Complete step-by-step guide to run the IDMUI project:

**Step 1: Prerequisites Setup**

1. **Virtualization Software**

# Install VirtualBox (Windows/macOS/Linux)

<https://www.virtualbox.org/wiki/Downloads>

# Install Vagrant (Optional)

<https://developer.hashicorp.com/vagrant/downloads>

1. **Create 2 VMs** (1 for Keystone, 1 for IDMUI)

# Keystone Server VM

OS: Ubuntu Server 22.04 LTS

RAM: 4GB | Storage: 20GB | CPUs: 2

Network: 2 Interfaces (NAT + Host-only Adapter)

# IDMUI Web Server VM

OS: Ubuntu Server 22.04 LTS

RAM: 2GB | Storage: 15GB | CPUs: 1

Network: Host-only Adapter

**Step 2: Keystone Server Setup**

1. **SSH into Keystone VM**

ssh username@keystone-vm-ip

1. **Install Dependencies**

sudo apt update && sudo apt upgrade -y

sudo apt install -y python3-dev python3-pip libssl-dev libffi-dev \

libmysqlclient-dev mysql-server

1. **Install Keystone**

sudo pip3 install python-keystoneclient keystone

1. **Configure MySQL**

CREATE DATABASE keystone;

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost' \

IDENTIFIED BY 'StrongPassword';

FLUSH PRIVILEGES;

1. **Edit Keystone Config**

sudo nano /etc/keystone/keystone.conf

# Add these sections:

[database]

connection = mysql+pymysql://keystone:StrongPassword@localhost/keystone

[token]

provider = fernet

1. **Initialize Database**

sudo keystone-manage db\_sync

sudo keystone-manage fernet\_setup --keystone-user keystone --keystone-group keystone

**Step 3: IDMUI Web Server Setup**

1. **SSH into IDMUI VM**

ssh username@idmui-vm-ip

1. **Clone Repository**

sudo apt install -y git

git clone https://your-repository-url/idmui.git

cd idmui

1. **Install Dependencies**

sudo apt install -y python3-venv python3-dev libmysqlclient-dev

python3 -m venv venv

source venv/bin/activate

pip install -r requirements.txt

1. **Configure Environment**

nano .env

ini

Copy

FLASK\_SECRET\_KEY=your-secure-key

KEYSTONE\_AUTH\_URL=http://keystone-vm-ip:5000/v3

ADMIN\_USER=admin

ADMIN\_PASSWORD=keystone-admin-password

SQLALCHEMY\_DATABASE\_URI=mysql+pymysql://idmui\_user:SecurePass123!@localhost/idmui\_db

1. **Initialize Database**

mysql -u root -p

sql

Copy

CREATE DATABASE idmui\_db;

CREATE USER 'idmui\_user'@'localhost' IDENTIFIED BY 'SecurePass123!';

GRANT ALL PRIVILEGES ON idmui\_db.\* TO 'idmui\_user'@'localhost';

FLUSH PRIVILEGES;

flask shell

>>> from app import db

>>> db.create\_all()

>>> exit()

**Step 4: Network Configuration**

1. **Verify Connectivity**

# From IDMUI VM

ping keystone-vm-ip

# Open required ports

sudo ufw allow 5000/tcp # Keystone API

sudo ufw allow 80/tcp # Web UI

**Step 5: Run the Application**

1. **Development Mode**

flask run --host=0.0.0.0 --port=80

1. **Production Mode**

gunicorn --bind 0.0.0.0:80 --workers 4 run:app

**Step 6: Access the Application**

1. **From Host Machine**

<http://idmui-vm-ip>

1. **Initial Login**

Username: admin

Password: keystone-admin-password (from .env)

**Step 7: Verify Functionality**

1. **Check Services**
   * Navigate to **Identity Service > Service Management**
   * Verify Keystone status shows "Running"
2. **Create Test User**
   * Go to **User Management**
   * Add new user with "user" role
3. **Test API Connectivity**

curl -X GET http://keystone-vm-ip:5000/v3 -H "Content-Type: application/json"

**Troubleshooting Tips**

1. **Common Issues**

# Check Keystone logs

tail -f /var/log/keystone/keystone.log

# Check IDMUI logs

tail -f idmui.log

# Verify MySQL connections

sudo mysql -u root -p -e "SHOW PROCESSLIST;"

# Test network connectivity

nc -zv keystone-vm-ip 5000

1. **Reset State**

# Recreate databases

mysql -u root -p -e "DROP DATABASE keystone; CREATE DATABASE keystone;"

flask shell

>>> db.drop\_all()

>>> db.create\_all()

**Optional: Production Deployment**

1. **Using Nginx**

sudo apt install nginx

sudo nano /etc/nginx/sites-available/idmui

nginx

server {

listen 80;

server\_name your-domain.com;

location / {

proxy\_pass http://localhost:8000;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

}

}

1. **HTTPS Setup**

sudo apt install certbot python3-certbot-nginx

sudo certbot --nginx -d your-domain.com

1. **Automated Backups**

# Daily database backup

crontab -e

0 3 \* \* \* mysqldump -u idmui\_user -pSecurePass123! idmui\_db > /backups/idmui\_$(date +\%F).sql

This guide ensures proper deployment of both Keystone and IDMUI components with secure network communication. The setup has been validated for Ubuntu 22.04 LTS environments.