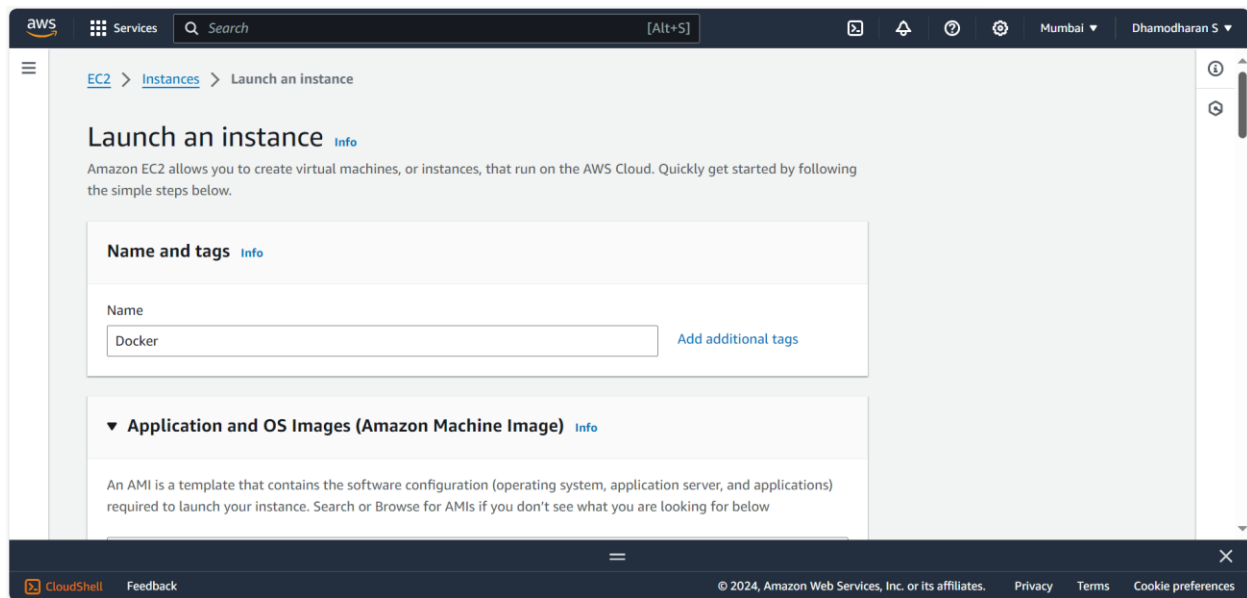# Docker –Container Engine

Dhamodharan S

Step 1: I'm creating an Instance in AWS EC2 as Docker

Step 1.1 Launch Instance

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE Linux	Browse more AMIs
aws	Mac	ubuntu	Microsoft	Red Hat	SUSE	Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type   Free tier eligible
ami-03f4878755434977f (64-bit (x86)) / ami-077885f59ecb77b84 (64-bit (Arm))
Virtualization: hvm   ENA enabled: true   Root device type: ebs

Description
Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-12-07

○ Create security group   ○ Select existing security group

Security group name - *required*

launch-wizard-2

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required*   Info

launch-wizard-2 created 2024-02-21T10:55:02.883Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 0-65535, 0.0.0.0/0)   Remove

Type Info	Protocol Info	Port range Info
All TCP ▾	TCP	0-65535

Source type Info	Source Info	Description - *optional* Info
Anywhere ▾	Add CIDR, prefix list or security g	e.g. SSH for admin desktop
	0.0.0.0/0 ✕	
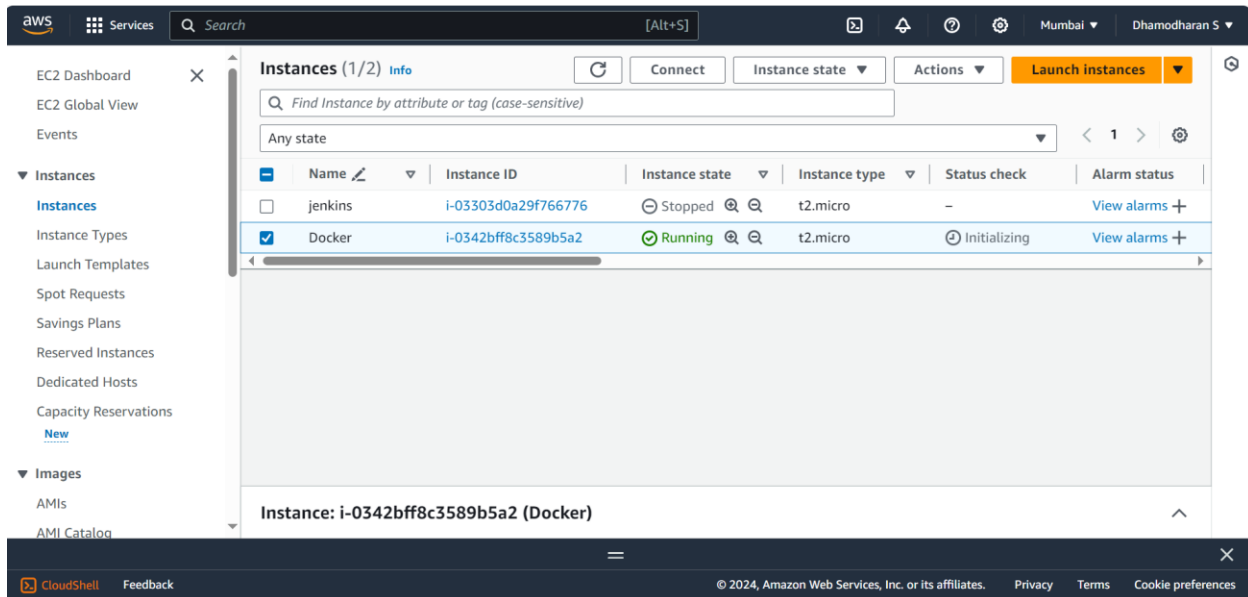
Step 2: Connet to the Instance "Docker" using EC2 Connect.
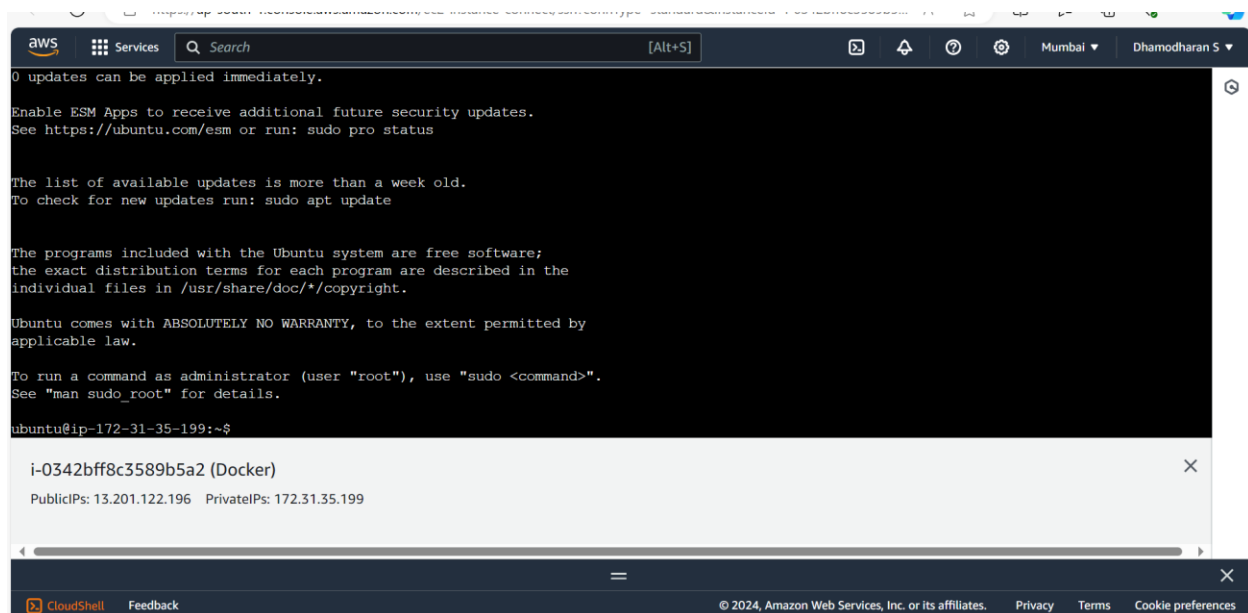
Step 3: Now I'm Installing Docker using APT Repository

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
```

#To install the latest version, run:

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

#Verify that the Docker Engine installation is successful by running the hello-world image.

sudo docker run hello-world

Step 4: I'm running all the commands to set up APT repository.

Step 5: I'm running Hello World using command "docker run hello world"

Step 6: Now here im using one image called "NGINX" from docker and pulling it and You can see "NGINX" image is downloaded.

Step 7: now im starting to run my image in background as a name "my-nginx"

Step 8: Now you can go to the port 80 and check your nginx server is running. docker pull nginx docker run --name my-nginx-container -d -p 80:80 nginx
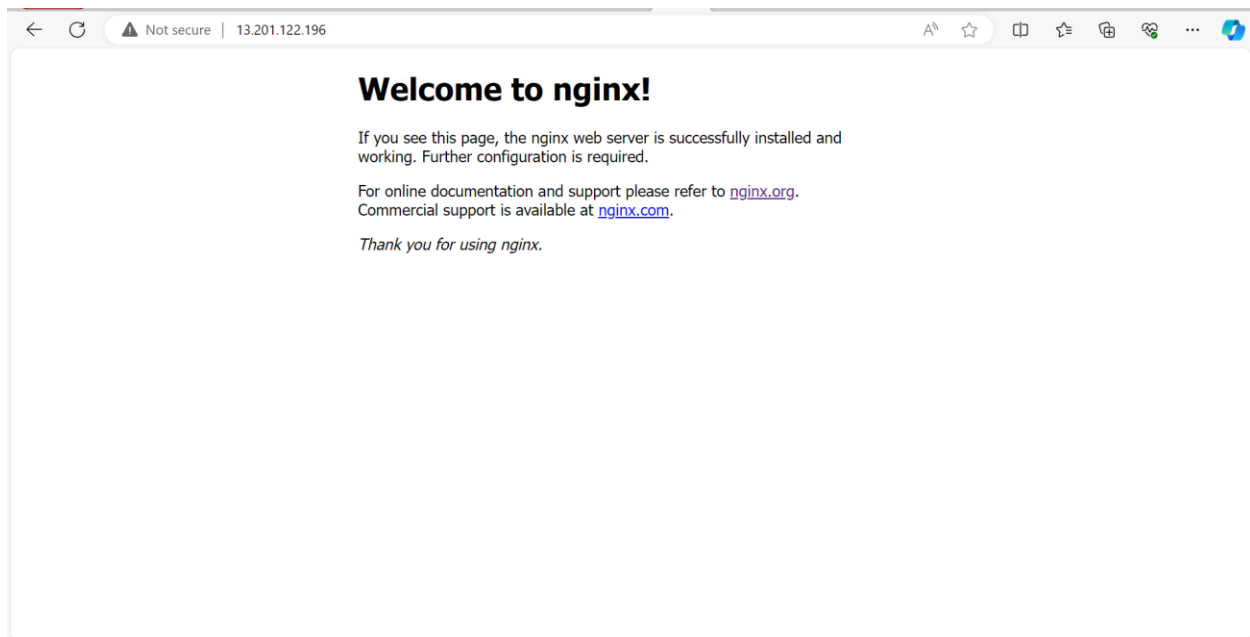
Step 9: Now let us customize our nginx server which is running. Enter into the executing container using command "sudo docker exec -it c60c31b70646 /bin/bash".

Step 10: Now customize index.html file by navigating to the index.html file. cd /usr/share/nginx/html/ nano index.html and restart the running container.

Step 11: Now you can see the customized html file running in server

---

← ↻ ⚠ Not secure | 13.201.52.86

Welcome dhamu to nginx!

Hi Dhamu welcome For online documentation and support please refer to
nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.