==========GIT==========

1.**can, you explain branching strategy?**
Ans: Branching strategy is a method used in software development to manage and organize code changes by creating separate branches for different features or versions of the code.

2.**differene between git pull and fetch?**
Ans: **git Pull** fetches and merges changes from a remote repository into the current branch. **git fetch** only downloads changes from the remote repository into the local repository, without merging them into the current branch.

3.**what is git merging? and merge conflicts?**
Ans: Git merging is the process of combining two or more branches in Git. Merge conflicts occur when Git is unable to automatically merge changes from two different branches, usually due to conflicting changes made to the same file(s).

4.**what is pull request?**
Ans: A pull request is a request to merge changes from one branch or repository to another in a code repository management system.

5.**what is the command to delete remote branch?**
Ans: git push origin --delete <Branch name>

6.**how to integrate Jenkins with git?**
Ans: To integrate Jenkins with Git, you need to install the Git plugin in Jenkins and then configure Jenkins to use Git for your project. Once the Git plugin is installed, you can configure your Jenkins project to use Git by specifying the Git repository URL and the branch to build. This can be done either through the Jenkins web interface or by using a Jenkins file in your project repository.

7.What is a webhook?
Ans: A webhook is a method of automating data transfer between web applications by sending real-time notifications from one application to another.

==========MAVEN==========

8. explain maven lifecycle goals?
Ans: Maven lifecycle goals are a predefined set of tasks that are executed during a project build process, from compiling source code to packaging the final artifact.

9.**what is m2 repository?**
Ans: The M2 repository is a local cache of all dependencies required by Maven to build a project.

10.How to integrate nexus repository in maven?
Ans: To integrate Nexus repository in Maven, you need to add the Nexus repository URL to your Maven settings.xml file as a mirror.

11. What is the goal you are using for uploading the antifactory Jenkins to maven?
Ans: The goal of uploading the Antifactory Jenkins to Maven is to make the Jenkins plugin available for distribution and consumption through the Maven repository system.

12.**what is dependency management?**
Ans: Dependency management is the process of identifying and tracking the relationships between software components and ensuring that the necessary dependencies are available and compatible with each other.

==========JENKINS==========

13.**how to upgrade the Jenkins?**
Ans: Upgrading Jenkins cannot be done in a single line as it involves multiple steps and may require different commands depending on the operating system and Jenkins installation.
Here's a general guide to upgrade Jenkins:

- Take a backup of your existing Jenkins instance.
- Download the latest version of Jenkins from the official website.
- Stop the Jenkins service.
- Install the new version of Jenkins by running the appropriate commands based on your operating system.
- Start the Jenkins service.

14.**what type of job you worked?**
Ans: I have worked with Jenkins in setting up and maintaining Continuous Integration and Continuous Deployment pipelines for various software projects.

15.**what are plugins and what type plugins you installed in Jenkins?**
Ans: A plugin is a software component that adds specific functionality to a larger software system. In Jenkins, I may install various types of plugins such as source code management, build tools, testing frameworks, deployment, and notification plugins, among others.

16.**how do you take backup for Jenkins jobs?**
Ans: To take a backup of Jenkins jobs, you can simply copy the Jenkins home directory to create a backup of all the jobs, configurations, and other related data.

17.how to migrate from one server to another server?
Ans: To migrate one server to another server, you need to transfer all the necessary data and configurations from the old server to the new server, which typically involves copying files, databases, and software installations, and reconfiguring any settings that are specific to the new server's environment.

18.**How to provide user access in Jenkins?**
Ans: To provide user access in Jenkins, you can follow these steps:

- Click on "Manage Jenkins" on the Jenkins dashboard.
- Click on "Manage Users" and then "Create User" to create a new user account.
- Enter the required details such as username, password, full name, and email address for the new user.
- Click on "Create User" to save the new user account.
- To assign permissions to the user, click on "Manage Jenkins" and then "Configure Global Security".
- Under the "Authorization" section, select "Matrix-based Security" and

click on "Add User/Group" to add the new user.

- Select the checkboxes for the desired permissions for the user, such as "Overall Read" or "Job Configure".
- Click on "Save" to apply the changes.

In summary:
Create a user -> Configure global security -> Add user to matrix -> Select permissions -> Save.

19. What is the purpose of Jenkins role?
Ans: Jenkins is a tool used for continuous integration and continuous delivery/deployment (CI/CD) purposes.

20. did you return any type ci/cd pipelinescripts?
Ans: I have written several CI/CD pipeline scripts using various tools and technologies such as Jenkins, GitLab CI/CD, Travis CI, AWS Code Pipeline, and Azure DevOps. However, it is not possible to provide a single line of code for a complete CI/CD pipeline script as it involves several stages such as build, test, deploy, and release, and each stage requires multiple commands and configurations. The pipeline script needs to be tailored based on the specific requirements of the project and the tools being used.

21.**what are the pipeline stages?**
Ans: Pipeline stages refer to the various steps or phases involved in a process or workflow, typically used in the context of software development or data analysis.

22.**what is continuous integration and continuous deployment?**
Ans: Continuous Integration (CI) and Continuous Deployment (CD) are software development practices that involve continuously integrating and testing code changes and automatically deploying those changes to production environments.

23.**differene between ci/cd?**
Ans: CI stands for Continuous Integration, which is the process of regularly merging code changes from multiple developers into a shared repository to ensure that the changes are compatible and working together.

CD stands for Continuous Delivery/Deployment, which is the process of automatically deploying code changes to production environments after they have passed through a series of automated tests and other quality checks.

24. What is agile methodology?
Ans: Agile methodology is an iterative and collaborative approach to project management that emphasizes flexibility and adaptability.

==========DOCKER==========

25. Did you write any docker files? give me explanation?
Ans: I have written Docker files which are configuration files used to build Docker images that can run containerized applications. A Docker file contains instructions on how to build an image by specifying a base image, adding custom configurations, and installing dependencies.

26.**what is difference between copy and add?**
Ans: COPY instruction just copies the files from the local host machine to the container file system. ADD is also used to copy the files and retrieve files from remote URL's.

27.**what is difference between CMD and entry point?**
Ans: **CMD** specifies the default command to be executed when running a container.
Entry point specifies the command to be executed when running the container, and any additional arguments are passed as parameters to the command.

28.**can, you explain docker structure?**
Ans: Docker is a software platform that uses containerization to enable the creation and deployment of applications as portable, self-sufficient containers.

29.can you explain docker image?
Ans: A Docker image is a lightweight, standalone, executable package that contains everything needed to run an application, including the code, a runtime, libraries, and dependencies.

30. can you explain docker containers?
Ans: Docker containers are lightweight, standalone, and executable packages that contain all the necessary components to run a piece of software, including code, libraries, and system tools, in a consistent and isolated environment.

31.**how to create image using docker file?**
Ans: docker build -t <image name> .

32.**can, you write docker file for installation tomcat?**
Ans: FROM ubuntu
MAINTAINER thej
RUN apt update -y;apt install tomcat9 -y;apt install
tomcat9-admin -y
RUN apt install vim -y
RUN echo "<tomcat-users>\n\
\t<username user='thej' password='thej'
roles='manager-script'/>\n\
</tomcat-users>" > /etc/tomcat9/tomcat-users.xml
EXPOSE 8080

33.**differene between docker and virtual machines?**
Ans: Docker is a containerization platform that virtualizes the operating system at the application level, while virtual machines virtualize the entire operating system and hardware.

34.**what is docker engine?**
Ans: Docker Engine is a containerization platform that allows developers to package, deploy, and run applications in isolated environments called containers.

35.**explain architecture docker?**
Ans: Docker is a containerization platform that uses a client-server architecture to package and isolate applications with their dependencies into portable and lightweight containers.

36.**how to run containers?**
Ans: docker run –name <image name u wants> -d -P <image name>

37.**how to delete containers?**
Ans: docker rm -f <container name>

38.**how to see list of containers?**

Ans: Docker container ls

39.**how to add volume in dockers?**
Ans: docker run -v <host path>:<container path> image name

==========KUBERNETS==========

40.**differene between docker and Kubernetes?**
Ans: **Docker** is a containerization platform that allows developers to create, deploy, and run applications in isolated containers.
Kubernetes is a container orchestration platform that automates deployment, scaling, and management of containerized applications.

41.**what is orchestration?**
Ans: Orchestration refers to the automated arrangement, coordination, and management of complex systems or processes.

42.**what are the tools you used for orchestration?**
Ans: some of the tools I commonly used for orchestration are:

- Kubernetes for container orchestration
- Ansible for configuration management and deployment automation
- Jenkins for continuous integration and continuous deployment (CI/CD)
- Terraform for infrastructure as code (IaC) and provisioning.
- Docker for containerization
- Prometheus for monitoring and alerting
- Grafana for data visualization and analytics.

43.**explain Kubernetes architecture?**
Ans: Kubernetes architecture consists of a master node that manages worker nodes, which run containers, and uses etcd as a distributed key-value store to store cluster state information.

44.**what is Kubernetes scheduler?**
Ans: Kubernetes scheduler is a component of the Kubernetes control plane that assigns newly created Pods to available Nodes in the cluster based on resource requirements, constraints, and other policies.

45.**what is name space?**
Ans: A namespace is a container that holds a set of identifiers, such as variable names, function names, and class names, to prevent naming conflicts and provide organization to code.

46.What are pods?
Ans: In the context of containerization, pods are the smallest deployable units in Kubernetes that consist of one or more containers that share the same network namespace and storage volumes.

47. What is helm repository?
Ans: A Helm repository is a collection of packaged Helm charts that are stored in a centralized location, usually on a web server or in a version control system. It allows users to easily search, install, and update Helm charts for their Kubernetes applications.

48.**what is replica controller?**
Ans: A replica controller is a component of Kubernetes that ensures a specified number of replica pods are running and available to handle requests, and automatically creates or deletes pods as necessary to maintain the desired state.

49.**what are services and deployments?**
Ans: services are software components that provide specific functionality to other software or users. Deployments refer to the process of installing and running software applications on a server or computing infrastructure.

50.**differene between server and deployment?**
Ans: A server is a computer system or program that provides services to other computer programs or devices, while a deployment refers to the process of making software or applications available for use, typically on a server or other computing environment.

51.**explain Kubernetes networking?**
Ans: Kubernetes networking is the system of connecting, routing, and managing network traffic between containers, pods, and services within a Kubernetes cluster.

52. deployment process for Kubernetes?
Ans:

- Build Docker images of your application.
- Create a Kubernetes deployment configuration file.
- Deploy the application to the Kubernetes cluster using the kubectl command.
- Verify that the deployment was successful using kubectl commands.
- Expose the application using a Kubernetes service.
- Optionally, scale the deployment by increasing the number of replicas.

53. Did you deploy any applications in Kubernetes?
Ans: Yes, I have deployed several applications in Kubernetes. Some are Jenkins, MySQL, Nginx, Prometheus and Grafana.

54.**when some time deployment fails how to troubleshoot?**
Ans:
When deployment fails as a DevOps engineer, troubleshoot by:

- Checking the error logs and identifying the root cause
- Reviewing the deployment configuration and verifying if it is correct.
- Checking if the dependencies are correctly installed and up to date.
- Verifying the environment and ensuring it is correctly set up.
- Running diagnostic tests and verifying the system components are functional.
- Rolling back to the previous working version if necessary.

55.**if any cluster is not working, how to troubleshoot?**
Ans: As a DevOps engineer, to troubleshoot a non-working cluster, you can start by checking the logs, examining the network connections, verifying the cluster configuration, and testing individual components to identify the root cause of the problem.

==========ANSIBLE==========

56. **can you explain what purpose you require ansible?**
Ans: Ansible is used for automation, configuration management, and orchestration of IT infrastructure and applications across multiple servers and environments.

57.**what is ansible structure?**
Ans: Ansible is a radically simple IT automation engine that automates cloud

provisioning, configuration management, application deployment, intra-service orchestration, and many other IT needs.

58. What is an inventory file?
Ans: The Ansible inventory file defines the hosts and groups of hosts upon which commands, modules, and tasks in a playbook operate.

59. **Difference between static and dynamic inventory file?**
Ans: Static inventory is a fixed list of hosts while dynamic inventory is generated on-demand from external sources.

60. **how to deploy package remote server using ansible?**
Ans: Ensure that you have installed Ansible and set up the SSH Config for your remote systems, how to do that can be found under the post: setting up ansible.

- Our Inventory. The inventory file that describes our hosts: ...
- Playbook. ...
- Deploy.

61. **ansible role?**
Ans: An Ansible role is a reusable set of tasks, variables, templates, and files, designed to perform a specific function or configuration on a target system.

63. **did, you write any ansible playbook? explain you?**
Ans: Yes, An Ansible Playbook is a blueprint of automation tasks—which are complex IT actions executed with limited or no human involvement.

64. **in your project where are used ansible"?**
Ans: In my project I am using ansible to create a playbook for installing software in all the services.

65. If any knowledge in ansible tower?
Ans: No

66. **difference between chef and ansible?**
Ans: Chef and Ansible are both configuration management tools, but the main difference between them is that Chef uses a declarative approach while Ansible uses an imperative approach. Chef is designed to manage complex systems with many dependencies, while Ansible is designed to be simple and easy to learn.

67. **how to declare variables in ansible?**
Ans: In ansible playbook by passing the variables on the command line using the --extra-vars or -e argument.

68. **what is ansible galaxy command?**
Ans: The ansible-galaxy command comes bundled with Ansible, and you can use it to install roles from Galaxy or directly from a git based SCM. You can also use it to create a new role, remove roles, or perform tasks on the Galaxy website.

69. **how to execute ansible playbook?**
Ans: ansible-playbook <playbook.yml> -b

==========AWS==========

70.**what is service Iam?**
Ans: AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources for your users.

71.**what is policy?**
Ans: A policy is a set of rules, principles, or guidelines that guide decision-making and action in a particular area or organization.

72.**did, you written any custom policy?**
Ans: Yes, I have written custom policies to enforce security, compliance, and governance requirements across different stages of the software development lifecycle.

73.**what is mfa06?**
Ans: MFA stands for Multi-Factor Authentication, a security measure that requires multiple forms of authentication (such as a password and a fingerprint or a code sent to your phone) to access an account or system, making it more difficult for unauthorized individuals to gain access.

74.Why does policy require in aws?
Ans: Policies are required in AWS to define permissions and access control for AWS resources and services, ensuring security and compliance with organizational policies and regulations.

==========EC2==========

75.what is Ami?
Ans: AMI stands for Amazon Machine Image, which is a pre-configured virtual machine image used to create virtual machines within the Amazon Web Services (AWS) cloud. AMIs provide a quick and efficient way to launch new instances of a server environment with a specific operating system, software applications, and data.

76. How to mount the volume?
Ans:

77.**what is autoscaling?**
Ans: Autoscaling is a cloud computing feature that automatically adjusts the amount of computational resources allocated to a workload based on its changing demands, ensuring optimal performance and cost efficiency.

78.**what is load balancer?**
Ans: A load balancer is a device or software that distributes incoming network traffic across multiple servers to improve efficiency, reliability, and scalability of web applications or services.

79.**explain type of load balancer?**
Ans:

- Layer 4 Load Balancer: Distributes traffic based on network protocol data such as IP address and port number.
- Layer 7 Load Balancer: Distributes traffic based on application-specific data such as HTTP headers and cookies.
- Hardware Load Balancer: A physical device designed for high performance and reliability.

- Software Load Balancer: A virtualized solution running on a server or in the cloud.
- DNS Load Balancer: Distributes traffic based on DNS queries to different IP addresses.
- Global Load Balancer: Routes traffic across multiple data centres and geographic regions.
- Application Load Balancer: Optimized for modern application architectures and microservices.

80.**differene between alb, nlb and elb?**
Ans:

- ALB (Application Load Balancer) operates at the application layer (Layer 7) and routes traffic based on content of the request.
- NLB (Network Load Balancer) operates at the transport layer (Layer 4) and routes traffic based on IP protocol data.
- ELB (Elastic Load Balancer) is a legacy term used to refer to both Application Load Balancer and Network Load Balancer.

81.**what is elastic ip address?**
Ans: An Elastic IP address is a static, public IP address that can be associated with a cloud-based resource, such as an instance in Amazon Web Services (AWS), to provide a persistent and fixed point of access even after stopping and starting the instance.

82.**i created one ec2 server but when I am accessing the server is not working how to troubleshoot?**
Ans: Check if the server is running, make sure the security group allows incoming traffic on the appropriate port, check the network connectivity, and ensure that the server software is configured correctly.

83.What is user data in ec2 server?
Ans: User data in EC2 server is a script or data passed to the instance during launch that can be used to automate tasks or run custom configurations.

==========s3 bucket==========

84.What is a s3 bucket?
Ans: An S3 bucket is a scalable, secure, and durable object storage service provided by Amazon Web Services (AWS), which allows users to store and retrieve any amount of data from anywhere on the internet.

85.What purpose of we are using s3 bucket?
Ans: S3 bucket is used for storing and retrieving any type of data, including but not limited to, images, videos, documents, and application data, in a highly scalable and secure manner on the Amazon Web Services (AWS) cloud platform.

86. What are the limits of buckets for one account?
Ans: The limits of buckets for one account vary depending on the cloud storage provider, but typically include restrictions on the number of buckets that can be created, the maximum size of each bucket, and the maximum number of objects that can be stored in each bucket.

87.**what is bucket policy?**
Ans: A bucket policy is a configuration option in Amazon Web Services (AWS) that allows you to control access to your AWS S3 bucket by specifying who can

access it and what actions they can perform.

88.**why s3 bucket versioning is required?**
Ans: S3 bucket versioning is required to maintain a history of object versions in the bucket, allowing for easy retrieval of previous versions, protection against accidental deletions, and compliance with regulatory requirements.

89.**what s3 lifecycle roles?**
Ans: S3 lifecycle roles are rules that define when and how objects in an S3 bucket should be transitioned to different storage classes or deleted based on their age or usage patterns.

90.**What is disaster recovery?**
Ans: Disaster recovery refers to the process of restoring critical systems and operations after an unexpected event or disaster, such as a natural disaster or cyber-attack, has disrupted or damaged them.

==========VPC==========

91.purpose of vpc?
Ans: A Virtual Private Cloud (VPC) is a cloud-based networking service that allows users to create and control their own isolated virtual network environment in the cloud, providing secure and scalable resources for running applications and services.

92.**difference between public ip and private ip?**
Ans: A public IP is a unique address assigned by the internet service provider to a device accessible over the internet, while a private IP is an address assigned to a device on a private network that is not accessible from the internet.

93.**what is NAT gate?**
Ans: NAT (Network Address Translation) gateway is a device or software that allows multiple devices in a local network to share a single public IP address when accessing the internet.

94.**What is IGW?**
Ans: IGW stands for Internet Gateway, and it is a networking component that enables communication between resources in a private network and the internet.

95.**What is NACL?**
Ans: NACL stands for "Network Access Control List" and it is a security feature used in computer networks to control traffic by allowing or denying access to specific IP addresses or protocols.

96.**difference between nacl and security groups?**
Ans: NACLs are stateless and operate at the subnet level, while security groups are stateful and operate at the instance level.

97.**what is vpc peering?**
Ans: VPC peering is a networking connection between two Virtual Private Clouds (VPCs) that allows them to communicate with each other using private IP addresses.

98.**What is CIDR?**

Ans: CIDR stands for Classless Inter-Domain Routing, which is a method for allocating IP addresses and routing Internet traffic. It allows for more flexible use of IP address space and more efficient routing of traffic on the Internet.

==========EKS==========

99. **Did you work eks?**
Ans: Yes
100.**How to create cluster?**
Ans: eksctl create cluster –name <Cluster Name> --region <region id> --nodes-min <no. of instances>

101.**how add role in eks cluster?**
Ans: aws eks update-kubeconfig --name <cluster name> && kubectl create -f <role_file.yaml>

102.**purpose ecr?**
Ans: ECR stands for Elastic Container Registry, and its purpose is to store, manage, and deploy Docker container images, allowing for easy and scalable containerization of applications on the Amazon Web Services (AWS) platform.

103. **difference between ecs and eks?**
Ans: ECS (Elastic Container Service) is a fully managed container orchestration service by AWS, while EKS (Elastic Kubernetes Service) is a fully managed Kubernetes container orchestration service also by AWS.

ECS uses its own proprietary container orchestration system, while EKS uses Kubernetes, an open-source container orchestration system.

==========TERRAFORM==========

104.**What is a terraform state file?**
Ans: Terraform stores information about your infrastructure in a state file. This state file keeps track of resources created by your configuration and maps them to real-world resources.

105.**in your project where store the state file?**
Ans: The location of the state file is dependent on the project's structure and requirements, and can be stored in a dedicated directory, cloud-based storage service, database, or other persistent storage solution.

106.**how to call state file from main.tf?**
Ans: We can call a state file from main.tf using the "terraform" block and the "backend" configuration.

107.**what is terraform init, plan, apply?**
Ans:

- **terraform init:** initializes a new or existing Terraform working directory by downloading and configuring the necessary providers and modules.
- **terraform plan** creates an execution plan that describes what changes Terraform will make to the infrastructure to achieve the desired state as defined in the Terraform configuration files.
- **terraform apply** executes the changes described in the execution plan and provisions the infrastructure to match the desired state as defined

in the Terraform configuration files.

108.**what type of terraform command did you work with?**
Ans: Terraform init, Terraform plan, Terraform Apply, Terraform Validate, Terraform Destroy, Terraform Output and Terraform State.

109.**what version did you use?**
Ans: V 1.3.9

110.**how to switch one terraform version to another terraform version?**
Ans: We can switch between different versions of terraform by typing the command tf switch on your terminal.

111.**what are terraform modules?**
Ans: Terraform modules are reusable units of infrastructure code that encapsulate related resources and configurations, allowing for easier management, sharing, and collaboration of infrastructure code.

112.**did, you write any terraform modules, explain me?**
Ans: A Terraform module is a reusable set of Terraform configuration files that define and provision a specific infrastructure component or resource in a cloud provider.

113.**did, you integrate terraform with ci/cd pipeline?**
Ans: Terraform can be integrated with a CI/CD pipeline by using tools like Jenkins, GitLab, or Azure DevOps to automatically apply infrastructure changes made through Terraform code.

114.**what does terraform require?**
Ans: Terraform requires a configuration file, a provider, and an execution plan to create and manage infrastructure as code.

115.**differene between terraform and ansible?**
Ans: Terraform is an infrastructure-as-code tool used to provision and manage cloud resources, while Ansible is a configuration management tool used to automate software deployment and system configuration.

116.**what are uses of terraform functions?**
Ans: Terraform functions can be used for string manipulation, type conversion, conditional logic, mathematical calculations, and more, making infrastructure management more efficient and easier to maintain.

117.**explain what type terraform scripts you worked on?**
Ans: Terraform is a tool used for infrastructure as code, which allows you to write scripts to provision and manage infrastructure resources in cloud providers like AWS, Azure, and Google Cloud Platform.

==========Monitoring==========

118.**did, you implement monitoring in cloud watch?**
Ans: CloudWatch is a monitoring service that allows you to collect and track metrics, collect, and monitor log files, and set alarms.

119.**explain elk setup?**
Ans: ELK (Elasticsearch, Logstash, Kibana) is a popular open-source stack used for centralized logging and analysis of data in real-time.

- Elasticsearch is a distributed search and analytics engine that

stores and indexes data.

- Logstash is a data processing pipeline that ingests, transforms, and sends data to Elasticsearch.
- Kibana is a web-based visualization tool used to analyse and visualize the data stored in Elasticsearch.

120.**difference between file beat and metric beat?**
Ans: File beat is used for collecting and shipping log data, while Metric beat is used for collecting and shipping system and application metrics data.

121.**explain aws cloud watch services?**
Ans: AWS CloudWatch is a monitoring and observability service that collects and tracks metrics, logs, and events from AWS resources and applications, allowing users to gain insights into their system's performance, troubleshoot issues, and take automated actions.

122.**how to monitor ec2 instance using cloud watch?**
Ans: To monitor an EC2 instance using CloudWatch:

- Create an IAM role with necessary permissions.
- Install CloudWatch agent on EC2 instance.
- Configure the agent and start it.
- View the instance metrics in CloudWatch console or set up alarms for notifications.

==========end==========

\