**1)What is DevOps beginner?**

DevOps is a society which supports collaboration between Development including Operations Team to deploy key to increase faster in an automated & repeatable way. In innocent words, DevOps backside is established as an association of development and IT operations including excellent communication and collaboration.

**2)What do the main benefits of DevOps?**

With a single team composed of cross-functional comments simply working in collaboration, DevOps organizations container produce including maximum speed, functionality, including innovation. Where continue special benefits: Continuous software control. Shorter complexity to manage.

**3)What are the DevOps Tools and there uses?**

The most popular DevOps tools are mentioned below

• Git. Git is one from the most successful DevOps tools, widely applied across the specific software industry.

• Jenkins. Jenkins is that go-to DevOps automation tool for many software community teams.

• Selenium Continuous Testing tool

• Puppet, Chef, Ansible are Configuration Management and Deployment tools

• Nagios Continuous Monitoring tool

• Docker Containerization tool

• Gradle. Your DevOps device stack will need a reliable build tool.

**4)What is the roles and responsibilities of the DevOps engineer?**

DevOps Engineer manages with developers including the IT system to manage the code releases. They are both developers cases become interested in deployment including practice settings or sysadmins who convert a passion for scripting and coding more move toward the development front where all can improve that planning from test and deployment.

**5) What are the tasks also responsibilities of DevOps engineer?**

In this role, you’ll work collaboratively including software engineering to use and operate our systems. Help automate also streamline our procedures and processes. Build also maintain tools for deployment, monitoring, including operations. And troubleshoot and resolve problems in our dev, search and production environments

**5) Advantages of DevOps Below are the major advantages**

Technical:

1. Continuous software delivery

2. Less Complexity

3. Faster

Resolution Business:

1. Faster delivery of the features

2. More stable operating environment

3. Improved communication and collaboration between various teams

**Linux:**

**Which command is used to check the size of file or directory?**

The command used to check the size of the file or directory is “du”. Here “du” stands for disk usage that is used to check information of disk usage of files and directories on a machine. It is also used to display files and directory sizes in a recursive manner.

**What is the function of grep command?**

Grep (Global regular expression print) is a command that is used to the global search for a string of characters in a specified file. The text search pattern is generally known as a regular expression. It simply makes use of pattern-based searching.   Syntax: grep [options] pattern [files] Example: $ grep -c "linux" interview.txt   The above command will usually print the total count of the word “Linux” in the file “interview.txt”.

**What is pipe?**

In Linux, a pipe is basically a form of redirection that is used to send the output of one command to another command for further processing. It simply takes the output from one command and uses it as an input for another. It provides asynchronous execution of commands with help of buffered I/O routines.

**What are file permissions in Linux?**

Name different types of file systems in Linux. There are three owners in the Linux System i.e., user, group, and others.

These owners have three types of permissions defined as listed below:

**Read (r):** It allows the user to open and read the file or list the directory.

**Write (w):** It allows the user to open and modify the file. One can also add new files to the directory. **Execute (x):** It allows the user to execute or run the file.  One can also lookup a specific file within a directory.

**Command is a command that is used to check connection status between source and destination Machine?**

Linux **ping (Packet Internet Groper)** command is a command that is used to check connection status between source and destination. In simple words, this command is used to check whether a network is available and if the host is reachable. It can also be used to troubleshoot different connectivity issues, verify connectivity at an IP - level to a second TCP/IP device, and name resolution. One can use this command to test both the computer name and IP address of the computer.

**What is Source Code Management?**

It is a process through which we can store and manage any code. Developers write code, Testers write test cases and DevOps engineers write scripts. This code, we can store and manage in Source Code Management. Different teams can store code simultaneously. It saves all changes separately. We can retrieve this code at any point of time.

**What is Git?**

Git is one of the Source Code Management tools where we can store any type of code. Git is the most advanced tool in the market now. We also call Git is version control system because every update stored as a new version. At any point of time, we can get any previous version. We can go back to previous versions. Every version will have a unique number. That number we call commit-ID. By using this commit ID, we can track each change i.e. who did what at what time. For every version, it takes incremental backup instead of taking the whole backup. That’s why Git occupies less space. Since it is occupying less space, it is very fast.

**What is GitHub?**

Git hub is central git repository where we can store code centrally. Git hub belongs to Microsoft Company. We can create any number of repositories in Git hub. All public repositories are free and can be accessible by everyone. Private repositories are not free and can restrict public access for security. We can copy the repository from one account to other accounts also. This process we call as “Fork”. In this repository also we can create branches. The default branch is “Master”.

**What is Git merge?**

By default, we get one branch in git local repository called “Master”. We can create any no of branches for parallel development. We write code for each feature in each branch so that development happens separately. Finally, we merge code off all branches in to Master and push to central repository. We can merge code to any other branch as well. But merging code into master is standard practice that being followed widely. Sometimes, while merging, conflict occurs. When same file is in different branches with different code, when try to merge those branches, conflict occurs. We need to resolve that conflict manually by rearranging the code.

**What is Git Reset?**

Git Reset command is used to remove changes form staging area. This is bringing back file form staging area to work directory. We use this command before commit. Often we go with git add accidentally. In this case if we commit, that file will be committed. Once you commit, commit ID will be generated and it will be in the knowledge of everyone. So to avoid this one, we use Git reset. If you add “–hard” flag to git reset command, in one go, file will be removed from staging area as well as working directory. We generally go with this one if we fell that something wrong in the file itself.

**Difference between Git pull and Git clone?**

We use these two commands to get changes from central repository. For the first time if you want whole central repository in your local server, we use git clone. It brings entire repository to your local server. Next time onwards you might want only changes instead of whole repository. In this case, we use Git pull. Git clone is to get whole copy of central repository Git pull is to get only new changes from central repository (Incremental data)

**What is the difference between Git pull and Fetch?**

We use Git pull command to get changes from central repository. In this operation, internally two commands will get executed. One is Git fetch and another one is Git merge. Git fetch means, only bringing changes from central repo to local repo. But these changes will not be integrated to local repo which is there in your server. Git merge means, merging changes to your local repository which is there in your server. Then only you can see these changes. So Git pull is the combination of Git pull and Git merge.

**What is the difference between Git merge and rebase?**

We often use these commands to merge code in multiple branches. Both are almost same but few differences. When you run Git merge, one new merge commit will be generated which is having the history of both development branches. It preserves the history of both branches. By seeing this merge commit, everyone will come to know that we merged two branches. If you do Git rebase, commits in new branch will be applied on top of base branch tip. There won’t be any merge commit here. It appears that you started working in one single branch form the beginning. This operation will not preserves the history of new branch

**Maven:**

**What is Maven?**

Maven is one of the Build tools. It is the most advance build tool in the market. In this, everything is already pre-configured. Maven belongs to Apache Company. We use maven to build Java code only. We can’t build other codes by using Maven. By default, we get so many plugins with Maven. You can write your own plug-in as well. Maven’s local repository is “.M2” where we can get required compilers and dependencies. Maven’s main configuration file is “pom.xml” where we keep all instructions to build.

**What is Maven’s Build Life Cycle?**

In maven, we have different goals. These are

• Generate resources (Dependencies)

• Compile code : mvn compile

• Unit test : mvn test

• Package (Build): mvn package

• Install (in to local repo & artifactory) : mvn install

• Deploy (to servers) : mvn deploy

• Clean (delete all run time files : mvn clean

**Maven Commands**

**What does POM.XML contains?**

POM.XML is maven’s main configuration file where we keep all details related to project.

It contains

• Metadata about that project

• Dependencies required to build the project

• The kind of project

• Kind of output you want (.jar, .war)

• Description about that project

**Jenkins:**

**What are the ways to install Jenkins?**

Jenkins can be installed using –

**1. Native System Package Manager like** - apt (Linux), brew (Mac), etc.

**2. Docker (popular docker images** for Jenkins is available for different platforms like Unix/Mac/Windows in the docker registry)

**3. Kubernetes** (available as a helm chart and can be installed on our Kubernetes clusters)

**4. Standalone** (on any machine with a Java Runtime Environment installed) For more detailed installation instructions refer official documentation

**What software requirements do I need to install Jenkins?**

* The below table lists the minimum software requirements, which need to be present on a machine on which we are planning to install the Jenkins: Java Development Kit (JDK) or Java Runtime Environment (JRE). Kindly refer to the tutorial How to Install Java.

**What is CI/CD?**

**Continuous Integration** is something that is used for streamlining the development and deployment process. These lead to the more rapid development of cohesive software.

**Continuous Delivery** is on the other hand is a process where your code after being pushed to a remote repository can be taken to production at any time.

**What are the common use cases Jenkins is used for?**

anity/Smoke/CI/Regression test jobs

Web/Data Scraping related jobs

Code coverage measurement jobs

General-purpose automation

Reverse Engineering jobs

Key Decoding jobs & many other jobs where so

**) What is a Jenkins job?**

Job/Project is the fundamental unit of a logical work (like a software build, an automation task, test execution, etc) using the Jenkins automation server and other required plugins, configurations & infrastructures.

**Jobs can be of different types like** - a freestyle project, a multi-configuration project, a pipeline project, a multi-branch project, etc.

**) What is a Jenkins Pipeline?**

The pipeline is a special type of Jenkins job - simply a sequence of steps controlled by a defined logic - which Orchestrates long-running activities that can span across multiple build agents. It is suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that cannot be easily achieved using a freestyle job.

**) What are the types of Jenkins pipelines?**

Jenkins Pipelines can be either - a **Declarative pipeline or a Scripted Pipeline**.

Declarative pipeline makes use of numerous, generic, predefined build steps/stages (i.e. code snippets) to build our job according to our build/automation needs whereas, with Scripted pipelines, the steps/stages can be custom-defined & used using a groovy syntax which provides better control & fine-tuned execution levels.

**) Where is Jenkins default password stored?**

* The Jenkins default password is stored in the $**JENKINS\_HOME/secrets/initialAdminPassword file** (the exact location of the Jenkins default password is indicated in the Jenkins console log). Cool Tip: Reset Jenkins admin password! Read more → The Jenkins default username is admin.

**How do I install a Jenkins plugin?**

* Navigate to the Manage Jenkins > Plugins page in the web UI. Choose the .hpi file from your system or enter a URL to the archive file under the Deploy Plugin section. Deploy the plugin file. Once a plugin file has been uploaded, the Jenkins controller must be manually restarted in order for the changes to take effect.

**Name some of the useful plugins in Jenkins.**

Some of the plugins in Jenkins include:

* [Maven 2 project](https://www.simplilearn.com/tutorials/maven-tutorial/maven-project-in-eclipse" \t "_blank" \o "Maven 2 project)
* [Amazon EC2](https://www.simplilearn.com/tutorials/aws-tutorial/aws-ec2" \t "_blank" \o "Amazon EC2)

**Name the two components that Jenkins is mostly integrated with.**

Jenkins is typically integrated with these two components:

1. Version Control systems like Git and SVN (Apache Subversion)
2. Build tools like [Maven](https://www.simplilearn.com/tutorials/maven-tutorial/what-is-maven" \t "_blank" \o "Maven)

### What is Jenkins file?

Jenkins file is a text file that has a definition of a Jenkins pipeline and is checked into the source control repository. It enables code review and iteration on the pipeline. It also permits an audit trail for the pipeline.

**How to make a backup of Jenkins?**

* Follow the steps given below to have a backup in place. Step 1 − Click on Manage Jenkins and choose the ‘Manage Plugins’ option. Step 2 − In the available tab, search for ‘**Backup Plugin’**. Click On Install without Restart. Once done, restart the Jenkins instance

How can the parameters be defined in Jenkins?

In Jenkins, a build can take many input parameters to execute.

* To define parameters for the job, select the “this project is parameterized” box.
* The drop down “Add Parameter” is enabled with the parameter types list. Any number of parameters may be added in the list.

There are several parameter types provided in the list.

 Explain the ways to configure Jenkins node agent to communicate with Jenkins master?

There are two ways to configure Jenkins node agent to communicate with Jenkins master:

* 1. Browser–If we launch the Jenkins node agent from a browser, a Java Web Start or JNLP file is downloaded. The downloaded file launches a new process on the client machine to run jobs.
  2. Command-line–If you want to start the node agent using the command line, you need the executable agent.jar file. When this file runs, it launches a client's process to communicate with the Jenkins master to run build jobs.

**Docker:**

**What is the Docker?**

Docker is a tool by using which, we create containers in less time. Docker uses light weight OS in the form of docker images that we will get from docker hub. Docker is open source now. It became so popular because of its unique virtualization concept called “Containerization” which is not there in other tools. We can use docker in both windows and Linux machines

**List of Docker components?**

**Docker image:** – Contains OS (very small) (almost negligible) + soft wares

**Docker Container:** – Container like a machine which is created from Docker image.

**Docker file:** – Describes steps to create a docker image.

**Docker hub/registry:** – Stores all docker images publicly.

**Docker daemon:** – Docker service runs at back end Above five components we call as Docker components

**What is difference between docker image and docker container?**

Docker image is a read only template that contains the instructions for a container to start. Docker container is a runnable instance of a docker image.

**What are the most common instructions in Docker file?**

**FROM:** This is used to set the base image for upcoming instructions. A docker file is considered to be valid if it starts with the FROM instruction.

**LABEL:** This is used for the image organization based on projects, modules, or licensing. It also helps in automation as we specify a key-value pair while defining a label that can be later accessed and handled programmatically.

**RUN:** This command is used to execute instructions following it on the top of the current image in a new layer. Note that with each RUN command execution, we add layers on top of the image and then use that in subsequent steps.

**CMD:** This command is used to provide default values of an executing container. In cases of multiple CMD commands the last instruction would be considered.

**Differentiate between COPY and ADD commands that are used in a Dockerfile?**

Both the commands have similar functionality, but COPY is more preferred because of its higher transparency level than that of ADD .

COPY provides just the basic support of copying local files into the container whereas ADD provides additional features like remote URL and tar extraction support.

**Important docker commands?** ‘

1. Docker ps (to see list of running containers)

2. Docker ps -a (to see list of all containers)

3. Docker images (to see list of all images)

4. Docker run (to create docker container)

5. Docker attach (to go inside container)

6. Docker stop (to stop container)

7. Docker start (to start container)

8. Docker commit (to create image out of docker file)

9. Docker rm (to delete container) 10. Docker rmi (to delete image

**What can you tell about Docker Compose?**

It is a YAML file consisting of all the details regarding various services, networks, and volumes that are needed for setting up the Docker-based application. So, dockercompose is used for creating multiple containers, host them and establish communication between them. For the purpose of communication amongst the containers, ports are exposed by each and every container.

**Ansible :**

**What is Ansible?**

Ansible is one of the configuration Management Tools. It is a method through we automate system admin tasks. Configuration refers to each and every minute details of a system. If we do any changes in system means we are changing the configuration of a machine. That means we are changing the configuration of the machine. All windows/Linux system administrators manage the configuration of a machine manually. All DevOps engineers are managing this configuration automatic way by using some tools which are available in the market. One such tool is Ansible. That’s why we call Ansible as configuration management tool.

**Ansible components?**

**Server: –** It is the place where we create playbooks and write code in YML format

**Node: –** It is the place where we apply code to create infrastructure. Server pushes code to nodes. **Ssh: –** It is an agent through ansible server pushes code to nodes.

**Setup: –** It is a module in ansible which gathers nodes information.

**Inventory file:-** In this file we keep IP/DNS of nodes.

**Advantages of Ansible over other SCM (Source Code Management) tools?**

• Agentless

• Relies on “ssh”

• Uses python

• Push mechanism

**What do you mean by Ad-Hoc commands in Ansible**?

These are simple one liner Linux commands we use to meet temporary requirements without actually saving for later. Here we don’t use ansible modules. So there, Idempotency will not work with Ad-Hoc commands. If at all we don’t get required YAML module to write to create infrastructure, then we go for it. Without using playbooks we can use these Ad-Hoc commands for temporary purpose.

**What is the ad-hoc command in Ansible?**

Ad-hoc commands are like one-line playbooks to perform a specific task only. The syntax for the ad-hoc command is

**ansible [pattern] -m [module] -a "[module options]"**

For example, we need to reboot all servers in the staging group

**ansible atlanta -a "/sbin/reboot" -u username --become [--ask-become-pass]**

**What are Ansible tasks?**

The task is a unit action of Ansible. It helps by breaking a configuration policy into smaller files or blocks of code. These blocks can be used in automating a process. For example, to install a package or update a software.

EX: Install <package\_name> , update <software\_name>

**Install apache2 using ansible-playbook?**

ubuntu@ip-172-31-12-17:~/ansibledemo$ cat install-apache.yml

---

- hosts: all

become: true

tasks:

- name: Install apache

apt: name=apache2 state=present

**What is EC2?**

EC2, a Virtual Machine in the cloud on which you have OS-level control. You can run this cloud server whenever you want and can be used when you need to deploy your own servers in the cloud, similar to your on-premises servers, and when you want to have full control over the choice of hardware and the updates on the machine.

**What is CloudWatch?**

CloudWatch helps you to monitor AWS environments like EC2, RDS Instances, and CPU utilization. It also triggers alarms depending on various metrics.

**Explain what S3 is?**

S3 stands for Simple Storage Service. You can use the S3 interface to store and retrieve any amount of data, at any time and from anywhere on the web. For S3, the payment model is “pay as you go”.

What is Elastic Ip & Why do we need?

**What are the different types of Instances?**

Following are the types of instances:

Compute Optimized

Memory-Optimized

Storage Optimized

Accelerated Computing

General Purpose

**What are the advantages of AWS IAM?**

AWS IAM enables an administrator to provide granular level access to different users and groups. Different users and user groups may need different levels of access to different resources created. With IAM, you can create roles with specific access-levels and assign the roles to the users. It also allows you to provide access to the resources to users and applications without creating the IAM Roles, which is known as Federated Access.

**What do you understand by VPC?**

VPC stands for Virtual Private Cloud. It allows you to customize your networking configuration. VPC is a network that is logically isolated from other networks in the cloud. It allows you to have your private IP Address range, internet gateways, subnets, and security groups.

### What is Kubernetes?

Kubernetes is a container management system developed in the Google platform. The purpose of [Kubernetes](https://www.guru99.com/kubernetes-tutorial.html) is to manage a containerized application in various types of physical, virtual, and cloud environments. Google Kubernetes is a highly flexible container tool to deliver even complex applications, consistently. Applications run on clusters of hundreds to thousands of individual servers.

### What is the work of a kube-scheduler?

Kube-scheduler is the default scheduler for Kubernetes. It assigns nodes to newly created pods.

### Define node in Kubernetes

A node the smallest unit of hardware. It defines a single machine in a cluster that can be a virtual machine from a cloud provider or physical machine in the data center. Every machine available in the Kubernetes cluster can substitute other machines.

### Why use namespace in Kubernetes?

Namespaces in Kubernetes are used for dividing cluster resources between users. It helps the environment where more than one user spread projects or teams and provides a scope of resources.

* **Master Node:**The master node is the first and most vital component which is responsible for the management of Kubernetes cluster. It is the entry point for all kinds of administrative tasks. There may be more than one master node in the cluster to check for fault tolerance.
* **API Server:** The API server acts as an entry point for all the REST commands used for controlling the cluster.
* **Scheduler:** The scheduler schedules the tasks to the slave node. It stores the resource usage information for every slave node. It is responsible for distributing the workload.
* **Etcd:**etcd components, store configuration detail, and wright values. It communicates with the most component to receive commands and work. It also manages network rules and port forwarding activity.
* **Worker/Slave nodes:**Worker nodes are another essential component that contains all the required services to manage the networking between the containers, communicate with the master node, which allows you to assign resources to the scheduled containers.
* **Kubelet:** It gets the configuration of a Pod from the API server and ensures that the described containers are up and running.
* **Docker Container:** Docker container runs on each of the worker nodes, which runs the configured pods.
* **Pods:** A pod is a combination of single or multiple containers that logically run together on nodes.

**13) List various services available in Kubernetes**

Various services available in Kubernetes are 1) Cluster IP service, 2) Load Balancer service, 3) Node Port service, 4) External Name Creation service.

### Define kubelet

The kubelet is a service agent which controls and maintains group pf pods by checking pod specification using Kubernetes. The kubelet runs on each node and allows to communicate between a master node and a slave node.

**What is the difference between Kubernetes and Docker Swarm?**

The difference between Kubernetes and Docker Swarm is:

| **Kubernetes** | **Docker Swarm** |
| --- | --- |
| Kubernetes Provides an auto-scaling feature. | Docker Swarm does not provide an auto-scaling feature. |
| Manually configure your load balancing settings. | Does auto load balancing |
| Installation is complicated & time-consuming. | Installation is easy & fast. |
| GUI is available. | GUI not available. |
| It provides a built-in load balancing technique. | Process scheduling is done to maintain services while updating. |

**Explain the types of Kubernetes pods**

There are two types of pods in Kubernetes:

* **Single Container Pod:** It can be created with the run command.
* **Multicontainer pods:** It can be created using the “create” command in Kubernetes.

**Mention the difference between Docker volumes and Kubernetes Volumes**

| **Kubernetes Volumes** | **Docker Volumes** |
| --- | --- |
| Volumes are not limited to any container. | Volumes are limited to a pod in the container. |
| Kubernetes volumes support all containers deployed in a pod of Kubernetes. | Docker volumes do not support all containers deployed in Docker. |
|  |  |

****Amazon Linux vs Ubuntu: What are the differences?****

****What is Amazon Linux?****

The Amazon Linux AMI is a supported and maintained Linux image provided by Amazon Web Services for use on Amazon EC2. The Amazon Linux AMI is a supported and maintained Linux image provided by Amazon Web Services for use on Amazon Elastic Compute Cloud (Amazon EC2).

****What is Ubuntu?****

The leading OS for PC, tablet, phone and cloud. Ubuntu is an ancient African word meaning ‘humanity to others’. It also means ‘I am what I am because of who we all are’. The Ubuntu operating system brings the spirit of Ubuntu to the world of computers.

Amazon Linux and Ubuntu can be categorized as ****"Operating Systems"**** tools.

According to the StackShare community, Ubuntu has a broader approval, being mentioned in ****1870**** company stacks & ****1757**** developers stacks; compared to Amazon Linux, which is listed in ****7**** company stacks and ****23**** developer stacks.

Ubuntu is much more faster over Windows and helps to get software and other utilities easier and within a short span of time compared to Windows.

Ubuntu helps to get robustness and resiliency over Windows. Ubuntu runs faster than Windows on every computer that I have ever tested. LibreOffice (Ubuntu's default office suite) runs much faster than Microsoft Office on every computer that I have ever tested.

* Which linux system will use in IT industry?
* Linux servers are often chosen over other server operating systems for their stability, security, and flexibility. Leading Linux server operating systems include CentOS, Debian, Ubuntu Server, Slackware, and Gentoo.