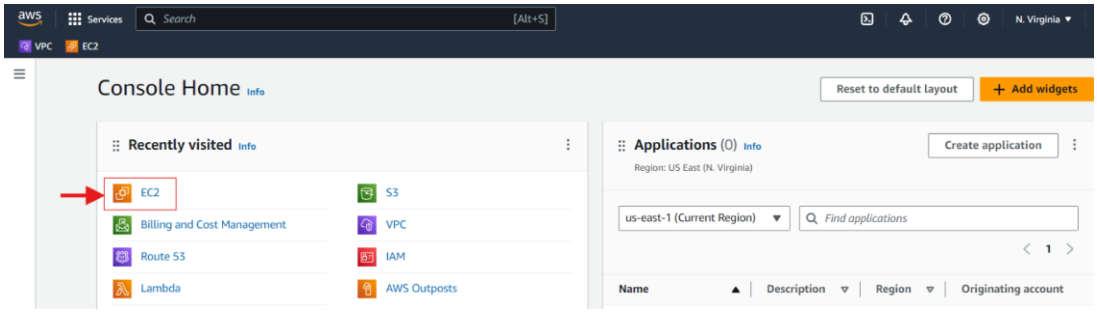
INSTALLING NEXUS & KUBERNETES

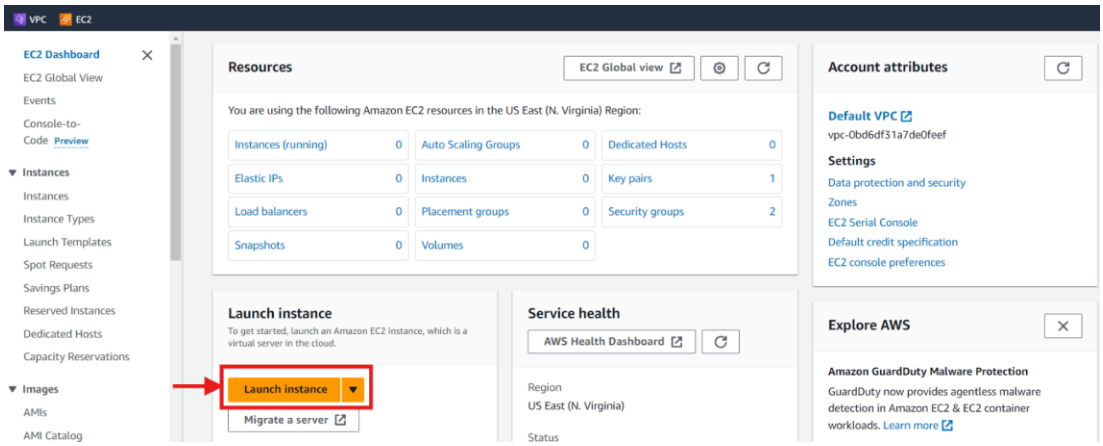
INSTALL AND SETUP NEXUS

✓ Spin Up EC2 instance in AWS

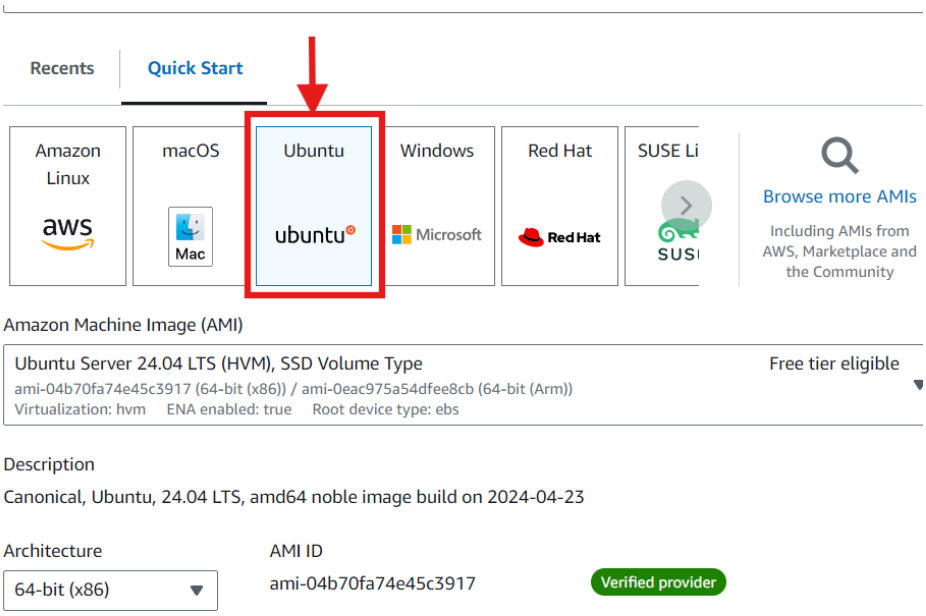
- Login to **AWS console and select EC2 service** (If you're using a noon root account make sure you have the necessary permissions/roles)



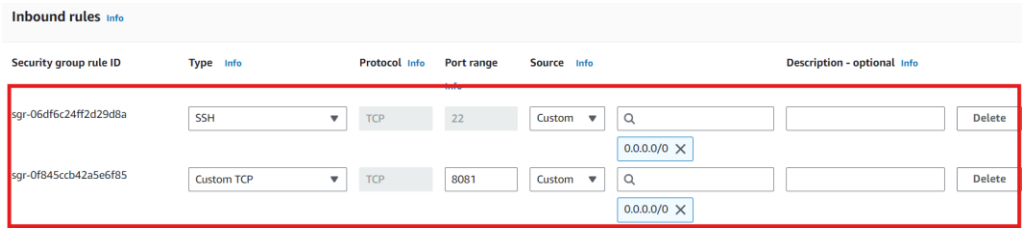
- Select **Launch Instance,** in EC2 service dashboard



- In the launch instance windows, configure the following:-
  - Give the instance a name
  - Select the **Ubuntu Server 20.04 LTS** Quick start Image



- Select any of the Instance types (min **T2.medium**)
- **Select a key pair** from available once (* need for SSH)
- In the network setting leave everything as defaults and **select a Security group with the following Inbound traffic ports opened 22 (SSH) 8081 (Nexus UI)**



- In a configure Storage section set minimum **20GB.**
- Finish and launch the instance.

✓ Install Docker

- Login to server, and run the following commands (*Ensure user has sudo user privileges)
- Add Docker's official GPG key:

```
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
```

- Add the repository to Apt sources:

```
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
```

- Install docker and its components

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

- Give non root user permission to docker

```
sudo usermod -aG docker $USER && newgrp docker
```

✓ Start Nexus Container

- Run the following commands

```
$ docker run -d -p 8081:8081 --name nexus sonatype/nexus3
```

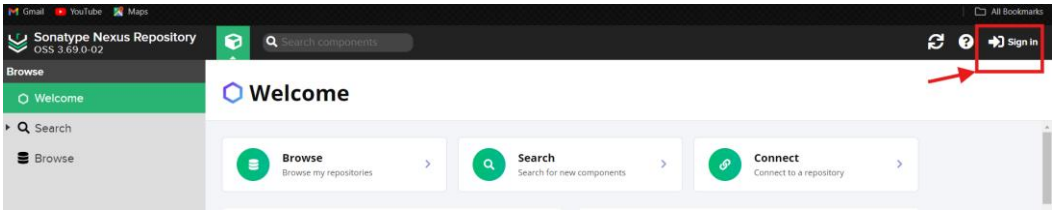
- Make sure that Nexus container is running by running the following commands.

```
docker ps
```

```
ubuntu@ip-172-31-92-77:~$ docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED          STATUS          PORTS                                       NAMES
7681b053806c   sonatype/nexus3   "/opt/sonatype/nexus…"   58 seconds ago   Up 52 seconds   0.0.0.0:8081->8081/tcp, :::8081->8081/tcp   nexus
ubuntu@ip-172-31-92-77:~$
```

✓ Setup Nexus

- Access the https URL of Jenkins (http://<server-public-ip>:8081) and setup the server.
- Click on Sign In the startup page.

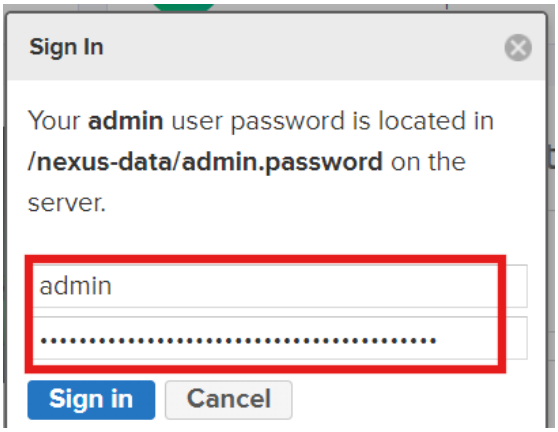


- Extract the Default Admin password, run the following command

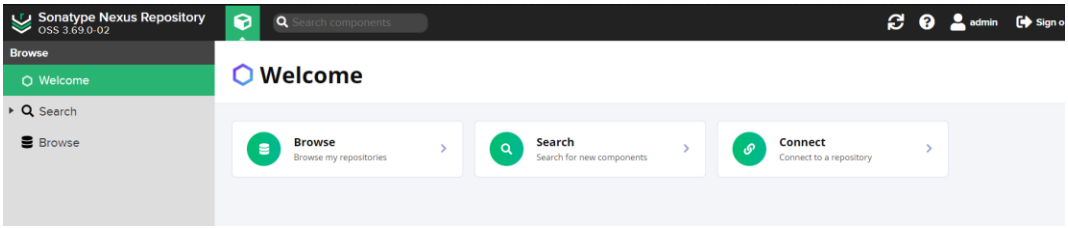
```
docker exec -it nexus /bin/bash
cat /nexus-data/admin.password
```

```
ubuntu@ip-172-31-92-77:~$ docker exec -it nexus  /bin/bash
bash-4.4$ cat /nexus-data/admin.password
2e253c2a-010e-4a2b-979a-0a8a6bc6c682bash-4.4$
```

- Use the credential **user : admin , password - *extracted password**



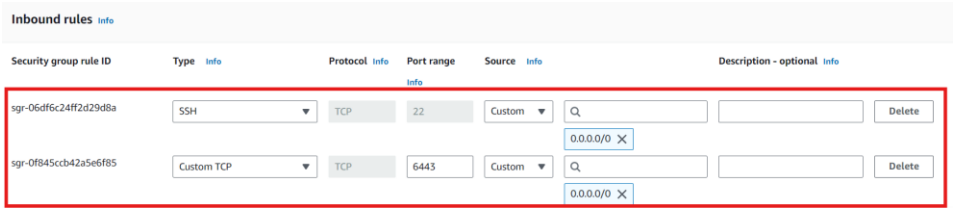
- Nexus is setup



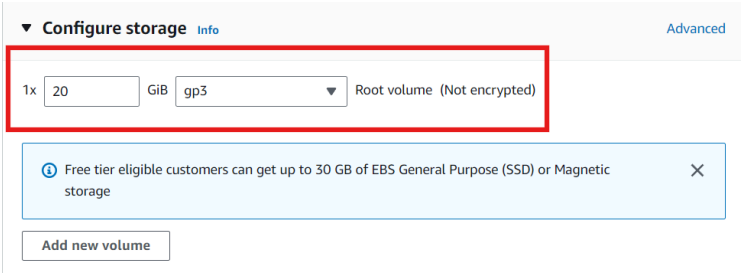
INSTALL AND SETUP KUBERNETES

✓ Spin Up EC2 instance in AWS

- Login to **AWS console and select EC2 service** (If you're using a noon root account make sure you have the necessary permissions/roles.
- Select **Launch Instance,** in EC2 service dashboard
- In the launch instance windows, configure the following:-
  - Give the instance a name
  - Select the **Ubuntu Server 20.04 LTS** Quick start Image
  - Select any of the Instance types (min **T2.medium**)
  - **Select a key pair** from available once (* need for SSH)
  - In the network setting leave everything as defaults and **select a Security group with the following Inbound traffic ports opened 22 (SSH) 6443 (Kubernetes API server)**



  - In a configure Storage section set minimum **20GB.**



  - Finish and launch the instance.

✓ Install Docker

- Login to server, and run the following commands (*Ensure user has sudo user privileges)
- Add Docker's official GPG key:

```
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
```

- Add the repository to Apt sources:

```
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
```

- Install docker and its components

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

- Give non root user permission to docker

```
sudo usermod -aG docker $USER && newgrp docker
```

✓ Install Minikube

- Run the following commands

```
https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-
amd64
```

✓ Setup Minikube cluster

- Run the following commands

```
minikube start
```

```
ubuntu@ip-172-31-86-113:~$ minikube start
* minikube v1.33.1 on Ubuntu 24.04 (xen/amd64)
* Automatically selected the docker driver. Other choices: ssh, none
* Using Docker driver with root privileges
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.44 ...
* Downloading Kubernetes v1.30.0 preload ...
    > preloaded-images-k8s-v18-v1...:  342.90 MiB / 342.90 MiB  100.00% 78.62 M
    > gcr.io/k8s-minikube/kicbase...:  481.58 MiB / 481.58 MiB  100.00% 60.41 M
* Creating docker container (CPUs=2, Memory=2200MB) ...
* Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: default-storageclass, storage-provisioner
* kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

- Check if the Kubernetes pods are up and running by running following commands

```
minikube kubectl -- get pods -A
ubuntu@ip-172-31-86-113:~$ minikube kubectl -- get pods -A
    > kubectl.sha256:  64 B / 64 B [-------------------------] 100.00% ? p/s 0s
    > kubectl:  49.07 MiB / 49.07 MiB [----------] 100.00% 296.60 MiB p/s 400ms
NAMESPACE     NAME                               READY   STATUS    RESTARTS   AGE
kube-system   coredns-7db6d8ff4d-nz46d           0/1     Running   0          16s
kube-system   etcd-minikube                      1/1     Running   0          30s
kube-system   kube-apiserver-minikube            1/1     Running   0          30s
kube-system   kube-controller-manager-minikube   1/1     Running   0          30s
kube-system   kube-proxy-2nskw                   1/1     Running   0          16s
kube-system   kube-scheduler-minikube            1/1     Running   0          31s
kube-system   storage-provisioner                1/1     Running   0          28s
ubuntu@ip-172-31-86-113:~$
```