Q 1: What is DevOps?

Answer: DevOps is a set of practices for achieving continuous delivery. In other words, it’s an integration of development and operations.

Q 2: What is the origin of DevOps?

Answer: DevOps term was coined by Patrick Debois in 2009 at a DevOpsDays conference.

Q 3: What are the main components of DevOps?

Answer:

The main components of DevOps are

* The organizational structure.
* The tools.
* The culture.

Q 4: What is Automated Testing?

Answer:

Automated Testing is a process for validating the functionality of a software product. It is an integral part of the software development cycle. Automated Testing verifies that a software application meets its specified requirements, that it performs as expected, and that it is easy to use. The process of automated testing is designed to be repeatable so that any number of regression tests can be run in an efficient manner. Automated testing is a primary requirement for DevOps.

Q 5. What is the difference between Manual and Automated Testing?

Answer:

Manual Testing gives the tester complete control of the execution of test cases, where as Automated Testing is completely non-interactive. Manual testing is a process of executing a test case by a tester or by an automated tool. Automated testing is a process of executing a test case by an automated tool. Manual testing can also be called as White Box Testing, where as Automated Testing can also be called as Black Box Testing.

Q 6. Mention the key benefits of Automated Testing in Agile Methodology?

Answer:

Automated testing provides faster feedback on the progress of the software development project. Automated testing is less expensive and easier to implement. Automated testing provides more accurate results. Automated testing is compatible with the development methodologies such as Agile and Continuous Integration.

Q 7. Why do we need to automate testing?

Answer:

Automated testing can be used to ensure that quality is built into the product from the beginning of the development lifecycle and automated testing can be used to provide fast and consistent feedback to developers.

Q 8. What are the types of Automated Testing?

Answer:

The following are the different types of automated testing:

Functional Testing – It verifies that the application is working as expected at the user level.

Non-functional Testing – It verifies that the application is working as expected at the non-user level.

Acceptance Testing – It verifies that the application meets the customer’s requirements.

Performance Testing – It verifies that the application performs as expected.

Security Testing – It verifies that the application is secure.

Regression Testing – It verifies that the application is stable.

Smoke Testing – It verifies that the application is not completely broken.

Load Testing – It verifies that the application can handle the load.

Stress Testing – It verifies that the application works under heavy load.

Sanity Testing – It verifies that the application is ready for development.

Unit Testing – It verifies that each unit of code is working as expected.

Integration Testing – It verifies that separate units of code are working together.

System Testing – It verifies that the application is functioning as a whole.

Acceptance Testing is one of the most popular types of automated testing.

Q 9. What is Acceptance Testing?

Answer:

Acceptance testing is also known as User Acceptance Testing (UAT). It is a form of non-functional testing that verifies that a software product meets the requirements of the customer. It is a part of the software development life cycle.

Q 10. What is the difference between Integration Testing and System Testing?

Answer:

Integration Testing is a type of testing that verifies that the application works as a whole. System Testing is a type of testing that verifies that the application works as a whole. It is an important part of the software development life cycle. It is an important part of the software development life cycle. System Testing is a form of non-functional testing.

Q 11. What is the difference between Smoke Testing and Sanity Testing?

Answer:

Smoke Testing is a type of testing that verifies that the application is not completely broken. Sanity Testing is a type of testing that verifies that the application is ready for development. It is used to determine whether a software product is ready for further testing. It is used to determine whether a software product is ready for further testing. It is a form of non-functional testing.

Q 12. Mention the key differences between Unit Testing and Integration Testing?

Answer:

Unit Testing is a type of testing that verifies that each unit of code is working as expected. Integration Testing is a type of testing that verifies that separate units of code are working together. Integration Testing is a form of non-functional testing. Unit Testing is a form of non-functional testing. It is a part of the software development life cycle.

Q 13. What is the difference between Smoke Testing and Regression Testing?

Answer:

Smoke Testing is a type of testing that verifies that the application is not completely broken. Regression Testing is a type of testing that verifies that the application is stable. It is used to determine whether a software product is ready for further testing. It is used to determine whether a software product is ready for further testing. Regression Testing is a form of non-functional testing.

Q 14. What is a Unit Test?

Answer:

A Unit Test is a type of automated test that verifies the functionality of a single unit of source code. It is a part of the software development life cycle.

Q 15. What are the different types of Unit Tests?

Answer:

There are different types of Unit tests:

In-Memory Unit Test – It runs in memory without writing any files.

Clean Room Unit Test – It is used to mock the external dependencies.

Fixture Unit Test – It is used to replace external dependencies.

In-Memory Unit Test – It runs in memory without writing any files.

Black-Box Unit Test – It is a type of Unit Test that uses the public interface of the code.

White-Box Unit Test – It is a type of Unit Test that uses the private interface of the code.

Fixture Unit Test – It is used to replace external dependencies.

Integration Test – It is a type of Unit Test that tests the integration between the classes.

State-Based Test – It is a type of Unit Test that tests the behavior of the methods.

E2E Test – It is a type of Unit Test that tests the whole application.

Q16. What is a System Test?

Answer:

A System Test is a type of automated test that verifies that the application works as a whole. It is a part of the software development life cycle.

Q17. Mention the different types of System Tests.

Answer:

There are different types of System tests. They are:

Black-Box System Test – It uses the external interface of the system to test the system.

White-Box System Test – It uses the internal interface of the system to test the system.

GUI System Test – It is a type of System Test that uses the graphical user interface to test the system.

E2E System Test – It is a type of System Test that tests the application as a whole.

Q 18. What is the difference between Integration Testing and System Testing?

Answer:

Integration Testing is a type of testing that verifies that the application works as a whole. System Testing is a type of testing that verifies that the application works as a whole. It is an important part of the software development life cycle. It is an important part of the software development life cycle. System Testing is a form of non-functional testing.

Q 19. What is the difference between Smoke Testing and System Testing?

Answer:

Smoke Testing is a type of testing that verifies that the application is not completely broken. System Testing is a type of testing that verifies that the application works as a whole. System Testing is an important part of the software development life cycle. Smoke Testing is a form of non-functional testing.

Q 20: Why are open source tools beneficial to DevOps?

Answer: Open source tools are beneficial to DevOps because they are not just available freely but are also available to be used by individual developers which is not the case with commercial software. Open source tools are also beneficial to DevOps because they are highly customizable and are available at no additional cost.

**Ansible:**

Q 21: What is the difference between ansible and chef?

Answer: Ansible is automation software that is written in Python language and it is used to automate the configuration and management of the software. Whereas Chef is also like ansible which automates the configuration of the software. It is also used to automate the configuration and management of the software. But it is written in Ruby and Perl language.

Q 22: What is the difference between Ansible and puppet?

Answer: Ansible and puppet are automation tools that do the same thing, that is, to automate the configuration and management of the software. Ansible is written in Python and puppet is written in Ruby.

Q 23: What is the difference between ansible and sysadm?

Answer: Ansible is automation software that is used to automate the configuration and management of the software. Whereas Sysadm is an application for the administrators which is used to manage the software and it is written in Ruby.

Q 24: What is ansible?

Answer: Ansible is a simple IT automation tool that allows you to configure your servers, deploy applications, and orchestrate more advanced IT tasks like continuous deployments or zero downtime rolling updates. It works over SSH and does not require any software or daemons to be installed on remote nodes.

Q 25: How to use ansible commands?

Answer: Ansible can be used to configure systems, deploy software, and orchestrate more advanced IT tasks like continuous deployments or zero downtime rolling updates. Ansible is written in Python and is used to manage nodes over SSH.

Q 26: What is ansible playbooks?

Answer: Ansible playbooks are YAML-formatted text files that contain lists of tasks, like installing packages or updating configuration files. Tasks are executed in the order listed. Each task can use one or more variables to make the task idempotent and only run when needed.

Q 27: What is ansible facts?

Answer: Facts are variables provided by plugins or the ansible host that can be used in playbooks, tasks, and templates. The following are some examples of facts: ansible\_ssh\_host, ansible\_linux\_distribution, and ansible\_processor\_count.

Q 28: What is ansible variables?

Answer: Variables are used to configure ansible in playbooks and templates. Variables are case sensitive and will throw an error if the variable name is not defined in the playbook or in a variable file. Ansible provides many variables out of the box, but you can create your own variables in order to keep your playbooks and templates clean.

Q 29: What is ansible module?

Answer: Ansible modules are used to manage nodes over SSH. Modules are executed in the order listed and can use variables, templates, and handlers. Modules can be used with the command line or with playbooks.

Q 30: What is the use of vaults in Ansible?

Answer: The vaults are used to store the sensitive data.

Q 31: What happens if the vaults are not configured in ansible?

Answer: The ansible vault command will not be able to encrypt or decrypt the data.

Q 32: What are the two ways to encrypt the data using ansible vault command?

Ans: The two ways to encrypt data using ansible vault command are as follows:

* By using command line
* By using ansible

Q 33: What is Ansible Galaxy?

Ans: AnsibleGalaxy is a community project for sharing Ansible roles. It is hosted on GitHub.

Q 34: How does Ansible Galaxy work?

Ans: AnsibleGalaxy provides a web-based repository for Ansible roles.

Q 35: What is Ansible’s Role?

Ans: Ansible Role is used in Ansible to create a reusable block of tasks to be executed on a remote machine.

Q 36: What is Ansible Tower?

Ans: Ansible Tower is a multi-user, multi-machine management and deployment tool used for the automation of the entire software delivery pipeline across the software development lifecycle.

Q 37: How to create a new project in Ansible Tower?

Ans: “Setting -> Project -> New project”

Q 38: Why Ansible?

Ans: Ansible is an automation and DevOps tool that makes the process easier than Puppet, which is a configuration management tool.

Q 39: Explain YAML?

Ans: YAML is a human-readable data serialization standard. It stands for ‘Yet Another Markup

Q 40: Ansible is written in?

Ans: Ansible is written in Python.

Q 41: Ansible playbooks are written in?

Ans: Ansible playbooks are written in YAML

Q 42: Could you explain the key distinctions between DevOps and Agile?

Ans: Although DevOps and Agile, one of the most prominent SDLC methodologies, have certain similarities, they are fundamentally different approaches to software development. The following are some of the key distinctions between the two:

Agile Approach - In Agile, the agile approach is exclusively for development, whereas in DevOps, the agile approach is for both development and operations.

Practices and Processes - While agile practises include Agile Scrum and Agile Kanban, DevOps processes include CD (Continuous Delivery), CI (Continuous Integration), and CT (Continuous Testing) (Continuous Testing).

Prioritization - Agile prioritises timeliness, but DevOps prioritises both timeliness and quality equally.

Release Cycles - DevOps provides shorter release cycles with faster feedback, whereas Agile merely provides shorter release cycles with delayed feedback.

Feedback Source - Agile relies on customer feedback, whereas DevOps incorporates feedback from self (monitoring tools).

Scope of Work - The scope of work for Agile is merely agility, whereas DevOps involves both agility and the need for automation.

**Jenkins:**

Q 43: What are the ways that pipelines can be created using jenkins?

Ans: There are different ways of creating a pipeline in Jenkins:

* From Manage Jenkins -> Configure System -> Pipeline
* Clone an existing pipeline
* Create a new pipeline using the Jenkins Pipeline plugin

Q 44: What are labels in Jenkins and where it can be utilized?

Ans: A label is a special kind of comment associated with a build. Labels in Jenkins can be used to differentiate your builds with meaningful descriptions. These labels can then be used for filtering and reports.

Q 45: Explain the purpose of build promotion in Jenkins?

Ans: Jenkins provides an option to promote the build to next stage. This is one of the most important features of Jenkins.

Q 46: What are the stages and promotion policies in Jenkins?

Ans: In Jenkins, a stage is a place in the pipeline to which a build may be promoted. The policies/rules that govern whether a build is promoted from one stage to another are called promotion policies. We can configure multiple promotion policies and can also create multiple stages to which a build is promoted.

Q 47: What is the main difference between manual and automatic promotion in Jenkins?

Ans: When we use the automatic promotion option, Jenkins automatically promotes the build to next stage. Whereas, when we use the manual promotion option, it means that the build will not be promoted to the next stage unless we manually promote it to the next stage.

Q 48: What are the different promotion policies available in Jenkins?

Ans: There are three different promotion policies available in Jenkins –

* Always
* Never
* Promote if...

Q 49: Explain the Always promotion policy in Jenkins?

Ans: The Always promotion policy always promotes the build to the next stage irrespective of the status of the build.

Q 50: Explain the Never promotion policy in Jenkins?

Ans: The Never promotion policy never promotes a build to the next stage.

Q 51: Explain the Promote if... promotion policy in Jenkins?

Ans: The Promote if... promotion policy promotes the build to next stage if the build status is successful or unstable.

Q 52: How can we add labels in Jenkins?

Ans: We can add labels in Jenkins by performing the following steps:

* Go to the Project dashboard.
* Click on the Configure link.
* Go to the Post-build Actions tab.
* Select the Add Label... option.

Q 53 What is the difference between creating a job and creating a stage?

Ans: When we create a job, it means that we are creating a new job for a project and this job will have one or more stages. Whereas, when we create a stage, it means that we are creating a new stage for a job and this stage will have one or more builds.

Ques 54: What is the use of the 'Run parameter' option in the Post-build Actions tab?

Ans: The 'Run parameter' option is used to pass a parameter to the build. This parameter can be used in the build script.

Ques 55: How can we pass a parameter to a build in Jenkins?

Ans: We can pass a parameter to a build by performing the following steps:

Go to the Project dashboard.

Click on the Configure link.

Go to the Post-build Actions tab.

Select the Add Parameter... option.

Ques 56: What is the difference between build parameters and build variables?

Ans: A build parameter is an input to the build. Whereas, a build variable is an output from the build.

Ques 57: Explain the 'Build Number Format' option in the Post-build Actions tab?

Ans: The 'Build Number Format' option is used to format the build number in Jenkins.

Ques 58: What is the difference between the 'Build number' and 'Build number with time stamp' options in the Post-build Actions tab?

Ans: The 'Build number' option is used to append a build number to the last portion of the job name. The 'Build number with time stamp' option is used to append a build number with a time stamp to the last portion of the job name.

Ques 59: What is the use of the 'View logs' option in the Post-build Actions tab?

Ans: The 'View logs' option is used to view the build logs.

Ques 60: What is the use of the 'Publish Javadoc' option in the Post-build Actions tab? Ans: The 'Publish Javadoc' option is used to publish the Javadoc of the build to the Jenkins dashboard.

Ques 61: What is the use of the 'Publish test result report' option in the Post-build Actions tab?

Ans: The 'Publish test result report' option is used to publish the test result report of the build to the Jenkins dashboard.

Ques 62: What is the use of the 'Publish performance report' option in the Post-build Actions tab?

Ans: The 'Publish performance report' option is used to publish the performance report of the build to the Jenkins dashboard.

Ques 63: What is the use of the 'Publish coverage report' option in the Post-build Actions tab?

Ans: The 'Publish coverage report' option is used to publish the code coverage report of the build to the Jenkins dashboard.

Ques 64: What is the use of the 'Publish PMD report' option in the Post-build Actions tab?

Ans: The 'Publish PMD report' option is used to publish the PMD report of the build to the Jenkins dashboard.

Ques 65: What is the use of the 'Publish FindBugs report' option in the Post-build Actions tab?

Ans: The 'Publish FindBugs report' option is used to publish the FindBugs report of the build to the Jenkins dashboard.

Ques 66: What is the use of the 'Publish Cobertura coverage report' option in the Post-build Actions tab?

Ans: The 'Publish Cobertura coverage report' option is used to publish the Cobertura coverage report of the build to the Jenkins dashboard.

Ques 67:What is the use of Blueocean in Jenkins?

Ans: Blueocean is a web-based continuous integration and deployment tool.Blueocean is a Jenkins plugin that facilitates the deployment of applications. It can easily create deployment plans and help you to scale your application. It also helps you to catch and solve the issues quickly and helps you to fix them by deploying your application.

Ques 68: What are the many ways to set up Jenkins?

Ans: Jenkins may be set up with the command -

1. apt (Linux), brew (Mac), and other native system package managers
2. Docker (in the Docker registry, popular docker images for Jenkins are available for many systems such as Unix/Mac/Windows).
3. Kubernetes is a container orchestration system (available as a helm chart and can be installed on our Kubernetes clusters)
4. Self-contained (on any machine with a Java Runtime Environment installed)

Ques 69: How to configure inclusions & exclusions in Artifacts Archival?

Ans: Inclusions and exclusions can be configured in Artifactory under Admin > Repository > Inclusions and Exclusions.

Inclusions are used to include the specified artifacts from their repository paths. In other words, if the artifact is found in the repository path specified, it will be included in the archive.

Exclusions are used to exclude the specified artifacts from their repository paths. In other words, if the artifact is found in the repository path specified, it will be excluded from the archive.

In order to use exclusions, you must enable the Include/Exclude option under the Advanced tab of the repository configuration.

Ques 70: How can I push my Artifactory repository to a remote location?

Ans: In order to push your Artifactory repository to a remote location, you must first configure a remote repository in Artifactory. You can then use the remote repository to clone your Artifactory repository.

Ques 71: How to create & use a Shared Library in Jenkins?

Ans: The following are the minimum criteria for a Jenkins shared library to be used in Pipeline Code:

In SCM, a repository containing pipeline common library code. For the Jenkins instance, a proper SCM Plugin configuration. Jenkins Global Configuration should be used to set up the Global Shared Library. Use the methods specified in the Jenkins Shared Library and include the Shared Library in the Pipeline Code.

E.g.

*@Library('fs jenkins shared library@v2.0.7')\_ #!/urs/bin/env groovy*

Ques 72: What is Jenkins Remote Access API?

Ans: The Jenkins remote access API allows you to write your own remote access plugins. To access it, use the Remote Access API Plugin. Example: *http://localhost:8080/jnlpJars/example.jnlp*

Ques 73: Give Some Continuous Monitoring Tools?

Ans: Nagios, Zabbix and Sensu

Ques 74: Give Some Tools to Check Database Performance?

Ans: pgAdmin, pgAgent, pgbouncer and pgAdmin

Ques 75: What is the Performance of PgBouncer?

Ans: It is very high unlike other proxy servers

Ques 76: Which PostgreSQL Commands Can Be Used to Monitor Database?

Ans: SHOW PROCESSLIST, SHOW DATABASES, SHOW SERVERS, and SHOW PAGES

Ques 77: What is the difference between SHOW PROCESSLIST and SHOW FULL PROCESSLIST in PostgreSQL?

Ans: The difference is that the SHOW FULL PROCESSLIST shows the query for the query.

Ques 78: What is the difference between SHOW DATABASES and SHOW TABLES in PostgreSQL?

Ans: SHOW DATABASES shows a list of databases and all tables of the database. On the other hand, SHOW TABLES shows the metadata of all tables in the database.

Ques 79: What is the difference between SHOW SERVERS and SHOW DATABASE in PostgreSQL?

Ans: SHOW SERVERS shows the server information, whereas SHOW DATABASE shows the metadata of the database.

Ques 80: What is the difference between SHOW PAGES and SHOW INDEXES in PostgresSQL?

Ans: SHOW PAGES shows the page information and SHOW INDEXES shows the index information.

Ques 81: What is the difference between SHOW FULL PROCESSLIST and SHOW PROCESSLIST in PostgresSQL?

Ans: SHOW FULL PROCESSLIST shows the query for the query, whereas SHOW PROCESSLIST shows only the query.

Ques 82: What is the difference between SHOW DATABASE and SHOW TABLES in PostgresSQL?

Ans: SHOW DATABASE shows the database information and SHOW TABLES shows the metadata of all tables in the database.

Ques 83: What is Docker?

Ans: Docker is a container platform that allows you to create, deploy and run apps without worrying about the infrastructure. Docker is open-source and its main goal is to make it easier to create, deploy and run applications by using containers.

Ques 84: What is Dockerfile?

Ans: Dockerfile is a file that contains the commands used to build a Docker image.

Ques 85: What is a Docker image?

Ans: A Docker image is a read-only template( a frozen copy) that contains everything needed to run an application.

Ques 86: What is a Docker container?

Ans: A Docker container is a running instance of a Docker image.

Ques 87: What are some of the features of Docker?

Ans: Lightweight Linux container Secure by default Self-contained application packages Fast and efficient.

Ques 88: How to install Docker in Linux?

Ans: Download Docker using the following command:

wget -qO- https://get.docker.com/ | sh

Test the Docker installation using the following command:

sudo docker run hello-world

Ques 89: What is Docker Compose?

Ans: Docker Compose is a tool for defining and running multi-container Docker applications.

Ques 90: What is Docker Swarm?

Ans: Docker Swarm is native clustering for Docker.

Ques 91: What is Docker Universal Control Plane (UCP)?

Ans: Docker Universal Control Plane is an open-source container management platform that allows cluster administrators and developers to manage multiple Docker engines running distributed applications.

Ques 92: What is Docker volume?

Ans: Docker volume is a network-attached storage (NAS) volume.

Ques 93: What is a Docker network?

Ans: Docker network is a group of Docker daemons that can communicate with each other.

Ques 94: What is Docker Compose file?

Ans: Docker Compose file is a YAML file that defines a multi-container application.

Ques 95: What is a Docker container?

Ans: Docker container is a lightweight, standalone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.

Ques 96: What is a Docker registry?

Ans: Docker registry is a centralized storage server that stores and distributes Docker images and allows users to search, download, and upload them.

Ques 97: What is Docker Hub?

Ans: Docker Hub is a cloud-based service that offers Docker management and collaboration tools for public and private repositories.

Ques 98: What are microservices?

Ans: Microservices are a collection of small applications that work together. Microservices are usually implemented in a way that they can be deployed independently and run in a distributed environment.

Ques 99: What is the main advantage of microservices?

Ans: Microservices allow us to decrease the time to market and reduce the time for development. In addition, microservices can be scaled separately, so that we can reduce the impact of an issue in a particular service.

Ques 100: How service communication happens in microservices?

Ans: Communication between microservices happens through APIs.

Ques 101: What is REST?

Ans: REST stands for REpresentational State Transfer. It is an architecture style for developing distributed systems over the internet. RESTful APIs are stateless.

Ques 102: What are the main principles of REST?

Ans:

* Uniform Interface – All resources are accessed via a uniform interface.
* All resources are accessed via a uniform interface. Stateless
* A RESTful service does not maintain any form of a client state.
* A RESTful service does not maintain any form of a client state. Cacheable – A RESTful service allows caching of representations that are served by the service.
* A RESTful service allows caching of representations that are served by the service. Layered System – A RESTful service does not include a complex, monolithic design and is intended to be used in layered system architecture.

Ques 103: Explain the advantages of RESTful architecture.

Ans: The advantages of RESTful architecture are:

Support for HTTP Verbs – RESTful services use HTTP verbs to define the action to be performed.

* RESTful services use HTTP verbs to define the action to be performed. Self-Describing Messages
* Each RESTful message includes information about the action, data format, and the resource being acted upon. – Each RESTful message includes information about the action, data format, and the resource being acted upon. Uniform Interface
* All resources are accessed via a uniform interface. – All resources are accessed via a uniform interface. Stateless – A RESTful service does not maintain any form of client state.
* A RESTful service does not maintain any form of client state. Cacheable – A RESTful service allows caching of representations that are served by the service.

Ques 104: What is the difference between REST and SOAP?

Ans: The main difference between REST and SOAP is the way they communicate. While REST uses HTTP protocol, SOAP uses XML.

Ques 105: What is the difference between GET and POST methods?

Ans: The GET method is used to retrieve data from a website, while the POST method is used to send data to a website.

Ques 106: What is the difference between DELETE and POST methods?

Ans: The POST method can be used to send data to a server while the DELETE method can be used to delete data from a server.

Ques 107: What is the difference between GET and PUT methods?

Ans: The GET method is used to retrieve data from a website, while the PUT method is used to update data on a website.

Ques 108: What is the difference between PUT and POST methods?

Ans: The POST method can be used to send data to a server while the PUT method can be used to update data on a server.

Ques 109: What is the difference between GET and DELETE methods?

Ans: The GET method is used to retrieve data from a website, while the DELETE method is used to delete data from a server.

Ques 110: What is the difference between PUT and PATCH methods?

Ans: The PUT method is used to update a single resource, while the PATCH method is used to update multiple resources.

Ques 111: What is the difference between GET and HEAD methods?

Ans: The GET method is used to retrieve data from a website, while the HEAD method is used to retrieve the headers from a website.

Ques 112: What is Prometheus monitoring?

Ans: Prometheus is monitoring the metrics exposed by the application.

Ques: How do we ensure that metrics are collected and exposed?

Ans: We can ensure that metrics are collected and exposed by writing a Prometheus rule.

Ques 113: What is Prometheus rule?

Ans: Prometheus rule is a config file thatspecifies what type of events should be exposed to Prometheus.

Ques 114: What are the different types of Prometheus rules?

Ans: There are three types of Prometheus rules:

scrape\_config, scrape\_config\_files and scrape\_config\_files\_regex.

Ques 115: What are the differences among scrape\_config, scrape\_config\_files, scrape\_config\_files\_regex and scrape\_config\_regex?

Ans: The differences among scrape\_config, scrape\_config\_files, scrape\_config\_files\_regex and scrape\_config\_regex are:

Scrape\_config: This rule can contain multiple rules in the same file.

Scrape\_config\_files: This rule can contain multiple files inside the same directory. Scrape\_config\_files\_regex: This rule can contain multiple file names inside the same directory with a specific pattern.

Scrape\_config\_regex: This rule can contain multiple regex patterns inside the same directory.

Ques 116: What happens if we add extra metrics to the Prometheus rule with scrape\_config\_regex?

Ans: Extra metrics are not added to Prometheus, as the Prometheus rule with scrape\_config\_regex only matches the metric name, not the metric value.

Ques 117: How do we use Prometheus rules when exposing metrics from applications? Ans: We can use Prometheus rules when exposing metrics from applications by putting the metrics into a Prometheus-supported data format like text or JSON.

Ques 118: What is the format of the metrics that Prometheus supports?

Ans: The metrics that Prometheus supports are text and JSON.

Ques 119: What is the metric format in Prometheus that specifies the data format as text? Ans: The format of the metrics that specifies the data format as text is text.

Ques 120: What is the metric format in Prometheus that specifies the data format as JSON? Ans: The format of the metrics that specifies the data format as JSON is json.

Ques 121: How do we expose metrics from applications through text or JSON format?

Ans: We can expose metrics from applications through text or JSON format by writing a Prometheus rule and exposing the metrics via a data format that Prometheus supports.

Ques 122: What is Kubernetes?

Ans: Kubernetes is an open source system for automating deployment, scaling, and management of containerized applications. It groups containers that make up an application into logical units for easy management and discovery. Kubernetes is designed to scale up from zero to thousands of containers, and it can run on-premise, in a private cloud, or on public cloud platforms like Google Cloud Platform.

Ques 123: What is a Pod?

Ans: A Pod is the smallest deployable unit in Kubernetes. It represents a group of one or more containers, a shared network namespace, and options about how containers in a pod should be scheduled.

Ques 124: What is a Deployment?

Ans: Deployment is a declarative way of saying, “Take this application, here’s its configuration, here’s the container image that you should run, and here’s the volume of replicas that you should run of it.” The advantage of using deployments rather than just kubectl run is that deployments are versioned and can be updated without restarting your containers. Deployments are always tied to a particular pod template. When you update deployment, the new image is pulled, traffic is redirected to the new pod, and the old pod is then killed.

Ques 125: What is a ReplicationController?

Ans: Replication controllers are responsible for ensuring that a specified number of pod “replicas” are running at any given time. A replication controller is an abstraction on top of a deployment, which is a declarative way of saying, “I want three replicas of this,” and the replication controller is responsible for making sure that the desired number of replicas is running.

Ques 126: What is the default Service Type in Kubernetes?

Ans: ClusterIP

Ques 127: What is the difference between ReplicationController and ReplicaSet?

Ans: ReplicationController is designed for stateless applications. ReplicaSet is designed for stateful applications.

Ques 128: List types of load balancer in Kubernetes?

Ans: There are several load balancer types in Kubernetes.

1. Ingress.

2. NodeIP.

3. External.

Ques 129: Explain the Architecture of K8s?

Ans:

Controller Manager:

It is the main component of K8s, which manages all components of K8s. It is responsible for the execution of K8s objects like Deployment, Pod, Service and ReplicaSet.

Kubelet:

Kubelet is a process responsible for the creation of the pods. It interacts with the API server by sending the API a command or a status of the node.

Kube-proxy:

Kube-proxy is responsible for the IPTables of the K8s cluster. It uses iptables to redirect the external traffic to the service containers based on the resource requests.

Etcd:

Etcd is a highly-available key-value store. All the components of K8s use etcd to store the data. It is a distributed database that enables high performance and reliability.

Scheduler:

It is responsible for the placement of the workload on the K8s cluster. K8s makes use of the resource requests to make the scheduling decisions.

Kube-apiserver:

It is the main entry point to K8s. It is responsible for handling all the requests coming from the API clients. It is the first component to be started in the K8s cluster.

Kube-controller-manager:

It is responsible for the creation and deletion of the controllers. It also registers and deregisters the controllers.

Kube-scheduler:

It is responsible for the scheduling of the workloads on the K8s cluster. It is responsible for assigning the pods to the nodes in the cluster.

Ques 130: Explain the K8s Architecture for Services?

Ans: The services architecture is responsible for the communication between the pods. The services make use of the labels to identify the pods. The services are always made available in the cluster. It is always recommended to use the services for the communication between the pods.

Ques 131: Explain the k8s Architecture for Deployments?

Ans: The deployments architecture is responsible for the deployment of the pods. The deployments are always available in the K8s cluster. The deployments are always dependent on the services.

Ques 132: Explain the k8s Architecture for ReplicaSets?

Ans: The replica sets are responsible for the replication of a pod. The replica sets are always dependent on the deployments.

Ques 133: Explain the k8s Architecture for Labels?

Ans: The labels are used for the identification of the pods. The labels are used for the replication of the pods. The labels are always dependent on the deployments.

Ques 134: Give an example of a Pod?

Ans: The below given example shows the definition of the pod.

apiVersion: v1 kind: Pod metadata: name: node-test-34-pod spec: containers: - name: test-container image: k8s.gcr.io/busybox command: [“/bin/sh”] args: [“-c”, “while true; do echo hello world; sleep 10; done”] ports: - containerPort: 80 hostPort: 80 name: http volumeMounts: - name: data mountPath: /data - name: config mountPath: /config

Ques 135: What is the use of Pods?

Ans: The pods are responsible for the running process in the cluster. It is a collective name for the container, volumes, and IP Address.

Ques 136: Explain the process of creating a Pod?

Ans: Below given is the command for creating a pod.

kubectl run mypod --image=nginx –replicas=2 –labels=‘node=node-1’

Ques 137: Explain the process of creating a ReplicaSet?

Ans: Below given is the command for creating a ReplicaSet.

kubectl run mypod --image=nginx –replicas=2

Ques 138: Explain the process of creating a Deployment?

Ans: Below given is the command for creating a Deployment.

kubectl run mypod –image=nginx –replicas=2

Ques 139: Explain the process of creating a service?

Ans: Below given is the command for creating a service.

kubectl expose deployment mypod --type=NodePort

Ques 140: Explain the process of creating a DaemonSet?

Ans: Below given is the command for creating a DaemonSet.

kubectl run mypod --image=nginx –replicas=2 –labels=‘node=node-1’

Ques 141: What is the use of Services?

Ans: The services are used for the communication between the Pods. The services are always available in the cluster. The services make use of the labels to identify the pods.

Ques 142: What exactly is GIT?

Ans: GIT is a distributed version control and source code management (SCM) solution designed to handle both small and big projects quickly and efficiently.

Ques 143: In GIT, what is a repository?

Ans: A repository has a.git directory in which git stores all of the repository's metadata. Git keeps the contents of the.git directory private.

Ques 144: What exactly is the distinction between GIT and SVN?

Ans: The distinction between GIT and SVN is as follows:

a) Git is not recommended for handling exceptionally large files or binary files that change regularly, but SVN can handle numerous projects in the same repository.

b) Commits across multiple branches or tags are not supported by GIT. Subversion allows you to create folders at any point in the repository hierarchy.

c) Gits are immutable, but Subversion allows committers to use a tag like a branch and produce numerous revisions under a single tag root.

Ques 145: What are the advantages of using GIT?

Ans: Checkout the advantages of using GIT:

1. GIT supports branching and merging

2. GIT supports distributed workflows

3. GIT supports rebase and recursive merge

4. GIT supports complicated histories

5. GIT support undo

6. GIT supports patches and diffs

7. GIT supports binary files

8. GIT supports partial checkouts

9. GIT supports non-linear development

10. GIT supports both centralized and decentralized workflows

Ques 146: What language is used in GIT?

Ans: GIT is fast, and the ‘C’ language makes this possible by reducing the overhead of runtimes associated with higher languages.

Ques 147: In GIT, what does "Staging Area" or "Index" mean?

Ans: It can be structured and examined in an intermediary region called as the 'Staging Area' or 'Index' before committing it.

Ques 148: What is GIT stash?

Ans: GIT stash takes the current state of the working directory and index and puts in on the stack for later and gives you back a clean working directory. So in case if you are in the middle of something and need to jump over to the other job, and at the same time you don’t want to lose your current edits then you can use GIT stash.

Ques 149: What is GIT stash drop?

Ans: When you are done with the stashed item or want to remove it from the list, run the git ‘stash drop’ command. It will remove the last added stash item by default, and it can also remove a specific item if you include it as an argument.

Ques 150: How will you know in GIT if a branch has been already merged into master?

Ans: Git branch—merged lists the branches that have been merged into the current branch

Git branch—-no merged lists the branches that have not been merged

Ques 151: What is the function of the git clone?

Ans: The git clone command creates a copy of an existing Git repository. To get a copy of a central repository, ‘cloning’ is the most common way used by programmers.

Ques 152: What is the function of ‘git config’?

Ans: The ‘git config’ command is a convenient way to set configuration options for your Git installation. Behavior of a repository, user info, preferences, etc. can be defined through this command.

Ques 153:What does commit object contain?

a) A set of files, representing the state of a project at a given point of time

b) Reference to parent commit objects

c) An SHAI name, a 40 character string that uniquely identifies the commit object.

Ques 154:How can you create a repository in Git?

Ans: In Git, to create a repository, create a directory for the project if it does not exist, and then run the command “git init”. By running this command .git directory will be created in the project directory, the directory does not need to be empty.

Ques 155: What is ‘head’ in git and how many heads can be created in a repository?

Ans: A ‘head’ is simply a reference to a commit object. In every repository, there is a default head referred to as “Master”. A repository can contain any number of heads.

Ques156: What is the purpose of branching in GIT?

Ans: The purpose of branching in GIT is that you can create your own branch and jump between those branches. It will allow you to go to your previous work keeping your recent work intact.

Ques 157:What is the common branching pattern in GIT?

Ans: The common way of creating a branch in GIT is to maintain one as “Main“branch and create another branch to implement new features. This pattern is particularly useful when there are multiple developers working on a single project.

Ques 158:How can you bring a new feature to the main branch?

Ans: To bring a new feature in the main branch, you can use the command “git merge” or “git pull command”.

Ques 159:What is a ‘conflict’ in git?

Ans: A ‘conflict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. Git will not be able to predict which change should take precedence.

Ques 160: How can conflict in git resolved?

Ans: To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running “git add” after that to commit the repaired merge, run “git commit”. Git remembers that you are in the middle of a merger, so it sets the parents of the commit correctly.

Ques 161: To delete a branch what is the command that is used?

Ans: Once your development branch is merged into the main branch, you don’t need

development branch. To delete a branch use, the command “git branch –d [head]”.

Ques 162: What is another option for merging in git?

Ans: “Rebasing” is an alternative to merging in git.

Ques 163: What is the syntax for “Rebasing” in Git?

Ans: The syntax used for rebase is “git rebase [new-commit] “

Ques 164: What is the function of ‘git diff ’ in git?

Ans: ‘git diff ’ shows the changes between commits, commit and working tree etc.

Ques 165: What is ‘git status’ is used for?

Ans: As ‘Git Status’ shows you the difference between the working directory and the index, it is helpful in understanding a git more comprehensively.

Ques 166: What is the difference between the ‘git diff ’and ‘git status’?

Ans: ‘git diff’ is similar to ‘git status’, but it shows the differences between various commits and also between the working directory and index.

Ques 167:What is the function of ‘git checkout’ in git?

Ans: A ‘git checkout’ command is used to update directories or specific files in your working tree with those from another branch without merging it in the whole branch.

Ques 168: What is the function of ‘git rm’?

Ans: To remove the file from the staging area and also of your disk ‘git rm’ is used.

Ques 169: What is the function of ‘git stash apply’?

Ans: When you want to continue working where you have left your work, ‘git stash apply’ command is used to bring back the saved changes onto the working directory.

Ques 170: What is the use of ‘git log’?

Ans: To find specific commits in your project history- by author, date, content or history ‘git log’ is used.

Ques: What is ‘git add’ is used for?

Ans: ‘git add’ adds file changes in your existing directory to your index.

Ques 171: What is the function of ‘git reset’?

Ans: The function of ‘Git Reset’ is to reset your index as well as the working directory to the state of your last commit.

Ques 172: What is git Is-tree?

Ans: ‘git Is-tree’ represents a tree object including the mode and the name of each item and the SHA-1 value of the blob or the tree.

Ques 173: How git instaweb is used?

Ans: ‘Git Instaweb’ automatically directs a web browser and runs a webserver with an interface into your local repository.

Ques 174: What does ‘hooks’ consist of in git?

Ans: This directory consists of Shell scripts that are activated after running the corresponding Git commands. For example, git will try to execute the post-commit script after you run a commit.

Ques 175: Explain the major components of Chef?

Ans: Chef is a configuration management tool. It mainly contains three major components as follows.

Chef-Server: It acts as the central data store for Chef. Where all the configurations and details of the nodes are stored.

Chef-Client: The chefdk is a client which is used to connect to the server. The Client sends all the details automatically to the server.

Knife: This is a command line interface that helps to automate the chef-client.

Ques 176: What are the features of Chef?

Ans: The features of Chef are as follows. Chef is an open-source as well as a community-based tool. Chef is compatible with all the major platforms like Windows, Linux, Mac, UNIX, IBM AIX, etc. Chef can be integrated with DevOps tools like Docker, Ansible, Puppet, and Jenkins. Chef works as a standalone tool as well. Chef provides us with centralized control of the configuration of the nodes.

Ques 177: How to install Chef?

Ans: Chef can be installed in two ways.

Standalone: We can install Chef as a standalone node. We can install the Chef on the local system.

Server: We can also install chef-server in a centralized system.

Ques 178: How to install the chef server?

Ans: Chef can be installed as a standalone or a server using the following steps: Download the latest version of chef-server from https://www.chef.io/download/. After downloading the archive, extract it in a directory. After extracting the archive, copy all the files to /opt directory. After that, run the following command to install the chef server. $ sudo ./install.sh

Ques 179: How to install chef-client?

Ans: Chef-client can be installed in two ways.

Standalone: We can install chef-client on the system where the chef is installed, we can even install the chefdk on the local system.

Server: We can also install the chef-client using the chef-server.

Ques 180: What is Continuous Integration?

Ans: Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily – leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible.

Ques 181: What is a build?

Ans: The process of compiling, assembling, and packaging source code, associated assets, test suites, and so on, into a format that can be deployed.

Ques 182: What is a unit test?

Ans: A unit test is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine if they are fit for use.

Ques 183: What is a component test?

Ans: Component testing, sometimes called component-level testing, is a black box testing-based software testing method. Component tests are usually written by the programmer in collaboration with the test team. They are intended to test the individual components of an application separately from the rest of the application and from its surrounding environment.

Ques 184: What is an integration test?

Ans: Integration testing (sometimes called Integration and Testing, abbreviated I&T) is the phase in software testing in which individual software components are combined and tested as a group.

Ques 185: What is a regression test?

Ans: A regression test is the retesting of a repaired or modified software to make sure the change has not introduced new bugs. A change that has been checked in but not yet released to the users is a candidate for regression testing.

Ques 186: What is a Load test?

Ans: Load testing is a form of software testing that attempts to determine the behavior of a system or network as the system is subjected to increasing workloads. It is essentially a more thorough version of stress testing. Load testing is often confused with performance testing, which is essentially load testing done for the purpose of benchmarking.

Ques 187: What is a performance test?

Ans: A performance test evaluates the behavior of a system or network as it is subjected to a specified workload.

Ques 188: What is a smoke test?

Ans: A smoke test (or sanity test or Sanity check) is a type of software test in which a developer or tester checks out a new build of software to see if it crashes or exhibits other obvious faults.

Ques 189: What is a Sanity test?

Ans: A sanity test is a form of software verification that builds on smoke testing by adding more detailed and specific test cases.

Ques 190: What is a Deployment test?

Ans: A deployment test is a software verification and validation test involving the deployment of a system configuration to a location for the purpose of determining if it will satisfy the user’s needs.

Ques 191: What is an exploratory test?

Ans: An exploratory test is a type of software testing that emphasizes the personal freedom of the individual tester to drive the test process to whatever level of detail he or she finds most appropriate.

Ques 192:

What is A/B testing?

Ans: A/B testing is a methodology of evaluating two variants of a web page, webpage, web-app, landing page or anything users interact with. A/B testing is used to determine the optimal variant of a webpage based on the users’ response.

Ques 193: How does A/B testing work?

Ans: A/B testing works by showing two variants of a webpage to the users. One variant is the original webpage and the other is the modified webpage. Users are shown the different variants in a ratio of 1:1. The users choose the variant they like the most. The variant which gets the highest conversion from the user is deemed as the optimal variant.

Ques 194: What is multivariate testing?

Ans: Multivariate testing is an extension of A/B testing. It evaluates multiple variants at the same time. So instead of showing two variants to the users, it shows a number of variants to the users.

Ques 195: What is the difference between A/B testing and multivariate testing?

Ans: In A/B testing, variants are shown to the users in a ratio of 1:1, while in multivariate testing, the ratio can be any number greater than 1.

Ques 196: How does multivariate testing work?

Ans: Multivariate testing works by showing a number of variants to the users in a ratio of any number greater than 1. It shows the different variants to the users and the users choose the variant they like the most. The variant which gets the highest conversion from the user is deemed as the optimal variant.

Ques 197: what is Openshift?

Ans: OpenShift is a platform as a service (PaaS) that is developed by Red Hat. OpenShift is based on containers and provides a development platform. It is a free and open-source platform. OpenShift is a hybrid of Platform as a service (PaaS) and container **as** a service (CaaS).

Ques 198: What is Selenium in testing?

Ans: Selenium is an open-source software framework for web application testing. The framework is written in web-based languages like PHP, Java and Python.

Ques 199: What is the benefit of using Selenium?

Ans: Selenium is a Behavior-Driven Development framework in which you can write your test cases in a standard language like Java or C#, and then run them in any environment.

Ques 200: What is the difference between Selenium RC and Selenium WebDriver?

Ans: Selenium RC is the first implementation of Selenium. It is Selenium version 1.0.0. Selenium Server is the server-side component of the Selenium system. It hosts the Selenium Remote Control (RC) server to support requests from the Selenium client. RC is a language-independent protocol for controlling a web browser. Selenium WebDriver is the new implementation of Selenium. It is Selenium version 2.0.0. Selenium WebDriver is the combination of Selenium RC and WebDriver. It is a browser-independent driver for automated testing of web applications.

Ques 201: What is the functionality of Selenium WebDriver?

Ans: Selenium WebDriver is a browser-independent driver for automated testing of web applications. Selenium WebDriver provides support for the following browsers: Mozilla Firefox Google Chrome Internet Explorer Opera Safari

Ques 202: What is Selenium Grid?

Ans: Selenium Grid is a distributed system to run tests in multiple browsers.

Ques 203: What is the use of Selenium Grid?

Ans: Selenium Grid allows you to run the test in multiple browsers simultaneously. Selenium Grid is used for load testing.

Ques 204: How Selenium Grid works?

Ans: Selenium Grid works as a hub and spoke model. The hub is a service running on a single machine that receives test requests from the client. All the spokes are the machines that actually run the tests.

Ques 205: What is Junit4?

Ans: Junit4 is the latest version of Junit which is used to test Java applications.

Ques 206: What is an assertion?

Ans: An assertion is a condition that Junit checks before running a test.

Ques 207: What is Jira?

Ans: Jira is a software-as-a-service (SaaS) issue tracking application from Atlassian that provides bug tracking, issue tracking, and project management.

Ques 208: What is BitBucket?

Ans: BitBucket is a hosted source code repository and online code management tool for software developers.

Ques 209: What is Rake in DevOps?

Ans: Rake is a tool used to build and deploy applications, in a controlled and repeatable way. It is an essential part of the Ruby ecosystem and is used to manage other tools, such as Capistrano.

Ques 210: What is the difference between Maven and Gradle? Which one is better?

Ans: The difference between Maven and Gradle is that Maven is a software model, whereas Gradle is a build tool. Maven is an XML-based dependency management tool for Java projects. It’s a tool that inspects the developing code and downloads the required dependency and builds the code. Gradle is a build automation tool. It is an open-source build system that builds, tests and deploys applications. It uses a Groovy-based domain-specific language (DSL) for writing build scripts, which are highly extensible.

Ques 211: What is a Job DSL in Gradle?

Ans: The Gradle “Job DSL” is a way to define a set of tasks and dependencies so that you can run them as a batch.

Ques 212: What is the difference between Java and Kotlin?

Ans: Kotlin is a statically-typed programming language that runs on the JVM. It is fully interoperable with existing Java code, but it also adds features that are inspired by languages like Scala and Groovy.

Ques 213: What is Gradle Wrapper?

Ans: Gradle is a repeatable, predictable build tool. The Gradle wrapper is a tool that downloads and installs Gradle automatically. The Gradle wrapper is a zip file that contains a small executable file.

Ques 214: What is a dependency?

Ans: A dependency is a set of classes, usually provided by a library, that a project needs to compile and run. A project declares its dependencies in a file called a “project. properties” file or “build.gradle” file, depending on whether you are using Maven or Gradle.

Ques 215: What is the difference between compile and runtime dependencies?

Ans: A compile dependency is a class that is used by the project during the compile process. A runtime dependency is a class that is used by the project when the application is running.

Ques 216: What is a maven?

Ans: Maven is a tool used to build, deploy and manageJava projects.

Ques 217: What is pom?

Ans: POM is metadata of the maven project.

Ques 218: What is POM?

Ans: POM is metadata of the maven project.

Ques 219: What is pom.xml?

Ans: POM.xml is metadata of the maven project.

Ques 220: What is a maven-compiler-plugin?

Ans: The maven compiler plugin is used to compile java files.

Ques 221: What is the maven plugin?

Ans: The maven plugin is used to create the plugin of the maven project.

Ques 222: What are build automation tools?

Ans: Build automation tools help to automate the build process of a software application by simplifying the process of executing the build command to compile the application code.

Ques 223: Why do we need to build automation tools?

Ans: Build automation tools help to automate the process of building the code and make it easier to track changes to the code base. Build automation tools also helps in improving the overall build time and quality by running the build at specified intervals.

Ques 224: What are the general tasks performed by building automation tools?

Ans: Build automation tools perform the following general tasks:

* Pre-processing the source code and compiling it.
* Building the code.
* Building the documentation.
* Packaging the output.
* Running unit tests.
* Building the web and GUI applications.

Ques 225: What is gradle?

Ans: “Gradle is a build tool that builds upon the concepts of Apache Ant and Apache Maven and introduces a Groovy-based domain-specific language (DSL) instead of the more traditional XML form of declaring the project configuration.”

Ques 226: What does Gradle build?

Ans: Gradle builds Java, Groovy, Scala, Android, C/C++, and more.

Ques 227: What are Gradle build scripts?

Ans: Gradle build scripts are Groovy scripts that describe how to build a project. These scripts are typically called build.gradle files and can be located in the root directory of a project or in subdirectories.

Ques 228: What are Gradle build phases?

Ans: Gradle build phases are the steps Gradle goes through to build a project.

Ques 229: What are Gradle tasks?

Ans: Gradle tasks are the actions that Gradle performs as it builds a project.

Ques 230: What is “gradlew” and “gradlew.bat”?

Ans: gradlew is the command-line wrapper script for the Gradle Daemon. It starts the Daemon as a background process, sets some environment variables, and runs the build. gradlew is a part of the Gradle distribution, included in the bin/ directory.

Ques 231: What is Nagios?

Ans: Nagios is an open source host and network monitoring system. It's designed to inform you when servers and other network hardware go down or when your resource usage gets higher than expected. This can help you avoid some of the negative consequences of failing hardware or services. It also helps you identify problems before your clients, end-users or managers do. This makes it ideal for server monitoring, network monitoring and infrastructure monitoring. In essence, Nagios is the swiss army knife for monitoring.

Ques 232: What are the main features of Nagios?

Ans: Nagios is feature rich, and can provide a variety of different monitoring services. Some of the main features are:

Monitoring of networks and hosts Event notification

Flexible notification methods

Parallelized service checks

Event correlation

Ability to define scheduled downtime

Performance and load data and graphs

Generation of reports Whole-site visibility

Ques 233: What is Terraform?

Ans: Terraform is a tool used for managing infrastructure as code.

Ques 234: What is the difference between Terraform and Chef?

Ans: Chef is a tool used for provisioning infrastructure. Terraform is a tool used for managing infrastructure as code.

Ques 235: What is the difference between Terraform and CloudFormation?

Ans: Terraform is a tool used for managing infrastructure as code. CloudFormation is a tool used for provisioning infrastructure.

Ques 236: What is the difference between Terraform Providers and Resources?

Ans: Terraform Providers are used for connecting to different cloud providers. Terraform Resources are used for describing infrastructure components.

Ques 237: What are the Terraform Providers that are supported by Azure?

Ans: The Terraform Providers that are supported by Azure are: Azure Azure\_Web\_App Azure\_Storage Azure\_Website

Ques 238: What is the syntax for Azure Provider?

Ans: The syntax for Azure Provider is: provider "azurerm" { }

Ques 239: What is AWS?

Ans: AWS is Amazon Web Server. You can store your data on this server. AWS can be used for storage as well as for processing.

Ques 240: What is AWS S3?

Ans: AWS S3 stands for Amazon Simple Storage Service. You can store your data on this server and get access to it anywhere.

Ques 241: What is EC2?

Ans: EC2 stands for Elastic Compute Cloud. It is a web service provided by Amazon Web Services.

Ques 242: What is RDS?

Ans: RDS stands for Relational Database Service. It is a database service provided by Amazon Web Services. You can set up your own database servers in the cloud.

Ques 243: What is DynamoDB?

Ans: DynamoDB is a database service provided by Amazon Web Services. You can store your data in the cloud and get access to it anywhere.

Ques 244: What is IAM?

Ans: IAM stands for Identity and Access Management. It is a security service provided by Amazon Web Services.

Ques 245: How can I search for data on AWS?

Ans: You can search for data on AWS using the search bar on the AWS home page. You can find out the data stored on AWS and sign up for it.

Ques 246: What is Aws Lambda?

Ans: AWS Lambda is a serverless compute service that runs code in response to events and automatically manages the compute resources required by that code.

Ques 247: What is API Gateway?

Ans: API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure REST APIs at any scale.

Ques 248: What is Apache Kafka?

Ans: Apache Kafka is a distributed streaming platform that enables real-time data processing.

Ques 249: What is the main advantage of using Kafka?

Ans: Kafka is highly scalable and supports millions of messages per second.

Ques 250: What is the replication factor in Kafka?

Ans: Replication factor is the number of copies of the data in the cluster.

Ques 251: What is the partitioner in Kafka?

Ans: Partitioner is used to determine how the data is divided among the nodes.

Ques 252: What is the default partitioner in Kafka?

Ans: The default partitioner in Kafka is ‘Murmur3Partitioner’.

Ques 253: What is the responsibility of the Broker in Kafka?

Ans: Brokers provide the API for writing and reading messages from the various topics in Kafka.

Ques 254: What is the responsibility of the Topic in Kafka?

Ans: Topic is a source of the data. It is used to store the messages in Kafka.

Ques 255: What are the various types of topics in Kafka?

Ans: Kafka supports three types of topics: Ordered Topic Partitioned Topic Unordered Topic

Ques 256: What is the role of a ZK in Kafka?

Ans: ZK isthe Zookeeper. It is used to manage the cluster metadata.

Ques 257: How many ZKs are there in a Kafka cluster?

Ans: There are three ZKs in a Kafka cluster.

Ques 258: What are the various tools used for monitoring Kafka?

Ans: The various tools used for monitoring Kafka are:

Kafka console

Kafka manager

Ques 259: What are the advantages of Kafka?

Ans:

1. Kafka is highly scalable and supports millions of messages per second.
2. Kafka is fault-tolerant and uses replication to increase fault tolerance.
3. Kafka is distributed and highly available.

Ques 260: **Ques: What is Rabbit Mq?**

Ans: Rabbit Mq is an open-source message broker written in Erlang. Rabbit MQ is based on the AMQP protocol.

Ques 261:What is the difference between AMQP and AMQP 0-9-1?

Ans: AMQP and AMQP 0-9-1 are the same things. AMQP 0-9-1 is the official specification of the protocol.

Ques 262: What is the difference between the Rabbit MQ and the Active MQ?

Ans: Rabbit Mq is based on the AMQP protocol and Active Mq is based on the JMS protocol.

Ques 263: What are the advantages of Rabbit Mq over Active MQ?

Ans: Rabbit Mq has a lower memory footprint and it is more scalable than the Active Mq.

Ques 264: What is the difference between the Active Mq and the Rabbit Mq?

Ans: Active Mq is a Java-based message broker and Rabbit Mq is an Erlang Based Message broker.

Ques 265: What is Apache Camel?

Ans: It is a powerful open-source enterprise service bus that is available for Java, .Net, and Python. It is mostly known for the integration of Enterprise Applications with Apache ServiceMix.

Ques 266: Explain Apache Camel Architecture?

Ans: It consists of a core library with a number of Java classes that implement the client and server-side of a messaging system. Apache Camel also has a number of Java extension modules that provide additional functionality such as support for FTP, JMS, AMQP, and others.

Ques 267: What are the advantages of using a camel?

Ans: It is an open-source framework. It is available in many languages. It has support for Mule, Apache ServiceMix, and JBoss. It has support for POJO. It is easy to use.

Ques 268: What are the Core components of Apache Camel?

Ans:

* Endpoint: It is used to connect to a remote service.
* Route: It is used to process messages.
* Component: It is used to do something with the message.
* Interceptors: It is used for intercepting a message.
* Producer: It is used for sending messages to the remote service.
* Consumer: It is used for receiving messages from the remote service.

Ques 269: What is DataDog?

Ans: DataDog is an open-source hosted monitoring service for real-time monitoring of servers, applications, and services. DataDog allows users to monitor the entire stack of their infrastructure and the applications and services running on top of it. It also provides users insights through a set of metrics and dashboards.

Ques 270: What does DataDog offer?

Ans: DataDog offers a whole array of features for monitoring, alerting, and analysis. This includes: Monitoring of services and applications Alerting and notification Performance graphs Dashboard Historical data Support.