 Sure! My name is Bellamkonda Siva Teja. I’m currently working as a DevOps Engineer with around **3 years of experience** in build and release management, cloud infrastructure, CI/CD pipelines, and automation.  
 I’ve worked on AWS cloud services like IAM, S3 EC2, VPC, Route53, ECR, ECS and RDS, and I’m skilled in containerization using **Docker** and orchestration with **Kubernetes**. I’ve automated deployments using **Jenkins pipelines**, written **Terraform scripts** for infrastructure provisioning, and used **Ansible** for configuration management.  
 I’m passionate about streamlining deployments, enhancing system reliability, and improving collaboration between development and operations teams.

2.     Can you walk me through your role and responsibilities in your current project at Kyndryl?  
My day-to-day tasks involve:

* Writing and maintaining **Jenkins pipelines** for automated application builds and deployments.
* Using **Shell scripting** to automate manual tasks like log collection, backups, and service checks.
* Handling **AWS security alerts** via Wiz portal, like securing S3 buckets, patching Log4J vulnerabilities, and enforcing IMDSv2 on EC2.
* Troubleshooting and fixing any issues in CI/CD pipelines and infrastructure.
* On-call support during critical hours and weekends to resolve deployment or server-related issues.

Collaborating with development and QA teams to resolve build issues and improve deployment pipeline

“In my last project, I worked as a **DevOps and Cloud Engineer** on the *Shell Fleet Hub* platform, specifically supporting the **Transaction Monitoring & Reporting** module. This module handles real-time fuel card transaction data and generated reports for fleet managers.

My responsibilities covered both DevOps automation and cloud infrastructure, including:

* **Provisioning AWS infrastructure using Terraform**, such as VPCs, subnets, EC2, ECS (Fargate), security groups, and IAM roles.
* **Deploying backend services in containers** using **ECR** and **ECS**, which handled ingestion and processing of transactional data.
* Implementing **CI/CD pipelines in Jenkins** integrated with GitHub for automated build, test, and deployment workflows.
* Using **Amazon RDS with Multi-AZ deployment** to ensure high availability and automatic failover for our PostgreSQL database. This was critical for keeping transaction logs and report data resilient and highly available.
* Configuring **CloudWatch** for centralized logging and alerts across ECS tasks, EC2 instances, and RDS, which helped in troubleshooting and proactive monitoring.
* **Enforcing IAM best practices** for secure service-to-service communication and least-privilege access control.
* As a Cloud Engineer, I also contributed to **cost optimization**, like selecting appropriate EC2 instance types, using auto-scaling, and managing reserved capacity.

Overall, my role was to ensure that deployments were automated, infrastructure was resilient, and the system met uptime and performance expectations.”  
  
  
  
  
  
  
  
“In the Shell Fleet Hub project, the platform was built on a **microservices architecture**. Overall, the application consisted of around **12–15 microservices**, each responsible for a specific domain — for example:

* **Transaction Processing**
* **User Management**
* **Fleet Vehicle Data Service**
* **Reporting Service**
* **Authentication & Authorization**
* **Notification Service (email/SMS alerts)**

As a **DevOps and Cloud Engineer**, I was directly handling deployments and maintenance for about **6–8 of those services**, particularly the ones under the **Transaction Monitoring and Reporting** module.

Each microservice was:  
  
  
**1. User Management Service**

* **Function:** Manages user registration, authentication, and roles.
* **Responsibilities:**
  + ECS (Fargate) deployment with secure IAM roles.
  + Secrets managed in AWS Secrets Manager.
  + Jenkins CI/CD pipeline with auto-deploy.
  + CloudWatch monitoring for login failures and performance.

**2. Notification Service**

* **Function:** Sends email and SMS alerts for account and vehicle notifications.
* **Responsibilities:**
  + Integrated with AWS SNS.
  + CloudWatch for delivery monitoring.
  + SSM Parameter Store used for configs.
  + Deployed with Jenkins automation.

**3. Audit Logging Service**

* **Function:** Logs user activity for compliance.
* **Responsibilities:**
  + Logs stored in CloudWatch and exported to S3.
  + Lambda used for log archival to S3.
  + Lifecycle rules for retention and storage cost optimization.

**4. Frontend Hosting Service**

* **Function:** Hosts web dashboard frontend.
* **Responsibilities:**
  + Code deployed to S3 static website.
  + Integrated with CloudFront for CDN delivery.
  + Route53 used for custom domain routing.
  + Automated deployments via Jenkins.

**5. File Upload Service**

* **Function:** Allows users to upload vehicle documents and fuel receipts.
* **Responsibilities:**
  + S3 used for file storage with bucket policies.
  + IAM roles configured for secure upload access.
  + CloudWatch logs for upload errors.
  + ECS-deployed microservice for handling metadata.

**6. Internal Dashboard Service**

* **Function:** Internal tool for admins to monitor and control system services.
* **Responsibilities:**
  + Deployed on EC2 inside a private subnet.
  + VPC setup includes NAT gateway and security groups.
  + Route53 used for internal DNS routing.
  + Jenkins used for manual approvals and deployments.