ManageEngine
DDI Central

DDI Central User Guide

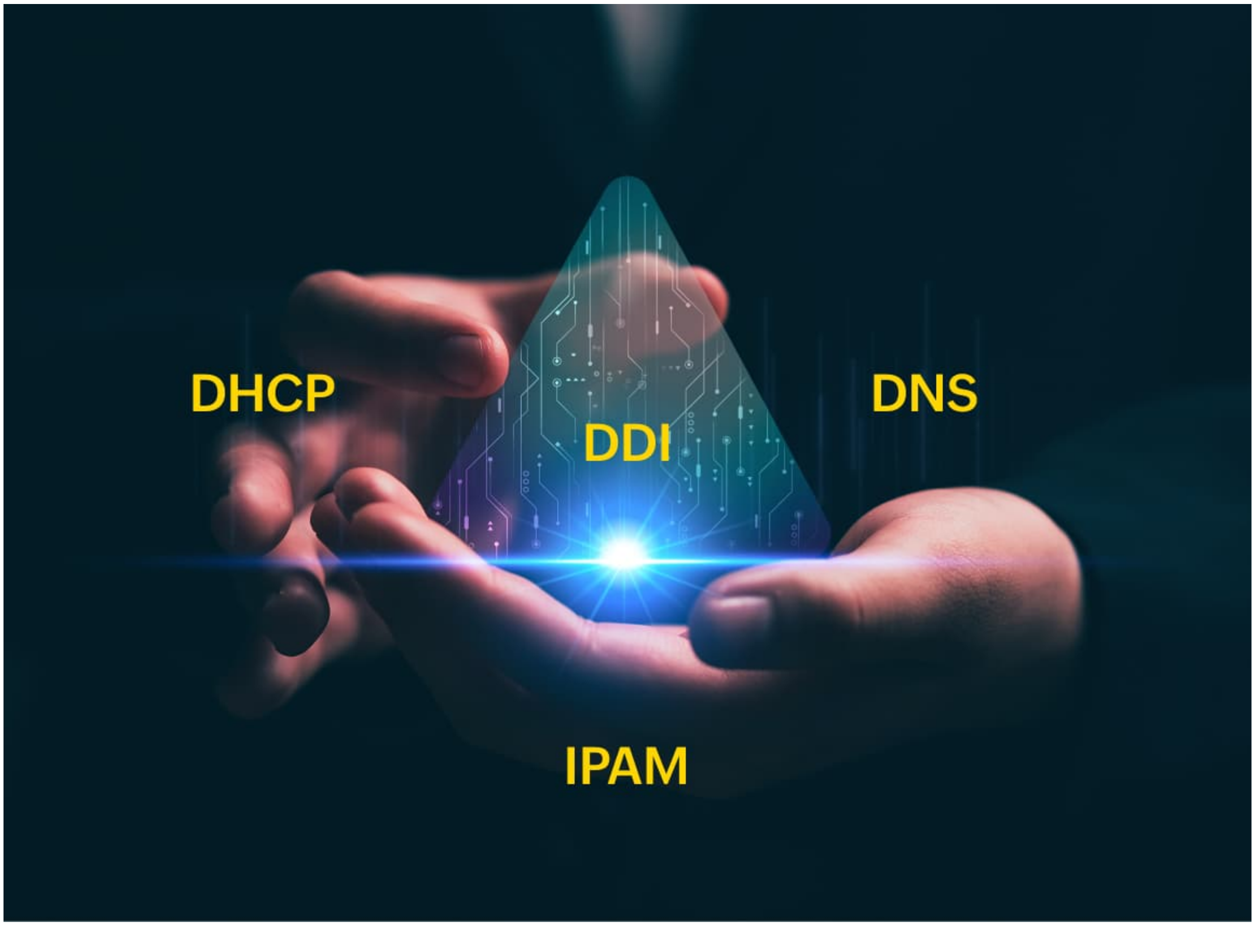


Table of Contents

About ManageEngine DDI Central

ManageEngine DDI Central is an comprehensive and easy to use DNS, DHCP and, IP Address Management (IPAM) application. It can be provisioned to manage both internal and external DNS and DHCP clusters.

ManageEngine DDI Central helps you to discover and manage existing installation of ISC-Bind9 and ISC-DHCP clusters. It can also be provisioned as new DNS and DHCP clusters to manage the infrastructure. ManageEngine DDI Central has DNS and DHCP bundled with the product and it gets deployed on your servers while installing the product.

DDI Central Application Architecture

Table of Contents

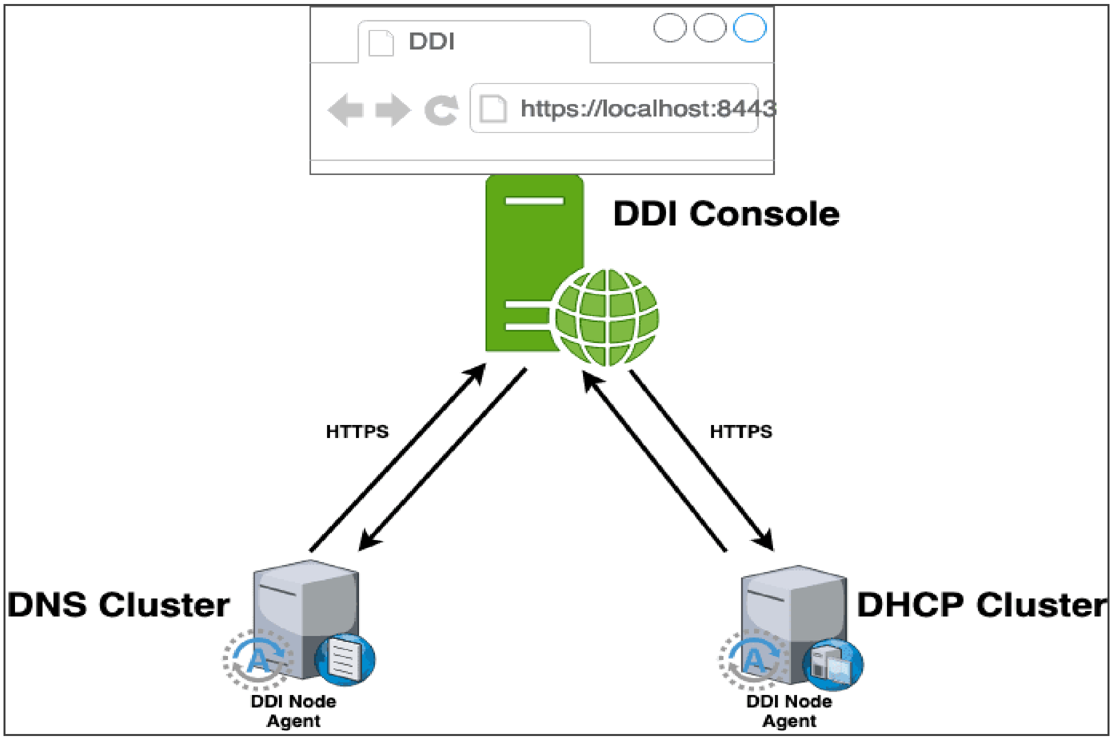
ManageEngine DDI Central has two components DDI Console and DDI Node Agent to be downloaded.

DDI Console

DDI Console provides centralized User Interface to manage all your DNS, DHCP clusters and also helps to manage your IP address inventory.

DDI Node Agent

DDI Node Agent must be deployed on all your DNS and DHCP servers to ensure continuous visibility of your DNS and DHCP clusters. It is a small light-weight agent which communicates with the DDI Console and helps to discover the existing configuration and provision new configuration changes from the DDI Console.



System requirements

Table of Contents

Hardware requirements for ManageEngine DDI Central Console - main server

ManageEngine DDI Central Console server hardware requirements for discovering and managing existing installation of ISC-DNS and ISC-DHCP is as follows:

Parameter	**Essential/Professional**
Processor	2.4 GHz Quad Core
RAM	32 GB
HDD	500 GB
Operating System	64 bit

Hardware requirements for ManageEngine DDI Central Node Agent server

ManageEngine DDI Central Node Agent is an small light-weight agent that has to be installed on all your existing ISC-DNS and ISC-DHCP clusters.

Parameter	**Essential/Professional**
Processor	2.4 GHz Quad Core
RAM	24 GB
HDD	500 GB
Operating System	64 bit

Software Requirements

Software	Minimum versions required for Evaluation	Version requirements for Production
Linux OS	Ubuntu 14 to 20.04/ CentOS 7/ Fedora 31/ Red Hat 7 to 9.1/ Debian	Ubuntu 14 to 20.04/ Red Hat version 7 to 9.1/ CentOS Stream 8/ CentOS 7/ Debian 10 and above
Browsers	Chrome/ Firefox/ Edge	Chrome (preferred)/ Firefox/ Edge

Software libraries required

1. Open SSL for secured network communication
2. Sudoers file for user authorizations

Port Requirements

Port Name	Default Port Numbers	Protocol	Usage	Inbound
Web server port	9090	TCP	This is the port on which you will connect to the DDI Central Console server from a web browser. You can change this at any time from the Settings tab.	Inboud
Embedded database port	33306	TCP	This is the default port used to connect to the PostgreSQL	N/A

			database in DDI	
HTTPS Port	9443	TCP	This is port through which DDI Central node Agent communicates with the console and also used to connect web console	Bidirectional

Database Requirements

PostgresSQL Comes bundled with the product.

Note: In case of failover, please use PostgresSQL that has replication configured.

Network Port Configurations

To ensure the optimal performance and seamless operation of the DDI Central solution, the following network port configurations are required:

1. **DNS (Domain Name System)**
    - **TCP and UDP Port 53**: Must be open for DNS query and response traffic. DNS uses this port for both TCP and UDP protocols to handle standard query and response action, as well as zone transfers (TCP).

2. **DHCP (Dynamic Host Configuration Protocol)**
    - **UDP Port 67 (Primary DHCP):** Must be open for receiving DHCP discovery requests from DHCP clients. This is the standard port on which the DHCP server listens for and responds to DHCP discovery messages and other requests from clients.
    - **TCP Port 647 (DHCP Failover):** Must be open to enable communication between primary and secondary DHCP servers in a failover configuration. This port is used for synchronization of DHCP lease

information and other configuration data to ensure continuity and consistency of DHCP services in case one of the servers becomes unavailable.

These port configurations are essential to allow the DDI Central system to communicate effectively within your network infrastructure. Ensure that any firewalls or network security systems are configured to permit traffic on these ports for the DDI Central solution's components.

Quick Installation Guide

Table of Contents

Note:

ManageEngine DDI is available only for **Linux platforms.** ManageEngine DDI can be deployed as an overlay for your existing Linux DNS and DHCP environment that supports Internet Systems Consortium : ISC DHCP and ISC BIND9 DNS.

Installing DDI Console

1. Download DDI Console for Linux
2. Assign **execute** permission using the command: chmod a+x ManageEngine_DDI_Console_xxxx.bin where ManageEngine_DDI_Console_xxxx is the name of the downloaded BIN file.
3. Execute the following command as **'root'.**

   ./ManageEngine_DDI_Console_xxxx.bin -i console
4. Follow the instructions as they appear on the screen to successfully install DDI Console on to your machine.

Installing DDI Node Agent

1. Download DDI Node Agent for Linux
2. Assign **execute** permission using the command: chmod a+x ManageEngine_DDI_Agent_xxxx.bin where ManageEngine_DDI_Agent_xxxx is the name of the downloaded BIN file.
3. Execute the following command as **'root'.**

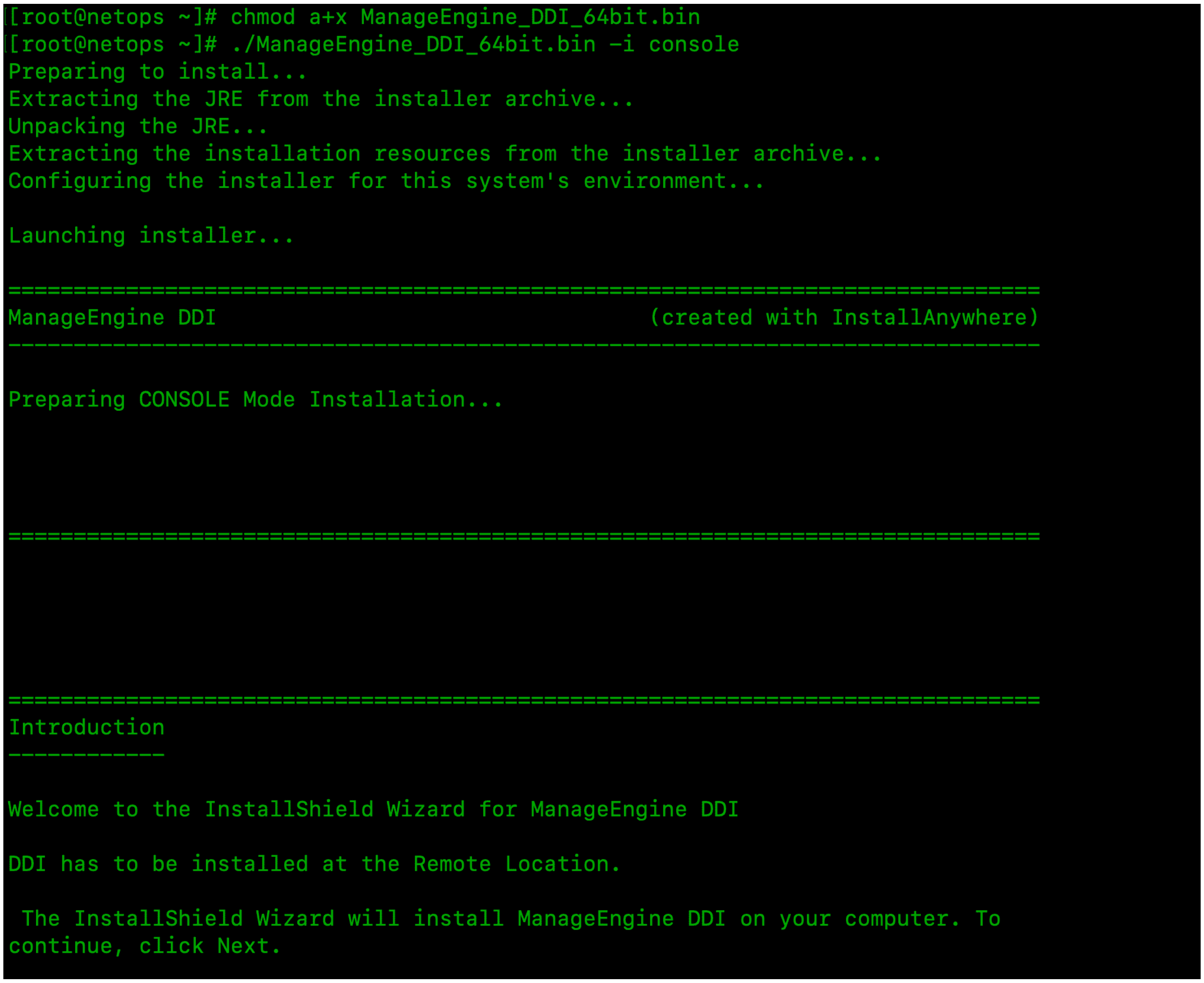
   ./ManageEngine_DDI_Agent_xxxx.bin -i console

Follow the instructions as they appear on the screen to successfully install DDINode Agent on to your machine.

DDI console mode installation

This is a quick walk-through of the console mode installation of DDI on a Linux box - an easy thing to do if you are working on a Windows box and want to install on a remote Linux system.

Step 1: Execute the binary with administrator privileges (sudo) and **-i console** option.

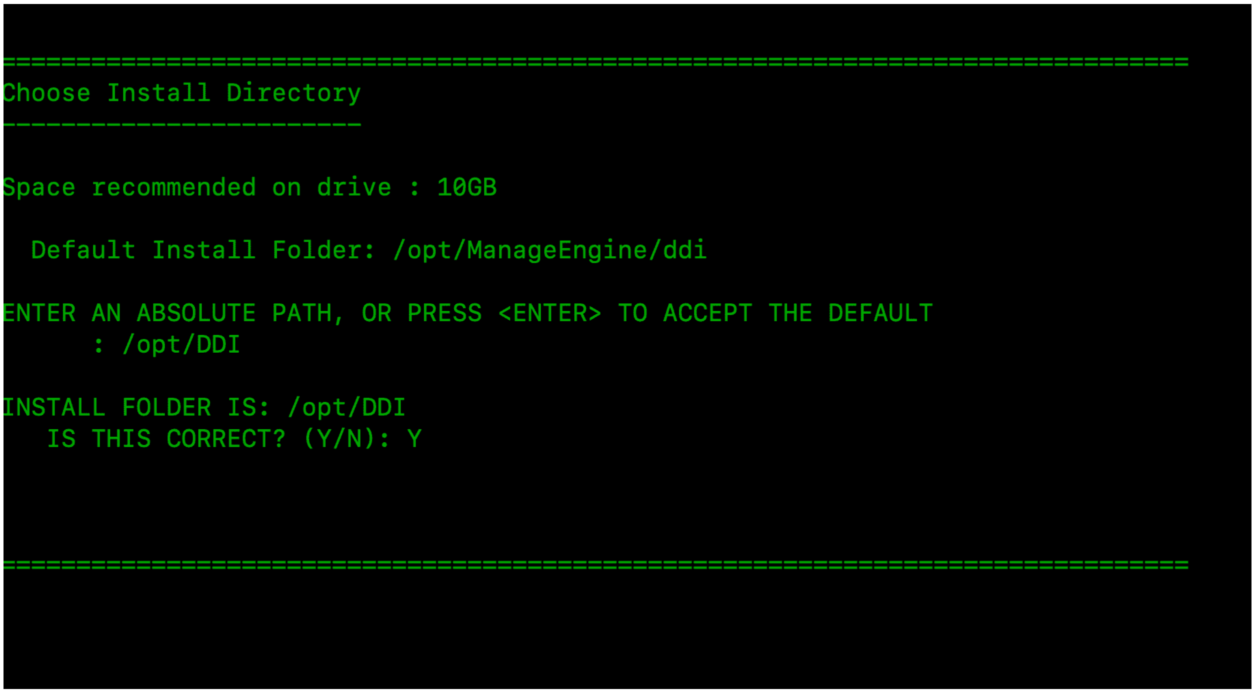


```
[root@netops ~]# chmod a+x ManageEngine_DDI_64bit.bin
[root@netops ~]# ./ManageEngine_DDI_64bit.bin -i console
Preparing to install...
Extracting the JRE from the installer archive...
Unpacking the JRE...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...

Launching installer...

===============================================================================
ManageEngine DDI                                 (created with InstallAnywhere)
-------------------------------------------------------------------------------

Preparing CONSOLE Mode Installation...




===============================================================================





===============================================================================
Introduction
------------

Welcome to the InstallShield Wizard for ManageEngine DDI

DDI has to be installed at the Remote Location.

 The InstallShield Wizard will install ManageEngine DDI on your computer. To
continue, click Next.
```

Step 2: Go through the license agreement and enter 'Y' to proceed. You can register for technical support by providing the required details. (Name, E-mail ID, Phone, Company Name)

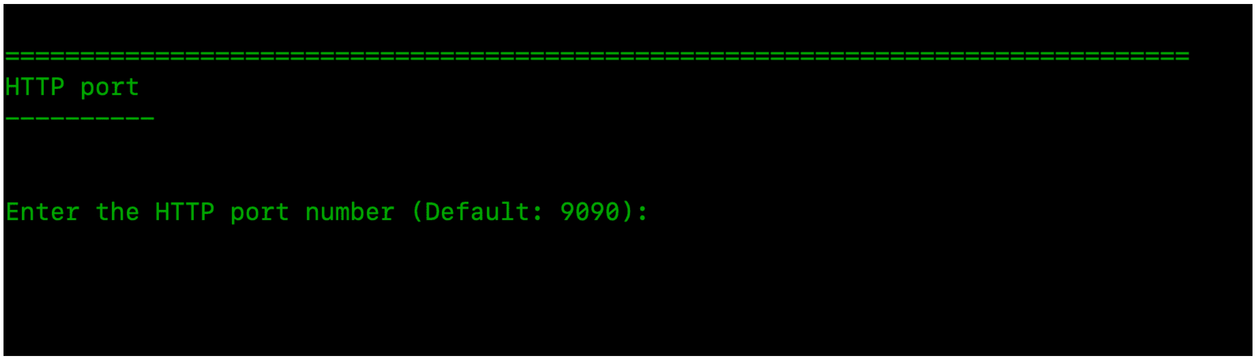
Step 3: Select the location

Step 4: Choose the installation directory



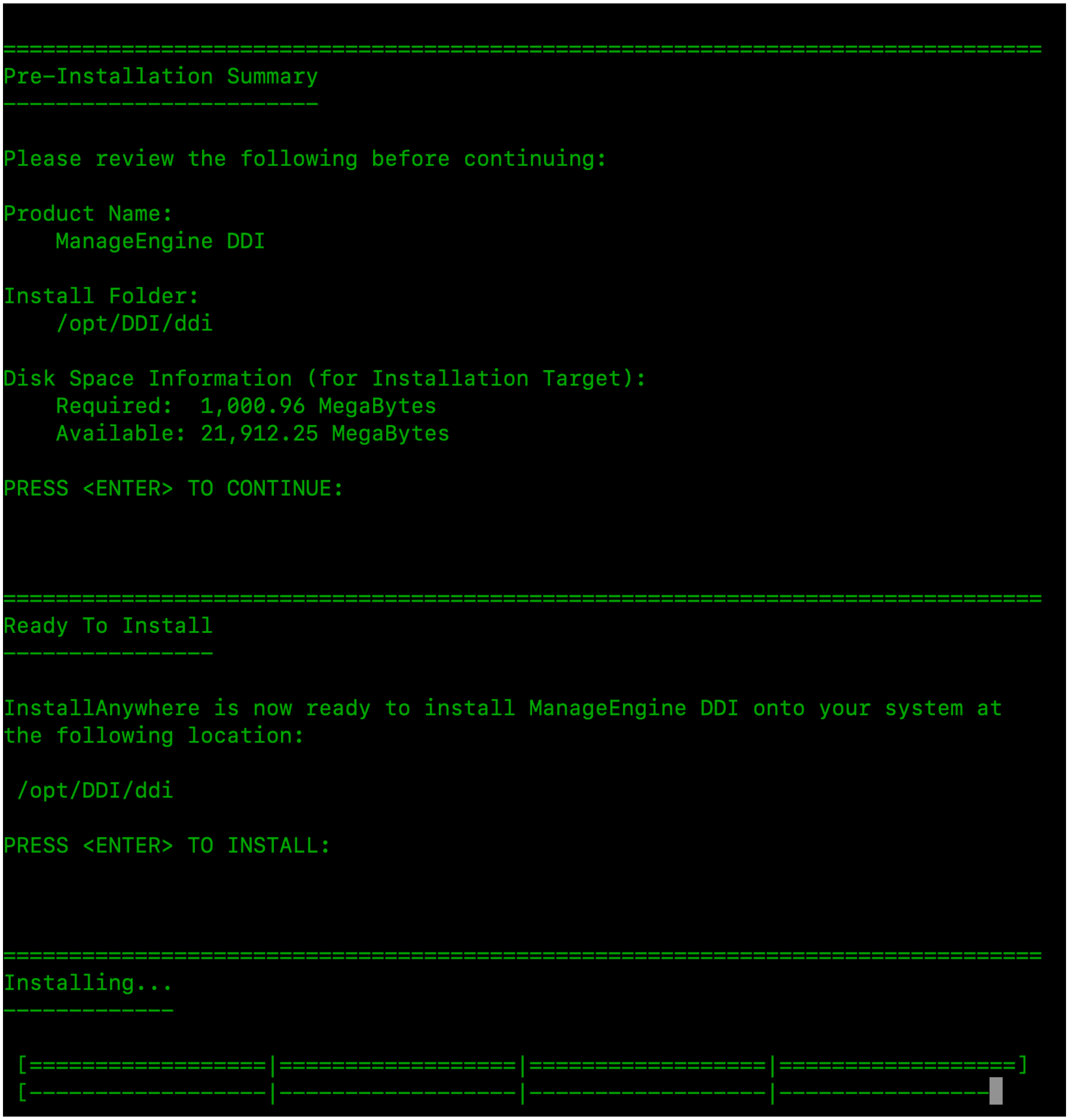
```
===============================================================================
Choose Install Directory
------------------------

Space recommended on drive : 10GB

  Default Install Folder: /opt/ManageEngine/ddi

ENTER AN ABSOLUTE PATH, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
      : /opt/DDI

INSTALL FOLDER IS: /opt/DDI
   IS THIS CORRECT? (Y/N): Y



===============================================================================
```

Step 5: Configure the Webserver and Listener Ports.



```
================================================================================
HTTP port
----------


Enter the HTTP port number (Default: 9090):
```

Step 6: Verify the installation details and press 'Enter' to complete the installation.



```
===============================================================================
Pre-Installation Summary
------------------------

Please review the following before continuing:

Product Name:
    ManageEngine DDI

Install Folder:
    /opt/DDI/ddi

Disk Space Information (for Installation Target):
    Required:  1,000.96 MegaBytes
    Available: 21,912.25 MegaBytes

PRESS <ENTER> TO CONTINUE:



===============================================================================
Ready To Install
----------------

InstallAnywhere is now ready to install ManageEngine DDI onto your system at
the following location:

 /opt/DDI/ddi

PRESS <ENTER> TO INSTALL:



===============================================================================
Installing...
-------------

 [==================|==================|==================|==================]
 [------------------|------------------|------------------|-----------------
```

i.e

During installation if you get an error message stating that the temp folder does not have enough space, try executing this command with the -is:tempdiroption, where is the absolute path of an existing directory.

./ManageEngine_DDI_xxxx.bin -is:tempdir

For non-x11 machines, use the following command:

./ManageEngine_DDIr_xxxx.bin -i console

Starting DDI Central

Once the ManageEngine DDI Central installation is completed, you can start the service by executing following command on your linux box

systemctl start DDI

To check whether service is started, execute the following command:

systemctl status DDI

After ensuring service is started successfully, you can connect the web client using both http and https as follows http://server_ip:9090

https://server_ip:9443

Accessing the Web Client

Once the service has successfully started, follow the steps below to access DDI Central.

1. Open a supported web browser window
2. Type the URL address as **http://**DDI_ConsoleServer_IP**:9090**(where **DDI Console Server** is the name of the machine on which **DDI Console Server** is running, and **9090** is the default web server port)
3. Log in to DDI using the default username/password combination of **admin/admin**

Licensing Information

- **Essential Edition**

Manage 1 DNS Cluster(3 servers max), 1 DHCP Cluster(2 Servers max), 5 DNS Zones, 5 IP Subnet, 1 User, and IPAM. This is entry level edition with basic DNS, DHCP and IP Address Management. Advanced features are available in Professional edition.

- **Professional Edition**

  Manage 1 DNS Cluster(3 servers max), 1 DHCP Cluster(2Servers max), 15 DNS Zones, 15 IP Subnet, 2 Users, and IPAM.Advanced feature like Domain Blocking, DHCP Fingerprinting, Audit, Query Analytics, etc.

License components in DDI Central

Clusters in ManageEngine DDI Central

Clusters are logical groupings of servers - DNS, DHCP or both organized for identification and administrative purposes. These clusters operate independently of other clusters configured within DDI Central. Each cluster you add has its own internalized IP address plans, IP inventory, IP Address Manager, DNS manager and DHCP manager.

A single cluster can accommodate any number of DNS servers and DHCP servers.

However, ManageEngine DDI enables you to add only one Primary DHCP server for each cluster and the rest of DHCP servers act secondary servers configured under DHCP failover configurations.

Essential Plan:

The Essential plan includes **one DNS cluster and one DHCP cluster**. With the single DNS cluster, you can add **up to three DNS servers** into the DDI Central console. Similarly, the single DHCP cluster allows for the addition of **up to two DHCP servers**.

These servers can be grouped together within a single cluster or distributed across multiple clusters in any combination as per your preference. There is **no restriction on the number of clusters** you can create.

Professional Plan:

The Professional plan provides **two DNS clusters and two DHCP clusters**. This way you can **add up to six DNS servers and four DHCP servers in total**. You have the flexibility to organize these servers by grouping them into a single cluster or by creating multiple clusters. You can configure the clusters in any combination that suits your needs, with no limit on the number of clusters.

You can also purchase additional DNS clusters, DHCP clusters, subnets, zones, and users as add-ons. Please refer the ManageEngine Store page to check the prices of DDI Central add-ons.

Once installed, DDI Central runs in evaluation mode for 30 days. You can obtain a registered license for DDI Central at any time during the evaluation period by contacting DDI Central Support. If you have not upgraded to the Licensed Edition by the end of the evaluation period, DDI Central evaluation license expires and you will have only read-only access.

Upgrading your license

After obtaining the new license from ZOHO Corp, save it on your computer, and follow the steps below to upgrade your DDI installation:

1. Log in to the DDI Central web client

2. Click **Admin** logo/icon in the top right corner of the web client.
3. Click the **Register** link present in that pop-up page.
4. In the License window that opens up, browse for the new license file and select it.
5. Click **Register** to apply for the new license file

Note : The new license is applied with immediate effect. You do not have to shut down or restart the DDI Central server after the license is applied.

Installing DDI Central using custom ISO image

Download the Debian ISO image of the custom Linux distribution that will help you run ManageEngine DDI Central in any environment like Windows using your VM software.

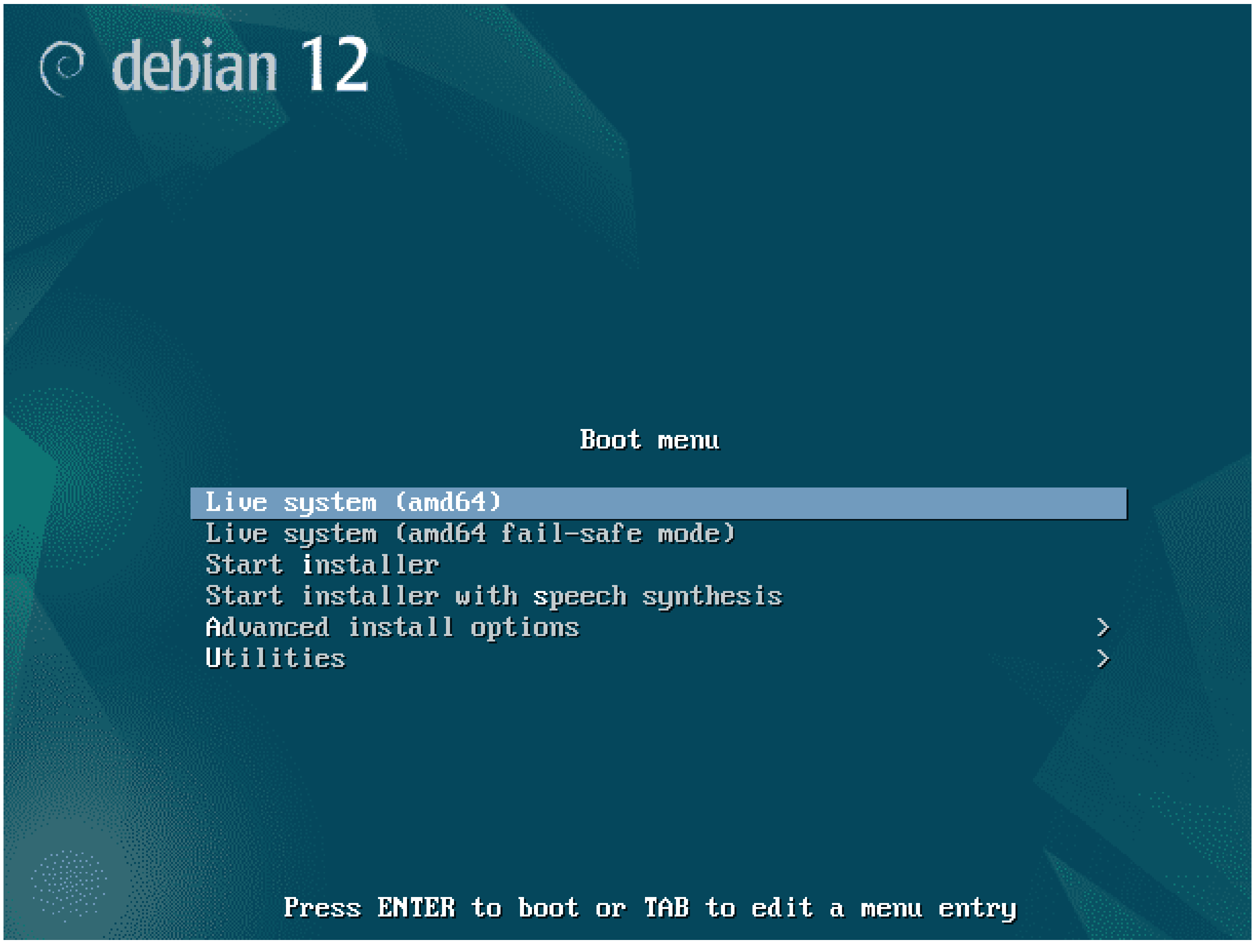
Prerequisites

Virtual machine software like: VirtualBox, VMware, Hyper V, and KVM

Installing the DDI Console

1. Start the Virtual Machine with the Debian ISO image

- Boot up your virtual machine software (like VirtualBox or VMware).
- Start the VM configured with the Debian ISO image.
- You will be greeted with a boot menu similar to the image below. Choose **Live System (amd64)** to begin the Debian installation process. This live ISO image allows you to boot into a fully functional Debian environment without installing it on your hard drive.


debian 12
Boot menu
Live system (amd64)
Live system (amd64 fail-safe mode)
Start installer
Start installer with speech synthesis
Advanced install options >
Utilities >

Press ENTER to boot or TAB to edit a menu entry

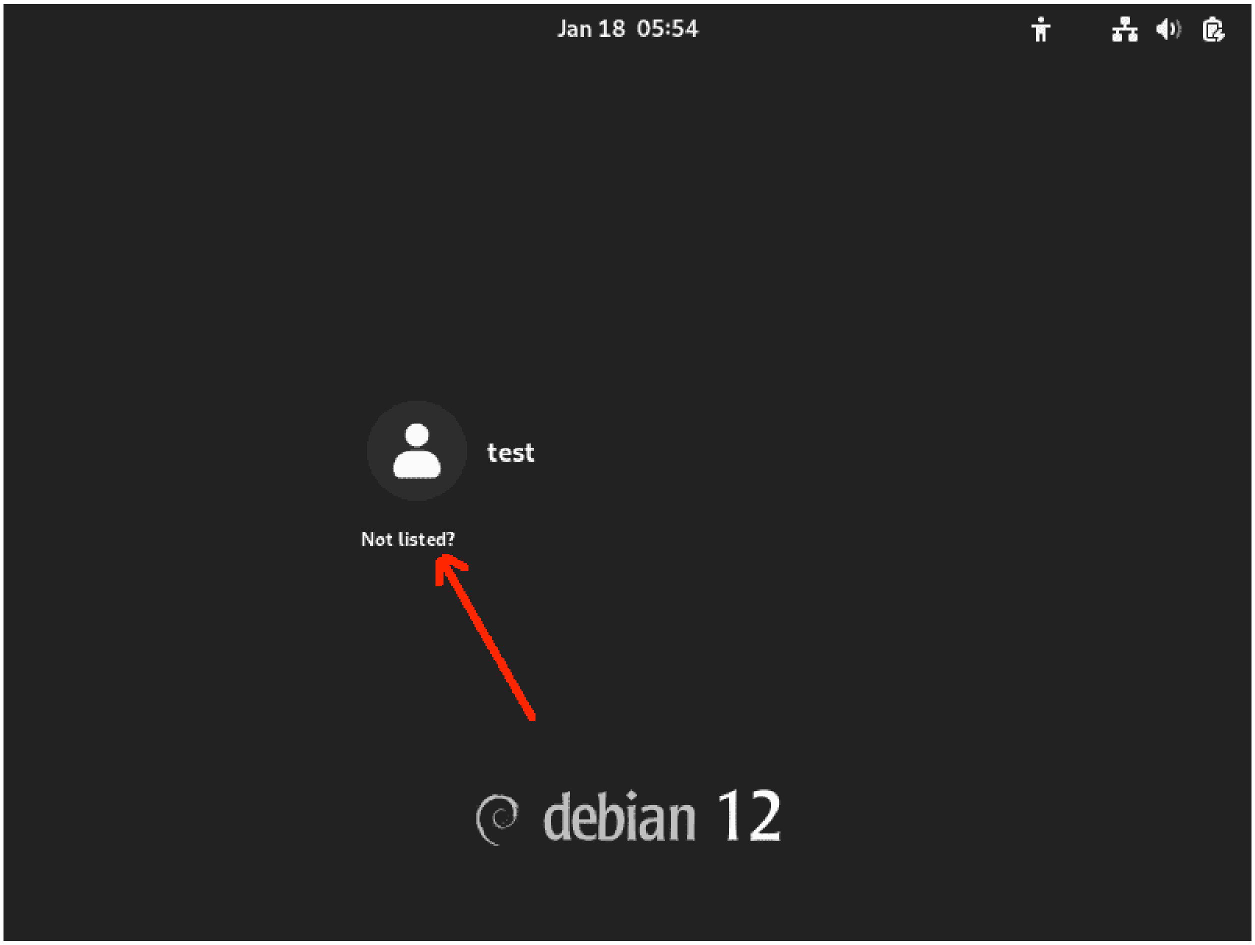
2. Explore the live Debian environment

You will reach a desktop environment, where you will see an unlisted desktop account. Click on it.

This will take you to a different screen that will ask you for the Username. Enter the username as **root**.

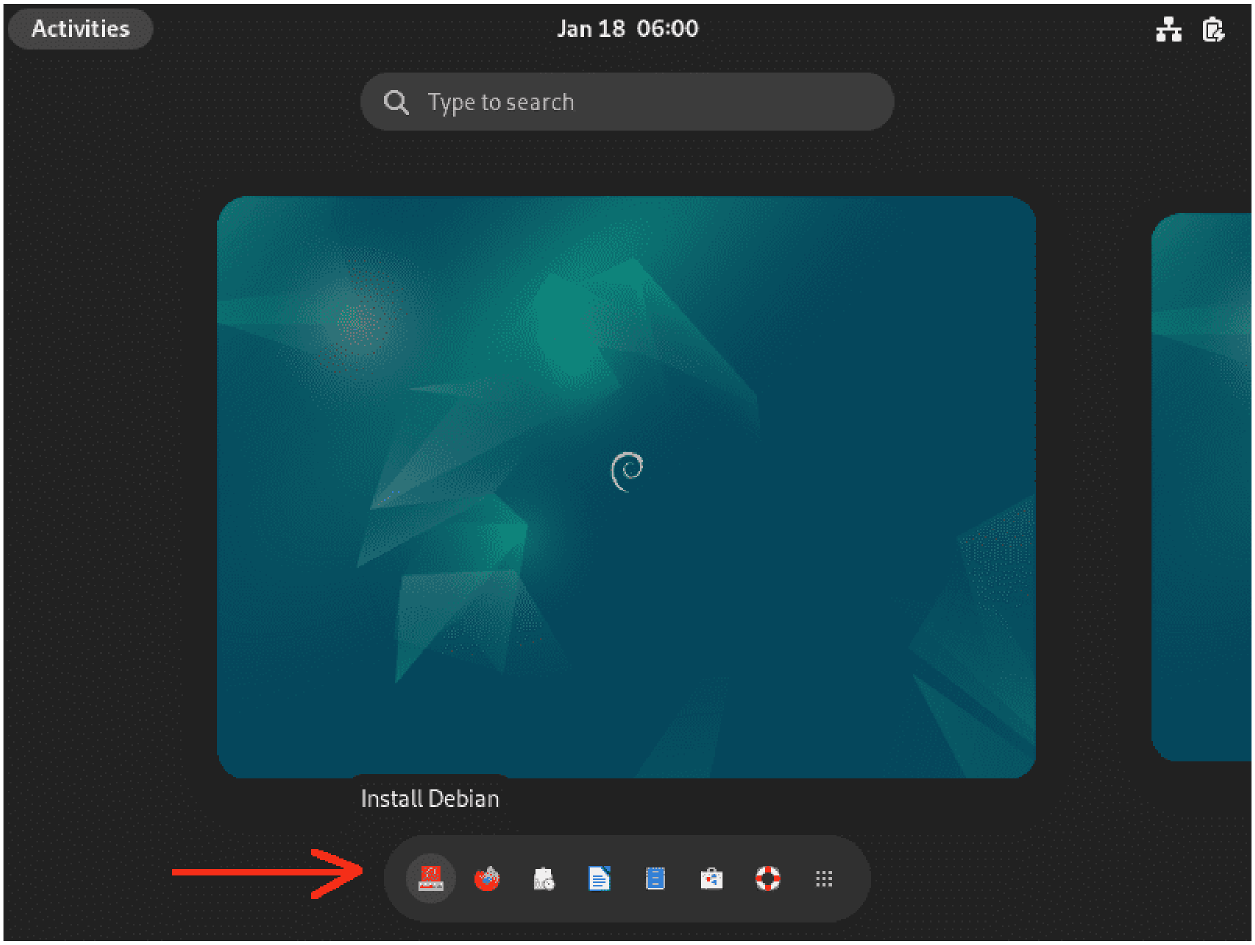
On entering the username, you'll be prompted to enter the password. Now, enter the password provided by the DDI support team via a dedicated email sent for downloading ISO image.

Note: The password sent via email is only temporary. You should reset it in the steps to follow as you proceed.



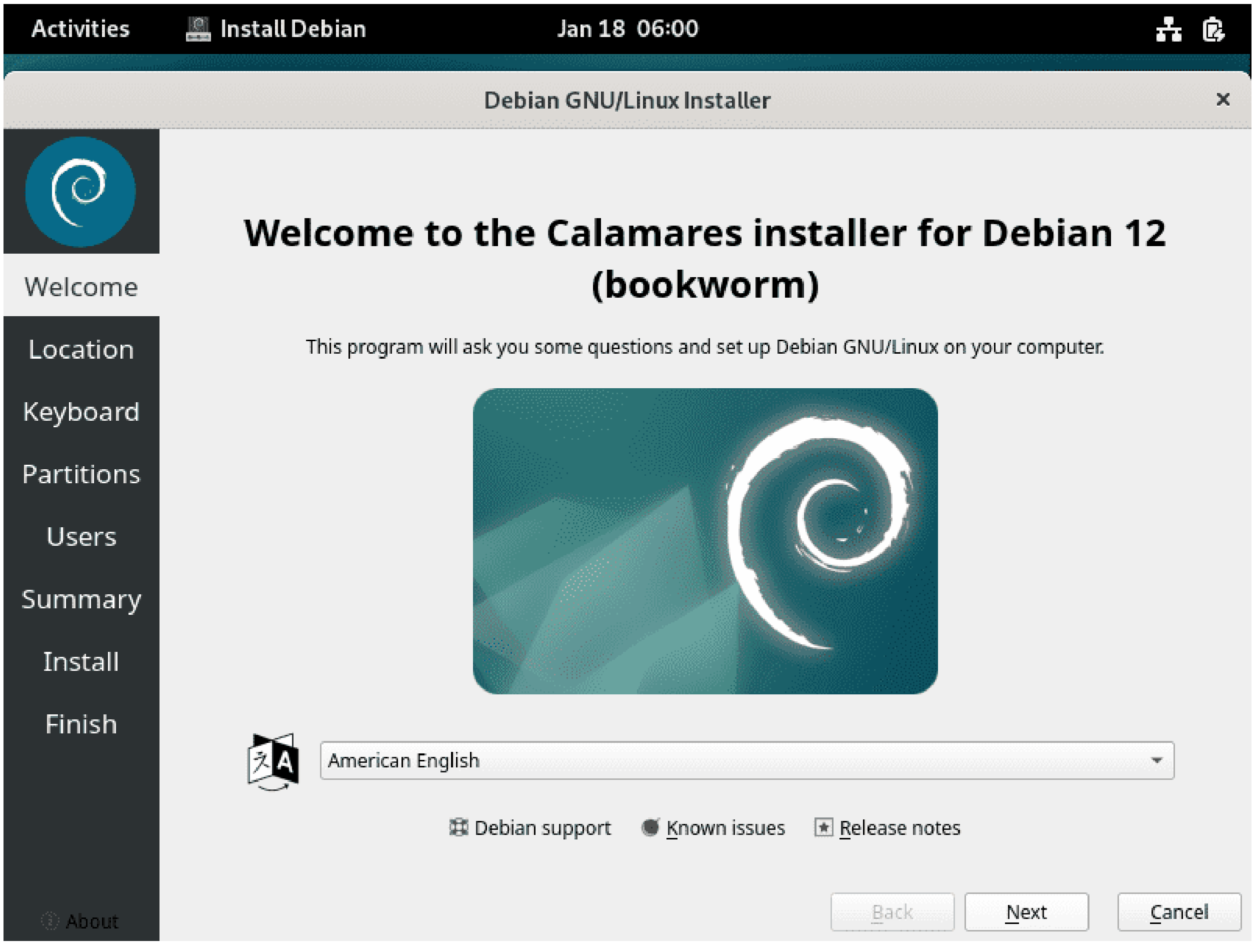
3. Start the installer from the live environment

On successful login, you can install Debian onto the virtual machine's hard drive by locating and double-clicking the **Install Debian** icon on the desktop, as shown in the image below.



4. Welcome screen of the installer

Now, the welcome screen of the **Calamares** installer appears. Here, choose the installation language as **American English** and proceed by clicking **Next.**



Debian GNU/Linux Installer

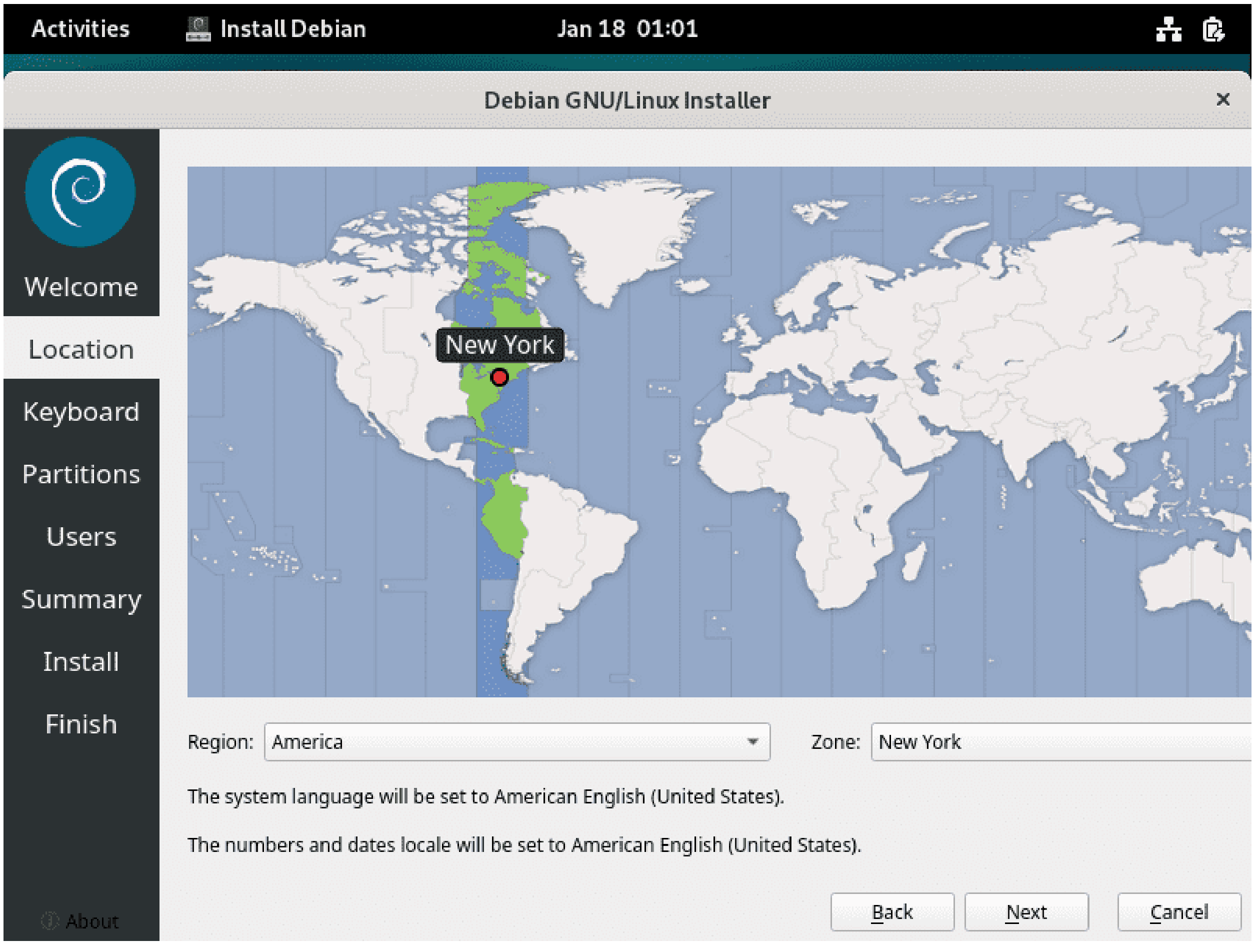
Welcome to the Calamares installer for Debian 12 (bookworm)

This program will ask you some questions and set up Debian GNU/Linux on your computer.

American English

5. Set up location and timezone

Select your location and timezone from the respective dropdown menus and click **Next**.



Region: America

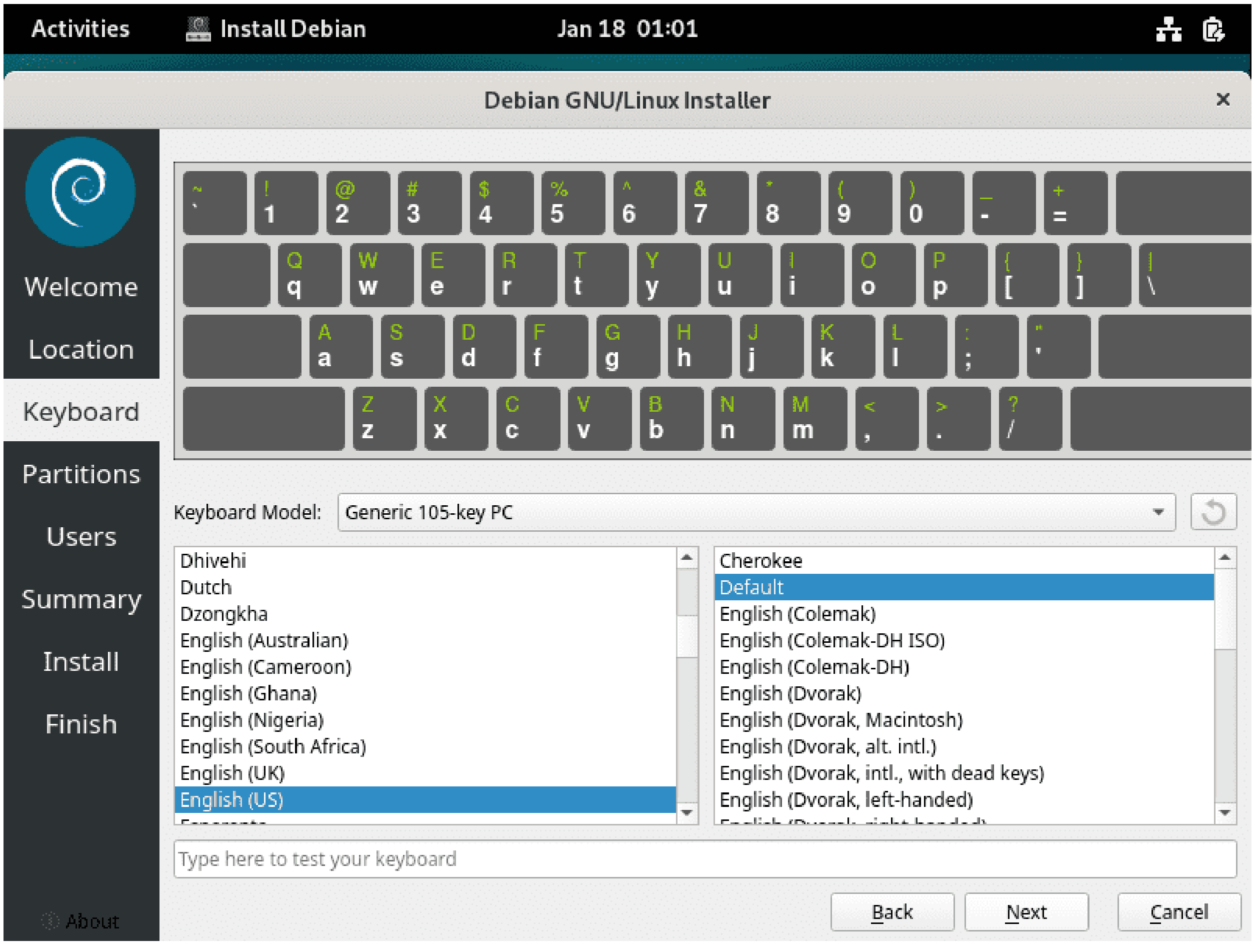
Zone: New York

The system language will be set to American English (United States).

The numbers and dates locale will be set to American English (United States).

6. Configure the keyboard

Choose your keyboard layout from the **Keyboard Model** dropdown menu. For an English (US) keyboard, you should select **English (US)**.



Keyboard Model: Generic 105-key PC

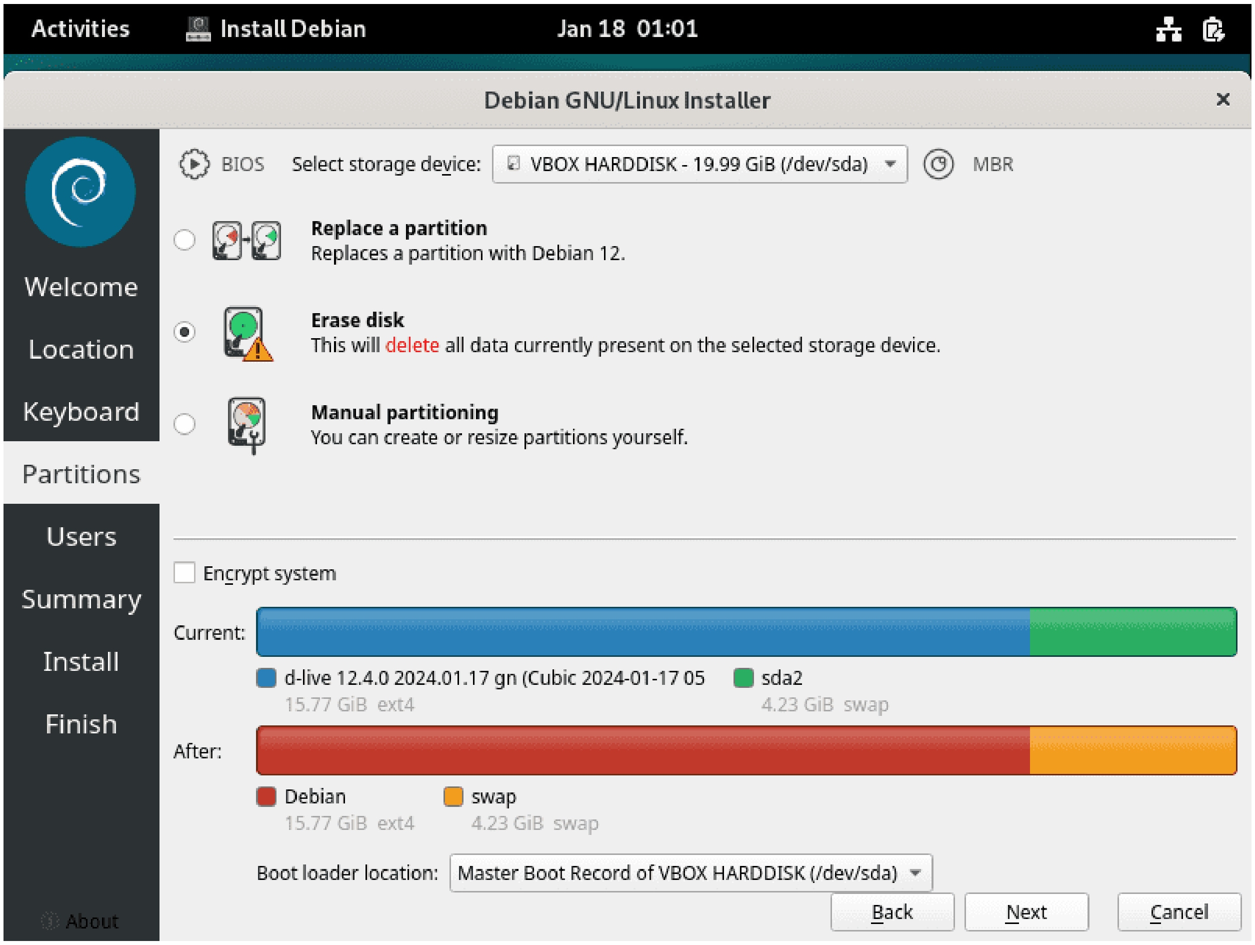
Dhivehi
Dutch
Dzongkha
English (Australian)
English (Cameroon)
English (Ghana)
English (Nigeria)
English (South Africa)
English (UK)
English (US)

Cherokee
Default
English (Colemak)
English (Colemak-DH ISO)
English (Colemak-DH)
English (Dvorak)
English (Dvorak, Macintosh)
English (Dvorak, alt. intl.)
English (Dvorak, intl., with dead keys)
English (Dvorak, left-handed)

Type here to test your keyboard

7. **Disk partitioning**

- On the partitioning screen, configure the partitioning of your disk. If you're okay with wiping the entire virtual disk, select **Erase disk**. This will delete all data and install Debian 12.
- Confirm that the bootloader is installed to the correct location, typically the Master Boot Record (MBR) of the VM's virtual disk.



Select storage device: VBOX HARDDISK - 19.99 GiB (/dev/sda)

○ Replace a partition
Replaces a partition with Debian 12.

● Erase disk
This will delete all data currently present on the selected storage device.

○ Manual partitioning
You can create or resize partitions yourself.

□ Encrypt system

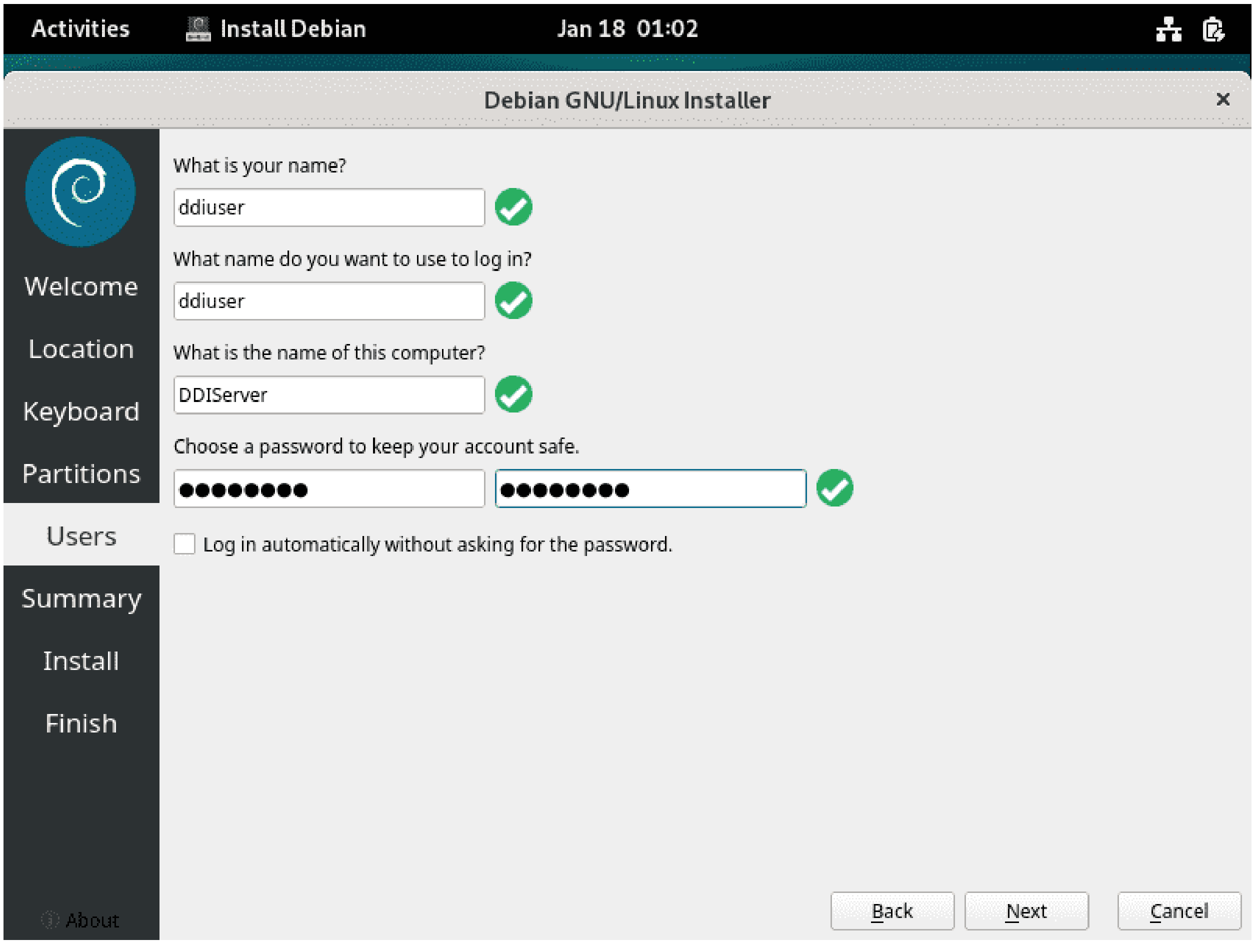
Current:
🟦 d-live 12.4.0 2024.01.17 gn (Cubic 2024-01-17 05 — 15.77 GiB ext4
🟩 sda2 — 4.23 GiB swap

After:
🟥 Debian — 15.77 GiB ext4
🟧 swap — 4.23 GiB swap

Boot loader location: Master Boot Record of VBOX HARDDISK (/dev/sda)

8. Create a User Account

Create a user account. Fill in the desired username and password, which will be used to log into Debian after the installation. Click **Next**.



What is your name?
ddiuser ✓

What name do you want to use to log in?
ddiuser ✓

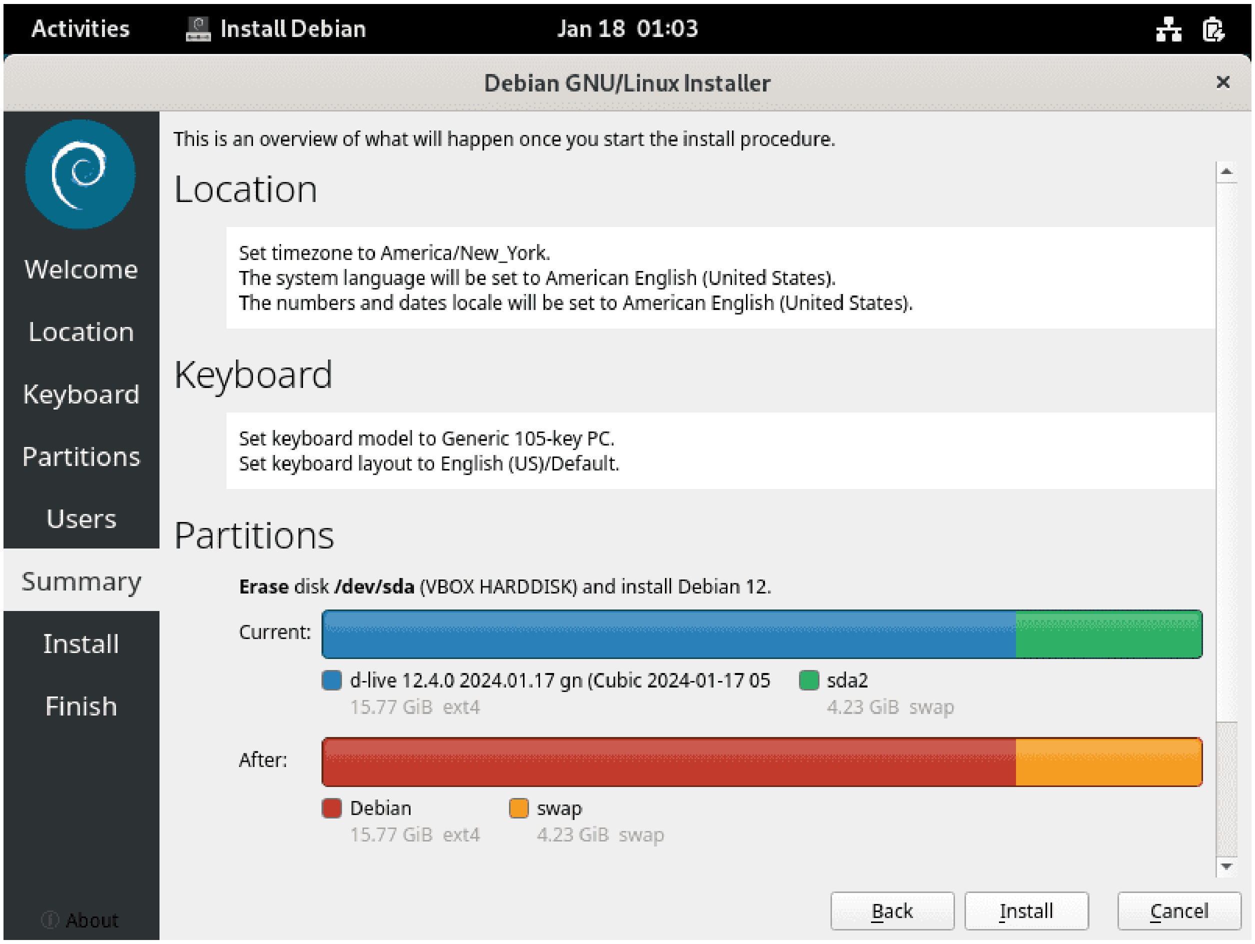
What is the name of this computer?
DDIServer ✓

Choose a password to keep your account safe.
●●●●●●●●● ●●●●●●●●● ✓

□ Log in automatically without asking for the password.

9. Review installation summary

The image below summarizes your choices. If all the details are correct, proceed by clicking **Install.**



This is an overview of what will happen once you start the install procedure.

Location

Set timezone to America/New_York.
The system language will be set to American English (United States).
The numbers and dates locale will be set to American English (United States).

Keyboard

Set keyboard model to Generic 105-key PC.
Set keyboard layout to English (US)/Default.

Partitions

Erase disk **/dev/sda** (VBOX HARDDISK) and install Debian 12.

Current:

🟦 d-live 12.4.0 2024.01.17 gn (Cubic 2024-01-17 05
15.77 GiB ext4

🟩 sda2
4.23 GiB swap

After:

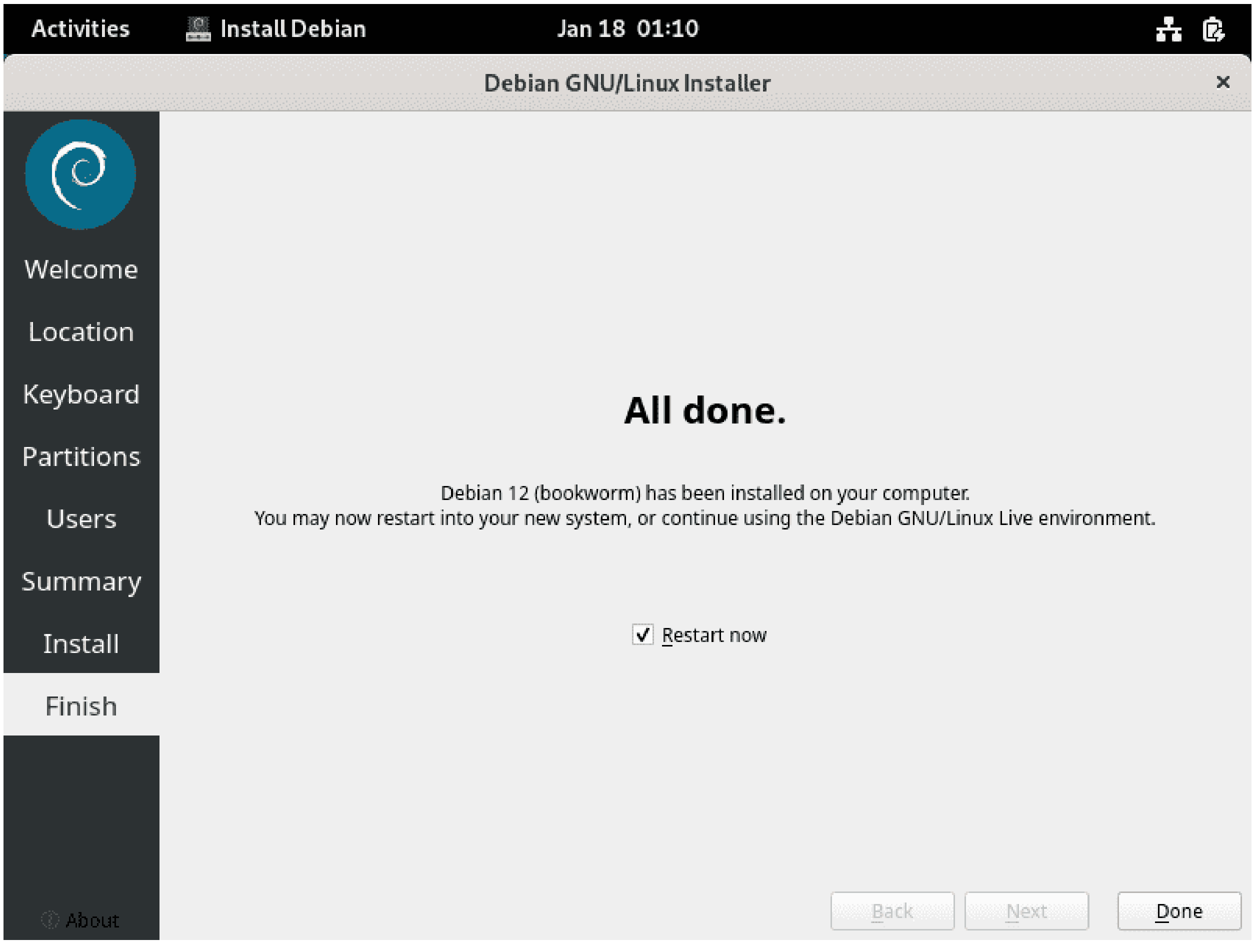
🟥 Debian
15.77 GiB ext4

🟧 swap
4.23 GiB swap

10. Completing the installation

As shown in the image below, the installation process will begin. The progress bar will indicate how far along the installation is. Wait for it to complete. After the installation process completes, you will usually be prompted to restart the VM.

Note: Upon reboot, ensure to remove the installation ISO from the VM's settings to avoid booting into the live environment again.



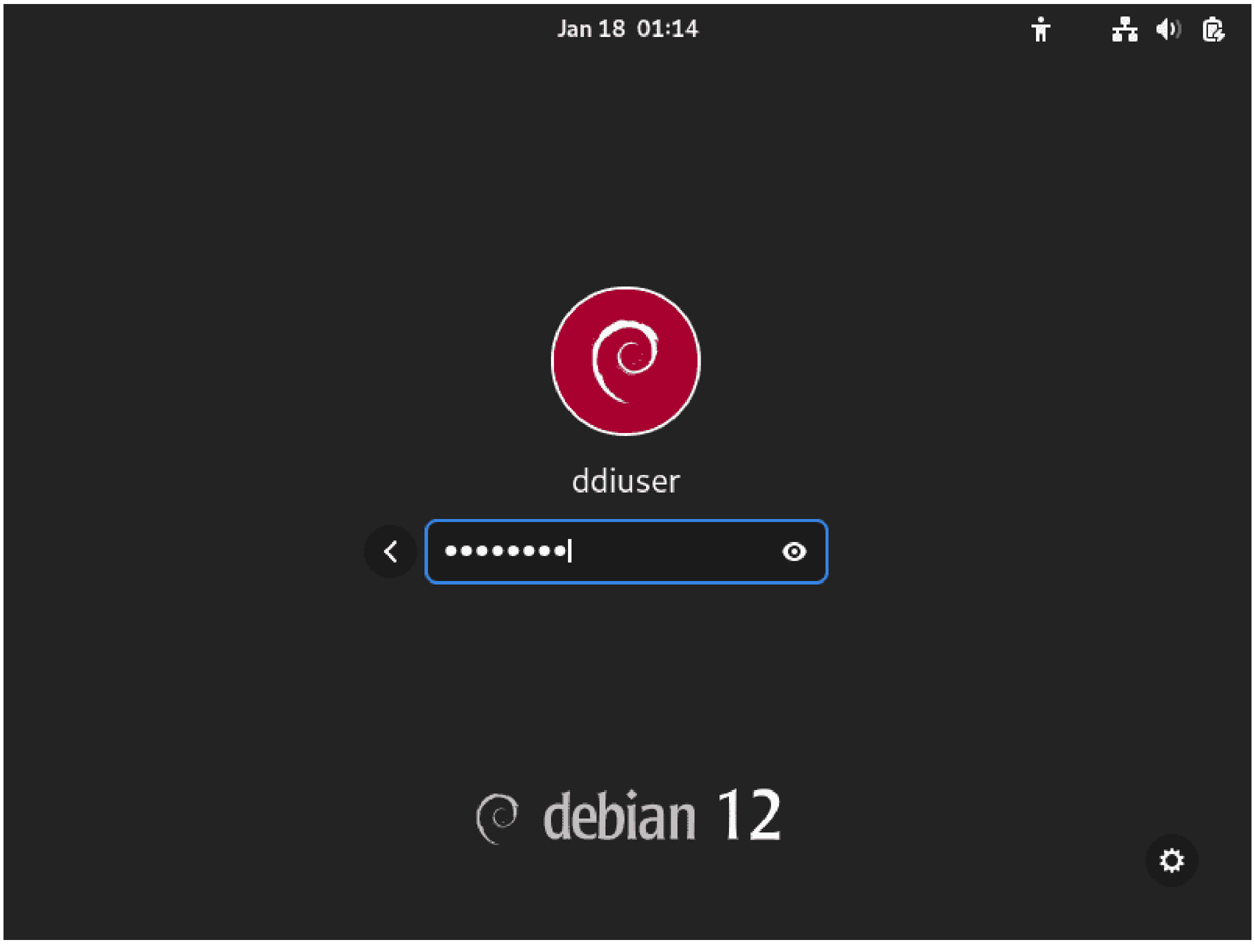
All done.

Debian 12 (bookworm) has been installed on your computer.
You may now restart into your new system, or continue using the Debian GNU/Linux Live environment.

☑ Restart now

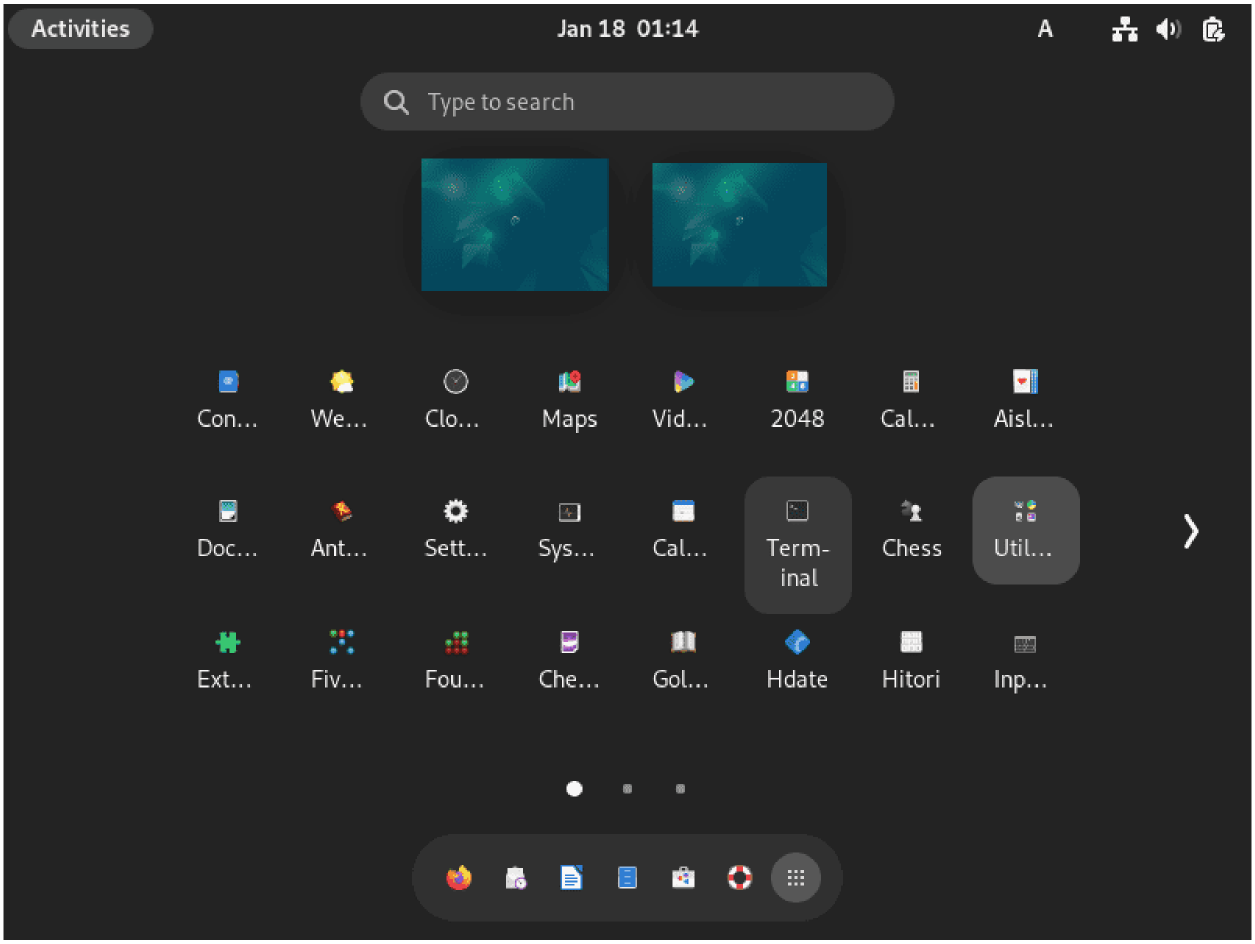
11. Post-installation steps

Once Debian is installed, you can login to your new user account with the username and password from **Step 8.**



12. Installing and configuring your application

After successfully installing Debian, you can now set up ManageEngine DDI Central on your system. Open the terminal window.



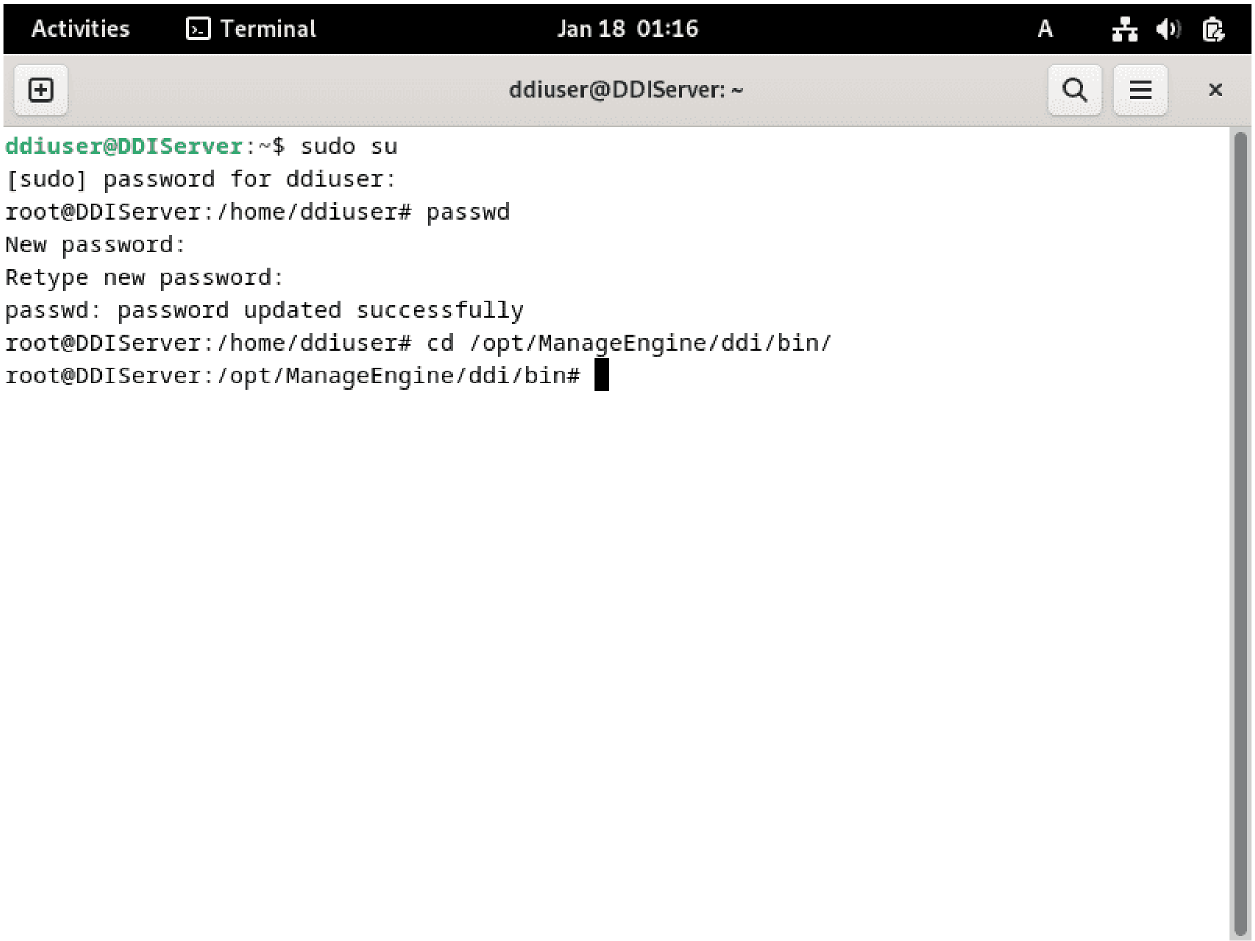
13. Reset the root password

On the terminal window, reset the temporary root password from **Step 2** following the steps below.

- Gain SuperUser access by entering the command sudo su to switch to the root user. You will be prompted to enter the password for your current user (the one with sudo privileges).
- Once you are logged in as root, reset the root password by typing the command passwd. You will then be prompted to enter a new root password.
- After entering passwd, you'll be asked to enter the new password twice to confirm.

14. Access the Application Installation Directory

- Navigate to the directory where your application's .bin files are located using the cd command in the terminal.
- cd /opt/ManageEngine/ddi/bin/



```
ddiuser@DDIServer:~$ sudo su
[sudo] password for ddiuser:
root@DDIServer:/home/ddiuser# passwd
New password:
Retype new password:
passwd: password updated successfully
root@DDIServer:/home/ddiuser# cd /opt/ManageEngine/ddi/bin/
root@DDIServer:/opt/ManageEngine/ddi/bin#
```

15. Change file permissions

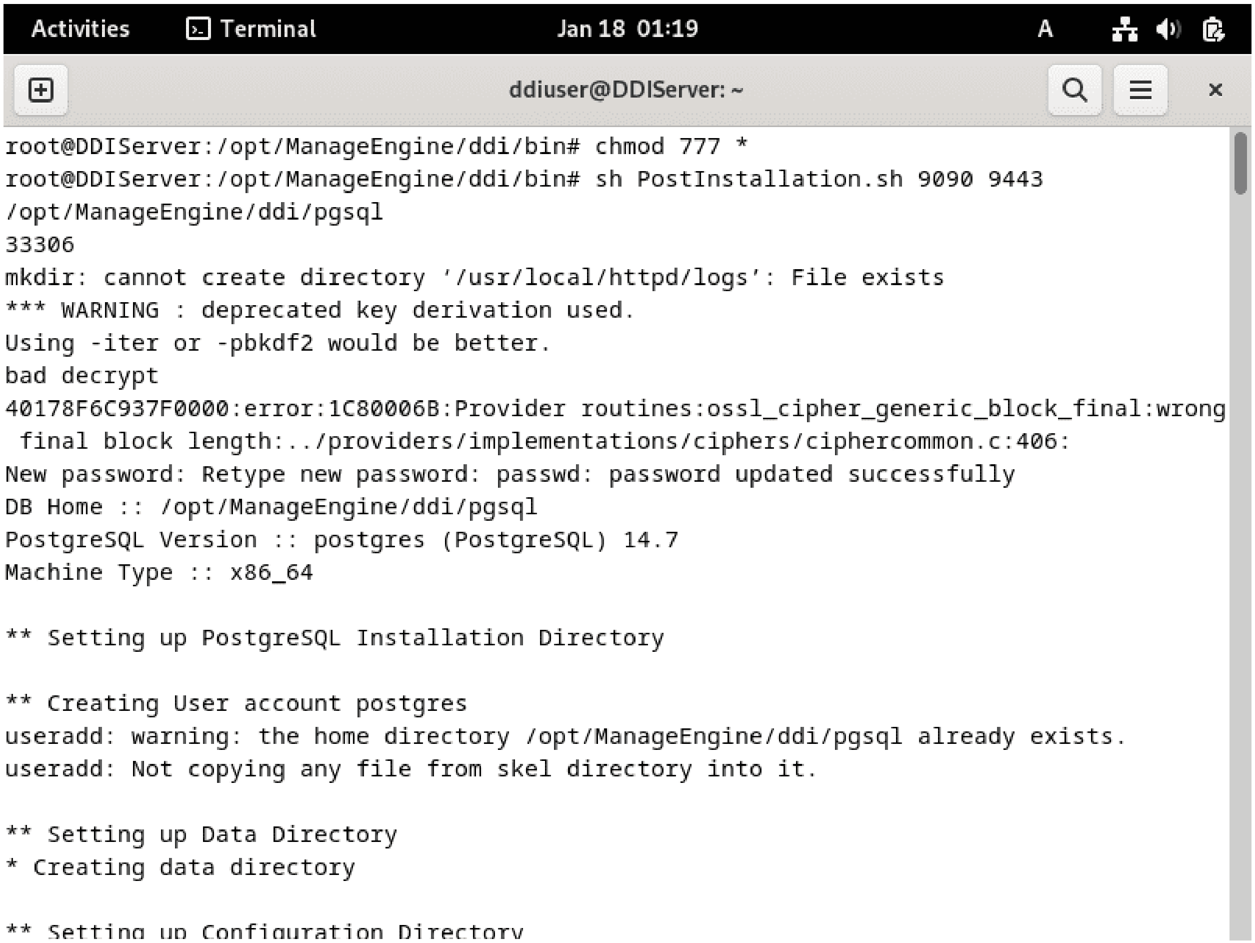
You might need to change the permissions of the scripts to make them executable.

This can be done using the chmod command.

Note: Use the chmod command as specified below, introducing a / (slash) anywhere within the chmod command would yield errors.

```
chmod 777 *
```

This command grants read, write, and execute permissions of all files in the current directory.



```
root@DDIServer:/opt/ManageEngine/ddi/bin# chmod 777 *
root@DDIServer:/opt/ManageEngine/ddi/bin# sh PostInstallation.sh 9090 9443
/opt/ManageEngine/ddi/pgsql
33306
mkdir: cannot create directory '/usr/local/httpd/logs': File exists
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
bad decrypt
40178F6C937F0000:error:1C80006B:Provider routines:ossl_cipher_generic_block_final:wrong
 final block length:../providers/implementations/ciphers/ciphercommon.c:406:
New password: Retype new password: passwd: password updated successfully
DB Home :: /opt/ManageEngine/ddi/pgsql
PostgreSQL Version :: postgres (PostgreSQL) 14.7
Machine Type :: x86_64

** Setting up PostgreSQL Installation Directory

** Creating User account postgres
useradd: warning: the home directory /opt/ManageEngine/ddi/pgsql already exists.
useradd: Not copying any file from skel directory into it.

** Setting up Data Directory
* Creating data directory

** Setting up Configuration Directory
```

16. Run the installation script

- Execute the command - sh PostInstallation.sh 9090 9443
- This executes a shell script named PostInstallation.sh with two arguments, 9090 and 9443, which are http and https port numbers that the script needs to

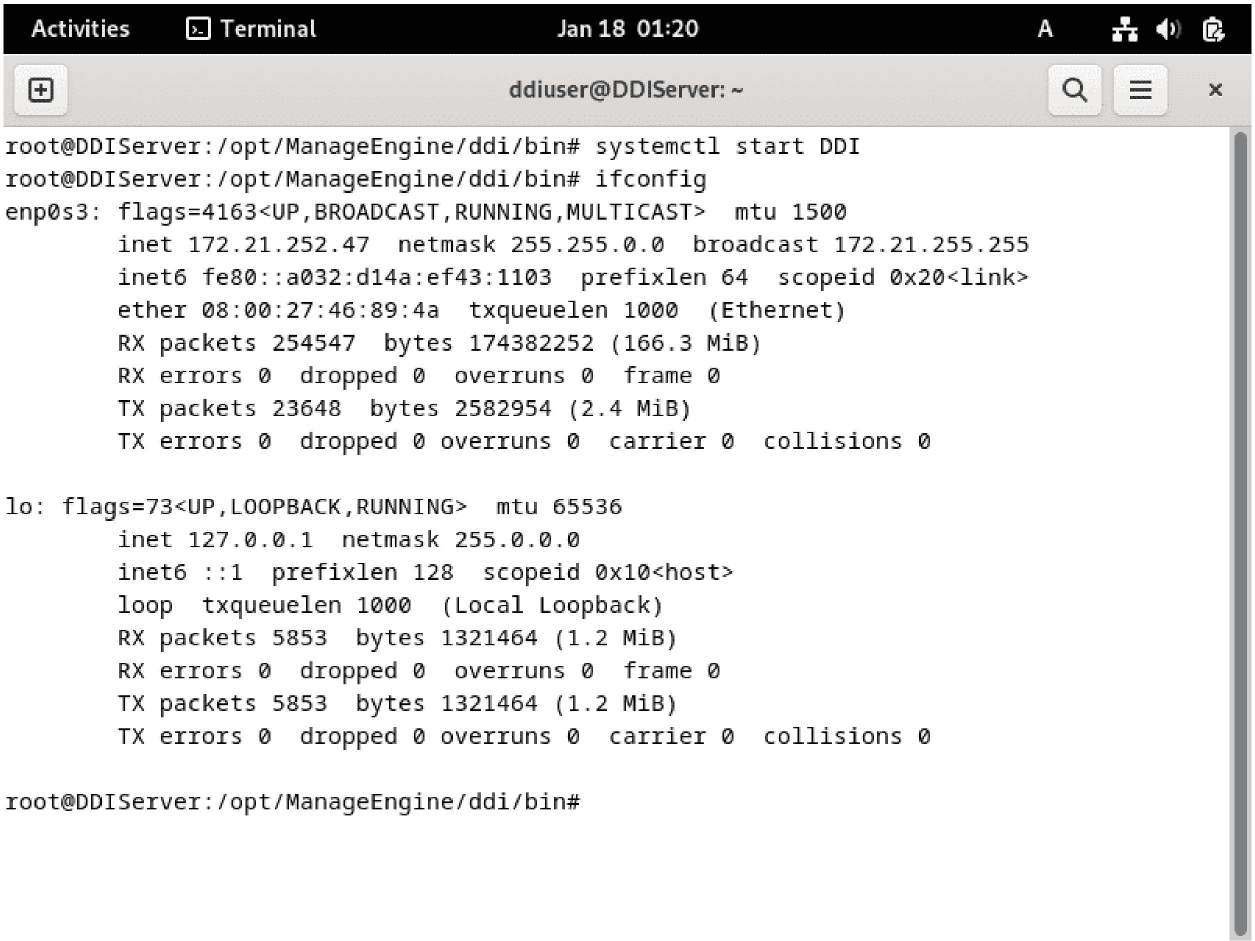
configure the application to.

- Wait till the script is run completely.

If you encounter errors, you can seek support from the ddi-support team.

17. Start the application services

Once the installation is complete and all configurations are set, start the application services. Use the command systemctl for this purpose: systemctl start DDI



```
root@DDIServer:/opt/ManageEngine/ddi/bin# systemctl start DDI
root@DDIServer:/opt/ManageEngine/ddi/bin# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.21.252.47  netmask 255.255.0.0  broadcast 172.21.255.255
        inet6 fe80::a032:d14a:ef43:1103  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:46:89:4a  txqueuelen 1000  (Ethernet)
        RX packets 254547  bytes 174382252 (166.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 23648  bytes 2582954 (2.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 5853  bytes 1321464 (1.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5853  bytes 1321464 (1.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@DDIServer:/opt/ManageEngine/ddi/bin#
```

18. Verify network configuration

- Use the ifconfig network command to view the configuration of network interfaces and IP addresses. From the response generated, the IP address near inet shows the IPv4 address assigned to the interface. This is the ip address that is used to run the DDI application using the web browser.

- The netmask indicates the network mask for the IP address, defining which portion of the address is the network and which part is the host.

Note: Make sure that the correct ports are open and listening for the application to communicate over the network.

19. Final verification

- Ensure that the application is running by accessing its web GUI through the browser.
- Example: Using the IPv4 address and the port number from the image. DDI application can be accessed using anyone of the following urls:

  http://172.21.252.47:9090/ or http://172.21.252.47:9443/
- Test the functionalities of the application to confirm that it is operating as expected.

Installing the DDI Node Agent

Follow the same steps from **Step 1** through **Step 13** to install DDI agent onto each of

DNS and DHCP servers.

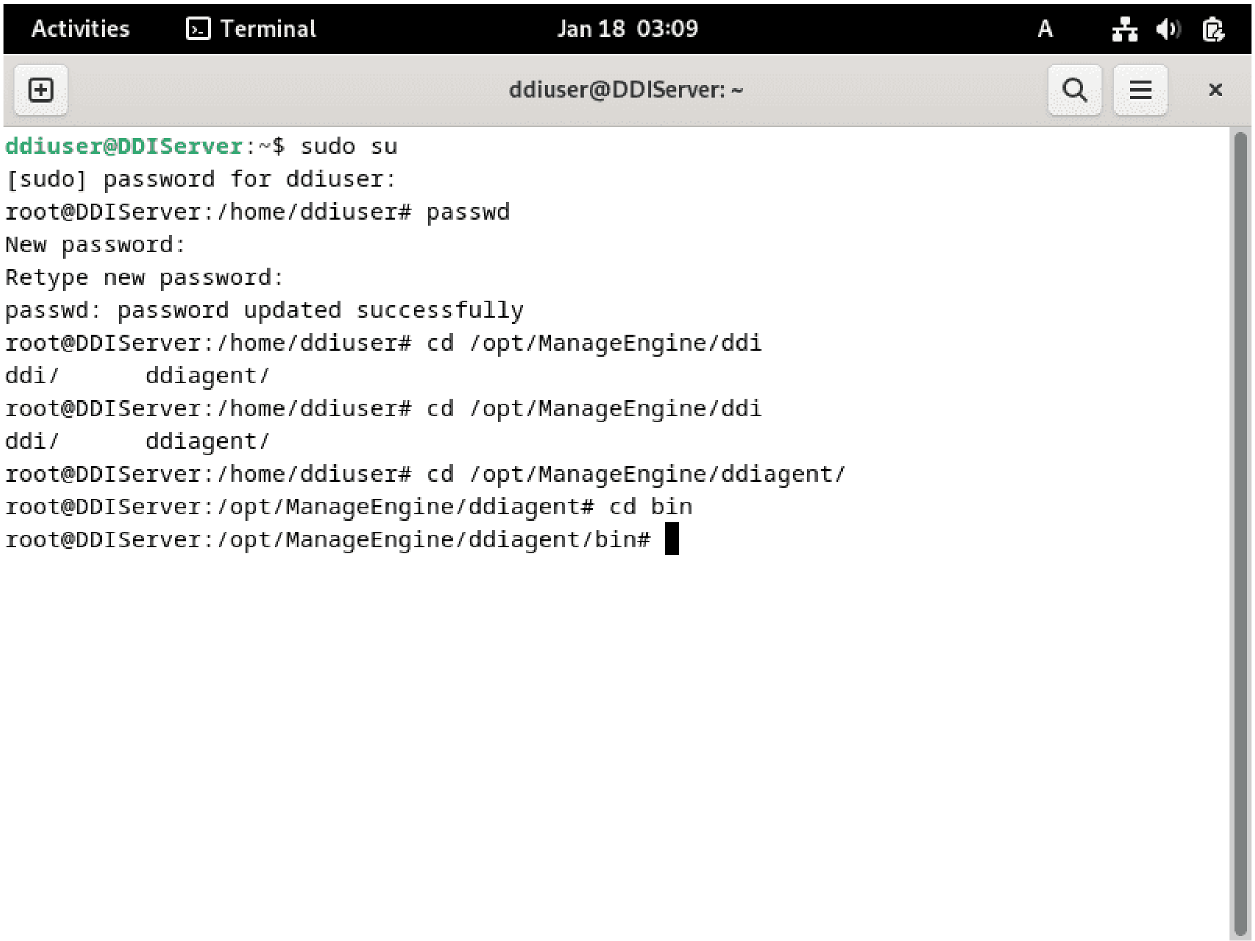
Note: DDI Console and DDI Node Agent cannot run in the same machine.Please ensure to install DDI Node Agent in a different machine you want to manage.

Now for the further steps

14. Access the Application Installation Directory

Navigate to the directory where your application's .bin files are located using the cd command in the terminal.

```
cd /opt/ManageEngine/ddiagent/bin/
```



```
ddiuser@DDIServer:~$ sudo su
[sudo] password for ddiuser:
root@DDIServer:/home/ddiuser# passwd
New password:
Retype new password:
passwd: password updated successfully
root@DDIServer:/home/ddiuser# cd /opt/ManageEngine/ddi
ddi/      ddiagent/
root@DDIServer:/home/ddiuser# cd /opt/ManageEngine/ddi
ddi/      ddiagent/
root@DDIServer:/home/ddiuser# cd /opt/ManageEngine/ddiagent/
root@DDIServer:/opt/ManageEngine/ddiagent# cd bin
root@DDIServer:/opt/ManageEngine/ddiagent/bin# 
```

15. Change file permissions

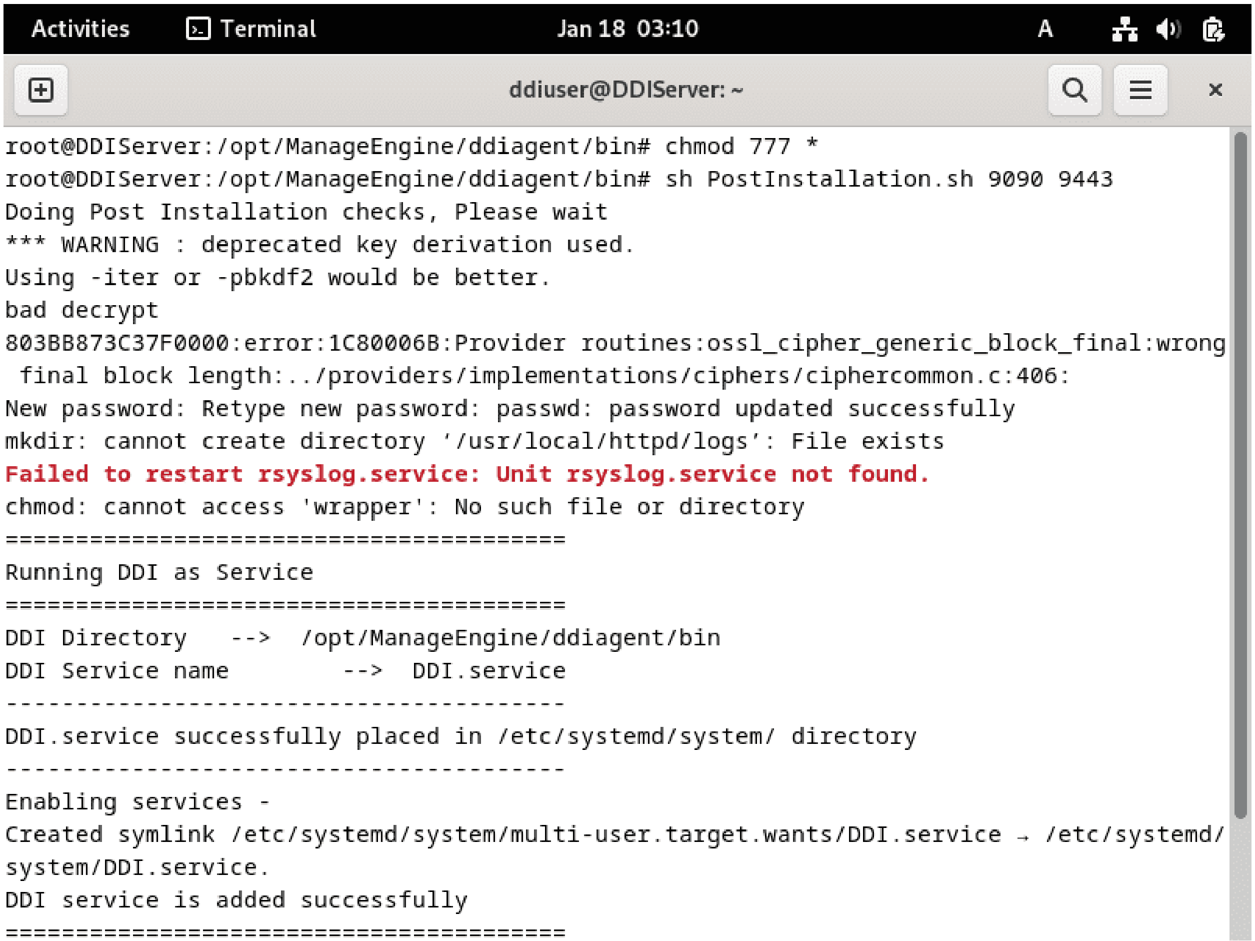
You might need to change the permissions of the scripts to make them executable.

This can be done using the chmod command.

Note: Use the chmod command as specified below, introducing a / (slash) anywhere within the chmod command would yield errors.

chmod 777 *

This command grants read, write, and execute permissions of all files in the current directory.



```
root@DDIServer:/opt/ManageEngine/ddiagent/bin# chmod 777 *
root@DDIServer:/opt/ManageEngine/ddiagent/bin# sh PostInstallation.sh 9090 9443
Doing Post Installation checks, Please wait
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
bad decrypt
803BB873C37F0000:error:1C80006B:Provider routines:ossl_cipher_generic_block_final:wrong
 final block length:../providers/implementations/ciphers/ciphercommon.c:406:
New password: Retype new password: passwd: password updated successfully
mkdir: cannot create directory '/usr/local/httpd/logs': File exists
Failed to restart rsyslog.service: Unit rsyslog.service not found.
chmod: cannot access 'wrapper': No such file or directory
========================================
Running DDI as Service
========================================
DDI Directory   -->  /opt/ManageEngine/ddiagent/bin
DDI Service name        -->  DDI.service
----------------------------------------
DDI.service successfully placed in /etc/systemd/system/ directory
----------------------------------------
Enabling services -
Created symlink /etc/systemd/system/multi-user.target.wants/DDI.service → /etc/systemd/
system/DDI.service.
DDI service is added successfully
========================================
```

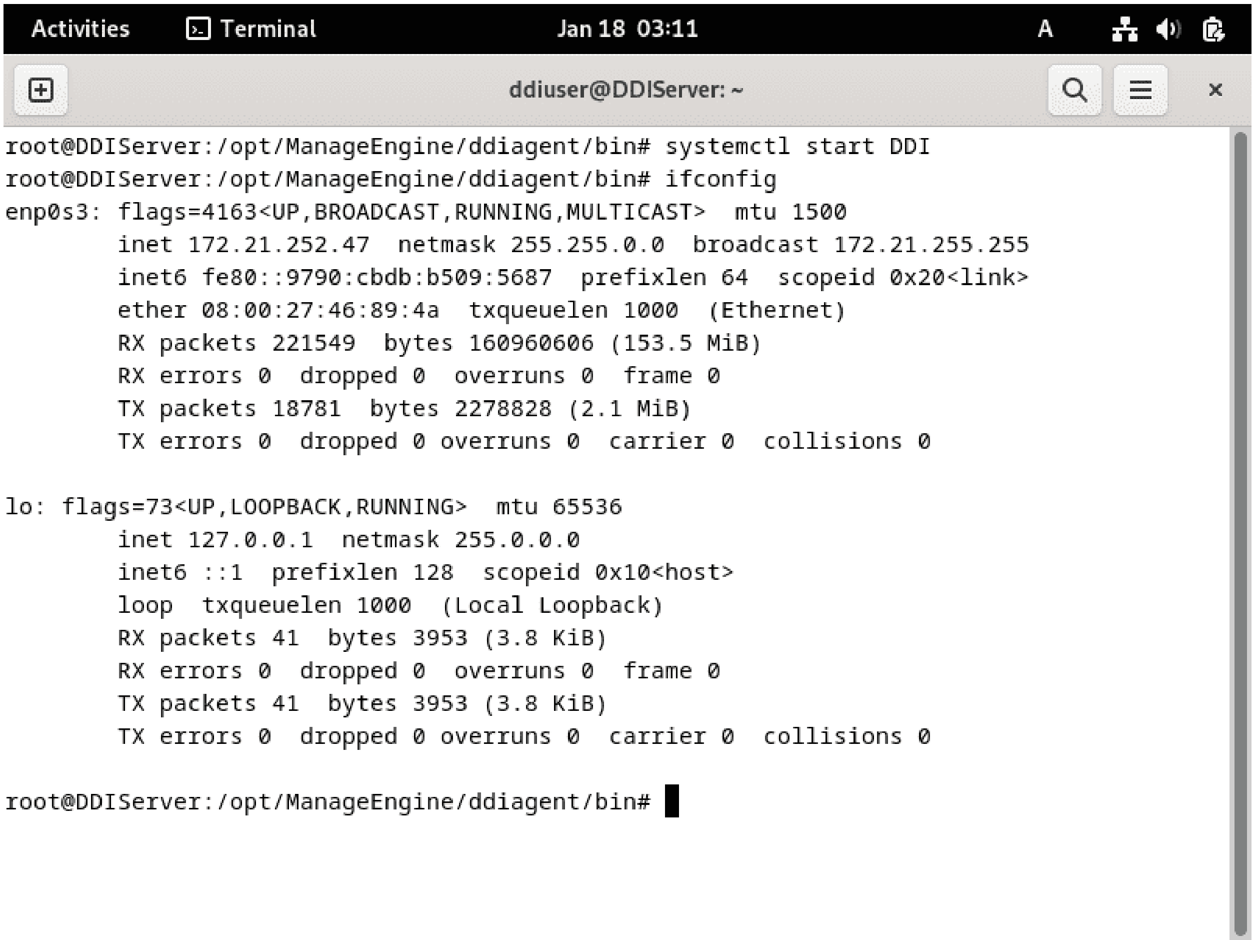
16. Run the installation script

- Execute the command - sh PostInstallation.sh 9090 9443
- This executes a shell script named PostInstallation.sh with two arguments, 9090 and 9443, which are http and https port numbers that the script needs to configure the application to.
- Wait till the script is run completely.

If you encounter errors, you can seek support from the ddi-support team.

17. Start the application services

Once the installation is complete and all configurations are set, start the application services. Use the command systemctl for this purpose: systemctl start DDI



```
root@DDIServer:/opt/ManageEngine/ddiagent/bin# systemctl start DDI
root@DDIServer:/opt/ManageEngine/ddiagent/bin# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.21.252.47  netmask 255.255.0.0  broadcast 172.21.255.255
        inet6 fe80::9790:cbdb:b509:5687  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:46:89:4a  txqueuelen 1000  (Ethernet)
        RX packets 221549  bytes 160960606 (153.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18781  bytes 2278828 (2.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 41  bytes 3953 (3.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 41  bytes 3953 (3.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@DDIServer:/opt/ManageEngine/ddiagent/bin# 
```

18. Verify network configuration

- Use the ifconfig network command to view the configuration of network interfaces and IP addresses. From the response generated, the IP address near inet shows the IPv4 address assigned to the interface. This is the ip address that is used to run the DDI application using the web browser.

- The netmask indicates the network mask for the IP address, defining which portion of the address is the network and which part is the host.

Note: Make sure that the correct ports are open and listening for the application to communicate over the network.

Getting Started

To get started with ManageEngine DDI Central,

Execute the following steps in the order given below

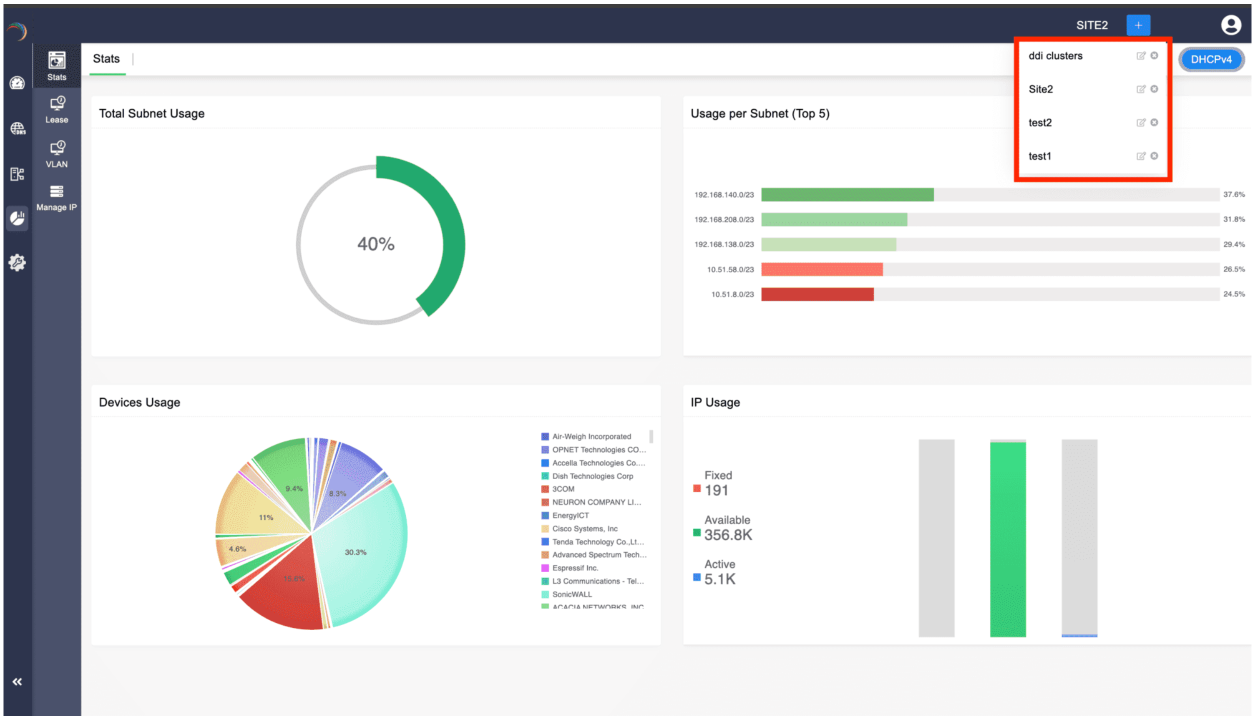
1. Add Clusters
2. Add Servers
3. Add Users (optional)
4. Configure an SMTP server
5. Add Contact Groups (optional)

Adding Clusters

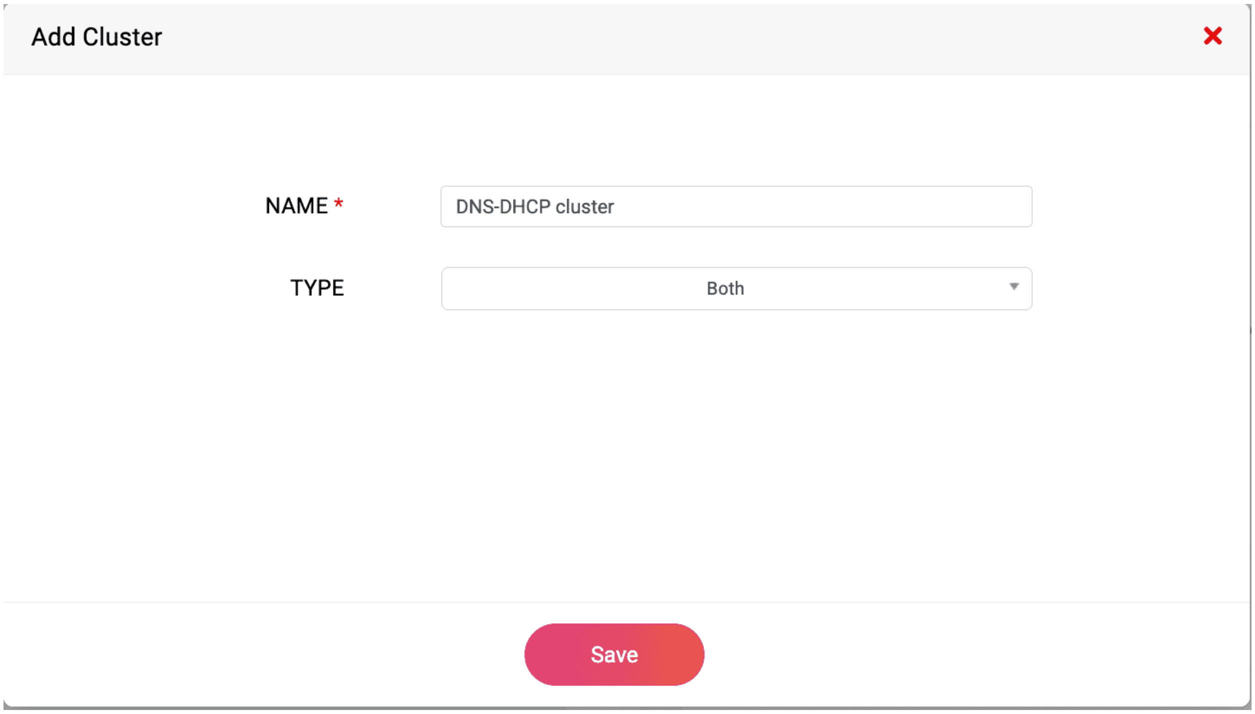
Upon successful signup, the first glimpse within the DDI Central UI reveals an empty dashboard. To get started, create clusters and add your DNS and DHCP servers to your clusters for effective management of your network infrastructure.

To create new clusters

- Click on the **plus (+) sign** at the top right corner.



- The **Add Cluster** window appears prompting you to enter the name and type of the cluster: **DNS, DHCP or Both.**



Add Cluster

NAME * DNS-DHCP cluster

TYPE Both

- Clusters are logical groupings of servers - DNS, DHCP or both organized for identification and administrative purposes. These clusters operate independently of other clusters configured within DDI Central. Each cluster you add has its own internalized IP address plans, IP inventory, IP Address Manager, DNS manager and DHCP manager. A single cluster can accommodate any number of DNS servers and DHCP servers.

- After making the necessary selection, click **Save** to create the cluster.

Once the cluster is created you will be directed to the servers page where you'll be prompted to add your DNS and DHCP servers.

Modes For Deploying Servers

DDI Central server incorporation modes

Servers can be added to the DDI Clusters using the Discovery mode or can be set up as new servers from the scratch.

DDI Central as an overlay

Enable DDI Central to seamlessly discover and integrate your on-premises infrastructure's complete DNS-DHCP server configurations, including the entire IP address footprint, into the intuitive DDI Central console interface.

DDI Central as a DNS-DHCP-IPAM service provider

As ManageEngine DDI Central is bundled with DNS and DHCP services you can set up new servers, enable ManageEngine DDI Central to implement, configure, and manage DNS, DHCP and IPAM services on your network infrastructure from scratch as you install

Core to periphery DDI

Deploy DDI Central flexibly to manage both your on-premises internal and external DNS-DHCP cluster of servers that are accessible via VPN, point-to-point connections, private networks connected via MPLS(Multiprotocol Label Switching) services offered by ISPs, and SD-WANs.

Note: ManageEngine DDI Central only supports Internet Systems Consortium (ISC)'s ISC DHCP and BIND9 DNS servers.

Adding Servers

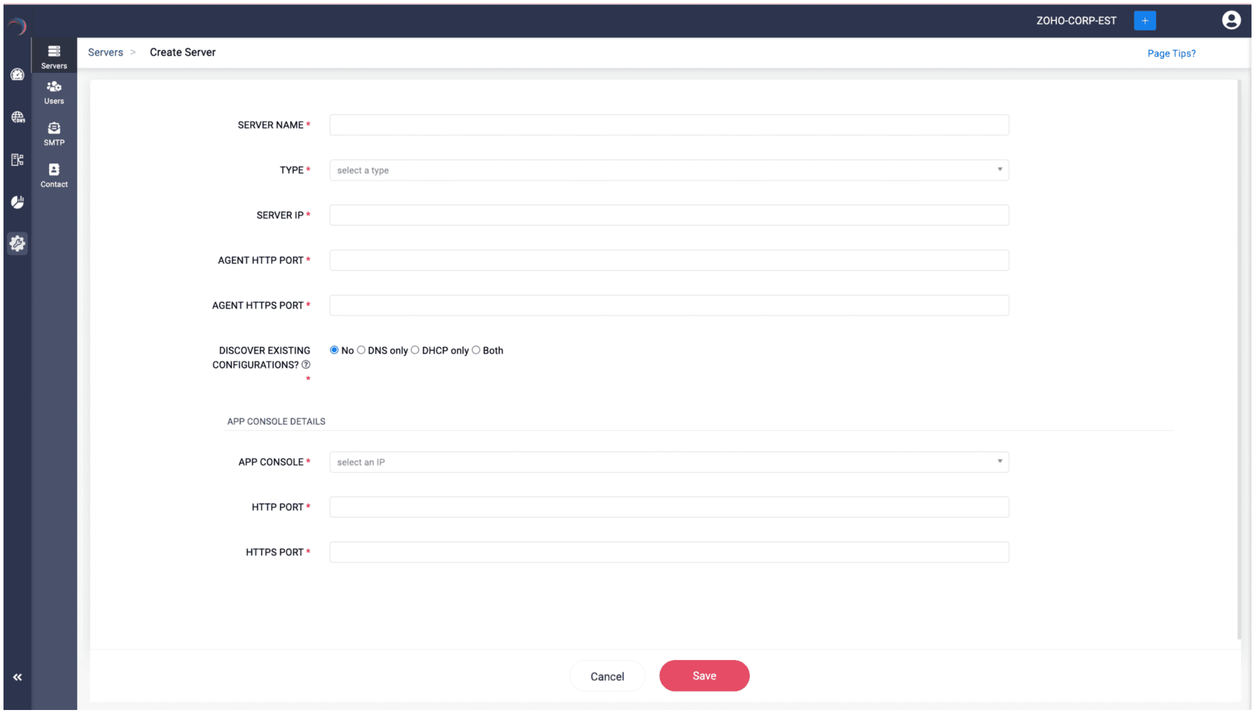
Table of Contents

Creating Servers

Once the cluster is created, you'll be immediately directed to the **Servers** page to add your DNS and DHCP servers. If not, you can add servers by selecting the **Settings** menu from the menu bar along the left side of the screen. From the submenus that appear in parallel, choose **Servers**.

- On the **Servers** page, click the **Add Server** button on the top left corner.

- The **Create Server** page appears on the screen. Here, you can add your DNS-DHCP servers either by discovering existing server configurations or by simply adding the server to the DDI Central console and configure it using the DDI Central user interface at later stages.



SERVER NAME *

TYPE * select a type

SERVER IP *

AGENT HTTP PORT *

AGENT HTTPS PORT *

DISCOVER EXISTING CONFIGURATIONS? *: ● No ○ DNS only ○ DHCP only ○ Both

APP CONSOLE DETAILS

APP CONSOLE * select an IP

HTTP PORT *

HTTPS PORT *

Enter the server details like

1. **SERVER NAME**: A required field where you assign a unique name to the server

being configured or added for identification.

Note: No two servers in the same or different clusters can have the same name.

2. **TYPE**: Select the type of server being set up, such as DNS, DHCP, or both (server that is configured for both DNS and DHCP services).
3. **SERVER IP**: Specify the IP address of the server being added.
4. **AGENT HTTP PORT**: Specify the port number used by the DDI Central Node Agent installed in the server for HTTP connections.
5. **AGENT HTTPS PORT**: Specify the port number used by the DDI Central Node Agent installed in the server for HTTPS connections.
6. **DISCOVER EXISTING CONFIGURATIONS?**: You have two choices to make here; opt for Step 7 or Step 8 depending on your requirement.

   Step 7 → Advanced DNS-DHCP-IP address discovery

   Specify any one of the options :**DNS, DHCP, or Both** to discover all the existing configurations from the server, or

   Step 8 → Adding and configuring servers using DDI Central

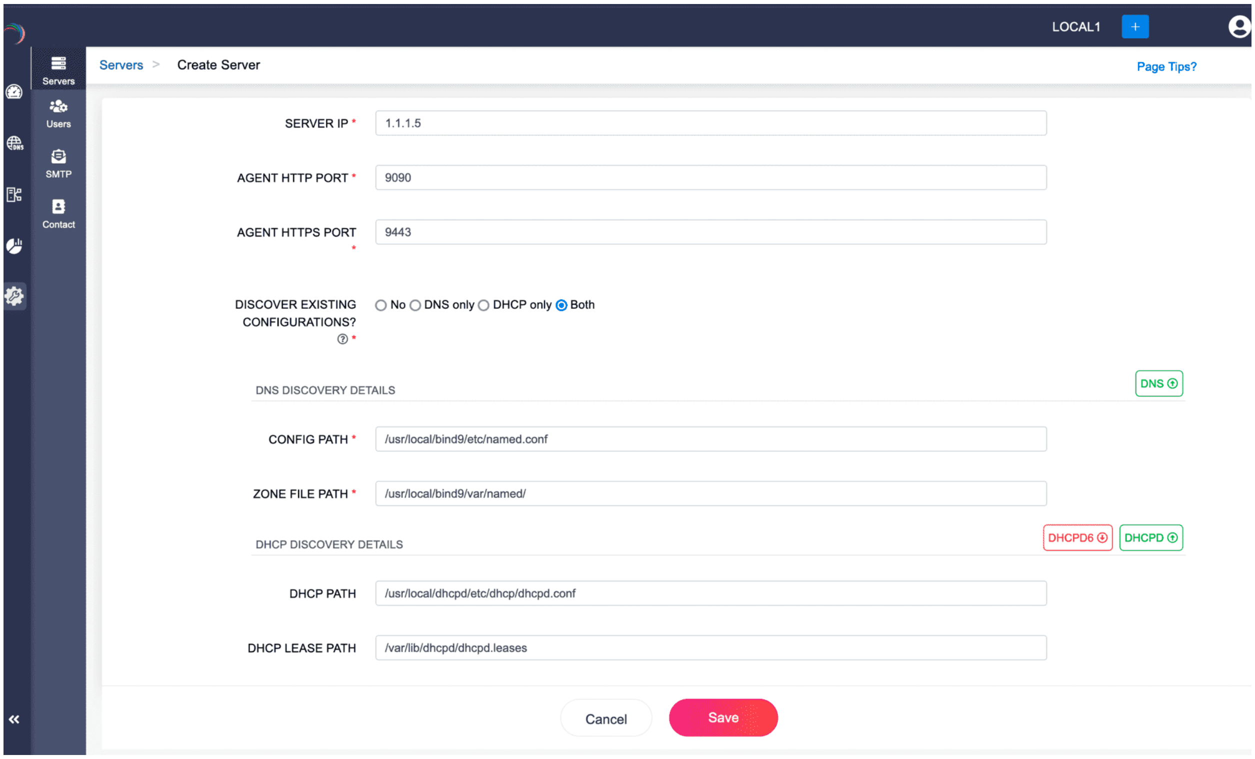
   Specify **No** if you just want to add and setup a new server from the scratch. You can setup the required DNS, DHCP or combined configurations to your server to get it configured through the user-friendly DDI Central user interface later.

Advanced DNS-DHCP-IP address discovery

7. To discover all the advanced configurations of DNS-DHCP services, the whole IP address plan and the current IP address inventory

   Choose any one of the three options: **DNS, DHCP, Both,** for the **Discover Existing Configurations?**

   Note: Selecting either DNS or DHCP will result in the discovery of only the DNS or DHCP configurations, respectively, from the server.
   When discovering a DNS server with DDNS-enabled domains, ensure that both DNS and DHCP servers are discovered at the same time for DDI Central to capture the combined configurations. Similarly, while discovering DHCP servers that provision IP addresses for dynamic domains, it is essential to discover the corresponding DNS servers as well.



SERVER IP * 1.1.1.5

AGENT HTTP PORT * 9090

AGENT HTTPS PORT * 9443

DISCOVER EXISTING CONFIGURATIONS? * ○ No ○ DNS only ○ DHCP only ● Both

DNS DISCOVERY DETAILS (DNS)

CONFIG PATH * /usr/local/bind9/etc/named.conf

ZONE FILE PATH * /usr/local/bind9/var/named/

DHCP DISCOVERY DETAILS (DHCPD6) (DHCPD)

DHCP PATH /usr/local/dhcpd/etc/dhcp/dhcpd.conf

DHCP LEASE PATH /var/lib/dhcpd/dhcpd.leases

Provide the essential ***Config Path*** and the ***Zone File path*** for the DNS servers, while providing the ***Lease Path*** and the ***DHCP server path*** for the DHCP servers.

Setting up servers through DDI Central

8. You can add new servers to DDI Central console and enable ManageEngine

DDI Central to implement, configure, and manage DNS, DHCP and IPAM services on your network infrastructure from scratch.

As DDI Central has DNS and DHCP bundled with the product and it gets deployed on your servers while installing the product.

For this, you'll have to choose **No** for **Discover Existing Configurations?** option.

App Console Details

9. **APP CONSOLE**: Enter the static IP address of the central server that hosts the DDI Central application console associated with this server.

   Note: It is crucial that this IP address remains constant to maintain consistent connection between the central DDI Central console server and the Node Agents installed in all your DNS and DHCP servers.
10. **HTTP PORT**: Specify the port number of the central DDI Central application console server for HTTP connections.
11. **HTTPS PORT**: Specify the port number of the central DDI Central application console server for HTTPS connections.

12. Click **Save** to add the server into the ME DDI Central console.

If you have chosen the discovery option as outlined in **Step 7,** ManageEngine DDI Central will begin to discover configurations from the designated paths for each service.

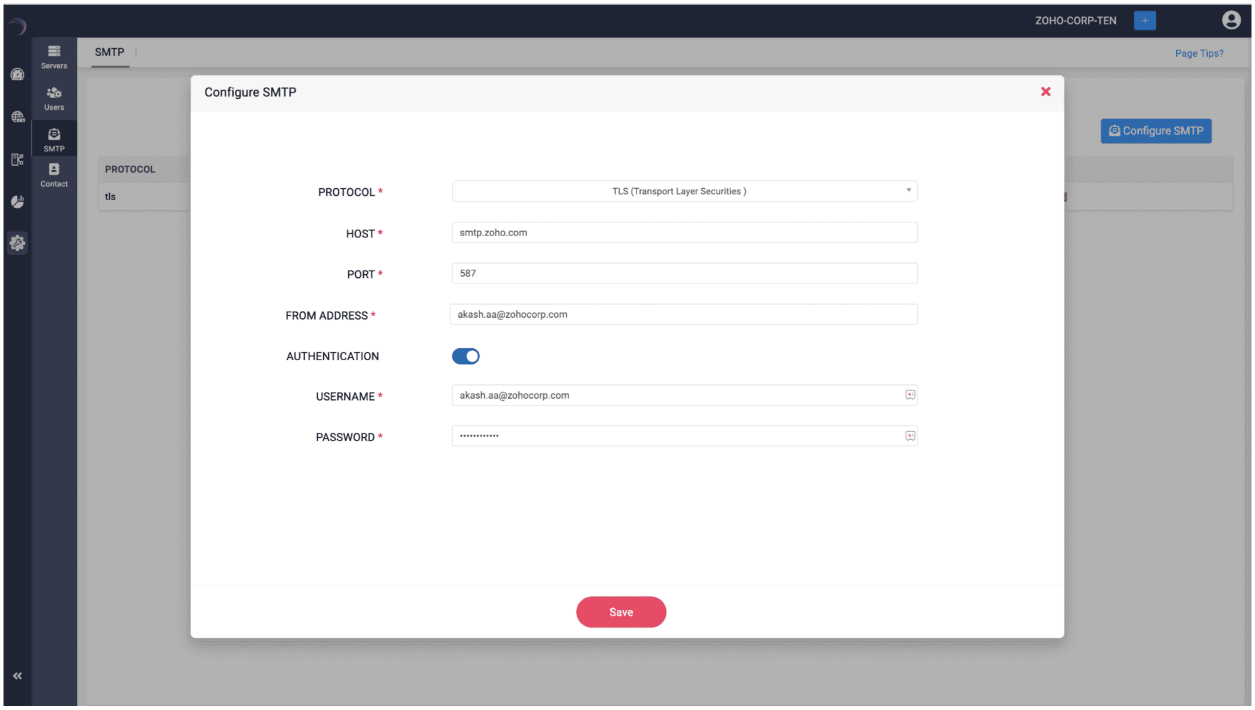
Note: The discovery process takes a considerable amount of time depending on the volume of configurations in the servers. Wait until the whole process completes.

Once you add your server into the DDI Central console you can further proceed modifying the discovered DNS-DHCP-IPAM configurations or quickly start setting up the DNS-DHCP-IPAM configurations for the new server through the user-friendly DDI Central user interface.

Configuring The SMTP Host

DDI Central Admins can configure to send email using a particular SMTP host.

- Provide an SMTP username and password for the authentication of email notifications. This is optional; you can enable or disable it anytime.
- Configure the SMTP host, sender address, and optional username and password.



Configure SMTP

PROTOCOL * TLS (Transport Layer Securities)

HOST * smtp.zoho.com

PORT * 587

FROM ADDRESS * akash.aa@zohocorp.com

AUTHENTICATION ●

USERNAME * akash.aa@zohocorp.com

PASSWORD * ••••••••••••

Save

1. **PROTOCOL**: Choose the encryption protocol for SMTP communication: either TLS (Transport Layer Security) or SSL (Secure Sockets Layer), both of which ensure that email communications are encrypted for security.
2. **HOST**: Provide the FQDN(Fully Qualified Domain Name) of your mail server in the following format: hostname. domain. tld
3. **PORT**: The port number used for SMTP connections. It's set to **587 for TLS**, 465 for SSL. If no encryption protocol is chosen the port number switches to the traditional SMTP port 25.

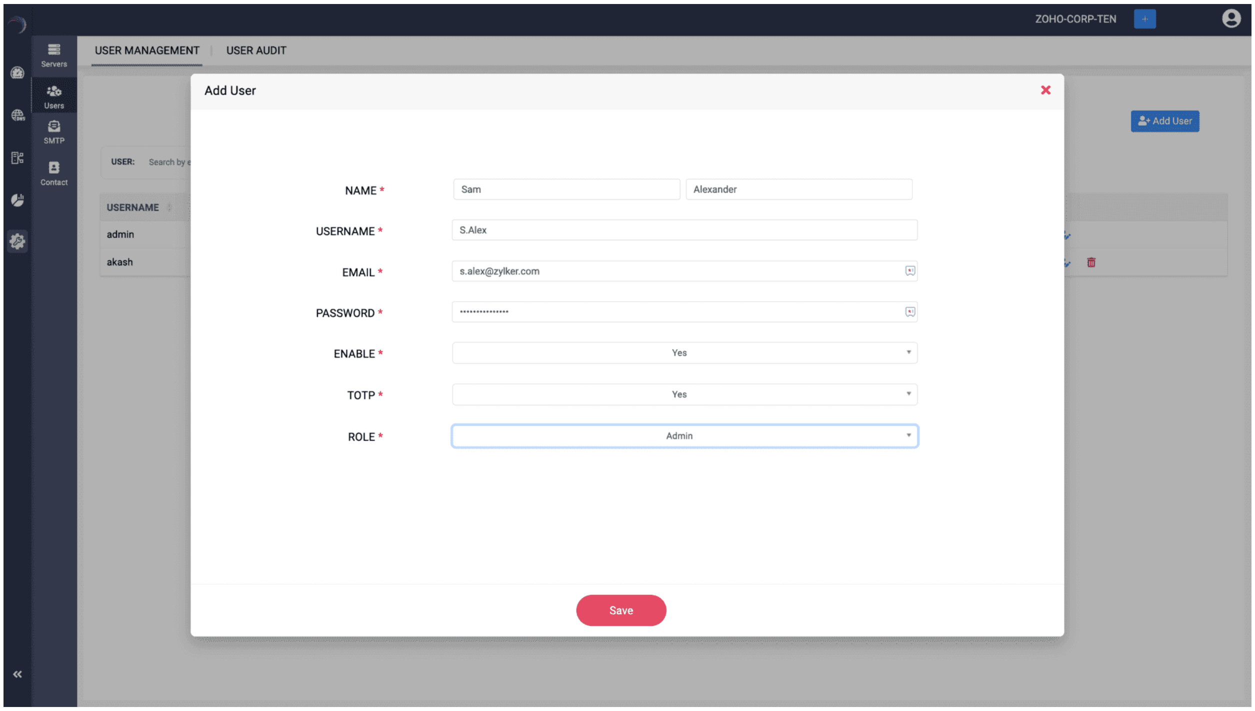
   Note: Port 25 does not imply any encryption and is often used for relaying emails across servers. Due to its lack of security features, it's generally not recommended for submitting emails from clients to servers. Additionally, many ISPs block outgoing connections on port 25 to reduce spam.
4. **FROM ADDRESS**: A valid sender email address DDI uses to send mails incase of password recovery and other notifications.
5. **AUTHENTICATION**: A toggle switch, which can enable or disable the authentication required for sending emails through this SMTP host.
6. **USERNAME**: The username for authenticating with the SMTP host, often the same as the email address.
7. **PASSWORD**: The password required for SMTP authentication.

To start the email notification service and subscribe to notifications pertaining to a cluster, set up a **Contact group** under the cluster by selecting **Settings→ Contact groups.**

Adding Users

To add users as an admin:

- Select **Settings→Users.**
- Under the **User Management** tab, click on the **Add User** button in the right corner.



Add User

NAME * Sam Alexander

USERNAME * S.Alex

EMAIL * s.alex@zylker.com

PASSWORD * •••••••••••••••

ENABLE * Yes

TOTP * Yes

ROLE * Admin

Save

- Enter the essential details of the user, including Name, Username, email, and password. You can enable or disable the login for this particular user. Set Yes to enable the login. Enable the TOTP login for the user to add an extra layer of security.
- Finally, Assign the appropriate role for the user.
- DDI Central provides two roles: **Admin** and **Operator**. The Admin role has unrestricted access, while the Operator role has limited access, which can be extended by granting specific permissions for each cluster or zone as needed.
- Click **Save**.
- Provide the Username, Password, and URL for the other users you've added. Make sure they login using the URL from their web browser.
- Once they login they'll be prompted to reset their password and login to the DDI Central system.
- **Enabling Two-factor authentication for the users**

DDI Central enhances user account security by mandating two-factor authentication (2FA) for all users associated with your organization. This additional security layer requires verification through a time-sensitive code generated by a compatible mobile authenticator application. The following steps outline the 2FA process.

1. Users need a mobile device capable of running a TOTP-enabled authenticator mobile app.
2. ManageEngine DDI Central is compatible with various mobile authenticator apps, including Google Authenticator, Zoho's OneAuth, Authy, and others.
3. Install your chosen authenticator app on your smartphone.
4. Link DDI Central to the authenticator app either by scanning the QR secret code displayed on the DDI Central login page or by entering the code manually. This is a one-time process.
5. On subsequent logins, enter the TOTP displayed in your authenticator app. The OTP adds an extra layer of security and can be generated without an internet connection.
6. Upon first accessing DDI Central, all users except the Admin who managed the installation process will need to reset their password.

This two-factor authentication approach ensures that access to DDI Central accounts is secure, combining something the user knows (their password) with something they have (a TOTP from the authenticator app).

User permissions

Admin can	**Operator can**
Create, update, and delete user	-
Add, update, and delete zones	Update zone if operator has zone permission
Create update and delete cluster	-
Giving cluster and zone permission to the operator	-
Add, update, and delete servers	-

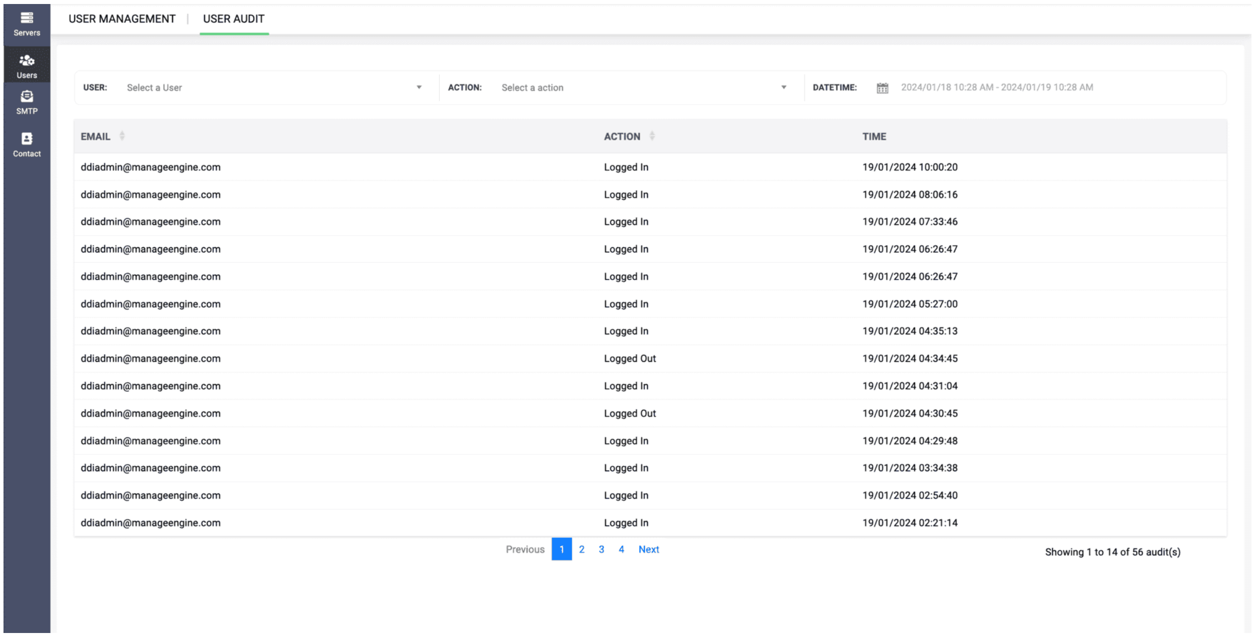
Add SMTP details	-
Able to see login and logout details of the user	-
Able to see DHCP and DNS audit report	-
Reset client credentials	Reset client credentials
Enable TOTP for an user	-
Delete TOTP device	-
Add, update, and delete records in zone	Add, update, and delete records in zone if the operator has zone permission
Add, update, and delete named options	Add, update, and delete named options if the operator has cluster permission

Add, update and delete dhcp options	Add, update and delete dhcp options if operator has cluster permission
Add, update, and delete custom options	Add, update, and delete custom options if the operator has cluster permission
Add, update, and delete subnet, shared network, client class, host, host group and vlan	Add, update and delete subnets, shared network, client classes, host, host group and vlan if the operator has cluster permission
Add, update and delete supernet	Add, update and delete supernet if operator has cluster permission
Add, update, and delete failover	Add, update, and delete failover if the

configurations	operator has cluster permission
Enable, add, update, and delete named views	update named_view if operator has cluster permission
Add, update and delete DHCP Zone	Add, update, and delete DHCP Zone if operator has cluster permission
Add, update, and delete records in views	Update view if operator has zone permission

User Audits

The **User Audit** tab can be accessed by selecting the **Audit** menu from the left menu bar. The User audit tab helps you monitor your users' login activities by capturing the username, date, and timestamp of the latest login activities.



USER: Select a User

ACTION: Select a action

DATETIME: 2024/01/18 10:28 AM - 2024/01/19 10:28 AM

EMAIL	ACTION	TIME
ddiadmin@manageengine.com	Logged In	19/01/2024 10:00:20
ddiadmin@manageengine.com	Logged In	19/01/2024 08:06:16
ddiadmin@manageengine.com	Logged In	19/01/2024 07:33:46
ddiadmin@manageengine.com	Logged In	19/01/2024 06:26:47
ddiadmin@manageengine.com	Logged In	19/01/2024 06:26:47
ddiadmin@manageengine.com	Logged In	19/01/2024 05:27:00
ddiadmin@manageengine.com	Logged In	19/01/2024 04:35:13
ddiadmin@manageengine.com	Logged Out	19/01/2024 04:34:45
ddiadmin@manageengine.com	Logged In	19/01/2024 04:31:04
ddiadmin@manageengine.com	Logged Out	19/01/2024 04:30:45
ddiadmin@manageengine.com	Logged In	19/01/2024 04:29:48
ddiadmin@manageengine.com	Logged In	19/01/2024 03:34:38
ddiadmin@manageengine.com	Logged In	19/01/2024 02:54:40
ddiadmin@manageengine.com	Logged In	19/01/2024 02:21:14

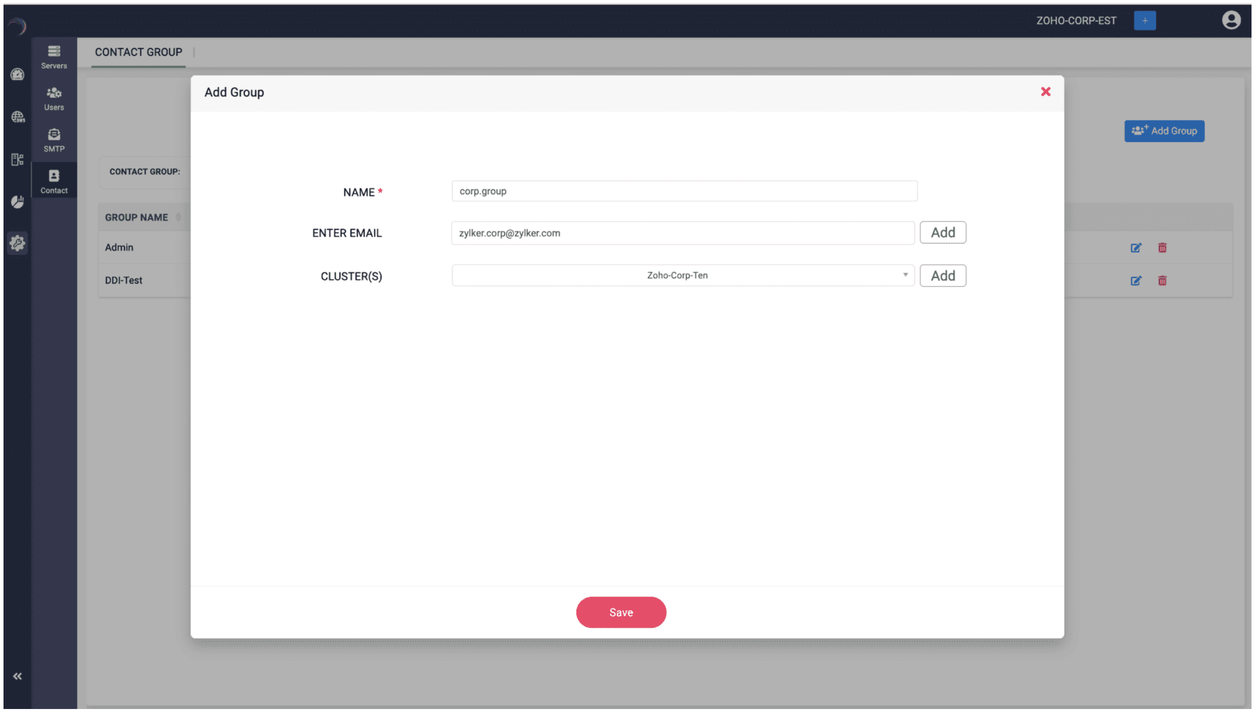
Showing 1 to 14 of 56 audit(s)

Creating Contact Groups

DDI Central enables you to group specific users under your organization to create special contact groups. You can associate the relevant contact group to be notified of alerts or incidences concerning that domain and associated monitor.

To create a contact group

1. Select **Contact** from the left menu bar. In the Contact page, under the Contact Groups tab, CloudDNS displays the list of contact groups created under the organization.
2. Click on the **Add Group** button to create a new contact group.



Add Group

NAME * corp.group

ENTER EMAIL zylker.corp@zylker.com

CLUSTER(S) Zoho-Corp-Ten

3. Enter the details of the group, like the group name.
4. For the group email, add the list of email ids of the members to send notifications to, one by one, and click **Add** after each selection.
5. Select the required clusters one by one and click the **Add** button after each selection. Click **Save**.
6. On successful association, the **Contact Group** tab on the **Contact** page displays the list of members in the Contact Group as well as the list of **Associated Clusters**.
7. To dissociate any cluster or contact from the Contact Group, click on the Edit

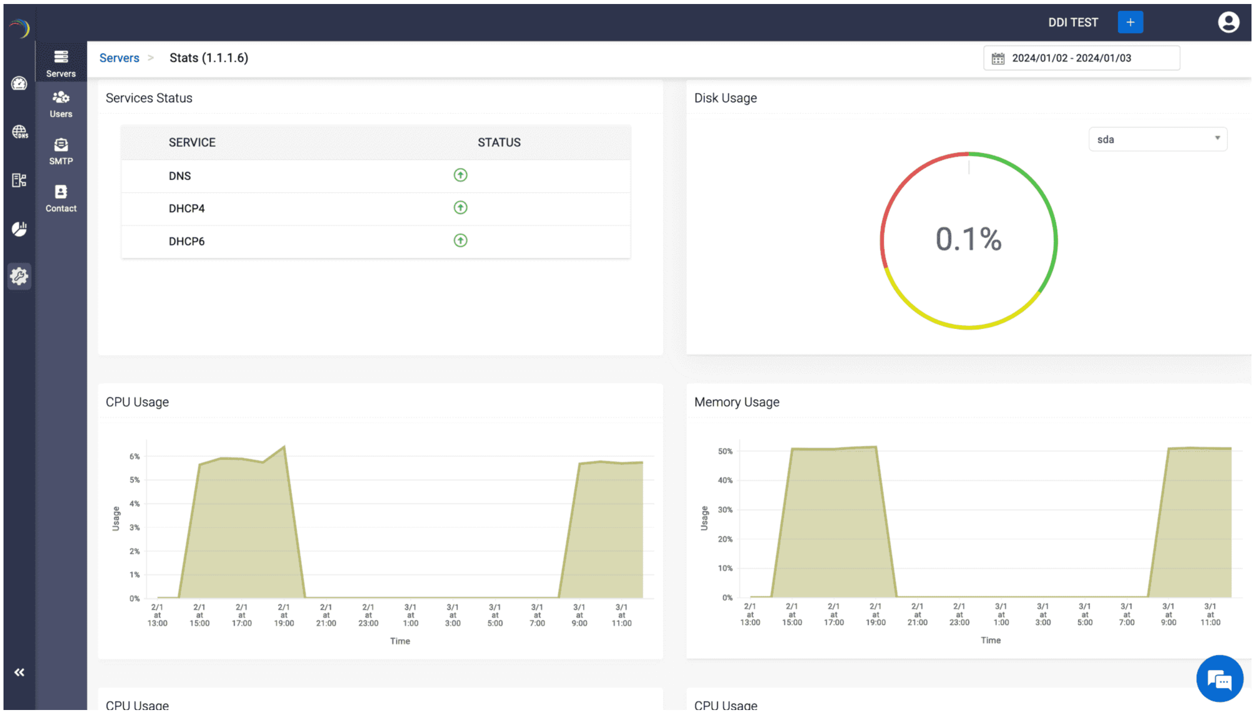
button on the extreme right. From the Edit Group window, deselect the email ids or the clusters using the minute close button at the top right corner of each selection.

Monitoring servers

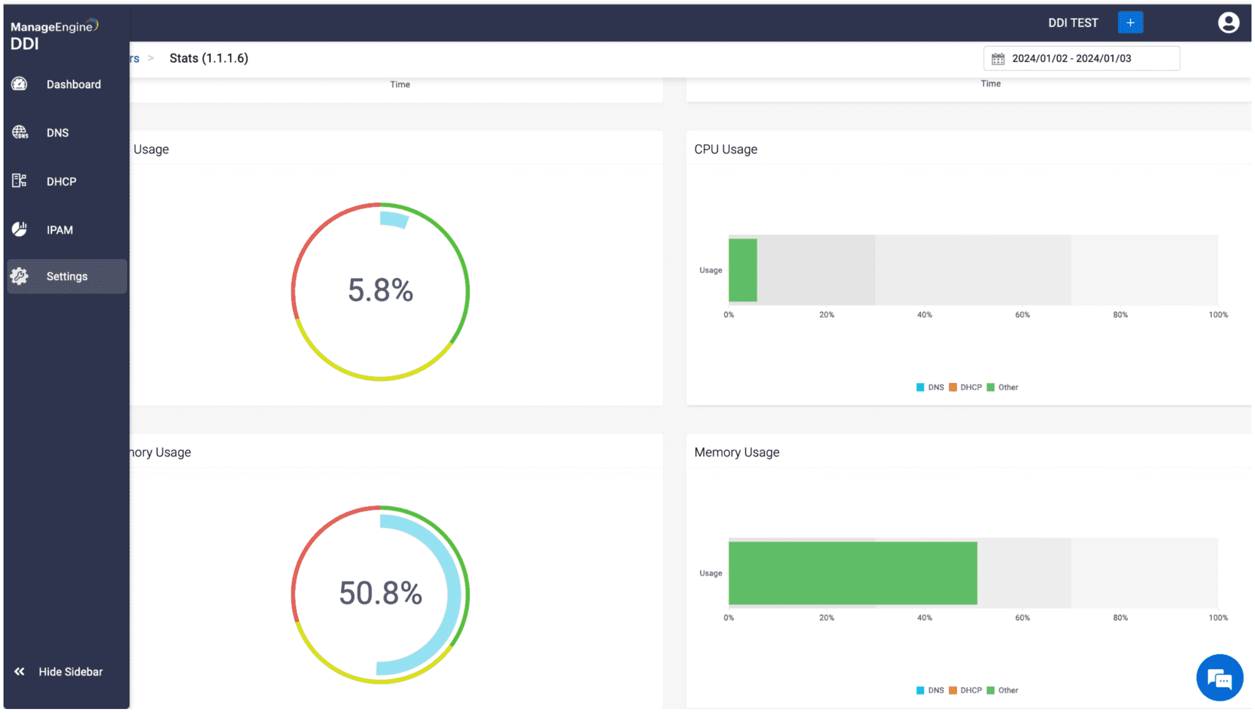
To monitor the load and performance of your DNS and DHCP servers:

Select **Settings→ Servers** . The **Servers** page appears listing the servers added.

First it displays the status of the DNS, DHCP4, and DHCP6 services of the cluster.



SERVICE	STATUS
DNS	↑
DHCP4	↑
DHCP6	↑



DDI Central also gives a visual snapshot of a different part of your server's load, health and performance. CPU, memory, and disk percentages of your server represent different aspects of the server's system resource usage, each playing a unique role in the server's overall performance. Understanding the differences between them can help diagnose performance issues or guide system upgrades.

1. **CPU Percentage**

- The CPU (Central Processing Unit) percentage indicates how much of the CPU's processing power is being used. It reflects the workload being processed by the CPU of your server at any given moment.
- **Implications:** A high CPU percentage can mean the processor is handling a lot of tasks simultaneously or dealing with a few very demanding tasks. If the CPU usage is consistently high, the server might slow down or become unresponsive, especially if it's attempting to process more data than it can handle efficiently.

2. **Memory (RAM) Percentage**

   - The memory percentage refers to the proportion of the computer's RAM (Random Access Memory) that is currently in use. RAM is used to store data and program instructions needed immediately or shortly by the CPU.
   - **Implications:** High memory usage indicates that a large amount of the system's RAM is being used. If the server runs out of RAM, it starts using disk space as virtual memory, which is much slower. Excessive memory usage can slow down the system, cause programs to respond more slowly, and may lead to system instability.

3. **Disk Usage Percentage:**

- **Definition:** Disk usage percentage shows how actively the server's hard drive (or SSD) is being read from or written to. It’s different from disk capacity, which refers to how much data is stored on the disk.
- **Implications:** High disk activity can indicate that a lot of data is being transferred to and from the storage device. This could be due to various reasons, like file copying, intensive read/write operations by applications, or because the server is using the disk for virtual memory. Prolonged high disk usage can slow down the server, as the disk is generally the slowest component in terms of data access.

For optimal performance, it's crucial to have a balanced server where no single resource consistently becomes a bottleneck. For example, a powerful CPU can be underutilized if the server doesn't have enough RAM or if the disk is too slow to provide data quickly. Similarly, having a lot of RAM is less useful if the CPU isn't fast enough to process the data held in the RAM, or if the disk is too slow to load new data into the RAM efficiently. Regular monitoring of these percentages can help in identifying and resolving performance bottlenecks in a computer system.

About DNS Management

Domain Name System (DNS) Management refers to the process of translating human-readable domain names (like ddi.manageengine.com) into IP addresses that computers use to communicate with each other. Efficient DNS management is crucial for ensuring that websites and online services are accessible, reliable, and secure. It involves tasks like configuring DNS records (such as A, AAAA, CNAME, MX records), managing domain names, setting up reverse DNS, and ensuring DNS security with configurations like DNSSEC, domain views, Response Policy Zones(RPZ) and more.

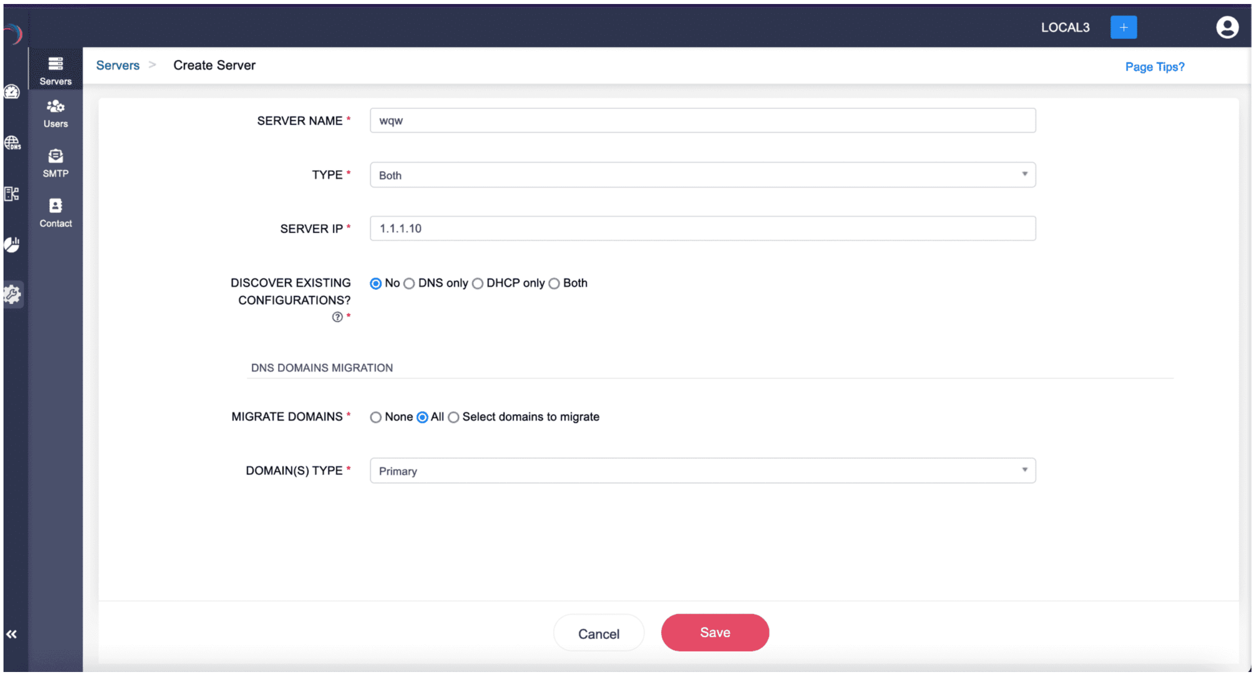
Organizations often use DNS management tools or services to streamline these tasks, handle large volumes of DNS traffic, and protect against DNS-related attacks. Good DNS management also plays a key role in optimizing website performance and uptime, crucial for providing a positive user experience.

DNS Domain Migrations

DDI Central enables you to add new DNS servers through a special mode called the **DNS Domain Migrations** mode. When using this mode, it is necessary to skip the discovery phase. This is done by selecting **No** for the **Discover Existing Configurations** option.

This approach is particularly advantageous when integrating a new DNS server that manages a vast number of domains. These domains can be added to your existing cluster as either primary or secondary.

The DNS Domain Migration mode facilitates the rapid inclusion of all domains or the selective migration of specific domains from the server being added. If you choose **None,** the new DNS server will be incorporated into your DDI cluster without any domains, allowing you to configure zones and other settings manually at a later stage.



SERVER NAME * wqw

TYPE * Both

SERVER IP * 1.1.1.10

DISCOVER EXISTING CONFIGURATIONS? ⓘ * ● No ○ DNS only ○ DHCP only ○ Both

DNS DOMAINS MIGRATION

MIGRATE DOMAINS * ○ None ● All ○ Select domains to migrate

DOMAIN(S) TYPE * Primary

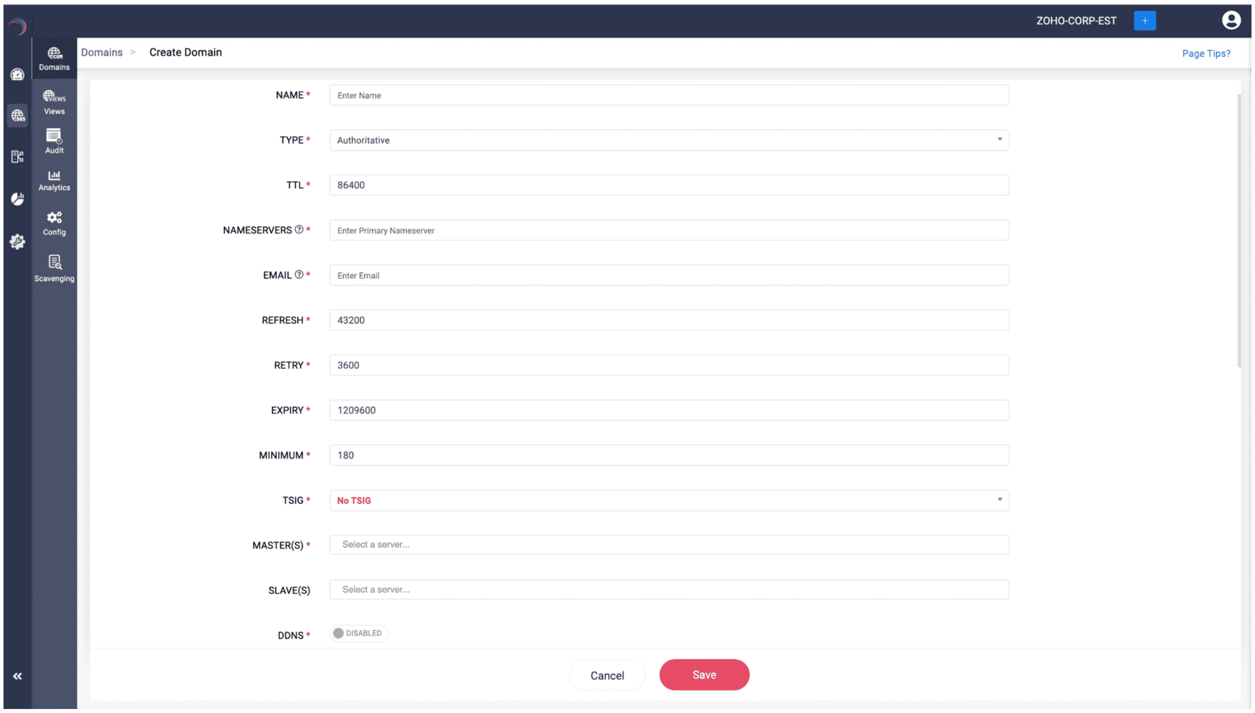
To proceed, select whether the domains of this particular server should be recognized as **Primary** or **Secondary** within your cluster, and click **Save** to begin the migration process. As this can take some time, it is advisable to wait for further prompts or indications on the screen before proceeding.

Creating Authoritative Zones

You can create a new domain using the **Add Domain** button or import domains in bulk using the **Import** button in the top right corner.

Add Domain

On clicking the **Add Domain** button, the **Create Domain** page appears as shown below.



Domains > Create Domain

NAME *	Enter Name
TYPE *	Authoritative
TTL *	86400
NAMESERVERS ⓘ *	Enter Primary Nameserver
EMAIL ⓘ *	Enter Email
REFRESH *	43200
RETRY *	3600
EXPIRY *	1209600
MINIMUM *	180
TSIG *	No TSIG
MASTER(S) *	Select a server...
SLAVE(S)	Select a server...
DDNS *	○ DISABLED

Cancel   Save

In the **Create Domain** page enter the value for the fields based on the descriptions below.

1. **NAME:** Name the new domain name you wish to create or manage.
2. **TYPE:** Select the type of DNS zone. DDI Central offers three types of Zones: Authoritative, Forward and Response Policy Zones (RPZ). Now select Authoritative.

   Authoritative: This type indicates that the DNS server has the authority to answer queries for the specified domain with authoritative data. This means it holds the actual DNS records, such as A records, MX records, and so on, for the domain. It is the definitive source for information about that domain, and its answers are considered official.

Note: The type of the domain once chosen cannot be updated. To update the type of the domain, delete the domain and re-configure it.

3. **TTL (Time to Live):** Specifies the resolving servers how long to cache information about the domain before querying for it again. Specified in seconds, and the default value is set to 86400.
4. **NAMESERVERS:** Enter the nameserver that has the authority to resolve queries and providing responses to queries for this domain.
5. **EMAIL:** Enter the email address of the domain administrator or the personnel responsible for managing the DNS zone.
6. **REFRESH:** Specified in seconds, tells secondary nameservers how often to check with the primary nameserver for updates.
7. **RETRY:** Incase the secondary nameserver fails to reach the primary, this value specifies how long it should wait before retrying.
8. **EXPIRY:** Determines the duration, in seconds, for which a secondary nameserver will attempt to contact the primary nameserver. After this period,

if no response is received, the secondary nameserver will consider the data stale and cease responding to queries with it.

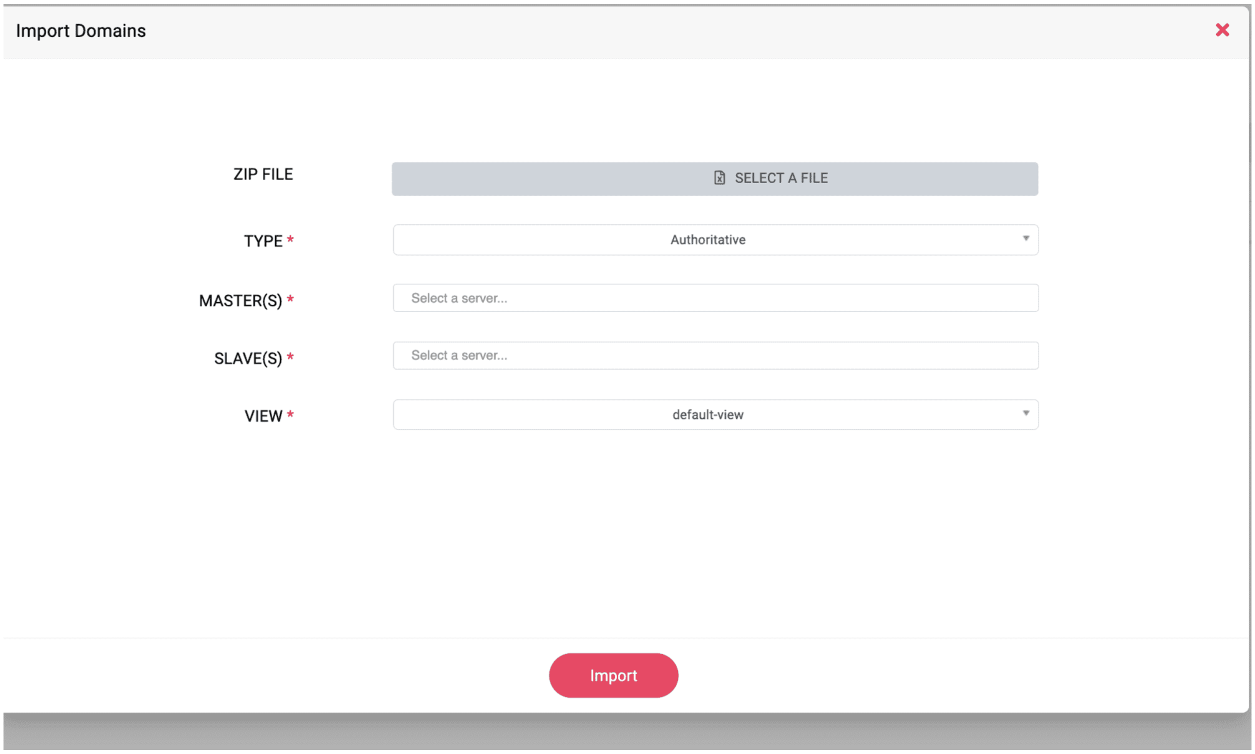
9. **MINIMUM:** Specifies the minimum TTL that tells the resolving servers how long to remember that a particular record does not exist.
10. **TSIG:** Transaction Signature is a security protocol used to secure zone transfer operations. "No TSIG" indicates that no transaction signature will be used.
11. **MASTER(S):** Specifies the master DNS server for the zone. In a primary-secondary setup, the master server is where the zone records are originally created and managed.
12. **SLAVE(S):** Specifies secondary or slave DNS servers. These servers get their zone data from the master server through zone transfers.
13. **DDNS:** Dynamic DNS allows for the automatic updating of a name server in the Domain Name System via DHCP servers.
14. Failing to enable DDNS here, you can alternative enable DDNS by navigating to DHCP→ Domains, Add the domain there and specify the TSIG key for secure dynamic updates.

Click **the Save button** at the bottom to create or update the domain with the specified parameters, while **Cancel** would discard any changes made.

Import Domains

DDI Central enables you to bulk import records into the DDI Central console using the **Import** button on the top right corner.

On clicking the **Import** button, the **Import Domains** window appears on screen.



Import Domains

ZIP FILE: SELECT A FILE

TYPE *: Authoritative

MASTER(S) *: Select a server...

SLAVE(S) *: Select a server...

VIEW *: default-view

On the Import Domains window, enter the values for the following fields:

1. **ZIP FILE:** Upload the ZIP file containing domain information in a specific format required by the system.
2. **TYPE:** Specify the type of DNS zone being imported. The default option shown is Authoritative, indicating that the data being imported is for an authoritative DNS zone. Authoritative DNS zones are responsible for containing the DNS records for a particular domain.
3. **MASTER(S):** Select one or more master DNS servers. The master server would be the primary source of data for the zones being imported and would handle DNS queries and updates.
4. **SLAVE(S):** Similar to the MASTER(S) field, select one or more slave (or secondary) DNS servers. Slave servers receive zone data from the master server and serve as backups to handle queries if the master is unavailable.
5. **VIEW:** Select a DNS view by its name to provide different information to different sets of clients based on some pre-defined criteria. For example, you might have an "internal" view for clients on your local network and an "external" view for clients on the internet.
6. **IMPORT:** Once all fields are filled out and you are ready to import the domain

data, click the **Import** button to initiate the import process.

The two methods discussed above can help you create authoritative zones.

Creating Reverse Authoritative Zones

Reverse DNS is the process of resolving an IP address back to a domain name. It is commonly used for services such as email servers to verify that an IP address maps to a domain name. Reverse DNS for an IP address is configured by setting up a PTR (Pointer) record in the reverse DNS zone. This zone is named after the reversed IP address in the case of IPv4, and after the reversed nibbles of the IP address in the case of IPv6. Reverse zones can only be configured for authoritative zones.

Reverse zones for IPv4 addresses

Follow the steps below to create reverse authoritative zones for IPv4 addresses.

Prerequisites

- Determine the subnet for which you want to create the reverse zone.
- Determine the authoritative DNS server for this zone.

1. Create a reverse authoritative zone just like how you create a standard authoritative zone by selecting **DNS→ Domain→ Add Domain**
2. However while entering the reverse zone name, make sure to derive it from your IP block in reverse order, followed by .in-addr.arpa. For example, for the IP block 192.168.1.0/24, the reverse zone name should be 1.168.192.in-addr.arpa.
3. Create PTR (Pointer) records within the reverse zone. PTR records map the IP addresses within your network block to the corresponding domain names. Each record will correlate an IP address to a hostname, with the IP address portion written in reverse.
4. You can enable DDNS Dynamic updates. Ensure the DHCP server is in the

same network as the Zone. This automates the creation of PTR records, configure the DHCP options to allow dynamic updates from the DHCP server to the domain.

5. If you want to secure the reverse zone with DNS Security Extensions (DNSSEC), you can sign the zone to generate the necessary keys and signing policies.
6. Click **Save** to save your new reverse zone configuration.
7. Perform reverse DNS lookups on IP addresses within the network block to ensure that the PTR records are correctly resolving to their respective domain names.

Reverse zones for IPv6 addresses

For IPv6 addresses, the process is similar to IPv4, but the notation and the domain used for reverse DNS delegation are different. The domain used for IPv6 reverse DNS is .ip6.arpa.

Here is how you derive the reverse zone name from an IPv6 address block:

Let's say you have an IPv6 address block of 2001:0db8:85a3::/48. To create a reverse zone name for this block:

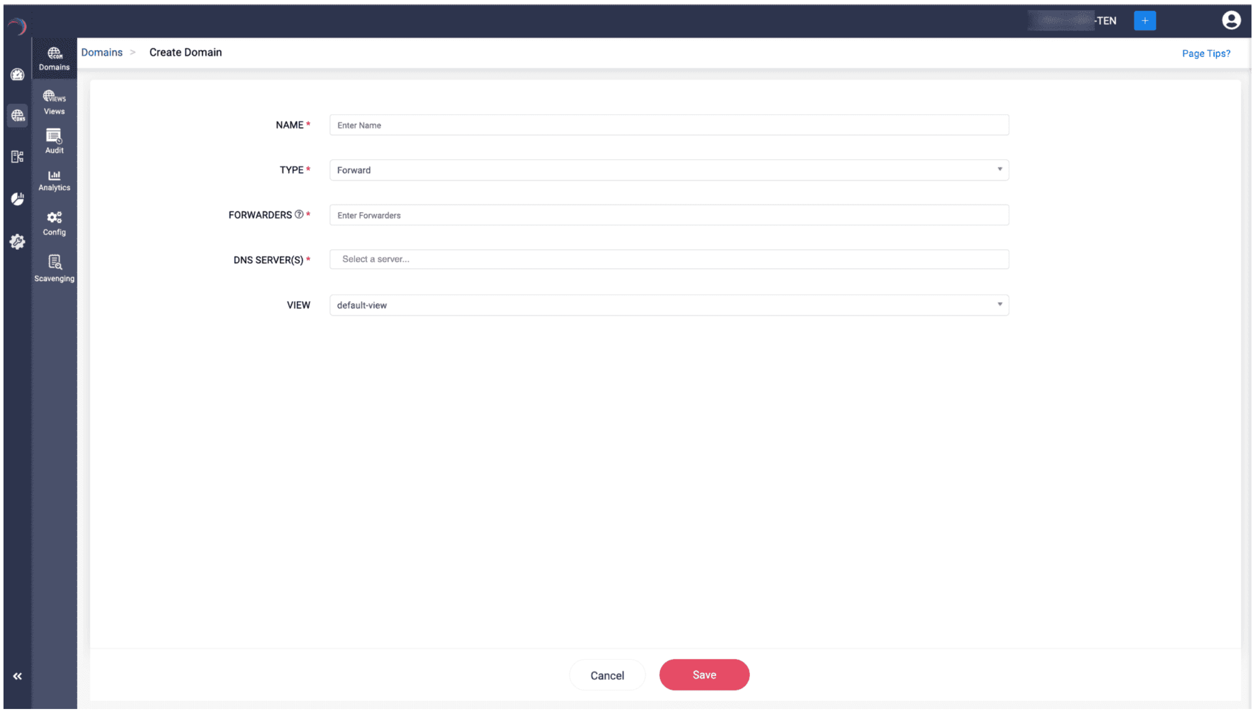
1. **Expand the IPv6 address:** Write out the full IPv6 address, filling in any omitted sections with zeros.
2. **Expanded IPv6 address:** 2001:0db8:85a3:0000:0000:0000:0000:0000
3. **Remove the bits beyond the prefix length:** Since the prefix is /48, you keep only the first 48 bits (which correspond to the first three blocks of the IPv6 address).
4. **Address Prefix:** 2001:0db8:85a3
5. **Reverse the nibbles:** Split the address into individual hexadecimal digits (nibbles), and write them in reverse order. Each hexadecimal digit corresponds to four bits.
6. **Reversed nibbles:** 3.a.5.8.8.b.d.0.1.0.0.2
7. **Add the .ip6.arpa domain:** Append this reversed string of nibbles to .ip6.arpa.
8. **Reverse Zone Name:** 3.a.5.8.8.b.d.0.1.0.0.2.ip6.arpa

Creating Forward Zones

DNS Zone Forwarding or Forward Zones in DDI Central refers to the process of redirecting queries for a specific DNS zone to another DNS server. This is typically used when a DNS server is not authoritative for a particular zone but is configured to pass queries for that zone to a server that is. This zone doesn’t contain actual DNS records but rather a forwarding instruction.

To create a new Forward Zone:

1. Select the **DNS** menu from the menu bar along the left side of the screen.From the submenus that appear, choose **Domains.**
2. Click on **Add Domain** and choose the type of the domain as **Forward Zone**.
3. The **Create Domain** page will appear as shown below with the following fields:



NAME * Enter Name

TYPE * Forward

FORWARDERS * Enter Forwarders

DNS SERVER(S) * Select a server...

VIEW default-view

NAME: Enter the name of the domain that you want to create.

TYPE: Select the type of the domain as Forward.

FORWARDERS: In this field, add the IP address of the DNS servers to forward queries to.

DNS SERVER(S): Here you select the server that will be authoritative for the domain you're creating.

VIEW: Allows you to select a DNS view, which can provide different data responses based on the source of the DNS query. You can configure views for forward zones in DDI Central. This would allow you to specify different forwarding behaviors based on the client making the request. For example, internal clients may be forwarded to an internal server, while external clients are forwarded to a public DNS service.

Note: Certain configurations like DDNS and DNSSEC do not exist for Forward zones.

DNS Firewall(FRW) Response Policy Zones (RPZ)

RPZ (Response Policy Zone) allows a nameserver to modify DNS responses based on policies. It's often used for implementing security measures, such as blocking known malicious domains, redirecting domains, or applying other customized policies. When a query matches an RPZ policy, the DNS server can return a different answer than what is stored in the authoritative data.

A DNS Firewall using Response Policy Zones (RPZ) is a powerful mechanism in DNS servers for implementing custom security policies. It's often used for implementing security measures, such as blocking known malicious domains, redirecting domains, or applying other customized policies. When a query matches an RPZ policy, the DNS server can return a different answer than what is stored in the authoritative data. It effectively allows DNS administrators to override DNS responses based on predefined policies, enhancing security and control over network traffic.

Here's what DNS Firewall RPZ does:

1. **Intercepts DNS Queries**: When a client device makes a DNS query, the DNS

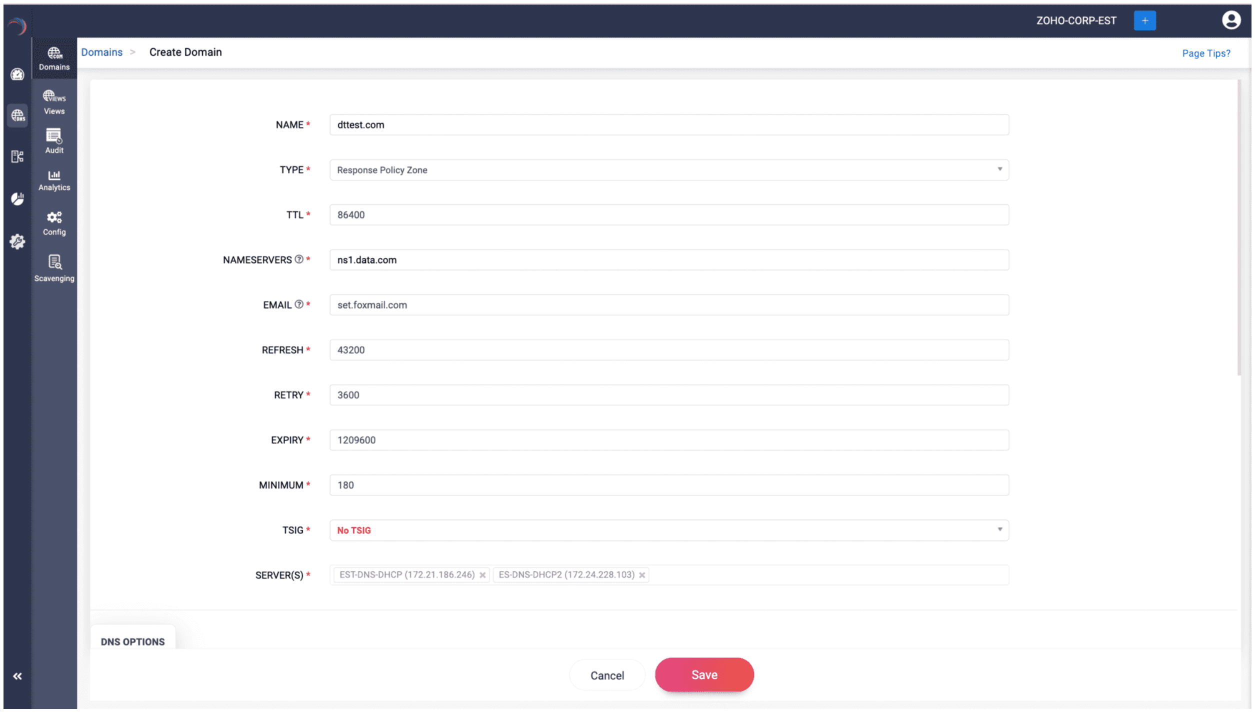
Firewall with RPZ intercepts this query. It then checks the requested domain name against a set of policy rules.

2. **Uses Policy Zones (RPZs)**: RPZs are special DNS zones that contain lists of domain names along with the policy actions to be applied to them. These can include known malicious domains, domains associated with phishing, spam, or domains that an organization wants to block for other reasons.
3. **Overrides Standard Responses**: Based on the RPZ rules, the DNS Firewall can modify the standard DNS response. For instance, if a client requests a domain that is listed in the RPZ as malicious, the DNS Firewall can redirect it to a safe page, block the request, or provide an alternate response.
4. **Prevents Access to Harmful Sites**: By redirecting or blocking requests to dangerous or unwanted domains, DNS Firewall RPZs protect users from malware, phishing attacks, and other cyber threats.
5. **Customizable and Flexible**: Administrators can create custom RPZs tailored to their organization's specific security needs. They can also subscribe to third-party RPZ feeds, which are regularly updated lists of harmful domains.
6. **Logging and Reporting**: DNS Firewall RPZs can log queries to blocked domains, providing valuable insights into attempted access to harmful sites and helping to identify patterns of malicious activity.

7. **Complements Other Security Measures:** While not a standalone security solution, DNS Firewall RPZ is an effective layer in a multi-layered security strategy, complementing firewalls, intrusion detection systems, and other security measures.

To create a RPZ in DDI Central:

- Go to **DNS→ Domains** . Click on **Add Domain** button on the top right corner.
- On the **Create Domain** page, Choose the type of the domain as **Response Policy Zone** (RPZ).



Domains > Create Domain

NAME *	dttest.com
TYPE *	Response Policy Zone
TTL *	86400
NAMESERVERS *	ns1.data.com
EMAIL *	set.foxmail.com
REFRESH *	43200
RETRY *	3600
EXPIRY *	1209600
MINIMUM *	180
TSIG *	No TSIG
SERVER(S) *	EST-DNS-DHCP (172.21.186.246) ES-DNS-DHCP2 (172.24.228.103)

DNS OPTIONS

- You can create the RPZ just like how an authoritative zone is created and the records are added. It is just you are controlling the local access to a publicly available suspicious domain with customized safe IPs.
- Configure various types of individual records for the RPZ offered by DDI Central, so that whenever a client in your network queries for any subdomain or domain configured as RPZ, it is the custom response you configured will be visible to the client in your network.
- DDI Central logs the queries to the RPZs and the different views configured for it. All the stats can be visualized under DNS→ Analytics page.

- An RPZ cannot have dynamic configurations. DDI Central enables you to apply a variety of DNS options to RPZs to have a granular control over the clients accessing it.
- DNS Firewall with RPZ is a proactive tool for enhancing network security by controlling and modifying DNS responses based on an organization's policies, thereby safeguarding the network from various online threats and undesirable content.

Managing DNS Resource Records

What are domain Resource Records (RR)?

Resource Records (RRs) are the fundamental information elements of the Domain Name System (DNS). Each RR defines a specific piece of information about the domain. Here are the general components of an RR:

1. **Name:** The domain name to which this record pertains.
2. **Type:** The type of the resource record, which defines the type of data contained in the record (e.g., A, MX, CNAME).
3. **TTL:** Time to Live, which specifies how long the record should be cached by DNS resolvers.
4. **RDATA:** Resource Data, the data of the resource record, varies according to the type (e.g., the IP address for an A record).

The combination of these elements within a DNS record allows DNS servers to accurately resolve queries by clients for various services related to a domain, such as website addresses, email servers, and service locations.

The following are the types of resource records supported by DDI Central:

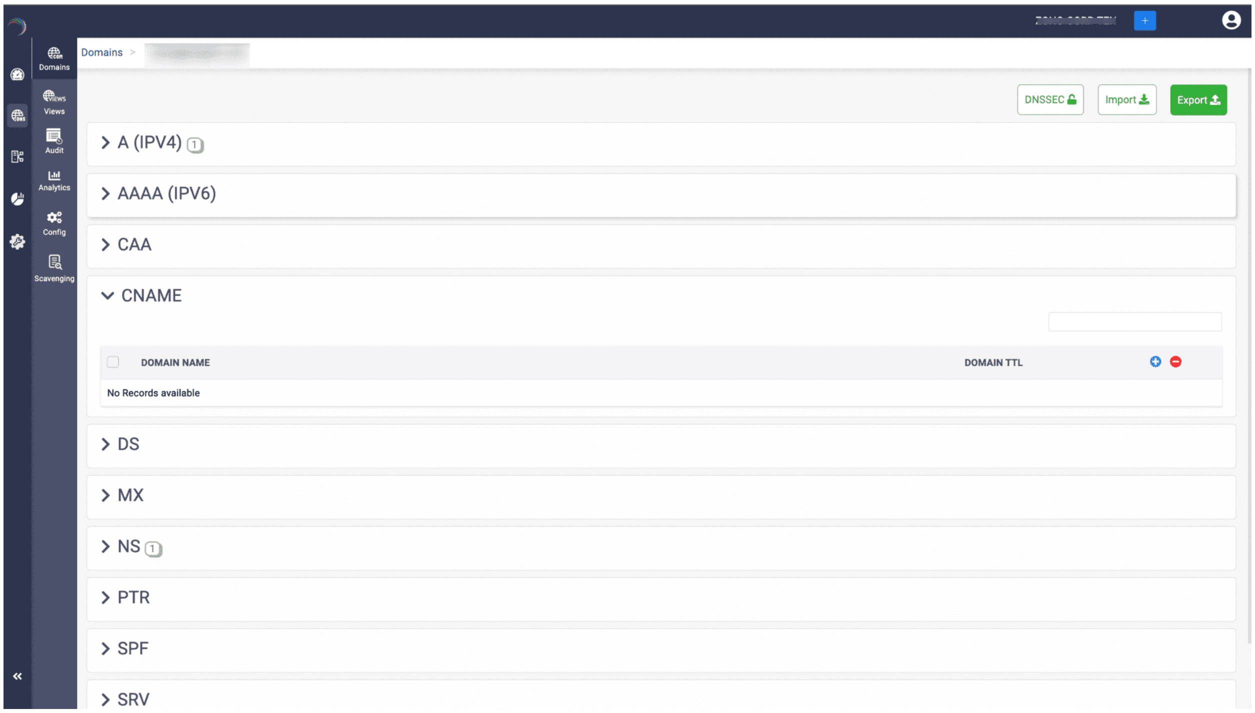
1. **A (IPv4):** Address record that maps a domain name to an IPv4 address, allowing a domain to be associated with a physical machine or resource on an IPv4 network.
2. **AAAA (IPv6):** Address record similar to the A record but for IPv6 addresses, mapping a domain name to an IPv6 address.
3. **CAA (Certificate Authority Authorization):** Specifies which certificate authorities (CAs) are allowed to issue certificates for a domain, enhancing security by restricting which CAs can issue certificates.
4. **CNAME (Canonical Name):** Redirects one domain name to another domain name, allowing multiple DNS records to map to the same server without specifying IP addresses.
5. **DS (Delegation Signer):** Holds the cryptographic signature for a DNSSEC-secured domain, which is used to securely delegate a subdomain to another DNS server or manager.

6. **MX (Mail Exchange):** Directs email to a mail server by specifying the server responsible for accepting email messages on behalf of a domain with a priority level.
7. **NS (Name Server):** Indicates the authoritative name server for a domain, which is responsible for presenting information about the domain's DNS zone.
8. **PTR (Pointer):** Used primarily for reverse DNS lookups, mapping an IP address (IPv4 or IPv6) to a domain name.
9. **SPF (Sender Policy Framework):** Defines which IP addresses are authorized to send email from a domain, helping to prevent email spoofing.
10. **SRV (Service Locator):** Specifies the location of servers for specified services, containing the hostname and port number for services such as VoIP, IM, etc.
11. **TXT (Text):** Allows administrators to insert arbitrary text into a DNS record. Often used to provide information to external sources, such as verification tokens for domain ownership or email security policies.

Creating resource records in DDI Central

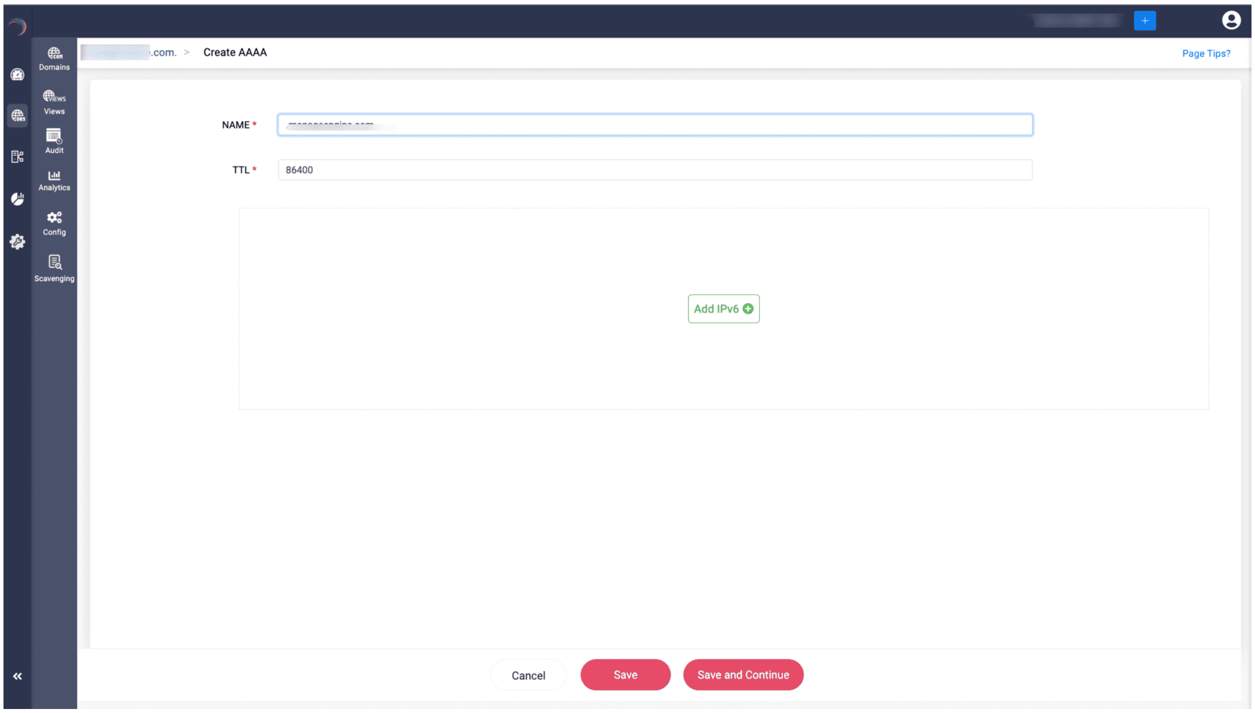
To add or update the resource records for a particular domain

1. Click on the domain name of your choice from the list of the domains that you intend to create or update DNS records.
2. This will take you inside that particular domain, displaying various types of records supported by DDI Central like A, AAAA, ANAME, CNAME, etc.



DOMAIN NAME	DOMAIN TTL
No Records available	

3. Select the relevant record type you'd like to configure for your domain and click on it.
4. To create a new record under the chosen record type, Click on the blue **Add** button at the extreme right corner of the table header under the chosen record type.
5. On the Create record type page, enter the subdomain or hostname.



[redacted].com. > Create AAAA

NAME * [redacted]

TTL * 86400

6. The **Time-To-Live(TTL)** attribute specifies the total number of seconds the local resolver ought to cache the response for a record before requesting a new one. The default is set to 86400 but can be modified as per your domain's requirements.
7. DDI Central enables you to configure multiple hosts to provide responses for a domain by clicking **Add IP**. To configure multiple hosts for a DNS record click on **Add IP** after each entry.

Importing and exporting zone data

DDI Central enables you to quickly create all of the records for your zone by importing a zone file in BIND format, that represents zone files in a text format.

Importing

To create DNS records by importing a zone file in BIND format:

1. Get the zone file exported and saved as BIND file from the other DNS server Make sure the zone file is in RFC-compliant format.
2. In the DDI Central, create a new zone or select a zone by clicking on its name.
3. Once you are inside the new zone, click the **Import** button in the top right corner.
4. Now you can import the zone file as a BIND file.

Click the **Import** button at the bottom to start importing. You may have to wait a few minutes for the records to be created as it depends on the number of records in your zone file.

Exporting

The same process is followed to export your zone files in DDI Central. Click the **Export** button in the top right corner. On clicking the **Export** button, the zone files are automatically downloaded as text files with the respective domain name in BIND format.

Managing DNS Views

What are DNS views?

DNS views or Domain views serve different responses to DNS queries based on various criteria, most commonly the source of the query or the host accessing it. This indicates that the DNS server can present one set of DNS information to one group of clients and a different set to another group, based on predefined views.

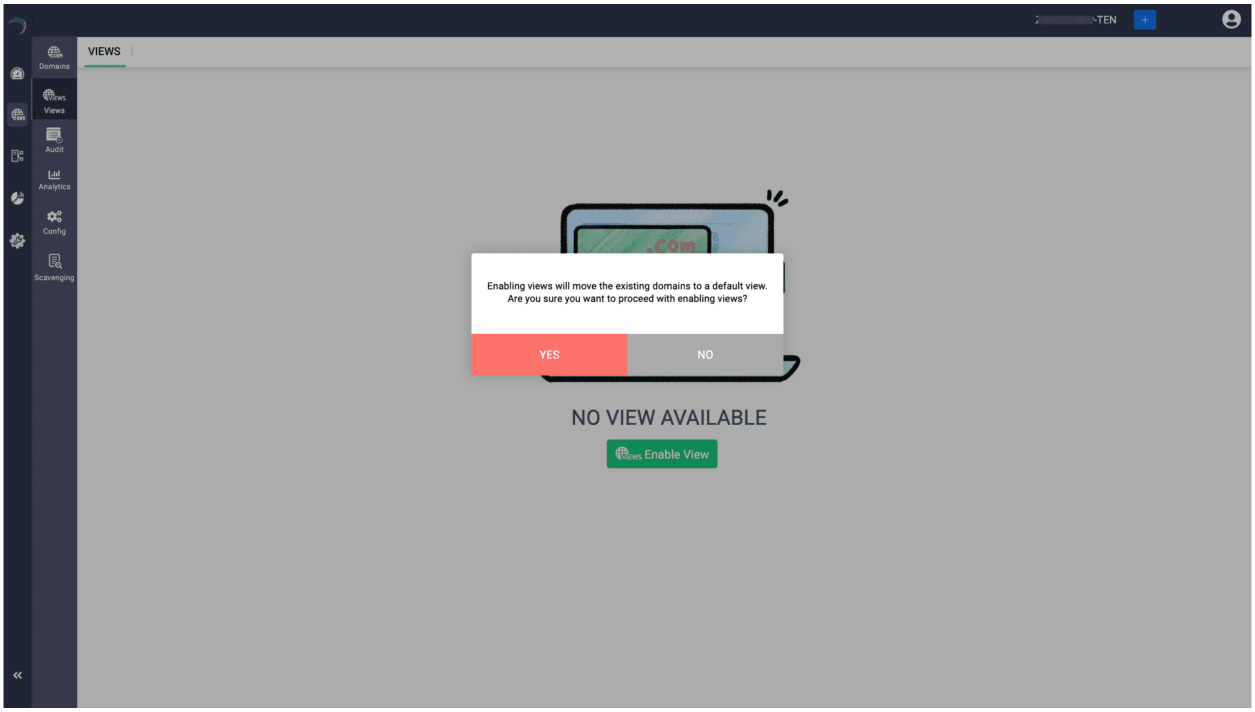
Configuring Named Views

DDI Central enables you to create multiple views and name them for better identification

Select the **DNS** menu the left menu bar. Now select **Views** from the submenus that appear on the inner menu bar.

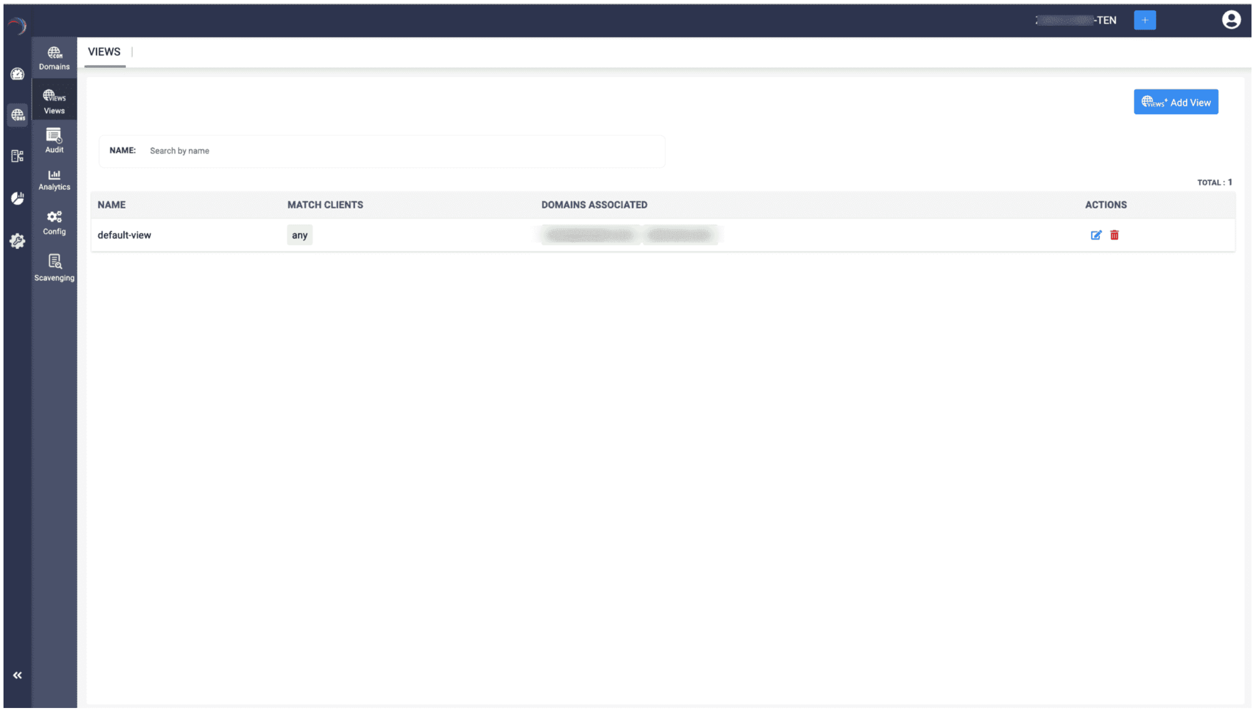
1. Select the **DNS** menu from the menu bar along the left side of the screen.

2. From the submenus that appear, choose **Views.**
3. If views have never been enabled for any of the existing zones, the screen will display the message **No View Available.**



Enabling views will move the existing domains to a default view.
Are you sure you want to proceed with enabling views?

4. Clicking on the **Enable View** button now will move all the existing domains to a **Default** view. Clicking **Yes** will create a Default View entry in the Views page. Here you can see under the match client field holding the value **any**, indicating this configuration will apply to all clients.



NAME: Search by name

TOTAL : 1

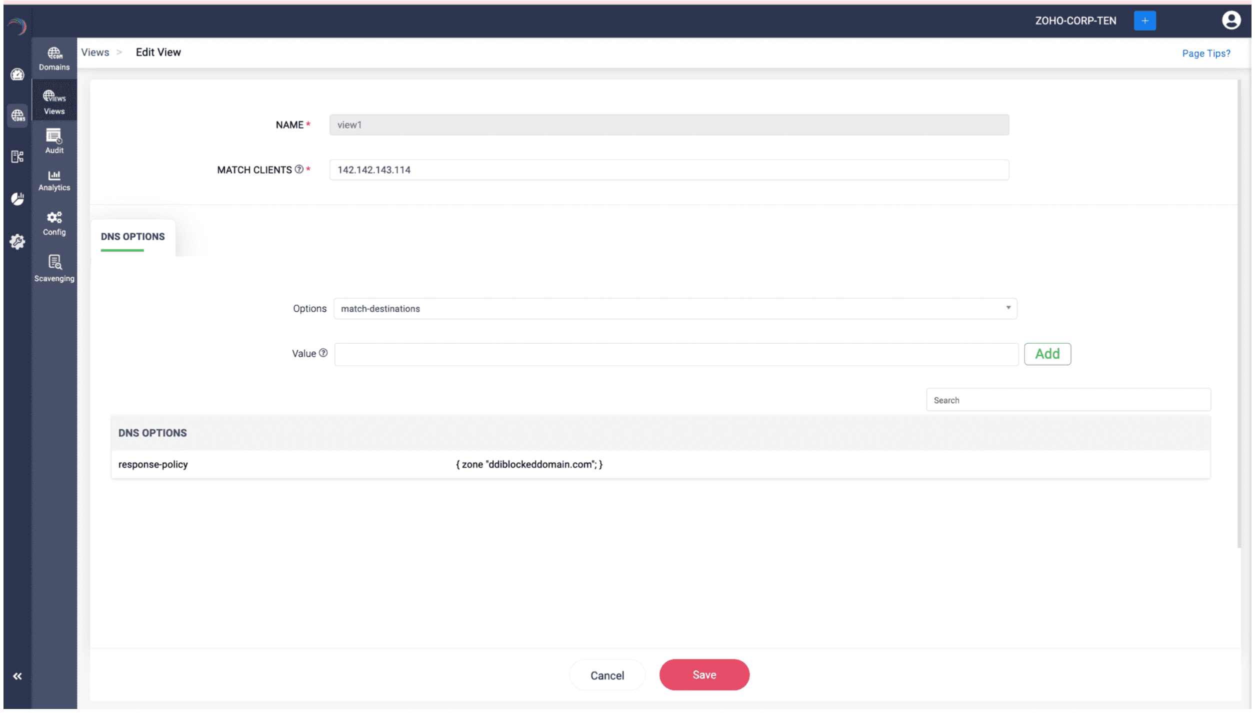
NAME	MATCH CLIENTS	DOMAINS ASSOCIATED	ACTIONS
default-view	any	[redacted]	

Default View

Default view for a domain refers to the unnamed or implicit view that is used when no specific view has been defined for a set of DNS queries. In the default view, BIND handles DNS queries as any standard DNS server would, without applying different rules or data sets based on the query source. It simply serves the DNS zones and records as configured.

In more complex configurations where named views are used, the default view can still exist. It would handle any queries that don't match the criteria of the named views. For example, if there are views for internal and external networks, the default view could handle queries from sources not covered by these specific views.

5. To create a named View, click on the **Add View** button in the top right corner. The **Create View** page appears.
6. Input a name for the new View in the designated field.
7. For the **Match Clients** field, input the list of IP addresses or specify named Access Control Lists (ACLs) as required.



NAME * view1

MATCH CLIENTS * 142.142.143.114

Options match-destinations

Value

DNS OPTIONS	
response-policy	{ zone "ddiblockeddomain.com"; }

8. Select the DNS options relevant for your selection of clients.
9. Once all the necessary information is provided and options are selected, click **Save**.

Popular DNS options for Domain views

1. **Match Clients**: Determines which clients (usually specified by IP address or network) the view applies to. It can be used to differentiate between internal

and external network clients.

2. **Match Destinations**: Similar to **match-clients**, but this matches on the destination address of the query instead of the source.
3. **Recursion**: Controls whether the server will perform recursive queries for clients using this view. This can be enabled for internal clients and disabled for external ones to prevent abuse.
4. **Forwarders**: Specifies different upstream servers for resolving DNS queries for clients that match the view. This can redirect query traffic based on client type or requested domain.
5. **Response Policy Zone (RPZ)**: Implements response policy service, allowing the server to modify or block DNS responses based on policies.
6. **Order of Precedence**: If a client matches multiple views, the order in which the views are defined in the configuration file determines which one finally applies.
7. **Allow-recursion, allow-query, allow-transfer**: These options within a view can be used to control which clients are allowed to perform recursive queries, make queries, or request zone transfers, respectively.
8. **DNSSEC Validation**: Controls whether DNSSEC validation is performed for the clients that match the view. This might be enabled for external views to

provide DNSSEC security for internet clients.

Managing Dynamic Domains

Table of Contents

Dynamic DNS (DDNS)

In DNS, a zone is a portion of the domain namespace, and the ability to create new zones dynamically is very essential, especially in environments where zones need to be added or removed without manual intervention.

Dynamic DNS (DDNS) can be enabled for various types of zones where it is necessary to dynamically update DNS records without manual intervention. Here are the types of zones where DDNS can be enabled:

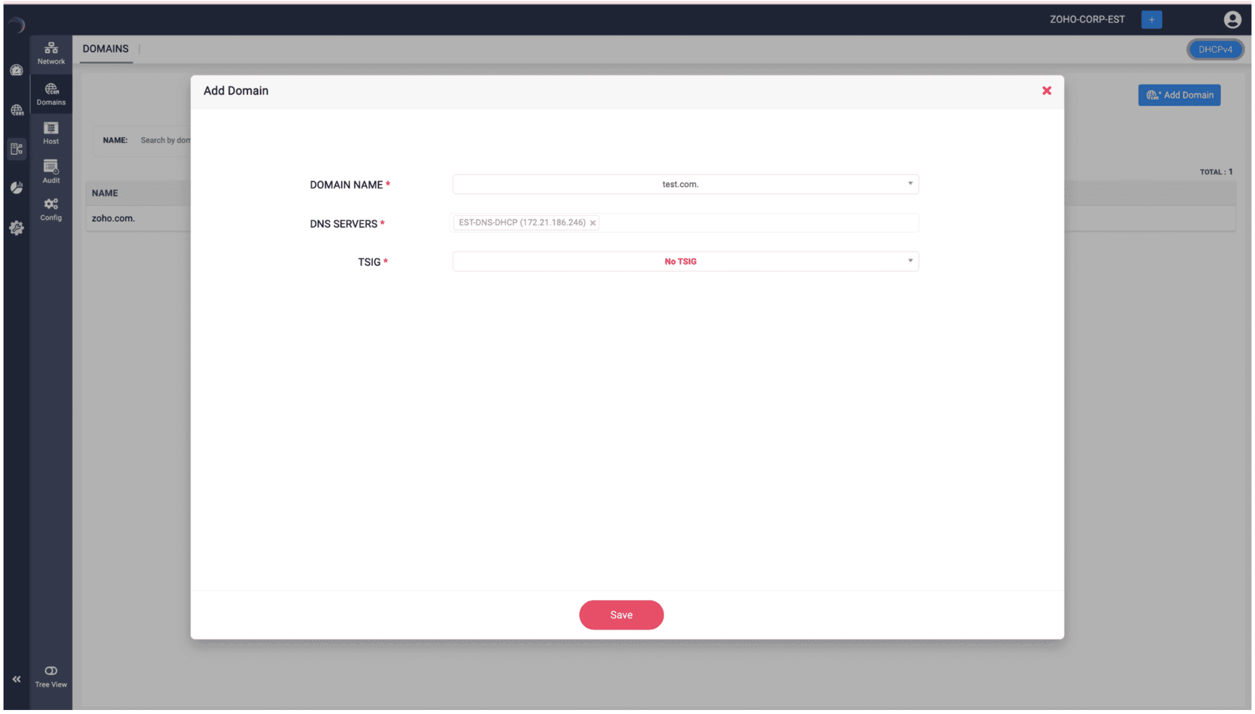
1. **Primary Zones**:
    - DDNS is most commonly enabled on primary zones. In a primary zone, the DNS records are stored and managed directly on the authoritative DNS server.
    - With DDNS, clients such as DHCP servers or DHCP clients can add, remove, or update DNS records in the primary zone dynamically. This is often used for automatically updating the DNS records of hosts as they obtain IP addresses from a DHCP server.
2. **Secondary Zones**:
    - While DDNS updates are not directly applied to secondary zones, secondary zones can receive updates indirectly via zone transfers from the primary zone.
    - When a DDNS update is made to the primary zone, the updated information is propagated to the secondary zones through the standard zone transfer mechanism (AXFR).
3. **Reverse Zones:**

    A reverse domain needs Dynamic DNS (DDNS) for several reasons, mainly related to the management of changing IP addresses and the need to maintain accurate reverse DNS records.

Enabling DDNS in ManageEngine DDI Central

To enable your DHCP server to dynamically provision IP address to your domains:

- When creating a domain via **Domains → Add Domain** , enable **DDNS** and assign a TSIG key for secure dynamic updates. If DDNS is not enabled at this stage, it cannot be activated later through the DNS menu.
- Alternatively, add domains that require dynamic configurations by navigating to **DHCP → Domains** .
- On the **Domains** page, Click on the **Add Domain** button in the top right corner.



DOMAIN NAME * test.com.

DNS SERVERS * EST-DNS-DHCP (172.21.186.246) ×

TSIG * No TSIG

Note: For your DHCP server to provision network parameters like IP addresses to your domains make sure your Domains and DHCP server are in the same network address.

Dynamic authoritative zones

Dynamic DNS (DDNS) allows the automatic updating of a DNS record when an IP address changes. This is often used for hosts with dynamic IP addresses assigned by a DHCP server.

Forward Mapping Dynamic Zones

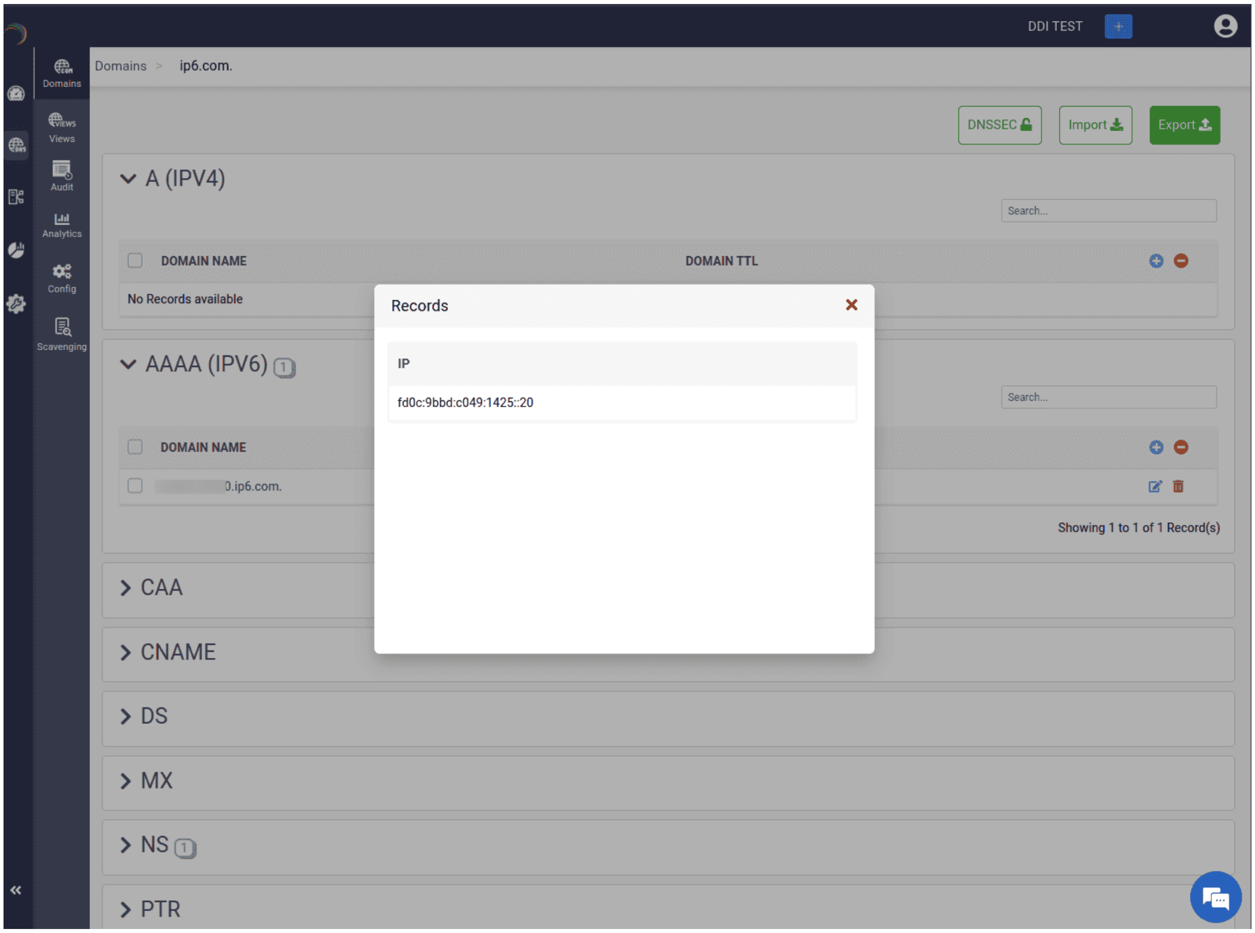
To create a dynamic Forward Mapping Authoritative Zone,

- Enable **DDNS** for the zone via one of the methods discussed above This would automatically enable the following DHCP options for the domain: **ddns-updates:true, ddns-update-style:interim, domainname**

- Now select the subnet that you would like to provision IP addresses to the domain.   Within that subnet specify the option domainname and specify the domain name you would like to create hosts to via dynamic updates like: **domainname: data.com.** This would enable the specific subnet to provision ip address to the hosts of that particular domain. Save the configurations.

- Now for an authoritative forward mapping zone, enabling DDNS would automatically create an A record with a host name assigned to it. the host name assumes variable IPs provisioned by the chosen subnet.

- **Example:** In the provided example, **ip6.com.** is an IPv6 based Forward mapping zone. Upon enabling DDNS, the DNS server will automatically generate a AAAA record for the zone. Within the AAAA record you can find the host name that holds dynamically variable IPv6 address provisioned by the DHCP server.



Domains > ip6.com.

A (IPV4)

□	DOMAIN NAME	DOMAIN TTL
No Records available		

AAAA (IPV6) 1

□	DOMAIN NAME
□	[redacted]0.ip6.com.

Showing 1 to 1 of 1 Record(s)

Records ×

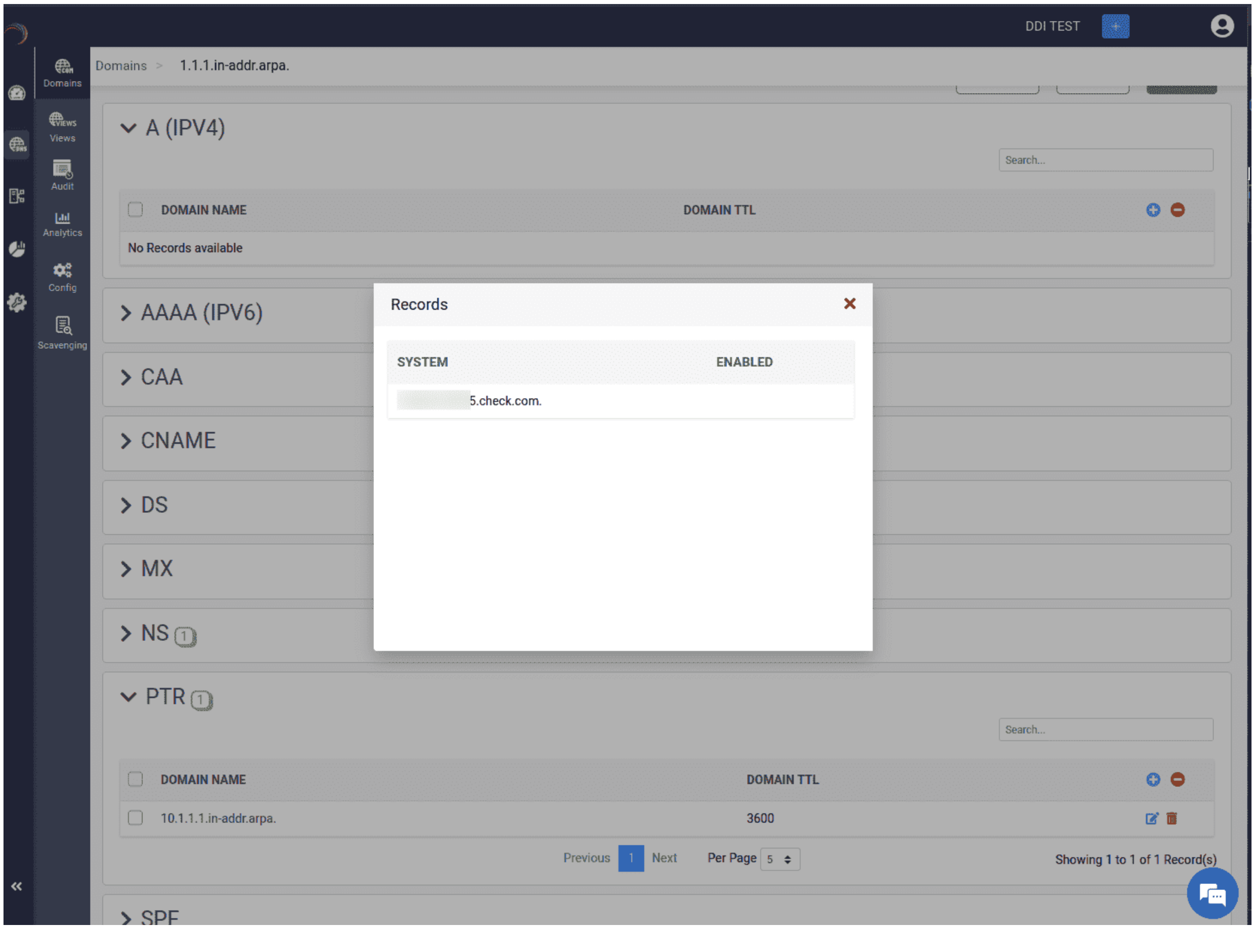
IP
fd0c:9bbd:c049:1425::20

Reverse dynamic Zones

For an authoritative reverse mapping zone, enabling DDNS will automatically generate PTR records that correspond to the hostnames within the authoritative A (or AAAA) records of a forward mapping zone.

Example:

In the provided example, **1.1.1.in-addr.arpa.** is an IPv4-based reverse zone. Upon enabling DDNS, the DNS server will automatically generate a PTR record within this reverse zone. These PTR records will correspond to the hostnames defined in the authoritative A records of the forward-mapping zone to which the reverse zone is linked.



DOMAIN NAME	DOMAIN TTL
No Records available	

Records

SYSTEM	ENABLED
[redacted]5.check.com.	

DOMAIN NAME	DOMAIN TTL
10.1.1.1.in-addr.arpa.	3600

Showing 1 to 1 of 1 Record(s)

The PTR record's name is the reverse of the IPv4 address appended to **in-addr.arpa.**

The reverse zone name **1.1.1.in-addr.arpa.** would be the reverse of the assigned IPv4 address and the corresponding PTR record within that zone points to **host.check.com.** which is the hostname of the system that was assigned the IPv4 address. This enables reverse DNS lookups, where querying the IPv4 address in reverse notation returns the hostname **host.check.com.**

Domain Scavenging

Domain scavenging, more commonly known as DNS scavenging, refers to the process of cleaning up stale DNS records that remain idle over time in the DNS database. This mechanism is typically used to automatically remove outdated records, such as those for IP addresses no longer in use, and can help prevent DNS-related issues such as name resolution conflicts and unwanted bloats in the DNS database. This practice is essential for maintaining an accurate and efficient Domain Name System, particularly in environments where IP addresses and host configurations frequently change. Here is an overview of domain scavenging:

1. **Purpose:** Scavenging helps remove stale resource records from DNS, which might no longer be valid due to changes in network configuration, such as decommissioned servers, expired DHCP leases, or devices that are no longer part of the network.
2. **Automated Cleanup:** The scavenging process is often automated. DNS servers are configured to periodically scan the DNS records and remove those

that are outdated or no longer in use.

3. **Aging and Refresh:** Scavenging relies on two key concepts: the aging of records and the refresh of these records. When a DNS record is created or updated, it’s given a timestamp. If this record is not refreshed or updated within a certain period (the aging time), it's considered stale.
4. **Scavenging Interval:** Administrators set a scavenging interval, which is the frequency at which the DNS server checks for stale records. If a record is older than the aging period by the time of this check, it will be notified to the user through scavenge reports.
5. **Prevents DNS Bloat:** Regular scavenging prevents the DNS database from becoming bloated with unnecessary records, which can slow down DNS query responses and lead to inefficiencies in network operation.
6. **Dynamic DNS Environments:** Scavenging is particularly important in dynamic DNS environments where DHCP is used to assign IP addresses. As clients come and go, their DNS records need to be updated or removed to reflect their current status.
7. **Careful Configuration:** Incorrectly configured scavenging can lead to the premature deletion of active DNS records. It’s important to set appropriate aging and scavenging intervals to avoid disrupting network services.

8. **Improves Network Security:** By removing outdated records, scavenging can also enhance network security. Stale DNS entries can be a security risk, as they may point to unused IP addresses that could be exploited by malicious actors.

Domain scavenging is a crucial maintenance activity for any network that uses DNS and DHCP. It helps ensure that the DNS database remains up-to-date and free from clutter, enhancing both the performance and security of the network.

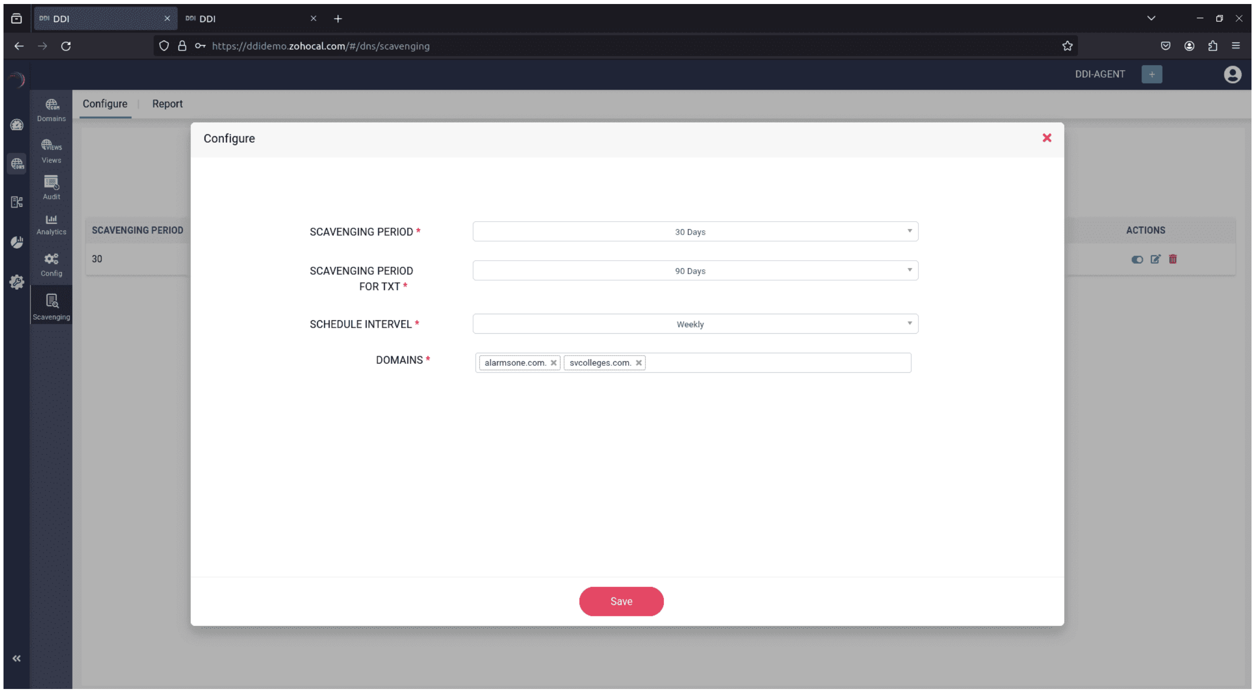
Configuring domain scavenging in DDI Central

To configure Domain scavenging in DDI Central:

Note: Scavenging can be configured only for A, AAAA. CNAME, PTR and TXT records, as only these records are capable of receiving dynamic updates.

- Select the **DNS** menu from the menu bar along the left side of the screen. From the submenus that appear, choose **Scavenging.**

- First configure scavenging for your DNS infrastructure under the Configure tab.
- On the **Configure** window that appears, the top field **SCAVENGING PERIOD** is meant for all the A, AAAA. CNAME, PTR of the domains selected. This is the duration after which a DNS record becomes eligible for scavenging if it has not been refreshed. If the DNS record still remains un refreshed after this period, DNS server considers the record stale and eligible for deletion and put up in the report for the user to delete or reclaim it .
- **SCHEDULE INTERVAL:** This dropdown menu allows the user to select how often the scavenging process should be scheduled to run. The options could range from daily to monthly intervals.
- **DOMAINS:** Here, you can specify which domains are subject to the scavenging process. Click Save.



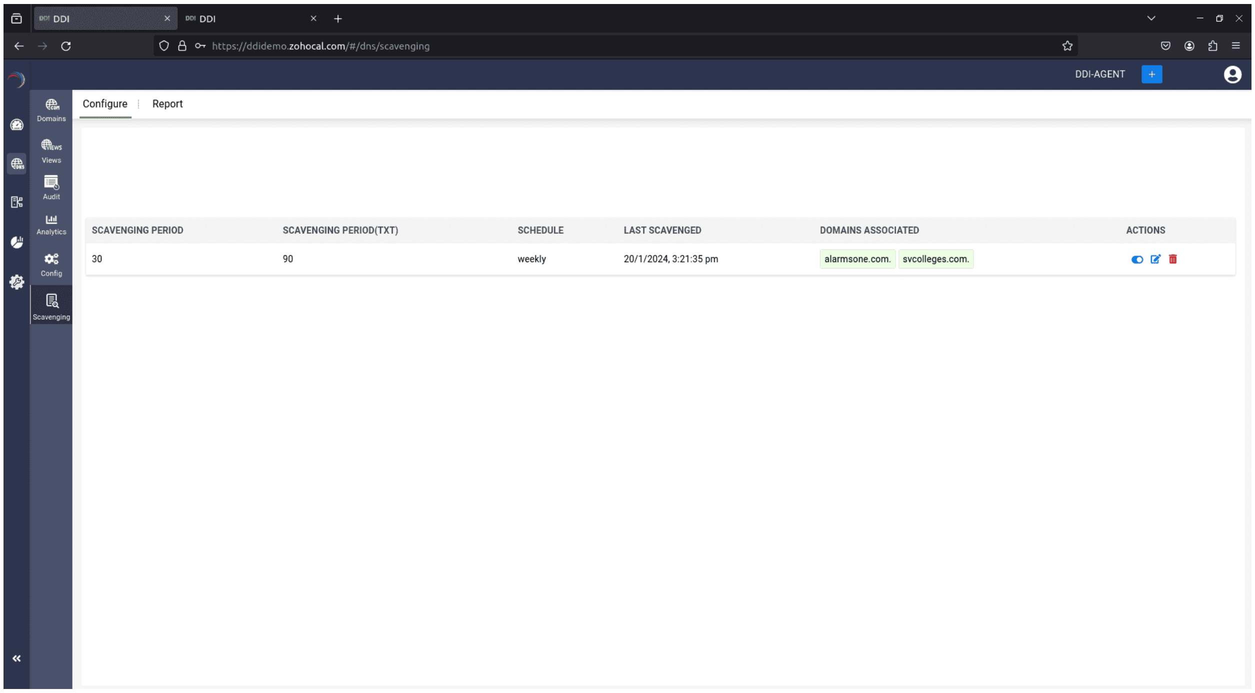
Configure

SCAVENGING PERIOD * 30 Days

SCAVENGING PERIOD FOR TXT * 90 Days

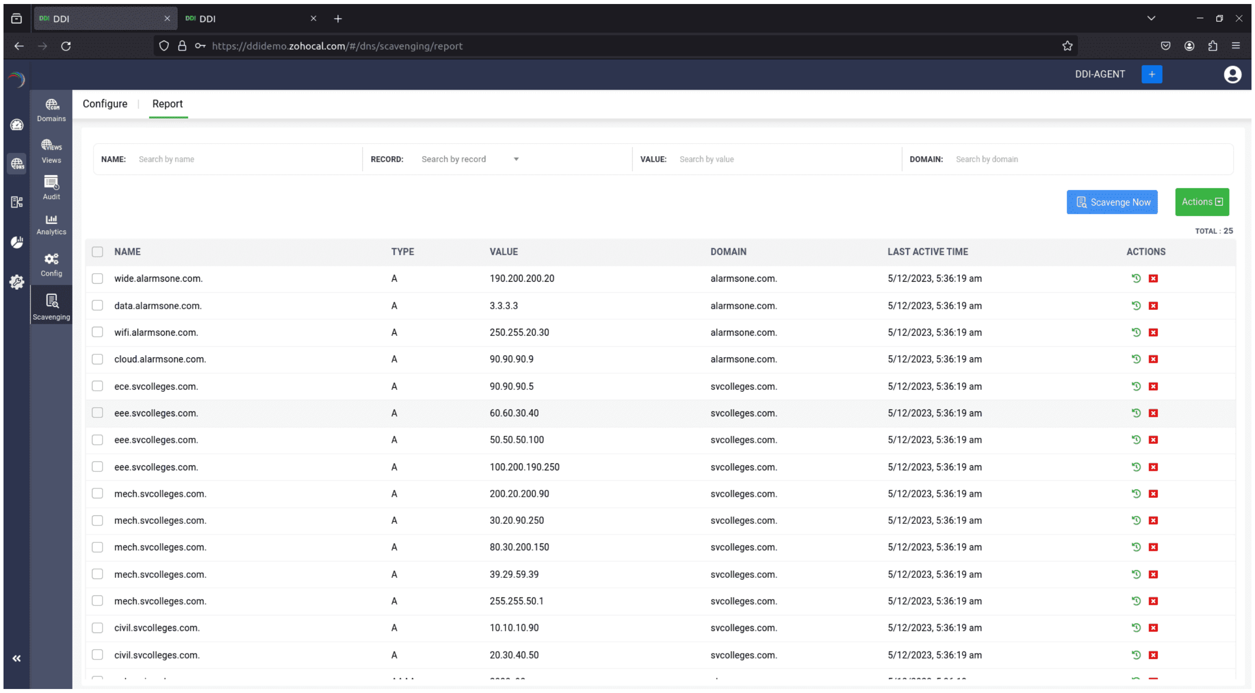
SCHEDULE INTERVEL * Weekly

DOMAINS * alarmsone.com. ✕ svcolleges.com. ✕



SCAVENGING PERIOD	SCAVENGING PERIOD(TXT)	SCHEDULE	LAST SCAVENGED	DOMAINS ASSOCIATED	ACTIONS
30	90	weekly	20/1/2024, 3:21:35 pm	alarmsone.com. svcolleges.com.	

- Once the scavenging is configured, the Configure page summarizes your selections and shows the domains it targets to scavenge.



NAME: Search by name RECORD: Search by record VALUE: Search by value DOMAIN: Search by domain

TOTAL : 25

NAME	TYPE	VALUE	DOMAIN	LAST ACTIVE TIME	ACTIONS
wide.alarmsone.com.	A	190.200.200.20	alarmsone.com.	5/12/2023, 5:36:19 am	
data.alarmsone.com.	A	3.3.3.3	alarmsone.com.	5/12/2023, 5:36:19 am	
wifi.alarmsone.com.	A	250.255.20.30	alarmsone.com.	5/12/2023, 5:36:19 am	
cloud.alarmsone.com.	A	90.90.90.9	alarmsone.com.	5/12/2023, 5:36:19 am	
ece.svcolleges.com.	A	90.90.90.5	svcolleges.com.	5/12/2023, 5:36:19 am	
eee.svcolleges.com.	A	60.60.30.40	svcolleges.com.	5/12/2023, 5:36:19 am	
eee.svcolleges.com.	A	50.50.50.100	svcolleges.com.	5/12/2023, 5:36:19 am	
eee.svcolleges.com.	A	100.200.190.250	svcolleges.com.	5/12/2023, 5:36:19 am	
mech.svcolleges.com.	A	200.20.200.90	svcolleges.com.	5/12/2023, 5:36:19 am	
mech.svcolleges.com.	A	30.20.90.250	svcolleges.com.	5/12/2023, 5:36:19 am	
mech.svcolleges.com.	A	80.30.200.150	svcolleges.com.	5/12/2023, 5:36:19 am	
mech.svcolleges.com.	A	39.29.59.39	svcolleges.com.	5/12/2023, 5:36:19 am	
mech.svcolleges.com.	A	255.255.50.1	svcolleges.com.	5/12/2023, 5:36:19 am	
civil.svcolleges.com.	A	10.10.10.90	svcolleges.com.	5/12/2023, 5:36:19 am	
civil.svcolleges.com.	A	20.30.40.50	svcolleges.com.	5/12/2023, 5:36:19 am	

- Once it detects stale records, the records will be displayed in the reports section. Depending on the current state of the records, the user can delete it or reclaim those records.

DNS64

DNS64 is a mechanism used in IPv6 networks to facilitate communication between IPv6-only clients and IPv4-only servers. This is especially important in the context of the ongoing transition from IPv4 to IPv6. Since these are two different protocols and not directly compatible, mechanisms like DNS64 are essential for interoperability. This is accomplished in ME DDI Central using the DNS option **dns64**.

Configuring dns64 option involves setting up a DNS server that can synthesize AAAA records (IPv6 addresses) from A records (IPv4 addresses) when no native AAAA records are available. This configuration is typically done on a DNS server that's designed to support DNS64 functionality.

DNS64 works by prefixing an IPv4 address with a specific IPv6 prefix. This prefix is usually a /96 prefix, which leaves room for the entire IPv4 address. A common prefix used is 64:ff9b::/96, but you can use a different one if required.

Example in ISC BIND format

```
options {
  // other options...
  dns64 64:ff9b::/96 {
    clients { any; };
    // more specific configurations if needed
  };
};
```

In this configuration:

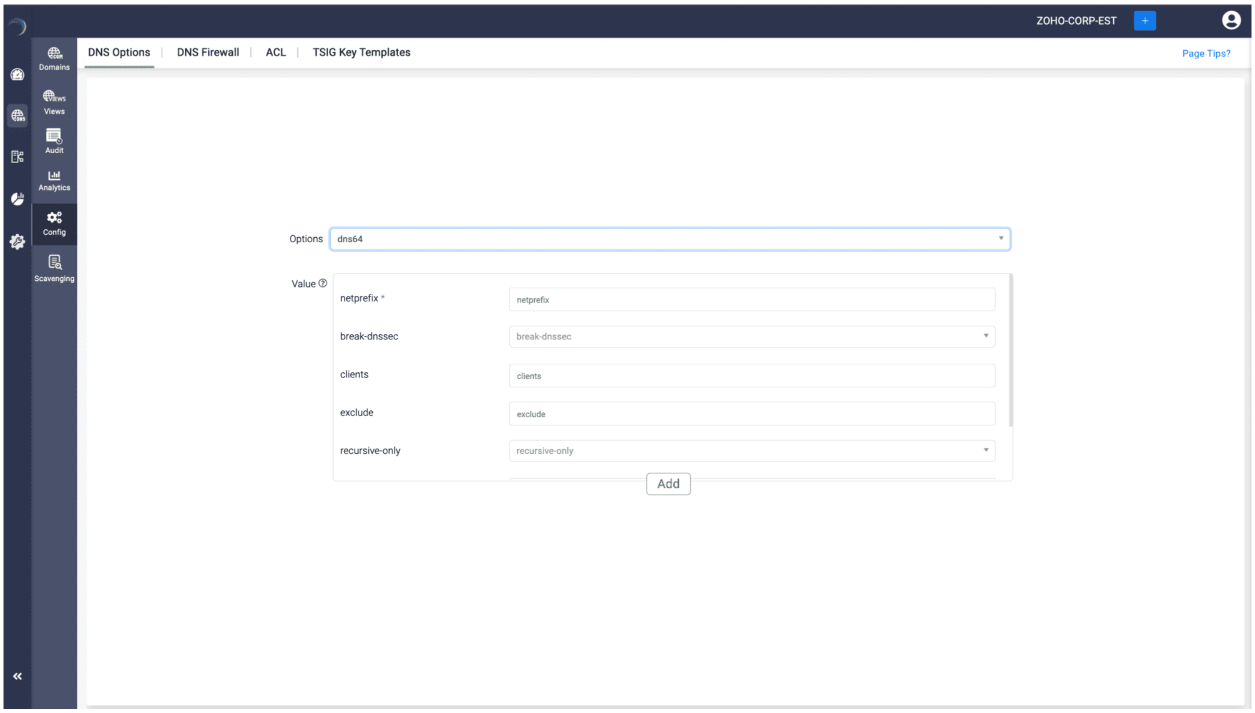
- dns64 64:ff9b::/96 specifies the DNS64 prefix.
- clients { any; }; indicates that DNS64 translation should be applied to requests from all clients. You can restrict this to certain clients or networks if necessary.

Configuring DNS64 in DDI Central

To configure it select **DNS→ Config→ DNS Options**

- On the **DNS options** page, click on the Options drop down box to search for dns64 option.
- The dns64 option appears with all its attributes. Fill in the values for each

attribute and click **Save**.



Options: dns64

Value

netprefix *	netprefix
break-dnssec	break-dnssec
clients	clients
exclude	exclude
recursive-only	recursive-only

Here are the attributes within the dns64 option:

netprefix: This is the IPv6 prefix that is used to synthesize AAAA records. It's typically a /96 prefix, and the IPv4 address is appended to this prefix to create the IPv6 address in the synthesized AAAA record. Example value: dns64 64:ff9b::/96 { ... };

break-dnssec: This attribute, when set, allows DNS64 to synthesize AAAA records even for DNSSEC-signed domains. This can potentially break DNSSEC validation, as the synthesized AAAA record does not actually exist in the DNS. Example value: break-dnssec yes;

clients: Specifies for which clients the DNS64 rule applies to. You can define a match list of IP addresses or subnets from which the clients are allowed to use DNS64. Example value: clients { any; };

exclude: Used to define IP address ranges for which DNS64 should not synthesize AAAA records. This is useful for networks or hosts that are reachable over native IPv6. Example value: exclude { 2001:db8::/32; };

recursive-only: When set to yes, DNS64 synthesis is performed only for recursive queries. It won’t synthesize records for authoritative answers. Example value: recursive-only yes;

mapped: This attribute controls whether DNS64 synthesis is applied to domains that have both A and AAAA records. If set, it will synthesize AAAA records even if native AAAA records exist. Example value: mapped yes;

suffix: This optional attribute specifies a suffix to append to the synthesized IPv6 address. It's rarely used as the default behavior (without a suffix) is generally preferred. Example value: suffix ::1;

Each of these attributes fine-tunes the behavior of DNS64, allowing for customization based on specific network needs, especially in environments transitioning to IPv6 or operating in dual-stack (IPv4 and IPv6) scenarios. It's important to configure these settings carefully to ensure proper network functionality and to avoid unintended disruptions, particularly with regard to DNSSEC and native IPv6 connectivity.

Click **Save** to see the dns64 option in effect.

DNS Security Management

DNS Security Management involves implementing and maintaining measures to protect the Domain Name System (DNS), a critical component of your network infrastructure. This includes safeguarding against DNS-based threats like cache poisoning, DDoS attacks, and domain hijacking. Effective DNS security management encompasses using secure protocols like DNSSEC to ensure data integrity, implementing response rate limiting (RRL) to mitigate DDoS attacks, and regularly monitoring and auditing DNS traffic for anomalies.

Additionally, it involves ensuring proper configuration of DNS servers, keeping software updated, and using access control lists (ACLs) to restrict unauthorized access. Such comprehensive security practices are essential to maintain the reliability and trustworthiness of DNS services, crucial for the seamless operation of internet-based communications and services.

DNSSEC

DNSSEC, short for Domain Name System Security Extensions, is a suite of specifications for securing certain kinds of information provided by the Domain Name System (DNS). It is designed to protect against a range of DNS attacks such as cache poisoning, where a DNS query is redirected from a legitimate to a malicious site.

Why You Need DNSSEC

1. **Integrity**: DNSSEC ensures that the information you receive from a DNS query is exactly what the domain owner entered, with no modifications en route, guaranteeing data integrity.
2. **Authentication**: It provides a means to verify that the source of your DNS data is legitimate (authenticity) and not a malicious actor trying to intercept or manipulate DNS queries.
3. **Trust**: By building a chain of trust from the root DNS servers down to the

specific DNS entry for a domain, DNSSEC prevents attackers from inserting malicious DNS data into the responses to DNS queries.

What DNSSEC Does

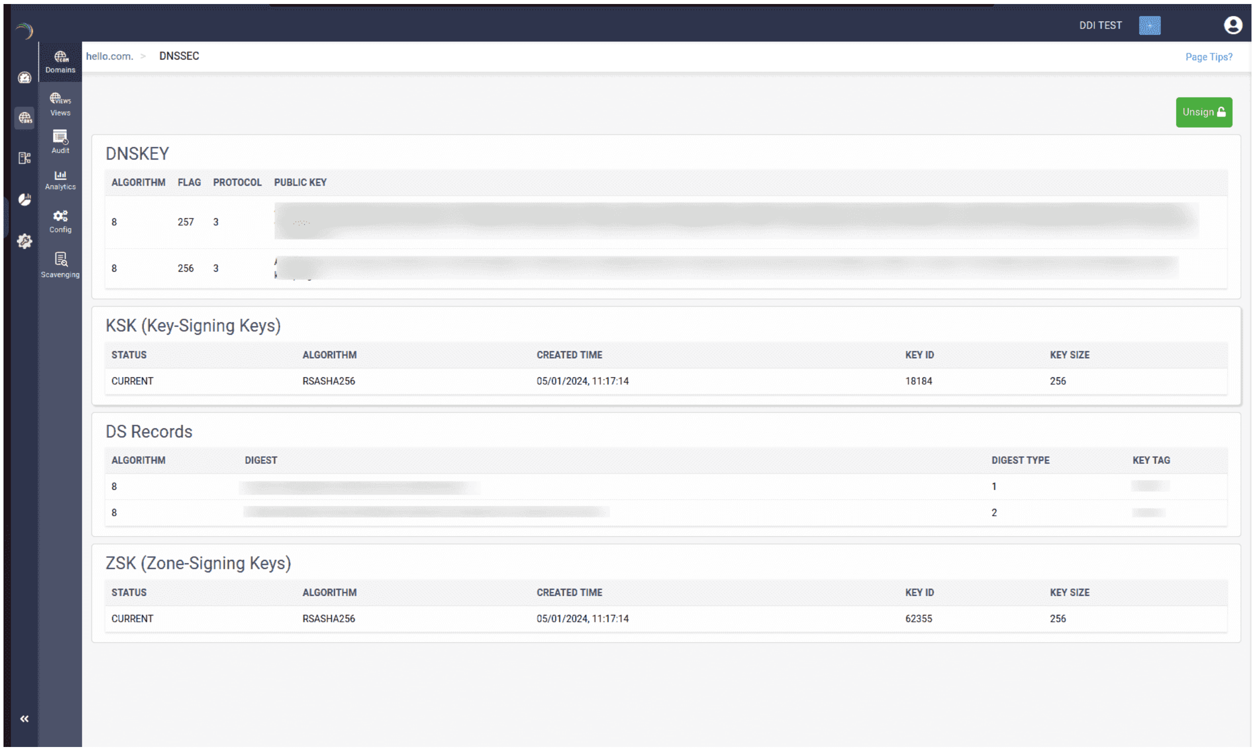
- **Digital Signing**: DNSSEC works by digitally signing these records for DNS lookup using public-key cryptography. Each DNS zone has a private key that is used to sign the zone's DNS records, and a public key that is used to validate the signatures.
- **Chain of Trust**: Starting from the DNS root zone, each level of the DNS hierarchy has its own pair of keys and signs the keys for the level below it, creating a chain of trust down to the individual DNS record level.
- **Non-Repudiation**: Because of the digital signatures, DNS data cannot be tampered with without detection, providing non-repudiation, which is the assurance that someone cannot deny the validity of something.
- **Validation**: Resolving name servers, which are configured to use DNSSEC, can then validate these signatures using the public key, ensuring that the DNS data has not been modified.

DNSSEC is necessary to combat the inherent vulnerabilities in the traditional DNS system that make it susceptible to various forms of attack. By providing a way to verify the authenticity of DNS data, DNSSEC adds a layer of security to the domain name lookup process.

Configuring DNSSEC in DDI Central

To enable DNSSEC:

- DDI Central deploys DNSSEC signing to sign the DNS responses of a particular zone. Navigate to the domain of your choice and click on the **DNSSEC** button with the icon of an opened lock on the top right corner.
- Click on the **Sign** button.



hello.com. > DNSSEC

DNSKEY

ALGORITHM	FLAG	PROTOCOL	PUBLIC KEY
8	257	3	[redacted]
8	256	3	[redacted]

KSK (Key-Signing Keys)

STATUS	ALGORITHM	CREATED TIME	KEY ID	KEY SIZE
CURRENT	RSASHA256	05/01/2024, 11:17:14	18184	256

DS Records

ALGORITHM	DIGEST	DIGEST TYPE	KEY TAG
8	[redacted]	1	[redacted]
8	[redacted]	2	[redacted]

ZSK (Zone-Signing Keys)

STATUS	ALGORITHM	CREATED TIME	KEY ID	KEY SIZE
CURRENT	RSASHA256	05/01/2024, 11:17:14	62355	256

- After the domain of your choice is signed successfully, a DNSKEY record, a DS record are created automatically within the zone. DDI Central displays the DNSSEC key tag, algorithm, digest type, digest under **DS Records**, flags along with a public key, Key Signing Keys(KSK), Zone Signing Keys(ZSK) associated with the particular zone. Copy these details in your clipboard as you'll need these details to update your registrar.
- You can also see the **Unsign** button with a closed lock on the top right corner

indicating DNSSEC is enabled for the zone.

- Once DNSSEC signing is enabled on a zone and the appropriate information is given to your registrar, DNSSEC supporting resolvers will begin to validate DNS responses returned by your on-prem nameservers.
- You can also revoke DNSSEC for a particular zone by clicking the **Unsign** button at the top right corner.

Response Rate Limiting (RRL)

RRL, or Response Rate Limiting, is a security feature implemented in DNS servers to mitigate the impact of Distributed Denial of Service (DDoS) attacks, particularly DNS amplification attacks. It works by limiting the rate at which DNS responses are sent from a server to a particular client or set of clients.

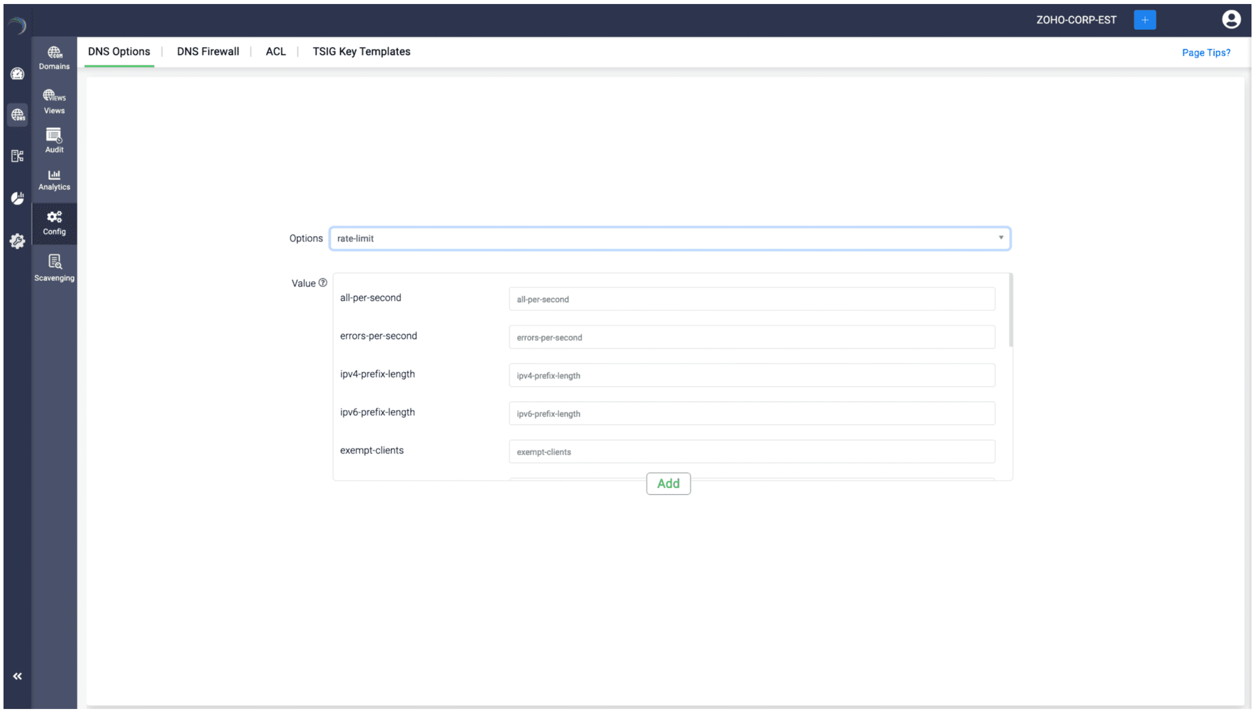
When a DNS server receives an unusually high volume of requests, possibly as part of an attack, RRL kicks in to restrict the number of responses sent back to any given requester over a specified period. This helps to prevent the server from being used as a tool in amplification attacks, where large numbers of responses are sent to a victim's network, overwhelming its bandwidth. It is implemented in ME DDI Central using the **rate-limit** DNS option.

Configuring RRL in DDI Central

To configure it select **DNS→ Config→ DNS Options**

On the **DNS options** page, click on the Options drop down box to search for rate-limit option.

The rate-limit option appears with all its attributes. Fill in the values for each attribute and click **Save**.


Options: rate-limit

Here are explanations for various attributes of the **rate-limit** option:

1. **all-per-second:** Limits the total number of all responses (regardless of type) per second.
2. **errors-per-second:** Limits the number of error responses (like SERVFAIL) per

second.

3. **ipv4-prefix-length and ipv6-prefix-length:** Define the subnet mask length for aggregating IPv4 and IPv6 addresses. This dictates how broadly the rate limiting is applied across a range of IP addresses.

   For example, **ipv4-prefix-length** of 24 means that the server will apply rate limits to all addresses in each /24 subnet as a group. Therefore, all requests originating from the 192.168.1.0/24 subnet, for instance, would be collectively subject to the specified rate limit.

4. **exempt-clients:** Specifies clients (usually by IP address or subnet) that are exempt from rate limiting. This is often used for trusted networks.
5. **log-only:** When enabled, BIND logs the rate-limited responses without actually enforcing the limits. This is useful for testing the configuration.
6. **max-table-size:** The maximum number of entries in the rate-limiting table. A larger table can track more clients but requires more memory.
7. **min-table-size:** The minimum size of the rate-limiting table.
8. **nodata-per-second:** Limits the number of responses per second that result in NODATA (no error but no data).
9. **nxdomains-per-second:** Limits the number of NXDOMAIN (non-existent domain) responses per second.

10. **qps-scale:** A factor by which to scale the queries per second calculation. It can be used to adjust the sensitivity of rate limiting.
11. **referrals-per-second:** Limits the number of DNS referral responses per second.
12. **responses-per-second:** Limits the number of identical responses per second from a single IP address or subnet.
13. **slip:** Defines the behavior when a rate limit is exceeded. Typically, every nth response will be truncated.

    The **slip** setting determines how often the DNS server will send a truncated response instead of dropping the response entirely when rate limiting is in effect. A truncated response is a response that tells the querying client that it should retry the request over TCP instead of UDP. Since TCP connections require more resources to establish, attackers are less likely to use them, making DDoS attacks less effective.

    Here's a breakdown of the slip option:

    1. **Value 0:** The server will drop all responses that exceed the rate limit.
    2. **Value 1:** The server will send a truncated response for every request that exceeds the rate limit.

Values 2 and higher: The server will send truncated responses for one out of every 'slip' number of requests that exceed the rate limit. For example, if the slip value is set to 2, then the server will send a truncated response for every second request that exceeds the limit.

14. **window:** The time period, in seconds, over which BIND calculates the rate of identical responses for rate limiting.

Example

```
rate-limit {
   responses-per-second 10;
   window 5;
   ipv4-prefix-length 24;
   ipv6-prefix-length 48;
   slip 2;
   nxdomains-per-second 5;
   nodata-per-second 5;
   errors-per-second 2;
   all-per-second 20;
   max-table-size 100000;
```

```
    exempt-clients { 192.168.0.0/24; };
    log-only yes;
};

```

From the above configuration example, ME DDI Central will limit identical DNS responses to 10 per second over a 5-seconds window. If the limit exceeds, DDI Central will start sending truncated responses every second request (split=2). The local network (192.168.1.0/24) is exempt from these limits, and the log-only setting means the limits will be logged but not enforced, which is helpful for initial testing.

Domain Blocking Using DNS Firewall

Domain blocking using a DNS Firewall is a security measure that prevents users from accessing specific websites or domains by intercepting DNS queries and filtering out requests to undesired or malicious domains. When a user attempts to visit a website, their device sends a DNS query to resolve the domain name into an IP address. A DNS Firewall steps in at this point to screen the query against a set of predefined security rules or blacklists.

The DNS Firewall first intercepts DNS queries from client devices on the network before they reach the internet. It analyzes the domain name requested against a database of blocked or suspicious domain names. If the domain is on the block list, the DNS Firewall applies the configured policy, which typically involves preventing the resolution of the domain name into an IP address. Finally, the DNS Firewall redirects the query to a safe page. DDI Central's Firewall based Domain Blocking measure blocks collections of recognized malicious domains and directs the users to a safe customized IP address.

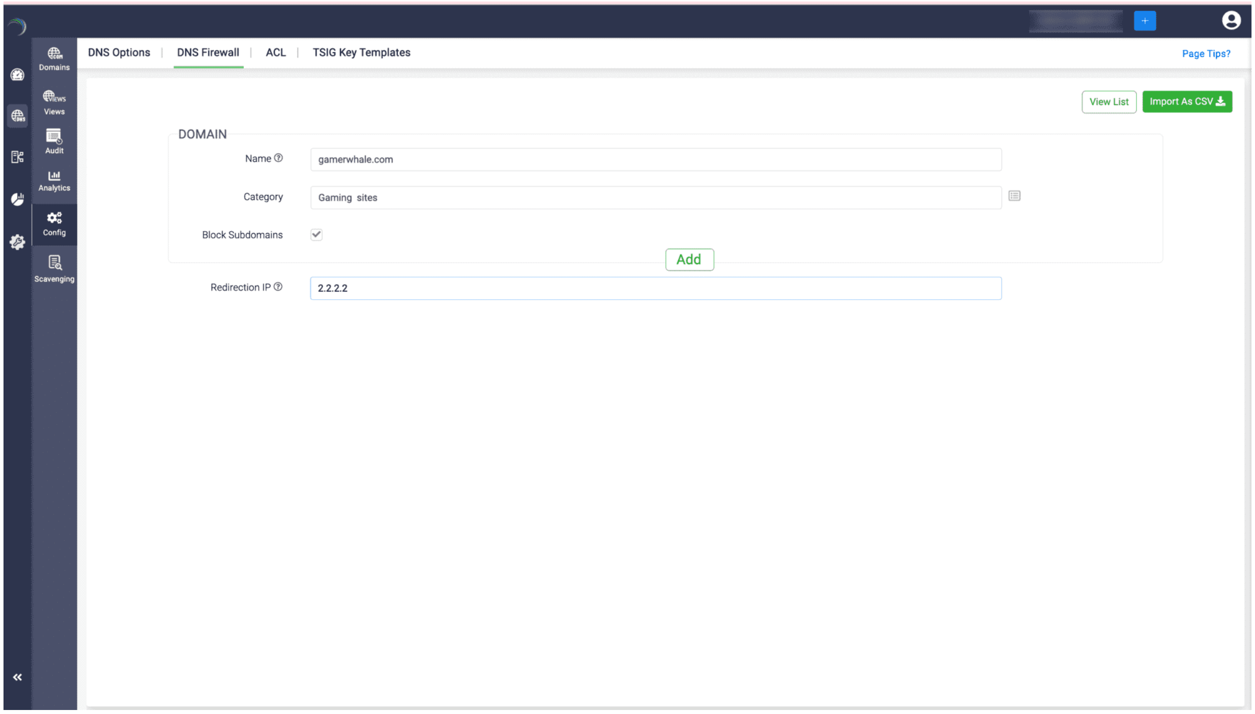
Components of DNS Firewall based

Domain Blocking

- **Blacklists:** Lists of known bad domains, which can be custom-defined by the organization or subscribed to from external security providers.
- **Category-Based Filtering:** Blocking domains based on categories, such as adult content, social media, or streaming services. DDI Central also curates most common collections of malicious or suspicious domains from third party services and enables you add your own custom collection of malicious sites.

To add a domain to the DNS Firewall Blacklist :

1. Go to **DNS→Config→ DNS Firewall.**
2. You can start adding the domains to the blacklist one by one under a particular category. Check the **Block subdomains** check box if you want to block all the subdomains of the domain as well.



DOMAIN

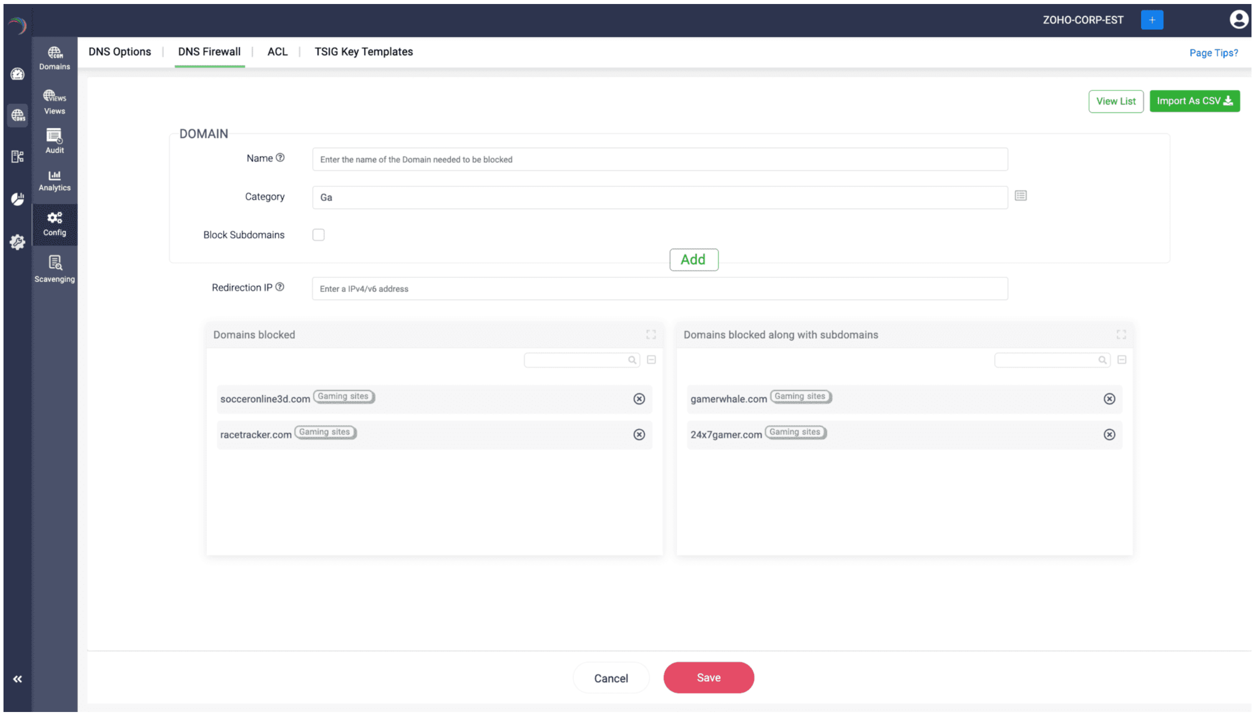
Name: gamerwhale.com

Category: Gaming sites

Block Subdomains: ☑

Redirection IP: 2.2.2.2

3. Once you click **Add**, you will see two separate lists, one that says **Domains Blocked** and the other says **Domains blocked along with subdomains**. This way, you can build your categories of malicious domains on your own.



DOMAIN

Name: Enter the name of the Domain needed to be blocked

Category: Ga

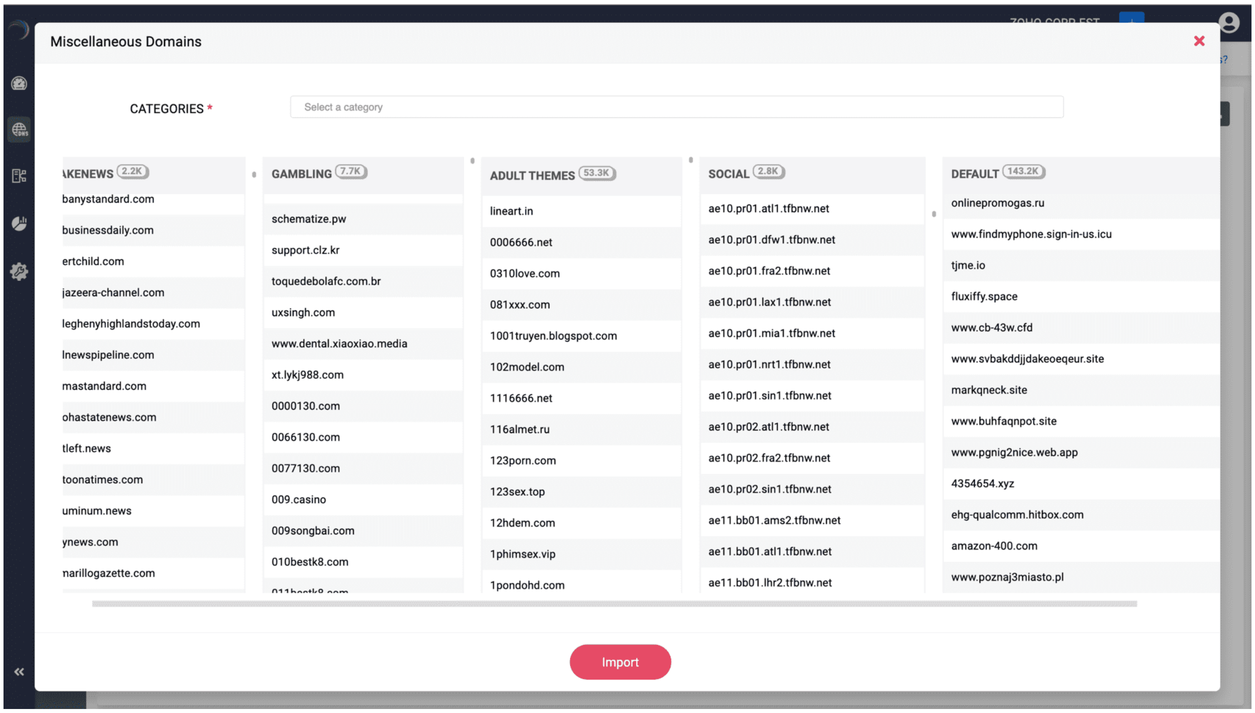
Block Subdomains □

Redirection IP: Enter a IPv4/v6 address

Domains blocked	
socceronline3d.com	Gaming sites
racetracker.com	Gaming sites

Domains blocked along with subdomains	
gamerwhale.com	Gaming sites
24x7gamer.com	Gaming sites

4. Once you are done building the list, specify the **Redirection IP** and click **Save.**
5. You can bulk import a customized list of malicious domains via CSV import for quicker addition. You can also block as many categories based on your organizational needs.
6. You can also click on the **View list** button on the top right corner of the page, to import already existing categories into the current blacklist you are building.



Miscellaneous Domains

CATEGORIES * Select a category

AKENEWS 2.2K	GAMBLING 7.7K	ADULT THEMES 53.3K	SOCIAL 2.8K	DEFAULT 143.2K
banystandard.com	schematize.pw	lineart.in	ae10.pr01.atl1.tfbnw.net	onlinepromogas.ru
businessdaily.com	support.clz.kr	0006666.net	ae10.pr01.dfw1.tfbnw.net	www.findmyphone.sign-in-us.icu
ertchild.com	toquedebolafc.com.br	0310love.com	ae10.pr01.fra2.tfbnw.net	tjme.io
jazeera-channel.com	uxsingh.com	081xxx.com	ae10.pr01.lax1.tfbnw.net	fluxiffy.space
leghenyhighlandstoday.com	www.dental.xiaoxiao.media	1001truyen.blogspot.com	ae10.pr01.mia1.tfbnw.net	www.cb-43w.cfd
lnewspipeline.com	xt.lykj988.com	102model.com	ae10.pr01.nrt1.tfbnw.net	www.svbakddjjdakeoeqeur.site
mastandard.com	0000130.com	1116666.net	ae10.pr01.sin1.tfbnw.net	markqneck.site
ohastatenews.com	0066130.com	116almet.ru	ae10.pr02.atl1.tfbnw.net	www.buhfaqnpot.site
tleft.news	0077130.com	123porn.com	ae10.pr02.fra2.tfbnw.net	www.pgnig2nice.web.app
toonatimes.com	009.casino	123sex.top	ae10.pr02.sin1.tfbnw.net	4354654.xyz
uminum.news	009songbai.com	12hdem.com	ae11.bb01.ams2.tfbnw.net	ehg-qualcomm.hitbox.com
ynews.com	010bestk8.com	1phimsex.vip	ae11.bb01.atl1.tfbnw.net	amazon-400.com
narillogazette.com	011bestk8.com	1pondohd.com	ae11.bb01.lhr2.tfbnw.net	www.poznaj3miasto.pl

7. This setup enhances network security by proactively preventing access to potentially harmful web content and mitigating cyber threats.

Configuring TSIG Keys

TSIG (Transaction Signature)

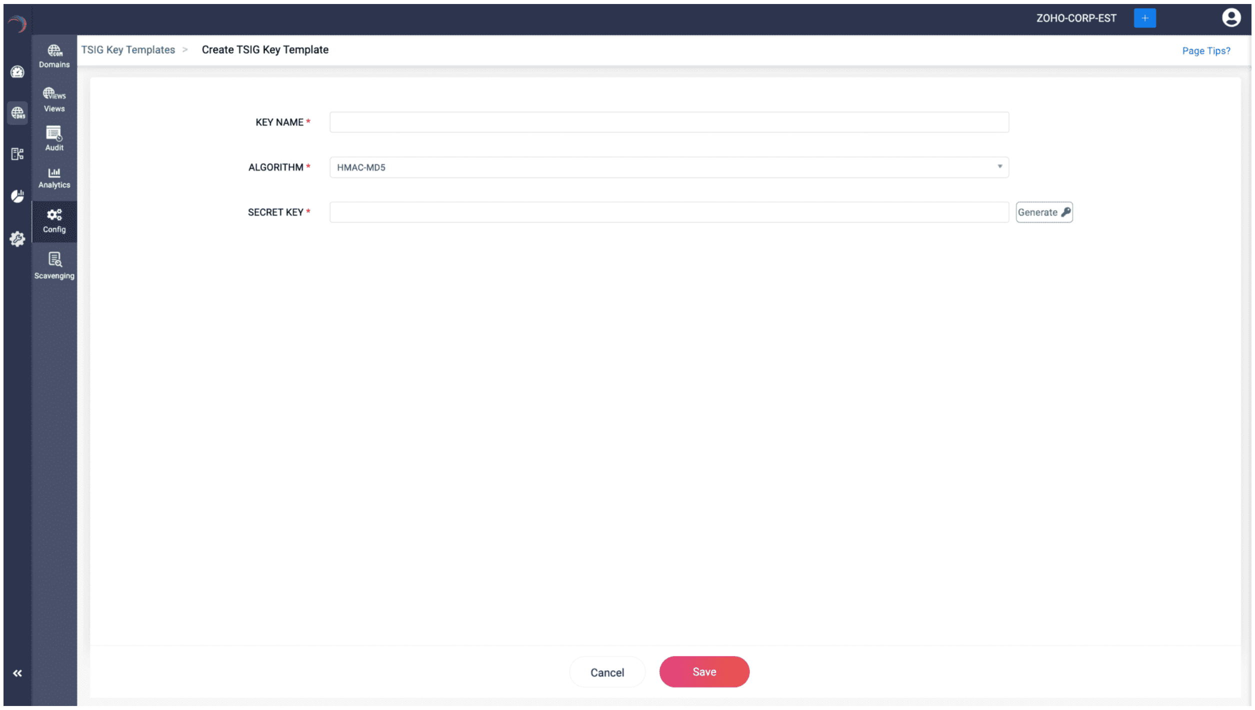
TSIG is a security protocol used in the Domain Name System (DNS) to provide authenticated and secure communications between DNS servers and between DNS servers and clients. TSIG uses shared secret keys and cryptographic signatures to validate that the DNS messages are authentic and have not been tampered with. It's primarily used for

1. **Securing Zone Transfers:** Ensuring that AXFR zone transfers occur only between authorized servers.
2. **Securing Dynamic Updates:** Authenticating requests to update DNS records dynamically, especially in Dynamic DNS (DDNS) environments.
3. **Authenticating DNS Queries and Responses:** Verifying the authenticity of both the query and the response in DNS transactions.

TSIG adds an additional layer of security to DNS operations that is not provided by standard DNS, which by itself has no mechanism for authenticating the source or integrity of DNS data.

TSIG Key Templates in DDI Central

The Key Templates are saved under the **TSIG Key Templates** tab on the **Config** page with the following fields as shown below:



KEY NAME *

ALGORITHM * HMAC-MD5

SECRET KEY *

Key Name

The *Key Name* is mainly used to identify the key across the primary and secondary name servers. Ensure a unique name is assigned to the key.

Algorithm

TSIG *Algorithm* serves essentially as a cryptographic hash function that executes HMAC operations to generate the TSIG key value. Currently, CloudDNS supports the following algorithms HMAC MD5, HMAC SHA1, HMAC SHA224, HMAC SHA256, HMAC SHA384, and HMAC SHA512 to generate the TSIG key.

Secret Key

The secret key value is an encoded base64 string with a maximum value of 255 characters that acts as a shared signature to provide transaction-level authentication for the name servers during zone transfer operations.

Configuring ACL (Access Control List)

An ACL in the context of network administration is a set of rules that control network traffic and limit access to networks and network resources based on predefined criteria. In DNS servers like ISC BIND, ACLs are used to define which clients (based on IP addresses or networks) are allowed or denied access to certain DNS services. Common uses of ACLs in DNS include:

1. **Restricting Query Access**: Defining which clients are allowed to query the DNS server.
2. **Controlling Zone Transfers**: Specifying which secondary servers are allowed to receive zone data from the primary server.
3. **Limiting Dynamic Updates**: Controlling which clients can dynamically update DNS records, often used in conjunction with TSIG for secure DDNS.

ACLs allow for the implementation of security policies by controlling who can access the DNS server and what actions they can perform, which is critical for maintaining the integrity and security of the DNS infrastructure.

Managing ACL templates

ACL templates are predefined configurations that simplify the creation of Access Control Lists (ACLs) in various network services, including DNS and DHCP servers. An ACL template allows administrators to define a set of rules or criteria once and then apply them across multiple instances, reducing redundancy and potential for error in configurations.

Usage of ACL Templates

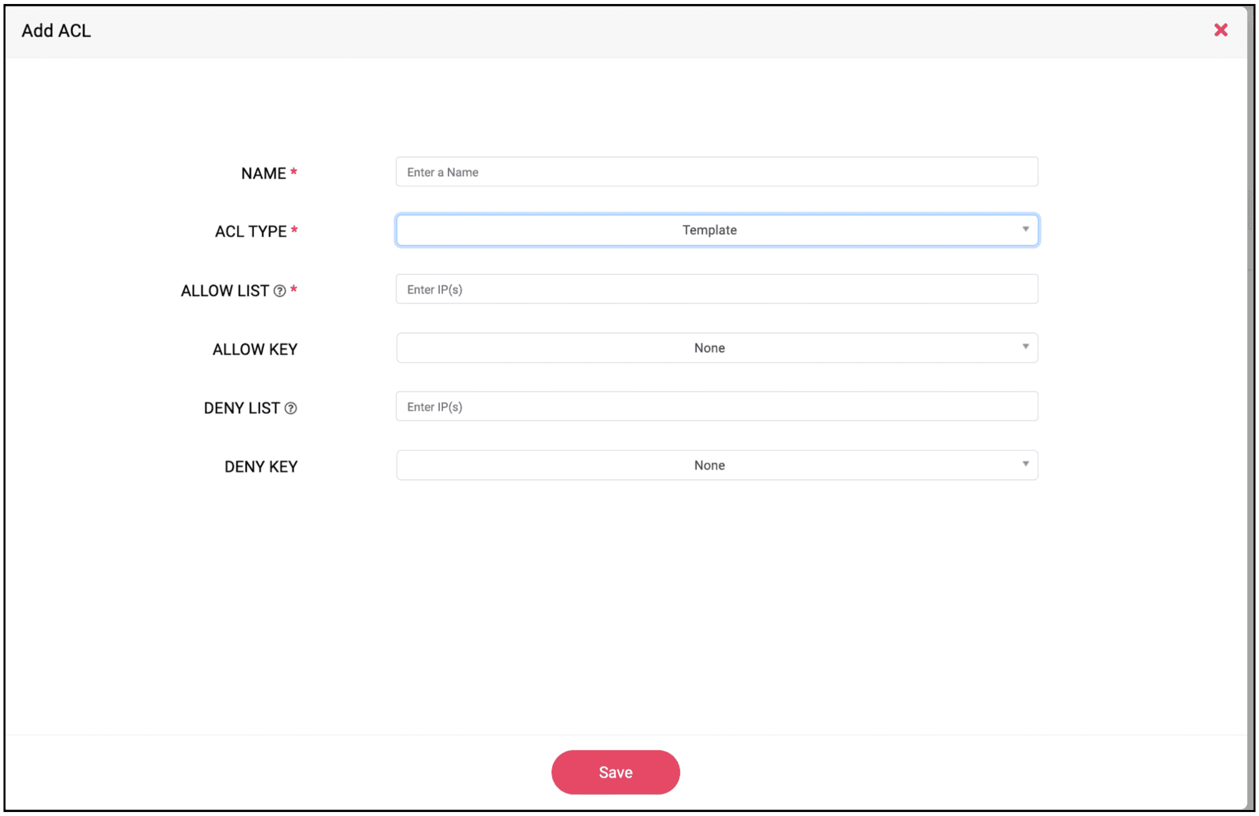
ACL templates are typically used in environments where the same access restrictions or permissions are needed across different zones, views, or services. Instead of defining the same ACL multiple times, a template is created once and then referenced wherever needed.

They can be applied gobally on the cluster level, within specific zones, views, or options.

Defining Named ACLs

To create Named ACLs

- Go to **DNS-> Config-> ACL**
- Click on **ADD ACL** button on the right.
- You can choose the type of the ACL: **ISC Format** or Template based ACL.
- For the DDI Central template, just enter the IPv4/IPv6 addresses one by one in the allow and Deny lists.


Add ACL

NAME *	Enter a Name
ACL TYPE *	Template
ALLOW LIST ⓘ *	Enter IP(s)
ALLOW KEY	None
DENY LIST ⓘ	Enter IP(s)
DENY KEY	None

- For the ISC format

  Follow the Example below:

  Here's an example of an ACL in ISC BIND format:

```
acl "internal-network" {
    192.168.0.0/24;          // An internal subnet in CIDR notation
    10.15.20.0/22;           // Another internal subnet in CIDR notation
    localhost;               // The keyword for the loopback address (127.0.0.1)
    localnets;               // A predefined match list for all local networks
    ! 192.168.0.100;         // Exclude a specific IP from the ACL
    2001:db8::/32;           // An IPv6 subnet in CIDR notation
    key "transfer-key";      // A TSIG key for secure
transactions
};
```

DNS Query Analytics

DNS analytics dashboard provides a network administrator with quick insights into the DNS and leased IP activity related to a particular domain or network segment. It helps in monitoring network usage, identifying potential issues, and understanding traffic patterns.

To access the domain analytics

- Select the Select the **DNS** menu from the menu bar along the left side of the screen.
- From the submenus that appear, choose **Analytics.**
- The analytics page appears, showing the current query rate and the total queries handled by all the DNS servers in the cluster. At the top right corner of the analytics Page, choose the type of Zones to view the query analytics.
    - **All:** Displays analytics for all domains.
    - **Hosted Domains**: Shows analytics specifically for domains that are hosted.
    - **Blocked Domains**: Presents analytics for domains that have been

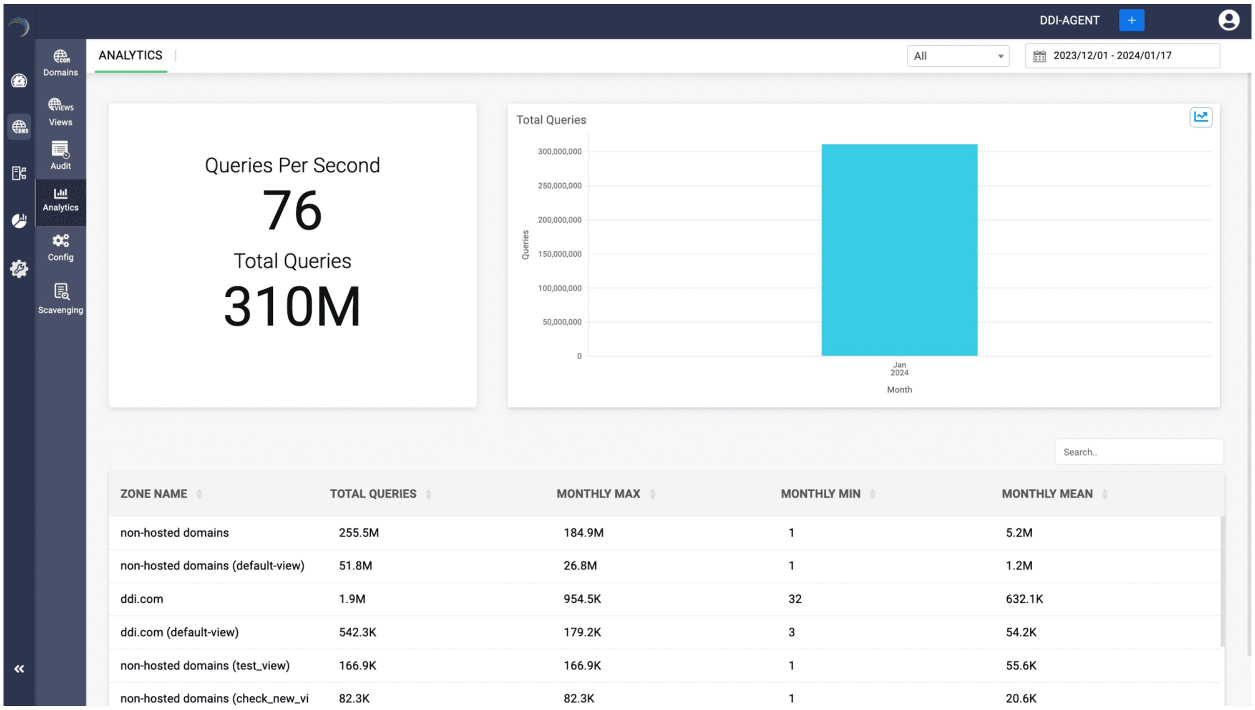
blocked.

Moreover, choose the required timeframe along which you want to analyze the performance of domains.

Queries Per Second: Indicates the current rate at which DNS queries are being processed by the server.

Total Queries: Displays the total volume of queries handled over a specific time period.

Below these metrics, you can find the list of domains and their views queried. The list also bears the query volume for even the non-hosted- domains un-resolvable by your DNS servers.

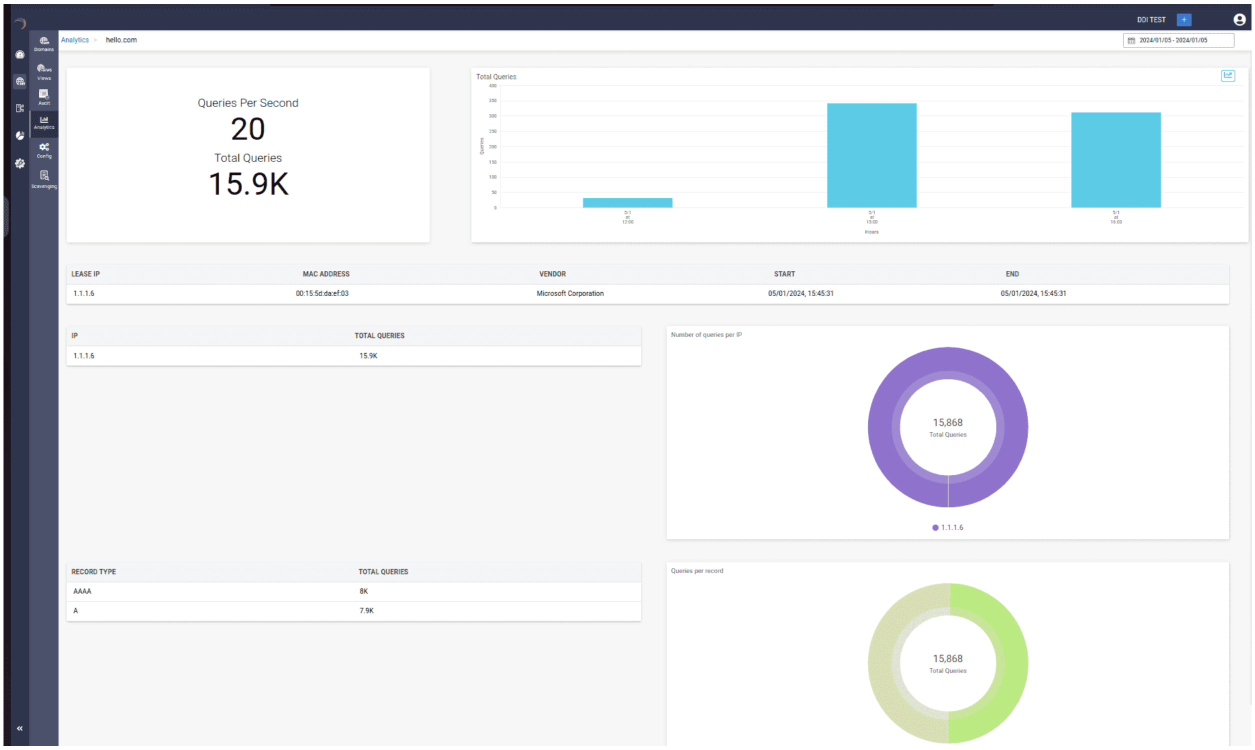


Queries Per Second
76
Total Queries
310M

ZONE NAME	TOTAL QUERIES	MONTHLY MAX	MONTHLY MIN	MONTHLY MEAN
non-hosted domains	255.5M	184.9M	1	5.2M
non-hosted domains (default-view)	51.8M	26.8M	1	1.2M
ddi.com	1.9M	954.5K	32	632.1K
ddi.com (default-view)	542.3K	179.2K	3	54.2K
non-hosted domains (test_view)	166.9K	166.9K	1	55.6K
non-hosted domains (check_new_vi	82.3K	82.3K	1	20.6K

- To thoroughly evaluate a domain's performance, select a specific domain from the list. This will display the domain's specific performance metrics, including hourly query load over a user-defined timeframe.
- You'll also see details for IP addresses leased under this domain, such as lease duration, MAC addresses identifying each host, and the vendors of the host machines.
- Additionally, the total query load across all IPs, as well as individual query loads, are visually represented in a doughnut chart accessible by selecting the respective IP.

- Furthermore, a separate doughnut chart provides a visual breakdown of the query volume for different types of DNS records, illustrating the distribution for each query type.



Queries Per Second
20
Total Queries
15.9K

LEASE IP	MAC ADDRESS	VENDOR	START	END
1.1.1.6	00:15:5d:da:ef:03	Microsoft Corporation	05/01/2024, 15:45:31	05/01/2024, 15:45:31

IP	TOTAL QUERIES
1.1.1.6	15.9K

15,868
Total Queries

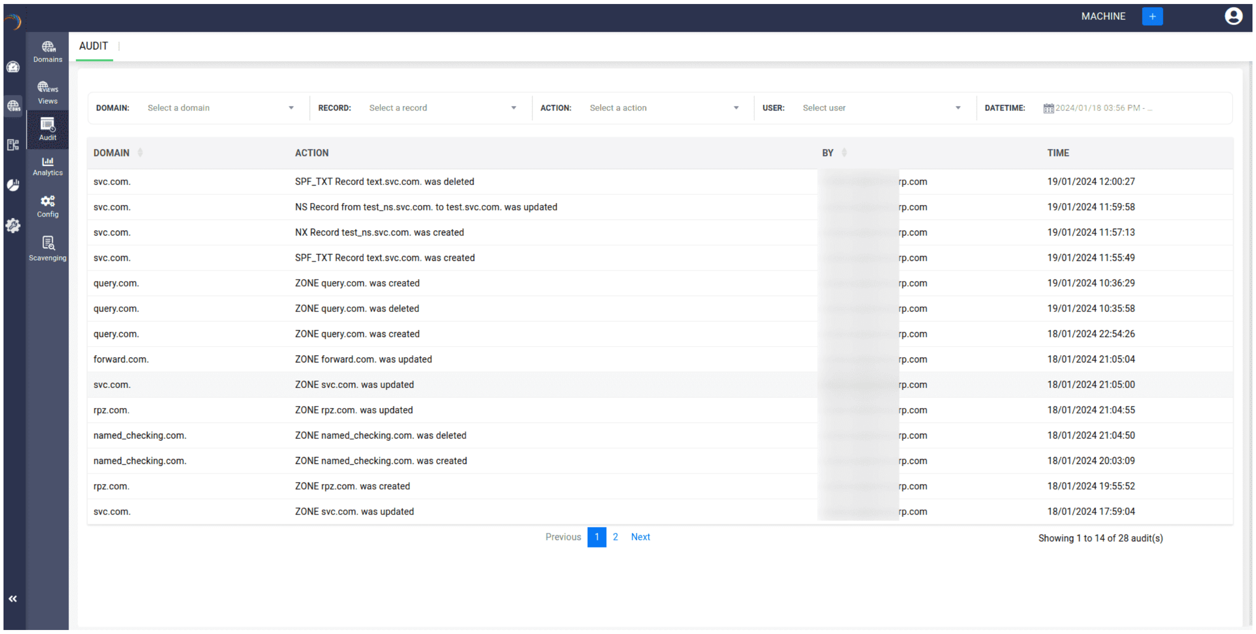
RECORD TYPE	TOTAL QUERIES
AAAA	8K
A	7.9K

15,868
Total Queries

DNS Audit Logs

ManageEngine DDI Central enables you to view the audit logs of specific domains.

Select the **DNS** menu from the menu bar along the left side of the screen. From the submenus that appear, choose **Audit.**



DOMAIN: Select a domain RECORD: Select a record ACTION: Select a action USER: Select user DATETIME: 2024/01/18 03:56 PM - ...

DOMAIN	ACTION	BY	TIME
svc.com.	SPF_TXT Record text.svc.com. was deleted	[redacted]	19/01/2024 12:00:27
svc.com.	NS Record from test_ns.svc.com. to test.svc.com. was updated	[redacted]	19/01/2024 11:59:58
svc.com.	NX Record test_ns.svc.com. was created	[redacted]	19/01/2024 11:57:13
svc.com.	SPF_TXT Record text.svc.com. was created	[redacted]	19/01/2024 11:55:49
query.com.	ZONE query.com. was created	[redacted]	19/01/2024 10:36:29
query.com.	ZONE query.com. was deleted	[redacted]	19/01/2024 10:35:58
query.com.	ZONE query.com. was created	[redacted]	18/01/2024 22:54:26
forward.com.	ZONE forward.com. was updated	[redacted]	18/01/2024 21:05:04
svc.com.	ZONE svc.com. was updated	[redacted]	18/01/2024 21:05:00
rpz.com.	ZONE rpz.com. was updated	[redacted]	18/01/2024 21:04:55
named_checking.com.	ZONE named_checking.com. was deleted	[redacted]	18/01/2024 21:04:50
named_checking.com.	ZONE named_checking.com. was created	[redacted]	18/01/2024 20:03:09
rpz.com.	ZONE rpz.com. was created	[redacted]	18/01/2024 19:55:52
svc.com.	ZONE svc.com. was updated	[redacted]	18/01/2024 17:59:04

Previous 1 2 Next

Showing 1 to 14 of 28 audit(s)

The **Audit** page helps you continuously evaluate the overall security posture of your domains and records using security audit logs to track who, what, and when with respect to domain management and record updates.

You can also filter the logs for filtering the specific activities carried out by a specific user or a specific activity carried out around a certain time frame to detect security breeches and malpractices.

Regularly reviewing your DNS infrastructure's security logs helps you ensure that the access control mechanisms are performing adequately, determine whether employees are sticking to your security practices, and catch new potential security weaknesses.

About DHCP Management

Dynamic Host Configuration Protocol (DHCP) is a critical network service that automatically assigns IP addresses and other network configuration parameters to each device on a network, enabling them to communicate with other IP networks. DHCP management streamlines the process of configuring devices on IP networks, reduces the potential for error in assigning IP addresses, and conserves the number of addresses used.

DHCP is a crucial aspect of network administration, as it ensures that devices can join the network with minimal manual setup, maintain connectivity, and have the correct network settings for accessing local resources and the Internet. Effective DHCP management includes overseeing IP address allocation, monitoring DHCP servers, and ensuring the reliability and security of the service within an organization's IT infrastructure.

Managing DHCP Scopes

What is a DHCP Scope?

A DHCP scope is a network topological element in DHCP defined as a pool of IP addresses that a DHCP server can dynamically assign to clients on a particular subnet. Each scope represents a range of IP addresses that are available for lease to client devices, as well as configuration options associated with those IP addresses.

ManageEngine DDI Central supports the following network topological elements that shape a network infrastructure:

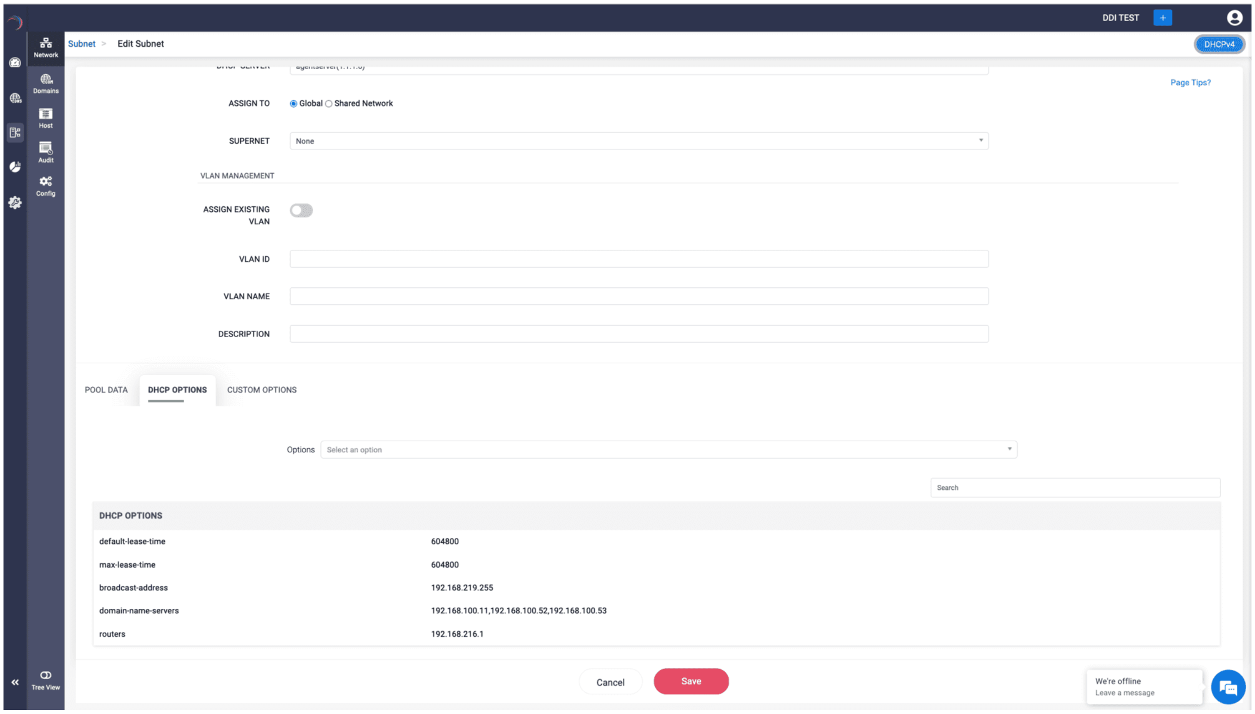
Subnets

- A subnet represents a basic segment of IP addresses (IPv4 or IPv6) within a network. Defining, a **subnet** in DDI Central is used to define a range of IP addresses that the DHCP server can assign to clients on a specific network segment.
- Each subnet is defined by a range of IP addresses and a subnet mask,

determining the network's address range.

- To create or update a subnet go to **DHCP→ Network→ Subnet.**
- Define a new subnet by providing values for various attributes of the subnet like:

  Provide the first address of the pool to be associated with the new subnet.
- Provide a suitable description for the subnet to quickly identify its purpose and convey the policy associated with it, for a common understanding of its layout.
- Specify the subnet size using an appropriate prefix, which denotes the number of IP addresses that the subnet can accommodate.
- Enable DHCP failover and select a DHCP server to take over the task of assigning IP addresses for the subnet without any significant downtime.
- Assign the necessary **DHCP options**.
- Click **Save**.
- **Note:** DDI Central also offers the option to clear the active subnets currently in lease. Clearing all the leases for a subnet removes it from your database, freeing up memory, but lease records stay intact, enabling IP addresses to revert to their original states as per the lease records after a short interval of 5 minutes.



Subnet > Edit Subnet

ASSIGN TO ● Global ○ Shared Network

SUPERNET None

VLAN MANAGEMENT

ASSIGN EXISTING VLAN

VLAN ID

VLAN NAME

DESCRIPTION

POOL DATA | DHCP OPTIONS | CUSTOM OPTIONS

Options Select an option

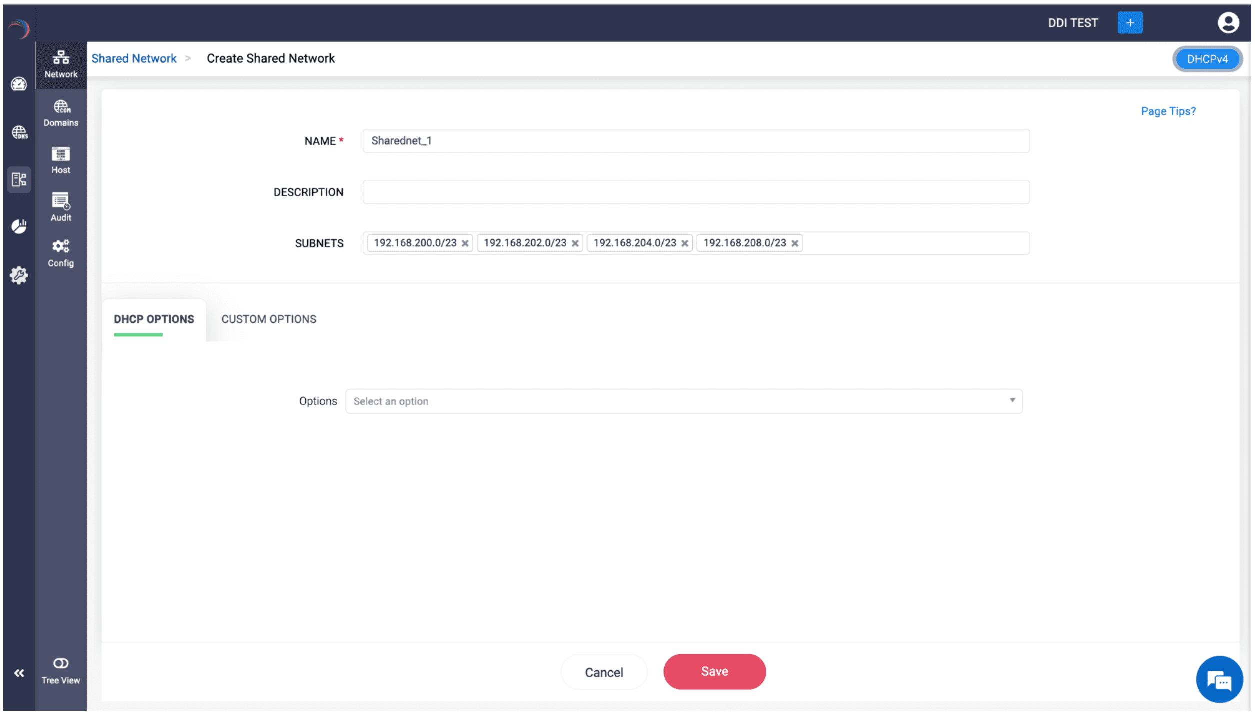
Search

DHCP OPTIONS	
default-lease-time	604800
max-lease-time	604800
broadcast-address	192.168.219.255
domain-name-servers	192.168.100.11,192.168.100.52,192.168.100.53
routers	192.168.216.1

Shared Networks

- A **shared-network** defined in DDI Central is used when multiple logical IP networks (subnets) share the same physical network segment.
- Shared networks allow DHCP to serve multiple subnets on a single physical network, providing different IP configurations to clients based on their network segment.
- To create a new shared network, go to DHCP→ Network→ Shared Network.
- Assign a unique name and description for the shared network.

- Just add the required subnets and apply the necessary DHCP or custom options .
- Click **Save.**



NAME * Sharednet_1

DESCRIPTION

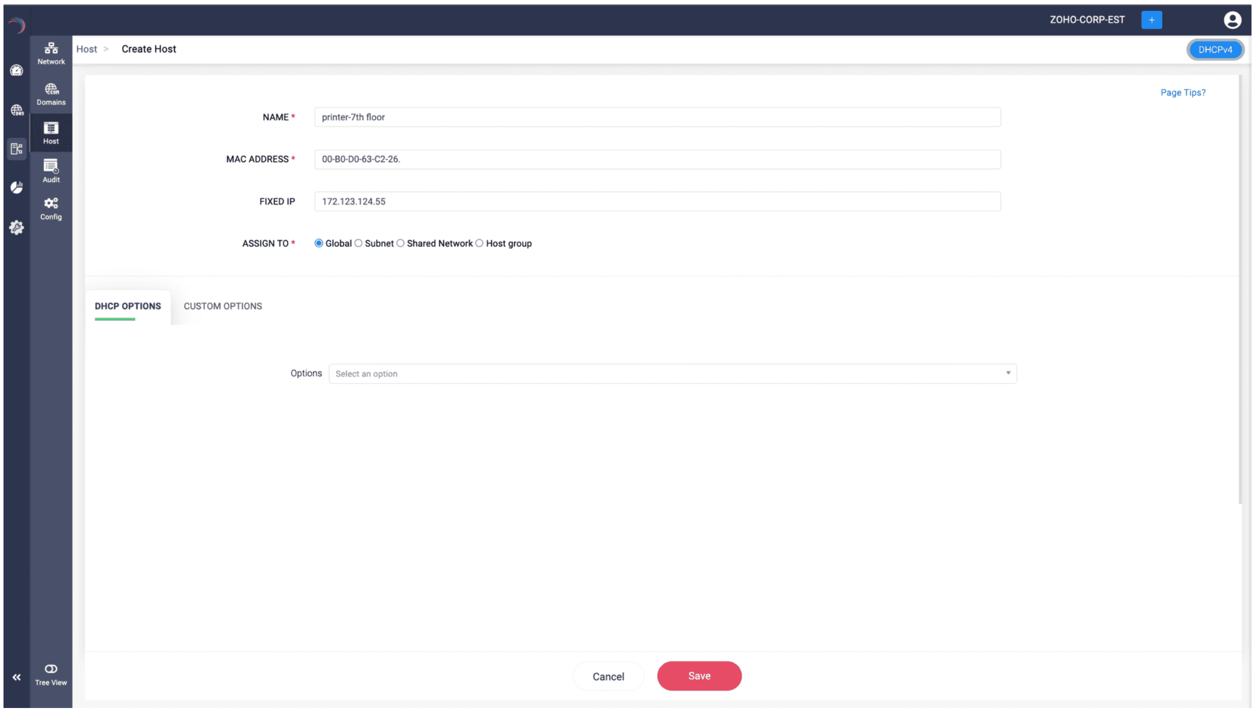
SUBNETS 192.168.200.0/23 × 192.168.202.0/23 × 192.168.204.0/23 × 192.168.208.0/23 ×

Options Select an option

Hosts

- A **host** declaration specifies settings for individual clients based on their hardware (MAC) address.

- This is used for assigning fixed IP addresses or specific configurations to particular clients, ensuring that a specific client always receives the same IP address and settings.
- To create a Host with a fixed address, go to **DHCP→ Network→ Host.**
- On the Host page, provide a unique name for the host, the mac address of the host.



NAME * printer-7th floor

MAC ADDRESS * 00-B0-D0-63-C2-26.

FIXED IP 172.123.124.55

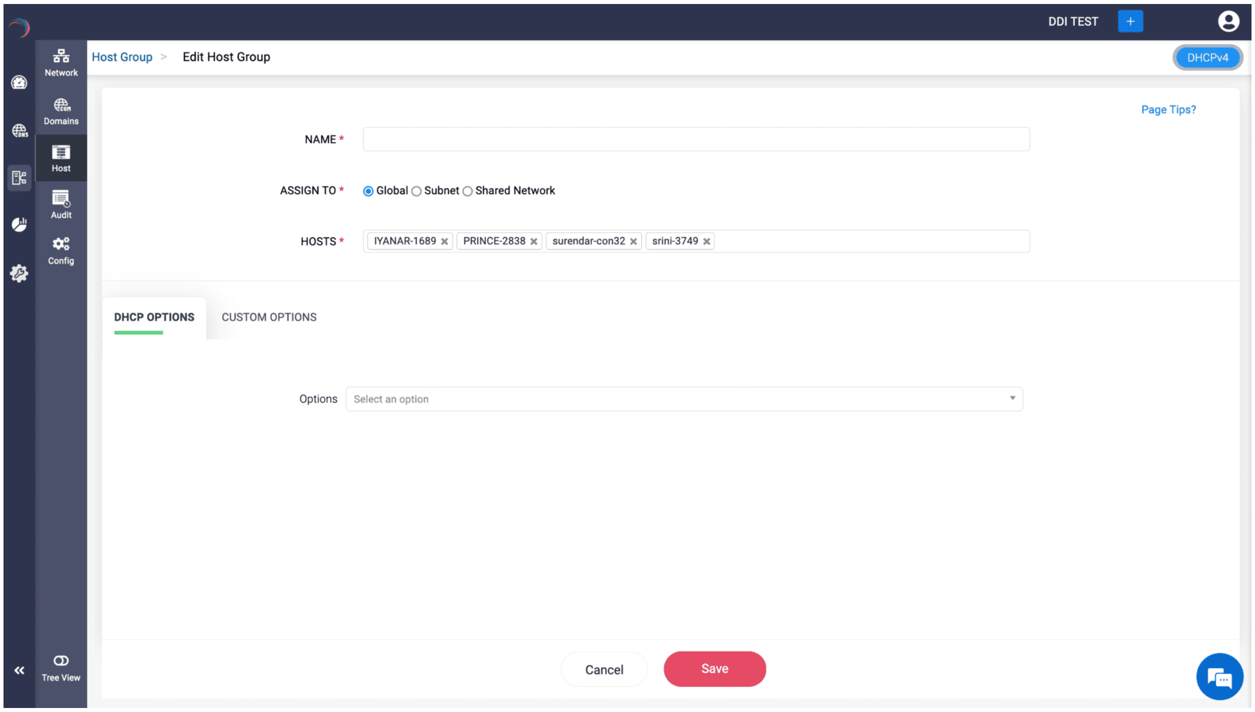
ASSIGN TO * ● Global ○ Subnet ○ Shared Network ○ Host group

Options Select an option

Host Groups

- Host groups are a group of **hosts** combined logically for easier management.

- Grouping hosts can simplify configuration, especially in large networks, by applying common settings to multiple hosts.
- You can apply a multiple DHCP options over this combination of hosts for customized management.



Host Group > Edit Host Group

NAME *

ASSIGN TO * ● Global ○ Subnet ○ Shared Network

HOSTS * IYANAR-1689 × PRINCE-2838 × surendar-con32 × srini-3749 ×

Options: Select an option

Supernets

- Supernets, or supernetting, refers to aggregating multiple networks into a larger network. In the context of DHCP, this is not a direct feature but rather a concept of network design.

- Supernetting is used in IP routing more than in DHCP configurations. It's about combining smaller subnets into a larger address space for routing purposes.

VLANs (Virtual LANs)

- VLANs are a network configuration that segments a physical network into multiple logical networks at the data link layer (Layer 2).
- DDI Central enables DHCP servers to serve different VLANs as distinct subnets or shared networks.
- Each VLAN you create within a subnet functions as a separate network, which improves performance by reducing broadcast traffic, enhances security by isolating sensitive data, and simplifies management by grouping devices according to function, department, or project.You can also associate an already existing VLAN to the subnet.
- Name and provide a suitable description to quickly identify the new VLAN. Also assign a suitable VLAN ID.
- **Note:** VLAN IDs are represented by a 12-bit number, but the usable range of VLAN IDs is from 2 to 4094.

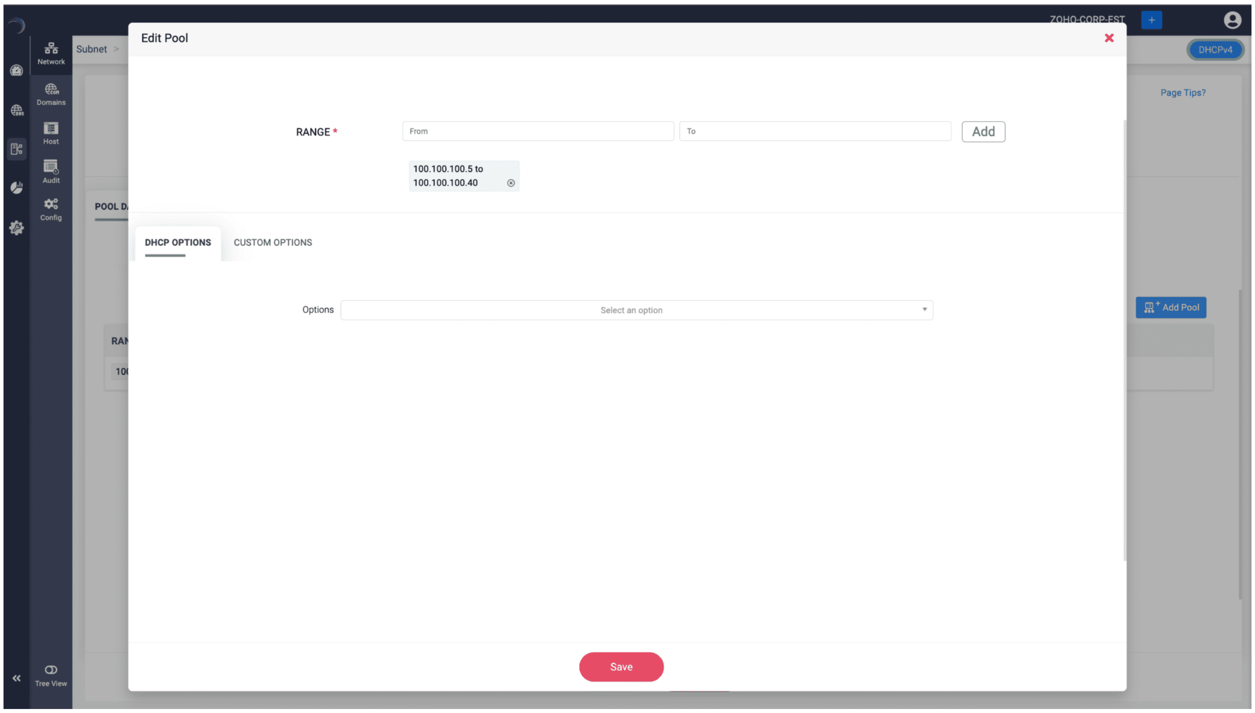
Note: DDI Central enables you to define Supernets and VLANs only to simplify network administration. However, no advanced DHCP configurations, such as DHCP options or Client Classes, can be implemented on the Supernets and VLANs.

Also, when discovering your current configurations from your network infrastructure using DDI Central discovery tools, it's crucial to note that VLANs and Supernets configured in your network will not be discovered. Therefore, ensure that you configure them separately in DDI Central for comprehensive and accurate network management.

Address Pools

- An address pool within a subnet specifies the range of IP addresses available for dynamic assignment.
- Pools are used to control the distribution of IP addresses to clients within a subnet. They allow for more granular management of IP address allocations, including setting different options or restrictions for different pools within the same subnet.

- When configuring options at the subnet level, you can add and define the pool or address range within the subnet that should be configured with a specific set of options. Multiple combinations of options can be applied to various address ranges within the same subnet.
- Address pool configurations in a subnet can either allow or deny specific client classes for dynamic IP provisioning. If "Allow" is set to "yes," the pool permits provisioning for the chosen client class, while setting it to "NO" excludes provisioning for that class. Choosing "none" means the address pool is open for dynamic provisioning to all clients in the subnet without class restrictions.



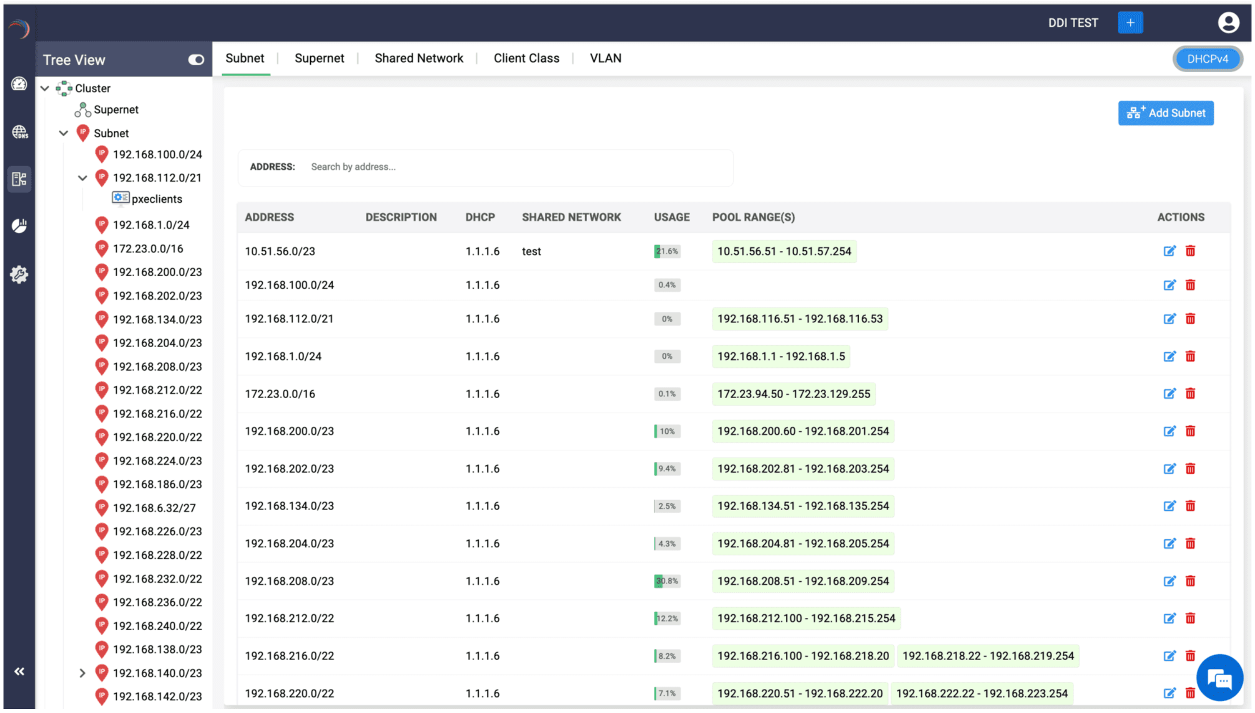
RANGE *

100.100.100.5 to 100.100.100.40

Options

DHCP scope visualization

DDI Central also lets you organize and manage the scopes in a hierarchical manner by providing hierarchical tree-view that show how different scopes relate to one another within the network. The DHCP scope tree view enables admins to quickly locate and access specific scopes, subnets, to manage configurations and troubleshooting tasks for a specific scope.



ADDRESS: Search by address...

ADDRESS	DESCRIPTION	DHCP	SHARED NETWORK	USAGE	POOL RANGE(S)	ACTIONS
10.51.56.0/23		1.1.1.6	test	21.6%	10.51.56.51 - 10.51.57.254	
192.168.100.0/24		1.1.1.6		0.4%		
192.168.112.0/21		1.1.1.6		0%	192.168.116.51 - 192.168.116.53	
192.168.1.0/24		1.1.1.6		0%	192.168.1.1 - 192.168.1.5	
172.23.0.0/16		1.1.1.6		0.1%	172.23.94.50 - 172.23.129.255	
192.168.200.0/23		1.1.1.6		10%	192.168.200.60 - 192.168.201.254	
192.168.202.0/23		1.1.1.6		9.4%	192.168.202.81 - 192.168.203.254	
192.168.134.0/23		1.1.1.6		2.5%	192.168.134.51 - 192.168.135.254	
192.168.204.0/23		1.1.1.6		4.3%	192.168.204.81 - 192.168.205.254	
192.168.208.0/23		1.1.1.6		30.8%	192.168.208.51 - 192.168.209.254	
192.168.212.0/22		1.1.1.6		12.2%	192.168.212.100 - 192.168.215.254	
192.168.216.0/22		1.1.1.6		8.2%	192.168.216.100 - 192.168.218.20 192.168.218.22 - 192.168.219.254	
192.168.220.0/22		1.1.1.6		7.1%	192.168.220.51 - 192.168.222.20 192.168.222.22 - 192.168.223.254	

DDI Central provides flexible and powerful ways to manage IP address assignment and network configurations. Understanding these elements is crucial for network administrators to effectively design and manage their network's IP addressing scheme.

DHCP Fingerprinting With Client Classes

Client Classes and Sub Classes

Client classes and Sub Classes are powerful features used to group clients (DHCP clients) and apply specific DHCP options or behaviors to those groups. These classes and subclasses enable more granular control over how DHCP services are delivered to different types of clients on the network.

Client Classes

- A client class in ISC DHCP is a grouping of DHCP clients that share common characteristics. These characteristics are usually defined by matching specific criteria in the DHCP discovery or request messages that the clients send.
- Classes are used to apply different DHCP configurations to different groups of

clients. For example, you might have different classes for different types of devices (like printers, laptops, and phones) or different operating systems.

Example of a Client Class:

```
class "Printers" {
  match if substring(hardware, 1, 3) = 00:11:22;
}
subnet 192.168.1.0 netmask 255.255.255.0 {
  pool {
    allow members of "Printers";
    range 192.168.1.50 192.168.1.60;
  }
}
```

- In this example, a class named "Printers" is defined, which includes any client whose MAC address starts with 00:11:22. Printers are then assigned IP addresses from a specific range.

Subclasses

- A subclass in ISC DHCP is a more specific grouping within a class. Subclasses are defined based on a subclass-specific value, such as a MAC address or a client identifier.
- Subclasses allow for even more specific targeting of DHCP options and configurations. They are useful in scenarios where a broad class needs to be divided into finer groups.

Example of Subclasses:

```
class "MobileDevices" {
  match if substring(option vendor-class-identifier, 0, 6) = "iPhone" or substring(option vendor-class-identifier, 0, 7) = "Android";
}

subclass "MobileDevices" "iPhone" {
  match if substring(option vendor-class-identifier, 0, 6) = "iPhone";
}

subclass "MobileDevices" "Android" {
  match if substring(option vendor-class-identifier, 0, 7) = "Android";
}
```

- **Description**: This configuration first defines a broad class for mobile devices, and then two subclasses for iPhones and Android devices, respectively. Each subclass can then be given different IP ranges, options, or policies.

Applications and Benefits

1. **Customized Configuration**: Allows network administrators to tailor DHCP settings to the specific needs of different devices or user groups.
2. **Network Management**: Easier management of network resources and policies by segmenting clients into manageable groups.

3. **Policy Enforcement**: Enforces different network policies for security, access control, or bandwidth allocation based on client type.

Configuring Classes and Sub Classes in DDI Central

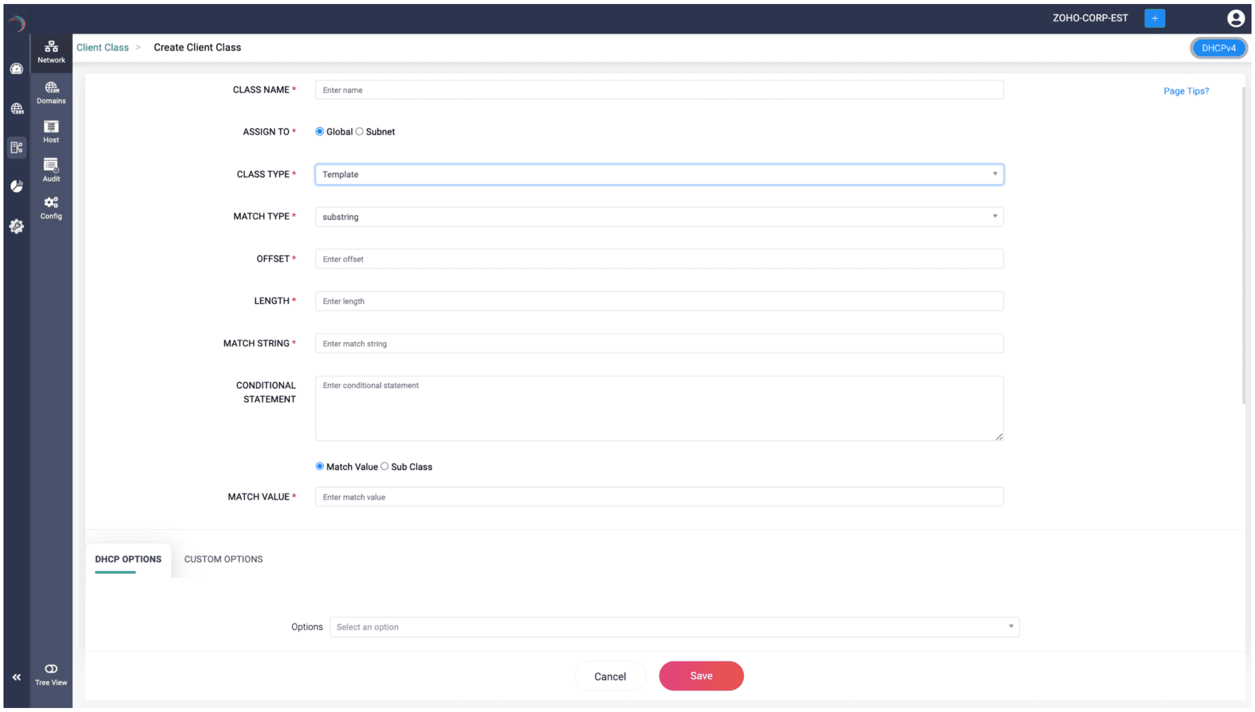
To create a client class;

- Go to **DHCP→ Network→ Client Class**
- The **Create Client Class** page appears on the screen.
- Assign the Client class a unique name.
- **ASSIGN TO:** Assign the scope level for the client class, whether its configurations should be applied for the matching client on a specific subnet level or global level. The Global option suggests it could be applied across all subnets, whereas a specific Subnet could be chosen to restrict the class to a particular network segment.
- **CLASS TYPE**: The class type field likely refers to the basis of the class definition. **Template** might be an option here indicating that this class is a

template that can be reused or that you are creating this class based on a pre-defined template.

- **MATCH TYPE**: This defines the method by which the DHCP server will match clients to this class. **Substring** indicates that the server will look for a matching string of characters within the client's DHCP messages.
- **OFFSET**: In the context of matching by substring, this defines the starting position in the client's DHCP message where the matching should begin.
- **LENGTH**: This specifies the length of the substring that the DHCP server should match against the client's DHCP message.
- **MATCH STRING**: The actual string of characters the DHCP server will look for in the client's DHCP message to determine if it belongs to this client class.
- **CONDITIONAL STATEMENT**: This field allows for more complex matching rules, perhaps using logical or comparison operators to evaluate whether clients meet the criteria for this class.
- **Match Value / Sub Class**: This section has a checkbox that is used to indicate whether a match value should be used to further define subclasses within this client class.
- **MATCH VALUE**: If subclasses are being defined, this field would be where you specify the value that differentiates each subclass.

- **DHCP OPTIONS**: Here, you would specify any DHCP options that should be applied to clients within this class. These could include options like DNS servers, domain name, lease time, etc.
- **CUSTOM OPTIONS**: This section is likely for defining additional DHCP options that are not part of the standard set, which could be specific to the organization or the DHCP server software being used.
- Cilck **Save**.



CLASS NAME * Enter name

ASSIGN TO * ● Global ○ Subnet

CLASS TYPE * Template

MATCH TYPE * substring

OFFSET * Enter offset

LENGTH * Enter length

MATCH STRING * Enter match string

CONDITIONAL STATEMENT Enter conditional statement

● Match Value ○ Sub Class

MATCH VALUE * Enter match value

Classes and subclasses in DDI Central add flexibility and precision to DHCP management, enabling complex scenarios and specific requirements to be met efficiently. This is particularly useful in large or diverse network environments.

DHCP Fingerprinting with Client Classes

DHCP fingerprinting, a method of device identification through DHCP, leverages client class parameters to provide a means for more granular network management and resource allocation. This process involves the DHCP client sending additional information to the DHCP server, which in turn uses this information to identify the type of client and assign IP addresses or parameters accordingly. This technique is especially useful in environments where different types of devices require distinct network configurations or policies.

How DHCP Fingerprinting Works:

1. **Client Class Parameters**: When a DHCP client requests an IP address, it can

provide additional information in the form of vendor class identifiers (VCI) or user class identifiers (UCI). These identifiers are part of the DHCP discovery or request packets.

2. **Server Recognition**: The DHCP server is configured to recognize these identifiers and categorize clients into different classes based on the provided information.

Applications of DHCP Fingerprinting:

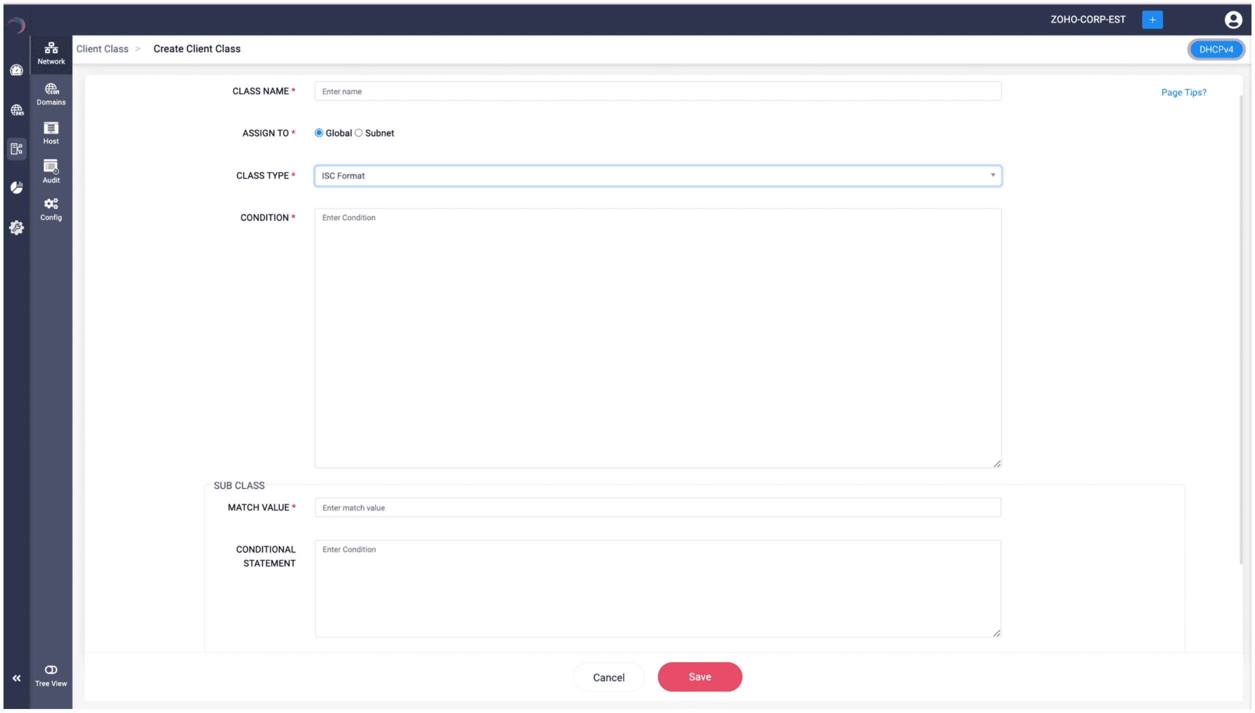
- **Differentiated Resource Allocation**: You can dedicate one address pool for specific types of devices, like VoIP devices, and a separate pool for data devices. This is useful in networks where different device types have different network requirements.
- **Policy Enforcement**: For source routing policies, where voice and data packets are routed differently, DHCP fingerprinting helps in applying these policies right from the point of network entry.
- **Administrative Segmentation**: In a large organization, managing devices based on their type (like printers, workstations, mobile devices) becomes easier with DHCP fingerprinting.

Example Scenario:

Consider a network where VoIP devices and data devices need to be segregated:

```
class "VoIP-Phones" {
  match if substring(option vendor-class-identifier, 0, 4) = "VoIP";
}
class "Data-Devices" {
  match if substring(option vendor-class-identifier, 0, 4) != "VoIP";
}

subnet 192.168.1.0 netmask 255.255.255.0 {
  pool {
    allow members of "VoIP-Phones";
    range 192.168.1.10 192.168.1.50;
  }
  pool {
    allow members of "Data-Devices";
    range 192.168.1.51 192.168.1.100;
  }
}
```

- In this configuration, two classes are defined based on the vendor class identifier. VoIP phones are assigned IP addresses from a specific range, separate from the range used for data devices. The same can be configured using DDI Central GUI using templates or the above can be given ISC bind format in the Condition text box and simply click **Save.**

- 

Client Class > Create Client Class

CLASS NAME * Enter name

ASSIGN TO * ● Global ○ Subnet

CLASS TYPE * ISC Format

CONDITION * Enter Condition

SUB CLASS

MATCH VALUE * Enter match value

CONDITIONAL STATEMENT Enter Condition

Benefits of DHCP Fingerprinting:

- **Efficient Network Management**: Allows for the dynamic assignment of IP addresses and configurations based on device type, improving network efficiency.
- **Enhanced Security**: Helps in implementing security policies tailored to different device types.
- **Quality of Service (QoS)**: Ensures that devices like VoIP phones that require higher QoS receive the necessary network configurations.
- **Scalability**: Makes the network more adaptable to the addition of new types of devices without requiring major configuration changes.

Considerations:

- **Accuracy**: The accuracy of DHCP fingerprinting depends on the uniqueness and consistency of the vendor or user class identifiers provided by the devices.
- **Configuration Complexity**: Implementing DHCP fingerprinting can add complexity to DHCP server configuration and requires thorough planning and testing.

DHCP fingerprinting is a powerful tool in network administration, enabling the categorization and appropriate management of different types of devices within the network. It enhances the capability to efficiently allocate network resources, enforce policies, and ensure optimal performance for all devices.

DHCP Options

Table of Contents

DHCP Options

DHCP (Dynamic Host Configuration Protocol) options are additional pieces of configuration information that can be provided by a DHCP server to DHCP clients during the lease negotiation process.

These options offer a standardized way to communicate various parameters and settings to devices on a network dynamically. Each DHCP option is identified by a specific code, and the values associated with these codes convey specific types of information.

Here are some common DHCP options and what they typically do:

1. **Subnet Mask (Option 1):** Provides the subnet mask information, allowing devices to understand the network's subnet structure.
2. **Router/Gateway (Option 3):** Specifies the IP address of the default gateway or router that devices should use for routing traffic outside their local subnet.
3. **Domain Name Server (DNS) (Option 6):** Supplies the IP addresses of DNS servers that devices can use to resolve domain names to IP addresses.

4. **Domain Name (Option 15):** Specifies the domain name for devices on the network.
5. **Time Offset (Option 2):** Provides the time zone offset in seconds from Coordinated Universal Time (UTC).
6. **NTP Servers (Option 42):** Supplies the IP addresses of Network Time Protocol (NTP) servers, allowing devices to synchronize their clocks.
7. **Hostname (Option 12):** Communicates the preferred host name for the client.
8. **Broadcast Address (Option 28):** Informs devices about the broadcast address for their subnet.
9. **TFTP Server Name (Option 66):** Specifies the hostname or IP address of a TFTP server. Often used in VoIP deployments for firmware updates.
10. **Bootfile Name (Option 67):** Specifies the name of the boot file that devices should load from a TFTP server.

These options enhance the DHCP process by providing crucial configuration details, allowing devices to function properly within a network without manual configuration. DHCP options are particularly useful in scenarios where a large number of devices need to be configured dynamically and consistently across the network. Different DHCP options cater to various aspects of network configuration, from addressing to naming and time synchronization.

Options for IPv4 and IPv6

DHCPv4 (Dynamic Host Configuration Protocol for IPv4) and DHCPv6 (Dynamic Host Configuration Protocol for IPv6) have separate sets of options. Each version of the DHCP protocol has its own set of option codes to convey specific configuration information to DHCP clients.

While some options may serve similar purposes in both DHCPv4 and DHCPv6 (e.g., DNS options), the option codes and formats are different due to the differences in the underlying IP versions and the specific requirements of each protocol.

DHCP Option Configuration Levels

DHCP options can be applied at different levels within a DHCP server's configuration. The application of options depends on the DHCP server software

Here are common places where DHCP options can be applied:

Global Configuration

Options can be set at the global level, affecting the entire DHCP server. This is useful for settings that are common to all scopes or subnets.

Subnet Level

DHCP options can be configured at the subnet level. Each subnet declaration in the DHCP server configuration can have its own set of options. This allows for customization based on the characteristics of each subnet.

Pool Level

Within a subnet, you define address pools. Options can be set at the pool level, affecting the devices that receive IP addresses from that specific pool. Useful for fine-grained control over specific ranges of IP addresses.

Client Class Level

Client classes group DHCP clients based on certain characteristics, such as hardware type, client identifier, or other parameters.This allows for customization based on the type or characteristics of DHCP clients. Once clients are grouped into classes, specific DHCP options can be applied selectively to those classes.

Host Level

Options can be set for specific hosts, providing individualized configuration parameters. This is useful when you need to apply specific settings to particular devices.

Shared Network Level

Options can be configured specifically for a shared network. Configurations at this level apply to all subnets within that shared network. It allows for common settings across multiple subnets.

Options precedence in DHCP

In ISC DHCP, options can be specified at multiple levels, and the server determines which options to apply to a client based on a defined precedence. The typical order of precedence for DHCP options in ISC DHCP is as follows:

1. **Host-Specific Options:** The highest precedence. If options are defined for a

specific host (identified by its MAC address or client identifier), these options override all others.

2. **Class and Subclass Options:** If the host is a member of a defined class or subclass, options set for these take precedence next, unless overridden by host-specific options.
3. **Pool-Level Options:** Options defined at the pool level come after class and subclass options. These are specific to a range of IP addresses and override subnet, shared-network, and global options for clients receiving an IP from that pool.
4. **Subnet-Specific Options:** Options defined for a specific subnet. These are applied to all IPs within the subnet unless overridden by higher precedence options (like pool, class, subclass, or host).
5. **Shared-Network and Global Options:** Shared-network level options come next, followed by global options which are the default settings across the entire DHCP server.
6. **Client-Requested Options:** Lastly, if clients request specific options in their DHCP requests, the server will provide these options if they are available and configured, respecting the above precedence rules.

DDI Central processes and applies the options according to these rules when determining the final set of options to send to a DHCP client in the offered lease. It's important to note that the most specific option will take precedence in the case of conflicts, with host declarations being the most specific and global options being the least specific.

Custom DHCP Options

Table of Contents

About Custom DHCP options

Defining custom DHCP options enable network administrators to extend and tailor DHCP functionality beyond the standard configuration parameters. Custom DHCP options provide a way to convey specific information to DHCP clients during the lease negotiation process. Here's a general process for defining custom DHCP options:

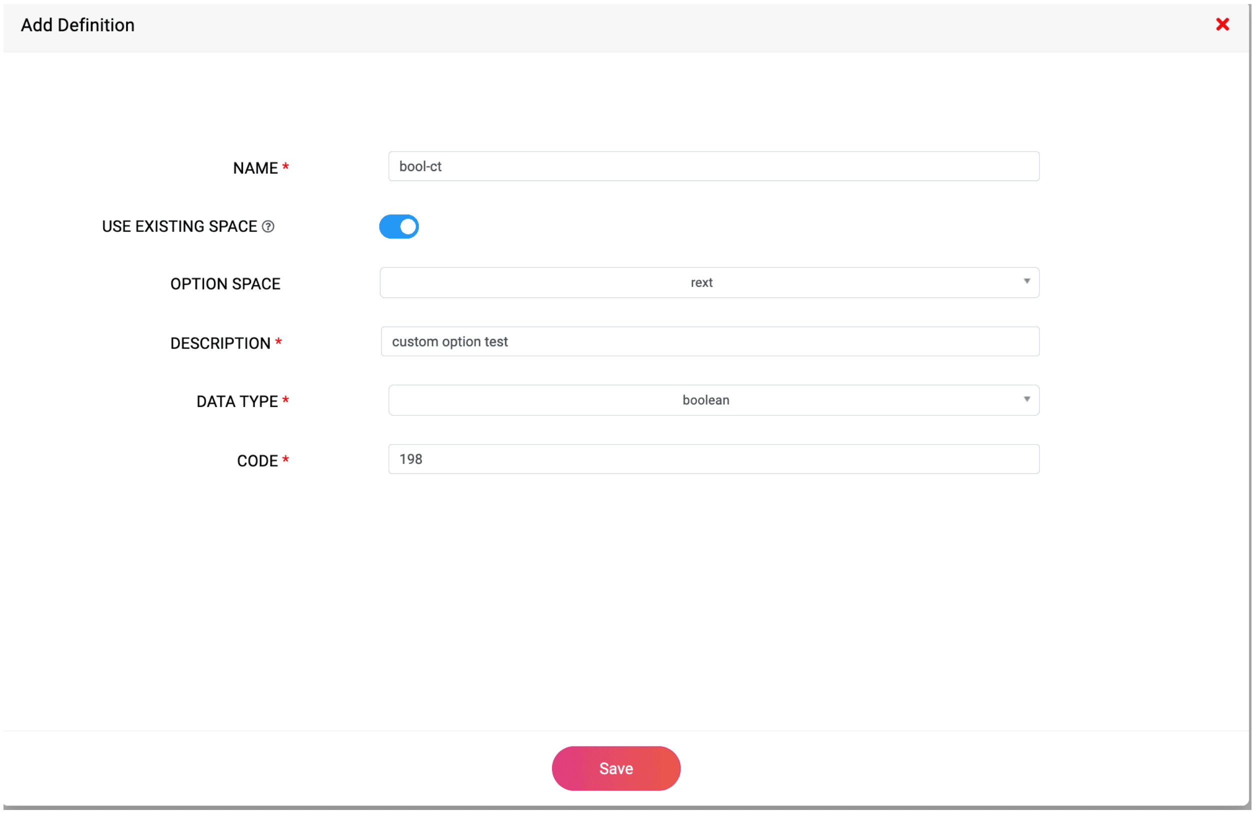
In ManageEngine DDI Central, Custom Options are defined and values are provided within the respective fields provided within the GUI by specifying details like: the user-defined option name, code, and any data type restrictions as shown in the image below. These custom options can be defined at multiple levels. To define cluster level custom options, go to **DHCP→ Config→ Custom Option.**

[option-name] [option-space] [option-description] [option- data-type] [option-code] are the parameters that help define a custom DHCP option.

To create a Custom option in DDI Central:

Click on the green + (plus) button besides the Options dropdown box.

The **Add Definition** window appears. Here you can declare and define your new Custom option in DDI Central:



Add Definition

NAME *	bool-ct
USE EXISTING SPACE	●
OPTION SPACE	rext
DESCRIPTION *	custom option test
DATA TYPE *	boolean
CODE *	198

1. **NAME**: This is a required field where you would enter the name of the new custom option you are defining. It should be a unique identifier that

accurately describes the option you're adding.

2. **USE EXISTING SPACE**: This toggle indicates whether the new custom option should be added to an existing set of options (an option space) or if it's going to define a new one. If the toggle is enabled (turned on), you should select an existing space from the **OPTION SPACE** field.
3. **OPTION SPACE**: If you are using an existing space (as indicated by the toggle above), you would enter the name of that space here. This would be the grouping or category under which your new option will be classified.
4. **DESCRIPTION**: This required field is where you would provide a detailed explanation of what the custom option does and what values it expects; the valid punctuation like: some data types accept only spaces while some options accept comma separated values. The description helps users understand the purpose, grammar, and usage of the option.
5. **DATA TYPE**: Here, you choose the type of data the custom option will use. The data types include boolean, string, integer, IP address, etc., depending on what the system allows.
6. **CODE**: This is a required field where you enter the specific code that identifies the custom option. This is often a numeric value that is used in configuration files or by the system to recognize the option.

7. **Save Button**: After filling out all the necessary information, click **Save** to save the new custom option to the system.

Note:

The **option-name** must be different from server-defined options and consist of alphanumeric characters and '-'.

The **option-code** is typically between 128 and 254.

Supported **option-types** include boolean, integer[(signed) integer8, 16, 32, unsigned integer 8, 16, 32], string, text and IPv4 or IPv6 address, array of IP addresses, record and encapsulation.

Custom Option- Data Types

Boolean Type Options

The ISC BIND declaration format is shown below along with an example definition of a boolean option named my-option with code 209.

Declaration: option my-option code 209 = boolean;

Once declared, this option can accept values based on the grammar defined.

Setting: option my-option true;

Integer Type Options

Options of data type integer include specification of signed or unsigned (or blank) and integer length of either 8, 16, or 32 bits.

Declaration: option bits-per-sec code 210 = unsigned integer 32;

Setting: option bits-per-sec 1544000;

String Type Options

A string type option consists of a hexadecimal-encoded colon-separated octet string.

Declaration: option mac-manufacturer code 211 = string;

Setting: option mac-manufacturer a4:80:1f;

Text Type Options

Text type options specify values encoded as ASCII text strings.

Declaration: option your-help-contact code 212 = text;

Setting: option your-help-contact 'John Smith';

IPv4 Address Type Options

The IP-address data type enables specification of an IPv4 address or resolvable domain name.

Declaration: option our-file-server code 213 = ip-address;

Setting: option our-file-server 10.0.209.12;

option our-file-server fileserv1.ipamww.com;

IPv6 Address Type Options

The IP6- address data type enables specification of an IPv6 address

Declaration: option our-video-server code 214 = ip6-address;

Setting: option our-video-server fc01:273e:90a:2::b1 ;

option dhcp6.some-server code 1234 = array of ip6-address;

option dhcp6.some-server 3ffe:bbbb:aaaa:aaaa::1, 3ffe:bbbb:aaaa:aaaa::2;

Array Type Options

Array options provide a way to specify multiple values for boolean integer or IP address data type values (all of the same type) by simply inserting 'array of' before the data type. Array elements are defined when setting the option values using comma-separated values.

Declaration: option my-ip-array code 198 = array of ip-address;

Setting: option my-ip-array 10.0.100.1 10.100.0.1;

Note:

Options can contain arrays of any of the supported data types except for the text and string types, which aren’t currently supported in arrays.

Encapsulated Type Options (Option Spaces)

An option space groups multiple options, typically with a common purpose. This grouping of options can be 'encapsulated' within a single user-defined option code.

Consider the example of creating a db option space to specify some database connection suboptions.

Declaration: option space db;

Setting value:

option db.db-server code 1 = ip-address;

option db.loginid code 2 = text;

option db.db-name code 3 = text;

option database-encapsulated code 221 = encapsulate db;

The first line option space db; defines the db option space. Next three suboptions are defined within this space. Each suboption has a unique code which is typically numbered from 1 since these are suboption code values. These suboptions are encapsulated within the parent option of code 221 named database-encapsulated. The setting statements below would set values to suboptions 1 2 and 3 encapsulated within option 221.

Setting: option db.db-server 10.199.200.37;

option db.loginid 'database';

option db.db-name 'mydatabase';

Record Type Options

While array options provide specification of multiple elements of the same type, Record types options **enable specification of multiple elements of different types.** Each element of the record is specified in order in the UI.

The following example defines an option of data type record including an integer (16 bit) text boolean and IP address as input values..

Note:

Unlike arrays, record element values accept only space separated not comma separated.

Definition: option my-rec code 198 = { integer 16 text boolean ip-address };

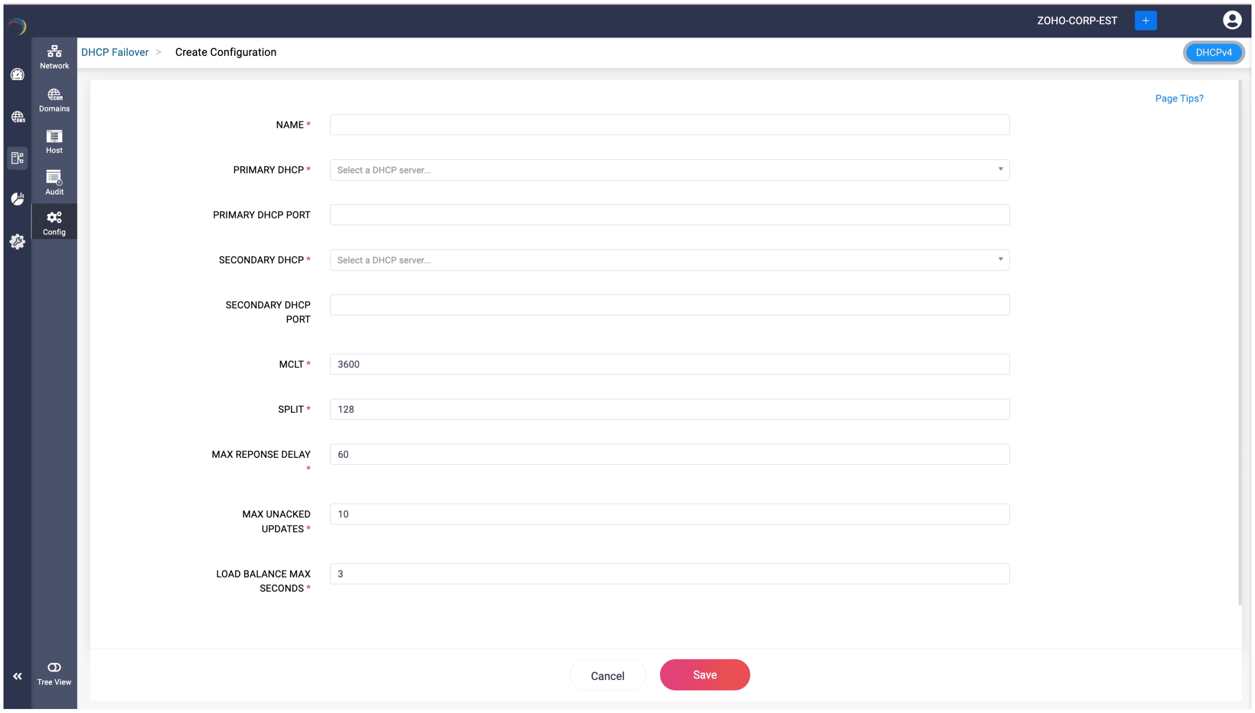
Values: option my-rec 4096 'cio' true 10.10.99.12;

Configuring DHCP Failover

Note: ManageEngine DDI Central does not offer DHCP failover for IPv6 address space. Failover is only available for IPv4 address space.

To configure the DHCP failover configurations:

- Go to **DHCP →Config→ DHCP Failover**
- Click on the **Add DHCP Failover** button on the top right corner.
- The **Create Configuration** page appears on the screen. Here enter the values for the fields as shown in the image



DHCP Failover > Create Configuration

NAME *	
PRIMARY DHCP *	Select a DHCP server...
PRIMARY DHCP PORT	
SECONDARY DHCP *	Select a DHCP server...
SECONDARY DHCP PORT	
MCLT *	3600
SPLIT *	128
MAX REPONSE DELAY *	60
MAX UNACKED UPDATES *	10
LOAD BALANCE MAX SECONDS *	3

Primary DHCP

The primary DHCP server is the main server responsible for handling DHCP requests and managing IP address leases. It works in conjunction with a secondary server to provide redundancy and high availability. The primary server typically handles the majority of DHCP requests and coordinates with the secondary server to ensure lease database synchronization and service continuity.

Primary DHCP Port

Specify the network port number that the primary DHCP server will use for its operations. The default port for DHCP services is typically 67 for servers. However, in certain network configurations or for specific security or operational reasons, you might choose to customize this port number. It's important to ensure that this port is consistent and properly configured in both the primary and secondary DHCP servers to facilitate smooth communication and service operation.

Secondary DHCP

Specify the configuration of the backup DHCP server. In a DHCP failover setup, there are typically two servers: a primary and a secondary. The secondary server is on standby to take over the DHCP responsibilities if the primary server becomes unavailable. This ensures continuity of service.

Secondary DHCP Port

Allows you to specify the network port number that the secondary DHCP server will use for communication. The standard port for DHCP is 67 for servers and 68 for clients, but this setting can be customized if needed.

MCLT (Maximum Client Lead Time)

MCLT is a crucial parameter in DHCP failover configurations. It defines the maximum time that a DHCP client can extend its lease on an IP address without contacting the server. This setting is important for ensuring consistency between the primary and secondary DHCP servers in terms of lease information. It is defined only on the primary DHCP server.

Split

Split is a special property that enables you to specify the percentage of the IP traffic to be handled by your Primary and Secondary Servers as a means of load balancing. It is defined only on the primary DHCP server. Its values range from 0 to 256.

A value 256 indicates no load balancing. Even if the failover is enabled for a DHCP server. The primary DHCP Server is the one solely responsible for listening and serving the address requests.

A value 0 indicates that most of the requests are handled by the Secondary DHCP server configured to handle the Failover .

A value of 128 means 50-50 load balancing where both the primary and the secondary DHCP servers configured for a specific range of IP addresses equally listen and serve the IP resources.

Max Response Delay

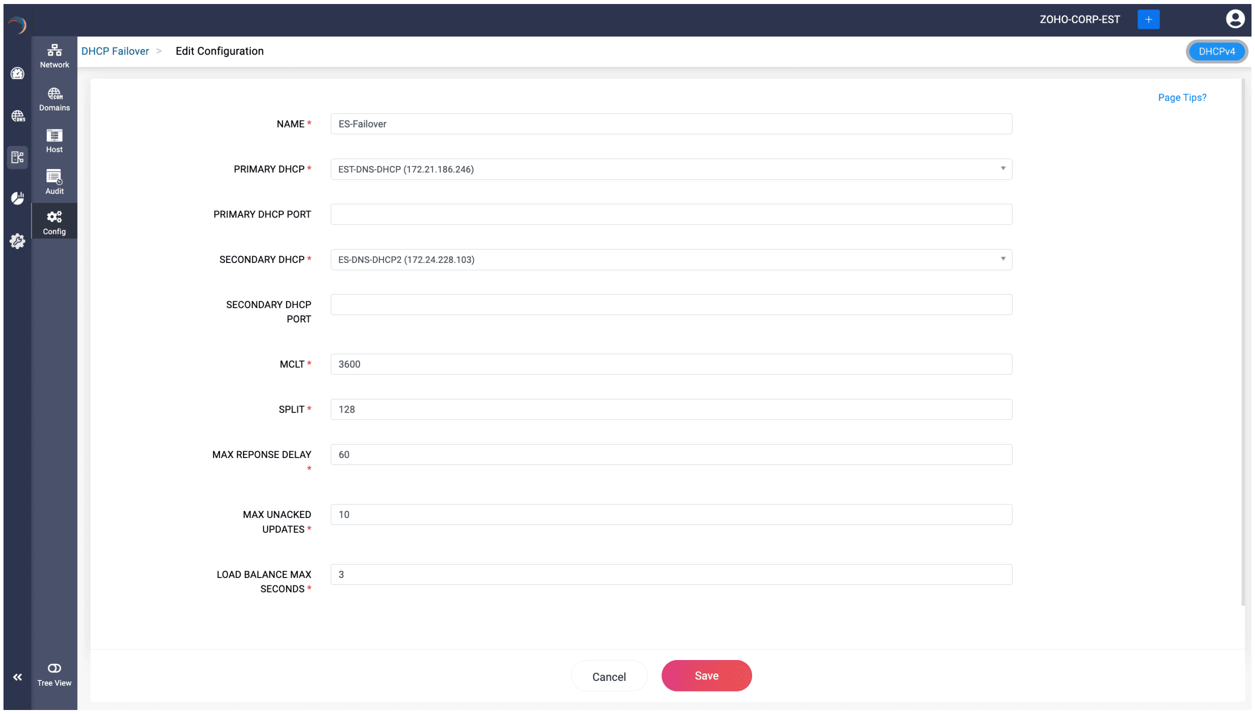
Determines the maximum time a DHCP server will wait before responding to a client request. This parameter is important for efficient allocation of IP leases and ensures that clients are not left waiting too long for a response, which could lead to network access issues.

Max Unacked Updates

In a DHCP failover configuration, the primary and secondary servers synchronize lease information with each other. This setting controls the maximum number of updates (regarding lease information) that can be sent from one server to another without receiving an acknowledgment. It's important for ensuring that both servers have consistent and up-to-date lease information.

Load Balance Max Seconds

This parameter sets the maximum time a DHCP server in a failover pair will wait to receive a response from its partner during load balancing operations. It ensures that if one server is not responding (possibly due to being down or overloaded), the other server can take over more of the load to maintain service continuity.



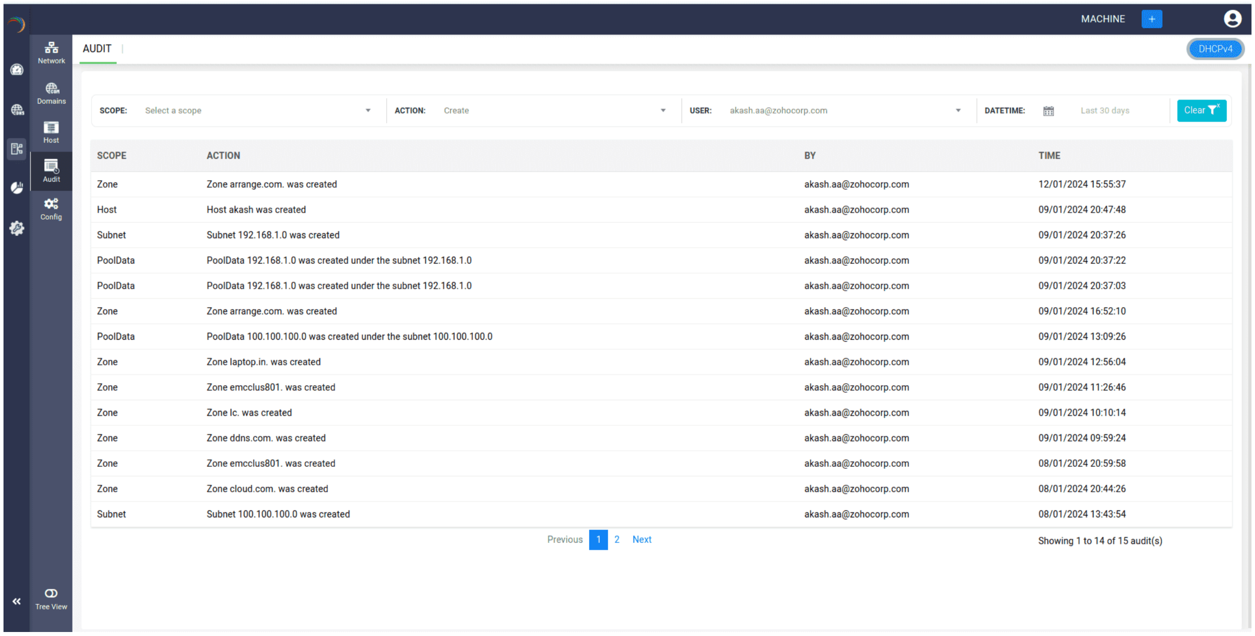
NAME *	ES-Failover
PRIMARY DHCP *	EST-DNS-DHCP (172.21.186.246)
PRIMARY DHCP PORT	
SECONDARY DHCP *	ES-DNS-DHCP2 (172.24.228.103)
SECONDARY DHCP PORT	
MCLT *	3600
SPLIT *	128
MAX REPONSE DELAY *	60
MAX UNACKED UPDATES *	10
LOAD BALANCE MAX SECONDS *	3

Click **Save** to bring the failover configurations into effect. Select the failover server you create while enabling failover for each scope you define in DDI Central.

DHCP Scope Audit Logs

The DHCP scope audit logs page provides you an overview of the actions performed on each scope configured in your network. It help you to continuously evaluate the overall security posture of your scopes using security audit logs to track the who, what, and when with respect to each DDNS Zone, VLAN, Supernet, Custom Options, Options, Pool Data, Client Class, Host, Host Group, Shared Network and Subnet.

Access the DHCP scope audit logs by navigating to **DHCP→ Audit** .



SCOPE: Select a scope ACTION: Create USER: akash.aa@zohocorp.com DATETIME: Last 30 days

SCOPE	ACTION	BY	TIME
Zone	Zone arrange.com. was created	akash.aa@zohocorp.com	12/01/2024 15:55:37
Host	Host akash was created	akash.aa@zohocorp.com	09/01/2024 20:47:48
Subnet	Subnet 192.168.1.0 was created	akash.aa@zohocorp.com	09/01/2024 20:37:26
PoolData	PoolData 192.168.1.0 was created under the subnet 192.168.1.0	akash.aa@zohocorp.com	09/01/2024 20:37:22
PoolData	PoolData 192.168.1.0 was created under the subnet 192.168.1.0	akash.aa@zohocorp.com	09/01/2024 20:37:03
Zone	Zone arrange.com. was created	akash.aa@zohocorp.com	09/01/2024 16:52:10
PoolData	PoolData 100.100.100.0 was created under the subnet 100.100.100.0	akash.aa@zohocorp.com	09/01/2024 13:09:26
Zone	Zone laptop.in. was created	akash.aa@zohocorp.com	09/01/2024 12:56:04
Zone	Zone emcclus801. was created	akash.aa@zohocorp.com	09/01/2024 11:26:46
Zone	Zone lc. was created	akash.aa@zohocorp.com	09/01/2024 10:10:14
Zone	Zone ddns.com. was created	akash.aa@zohocorp.com	09/01/2024 09:59:24
Zone	Zone emcclus801. was created	akash.aa@zohocorp.com	08/01/2024 20:59:58
Zone	Zone cloud.com. was created	akash.aa@zohocorp.com	08/01/2024 20:44:26
Subnet	Subnet 100.100.100.0 was created	akash.aa@zohocorp.com	08/01/2024 13:43:54

Previous 1 2 Next

Showing 1 to 14 of 15 audit(s)

You can also filter the logs for filtering the specific activities carried out by a specific user, or a specific activity carried out around a certain time frame on a particular DHCP scope to detect security breeches and malpractices.

Regularly reviewing your DHCP security logs helps you ensure that the access control mechanisms are performing adequately, determine whether users are sticking to your security practices, and catch new potential security weaknesses.

About IP Address Management

IPAM is a comprehensive system designed to plan, track, and manage IP address space within a network for smooth identification and communication. It provides a centralized repository for IP address information, offering administrators a bird’s eye view of the network infrastructure to oversee the allocation and usage of IP addresses.

ManageEngine DDI Central's IP Address Manager serves as the linchpin in maintaining a robust and efficient network infrastructure. By seamlessly integrating with your DNS and DHCP services, IPAM emerges as a key player in the evolving landscape of network administration, offering not just solutions but a proactive approach to network management.

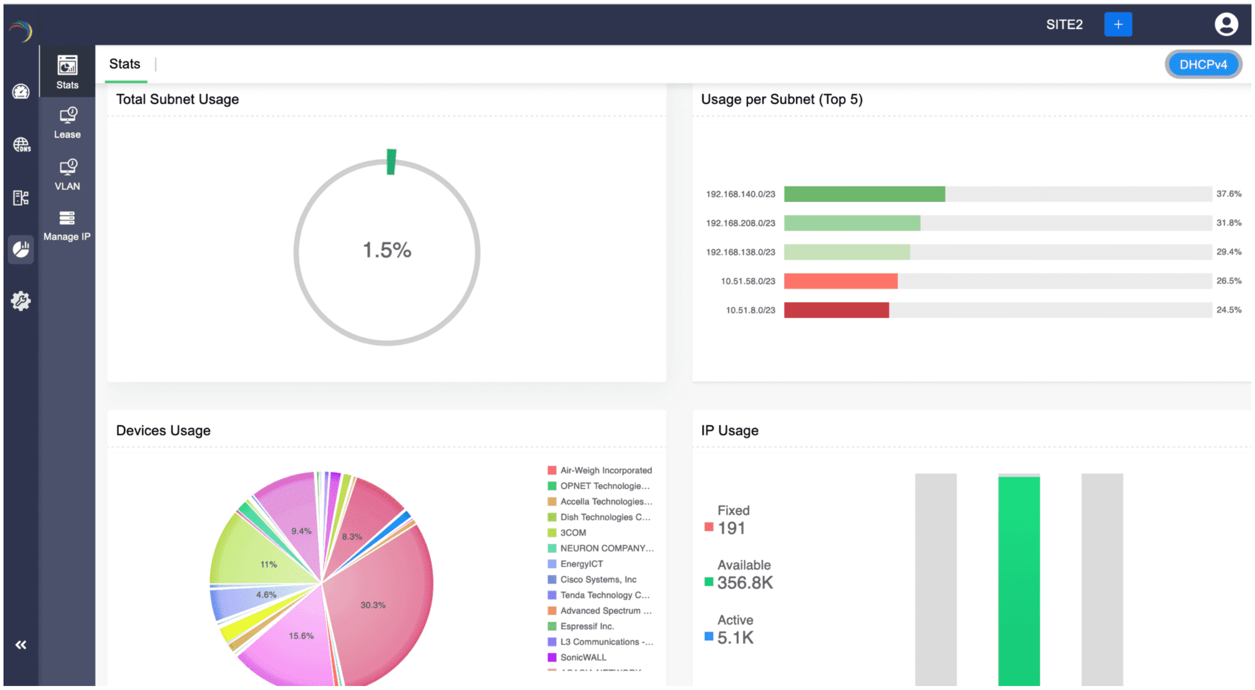
The IPAM Stats Dashboard

The IPAM stats dashboard presents an overview of the DHCP scopes or network topological units.

To access ManageEngine DDI Central's IPAM:

1. Log into ManageEngine DDI Central with your login credentials.
2. Select the **IPAM** menu from the left menu bar. With the IPAM selected, an inner menu bar appears parallel to the left menu bar.
3. By default, the **Stats** menu is selected in the inner menu bar.

The IPAM stats offering insights on the IP address inventory managed by your DHCP server, referred as the **Stats** dashboard appears on screen. View IP address statistics for DHCPv4 and DHCPv6 address spaces using the toggle at the top right corner.



Total Subnet Usage

Provides quick insights on the overall utilization of all subnets within a network infrastructure. It provides an aggregate view of the allocated and available subnets, offering a comprehensive understanding of the subnet distribution across the network.

Usage per Subnet

Displays a bar graph showcasing the top 5 subnets with the highest IP utilization percentage within each subnet managed by the DHCP server.

Devices Usage

Device Usage statistics provide insights into the distribution of assigned IPs among various hardware vendors.

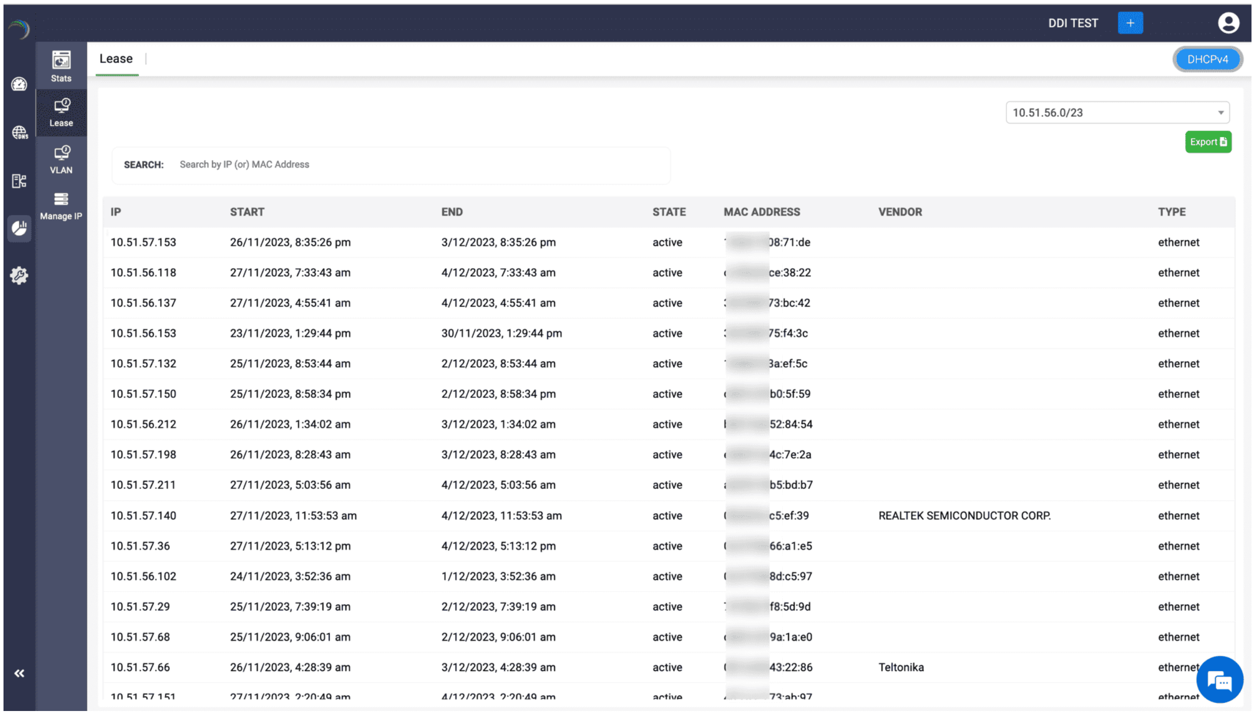
IP Usage

Displays bar graphs illustrating the total number of fixed, available, and active IPs in the entire IPv4/IPv6 address space managed by the DHCP server.

Analyzing Lease and Lease History

To access the lease records of the DHCP server:

- Select the **Lease** menu from the left inner menu bar.



IP	START	END	STATE	MAC ADDRESS	VENDOR	TYPE
10.51.57.153	26/11/2023, 8:35:26 pm	3/12/2023, 8:35:26 pm	active	[redacted]08:71:de		ethernet
10.51.56.118	27/11/2023, 7:33:43 am	4/12/2023, 7:33:43 am	active	[redacted]ce:38:22		ethernet
10.51.56.137	27/11/2023, 4:55:41 am	4/12/2023, 4:55:41 am	active	[redacted]73:bc:42		ethernet
10.51.56.153	23/11/2023, 1:29:44 pm	30/11/2023, 1:29:44 pm	active	[redacted]75:f4:3c		ethernet
10.51.57.132	25/11/2023, 8:53:44 am	2/12/2023, 8:53:44 am	active	[redacted]3a:ef:5c		ethernet
10.51.57.150	25/11/2023, 8:58:34 pm	2/12/2023, 8:58:34 pm	active	[redacted]b0:5f:59		ethernet
10.51.56.212	26/11/2023, 1:34:02 am	3/12/2023, 1:34:02 am	active	[redacted]52:84:54		ethernet
10.51.57.198	26/11/2023, 8:28:43 am	3/12/2023, 8:28:43 am	active	[redacted]4c:7e:2a		ethernet
10.51.57.211	27/11/2023, 5:03:56 am	4/12/2023, 5:03:56 am	active	[redacted]b5:bd:b7		ethernet
10.51.57.140	27/11/2023, 11:53:53 am	4/12/2023, 11:53:53 am	active	[redacted]c5:ef:39	REALTEK SEMICONDUCTOR CORP.	ethernet
10.51.57.36	27/11/2023, 5:13:12 pm	4/12/2023, 5:13:12 pm	active	[redacted]66:a1:e5		ethernet
10.51.56.102	24/11/2023, 3:52:36 am	1/12/2023, 3:52:36 am	active	[redacted]8d:c5:97		ethernet
10.51.57.29	25/11/2023, 7:39:19 am	2/12/2023, 7:39:19 am	active	[redacted]f8:5d:9d		ethernet
10.51.57.68	25/11/2023, 9:06:01 am	2/12/2023, 9:06:01 am	active	[redacted]9a:1a:e0		ethernet
10.51.57.66	26/11/2023, 4:28:39 am	3/12/2023, 4:28:39 am	active	[redacted]43:22:86	Teltonika	ethernet
10.51.57.151	27/11/2023, 2:20:49 am	4/12/2023, 2:20:49 am	active	[redacted]73:ab:97		ethernet

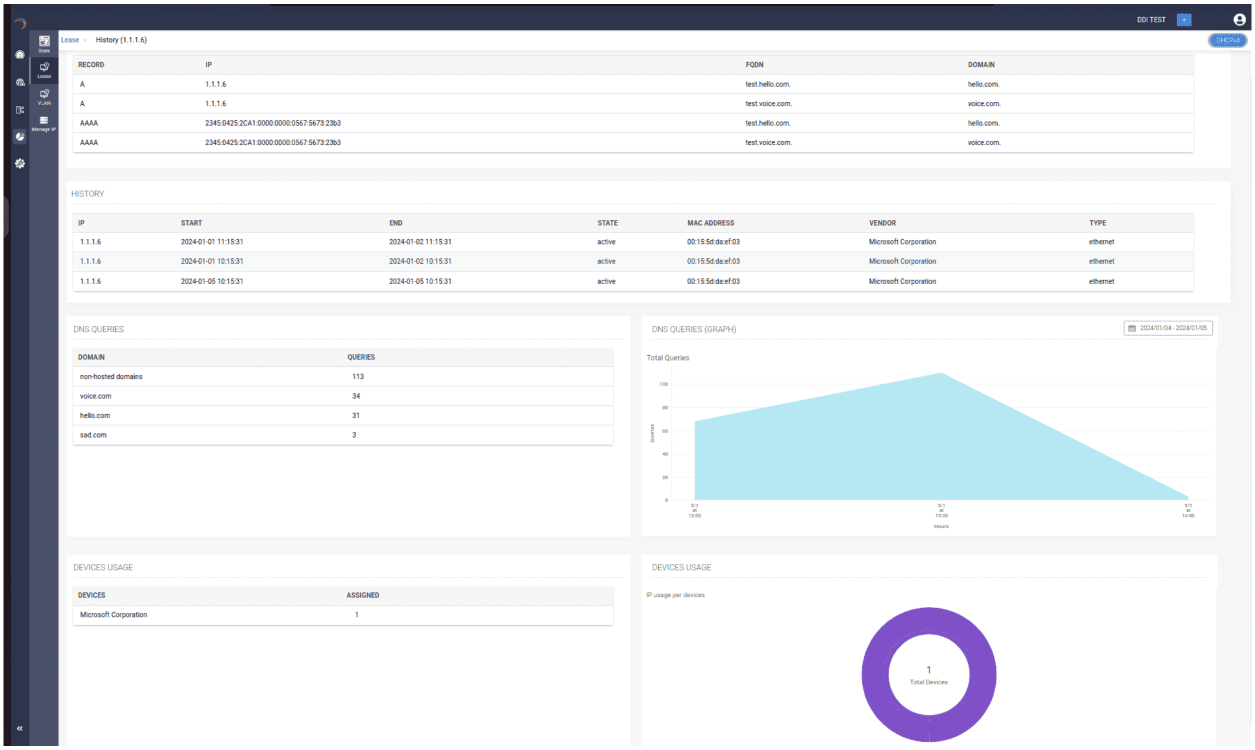
- The **Lease** page appears displaying the list of IP addresses currently leased by the DHCP server from a specific subnet, along with its the total lease duration, the current availability state of the IP, the MAC address and the manufacturer details of the host device associated with the IP during the lease period.

Note:

You can also export these lease records and download them as a CSV file for future references.

To select a different subnet, click on the dropdown box at the top right end and choose the specific subnet by its network address.

Click on an IP address to probe through the lease history of that particular address.



RECORD	IP	FQDN	DOMAIN
A	1.1.1.6	test.hello.com.	hello.com.
A	1.1.1.6	test.voice.com.	voice.com.
AAAA	2345:0425:2CA1:0000:0000:0567:5673:23b3	test.hello.com.	hello.com.
AAAA	2345:0425:2CA1:0000:0000:0567:5673:23b3	test.voice.com.	voice.com.

HISTORY

IP	START	END	STATE	MAC ADDRESS	VENDOR	TYPE
1.1.1.6	2024-01-01 11:15:31	2024-01-02 11:15:31	active	00:15:5d:da:ef:03	Microsoft Corporation	ethernet
1.1.1.6	2024-01-01 10:15:31	2024-01-02 10:15:31	active	00:15:5d:da:ef:03	Microsoft Corporation	ethernet
1.1.1.6	2024-01-05 10:15:31	2024-01-05 10:15:31	active	00:15:5d:da:ef:03	Microsoft Corporation	ethernet

DNS QUERIES

DOMAIN	QUERIES
non-hosted domains	113
voice.com	34
hello.com	31
sad.com	3

DEVICES USAGE

DEVICES	ASSIGNED
Microsoft Corporation	1

The **History** page for the IP address appears displaying the following sections:

DNS Relations

The DNS relations section displays a list of domain name records that was previously leased with the selected IP address. It includes information such as the type of record, the exact Fully Qualified Domain Name (FQDN) linked to the record, and the root domain of the record.

History

The history section provides a comprehensive audit trail detailing the evolution of the IP over time. It includes information on the host, identified by its MAC address and the manufacturer of the host device, to which the IP was leased and the duration of that lease. Additionally, it records the type of connection and precisely indicates the availability state of the IP during the entire span of the lease.

DNS Queries

The DNS Queries provides two sections to help you quantify and visualize the query volume handled by the IP during its association with various FQDNs.

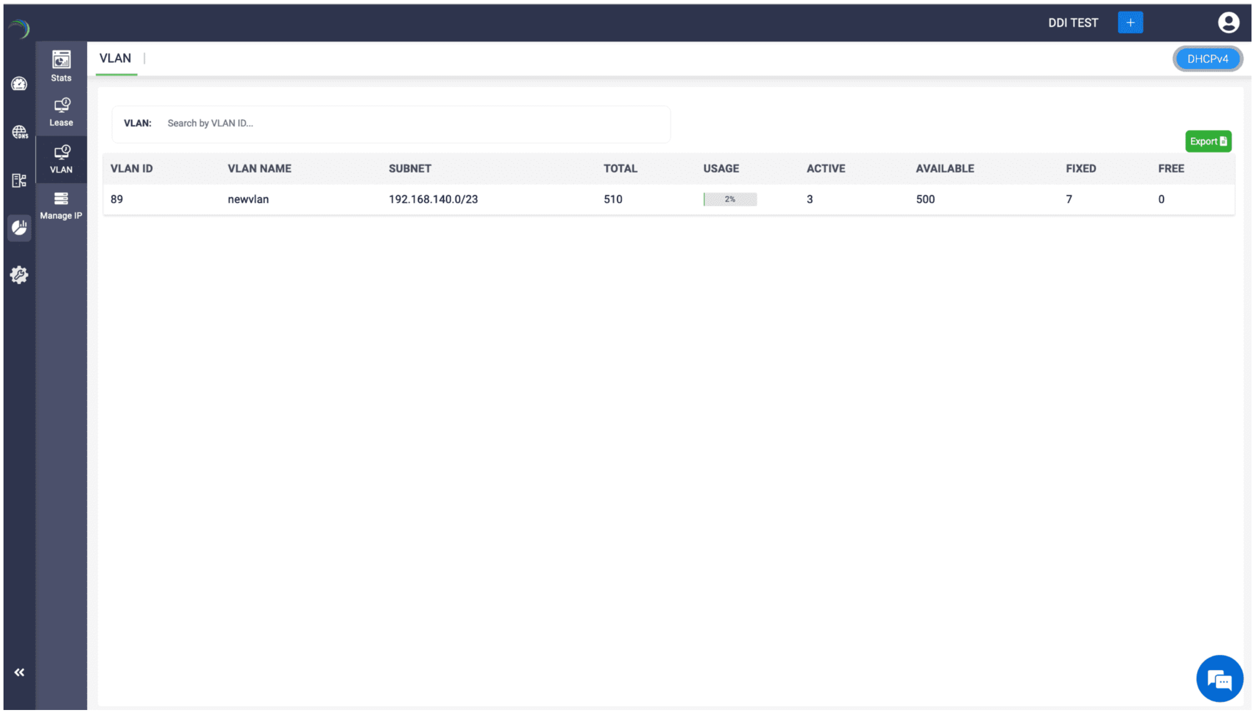
The section on the left, lists a historical overview of the total query volume to each specific FQDNs when the IP was associated with them.

Additionally, the section on the right helps you visualize, in the form of line graphs, the hourly query volume handled by the IP when it was associated with different domain names along a custom time frame. To analyze these hourly readings more accurately, make sure you select a custom time frame from the drop down calendar at the left corner within the same section.

Managing VLAN IP Address Inventory

To visualize and take control of the IP allocations for the VLANs managed by the DHCP server:

- Select the **DHCP → VLAN** menu from the left menu bar.

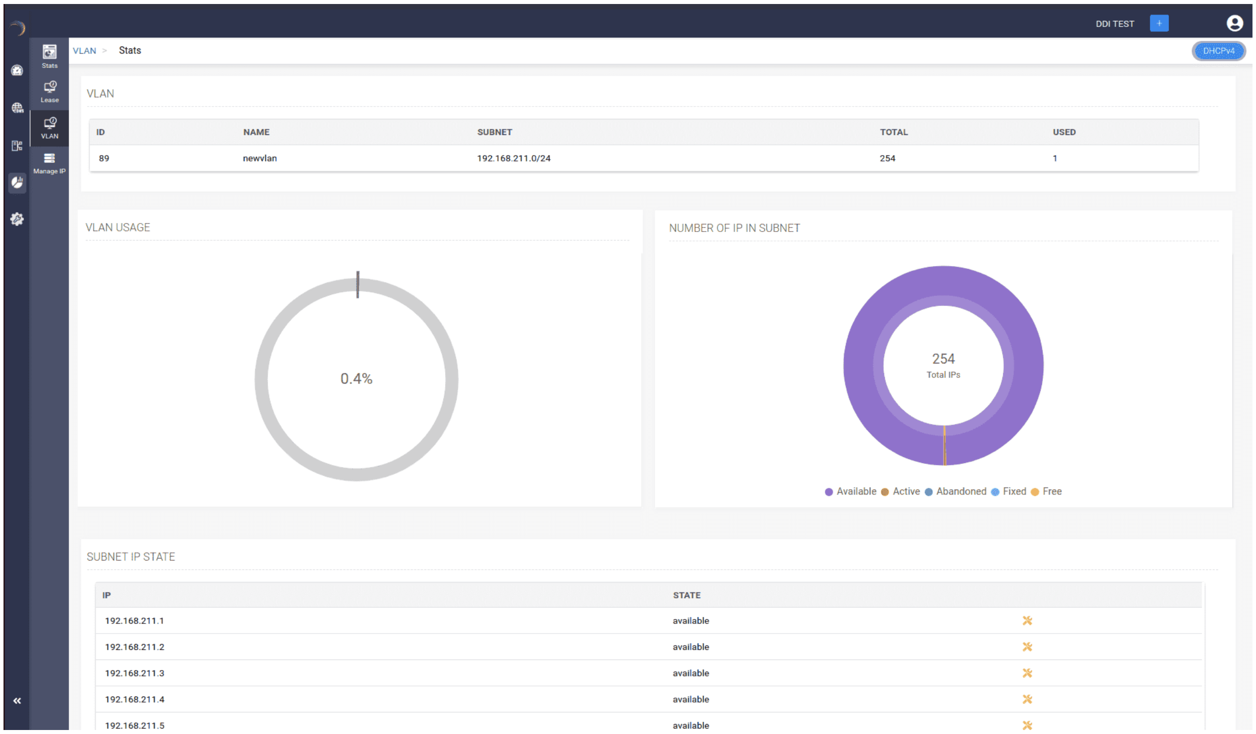


VLAN: Search by VLAN ID...

VLAN ID	VLAN NAME	SUBNET	TOTAL	USAGE	ACTIVE	AVAILABLE	FIXED	FREE
89	newvlan	192.168.140.0/23	510	2%	3	500	7	0

- The **VLAN** page appears displaying the list of all VLANS serviced by the DHCP server.

- Click on a VLAN entry from the list to view detailed stats on that specific VLAN from the **Stats** page.



VLAN

ID	NAME	SUBNET	TOTAL	USED
89	newvlan	192.168.211.0/24	254	1

SUBNET IP STATE

IP	STATE	
192.168.211.1	available	⚒
192.168.211.2	available	⚒
192.168.211.3	available	⚒
192.168.211.4	available	⚒
192.168.211.5	available	⚒

- Along the the top section of the **Stats** page, find the VLAN ID and the VLAN Name along with the essential details like: the dedicated subnet leasing IP addresses to the VLAN, the total number of available IPs in the subnet for the VLAN, the number of active IP addresses allocated to the VLAN.

- The mid section of the Stats page displays the following infographics:

  VLAN Usage: Illustrates the percentage of IP usage within the VLAN.

  Number of IPs in the subnet: A doughnut plot that illustrates and quantifies the volume of Subnet's IP addresses based on their availability states like:

  - **Available:** IPs that are not currently assigned and are ready for allocation.
  - **Active:** IPs that are currently in use and assigned to active devices on the network.
  - **Abandoned:** IPs that were previously assigned but are no longer in use or have been released.
  - **Fixed:** IPs that are reserved for specific devices or purposes, ensuring they are not allocated to other devices.
  - **Free:** Typically includes both IPs that are available for immediate assignment and those that are reserved but not currently in use.

At the bottom section, you'll discover a list of each IP address in the subnet responsible for provisioning IP addresses to the VLAN, along with their availability states. You can click on any Available or Free IP address to directly configure it for a host that is supposed to connect with the VLAN.

Note:

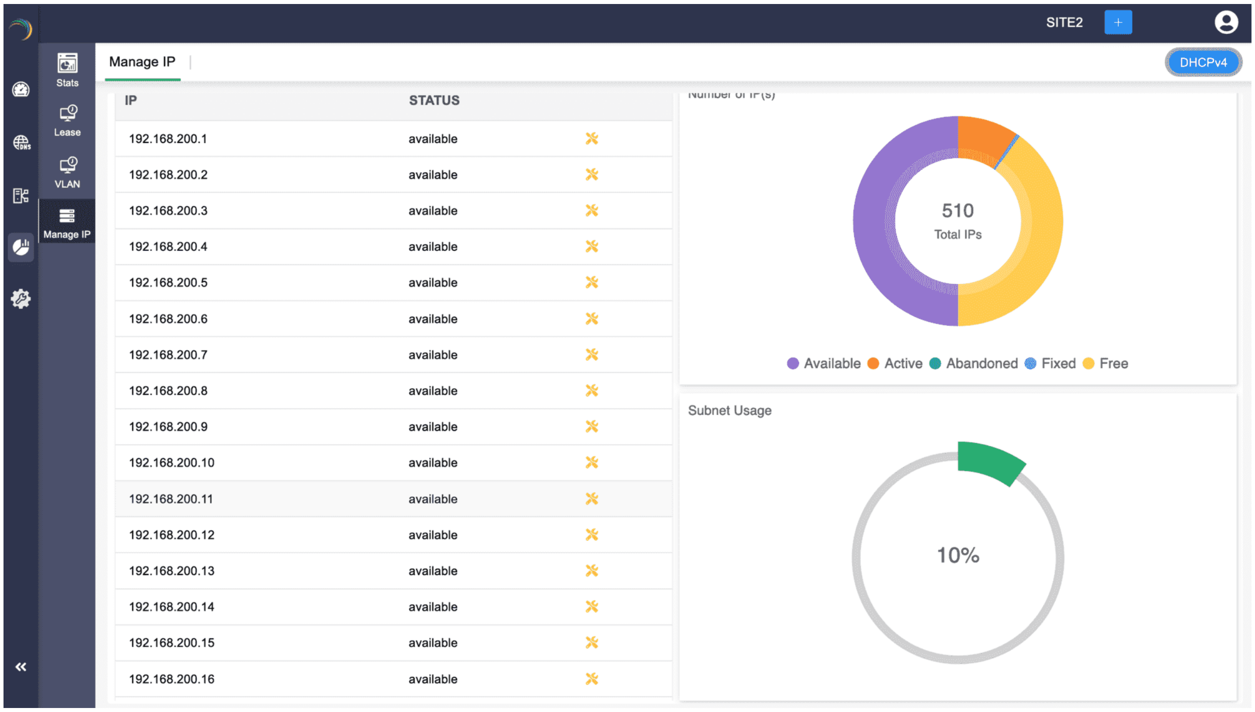
You can also search through the list for a specific VLAN by its VLAN ID. In addition, You can directly export the VLAN stats and download them as a CSV file for future references.

Managing Network IP Address Inventory

To visualize and take control of the IP allocations of each subnet managed by the DHCP server:

Select **DHCP → Manage IP** menu from the left menu bar.

The **Manage IP** tab displays the following three sections:



Manage IP

IP	STATUS	
192.168.200.1	available	🛠
192.168.200.2	available	🛠
192.168.200.3	available	🛠
192.168.200.4	available	🛠
192.168.200.5	available	🛠
192.168.200.6	available	🛠
192.168.200.7	available	🛠
192.168.200.8	available	🛠
192.168.200.9	available	🛠
192.168.200.10	available	🛠
192.168.200.11	available	🛠
192.168.200.12	available	🛠
192.168.200.13	available	🛠
192.168.200.14	available	🛠
192.168.200.15	available	🛠
192.168.200.16	available	🛠

The first section - Manage IP on the left displays the list of available/free IP(s) within the subnet. Click on any desired IP from the list. This directly takes you to the **Host** Page. Here you can directly assign the chosen IP address to any host or client.

The **second section- Number of IP (s)** on the top right visually depicts the volume of IP addresses in different availability states along a doughnut plot :

- **Available:** IPs that are not currently assigned and are ready for allocation.

- **Active:** IPs that are currently in use and assigned to active devices on the network.
- **Abandoned:** IPs that were previously assigned but are no longer in use or have been released.
- **Fixed:** IPs that are reserved for specific devices or purposes, ensuring they are not allocated to other devices.
- **Free:** Typically includes both IPs that are available for immediate assignment and those that are reserved but not currently in use.

The **third section-Subnet Usage,** on the bottom right corner, illustrates the overall percentage of IP utilization within that subnet.

Option Name	Option Description	Data Type	Block	Tags	Supported Versions	Example	Grammar
allow-new-zones	Controls whether zones can be added to the BIND nameserver at runtime using **rndc addzone**. The default value is no. If set to "yes," the BIND server would allow the dynamic addition of new zones through rndc addzone.	boolean	options, view	server,zone	all	Enter yes or no. Eg: yes	<boolean>
allow-notify	Controls which specific servers (identified by the [address-match-list]) are authorized to send **NOTIFY** messages to inform a name server about the changes to a specific zone. This is crucial for maintaining security and ensuring that only trusted servers trigger zone transfers. These specific servers defined by the allow-notify option are in addition to addresses defined in the primaries option for the zone. If not specified, by default the name server of a zone accepts Notify messages from the configured zone Primary server(s) .	address_list	view, options, zone(secondary)	transfer	all	Enter a list of valid ip_addresses/netprefixes/acl_names/server_keys enclosed within curly braces seperated by semicolons Eg: { 192.168.1.1; 192.168.1.2; }	{ <address_match_element>; ... }
allow-query	Defines an ACL to control who can query this server based on the address match list definition. [allow-query] may also be specified under the zone statement, in which case it overrides the [allow-query] under the options statement. If not specified, the default is to allow queries from all hosts.	address_list	zone(primary, secondary), view, options	query	all	Enter a list of valid ip_addresses/netprefixes/acl_names/server_keys enclosed within curly braces seperated by semicolons Eg: { 192.168.1.1; 192.168.1.2; }	{ <address_match_element>; ... }
allow-query-on	Specifies on which name server interface(s) queries will be accepted. For example, this option could be configured to allow queries on the interface(s) facing the internal network. For a query to be accepted, it must be allowed by both allow-query and allow-query-on ACLs. If a query is not permitted by both ACLs, it will be refused.	address_list	options, zone(primary, secondary), view	query	all	Enter a list of valid ip_addresses/netprefixes/acl_names/server_keys enclosed within curly braces seperated by semicolons Eg: { 192.168.1.1; 192.168.1.2; }	{ <address_match_element>; ... }

allow-query-cache	Specifies which hosts based on the address match list may receive query answers from the server's cache. If not specified this option defaults to the address match list specified in the allow-recursion option; if this is not set then that set in the allow-query option is used; otherwise this option defaults to {localnets; localhosts;}.	address_list	options, view	transfer	all	Enter a list of valid ip_addresses/netprefixes/acl_names/server_keys enclosed within curly braces seperated by semicolons Eg: { 192.168.1.1; 192.168.1.2; }	{ <address_match_element>; ... }
allow-query-cache-on	Specifies on which name server interface(s) queries will be accepted that may receive answers from the server's cache. For example this option could be configured to allow cache queries on the interface(s) facing the internal network.	address_list	options, view	transfer	all	Enter a list of valid ip_addresses/netprefixes/acl_names/server_keys enclosed within curly braces seperated by semicolons Eg: { 192.168.1.1; 192.168.1.2; }	{ <address_match_element>; ... }
allow-recursion	Defines an ACL on who can issue recursive queries to this server based on the address match list definition. If not specified this option defaults to the address match list specified in the allow-query-cache option; if this is not set then that set in the allow-query option is used; otherwise this option defaults to {localnets; localhosts;}.	address_list	view, options	dnssec, transfer	all	Enter a list of valid ip_addresses/netprefixes/acl_names/server_keys enclosed within curly braces seperated by semicolons Eg: { 192.168.1.1; 192.168.1.2; }	{ <address_match_element>; ... }
allow-recursion-on	Specifies on which name server interface(s) recursive queries will be accepted. For example this option could be configured to allow recursive queries on the interface(s) facing the internal network. The default is to accept recursive queries on all server interfaces.	address_list	options, view	server, query	all	Enter a list of valid ip_addresses/netprefixes/acl_names/server_keys enclosed within curly braces seperated by semicolons Eg: { 192.168.1.1; 192.168.1.2; }	{ <address_match_element>; ... }
allow-transfer	Specifies an ACL on who can receive a zone transfer from this server. The default is any.	string	options, view, zone(rimary, secondary)	dnssec, zone	all	Enter address_match_list of hosts that are allowed to transfer the zone information from this server. Eg: { 192.168.0.3; 192.168.0.4; }	[port <integer>] [transport <string>] { <address_match_element>; ... }

allow-update	Defines an ACL on who can perform a dynamic DNS update based on the address match list definition. The default is none. If the more granular update-policy option is specified within options view or zone blocks allow-update must not also be specified within the corresponding statement block.	address_list	view, options,zone(primary)	dnssec,server,query	all	Enter a list of valid ip_addresses/netprefixes/acl_names/server_keys enclosed within curly braces seperated by semicolons Eg: { 192.168.1.1; 192.168.1.2; }	{ <address_match_element>; ... }
allow-update-forwarding	Specifies an ACL defining from whom dynamic updates will be accepted for slave zones which will in turn be forwarded to the zone's master server. The default is none. ISC recommends using either any or none the default. This pushes the enforcement of update acceptance from this slave server to the master server.	address_list	view, options, zone(secondary)	dnssec, transfer	all	Enter a list of valid ip_addresses/netprefixes/acl_names/server_keys enclosed within curly braces seperated by semicolons Eg: { 192.168.1.1; 192.168.1.2; }	{ <address_match_element>; ... }
also-notify	Defines a set of IP addresses with or without corresponding port numbers to which to send Notify messages when a zone is updated (default = empty i.e. none). This option specifies additional Notify recipients to those specified in the zone's NS records.	string	options,zone, view	deprecated, logging	all	Enter the list of servers that are allowed to transfer the zone information. Eg: { 192.168.1.1; 192.168.1.2; }	[port <integer>] [source (<ipv4_address> | *)] [source-v6 (<ipv6_address> | *)] { (<remote-servers> | <ipv4_address> [port <integer>] | <ipv6_address> [port <integer>]) [key <string>] [tls <string>]; ... }
alt-transfer-source	Specifies an alternate transfer source IPv4 address and optionally port for performing inbound zone transfers or for issuing SOA queries or forwarded dynamic updates if the transaction failed with transfer-source parameters. Note that the use-alt-transfer-source yes option must be set.	string	options,view, zone(secondar)	dnssec	all	Enter alternative transfer source. Eg: 192.168.9.6	(<ipv4_address> | *) [port (<integer> | *)]
alt-transfer-source-v6	Specifies an alternate transfer source IPv6 address and optionally port for performing inbound zone transfers or for issuing SOA queries or forwarded dynamic updates if the transaction failed with transfer-source-v6 parameters. Note that the use-alt-transfer-source yes option must be set.	string	options,view, zone(secondary)	transfer	all	Enter alternative transfer-source-v6. Eg: 2001:db8::1	(<ipv6_address> | *) [port (<integer> | *)]
answer-cookie	Specifies whether the server will include the COOKIE EDNS options in responses. The default is yes and it is suggested this be set to no only to rectify operational problems	boolean	options	dnssec	all	Enter yes or no. Eg: yes	<boolean>

attach-cache	By default each view has its own cache database. This option enables the sharing of a common cache database across some or all views. When set in the options directive, all views will use the specified [cache-name] cache. Particular views may use their own cache by specifying a different cache-name within the view statement block. Cache sharing among views requires each view to support common cache-impacting parameters: check-names cleaning-interval dnssec-accept-expired dnssec-validation max-cache-ttl max-ncache-ttl max-cache-size and zero-no-soa-ttl.	string	options, view	view	all	Enter the cache name. Eg: named-cache	<string>
auth-nxdomain	Allows the server to always claim that a negative answer from its cache is actually authoritative even if it isn't; the default is no do not always claim authoritative answers.	boolean	options, view	dnssec	all	Enter yes or no. Eg: yes	<boolean>
auto-dnssec	This option defines the degree of automation for BIND's automated DNSSEC key and signature management features. The allow setting enables key updates and zone resigning when the user initiates the rndc sign zone command corresponding to this zone. maintain includes the allow setting capability and adds the automation of key activation revocation retirement and deletion according to each key's timing metadata as specified using the dnssec-keygen utility. create adds to the maintain functionality the ability to automatically create new keys when needed (Note - this has not yet been implemented). The off setting (default) disables automated DNSSEC management.	string	options, zone, view	dnssec	all	Valid values: allow, maintain, create or off Eg: allow	(allow | maintain | create | off)
automatic-interface-scan	Configures named to recan network interfaces on the server when interface addresses are added or removed. The default is yes.	boolean	options	dnssec	all	Enter yes or no. Eg: yes	<boolean>

avoid-v4-udp-ports	Specifies which port numbers to avoid as system-assigned source UDP ports over IPv4 typically to avoid firewall-blocked port numbers	string	options	dnssec	all	Enter a list of ports that are valid sources for UDP/IPv4 messages. Valid values: list of ports or port ranges Eg: { 7080; range 480 500; }	{ <portrange>; ... }
avoid-v6-udp-ports	Specifies which port numbers to avoid as system-assigned source UDP ports over IPv6 typically to avoid firewall-blocked port numbers	string	options	transfer	all	Enter a list of ports that are valid sources for UDP/IPv4 messages. Valid values: list of ports or port ranges Eg: { 7080; range 480 500; }	{ <portrange>; ... }
bindkeys-file	Specifies the pathname on the server for the trusted keys for use in DNSSEC Lookaside Validation. The default is /etc/bind.keys.	quoted_string	options	query	all	Enter the pathname of a file to override the built-in trusted keys provided by named. Eg: "/etc/bind/keys.bind"	<quoted_string>
blackhole	Defines an ACL defined by the address match list from which this server will not accept queries nor use to resolve a query. The default is none.	address_list	options	dnssec	all	Enter a list of valid ip_addresses/netprefixes/acl_names/server_keys enclosed within curly braces seperated by semicolons Eg: { 192.168.1.1; 192.168.1.2; }	{ <address_match_element>; ... }
check-dup-records	Configures the server to check its master zones for resource records that are treated differently by DNSSEC but are semantically equal in plain DNS. The default is warn.	string	options, view,zone	deprecated, dnssec, query	all	Enter fail, warn or ignore Eg: warn	(fail | warn | ignore)
check-integrity	When set to yes, it configures the server to perform zone integrity checks after the loading of master zones; the integrity check consists of assuring MX and SRV records refer to hosts that have corresponding A or AAAA records (intra-zone checks only) and that glue records exist for delegated zones. The default is yes.	boolean	options, view, zone	zone	all	Enter yes or no. Eg: yes	<boolean>
check-mx	Performs checking on MX records and will fail warn (default) or ignore based on whether the RDATA contains an IP address.	string	view, options, zone	dnssec, zone, logging	all	Enter fail, warn or ignore Eg: warn	(fail | warn | ignore)

check-names	Configures the server to validate owner names of A AAAA and MX records as well as RDATA names in NS SOA and MX records and also PTR records resolved based on queries for owners within ip6.arpa or in-addr.arpa zones. When defined within the options or view statement but not within zone declarations checking can be focused to master zones (default = fail) slave zones (default = warn) or responses received from other servers (response default = ignore).	string	options, view,zone	dnssec,zone	all	Grammmer: (primary/ master | secondary / slave | response) (fail | warn | ignore)	(primary / master | secondary / slave | response) (fail | warn | ignore)
check-sibling	Configures the server to verify that glue records (A/AAAA) exist for sibling zones i.e. other zones delegated by this server (as a common parent). For example the Rdata field of an NS record for a delegated zone may refer to a name server in a sibling zone: blog.manageengine.com. IN NS ns.products.manageengine.com. Setting this option to yes drives the server to verify that a glue (A/AAAA) record exists for ns.products.manageengine.com. The default value is yes. check-sibling only takes effect when check-integrity is set to "yes." This ensures that the DNS server performs integrity checks, including the verification of glue records for sibling zones. If not specified, the server, by default, verifies the existence of glue records for sibling zones when check-integrity is enabled.	boolean	options, view, zone (primary)	zone	all	Enter yes or no. Eg: yes	<boolean>
check-spf	If check-integrity is set, this option dictates whether to check for the presence of a TXT record if an SPF record is found. Sender Policy Framework (SPF) RR Types have been deprecated given the embedded deployments of SPF using the TXT record instead. The default value is warn.	string	options,view, zone(primary)	zone	all	Enter warn or ignore Eg: warn	(warn | ignore)

check-srv-cname	Configures the BIND server to verify that SRV records do not refer to CNAME records.This option is particularly relevant when the overall integrity of DNS responses is being checked (check-integrity is set to yes). **fail:** The server fails to process the SRV records that refer to CNAMEs. It may reject or ignore such records. **warn:** The server issues a warning when it encounters SRV records referring to CNAMEs. The processing continues, but a warning is logged. **ignore:** The server ignores the fact that SRV records refer to CNAMEs and proceeds with processing without generating warnings or errors. If check-srv-cname is not explicitly configured, the default response is set to warn. This makes the server to issue a warning when SRV records refer to CNAMEs, but it still processes the records.	string	options, view, zone (primary)	zone	all	Enter fail, warn or ignore Eg: warn	(fail | warn | ignore)
check-wildcard	If set to yes, it instructs the server to issue a warning upon detecting a non-fully resolvable wildcard (*) in its master zones. The default is yes.	boolean	options, view, zone(primary)	zone	all	Enter yes or no. Eg: yes	<boolean>
clients-per-query	Defines the minimum initial number of simultaneous outstanding recursive queries for a given name (i.e. of the same qname qtype qclass). In this context the server issuing such queries is the ""client"" referred to by the option name. (default = 10)	integer	options, view	server	all	Enter a valid integer. Eg:100	<integer>
cookie-algorithm	Defines the algorithm to be used when generating the server cookie, which serves as a lightweight DNS message authentication mechanism. The default is aes if supported by the server, otherwise siphash24.	string	options	server	all	Valid values: aes, siphash24 Eg: aes	(aes | siphash24)
cookie-secret	Specifies the shared secret used for generating and verifying EDNS COOKIE options within an anycast cluster. The shared secret is encoded as a hex string and needs to be 128 bits for AES128, 160 bits for SHA1 and 256 bits for SHA256. If not set, the system will generate a random secret at startup.	string	options	server	all	Use this option to pre-set the server cookie string. Eg:"8217891bcbbca1b7903069d20d20c4c2"	<string>

coresize	Defines the maximum size of a core dump file. The default is default which is the default core dump file size permitted by the operating system.	string	options	deprecated	all	Enter size in bytes followed by an additional suffix 'k', 'm or 'g'. Eg:3k	<sizeval>
datasize	Defines the maximum size of memory the server may use. The default is default which is the amount of memory allocated by the operating system by default though this option is useful when specifying a size greater than the operating system default if this amount is too small.	string	options	deprecated	all	Enter size in bytes followed by an additional suffix 'k', 'm or 'g'. Eg:3k	<sizeval>
deny-answer-addresses	Configures the server to filter out (drop) address (A or AAAA) query responses from external DNS servers where the address(es) contained in the answer section fall within the address_match_list definition to mitigate rebinding attacks. However, all address answers where the query name matches the [except-from name-list] will be accepted. For example a server configured with deny-answer-addresses {192.0.2.0/24;} except-from {"manageengine.com";}; will drop A records in the answer section containing an address within the 192.0.2.0/24 space except where the query name falls within the manageengine.com domain or subdomain.	string	view, options	query, Content filtering	all	Enter ipv4 addresses of A or AAAA records to be rejected. Eg:{ 192.168.1.1; 192.168.1.2; } except-from {ns3.example.com;}	{ <address_match_element>; ... } [except-from { <string>; ... }]

deny-answer-aliases	Configures the server to filter out (drop) alias (CNAME or DNAME) query responses from external DNS servers where the alias(es) contained in the answer section fall within the alias-list definition to mitigate rebinding attacks. However all alias answers where the query name matches the except-from name-list will be accepted. For example a server configured with deny-answer-aliases {"manageengine.com";} except-from {"blog.manageengine.com";}; will drop CNAME or DNAME records within the answer section of the response containing an answer within the manageengine.com domain or subdomains except where the query name falls within the blog.manageengine.com domain or subdomains.	string	options, view	query, Content fintering	all	Enter list of aliases of CNAME or DNAME records to be rejected. Eg: {ns1.example.com;ns2.example.com;} except-from {ns3.example.com;}	{ <string>; ... } [except-from { <string>; ... }]
dialup	Concentrates all communications between servers to the time when a dialup connection is made based on timing set in the heartbeat-interval option overriding the refresh timer to send out SOA (refresh) queries and NOTIFYs only at this interval. More granular control is available using: **notify** parameter, which directs the server to send only NOTIFYs during the connection with normal refresh processing. **notify-passive** parameter which indicates the server will send NOTIFYs during the heartbeat interval while suspending normal refresh processing. **refresh** suspends NOTIFYs during heartbeat intervals but sends refresh queries during the heartbeat interval. **passive** disables normal refresh processing notify-passive sends NOTIFYs during the heartbeat and suppresses refresh processing.	string	view, options, zone(primary, secondary)	transfer	all	Valid Values: notify, notify-passive, passive, refresh or <boolean> Eg: notify	(notify | notify-passive | passive | refresh | <boolean>)

directory	Specifies the location of the current working directory on the server. Any relative (non-absolute) pathnames are interpreted as relative to this directory. If a directory is not specified, the working directory defaults to ".", the directory from which the server was started.	quoted_string	options	server	all	Enter the server's working directory. Eg:"var/local/named/working directory"	<quoted_string>
disable-algorithms	Disables the specified DNSSEC algorithm(s) when processing queries for the specified domain and its subdomains. Multiple occurrences of this statement are permitted.	string	options, view	dnssec	all	Enter zone name and dnssec algorithms that need to be disabled. Eg: "example.com" { "NSECRSASHA1"; "DH"; }	<string> { <string>; ... }
disable-ds-digests	Disables specified DS/DLV digest types at and below the specified domain. Multiple statements are permitted.	string	view, options	zone, dnssec	all	Enter zone name and ds digest algorithms that need to be disabled. Eg: "example.com" { "SHA-384"; "SHA-256"; }	<string> { <string>; ... }
disable-empty-zone	Disables an individual empty zone identified by zone_name. Multiple statements are permitted.	string	options, view	zone, server	all	Enter zone name in which empty zones are to be disabled. Eg: "zone_name"	<string>

dns64	Supports the DNS64 IPv4-IPv6 co-existence strategy by allowing an IPv6 host to connect to an IPv4 destination via a NAT64 gateway, whose IP address is a concatenation of the specified IPv6 prefix and an IPv4 address returned via A record queries (when no native AAAA record answers are provided). The DNS64 service provides this mapping function.
	The **clients** parameter indicates an address match list of clients for whom the service is provided; the default is any. The **mapped** parameter indicates which IPv4 addresses within the A resource record set shall be mapped to corresponding AAAA answers. The **exclude** pa**rameter** defines that any queried IPv6 addresses falling within the specified network will not be subject to DNS64 translation. If set to exclude { 2001:db8::/32; }, DNS64 translation will be bypassed for IPv6 addresses within the specified network.The **suffix** can be used to specify additional bits to include in the mapped response following the IPv4 address (the default is ::). The **recursive-only** parameter indicates whether to apply DNS64 mapping to recursive queries only, and the **break-dnssec** will not modify(add or remove) DNSSEC records from the authoritative server response if the value is **no** and will do so if the value is set to **yes.**
		string	options, view	query	all		dns64 <netprefix> {
							break-dnssec <boolean>;
							clients { <address_match_element>; ... };
							exclude { <address_match_element>; ... };
							mapped { <address_match_element>; ... };
							recursive-only <boolean>;
							suffix <ipv6_address>;
							}

dns64-contact	Supports the DNS64 IPv4-IPv6 co-existence strategy as discussed above. This option defines the administrative contact name that will appear in the SOA record for the ipv6.arpa zone corresponding to the mapped AAAA records created by appending the IPv4 address to the IPv6 prefix during a DNS64 transaction.	string	options, view	server	all	Enter the name of the contact for dns64 zones. Eg: "contacts.example.com"	<string>

dns64-server	Supports the DNS64 IPv4-IPv6 co-existence strategy as described above. This option defines the DNS server name that will appear in the SOA record for the ipv6.arpa zone corresponding to the mapped AAAA records created by appending the IPv4 address to the IPv6 prefix during a DNS64 transaction.	string	view, options	server	all	Enter the name of the server for dns64 zones. Eg: "ns1.example.com"	<string>
dnskey-sig-validity	Defines the number of days in the future when DNSSEC signatures that are automatically generated for DNSKEY RRsets as a result of dynamic updates will expire. This option is disabled if set to 0; otherwise it overrides the sig-validity-interval option for DNSKEY records. The maximum value is 3660 (10 years).	integer	options, view, zone	deprecated	all	Enter a valid integer. Eg:100	<integer>
dnsrps-enable	Enables or disables the DNS Response Policy Service (DNSRPS) API, which enables use of an external response policy provider as an alternative to response policy zones.	boolean	options, view	server, security	all	Enter yes or no. Eg: yes	<boolean>
dnsrps-options	Configures the DNS Response Policy Service (RPS) provider library, librpz; the text is passed to the library, concatenated with settings derived from the response policy statement.	string	options, view	server, security	all		{ <unspecified-text> }
dnssec-accept-expired	Instructs the server to accept expired signatures for DNSSEC validation. The default is no.	boolean	view, options	query	all	Enter yes or no. Eg: yes	<boolean>
dnssec-dnskey-kskonly	This option is a parameter for BIND's automated DNSSEC key and signature management features. When set to yes and update-check-ksk is set to yes only KSKs will be used to sign the DNSKEY,CDNSKEY, and CDS RRsets at the zone apex; otherwise ZSKs may be used to sign the DNSKEY RRset. When the option **update-check-ksk** is set to **no** this option is ignored.	boolean	view, zone, options	query	all	Enter yes or no. Eg: yes	<boolean>
dnssec-enable	Enables or disables DNS Security Extensions (DNSSEC) validation in the BIND DNS server. When set to "yes," it indicates that DNSSEC validation should be enabled, enhancing the security of DNS responses by verifying their cryptographic signatures.	boolean	options	security	9.9,9.11,9.16	Enter yes or no. Eg: yes	<boolean>

dnssec-loadkeys-interval	Specifies the interval between checks for new keys or changes in key timing metadata when auto-dnssec maintain; is configured. The default is 60 (minutes) the minimum value is 1 and the maximum value is 1440.	integer	options, view,zone(primary, secondary)	query	all	Enter a valid integer. Eg:100	<integer>
dnssec-lookaside	refers to the mechanism for managing DNS Security Extensions (DNSSEC) trust anchors using a lookaside validation approach. When configured, BIND can use a separate repository, known as a lookaside, to fetch and manage DNSSEC trust anchors. This allows administrators to maintain and update trust anchors outside of the DNS tree, providing flexibility in managing DNSSEC trust relationships.	string	options	query	9.9,9.11,9.16	Enter a valid domain name. Eg:dnssec-lookaside "manageengine.com"; This example suggests that BIND should use a lookaside validation approach for DNSSEC and fetch trust anchors associated with the "manageengine.com" domain. The actual value would depend on the specific configuration and requirements of the DNSSEC implementation.	<string>
dnssec-must-be-secure	Specifies a domain (including subdomains) that must provide secure resolution as validated by trusted-key configuration or DLV when set to yes. When set to no secure resolution is not required for this domain.	string	view, options	dnssec	all		<string> <boolean>
dnssec-policy	Defines a DNSSEC key and signing policy (KASP) for a zone. This is a string referring to a dnssec-policy block. The default is none.	quoted_string	options, view, zone(primary)	dnssec	9.16,9.18		<string>
dnssec-secure-to-insecure	When set to yes this allows the DNSKEY record(s) to be deleted in the zone(s) via BIND's automated DNSSEC key and signature management features introduced in BIND 9.7.0. Deleting these records effectively transitions the zone(s) from secure to insecure. The default is no. If set to yes, and if the DNSKEY RRset at the zone apex is deleted, all RRSIG and NSEC records are removed from the zone as well.	boolean	options, view, zone(primary)	dnssec	all	Enter yes or no. Eg: yes	<boolean>

dnssec-update-mode	Configures automated signing of new or changed resource records and automated resigning of RRSets when nearing signature expiration when set to **maintain**. When set to **no-resign** new or changed resource records will be signed but automated resigning of RRSets when nearing signature expiration will be disabled.	string	options, view	dnssec	all	Valid values: maintain, no-resign Eg: maintain	(maintain | no-resign)
dnssec-validation	Turns on DNSSEC validation processing when set to yes. dnssec-enable must also be set to yes. The default is yes.	string	options, view	dnssec	all	Valid values: yes, no, auto Eg: auto	(yes | no | auto)
dnstap	Defines message types to be logged under the dnstap query logging feature. Message type can be client, auth, resolver, forwarder, or all. Specifying type all causes all dnstap messages to be logged, regardless of type. Each type may take an additional argument to indicate whether to log query messages or response messages; if not specified, both queries and responses are logged.	string	options, view	logging	all	Eg:{auth; client response; resolver query;}	{ (all | auth | client | forwarder | resolver | update) [(query | response)]; ... }
dnstap-identity	Defines an identity to include in dnstap messages. The default is hostname, i.e., the server's hostname.	string	options	logging	all	Enter an identity string to send in dnstap messages. example: "my-dns"	(<quoted_string> | none | hostname)
dnstap-output	dnstap logging destination including specification of destination as a file or a UNIX domain socket followed by the path of the file or socket.	string	options	logging	all	Enter the path to which the dnstap frame stream should be sent. Eg: unix "/var/run/bind/dnstap.sock"	(file | unix) <quoted_string> [size (unlimited | <size>)] [versions (unlimited | <integer>)] [suffix (increment | timestamp)]
dnstap-version	Specifies a version string to inclue in dnstap messages.	string	options	logging	all	Enter version value in quoted string Valid values: quoted version string, none Eg:"version2"	(<quoted_string> | none)

dscp	Specifies the value of the differentiated services code point (DSCP) in the IPv4 header to classify outgoing DNS traffic on operating systems that support DSCP. Valid values for ip_dscp are 0-63 and the default is "not configured". It is now **obsolete** and has no effect.	integer	options, view, zone(secondary)	logging	all	Enter a valid integer. Eg:100	<integer>
dual-stack-servers	Specifies external name server IP addresses or hostnames that have access to both IPv4 and IPv6 transport. This option has no effect if the server on which this option is configured is itself dual-stacked.	string	view, options	dnssec	all	Eg: {"usr/app" port 70; 192.168.0.1 port 80;}	[port <integer>] { (<quoted_string> [port <integer>] | <ipv4_address> [port <integer>] | <ipv6_address> [port <integer>]); ... }
dump-file	Specifies the file pathname to place the dump file when told to dump its database via rndc dumpdb; the default is named_dump.db	quoted_string	options	logging	all	Enter the pathname of the file where the server dumps the database after rndc dumpdb. Eg:"/var/named/dumpfile"	<quoted_string>
edns-udp-size	Defines the advertised EDNS UDP buffer size in bytes ranging from 512 to 4096 (default)	integer	server, options, view	query	all	Enter a valid integer. Eg:100	<integer>
empty-contact	Specifies the zone contact that will appear in the SOA record created in empty zones. If not specified "." is used.	string	options, view	zone, server	all	Enter the contact name in the returned SOA record for empty zones. Eg: "one.example.com"	<string>
empty-server	Specifies the server name that will appear in the SOA record created in empty zones. If not specified the empty zone's name will be used.	string	options, view	zone, server	all	Enter the server name in the returned SOA record for empty zones. Eg: "ns1.example.com"	<string>
empty-zones-enable	Enables (yes) or disables (no) creation of empty zones on the server. Empty zones are enabled by default.	boolean	options, view	zone, server	all	Enter yes or no. Eg: yes	<boolean>

fetch-quota-params	Defines parameters for the dynamic resizing of the fetches-per-server option in response to detected congestion. The number parameter indicates how often the moving average ratio of timeouts to responses should be calculated based on the number of queries received (default = 100 queries). The remaining arguments define the low ratio threshold (default 0.1), the high threshold (default 0.3) and the discount parameter (default 0.7) respectively where a higher discount weighs more recent events higher than earlier events. This option requires BIND to be built with configure -enable-fetchlimit .	string	options, view	query, server	all	Enter the parameters for dynamic resizing of the fetches-per-server quota in response to detected congestion. Eg:100 0.1 0.3 0.4	<integer> <fixedpoint> <fixedpoint> <fixedpoint>
fetches-per-server	Defines the maximum number of simultaneous iterative queries that may be sent to a single name server. The default is 0 which indicates no limit. This quota is dynamically adjusted based on the setting of the fetch-quota-params option. The optional drop or fail keyword indicates whether the server shall drop those queries exceeding the quota with no response or respond with a SERVFAIL. The default is fail.	string	options, view	query, server	all	Enter the maximum number of simultaneous iterative queries allowed to be sent by a server to an upstream name server before the server blocks additional queries. Eg:100	<integer> [(drop | fail)]
fetches-per-zone	Defines the maximum number of simultaneous iterative queries that may be sent for a given domain. The default is 0 which indicates no limit. This quota is dynamically adjusted based on the setting of the fetch-quota-params option. The optional drop or fail keyword indicates whether the server shall drop those queries exceeding the quota with no response or respond with a SERVFAIL. The default is fail.	string	options, view	query, server	all	Enter the maximum number of simultaneous iterative queries allowed to any one domain before the server blocks new queries for data in or beneath that zone. Eg:200	<integer> [(drop | fail)]
files	Defines the maximum number of files the DNS service may have open concurrently. The default is unlimited.	integer	options	query	all	Enter a valid integer. Eg:100	<integer>
flush-zones-on-shutdown	When signaled to exit via the SIGTERM signal the server will discard any pending zone writes from journal files; the default is no indicating zone writes should first be performed	boolean	options	zone	all	Enter yes or no. Eg: yes	<boolean>

forward	Configures the server to either: use only those servers configured in the forwarders statement to resolve queries (forward only) or to first query a server listed in the forwarders statement and upon receiving no resolution answer query another server (e.g. based on cached information or hints file configuration) (forward first).	string	options, zone, view	query	all	Enter first or only Eg: first	(first | only)
forwarders	Specifies the IP address(es) of servers to query when using forwarding. The default is an empty list i.e. no forwarding but when the empty list is used within a zone statement while forwarders are configured within the server options statement then those forwarders are enabled on the server but not for the zone with the empty forwarders list (i.e. acts as negation).	string	options, zone, view	query	all	Eg:port 8080 {192.168.0.1 port 80;}	[port <integer>] [tls <string>] { (<ipv4_address> | <ipv6_address>) [port <integer>] [tls <string>]; ... }
fstrm-set-buffer-hint	Configures the threshold number of bytes to accumulate in the output buffer before forcing a buffer flush in the high speed framing library, libfstrm used by dnstap. The minimum is 1024, the maximum is 65536, and the default is 8192.	integer	options	logging	all	Enter a valid integer. Eg:100	<integer>
fstrm-set-flush-timeout	Defines the number of seconds to allow unflushed data to remain in the output buffer in the high speed framing library, libfstrm used by dnstap. The minimum (and default) is 1 second, the maximum is 600 seconds (10 minutes).	integer	options	logging	all	Enter a valid integer. Eg:100	<integer>
fstrm-set-input-queue-size	Specifies the number of queue entries to allocate for each input queue for the high speed framing library, libfstrm used by dnstap. This value must be a power of 2. The minimum is 2, the maximum is 16384, and the default is 512.	integer	options	logging	all	Enter a valid integer. Eg:100	<integer>
fstrm-set-output-notify-threshold	The number of outstanding queue entries to allow on an input queue before waking the I/O thread for the high speed framing library, libfstrm used by dnstap. The minimum is 1 and the default is 32.	integer	options	logging	all	Enter a valid integer. Eg:100	<integer>

fstrm-set-output-queue-model	Controls the queuing semantics to use for queue objects for the high speed framing library, libfstrm used by dnstap. The default is mpsc (multiple producer, single consumer); the other option is spsc (single producer, single consumer).	string	options	logging	all	Enter the queuing semantics to use for queue objects. Valid values: mpsc or spsc Eg: mpsc	(mpsc | spsc)
fstrm-set-output-queue-size	Defines the number of queue entries to allocate for each output queue for the high speed framing library, libfstrm used by dnstap. The minimum is 2, the maximum is system-dependent, and the default is 64.	integer	options	logging	all	Enter a valid integer. Eg:100	<integer>
fstrm-set-reopen-interval	Defines the number of seconds to wait between attempts to reopen a closed output stream for the high speed framing library, libfstrm used by dnstap. The default is 5 seconds, the minimum is 1 second, and the maximum is 600 seconds (10 minutes).	duration	options	logging	all	Enter time in seconds or a valid ISO 8601 duration Eg: 40	<duration>
geoip-directory	Defines the directory containing the GeoIP .dat database files for GeoIP initialization. By default this option is not configured and the libGeoIP built-in directory is used for GeoIP features.	string	options	server	all	Enter the directory containing GeoIP database or none. Eg: "/usr/data/geoip"	(<quoted_string> | none)
glue-cache	Enables caching of address (A and AAAA) glue records to speed performance when adding these records tothe additional section of DNS response messages. The default is yes.	boolean	options, view	deprecated	all	Enter yes or no. Eg: yes	<boolean>
heartbeat-interval	Defines the heartbeat interval governing frequency of tasks for zones defined with the dialup option set to a value other than no (default = 60 [minutes]).	integer	options	deprecated	all	Enter a valid integer. Eg:100	<integer>
hostname	Defines a host name to be provided in response to a TXT query of class CHAOS for owner hostname.bind. The default is the hostname of the server on which named is running as determined by a gethostname() call. Setting hostname_string to none disables processing of these queries.	string	options	server	all	Enter the hostname of the server to return in response to a hostname.bind query. Eg: "example.com"	(<quoted_string> | none)

http-listener-clients	Sets a limit on the number of concurrent DoH clients (HTTP/2 connections) on a per-listener basis, rather than globally. This is important because it allows for more granular control over resource allocation and can prevent idle HTTP clients from hogging resources that could be used by other TCP clients. The http-listener-clients option sets a default quota size for each listener, which can be overridden by a listener-clients option within an http clause in the BIND configuration. The default value for http-listener-clients is 300. Setting it to 0 disables the quota facility, which is useful for testing and benchmarking purposes. We settled for the value 300 for now because this value is large enough to serve some clients while not large enough to let the server be abused too much, taking into consideration that it might need to serve clients over other DNS transports.	integer	options	server	all	Enter a valid integer. Eg:100	<integer>

http-streams-per-connection	Sets a hard limit on the number of concurrent HTTP/2 streams that can be open on a single DoH connection. When this limit is reached, the HTTP/2 session will be closed by the server. This is a crucial setting to prevent overloading the server with too many concurrent requests on a single connection. The default limit is set to 100 streams per connection, based on the assumption that most libnghttp2-based clients will not exceed this number. However, administrators can lower this number if needed. It can be set globally using the http-streams-per-connection option or within an individual http clause using the streams-per-connection option. Setting the limit to 0 disables it, which is not recommended except for benchmarking in controlled environments. Together, the two options (http-listener-clients and http-streams-per-connection) allow administrators to effectively manage the load that HTTP clients can place on a DNS server operating over HTTP/2, ensuring efficient resource usage and preventing server overload.	integer	options	transfer	all	Enter a valid integer. Eg:100	<integer>
http-port	Specifies the port number on which the DNS server listens for incoming HTTP connections. This is typically used for DNS-over-HTTP (DoH) services, where DNS queries are sent over HTTP rather than traditional DNS transport protocols. By setting this option, an administrator can define a custom port for HTTP traffic if the default one is not suitable or needs to be changed due to conflicts or security policies.	integer	options	query,server	all	Enter a valid integer. Eg:100	<integer>

inline-signing	The inline-signing DNS option is related to DNSSEC (Domain Name System Security Extensions), which adds security measures to the DNS protocol to counteract threats like cache poisoning and man-in-the-middle attacks. When inline-signing is enabled for a zone, the DNS server automatically signs the zone's data with DNSSEC keys according to the configured parameters. This process happens "inline" during the zone publishing process, which means that the server generates DNSSEC signatures on-the-fly as it serves responses, simplifying the management of signed zones. This is particularly useful for operators who want to maintain unsigned zone files while still serving signed DNS data, as it removes the need for separate zone-signing procedures.	boolean	dnssec-policy, zone	dnssec, zone	all	Enter yes or no. Eg: yes	<boolean>
interface-interval	Defines the interval governing the frequency of scans for new or removed network interfaces on the server to begin listening on new interfaces and stop listening on deleted interfaces as permitted with corresponding listen-on settings. The default is 60 [minutes].	duration	options	server	all	Enter time in seconds or a valid ISO 8601 duration example: 40	<duration>
ipv4only-contact	This configuration option specifies a contact address or mechanism that is used exclusively for IPv4 communication. It is used to define an email, IP address, or a URL for contacting the DNS server administrator or support team using only IPv4. This is important in networks that maintain separate contact points for IPv4 and IPv6 for administrative or technical reasons.	string	options, view	server	all	Enter the contact for the IPV4ONLY.ARPA zone created by dns64. Eg: "contact.example.com"	<string>

ipv4only-enable	This setting is typically used to enable or disable a service or feature that is intended to operate only over IPv4. By setting this to true (or enabled), the DNS server might restrict its operations to IPv4, either globally or for a specific service, such as zone transfers or dynamic updates. This could be useful for compatibility with legacy systems or in environments where IPv6 is not supported or desired.	boolean	options, view	query	all	Enter yes or no. Eg: yes	<boolean>
ipv4only-server	This option defines the server or set of servers that the DNS service should use exclusively for IPv4 traffic. It may designate specific DNS servers that are reachable only via IPv4, or it could configure the server to respond to DNS queries only over IPv4. Such a configuration might be necessary in networks that are segregated based on IP versions or where IPv6 connectivity is not reliable or available.	string	options, view	query	all	Enter the name of the server for the IPV4ONLY.ARPA zone created by dns64. Eg: "ns1.example.com"	<string>
ixfr-from-differences	When set to yes the server will compute the differences between a new version of a zone (upon reload as a master or zone transfer receipt as a slave) and use the differences between these for IXFR processing. The parameters master and slave may be defined at the view and zone statements to apply this processing to master zones or slave zones respectively within the view or zone.	string	options, view, zone	transfer	all	Valid values: primary, master, secondary, slave or a boolean value. Eg: master	(primary | master | secondary | slave | <boolean>)
keep-response-order	Specifies the set of addresses with the address match list to which the server will send responses to TCP queries in the same order in which they were received. The default is none.	address_ list	options	dnssec	all	Enter a list of valid ip_addresses/netprefixes/acl_ names/server_keys enclosed within curly braces seperated by semicolons Eg: { 192.168.1.1; 192.168.1.2; }	{ <address_match_element>; ... }

key-directory	The full directory pathname in which public and private key files are stored on the server for processing of dynamic updates of DNSSEC secure zones. If not specified the current working directory is used.	quoted_string	options, view, zone	dnssec	all	Enter the directory where the public and private DNSSEC key files should be found. Eg: "/var/dnssec/keyfolder"	<quoted_string>
lame-ttl	Defines the number of seconds the server will cache a lame server designation; i.e. a given server is not authoritative for a zone that's delegated to it (default = 600 [seconds]).	duration	view, options	server	all	Enter time in seconds or a valid ISO 8601 duration example: 40	<duration>
listen-on	Specifies the network interface the server listens for queries; the default is to listen on port 53 on all interfaces. Multiple listen-on statements may be defined.	string	options	server	all	Eg: { 192.168.1.1; 192.168.1.2; }	[port <integer>] [tls <string>] [http <string>] { <address_match_element>; ... }
listen-on-v6	Specifies the network interface parameters on which the server will listen for queries using IPv6 transport. If this option is not specified the server will not listen on any of the server's IPv6 addresses unless BIND was invoked with the -6 option when it will listen on all IPv6 interfaces.	string	options	server	all	Eg: { 192.168.1.1; 192.168.1.2; }	[port <integer>] [tls <string>] [http <string>] { <address_match_element>; ... }
lmdb-mapsize	Sets the maximum size for the memory map of the new-zone database (NZD) in Lightning Memory-Mapped Database (LMDB) format when BIND is built with liblmdb. The LMDB stores zone configuration information when using rndc addzone. The default is 32MB.	string	view, options	server	all	Enter size in bytes followed by an additional suffix 'k', 'm or 'g'. Eg:3k	<sizeval>
lock-file	Specifies the pathname of a file on which named will attempt to acquire a file lock when starting up for the first time as confirmation another server is not already running. Setting path_name to none disables this feature and the default is /var/run/named/named.lock.	string	options	server	all	Enter the pathname of the file on which named attempts to acquire a file lock when starting for the first time. Valid values: none, filepath Eg: "/var/locks/named.lock"	(<quoted_string> | none)
managed-keys-directory	The directory in which files used to track managed keys are located. By default this is the named working directory.	quoted_string	options	dnssec	all	Enter the directory in which to store the files that track managed DNSSEC keys. Eg: "/var/named/example.mkeys"	<quoted_string>

masterfile-format	Specifies the format of zone files on the server. The default is text . Setting to raw will omit some name checking features and setting to map uses an image of a BIND 9 in-memory zone database but is very server architecture specific.	string	options, zone, view	zone, server	all	Valid values: raw, text Eg: raw	(raw | text)
masterfile-style	When masterfile-format is set to text , this option specifies whether a dump of the zone files is formatted in multi-line format with owner names expressed relative to a shared origin when set to relative which may be easier for human consumption or with fully qualified owner names when set to full which may be easier for script processing.	string	options, zone, view	server	all	Valid values: full, relative Eg: full	(full | relative)
match-mapped-addresses	Specifies that the server should map IPv4 addresses associated with an IPv4-mapped IPv6 address against defined address match lists for processing. This option is intended solely for use as a work around for a Linux kernel quirk for IPv6-enabled Linux servers.	boolean	options	query	all	Enter yes or no. Eg: yes	<boolean>
max-cache-size	Sets the maximum memory size to be used for the server's cache. If using DNS views the specified size applies to the cache size for each view. When the amount of data in the cache approaches the limit the server will prematurely expire records to remain within the bound (default = 0 which means that records are purged from cache when their TTLs expire).	string	view, options	dnssec,zone	all	Enter default, unlimited, or percentage size or enter size in bytes followed by an additional suffix 'k', 'm or 'g'. Eg:3k	(default | unlimited | <sizeval> | <percentage>)
max-cache-ttl	Defines the maximum retention time for cached [positive] information. The default is 7 days.	duration	options, view	server	all	Enter time in seconds or a valid ISO 8601 duration example: 40	<duration>
max-clients-per-query	Defines the maximum number of simultaneous outstanding recursive queries for a given name (i.e. of the same qname qtype qclass) before dropping additional clients. In this context the server issuing such queries is the "client" referred to by the option name (default = 100).	integer	options, view	server	all	Enter a valid integer. Eg:100	<integer>

max-ixfr-ratio	Sets the threshold expressed as a percentage of pending ixfr size to the full zone size above which an AXFR will be used instead of an IXFR for a zone transfer request. The default, unlimited, disables ratio checking.The minimum percentage value is 1%.	string	options, zone, view	transfer	all	Valid values: unlimited, percentage value Eg: 25	(unlimited | <percentage>)
max-journal-size	Specifies the maximum size of each journal file. The default is unlimited.	string	options, zone, view	transfer	all	Enter size of journal files. Valid values: default, unlimited, size in bytes followed by additional suffix 'k', 'm' or 'g'. Eg: 1g	(default | unlimited | <sizeval>)
max-ncache-ttl	Defines the maximum number of seconds the server will cache negative answers. The default is 10800 [seconds] or 3 days and the maximum value is 7 days.	duration	options, view	server	all	Enter time in seconds or a valid ISO 8601 duration example: 40	<duration>
max-records	Specfies the maximum number of records permitted in a zone. The default is 0 which means unlimited.	integer	options, view, zone	zone, transfer	all	Enter a valid integer. Eg:100	<integer>
max-recursion-depth	Defines the maximum number of resolution redirections permitted for a given query. A redirection occurs when resolving a domain name requires the resolution of another name. The default is 7.	integer	options, view	server	all	Enter a valid integer. Eg:100	<integer>
max-recursion-queries	Defines the maximum number of iterative queries that may be sent for a given recursive query. The root and TLD iterative queries are not counted against this max and the default is 75.	integer	options, view	query, server	all	Enter a valid integer. Eg:100	<integer>
max-refresh-time	Defines the maximum refresh interval for SOA refresh attempts to the master.	integer	options, view, zone	transfer	all	Enter a valid integer. Eg:100	<integer>
max-retry-time	Defines the maximum retry time at which the server should retry a failed zone transfer.	integer	options, view, zone	transfer	all	Enter a valid integer. Eg:100	<integer>
max-rsa-exponent-size	Defines the maximum RSA exponent size that will be accepted when validating DNSSEC responses (in bits). Valid values are 0 (default, equivalent to 4096), 35 to 4096.	integer	options	dnssec, query	all	Enter a valid integer. Eg:100	<integer>

max-stale-ttl	If the stale answers feature is enabled (via option stale-answer-enable yes or rndc serve-stale on), this option sets the maximum time beyond the TTL expiry of a record to retain it in cache. The default is one week.	duration	options, view	server	all	Enter time in seconds or a valid ISO 8601 duration example: 40	<duration>
max-transfer-idle-in	Specifies a limit on the duration of idle time during an inbound zone transfer (default = 60 [minutes]). Once exceeded the zone transfer will be terminated.	integer	options, view, zone	transfer	all	Enter a valid integer. Eg:100	<integer>
max-transfer-idle-out	Specifies a limit on the duration of idle time during an outbound zone transfer (default = 60 [minutes]). Once exceeded the zone transfer will be terminated.	integer	options, view, zone	transfer	all	Enter a valid integer. Eg:100	<integer>
max-transfer-time-in	Specifies a limit on the duration of an inbound zone transfer (default = 120 [minutes]). Once exceeded the zone transfer will be terminated.	integer	options, view, zone	transfer	all	Enter a valid integer. Eg:100	<integer>
max-transfer-time-out	Specifies a limit on the duration of an outbound zone transfer (default = 120 [minutes]). Once exceeded the zone transfer will be terminated.	integer	options, view, zone	transfer	all	Enter a valid integer. Eg:100	<integer>
max-udp-size	Defines the maximum EDNS UDP packet size the server will send in bytes ranging from 512 to 4096 (default)	integer	options, server, view	query	all	Enter a valid integer. Eg:100	<integer>
max-zone-ttl	Defines the maximum permissible TTL value for all zones or a particular zone on the server. This is useful when rolling DNSSEC keys to to enable the to-be-rolled key to remain available until corresponding RRSIG records have expired from cahces.	string	options, view, zone, dnssec-policy	zone, query	all	Enter maximum permissible ttl in seconds. Valid values: unlimited, ISO duration, Eg: 10	(unlimited | <duration>)
memstatistics	Turns on (yes) or off (no) writing of memory statistics to the file specified in the memstatistics-file option. The default is no unless named was started with the "-m record" switch.	boolean	options	logging, server	all	Enter yes or no. Eg: yes	<boolean>

memstatistics-file	This specifies the pathname of the file to which the server will write memory usage statistics. The default is named.memstats.	quoted_string	options	logging, server	all	Enter the pathname of the file where the server writes memory usage statistics on exit. Eg: "/var/bind/example.memstats".	<quoted_string>
message-compression	Configures the server to use DNS name compression for regular queries (compression is always used for incremental or absolute zone transfers)	boolean	options, view	query	all	Enter yes or no. Eg: yes	<boolean>
min-cache-ttl	Defines the minimum time the server will cache affirmative answers. Valid values range from 0 to 90s.	duration	options, view	server	all	Enter time in seconds or a valid ISO 8601 duration example: 40	<duration>
min-ncache-ttl	Defines the minimum time the server will cache negative answers. Valid values range from 0 to 90s.	duration	options, view	server	all	Enter time in seconds or a valid ISO 8601 duration example: 40	<duration>
min-refresh-time	Defines the minimum SOA refresh time to query the master.	integer	options, view, zone	transfer	all	Enter a valid integer. Eg:100	<integer>
min-retry-time	Defines the minimum retry time at which the server should retry a failed zone transfer.	integer	options, view, zone	transfer	all	Enter a valid integer. Eg:100	<integer>
minimal-any	This option governs responses to ANY queries, i.e., RRType of "*" for a given qname. If set to yes, only one RRType (and associated DNSSEC signatures) for the queried name will be provided in the response instead of all RRTypes for the queried name if set to no (default).	boolean	options, view	query	all	Enter yes or no. Eg: yes	<boolean>
minimal-responses	When set to yes this option instructs the server to only add records to the authority and additional sections of the response when required e.g. for negative responses or delegations. When set to no-auth, the server will only add records to the authority section if required but may add records to the additional section. When set to no-auth-recursive, limiting of authority and additional section resource records applies to recursive queries. The default is no.	string	options, view	query	all	Valid values: no-auth, no-auth-recursive, boolean value(yes or no). Eg: yes	(no-auth | no-auth-recursive | <boolean>)

multi-master	When set to yes the server will not log when its serial number is greater than that on another master	boolean	options, view, zone	transfer	all	Enter yes or no. Eg: yes	<boolean>
new-zones-directory	Specifies the directory in which to store configuration parameters added via rndc addzone.	quoted_string	options, view	zone	all	Enter the directory where configuration parameters are stored for zones added by rndc addzone. Eg: "/var/named/new.zone"	<quoted_string>
no-case-compress	Responses to queriers within the scope of the address match list will include non-compression of case-sensitive answers. With case compression (default), example.com and example.COM are the same and hence compressed; with no-case-compression, both versions of the answer are included in the response.		options, view	server	all	Enter a list of valid ip_addresses/netprefixes/acl_names/server_keys enclosed within curly braces seperated by semicolons Eg: { 192.168.1.1; 192.168.1.2; }	{ <address_match_element>; ... }
nocookie-udp-size	Defines the maximum size in bytes of UDP responses to queries without a valid server cookie. The default is 4096 but the max-udp-size option may further limit the response size.	integer	options, view	query	all	Enter a valid integer. Eg:100	<integer>
notify	This option governs the sending of NOTIFY messages: yes - NOTIFY messages are sent to all servers with NS records for the zone except the zone master (primary) identified by the MNAME field of the zone's SOA record; NOTIFY messages are also sent to those defined in the also-notify option explicit - NOTIFY messages are sent only to those servers identified in the also-notify option. master-only - NOTIFY messages are sent only for master/primary zones no - no NOTIFY messages are sent	string	options	query	all	Valid values: explicit, master-only, primary, a boolean value. Eg: master-only	(explicit | master-only | primary-only | <boolean>)
notify-delay	This option defines the number of seconds to wait between sending sets of Notify messages. The default is 0.	integer	options, view, zone	transfer, zone	all	Enter a valid integer. Eg:100	<integer>
notify-rate	This option defines the rate of notify requests per second. The default is 20.	integer	options	transfer, zone	all	Enter a valid integer. Eg:100	<integer>

notify-source	Defines the server's network interface (IPv4 address) and optionally source UDP port for sending Notify messages.	string	options, server, view, zone	transfer	all	Enter the IPv4 address to be used for outgoing NOTIFY messages. Eg: 192.168.2.4	(<ipv4_address> | *)
notify-to-soa	Facilitates hidden master configurations when set to yes by instructing the server to send a Notify message as appropriate to the server listed in the SOA record master name (MNAME) field. In hidden master configurations MNAME may be configured with the name of a slave server. If set to no a Notify will not be sent to the server listed in the MNAME field.	boolean	options, view, zone	transfer	all	Enter yes or no. Eg: yes	<boolean>
nta-lifetime	This parameter configures the default time that a negative trust anchor (nta) is ignored when added via rndc nta. An nta disables DNSSEC validation for zones known to be failing validation due to misconfiguration. The duration may be entered using TTL-style formats for seconds, minutes or hours. The default is one hour.	duration	options, view	dnssec	all	Enter time in seconds or a valid ISO 8601 duration example: 40	<duration>
nta-recheck	Negative trust anchor (nta) configuration enables you to disable DNSSEC validation for a given domain due to know misconfiguration issues. Named will periodically issue a query to each nta domain to determine if it has been repaired, i.e., whether DNSSEC validation is accurate. This option sets the duration of the periodicity of these checks. These checks can be disabled by setting the valude to 0; the default is 5s.	duration	options, view	dnssec	all	Enter time in seconds or a valid ISO 8601 duration example: 40	<duration>
nxdomain-redirect	Defines a redirect namespace to replace an NXDOMAIN received from an authoritative server with the original query name plus the specified string. If a relevant zone of type redirect is defined, it shall override the setting of this option.	string	options, view	query	all	Enter redirect url for non existent domain Eg:redirect.example.com	<string>

parental-source	The concept of a "parental source" in DNS configurations involves using dedicated DNS IP addresses specifically for parental control. This setup aims to maintain consistent content filtering and protection, even during user IP address changes. When the parental control DNS receives queries from an unrecognized IP, it defaults to a safe mode, blocking all sites or allowing only a pre-approved list, thereby ensuring continuous protection and reducing the risk of bypassing content restrictions due to IP address updates. This approach requires a separate DNS infrastructure, distinct from standard DNS services, to manage and enforce these safety policies effectively.	string	options, view, zone	dnssec	all	Enter the local IPv4 source address to be used to send parental DS queries. Eg: 192.168.3.2	(<ipv4_address> | *)
pid-file	Specifies the pathname of the file to which the server writes its process ID. The default is /var/run/named.pid (pre BIND 9.6) or /var/run/named/named.pid (BIND 9.6+). If the pathname parameter is specified as none no pid file will be written.	string	options	server	all	Enter the filepath of pid-file. Valid values: none, filename enclosed in quotes Eg:"/var/process/example.pid "	(<quoted_string> | none)
port	Specifies the UDP/TCP port number used by the server for sending and receiving DNS messages. This option is intended primarily for server testing purposes as setting the value to other than 53 the default will inhibit communications with the global DNS	integer	options	transfer	all	Enter a valid integer. Eg:100	<integer>
preferred-glue	Specifies the preferred resource record type that will be specified first in the additional section of a query response for an NS record. The default is NONE no preference.	string	options, view	query	all	Valid values: A, AAAA or NONE Eg: A	<string>

prefetch	Specifies whether the server should refresh its cache for soon-to-expire cached data ensuring the cache always has an answer. The number parameter defines the trigger TTL at which prefetch will take place when a cached record with a lower TTL is encountered durign query processing. The default value is 0 which disables prefetch and other valid values are 1-10. The second optional parameter defines the eligibility TTL, or the smallest original TTL value that will be accepted for eligibility for prefetch. The default value is 9 and the value must be at least six seconds greater than the trigger TTL value.	string	options, view	query	all	Enter a trigger ttl and eligibility ttl seperated by space. Eg: 2 9	<integer> [<integer>]
provide-ixfr	Used in options or server statements to configure a server configured as master for its zones to honor IXFR requests from slaves or not.	boolean	options, server, view	transfer	all	Enter yes or no. Eg: yes	<boolean>
qname-minimization	Qname minimization calls for servers to convey the queryname (Qname) in queries in the context of the authoritative server being queried. For example, the server would include only the TLD in teh Qname when querying the root servers. This reduces the number of queries on the Internet with the fully qualified query intact to reduce exposure. Setting to strict follows this process as defined in RFC 7816 while relaxed (default value) supports this as well with a fallback to non-minimized qnames upon receipt of an NXDOMAIN or other error response. Disabled and off disables qname minimization on queries.	string	options, view	query	all	Valid values: strict, relaxed, disabled or off Eg: strict	(strict | relaxed | disabled | off)
query-source	Defines the local network interface (IPv4 address) and source port for UDP-based queries issued to other servers to obtain a query answer TCP-based queries always use a random source port and it's recommended that UDP also do so to reduce the risk of cache poisoning. Therefore the port parameter should generally not be specified.	string	options, server, view	query	all	Enter the IPV4 address used as the source for outgoing queries from the server. Eg:192.168.2.3	[address] (<ipv4_address> | *)

query-source-v6	Defines the local network interface (IPv6 address) and source port for UDP-based queries issued to other servers to obtain a query answer TCP-based queries always use a random source port and it's recommended that UDP also do so to reduce the risk of cache poisoning. Therefore the port parameter should generally not be specified.	string	options, server, view	query	all	Enter the IPV6 address used as the source for outgoing queries from the server. Eg: 2001:db8::1	[address] (<ipv6_address> | *)
querylog	When set to yes logging of queries is enabled upon named startup; query logging is otherwise determined by the queries logging category setting	boolean	options	logging, server	all	Enter yes or no. Eg: yes	<boolean>

rate-limit	Enables specification of parameters designed to minimize the use of this server in amplifying reflection denial of service attacks which inundate a spoofed (target) IP address. The server will limit nearly identical answers for a given IP address (or addresses within a block if ipv4-prefix-number (default = 24) or ipv6-prefix-number (default = 56) are specified and/or for a given namespace is a domain is specified.
	Responses by type (responses, referrals, nodata, nxdomains or errors) or all can be limited based on the quantity of responses already provided as specified in the respective "per-second" parameter.
	
	The qps-scale parameter dampens the responses/errors/nxdomains or all per second values to tighten defenses during an attack based on the overall query rate. For example if qps-scale is set to 250 and responses-per-second is 20, then a total query rate of 1000 qps changes the effective responses-per-second to (250/1000)*20 = 5. The optional parameters on responses-per-second control initiation of rate limiting to response or amplification factors to minimum sizes. Size applies to the minimum response size that will trigger this parameter; ratio indicates the policy applies for responses where the response size/request size ratio exceeds this value.
		string	options, view	query	all	rate-limit {
						[responses-per-second number ;]
						[referrals-per-second number ;]
						[nodata-per-second number ;]
						[nxdomains-per-second number ;]
						[errors-per-second number ;]
						[all-per-second number ;]
						[window number ;]
						[log-only (yes | no) ;]
						[qps-scale number ;]
						[ipv4-prefix-length number ;]
						[ipv6-prefix-length number ;]
						[slip number ;]
						[exempt-clients {addr_match_list} ;]
						[max-table-size number ;]
						[min-table-size number ;] };
							rate-limit {
							all-per-second <integer>;
							errors-per-second <integer>;
							exempt-clients { <address_match_element>; ... };
							ipv4-prefix-length <integer>;
							ipv6-prefix-length <integer>;
							log-only <boolean>;
							max-table-size <integer>;
							min-table-size <integer>;
							nodata-per-second <integer>;
							nxdomains-per-second <integer>;
							qps-scale <integer>;
							referrals-per-second <integer>;
							responses-per-second <integer>;
							slip <integer>;
							window <integer>;
							}

recursion	Turn recursion on or off. If set to yes the server will perform recursion to obtain the answer for the client; if no the server will attempt to give an authoritative answer cached information or a referral to another name server.	boolean	options, view	query	all	Enter yes or no. Eg: yes	<boolean>
recursive-clients	Defines the maximum number of simultaneous recursive lookups the server will perform on behalf of clients (default = 1000).	integer	options	query	all	Enter a valid integer. Eg:100	<integer>
request-expire	Configures the server to request the EDNS EXPIRE value from its master server. This value indicates the time remaining until the zone expires if not refreshed. The use case for this option applies when a server, configured as a slave requests zone transfers from another slave. The default is yes.	boolean	options,server,view, zone	transfer. query	all	Enter yes or no. Eg: yes	<boolean>
request-ixfr	Used in options or server statement to configure a slave to request IXFRs of its master or not.	boolean	options,server,view, zone	transfer	all	Enter yes or no. Eg: yes	<boolean>
request-nsid	When set to yes, an empty EDNS0 Name Server Identifier (NSID) option is sent with all queries to authoritative name servers during iterative name resolution. Returned NSID values are logged in the resolver logging category at level info. Default = no.	boolean	options,server,view	query	all	Enter yes or no. Eg: yes	<boolean>
require-server-cookie	Configures the server to require a valid server cookie within a query from a cookie aware client before sending a full response in reply. The BADCOOKIE error is sent if the cookie is absent or invalid.	boolean	options, view	query	all	Enter yes or no. Eg: yes	<boolean>
reserved-sockets	Enables specification of the number of file descriptors supported by the operating system to keep named within this constraint (Default =512).	integer	options	transfer	all	Enter a valid integer. Eg:100	<integer>
resolver-nonbackoff-tries	Defines the number of queries sent prior to applying backoff retries.	integer	options, view	server	all	Enter a valid integer. Eg:100	<integer>
resolver-query-timeout	Enables specification of the number of seconds the server should await a response to a query before failing (SERVFAIL). The default is 10 and the maximum is 30.	integer	options, view	query	all	Enter a valid integer. Eg:100	<integer>

resolver-retry-interval	Defines the time between successive query retries.	integer	options, view	server,query	all	Enter a valid integer. Eg:100	<integer>
response-padding	Enables padding of responses using the EDNS Padding option to maintain consistent packet sizes to improve confidentiality of DNS queries transmitted over encrypted channels. The response will be padded up to blocksize bytes if and only if the query a)contains an EDNS Padding option, b) includes a valid server cookie or uses TCP, c) is not signed using TSIG or SIG(0), and d) is from a client falling within the specified address_match_element.	string	response-padding	query	all	Enter EDNS padding size in bytes. Eg:{ 192.168.0.1; 192.168.0.2;} block-size 256	{ <address_match_element>; ... } block-size <integer>
response-policy	Also known as "DNS Firewall," this option enables specification of modified responses to queries for the specified zone in accordance with the response policy zone initiative where domain registrars may share valid (e.g. non-spammers) domain names to enable resolution while not resolving others modifying or otherwise processing responses for "invalid" domain names as identified via backlist/whitelist queries. Please consult our DNS firewall section for specification details.	string	options, view	security, query, zone, server	all	Eg: { zone "badlist" add-soa yes; }	{ zone <string> [add-soa <boolean>] [log <boolean>] [max-policy-ttl <duration>] [min-update-interval <duration>] [policy (cname | disabled | drop | given | no-op | nodata | nxdomain | passthru | tcp-only <quoted_string>)] [recursive-only <boolean>] [nsip-enable <boolean>] [nsdname-enable <boolean>] [ede <string>]; ... } [add-soa <boolean>] [break-dnssec <boolean>] [max-policy-ttl <duration>] [min-update-interval <duration>] [min-ns-dots <integer>] [nsip-wait-recurse <boolean>] [nsdname-wait-recurse <boolean>] [qname-wait-recurse <boolean>] [recursive-only <boolean>] [nsip-enable <boolean>] [nsdname-enable <boolean>] [dnsrps-enable <boolean>] [dnsrps-options { <unspecified-text> }]

reuseport	The reuseport DNS option allows a DNS server to enable multiple processes to bind to the same port number for incoming traffic, facilitating load balancing across processes and improving the server's ability to handle high volumes of parallel requests.	boolean	options	server	all	Enter yes or no. Eg: yes	<boolean>
root-delegation-only	Enables enforcement of delegation-only processing in root and TLDs except those domains listed within the namelist after the exclude keyword.	string	options	query	all	Enter exclude list of domain names. Eg: exclude { "com"; "net"; "example.org"; }	[exclude { <string>; ... }]
root-key-sentinel	Enables the server to respond to DNS root key sentinal queries to enable the querier to deduce the trusted root zone key configured on the server. These queries are useful for administrators and Internet researchers to verify key configurations, e.g., prior to a key rollover.	boolean	options, view	server	all	Enter yes or no. Eg: yes	<boolean>

rrset-order	It enables the specification of the ordering of resource records when multiple records apply to the query.
	The rrtype parameter refers to a resource record type (e.g., MX) and a given domain name (e.g., manageengine.com).
	Ordering (ordertype) may be:
	fixed -the order in which they are defined in the zone,
	random or **cyclic** -roundrobin. **none**
	Records are returned in the order they were retrieved from the database. This order is indeterminate but remains consistent as long as the database is not modified.
		string	options, view	query	all	Enter the order in which equal RRs (RRsets) are returned.
						Eg:rrset-order {
						type A name "foo.isc.org" order random;
						type AAAA name "foo.isc.org" order cyclic;
						name "bar.isc.org" order fixed;
						name "*.bar.isc.org" order random;
						name "*.baz.isc.org" order cyclic;
						};
							{ [class <string>] [type <string>] [name <quoted_string>] <string> <string>; ... }

secroots-file	Specifies the pathname of the file to which the **rndc secroots** command dumps security roots. (default = named.secroots).	quoted_string	options	dnssec	all	Enter the pathname of the file where the server dumps security roots. Eg: "/var/named/example.secroots"	<quoted_string>

send-cookie	Configures the server to send an EDNS COOKIE with each query to provide identification to the queried server to avoid potential rate limiting treatment.	boolean	options, server, view	query	all	Enter yes or no. Eg: yes	<boolean>
serial-query-rate	Specifies the maximum number of serial number queries per second to be sent to the master (across all zones) (default = 20).	integer	options	transfer	all	Enter a valid integer. Eg:100	<integer>
serial-update-method	Configures the server with the zone serial number format in its SOA record. Setting to increment sets the format to a monotontically increasing integer. The unixtime format indicates the number of seconds since the UNIX epoch unless the serial number is already greater than this value, in whihc case it is incremented by 1. The date method defines the serial number format as YYYYMMDDXX where XX is an incremented value from 00 to 99.	string	options, view, zone	zone	all	Enter the update method to be used for the zone serial number in the SOA record. Valid values: date, increment or unixtime Eg: date	(date | increment | unixtime)
server-id	Specifies the ID that the server should provide in response to a name server identifier (NSID) query or a query for owner ID.SERVER of type TXT in class CHAOS. This information can be helpful in identifying the responding server in an anycast deployment. Defining the server-id_string as none (the default) disables responses to such queries and setting it to hostname returns the configured hostname (per gethostbyname() sockets call).	string	options	server	all	Enter the ID of the server to return in response to a ID.SERVER query. Valid values: none, hostname, custom server id enclosed in quote Eg: "example_server"	(<quoted_string> | none | hostname)
servfail-ttl	Defines the number of seconds to cache a SERVFAIL response due to DNSSEC validation or other server failure. This cache is ignored for queries with the Checking Disabled (CD) bit set to enable querying without validation if desired. The default is 1s, a value of 0 disables such caching and the maximum value is 30s.	duration	options, view	server	all	Enter time in seconds or a valid ISO 8601 duration example: 40	<duration>

session-keyalg	When BIND's pre-defined update-policy local; is configured named automatically creates a TSIG key to sign local dynamic updates. By default the key generation algorithm is HMAC-SHA256 but this option enables overriding this default. Valid values of algorithm are: hmac-sha1 hmac-sha224 hmac-sha256 hmac-sha384 and hmac-md5.	string	options	security	all	Enter a valid algorithm to use for the TSIG session key. Eg: hmac-sha256	<string>
session-keyfile	When BIND's pre-defined update-policy local; is configured named automatically creates a TSIG key to sign local dynamic updates. By default the file is /var/run/named/session.key though an alternative pathname may be defined using this option.	string	options	security	all	Enter pathname of the file where a TSIG session key is written. Eg: "/var/keyfolder/example.key"	(<quoted_string> | none)
session-keyname	When BIND's pre-defined update-policy local; is configured named automatically creates a TSIG key to sign local dynamic updates. By default the keyname is local-ddns though this option may be specified to define a different keyname.	string	options	security	all	Enter a name for the TSIG session key. Eg: example-key	<string>
sig-signing-nodes	Specifies the maximum number of "nodes" (unique RRSet owners) that are examined during a zone re-signing evaluation to determine if re-signature is required or not for each. The default is 100.	integer	options, view, zone	dnssec	all	Enter a valid integer. Eg:100	<integer>
sig-signing-signatures	Specifies the maximum number of RRSets that will be re-signed during an automatic re-signing process. This option bounds the number of signatures performed during a re-sign. The default is 10.	integer	options, view, zone	dnssec	all	Enter a valid integer. Eg:100	<integer>
sig-signing-type	Specifies the RData Type to be used when generating key signing records. The default is 65535.	integer	options, view, zone	dnssec	all	Enter a valid integer. Eg:100	<integer>

sig-validity-interval	Defines the expiration date as the number of days in the future for DNSSEC signatures automatically generated for dynamic updates to a secure zone. The default is 30 days and the maximum value is 10 years. The re-sign parameter defines the remaining time on RRSet signatures within which the server should re-sign the RRSet. If days is < 7 then re-sign is defined in units of hours; otherwise it is in days. If re-sign is not specified days/4 will be used as the assumed re-sign value. This option can be overidden for DNSKEY records via the dnskey-sig-validity option.	string	view, options, zone	obsolete	all	Enter number of days. Eg: 60	<integer> [<integer>]
sortlist	Enables specification of the order of query responses based on source of query respond with preferred list of responses. Here are the details on the syntax and interpretation of the sortlist option.	address_ list	view, options	query	all	Enter a list of valid ip_addresses/netprefixes/acl_ names/server_keys enclosed within curly braces seperated by semicolons Eg: { 192.168.1.1; 192.168.1.2; }	{ <address_match_element>; ... }
stacksize	Defines the maximum size of stack memory the server may use. The default is default which is the amount of stack memory allocated by the operating system by default.	string	options	query	all	Enter size in bytes followed by an additional suffix 'k', 'm or 'g'. Eg:3k	<sizeval>
stale-answer-client-timeout	Defines the duration the server will wait before attempting to answer the query with a stale resource record from cache; if an answer is resolved in the meantime, the server will answer and refresh its cache wtih the resolved value. The minimum value is 0 (immediately return stale records) and the maximum is the value of resolver-query-timeout minus one second. The default is off (which is equivalent to disabled) and this option is ignored if stale-answer-enable is set to no.	string	options, view	server,query	all	Enter amount of time in milliseconds.To dusable this option enter disabled or off. Eg: 100	(disabled | off | <integer>)
stale-answer-enable	Enables the server to respond with cached resource records whose TTL has expired when an authoritative server cannot be reached. The default is no. When set to yes, stale-cache-enable should also be set to yes.	boolean	options, view	server,query	all	Enter yes or no. Eg: yes	<boolean>

stale-answer-ttl	Defines the TTL to be transmitted on stale resource records (records retained in cache whose TTL has expired). The default is 1s.	duration	options, view	query	all	Enter time in seconds or a valid ISO 8601 duration example: 40	<duration>
stale-cache-enable	Enables the server to cache rather than expire state resource records, i.e., those whose TTL has expired when an authoritative server cannot be reached. The default is yes.	boolean	options, view	server,query	all	Enter yes or no. Eg: yes	<boolean>
stale-refresh-time	If authoritative name servers for a queried zone are not answering queries, the recursive server will reply to its clients' queries with stale resource records without attempting to query the authoritative servers for the specified duration. The default is 30s and a value of 0 disables this feature, enabling full resolution attempts for every query regardless of authoritative servers status.	duration	options, view	server,query	all	Enter time in seconds or a valid ISO 8601 duration example: 40	<duration>
startup-notify-rate	Defines the rate of Notify requests sent when the name server is first starting up or when zones have been newly added. The default rate is 20 per second.	integer	options	zone,transfer	all	Enter a valid integer. Eg:100	<integer>
statistics-file	Specifies this pathname of the file to which the server appends statistics when the rndc stats command is executed. The default is named.stats.	quoted_string	options	logging, server	all	Enter pathname of files enclosed in quotes. Eg: "/var/named/named.stats"	<quoted_string>
synth-from-dnssec	Setting to yes (default) could improve DNSSEC resolution performance by enabling synthesized validated responses based on cached NSEC (NSEC3 support not yet implemented) records and other RRsets that have been previously validated.	boolean	options, view	dnssec	all	Enter yes or no. Eg: yes	<boolean>
tcp-advertised-timeout	Sets the timeout value the server will send in reponses containing the EDNS TCP Keepalive option specified in units of 100ms. Valid values range from 0 (close TCP connections immediately) to 65535, and the default is 300 (30s).	integer	options	query	all	Enter a valid integer. Eg:100	<integer>
tcp-clients	Limits number of concurrent TCP connections (default = 100).	integer	options	server	all	Enter a valid integer. Eg:100	<integer>

tcp-idle-timeout	Defines the length of time the server waits on an idle TCP connection before closing it when the client is not using the EDNS TCP Keepalive option. This timeout is specified in units of 100ms. Valid values range from 1 (0.1s) to 1200 (2m), and the default is 300 (30s).	integer	options	query	all	Enter a valid integer. Eg:100	<integer>
tcp-initial-timeout	Defines the length of time the server waits on a new TCP connection for the first message from the client before closing the connection. This timeout is specified in units of 100ms. Valid values range from 1 (0.1s) to 65535, and the default is 300 (30s).	integer	options	query, server	all	Enter a valid integer. Eg:100	<integer>
tcp-keepalive-timeout	Specifies the length of time the server waits on an idle TCP connection before closing it when the client is using the EDNS TCP Keepalive option. This timeout is specified in units of 100ms. Valid values range from 1 (0.1s) to 1200 (2m), and the default is 300 (30s). If you have plans to serve DNS-over-HTTPS, you might want to set tcp-initial-timeout, tcp-keepalive-timeout, and tcp-idle-timeout to the minimum values that work for you. Setting them to tcp-initial-timeout 100, tcp-keepalive-timeout 100 and tcp-idle-timeout 100 (ten seconds) is a good setting to try.	integer	options	query	all	Enter a valid integer. Eg:100	<integer>
tcp-listen-queue	Specifies the queue depth for listening for TCP connections (default and minimum = 3).	integer	options	server	all	Enter a valid integer. Eg:100	<integer>
tcp-receive-buffer	Specifies the size of the buffer that the DNS server uses to receive data over TCP connections. The buffer is a block of memory allocated for storing incoming data until it can be processed. A receive buffer of the correct size can improve the server's ability to handle incoming data efficiently, especially when multiple or large DNS queries are received simultaneously over TCP. If the buffer is too small, it may result in slow data processing and the need for retransmissions if packets are dropped. If too large, it may waste system resources without any added performance benefit.	integer	options	server	all	Enter a valid integer. Eg:100	<integer>

tcp-send-buffer	Determines the size of the buffer for sending data to clients or other DNS servers via TCP. The send buffer holds outgoing data before it is transmitted over the network. Configuring an appropriate send buffer size is crucial for the efficient handling of outbound traffic, particularly when the server is sending large amounts of data, such as DNSSEC responses or during DNS zone transfers. A send buffer that is too small might lead to underutilization of the available network bandwidth, causing delays in data transmission. Conversely, a buffer that is too large could unnecessarily consume memory resources. Both tcp-receive-buffer and tcp-send-buffer need to be set in consideration of the operating system limits, network conditions, and expected traffic loads to balance resource usage with performance and reliability. Proper tuning of these parameters can lead to a more responsive and stable DNS service, especially under high load or in networks with high latency.	integer	options	server	all	Enter a valid integer. Eg:100	<integer>
tkey-dhkey	This option specified the Diffie-Hellman key to use to generate shared keys with clients when using the Diffie-Hellman mode of TKEY. In most cases this should be the server's hostname.	string	options	security	all	Enter key-id and key-tag seperated by space Eg: "key-id" "key-tag"	<quoted_string> <quoted_string>
tkey-domain	This option specifies the domainname that should be appended to the names of all shared keys generated during a TKEY exchange. In most cases the domainname should be the server's domain name.	quoted_s tring	options	security	all	Enter tkey domain in quoted string Eg: "example.com"	<quoted_string>
tkey-gssapi-credential	This option configures the credential to be used to authenticate keys for use with the GSS-TSIG protocol e.g. when performing secure updates to Microsoft Windows DNS. Currently a Kerberos principal is supported	quoted_s tring	options	security	all	Eg: "DNS/example.com";	<quoted_string>
tkey-gssapi-keytab	Defines the pathname to the key file used to authenticate Kerberos 5 credentials. If not set the typical system key file is /etc/krb5.keytab.	quoted_s tring	options	security	all	Eg: "/etc/krb5.keytab"	<quoted_string>

tls-port	Designates the specific port number for secure DNS queries over TLS. The standard port for DNS over TLS is 853, as defined by IANA (Internet Assigned Numbers Authority). By setting the tls port option, a DNS server is instructed to establish secure connections with clients that request DNS resolution through TLS, ensuring that DNS queries and responses are encrypted and secure from eavesdropping as well as man-in-the-middle attacks. It's an essential part of implementing DoT on a DNS server, which is increasingly important for enhancing privacy and security in DNS transactions.	integer	options	query, server	all	Enter a valid integer. Eg:100	<integer>
transfer-format	Specifies on a master server which format to employ for zone transfers: one-answer means one resource record per message while many-answers (the default) means multiple records as many as will fit within the message size are placed within each transfer message.	string	options, server, view	transfer	all	Valid values: many-answers, one-answer Eg: many-answers	(many-answers | one-answer)
transfer-message-size	Intended primarily for testing, this option defines a soft upper bound on uncompressed zone transfer messages over TCP. If the message size exceeds this bound, multiple messages will be sent unless the Rdata of a single resource record exceeds the bound it will be sent regardless. Valid values range from 512 to 65535 and the default is 20480.	integer	options	transfer	all	Enter a valid integer. Eg:100	<integer>
transfer-source	Defines the server's network interface (IPv4 address and optionally port number) on which incoming zone transfers will be bound. This option also specifies the source IP address and optionally source UDP port for SOA query messages and forwarded dynamic updates.	string	options, server, view, zone	transfer	all	Enter which local IPv4 address is bounded to TCP connections used to fetch zones transferred inbound by the server. Eg: 192.168.0.2	(<ipv4_address> | *)

transfer-source-v6	Defines the server's network interface (IPv6 address and optionally port number) on which inbound zone transfers will be bound. This option also specifies the source IPv6 address and optionally source UDP port for SOA query messages and forwarded dynamic updates.	string	options, server, view, zone	transfer	all	Enter which local IPv6 address is bounded to TCP connections used to fetch zones transferred inbound by the server. Eg: 2001:db8::1	(<ipv6_address> | *)
transfers-in	Specifies a limit to the total number of concurrently running inbound zone transfers (default = 10)	integer	options	transfer	all	Enter a valid integer. Eg:100	<integer>
transfers-out	Specifies a limit to the total number of concurrently running outbound zone transfers (default = 10)	integer	options	transfer	all	Enter a valid integer. Eg:100	<integer>
transfers-per-ns	Specifies a limit on the number of concurrently running inbound zone transfers from any given server (default = 2)	integer	options	transfer	all	Enter a valid integer. Eg:100	<integer>
trust-anchor-telemetry	The trust-anchor-telemetry DNS option enables a DNS server to automatically query for DNSSEC trust anchors to monitor their status, ensuring they are current and valid for the authentication of DNS responses. This feature helps maintain DNS security by facilitating the automatic update of trust anchors during key rollover events, in compliance with DNSSEC standards.	boolean	options, view	dnssec	all	Enter yes or no. Eg: yes	<boolean>
try-tcp-refresh	If a zone refresh query via UDP fails this option when set to yes configures the server to reattempt using TCP. The default is yes.	boolean	options, view, zone	transfer	all	Enter yes or no. Eg: yes	<boolean>

udp-receive-buffer	This option sets the size of the buffer that the DNS server uses to receive UDP (User Datagram Protocol) packets. A larger receive buffer can improve performance by allowing the server to handle more incoming requests simultaneously, especially during traffic spikes. If the buffer is too small, the server may drop incoming requests because it doesn't have enough space to store them, leading to increased query times or failed requests. Configuring the right size for the UDP receive buffer is a balance: too small, and you risk losing packets; too large, and you may waste system resources or run into other limits set by the operating system.	integer	options	server	all	Enter a valid integer. Eg:100	<integer>
udp-send-buffer	This option sets the size of the buffer used for sending UDP packets from the DNS server. Just as with the receive buffer, a send buffer that's too small can lead to packet loss and poor performance, especially when responding to a large volume of requests.Conversely, a buffer that's too large could be inefficient, potentially holding onto data longer than necessary and using more memory than needed. The send buffer size can affect how quickly the server can respond to requests and handle outgoing traffic loads.	integer	options	server	all	Enter a valid integer. Eg:100	<integer>
update-check-ksk	Configures the server to use KSK(s) (KSK flag on the corresponding DNSKEY resource record is set) to sign the DNSKEY RRset only (if set to yes) or to ignore the KSK flag and use all zone keys to sign the zone (if set to no). The default of yes effectively requires the use of separate zone KSKs and ZSKs while a setting of no enables use of one key per zone.	boolean	options, view, zone	obsolete	all	Enter yes or no. Eg: yes	<boolean>

update-quota	Specifies the maximum number of concurrent DNS UPDATE messages that can be processed by the server. This is the maximum number of simultaneous DNS UPDATE messages that the server will accept for updating local authoritiative zones or forwarding to a primary server. The default is 100.	integer	options	server	all	Enter a valid integer. Eg:100	<integer>
use-alt-transfer-source	Controls the use of alternative transfer source options for v4 and v6 (alt-transfer-source and alt-transfer-source-v6 respectively).	boolean	options	server	all	Enter yes or no. Eg: yes	<boolean>
use-v4-udp-ports	Enables specification of a range or pool of port numbers from which a randomly selected value will be used to set the source port for outbound IPv4 queries. Values set in the avoid-v4-udp-ports option will be excluded from this port list for port number generation. The default range is 1024 65535. Note: Make sure your port range coincides with those permitted by the operating system on which named is running for named; otherwise queries using these port numbers will fail.	string	options	deprecated	all	Enter a list of ports that are valid sources for UDP/IPv4 messages. Valid values: list of ports or port ranges Eg: { 7080; range 480 500; }	{ <portrange>; … }
use-v6-udp-ports	Enables specification of a range or pool of port numbers from which a randomly selected value will be used to set the source port for outbound IPv6 queries. Values set in the avoid-v6-udp-ports option will be excluded from this port list for port number generation. The default range is 1024 65535. Note: Make sure your port range coincides with those permitted by the operating system on which named is running for named; otherwise queries using these port numbers will fail.	string	options	deprecated	all	Enter a list of ports that are valid sources for UDP/IPv4 messages. Valid values: list of ports or port ranges Eg: { 7080; range 480 500; }	{ <portrange>; … }
v6-bias	Indicates the number of milliseconds of preference to give to IPv6 name servers. When determining the next name server to try, this indicates by how many milliseconds to prefer IPv6 name servers. The default is 50 milliseconds.	integer	options, view	query,server	all	Enter a valid integer. Eg:100	<integer>

validate-except	This option disables DNSSEC validation for specified domains and respective subdomains. While negative trust anchors enable this functionality on a temporary basis, this option enables permanent disabling of validation for these domains, such as unsigned local-use domains for example.	string	options, view	dnssec	all	Enter a list of domain names at and beneath which DNSSEC validation should not be performed. Eg:{ "example1.com"; "example2.com"; }	{ <string>; ... }
version	This option specifies the string the server should provide in response to give to a TXT query of class CHAOS for name version.bind. Setting version_string to "none" disables responding to these queries.	string	options	server	all	Enter a version number enclosed in quotes that will be returned on a version.bind query. If not required enter none. Eg:"12.0.4"	(<quoted_string> | none)
zero-no-soa-ttl	Instructs the server to set the TTL to zero when returning an authoritative negative response to an SOA query (default = yes).	boolean	view, zone, options	server,query, zone	all	Enter yes or no. Eg: yes	<boolean>
zero-no-soa-ttl-cache	Instructs the server when caching a negative response to an SOA query to set the TTL to zero (default = no).	boolean	options, view	server,query,zone	all	Enter yes or no. Eg: yes	<boolean>
zone-statistics	Instructs the server to collect statistical data on all zones (or per zone control/override in zone statement).	string	view,zone, options	zone, logging	all	Either one of the three values : full/terse/none or a boolean value(yes or no) Eg: full	(full | terse | none | <boolean>)

Option Code	Option Name	Description	Data Type	Supported	Is Predefined?	Grammar	Example
1	subnet-mask	Specifies the client's subnet mask. If not provided, the DHCP server defaults to the subnet mask in the subnet declaration, unless overridden by a subnet-mask option in scope for the assigned address.	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
2	time-offset	Specifies the offset of the client's subnet in seconds from Coordinated Universal Time (UTC).	int32	TRUE	TRUE	int32	Range : -2,147,483,648 to +2,147,483,647 Example : 12345
3	routers	Specifies a list of IP addresses for routers on the client's subnet. Routers should be listed in order of preference.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
4	time-servers	Specifies a list of Time Servers available to the client. Time Servers are listed in order of preference.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
5	ien116-name-servers	Specifies a list of Name Servers available to the client. Name Servers are listed in order of preference.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
6	domain-name-servers	Specifies a list of Domain Name System (RFC 1035) name servers available to the client. Servers are listed in order of preference.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
7	log-servers	Specifies a list of UDP log servers available to the client. Servers are listed in order of preference.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
8	cookie-servers	The cookie server option specifies a list of RFC 865 cookie servers available to the client. Servers should be listed in order of preference.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
9	lpr-servers	Specifies a list of Line Printer Servers available to the client. Servers are listed in order of preference.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
10	impress-servers	Specifies a list of Imagen Impress Servers available to the client. Servers are listed in order of preference.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
11	resource-location-servers	Specifies a list of Resource Location Servers available to the client. Servers are listed in order of preference.	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
12	host-name	Configures the host name string that can be assigned to the DHCP clients.	quoted_string	TRUE	TRUE	string	Example : host_name
13	boot-size	Specifies the size of the boot file in 512 byte blocks	uint16	TRUE	TRUE	uint16	Range : 0-65535 Example : 1024
14	merit-dump	Specifies the file path name to which the client should dump its core image in the event of a client crash	quoted_string	TRUE	TRUE	string	Example : string_value
15	domain-name	Specifies the domain name the client should use when resolving host names using the Domain Name System.	quoted_string	TRUE	TRUE	text	Example : example.com

16	swap-server	Specifies the IP address of the client Swap Server.	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
17	root-path	Specifies the path name (entered as an ASCII character string) that contains the client root disk.	quoted_string	TRUE	TRUE	text	Example : /home/path/dhcpd/
18	extensions-path	Specifies a file, retrievable through TFTP, that contains information that can be interpreted in the same way as the vendor-extension field within the BOOTP response, with the following exceptions: 1) the length of the file is unconstrained 2) all references to instances of this option in the file are ignored	quoted_string	TRUE	TRUE	text	Example : path/dhcpd/
19	ip-forwarding	Enable/Disable IP packet forwarding	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
20	non-local-source-routing	Enable/Disable IP packet forwarding for packets specifying non-local source routes.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
21	policy-filter	Specifies Policy Filters for non-local source routing. The filters consist of a list of IP addresses and masks that specify destination/mask pairs with which to filter incoming source routes. Any source-routed datagram whose next-hop address does not match one of the filters should be discarded by the client. In other words, it specifies acceptable non-local next hops to which IP packets may be forwarded for packets specifying non-local source routes.	ipv4address	TRUE	TRUE	ip-address ip-address [, ip-address ip-address...]	Type : ipv4 Example : 1.1.1.1 2.2.2.2 ,3.3.3.3 4.4.4.4,..,etc
22	max-dgram-reassembly	The maximum size datagram the client should be ready to reassemble specified as a 16-bit unsigned integer	uint16	TRUE	TRUE	uint16	Range : 0-65535 Example : 1024
23	default-ip-ttl	Default IP time to live value for use in outgoing packet IP header TTL field	uint8	TRUE	TRUE	uint8	Range : 0-255 Example : 1
24	path-mtu-aging-timeout	The timeout in seconds when performing path maximum transmission unit (MTU) discovery in accordance with RFC 1191; MTU discovery helps minimize packet fragmentation along the path	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
25	path-mtu-plateau-table	Specifies a table listing MTU sizes to use when performing Path MTU Discovery. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest. The minimum MTU value cannot be smaller than 68.	uint16	TRUE	TRUE	uint16 [, uint6...]	Range : 0-65535 Example : 1024,1024,..,etc
26	interface-mtu	Specifies the value of the Maximum Transmission Unit (MTU) to use on this interface. The minimum legal value for the MTU is 68.	uint16	TRUE	TRUE	uint16	Range : 0-65535 Example : 1024

27	all-subnets-local	Indicates whether all subnets within the client's network use the same maximum transmission unit (MTU) as the local subnet to which the client is connected.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
28	broadcast-address	Specifies the Broadcast Address in use on the client subnet.	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
29	perform-mask-discovery	Specifies whether the client should perform subnet mask discovery or not	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
30	mask-supplier	Specifies whether the client should respond to other clients performing mask discovery	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
31	router-discovery	Specifies whether the client should perform router discovery or not	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
32	router-solicitation-address	Specifies the address to which the client should transmit router solicitation requests.	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
33	static-routes	Specifies a list of Static Routes that the client should install in its routing cache. If multiple routes to the same destination are specified, they are listed in descending order of priority. The routes consist of a list of IP address pairs. The first address is the destination address, and the second address is the router for the destination. Note that the default route (0.0.0.0) is an illegal destination for a static route.	ipv4address	TRUE	TRUE	ip-address ip-address [, ip-address ip-address...]	Type : ipv4 Example : 1.1.1.1 2.2.2.2 ,3.3.3.3 4.4.4.4,..,etc
34	trailer-encapsulation	Specifies whether the client should attempt to negotiate the use of layer 2 frame trailers (like headers but at the end of the frame payload) in ARP messages	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
35	arp-cache-timeout	This option specifies the timeout in seconds for ARP cache entries.	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
36	ethernet-encapsulation	Specifies whether the client should use Ethernet II or IEEE 802.3 on an Ethernet interface	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
37	default-tcp-ttl	Default TCP time to live value	uint8	TRUE	TRUE	**uint8**	Range : 0-255 Example : 1
38	tcp-keepalive-interval	TCP keepalive interval in seconds	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
39	tcp-keepalive-garbage	Specifies whether the client should send an octet of **garbage** within TCP keepalive messages for compatibility with older implementations	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
40	nis-domain	Specifies the NIS domain (entered as an ASCII character string) for the client.	quoted_string	TRUE	TRUE	text	Example : domain.com
41	nis-servers	Specifies a list of IP addresses for NIS servers available to the client. Servers are listed in order of preference.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc

42	ntp-servers	Specifies a list of IP addresses for NTP servers available to the client. Servers are listed in order of preference.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
43	vendor-encapsulated-options	Specifies vendor-specific information. This allows clients and servers to exchange vendor-specific information. The vendor is specified in the Vendor Class Identifier option (option 60).	quoted_string	TRUE	TRUE	string	Example : string_value
44	netbios-name-servers	Specifies a list of NetBIOS Name Servers (NBNS) aka (WINS servers) (RFC 1001 and RFC 1002). NBNS servers are listed in order of preference.	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
45	netbios-dd-server	Specifies a list of NetBIOS Datagram Distribution Servers (NBDD) addresses (RFC 1001 and RFC 1002). NBDD server addresses are listed in order of preference.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
46	netbios-node-type	Specifies the client as a specific NETBIOS Node Type	uint8	TRUE	TRUE	uint8	Range : 0-255 Example : 1
47	netbios-scope	Specifies the NetBIOS over TCP/IP scope parameter (RFC 1001 and RFC 1002) for the client. While the DHCP server itself may not have a specific option for NetBIOS scope, it can be configured to provide NetBIOS settings to DHCP clients through DHCP options. DHCP Option 44 (WINS/NBT Servers) and DHCP Option 46 (WINS/NBT Node Type) are examples of DHCP options that can be used to provide NetBIOS-related information to clients.	quoted_string	TRUE	TRUE	string	Example : string_value
48	font-servers	Specifies a list of IP addresses of X Window System Font servers available to the client. X Window System Font servers are listed in order of preference.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc

49	x-display-manager	X Window System display manager addresses, such as those associated with XDMCP (X Display Manager Control Protocol), are typically configured separately from DHCP. The XDMCP addresses are commonly set in the configuration of the X display manager on the client or server. If you are looking to configure XDMCP addresses for X Window System display managers, you would typically do so through the configuration files specific to the X display manager software you are using (e.g., LightDM, GDM, XDM, etc.). The configuration files might include settings for specifying XDMCP servers, display addresses, and related parameters.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
50	dhcp-requested-address	IP address requested by the client (within a Discover message).When a DHCP client is in the initial stages of obtaining an IP address, it sends a DHCP Discover message to the network, indicating that it is seeking an available IP address. In this Discover message, the client may include DHCP Option 50 to express its preference or specific request for a particular IP address.	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
51	dhcp-lease-time	IP address lease time requested by the client (within a Discover or Request message).This option is used in a client request (DHCPDISCOVER or DHCPREQUEST) to allow the client to request a lease time for the IP address. In a server reply (DHCPOFFER), a DHCP server uses this option to specify the lease time it is willing to offer.	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
52	dhcp-option-overload	Indicates that the **sname** and/or **file** DHCP header fields contain additional DHCP option information if options to return to the client exceed the normal option space in the message	uint8	TRUE	TRUE	uint8	Range : 0-255 Example : 1

53	dhcp-message-type	DHCP message type option is used to convey the type of DHCP message being sent between the DHCP client and server. This option is essential for both the initiation and acknowledgment of DHCP messages, helping devices understand the purpose of the received DHCP packet. The DHCP message type option is specified by Option 53 in the DHCP protocol. The values associated with this option indicate different DHCP message types. Here are some common DHCP message types and their corresponding values: DHCPDISCOVER (Client to Server) - Value: 1: The DHCPDISCOVER message is sent by a DHCP client to discover available DHCP servers on the network. DHCPOFFER (Server to Client) - Value: 2: The DHCPOFFER message is sent by a DHCP server in response to a DHCPDISCOVER message. It offers configuration parameters, including IP address lease information, to the client. DHCPREQUEST (Client to Server) - Value: 3: The DHCPREQUEST message is sent by a DHCP client to formally request the offered configuration parameters from a specific DHCP server. DHCPACK (Server to Client) - Value: 5: The DHCPACK message is sent by the DHCP server to acknowledge the DHCPREQUEST from the client and confirm the lease of the offered configuration parameters. DHCPNAK (Server to Client) - Value: 6: The DHCPNAK message is sent by the DHCP server to indicate that the requested configuration parameters in the DHCPREQUEST are not available, and the client needs to restart the configuration process. DHCPRELEASE (Client to Server) - Value: 7: The DHCPRELEASE message is sent by a DHCP client to inform the server that it is releasing its IP address lease and will no longer use it. These values are included in the DHCP message type option (Option 53) as part of the DHCP packet. The DHCP client and server use these values to understand the purpose of the received DHCP message and respond accordingly during the dynamic IP address allocation process.	uint8	TRUE	TRUE	uint8	Range : 0-255 Example : 1

54	dhcp-server-identifier	DHCP server identification provided in the Offer (and Request and optionally ACK, NAK) to identify the server, e.g. to distinguish among multiple offers	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
55	dhcp-parameter-request-list	List of DHCP option code numbers for parameters requested by the client	uint8	TRUE	TRUE	uint8 [, uint8...]	Range : 0-255 Example : 123,153,..,etc
56	dhcp-message	Text containing an error message; can be used by the server in a Nak message to the client or by the client in a Decline message; e.g. this text could be included in logging details	quoted_string	TRUE	TRUE	text	Example : message
57	dhcp-max-message-size	The maximum DHCP message length the client is willing to accept	uint16	TRUE	TRUE	uint16	Range : 0-65535 Example : 1024
58	dhcp-renewal-time	Interval from address assignment time to the time the client enters the Renewing state	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
59	dhcp-rebinding-time	Interval from address assignment time to the time the client enters the Rebinding state.	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
60	vendor-class-identifier		quoted_string	TRUE	TRUE	string	Example : string_value
61	dhcp-client-identifier	Client identifier used by DHCP clients to uniquely identify themselves to the DHCP server during the DHCP negotiation process. The client identifier is typically associated with a specific hardware address or other unique identifier for the client device. Example: option dhcp-client-identifier 01:02:03:04:05:06; In this example: 01 indicates that the subsequent bytes represent the client's hardware address. 02:03:04:05:06 is the actual hardware address (MAC address) of the DHCP client.	quoted_string	TRUE	TRUE	string	Example : value
62	nwip-domain	Netware/IP Domain Name	quoted_string	TRUE	TRUE	string	Example : domain.com
63	nwip-suboptions	Netware/IP sub Options	quoted_string	TRUE	TRUE	string	Example : string_data
64	nisplus-domain	Network Information Services+ (NIS+) client domain name	quoted_string	TRUE	TRUE	text	Example : domain.com
65	nisplus-servers	Network Information Services+ (NIS+) server addresses	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc

66	tftp-server-name	TFTP server name can be used when the **sname** DHCP header field has been overloaded with other options. When a DHCP client receives Option 66 as part of its DHCP configuration, it uses the provided information to locate and connect to a TFTP server. This is particularly useful in scenarios where network devices need to retrieve configuration files or firmware updates during the initialization process.	quoted_string	TRUE	TRUE	text	Example : text_server_name
67	bootfile-name	Boot file name; can be used when the file DHCP header field has been overloaded with other options	quoted_string	TRUE	TRUE		Example : string_filename
68	mobile-ip-home-agent	Mobile IP home agent addresses	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
69	smtp-server	Simple Mail Transfer Protocol (SMTP) server addresses for outgoing e-mail.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
70	pop-server	Post Office Protocol v3 (POP3) server addresses for incoming e-mail retrieval.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
71	nntp-server	Network News Transport Protocol (NNTP) server addresses.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
72	www-server	The WWW server option specifies a list of WWW servers available to the client. Servers should be listed in order of preference.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
73	finger-server	Finger server addresses; finger servers enable retrieval of host user information regarding login name, login duration, and more	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
74	irc-server	Internet Relay Chat (IRC) server addresses	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
75	streettalk-server	StreetTalk server addresses; StreetTalk was a Banyan Vines user and resource directory	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
76	streettalk-directory-assistance-server	StreetTalk Directory Assistance (STDA) server addresses; StreetTalk was a Banyan Vines user and resource directory	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
77	user-class	User Class Identifier	quoted_string	TRUE	TRUE	string	Example : string_value
78	slp-directory-agent	Service Location Protocol (SLP) Directory Agent IP address(es)	string	TRUE	TRUE	boolean ip-address [, ip-address...]	values : true or false or on or off , Type : ipv4 Example : true 1.1.1.1,2.2.2.2,..,etc
79	slp-service-scope	Service Location Protocol (SLP) service scope the SLP agent is configured to use.	string	TRUE	TRUE	boolean text	values : true or false or on or off Example : true "text_value"
80	rapid-commit	[Not supported] Rapid Commit - requests a two-packet DHCP transaction instead of the normal four packet DORA process for mobility or overhead-constrained applications		FALSE			

81	fqdn	Conveys the client's fully qualified domain name to the DHCP server. This option helps in associating a specific FQDN with a DHCP client, providing a human-readable identifier for the client. The DHCP option 81 is structured as follows: Code: 81 Length: Length of the FQDN field Flags: Flags indicating the format of the FQDN RCODE1: Return code for the server's use RCODE2: Return code for the client's use Fully Qualified Domain Name: The actual FQDN of the client Here is an example of a DHCP option 81: rust Copy code Option: (t=81,l=15) domain-search "example.com". This option is useful in scenarios where clients need to be identified and associated with specific FQDNs within the network. The server can use this information for various purposes, including updating DNS records or maintaining a more human-readable record of leased addresses.	quoted_string	TRUE	TRUE		Type : fqdn
82	relay-agent-information	Relay Agent Information - additional client information supplied by the intervening relay agent	string	TRUE	TRUE		Example : "string_value"
83	iSNS [Internet Storage Name Service]	Internet Storage Name Service (ISNS) server addresses and iSNS application information					
84	Unassigned	These options are supported if the option value type is IP, ASCII, or HEX					
85	nds-servers	Novell Directory Services (NDS) Server IP addresses to contact for NDS client authentication and access the NDS directory repository	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
86	nds-tree-name	Novell Directory Services (NDS) tree name of the NDS repository the client should contact	quoted_string	TRUE	TRUE		Example : string_name
87	nds-context	Novell Directory Services (NDS) initial context within the NDS repository the NDS client should use	quoted_string	TRUE	TRUE		Example : string_value

88	bcms-controller-names	Broadcast and Multicast Server domain name (FQDN) list, used to construct follow-up SRV query(ies) (BCMCS is used in 3G wireless networks to enable mobiles to receive broadcast and multicast services)	quoted_string	TRUE	TRUE	domain-list	Example : example.com,domainexample.com,..,etc
89	bcms-controller-address [BCMCS Controller IPv4 Address Option]	Broadcast and Multicast Server (BCMCS) Controller IP address(es) (BCMCS is used in 3G wireless networks to enable mobiles to receive broadcast and multicast services)	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
90	authentication	Authentication option used to communicate authentication information between the client and server in accordance with the DHCP authentication protocol	quoted_string	FALSE	TRUE		
91	client-last-transaction-time	Seconds since the last DHCP transaction with the client on this lease as queried in a DHCP Lease Query message	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
92	associated-ip	List of IP addresses associated with the client as queried in a DHCP Lease Query message	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
93	pxe-system-type	PXE client system architecture type(s) each encoded as 16-bit code, e.g. Intel x86PC, DEC Alpha, EFI x86-64, etc.	uint16	TRUE	TRUE	uint16 [, uint6...]	Range : 0-65535 Example : 1024,1024,..,etc
94	pxe-interface-id	PXE client network interface identifier with individual octets encoded for interface type, interface major version number, and interface minor version number	uint8	TRUE	TRUE	uint8 uint8 uint8	Range : 0-255 Example : 1 2 3
95	LDAP	Lightweight Directory Access Protocol servers; this option is used by Apple Computer though no governing RFC has been published		FALSE	TRUE		
96	Unassigned						
97	pxe-client-id	PXE client machine identifier with encoded type and identifier value	string	TRUE	TRUE	uint8 string	Range : 0-255 and string Example : 1 "string_value"
98	uap-servers	List of locations (URLs) for services capable of processing authentication requests encapsulated using Open Group's User Authentication Protocol (UAP)	quoted_string	TRUE	TRUE	text	Example : text_value
99	geoconf-civic	Location of the server, network element closest to the client or the client itself as provided by the server encoded in country-specific civic (e.g. postal) format	quoted_string	TRUE	TRUE	string	Example : string_value
100	pcode	Time Zone encoded as IEEE 1003.1 TZ (POSIX)	quoted_string	TRUE	TRUE	text	Example : text_value
101	tcode	Reference to a local (on the client) TZ database for lookup of time zone	quoted_string	TRUE	TRUE	text	Example : text_value
102-111	Unassigned	These options are supported if the option value type is IP, ASCII, or HEX					

108	v6-only-preferred	specifies a method to use a DHCP option (delivered over IPv4) to disable the IPv4 protocol on a host (resulting in it being IPv6-only). DHCP Option 108 contains a 32-bit unsigned integer that represents the number of seconds the client should disable DHCPv4. Following are the timers that are defined in the RFC: V6ONLY_WAIT (default = 1800 seconds, 30 min) MIN_V6ONLY_WAIT (default = 300 seconds, 5 min), you can select assign the timer more value like 3600 seconds or more depending on your requirements.	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
109	softwire-address	IPv6 address assigned via DHCPv4 over DHCPv6					
110-111	Unassigned						
112	netinfo-server-address	NetInfo Parent Server Address; this option is used by Apple Computer though no governing RFC has been published; NetInfo is a distributed database user and resource information for Apple devices.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,...,etc
113	netinfo-server-tag	NetInfo Parent Server Tag; this option is used by Apple Computer though no governing RFC has been published. NetInfo is distributed database user and resource information for Apple devices.	quoted_string	TRUE	TRUE	text	Example : text_value
114	dhcp-captive-portal	Informs the client that they are behind a captive portal with a URI to an authentication function	quoted_string	TRUE	TRUE	string	http://www.YourDomain.com.
115	Unassigned	These options are supported if the option value type is IP, ASCII, or HEX					
116	auto-config	Instructs the client to auto-configure a link local address (69.254.0.0/16) or not. This can be used by the DHCP server to inform the client that it has no IP addresses to assign and that the client may or may not auto-configure	uint8	TRUE	TRUE	uint8	Range : 0-255 Example : 1
117	name-service-search	Lists one or more name services in priority order that the client should use for name resolution: DNS, NIS, NIS+, or WINS	uint16	TRUE	TRUE	uint16 [, uint6...]	Range : 0-65535 Example : 1024,1024,...,etc
118	subnet-selection	Identifies an IP subnet (address) from which to allocate an IP address to this client - overrides the GIAddr setting or DHCP server interface on which a broadcast Discover was received	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
119	domain-search	List of one or more domains for configuration of the client's resolver. If the application requests a resolution for a non-FQDN hostname, these domain(s) will successively be appended to the hostname prior to querying	quoted_string	TRUE	TRUE	domain-list	Example : example.com,sales.example.com,eng.example.com,...,etc

123	Location Configuration information	Provides the client its Location Configuration Information (LCI), including latitude, longitude, altitude and resolution of each coordinate					
124	vivco [Vendor-Identifying Vendor Class]	Enables specification of multiple vendor classes, each identified by IANA-assigned Enterprise Number (EN); this is useful to identify the hardware vendor, software vendor, application vendor, etc. supporting the device	quoted_string	TRUE	TRUE	string	Example : string_value
125	vivso [Vendor-Identifying Vendor-Specific Information]	Set of DHCP options grouped by vendor as identified by IANA-assigned Enterprise Number (EN);	quoted_string	TRUE	TRUE	string	Example : string_value
126-127	Unassigned	These options are supported if the option value type is IP, ASCII, or HEX					
128-135	PXE - Udefined [Vendor Specific]	These options are supported if the option value type is IP, ASCII, or HEX					
136	pana-agent	Identifies one or more IPv4 addresses of PANA (Protocol for carrying Authentication for Network Access) Authentication Agents for use by the client for authentication and authorization for network access service.	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
137	v4-lost	Location to Service Translation (LOST) server domain name; LOST protocol maps service identifiers and location information to service URLS	quoted_string	FALSE	TRUE	domain-name	Example : domainname.com
138	capwap-ac-v4	Control and Provisioning of Wireless Access Points (CAPWAP) Access Controller IP address(es) to which the client may connect	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
139	IPv4 Address MOS	IPv4 address(es) for servers providing particular types of IEEE 802.21 Mobility Service (MOS)					
140	MOS Service FQDNS	FQDN(s) for servers providing particular types of IEEE 802.21 Mobility Service (MOS)					
141	sip-ua-cs-domains [SIP UA Configuration Service Domains]	DHCP SIP user agent configuration service domains	quoted_string	FALSE	TRUE		
142	ipv4-address-andsf	Specifies the IP addresses of the Access Network Discovery and Selection Function (ANDSF) servers. available to the client. The servers are listed in order of priority.	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
143	SZTP bootstrap server URIs	Secure Zero Touch Provisioning (SZTP) bootstrap server list					
144	GoSpatial Location	Geographic location (latitude, longitude and altitude) of the client as provided by the DHCP server					

145	Force Renew Nonce Capable	Nonce value sent from the server to the client for authentication of possible future ForceRenew messages					
146	rdnss-selection	Recursive DNS Server selection for multi-interfaced nodes. This option is supported if the option value type is IP, ASCII, or HEX	string	TRUE	TRUE	uint8 ip-address ip-address domain-name	Range : 0-255 , Type : ipv4 Example : 5 1.1.1.1 2.2.2.2 "domainName.com"
147-149	Unassigned	These options are supported if the option value type is IP, ASCII, or HEX					
150	tftp-server-address	Specific to VoIP (Voice over Internet Protocol) environments, and it is used to provide the IP address or hostname of a TFTP (Trivial File Transfer Protocol) server to VoIP phones or devices during the DHCP lease negotiation process. TFTP is commonly employed in VoIP deployments to download configuration files or firmware updates required for the proper functioning of VoIP devices. The option contains the IP address or hostname of the TFTP server, allowing the VoIP device to establish a connection to the specified server and retrieve its configuration files.	ipv4address	TRUE	TRUE	ip-address [, ip-address...]	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
151	DHCP Bulk Lease Query- status code	Not supported					
152	DHCP Bulk Lease Query- base time	DHCP Bulk LeaseQuery base-time at which the LeaseQuery was created					
153	DHCP Bulk Lease Query- start time of state	DHCP Bulk LeaseQuery elapsed time since the given IP address entered its current state					
154	DHCP Bulk Lease Query- query start time	DHCP Bulk LeaseQuery filters results by the time **after** which lease binding changes are requested					
155	DHCP Bulk Lease Query- query end time	DHCP Bulk LeaseQuery filters results by the time **before** which lease binding changes are requested					
156	DHCP Bulk Lease Query- dhcp state	DHCP Bulk LeaseQuery dhcp lease state. This option is supported if the option value type is IP, ASCII, or HEX					
157	DHCP Bulk Lease Query- data source	DHCP Bulk LeaseQuery data source when two or more servers have information about the IP address binding					
158	DCHPv4 PCP Server	Port Control Protocol(PCP) server IP address(es).					
159	v4-portparams [DHCPv4 Port Parameters]	Shared IPv4 address space port parameters	quoted_string	FALSE	TRUE		

160	v4-captive-portal	The contact URI for the captive portal that the user should connect to. This option is supported if the option value type is IP, ASCII, or HEX	quoted_string	FALSE	TRUE		
161-174	Unassigned	These options are supported if the option value type is IP, ASCII, or HEX					
175	Etherboot	These options are supported if the option value type is IP, ASCII, or HEX					
176	IP Telephone-Voice-Server	Configures the IP telephone voice parameters for Avaya IP phones running as DHCP clients.					
177	Etherboot[Pkt Cable-Cable Home]	This option is supported if the option value type is IP, ASCII, or HEX					
178-207	Unassigned	These options are supported if theoption value type is IP, ASCII, or HEX					
209	PXE Configuration File	Specifies the configuration filename or file pathname to be used in a PXELINUX environment. for second stage PXE boot loading.					
210	PXE Path Prefix	Specifies a path prefix for the configuration file used in a PXELINUX environment, as specified in the PXE configuration file option [204]					
211	PXE Reboot Time	Number of seconds to wait to reboot if TFTP server is unreachable					
212	option-6rd	Service provider's 6rd prefix and 6rd border relay IPv4 address(es)	string	TRUE	TRUE	uint8 uint8 ip6-address ip-address [, ip-address ...]	uint8 Range : 0-255, Address Type : ipv6 and ipv4 Example : 1 2 2001:db8:3333:4444:5555:6666:7777:8888 1.1.1.1,2.2.2.2,..,etc
213	v4-access-domain	Local Location Information Server (LIS) discovery.Specifies the access network domain name available to the client for the purposes of discovering a Local Information Server (LIS).	quoted_string	FALSE	TRUE	domain-name	Example : domainname.com
214-219	Unassigned	These options are supported if theoption value type is IP, ASCII, or HEX					
220	Subnet allocation option (Tentatively assigned)	This options is supported if the option value type is IP, ASCII, or HEX					
221	Virtual subnet selection option (Tentatively assigned)	This option is supported if the option value type is IP, ASCII, or HEX					
222-223	Unassigned	These options are supported if the option value type is IP, ASCII, or HEX					
224-241	Reserved (Private Use)	These options are supported if the option value type is IP, ASCII, or HEX					

242	IP Tele-DataSrvr	Configures the IP telephone data parameters for Avaya IP phones running as DHCP clients.					
243-251	Reserved (Private Use)	These options are supported if the option value type is IP, ASCII, or HEX					
252	WPAD	Configures the Proxy-Auto Config (PAC) file location string for the Web Proxy Auto-Discovery (WPAD) supported DHCP clients.					
253-254	Reserved (Private Use)	These options are supported if the option value type is P, ASCII, or HEX					
255	End (None)						

Option Code	Option name	Description	Data type	Supported	Is Predefined?	Grammar	Example
1	client-id	Client Identifier (DUID of client)	string	TRUE	TRUE	string	Example : string_id_value
2	server-id	Server Identifier (DUID of server)	string	TRUE	TRUE	string	Example : string_id_value
3	ia-na	Identity Association for non-temporary addresses - includes the IAID, TI time, T2 time, and additional options for the IA for non-temporary addresses. Identity Association for Temporary addresses - includes the IAID and additional options for this IA for temporary addresses.	quoted_string	TRUE	TRUE	string	Example : string_value
4	ia-ta	Identity association for temporary addresses- includes the IAID and additional options for this IA for temporary addresses.	quoted_string	TRUE	TRUE	string	Example : string_value
5	ia-addr	IA Address option - specifies IPv6 addresses and associated preferred lifetime, valid lifetime, and options associated with an IA_NA or IA_TA. As such, this option may only appear as an option to the DHCPv6 message option OPTION_IA_TA or OPTION_IA_NA.	quoted_string	TRUE	TRUE	string	Example : string_value
6	oro	Option Request Option - used by clients to list option codes for which values are requested or by servers in a Reconfigure message to indicate which options the client should request in its subsequent Renew or Information- Request message.	uint16	TRUE	TRUE	uint16 [, uint6...]	Range : 0-65535 Example : 1024,1024,..,etc
7	preference	Preference setting by the server to facilitate client selection of DHCP server. The amount of time since the client began the current DHCP transaction in hundredths of a second. Clients are required to use this option.	uint8	TRUE	TRUE	uint8	Range : 0-255 Example : 123
8	elapsed-time	The amount of time since the client began the current DHCP transaction in hundredths of a second. Clients are required to use this option.	uint16	TRUE	TRUE	uint16	Range : 0-65535 Example : 1024
9	relay-msg	The DHCP message being relayed by a relay agent.	quoted_string	TRUE	TRUE	string	Example : string_value

11	auth	The auth DHCPv6 option is used for authentication in DHCPv6 communications. It ensures that DHCP servers and clients are verified to prevent unauthorized configurations and access. This option includes details like the authentication protocol, algorithm, and a method to detect replay attacks, providing a secure DHCPv6 environment.	string	FALSE	TRUE		
12	unicast	Server unicast option indicates the IP address to which the client may unicast messages to this server.	ipv6address	TRUE	TRUE	ip6-address	Type : ipv6 Example : 2001:db8:3333:4444:5555:6666:1.2.3.4
13	status-code	Status code option indicates a 2-byte status code and variable length status message. This option may be used as a DHCP message option or as an option within another DHCP message option	string	FALSE	TRUE	status-code [string]	
14	rapid-commit	Rapid commit option - enables a client to request a direct Reply message from the server with an IP address and parameters, bypassing the Advertise and Request messages.	flag	FALSE	TRUE	flag	values : true or false or on or off Example : true
15	user-class	User class option - analogous to user class in DHCPv4 in assisting the server in making address assignment decisions.	string	FALSE	TRUE		
16	vendor-class	Vendor class option - analogous to vendor class in DHCPv4 in conveying the vendor or manufacturer of the device or interface to assist the server in making address assignment decisions. The vendor class option includes the IANA-assigned Enterprise Number for the vendor.	quoted_string	FALSE	TRUE		
17	vendor-opts	Vendor specific information - this option includes the IANA-assigned Enterprise Number as well as one or more options, each defined with option code, length, and value.	quoted_string	TRUE	TRUE	string	Example : string_value
18	interface-id	Interface ID option - used by relay agents to convey the agent&#039;s interface ID on which the client message was received. This option may only appear in RELAY-FORW messages, and when it does, it is copied by the server to the RELAYREPL message.	string	TRUE	TRUE	string	Example : string_id_value
19	reconf-msg	Reconfigure Message option - for use in the Reconfigure message to inform the client which message to use to reconfigure: either Renew or	quoted_string	FALSE	TRUE	dhcpv6-message	Example : string_dhcpv6_message

20	reconf-accept	Reconfigure Accept option - the client populates this option if it is willing to accept Reconfigure messages from the server.	flag	FALSE	TRUE	flag	values : true or false or on or off Example : true
21	sip-servers-names	SIP Servers Domain Names option - lists domain names of the SIP outbound proxy servers that the client can use.	string	TRUE	TRUE	domain-list	Example : "domainname1.com, domainname2.com,..,etc"
22	sip-servers-addresses	SIP Servers IPv6 Address List option - lists the IPv6 addresses of the SIP outbound proxy servers that the client can use.	ipv6address	TRUE	TRUE	ip6-address [, ip6-address ...]	Type : ipv6 Example : 2001:db8:3333:4444:5555:6666:1.2.3.4, 2001:db8::1234:5678:5.6,.., etc
23	name-servers	DNS Recursive Name Server Option - lists IPv6 address(es) of DNS recursive name servers to which DNS queries may be sent by the client resolver in order of preference.	ipv6address	TRUE	TRUE	ip6-address [, ip6-address ...]	Type : ipv6 Example : 2001:db8:3333:4444:5555:6666:1.2.3.4, 2001:db8::1234:5678:5.6,.., etc
24	domain-search	Domain Search List option - provides a domain search list for client use when resolving hostnames via DNS.	string	TRUE	TRUE	domain-list	Example : "domainname1.com, domainname2.com,..,etc"
25	ia-pd	Identity Association for Prefix Delegation - includes the IAID, T1 time, T2 time and additional options for the IA_PD, including the associated prefix(es) defined within option code 26.	quoted_string	TRUE	TRUE	string	Example : string_value
26	ia-prefix	IA_PD Prefix option - specifies the IPv6 prefixes associated with the IA_PD, along with associated options and preferred and valid lifetimes. This option may only appear as an option to the DHCPv6 message option OPTION_IA PD. The prefix is specified with an 8-bit prefix length and a 128-bit IPv6 prefix.	quoted_string	TRUE	TRUE	string	Example : string_value
27	nis-servers	Network Information Service (NIS) Servers - ordered list of NIS servers by IPv6 address available to the client	ipv6address	TRUE	TRUE	ip6-address [, ip6-address ...]	Type : ipv6 Example : 2001:db8:3333:4444:5555:6666:1.2.3.4, 2001:db8::1234:5678:5.6,.., etc

28	nisp-servers	OPTION_NISP_SERVERS Network Information Service v2 (NIS+) Servers - ordered list of NIS+ servers by IPv6 address available to the client.	ipv6address	TRUE	TRUE	ip6-address [, ip6-address ...]	Type : ipv6 Example : 2001:db8:3333:4444:5555: 6666:1.2.3.4, 2001:db8::1234:5678:5.6,.., etc
29	nis-domain-name	Network Information Service (NIS) domain name - NIS domain name to be used by the client.	quoted_string	TRUE	TRUE	domain-name	Example : domainname.com
30	nisp-domain-name	OPTION_NISP_DOMAIN_ Network Information Service v2 (NIS+) domain name - NIS+ domain name to be used by the client	quoted_string	TRUE	TRUE	domain-name	Example : domainname.com
31	sntp-servers	Simple Network Time Protocol (SNTP) servers - ordered list of SNTP servers by IPv6 address available to the client.	ipv6address	TRUE	TRUE	ip6-address [, ip6-address ...]	Type : ipv6 Example : 2001:db8:3333:4444:5555: 6666:1.2.3.4, 2001:db8::1234:5678:5.6,.., etc
32	info-refresh-time	Information Refresh Option - specifies the upper bound of the number or seconds from the current time that a client should wait before refreshing information received from the DHCPv6 server, particularly for stateless DHCPv6 scenarios.	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
33	bcms-server-d	Broadcast and Multicast Service (BCMCS) Domain Name List - list of one or more FQDNS corresponding to BCMCS server(s). (BCMCS is used in 3G wireless networks to enable mobiles to receive broadcast and multicast services).	string	TRUE	TRUE	domain-list	Example : "domainname1.com, domainname2.com,..,etc"
34	bcms-server-a	Broadcast and Multicast Service (BCMCS) IPv6 Address List - list of one or more IPv6 address(es) corresponding to BCMCS server(s). (BCMCS is used in 3G wireless networks to enable mobiles to receive broadcast and multicast services).	ipv6address	TRUE	TRUE	ip6-address [, ip6-address ...]	Type : ipv6 Example : 2001:db8:3333:4444:5555: 6666:1.2.3.4, 2001:db8::1234:5678:5.6,.., etc

36	geoconf-civic	Geographical location in civic (e.g. postal) format. This option can be provided by the server to relate the location of the server, the closest network element (e.g. router) to the client or the client itself. The location information includes an ISO 3166 country code (US, DE, JP, etc.) and country-specific location information such as state, province, county, city, block, group of streets and more.	quoted_string	TRUE	TRUE	string	Example : string_value
37	remote-id	Relay Agent Remote ID option - remote identity inserted by the relay agent in RELAY-FORW message to the DHCPv6 server. This is useful in service provider environments where the &quot;edge&quot; device facing the subscriber device, inserts an identifier for the subscriber connection prior to relaying to the DHCPv6 server.	string	TRUE	TRUE	string	Example : string_value
38	subscriber-id	Relay Agent Subscriber ID option - subscriber identity inserted by the relay agent in RELAY-FORW message to the DHCPv6 server. This is useful in service provider environments where the &quot;edge&quot; device facing the subscriber device, inserts an identifier for the subscriber from which the message originated, prior to relaying to the DHCPv6 server.	quoted_string	TRUE	TRUE	string	Example : string_value
39	fqdn	Client FQDN option - indicates whether the client or the DHCP server should update DNS with the AAAA record corresponding to the assigned IPv6 address and the FQDN provided in this option. The DHCP server always updates the PTR record	quoted_string	TRUE	TRUE	string	Example : string_value
40	pana-agent	This option provides one or more IPv6 address(es) associated with PANA (Protocol for carrying Authentication for Network Access) Authentication Agents that a client can use	ipv6address	TRUE	TRUE	ip6-address [, ip6-address …]	Type : ipv6 Example : 2001:db8:3333:4444:5555:6666:1.2.3.4, 2001:db8::1234:5678:5.6,.., etc
41	new-posix-timezone	Time zone to be used by the client in IEEE 1003.1 format (POSIX - portable operating system interface). This format enables textual representation of time zone and daylight savings time information.	string	TRUE	TRUE	text	Example : string_timezone_value
42	new-tzdb-timezone	Time zone (TZ) database entry referred to by entry name. The client must have a copy of the TZ database, which it queries for the corresponding entry to determine its time zone.	string	TRUE	TRUE	text	Example : string_timezone_value

43	ero	Relay Agent Echo Request option - used by relay agents in the RELAY_ FORW message to request that the DHCPv6 sever echo back certain requested relay agent options, even if not supported on the sever. (DHCPv4 servers always echo back relay agent option [80] information but this is not required in DHCPv6, hence this option for relay agents requiring such echo back).	uint16	TRUE	TRUE	uint16 [, uint6...]	Range : 0-65535 Example : 1024,1024,..,etc
44	lq-query	The Query option is used in the LEASEQUERY message to identify the query information being requested. This option includes the Query type (by IA address or client ID option), link address to which the query applies and query options.	quoted_string	TRUE	TRUE	string	Example : string_value
45	client-data	Client Data - this option contains the query response information for the requested client data within a LEASEQUERY-REPLY message. At a minimum this option includes the client identifier (OPTION_CLIENTID), the IA address or prefix (OPTION_IAADDR and/or OPTION_IAPREFIX) and client last transaction time (OPTION_CLT_TIME).	quoted_string	TRUE	TRUE	string	Example : string_value
46	clt-time	Client Last Transaction Time - indicates the number of seconds since the server last communicated with the client referenced by the lease query. This option is encapsulated within the OPTION_CLIENT_DATA option within a LEASEQUERY-REPLY message.	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
47	lq-relay-data	Relay data - used in a LEASEQUERY-REPLY message to provide the relay agent information associated with the client information requested. This option includes the relay agent address from which the client&#039;s relay information was received along with the complete relayed message. Client link - identifies one or more links on which the queried client has DHCPv6 bindings. The queried client can be identified by address or client ID.	string	TRUE	TRUE	ip6-address string	Example : 2001:db8:3333:4444:5555:6666:1.2. 3.4, "string_value"
48	lq-client-link	Client link - identifies one or more links on which the queried client has DHCPv6 bindings. The queried client can be identified by address or client ID.	ipv6address	TRUE	TRUE	ip6-address [, ip6-address ...]	Type : ipv6 Example : 2001:db8:3333:4444:5555:6666:1.2.3.4, 2001:db8::1234:5678:5.6,.., etc
51	v6-lost	Location to Service Translation (LOST) server domain name; LOST protocol maps service identifiers and location information to service URLS.	string	FALSE	TRUE	domain-name	Example : domainname.com

52	capwap-ac-v6	Control and Provisioning of Wireless Access Points (CAPWAP) Access Controller IPv6 address(es) to which the client may connect.	ipv6address	TRUE	TRUE	ip6-address [, ip6-address ...]	Type : ipv6 Example : 2001:db8:3333:4444:5555: 6666:1.2.3.4, 2001:db8::1234:5678:5.6,.., etc
53	relay-id	DHCPv6 Bulk LeaseQuery - requests lease and prefix delegation bindings for a specified relay agent identified by its DUID in this option.	string	TRUE	TRUE	string	Example : string_value
57	v6-access-domain	OPTION_V6_ACCESS_ Local Location Information Server (LIS) discovery.	quoted_string	FALSE	TRUE	domain-name	Example : domainname.com
58	sip-ua-cs-list	OPTION_SIP_UA_CS_LIST DHCP SIP user agent configuration service domains.	string	TRUE	TRUE	domain-list	Example : "domainname1.com, domainname2.com,..,etc"
59	bootfile-url	URLto a boot file.	quoted_string	TRUE	TRUE	text	Example : text_url_value
60	bootfile-param	Boot file parameter list, similar to command line arguments in modern operating systems	quoted_string	TRUE	TRUE	string	Example : string_value
61	client-arch-type	Client system architecture type for network boot.	uint16	TRUE	TRUE	uint16 [, uint6...]	Range : 0-65535 Example : 1024,1024,..,etc
62	nii	Client universal network device interface (UNDI) identifier for network boot.	uint8	TRUE	TRUE	uint8 uint8 uint8	Range : 0-255 Example : 1 2 3
64	aftr-name	FQDN of the Address Family Transition Router (AFTR) used for Dual-Stack Lite.	string	FALSE	TRUE	domain-name	Example : domainname.com
65	erp-local-domain-name	Extensible Authentication Protocol (EAP) Re-authentication Protocol (ERP) local domain.	string	FALSE	TRUE	domain-name	Example : domainname.com
66	rsoo	Relay agent-supplied DHCP options.	string	FALSE	TRUE	TRUE	
67	pd-exclude	Prefix exclusion from a prefix set.	string	FALSE	TRUE	TRUE	
74	rdnss-selection	Recursive DNS Server selection for multi-interfaced nodes.	string	TRUE	TRUE	ip6-address uint8 domain-name	Example : 2001:db8:3333:4444:5555: 6666:1.2. 3.4 10 "domainname.com"
79	client-linklayer-addr	Client's link layer address provided by an on-link relay agent	string	TRUE	TRUE	string	Example : string_value
80	link-address	IPv6 address of the DHCP client for which a reconfigure command is requested by the relay agent.	ipv6address	TRUE	TRUE	ip6-address	Type : ipv6 Example : 2001:db8:3333:4444:5555: 6666:1.2.3.4

82	solmax-rt	Provided by the server to override the client default value of SOL_MAX_RT (maximum Solicit timeout).	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
83	inf-max-rt	Provided by the server to override the client default value of INF_MAX_RT (maximum Information-Request timeout).	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
87	dhcpv4-msg	DHCPv4 message encoded within a DHCPv6 message (DHCPv4 over DHCPv6, DHCP 406).	quoted_string	TRUE	TRUE	string	Example : string_dhcpv4_message
88	dhcp4-o-dhcp6-	DHCP 406 server IPv6 address(es).	ipv6address	TRUE	TRUE	ip6-address	ipv6
103	v6-captive-portal	Informs a client that it is behind a captive portal device and provides a URI to access an authentication page.	string	TRUE	TRUE	string	Example : string_value
135	relay-source-port	UDP port the server should use in response to this relay agent. Some configurations do not use the well-known DHCP source port (547) for scalability reasons.	uint16	TRUE	TRUE	uint16	Range : 0-65535 Example : 1024
143	ipv6-address-andsf	IP address of the Access Network Discovery and Selection Function (ANDSF).	ipv6address	TRUE	TRUE	ip6-address	ipv6
144-65535	Unassigned						

Option Code	Option name	Description	Data type	Supported	Is Predefined ?	Grammar	Example
1	default-lease-time	Specify the length of time in seconds to be assigned to a lease if the client requesting the lease doesn't provide a specific expiration time. This applies to both DHCPv4 and DHCPv6 leases, also referred to as the "valid lifetime" in DHCPv6. The default duration is set to 43200 seconds.	uint32	TRUE	TRUE	time	Range : 0-4,294,967,295 Example : 12345
2	max-lease-time	Time should be the maximum length in seconds that will be assigned to a lease. If not defined, the default maximum lease time is 86400. The only exception to this is that Dynamic BOOTP lease lengths, which are not specified by the client, are not limited by this maximum.	uint32	TRUE	TRUE	time	Range : 0-4,294,967,295 Example : 12345
3	min-lease-time	Time should be the minimum length in seconds that will be assigned to a lease. The default is the minimum of 300 seconds or max-lease-time.	uint32	TRUE	TRUE	time	Range : 0-4,294,967,295 Example : 12345
4	dynamic-bootp-lease-cutoff	The "dynamic-bootp-lease-cutoff" DHCP parameter is used to set a time limit for dynamically allocated BOOTP leases. When configured, this parameter determines the maximum duration for BOOTP leases obtained dynamically from the DHCP server. After this specified cutoff time, BOOTP leases expire, and clients need to renew their leases or request new ones. This parameter helps in managing lease durations for BOOTP clients, ensuring efficient address allocation within the network.	uint32	TRUE	TRUE	date	Range : 0-4,294,967,295 Example : 12345
5	dynamic-bootp-lease-length	The dynamic-bootp-lease-length DHCP parameter sets the duration, in seconds, for a dynamic BOOTP lease. This parameter influences the length of time an IP address is leased to a client that uses BOOTP to obtain its configuration. If a client doesn't request a specific lease duration, this parameter determines the default length of the lease. Adjusting this parameter allows administrators to control the lease duration for BOOTP clients in a DHCP environment.	string	TRUE	TRUE	length	Example : string_lease_length

6	boot-unknown-clients	If the boot-unknown-clients statement is present and has a value of false or off, then clients for which there is no host declaration will not be allowed to obtain IP addresses. If this statement is not present or has a value of true or on, then clients without host declarations will be allowed to obtain IP addresses, as long as those addresses are not restricted by allow and deny statements within their pool declarations.	control	TRUE	TRUE	allow unknown-clients;
						deny unknown-clients;
						ignore unknown-clients;
							Example : allow unknown-clients / deny unknown-clients / ignore unknown-clients

7	dynamic-bootp	The dynamic-bootp DHCP parameter is related to the behavior of the DHCP server in handling BOOTP (Bootstrap Protocol) clients. When set to dynamic-bootp, it allows the DHCP server to dynamically allocate IP addresses to BOOTP clients, treating them similarly to DHCP clients. This parameter enables the DHCP server to respond to BOOTP requests by allocating IP addresses dynamically rather than requiring fixed or manual configurations. This flexibility is particularly useful when dealing with devices that use BOOTP but benefit from dynamic IP assignment, such as diskless workstations or embedded systems.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true

8	bootp	The bootp flag is used to tell dhcpd whether or not to respond to bootp queries. Bootp queries are allowed by default.	control	TRUE	TRUE	allow bootp;
						deny bootp;
						ignore bootp;
							Example : allow bootp / deny bootp / ignore bootp

9	booting	The booting flag is used to tell dhcpd whether or not to respond to queries from a particular client. This keyword only has meaning when it appears in a host declaration. By default, booting is allowed, but if it is disabled for a particular client, then that client will not be able to get an address from the DHCP server.	control	TRUE	TRUE	allow booting;
						deny booting;
						ignore booting;
							Example : allow booting / deny booting / ignore booting

10	one-lease-per-client	If this flag is enabled, whenever a client sends a DHCPREQUEST for a particular lease, the server will automatically free any other leases the client holds. This presumes that when the client sends a DHCPREQUEST, it has forgotten any lease not mentioned in the DHCPREQUEST - i.e., the client has only a single network interface and it does not remember leases it's holding on networks to which it is not currently attached. Neither of these assumptions are guaranteed or provable, so we urge caution in the use of this statement.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
11	get-lease-hostnames	The get-lease-hostnames statement is used to tell dhcpd whether or not to look up the domain name corresponding to the IP address of each address in the lease pool and use that address for the DHCP hostname option. If flag is true, then this lookup is done for all addresses in the current scope. By default, or if flag is false, no lookups are done.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
12	use-host-decl-names	The use-host-decl-names DHCP parameter determines whether the DHCP server should use the host declarations' names as unique identifiers. When enabled, the server utilizes the names specified in the host declarations to identify clients. This parameter is useful for cases where host names are unique and provide an additional layer of identification, allowing for more straightforward and intuitive DHCP configuration management.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true

13	use-lease-addr-for-default-route	If the use-lease-addr-for-default-route parameter is true in a given scope, then instead of sending the value specified in the routers option (or sending no value at all), the IP address of the lease being assigned is sent to the client. This supposedly causes Win95 machines to ARP for all IP addresses, which can be helpful if your router is configured for proxy ARP. The use of this feature is not recommended, because it won’t work for many DHCP clients.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
14	min-secs	The min-secs DHCP parameter specifies the minimum elapsed time, in seconds, that a DHCP client must wait before sending a DHCPREQUEST message. This parameter helps prevent DHCP clients from immediately renewing their lease, allowing the DHCP server sufficient time to process requests and allocate IP addresses. It contributes to the efficient and orderly operation of the DHCP protocol.	uint8	TRUE	TRUE	seconds	Range : 0-255 Example : 5
15	filename	The filename statement can be used to specify the name of the initial boot file which is to be loaded by a client. The filename should be a filename recognizable to whatever file transfer protocol the client can be expected to use to load the file.	quoted_string	TRUE	TRUE	filename	Example : /tftpboot/netbsd.alphapc-diskless
16	server-name	The server-name DHCP parameter is used to specify the hostname of the DHCP server. This information is provided to DHCP clients during the lease negotiation process. Clients can use the server name for identification or informational purposes. The server-name parameter is part of DHCP option 66 and is commonly used in scenarios where clients need to know the name of the server from which they are obtaining their configuration.	quoted_string	TRUE	TRUE	name	Example : server_name
17	next-server	The next-server statement is used to specify the host address of the server from which the initial boot file (specified in the filename statement) is to be loaded. Server-name should be a numeric IP address or a domain name.	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1

18	authoritative	The authoritative statement in DHCP configuration indicates whether the server should consider configuration information for a network segment as authoritative. By default, DHCP servers assume a non-authoritative stance to avoid disruptions. For authoritative networks, include authoritative; at the top of the configuration file, allowing sending DHCPNAK messages to misconfigured clients, crucial for correct IP address acquisition during subnet changes.	flag	TRUE	TRUE	authoritative;
						not authoritative;
							Example : authoritative / not authoritative

19	vendor-option-space	This allows administrators to specify custom DHCP options tailored for specific vendors or devices. Each vendor-option-space can include unique configuration parameters that are applicable only to devices from a particular vendor. This parameter provides a flexible way to customize DHCP settings for diverse network environments and ensure compatibility with devices that have vendor-specific requirements.	string	TRUE	TRUE	string	Example : string_value
20	always-reply-rfc1048	The always-reply-rfc1048 flag in DHCP addresses issues with BOOTP clients expecting RFC1048-style responses but deviating when sending requests. If clients are not receiving configured options, setting this flag in the client's host declaration ensures the DHCP server responds with an RFC-1048-style vendor options field, applicable at any scope level.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
21	site-option-space	The site-option-space DHCP parameter is used to define a custom option space that can be associated with a particular site or location within a DHCP server configuration. This parameter allows network administrators to tailor and configure specific DHCP options for a defined site, enabling customized settings and parameters for devices within that location. By using site-option-space, administrators can efficiently manage and apply distinct DHCP configurations based on the requirements of different sites or geographical segments within a network.	quoted_string	TRUE	TRUE	name	Example : string_value
22	always-broadcast	To ensure DHCP and BOOTP clients adhere to the protocol requirements, it is necessary for them to set the broadcast bit in the flags field of the BOOTP message header. Regrettably, some clients neglect this step, risking non-receipt of responses from the DHCP server. To address this, the DHCP server can consistently broadcast its responses to clients by activating this flag for the relevant scope. These scopes can be defined within a conditional statement, as a parameter for a class, or as a parameter for a host declaration. To mitigate excessive broadcast traffic on your network, it is advisable to limit the use of this option to as few clients as possible. Notably, certain clients like the Microsoft DHCP client, OpenTransport, and ISC DHCP clients do not exhibit this issue.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
23	ddns-domainname	The name parameter should be the domain name that will be appended to the client’s hostname to form a fully-qualified domain-name (FQDN).	quoted_string	TRUE	TRUE	name	Example : domainname.com
24	ddns-hostname	The name parameter should be the hostname that will be used in setting up the client’s A and PTR records. If no ddns-hostname is specified in scope, then the server will derive the hostname automatically, using an algorithm that varies for each of the different update methods.	quoted_string	TRUE	TRUE	name	Example : string_hostname

25	ddns-rev-domainname	The name parameter should be the domain name that will be appended to the client’s reversed IP address to produce a name for use in the client’s PTR record. By default, this is "in-addr.arpa.", but the default can be overridden here. The reversed IP address to which this domain name is appended is always the IP address of the client, in dotted quad notation, reversed - for example, if the IP address assigned to the client is 10.17.92.74, then the reversed IP address is 74.92.17.10. So a client with that IP address would, by default, be given a PTR record of 10.17.92.74.in-addr.arpa.	quoted_string	TRUE	TRUE	name	Example : domainname.com
26	lease-file-name	The lease-file-name parameter in DHCP configuration specifies the file name and location where the DHCP server stores information about active leases. This file is crucial for maintaining a record of leased IP addresses, lease durations, and associated client information. Accurate management of the lease file is essential for the DHCP server to allocate and renew leases effectively.	quoted_string	TRUE	TRUE	name	Example : string_lease_file_name
27	pid-file-name	The pid-file-name parameter designates the file name and location where the DHCP server stores its process ID (PID). The PID file is essential for system administration, enabling easy identification and control of the DHCP server's process. It is commonly used in scripts or commands to manage the DHCP server, such as starting, stopping, or restarting the server process.	quoted_string	TRUE	TRUE	name	Example : string_value

28	duplicates	The "duplicates" DHCP parameter is used to control how the DHCP server handles duplicate client identifiers within the same network segment. This parameter is crucial in scenarios where multiple clients attempt to obtain IP addresses with the same client identifier, creating potential conflicts. By setting the "duplicates" parameter, administrators can determine whether the DHCP server allows or denies such duplicate client identifiers. The two main values for this parameter are "deny" (default) and "allow," each dictating the server's response to conflicting client identifiers. This parameter helps manage IP address allocation and prevent issues arising from duplicate client identifiers on the network.	control	TRUE	TRUE	allow duplicates;
						deny duplicates;
							Example : allow duplicates / deny duplicates

29	declines	The "declines" DHCP parameter refers to a mechanism in DHCP (Dynamic Host Configuration Protocol) where a client can inform the DHCP server that it is rejecting or declining a offered lease. When a client declines a lease, it indicates that it cannot or does not want to use the provided network configuration. This could happen, for example, if the client detects a conflict with the offered IP address. The DHCP server keeps track of declined leases, helping to manage IP address assignments and avoid potential conflicts.	control	TRUE	TRUE	allow declines;
						deny declines;
						ignore declines;
							Example : allow declines / deny declines / ignore declines

30	ddns-updates	The ddns-updates parameter controls whether or not the server will attempt to do a DNS update when a lease is confirmed. Set this to off if the server should not attempt to do updates within a certain scope. The ddns-updates parameter is on by default. To disable DNS updates in all scopes, it is preferable to use the ddns-update-style statement, setting the style to none.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
31	omapi-port	The omapi-port statement causes the DHCP server to listen for OMAPI connections on the specified port. This statement is required to enable the OMAPI protocol, which is used to examine and modify the state of the DHCP server as it is running.	uint16	TRUE	TRUE	port	Range : 0-65535 Example : 8000
32	local-port	This statement causes the DHCP server to listen for DHCP requests on the UDP port specified in port, rather than on port 67.	uint16	TRUE	TRUE	port	Range : 0-65535 Example : 8000
33	limited-broadcast-address	Specifies the limited broadcast address that the DHCP server should use when responding to DHCPINFORM messages. DHCPINFORM messages are typically used by clients to request specific configuration information without obtaining a new IP address lease. The limited broadcast address is a broadcast address that restricts the scope of the response to the local network segment, preventing unnecessary traffic propagation. This parameter helps optimize the handling of DHCPINFORM requests by ensuring that responses are directed only to the local network, improving network efficiency.	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
34	remote-port	This statement causes the DHCP server to transmit DHCP responses to DHCP clients upon the UDP port specified in port, rather than on port 68. In the event that the UDP response is transmitted to a DHCP Relay, the server generally uses the local-port configuration value. Should the DHCP Relay happen to be addressed as 127.0.0.1, however, the DHCP Server transmits its response to the remote-port configuration value. This is generally only useful for testing purposes, and this configuration value should generally not be used.	uint16	TRUE	TRUE	port	Range : 0-65535 Example : 8000

35	local-address	Used to specify the local address that the DHCP server should use for outgoing packets. It allows network administrators to set a specific source address for DHCP messages, ensuring they are sent from a designated IP address on the server. This parameter can be beneficial in multi-homed or complex network configurations where the DHCP server has multiple network interfaces, and administrators want to control the source address for DHCP communication. The local-address parameter helps in managing network traffic and ensuring DHCP messages are appropriately sourced from a specified IP address on the server.	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1
36	omapi-key	Used to define the key that grants access to the OMAPI (Object Management Application Programming Interface) in ISC DHCP. OMAPI is a feature that enables external applications to manage and manipulate DHCP server configuration dynamically. This parameter involves setting up a cryptographic key, and the omapi-key specifies the key's value. The key acts as a form of authentication, allowing external applications to securely interact with the DHCP server through OMAPI. It enhances security by ensuring that only authorized entities with the correct key can make dynamic changes to the DHCP server configuration using OMAPI.	quoted_string	FALSE	TRUE		
37	stash-agent-options	used to store DHCP options received from clients in a temporary storage, commonly known as a "stash." This feature is particularly useful when DHCP servers receive options from DHCP relay agents but need to process them later or under specific conditions. The stash-agent-options parameter allows DHCP servers to retain the received options until they are explicitly referenced or needed for further processing, enabling more flexible and controlled DHCP option handling.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
38	ddns-ttl	The ddns-ttl DHCP parameter governs the Time-To-Live (TTL) value for Dynamic DNS (DDNS) updates. It specifies the duration for which DDNS records should be considered valid. When set, this parameter influences how long DNS servers cache and use DDNS records, ensuring timely updates and efficient management of DNS entries related to DHCP-assigned addresses. Adjusting ddns-ttl allows network administrators to control the lifespan of DDNS records in DHCP environments.	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
39	ddns-update-style	The style parameter must be one of **ad-hoc, interim or none.** The ddns-update-style statement is only meaningful in the outer scope - it is evaluated once after reading the dhcpd.conf file, rather than each time a client is assigned an IP address, so there is no way to use different DNS update styles for different clients. The default is none.	string	TRUE	TRUE	style	Example : string_style

40	client-updates	The client-updates flag tells the DHCP server whether or not to honor the client's intention to do its own update of its A record. This is only relevant when doing interim DNS updates. See the documentation under the heading THE INTERIM DNS UPDATE SCHEME for details.	control	TRUE	TRUE	allow client-updates;
						deny client-updates;
							Example : allow client-updates / deny client-updates

41	update-optimization	The update-optimization DHCP parameter is related to DNS updates in DHCP. It optimizes the process of updating DNS records by allowing the DHCP server to determine whether an update is necessary before attempting to update the DNS server. This parameter helps improve efficiency by reducing unnecessary DNS update traffic, ensuring that DNS records are updated only when required, and minimizing the impact on both DHCP and DNS servers.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
42	ping-check	The ping-check DHCP parameter is used to enable or disable the DHCP server's ability to perform a ping check before allocating an IP address to a client. When enabled, the DHCP server sends an ICMP echo (ping) request to the IP address it intends to assign to the client. If the address responds, it is considered in use, and the server will not assign it to the client. This helps prevent IP address conflicts. If disabled, the server assigns the IP address without checking for its current usage.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
43	update-static-leases	The update-static-leases DHCP parameter is a configuration option that, when enabled, allows the DHCP server to update static leases in its database dynamically. This means that if a client with a static lease configuration sends a DHCP request with a different IP address, the server will update its records to reflect the new address. This parameter is useful in scenarios where administrators want to accommodate changes in network configurations without manually adjusting static lease entries. Enabling update-static-leases ensures that static lease information remains synchronized with the actual IP addresses assigned to clients, providing greater flexibility in network management.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
44	log-facility	The log-facility DHCP parameter determines the facility or destination where DHCP server log messages are sent. It specifies the syslog facility that will record DHCP-related events, aiding in monitoring and troubleshooting. This parameter allows administrators to customize log handling, directing DHCP logs to specific locations or systems for efficient management of DHCP server activities.	string	TRUE	TRUE	facility	values : auth, authpriv, cron, daemon, ftp, kern, lpr, mail, mark, news, ntp, security, syslog, user, uucp, and local0 through local7 Example : local7

45	do-forward-updates	The do-forward-updates DHCP parameter determines whether the DHCP server should forward DNS updates that it receives from clients. When set to "true," the server forwards these updates to the DNS server. This parameter is particularly relevant in environments where DHCP is integrated with DNS to ensure that DNS records are accurately updated when clients obtain or release IP addresses.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
46	ping-timeout	The ping-timeout DHCP parameter determines the maximum duration the DHCP server will wait for a response to a ping before considering the IP address as available for assignment. If a client doesn't respond to the ping within the specified timeout, the DHCP server assumes the address is unused and may offer it to a requesting client. This parameter helps manage IP address allocation by setting a threshold for determining the availability of addresses based on client responsiveness to ping requests.	uint32	TRUE	TRUE	seconds	Range : 0-4,294,967,295 Example : 12345
47	infinite-is-reserved	The infinite-is-reserved DHCP parameter is used to reserve the IP address representing infinity or unlimited time within the DHCP lease duration. By default, the concept of "infinite" in DHCP leases implies an indefinite lease time. However, certain scenarios or implementations might require reserving a specific IP address to represent infinity, and the infinite-is-reserved parameter facilitates this customization within the DHCP server configuration. When enabled, the DHCP server designates a reserved IP address to represent infinite lease times, providing flexibility in managing lease durations for specific clients or purposes.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
48	update-conflict-detection	If the update-conflict-detection parameter is true, the server will perform standard DHCID multiple-client, one-name conflict detection. If the parameter has been set false, the server will skip this check and instead simply tear down any previous bindings to install the new binding without question. The default is true.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true

49	leasequery	The leasequery flag tells the DHCP server whether or not to answer DHCPLEASEQUERY packets. The answer to a DHCPLEASEQUERY packet includes information about a specific lease, such as when it was issued and when it will expire. By default, the server will not respond to these packets.	control	TRUE	TRUE	allow leasequery;
						deny leasequery;
							Example : allow leasequery / deny leasequery

50	adaptive-lease-time-threshold	When the number of allocated leases within a pool rises above the percentage specified along with this option, the DHCP server decreases the lease length for new clients within this pool to min-lease-time seconds. Clients renewing an already valid (long) leases get at least the remaining time from the current lease. Since the leases expire faster, the server may either recover more quickly or avoid pool exhaustion entirely. Once the number of allocated leases drop below the threshold, the server reverts back to normal lease times. Valid percentages are between 1 and 99.	uint8	TRUE	TRUE	percentage	Range : 0-255 Example : 5
51	do-reverse-updates	The do-reverse-updates DHCP parameter is used to control whether the DHCP server will perform reverse DNS updates for dynamically assigned IP addresses. When this parameter is set to "true," the DHCP server will attempt to update the reverse DNS mapping (PTR record) in the DNS server corresponding to the dynamically assigned IP address. If set to "false," the DHCP server will refrain from performing reverse DNS updates. This parameter provides administrators with flexibility in managing DNS records associated with dynamically assigned addresses, allowing them to align DHCP and DNS configurations based on specific network requirements.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
52	fqdn-reply	The fqdn-reply DHCP parameter is used to control how the DHCP server responds to Fully Qualified Domain Name (FQDN) requests from DHCP clients. When enabled, it allows the DHCP server to include the FQDN option in the DHCPACK response, providing the client with its fully qualified domain name. This parameter is particularly useful in scenarios where DHCP clients need to obtain their FQDN dynamically. The FQDN can be utilized for various network operations, including hostname resolution.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
53	preferred-lifetime	Determines the preferred duration, in seconds, for which a client can retain a dynamically assigned IP address. This parameter is primarily associated with DHCPv6 (Dynamic Host Configuration Protocol for IPv6). During the preferred lifetime, the client is encouraged to renew its lease, and the DHCP server may extend the lease if the client requests. After the preferred lifetime elapses, the client is still able to use the assigned IP address until the lease expires, but the server may offer a different address upon renewal. The preferred-lifetime parameter helps manage IP address assignment and renewal in IPv6 networks.	uint32	TRUE	TRUE	seconds	Range : 0-4,294,967,295 Example : 12345

54	dhcpv6-lease-file-name	The dhcpv6-lease-file-name DHCP parameter refers to the configuration setting specifying the file name used to store DHCPv6 lease information. This parameter defines the location and name of the file where the DHCP server records lease details for IPv6 addresses assigned to clients. It plays a crucial role in DHCPv6 server management, ensuring the persistence and retrieval of lease data even in the event of server restarts or failures. The specified file captures essential information about leased IPv6 addresses, aiding in the efficient allocation and tracking of address assignments within the DHCPv6 environment.	quoted_string	TRUE	TRUE	name	Example : string_file_name
55	dhcpv6-pid-file-name	The dhcpv6-pid-file-name DHCP parameter is used to specify the file where the DHCPv6 server process ID (PID) is stored. This file contains the unique identifier assigned to the DHCPv6 server process, facilitating monitoring, control, and management of the DHCPv6 service. This parameter allows administrators to customize the location and filename for storing the DHCPv6 server process ID information.	quoted_string	TRUE	TRUE	name	Example : string_file_name
56	limit-addrs-per-ia	By default, the DHCPv6 server will limit clients to one IAADDR per IA option, meaning one address. If you wish to permit clients to hang onto multiple addresses at a time, configure a larger number here. Note that there is no present method to configure the server to forcibly configure the client with one IP address per each subnet on a shared network. This is left to future work.	uint32	TRUE	TRUE	number	Range : 0-4,294,967,295 Example : 12345
57	limit-prefs-per-ia	The limit-prefs-per-ia DHCP parameter is designed to restrict the number of preference values assigned to individual Identity Associations (IA) within DHCPv6. In DHCPv6, Identity Associations represent address assignments to clients. This parameter enables administrators to set a limit on the number of preferences (IA_Prefix and IA_NA options) a client can include in its requests. By controlling the preferences per IA, network administrators can manage address assignment policies more effectively and prevent potential misconfigurations or excessive requests from clients.	uint32	TRUE	TRUE	uint32	Range : 0-4,294,967,295 Example : 12345
58	delayed-ack	The delayed-ack DHCP parameter is designed to introduce a delay before sending acknowledgment (ACK) messages in response to DHCP requests. By enabling delayed-ack, the DHCP server intentionally delays sending ACK messages to clients, which can help optimize network performance and reduce the potential for network congestion.	uint16	FALSE	TRUE	count	

59	max-ack-delay	The max-ack-delay DHCP parameter sets a maximum limit on the delay introduced by the delayed-ack feature. Admins can configure max-ack-delay to control the upper limit of the delay before ACK messages are sent. This ensures that the delay introduced by delayed-ack does not exceed a specified threshold, providing a balance between optimization and timely responses.	uint32	FALSE	TRUE	microseconds	
78	dhcp-cache-threshold	The dhcp-cache-threshold parameter in DHCP configuration determines the threshold at which the DHCP server starts caching information about clients. When the number of active leases exceeds this threshold, the server initiates caching, improving performance by storing client information. This parameter helps optimize DHCP server efficiency, particularly in environments with a large number of active leases. Adjusting the dhcp-cache-threshold value allows administrators to tailor caching behavior based on network dynamics and resource considerations.	uint8	FALSE	TRUE	percentage	
79	dont-use-fsync	The dont-use-fsync DHCP parameter is a configuration option that advises the DHCP server not to use the fsync system call when writing lease state information to disk. By default, DHCP servers often use fsync to ensure that lease information is durably stored on disk. However, using fsync can impact performance, and in some cases, administrators might choose to disable it using the dont-use-fsync parameter to optimize DHCP server performance at the cost of potential data loss in the event of a server crash or failure.	flag	FALSE	TRUE	flag	
80	ddns-local-address4	The ddns-local-address4 DHCP parameter is used to specify the local IPv4 address that the DHCP server should use when communicating with the Dynamic Domain Name System (DDNS) server for updates. It allows administrators to define a specific IPv4 address for the DHCP server to utilize when performing updates to the DDNS, providing flexibility in network configurations. This parameter is particularly useful when the DHCP server has multiple network interfaces, and the administrator wants to designate a specific source address for DDNS updates.	ipv4address	FALSE	TRUE	ip-address	
81	ddns-local-address6	The ddns-local-address6 DHCP parameter is used to specify the local IPv6 address that the DHCP server should use when sending Dynamic DNS (DDNS) updates. This parameter allows administrators to control which IPv6 address the server uses for DDNS updates, providing flexibility in network configurations. The specified IPv6 address is used as the source address for the updates sent to the DNS server, ensuring accurate and secure updates to the DNS records associated with DHCP clients.	ipv6address	FALSE	TRUE	ip6-address	

82	ignore-client-uids	The ignore-client-uids DHCP parameter is utilized to handle situations where clients might send non-standard or inconsistent Client Identifier (UID) information. When enabled, this parameter instructs the DHCP server to disregard irregular client UIDs and focus on other identification methods. This can be useful in scenarios where clients deviate from standard UID formats, ensuring more robust and flexible DHCP service delivery.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
83	log-threshold-low	The log-threshold-low DHCP parameter establishes a threshold for logging low-level informational messages in the DHCP server's log. When set, it determines the lower limit for messages that will be recorded, helping administrators manage the volume of log entries. Fine-tuning this parameter allows DHCP server operators to control the verbosity of log information, aiding in efficient log analysis and troubleshooting.	uint8	FALSE	TRUE	percentage	
84	log-threshold-high	The log-threshold-high DHCP parameter sets the threshold for high-level logging in the DHCP server. This parameter controls when the server should generate detailed log entries, helping administrators monitor and troubleshoot DHCP activity. By adjusting the log-threshold-high value, administrators can control the level of detail recorded in logs, striking a balance between comprehensive logging for diagnostic purposes and minimizing unnecessary log entries.	uint8	FALSE	TRUE	percentage	
85	echo-client-id	The echo-client-id DHCP parameter controls whether the DHCP server echoes back the Client Identifier (option 61) in its response. When set, the server includes the received Client Identifier in the DHCP Offer or ACK messages. If not configured, the server generates its own Client Identifier. This parameter is useful in scenarios where the client relies on a specific identifier for consistent lease management. The echo-client-id parameter provides control over whether the server echoes the client's identifier or assigns its own.	flag	FALSE	TRUE	flag	
86	server-id-check	The server-id-check DHCP parameter is used to enforce stricter checking of DHCP server identifiers in incoming DHCP messages. When enabled, this parameter ensures that the DHCP server only responds to messages if the server identifier matches the server's own identifier. This helps enhance security by mitigating the risk of unauthorized or rogue DHCP servers on the network. The server-id-check parameter adds an extra layer of validation to DHCP communications, reducing the likelihood of unintended interactions with unauthorized DHCP servers.	flag	FALSE	TRUE	flag	

87	prefix-length-mode	The prefix-length-mode DHCP parameter is used to define the mode for specifying prefix lengths in DHCP configurations. It allows the network administrator to choose how to specify the length of IPv6 prefixes assigned by the DHCP server. This parameter supports two modes: non-strict: In this mode, the prefix length can be specified without adhering strictly to the standards, allowing flexibility in configuration. It provides a lenient approach to accommodate various network requirements. strict: This mode enforces strict adherence to the standards when specifying IPv6 prefix lengths. It ensures that the configured prefix lengths align precisely with the established standards, promoting consistency and compliance with IPv6 addressing norms. Network administrators can choose the appropriate prefix-length-mode based on their specific needs and the desired level of adherence to IPv6 standards in their DHCP configurations.	string	FALSE	TRUE	mode	
88	dhcpv6-set-tee-times	The "dhcpv6-set-tee-times" flag, when enabled, allows for the automatic setting of T1 and T2 timers. T1 and T2 values are derived from dhcp-renewal-time and dhcp-rebinding-time, respectively, or can be left for the client to choose with a value of zero. It's essential to note that "dhcpv6-set-tee-times" is intended as a transitional feature and may be removed in future releases. Once removed, DHCPv6 servers by default will be using configured values.	flag	FALSE	TRUE	flag	
89	abandon-lease-time	Specifies the maximum amount of time (in seconds) that an abandoned IPv4 lease remains unavailable for assignment to a client. Abandoned leases will only be offered to clients if there are no free leases. If not defined, the default abandon lease time is 86400 seconds (24 hours). Note the abandoned lease time for a given lease is preserved across server restarts. The parameter may only be set at the global scope and is evaluated only once during server startup. Values less than sixty seconds are not recommended as this is below the ping check threshold and can cause leases once abandoned but since returned to the free state to not be pinged before being offered. If the requested time is larger than 0x7FFFFFFF - 1 or the sum of the current time plus the abandoned time isgreater than 0x7FFFFFFF it is treated as infinite.	uint32	FALSE	TRUE	time	

90	use-eui-64	The use-eui-64 DHCP parameter is employed to enable or disable the generation of IPv6 addresses based on the EUI-64 format. When activated, this parameter allows DHCPv6 to construct IPv6 addresses using the Extended Unique Identifier (EUI-64) derived from the client's MAC address. This setting enhances address predictability and simplifies IPv6 address configuration. However, its use depends on network policies and security considerations, as it may expose information about the device's network interface.	flag	FALSE	TRUE	flag	
91	check-secs-byte-order	deprecated	flag	FALSE	TRUE	flag	
92	persist-eui-64-leases	The persist-eui-64-leases DHCP parameter is used to enable the persistence of leases based on Extended Unique Identifier (EUI-64) for IPv6 addresses. When set, this parameter ensures that DHCPv6 leases, specifically those generated using the EUI-64 mechanism, persist across server restarts. EUI-64 is a method for deriving IPv6 addresses from MAC addresses. Enabling persist-eui-64-leases helps maintain consistent addressing for devices using EUI-64 when the DHCPv6 server undergoes restarts or reboots.	flag	FALSE	TRUE	flag	
93	ddns-dual-stack-mixed-mode	The ddns-dual-stack-mixed-mode DHCP parameter enables dual-stack operation with mixed-mode support for Dynamic Domain Name System (DDNS) updates. This parameter is relevant in environments where both IPv4 and IPv6 addresses are utilized. When enabled, it allows DHCP servers to handle DDNS updates for both IPv4 and IPv6 addresses simultaneously, ensuring compatibility and efficient management in dual-stack network configurations.	flag	FALSE	TRUE	flag	
94	ddns-guard-id-must-match	The ddns-guard-id-must-match DHCP parameter is used to enforce consistency between the Guard ID in the DNS Update message and the client's Client Identifier (DUID-LLT or DUID-EN). This ensures that the Guard ID, a security feature in Dynamic DNS (DDNS), matches the Client Identifier, enhancing security by preventing unauthorized updates. When set, this parameter mandates that the Guard ID and Client Identifier must align, providing an additional layer of validation in DDNS updates.	flag	FALSE	TRUE	flag	
95	ddns-other-guard-is-dynamic	The ddns-other-guard-is-dynamic DHCP parameter is used to guard against dynamic DNS (DDNS) updates from non-ISC DHCP servers. When enabled, this parameter ensures that only updates from ISC DHCP servers are accepted, preventing potential issues that might arise from updates initiated by non-ISC DHCP servers. It adds a layer of security and control to the dynamic DNS update process within a DHCP environment, promoting consistency and reliability in DNS records.	flag	FALSE	TRUE	flag	
96	release-on-roam	deprecated	flag	FALSE	TRUE	flag	

97	local-address6	The local-address6 DHCP parameter allows the specification of the IPv6 address that the DHCPv6 server should use as its source address when communicating with clients. By setting this parameter, administrators can control the server's outgoing IPv6 address, influencing communication in DHCPv6 processes. It provides flexibility in network configurations and ensures proper addressing for DHCPv6 interactions.	ipv6address	FALSE	TRUE	ip6-address	
98	bind-local-address6	The bind-local-address6 DHCP parameter is utilized to specify the local IPv6 address that the DHCPv6 server binds to. This parameter allows network administrators to control the IPv6 address on which the DHCPv6 server listens for incoming requests. By setting the bind-local-address6 parameter in the DHCPv6 server configuration, administrators can influence the server's behavior in handling IPv6 communication.	flag	FALSE	TRUE	flag	
99	ping-cltt-secs	Specifies the time, in seconds, that a client must remain silent before the DHCP server considers the client as unreachable or "dead." This parameter is part of the DHCP protocol's mechanism for checking the liveness of DHCP clients. If a client fails to communicate within the defined duration (specified by ping-cltt-secs), the DHCP server may assume that the client is no longer reachable and may release the associated lease. This helps manage and reclaim IP addresses in cases where clients become unresponsive or disconnected from the network.	uint32	FALSE	TRUE	seconds	
100	ping-timeout-ms	If the DHCP server determined it should send an ICMP echo request (a ping) because the ping-check statement is true, ping-timeout allows you to configure how many seconds the DHCP server should wait for an ICMP Echo response to be heard, if no ICMP Echo response has been received before the timeout expires, it assigns the address. If a response is heard, the lease is abandoned, and the server does not respond to the client. If no value is set, ping-timeout defaults to 1 second.	uint32	FALSE	TRUE	milliseconds	

Option Code	Option Name	Option Description	Data Type	Supported	Is predefined?	Grammar
1	agent.circuit-id	The circuit-id suboption encodes an agent-local identifier of the circuit from which a DHCP client-to-server packet was received. It is intended for use by agents in relaying DHCP responses back to the proper circuit. The format of this option is currently defined to be vendor-dependent, and will probably remain that way, although the current draft allows for the possibility of standardizing the format in the future.	string	TRUE	TRUE	string
2	remote-id	The remote-id suboption encodes information about the remote host end of a circuit. Examples of what it might contain include caller ID information, username information, remote ATM address, cable modem ID, and similar things. In principal, the meaning is not well-specified, and it should generally be assumed to be an opaque object that is administratively guaranteed to be unique to a particular remote end of a circuit.	quoted_string	TRUE	TRUE	string
3	agent-id	The "Agent ID" sub-option is used to convey information about the relay agent that forwarded the DHCP message. It helps in identifying the relay agent and its characteristics in the DHCP communication. Here's a basic structure for the "Agent ID" sub-option within the "Relay Agent Information" option: Code: Specific code assigned to the "Agent ID" sub-option (e.g., 1). Length: Length of the sub-option data field. Agent ID Data: Information about the relay agent. The "Agent ID" can contain various details about the relay agent, such as its hardware type, hardware address, or other identifying information.	string	TRUE	TRUE	string
19	relay-port	The "relay-port" suboption is particularly useful for the DHCP server to identify the specific UDP port number used by the relay agent when forwarding DHCP messages.	string	TRUE	TRUE	port

4	DOCSIS-device-class	The DOCSIS-device-class suboption is intended to convey information about the host endpoint, hardware, and software, that either the host operating system or the DHCP server may not otherwise be aware of (but the relay is able to distinguish). This is implemented as a 32-bit field (4 octets), each bit representing a flag describing the host in one of these ways. So far, only bit zero (being the least significant bit) is defined in RFC3256. If this bit is set to one, the host is considered a CPE Controlled Cable Modem (CCCM). All other bits are reserved.	uint32	TRUE	TRUE	uint32
5	link-selection	The link-selection suboption is provided by relay agents to inform servers what subnet the client is actually attached to. This is useful in those cases where the giaddr (where responses must be sent to the relay agent) is not on the same subnet as the client. When this option is present in a packet from a relay agent, the DHCP server will use its contents to find a subnet declared in configuration, and from here take one step further backwards to any shared-network the subnet may be defined within; the client may be given any address within that shared network, as normally appropriate.	ipv4address	TRUE	TRUE	ip-address

Option Code	Option name	Description	Data Type	Supported	Is Predefined?	Grammar	Example
1	no-client-update	When the client sends this, if it is true, it means the client will not attempt to update its A record. When sent by the server to the client, it means that the client should not update its own A record.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
2	server-update	When the client sends this to the server, it is requesting that the server update its A record. When sent by the server, it means that the server has updated (or is about to update) the client's A record.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
3	encoded	If true, this indicates that the domain name included in the option is encoded in DNS wire format, rather than as plain ASCII text. The client normally sets this to false if it doesn't support DNS wire format in the FQDN option. The server should always send back the same value that the client sent. When this value is set on the configuration side, it controls the format in which the fqdn.fqdn suboption is encoded.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
4, 5	rcode1, rcode2	These options specify the result of the updates of the A and PTR records, respectively, and are only sent by the DHCP server to the DHCP client. The values of these fields are those defined in the DNS protocol specification.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
6	fqdn	Specifies the domain name that the client wishes to use. This can be a fully-qualified domain name, or a single label. If there is no trailing ´.´ character in the name, it is not fully-qualified, and the server will generally update that name in some locally-defined domain.	quoted_string	TRUE	TRUE	text	Type : fqdn
7	hostname	This option should never be set, but it can be read back using the option and config-option operators in an expression, in which case it returns the first label in the fqdn.fqdn suboption - for example, if the value of fqdn.fqdn is "foo.example.com.", then fqdn.hostname will be "foo".	flag	TRUE	TRUE	--never set--	values : true or false or on or off Example : true
8	domainname	This option should never be set, but it can be read back using the option and config-option operators in an expression, in which case it returns all labels after the first label in the fqdn.fqdn suboption - for example, if the value of fqdn.fqdn is "foo.example.com.", then fqdn.domainname will be "example.com.". If this suboption value is not set, it means that an unqualified name was sent in the fqdn option, or that no fqdn option was sent at all.	flag	TRUE	TRUE	--never set--	values : true or false or on or off Example : true

Option Code	Option name	Description	Data Type	Supported	Is Predefined?	Grammar	Example
1	nsq-broadcast	If true, the client should use the NetWare Nearest Server Query to locate a NetWare/IP server. The behaviour of the Novell client if this suboption is false, or is not present, is not specified.	flag	TRUE	TRUE	flag	values : true or false or on or off Example : true
2	preferred-dss	This suboption specifies a list of up to five IP addresses, each of which should be the IP address of a NetWare Domain SAP/RIP server (DSS).	ipv4address	TRUE	TRUE	ip-address [, ip-addr	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
3	nearest-nwip-server	This suboption specifies a list of up to five IP addresses, each of which should be the IP address of a Nearest NetWare IP server.	ipv4address	TRUE	TRUE	ip-address [, ip-addr	Type : ipv4 Example : 1.1.1.1,2.2.2.2,..,etc
4	autoretries	Specifies the number of times that a NetWare/IP client should attempt to communicate with a given DSS server at startup.	uint8	TRUE	TRUE	uint8	Range : 0-255 Example : 123
5	autoretry-secs	Specifies the number of seconds that a Netware/IP client should wait between retries when attempting to establish communications with a DSS server at startup.	uint8	TRUE	TRUE	uint8	Range : 0-255 Example : 123
6	nwip-1-1	If true, the NetWare/IP client should support NetWare/IP version 1.1 compatibility. This is only needed if the client will be contacting Netware/IP version 1.1 servers.	uint8	TRUE	TRUE	uint8	Range : 0-255 Example : 123
7	primary-dss	Specifies the IP address of the Primary Domain SAP/RIP Service server (DSS) for this NetWare/IP domain. The NetWare/IP administration utility uses this value as Primary DSS server when configuring a secondary DSS server.	ipv4address	TRUE	TRUE	ip-address	Type : ipv4 Example : 1.1.1.1