**DevOps Questions**

**AWS**

1. AWS services used in DevOps.

Ans : (AWS) offers many services for DevOps, including Amazon EC2, Amazon S3, AWS Lambda, and more. These services can help with building, testing, deploying, and monitoring applications.

2. What happens if we forget the public key of an EC2 instance?

If you lose the public key of an Amazon EC2 instance, you can create a new key pair and copy the public key to regain access to the instance.

3. What are security groups?

Security groups are virtual firewalls that control traffic to and from resources like EC2 instances and domains

4.. What is RDS?

Amazon Relational Database Service (RDS) is a database service that allows users to create, manage, and scale relational databases in the cloud

5. How to connect EC2 instances to the local machine

You can connect to an Amazon EC2 instance from your local machine using SSH or the EC2 console

6. What is the difference between a public and private subnet?

A public subnet having a route to an internet gateway and a private subnet not having one.

7. S3 bucket globally rename it or not?

No, you cannot directly rename an S3 bucket globally; once created, an S3 bucket name is immutable, meaning you cannot change it,

8. How to access the internet in a private subnet?

To access the internet from a private subnet, you need to use a Network Address Translation (NAT) gateway

9. What are default settings of security groups?

The default security group settings for AWS and IBM Cloud deny all inbound traffic and allow all outbound traffic.

10. Elastic Load Balancer (ELB): Working methodology.

An Elastic Load Balancer (ELB) works by acting as a single entry point for incoming traffic, distributing it across multiple target servers (like EC2 instances) in a balanced manner, ensuring high availability and optimal performance by monitoring the health of each target and only routing traffic to healthy ones

11. Auto Scaling Groups (ASG) uses.

An Auto Scaling Group (ASG) is primarily used to automatically manage and scale the number of EC2 instances within a group based on predefined criteria

12. EBS: How to mount volume in instance.

To mount an EBS volume on an EC2 instance, you need to first attach the volume to the instance using the AWS console or CLI, then identify the device name assigned to the volume, create a mount point directory on the instance, and finally use the mount command to mount the volume to that directory; if the volume is new, you'll need to format it with a filesystem first.

13. Types of storage in AWS.

S3, EBS, EFS, **AWS Storage Gateway**

14. CloudWatch for EC2.

Amazon CloudWatch for EC2 is a monitoring service that tracks and collects metrics for Amazon Elastic Compute Cloud (EC2) instances. It also monitors applications that run on EC2.

15. CloudFront.

Amazon CloudFront is a content delivery network (CDN) that speeds up how content is delivered to users. It's offered by Amazon Web Services (AWS).

16. Edge Locations.

Edge locations are data centers that are located near users and are used to deliver content and services. They are also known as mini data centers.

17. How do you monitor Amazon VPC?

You can monitor Amazon VPC using tools like VPC Flow Logs, Amazon CloudWatch, and Traffic Mirroring. You can also monitor subnets and network interfaces

18. What is an Elastic IP?

An Elastic IP address (EIP) is a static public IP address that can be associated with an instance or network interface in a virtual private cloud (VPC). EIPs are used for network address translation (NAT).

19. What is a Terraform State file?

A Terraform state file is a JSON file that tracks the resources and configuration of your infrastructure. It's a key part of Terraform's functionality.

20. Security groups vs NACLs.

Security groups and network access control lists (NACLs) are both used to control access to AWS resources. Security groups are used at the instance level, while NACLs are used at the subnet level.

21. VPC components.

The components of an Amazon Virtual Private Cloud (VPC) include subnets, VPC endpoints, VPC peering, network access control lists (ACLs), internet gateways, NAT gateways, and Transit Gateways.

22. How do you launch instances in AWS EC2?

To launch an instance in AWS EC2, you can use the EC2 launch instance wizard. You can also use the AWS CLI to launch an instance.

23. What is EC2?

Amazon Elastic Compute Cloud (EC2) is a web service that lets users build and run applications in the cloud. It provides virtual servers, or EC2 instances, that can be resized.

24. How do you manage users and permissions in AWS?

You can use AWS Identity and Access Management (IAM) to manage users and permissions in AWS. You can use IAM to create users, groups, roles, and policies, and assign them permissions.

25. What is Amazon EKS?

Amazon Elastic Kubernetes Service (EKS) is a managed service that allows users to run Kubernetes on AWS and on-premises. Kubernetes is an open-source system that automates the management of containerized applications.

26. What is IAM?

AWS Identity and Access Management (IAM) is a web service that controls access to AWS resources. It helps you manage who can use AWS services and resources, and under what conditions.

27. What is AMI?

An Amazon Machine Image (AMI) is a template that is used to create virtual servers, also known as Amazon Elastic Compute Cloud (EC2) instances, in Amazon Web Services (AWS)

28. How can you provide permission in IAM?

You can grant permissions in IAM by attaching policies to users, roles, or groups.

29. What is ELB?

Elastic Load Balancing (ELB) is a service that automatically distributes incoming traffic to multiple targets. It's offered by Amazon Web Services (AWS)

30. What are the main services in AWS?

* Amazon EC2 (Elastic Compute Cloud) EC2 is a cloud platform provided by Amazon that offers secure, and resizable compute capacity. ...
* Amazon RDS (Relational Database Services) ...
* Bonus Service: Amazon Connect. ...
* Amazon S3 (Simple Storage Service) ...
* Amazon Lambda. ...
* Amazon Cognito. ...
* Amazon Glacier. ...
* Amazon Lightsail.

31. Relational database in AWS.

A managed database service that makes it easy to set up and run a relational database in the AWS cloud

Supports a variety of database engines

Works with Oracle and Microsoft SQL Server

32. EC2 instance auto-scaling.

The dynamic scaling capabilities of Amazon EC2 Auto Scaling refers to the functionality that automatically increases or decreases capacity based on load or other metrics. For example, if your CPU spikes above 80% (and you have an alarm setup) Amazon EC2 Auto Scaling can add a new instance dynamically.

33. Explain VPC and its components like Public & Private subnets, NAT, IGW, and security groups.

A Virtual Private Cloud (VPC) is a logically isolated virtual network within a public cloud, essentially creating a private network space with customizable configurations for your cloud resources; key components of a VPC include public and private subnets, a NAT (Network Address Translation) gateway, an internet gateway (GW), and security groups which act as virtual firewalls to control network traffic in and out of your instances within the VPC.

34. Explain EC2 and RDS backups and how they work.

In AWS, EC2 backups refer to creating snapshots of Elastic Block Storage (EBS) volumes attached to EC2 instances, essentially backing up the data on your virtual servers, while RDS backups are automated snapshots of your entire relational database instance within the Amazon Relational Database Service (RDS), managed by AWS and designed specifically for database data protection;

35. What is S3?

Amazon S3, which stands for "Amazon Simple Storage Service," is a cloud-based object storage service offered by Amazon Web Services (AWS) that allows users to store and retrieve any amount of data, like files, images, and videos, from anywhere at any time, providing high scalability, data availability, and security with features like different storage classes to optimize costs based on access patterns and fine-tuned access controls for managing data permissions; essentially acting as a digital "bucket" where you can store and organize your data in the cloud.

36. S3 versioning.

Amazon S3 Versioning is a feature that allows you to keep multiple versions of an object in the same bucket. This helps protect your data from accidental deletions or overwrites.

37. How to access S3 and RDS from an EC2 instance?

To access S3 and RDS from an EC2 instance, you need to create an IAM role with the appropriate permissions for both services, attach that role to your EC2 instance, and ensure your security groups allow the necessary network connections, effectively granting your EC2 instance access to both S3 and RDS by using the AWS CLI or SDK within your application running on the EC2 instance.

38. What is EBS?

EBS could refer to Amazon Elastic Block Store,  is an easy-to-use, scalable, high-performance block-storage service designed for Amazon Elastic Compute Cloud (Amazon EC2).

39. What is EFS?

Amazon Elastic File System (EFS) is designed to provide serverless, fully elastic file storage that lets you share file data without provisioning or managing storage capacity and performance

40. How to configure VPC in Terraform?

**With this as the background, let us start building this VPC design from scratch.**

1. Step 1: Create a VPC. ...
2. Step 2: Create Subnets. ...
3. Step 3: Set up Internet Gateway. ...
4. Step 4: Create a Second Route Table. ...
5. Step 5: Associate Public Subnets with the Second Route Table.

41. AWS services (EC2, RDS, EFS, EBS, Load Balancer, IGW, NAT Gateway).

In AWS, EC2 provides virtual servers, RDS manages relational databases, EFS is a file system service, EBS offers block storage for EC2 instances, Load Balancer distributes traffic across multiple servers, IGW (Internet Gateway) allows access to the public internet from a VPC, and a NAT Gateway enables instances in a private subnet to access the internet through a public IP address.

42. T2 micro.

The t2.micro is a type of Amazon EC2 instance that provides a baseline level of CPU performance with the ability to burst to higher performance when needed. It's part of the t2 series of general purpose instances.

43. Difference between on-prem and cloud.

On-premises (on-prem) refers to hardware and software that's owned and operated by a company, while cloud computing refers to services that are hosted by a third party

44. How to migrate apps and databases from on-prem to cloud.

To migrate applications and databases from on-premise to the cloud, you need to follow a structured process involving assessment, choosing a migration strategy (like "lift and shift" or re-architecting), preparing your data, migrating the applications and databases, and then validating and optimizing the new cloud environment;

45. AWS Elastic Search.

Elasticsearch is a distributed search and analytics engine built on Apache Lucene. Since its release in 2010, Elasticsearch has quickly become the most popular search engine and is commonly used for log analytics, full-text search, security intelligence, business analytics, and operational intelligence use cases

46. AWS IAM roles.

AWS Identity and Access Management (IAM) roles are entities that allow users to access AWS resources with temporary credentials. You can create and assign permissions to these roles.

47. Amazon EBS-backed vs. instance-store backed instances.

An Amazon EBS-backed instance uses a persistent storage volume from Amazon Elastic Block Store (EBS) as its root disk, meaning data is saved even after the instance is stopped, while an instance-store backed instance uses temporary storage directly attached to the host server, which means data is lost when the instance is terminated

48. AWS EC2 Instance types.

**Instance Types**

* General purpose.
* Compute optimized.
* Memory optimized.
* Storage optimized.
* Accelerated computing.
* High-performance computing.
* Previous generation.

49. Cross-region S3.

With cross-region replication, every object uploaded to an S3 bucket is automatically replicated to a destination bucket in a different AWS region that you choose. For example, you can use cross-region replication to provide lower-latency data access in different geographic regions.

50. NAT gateway purpose.

A NAT gateway's primary purpose is to enable instances within a private subnet to access the internet or other external services while preventing external services from initiating connections with those instances,

51. The pem key is lost, and I need to connect via SSH.

f you've lost your PEM key, the only way to regain SSH access to your server is to create a new key pair and add the public key to your server's authorized\_keys file

52. Connect to a private subnet via SSH from a local machine.

To connect to a private subnet via SSH from a local machine, you typically need to use a "bastion host," which is a server in a public subnet that acts as a gateway to access private instances

53. Connectivity between two VPCs in private subnets.

Connectivity between two VPCs in private subnets is established through a "VPC peering connection

54. AutoScaling in AWS why?

AWS Auto Scaling can help you optimize your utilization and cost efficiencies when consuming AWS services so you only pay for the resources you actually need

55. What is meant by IAM server?

An "IAM server" refers to a dedicated server or system that manages "Identity and Access Management" (IAM), which essentially means it controls who has access to what resources within a network or system, by verifying user identities and assigning appropriate permissions based on their roles and access levels.

56. What do you mean by AMI in a server?

In server terminology, "AMI" stands for "Amazon Machine Image," which is essentially a template or snapshot of a virtual server configuration including the operating system and pre-installed software, used to launch new virtual machines on Amazon Web Services (AWS) Elastic Compute Cloud (EC2) platform; essentially, it's like a blueprint for creating identical server instances with specific settings.

**Linux**

1. Linux basic commands.

* **mkdir**: Creates directories and sets permissions for them
* **rm**: Deletes files or directories
* **cat**: Lists the contents of a file
* **chmod**: Changes the permissions of a file
* **grep**: Searches for text patterns in files or output
* **touch**: Creates an empty file or updates the timestamp of an existing file
* **pwd**: Displays the full path of the current working directory
* **rmdir**: Removes a directory
* **df**: Displays information about file system disk space usage
* **sudo**: Allows a system administrator to give certain users the ability to run commands as root
* **chown**: Changes the owner of files and directories
* **cd**: Changes the current working directory
* **ls**: Lists directory contents, including files and subdirectories
* **mv**: Moves or renames files or directories
* **cp**: Copies files or directories

1. What is Bash?

Bash is a command-line interface (CLI) and scripting language that allows users to interact with their computer's operating system.

1. Disk management.

Disk Management is a Windows tool that allows users to perform advanced storage tasks

1. Bash scripting.

Bash scripting is a way to automate tasks in the Unix shell. For instance, it can be used to automate system maintenance tasks, like cleaning up log files or backing up data, which helps reduce manual errors and save time

5. How to run Bash scripts.

6. Linux distributions.

7. Difference between top and nice commands.

8. Command to check server load.

9. How to rename a file in Linux?

10. Basic Linux commands.

11. Linux troubleshooting commands.

12. Commands for disk, memory, and CPU usage.

13. Linux patching.

14. How to add a user to your system?

15. Authentication methods used by SSH.

16. Operating system-related questions.

17. About Linux.

18. Write and explain all Linux commands you know.

19. Linux commands for disk, memory, CPU usage, and grep.

20. IP address and hostname change on Linux (by command line).

21. Configuration of DNS on Linux.

22. Public and Private DNS.

23. Command executed when a user logs into Linux.

24. How to check IP address in Linux.

25. How to log in to an Ubuntu machine.

26. Steps to launch an EC2 instance (using Ubuntu).

27. Can we connect to an EC2 server without using a key pair?

28. rm -i command in Linux.

29. How to save and exit a file in the Vi editor.

30. Can you name some commands in Linux?

31. SCP & SSH differentiate?

**Git**

1. Robert John is using Git for his work. What is the most effective approach to restructure his commit history when he has multiple commits for a single task?

2. What happens when you run the following command on the Git repository? git reset --soft HEAD^

3. Which Git command can configure the use of vimdiff as the default Git merge tool?

4. How does Git merge handle whitespace differences between versions?

5. Difference between centralized vs. distributed VCS.

6. What is the difference between Git pull and Git fetch?

7. Git rebase vs merge.

8. Command used to create a branch.

9. Differentiates between Git remote and Git clone.

10. Branching strategies.

11. Types of version controls.

12. Basic Git commands.

13. Command to rename a directory or a file.

14. How to checkout code from GitHub?

15. Difference between Git merge and Git rebase.

16. Difference between Git pull and Git fetch.

17. What is a merge conflict in Git and how do you resolve it?

18. Command to create a new branch in Git.

19. Command to checkout from one branch to another in Git.

20. Git vs. GitHub differences.

21. Git commands.

22. User access management in Git.

23. Checkout command in Git.

24. Purpose of the git rebase command.

25. Difference between git pull and fetch.

26. What are pull request strategies?

27. What are all git commands? What all the things you performed in your project?

28. What is .gitignore file?

29. How do u configure git in Jenkins?

**Docker**

1. Commands of Docker.

2. What is Docker Hub?

3. Dockerfile: How will you write it?

4. Difference between Docker image and container.

5. Where to use RUN and EXEC commands in Docker?

6. Docker volume creation and mounting.

7. Docker log.

8. Execute a command in Docker container (docker exec -it ...).

9. Run vs CMD in Dockerfile.

10. What is Docker containerization?

11. Difference between CMD and ENTRYPOINT in Dockerfile.

12. What is Docker?

13. What is a Dockerfile?

14. Difference between CMD and RUN in Dockerfile.

15. Difference between CMD and ENTRYPOINT in Dockerfile.

16. What is ARG in Dockerfile?

17. How to build an image in Docker?

18. Docker scenario: Updating an application in a container without disrupting others.

19. Docker networks.

20. Docker compose.

21. How to stop a Docker container?

22. What is Docker?

23. Why use Docker instead of directly providing code to the client?

24. Steps to do containerization.

25. Docker networking.

26. Docker file and its use.

27. Commands for containers (list, start, stop, remove).

28. How to run a Docker container.

29. Docker commands for creating containers and volume mapping.

30. Deployment file for Docker.

31. Write a Dockerfile for nginx.

32. What is the logging driver in Docker?

33. Docker-compose file to build an image.

34. Docker up, Docker run, Docker start in detail.

35. List Docker commands?

36. What is meant by Dockerfile?

**Kubernetes**

1. Difference between Kubernetes and Docker.

2. What is a replication set?

3. Kubernetes structure.

4. Networking in Kubernetes.

5. Replica set.

6. How to get a pod?

7. How to launch a pod in Kubernetes?

8. What is Kubernetes architecture?

9. Kubernetes scaling deployment.

10. Deployment strategies in Kubernetes.

11. What is node affinity and pod affinity?

12. What is the default Kubernetes network?

13. Commands in Kubernetes.

14. How to kill Kubernetes pods?

15. How to deploy a microservice in Kubernetes?

16. Kubernetes fmt command.

17. How to handle replicas and crash issues?

18. StatefulSet with PV.

19. What is Ingress? Which Ingress are you working on?

20. What is Kubernetes, and why use it?

21. Kubernetes architecture (control plane, worker, and master nodes).

22. Different Kubernetes strategies.

23. Kubernetes stateful set and daemon set.

24. Deployment vs. StatefulSet.

25. Commands for Kubernetes (create deployment, pods, services).

26. Service types in Kubernetes.

27. How to deploy MySQL DB in a Kubernetes cluster.

28. PV and PVC in Kubernetes.

29. Kubernetes manifest files.

30. How to resolve pod issues.

31. How to connect two Kubernetes nodes.

32. Explain Kubernetes architecture.

33. How do u access the pod through browser?

34. How do u check logs of pod?

**Jenkins**

1. How to manage credentials in Jenkins.

2. What plugins are used in Jenkins?

3. Jenkins pipeline.

4. Multi-pipeline in Jenkins.

5. Working of Jenkins.

6. Jenkins workspace.

7. Jenkins default port number?

8. Jenkins port number?

9. How do you backup Jenkins?

10. How can you integrate monitoring tools in Jenkins?

11. How will you configure Kubernetes in Jenkins?

12. What is a node in Jenkins?

13. Write a Groovy script for the Jenkins pipeline and explain the stages.

14. What are Jenkins environment variables?

15. What are the types of jobs in Jenkins?

16. How to secure Jenkins?

17. Where are Jenkins files and source code stored?

18. Master and slave in Jenkins.

19. Run deployment on a specific slave node.

20. Jenkins server backup.

21. How to get Jenkins password back if lost?

22. How to take Jenkins Backup plugin?

23. What will u do when build is failing in Jenkins?

**CI/CD Pipeline**

1. Describe CI/CD Pipeline and how it works.

2. How will you integrate with Maven?

3. Maven commands.

4. What is a CI/CD deployment flow?

5. Tools integrated into Jenkins.

6. Have you written any Jenkins pipelines? Explain each step.

7. Jenkins job migration between servers.

8. Dockerfile and Jenkins pipeline integration.

9. Observability/monitoring in production (e.g., CloudWatch).

**Terraform**

1. What is Terraform, and what does it provide?

2. Terraform components.

3. Terraform plan vs Terraform apply.

4. Stages in Terraform.

5. How to configure a server using Terraform?

6. How to rename resources in Terraform?

7. How does Terraform trigger changes after editing or updating resources?

8. What is a Terraform state file?

9. What is Terraform?

10. Use of the provider in Terraform.

11. Difference between Ansible and Terraform.

12. Terraform commands.

13. Terraform modules.

14. Terraform state file.

15. Terraform code for an S3 bucket.

16. Integration of Jenkins with Terraform for managing AWS infrastructure.

**DevOps**

1. Why DevOps? How does it help in IT?

2. What is Prometheus and Grafana?

3. Prometheus and Grafana log case study.

4. Default port number of Prometheus and Grafana.

5. How Prometheus and Grafana work.

6. What is ECS?

7. What do you mean by DevOps?

8. What are the tools in DevOps?

9. What do you mean by CICD?

**Testing**

1. What is testNG and its annotations?

2. New feature added: What type of testing will you do?

3. Difference between verification and validation.

4. What is the defect lifecycle?

5. Examples of high severity and low priority bugs.

**Miscellaneous**

1. What is Ansible?

2. Ansible roles and playbooks.

3. Inventory file in Ansible.

4. Define Chef.

5. What is the Terraform state file?

6. How to change the username to another username in AWS?

**General DevOps**

● What is Terraform?

● What do you know about DevOps?

● What are the popular tools in DevOps?

● Explain the approach of DevOps.

● What challenges you faced in DevOps and how did you overcome them?

● What are CI/CD tools?

● How to handle secrets?

● Steps in Jenkins pipeline to trigger Kubernetes deployment.

● What are the stages of Jenkins?

● Define Maven.

● Explain troubleshooting performance for production systems.

**Monitoring and Logging**

● Use of Prometheus.

● How to connect Grafana with Prometheus?

● Creation of dashboards in Grafana.

● SonarQube integration with Git and Jenkins.

● How can you check application logs, memory, and CPU utilization?

**Scripting and Configuration Management**

● Playbook format (YAML or JSON).

● What is Ansible?

● How to use Ansible Vault?

● Integration process in Ansible.

● Explain bash scripting for automation.

**Case Scenarios**

● Deploy microservices with RDS as a dependency.

● Handle downtime in applications.

● How to deploy a microservice with port as an environment variable?

● How to resolve a situation with two failed nodes out of five in Kubernetes?

● If a pod restarts, how do you handle the issue?

**Ansible**

● About Ansible.

● Ansible theory questions (e.g., how instances are attached to each other).

● Difference between Ansible and Terraform.

**Python/Shell Scripting**

● Write a shell script/python script to send an alert via email when disk usage exceeds 75%.

● Write a bash script.

● Python features.

● What is a shell script?

**Maven**

● Maven lifecycle.

● What happens in mvn clean package command?

● What is a Maven build profile?

● POM stands for?

**Networking**

● Process of searching something on the internet.

● What is DNS and how does it work?

● Difference between DNS and CNAME.

● Difference between proxy, reverse proxy, and load balancer.

● Port numbers (HTTP, HTTPS, DNS, MySQL, FTP).

● Public vs. Private IP address.

● How private IP is mapped to a public IP.

● Subnets (e.g., /16, /24, /32 notation)

**Monitoring Tools**

● What are Prometheus and Grafana? How do they work?

**Other Questions**

● Explain project details.

● Explain a complex problem or difficulty and its resolution.

● Responsibilities in your current organization.

● Knowledge of web applications.

● HTTP vs. HTTPS.

● HTTP request types (GET, POST, PUT).

● What is auto-scaling?

● What is crontab?

● How to migrate app and DB from on-prem to cloud.

**General/Project-Based**

● Explain about your project.