Devops and aws interview preparation

IAM Roles

"An IAM role is an IAM identity that you can create in your account that has specific permissions." It is not uniquely associated with a single person; it can be used by anyone who needs it. A role does not have any security credential l, i.e., username password or security key. It can be used to grant access to the AWS resources, services IAM users, third parties, federated Users etc.

5. Name some of the main Kubernetes features.
Some of the main Kubernetes features that candidates may name include:
1. Simultaneous, multiple cluster management
2. Container management
3. Self-monitoring features for nodes and containers
4. Resource scaling options – both vertically and horizontally

Q. What is Kubernetes?
Kubernetes is an open-source container orchestration system for deploying, scaling, and managing automated applications. It offers an excellent community and works with all cloud providers. Hence, it is a multi-container management solution.

Q. What is a container?
Containers are a technology for collecting the compiled code for an application when it is required at run-time. Each container allows you to run repeatable, standard dependencies and the same behavior whenever the container runs. It divides the application from the underlying host infrastructure to make the deployment much easier in cloud or OS platforms.

Q3. What are the nodes that run inside Kubernetes?
A node is a worker machine or VM depending on the cluster. Each node contains services to run the pods and the pods are managed by the master components.

Q4. What are the services that a node gives and its responsibilities?
The services that include in a node are as follows:
1. Container run-time
2. Kubelet
3. Kube-proxy
The Container run-time is responsible to start and manage the containers.
The kubelet is responsible for running the state of each node and receives commands from the master to work on it and it is also responsible for the metric collection of pods.
The Kube-proxy is a component that manages the subnets and makes services available for all other components.

Q5. What is a master node in Kubernetes?
A master node is a node that controls and manages the set of worker nodes and resembles a cluster in Kubernetes.

Q6. What are the main components of the master node?
The main components of the master node that help to manage worker nodes are as follows:

Kube-server: It acts as a front end of the cluster and communicates with the cluster through the API server.
Kube controller: It implements governance across the cluster and runs the set of controllers for the running cluster.
Kube scheduler: It schedules the activities of the nodes and holds the node resource to determine the proper action for triggering events.

Q7. What is a pod and what does it do?

A pod is a group of containers that are deployed together on the same host. It is the basic execution unit of the Kubernetes application that can create or deploy the Kubernetes unit of object models.

Kubernetes pods can be used in two ways. they are as follows:

Pods that can run in a single container

Pods that can run with multiple containers when it is required to work together

Q8. What are the different types of multiple-container pods?

There are three different types of multi-container pods. They are as follows:

Sidecar: The Sidecar pattern is a single node pattern made of two containers of the application. It contains the core logic of the application and it sends the logic files to the bucket.

Adapter: It is used to standardize and normalize the output application or monitor data for aggregation. It performs restructuring, and reformatting and can write the correct formatted output for the application.

Ambassador: It is a proxy pattern that allows connecting other containers with a port on the localhost.

Q9. What is the Namespace? How many namespaces are there in Kubernetes?

A namespace is used to work with multiple teams or projects spread across. It is used to divide the cluster resources for multiple users.

Q. What is kubectl?

Kubectl is the command-line tool used to control the Kubernetes clusters. It provides the CLI to run the command against clusters to create and manage the Kubernetes components.

Q24. What are the different types of services in Kubernetes?

The different types of services that support Kubernetes are as follows:

1. Cluster IP: It exposes the services on cluster internal IP and makes the services reachable within the cluster only.
2. Node port: It exposes the services on each node’s IP at the static port.
3. Load balancer: It provides services externally using a cloud provider’s load balancer. It creates the service to route the external load balancer automatically.
4. External name: It navigates the service to the contents of the external field by returning the CNAME record by its value.

Q25. Mention the various container resource monitoring tools.

The various container monitoring tools are as follows:

1.Grafana
2. Heapster
3. CAdvisor
4. InfluxDB
5. Prometheus

Q38. What is the role of clusters in Kubernetes?

Kubernetes allows the required state management by cluster services of a specified configuration. These cluster services run the configurations in the infrastructure. The following are steps that are involved in this process as follows:

1. The deployment file contains all the configuration that is fed into the cluster
2. These deployments are fed into the API server
3. The cluster services will schedule the pods in the environment
4. It also ensures the right number of pods were running

Q. What are the advantages of Kubernetes?
The advantages of Kubernetes are as follows:

1. Kubernetes is open-source and free
2. It is highly scalable and runs in any operating system
3. It provides more concepts and is more powerful than Docker swarm
4. It provides a scheduler, auto-scaling, rolling upgrades, and health checks
5. It has a flat network space and customized functionalities
6. It is easy to make effective CI/CD pipelines
7. It can improve productivity

Q. What are the disadvantages of Kubernetes?
The disadvantages of Kubernetes are as follows:

1. The installation process and configuration is highly difficult
2. It is not easy to manage the services
3. It takes a lot of time to run and compile
4. It is more expensive than the other alternatives
5. It can be overkill for simple application

Q11. How are Kubernetes related to docker?
Docker provides the lifecycle management of a container and the docker image builds the run-time of a container. The containers run on multiple hosts through a link and are orchestrated using Kubernetes. Dockers build these containers and help to communicate with multiple hosts through Kubernetes

Q13. Why do we need Container orchestration in Kubernetes?
Container orchestration is used to communicate with several micro-services that are placed inside a single container of an application to perform various tasks.

The use of container orchestration is as follows:

1. It controls and automates various tasks such as deployment, scaling, etc.,
2. Reduces the complexity of running time
3. Scaling becomes easy
4. It is used to deploy and manage complex containerized applications
5. Reduces manual setting up services

Q14. What are the tools of container orchestration?
There are many Container orchestration tools that provide a framework for managing microservices and containers at scale. The popular tools for container orchestration are as follows:

1. Kubernetes
2. Docker swarm
3. Apache Mesos

Q15. What are the major operations of Kubelet as a node service component in Kubernetes?
The major operations that the Kubelet do as follows:

1. The Kubelet is a node that communicates with master components to work on all the parts of the Kubernetes cluster.
2. It merges the available CPU, memory, and disk of a node into a large Kubernetes cluster.
3. It provides access to the controller to check and report the status of the cluster.
4. It is responsible for the collection of metric pods

Q16. Mention the list of objects of Kubernetes.

The following is the list of objects used to define the workloads.

1. Pods
2. Replication sets and controllers
3. Deployments
4. Distinctive identities
5. Stateful sets
6. Daemon sets
7. Jobs and cron jobs

Q17. What is the difference between the pod and the container?
Pods are the collection of containers used as the unit of replication in Kubernetes. Containers are the set of codes to compile in a pod of the application. Containers can communicate with other containers in the same pod

Q18. Explain Stateful sets in Kubernetes?
Ans: Stateful set is a workload API object used to manage the stateful application. It is used to manage deployments and scale the sets of pods. The state information and other resilient data of stateful pods were stored and maintained in the disk storage that connects with the stateful set.

Q19. How to determine the status of deployment?
To determine the status of the deployment, use the command below:

```
kubectl rollout status
```

If the output runs, then the deployment is successfully completed.

Q20. Explain Replication controllers?
Replication controllers act as supervisors for all long-running pods. It ensures that the specified number of pods are running at the run-time and also ensures that a pod or a set of pods are homogeneous in nature. It maintains the desired number of pods if the number of pods it will terminate the extra pod. And if there is a failed pod, the controller will automatically replace the failed pod.

Q21. What are the features of Kubernetes?
The features of Kubernetes are as follows:

1. It provides an automated and advanced scheduler to launch the containers on the cluster
2. Replacing, rescheduling, and restarting the containers that failed while compilation
3. It supports rollouts and rollback for the desired state of the containerized application
4. It can scale up and scale down as per the requirements.

[11:48 am, 04/04/2023] Institute: Krishna:
Devops and aws interview preparation Hyderabad:
Q22. What is kubectl?
Kubectl is the command-line tool used to control the Kubernetes clusters. It provides the CLI to run the command against clusters to create and manage the Kubernetes components.
[11:48 am, 04/04/2023] Institute: Q23. What is the Google container engine?
The Google Container Engine (GKE) is the open-source management for the Docker containers and the clusters. This Kubernetes-based container engine supports only the clusters that run within the Google public cloud service.

Q24. What are the different types of services in Kubernetes?
The different types of services that support Kubernetes are as follows:

1. Cluster IP: It exposes the services on cluster internal IP and makes the services reachable within the cluster only.

2. Node port: It exposes the services on each node’s IP at the static port.
3. Load balancer: It provides services externally using a cloud provider’s load balancer. It creates the service to route the external load balancer automatically.
4. External name: It navigates the service to the contents of the external field by returning the CNAME record by its value.

Q25. Mention the various container resource monitoring tools.

The various container monitoring tools are as follows:

1. Grafana
2. Heapster
3. CAdvisor
4. InfluxDB
5. Prometheus

Q26. What is Heapster?

Heapster is a performance monitoring and metric collection system. It provides cluster-wide data aggregation by running with a kubelet on each node. It allows for the collection of metrics, pods, workloads, containers, and other signals that are generated by the clusters.

Q27. Explain Daemon sets?

A daemon set ensures that all the eligible nodes run a copy of the pod runs only once in a host. It was created and scheduled by the daemon controller. It is a process that runs in the background and does not produce any visible output.

Q28. What are the uses of Daemon sets?

The uses of Daemon sets are as follows:

1. It runs cluster storage such as ceph, glusterd on each node.
2. It runs the logs collection of daemons on every node such as fluentd or filebeat.
3. It runs node monitoring on every node.

Q29. Explain the Replica set?

A Replica set is used to maintain a stable set of replica pods. It is used to specify the available number of identical pods. It was also considered as a replacement for the replication controller sometimes.

Q30. What is ETCD in Kubernetes?

ETCD is the distributed key-value store. It stores and replicates the configuring data of the Kubernetes cluster.

Q31. Explain the Ingress controller?

An ingress controller is a pod that acts as an inbound traffic handler. It is responsible for reading the ingress resource information and processing the data accordingly.

Q32. What is the based selector that is used in the replication controller?

The Replication controller uses the Equity-Based selector that allows filtering by labels key and values. It only looks for the pods which have the same values as that of the label.

Q33. Explain the Load balancer in Kubernetes?

The load balancer is a way of distributing the loads, which is easy to implement at the dispatch level. Each load balancer sits between the client devices and the backend servers. It receives and distributes the incoming requests to all available servers.

Q34. Explain the two different types of load balancers.

The two different load balancers are one is an internal load balancer that balances the load and allocates the pods automatically with the required configuration. And the other is the External load balancer that directs the traffic from external loads to the backend pods.

Q35. What is Minikube?
Minikube is a type of tool that helps to run Kubernetes locally. It runs on a single-node Kubernetes cluster inside a Virtual machine (VM).

Q36. What are the uses of the Google Kubernetes Engine?
The uses of Google Kubernetes Engine are as follows:

1. It creates the Docker container cluster
2. It resizes the application controllers
3. It creates the containers pods, load balancer, services, replication controller
4. It updates and upgrades the container cluster
5. It helps to debug the container cluster

Q37. Explain Prometheus in Kubernetes.
Prometheus is an open-source toolkit that is used for metric-based monitoring and alerting the application. It provides a data model and a query language and can provide details and actions of metrics. It supports the instrumental application of language for many languages. The Prometheus operator provides easy monitoring for deployments and k8s services, besides Alertmanager and Grafana.

Q38. What is the role of clusters in Kubernetes?
Kubernetes allows the required state management by cluster services of a specified configuration. These cluster services run the configurations in the infrastructure. The following are steps that are involved in this process as follows:

1. The deployment file contains all the configuration that is fed into the cluster
2. These deployments are fed into the API server
3. The cluster services will schedule the pods in the environment
4. It also ensures the right number of pods were running

Q39. What is the Cluster IP?
The cluster Ip is a default Kubernetes service that provides a link between the pods or map container port and the host ports. It provides the services within the cluster and gives access to other apps which are inside the same cluster.

Q40. What are the types of controller managers?
The Different types of controller managers that can run on the master node are as follows:

1. Endpoints controller
2. Namespace controller
3. Service account controller
4. Replication controller
5. Node controller
6. Token controller

Q41. What is Kubernetes architecture?
The Kubernetes architecture provides a flexible, coupled mechanism for the service. It consists of one master node and multiple containers. The master node is responsible for managing the clusters, API, and scheduling the pods. Each node runs on the container runtime such as Docker, rkt along with the node that communicates with the master.

Q42. What are the main components of Kubernetes architecture?
The two main components of the Kubernetes architecture are as follows:

1. Master node
2. Worker node

Q43. Define Kube-api server?
The Kube-API is the frontend of the master node that exposes all the components in the API server. It provides communication between the Kubernetes nodes and the master components.

Q44. What are the advantages of Kubernetes?
The advantages of Kubernetes are as follows:

1. Kubernetes is open-source and free
2. It is highly scalable and runs in any operating system
3. It provides more concepts and is more powerful than Docker swarm
4. It provides a scheduler, auto-scaling, rolling upgrades, and health checks
5. It has a flat network space and customized functionalities
6. It is easy to make effective CI/CD pipelines
7. It can improve productivity

Q45. What are the disadvantages of Kubernetes?
The disadvantages of Kubernetes are as follows:

1. The installation process and configuration is highly difficult
2. It is not easy to manage the services
3. It takes a lot of time to run and compile
4. It is more expensive than the other alternatives
5. It can be overkill for simple application

Devops and aws interview preparation Hyderabad:
1. What is meant by Kibana?
Ans: Kibana is an open-source and free frontend tool that sits at the top of the Elastic Stack, allowing users to search and visualize the data which is indexed in Elasticsearch. Kibana is the user-friendly interface for managing, protecting and monitoring an Elastic Stack cluster, as well as the administrative hub for built-in solutions based on the Elastic Stack (formerly known as the ELK Stack after Logstash, Kibana, and Elasticsearch). Kibana, which was developed in 2013 by members of the Elasticsearch forum, has evolved into a window into the Elastic Stack itself, providing a platform for individuals and businesses.

2. What is Kibana used for?
Ans: Kibana’s strong integration with Elasticsearch and the significant Elastic Stack is a perfect ideal for supporting the accompanying points:

A. Data indexed in Elasticsearch is searched, viewed, and visualized, and the data is analyzed using tables, maps, pie charts, bar charts, and histograms. A dashboard overview integrates various visual features to provide real-time analytical perspectives into enormous data volumes in support of use cases like:
    1. Business analytics
    2. Infrastructure metrics and container monitoring
    3. Geospatial data analysis and visualization
    4. Logging and log analytics
    5. Application performance monitoring (APM)
    6. Security analytics
B. Using the web interface, you can secure, monitor and manage an Elastic Stack instance.
C. Providing centralized access to built-in Elastic Stack solutions for enterprise search applications, security and visibility.

3. Why use Kibana?

Ans: Elasticsearch’s official interface is Kibana. Kibana is the most efficient platform for obtaining data insights and actively managing the integrity of their Elastic Stack for Elasticsearch users.
Kibana covers a wide range of scenarios. Elastic has put a lot of effort into improving the visualization interface. Kibana’s built-in abilities are used for uptime monitoring, business analytics, security analytics, APM, geospatial analytics, and other applications.
Kibana has a large and active user base. Kibana has a large community of users and contributors since it is an open and free interface. Kibana users have a wide range of experience, which is reflected in the documentation, training, and community support they receive. Elastic also provides training and one-on-one support to help users get started.

4. Explain about Elasticsearch?
Ans: Elasticsearch is a Java-based, open-source search, analytical engine distributed based on Apache Lucene. It began as a modular implementation of the open-source Lucene search platform, then incorporated the ability to expand the Lucene indices dynamically. Elasticsearch allows you to easily analyze, store, and search large amounts of data in near real-time, with results arriving in milliseconds. It can produce quick search results because it searches an index rather than searching the text directly. It has a document-based structure rather than tables and schemas, and it has comprehensive REST APIs for searching and storing the data. At its most basic level, Elasticsearch can be thought of as a server that can accept JSON requests and return JSON data.

6. Explain about the Kibana dashboard?
Ans: A Kibana dashboard is a single pane that contains a variety of maps, graphs, searches, and charts. Dashboards provide a glance of data insights from different angles and allow users to delve deeper into the details.

Q1) What is Jenkins and why use it?
Ans: Jenkins is one of the leading open-source continuous integration tools. The main functionality of this tool is to keep track of the version control system and monitor the build system and provide notifications and reports to alert. It enables you to deliver software by integrating with a large number of testing and deployment technologies.

The following are the reasons to use Jenkins:

1. It possesses an installer package for major operating systems.

2. Integrates individual projects for a larger purpose

3. To keep your team in sync

4. Troubleshoot and audit past jobs effortlessly

5. Provides accurate data support for project management

Q2) What is continuous integration?
Ans: Continuous integration is a process of continuously checking the developer’s code into a version control system several times a day and automating the build to check and detect bugs in the written code. Continuous Integration includes the following:

1. Development and Compilation

2. Database Integration

3. Unit Testing

4. Production Deployment

5. Code Labeling

6. Functional Testing

7. Generating and Analyzing Reports

Q3) What are the features of Jenkins?
Ans: Jenkins comes with the following features:

1. Free open source.

2. Easy installation on various operating systems.

3. Build Pipeline Support.

4. Workflow Plugin.

5. Test harness built around JUnit.

6. Easy upgrades.

7. Rapid release cycle.

8. Easy configuration setup.

9. Excellent community and documentation

10. Extensible with the use of third-party plugins.

Q4) What are the advantages of using Jenkins?
Ans: The advantages of using Jenkins are the following:

Open-source tool and user-friendly

Easy to install

Provides great collaboration between development and operations teams.

Code deployment is easy and happens in minutes, along with the generation of reports.

Free of cost

Platform independent

Rich plugin ecosystem

Code errors can be detected as early as possible.

Automation of integration work, thereby reducing the number of integration issues.

Q5) What are the prerequisites to use Jenkins?
Ans: We require the following to use Jenkins:

A source code repository that is accessible, for instance, and a Git repository

A working build script. e.g., a Maven Script checked into the repository

Q6) Name some of the plugins in Jenkin?
Ans: Some of the important plugins in Jenkin includes:

Maven 2 project
Amazon EC2
HTML publisher
Copy Artifact
Join
Green Balls
Git plugin
Multi job plugin
Test Results Analyzer
Metrics

Q7) How to restart Jenkins manually?
Ans: To restart Jenkins manually, you can use any one of the following commands:

(jenkins_url)/safe restart - Allows all running jobs to complete. New jobs will remain in the queue to run after the restart is complete.

(jenkins_url)/restart - Forces a restart without waiting for builds to complete.

Q8) What are the components that you can integrate Jenkins with?
Ans: Jenkins is mainly integrated with the following:

Version Control system like GIT, SVN
And build tools like Apache Maven.

Q9) How does Hudson relate to Jenkins?
Ans: Jenkins was a renamed version of Hudson.

Q10) How to install Jenkins?
Ans:

Install Java Version 8

Install Apache Tomcat Version 9

Download Jenkins war File

Deploy Jenkins war File

Install Suggested Plugins

Q11) How can you start Jenkins manually?
Ans: Jenkins can be manually started by opening the Console/Command line and using the below commands:

Start Jenkins: jenkins.exe start

Stop Jenkins: jenkins.exe stop

Restart Jenkins: jenkins.exe restart

Q13) Name a few Jenkins environment variables.
Ans: By default, there are numerous variables available in Jenkins. Some of them are:

$NODE_NAME

$JOB_NAME

$WORKSPACE

$JOB_URL

$BUILD_URL

Q14) How can you deploy a custom build of a core plugin?
Ans: To deploy a custom build of a core plugin, you have to do the following:

Stop Jenkins.
Copy the custom HPI to $Jenkins_Home/plugins.
Delete the previously expanded plugin directory.
Make an empty file called <plugin>. hpi. pinned.
Start Jenkins.

Devops and aws interview preparation Hyderabad:
1. Version Control Tool: Git (GitLab, GitHub, Bitbucket)
Git is perhaps the best and most widely used version control tool in a development era characterized by dynamism and collaboration. Version control provides developers with a means by which they can keep track of all the changes and updates in their codes such that in the event of a mishap, it is quite easy to return to and use the previous versions of the code and Git happens to be the best for many reasons.

Git DevOps tool is easy to implement as it is compatible with most protocols including HTTP, SSH, and FTP. It offers the best advantage for non-linear shared-repository development projects, unlike most other centralized version control tools. This makes it a good deal for mission-critical software.

Git features three storage tools including, GitHub and GitLab cloud-hosted code repository services as well as BitBucket the source code hosting service. Of the three, GitLab and BitBucket are specifically designed for enterprise-range version control.

2. Build Tool: Maven
Maven is one of the important DevOps tools for building projects. Unlike the ANT build system, Apache Maven is more than just an automation build framework. It is also designed to manage reporting, documentation, distribution, releases, and dependencies processes. Written in Java language, Maven can build and manage projects written in Java or C#, Ruby, Scala, and other languages using project object model (POM) plugins.

Maven offers a host of benefits to its users. It eases the build and monitoring process through automation and maintains a uniform build process allowing for consistency and efficiency. This tool also offers comprehensive project information through quality documentation, a valuable resource for the development of best practices hence the name Maven, translated from the Yiddish language to mean accumulator of knowledge. Finally, Maven provides a very simplified feature migration process.

It has a rich repository of plugins to enhance the build process and wide compatibility with IDEs like Eclipse, JBuilder, MyEclipse, NetBeans, IntelliJ IDEA, and others.

3. Continuous Integration Tool: Jenkins
Jenkins is an integration DevOps tool. For continuous integration (CI), Jenkins stands out as it is designed for both internal and plugin extensions. Jenkins is an open-source Java-based automation CI server that is supported by multiple operating systems including Windows, macOS, and other Unix OSs. Jenkins can also be deployed on cloud-based platforms.

Continuous Integration and Continuous Delivery are two core practices of the DevOps methodology which makes Jenkins an indispensable DevOps tool. Jenkins is compatible with most CI/CD integration tools and services thanks to the over 1,500 plugins available to provide integration points for delivering customized functionality during software development.

A valuable automation CI tool, Jenkins is pretty easy to install and configure. It is designed to support distributed workflows for accelerated and transparent builds, tests, and deployments across platforms.

4. Configuration Management Tool: Ansible
Ansible is an open-source CM DevOps tool that is also used for deployment, automation, and orchestration. While Ansible leverages infrastructure as a code architecture, it uses SSH connection for its push nodes thus agentless. Of the three, Ansible is considered easy to learn and use as its Playbooks are written in YAML with minimal commands and are readable by humans.

5. Container Platforms: Docker
Container platforms are application solutions that allow developers to build, test, and ship applications in resource-independent environments. Each container comprises a complete runtime environment including the specific application, its libraries, source code, configurations, and all its dependencies. Container platforms offer orchestration, automation, security, governance, and other capabilities.

DevOps heavily relies on containerization and microservices for efficient application development and deployment with Docker and Kubernetes as the most widely used container technologies.

6. Docker
The Docker engine is designed to automate the development, deployment, and management of containerized applications on single nodes. Docker is open-source and compatible with cloud services like AWS, GCP, and Azure Cloud. Docker also runs on Windows and Linux operating systems.

8. Container Platforms: Kubernetes
Kubernetes, on the other hand, is an automation orchestration platform that enables developers to run containerized applications across Kubernetes clusters referring to a group of nodes. Developers harness Kubernetes to automate such processes as container configuration, scaling, networking, security, and more to achieve speed and efficiency in production.

9. AWS Cloud Computing and Storage in DevOps
AWS features the widest range of service offerings under PaaS, SaaS, and IaaS categories including compute, identity and access management (ACM), networking, and storage. While AWS offers public, private, and hybrid clouds, its focus is more on the public cloud.

Devops and aws interview preparation Hyderabad:
Q 1: What do you mean by Terraform?
Answer: Terraform is open-source communication as a system software tool created by HashiCorp. It is an instrument for building, altering, and versioning transportation safely and professionally. Terraform can direct existing and accepted service providers as well as convention in-house solutions.

Q 2: What are the reasons for choosing Terraform for DevOps?
Answer: Below are the reasons for choosing Terraform for DevOps:

1. It can do complete orchestration and not just configuration management (like Ansible and Puppet).

2. Has amazing support of almost all the popular cloud providers like AWS, Azure, GCP, DigitalOcean etc .
3. Easily manages the configuration of an immutable (dynamic) infrastructure.
4. Provide immutable infrastructure where configuration changes smoothly.
5. Works on HCL (HashiCorp configuration language), which is very easy to learn and understand.
6. Easily portable from one provider to another.
7. Easy Installation.

Question 3: Define Terraform init?
Answer: Terraform initialises the code with the command terraform init. This command is used to set up the working directory for Terraform configuration files. It is safe to run this command multiple times.

You can use the init command for:

Installing Plugins
Installation of a Child Module
Initialization of the backend

Question 4: Name some major competitors of Terraform?
Answer: Some of them are:

Packer
Cloud Foundry
Ansible
Kubernetes

Question 5: Define Terraform provider?
Answer: Terraform is a tool for managing and informing infrastructure resources such as physical machines, virtual machines (VMs), network switches, containers, and more. A provider is responsible for API interactions that are thoughtful and reveal resources. Terraform is compatible with a wide range of cloud providers.

Question 6: How does Terraform work?
Answer: Terraform creates an implementation plan, defines what it will do to achieve the desired state, and then executes it to build the infrastructure described. Terraform is capable of determining what changed and generating incremental execution plans that are practical as the configuration changes.

Devops and aws interview preparation Hyderabad:
Q. What is AWS in DevOps?
Ans: AWS is Amazon’s cloud service platform that lets users carry out DevOps practices easily. The tools provided will help immensely to automate manual tasks, thereby assisting teams to manage complex environments and engineers to work efficiently with the high velocity that DevOps provides.

Q. DevOps and Cloud computing: What is the need?
Ans: Development and Operations are considered to be one single entity in the DevOps practice. This means that any form of Agile development, alongside Cloud Computing, will give it a straight-up advantage in scaling practices and creating strategies to bring about a change in business adaptability. If the cloud is considered to be a car, then DevOps would be its wheels.

3. Why AWS for DevOps?

Ans: There are numerous benefits of using AWS for DevOps. Some of them are as follows:

1. AWS is a ready-to-use service, which does not require any headroom for software and setups to get started with.

2. Be it one instance or scaling up to hundreds at a time, with AWS, the provision of computational resources are endless.
3. The pay-as-you-go policy with AWS will keep your pricing and budgets in check to ensure that you can mobilize enough and get an equal return on investment.
4. AWS brings DevOps practices closer to automation to help you build faster and achieve effective results in terms of development, deployment, and testing processes.
5. AWS services can easily be used via the command-line interface or by using SDKs and APIs, which make it highly programmable and effective.

Devops and aws interview preparation Hyderabad:
What is Ansible?
A product of RedHat, Ansible is a management tool for service deployment. It is an open-source solution for software provisioning, application deployment, and configuration management; Ansible has become increasingly popular because it offers its users numerous facilities. You can automate multiple IT processes by using Ansible. Moreover, its design is for multi-tier deployment so it can handle your different systems together.

Advantages of Ansible
1. Allows you to model complicated IT workflows easily
2. It is open-source
3. No requirements for downloading additional software on the client platform
4. No requirements for setting up separate management structures
5. Easy to use language with simple English-focused syntax

What is Jenkins?
Jenkins is an open-source tool in Java for automation and Continuous Integration tasks. Jenkins allows users to create and test their projects continuously while integrating changes quickly. One of the biggest highlights of Jenkins is its large number of plugins. Plugins allow Jenkins to integrate with other software solutions and enhance its capabilities in multitudes.

Jenkins can integrate the entire development life-cycle process of an application. This means it can handle creation, testing, packaging, deployment, analysis, and other operations.

Advantages of Jenkins
1. It is open-source, so you can use it for free
2. It has an active and thriving community
3. It has various plugins that enable it to work well with other CI and CD tools
4. Jenkins supports distributed builds
5. Easy installation, configuration, and upgrade
6. Easy to monitor external jobs

What is Terraform?
Terraform is one of the most popular IAC tools used by every cloud engineer. It allows us to define both cloud and on-premise resources in human-readable configuration files and thereby provision these resources programmatically. The most notable feature of Terraform is that, unlike most IAC tools out there, it is not limited to a single cloud provider. You can use Terraform to run your applications on multiple cloud platforms simultaneously.

#1. What do you understand by Terraform?
Terraform is an open-source IAC tool created by HashiCorp. It is used to create, update, delete and version your infrastructure on multiple cloud platforms.

#2. What are the reasons to choose Terraform for DevOps?
Using Terraform for provisioning infrastructure leaves no room for human errors, hence improving the quality, consistency, and efficiency of Cloud and on-prem infrastructure. Terraform uses the HCL language, w

hich is fairly similar to JSON and easy to learn and use. Unlike the other IAC tools offered by cloud providers like Cloudformation for AWS, we can use Terraform with a number of cloud platforms simultaneously. This avoids the need to learn multiple IAC tools and improves the scope of collaboration.

#3. How does Terraform work?
Terraform uses plugins called the Terraform providers to interact with APIs on Cloud Platforms and provision our resources. As an end-user, terraform workflow has three steps.

Write: Author the infrastructure as code.

Plan: Preview changes Terraform will make before applying.

Apply: Provision the infrastructure and apply the changes.

#4. What do you mean by Terraform cloud?
Terraform Cloud is a remote environment that is optimized for the Terraform workflow. It provides features like workspaces and state locking, which allows people in big teams to collaborate.

#5. What do you understand by State in Terraform?
As an IAC tool, terraform should know the current state of configurations and infrastructure under its management. Terraform stores this information in a file called the state file.

#6. What is the benefit of Terraform State?
The Terraform State allows Terraform to map real-world resources to your configuration, keep track of metadata, and improve performance when planning changes for complex infrastructures. It is a critical component of Terraform.

#7. What do you understand by Terraform Backend?
Terraform backend is the platform where the Terraform State Snapshots are stored. By default, Terraform uses a backend called local to store state as a local file on your disk. All other supported backends are some kind of remote storage service.

#11. What do you understand by modules in Terraform?
A Terraform module is a standard container for multiple resources used together to provision and configure resources. For example, you can create a “VPC module” for your organization that provisions a standard VPC and other resources like Subnets and Internet Gateways. Modules can be shared publically via the Public module registry and privately via the Private Module registry.

Devops and aws interview preparation Hyderabad:
Q #1) What do you mean by Maven?

Answer: Maven is a project management tool (introduced by Apache Software Foundation) that provides an entire framework for the build cycle. It is open-source and is mainly used for the project developed in Java.

Maven is driven by a project object model popularly known as POM. It is the central repository for all the dependencies. It maintains the same folder convention across organizations and can be easily integrated with continuous integration tools like Jenkins.

Besides, it takes care of the reporting and documentation of the project. It is developed in Java language.

Q #2) What are the features or advantages of Maven?

Answer:

The features or advantages of Maven are as follows:

Not required to manually add jars for the project. The updates to the project dependencies and transitive dependencies are carried out automatically by Maven.
Maven maintains a uniform directory structure across the organization.
Both the deployment and build activities are taken care by Maven.
Maven is simple, easy to grasp, to set up and utilize in the projects as it is driven by the POM file.
Rapidly expanding repository of Maven contains a large number of libraries that can be used for multiple projects at a time.
Maven encourages the use of extensible code design.

Q #3) Name the aspects that are managed by Maven.

Answer: The aspects that are managed by Maven are documentation, builds, SCMs, releases, distribution, mailing lists, dependencies, and reporting.

Q #4) What is meant by Maven build lifecycle? Name these cycles.

Answer: A Maven build life cycle comprises of certain phases. It determines the order in which the Maven goals need to be performed. If a particular phase is to be accomplished, then all the previous phases prior to that particular phase need to be run successfully in sequence.

Maven build life cycles are listed below as the followings:

1. Clean: Removes the artifact produced from the preceding build processes.
2. Default: Deployment and build process is taken care of.
3. Site: Project documentation is carried out in this cycle.

Q #5) What is a build tool responsible for?

Answer: Build tools are primarily responsible for producing the source code (provided the auto-generated code is utilized). It creates project documentation from the source code.

It compiles and packages the code in the form of JAR/ZIP file. Finally, places that code in the local, central, or remote repository.

Q #6) What do you mean by POM and what does it contain?

Answer: Project Object Model or pom forms the elementary part of the working of Maven. While performing a Maven goal or a task, the pom file residing in the present directory is searched and referred to collect information on the project.

It is basically an xml file. It contains configuration details like plugins, goals, developers, dependencies, profiles, versions, and mailing lists.

Q #7) What do you mean by a Maven Repository?

Answer: Maven repository is the location of a directory where all the related project artifacts, jars, libraries, plugins are kept and can be utilized by Maven easily.

Q #8) Explain the various types of Maven Repositories.

Answer: There are three types of Maven repositories.

They are:

1. Local Repository: This is placed in our local machine generally in the .m2 directory. This is generated once we are able to execute a Maven command successfully. All the project dependencies reside here. Once Maven scans the pom file, it first searches for its dependencies in the local repository.
2. Central Repository: This is supplied by Apache Maven. It contains most of the routinely used libraries. Once any of the dependencies are not found in the local repository, then Maven connects to a central repository.
3. Remote Repository: Sometimes companies develop their own custom repository comprising of their project artifacts, jars, and libraries. This is a type of repository that remains private for use inside that organization.

Q #9) Explain Maven Artifact.
[11:48 am, 04/04/2023] Institute: Krishna:
Answer: Maven artifact is known as a file (generally a jar) that is extended to the Maven repository. Maven build generates multiple artifacts comprising of source jar and compiled jar. GroupId, artifactId, and version together form an artifact and all of the three combined solely identifies it.

Q #10) List down the various scopes of Maven Dependency.

Answer:

The various scopes of Maven dependency include:

1. Compile: This scope is required to build, test, and run the project and is available by default.
2. Provided: This scope is needed to build and test the project and is available at runtime.
3. Runtime: This scope is not required for compilation but needed for execution.
4. Test: This scope is needed for compilation and running of the unit test cases.
5. System: This scope cannot be taken from the remote repository and need to be placed in the local project path.
6. Import: This scope is utilized when the dependencies are of pom type.

Q #11) How to determine the version of Maven in our system?

Answer: To determine the version of Maven we are using in our system, we need to enter the below command in the console.

```
mvn –version
```

Q #12) How to mention profiles in Maven?

Answer: The profiles are mentioned in Maven with the help of a subset of elements present in the pom file.

Q #13) What are Maven Plugins?

Answer: Maven plugins are the basic component of a Maven framework. Each of the plugins has a specific task to be performed.

Maven generally performs the following functionalities:

1. Generates jar files.
2. Generates war files.
3. Compiles the code.
4. Executes unit testing of code.
5. Generates documentation of the project.

6. Generates customized reports.
Maven plugin gives a group of goals that can be run with the following command syntax:

```
mvn [plugin-name]:[goal-name]
```

Q #17) What are the different types of Maven Plugins?

Answer:

The different types of Maven plugins are listed below:

1. Building Plugins: These plugins are used at the time of build and are defined in the building element of the pom file.
2. Reporting Plugins: These plugins are used at the time of site generation and are defined in the reporting element of the pom file.

Devops and aws interview preparation Hyderabad:
Q1) What is Jenkins and why use it?
Ans: Jenkins is one of the leading open-source continuous integration tools. The main functionality of this tool is to keep track of the version control system and monitor the build system and provide notifications and reports to alert. It enables you to deliver software by integrating with a large number of testing and deployment technologies.

The following are the reasons to use Jenkins:

1. It possesses an installer package for major operating systems.

2. Integrates individual projects for a larger purpose

3. To keep your team in sync

4. Troubleshoot and audit past jobs effortlessly

5. Provides accurate data support for project management

Q2) What is continuous integration?
Ans: Continuous integration is a process of continuously checking the developer’s code into a version control system several times a day and automating the build to check and detect bugs in the written code. Continuous Integration includes the following:

1. Development and Compilation

2. Database Integration

3. Unit Testing

4. Production Deployment

5. Code Labeling

6. Functional Testing

7. Generating and Analyzing Reports

Q3) What are the features of Jenkins?

Ans: Jenkins comes with the following features:

1. Free open source.

2. Easy installation on various operating systems.

3. Build Pipeline Support.

4. Workflow Plugin.

5. Test harness built around JUnit.

6. Easy upgrades.

7. Rapid release cycle.

8. Easy configuration setup.

9. Excellent community and documentation

10. Extensible with the use of third-party plugins.

Q4) What are the advantages of using Jenkins?
Ans: The advantages of using Jenkins are the following:

1. Open-source tool and user-friendly

2. Easy to install

3. Provides great collaboration between development and operations teams.

4. Code deployment is easy and happens in minutes, along with the generation of reports.

5. Free of cost

6. Platform independent

7. Rich plugin ecosystem

8. Code errors can be detected as early as possible.

9. Automation of integration work, thereby reducing the number of integration issues.

Q5) What are the prerequisites to use Jenkins?
Ans: We require the following to use Jenkins:

1. A source code repository that is accessible, for instance, and a Git repository

2. A working build script. e.g., a Maven Script checked into the repository

Q6) Name some of the plugins in Jenkin?
Ans: Some of the important plugins in Jenkin includes:

1. Maven 2 project
2. Amazon EC2

3. HTML publisher
4. Copy Artifact
5. Join
6. Green Balls
7. Git plugin
8. Multi job plugin
9. Test Results Analyzer
10. Metrics

Q7) How to restart Jenkins manually?
Ans: To restart Jenkins manually, you can use any one of the following commands:

1. (jenkins_url)/safe restart - Allows all running jobs to complete. New jobs will remain in the queue to run after the restart is complete.

2. (jenkins_url)/restart - Forces a restart without waiting for builds to complete.

Q8) What are the components that you can integrate Jenkins with?
Ans: Jenkins is mainly integrated with the following:

1. Version Control system like GIT, SVN
2. And build tools like Apache Maven.

Q9) How does Hudson relate to Jenkins?
Ans: Jenkins was a renamed version of Hudson.

Q10) How to install Jenkins?
Ans:

1. Install Java Version 8

2. Install Apache Tomcat Version 9

3. Download Jenkins war File

4. Deploy Jenkins war File

5. Install Suggested Plugins

Q11) How can you start Jenkins manually?
Ans: Jenkins can be manually started by opening the Console/Command line and using the below commands:

1. Start Jenkins: jenkins.exe start

2. Stop Jenkins: jenkins.exe stop

3. Restart Jenkins: jenkins.exe restart

Q12) What is the difference between Jenkins, Maven, and Ant?
Ans:

Jenkins

1. Continuous Integration Tool

2. Automates software development process through continuous integration and facilitates continuous delivery
3. Supports version control tools like Git, AccuRev.

Maven

1. Build automation tool
2. Describes software dependencies and explains how the software is built
3. Supports projects written in C#, Ruby.

Ant

1. Command Line/Java Library Tool.
2. Drives build process
3. Supports projects written in C and C++.

Q13) Name a few Jenkins environment variables.
Ans: By default, there are numerous variables available in Jenkins. Some of them are:

$NODE_NAME

$JOB_NAME

$WORKSPACE

$JOB_URL

$BUILD_URL

Q14) How can you deploy a custom build of a core plugin?
Ans: To deploy a custom build of a core plugin, you have to do the following:

Stop Jenkins.
Copy the custom HPI to $Jenkins_Home/plugins.
Delete the previously expanded plugin directory.
Make an empty file called <plugin>. hpi. pinned.
Start Jenkins.

Q15) What are the Parameters in Jenkins?
Ans: Parameters are supported by the Agent section. They are used to support several use-cases pipelines and are defined at the top-level of the pipeline or inside an individual stage directive.

Download the plug-in that supports the third-party tool through the Jenkins administrator console.

The third-party tool must be configured in the administrator console.

At last, the plug-ins can be utilized from inside a Jenkins build job.

Q28) What are the different ways of scheduling a build in Jenkins?
Ans:

Builds can be triggered by source code management commits sequentially.

Manual requests the Builds.

Can be scheduled to run at a specified time using the CRON jobs.

Q29) How will you secure Jenkins?
Ans: The following ways will help you to secure Jenkins:

Check the global security is on.

Make sure Jenkins is integrated with the appropriate login with my company’s user directory.

The project matrix/matrix is allowed to fine-tune access.

Automate the process of setting rights/privileges with custom version controlled script in Jenkins.

Limit physical access to Jenkins data/folders.

Run security audits periodically on the same.

Q30) What are the steps involved in deploying a custom build of a core plugin?
Ans: To deploy a custom build of a core plugin, you should follow the below steps:

Stop Jenkins.

Copy the custom HPI to $Jenkins_Home/plugins.

Delete the previously expanded plugin directory.

Make an empty file called. hpi. pinned.

Start Jenkins

Q31) What are declarative pipelines?
Ans: A declarative pipeline is a new feature in Jenkins that maintains the pipeline as a code and makes it easier to write and read. It is defined within a block labeled pipeline.

Syntax:

The common syntax is:

```
pipeline {
/* Declarative Pipeline */
}
```

Structure of the declarative pipeline:

any – It represents the whole pipeline will run on any available agent.

docker – This is to run the pipeline in the Docker environment.

none – It indicates all the stages under the block will have to be declared with the agent separately.

label – This is just a label for the Jenkins environment

Q32) How to define parameters for a build in Jenkins?

Ans: Build can use various input parameters for execution. For suppose, you have different test suites and you want to run only one then you can select a parameter to decide which one should be run. To define parameters for your job, first, you need to define the same while specifying the parameter. There are different parameter types like string, file, or custom.

Q34) What does SCM mean in Jenkins?
Ans: Source Code Management (SCM) specifies the source code location in Jenkins. The entry point to SCM is defined as jenkins_jobs.scm. The job defined with the SCM attribute accepts many numbers of SCM definitions. Some of the SCM tools are CVS, Git, Perforce, AccuRev, Subversion, Clearcase, RTC, and Mercurial.

SCM can be defined as:

```
scm:
  name: eloc - scm
  scm:
        git:
        url: ssh://Jenkins.org/eloc.git
```

Q35) What is the difference between Agent, post-section, and Jenkinsfile?
Ans:

Agent: A directive that specifies Jenkins on how to execute the pipeline in a particular order or manner.

Jenkinsfile: The text file which defines all the pipelines is called Jenkinsfile. It is being checked in the source control repository.

Post-section: It runs at the end of every pipeline’s execution.

Q36) How to achieve continuous integration using Jenkins?
Ans:

Developers must commit their source code changes to the shared Git repository.

The test results and build results are shared with the respective developers.

Jenkins server checks the shared Git repository at particular time periods and identifies the changes taken in the build.

The clean and tested build is deployed to the production server.

Q37) How do you define a Continuous Delivery Workflow?
Ans: The flowchart below shows the Continuous Delivery Workflow.

Q38) How do you integrate Git with Jenkins?
Ans: The below steps show you how to integrate Git with Jenkins:

Manage the Jenkins button on your Jenkins dashboard.

Click on Manage Plugins.

In the Plugins Page

Select the GIT Plugin and click on Install without restart.

You can also select the option Download now and install after restart.