Odoo Production Deployment Document

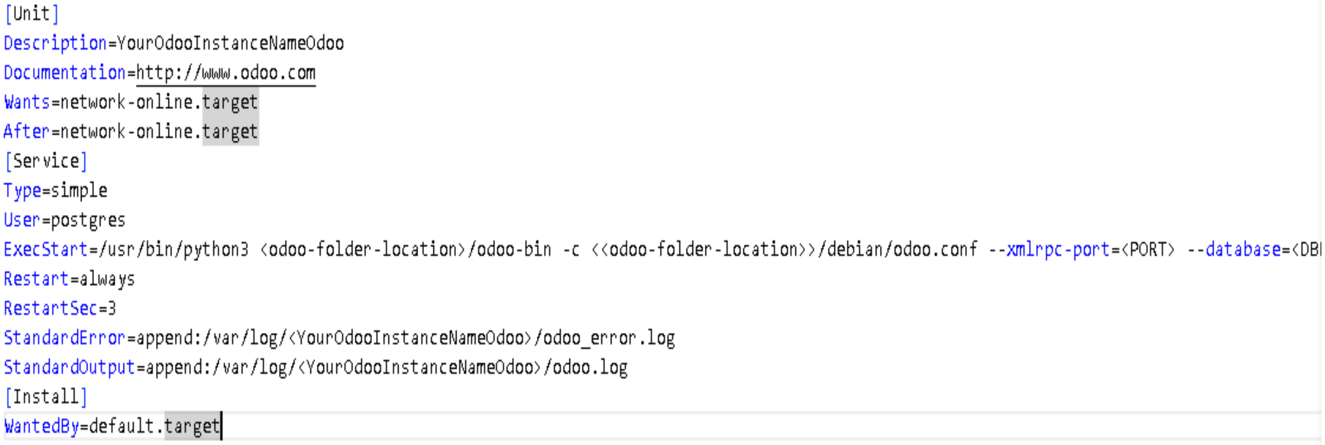
1. Copy the odoo community code to a specific location on server.

2. Install all the python dependency using pip command (pip -r requiremt.txt)

3. Create a separate service for the odoo instance by using below steps
   a. Navigate to location **/etc/systemd/system**

   b. Create a service file using command **nano <yourodooinstancename>.service**

   c. Paste below content on file **<yourodooinstancename>.service**



```
[Unit]
Description=YourOdooInstanceNameOdoo
Documentation=http://www.odoo.com
Wants=network-online.target
After=network-online.target
[Service]
Type=simple
User=postgres
ExecStart=/usr/bin/python3 <odoo-folder-location>/odoo-bin -c <<odoo-folder-location>>/debian/odoo.conf --xmlrpc-port=<PORT> --database=<DB
Restart=always
RestartSec=3
StandardError=append:/var/log/<YourOdooInstanceNameOdoo>/odoo_error.log
StandardOutput=append:/var/log/<YourOdooInstanceNameOdoo>/odoo.log
[Install]
WantedBy=default.target
```

   d. Change file permission using command **sudo chmod 755 <yourodooinstancename>.service**

   e. Change file owner using command **sudo chown root: <yourodooinstancename>.service**

   f. Go the **<odoo-folder-location>/Debian/odoo.conf** and search for logfile key and paste the value **/var/log/< yourodooinstancename>/odoo.log**

   g. Create log folder using command **mkdir -p /var/log/< yourodooinstancename>**

   h. Navigate to location **/var/log/yourodooinstancename** using command **cd /var/log/yourodooinstancename** and create two file using command **touch odoo.log odoo_error.log**

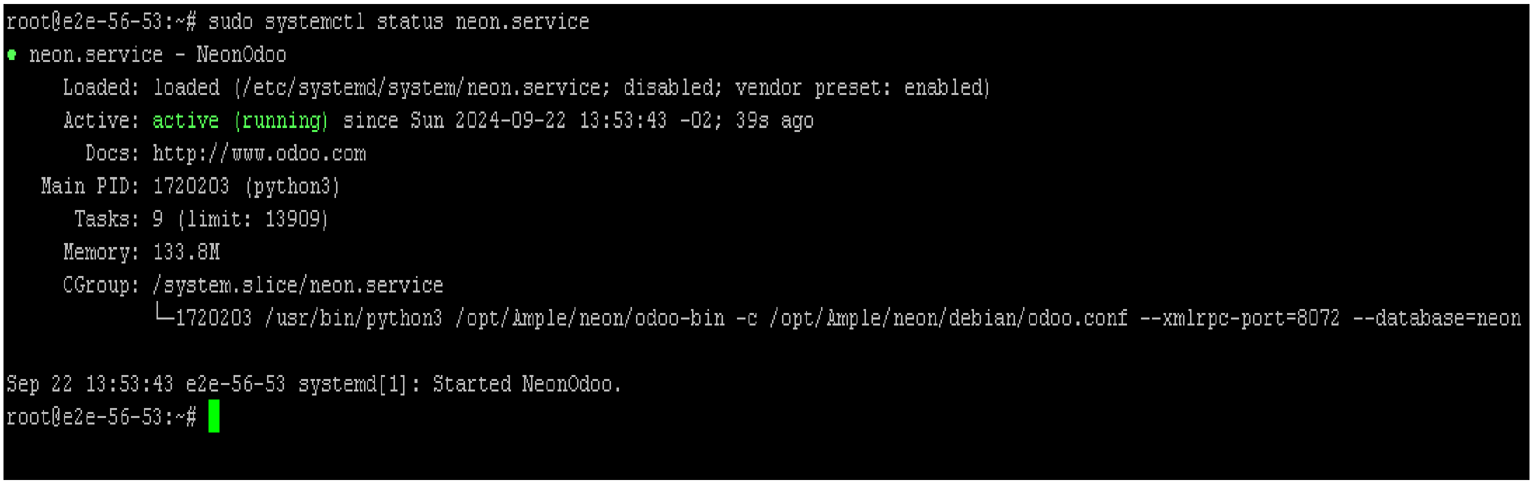
   i. Change the file permission using command **sudo chmod 640 odoo.log odoo_error.log**

   j. Chnage the file owner using command **sudo chown postgres: odoo.log odoo_error.log**

k. Execute the command for service reload **sudo systemctl daemon-reload**

l. Now execute the command **sudo systemctl start <yourodooinstancename>.service**

m. Execute the command **sudo systemctl status <yourodooinstancename>.service** to check if service is running or not. If service is running then you will below response.



```
root@e2e-56-53:~# sudo systemctl status neon.service
● neon.service - NeonOdoo
     Loaded: loaded (/etc/systemd/system/neon.service; disabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-09-22 13:53:43 -02; 39s ago
       Docs: http://www.odoo.com
   Main PID: 1720203 (python3)
      Tasks: 9 (limit: 13909)
     Memory: 133.8M
     CGroup: /system.slice/neon.service
             └─1720203 /usr/bin/python3 /opt/Ample/neon/odoo-bin -c /opt/Ample/neon/debian/odoo.conf --xmlrpc-port=8072 --database=neon

Sep 22 13:53:43 e2e-56-53 systemd[1]: Started NeonOdoo.
root@e2e-56-53:~# 
```

n. If service is not **active(running)** go to location **/var/log/<yourodooinstancename>/odoo_error.log** to check the issue.

o. Execute the command **sudo systemctl stop <yourodooinstancename>.service** to stop the service/instance

p. Execute the command command **sudo systemctl enable <yourodooinstancename>.service** to auto start the service wherever system rebooted.

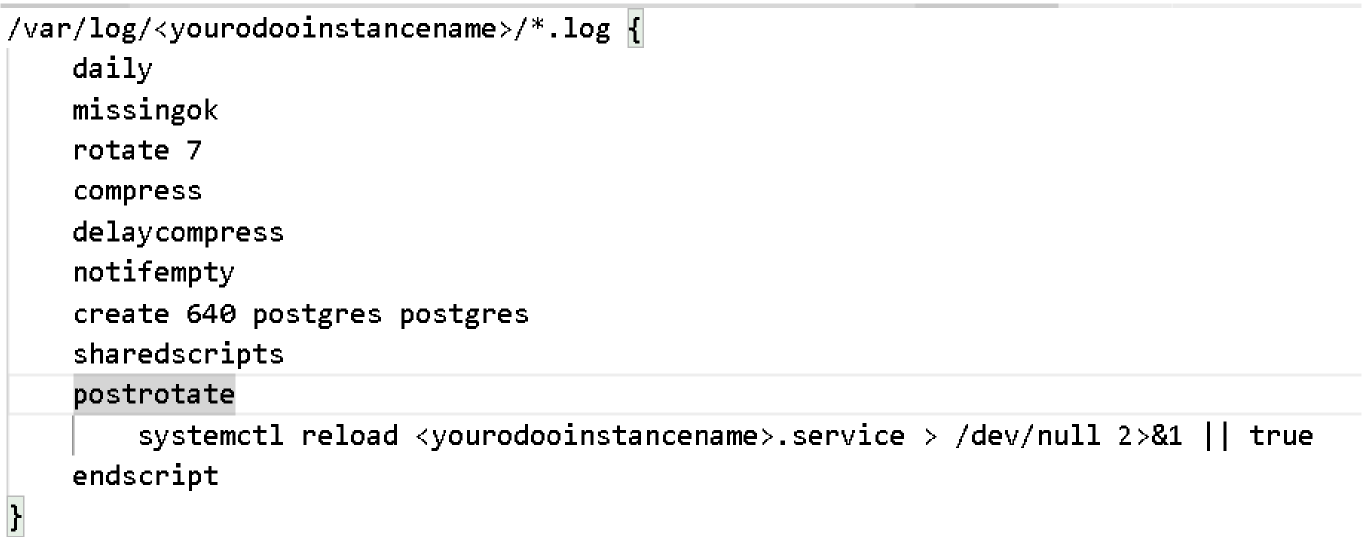
q. Now implement the log rotation for the logs.

Odoo Log Rotation Step

1. Navigate to location **/etc/logrotate.d**

2. Create a file using command **nano <yourodooinstancename>**

3. Paste the below code to the file **<yourodooinstancename>**



```
/var/log/<yourodooinstancename>/*.log {
    daily
    missingok
    rotate 7
    compress
    delaycompress
    notifempty
    create 640 postgres postgres
    sharedscripts
    postrotate
        systemctl reload <yourodooinstancename>.service > /dev/null 2>&1 || true
    endscript
}
```

4. Execute the command **sudo logrotate -d yourodooinstancename** to implement log rotation.

5. Execute the command **sudo logrotate -f yourodooinstancename** to test the log rotation.