

## Linux

**How to copy a file in Linux?** \$ cp <source> <destination> Suppose you want to copy a file named questions.txt from the directory /new/linux to /linux/interview, then the command will be: \$ cp questions.txt /new/linux /linux/interview

**What are inode and process id?** inode is the unique name given by the operating system to each file.

Similarly, process id is the unique id given to each process.

**Command to check how much memory is being used by Linux?** free -m, vmstat, top, htop

**What is Swap Space?** Swap Space is the additional spaced used by Linux that temporarily holds concurrently running programs when the RAM does not have enough space to hold the programs. When you run a program, it resides on the RAM so that the processor can fetch data quickly. Suppose you are running more programs than the RAM can hold, then these running programs are stored in the Swap Space. The processor will now look for data in the RAM and the Swap Space. Swap Space is used as an extension of RAM by Linux

**How can you check the memory status?** free -m to display output in MB, free -g to display output in GB

**What are symbolic links?** Symbolic links act similarly to shortcuts in Windows. Such links point to programs, files or directories. It also allows you instant access to it without having to go directly to the entire pathname.

**What are hard links?** Hard links point directly to the physical file on disk, and not on the pathname. This means that if you rename or move the original file, the link will not break since the link is for the file itself, not the path where the file is located.

**5 VPC Elastic IP addresses are allowed for each AWS account.**

**Explain default storage class in S3?** The default storage class is a Standard frequently accessed.

**What are the Roles?** Roles are used to provide permissions to entities which you can trust within your AWS account. Roles are very similar to users. However, with roles, you do not require to create any username and password to work with the resources.

**Explain the GIT push command.**

The GIT push command is used to push the content in a local repository to a remote repository. After a local repository has been modified, a push is executed to share the modifications with remote team members.

**Explain the GIT pull command?** GIT pull is used to fetch and merge changes from the remote repository to the local repository. GIT pull is a combination of two commands: git fetch; followed by git merge.

**Difference between GIT fetch and GIT pull.?**

**GIT Fetch:** GIT fetches only downloads new data from a remote repository.

It does not integrate any of these new data into your working files.

**GIT pull:** Can be done any time to update the remote-tracking branches

Command - git fetch origin  
git fetch --all

GIT pull updates the current HEAD branch with the latest changes from the remote server.

Downloads new data and integrate it with the current working files.

Tries to merge remote changes with your local ones.

Command - git pull origin master

**What is a merge conflict in GIT?** A merge conflict is an event that takes place when GIT is unable to resolve differences in code between the two commits automatically.

GIT is able to automatically merge the changes only if the commits are on different lines or branches.

**How do you resolve a merge conflict?**

The simplest way to resolve the conflicted file is to open it and make the required changes. After editing the file, we can use the git add command to stage the new merged content. The final step is to create a new commit with the help of the git commit command. GIT will create a new merge commit to finalize the merge.

**What is GIT stash?** Let's say you're a developer and you want to switch branches to work on something else. The issue is you don't want to make commits in uncompleted work, so you just want to get back to this point later. The solution here is the GIT stash. GIT stash takes your modified tracked files and saves it on a stack of unfinished changes that you can reapply at any time. To go back to the work you can use the stash pop.

**What is the difference between fork, branch, and clone?**

**Fork:** Process when a copy of the repository is made. It's usually experimentation in the project without affecting the original project. They're used to advise changes or take inspiration from someone else's project.

**Branch:** GIT branches refer to individual projects within a git repository. If there are several branches in a repository, then each branch can have entirely different files and folders.

**Clone:** GIT clone refers to creating a clone or a copy of an existing git repository in a new directory. Cloning automatically creates a connection that points back to the original repository, which makes it very easy to interact with the central repository.

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

To incorporate new commits into your feature branch, you use merge

Creates an extra merge commit every time you need to incorporate changes

Pollutes your feature branch history

As an alternative to merging, you can rebase the feature branch into master.

Incorporates all the new commits in the master branch

Rewrites the project history by creating brand new commits for each commit in the original branch

**How is 'GIT remote' different from 'GIT clone'?**

**Cloud formation is an orchestration or infrastructure or server deployment tool**

Managed server deployment which Amazon provides, It takes JSON as an input, It build entire infrastructure stats

AWS Cloud Formation provides a common language for you to model and provision AWS and third part application resources in your cloud environment. AWS Cloud Formation allows you to use programming Languages or a simple text file to model and provision, in an automated and secure manner, all the resources needed for your applications across all regions and accounts. This gives you a single source of truth for your AWS and third party resources. It is basically a service. Given a scenario where we have executable file, we first install the application. Subsequently, a folder is created and certain files of a package are copied. In short, it is a step to be followed with all files to be executed.

- 1) Creation of a Security Group
- 2) Creation of a Key Pair File
- 3) Launch of an EC2 Instance
- 4) Installation of Apache, MySQL or any other file.
- 5) Get Code from S3 and deploy in Web Server
- 6) Creation of Elastic Load Balancer and give the URL.

Template in Amazon Cloud Formation

It consists of 6 main objects: Format version, Description, Parameters, Mappings, Resources, and Output

**Security Group With VPC :** A security groups acts a virtual firewall for your instance to control inbound and outbound traffic

Security Groups act at the instance level, not the subnet level

Can create up to 500 security groups per VPC

Can specify allow rules, but not deny rules

Internet Gateways helps you public IP available

NAT gateway is used for private subnets for outside

Virtual Private Gateway is used to connect VPN connection

**Private IP address :** Private IP addresses are not reachable over the internet. It can be used for communication between the instances in the VPC

**Public IP address:** Public IP address are reachable over the internet, It can be used for communication between your instances and the internet, or with other AWS services that have public endpoints.

**Internet gateway** is only one thing which determines public subnet or a private Subnet.

**Route Tables:** A route table consists a set of rules called routes, that are used to determine where network traffic is directed

Each subnet in VPC must be associated with a route table

**Access Control Lists (ACL):** An optional layer of security for VPC that acts as a firewall for controlling traffic in and out of one or more subnets. A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.

**Note :** The set of ACL permissions is the same for object ACL and bucket ACL

NAT (Network Address Translation)

**AWS Snowball:** It is a migrating service which can move resources from on Premises to cloud. Snowball is a physical device which is delivered to offices or homes. It looks like a briefcase When ordered it charges 250 \$ for shipping and copy the data up to 80 TB up to a snowball and then send it to AWS nearest data centers, They will copy the data using their internal resources. Snow ball is a device which helps to copy the data. Snowball is the fastest means of migrating the data

**Amazon Cloud Front:** Helps to provide the distribution of content delivery network. It helps in casting the services using edge location. Edge location are subsets of data centers

**EFS:** same files system can be mapped to different OS

Network files system are those where one file system can be mapped to multiple file system

1 EBS volume can be attached to 1 OS at one time

S3 has either magnetic tape attached to it or volume attached to it

Static websites are nothing but index.html

They are static in nature

Something which cannot take dynamic request

Bucket can be created via Access Control List or bucket policy

By default object is private, we have to make it public

If you want an object to create permissions just create a bucket policy

Old objects will go after creating a public policy

**Vertical Scaling :** Resizing the existing server is called vertical scaling When you add a no other server is called Horizontal Scaling

**What is Glacier?** Amazon Glacier is a low-cost cloud storage service for data with longer retrieval times offered by Amazon Web Services (AWS). A developer uses a cold data cloud service such as Amazon Glacier to move infrequently accessed data to archival storage to save money on storage costs

Amazon Simple Storage Service is storage for the Internet. It is designed to make web-scale computing easier for developers.

Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data,

**Git remote:** GIT remote enables you to create, view, and delete connections to other repositories. It's used to refer to a remote repository or a central repository.

**Git clone:** GIT clone enables you to create a clone or copy of the target repository. It's used to target a different already existing repository.

#### What's the difference between reverting and resetting?

**Reverting:** revert command in GIT is used to create a new commit that undoes the changes made in the previous commit. When you use this command, a new history is added to the project; the existing history is not modified.

**Resetting:** GIT reset is a command that is used to undo the local changes that have been made to a Git repository. GIT reset operates on the following: commit history, the staging index, and the working directory.

#### What is "git cherry-pick"?

The command git cherry-pick enables you to pick up commits from a branch within a repository and apply it to another branch. This command is useful to undo changes when any commit is accidentally made to the wrong branch. Then, you can switch to the correct branch and use this command to cherry-pick the commit.

#### What is a 'conflict' in GIT?

GIT can handle on its own most merges by using its automatic merging features. There arises a conflict when two separate branches have made edits to the same line in a file, or when a file has been deleted in one branch but edited in the other. Conflicts are most likely to happen when working in a team environment

#### Describe the branching strategies you have used.

**Feature branching** – A feature branch model keeps all of the changes for a particular feature inside of a branch. When the feature is fully tested and validated by automated tests, the branch is then merged into master.

**Task branching** – In this model, each task is implemented on its own branch with the task key included in the branch name. It is easy to see which code implements which task, just look for the task key in the branch name.

**Release branching** – Once the develop branch has acquired enough features for a release, you can clone that branch to form a Release branch. Creating this branch starts the next release cycle, so no new features can be added after this point, only bug fixes, documentation generation, and other release-oriented tasks should go in this branch. Once it is ready to ship, the release gets merged into master and tagged with a version number. In addition, it should be merged back into the develop branch, which may have progressed since the release was initiated.

In the end tell them that branching strategies vary from one organization to another.

**What is the difference between rebasing and merge in GIT?** In Git, the rebase command is used to integrate changes from one branch into another. It is an alternative to the "merge" command. The difference between rebasing and merge is that rebase rewrites the commit history in order to produce a straight, linear succession of commits.

Merging is GIT's way of putting a forked history back together again. The git merge command helps you take the independent lines of development created by git branch and integrate them into a single branch.

**Can you explain the GIT flow workflow?** To record the history of the project, Gitflow workflow employs two parallel long-running branches – master and develop:

Master – this branch is always ready to be released on LIVE, with everything fully tested and approved (production-ready).

Hotfix – these branches are used to quickly patch production releases. These branches are a lot like release branches and feature branches except they're based on master instead of develop.

Develop – this is the branch to which all feature branches are merged and where all tests are performed. Only when everything's been thoroughly checked and fixed it can be merged to the master.

Feature – each new feature should reside in its own branch, which can be pushed to the develop branch as their parent one.

#### Docker :

**What is Hypervisor?** A hypervisor is software that makes virtualization possible. It is also called as virtualization monitor. It divides the host system and allocates the resources to each divided virtual environment. You can basically have multiple OS on a single host system.

The hypervisor allows you to create a virtual environment in which the guest virtual machines operate. It controls the guest systems and checks if the resources are allocated to the guests as necessary.

#### Difference between virtualization and containerization?

Once you've explained containerization and virtualization, the next expected question would be differences. The question could either be differences between virtualization and containerization or differences between virtual machines and containers. Either way, this is how you respond.

Containers provide an isolated environment for running the application. The entire user space is explicitly dedicated to the application. Any changes made inside the container is never reflected on the host or even other containers running on the same host. Containers are an abstraction of the application layer. Each container is a different application.

Whereas in Virtualization, hypervisors provide an entire virtual machine to the guest (including Kernel). Virtual machines are an abstraction of the hardware layer. Each VM is a physical machine

**What are Docker Images?** Docker image is the source of Docker container. In other words, Docker images are

at any time, from anywhere on the web.

**What is an example of an on premise establishment?** A local pub qualifies as an on premise establishment because the sale of goods or services can be consumed or used as intended while remaining at the establishment.

**What is on premise and off premise cloud?** The biggest difference between these two systems is how they are deployed.

Cloud-based software is hosted on the vendor's servers and accessed through a web browser.

On-premise software is installed locally, on a company's own computers and servers

**On premise:** On-premises software (also known as on-premise, and abbreviated "on-prem") is installed and runs on computers on the premises of the person or organization using the software, rather than at a remote facility such as a server farm

#### General DevOps

##### What is the difference between Continuous Delivery and Continuous Deployment?

**Continuous Delivery:** (Manual Deployment to Production. Does not involve every change to be deployed.)

**Continuous Delivery** is a software development practice where you build software in such a way that the software can be released to the production at any time. You achieve Continuous Delivery by continuously integrating the products built by the development team, running automated tests on those built products to detect problems and then push those files into production-like environments to ensure that the software works in production.

**Continuous deployment** means that every change that you make goes through the pipeline, and if it passes all the tests, it automatically gets deployed into production. So, with this approach, the quality of the software release completely depends on the quality of the test suite as you have automated everything.

**Elastic search** allows you to store, search, and analyze huge volumes of data quickly and in near real-time and give back answers in milliseconds. It's able to achieve fast search responses because instead of searching the text directly, it searches an index. **Elastic search** takes in unstructured data from different locations, stores and indexes it according to user-specified mapping (which can also be derived automatically from data), and makes it searchable. Its distributed architecture makes it possible to search and analyze huge volumes of data in near real time

**How to scale a production web service?** If you're on the cloud we can use auto scaling group based which will scale automatically on demand, If you don't want to go to instances and manage instances you can use Elastic Bean stack for a managed stack, which can take care of the Load Balancers yours

**Define Terraform provider?** Terraform is used to manage and inform infrastructure resources such as bodily machines, VMs, network switches, containers, and more. A provider is accountable for thoughtful API interactions and revealing resources.

#### Ansible

**What is Configuration Management and how does it help an organization?** Configuration Management is the practice of handling updates and changes systematically so that a system maintains its integrity over time. Configuration Management (CM) keeps a track of all the updates that are needed in a system and it ensures that the current design and build state of the system is up to date and functioning correctly.

**Configuration Management can help an organization by overcoming the following challenges:** Finding out what changes need to be implemented when user requirements change.

Redoing and updating an implementation due to change in the requirements since the last implementation. Reverting to an older version of the component because the latest version is flawed.

Replacing the wrong component because you couldn't accurately determine which component needed replacing.

**How does Ansible work?** Ansible, unlike other configuration management tools, is categorized into two types of servers – Controlling machines and Nodes. Controlling machine is where Ansible is installed and nodes are the ones that are managed by the controlling machines through SSH. There is an inventory file in the controlling machine that holds the location of the node systems. Ansible deploys modules on the node systems by running the playbook on the controlling machine. Ansible is agentless, that means there is no need to have a third party tool to make a connection between one node and the other.

**What are roles?** Roles let you automatically load related vars\_files, tasks, handlers, and other Ansible artifacts based on a known file structure. Once you group your content in roles, you can easily reuse them and share them with other users.

**What are playbooks?** Playbooks record and execute Ansible's configuration, deployment, and orchestration functions. They can describe a policy you want your remote systems to enforce, or a set of steps in a general IT process.

#### What are the different components of Ansible? Explain Ansible architecture?

The main component of Ansible is the Ansible automation engine. This engine directly interacts with various cloud services, Configuration Management Database (CMDB) and different users who write various playbooks to execute the Ansible Automation engine.

The Ansible Automation engine consists of the following components:

**Inventories:** These are a list of nodes containing their respective IP addresses, servers, databases, etc. which needs to be managed.

**APIs:** Just like any other API, the Ansible APIs are used for commuting various Cloud services, public or

used to create containers. When a user runs a Docker image, an instance of a container is created. These Docker images can be deployed to any Docker environment.

#### **Are you aware of load balancing across containers and hosts? How does it work?**

While using Docker service with multiple containers across different hosts, you come across the need to load balance the incoming traffic. Load balancing and HAProxy is basically used to balance the incoming traffic across different available (healthy) containers. If one container crashes, another container should automatically start running and the traffic should be re-routed to this new running container. Load balancing and HAProxy works around this concept.

**What is a Docker file?** Docker can build images automatically by reading the instructions from a file called Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using Docker build, users can create an automated build that executes several command-line instructions in succession.

**Tell us something about Docker Compose.?** Docker Compose is a YAML file which contains details about the services, networks, and volumes for setting up the Docker application. So, you can use Docker Compose to create separate containers, host them and get them to communicate with each other. Each container will expose a port for communicating with other containers.

**What is Docker Swarm?** 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.

**Do you know why Docker system prune is used? What does it do?** `$ docker system prune` used to remove all the stopped containers, all the networks that are not used, all dangling images and all build caches.

**Where all do you think Docker is being used?** Simplifying configuration: Docker lets you put your environment and configuration into code and deploy it.

Code Pipeline Management: There are different systems used for development and production. As the code travels from development to testing to production, it goes through a difference in the environment. Docker helps in maintaining the code pipeline consistency.

Developer Productivity: Using Docker for development gives us two things – We're closer to production and development environment is built faster.

Application Isolation: As containers are applications wrapped together with all dependencies, your apps are isolated. They can work by themselves on any hardware that supports Docker.

Debugging Capabilities: Docker supports various debugging tools that are not specific to containers but work well with containers.

Multi-tenancy: Docker lets you have multi-tenant applications avoiding redundancy in your codes and deployments.

Rapid Deployment: Docker eliminates the need to build an entire OS from scratch, reducing the deployment time.

**To look for all containers, use the following command:** `$ docker ps -a`

**How will you monitor Docker in production?** Docker provides functionalities like Docker stats and Docker events to monitor Docker in production. Docker stats provides CPU and memory usage of the container. Docker events provide information about the activities taking place in the Docker daemon.

**Is it a good practice to run Docker compose in production?** Yes, using Docker compose in production is the best practical application of Docker compose. When you define applications with compose, you can use this compose definition in various production stages like CI, staging, testing, etc.

#### **What changes are expected in your Docker compose file while moving it to production?**

Remove volume bindings, so the code stays inside the container and cannot be changed from outside the container.

Binding to different ports on the host.

Specify a restart policy

Add extra services like log aggregator

#### **Are you aware of load balancing across containers and hosts? How does it work?**

While using Docker service with multiple containers across different hosts, you come across the need to load balance the incoming traffic. Load balancing and HAProxy is basically used to balance the incoming traffic across different available (healthy) containers. If one container crashes, another container should automatically start running and the traffic should be re-routed to this new running container. Load balancing and HAProxy works around this concept.

**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.

**Explain Registries?** There are two types of registry is Public Registry and Private Registry

Docker's public registry is called Docker hub, which allows you to store images privately. In Docker hub, you can store millions of images

**What is the command to run the image as a container?** `sudo docker run -i -t alpine /bin/bash`

The common instruction in Docker file is: FROM, LABEL, RUN, and CMD.

**Explain Docker Swarm?** Docker Swarm is native gathering for Docker which helps you to a group of Docker hosts into a single and virtual Docker host. It offers the standard Docker application program interface.

private services.

**Modules:** The modules are used to manage system resources, packages, libraries, files, etc. Ansible modules can be used to automate a wide range of tasks. Ansible provides around 450 modules that automate nearly every part of your environment.

**Plugins:** If you want to execute Ansible tasks as a job, Ansible Plugins can be used. They simplify the execution of a task by building a job like an environment that basically contains pieces of code corresponding to some specific functionality. There are 100s of Plugins provided by Ansible. An example is the Action plugin, which acts as front ends to modules and can execute tasks on the controller before calling the modules themselves.

**Networking:** Ansible can also be used to automate different networks and services. It can do this by creating a playbook or an Ansible role that easily spans different network hardware.

**Hosts:** The Ansible Hosts/ Node systems are machines (Linux, Windows, etc) that are getting automated.

**Playbooks:** Playbooks are simple code files which describe the tasks that need to be executed. The Playbooks are written in YAML format. They can be used to automate tasks, declare configurations, etc.

**CMDB:** It is a database that acts as a storehouse for various IT installations. It holds data about various IT assets (also known as configuration items (CI)) and describes the relationships between such assets.

**Cloud:** It is a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server.

**What are Ansible Server requirements?** If you are a windows user then you need to have a virtual machine in which Linux should be installed. It requires Python 2.6 version or higher.

**Controller Machine:** The Controller machine is responsible for provisioning the servers that are being managed. It is the machine where Ansible is installed.

**Inventory:** An inventory is an initialization file that has details about the different servers you are managing.

**Task:** Each task represents a single procedure that needs to be executed, e.g. Install a library.

**Module:** A module is a set of tasks that can be executed. Ansible has 100s of built-in modules, but you can also create custom ones.

**Role:** An Ansible role is a pre-defined way for organizing playbooks and other files in order to facilitate sharing and reusing portions of provisioning.

**Play:** A task executed from start to finish or the execution of a playbook is called a play.

**Facts:** Facts are global variables that store details about the system, like network interfaces or operating system.

**Handlers:** Are used to trigger the status of a service, such as restarting or stopping a service.

#### **What is the difference between a variable name and an environment variable?**

Variable name :You need to add strings to create variable names

You can easily create multiple variable names by adding strings

We use the ipv4 address for variable names

You need existing variables to access environment variables.

Environment variable: To create environment variables we must refer advanced Ansible playbook

We use `{{ ansible_env.SOME_VARIABLE }}` for remote environment variables.

#### **What are the Ansible Modules? Explain the different types?**

Ansible modules are a small set of programs that perform a specific task. Modules can be used to automate a wide range of tasks. Modules in Ansible are considered to be idempotent or in other words, making multiple identical requests has the same effect as making a single request.

**Core Modules:** These are modules that the core Ansible team maintains and will always ship with Ansible itself. They will also receive a slightly higher priority for all requests than those in the "extras" repos. The source of these modules is hosted by Ansible on GitHub in the Ansible-modules-core.

**Extras Modules:** These modules are currently shipped with Ansible but might be shipped separately in the future. They are also mostly maintained by the Ansible Community. Non-core modules are still fully usable but may receive slightly lower response rates for issues and pull requests.

Popular "extras" modules may be promoted to core modules over time. The source for these modules is hosted by Ansible on GitHub in the Ansible-modules-extras.

#### **What is Ansible Tower?**

It's an enterprise-level web-based solution that increases Ansible's accessibility to other IT teams by including an easy to use UI (user interface). Tower's primary function is to serve as the hub for all of an organization's automation tasks, allowing users to monitor configurations and conduct rapid deployments.

#### **Puppet**

**Puppet Master and Agent Communicates:** Puppet has a master Slave Architecture in which slave has to first send a certificate signing request to master and master has to sign the certificate in order to establish a secure connection between a puppet master and puppet slave. Puppet Slave send a request to puppet master and then puppet master then pushes configuration on Slave

**How Puppet Works?** Requires agent

In one servers there is a puppet agent

In order to automate Puppet agent we should call

**Provisioning :** Give access to server , there is already agent so puppet access comes in Puppet Master and puppet slave

### How can you monitor the Docker in production environments?

Docker states and Docker Events are used to monitoring Docker in the production environment.

### Write a Docker file to create and copy a directory and built it using python modules?

```
FROM python:2.7-slim
```

```
WORKDIR /app
```

```
COPY ./app
```

```
docker build - tag
```

**What happens during Docker host owner maintenance?** Containers will not work during host maintenance, due to which in Docker we maintain high availability which is called as Docker swarm that is a cluster way of maintaining a container, in that case we have n number of machines, A few containers will be running on one machine and other on another machines. Even if a machine goes down for a maintenance issue other containers runs from other machines, they serve the end users requests

**How to view docker logs?** docker logs containerid

**How to view docker logs continuously?** docker logs -f containerid

**What is the command to remove all stopped containers?** docker container prune

**How to remove images?** docker rmi nameoftheimage

**How to remove unused image?** docker image prune -a

**Where are image layers stored?** Image layers will be saved cd /var/lib/docker/image/overlay2

/layer db

ls -l

### Nagios

Nagios will Monitor live applications configured on server, It will not monitor Jenkins job but will check if server goes down where the application is running, It will also monitor any unusual behavior of applications running on server.

**Why Nagios?** It can monitor database servers such as SQL Server, MY SQL, POSTGRE SQL,

It gives application level information(Apache, Postfix LDAP, Citrix)

### AWS

**How do you allow a user to gain access to a specific bucket?** They are: Categorize your instances

Define how authorized users can manage specific servers.

Lockdown your tags

Attach your policies to IAM users

VPC is not resolving the server through DNS. What might be the issue, and how can you fix it?

To fix this problem, you need to enable the DNS hostname resolution, so that the problem resolves itself.

**How do you connect multiple sites to a VPC?** If you have multiple VPN connections, you can provide secure communication between sites using the AWS VPN CloudHub.

### **What are CD and CI, and what is Ansible's relationship with them?**

In CD, developers build software that can be released into production at any given time. CI, on the other hand, consists of each developer uploading regularly scheduled integrations (usually daily), resulting in multiple integrations every day. Ansible is an ideal tool for CI/CD processes, providing a stable infrastructure for provisioning the target environment and then deploying the application to it.

### **What are the differences between NAT Gateways and NAT Instances?**

While both NAT Gateways and NAT Instances serve the same function, they still have some key differences.

**How do you configure Cloud Watch to recover an EC2 instance?** Create an Alarm using Amazon Cloud Watch

In the Alarm, go to Define Alarm -> Actions tab

**How can you recover/login to an EC2 instance for which you have lost the key?**

Verify that the EC2Config service is running

Detach the root volume for the instance

Attach the volume to a temporary instance

Modify the configuration file

Restart the original instance

### **What is the relation between the Availability Zone and Region?**

AWS regions are separate geographical areas, like the US-West 1 (North California) and Asia South (Mumbai).

On the other hand, availability zones are the areas that are present inside the regions. These are generally isolated zones that can replicate themselves whenever required.

**What is auto-scaling?** Auto-scaling is a function that allows you to provision and launch new instances whenever there is a demand. It allows you to automatically increase or decrease resource capacity in relation to the demand.

### **What are the steps involved in a Cloud Formation Solution?**

Create or use an existing Cloud Formation template using JSON or YAML format.

Save the code in an S3 bucket, which serves as a repository for the code?

Use AWS Cloud Formation to call the bucket and create a stack on your template.

Cloud Formation reads the file and understands the services that are called, their order, the relationship between the services, and provisions the services one after the other.

### **Mention what the key components of AWS are?**

Route 53:A DNS web service

Puppet master sends a request to puppet slave - will send a request to slave

Puppet master will send a request to slave

It will deny and send

### **Kubernetes:**

**How does Kubernetes simplify containerized Deployment?** As a typical application would have a cluster of containers running across multiple hosts, all these containers would need to talk to each other. So, to do this you need something big that would load balance, scale & monitor the containers. Since Kubernetes is cloud-agnostic and can run on any public/private providers it must be your choice simplify containerized deployment.

**What do you know about clusters in Kubernetes?** The fundamental behind Kubernetes is that we can enforce the desired state management, by which I mean that we can feed the cluster services of a specific configuration, and it will be up to the cluster services to go out and run that configuration in the infrastructure.

The deployment file will have all the configurations required to be fed into the cluster services. Now, the deployment file will be fed to the API and then it will be up to the cluster services to figure out how to schedule these pods in the environment and make sure that the right numbers of pods are running.

So, the API which sits in front of services, the worker nodes & the Kubelet process that the nodes run, all together make up the Kubernetes Cluster.

**What is Kubectl?** Kubectl is the platform using which you can pass commands to the cluster. So, it basically provides the CLI to run commands against the Kubernetes cluster with various ways to create and manage the Kubernetes component

**What is Kubelet?** This is an agent service which runs on each node and enables the slave to communicate with the master. So, Kubelet works on the description of containers provided to it in the PodSpec and makes sure that the containers described in the PodSpec are healthy and running.

**What are the different components of Kubernetes Architecture?** The Kubernetes Architecture has mainly 2 components – the master node and the worker node. The master and the worker nodes have many inbuilt components within them. The master node has the kube-controller-manager, kube-apiserver, kube-scheduler, etc. Whereas the worker node has kubelet and kube-proxy running on each node.

**What do you understand by Kube-proxy?** Kube-proxy can run on each and every node and can do simple TCP/UDP packet forwarding across backend network service. So basically, it is a network proxy that reflects the services as configured in Kubernetes API on each node. So, the Docker-linkable compatible environment variables provide the cluster IPs and ports which are opened by proxy.

**Can you brief on the working of the master node in Kubernetes?** Kubernetes master controls the nodes and inside the nodes the containers are present. Now, these individual containers are contained inside pods and inside each pod, you can have a various number of containers based upon the configuration and requirements. So, if the pods have to be deployed, then they can either be deployed using user interface or command-line interface. Then, these pods are scheduled on the nodes, and based on the resource requirements, the pods are allocated to these nodes. The kube-apiserver makes sure that there is communication established between the Kubernetes node and the master components.

### **What is the role of kube-apiserver and kube-scheduler?**

The kube – apiserver follows the scale-out architecture and, is the front-end of the master node control panel. This exposes all the APIs of the Kubernetes Master node components and is responsible for establishing communication between Kubernetes Node and the Kubernetes master components. The kube-scheduler is responsible for the distribution and management of workload on the worker nodes. So, it selects the most suitable node to run the unscheduled pod based on resource requirement and keeps a track of resource utilization. It makes sure that the workload is not scheduled on nodes that are already full.

**What is ETCD?** Etcd is a distributed key-value store used for coordinating distributed work. So, Etcd stores the configuration data of the Kubernetes cluster, representing the state of the cluster at any given point in time.

**What do you understand by load balancer in Kubernetes?** A load balancer is one of the most common and standard ways of exposing service. There are two types of load balancer used based on the working environment i.e. either the Internal Load Balancer or the External Load Balancer. The Internal Load Balancer automatically balances load and allocates the pods with the required configuration whereas the External Load Balancer directs the traffic from the external load to the backend pods

**What is Ingress network?** Ingress exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. Traffic routing is controlled by rules defined on the Ingress resource.

Ingress may be configured to give Services externally-reachable URLs, load balance traffic, terminate SSL / TLS, and offer name-based virtual hosting. An Ingress controller is responsible for fulfilling the Ingress, usually with a load balancer, though it may also configure your edge router or additional frontends to help handle the traffic.

Ingress does not expose arbitrary ports or protocols. Exposing services other than HTTP and HTTPS to the internet typically uses a service of type Service. Type=NodePort or Service.Type=LoadBalancer.

**What is the difference between a replica set and replication controller?** Replica Set and Replication Controller do almost the same thing. Both of them ensure that a specified number of pod replicas are running at any given time. The difference comes with the usage of selectors to replicate pods. Replica Set

Simple E-mail Service: It allows sending e-mail using RESTFUL API call or via regular SMTP

**Identity and Access Management:** It provides enhanced security and identity management for your AWS account

Simple Storage Device or (S3): It is a storage device and the most widely used AWS service

Elastic Compute Cloud (EC2): It provides on-demand computing resources for hosting applications. It is handy in case of unpredictable workloads

Elastic Block Store (EBS): It offers persistent storage volumes that attach to EC2 to allow you to persist data past the lifespan of a single Amazon EC2 instance

Cloud Watch: To monitor AWS resources, It allows administrators to view and collect key Also, one can set a notification alarm in case of trouble.

**What is AMI?** AMI stands for Amazon Machine Image. It's a template that provides the information (an operating system, an application server, and applications) required to launch an instance, which is a copy of the AMI running as a virtual server in the cloud. You can launch instances from as many different AMIs as you need.

**Mention what the relationship between an instance and AMI is?** From a single AMI, you can launch multiple types of instances. An instance type defines the hardware of the host computer used for your instance. Each instance type provides different computer and memory capabilities. Once you launch an instance, it looks like a traditional host, and we can interact with it as we would with any computer.

**How can you send a request to Amazon S3?** Amazon S3 is a REST service, and you can send a request by using the REST API or the AWS SDK wrapper libraries that wrap the underlying Amazon S3 REST API.

**The difference between EC2 and Amazon S3 is that?** EC2: It is a cloud web service used for hosting your application

It is like a huge computer machine which can run either Linux or Windows and can handle application like PHP, Python, Apache or any databases.

S3: It is a data storage system where any amount of data can be stored

It has a REST interface and uses secure HMAC-SHA1 authentication keys

**How many buckets can you create in AWS by default?** By default, you can create up to 100 buckets in each of your AWS accounts.

**Explain can you vertically scale an Amazon instance? How?**

Spin up a new larger instance than the one you are currently running

Pause that instance and detach the root volume from the server and discard

Then stop your live instance and detach its root volume

Note the unique device ID and attach that root volume to your new server

And start it again

**In VPC with private and public subnets, database servers should ideally be launched into which subnet?**

With private and public subnets in VPC, database servers should ideally launch into private subnets.

**How many Elastic IPs are allowed by AWS?** 5 VPC Elastic IP addresses are allowed for each AWS account.

**Explain default storage class in S3?** The default storage class is a Standard frequently accessed.

**What are the Roles?** Roles are used to provide permissions to entities which you can trust within your AWS account. Roles are very similar to users. However, with roles, you do not require to create any username and password to work with the resources.

**Explain snowball?** Snowball is a data transport option. It used source appliances to a large amount of data into and out of AWS. With the help of snowball, you can transfer a massive amount of data from one place to another. It helps you to reduce networking costs.

**What are the advantages of auto-scaling?** Offers fault tolerance, Better availability, Better cost management

**How many subnets can you have per VPC?** You can have 200 subnets per VPC.

**What is the role of AWS CloudTrail?**

Cloud Trail: is a specially designed tool for logging and tracking API calls. It helps to audit all S3 bucket accesses.

**What is AWS Lambda?** Lambda is an Amazon compute service which allows you to run code in the AWS Cloud without managing servers.

**Name the types of AMI provided by AWS?** Instance store backed and EBS backed

**What is boot time taken for the instance stored backed AMI?** is less than 5 minutes.

**Can vertically scaling is allowed in Amazon Instance?** Yes, you can vertically estimate one Amazon instance.

**What are the storage classes available in Amazon S3?** Amazon S3 standard, Amazon S3 standard-infrequent Access, Amazon S3 Reduced Redundancy Storage, Amazon Glacier.

**Name some of the DB engines which can be used in AWS RDS?** MS-SQL DB, Maria DB, MYSQL DB, Oracle DB, PostgreSQL

**Latency** is a delay between a user action and web applications response to that action, often referred to in networking terms as a total round trip time it takes for a data packet to travel. It is not a concept related to AWS but rather a concept used in terms of web application. In practical terms, latency is the time between user action and the response from the website or application to this action of instance, the delay between a user's clicks a link to a webpage and when the browser displays that webpage.

Lower the latency the better the region

use Set-Based selectors while replication controllers use Equity-Based selectors.

Equity-Based Selectors: This type of selector allows filtering by label key and values. The equity-based selector will only look for the pods which will have the exact same phrase as that of the label.

Example: Suppose your label key says app=apache, then, with this selector, you can only look for those pods with label app equal to apache.

Selector-Based Selectors: This type of selector allows filtering keys according to a set of values. So, in other words, the selector based selector will look for pods whose label has been mentioned in the set.

Example: Say your label key says app in (nginx, NPS, Apache). Then, with this selector, if your app is equal to any of nginx, NPS, or Apache, then the selector will take it as a true result.

**What is Mini kube?** Mini kube is a tool that makes it easy to run Kubernetes locally. This runs a single-node Kubernetes cluster inside a virtual **machine** not a containerization platform, but it is a multi-container management solution.

Jenkins

**What do you mean by Pipeline as a Code?** Pipeline as Code describes a set of features that allow Jenkins users to define pipelined job processes with code, stored and versioned in a source repository. These features allow Jenkins to discover, manage, and run jobs for multiple source repositories and branches — eliminating the need for manual job creation and management.

To use Pipeline as Code, projects must contain a file named Jenkinsfile in the repository root, which contains a "Pipeline script."

Additionally, one of the enabling jobs needs to be configured in Jenkins:

**Multi branch Pipeline:** build multiple branches of a single repository automatically

**Organization Folders:** scan a GitHub Organization or Bit bucket Team to discover an organization's repositories, automatically creating managed Multi branch Pipeline jobs for them.

**What is PMD?** An extensible cross-language static code analyzer. It is a source code analyzer. It finds common programming flaws like unused variables, empty catch blocks, unnecessary object creation, and so forth. It includes CPD, the copy-paste-detector.

**What is Sonar Qube?** Continuous Code Quality. Sonar Qube provides an overview of the overall health of your source code and even more importantly, it highlights issues found on new code. With a Quality Gate set on your project, you will simply fix the Leak and start mechanically improving.

PMD and Sonar Qube can be categorized as "Code Review" tools.

Some of the features offered by PMD are:

supports multiple languages

enforce a coding standard for your codebase

built-in checks

On the other hand, Sonar Qube provides the following key features: Multi-language, Detect tricky issues, Security analysis

**Explain how you can move or copy Jenkins from one server to another?**

I will approach this task by copying the jobs directory from the old server to the new one. There are multiple ways to do that

You can

Move a job from one installation of Jenkins to another by simply copying the corresponding job directory.

Make a copy of an existing job by making a clone of a job directory by a different name.

Rename an existing job by renaming a directory. Note that if you change a job name you will need to change any other job that tries to call the renamed job.

**What are Parameters in Jenkins?** Parameters are supported by Agent section and they are used to support various use-cases pipelines. Parameters are defined at the top-level of the pipeline or inside an individual stage directive.

**How to create a backup and copy files in Jenkins?**

To create a backup all you need to do is to periodically back up your JENKINS\_HOME directory. This contains all of your build jobs configurations, your slave node configurations, and your build history. To create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

**What you do when you see a broken build for your project in Jenkins?**

I will open the console output for the broken build and try to see if any file changes were missed. If I am unable to find the issue that way, then I will clean and update my local workspace to replicate the problem on my local and try to solve it.

**Explain the terms Agent, Jenkinsfile?**

Agent: It is directive to tell Jenkins to execute the pipeline in a particular manner and order.

Jenkinsfile: The text file where all the definitions of pipelines are defined is called Jenkinsfile. It is being checked in the source control repository.

**What is Jenkins?** Jenkins is an open source tool with plugin built for continuous integration purpose. The principle functionality of Jenkins is to keep a track of version control system and to initiate and monitor a build system if changes occur. It monitors the whole process and provides reports and notifications to alert.

**Explain how you can move or copy Jenkins from one server to another?**

Slide a job from one installation of Jenkins to another by copying the related job directory

## **AWS Global Infrastructure consist of regions, availability zone, Edge locations**

By default they have two data centers. If anyone datacenter goes down there can be a backup. Availability Centers are also called as Data Centers

**Edge location:** They are subsets of AWS data centers. They are partial datacenters of AWS. They are used for cloud front distributions to cast the contents for the users, to give the fastest response to them. AWS partners with local data centers. Highest No of AWS data centers available are 150 +

**What is Route53?** Route 53 is a DNS service, Which helps to achieve High Availability within the region. If region 1 goes down then we are redirected to region2. Availability Zones. In these AZ there are again multiple servers, If these AZ goes down (These happens only when AWS goes down) then we go to another cloud service that is Azure)

**Amazon EBS:** Elastic Block Storage: A service where we can create Volumes (Ex: HDD Volume, SSD volume, Magneto Volume, Performance Storage Volumes) different kinds of volumes which are attached to an OS.

**Amazon Glacier** is an unlimited storage archival storage. Organizations which has unfrequently accessed data or less frequently accessed data or retrospective data. Long term archival service is also called glacier. When storage service is for many years, this is the cheaper service to use

**What happens to my backups and DB Snapshots if I delete my DB Instance?** When you delete a DB instance, you have an option of creating a final DB snapshot, if you do that you can restore your database from that snapshot. RDS retains this user-created DB snapshot along with all other manually created DB snapshots after the instance is deleted, also automated backups are deleted and only manually created DB Snapshots are retained.

### **When should I use a Classic Load Balancer and when should I use an Application load balancer?**

Classic Load Balancer is ideal for simple load balancing of traffic across multiple EC2 instances, while an Application Load Balancer is ideal for micro services or container-based architectures where there is a need to route traffic to multiple services or load balance across multiple ports on the same EC2 instance.

### **What is the biggest issue you have faced in a production environment?**

There was an application we used to run on the cloud, there was an application deployment error, there was a logical. Error in the application that cause an error in the elastic load balancer. So We had an elastic LB which was tied to the Auto scaling group, so it kind of snow balled the application we were not able to control it. The application scaled infinitely and it was snowballing (increase rapidly in size) the instances and we had to go and manually free the size of the auto scaling group, once the size was frozen we were able to go to into the instances check the logs fix the issue rebuild the AMI and restart the auto scaling process so that was the biggest production issue I faced recently.

**What is a Subnet?** A portion of Network that shares a common address component.

Subnets are defined as all devices whose IP addresses have the same prefix

For examples, all devices with IP addresses that start with 100.100.100 would be part of the same subnet

**Benefits of VPC?** Launch instances into a subnet

Define custom IP address range inside each subnet

Configure route tables between subnets

Configure internet gateways and attach them to subnets

Create a layered network of resources

More fine grained security settings to protect your cloud assets

Layered security

Security groups----> Instances

Access control lists--> Subnet

Extend your network onto the cloud using VPG through Ipsec VPN tunnel

**Amazon Storage Gateway** is an application which helps to correct on premises application with storage gateway with the cloud. This can be installed on Hypervisor or ESXI we can achieve the connectivity between between the on prem and the cloud

**What is ECS in AWS?** Amazon Elastic Container Service (ECS) is a highly scalable, high performance container management service that supports Docker containers and allows you to easily run applications on a managed cluster of Amazon EC2 instances.

**Amazon Elastic Container Service (ECS)** is a cloud computing service in Amazon Web Services (AWS) that manages containers and allows developers to run applications in the cloud without having to configure an environment for the code to run in ECS supports Docker, an open source Linux container service.

Make a copy of an already existing job by making clone of a job directory by a different name

Renaming an existing job by renaming a directory.

### **Explain how can create a backup and copy files in Jenkins?**

Jenkins saves all the setting, build artifacts and logs in its home directory, to create a back-up of your

Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

**What is the relation between Hudson and Jenkins?** Hudson was the earlier name and version of current Jenkins. After some issues, they renamed the project from Hudson to Jenkins.

**Mention some of the useful plugins in Jenkins?** Maven 2 project, GIT, Amazon EC2, HTML publisher, Copy artifact, Join, Green Balls

### **What is Maven? What is the benefit of integrating Maven with Jenkins?**

Maven is a build management tool. It uses a simple pom.xml to configure all the dependencies needed to build, test and run the code. Maven manages the full lifecycle of a test project. Once integrated with Jenkins, the maven Web driver will build the project and execute all tests efficiently.

### **What is Kubernetes? How can you integrate Jenkins with Kubernetes?**

Kubernetes is a container orchestration tool. With Kubernetes, one can create multiple container instances to achieve more fault tolerance. You can use the Kubernetes deploy plugin to use it with Jenkins for continuous deploy.

**What is the use of JENKINS HOME directory?** All the settings, logs and configurations are stored in the JENKINS\_HOME directory.

**What is a backup plugin? Why is it used?** This is a helpful plugin that backs up all the critical settings and configurations to be used in the future. This is useful in cases when there is a failure so that we don't lose the settings.

### **What is DevOps and in which stage does Jenkins fit in?**

DevOps is a software development practice that blends software development (Dev) with the IT operations (Ops) making The whole development lifecycle simpler and shorter by constantly delivering builds, fixes, updates, and features. Jenkins plays a Crucial role because it helps in this integration by automating the build, test and deployment process.

**How do you define parameters for a build in Jenkins?** 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 so that you are able 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 anything like a string, a file or a custom.

**What are the ways to configure Jenkins node agent to communicate with Jenkins master?** 2 ways to start the node agent –

Browser – if Jenkins node agent is launched from a browser, a JNLP (Java Web Start) file is downloaded. This file launches a new process on the client machine to run jobs.

Command-line – to start the node agent using the command line, the client needs the executable agent.jar file. When this file is run, it simply launches a process on the client to communicate with the Jenkins master to run build jobs.

**How does Jenkins authenticate users?** There are 3 ways –

The default way is to store user data and credentials in an internal database.

Configure Jenkins to use the authentication mechanism defined by the application server on which it is deployed.

Configure Jenkins to authenticate against LDAP server

**How can you temporarily turn off Jenkins security if the administrative users have locked themselves out of the admin Console?**

The JENKINS\_HOME folder contains a file named config.xml. When you enable the security, this file contains an XML element named useSecurity that changes to true. If you change this setting to false, security will be disabled the next time Jenkins is restarted.<useSecurity>>false</useSecurity>

However, we must understand that disabling security should always be both a last resort and a temporary measure. Once you resolve the authentication issues, make sure that you re-enable Jenkins security and reboot the CI server.

**What is a trigger? Give an example of how the repository is polled when a new commit is detected.**

Triggers are used to define when and how pipelines should be executed.

When Jenkins is integrated with an SCM tool, for example, Git, the repository can be polled every time there is a commit. The Git plugin should be first installed and set up.

After this, you can build a trigger that specifies when a new build should be started. For example, you can create a job that polls the repository and triggers a build when a change is committed