**COPY** takes in a src and destination. It only lets you copy in a local file or directory from host (the machine building the Docker image) into the Docker image itself.

**ADD** lets you do that too, but it also supports 2 other sources. First, you can use a URL instead of a local file / directory. Secondly, **you can extract a tar file from the source directly into the destination.**

**CMD** sets default command and/or parameters, which can be overwritten from command line when docker container runs.

**ENTRYPOINT** configures a container that will run as an executable and it cannot be overwritten.

What is CMD command in Docker?

The **CMD command**​ specifies the instruction that is to be executed when a **Docker** container starts. ... The main purpose of the **CMD command** is to launch the software required in a container.

What is difference between run and CMD in Docker?

**RUN** lets you execute commands inside of your **Docker** image. ... **CMD** lets you define a default command to **run** when your container starts.

**Docker** is a platform that runs each and every application segregated and securely

**Container** is the run time instance of images

An **image** is a read-only template with instructions for creating a **Docker container**.

Kubernetes

**Kubernetes** is an open-source platform and container orchestration tool. It is mainly used to automate the deployment, scaling, and operations of the container-based applications across the cluster of nodes.

**port**

Expose the *service* on the specified port internally within the cluster. That is, the service becomes visible on this port, and will send requests made to this port to the pods selected by the service.

**nodePort**

This setting makes the service visible *outside* the Kubernetes cluster by the node’s IP address and the port number declared in this property. The service also has to be of type NodePort (if this field isn’t specified, Kubernetes will allocate a node port automatically).

**targetPort**

This is the port on the *pod* that the request gets sent to. Your application needs to be listening for network requests on this port for the service to work.

**There are four types of Kubernetes services:**

* ClusterIP. This default type exposes the service on a cluster-internal IP. ...
* NodePort. This type of service exposes the service on each node's IP at a static port. ...
* LoadBalancer. ...
* ExternalName.

**Following are the components of Kubernetes Master Machine.**

* etcd. It stores the configuration information which can be used by each of the nodes in the cluster. ...
* API Server. ...
* Controller Manager. ...
* Scheduler. ...
* Docker. ...
* Kubelet Service. ...
* Kubernetes Proxy Service.

Terraform

Command: **init**. The **terraform init** command is used to **initialize** a working directory containing **Terraform** configuration files. This is the first command that should be run after writing a new **Terraform** configuration or cloning an existing one from version control.

**Providers**. **Terraform** is used to create, manage, and update infrastructure resources such as physical machines, VMs, network switches, containers, and more. Almost any infrastructure type can be represented as a resource in **Terraform**. A **provider** is responsible for understanding API interactions and exposing resources.

**Resources** are the most important element in the **Terraform** language. Each **resource** block describes one or more infrastructure objects, such as virtual networks, compute instances, or higher-level components such as DNS records.