

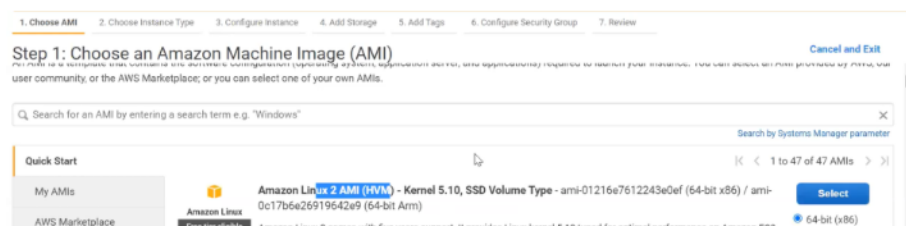


Summary

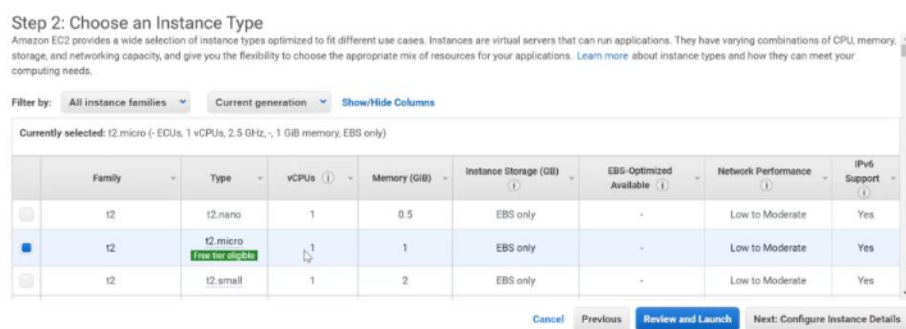
Session No 1

- From a user perspective main use of the operating system is to run the program
- If we want to run any program or application we need the operating system
- There is no way to interact with the operating system without the program
- For getting the operating system we need physical hardware (RAM, CPU, Network card, HD) and we need to install the operating system
- Running the program is also called deploy
- Different ways of installing the operating system
 - Bare metal
 - Cloud computing
 - Virtualization
 - Containerization
- Docker is a product that works on the concept of containerization
- Case study on Hotstar
 - Hotstar created the world record on the ICC world cup 2019 IND vs NZ match of having 25.3 million concurrent users
 - For launching the Operating system Hotstar used containerization
 - For launching the server we need physical hardware and we need to install the operating system on it and deploy the application it is also called tech stack
 - Increasing the servers is also called scaling

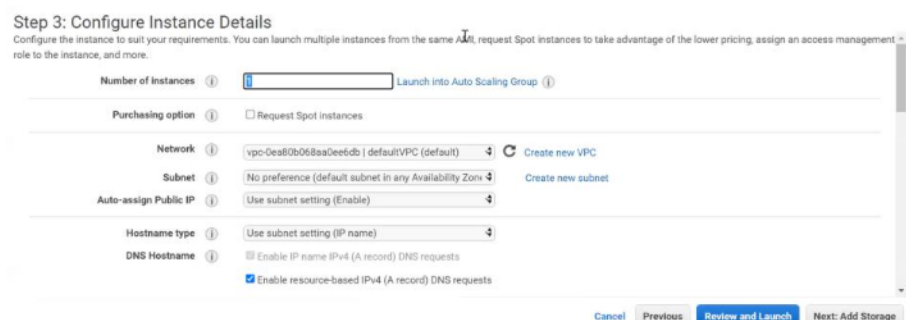
- Before 2018 Hotstar was using Ec2 instance to launch an operating system
- If a surge of requests comes up Ec2 instance will not help because it requires some time to launch the operating system
- If we want to launch the operating system in 1 second then we can use Docker which works on the containerization concept
- Docker is software, to run it we need the operating system. Mostly Linux operating system is used
- Launching instance on Aws cloud
 - Step 1 - Choose Amazon Machine image



- Step 2 –Choose instance type



- Step 3 – Configure instance details



○ Step 4 – Adding a tag

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

○ Step 5 – Configuring security groups and allowing all traffic

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

- **su** command is used to switch between users

```
[ec2-user@ip-172-31-40-68 ~]$ whoami
ec2-user
[ec2-user@ip-172-31-40-68 ~]$ sudo su - root
[root@ip-172-31-40-68 ~]# whoami
root
[root@ip-172-31-40-68 ~]#
```

- Installing Docker

```
[root@ip-172-31-40-68 ~]# rpm -q docker-ce
package docker-ce is not installed
[root@ip-172-31-40-68 ~]# yum install docker
```

- Starting Docker services

```
[root@ip-172-31-40-68 ~]# systemctl status docker
• docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
  Active: inactive (dead)
  Docs: https://docs.docker.com
[root@ip-172-31-40-68 ~]# systemctl start docker
[root@ip-172-31-40-68 ~]#
```

- Docker program or service is also known as docker-engine

- If we launch any OS on top of the cloud it is called an instance similarly if we launch any OS on top of Virtualization it is called Host/VM and on top of containerization if we launch any OS it is called Container
- **docker ps** command is used to see the running containers
- Minimum requirement to launch the Operating system we need an image
- Without an OS image we can not install or launch the operating system
- Registry contains the container image. The famous public registry is DOCKER HUB
- **docker images** command is used to display all the docker images
- **docker pull (image name)** command is used to pull or download the image

```
[root@ip-172-31-40-68 ~]# docker pull ubuntu:14.04
14.04: Pulling from library/ubuntu
2e6e20c8e2e6: Pull complete
0551a797c01d: Pull complete
512123a864da: Pull complete
Digest: sha256:d7a459ecd77ebb09525584f2c3e1bb7f6a2879d90df8a3523c1b899dfc2a226f
Status: Downloaded newer image for ubuntu:14.04
docker.io/library/ubuntu:14.04
[root@ip-172-31-40-68 ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	14.04	13b66b487594	18 months ago	197MB

```
[root@ip-172-31-40-68 ~]#
```

- Launching the container
 - Command - **docker run -t -i image name**
 - -i means interactive
 - -t for a terminal

```
[root@ip-172-31-40-68 ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	14.04	13b66b487594	18 months ago	197MB

```
[root@ip-172-31-40-68 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
[root@ip-172-31-40-68 ~]# docker run -t -i ubuntu:14.04						

```
root@474472007da0:/#
root@474472007da0:/#
root@474472007da0:/#
root@474472007da0:/#
```

- **docker ps -a** command is used to see all the container

```
[root@ip-172-31-40-68 ~]# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
cal5ebc88b5a   ubuntu:14.04   "/bin/bash"             33 seconds ago Up 32 seconds             friendly_lamport
474472007da0   ubuntu:14.04   "/bin/bash"             3 minutes ago  Exited (0) 54 seconds ago intelligent_kilby
[root@ip-172-31-40-68 ~]#
```

- **docker stop (Container ID)** to stop the container from the base system

```
[root@ip-172-31-40-68 ~]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
cal5ebc88b5a   ubuntu:14.04   "/bin/bash"             About a minute ago Up About a minute             friendly_lamport
[root@ip-172-31-40-68 ~]# docker stop cal5ebc88b5a
cal5ebc88b5a
[root@ip-172-31-40-68 ~]#
```