VM SETUP

Step by Step Tutorial

---

1. Redis Setup in the VM

```
$ sudo apt-get install lsb-release curl gpg
$ curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/redis-archive-keyring.gpg
$ sudo chmod 644 /usr/share/keyrings/redis-archive-keyring.gpg
$ echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg]
https://packages.redis.io/deb $(lsb_release -cs) main" | sudo tee
/etc/apt/sources.list.d/redis.list
$ sudo apt-get update
$ sudo apt-get install redis
```

> **Info**
>
> Redis Automatic Start with system start

```
    sudo systemctl enable redis-server
    sudo systemctl start redis-server
```

2. Git Setup
   - SSH key setup

```
    ssh-keygen -t ed25519 -C "your_email@example.com"
```

> **Note**
>
> Enter every question leave to default

```
- Add Key to your SSH agent
```shell
    eval "$(ssh-agent -s)"
    ssh-add ~/.ssh/id_ed25519
```
```

Now add the Public key to the Github in settings

```bash
    ssh -T git@github.com
```

> **Note**
>
> Should received Authenticated but does'nt have shell access.

3. Clone the repository.
   - Create a Directory at `/home/azureuser/`

     ```
     mkdir SCM
     ```

   - Now Clone the repository

```
$ cd SCM
$ git clone
git@github.com:SustainableCityManagement/SustainableCityManagement.git
$ cd SustainableCityManagement
```

4. Installing all the VM requirements

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install -y python3 python3-pip python3-venv nginx
```

5. Creation of environment and requirements installation

> **⚠ Warning**
> You cant directly download packages with pip because system is defaulted to apt

```
python3 -m venv venv
source venv/bin/activate

pip install django gunicorn django-prometheus redis
pip install -r requirements.txt
```

6. Install Prometheus

```
wget https://github.com/prometheus/prometheus/releases/download/v3.3.0-rc.0/prometheus-3.3.0-rc.0.linux-amd64.tar.gz

tar xvfz prometheus-3.3.0-rc.0.linux-amd64.tar.gz

sudo mv prometheus-3.3.0-rc.0.linux-amd64 /usr/local/prometheus
sudo useradd --no-create-home --shell /bin/false prometheus


sudo mkdir /etc/prometheus
sudo mkdir /var/lib/prometheus

sudo chown prometheus:prometheus /usr/local/prometheus
sudo chown -R prometheus:prometheus /etc/prometheus
sudo chown -R prometheus:prometheus /var/lib/prometheus

$ sudo cp
/home/azureuser/SCM/SustainableCityManagement/prometheus/prometheus.yml
/etc/prometheus/

#sudo cp -r /usr/local/prometheus/consoles /etc/prometheus
#$ sudo cp -r /usr/local/prometheus/console_libraries /etc/prometheus/
```

6. Creating System Control scripts

```
# prometheus
sudo nano /etc/systemd/system/prometheus.service
```

```
#Script for prometheus.service
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/prometheus/prometheus \
  --config.file=/etc/prometheus/prometheus.yml \
  --storage.tsdb.path=/var/lib/prometheus/ \
  --web.console.templates=/usr/local/prometheus/consoles \
  --web.console.libraries=/usr/local/prometheus/console_libraries

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

```
sudo systemctl daemon-reload
sudo systemctl start prometheus
#check
sudo systemctl status prometheus
```

```
sudo nano /etc/systemd/system/celeryworker.service
```

```
[Unit]
Description=Celery Worker Service
After=network.target

[Service]
User=scm
Group=scm
WorkingDirectory=/home/azureuser/SCM/SustainableCityManagement
Environment="DJANGO_SETTINGS_MODULE=SustainableCityMgmtApp.settings"
ExecStart=/home/azureuser/SCM/SustainableCityManagement/venv/bin/celery -A
apps.city_manager worker --pool=solo --loglevel=info
Restart=always

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

```
sudo systemctl daemon-reload
sudo systemctl start celeryworker
#check
sudo systemctl status celeryworker
```

```
sudo nano /etc/systemd/system/celeryworker.service
```

```
[Unit]
Description=Celery Beat Service
After=network.target

[Service]
User=scm
Group=scm
WorkingDirectory=/home/azureuser/SCM/SustainableCityManagement
Environment="DJANGO_SETTINGS_MODULE=SustainableCityMgmtApp.settings"
ExecStart=/home/scm/SCM/SustainableCityManagement/venv/bin/celery -A
apps.city_manager beat --loglevel=info
Restart=always
```

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

```
sudo systemctl daemon-reload
sudo systemctl start celerybeat
#check
sudo systemctl status celerybeat
```

```
sudo systemctl enable celeryworker
sudo systemctl enable celerybeat
sudo systemctl enable prometheus
```

7. Gunicorn Service

```
sudo nano /etc/systemd/system/gunicorn.service
```

```
[Unit]
Description=gunicorn daemon for Django application
After=network.target
Wants=network-online.target
Requires=network-online.target

[Service]
User=azureuser
Group=azureuser
WorkingDirectory=/home/azureuser/SCM/SustainableCityManagement
ExecStart=/home/azureuser/SCM/SustainableCityManagement/venv/bin/gunicorn \
    --workers 2 \
    --threads 2 \
    --bind 0.0.0.0:8000 \
    --timeout 300 \
    --max-requests 1000 \
    --max-requests-jitter 50 \
    --log-level info \
    --access-logfile - \
    --error-logfile - \
    SustainableCityMgmtApp.wsgi:application

# Restart on crash but with increasing delays to prevent thrashing
Restart=on-failure
RestartSec=5s
StartLimitInterval=600s
StartLimitBurst=5

# Environment variables if needed
# Environment=DEBUG=False
```

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

```
sudo systemctl daemon-reload
sudo systemctl enable gunicorn
sudo systemctl start gunicorn
# status check
sudo systemctl status gunicorn
```

8. Nginx Service:

```
sudo nano /etc/nginx/sites-available/djangoapp
```

```
server {
    # The default_server directive tells Nginx to use this block for any
requests
    # that don't match a specific server_name
    listen 80 default_server;
    listen [::]:80 default_server;

    # This will match any hostname (you don't need to change IPs here)
    server_name _;

    # Increase timeouts for long-running requests
    proxy_connect_timeout 600s;
    proxy_send_timeout 600s;
    proxy_read_timeout 600s;
    send_timeout 600s;

    # Security headers
    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-XSS-Protection "1; mode=block";

    # Main Django application
    location / {
        # This proxies to your Gunicorn server running on port 8000
        proxy_pass http://127.0.0.1:8000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_buffering off;
        proxy_redirect off;
```

```
    }

    # Static files - corrected path with azureuser username
    location /static/ {
        alias /home/azureuser/SCM/SustainableCityManagement/static/;
        expires 30d;
        access_log off;
        add_header Cache-Control "public, max-age=2592000";
    }

    # Media files (if needed) - corrected path with azureuser username
    location /media/ {
        alias /home/azureuser/SCM/SustainableCityManagement/media/;
        expires 30d;
        access_log off;
        add_header Cache-Control "public, max-age=2592000";
    }

    # Favicon handling
    location = /favicon.ico {
        access_log off;
        log_not_found off;
    }

    # Robots.txt handling
    location = /robots.txt {
        access_log off;
        log_not_found off;
    }

    # Deny access to .git, etc
    location ~ /\.(?!well-known).* {
        deny all;
        access_log off;
        log_not_found off;
    }

    # Error pages
    error_page 404 /404.html;
    error_page 500 502 503 504 /50x.html;
}
```

```
# Enable your site
sudo ln -sf /etc/nginx/sites-available/djangoapp /etc/nginx/sites-enabled/

# Remove default site
sudo rm -f /etc/nginx/sites-enabled/default
```

```
# Check for any other enabled sites that might conflict
ls -la /etc/nginx/sites-enabled/
```

```
nano
/home/azureuser/SCM/SustainableCityManagement/SustainableCityMgmtApp/setting
s.py
```

> **Note**
>
> Change the allowed hosts to `*` for testing purposes will set server ip for the production.

```
ALLOWED_HOSTS = [`*`]
```

- Applying all the chnages

```
sudo nginx -t
# we will receive successful and ok then
sudo systemctl restart nginx
```

9. File access permissions to nginix

```
sudo chmod 755 /home/azureuser
sudo chmod 755 /home/azureuser/SCM
sudo chmod 755 /home/azureuser/SCM/SustainableCityManagement
sudo chmod 755 /home/azureuser/SCM/SustainableCityManagement/static
sudo find /home/azureuser/SCM/SustainableCityManagement/static -type d -exec
chmod 755 {} \;
sudo find /home/azureuser/SCM/SustainableCityManagement/static -type f -exec
chmod 644 {} \;
```

10. verifying with logs

> **Important**
> This shows all HTTP requests hitting your Nginx server.

```
sudo tail -f /var/log/nginx/access.log
```

> **Info**

This shows logs from the Gunicorn process serving your Django application.

```
sudo journalctl -fu gunicorn
```

Danger

Do this only if you receive the error of unable to resolve the host ASE: Temporary failure in name resolution

```
sudo nano /etc/hosts
```

```
127.0.0.1 localhost ASE
```