Após instalar o sistema operacional a primeira providencia será definir um endereço IP estático, para isso, entrando como administrador do sistema utilizaremos o comando:
nano /etc/network/interfaces

```
  GNU nano 7.2                  /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp2s0
iface enp2s0 inet static
        address 192.168.3.2/24
        netmask 255.255.255.0
        gateway 192.168.3.1
```

Após esta configuração você pode reiniciar a maquina ou apenas a interface com o comando *systemctl reboot*

Agora vamos instalar nosso servidor DNS com o comando:
apt-get install bind9

Aguardamos a conclusão da instalação e acessamos o endereço das configurações com o comando: *cd /etc/bind* e podemos visualizar os arquivos da pasta com o comando ls.

```
root@debian:/etc/bind# ls
bind.keys   db.255    named.conf                named.conf.options
db.0        db.empty  named.conf.default-zones  rndc.key
db.127      db.local  named.conf.local          zones.rfc1918
```

O primeiro arquivo que vamos configurar é o *nano named.conf.default-zones* e no final do arquivo adicionamos nosso domínio:

```
zone "inneo.io" {
        type master;
        file "/etc/bind/forward.inneo.io";
};

zone "3.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/reverse.inneo.io";
};
```

Finalizando essa parte vamos criar nossas zonas definidas no arquivo acima e para isso, copiamos os arquivos *db.local* e *db.127* renomeando para *forward.inneo.io* e *reverse.inneo.io*, no caso você utiliza seu domínio.

```
root@debian:/etc/bind# cp db.local forward.inneo.io
root@debian:/etc/bind# cp db.127 reverse.inneo.io
```

Com os modelos copiados, iniciamos a configuração pelo arquivo *forward.inneo.io* com o comando: *nano forward.inneo.io*

```
$TTL    604800
@       IN      SOA     inneo.io. root.inneo.io. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      inneo.io.
@       IN      A       192.168.3.2
```

E agora nosso DNS reverso: *nano reverse.inneo.io*

```
$TTL    604800
@       IN      SOA     inneo.io. root.inneo.io. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      inneo.io.
2       IN      PTR     inneo.io.
```

Com isso temos nosso servidor DNS pronto e vamos instalar o DOCKER para podermos iniciar nossos containers: recomendo executar uma linha por vez para evitar erros.

```
apt-get update

apt-get install ca-certificates curl

install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/debian/gpg -o
/etc/apt/keyrings/docker.asc

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/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  tee /etc/apt/sources.list.d/docker.list > /dev/null

apt-get update
```

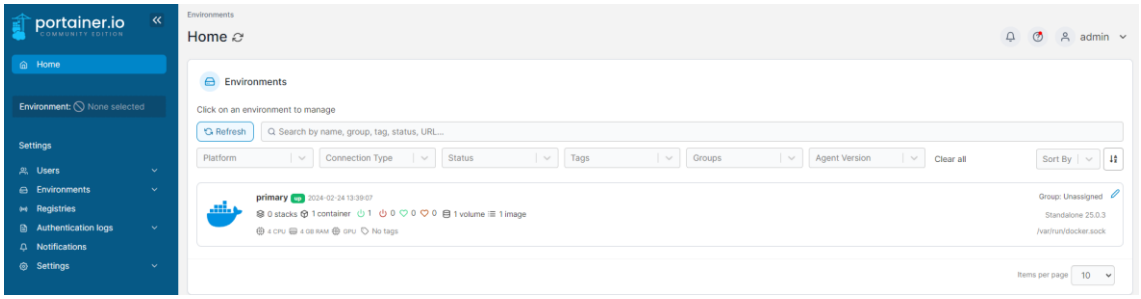
apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

e agora instalaremos o portainer.io para gerenciar nossas aplicações docker. Começamos criando um volume com o comando:

docker volume create portainer_data

docker run -d -p 8000:9000 --name portainer --restart always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer -H unix:///var/run/docker.sock

finalizado a execução do comando basta acessarmos nosso portainer e adicionar nossa nova senha de admin: *http://192.168.3.2:8000* e teremos acesso ao nosso portainer.



Com isso temos nosso servidor rodando e pronto para receber seus containers docker.