**✅ CI/CD Pipeline using Jenkins**

**Q: Can you explain your Jenkins CI/CD pipeline project?**

**Answer:**

Yes. In this project, I worked on a **Todo application made with Node.js**.  
The goal was to make the **deployment automatic** — so that whenever the code changes, it should go live automatically.

I used **Jenkins** for this. I connected it to **GitHub using webhooks**. When we push new code, Jenkins **builds the Docker image**, **runs tests**, and **deploys it to AWS EC2**.  
This saved time, and we didn’t need to deploy manually every time. It also made sure the app runs the same way on every machine using Docker.

**✅ Terraform Infrastructure Project**

**Q: Tell me about your Terraform project.**

**Answer:**

This project was about **creating AWS resources automatically** like EC2, S3, DynamoDB using **Terraform**.

I made **reusable modules** so we don’t write the same code again and again.  
I also managed **different environments** like Dev, QA, and Production separately.  
With one command, we could **create or destroy** all resources easily.

It helped to save time and avoid mistakes that happen when we do things manually.

**✅ Internship at BISAG (Bhaskaracharya Institute)**

**Q: What did you do during this internship?**

**Answer:**

In this internship, I worked on **AWS cloud services**.

I created **Linux EC2 instances**, added **Elastic IPs** to access them easily. I also added **EBS volumes for storage** and created **snapshots** for backup.

I managed **security groups** to allow only specific traffic. I also set up **CloudWatch alarms** with **SNS** to send alerts if something goes wrong.

This helped me learn how to manage and secure AWS servers properly.

**✅ Internship at Brainwave Matrix Solutions**

**Q: What was your role in this remote internship?**

**Answer:**

Here, I worked on **deploying web applications on AWS**.

I used **EC2, S3, and RDS** to make the app run smoothly and securely. I also worked with **IAM roles** and **security groups** to control access.

Apart from that, I built a **real-time chat app** using **Firebase Auth and Firestore**. I hosted it using **Firebase Hosting**.

This internship helped me learn both AWS and Firebase cloud platforms.

**1. What is cloud computing? How is it different from traditional on-premise computing?**  
Cloud computing means using servers, storage, or applications over the internet. In traditional computing, you manage everything yourself on your own computers. With cloud, you rent what you need and use it anytime from anywhere.

**2. What are the main service models in cloud computing (IaaS, PaaS, SaaS)?**

* **IaaS** – Infrastructure as a Service (e.g., EC2): gives virtual servers.
* **PaaS** – Platform as a Service (e.g., Elastic Beanstalk): gives a ready platform to run apps.
* **SaaS** – Software as a Service (e.g., Gmail): ready-made apps to use directly.

**3. Difference between public, private, and hybrid cloud?**

* **Public Cloud**: Shared by many users (like AWS).
* **Private Cloud**: Only for one company.
* **Hybrid Cloud**: Mix of both public and private.

**☁️ AWS-Specific Questions**

**4. What is an EC2 instance? How do you launch and connect to it?**  
EC2 is a virtual server in AWS. I launch it from the AWS Console by choosing OS, size, and key pair. I connect using SSH from the terminal.

**5. How do you attach an EBS volume to EC2? Purpose?**  
EBS is a storage disk. I attach it in the EC2 settings. It's used to store files even after the instance is stopped.

**6. Difference between EBS and S3?**

* **EBS**: Like a hard disk for EC2.
* **S3**: Object storage for any kind of files (images, backups, etc.).

**7. What is an Elastic IP? Why did you use it?**  
It’s a static public IP. I used it so the EC2 instance keeps the same IP even if restarted.

**8. How do IAM roles and policies work?**  
IAM roles give permissions to users or services. Policies define what actions they can do (like read/write on S3).

**9. What are security groups vs NACLs?**

* **Security Groups**: Work at instance level (virtual firewall).
* **NACLs**: Work at subnet level and support rules for allow and deny.

**10. How does S3 ensure durability and availability?**  
S3 stores data in multiple places. Even if one server fails, data is safe and always available.

**11. What is RDS? When to use it?**  
RDS is a managed database service. We use it when we want to run databases like MySQL without managing servers.

**12. What is a VPC and its components?**  
VPC is a private network in AWS. It includes subnets, route tables, internet gateway, NAT gateway, and NACLs.

**13. How does CloudWatch work? How did you use it?**  
CloudWatch monitors AWS resources. I used it to create alarms (like high CPU usage) and send alerts using SNS.

**14. Difference between horizontal and vertical scaling?**

* **Horizontal**: Add more machines.
* **Vertical**: Increase size of one machine (more CPU/RAM).

**15. How to configure a custom VPC with private and public subnets?**  
Create a VPC, add subnets. Public subnet gets internet gateway. Private subnet uses NAT gateway to access internet securely.

**16. How do snapshots help in disaster recovery?**  
Snapshots are backups of volumes. If data is lost, we can restore it using snapshots.

**🔧 DevOps Tools & Infrastructure**

**17. What is Jenkins? How did you use it?**  
Jenkins is a tool for automation. I used it to create a CI/CD pipeline to deploy my Node.js app automatically.

**18. What are webhooks in Jenkins?**  
Webhooks send messages when code changes on GitHub. Jenkins gets this and starts the pipeline automatically.

**19. How does Docker help in deployment?**  
Docker puts apps in containers. It works the same everywhere and avoids environment problems.

**20. What is Terraform? Why did you use it?**  
Terraform is a tool to create cloud resources using code. I used it to create EC2, S3, and DynamoDB quickly.

**21. What are Terraform modules?**  
Modules are reusable code blocks in Terraform. I used them to avoid repeating the same code for multiple environments.

**22. Use of terraform destroy?**  
It deletes all cloud resources created by Terraform. Useful for cleanup.

**23. How to manage multiple environments in Terraform?**  
I used different variables and workspaces to handle Dev, QA, and Production environments separately.

**24. Role of Git in DevOps?**  
Git stores and tracks code changes. Teams can work together without confusion.

**25. How do you monitor app health in pipeline?**  
I used CloudWatch and logs to check if apps are working. I also set alerts for failures.

**🧠 Networking Fundamentals**

**26. Difference between public and private IPs?**

* **Public IP**: Can be accessed from internet.
* **Private IP**: Only for inside the VPC.

**27. What is DNS and how does it work?**  
DNS changes domain names (like google.com) to IP addresses. Helps users connect to websites easily.

**28. How does NAT help in networking?**  
NAT allows private instances to access the internet without having public IPs.

**29. What is DHCP?**  
DHCP gives IP addresses to devices in a network automatically.

**30. Difference between TCP and UDP?**

* **TCP**: Reliable and slow (e.g., emails).
* **UDP**: Fast but not reliable (e.g., video calls).

**31. What is a firewall? How did you use it in AWS?**  
A firewall controls traffic. In AWS, I used security groups to allow only required ports like 22 (SSH) or 80 (HTTP).

**32. What is a default gateway?**  
It’s a device/router that sends traffic from inside the network to the internet.

**👩‍💻 Internship & Project-Based**

**33. In Brainwave Matrix project, how did you ensure secure access?**  
I used IAM roles and security groups. Only allowed required ports and access to specific users.

**34. How did the real-time chat app work?**  
Frontend was built using HTML and JS. Backend was Firebase. Used Firebase Auth for login, Firestore to store messages, and Firebase Hosting to run the app.

**35. How did you deploy web apps on AWS?**  
I launched EC2, installed app code, used RDS for database, and stored files in S3. Managed everything with security groups.

**36. Biggest challenge in internship?**  
At first, I found IAM policies confusing. I solved it by reading AWS docs and testing with different permissions.

**37. How did you manage backup in EC2?**  
I created EBS snapshots regularly. If anything went wrong, I could restore the instance using that snapshot.

**📑 Scripting and Programming**

**38. What scripts have you written in Shell or Python?**  
I wrote scripts to automate backups, check system status, install packages, and run simple monitoring tasks.

**39. Bash script to monitor disk usage?**

#!/bin/bash

df -h > disk\_usage.txt

**40. Any tasks automated during internship?**  
Yes, I automated EC2 snapshot creation using a shell script and cron jobs.

**🧠 Behavioral / HR-Type**

**41. Why are you interested in AWS Support role?**  
Because I enjoy solving cloud-related problems. I want to help users and grow my skills in AWS.

**42. Time you solved a technical problem?**  
Once, an EC2 instance stopped working. I checked logs, found a boot issue, restored it using snapshot and fixed it.

**43. How do you handle high-pressure or urgent tasks?**  
I stay calm, check logs or errors, break down the issue step by step, and ask for help if needed.

**44. What if you don’t know the answer?**  
I will check AWS docs or forums like Stack Overflow. If still unclear, I’ll ask my team or senior for help.

**45. How do you stay updated?**  
I read AWS blogs, watch YouTube tutorials, and follow tech people on LinkedIn.

**46. Why should we hire you over others?**  
Because I have hands-on AWS and DevOps experience, I’m a quick learner, and I’m passionate about cloud technologies.

**🐳 Docker Interview Questions & Answers (Simple)**

**1. What is Docker?**  
Docker is a tool that lets you **package and run applications in containers**. Containers are like small virtual machines but lightweight.

**2. What is the difference between a container and a virtual machine?**

* A **container** shares the host system's OS. It's faster and uses less space.
* A **VM** has its own OS and takes more resources.

**3. What is a Dockerfile?**  
It’s a text file with instructions to build a Docker image. Example: install Node.js, copy code, and run app.

**4. What is the difference between Docker image and Docker container?**

* **Image**: Blueprint of the app.
* **Container**: Running version of the image.

**5. How do you run a Docker container?**

docker run image\_name

**6. How do you see running containers?**

docker ps

**7. What is Docker Hub?**  
It’s an online place to store and share Docker images.

**8. Why is Docker useful in DevOps?**  
It helps run apps in the same environment everywhere — no more "it works on my machine" problem.

**🔧 Jenkins Interview Questions & Answers (Simple)**

**1. What is Jenkins?**  
Jenkins is an **open-source automation tool** used to build, test, and deploy code automatically (CI/CD tool).

**2. What is CI/CD?**

* **CI (Continuous Integration)**: Developers push code often and automatically test it.
* **CD (Continuous Deployment)**: Code is automatically deployed to production after tests.

**3. What is a Jenkins pipeline?**  
It’s a series of steps in a Jenkins job like:  
Pull code → Build → Test → Deploy.

**4. How did you use Jenkins in your project?**  
I used Jenkins to set up a pipeline. When I pushed code to GitHub, Jenkins automatically built a Docker image and deployed it to AWS EC2.

**5. What is a webhook in Jenkins?**  
A webhook is a way for GitHub to notify Jenkins when new code is pushed, so Jenkins can start the job.

**6. How do you install Jenkins plugins?**  
Go to **Manage Jenkins > Manage Plugins**, then install from the list.

**7. Where are Jenkins jobs stored?**  
They are stored as XML files inside the Jenkins home directory.

**🌱 Terraform Interview Questions & Answers (Simple)**

**1. What is Terraform?**  
Terraform is an **infrastructure as code (IaC)** tool. It helps you create and manage cloud resources like EC2, S3, etc. using code.

**2. Why do we use Terraform?**  
It automates the process of creating servers, databases, and networks. It saves time and reduces manual errors.

**3. What is the difference between Terraform and CloudFormation?**  
Both do similar things, but:

* **Terraform**: Open-source, supports many cloud providers.
* **CloudFormation**: Only for AWS.

**4. What is a Terraform provider?**  
A provider is a plugin that tells Terraform how to manage resources — like AWS, Azure, or GCP.

**5. What is the use of terraform init?**  
It sets up the Terraform project and downloads required providers.

**6. What does terraform plan do?**  
It shows what Terraform will do before making any changes.

**7. What does terraform apply do?**  
It creates or updates the resources as defined in your code.

**8. What does terraform destroy do?**  
It deletes all the resources created by Terraform.

**9. What are Terraform modules?**  
Modules are **reusable code blocks**. You write once and use it for multiple environments.

**10. How do you manage environments (Dev, QA, Prod) in Terraform?**  
By using **different variable files** or workspaces for each environment.