Explain DevOps Project In Interview

Project Overview: -

- Frontend
- Backend
- Jira Ticket System
- Confluence – For Documentations
- Slack or MS Teams – For Internal Chat, Calls & Meetings
- Lastpass or 1Password - To store credentials and share with other team members Securely
- GitHub – To store project application codes

Frontend:

Tech Stack Details

- ECS Farget Service - Container
- Route53
- Load Balancer
- SSL Certificate
- ECR Registry
- Custom VPC

Backend: -

Tech Stack Details

- Lambda Functions (Nodejs-16)
- API Gateway
- RDS - MySQL Database with Replication
- S3 Bucket
- Route53
- SSL Certificate
- Custom VPC

How alerts triggered?

- ✓ AWS SNS Topic with Email Subscriptions
- ✓ Alerts are integrated with Slack Channel

Infra Network Setup

- ✓ Frontend to Backend - VPC Peering or Direct Connect Service to make two different account VPC private connections
- ✓ RDS – It should be in private subnet and secured
- ✓ RDS password should be stored in secret manager

Deployment End to End Process: -

Frontend: -
CICD Pipeline stage should be like this: - Build → Test → Deploy → Test →Prod
Follow proper Git branching strategy during deployment
Branching strategy could be like, dev, hotfix, features, release branches
Always follow code review process before merging into master branch

CICD Pipeline Work
Build: - Build Docker image and pushed into ECR registry
Test: - SonarQube should be there integrated to check Code quality.
Deploy: - Deploy latest image from ECR to ECS container

Backend: -
CICD Pipeline stage should be like this: - Build → Test → Deploy → Test →Prod
Follow proper Git branching strategy during deployment
Branching strategy could be like, dev, hotfix, features, release branches
Always follow code review process before merging into master branch

CICD Pipeline Work
Build: - Install required package to make a bundler for lambda functions
Test: - SonarQube should be there integrated to check Code quality.
Deploy: - Deploy latest changes related to Lambda, API Gateway & RDS

Monitoring:

- ✓ Grafana
- ✓ CloudWatch

What Is Covered in Monitoring?

- ✓ **RDS: -** CPU, Memory Utilization, DB Connections, Replica Lags
- ✓ **Lambda Functions: -** Errors, Durations, Invocations
- ✓ **ECS Container: -** CPU, Memory Utilization
- ✓ **API Gateway: -** 5xx error, Hit count, Latency

Day to Day Activities:

- Monitor Infrastructure status by using Grafana and CloudWatch

- Check Jira ticket status and work on pending task
- Production release management if any
- Setup CICD pipeline according to project requirement
- Follow best practices Git branching strategy in CICD for deployments
- Write Docker file as per the application
- Create/Manage infra on AWS using terraform.
- Add new users or provide access to users as per request in IAM.
- Always find a way to automate the tasks and do the enhance wherever I see the opportunity
- Daily standups, client meetings and internal team meetings
- Create infrastructure related documents in confluence

Real Time Issues & Troubleshooting

AWS: -

- ✓ Increase EBS Volume Size for EC2 without Downtime
- ✓ Configure Auto Scaling for Better Optimized Setup
- ✓ Enable termination protection for RDS, Load Balancer, and EC2
- ✓ Delete older files from S3 Bucket
- ✓ Server performance is very slow. Increase Instance types
- ✓ RDS database server working slow.
- ✓ Server and Database not able to connect
- ✓ Lambda function timeouts
- ✓ Security group policy
- ✓ IAM User or Role with policy management
- ✓ S3 Bucket Security. Don't make S3 bucket public
- ✓ Automate the EC2, Database Backup

Jenkins or GOCD: -

- ✓ Pipeline failure due to server not connect
- ✓ Plugins upgradation issue
- ✓ Configuration issue with pipeline like variables, SSH etc.
- ✓ Agent failure during pipeline execution
- ✓ Jenkins master server crash or failure
- ✓ Limitation of build executors
- ✓ Store credentials securely
- ✓ Security Vulnerabilities like port open, unsecured configurations etc
- ✓ Outdated Jenkins version
- ✓ Master-slave server failure

Kubernetes: -

- ✓ Infrastructure capacity issue for node to launch new container
- ✓ Networking configuration challenges
- ✓ Log management or export logs to CloudWatch or Grafana etc
- ✓ Cluster Setup and Connection
- ✓ Pod monitoring
- ✓ Check pod status after every deployment and make sure it should be running
- ✓ Setup CICD pipeline for new deployment in k8s cluster
- ✓ If any error logs then according to error, send it to developer team to fix it

Real Time Issues:

Pod Deployment Failures:
Issue: Pods fail to deploy, and troubleshooting the root cause, whether its misconfigured resources, image availability, or connectivity issues, can be challenging.

Ingress Configuration Problems:
Issue: Ingress rules not working as expected, leading to routing or load balancing issues. Debugging involves checking configuration syntax, backend services, and networking.

Persistent Volume (PV) and Persistent Volume Claim (PVC) Mismatches:
Issue: Mismatched PV and PVC configurations can lead to data access problems. Resolving this involves aligning storage classes, access modes, and reclaim policies.

Networking Issues:
Issue: Networking challenges like pod-to-pod communication failures, service discovery issues, or external access problems. Diagnosing involves examining network policies, service configurations, and firewall settings.

Resource Constraints:
Issue: Pods experiencing resource limitations or excessive resource usage, causing performance degradation. Addressing this requires optimizing resource allocations and scaling strategies.

Scaling Challenges:
Issue: Difficulty in scaling applications horizontally or vertically due to misconfigurations, improper auto-scaling settings, or limitations in cluster capacity.

Secrets Management:
Issue: Problems with managing and securing sensitive information using Kubernetes secrets, including issues with encryption, distribution, and updates.

Node Failures and Recovery:
Issue: Nodes going down unexpectedly, affecting application availability. Handling this involves implementing node health checks, redundancy, and automated recovery mechanisms.

Image Registry Access Issues:
Issue: Problems pulling container images from registries during pod initialization, often related to authentication, authorization, or image availability.

Rolling Updates and Rollbacks:
Issue: Challenges in orchestrating rolling updates without downtime or rolling back to a previous version when issues arise. This requires careful management of deployment strategies and versioning.

Terraform: -

State File Corruption:
Issue: Terraform state file corruption can occur due to unexpected interruptions or conflicts, leading to inconsistencies in infrastructure management.

Resource Dependencies:
Issue: Managing dependencies between resources can be challenging, especially when creating resources that depend on outputs from other resources.

Variable Validation:
Issue: Ensuring proper validation of input variables can be tricky, leading to misconfigurations or unexpected behavior.

Sensitive Data Handling:
Issue: Managing sensitive data like API keys or passwords in Terraform can pose security risks.

Provider Version Compatibility:
Issue: Upgrading Terraform versions might lead to compatibility issues with specific providers or modules.

State Locking:
Issue: Concurrent Terraform runs can result in state locking issues, causing conflicts and potential data corruption.

Dynamic Resource Creation:
Issue: Dynamically creating resources based on variable inputs can be complex and prone to errors.

Module Versioning:
Issue: Managing module versions across different environments can lead to inconsistencies.

Rollback Challenges:
Issue: Rolling back infrastructure changes can be difficult, especially when dealing with destructive changes.

Provider Rate Limiting:
Issue: Some cloud providers impose rate limits, causing Terraform to fail during rapid or large-scale deployments.

For More AWS & DevOps Related Content Contact Me On: -

WhatsApp: - +91 – 7249 0590 06
Email ID: - support@devopswithnamdev.com
YouTube: - DevOps with Namdev

https://www.youtube.com/@namdev.devops