**Python Code Deployment Guide (Flask Project)**

This guide documents step-by-step deployment of a Flask-based Python application on a Linux server using system tools and Gunicorn.

**1. System Update & Package Installation**

apt update -y

apt install git -y

Updates system packages and installs Git to manage source code.

**2. Clone the Application Repository**

git clone https://github.com/daniyel7devops/myprojects1.git

cd redeemx\_be\_app-uat

cd redeemx

Clones your project repository and navigates to the correct sub-directory.

**3. Verify Files & Initial Check**

python3

cat requirements.txt

Lists files in the directory and confirms Python is installed.  
Checks if a requirements.txt file exists for dependencies.

**4. Install Python and Build Tools**

apt install pip3

sudo apt install -y libmariadb-dev gcc g++ build-essential python3-dev dh-python

apt install python3-pip

Installs Python pip and required build tools for compiling Python packages.

**5. Install Python Modules**

pip3 install -r requirements.txt

apt install python3-flask

apt install python3-redis

apt install python3-gunicorn

Installs Flask, Redis, Gunicorn, and other dependencies either via pip or system packages.

**6. Setup Python Virtual Environment**

apt install python3-venv

python3 -m venv myenv

source myenv/bin/activate

Creates and activates a virtual environment to isolate project dependencies.

**7. Run and Debug the Application**

python3 app.py

pip3 install flask

pip3 install redis

pip3 install gunicorn

Runs the Flask application for testing.  
Installs additional missing modules one-by-one based on error messages.

**8. View and Edit Application Files (Optional)**

cat app.py

vi app.py

vi wsgi.py

Views or modifies main application files and Gunicorn WSGI entry point.

**9. Production Deployment Using Gunicorn**

gunicorn app:app -b 0.0.0.0:5000 --log-file - --access-logfile - --workers 4 --keep-alive 0

Deploys the app on port 5000 using Gunicorn with 4 worker processes.

