

Sprint-5 (IAC)

Docker - Day 1

Q: What is Docker?

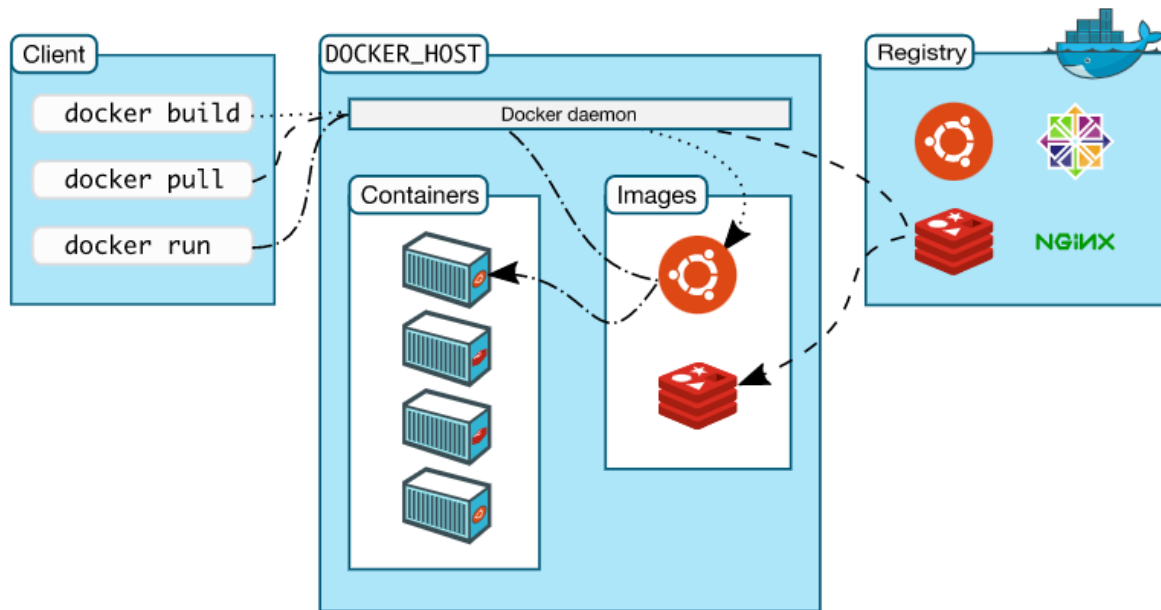
Answer: Docker is a software packaging tool to package the software as a container which uses the OS virtualization. Docker can package an application and its dependencies in a virtual container that can run on any Linux, Windows, or MacOS computer.

Docker is used for:

- Fast, consistent Delivery of your applications - Containers are great for continuous integration and continuous delivery (CI/CD) workflows.
- Responsive Deployment and scalability- Docker's container-based platform allows for highly portable workloads.
- Running more workloads on same hardware- Docker is perfect for high density environments and for small and medium deployments where you need to do more with fewer resources.

Docker Architecture:

Docker uses a client-server architecture. The Docker *client* talks to the Docker *daemon*, which does the heavy lifting of building, running, and distributing your Docker containers. The Docker client and daemon *can* run on the same system, or you can connect a Docker client to a remote Docker daemon.



The Docker daemon:

The Docker daemon listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes. A daemon can also communicate with other daemons to manage Docker services.

The Docker Client:

The Docker client is the primary way that many Docker users interact with Docker. When you use commands such as `docker run`, the client sends these commands to docker, which carries them out. The docker command uses the Docker API. The Docker client can communicate with more than one daemon.

Docker Desktop:

Docker Desktop is an easy-to-install application for your Mac, Windows or Linux environment that enables you to build and share containerized applications and microservices. Docker Desktop includes the Docker daemon, the Docker client, Docker Compose, Docker Content Trust, Kubernetes, and Credential Helper.

Docker Registry:

A Docker registry stores Docker images. Docker Hub is a public registry that anyone can use, and Docker is configured to look for images on Docker Hub by default. You can even run your own private registry.

When you use the `docker pull` or `docker run` commands, the required images are pulled from your configured registry. When you use the `docker push` command, your image is pushed to your configured registry.

Docker objects:

When you use Docker, you are creating and using images, containers, networks, volumes, plugins, and other objects. This section is a brief overview of some of those objects.

Image:

An image is a read-only template with instructions for creating a Docker container. Often, an image is based on another image, with some additional customization.

Q: What is container in Docker.

Answer: A container is a standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Container images become containers at runtime and in the case of Docker containers – images become containers when they run on Docker Engine.

Q: What is EC2?

Answer: EC2 (Elastic Compute Cloud). It provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. It is like a virtual machine. Using EC2 eliminates our need to invest in hardware up front, so we can develop and deploy applications faster.

Features of EC2:

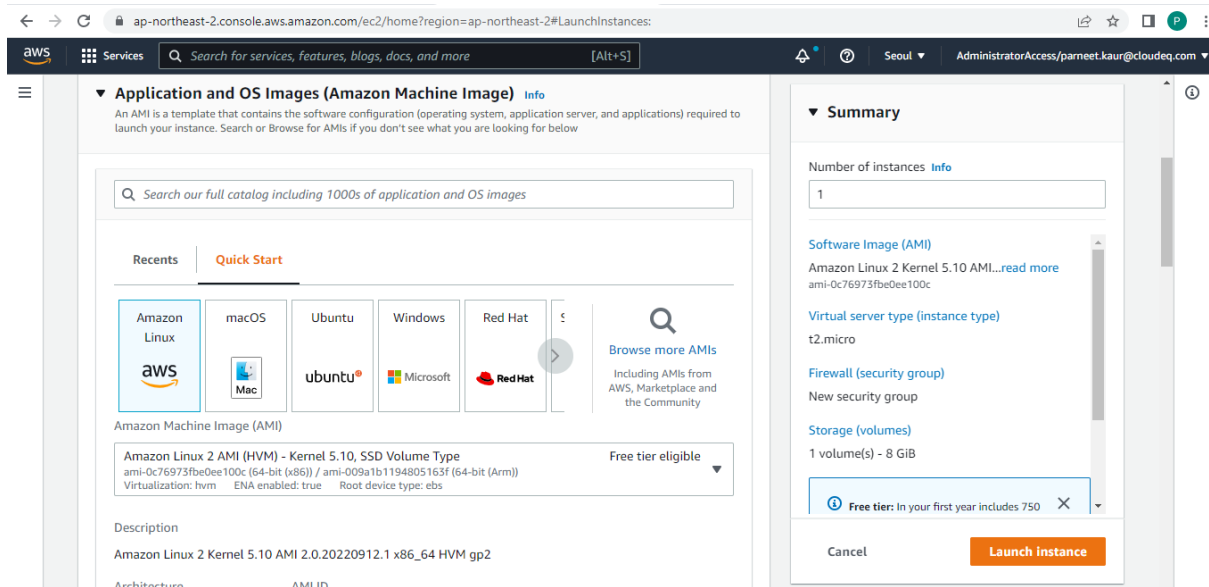
- Virtual computing environments, known *as* instances
- Preconfigured templates for your instances, known as Amazon Machine Images (AMIs).
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as instance types.
- Secure login information for your instances using key pairs (AWS stores the public key, and we store the private key in a secure place).
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as Regions and Availability Zones.

- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using security groups.

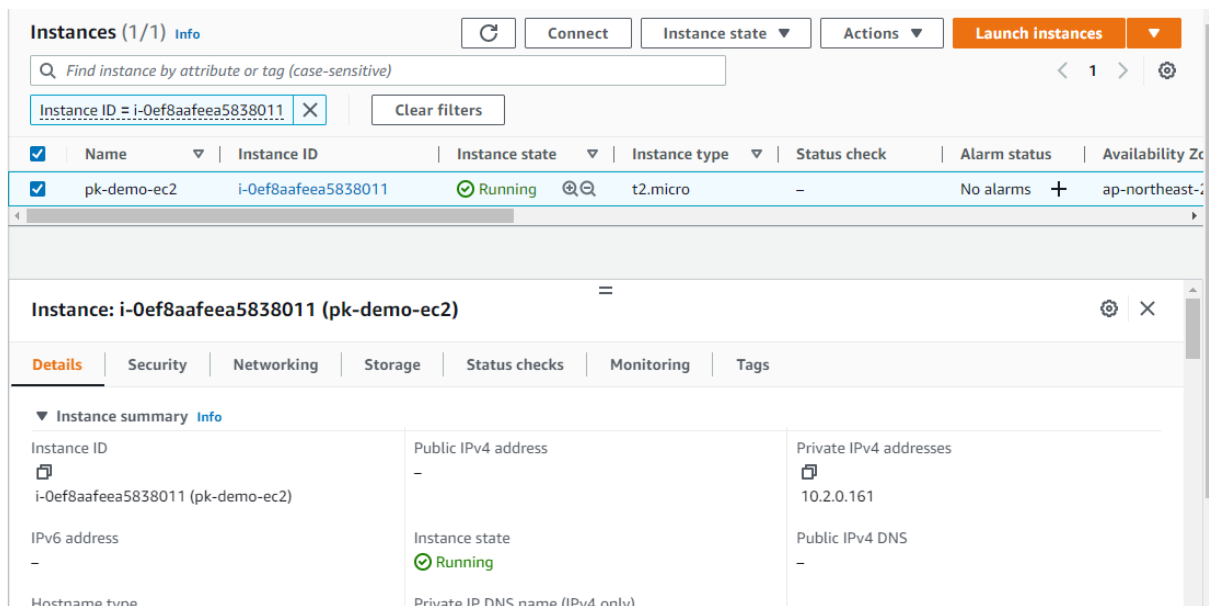
Steps to create EC2:

Step1: We will login to our AWS console and open EC2 and click on “Launch Instance”.

Step 2: After selecting all the properties click on “Launch instance” and our EC2 instance will be created.



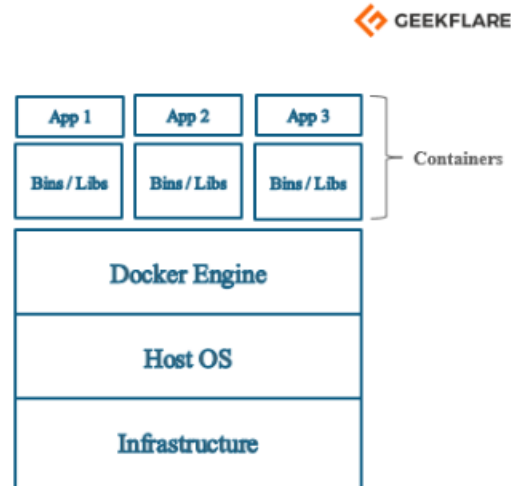
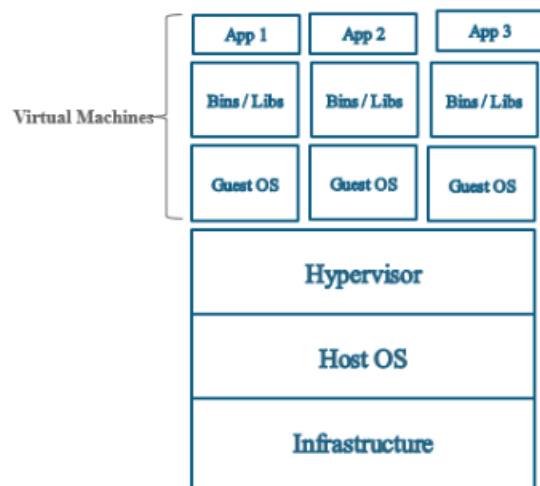
We can see our instance is running successfully:



Q: Difference between Docker and VM:

Answer: The main difference between these two technologies is that VMs (Virtual machines) run as virtual environments on the same hardware, whereas Docker runs on virtualizations of the same operating system.

Basis	Docker	Virtual Machine (VM)
Definition	Docker is a software packaging tool that uses containers to make creation, deployment, and running of application a lot easier. It binds application and its dependencies inside a container.	A virtual machine is a system which acts exactly like a computer. It makes it possible to run what appears to be on many separate computers on hardware, that is one computer. Each virtual machine requires its underlying operating system, and then the hardware is virtualized.
Security	A container has a lot of security risks, and vulnerabilities as the containers have shared host kernel or OS. Also, since docker resources are shared and not namespace, an attacker can exploit all the containers in a cluster if he/she gets access to even one container.	The virtual machine does not share operating system, and there is strong isolation in the host kernel or OS. Hence, they are more secure as compared to Containers. In a virtual machine, you don't get direct access to the resources, and hypervisor is there to restrict the usage of resources in a VM.
Portability	Docker containers are easily portable because they do not have separate operating systems. A container can be ported to a different OS, and it can start immediately.	Virtual machines have separate OS, so porting a virtual machine is difficult as compared to containers, and it also takes a lot of time to port a virtual machine because of its size.
Performance	Its Boots in seconds.	It Boots in Minutes.



Q: Download Docker and MobaXterm.
Solution: Installed and Signed in.