UBUNTU (22.04) SERVER SETTING

```
Ssh root@127.0.0.1
initial passwd
```

ROOT PASSWD

```
set new root passwd:
sudo passwd root
```

SSH

```
vim /root/.ssh/authorized_keys
paste id_rsa.pub  inside
vim ~/.ssh/authorized_keys
paste id_rsa.pub  inside

update ssh port:
vim /etc/ssh/sshd_config

change the port & password authentication:
Port 30
...
PasswordAuthentication no

sshd -T | grep passwordauthentication

if passwordauthentication yes:
rm /etc/ssh/sshd_config.d/*


restart the deamon:
systemctl restart ssh
service ssh restart

change the port and verify:
ss -tlpn| grep ssh
netstat -tlpn| grep ssh

ask access to firewall:
ufw allow 30/tcp

Ionos firewall (MENU/NETWORK/ add rule)
```

CREATE USER

```
sudo adduser username
vim /etc/sudoers

add this line and force save for readonly file:
username ALL=(ALL:ALL) ALL
:w!
```

CONFIG COMMAND

```
apt-get update
apt-get upgrade   (KEEP the sshd_config as modified)
apt install build-essential
exec bash
```

INSTALL GIT

```
apt install git
apt install libz-dev libssl-dev libcurl4-gnutls-dev libexpat1-dev gettext cmake gcc


as username, not root: (to get the .gitconfig in /home/username)
wget https://www.kernel.org/pub/software/scm/git/git-2.9.5.tar.gz -O git.tar.gz
tar -zxf git.tar.gz
cd git-*
make prefix=/usr/local all
make prefix=/usr/local install

git config --list

git config --global user.name "4nkh"
git config --global user.email "admin@4nkh.ca"
git config --list

cd /var/www:
sudo git clone https://...
sudo chown -R username:username directory
```

SETUP DNS

TYPE	A
HOST NAME	www
VALUE	222.22.22.222

TYPE	A
HOST NAME	@
VALUE	222.22.22.222

NGINX

```
apt install nginx
ufw app list
ufw allow 'Nginx HTTP' or ufw allow 'Nginx HTTPS' or ufw allow 'Nginx Full'
ufw status
# verify its running
systemctl status nginx
# TEST
http://222.22.22.222

sudo service nginx restart
```

```
systemctl stop nginx
systemctl start nginx
systemctl restart nginx

load config change:
systemctl reload nginx

to disable starting automaticly on reboot:
systemctl disable nginx

to disable starting automaticly on reboot:
systemctl enable nginx

test config get problem:
nginx -t

/etc/nginx
```

SSL

https://www.digicert.com/kb/ssl-certificate-installation.htm

```
copy pem, cert & key to:
/etc/ssl/4nkh/...pem, cert,key

specify the ssl in nginx
```

INSTALL POSTGRESQL

```
apt install postgresql postgresql-contrib
systemctl start postgresql.service

sudo -i -u postgres
psql
CREATE ROLE username WITH LOGIN SUPERUSER PASSWORD ‘passwd’;
```

POSTGRESQL DB EXPORT/IMPORT

```
EXPORT DB:
pg_dump -h 127.0.0.1 -p 5432 -U username -W dbname > dbexport.pgsql


COPY ON THE OTHER SERVER:
scp -P 1267 dbexport.pgsql username@74.208.77.31:/home/username/db/dbexport.pgsql

BUILD DB:
sudo -i -u postgres
SET ROLE username;
CREATE DATABASE dbname;
```

```
\q

IMPORT DB:
psql – h 127.0.0.1 -p 5432 -W -U username dbname < dbexport.pgsql
```

RUBY

```
as username, not root: (to get the .rbenv in /home/username)
sudo apt install git curl libssl-dev libreadline-dev zlib1g-dev autoconf bison
build-essential libyaml-dev libreadline-dev libncurses5-dev libffi-dev libgdbm-dev
curl -fsSL https://github.com/rbenv/rbenv-installer/raw/HEAD/bin/rbenv-installer |
bash
echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc
echo 'eval "$(rbenv init -)"' >> ~/.bashrc
source ~/.bashrc
rbenv

rbenv install -l
rbenv install 3.3.4

rbenv global 3.3.4
ruby -v

# install gems
echo "gem: --no-document" > ~/.gemrc
gem install bundler

gem env home
```

RAILS

```
see GIT to clone an app

cd /var/www/rails_repo
rbenv local 3.3.4
bundle install

if sqlite3 (WATCH OUT ubuntu)
#apt-get install libsqlite3-devapt-get remove --auto-remove ruby-railties

if pg
sudo apt install libpq-dev
#sudo yum install postgresql-devel
#sudo zypper in postgresql-devel
#sudo pacman -S postgresql-libs

if mysql2
sudo apt-get install libmysqlclient-dev
```

```
#or  sudo yum install mysql-devel
#sudo apt-get install libmariadb-dev',

if rmagick
sudo apt-get install libmagick++-dev

rake db:create
rake db:migrate
rake db:seed

copy the bin folder (if not there)
copy config/credentials/production.key (if not there)


clean before compiling:
rails assets:clobber
RAILS_ENV=production rails assets:precompile

RAILS_ENV=production bundle exec puma -C config/puma.rb

show how many process:
grep -c processor /proc/cpuinfo
```

ANACONDA

```
cd /home/username
curl -O https://repo.anaconda.com/archive/Anaconda3-<INSTALLER_VERSION>-Linux-
x86_64.sh

curl -O https://repo.anaconda.com/archive/Anaconda3-2024.06-1-Linux-x86_64.sh

install in home/username/anaconda3
say yes to modify the .bashrc file:
yes
conda init
conda list



# The base environment is activated by default

# The base environment is not activated by default
conda config --set auto_activate_base False

# conda config init is available in conda version 4.6.12
conda config --set auto_activate_base True

conda create -n web_env
conda install python=3.12.4
```

DOCKER

```
only compatible with iptables (not ufw)

sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc


echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin

sudo docker run hello-world
sudo docker ps -a
sudo docker stop unruffled_davinci
sudo docker rm unruffled_davinci

sudo docker run -p port:port --name redis_container -d redis:5

next_time:
docker start redis_container
docker stop redis_container
```

DJANGO

```
conda activate web_env
python -m pip install Django —break-system-packages

package:
...
pip install python-dotenv
pip install daphne
pip install django-debug-toolbar
pip install django-user-agents
pip install djangorestframework
pip install psycopg2
pip install django-elasticsearch-dsl
pip install openai
pip install boto3
pip install google-generativeai
pip install channels
pip install channels-redis
pip install awscli
...

BEDROCK credential:
aws configure
...
```

DAPHNE:
sudo vim /etc/systemd/system/daphne.service

```
[Unit]
Description=daphne daemon
After=network.target

[Service]
Type=simple
User=username
WorkingDirectory=/var/www/blog
ExecStart=/home/username/anaconda3/envs/web_env/bin/daphne -u /tmp/blog.sock
blog.asgi:application

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

vim /home/username/unix_script/django_script.sh

```
#!/bin/bash

echo 'start application script on reboot'

#echo 'start postgesql'
#systemctl start postgresql.service

echo 'start redis in docker'
sudo docker start redis_chatbot

echo 'start daphne'
sudo systemctl start daphne.service

#echo 'start nginx'
#service nginx start

#bash "name_of_file".sh
```

```
sudo vim /lib/systemd/system/django_startup.service
```

```
[Unit]
Description=Django Startup Script
[Service]
ExecStart=/bin/bash "/home/username/unix_script/django_script.sh"
[Install]
WantedBy=multi-user.target
```

```
systemctl enable django_startup.service --now
```