Implementation of a Linux-based Domain Controller with Active Directory and Zabbix Monitoring

Executive Summary

This project focuses on the implementation of a Linux-based Domain Controller combined with a centralized monitoring system. The domain controller was built using Samba Active Directory on Ubuntu 20.04, while Zabbix 6.0 LTS was deployed for monitoring purposes backed by PostgreSQL 14 as the database server. A Fedora 39 client machine was successfully joined to the Samba domain and configured for real-time monitoring through the Zabbix server.

The key objectives of this project were to:

- Establish a fully functional Active Directory domain environment.
- Set up a stable and efficient Zabbix monitoring solution.
- Integrate both the domain controller and client machine into Zabbix monitoring.
- Ensure secure and reliable communication between all components.

During the implementation, Zabbix Agent (classic) was installed and configured on both the server and client machines, aligning with Zabbix best practices. Templates such as "Template OS Linux" were applied for detailed system monitoring, including system services and database performance.

The setup was thoroughly tested to ensure user authentication through the domain controller, proper system resource tracking, and health monitoring via Zabbix dashboards. Through this project, a scalable and maintainable monitoring infrastructure was successfully achieved, demonstrating the effective use of open-source tools in a professional network environment

Table of Contents

01. Introduction

In modern IT infrastructures, centralised authentication and efficient monitoring play crucial roles in maintaining operational reliability and security. This project focuses on the implementation of a Linux-based Active Directory Domain Controller using Samba, combined with the deployment of a comprehensive monitoring solution through Zabbix.

The Domain Controller is responsible for authenticating and authorizing users and devices within the network, ensuring centralised management and enhanced security. Simultaneously, Zabbix provides a robust platform for real-time monitoring of system performance, availability and health across both servers and client machines.

This assignment specifically involves setting up a Samba Active Directory Domain Controller on Ubuntu 20.04, integrating a PostgreSQL-backed Zabbix 6.0 LTS monitoring server and configuring Zabbix agents on both the domain controller and a Fedora 39 client. The project demonstrates not only technical proficiency in system administration and network management but also highlights best practices in open-source system integration.

The following sections provide a detailed walkthrough of the system setup, configuration, and validation processes undertaken to achieve a fully operational monitoring environment for the implemented domain infrastructure

02. System Requirements and Environment

Server Machine (Ubuntu 20.04 LTS)

- **Operating System:** Ubuntu 20.04 LTS (Focal Fossa) – 64-bit
- **CPU:** Minimum 2 vCPUs (Used: 4 vCPUs)
- **Memory:** 3 GB RAM
- **Storage:** 20 GB available disk space
- **Network:** Static IP configuration with internet access
- **Software Installed:**

Samba (for Active Directory Domain Controller)

Bind9 (optional for internal DNS services)

PostgreSQL 14 (Database server for Zabbix)

Zabbix Server 6.0 LTS

Zabbix Frontend (Apache, PHP 8.1)

Zabbix Agent (for server monitoring)

Client Machine (Fedora 39)

- **Operating System:** Fedora Linux 39 (Workstation Edition) – 64-bit
- **CPU:** 4 vCPUs
- **Memory:** 6 GB RAM
- **Storage:** Minimum 20 GB available disk space
- **Network:** Static IP
- **Software Installed:**

SSSD (for domain join and authentication)

Zabbix Agent (for client monitoring)

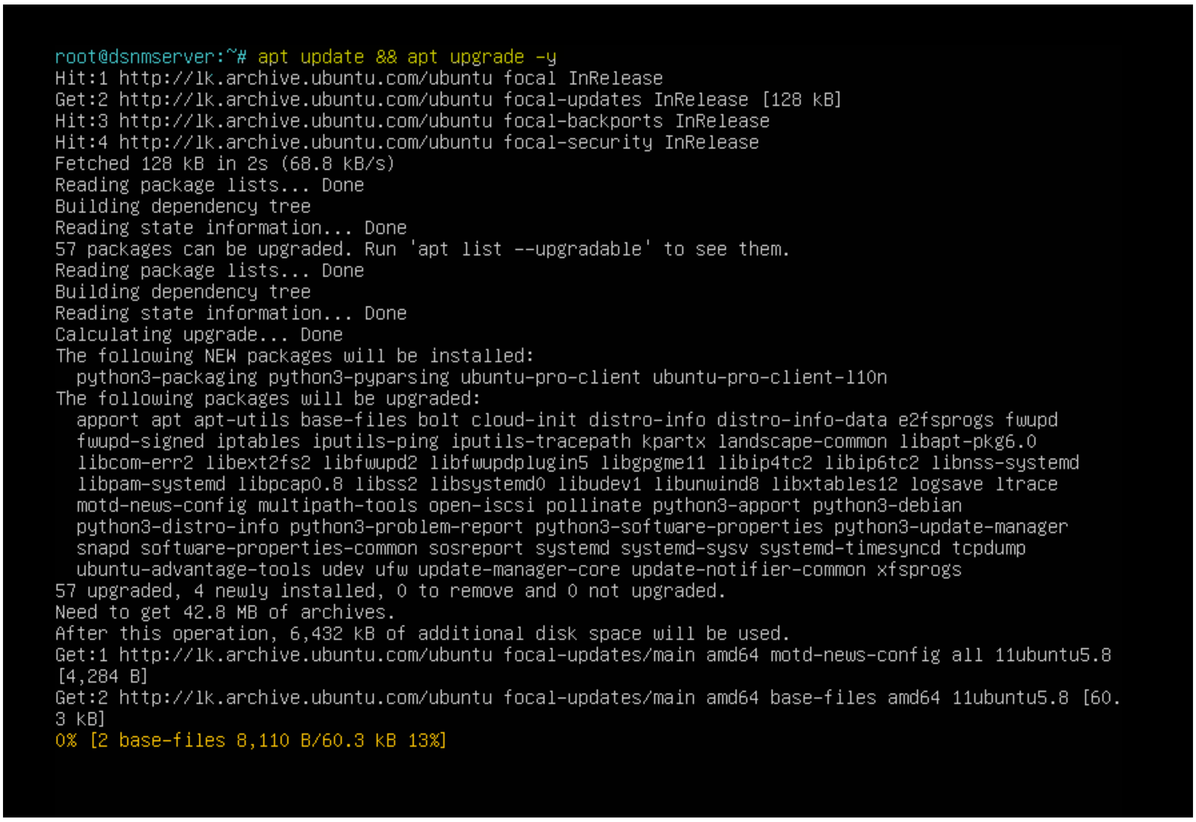
03. Implementation Details

3.1. Setting up Samba Active Directory Domain Controller

01. **Update System Packages**

Ensured all system packages were up to date using:

```
sudo apt update && sudo apt upgrade -y
```

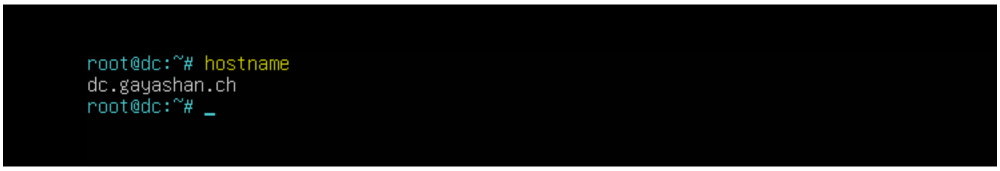


```
root@dsnmserver:~# apt update && apt upgrade -y
Hit:1 http://lk.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://lk.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Hit:3 http://lk.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://lk.archive.ubuntu.com/ubuntu focal-security InRelease
Fetched 128 kB in 2s (68.8 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
57 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  python3-packaging python3-pyparsing ubuntu-pro-client ubuntu-pro-client-l10n
The following packages will be upgraded:
  apport apt apt-utils base-files bolt cloud-init distro-info distro-info-data e2fsprogs fwupd
  fwupd-signed iptables iputils-ping iputils-tracepath kpartx landscape-common libapt-pkg6.0
  libcom-err2 libext2fs2 libfwupd2 libfwupdplugin5 libgpgme11 libip4tc2 libip6tc2 libnss-systemd
  libpam-systemd libpcap0.8 libss2 libsystemd0 libudev1 libunwind8 libxtables12 logsave ltrace
  motd-news-config multipath-tools open-iscsi pollinate python3-apport python3-debian
  python3-distro-info python3-problem-report python3-software-properties python3-update-manager
  snapd software-properties-common sosreport systemd systemd-sysv systemd-timesyncd tcpdump
  ubuntu-advantage-tools udev ufw update-manager-core update-notifier-common xfsprogs
57 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 42.8 MB of archives.
After this operation, 6,432 kB of additional disk space will be used.
Get:1 http://lk.archive.ubuntu.com/ubuntu focal-updates/main amd64 motd-news-config all 11ubuntu5.8
[4,284 B]
Get:2 http://lk.archive.ubuntu.com/ubuntu focal-updates/main amd64 base-files amd64 11ubuntu5.8 [60.
3 kB]
0% [2 base-files 8,110 B/60.3 kB 13%]
```

02. Set Hostname

Configured the server’s hostname to match the domain controller's FQDN:

```
sudo hostnamectl set-hostname dc.gayashan.ch
```

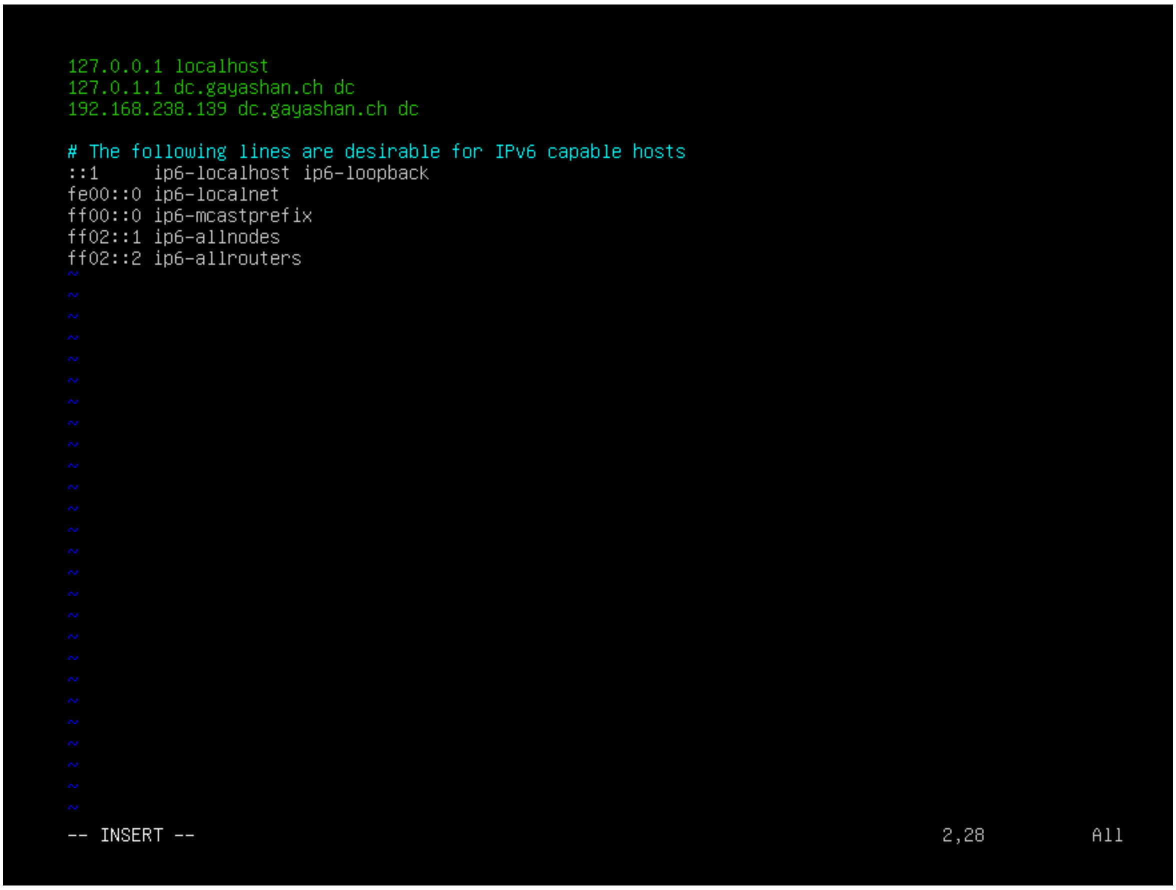


```
root@dc:~# hostname
dc.gayashan.ch
root@dc:~# _
```

03. Edit /etc/hosts File

Mapped the static IP and FQDN to 127.0.0.1 and the system's IP address.

```
vi /etc/hosts
```

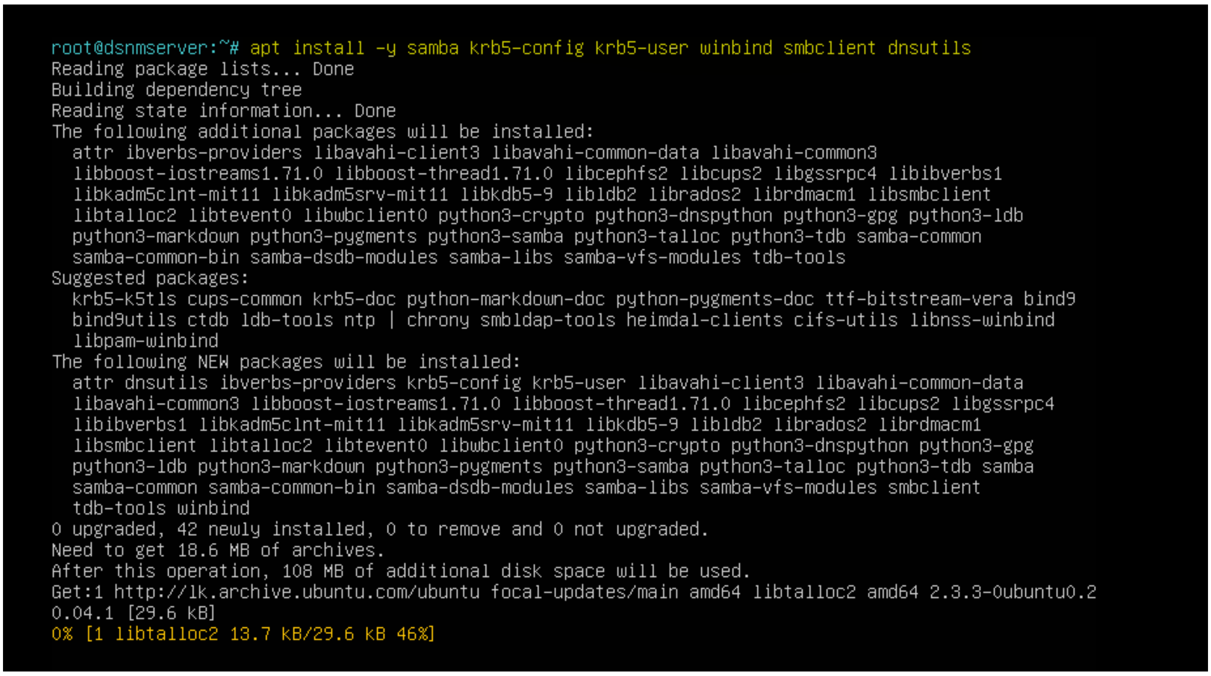


```
127.0.0.1 localhost
127.0.1.1 dc.gayashan.ch dc
192.168.238.139 dc.gayashan.ch dc

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                        2,28          All
```

04. Install Required Packages

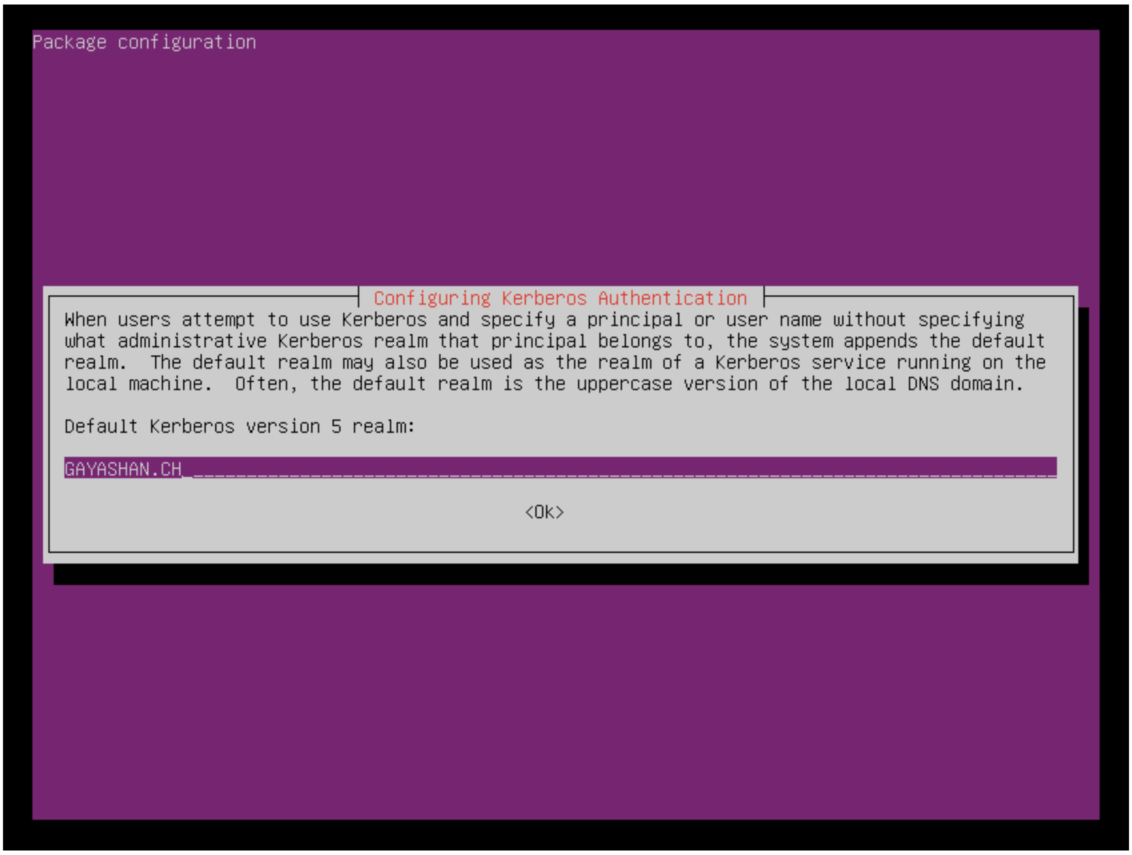
Installed Samba and related utilities:

```
sudo apt install -y samba krb5-config krb5-user winbind smbclient dnsutils
```



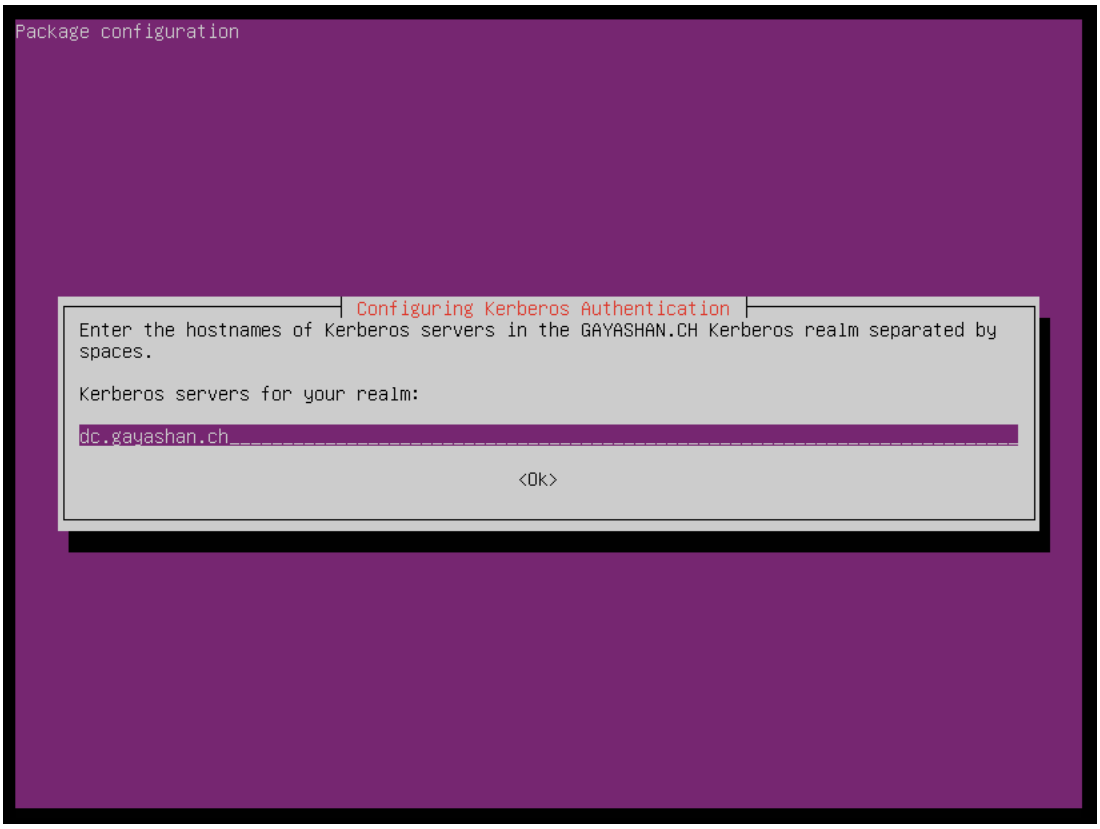
```
root@dsnmserver:~# apt install -y samba krb5-config krb5-user winbind smbclient dnsutils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  attr ibverbs-providers libavahi-client3 libavahi-common-data libavahi-common3
  libboost-iostreams1.71.0 libboost-thread1.71.0 libcephfs2 libcups2 libgssrpc4 libibverbs1
  libkadm5clnt-mit11 libkadm5srv-mit11 libkdb5-9 libldb2 librados2 librdmacm1 libsmbclient
  libtalloc2 libtevent0 libwbclient0 python3-crypto python3-dnspython python3-gpg python3-ldb
  python3-markdown python3-pygments python3-samba python3-talloc python3-tdb samba-common
  samba-common-bin samba-dsdb-modules samba-libs samba-vfs-modules tdb-tools
Suggested packages:
  krb5-k5tls cups-common krb5-doc python-markdown-doc python-pygments-doc ttf-bitstream-vera bind9
  bind9utils ctdb ldb-tools ntp | chrony smbldap-tools heimdal-clients cifs-utils libnss-winbind
  libpam-winbind
The following NEW packages will be installed:
  attr dnsutils ibverbs-providers krb5-config krb5-user libavahi-client3 libavahi-common-data
  libavahi-common3 libboost-iostreams1.71.0 libboost-thread1.71.0 libcephfs2 libcups2 libgssrpc4
  libibverbs1 libkadm5clnt-mit11 libkadm5srv-mit11 libkdb5-9 libldb2 librados2 librdmacm1
  libsmbclient libtalloc2 libtevent0 libwbclient0 python3-crypto python3-dnspython python3-gpg
  python3-ldb python3-markdown python3-pygments python3-samba python3-talloc python3-tdb samba
  samba-common samba-common-bin samba-dsdb-modules samba-libs samba-vfs-modules smbclient
  tdb-tools winbind
0 upgraded, 42 newly installed, 0 to remove and 0 not upgraded.
Need to get 18.6 MB of archives.
After this operation, 108 MB of additional disk space will be used.
Get:1 http://lk.archive.ubuntu.com/ubuntu focal-updates/main amd64 libtalloc2 amd64 2.3.3-0ubuntu0.2
0.04.1 [29.6 kB]
0% [1 libtalloc2 13.7 kB/29.6 kB 46%]
```

Kerberos Realm



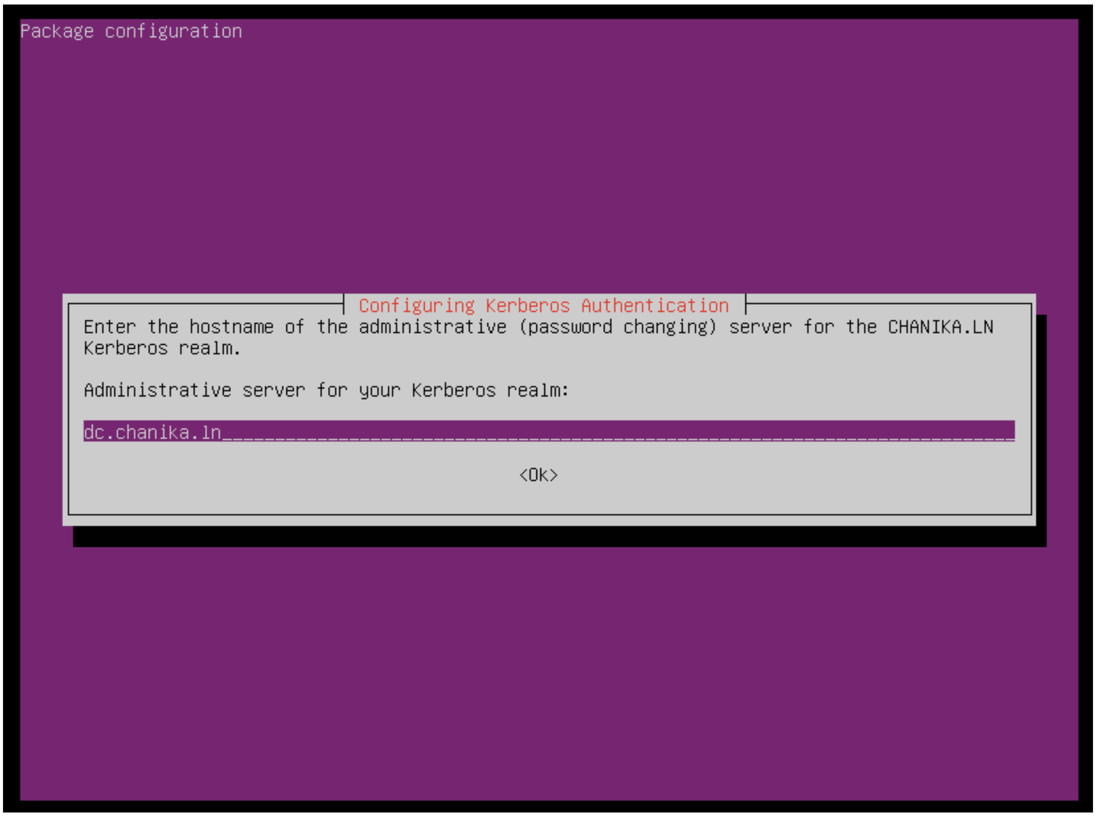
```
Package configuration

Configuring Kerberos Authentication

When users attempt to use Kerberos and specify a principal or user name without specifying
what administrative Kerberos realm that principal belongs to, the system appends the default
realm.  The default realm may also be used as the realm of a Kerberos service running on the
local machine.  Often, the default realm is the uppercase version of the local DNS domain.

Default Kerberos version 5 realm:

GAYASHAN.CH

<Ok>
```

Kerberos Server



```
Package configuration

            ┤ Configuring Kerberos Authentication ├
 Enter the hostnames of Kerberos servers in the GAYASHAN.CH Kerberos realm separated by
 spaces.

 Kerberos servers for your realm:

 dc.gayashan.ch_________________________________________________________________________

                                      <Ok>
```

Admin Sever



```
Package configuration

            ┤ Configuring Kerberos Authentication ├
 Enter the hostname of the administrative (password changing) server for the CHANIKA.LN
 Kerberos realm.

 Administrative server for your Kerberos realm:

 dc.chanika.ln__________________________________________________________________________

                                      <Ok>
```

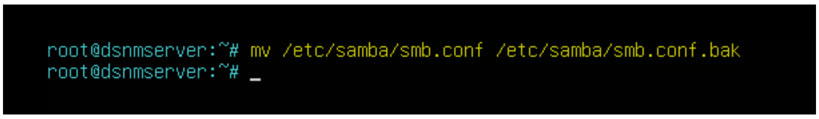
In the above screenshot, the admin server is named as “`dc.chanika.ln`” but the correct server name should be “`dc.gayashan.ch`”

During the installation of Kerberos packages (`krb5-config, krb5-user`), the system prompted for Kerberos realm settings. These inputs are required for Active Directory integration and were automatically written to `/etc/krb5.conf`.

05. Provision the Domain Controller

Move the default config aside to avoid conflicts:

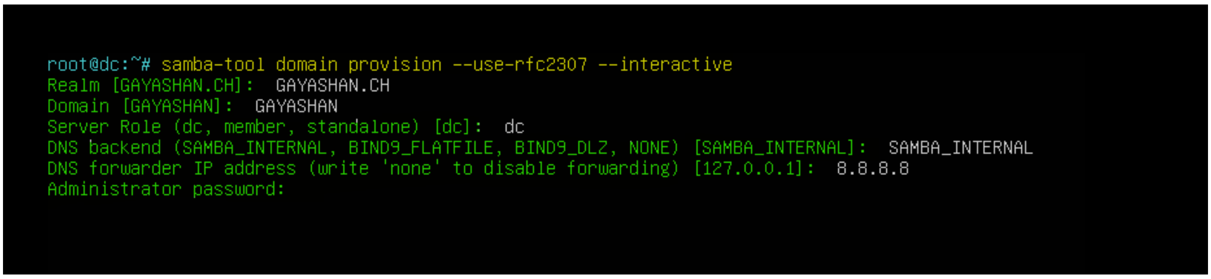
```
sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.bak
```



```
root@dsnmserver:~# mv /etc/samba/smb.conf /etc/samba/smb.conf.bak
root@dsnmserver:~# _
```

Used samba-tool to provision the domain:

```
sudo samba-tool domain provision --use-rfc2307 –interactive
```



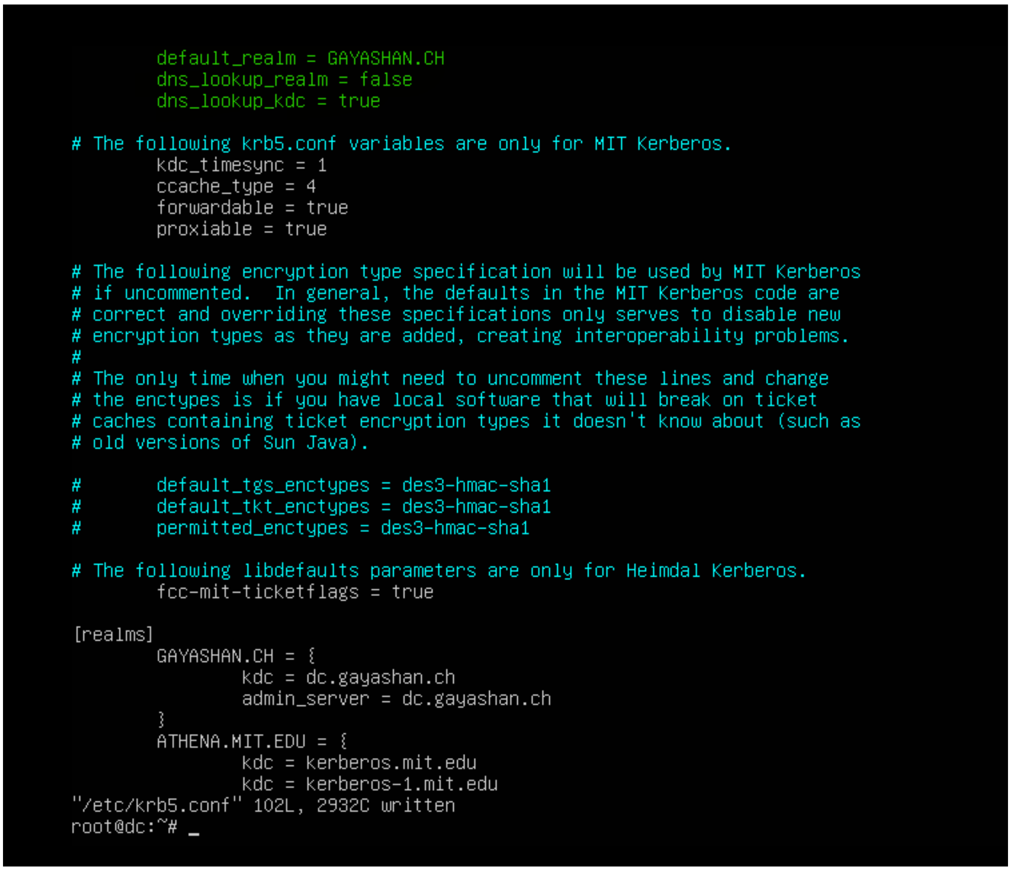
```
root@dc:~# samba-tool domain provision --use-rfc2307 --interactive
Realm [GAYASHAN.CH]:  GAYASHAN.CH
Domain [GAYASHAN]:  GAYASHAN
Server Role (dc, member, standalone) [dc]:  dc
DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE) [SAMBA_INTERNAL]:  SAMBA_INTERNAL
DNS forwarder IP address (write 'none' to disable forwarding) [127.0.0.1]:  8.8.8.8
Administrator password:
```

Entered domain name, NetBIOS name, and administrative password.

06. Configure DNS and Kerberos

Updated `/etc/krb5.conf` to reflect the correct realm and domain settings.

```
[libdefaults]

    default_realm = GAYASHAN.CH

    dns_lookup_realm = false

    dns_lookup_kdc = true
```



```
        default_realm = GAYASHAN.CH
        dns_lookup_realm = false
        dns_lookup_kdc = true

# The following krb5.conf variables are only for MIT Kerberos.
        kdc_timesync = 1
        ccache_type = 4
        forwardable = true
        proxiable = true

# The following encryption type specification will be used by MIT Kerberos
# if uncommented.  In general, the defaults in the MIT Kerberos code are
# correct and overriding these specifications only serves to disable new
# encryption types as they are added, creating interoperability problems.
#
# The only time when you might need to uncomment these lines and change
# the enctypes is if you have local software that will break on ticket
# caches containing ticket encryption types it doesn't know about (such as
# old versions of Sun Java).

#       default_tgs_enctypes = des3-hmac-sha1
#       default_tkt_enctypes = des3-hmac-sha1
#       permitted_enctypes = des3-hmac-sha1

# The following libdefaults parameters are only for Heimdal Kerberos.
        fcc-mit-ticketflags = true

[realms]
        GAYASHAN.CH = {
                kdc = dc.gayashan.ch
                admin_server = dc.gayashan.ch
        }
        ATHENA.MIT.EDU = {
                kdc = kerberos.mit.edu
                kdc = kerberos-1.mit.edu
"/etc/krb5.conf" 102L, 2932C written
root@dc:~# _
```

07. Start Samba Services

```
sudo systemctl stop smbd nmbd winbind
```

- These are the traditional Samba services used for standalone file sharing and authentication:
  - `smbd` – handles SMB/CIFS file sharing.
  - `nmbd` – handles NetBIOS name service.
  - `winbind` – used for Windows authentication.
- In an AD DC setup, we don't need these services separately, because:
  - samba-ad-dc handles everything (SMB, LDAP, Kerberos, DNS) under one unified service.
  - Leaving smbd/nmbd/winbind running can conflict with samba-ad-dc.

  So, stop them to avoid conflicts.

`sudo systemctl unmask samba-ad-dc`

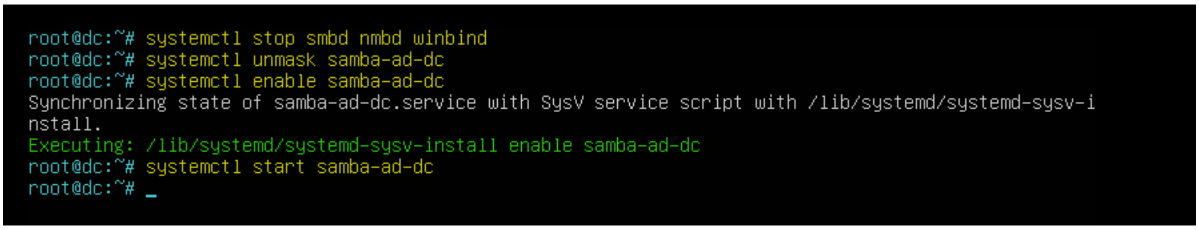
Sometimes, the `samba-ad-dc service` might be masked (disabled at the system level) by default to prevent accidental use. A masked service can’t be started or enabled. This command unblocks the AD DC service so we can enable/start it.

`sudo systemctl enable samba-ad-dc`

Ensures that samba-ad-dc starts automatically every time the server reboots. A crucial step for keeping the domain controller always online. Enables automatic startup on boot.

`sudo systemctl start samba-ad-dc`

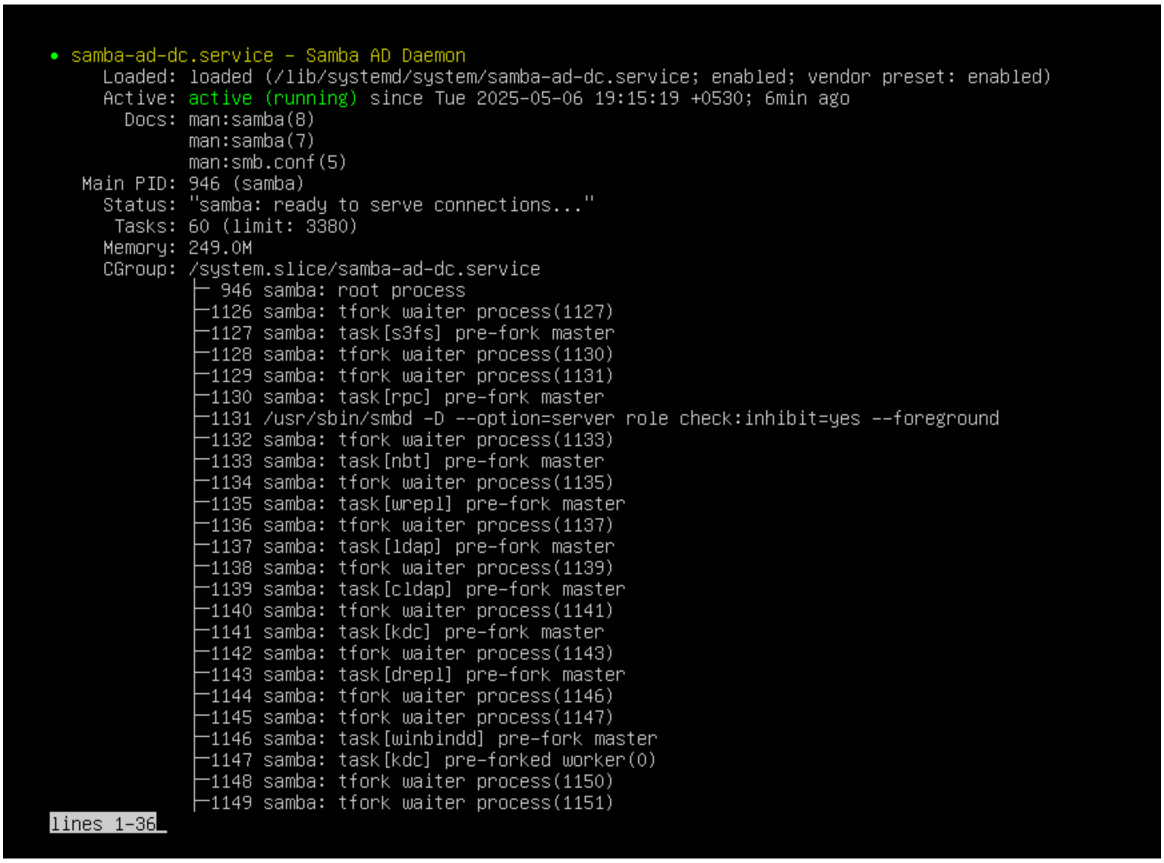
Starts the Samba AD DC service immediately. This brings the domain controller online, allowing domain authentication, DNS, LDAP, etc.



```
root@dc:~# systemctl stop smbd nmbd winbind
root@dc:~# systemctl unmask samba-ad-dc
root@dc:~# systemctl enable samba-ad-dc
Synchronizing state of samba-ad-dc.service with SysV service script with /lib/systemd/systemd-sysv-i
nstall.
Executing: /lib/systemd/systemd-sysv-install enable samba-ad-dc
root@dc:~# systemctl start samba-ad-dc
root@dc:~# _
```

`sudo systemctl status samba-ad-dc`

Run this to confirm the AD DC service is up and healthy.

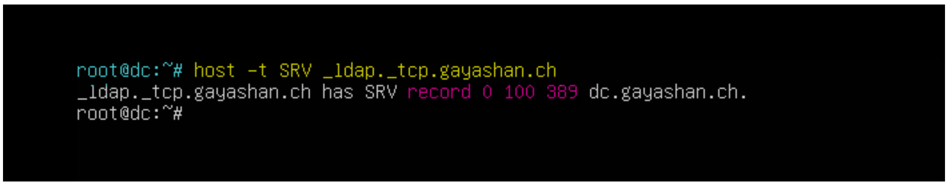


```
● samba-ad-dc.service - Samba AD Daemon
     Loaded: loaded (/lib/systemd/system/samba-ad-dc.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2025-05-06 19:15:19 +0530; 6min ago
       Docs: man:samba(8)
             man:samba(7)
             man:smb.conf(5)
   Main PID: 946 (samba)
     Status: "samba: ready to serve connections..."
      Tasks: 60 (limit: 3380)
     Memory: 249.0M
     CGroup: /system.slice/samba-ad-dc.service
             ├─ 946 samba: root process
             ├─1126 samba: tfork waiter process(1127)
             ├─1127 samba: task[s3fs] pre-fork master
             ├─1128 samba: tfork waiter process(1130)
             ├─1129 samba: tfork waiter process(1131)
             ├─1130 samba: task[rpc] pre-fork master
             ├─1131 /usr/sbin/smbd -D --option=server role check:inhibit=yes --foreground
             ├─1132 samba: tfork waiter process(1133)
             ├─1133 samba: task[nbt] pre-fork master
             ├─1134 samba: tfork waiter process(1135)
             ├─1135 samba: task[wrepl] pre-fork master
             ├─1136 samba: tfork waiter process(1137)
             ├─1137 samba: task[ldap] pre-fork master
             ├─1138 samba: tfork waiter process(1139)
             ├─1139 samba: task[cldap] pre-fork master
             ├─1140 samba: tfork waiter process(1141)
             ├─1141 samba: task[kdc] pre-fork master
             ├─1142 samba: tfork waiter process(1143)
             ├─1143 samba: task[drepl] pre-fork master
             ├─1144 samba: tfork waiter process(1146)
             ├─1145 samba: tfork waiter process(1147)
             ├─1146 samba: task[winbindd] pre-fork master
             ├─1147 samba: task[kdc] pre-forked worker(0)
             ├─1148 samba: tfork waiter process(1150)
             ├─1149 samba: tfork waiter process(1151)
lines 1-36
```

08. Verifying Active Directory DNS and Authentication

To ensure the proper functionality of the Samba-based Domain Controller, DNS resolution and Kerberos authentication must be verified. These steps confirm that the essential AD services are correctly advertised and accessible within the network.

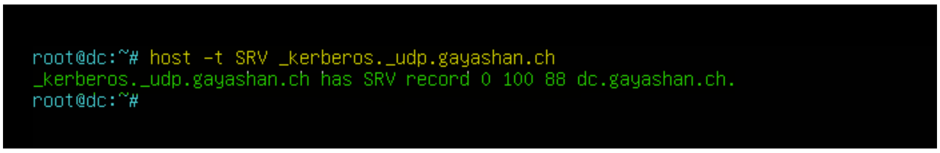
```
host -t SRV _ldap._tcp.gayashan.ch
```



```
root@dc:~# host -t SRV _ldap._tcp.gayashan.ch
_ldap._tcp.gayashan.ch has SRV record 0 100 389 dc.gayashan.ch.
root@dc:~#
```

These commands check if the DNS SRV records for LDAP and Kerberos services are properly registered. A successful response confirms that the Domain Controller is discoverable by clients attempting to use AD services.

```
host -t SRV _kerberos._udp.gayashan.ch
```

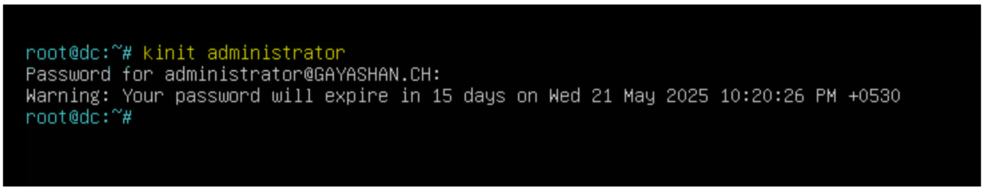


```
root@dc:~# host -t SRV _kerberos._udp.gayashan.ch
_kerberos._udp.gayashan.ch has SRV record 0 100 88 dc.gayashan.ch.
root@dc:~#
```

This Response indicates that DNS is properly pointing Kerberos authentication traffic (UDP port 88) to the domain controller dc.gayashan.ch, which is crucial for AD-integrated authentication.

Kerberos Authentication Test

```
kinit administrator
```



```
root@dc:~# kinit administrator
Password for administrator@GAYASHAN.CH:
Warning: Your password will expire in 15 days on Wed 21 May 2025 10:20:26 PM +0530
root@dc:~#
```

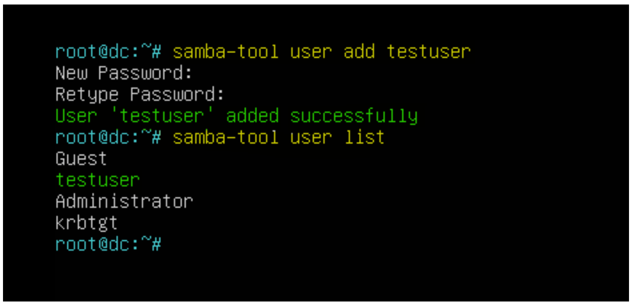
This command initializes a Kerberos ticket for the administrator account. It validates whether the Kerberos Key Distribution Center (KDC) is functioning and if user credentials are being handled correctly. A successful login with no errors confirms that authentication is working as expected.

Test Samba authentication by creating a test user

```
sudo samba-tool user add testuser
```

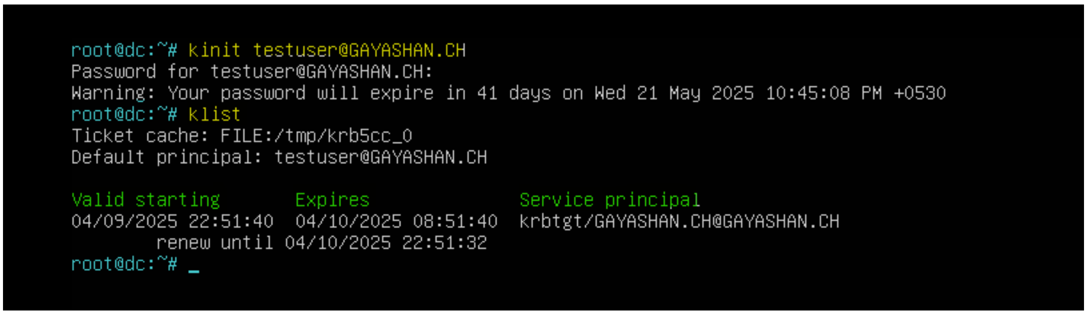
Then check if it exists:

```
sudo samba-tool user list
```



```
root@dc:~# samba-tool user add testuser
New Password:
Retype Password:
User 'testuser' added successfully
root@dc:~# samba-tool user list
Guest
testuser
Administrator
krbtgt
root@dc:~#
```

Test by authenticating with the test user



```
root@dc:~# kinit testuser@GAYASHAN.CH
Password for testuser@GAYASHAN.CH:
Warning: Your password will expire in 41 days on Wed 21 May 2025 10:45:08 PM +0530
root@dc:~# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: testuser@GAYASHAN.CH

Valid starting       Expires              Service principal
04/09/2025 22:51:40  04/10/2025 08:51:40  krbtgt/GAYASHAN.CH@GAYASHAN.CH
        renew until 04/10/2025 22:51:32
root@dc:~# _
```

This output confirms that a valid Kerberos ticket-granting ticket (TGT) has been issued for the domain user `testuser@GAYASHAN.CH`. It proves that the system has successfully authenticated against the domain controller and can now access domain services securely.

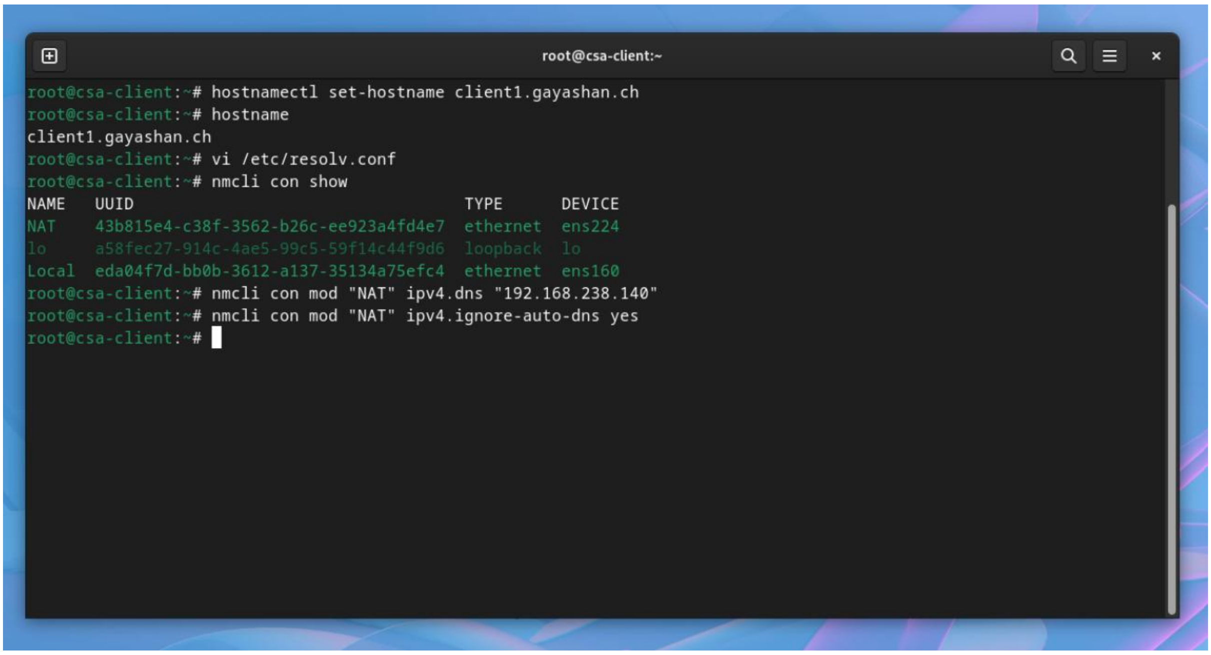
3.2 Joining Fedora Client to the Domain

To allow centralised authentication and management, the Fedora client was successfully joined to the Samba-based Active Directory domain gayashan.ch. Below are the key steps involved in the process:

01. Set the Hostname on the Fedora Client

To ensure proper identity resolution and avoid conflicts the client machine was assigned a unique hostname:

```
sudo hostnamectl set-hostname client1.gayashan.ch
```



```
root@csa-client:~# hostnamectl set-hostname client1.gayashan.ch
root@csa-client:~# hostname
client1.gayashan.ch
root@csa-client:~# vi /etc/resolv.conf
root@csa-client:~# nmcli con show
NAME   UUID                                  TYPE      DEVICE
NAT    43b815e4-c38f-3562-b26c-ee923a4fd4e7  ethernet  ens224
lo     a58fec27-914c-4ae5-99c5-59f14c44f9d6  loopback  lo
Local  eda04f7d-bb0b-3612-a137-35134a75efc4  ethernet  ens160
root@csa-client:~# nmcli con mod "NAT" ipv4.dns "192.168.238.140"
root@csa-client:~# nmcli con mod "NAT" ipv4.ignore-auto-dns yes
root@csa-client:~# 
```

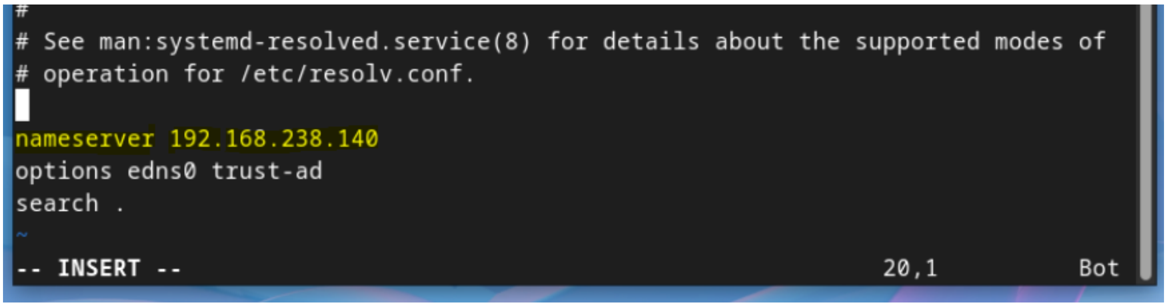
This helps maintain naming consistency within the domain network.

02. Point Fedora Client's DNS to the Domain Controller

Edit `/etc/resolv.conf` and make sure the DNS server is your Samba AD/DC server:

For domain discovery and authentication to work properly the client must use the domain controller as its DNS server

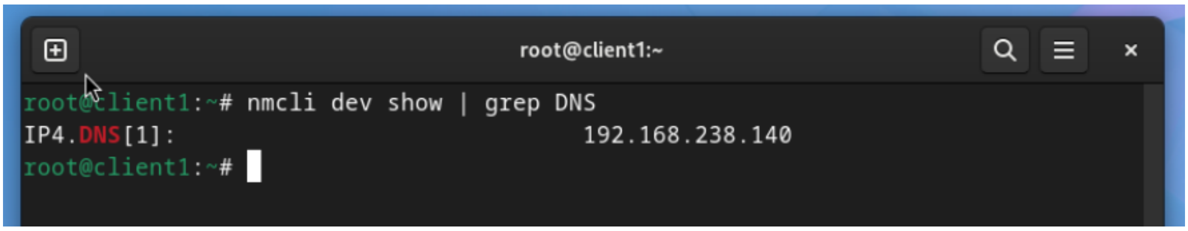
```
nameserver <IP-of-dc.gayashan.ch>
```



```
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 192.168.238.140
options edns0 trust-ad
search .
~
-- INSERT --                                                  20,1          Bot
```

Verify DNS

`nmcli dev show | grep DNS` - Check if the DNS is updated

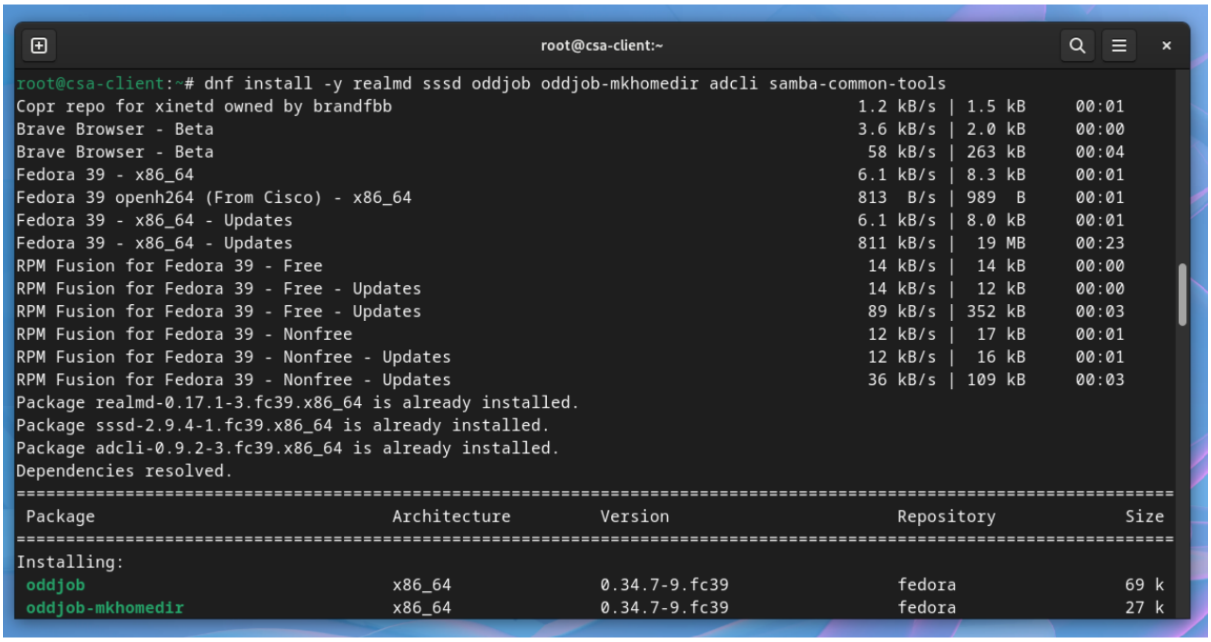


```
root@client1:~# nmcli dev show | grep DNS
IP4.DNS[1]:                             192.168.238.140
root@client1:~# 
```

03. Install Required Packages

The following packages are essential for realmd, SSSD-based authentication, and Kerberos integration:

```
sudo dnf install -y realmd sssd oddjob oddjob-mkhomedir adcli samba-common-tools
```



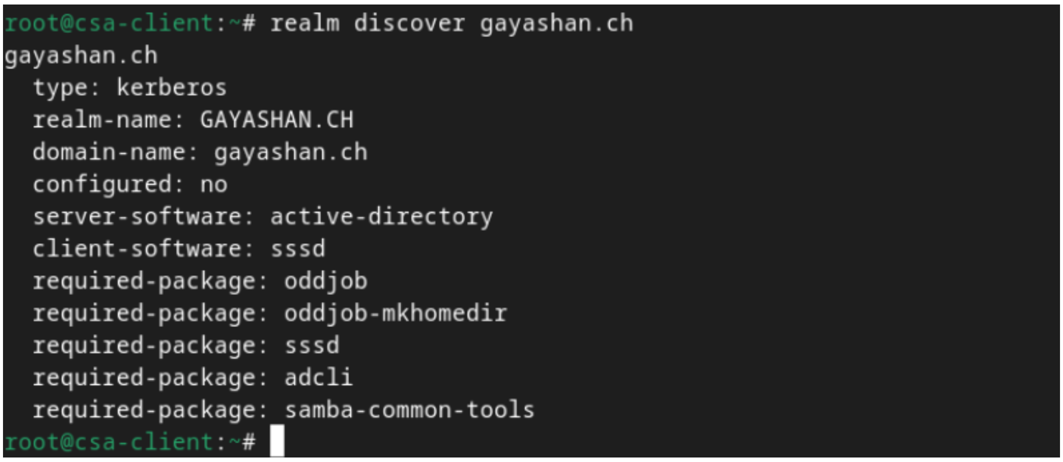
```
root@csa-client:~# dnf install -y realmd sssd oddjob oddjob-mkhomedir adcli samba-common-tools
Copr repo for xinetd owned by brandfbb                                  1.2 kB/s | 1.5 kB     00:01
Brave Browser - Beta                                                    3.6 kB/s | 2.0 kB     00:00
Brave Browser - Beta                                                     58 kB/s | 263 kB     00:04
Fedora 39 - x86_64                                                      6.1 kB/s | 8.3 kB     00:01
Fedora 39 openh264 (From Cisco) - x86_64                                813  B/s | 989  B     00:01
Fedora 39 - x86_64 - Updates                                            6.1 kB/s | 8.0 kB     00:01
Fedora 39 - x86_64 - Updates                                            811 kB/s |  19 MB     00:23
RPM Fusion for Fedora 39 - Free                                          14 kB/s |  14 kB     00:00
RPM Fusion for Fedora 39 - Free - Updates                                14 kB/s |  12 kB     00:00
RPM Fusion for Fedora 39 - Free - Updates                                89 kB/s | 352 kB     00:03
RPM Fusion for Fedora 39 - Nonfree                                       12 kB/s |  17 kB     00:01
RPM Fusion for Fedora 39 - Nonfree - Updates                             12 kB/s |  16 kB     00:01
RPM Fusion for Fedora 39 - Nonfree - Updates                             36 kB/s | 109 kB     00:03
Package realmd-0.17.1-3.fc39.x86_64 is already installed.
Package sssd-2.9.4-1.fc39.x86_64 is already installed.
Package adcli-0.9.2-3.fc39.x86_64 is already installed.
Dependencies resolved.
======================================================================================================
 Package                             Architecture         Version                      Repository  Size
======================================================================================================
Installing:
 oddjob                              x86_64               0.34.7-9.fc39                fedora      69 k
 oddjob-mkhomedir                    x86_64               0.34.7-9.fc39                fedora      27 k
```

These tools handle domain discovery, authentication, and user/group mapping.

04. Discover the Domain

To verify that the domain is visible and compatible:

```
realm discover gayashan.ch
```



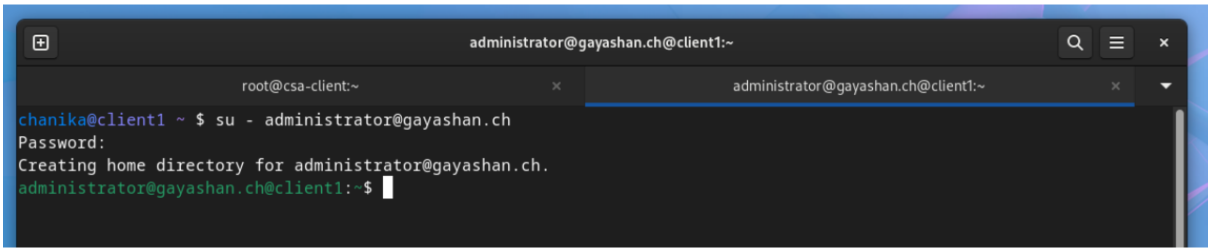
```
root@csa-client:~# realm discover gayashan.ch
gayashan.ch
  type: kerberos
  realm-name: GAYASHAN.CH
  domain-name: gayashan.ch
  configured: no
  server-software: active-directory
  client-software: sssd
  required-package: oddjob
  required-package: oddjob-mkhomedir
  required-package: sssd
  required-package: adcli
  required-package: samba-common-tools
root@csa-client:~#
```

If the output lists realm configuration, Kerberos settings, and login formats, the discovery was successful. This is a successful discovery

05. Join the Domain

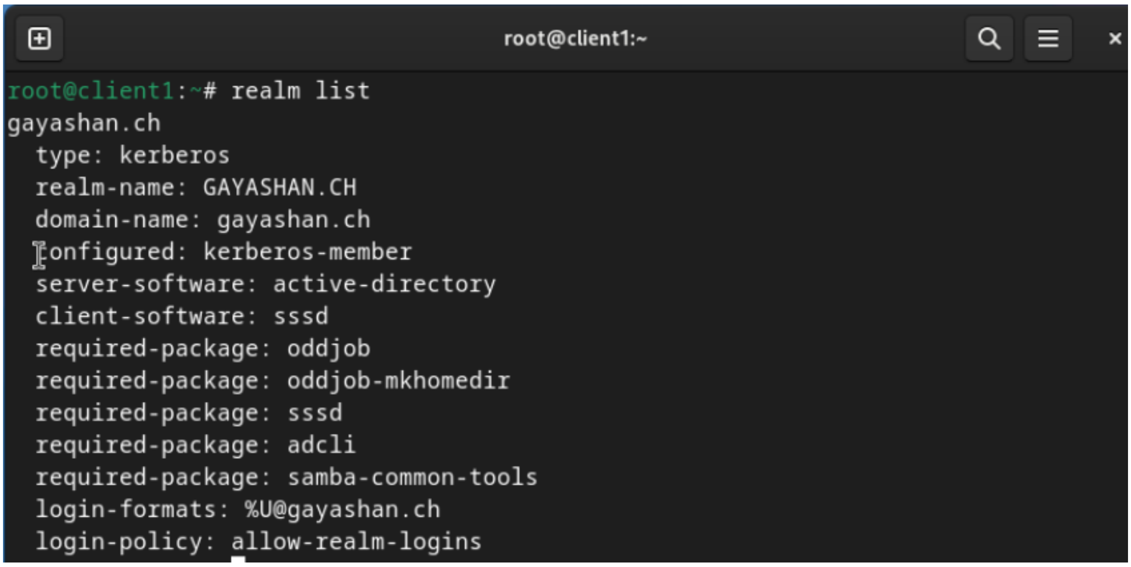
The client was then joined to the domain using an administrative account:

```
sudo realm join -U administrator gayashan.ch
```


administrator@gayashan.ch@client1:~
root@csa-client:~
administrator@gayashan.ch@client1:~

```
chanika@client1 ~ $ su - administrator@gayashan.ch
Password:
Creating home directory for administrator@gayashan.ch.
administrator@gayashan.ch@client1:~$
```

06. Verify Domain Join



```
root@client1:~# realm list
gayashan.ch
  type: kerberos
  realm-name: GAYASHAN.CH
  domain-name: gayashan.ch
  configured: kerberos-member
  server-software: active-directory
  client-software: sssd
  required-package: oddjob
  required-package: oddjob-mkhomedir
  required-package: sssd
  required-package: adcli
  required-package: samba-common-tools
  login-formats: %U@gayashan.ch
  login-policy: allow-realm-logins
```

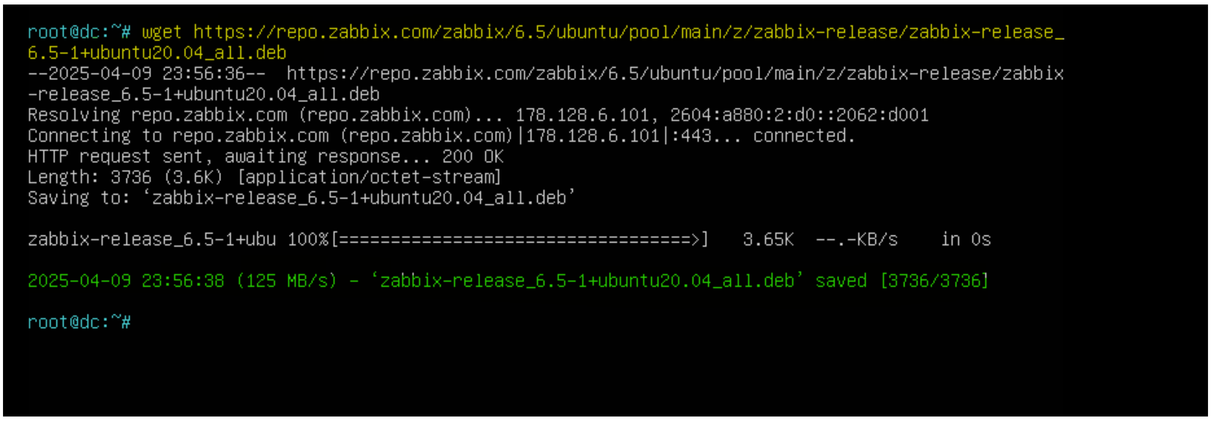
3.3 Installing and Configuring Zabbix Server

This section covers the complete installation and configuration process for setting up Zabbix 6.0 LTS on Ubuntu 20.04 with PostgreSQL 14 as the backend database.

01. Add the Official Zabbix Repository

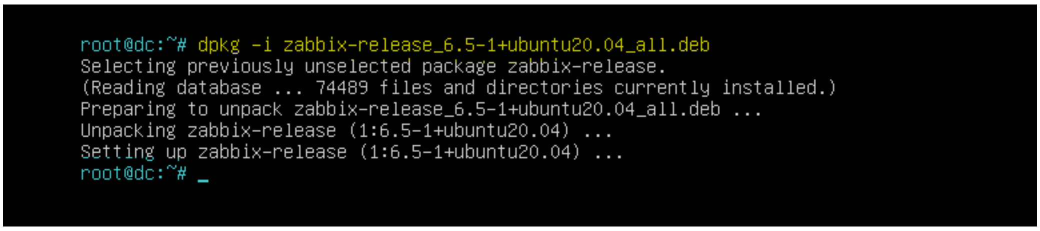
To ensure the correct and stable version (6.5 LTS), the official Zabbix repository for Ubuntu 20.04 was added:

```
wget https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.5-4+ubuntu20.04_all.deb
```



```
root@dc:~# wget https://repo.zabbix.com/zabbix/6.5/ubuntu/pool/main/z/zabbix-release/zabbix-release_
6.5-1+ubuntu20.04_all.deb
--2025-04-09 23:56:36--  https://repo.zabbix.com/zabbix/6.5/ubuntu/pool/main/z/zabbix-release/zabbix
-release_6.5-1+ubuntu20.04_all.deb
Resolving repo.zabbix.com (repo.zabbix.com)... 178.128.6.101, 2604:a880:2:d0::2062:d001
Connecting to repo.zabbix.com (repo.zabbix.com)|178.128.6.101|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3736 (3.6K) [application/octet-stream]
Saving to: ‘zabbix-release_6.5-1+ubuntu20.04_all.deb’

zabbix-release_6.5-1+ubu 100%[===================================>]   3.65K  --.-KB/s    in 0s

2025-04-09 23:56:38 (125 MB/s) - ‘zabbix-release_6.5-1+ubuntu20.04_all.deb’ saved [3736/3736]

root@dc:~#
```

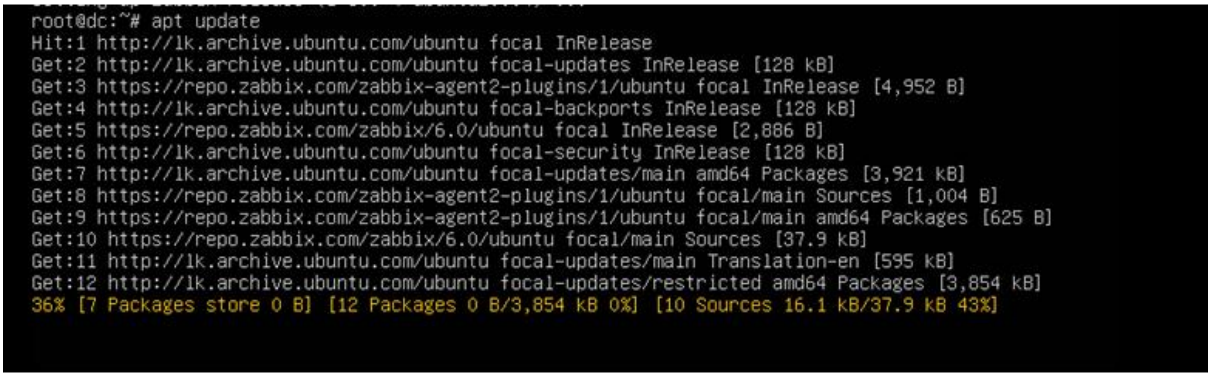
```
sudo dpkg -i zabbix-release_6.5-4+ubuntu20.04_all.deb
```



```
root@dc:~# dpkg -i zabbix-release_6.5-1+ubuntu20.04_all.deb
Selecting previously unselected package zabbix-release.
(Reading database ... 74489 files and directories currently installed.)
Preparing to unpack zabbix-release_6.5-1+ubuntu20.04_all.deb ...
Unpacking zabbix-release (1:6.5-1+ubuntu20.04) ...
Setting up zabbix-release (1:6.5-1+ubuntu20.04) ...
root@dc:~# _
```

```
sudo apt update
```

Ran sudo apt update to refresh the system package index so the new Zabbix repo is recognized.

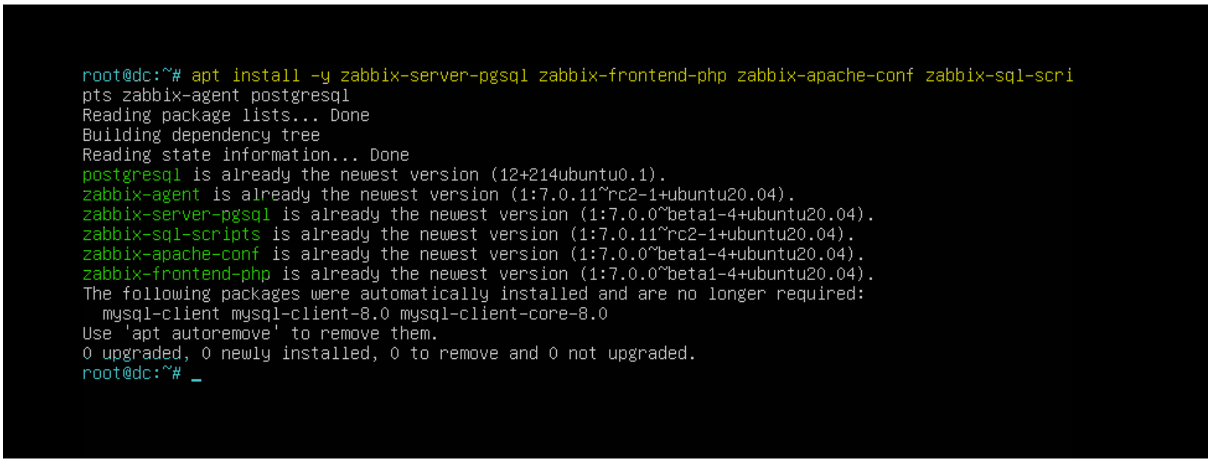


```
root@dc:~# apt update
Hit:1 http://lk.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://lk.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:3 https://repo.zabbix.com/zabbix-agent2-plugins/1/ubuntu focal InRelease [4,952 B]
Get:4 http://lk.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Get:5 https://repo.zabbix.com/zabbix/6.0/ubuntu focal InRelease [2,886 B]
Get:6 http://lk.archive.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Get:7 http://lk.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3,921 kB]
Get:8 https://repo.zabbix.com/zabbix-agent2-plugins/1/ubuntu focal/main Sources [1,004 B]
Get:9 https://repo.zabbix.com/zabbix-agent2-plugins/1/ubuntu focal/main amd64 Packages [625 B]
Get:10 https://repo.zabbix.com/zabbix/6.0/ubuntu focal/main Sources [37.9 kB]
Get:11 http://lk.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [595 kB]
Get:12 http://lk.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [3,854 kB]
36% [7 Packages store 0 B] [12 Packages 0 B/3,854 kB 0%] [10 Sources 16.1 kB/37.9 kB 43%]
```

02. Install the Required Zabbix Components

The Zabbix server, frontend and agent packages were installed:

```
sudo apt install -y zabbix-server-pgsql zabbix-frontend-php zabbix-apache-conf zabbix-sql-scripts zabbix-agent
```



```
root@dc:~# apt install -y zabbix-server-pgsql zabbix-frontend-php zabbix-apache-conf zabbix-sql-scri
pts zabbix-agent postgresql
Reading package lists... Done
Building dependency tree
Reading state information... Done
postgresql is already the newest version (12+214ubuntu0.1).
zabbix-agent is already the newest version (1:7.0.11~rc2-1+ubuntu20.04).
zabbix-server-pgsql is already the newest version (1:7.0.0~beta1-4+ubuntu20.04).
zabbix-sql-scripts is already the newest version (1:7.0.11~rc2-1+ubuntu20.04).
zabbix-apache-conf is already the newest version (1:7.0.0~beta1-4+ubuntu20.04).
zabbix-frontend-php is already the newest version (1:7.0.0~beta1-4+ubuntu20.04).
The following packages were automatically installed and are no longer required:
  mysql-client mysql-client-8.0 mysql-client-core-8.0
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@dc:~# _
```

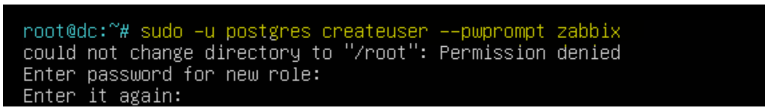
03. Install and Configure PostgreSQL 14

Since Zabbix 6.0 is compatible with PostgreSQL 14, it was installed via the PostgreSQL APT repository:

```
sudo apt install -y postgresql-14
```

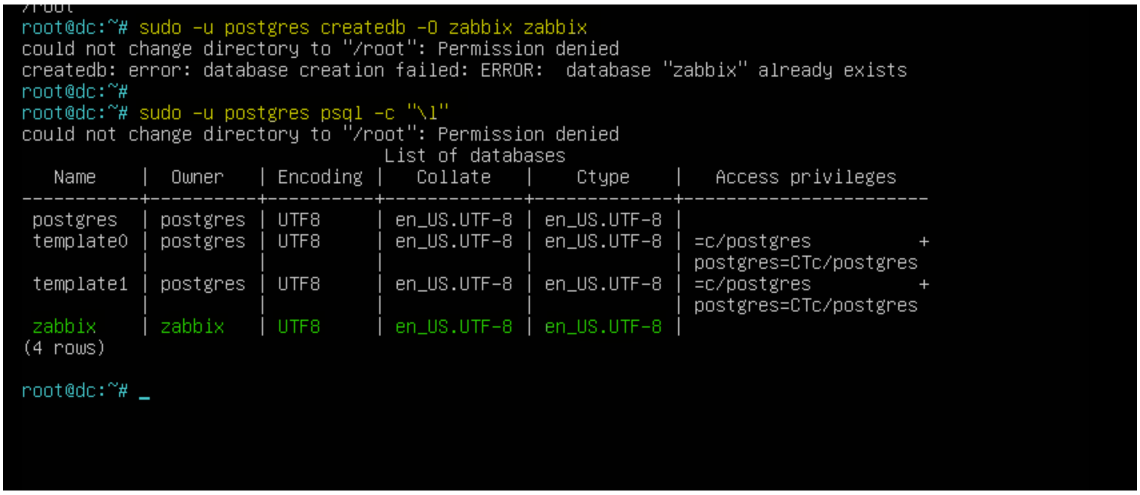
04. Create the Zabbix Database and User

```
sudo -u postgres createuser --pwprompt zabbix
```



```
root@dc:~# sudo -u postgres createuser --pwprompt zabbix
could not change directory to "/root": Permission denied
Enter password for new role:
Enter it again:
```

```
sudo -u postgres createdb -O zabbix Zabbix
```

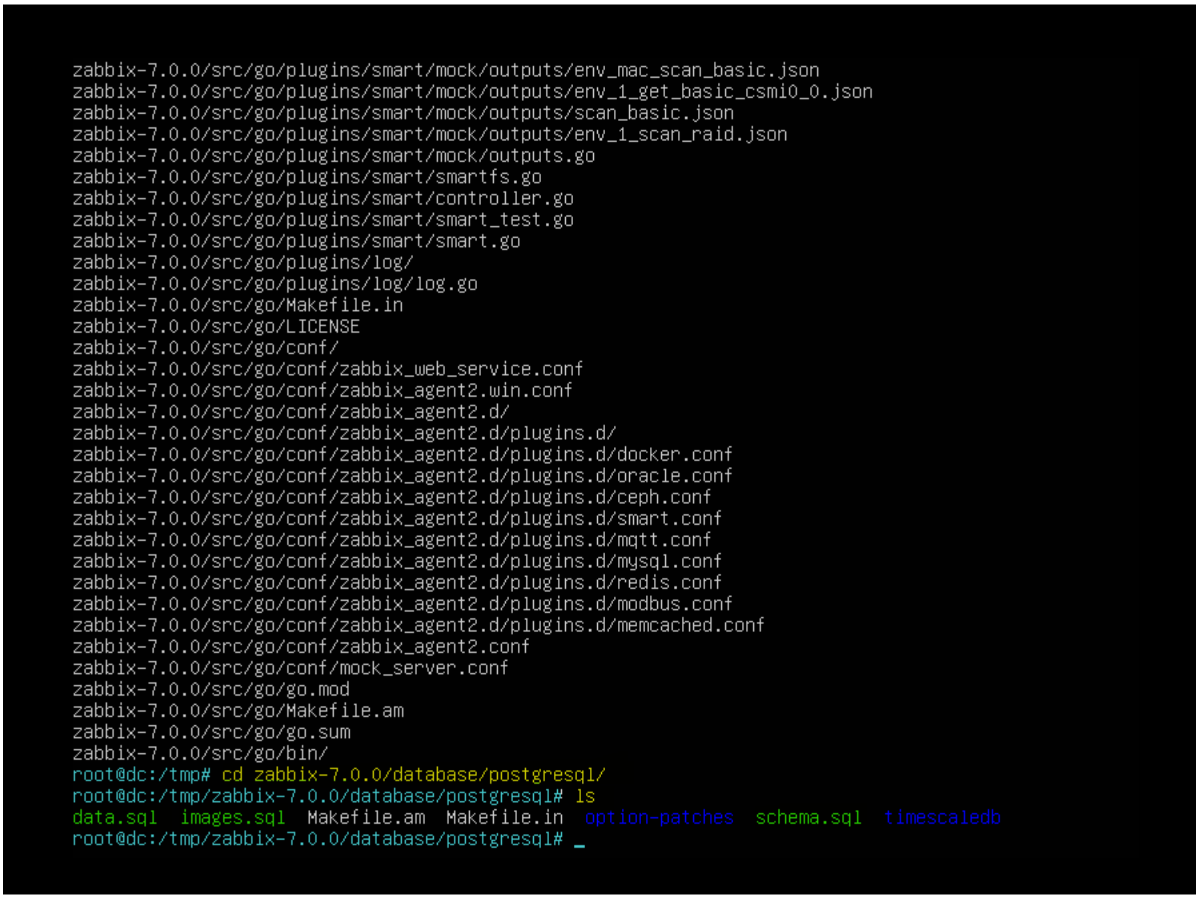


```
/root
root@dc:~# sudo -u postgres createdb -O zabbix zabbix
could not change directory to "/root": Permission denied
createdb: error: database creation failed: ERROR:  database "zabbix" already exists
root@dc:~#
root@dc:~# sudo -u postgres psql -c "\l"
could not change directory to "/root": Permission denied
                                  List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
 zabbix    | zabbix   | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
(4 rows)

root@dc:~# _
```

05. Import the Zabbix Schema and Data

The initial schema and data were imported into the newly created database:

```
zcat /usr/share/zabbix-sql-scripts/postgresql/server.sql.gz | sudo -u
zabbix psql Zabbix
```



```
zabbix-7.0.0/src/go/plugins/smart/mock/outputs/env_mac_scan_basic.json
zabbix-7.0.0/src/go/plugins/smart/mock/outputs/env_1_get_basic_csmi0_0.json
zabbix-7.0.0/src/go/plugins/smart/mock/outputs/scan_basic.json
zabbix-7.0.0/src/go/plugins/smart/mock/outputs/env_1_scan_raid.json
zabbix-7.0.0/src/go/plugins/smart/mock/outputs.go
zabbix-7.0.0/src/go/plugins/smart/smartfs.go
zabbix-7.0.0/src/go/plugins/smart/controller.go
zabbix-7.0.0/src/go/plugins/smart/smart_test.go
zabbix-7.0.0/src/go/plugins/smart/smart.go
zabbix-7.0.0/src/go/plugins/log/
zabbix-7.0.0/src/go/plugins/log/log.go
zabbix-7.0.0/src/go/Makefile.in
zabbix-7.0.0/src/go/LICENSE
zabbix-7.0.0/src/go/conf/
zabbix-7.0.0/src/go/conf/zabbix_web_service.conf
zabbix-7.0.0/src/go/conf/zabbix_agent2.win.conf
zabbix-7.0.0/src/go/conf/zabbix_agent2.d/
zabbix-7.0.0/src/go/conf/zabbix_agent2.d/plugins.d/
zabbix-7.0.0/src/go/conf/zabbix_agent2.d/plugins.d/docker.conf
zabbix-7.0.0/src/go/conf/zabbix_agent2.d/plugins.d/oracle.conf
zabbix-7.0.0/src/go/conf/zabbix_agent2.d/plugins.d/ceph.conf
zabbix-7.0.0/src/go/conf/zabbix_agent2.d/plugins.d/smart.conf
zabbix-7.0.0/src/go/conf/zabbix_agent2.d/plugins.d/mqtt.conf
zabbix-7.0.0/src/go/conf/zabbix_agent2.d/plugins.d/mysql.conf
zabbix-7.0.0/src/go/conf/zabbix_agent2.d/plugins.d/redis.conf
zabbix-7.0.0/src/go/conf/zabbix_agent2.d/plugins.d/modbus.conf
zabbix-7.0.0/src/go/conf/zabbix_agent2.d/plugins.d/memcached.conf
zabbix-7.0.0/src/go/conf/zabbix_agent2.conf
zabbix-7.0.0/src/go/conf/mock_server.conf
zabbix-7.0.0/src/go/go.mod
zabbix-7.0.0/src/go/Makefile.am
zabbix-7.0.0/src/go/go.sum
zabbix-7.0.0/src/go/bin/
root@dc:/tmp# cd zabbix-7.0.0/database/postgresql/
root@dc:/tmp/zabbix-7.0.0/database/postgresql# ls
data.sql  images.sql  Makefile.am  Makefile.in  option-patches  schema.sql  timescaledb
root@dc:/tmp/zabbix-7.0.0/database/postgresql# _
```

06. Configure the Zabbix Server

Database connection settings were defined in the Zabbix server configuration file:

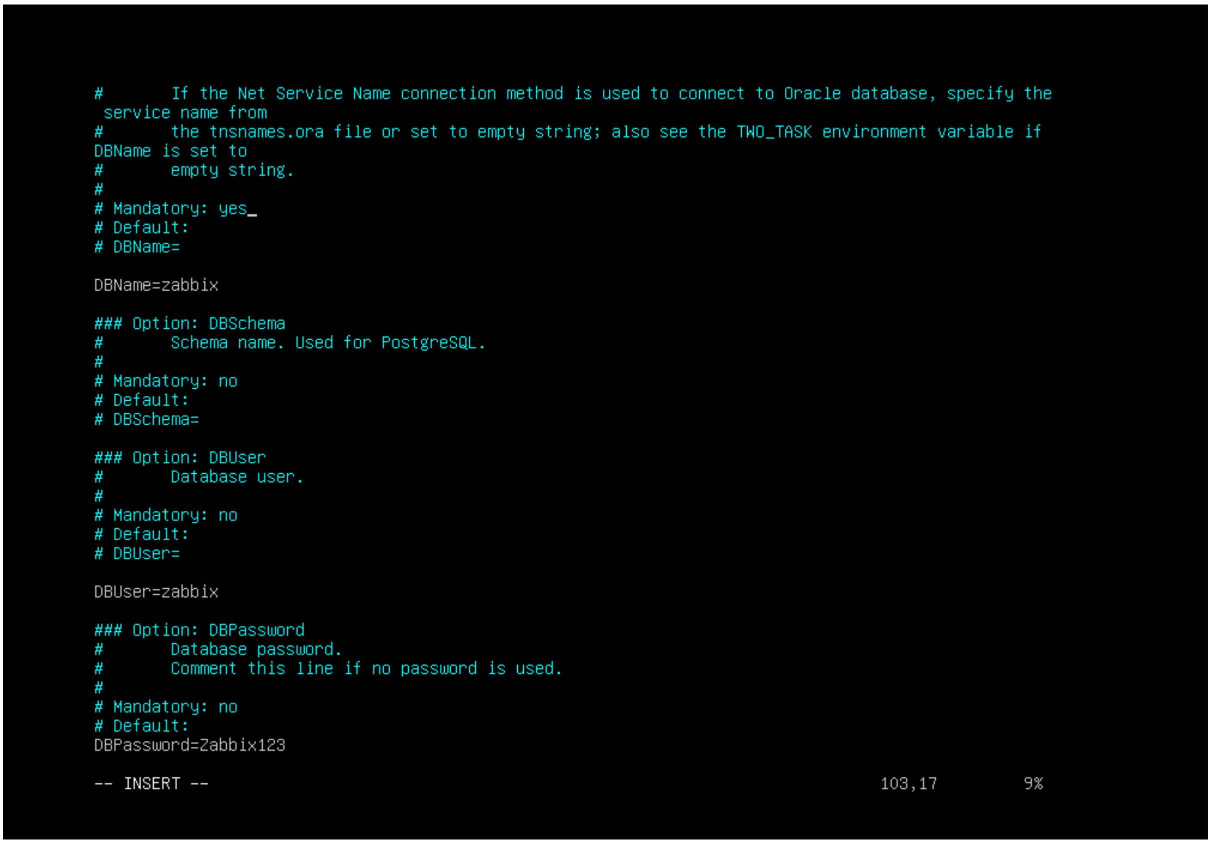
```
sudo vi /etc/zabbix/zabbix_server.conf
```

Set the following:

```
DBName=zabbix
```

```
DBUser=zabbix
```

```
DBPassword=password
```



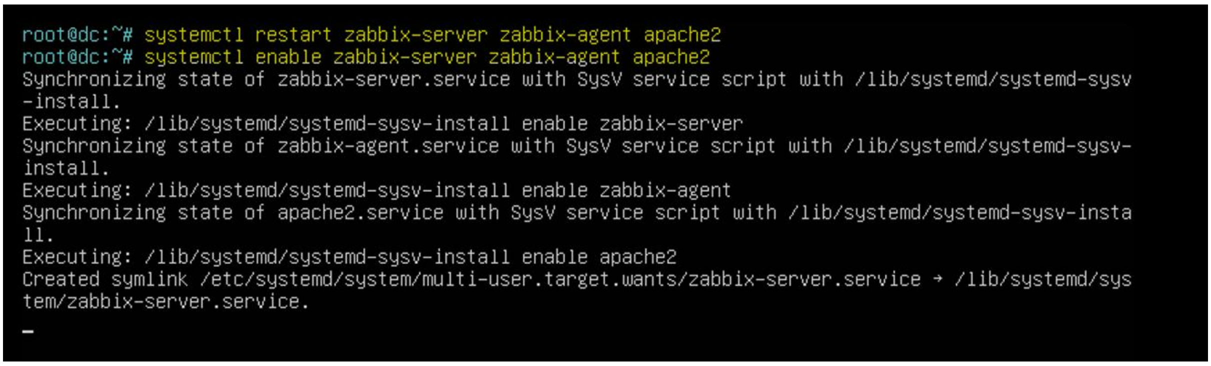
```
#       If the Net Service Name connection method is used to connect to Oracle database, specify the
 service name from
#       the tnsnames.ora file or set to empty string; also see the TWO_TASK environment variable if
DBName is set to
#       empty string.
#
# Mandatory: yes_
# Default:
# DBName=

DBName=zabbix

### Option: DBSchema
#       Schema name. Used for PostgreSQL.
#
# Mandatory: no
# Default:
# DBSchema=

### Option: DBUser
#       Database user.
#
# Mandatory: no
# Default:
# DBUser=

DBUser=zabbix

### Option: DBPassword
#       Database password.
#       Comment this line if no password is used.
#
# Mandatory: no
# Default:
DBPassword=Zabbix123

-- INSERT --                                                                  103,17        9%
```

07. Start and Enable Zabbix Services

The Zabbix server and agent services were enabled and started:

```
sudo systemctl restart zabbix-server zabbix-agent apache2
```

```
sudo systemctl enable zabbix-server zabbix-agent apache2
```



```
root@dc:~# systemctl restart zabbix-server zabbix-agent apache2
root@dc:~# systemctl enable zabbix-server zabbix-agent apache2
Synchronizing state of zabbix-server.service with SysV service script with /lib/systemd/systemd-sysv
-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-server
Synchronizing state of zabbix-agent.service with SysV service script with /lib/systemd/systemd-sysv-
install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-agent
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-insta
ll.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-server.service → /lib/systemd/sys
tem/zabbix-server.service.
_
```

08. Configure the Zabbix Frontend

Accessed the Zabbix frontend via web browser (http://localhost/zabbix) to complete the setup wizard.

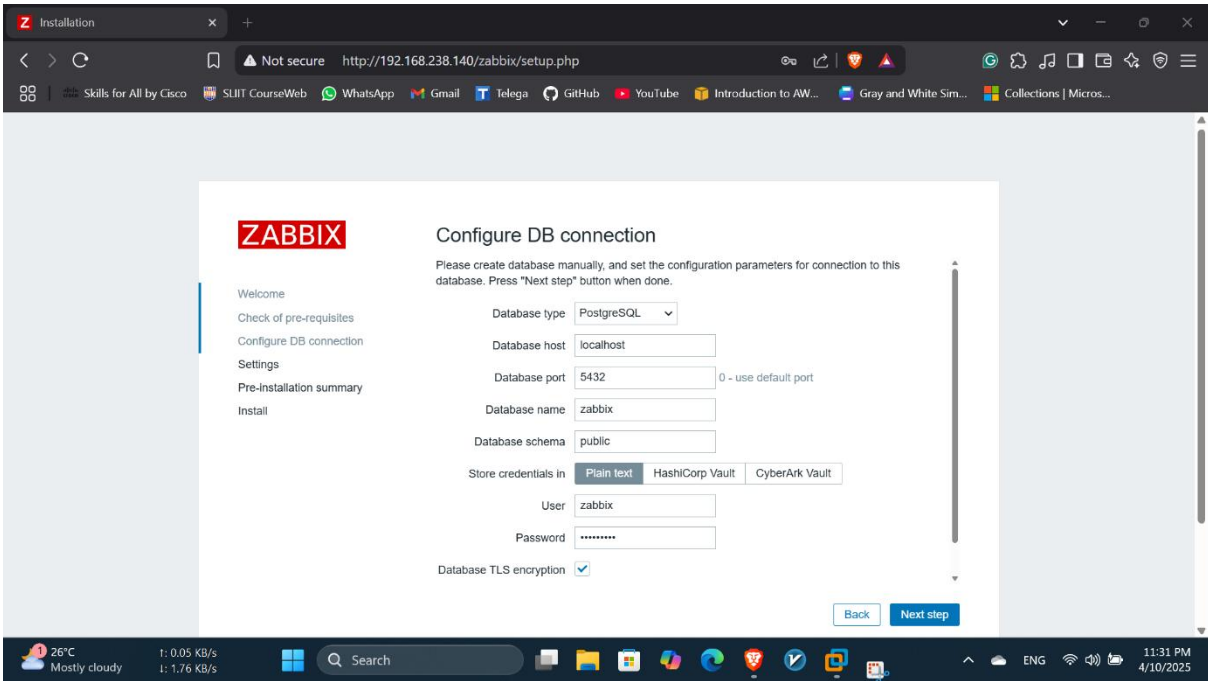
Follow the installation wizard:

- Enter DB credentials
- Verify requirements

- Finish installation

Default login:

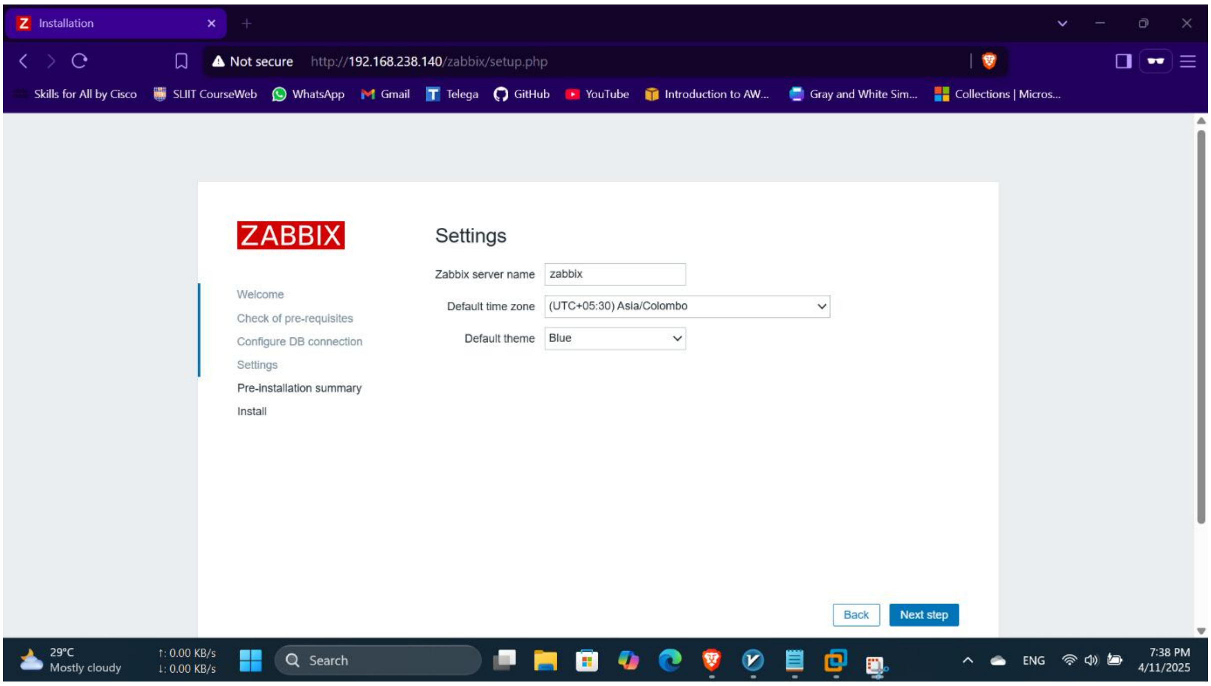
- **Username:** Admin
- **Password:** Zabbix



Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

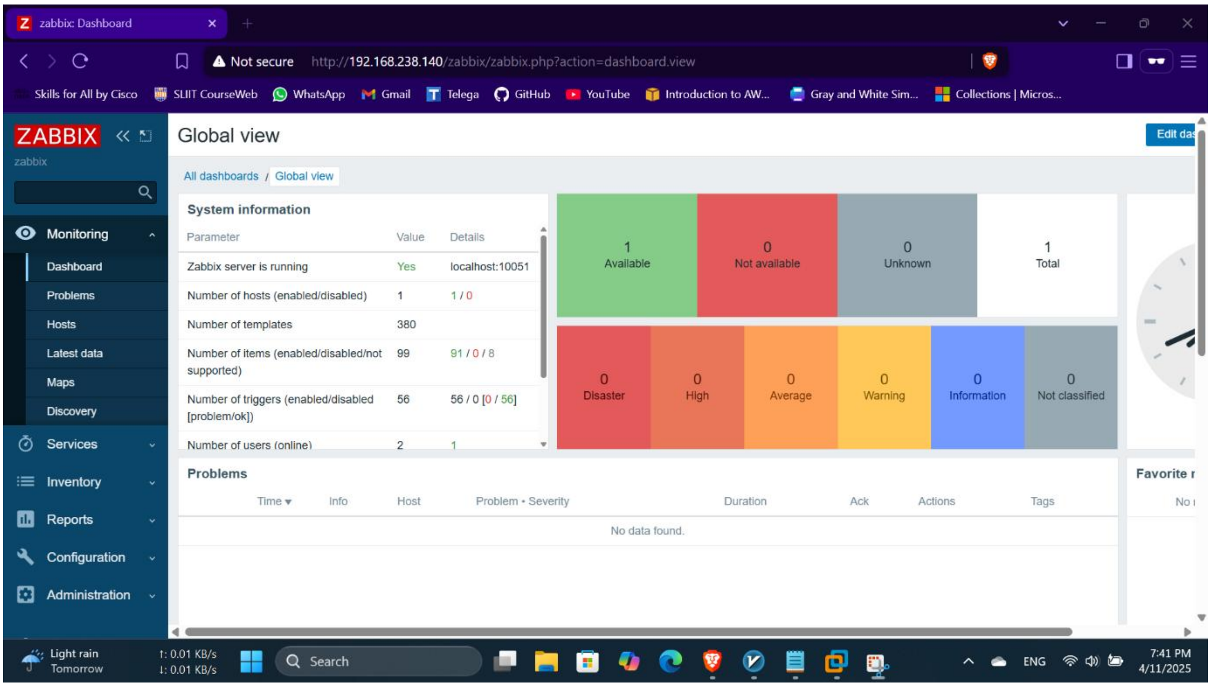
Database type: PostgreSQL
Database host: localhost
Database port: 5432 (0 - use default port)
Database name: zabbix
Database schema: public
Store credentials in: Plain text | HashiCorp Vault | CyberArk Vault
User: zabbix
Password: ••••••••••
Database TLS encryption: ☑



Settings

Zabbix server name: zabbix
Default time zone: (UTC+05:30) Asia/Colombo
Default theme: Blue

The Zabbix Dashboard is the central hub for monitoring and visualizing the health and performance of your IT infrastructure. It displays real-time critical metrics, including system uptime, CPU usage, memory load, network traffic, disk space, and more. The dashboard also shows problem statuses, recent alerts, trigger events, and monitored host availability using widgets and graphs. This helps administrators quickly identify and respond to system issues, improving uptime and operational awareness.



Global view

All dashboards / Global view

System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (enabled/disabled)	1	1 / 0
Number of templates	380	
Number of items (enabled/disabled/not supported)	99	91 / 0 / 8
Number of triggers (enabled/disabled [problem/ok])	56	56 / 0 [0 / 56]
Number of users (online)	2	1

Available	Not available	Unknown	Total
1	0	0	1

Disaster	High	Average	Warning	Information	Not classified
0	0	0	0	0	0

Problems

Time ▾	Info	Host	Problem • Severity	Duration	Ack	Actions	Tags
No data found.							

Favorite r

No r

3.4 Installing and Configuring Zabbix Agent on Domain Controller

01. Install Zabbix Agent

Installs the standard Zabbix agent package on the Domain Controller enabling it to be monitored by the Zabbix server.

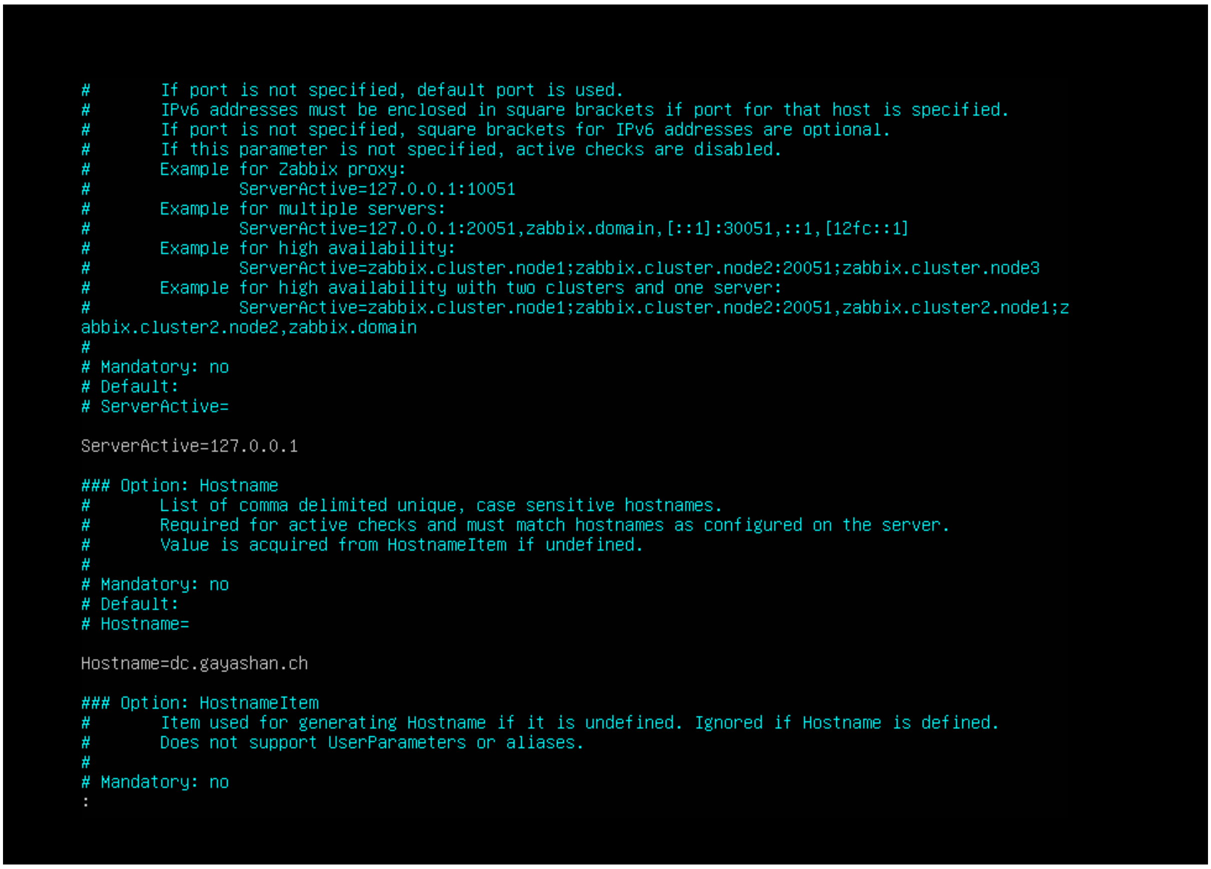
```
sudo apt install zabbix-agent
```

02. Configure the Zabbix Agent

Specifies which server is allowed to communicate with the agent and sets the hostname as it will appear in the Zabbix frontend.

Edit the configuration file:

```
sudo vi /etc/zabbix/zabbix_agentd.conf
```



```
#       If port is not specified, default port is used.
#       IPv6 addresses must be enclosed in square brackets if port for that host is specified.
#       If port is not specified, square brackets for IPv6 addresses are optional.
#       If this parameter is not specified, active checks are disabled.
#       Example for Zabbix proxy:
#               ServerActive=127.0.0.1:10051
#       Example for multiple servers:
#               ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
#       Example for high availability:
#               ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
#       Example for high availability with two clusters and one server:
#               ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;z
abbix.cluster2.node2,zabbix.domain
#
# Mandatory: no
# Default:
# ServerActive=

ServerActive=127.0.0.1

### Option: Hostname
#       List of comma delimited unique, case sensitive hostnames.
#       Required for active checks and must match hostnames as configured on the server.
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

Hostname=dc.gayashan.ch

### Option: HostnameItem
#       Item used for generating Hostname if it is undefined. Ignored if Hostname is defined.
#       Does not support UserParameters or aliases.
#
# Mandatory: no
:
```

Update or set the following parameters:

```
Server=<Zabbix_Server_IP>
```

```
ServerActive=<Zabbix_Server_IP>
```

```
Hostname=dc.gayashan.ch
```

03. Enable and Start Zabbix Agent Service

Ensures that the Zabbix agent starts on boot and immediately begins sending data to the Zabbix server.

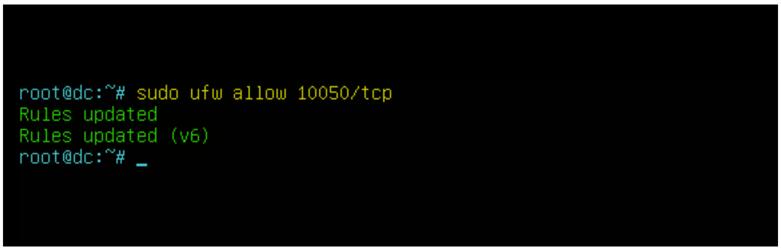
```
sudo systemctl enable zabbix-agent
```

```
sudo systemctl start zabbix-agent
```

04. Open Firewall Port (If UFW is enabled)

Allows incoming connections from the Zabbix server on the default agent port.

```
sudo ufw allow 10050/tcp
```

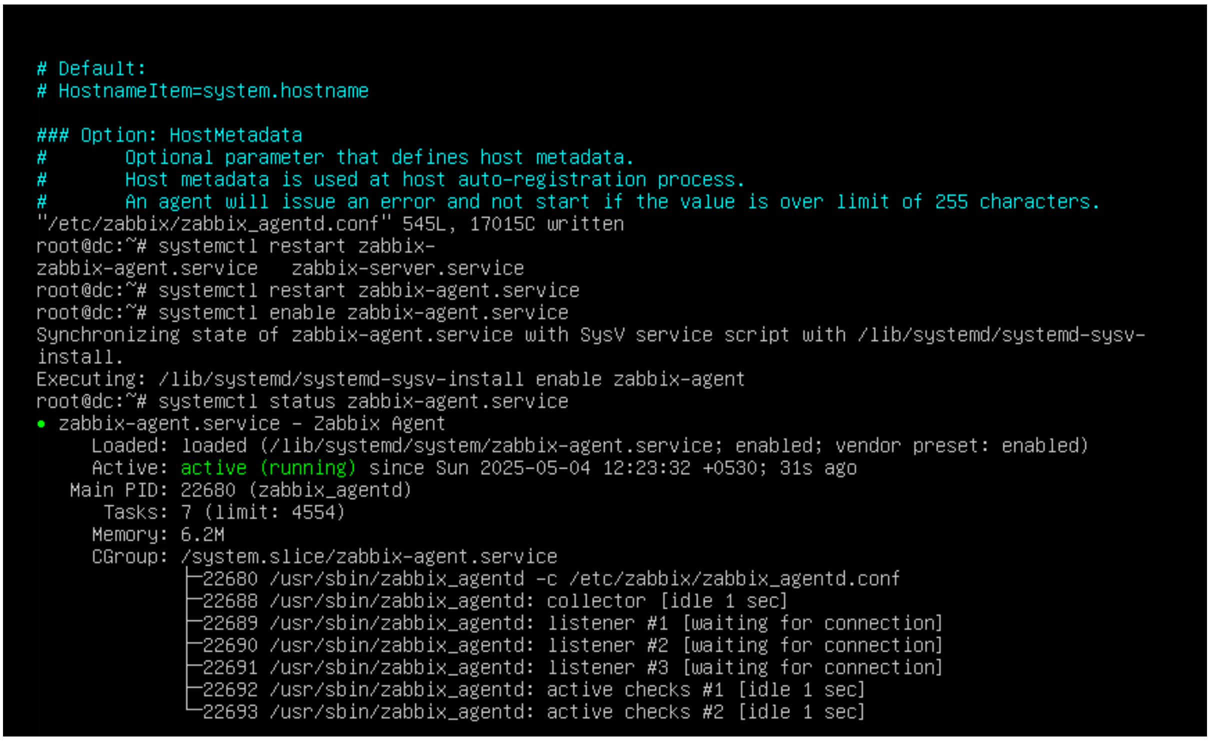


```
root@dc:~# sudo ufw allow 10050/tcp
Rules updated
Rules updated (v6)
root@dc:~# _
```

05. Verify Agent Status

Confirms that the agent is running correctly without any startup issues.

```
sudo systemctl status zabbix-agent
```

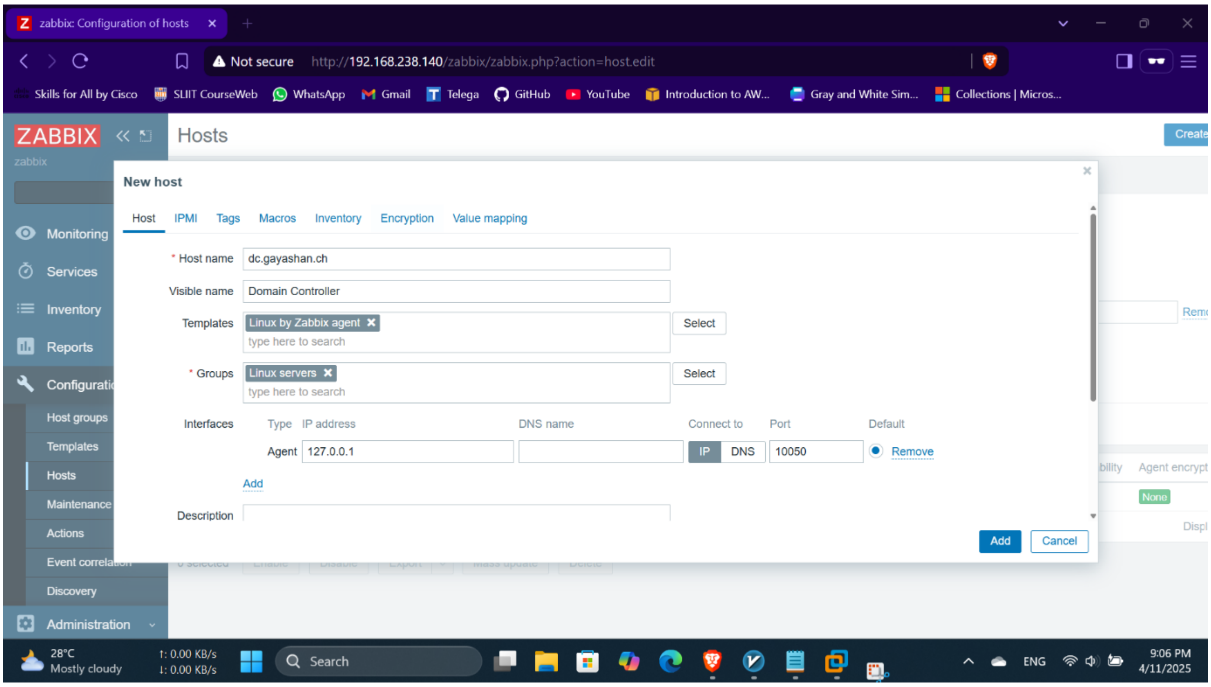


```
# Default:
# HostnameItem=system.hostname

### Option: HostMetadata
#       Optional parameter that defines host metadata.
#       Host metadata is used at host auto-registration process.
#       An agent will issue an error and not start if the value is over limit of 255 characters.
"/etc/zabbix/zabbix_agentd.conf" 545L, 17015C written
root@dc:~# systemctl restart zabbix-
zabbix-agent.service   zabbix-server.service
root@dc:~# systemctl restart zabbix-agent.service
root@dc:~# systemctl enable zabbix-agent.service
Synchronizing state of zabbix-agent.service with SysV service script with /lib/systemd/systemd-sysv-
install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-agent
root@dc:~# systemctl status zabbix-agent.service
● zabbix-agent.service - Zabbix Agent
     Loaded: loaded (/lib/systemd/system/zabbix-agent.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-05-04 12:23:32 +0530; 31s ago
   Main PID: 22680 (zabbix_agentd)
      Tasks: 7 (limit: 4554)
     Memory: 6.2M
     CGroup: /system.slice/zabbix-agent.service
             ├─22680 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
             ├─22688 /usr/sbin/zabbix_agentd: collector [idle 1 sec]
             ├─22689 /usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
             ├─22690 /usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
             ├─22691 /usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
             ├─22692 /usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]
             └─22693 /usr/sbin/zabbix_agentd: active checks #2 [idle 1 sec]
```

06. Add Host to Zabbix Frontend

- Go to Configuration > Hosts in Zabbix frontend.
- Click "Create Host".
- Set hostname to dc.gayashan.ch and assign it to a group (e.g., "Linux servers").
- Set the Agent interface to the Domain Controller's IP.
- Link the template: Template OS Linux by Zabbix agent.

This registers the Domain Controller as a monitored host in Zabbix and links it to a Linux monitoring template for data collection.



New host

* Host name: dc.gayashan.ch

Visible name: Domain Controller

Templates: Linux by Zabbix agent

* Groups: Linux servers

Interfaces:

Type	IP address	DNS name	Connect to	Port	Default
Agent	127.0.0.1		IP	10050	●

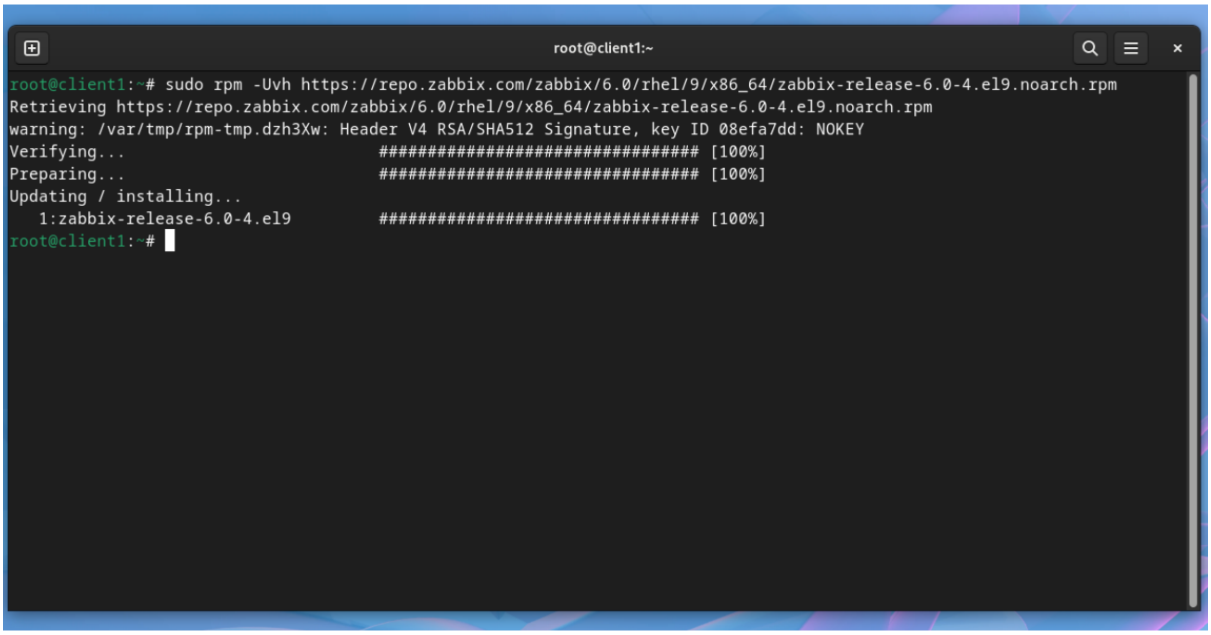
Description:

3.5 Installing and Configuring Zabbix Agent on Fedora Client

01. Add Zabbix Repository for Fedora 39

Adds the official Zabbix 6.0 repository compatible with Fedora 39 via RHEL 9 packages.

```
sudo rpm -Uvh https://repo.zabbix.com/zabbix/6.0/rhel/9/x86_64/zabbix-release-6.0-4.el9.noarch.rpm
```

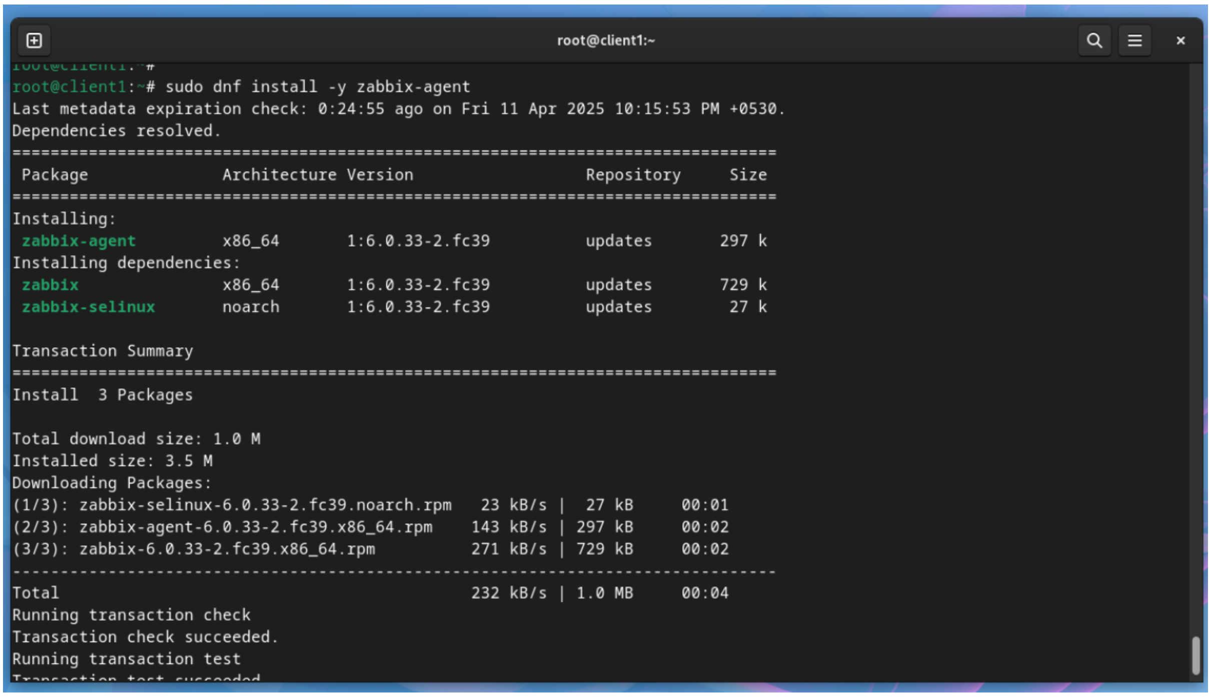


```
root@client1:~# sudo rpm -Uvh https://repo.zabbix.com/zabbix/6.0/rhel/9/x86_64/zabbix-release-6.0-4.el9.noarch.rpm
Retrieving https://repo.zabbix.com/zabbix/6.0/rhel/9/x86_64/zabbix-release-6.0-4.el9.noarch.rpm
warning: /var/tmp/rpm-tmp.dzh3Xw: Header V4 RSA/SHA512 Signature, key ID 08efa7dd: NOKEY
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:zabbix-release-6.0-4.el9         ################################# [100%]
root@client1:~# 
```

02. Install Zabbix Agent

Installs the Zabbix agent package which collects and sends monitoring data from the Fedora client.

```
sudo dnf install zabbix-agent
```

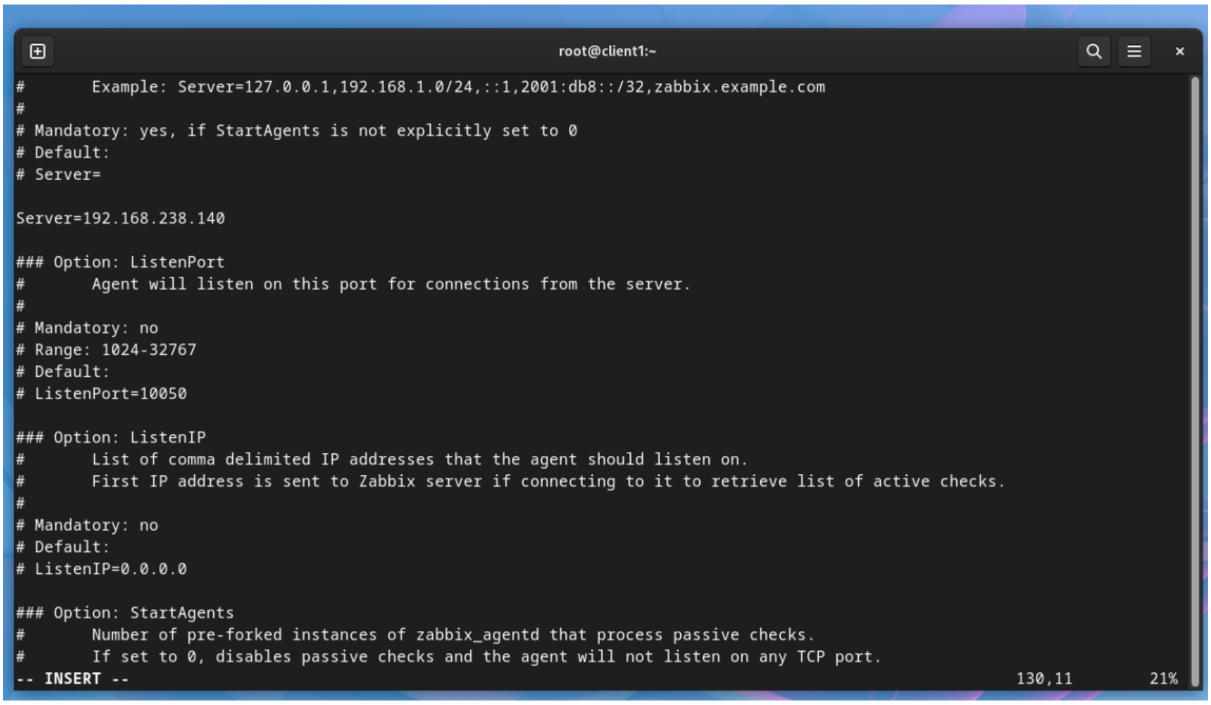


```
root@client1:~#
root@client1:~# sudo dnf install -y zabbix-agent
Last metadata expiration check: 0:24:55 ago on Fri 11 Apr 2025 10:15:53 PM +0530.
Dependencies resolved.
================================================================================
 Package              Architecture Version                 Repository      Size
================================================================================
Installing:
 zabbix-agent         x86_64       1:6.0.33-2.fc39         updates        297 k
Installing dependencies:
 zabbix               x86_64       1:6.0.33-2.fc39         updates        729 k
 zabbix-selinux       noarch       1:6.0.33-2.fc39         updates         27 k

Transaction Summary
================================================================================
Install  3 Packages

Total download size: 1.0 M
Installed size: 3.5 M
Downloading Packages:
(1/3): zabbix-selinux-6.0.33-2.fc39.noarch.rpm   23 kB/s |  27 kB     00:01
(2/3): zabbix-agent-6.0.33-2.fc39.x86_64.rpm    143 kB/s | 297 kB     00:02
(3/3): zabbix-6.0.33-2.fc39.x86_64.rpm          271 kB/s | 729 kB     00:02
--------------------------------------------------------------------------------
Total                                           232 kB/s | 1.0 MB     00:04
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
```

03. Configure the Zabbix Agent

Edit the config file:

```
sudo vi /etc/zabbix/zabbix_agentd.conf
```



```
#       Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes, if StartAgents is not explicitly set to 0
# Default:
# Server=

Server=192.168.238.140

### Option: ListenPort
#       Agent will listen on this port for connections from the server.
#
# Mandatory: no
# Range: 1024-32767
# Default:
# ListenPort=10050

### Option: ListenIP
#       List of comma delimited IP addresses that the agent should listen on.
#       First IP address is sent to Zabbix server if connecting to it to retrieve list of active checks.
#
# Mandatory: no
# Default:
# ListenIP=0.0.0.0

### Option: StartAgents
#       Number of pre-forked instances of zabbix_agentd that process passive checks.
#       If set to 0, disables passive checks and the agent will not listen on any TCP port.
-- INSERT --                                                                    130,11        21%
```

Update these fields:

```
Server=<Zabbix_Server_IP>
```

```
ServerActive=<Zabbix_Server_IP>
```

```
Hostname=fedora-client
```

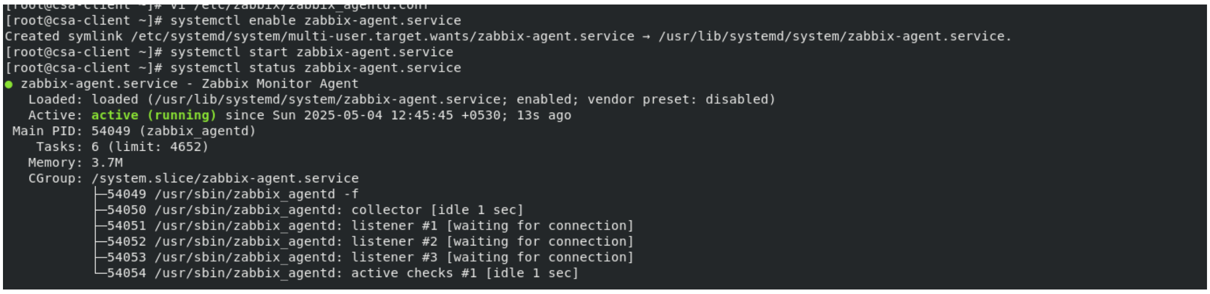
Defines which Zabbix server can communicate with the agent and assigns a recognizable hostname for identification.

04. Enable and Start Zabbix Agent

Enables the agent to run at boot and starts the service to begin monitoring immediately.

```
sudo systemctl enable zabbix-agent
```

```
sudo systemctl start zabbix-agent
```

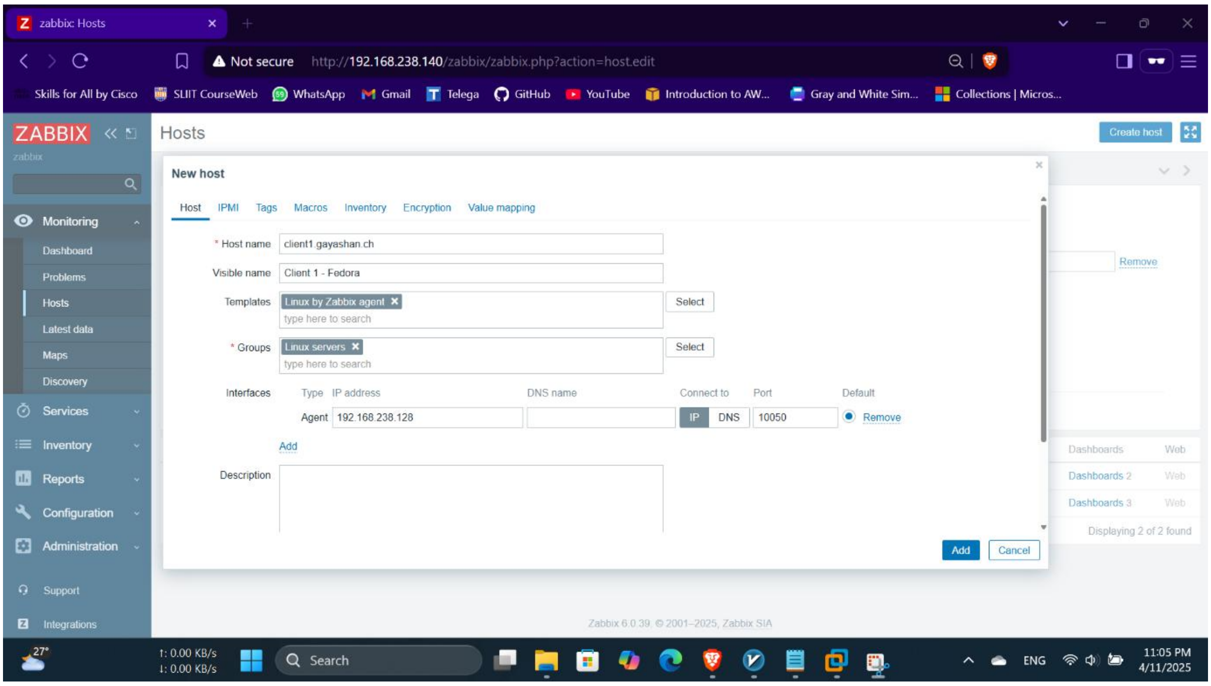


```
[root@csa-client ~]# vi /etc/zabbix/zabbix_agentd.conf
[root@csa-client ~]# systemctl enable zabbix-agent.service
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-agent.service → /usr/lib/systemd/system/zabbix-agent.service.
[root@csa-client ~]# systemctl start zabbix-agent.service
[root@csa-client ~]# systemctl status zabbix-agent.service
● zabbix-agent.service - Zabbix Monitor Agent
   Loaded: loaded (/usr/lib/systemd/system/zabbix-agent.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2025-05-04 12:45:45 +0530; 13s ago
 Main PID: 54049 (zabbix_agentd)
    Tasks: 6 (limit: 4652)
   Memory: 3.7M
   CGroup: /system.slice/zabbix-agent.service
           ├─54049 /usr/sbin/zabbix_agentd -f
           ├─54050 /usr/sbin/zabbix_agentd: collector [idle 1 sec]
           ├─54051 /usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
           ├─54052 /usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
           ├─54053 /usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
           └─54054 /usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]
```

05. Add Fedora Client as Host in Zabbix Frontend

- Navigate to Configuration > Hosts.
- Click "Create Host".
- Set fedora-client as the hostname.
- Set the Agent interface to the Fedora client’s IP.
- Link the template: Template OS Linux by Zabbix agent.

This step registers the Fedora machine as a monitored host in the Zabbix server, applying the Linux monitoring template.



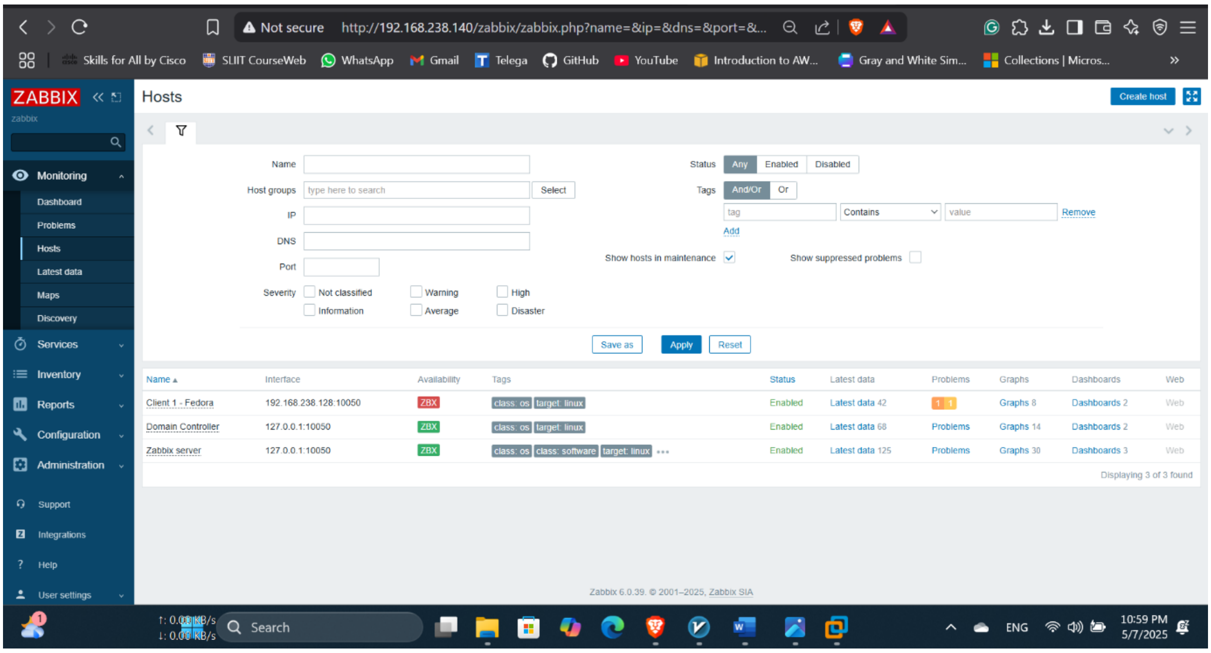
New host

Host | IPMI | Tags | Macros | Inventory | Encryption | Value mapping

* Host name	client1.gayashan.ch
Visible name	Client 1 - Fedora
Templates	Linux by Zabbix agent
* Groups	Linux servers

Interfaces

Type	IP address	DNS name	Connect to	Port	Default
Agent	192.168.238.128		IP	10050	●



Hosts

Name ▲	Interface	Availability	Tags	Status	Latest data	Problems	Graphs	Dashboards	Web
Client 1 - Fedora	192.168.238.128:10050	ZBX	class: os target: linux	Enabled	Latest data 42	1 1	Graphs 8	Dashboards 2	Web
Domain Controller	127.0.0.1:10050	ZBX	class: os target: linux	Enabled	Latest data 68	Problems	Graphs 14	Dashboards 2	Web
Zabbix server	127.0.0.1:10050	ZBX	class: os class: software target: linux •••	Enabled	Latest data 125	Problems	Graphs 30	Dashboards 3	Web

Displaying 3 of 3 found

Zabbix 6.0.39. © 2001–2025, Zabbix SIA

04. Challenges and Solutions

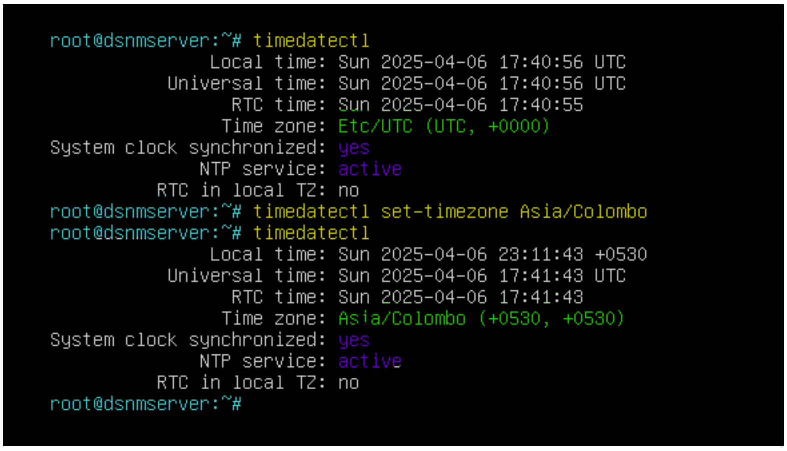
01. Incorrect Time Display in Zabbix Monitoring

Description:
After setting up Zabbix Server and Agent on the domain controller (Ubuntu 20.04), system metrics and logs in the Zabbix dashboard showed incorrect timestamps. This discrepancy affected monitoring accuracy and made it difficult to align events and alerts with the actual local time.

Root Cause:
By default, Ubuntu uses UTC (Coordinated Universal Time) as the system time zone, while Zabbix reflects server time for logs and metrics. Since the system was running in Sri Lanka, this led to a mismatch between observed times in logs and the actual local time.

Solution:
The system time zone was updated to match the local time zone used in Sri Lanka (Asia/Colombo). The following command was executed:

```
sudo timedatectl set-timezone Asia/Colombo
```



```
root@dsnmserver:~# timedatectl
               Local time: Sun 2025-04-06 17:40:56 UTC
           Universal time: Sun 2025-04-06 17:40:56 UTC
                 RTC time: Sun 2025-04-06 17:40:55
                Time zone: Etc/UTC (UTC, +0000)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
root@dsnmserver:~# timedatectl set-timezone Asia/Colombo
root@dsnmserver:~# timedatectl
               Local time: Sun 2025-04-06 23:11:43 +0530
           Universal time: Sun 2025-04-06 17:41:43 UTC
                 RTC time: Sun 2025-04-06 17:41:43
                Time zone: Asia/Colombo (+0530, +0530)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
root@dsnmserver:~#
```

This ensured that all time-based logs, triggers, and graphs in the Zabbix dashboard aligned accurately with local time, enabling correct event tracking and correlation.

Verification:
To verify the new time zone setting, this command was used:

```
timedatectl
```

The output confirmed that the system time was set to `Asia/Colombo` and synchronized correctly with the hardware clock.

02. PostgreSQL Version Compatibility with Zabbix

Description:
During the initial setup of the Zabbix server, an error occurred indicating that the installed PostgreSQL version was not supported. The Zabbix server log displayed the following error message:

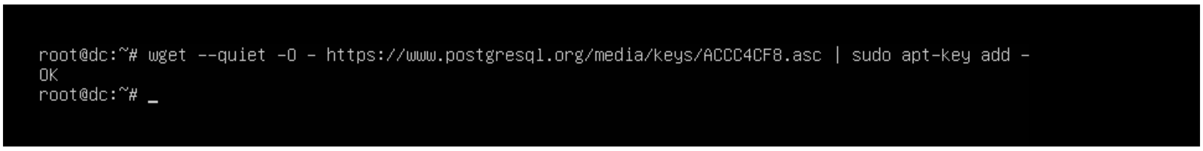
"Unable to start Zabbix server due to unsupported PostgreSQL version (12.2). Minimum required version: 13.0"

Root Cause:
Zabbix 7.0 and 6.0 require a minimum PostgreSQL version of 13.0 to ensure compatibility with the database schema and features used by Zabbix. However, Ubuntu 20.04's default repositories included an older version (12.2), which was incompatible with Zabbix's requirements.

Solution:
To resolve this issue, PostgreSQL 14 was manually added and installed from the official PostgreSQL repository. The following steps were taken:

01. Import the PostgreSQL GPG key

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc |
sudo apt-key add –
```

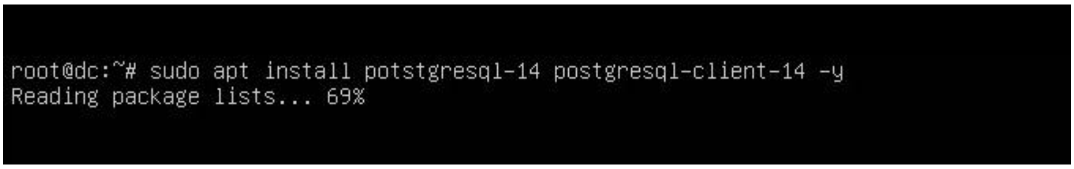


```
root@dc:~# wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
OK
root@dc:~# _
```

02. Add the PostgreSQL 14 repository

```
echo "deb http://apt.postgresql.org/pub/repos/apt/ focal-pgdg main" | sudo
tee /etc/apt/sources.list.d/pgdg.list
```

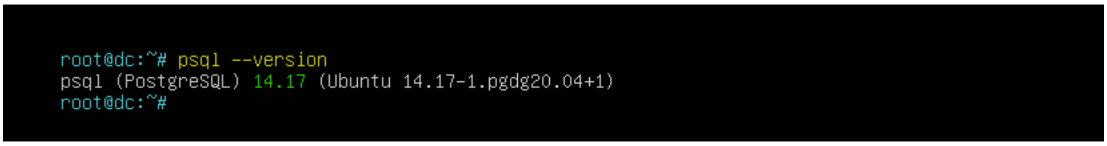
03. Update package lists and install PostgreSQL 14



```
root@dc:~# sudo apt install potstgresql-14 postgresql-client-14 -y
Reading package lists... 69%
```

04. Verify the installed version

```
psql -version
```



```
root@dc:~# psql --version
psql (PostgreSQL) 14.17 (Ubuntu 14.17-1.pgdg20.04+1)
root@dc:~#
```

Outcome:
After installing PostgreSQL 14, Zabbix server started successfully, and the database schema was loaded without errors. Monitoring operations continued smoothly with the supported database backend.

05. Conclusion

This project successfully demonstrated the implementation of a Linux-based Domain Controller using Samba, integrated with Zabbix monitoring for both the server and a Fedora client machine. Through the course of the setup, various real-world challenges such as software version mismatches, PostgreSQL compatibility issues, and time synchronization problems were encountered and effectively resolved. The system was carefully configured to ensure centralized authentication through Active Directory and proactive infrastructure monitoring via Zabbix. By completing this assignment, valuable hands-on experience was gained in system administration, domain services, network monitoring, and troubleshooting in a mixed operating system environment. This solution lays a strong foundation for scalable and secure enterprise IT infrastructure management.

06. References

[1] S. Samba Team, "Samba - Opening Windows to a Wider World," *Samba Official Documentation*, [Online]. Available: https://www.samba.org/samba/docs/. [Accessed: May 7, 2025].

[2] Zabbix LLC, "Zabbix Documentation 6.0 LTS," *Zabbix Official Documentation*, 2024. [Online]. Available: https://www.zabbix.com/documentation/6.0/manual. [Accessed: May 7, 2025].

[3] PostgreSQL Global Development Group, "PostgreSQL 14 Documentation," *PostgreSQL Official Documentation*, 2024. [Online]. Available: https://www.postgresql.org/docs/14/. [Accessed: May 7, 2025].

[4] Red Hat, Inc., "Fedora 39 System Administrator's Guide," *Fedora Documentation*, 2024. [Online]. Available: https://docs.fedoraproject.org/en-US/fedora/latest/system-administrators-guide/. [Accessed: May 7, 2025].

[5] Canonical Ltd., "Ubuntu 20.04 LTS Documentation," *Ubuntu Documentation*, 2020. [Online]. Available: https://help.ubuntu.com/lts/serverguide/. [Accessed: May 7, 2025].