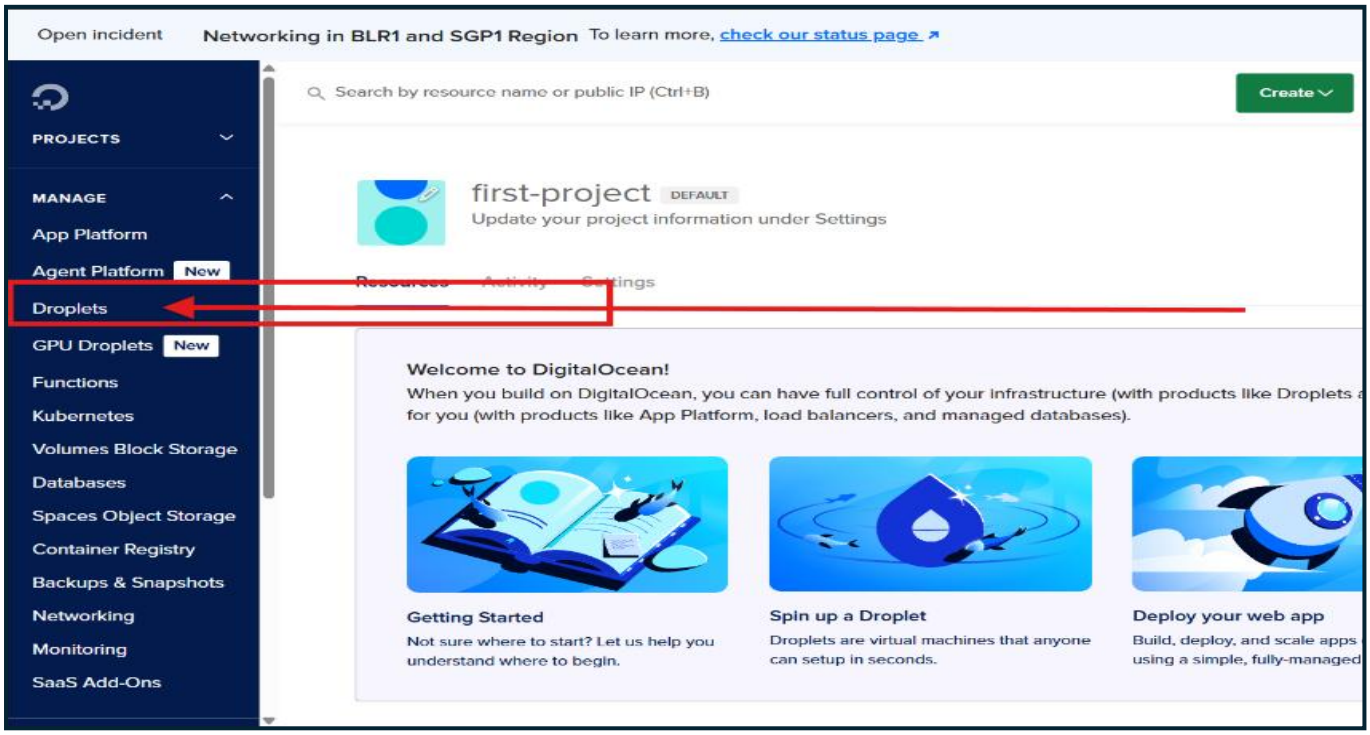
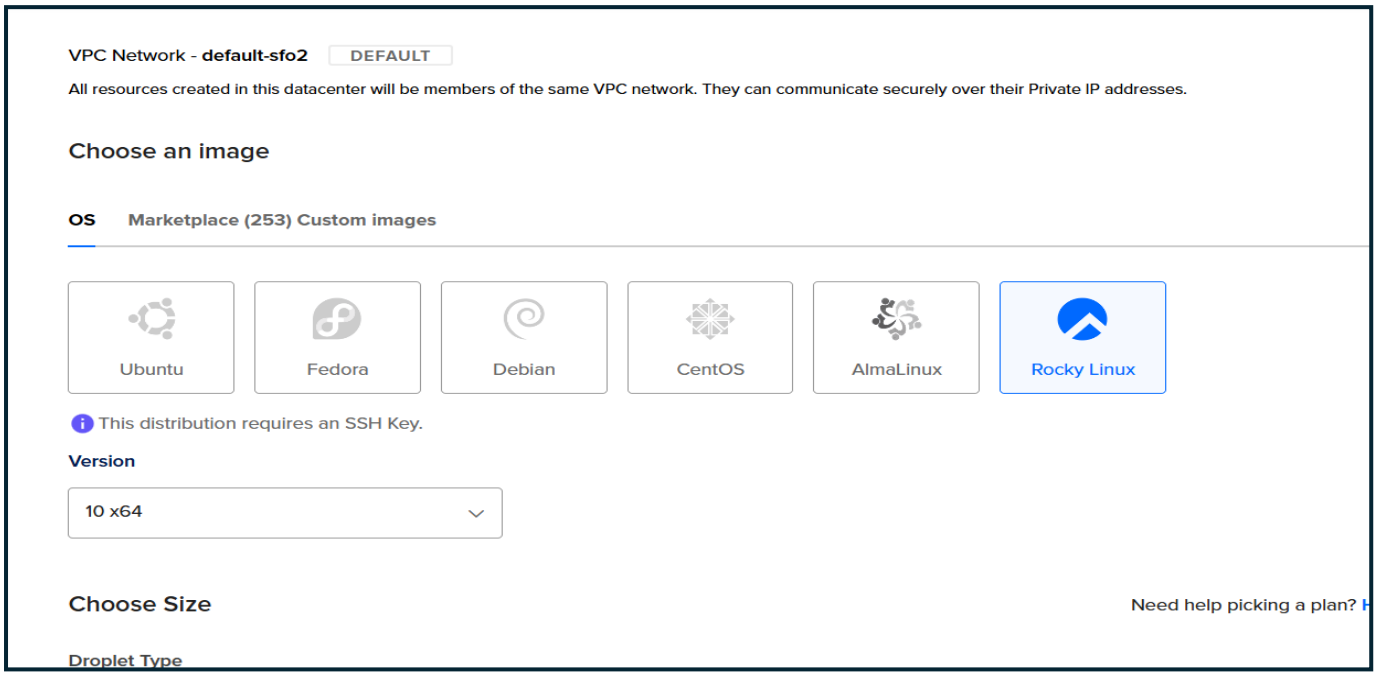
Digital Ocean Server Setup with Keycloak SSO

Page 1: Infrastructure & Keycloak Foundation (SETUP)

1. **Digital Ocean Droplet & Initial Setup**
   A. Create the Droplet:
      1. Log in to your Digital Ocean control panel.
      2. Click Create > **Droplets**.
      3. **Operating System:** Choose Rocky Linux 10.
      4. Plan: Select a plan that meets your needs (e.g., **Basic Shared CPU with 2GB+ RAM**).
      5. IP: **IPv6** should be enabled along with IPv4.
      6. **Data Center:** Choose a region close to your users.
      7. **Authentication**: Add your SSH key. This is more secure than using a password.
      8. Finalize: Set a hostname and click **Create Droplet**.





Choose Size

Need help picking a plan? Help me choose

Droplet Type

SHARED CPU	DEDICATED CPU			
Basic (Plan selected)	General Purpose	CPU-Optimized	Memory-Optimized	Storage-Optimized

Basic virtual machines with a mix of memory and compute resources. Best for small projects that can handle variable levels of CPU performance, like blogs, web apps and dev/test environments.

CPU options

- Regular — Disk type: SSD
- Premium Intel — Disk: NVMe SSD
- Premium AMD — Disk: NVMe SSD

4/mo, 06/hour	$6/mo, $0.009/hour	$12/mo, $0.018/hour	$18/mo, $0.027/hour	$24/mo, $0.036/hour	$48/mo, $0.071/hour	$9, $0.1
IB / 1 CPU, SSD Disk	1 GB / 1 CPU, 25 GB SSD Disk	2 GB / 1 CPU, 50 GB SSD Disk	2 GB / 2 CPUs, 60 GB SSD Disk	4 GB / 2 CPUs, 80 GB SSD Disk	8 GB / 4 CPUs, 160 GB SSD Disk	320 GI

Choose Authentication Method

- SSH Key — Connect to your Droplet with an SSH key pair
- Password — Connect to your Droplet as the "root" user via password

Choose your SSH keys

- [x] xyushman

New SSH Key

We recommend these options

- [ ] **Add improved metrics monitoring and alerting (free)**
  Collect and graph expanded system-level metrics, track performance, and set up alerts instantly within the control panel.
- [ ] **Add a worry-free Managed Database (+$15.00)**

— Advanced Options

- [x] Enable IPv6 (free)
  Enables public IPv6 networking
- [ ] Add Initialization scripts (free)
  Add scripts to run on initial droplet boot up - great for repetitive or initialization tasks

Finalize Details

Quantity
Deploy multiple Droplets with the same configuration.

— 1 Droplet +

Hostname
Give your Droplets an identifying name you will remember them by.

xyushman

Tags

xyushman-Droplet Type tags here

Project

first-project

$12.00/month
$0.018/hour

CREATE VIA COMMAND LINE | Create Droplet

NOTE :

Please Note down your Hostename and save your ssh key

Hostname : xyushman

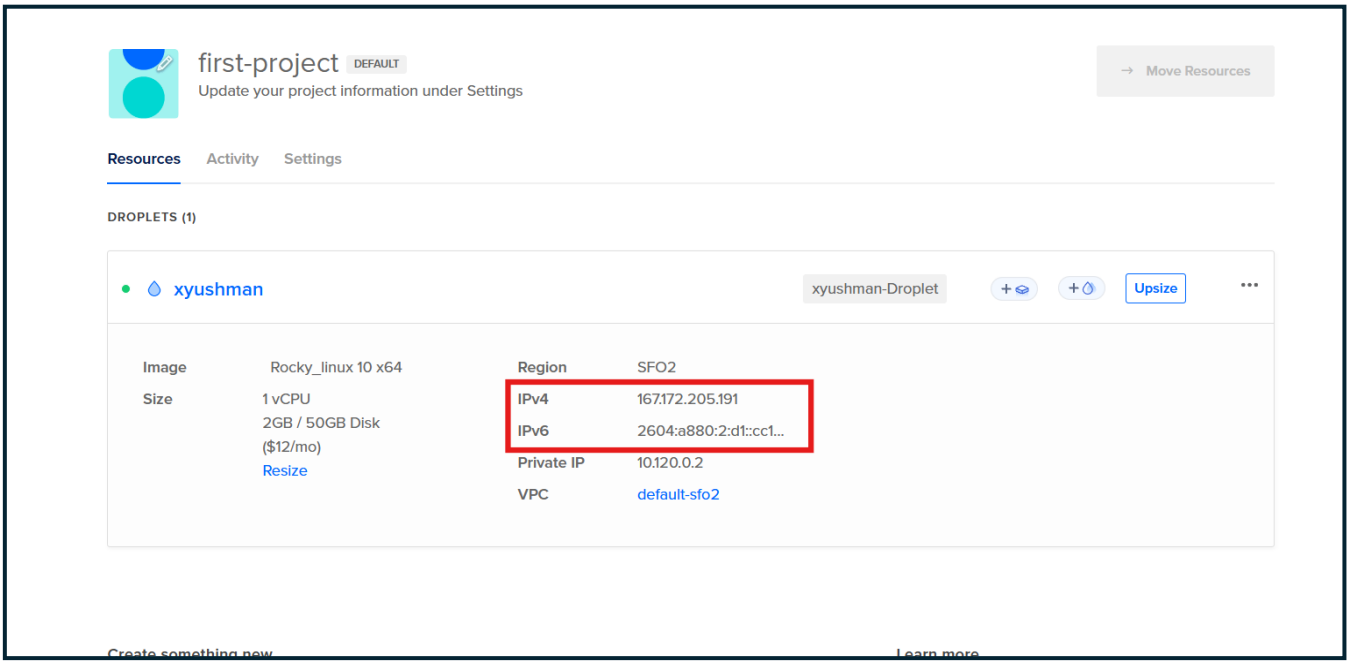
B. Initial Server Hardening:

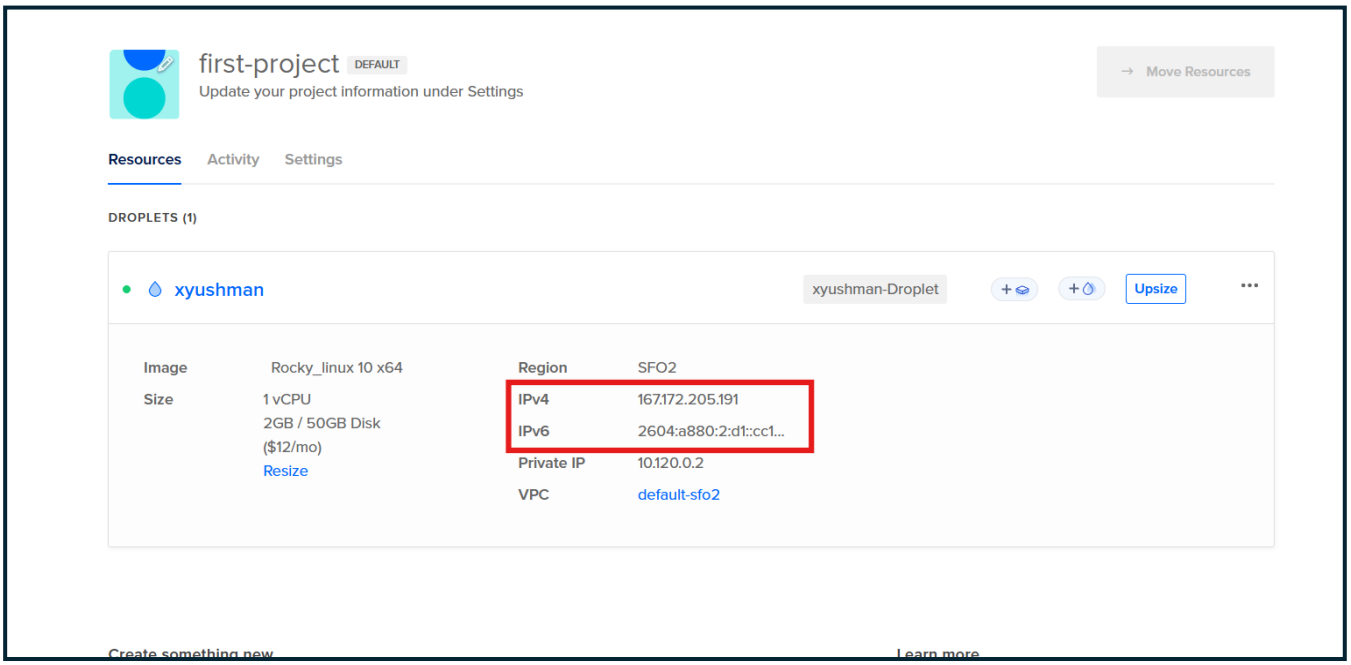
Connect to your new Droplet as the root user.

ssh root@your_droplet_ip

eg : ssh root@167.172.205.191

NOTE : This will not work for you have to use your own ip





Create a new user and grant administrative privileges.

```
# Create the user and set a strong password
adduser your_username
passwd your_username

# Add the user to the 'wheel' group to grant sudo access
usermod -aG wheel your_username
```

Copy your SSH key to the new user to allow direct SSH access.

```
#Install Text editor nano
dnf install nano
# Copy SSH key for passwordless login
rsync --archive --chown=your_username:your_username ~/.ssh /home/your_username
```

```
[root@xyushman ~]# adduser xyushman
useradd: user 'xyushman' already exists
[root@xyushman ~]# passwd xyushman
New password:
Retype new password:
passwd: password updated successfully
[root@xyushman ~]# usermod -aG wheel xyushman
[root@xyushman ~]# rsync --archive --chown=xyushman:xyushman ~/.ssh /home/xyushman
[root@xyushman ~]# nano /etc/ssh/sshd_config
-bash: nano: command not found
[root@xyushman ~]# apt install nano
-bash: apt: command not found
[root@xyushman ~]# dnf install nano
```

Disable root SSH login for better security. Edit **/etc/ssh/sshd_config** and change **PermitRootLogin yes** to **PermitRootLogin no.**

```
# Apply the change
sudo systemctl restart sshd
```

```
  GNU nano 8.1                                    /etc/ssh/sshd_config
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none

^G Help      ^O Write Out  ^F Where Is   ^K Cut      ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste    ^J Justify    ^/ Go To Line
```

Type exit to log out

Log out and **log back** in as your new user: ssh your_username@your_droplet_ip

Eg : ssh xyushman@167.172.205.191

C. Configure the Firewall:

#Switch to root user
sudo -i

# Enable and start the firewall
sudo systemctl enable --now firewalld

#Error
Failed to enable unit: Unit firewalld.service does not exist
dnf install firewalld -y
sudo systemctl enable --now firewalld
sudo systemctl status firewalld

# Allow essential services permanently
sudo firewall-cm`d --permanent --add-service=http
sudo firewall-cmd --permanent --add-service=https
sudo firewall-cmd --permanent --add-port=8080/tcp # For Keycloak
sudo firewall-cmd --permanent --add-service=ssh

# Apply the new rules
sudo firewall-cmd –reload

```
[root@xyushman ~]# sudo systemctl enable --now firewalld
[root@xyushman ~]# sudo systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-09-10 17:40:19 UTC; 8s ago
 Invocation: f22ea0a1759a4e62950cf0e3c1a0330f
       Docs: man:firewalld(1)
   Main PID: 25673 (firewalld)
      Tasks: 2 (limit: 10886)
     Memory: 26.6M (peak: 26.8M)
        CPU: 590ms
     CGroup: /system.slice/firewalld.service
             └─25673 /usr/bin/python3 -sP /usr/sbin/firewalld --nofork --nopid
[root@xyushman ~]# sudo firewall-cmd --permanent --add-service=http
success
[root@xyushman ~]# sudo firewall-cmd --permanent --add-service=https
success
[root@xyushman ~]# sudo firewall-cmd --permanent --add-port=8080/tcp # For Keycloak
success
[root@xyushman ~]# sudo firewall-cmd --permanent --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
[root@xyushman ~]# sudo firewall-cmd --reload
success
[root@xyushman ~]#
```

D. **Update System & Install Core Components:**

```
# Update all system packages
sudo dnf update -y

# Install EPEL and Remi repositories for up-to-date packages
sudo dnf install epel-release -y
sudo dnf install https://rpms.remirepo.net/enterprise/remi-release-10.rpm -y
sudo dnf module enable php:remi-8.3 -y

# Install Apache, PHP, MariaDB, Python, and other tools
sudo dnf install httpd php php-cli php-mysqlnd php-gd php-xml php-mbstring php-json php-fpm
mariadb-server python3 python3-pip unzip wget -y

# Enable and start core services
sudo systemctl enable --now httpd
sudo systemctl enable --now php-fpm
sudo systemctl enable --now mariadb

# Secure your database installation
sudo mysql_secure_installation
```

mysql_secure_installation Options & What to Set

1. **Enter current password for root**
   - If just installed: **Press Enter**
   - If password already set: **Enter it**
2. **Switch to unix_socket authentication [Y/n]**
   - **Development:** n (so you can log in with password easily)
   - **Production:** Y (root only logs in via system root user, much safer)
3. **Change the root password? [Y/n]**
   - **Development:** n (if already set and you remember it)
   - **Production:** Y (set a **very strong password**)
4. **Remove anonymous users? [Y/n]**
   - **Development:** y
   - **Production:** y (absolutely required!)
5. **Disallow root login remotely? [Y/n]**
   - **Development:** n (if you want to connect remotely for testing)
   - **Production:** Y (never allow root over the network — big security risk!)

(Instead: create a new admin user, e.g., admin@'your_ip' with proper privileges.)

6. **Remove test database and access to it? [Y/n]**
   - **Development:** n (can be useful for quick testing)
   - **Production:** Y (removes unnecessary exposure)
7. **Reload privilege tables now? [Y/n]**
   - **Development:** Y (always good to apply changes immediately)
   - **Production:** Y (mandatory)

```
You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
 ... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
 ... skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] n
 ... skipping.

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
[xyushman@xyushman ~]$
```

2. Keycloak Installation & Configuration

A. Install Java & Keycloak:

# Install Java JDK 17
sudo dnf install java-17-openjdk-devel -y
(**Rocky 10's** official repositories have moved forward, **Java 21** is the current default OpenJDK version available. Let move on with Java 21 if we face error we will manually install **Java17**)
sudo dnf install java-21-openjdk-devel -y

Navigate to /opt for the installation
cd /opt

# Download the latest stable Keycloak (check keycloak.org/downloads for new versions)
sudo wget https://github.com/keycloak/keycloak/releases/download/24.0.4/keycloak-24.0.4.zip
sudo unzip keycloak-24.0.4.zip
sudo mv keycloak-24.0.4 keycloak

# Create a dedicated user for Keycloak
sudo groupadd keycloak
sudo useradd -r -g keycloak -d /opt/keycloak -s /sbin/nologin keycloak
sudo chown -R keycloak:keycloak /opt/keycloak

```
r
  inflating: keycloak-24.0.4/lib/lib/main/com.github.ua-parser.uap-java-1.5.4.jar
  inflating: keycloak-24.0.4/lib/lib/main/org.keycloak.keycloak-themes-24.0.4.jar
  inflating: keycloak-24.0.4/lib/lib/deployment/io.quarkus.quarkus-hibernate-orm-deployment-spi-3.8.4.jar
  inflating: keycloak-24.0.4/lib/lib/deployment/io.quarkus.quarkus-arc-deployment-3.8.4.jar
  inflating: keycloak-24.0.4/lib/lib/deployment/io.quarkus.quarkus-jdbc-mariadb-deployment-3.8.4.jar
  inflating: keycloak-24.0.4/lib/lib/deployment/io.quarkus.quarkus-agroal-deployment-3.8.4.jar
  inflating: keycloak-24.0.4/lib/lib/deployment/io.quarkus.quarkus-class-change-agent-3.8.4.jar
  inflating: keycloak-24.0.4/lib/lib/deployment/io.quarkus.quarkus-mutiny-deployment-3.8.4.jar
  inflating: keycloak-24.0.4/lib/lib/deployment/io.quarkus.quarkus-core-deployment-3.8.4.jar
  inflating: keycloak-24.0.4/lib/lib/deployment/io.quarkus.quarkus-kubernetes-service-binding-spi-3.8.4.jar
  inflating: keycloak-24.0.4/lib/lib/deployment/io.quarkus.quarkus-builder-3.8.4.jar
  inflating: keycloak-24.0.4/lib/lib/deployment/org.graalvm.sdk.word-23.1.2.jar
  inflating: keycloak-24.0.4/lib/lib/deployment/io.quarkus.quarkus-micrometer-registry-prometheus-deployment-3
.8.4.jar
  inflating: keycloak-24.0.4/lib/lib/deployment/io.quarkus.quarkus-caffeine-deployment-3.8.4.jar
  inflating: keycloak-24.0.4/lib/app/keycloak.jar
  inflating: keycloak-24.0.4/lib/quarkus/transformed-bytecode.jar
  inflating: keycloak-24.0.4/lib/quarkus-run.jar
  inflating: keycloak-24.0.4/conf/README.md
  inflating: keycloak-24.0.4/conf/keycloak.conf
  inflating: keycloak-24.0.4/LICENSE.txt
[xyushman@xyushman opt]$ sudo mv keycloak-24.0.4 keycloak
[xyushman@xyushman opt]$ sudo groupadd keycloak
[xyushman@xyushman opt]$ sudo useradd -r -g keycloak -d /opt/keycloak -s /sbin/nologin keycloak
[xyushman@xyushman opt]$ sudo chown -R keycloak:keycloak /opt/keycloak
[xyushman@xyushman opt]$
```

B. Configure and Run Keycloak:

```
# Start Keycloak to create an initial admin user
# The --auto-build flag is for development; we'll create a service for production
export KEYCLOAK_ADMIN=admin
export KEYCLOAK_ADMIN_PASSWORD=q4tdqs7a
/opt/keycloak/bin/kc.sh start-dev --http-port=8080
```

Access http://your_droplet_ip:8080, create an admin user, and log in to the **Administration Console**.

Eg : http://167.172.205.191:8080

C. Create a Systemd Service for Production:

sudo nano /etc/systemd/system/keycloak.service

Paste in:

```
[Unit]
Description=Keycloak Authorization Server
After=network.target

[Service]
Type=idle
User=keycloak
Group=keycloak
ExecStart=/opt/keycloak/bin/kc.sh start --optimized --http-port=8080
LimitNOFILE=102400
LimitNPROC=102400
TimeoutStartSec=600
Restart=on-failure
RestartSec=30

[Install]
WantedBy=multi-user.target
```

Save and exit (CTRL+O, CTRL+X).

Enable and start the service:

```
sudo systemctl daemon-reload
sudo systemctl enable --now keycloak
sudo systemctl status keycloak
```

```
[xyushman@xyushman ~]$ sudo nano /etc/systemd/system/keycloak.service
[xyushman@xyushman ~]$ sudo systemctl daemon-reload
[xyushman@xyushman ~]$ sudo systemctl enable --now keycloak
[xyushman@xyushman ~]$ sudo systemctl status keycloak
● keycloak.service - Keycloak Authorization Server
     Loaded: loaded (/etc/systemd/system/keycloak.service; enabled; preset: disabled)
     Active: active (running) since Thu 2025-09-11 11:15:52 UTC; 2min 12s ago
 Invocation: 8a9b5670e9a24403ba9b69a542adb8d7
   Main PID: 17715 (java)
      Tasks: 41 (limit: 10429)
     Memory: 291.9M (peak: 294M)
        CPU: 24.163s
     CGroup: /system.slice/keycloak.service
             └─17715 java -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m -Dfile.encoding=UTF-8 -Dsun.stdout.>

Sep 11 11:16:07 xyushman kc.sh[17715]: 2025-09-11 11:16:02,659 INFO  [org.jgroups.protocols.pbcast.GMS] (keyc>
Sep 11 11:16:07 xyushman kc.sh[17715]: 2025-09-11 11:16:02,699 INFO  [org.infinispan.CLUSTER] (keycloak-cache>
Sep 11 11:16:07 xyushman kc.sh[17715]: 2025-09-11 11:16:02,792 INFO  [org.infinispan.CLUSTER] (keycloak-cache>
Sep 11 11:16:07 xyushman kc.sh[17715]: 2025-09-11 11:16:02,852 WARN  [org.infinispan.CONFIG] (keycloak-cache->
Sep 11 11:16:07 xyushman kc.sh[17715]: 2025-09-11 11:16:07,375 WARN  [io.quarkus.agroal.runtime.DataSources]  >
Sep 11 11:16:12 xyushman kc.sh[17715]: 2025-09-11 11:16:12,702 INFO  [org.keycloak.connections.infinispan.Def>
Sep 11 11:16:12 xyushman kc.sh[17715]: 2025-09-11 11:16:12,716 INFO  [org.keycloak.broker.provider.AbstractId>
Sep 11 11:16:15 xyushman kc.sh[17715]: 2025-09-11 11:16:15,986 INFO  [io.quarkus] (main) Keycloak 24.0.4 on J>
Sep 11 11:16:15 xyushman kc.sh[17715]: 2025-09-11 11:16:15,989 INFO  [io.quarkus] (main) Profile prod activat>
Sep 11 11:16:15 xyushman kc.sh[17715]: 2025-09-11 11:16:15,989 INFO  [io.quarkus] (main) Installed features: >
```

Error :: HTTPS required

Just ssh -L 9080:localhost:8080 xyushman@167.172.205.191 and Sign up

Page 2: Application Deployment & SSO Integration

1. **Drupal 11 Setup & SSO**
   A. **Create Database:**

```
-- Log in to MariaDB
sudo mysql -u root -p

-- Create database and user
CREATE DATABASE drupaldb CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
CREATE USER 'drupaluser'@'localhost' IDENTIFIED BY 'q4tdqs7a';
GRANT ALL PRIVILEGES ON drupaldb.* TO 'drupaluser'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

```
[xyushman@xyushman ~]$ sudo mysql -u root -p
[sudo] password for xyushman:
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.11.11-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE drupaldb CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
Query OK, 1 row affected (0.004 sec)

MariaDB [(none)]> CREATE USER 'drupaluser'@'localhost' IDENTIFIED BY 'q4tdqs7a';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON drupaldb.* TO 'drupaluser'@'localhost';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> EXIT;
Bye
[xyushman@xyushman ~]$
```

B. Install Drupal & Configure Apache:

Install Drupal using Composer for better dependency management

cd /var/www/

sudo dnf install composer -y

sudo composer create-project drupal/recommended-project drupal

sudo chown -R apache:apache /var/www/drupal

sudo chmod -R 755 /var/www/drupal/web

```
  - Copy [web-root]/themes/README.txt from assets/scaffold/files/themes.README.txt
  * Homepage: https://www.drupal.org/project/drupal
  * Support:
    * docs: https://www.drupal.org/docs/user_guide/en/index.html
    * chat: https://www.drupal.org/node/314178
No security vulnerability advisories found.

  Congratulations, you've installed the Drupal codebase
  from the drupal/recommended-project template!


Next steps:
  * Install the site: https://www.drupal.org/docs/installing-drupal
  * Read the user guide: https://www.drupal.org/docs/user_guide/en/index.html
  * Get support: https://www.drupal.org/support
  * Get involved with the Drupal community:
      https://www.drupal.org/getting-involved
  * Remove the plugin that prints this message:
      composer remove drupal/core-project-message
  * Homepage: https://www.drupal.org/project/drupal
  * Support:
    * docs: https://www.drupal.org/docs/user_guide/en/index.html
    * chat: https://www.drupal.org/node/314178
[xyushman@xyushman www]$ sudo chown -R apache:apache /var/www/drupal
[sudo] password for xyushman:
[xyushman@xyushman www]$ sudo chmod -R 755 /var/www/drupal/web
[xyushman@xyushman www]$
```

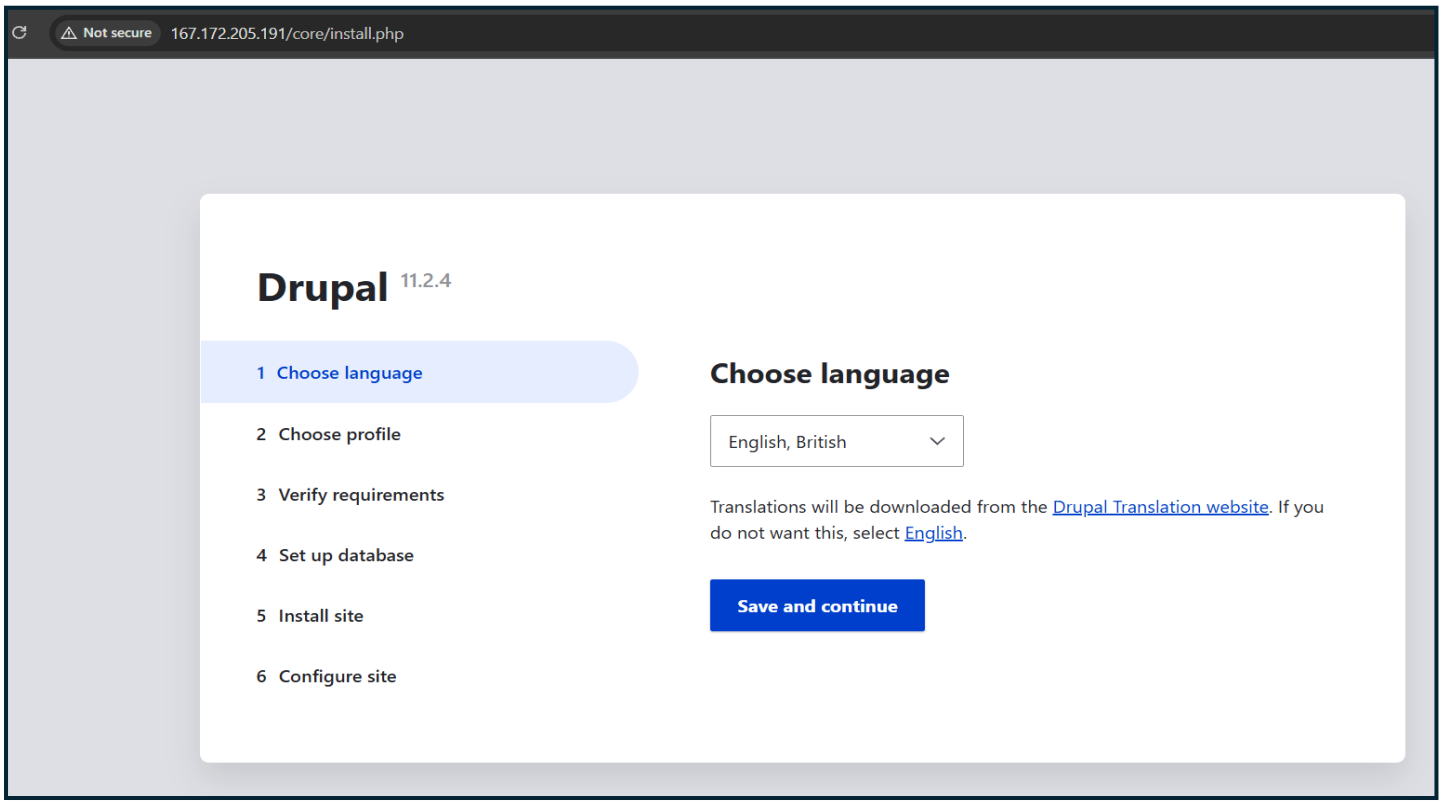
Create an Apache virtual host at **/etc/httpd/conf.d/drupal.conf:**

```
<VirtualHost *:80>
  ServerName your_drupal_domain.com
  DocumentRoot /var/www/drupal/web

  <Directory /var/www/drupal/web>
    AllowOverride All
    Require all granted
  </Directory>

  ErrorLog /var/log/httpd/drupal_error.log
  CustomLog /var/log/httpd/drupal_access.log combined
</VirtualHost>
```

Restart Apache: sudo systemctl restart httpd. Then,
navigate http://your_drupal_domain.com to complete the web installation.



If you get some error missing files/ directory and settings.php

1. **Create the files directory**

```
sudo mkdir -p /var/www/drupal/web/sites/default/files
sudo chown -R apache:apache /var/www/drupal/web/sites/default/files
sudo chmod -R 775 /var/www/drupal/web/sites/default/files
```

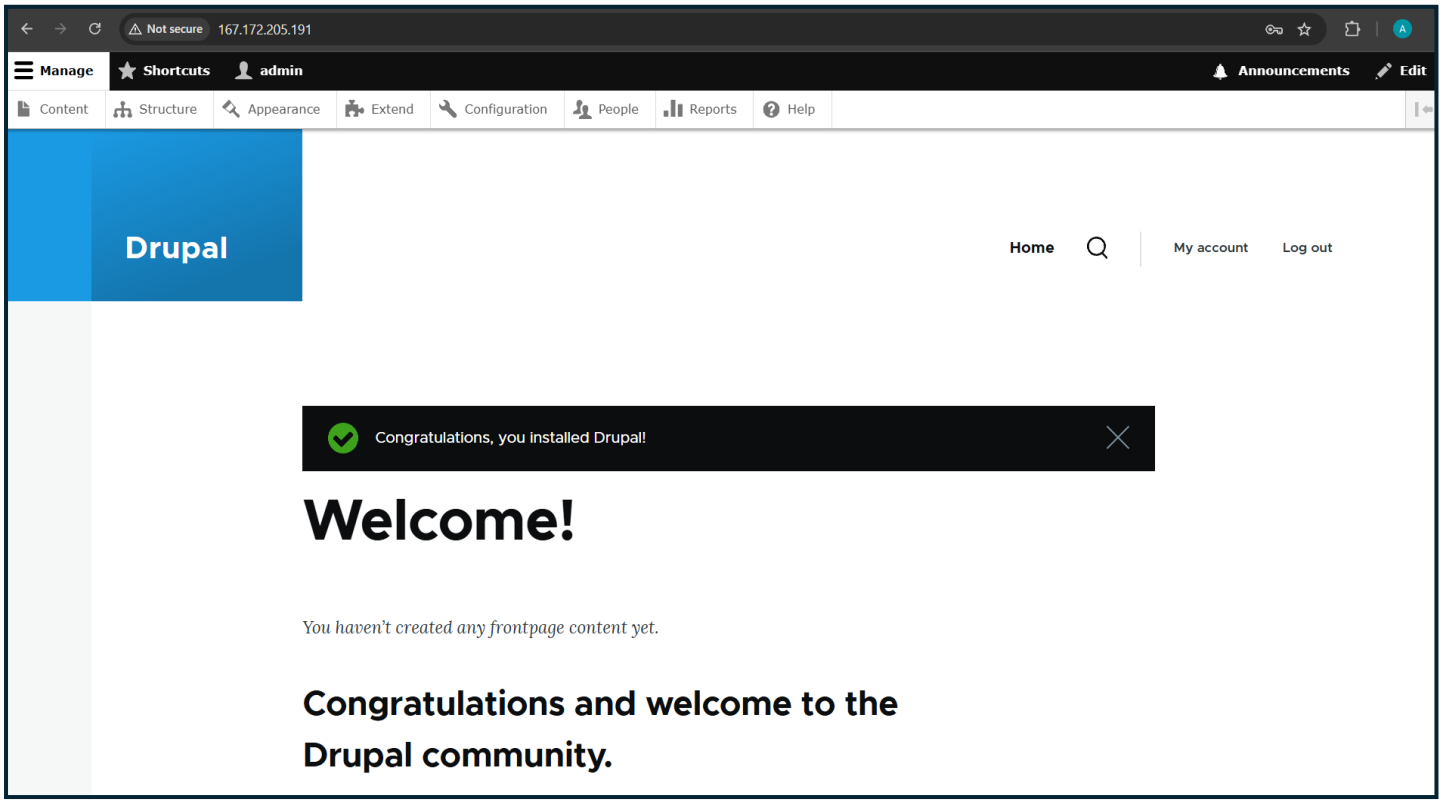
2. **Copy and set up the settings.php file**

```
cd /var/www/drupal/web/sites/default
sudo cp default.settings.php settings.php
sudo chown apache:apache settings.php
sudo chmod 664 settings.php
```

3. **SELinux (only if it's enabled on Rocky Linux)(**If SELinux is enforcing, you need to give Apache write permissions**)**

```
sudo chcon -R -t httpd_sys_rw_content_t /var/www/drupal/web/sites/default/files
sudo chcon -t httpd_sys_rw_content_t /var/www/drupal/web/sites/default/settings.php
sudo systemctl restart httpd
```

Username : admin
Password : q4tdqs7a



C. Integrate Keycloak SSO:

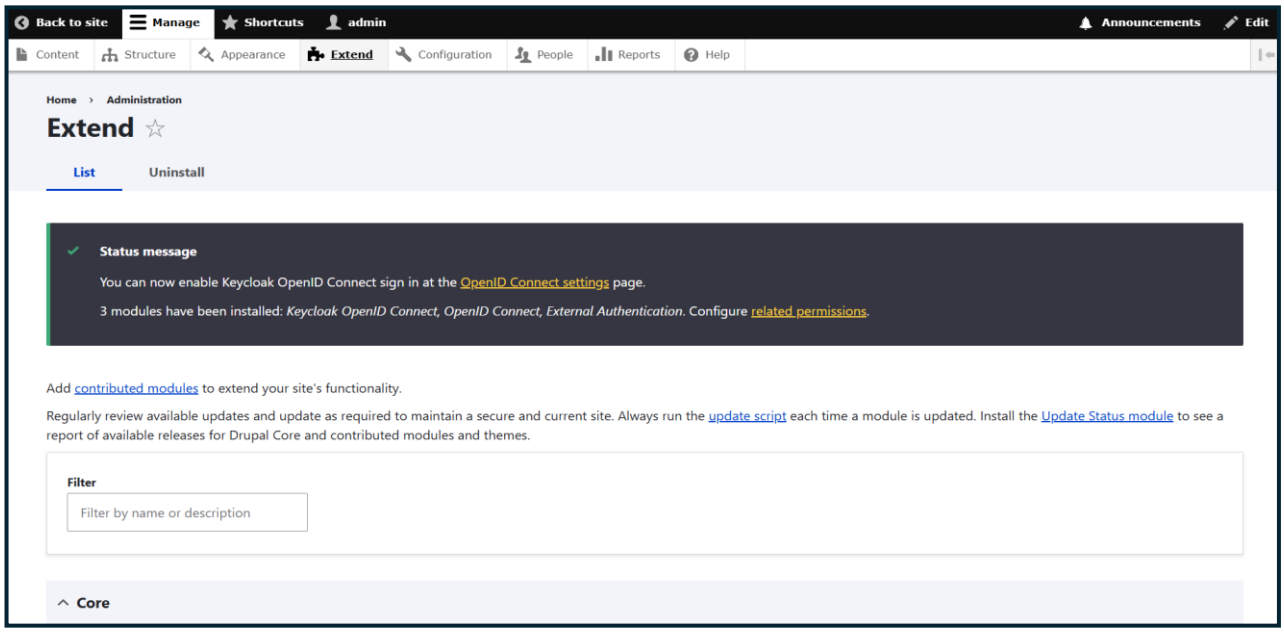
i. **Install Module**: In your Drupal directory (/var/www/drupal), run:

- sudo composer require drupal/keycloak

Error : The error occurs because the Keycloak module for Drupal has dependencies that require a higher minimum stability than your current Composer configuration allows.

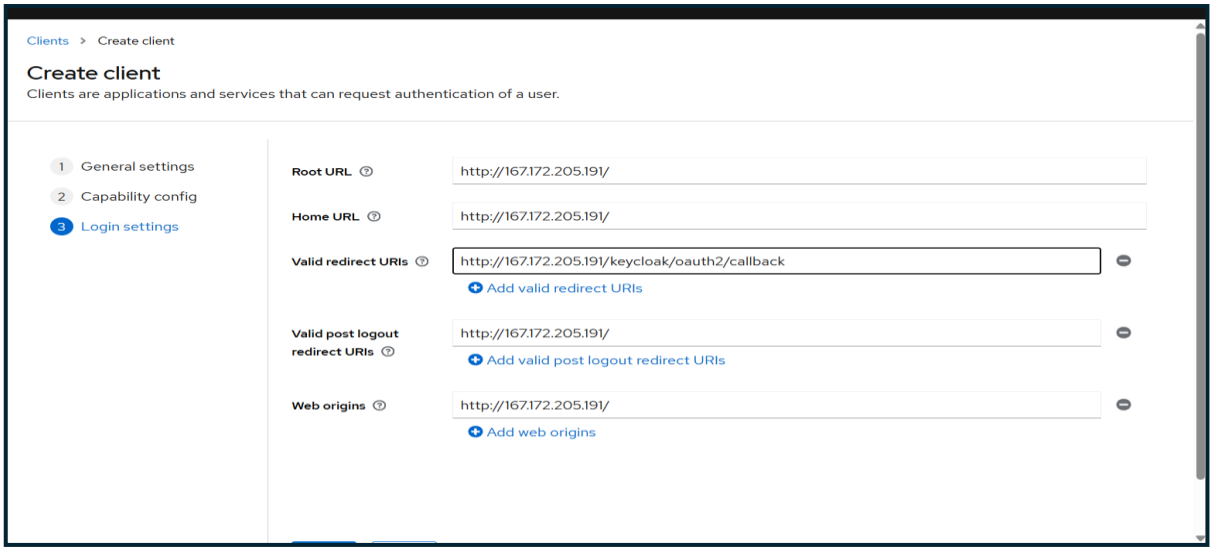
```
composer config minimum-stability dev
composer config prefer-stable true
composer require drupal/keycloak:2.2.x-dev --dev --with-all-dependencies
```

ii. **Enable Module:** In the Drupal admin UI, go to Extend and enable the Keycloak module.



iii. **Create Keycloak Client:**

- In the Keycloak Admin Console, go to Clients > Create client.
- Client ID: drupal
- Client authentication: ON
- Valid redirect URIs: http://your_drupal_domain.com/keycloak/oauth2/callback
- Web origins: http://your_drupal_domain.com
- Save, then copy the Client Secret from the Credentials tab.



iv. **Configure Drupal Module:**

- In Drupal, go to Configuration > Web services > Keycloak. (I didn't find it here Go to **Configuration → People → OpenID Connect → Clients**)
- Enter your Keycloak URL (http://your_droplet_ip:8080), Realm (master), Client ID, and Client Secret.
- Redirect URL (http://167.172.205.191/openid-connect/keycloak) change to this

2. Django Project Setup & SSO

A. Create Database:

```
sudo mysql -u root -p
CREATE DATABASE djangodb;
CREATE USER 'djangouser'@'localhost' IDENTIFIED BY 'q4tdqs7a';
GRANT ALL PRIVILEGES ON djangodb.* TO 'djangouser'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

```
[xyushman@xyushman default]$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 163
Server version: 10.11.11-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE djangodb;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> CREATE USER 'djangouser'@'localhost' IDENTIFIED BY 'another_secure_password';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> DROP USER 'djangouser'@'localhost';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> CREATE USER 'djangouser'@'localhost' IDENTIFIED BY 'q4tdqs7a';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON djangodb.* TO 'djangouser'@'localhost';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> EXIT;
Bye
```

B. Set Up Django Project with Gunicorn:

```
# Create project directory and virtual environment
sudo mkdir /var/www/django_project
sudo chown your_username:your_username /var/www/django_project
sudo dnf install gcc python3-devel mariadb-devel pkg-config -y
cd /var/www/django_project
python3 -m venv venv
source venv/bin/activate

# Install Django, Gunicorn, and a robust OIDC library
pip install django gunicorn mozilla-django-oidc mysqlclient

# Create project and configure database in settings.py
django-admin startproject mysite .
# ... edit mysite/settings.py to configure your MariaDB database ...
nano mysite/settings.py
```

```
  GNU nano 8.1                                   mysite/settings.py
        },
    },
]

WSGI_APPLICATION = 'mysite.wsgi.application'


# Database
# https://docs.djangoproject.com/en/5.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'djangodb',
        'USER': 'djangouser',
        'PASSWORD': 'q4tdqs7a',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}


# Password validation
# https://docs.djangoproject.com/en/5.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [

^G Help      ^O Write Out  ^F Where Is   ^K Cut      ^T Execute    ^C Location   M-U Unc
^X Exit      ^R Read File  ^\ Replace    ^U Paste    ^J Justify    ^/ Go To Line M-E Rec
```

python manage.py migrate

python manage.py createsuperuser

Username : xyushman

Password : q4tdqs7a

Deactivate

```
(venv) [xyushman@xyushman django_project]$ django-admin startproject mysite .
(venv) [xyushman@xyushman django_project]$ nano mysite/settings.py
(venv) [xyushman@xyushman django_project]$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(venv) [xyushman@xyushman django_project]$ python manage.py createsuperuser
Username (leave blank to use 'xyushman'):
Email address:
Password:
Password (again):
Superuser created successfully.
(venv) [xyushman@xyushman django_project]$ deactivate
```

C. Configure Apache as a Reverse Proxy:

Create /etc/httpd/conf.d/django.conf: (sudo nano /etc/httpd/conf.d/django.conf)

```
<VirtualHost *:8000>
  ServerName 167.172.205.191
  ProxyPass / http://127.0.0.1:8000/
  ProxyPassReverse / http://127.0.0.1:8000/
  ErrorLog /var/log/httpd/django_error.log
  CustomLog /var/log/httpd/django_access.log combined
</VirtualHost>
```

For this to work, you'll run Gunicorn listening on port 8000. Create a systemd service for Gunicorn to manage it properly. Restart Apache after setup.

pip install gunicorn

1. **Create a Gunicorn systemd service**
   Create file: sudo nano /etc/systemd/system/gunicorn.service
2. **Add this configuration (adjust paths and usernames where needed**

```
[Unit]
Description=Gunicorn daemon for Django project
After=network.target

[Service]
User=xyushman
Group=xyushman
```

```
WorkingDirectory=/var/www/django_project
ExecStart=/var/www/django_project/venv/bin/gunicorn \
    --access-logfile - \
    --workers 3 \
    --bind 127.0.0.1:8000 \
    mysite.wsgi:application

Restart=always

[Install]
WantedBy=multi-user.target
```

3. **Reload and Start Service**

```
sudo systemctl daemon-reload
sudo systemctl restart gunicorn
sudo systemctl status gunicorn
```

```
[xyushman@xyushman ~]$ sudo systemctl daemon-reload
sudo systemctl restart gunicorn
sudo systemctl status gunicorn
● gunicorn.service - Gunicorn daemon for Django project
     Loaded: loaded (/etc/systemd/system/gunicorn.service; enabled; preset: disabled)
     Active: active (running) since Thu 2025-09-11 16:11:54 UTC; 173ms ago
 Invocation: 4e7b7bbb16eb48c0a402d98136ac5c08
   Main PID: 23317 (gunicorn)
      Tasks: 1 (limit: 10429)
     Memory: 2.9M (peak: 3.1M)
        CPU: 32ms
     CGroup: /system.slice/gunicorn.service
             └─23317 /var/www/django_project/venv/bin/python3 /var/www/django_project/venv/bin/gunic

Sep 11 16:11:54 xyushman systemd[1]: Started gunicorn.service - Gunicorn daemon for Django project.
lines 1-12/12 (END)
```

D. **Integrate Keycloak SSO with mozilla-django-oidc:**

1. Create **Keycloak Client:**
   - **Client ID:** django
   - **Valid redirect URIs**: http://your_django_domain.com/oidc/callback/
   - Copy the **Client Secret.**
2. Configure Django **(settings.py):**

```
# settings.py
INSTALLED_APPS = [
    # ...
    'mozilla_django_oidc',
]
AUTHENTICATION_BACKENDS = [
    'mozilla_django_oidc.auth.OIDCAuthenticationBackend',
    'django.contrib.auth.backends.ModelBackend',
]
# Keycloak OIDC configuration
OIDC_RP_CLIENT_ID = "django"
OIDC_RP_CLIENT_SECRET = "lAFTzIbo1e7L9rQsB6zHmSkCW4n2iR6F"
```

```
OIDC_OP_AUTHORIZATION_ENDPOINT =
"http://167.172.205.191:8080/realms/master/protocol/openid-
connect/auth"
OIDC_OP_TOKEN_ENDPOINT =
"http://167.172.205.191:8080/realms/master/protocol/openid-
connect/token"
OIDC_OP_USER_ENDPOINT =
"http://167.172.205.191:8080/realms/master/protocol/openid-
connect/userinfo"
OIDC_RP_SIGN_ALGO = "RS256"
OIDC_OP_JWKS_ENDPOINT =
"http://167.172.205.191:8080/realms/master/protocol/openid-
connect/certs"

# Redirect URLs after login/logout
LOGIN_REDIRECT_URL = "/"
LOGOUT_REDIRECT_URL = "/"
```

Page 3: PHP Application & Final Submission

1. Generic PHP Project Setup

A. Deploy Application:

Place your PHP application files in a dedicated directory.

```
sudo mkdir /var/www/php_app
# Copy your PHP application files to this directory
sudo chown -R apache:apache /var/www/php_app
sudo chmod -R 755 /var/www/php_app
```

B. Configure Apache:

Create /etc/httpd/conf.d/php_app.conf:

2. PHP Keycloak SSO Integration

We'll use a standard OIDC client library for PHP.

1. **Install Library:**

   cd /var/www/php_app
   sudo composer require jumbojett/openid-connect-php

```
[xyushman@xyushman php_app]$ sudo chmod -R 755 /var/www/php_app
[xyushman@xyushman php_app]$ composer require jumbojett/openid-connect-php
./composer.json has been created
Running composer update jumbojett/openid-connect-php
Loading composer repositories with package information
Updating dependencies
Lock file operations: 4 installs, 0 updates, 0 removals
  - Locking jumbojett/openid-connect-php (v1.0.2)
  - Locking paragonie/constant_time_encoding (v3.0.0)
  - Locking paragonie/random_compat (v9.99.100)
  - Locking phpseclib/phpseclib (3.0.46)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 4 installs, 0 updates, 0 removals
  - Downloading paragonie/random_compat (v9.99.100)
  - Downloading paragonie/constant_time_encoding (v3.0.0)
  - Downloading phpseclib/phpseclib (3.0.46)
  - Downloading jumbojett/openid-connect-php (v1.0.2)
  - Installing paragonie/random_compat (v9.99.100): Extracting archive
  - Installing paragonie/constant_time_encoding (v3.0.0): Extracting archive
  - Installing phpseclib/phpseclib (3.0.46): Extracting archive
  - Installing jumbojett/openid-connect-php (v1.0.2): Extracting archive
4 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating autoload files
1 package you are using is looking for funding.
Use the `composer fund` command to find out more!
No security vulnerability advisories found.
Using version ^1.0 for jumbojett/openid-connect-php
[xyushman@xyushman php_app]$ |
```

2. **Create Keycloak Client:**

   1. **Client ID:** php-app
   2. **Valid redirect URIs:** http://your_php_app_domain.com/callback.php
   3. Copy the **Client Secret**.

3. **Example PHP Code:**

Create a file login.php To initiate the SSO flow:

```php
<?php
require 'vendor/autoload.php';
use Jumbojett\OpenIDConnectClient;

session_start();

$oidc = new OpenIDConnectClient(
    'http://your_droplet_ip:8080/realms/master', // Keycloak provider URL
    'php-app',                                   // Client ID
    'your_client_secret_from_keycloak'           // Client Secret
);

// This triggers the authentication flow
$oidc->authenticate();

// Store user info in the session
$_SESSION['user_info'] = $oidc->requestUserInfo();

// Redirect to a protected page
header("Location: /profile.php");
exit();
```

Create a profile.php To display user data:

```php
<?php
session_start();
if (empty($_SESSION['user_info'])) {
    header("Location: /login.php");
    exit();
}
$userInfo = $_SESSION['user_info'];
echo "<h1>Welcome, " . htmlspecialchars($userInfo->name) . "</h1>";
echo "<p>Email: " . htmlspecialchars($userInfo->email) . "</p>";
?>
```