**Question Jenkins:**

**What are the advantages of using Jenkins?**

**Answer:** Advantages of Jenkins are –

* Provides great collaboration between development and operations team, making it into a single DevOps team
* Code errors can be detected as early as possible.
* Code deployment is easy and happens in minutes, along with the generation of reports.
* Automation of integration work, thereby reducing the number of integration issues.

**Question: Do you know any other continuous integration tools? How is Jenkins better than any of those?**

**Answer:** There are many other CI tools, the prominent ones being –

* TeamCity
* Bamboo
* Perforce
* Circle CI
* Go
* ThoughtWorks
* Integrity
* Travis CI

#### ****What is Groovy?****

**Answer:** [Groovy](https://hackr.io/tutorials/learn-apache-groovy?ref=blog-post) from Apache is a language for Java platform. It is the native scripting language for Jenkins. Groovy-based plugins enhance Jenkins with great interfaces and build reports that are dynamic and consistent.

#### ****Question: Can you start Jenkins using command line? How?****

**Answer:** Yes, using jenkins.exe start

**Question: What are the types of pipelines in Jenkins?**

**Answer:** There are 3 types –

1. CI CD pipeline (Continuous Integration Continuous Delivery)
2. Scripted pipeline
3. Declarative pipeline

#### ****Question: What is an agent?****

**Answer:** Agent specifies the execution point for a particular stage in the pipeline or the entire pipeline itself.

#### ****Question: How do you define parameters in Jenkins?****

**Answer:** A build can take several input parameters to execute. For example, if you have multiple test suites, but you want to run only one. You can set a parameter to decide which one should be run. To have parameters in a job, you need to specify the same while defining the parameter. The parameter can be a string, file or custom

### **Terraform.**

### **2. What are the reasons to choose Terraform for DevOps?**

Answer: Terraform supports automation and helps in running infrastructure as code. Another potential reason for choosing Terraform is the facility for implementing almost any type of coding principle.

### **3. What are the notable features of Terraform?**

Answer: The features of Terraform would be one of the common topics of the latest Terraform interview questions. The key features of Terraform are as follows.

1. In-built graphing features for visualization of infrastructure
2. Friendly custom syntax helps in improving efficiency
3. The ability for understanding resource relationships
4. Contribution of updates and new features by the Open Source Project
5. Capability for breaking down a configuration into smaller parts for ease of organization and maintenance

### **4. How does Terraform work?**

Answer: The working of Terraform is a formidable topic that churns out many relevant Terraform interview questions. The best answer to this question would be to point towards the plugin-based architecture of Terraform. The plugin-based architecture helps developers in extending functionalities of Terraform. Developers could write new plugins or compile the modified versions of current plugins.

Github:

Git fetch and git pull

**git fetch** is the command that tells your local **git** to retrieve the latest meta-data info from the original (yet doesn't do any file transferring. It's more like just checking to see if there are any changes available). **git pull** on the other hand does that AND brings (copy) those changes from the remote repository

### **17. How to resolve a conflict in Git?**

The following steps will resolve conflict in Git-

1. Identify the files that have caused the conflict.
2. Make the necessary changes in the files so that conflict does not arise again.
3. Add these files by the command git add.
4. Finally to commit the changed file using the command git commit

**VM ware:**

**What is VMware and what are their benefits?**

VMware provides different applications and software for virtualization.  VMware products are categorized into two levels, desktop applications, and Server applications.

**3) Mention different types of server software do VMware provides?**

VMware provides three different types of server software

* VMware ESX Server
* VMware ESXi Server
* VMware Server

**4) Explain what is hypervisor**

A hypervisor is a program that enables multiple operating systems to share a single hardware host.  Each operating system has the host’s processor, memory and other resources all to itself.  The hypervisor controls the resources and host processor, allocating what is required for each operating system in turn and make sure that the guest operating system cannot disrupt each other.

**5) Explain VMware DRS?**

VMware DRS stands for Distributed Resource Scheduler; it dynamically balances resources across various host under a cluster or resource pool.  It enables users to determine the rules and policies which decide how virtual machines deploy resources, and these resources should be prioritized to multiple virtual machines.

**Maven:**

**1) Explain what is Maven? How does it work?**

Maven is a project management tool. It provides the developer a complete build lifecycle framework. On executing Maven commands, it will look for POM file in Maven; it will run the command on the resources described in the POM.

**2) List out what are the aspects does Maven Manages?**

Maven handles following activities of a developer

* Build
* Documentation
* Reporting
* Dependencies
* SCMs
* Releases
* Distribution
* Mailing list

**3) Mention the three build lifecycle of Maven?**

* Clean: Cleans up artifacts that are created by prior builds
* Default (build): Used to create the application
* Site: For the project generates site documentation

**4) Explain what is POM?**

**Docker:**

**. What is Docker?**

Docker is an open-source lightweight containerization technology. It has gained widespread popularity in the cloud and application packaging world. It allows you to automate the deployment of applications in lightweight and portable containers.

**. What is Docker image?**

The Docker image help to create Docker containers. You can create the Docker image with the build command. Due to this, it creates a container that starts when it begins to run. Every docker images are stored in the Docker registry.

**8. What command should you run to see all running container in Docker?**

$ docker ps

**9. Write the command to stop the docker container**

$ sudo docker stop container name

### **What is Docker Swarm?**

You are expected to have worked with Docker Swarm as it’s an important concept of Docker.

Docker Swarm is native clustering for Docker. It turns a pool of Docker hosts into a single, virtual Docker host. Docker Swarm serves the standard Docker API, any tool that already communicates with a Docker daemon can use Swarm to transparently scale to multiple hosts.

### **What is the lifecycle of a Docker Container?**

This is one of the most popular questions asked in Docker interviews. Docker containers have the following lifecycle:

* Create a container
* Run the container
* Pause the container(optional)
* Un-pause the container(optional)
* Start the container
* Stop the container
* Restart the container
* Kill the container
* Destroy the container

### **Q.  How do you get the number of containers running, paused and stopped?**

You can use the following command to get detailed information about the docker installed on your system.

$ docker info

### **20. If you wish to use a base image and make modifications or personalize it, how do you do that?**

You pull an image from docker hub onto your local system

It’s one simple command to pull an image from docker hub:

$ docker pull <image\_name>

### **21. How do you create a docker container from an image?**

Pull an image from docker repository with the above command and run it to create a container. Use the following command:

$ docker run -it -d <image\_name>

Most probably the next question would be, what does the ‘-d’ flag mean in the command?

**-d** means the container needs to start in the detached mode. Explain a little about the detach mode. Have a look at [this](https://www.edureka.co/blog/docker-commands/) blog to get a better understanding of different docker commands.

### **23. Suppose you have 3 containers running and out of these, you wish to access one of them. How do you access a running container?**

The following command lets us access a running container:

$ docker exec -it <container id> bash

The exec command lets you get inside a container and work with it.

**24. How to start, stop and kill a container?**

The following command is used to start a docker container:

$ docker start <container\_id>

and the following for stopping a running container:

$ docker stop <container\_id>

kill a container with the following command:

$ docker kill <container\_id>

### **25. Can you use a container, edit it, and update it? Also, how do you make it a new and store it on the local system?**

Of course, you can use a container, edit it and update it. This sounds complicated but its actually just one command.

$ docker commit <conatainer id> <username/imagename>

### **26. Once you’ve worked with an image, how do you push it to docker hub?**

$ docker push <username/image name>

### **27. How to delete a stopped container?**

Use the following command to delete a stopped container:

$ docker rm <container id>

### **28. How to delete an image from the local storage system?**

The following command lets you delete an image from the local system:

$ docker rmi <image-id>

### **29. How to build a Dockerfile?**

Once you’ve written a Dockerfile, you need to build it to create an image with those specifications. Use the following command to build a Dockerfile:

$ docker build <path to docker file>

### **34. Can I use JSON instead of YAML for my compose file in Docker?**

You can use JSON instead of YAML for your compose file, to use JSON file with compose, specify the JSON filename to use, for eg:

$ docker-compose -f docker-compose.json up

**Kubernetes: orchestration and scheduling tool**

**Why we use it.**

Linux and scripting:

Su and Su-

AWS and Azure:

CM tools:

Ansible/Puppet:

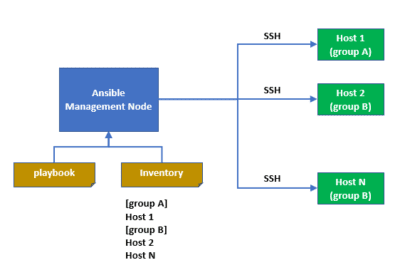
**2) What's the use of Ansible?**

Ansible is used for managing IT infrastructure and deploy software apps to remote nodes.

For example, Ansible allows you to deploy as an application to many nodes with one single command. However, for that, there is a need for some programming knowledge to understand the ansible scripts.

**webapp.yml** is an Ansible playbook file written in YAML format. [Ansible Playbooks](https://docs.ansible.com/ansible/latest/user_guide/playbooks.html) are Ansible’s configuration, deployment, and orchestration language. They can describe a policy you want your remote systems to enforce or a set of steps in a general IT process. These playbooks use YAML file format to define a model of a configuration or a process.

Ansible includes a suite of modules for interacting with Azure Resource Manager, giving you the tools to easily create and orchestrate infrastructure on the Microsoft Azure Cloud.



Automation/

Monitoring

### **What is Docker Swarm?**

You are expected to have worked with Docker Swarm as it’s an important concept of Docker.

Docker Swarm is native clustering for Docker. It turns a pool of Docker hosts into a single, virtual Docker host. Docker Swarm serves the standard Docker API, any tool that already communicates with a Docker daemon can use Swarm to transparently scale to multiple hosts.