**Devops Self Introduction**

**Terraform**

**Kubernetes**

**Tell me about yourself**

**My name is Sivakumar, I have 4 years of experience in DevOps and AWS. We are using terraform as IaaC tool to create our infra in AWS. We have created our own modules and maintaining them. We are using EKS as our deployment platform. We have few legacy applications in VM as well. We configured autoscaling for VM as well EKS.**

**We have applications in NodeJS, Java, Python. We have different kinds of DB like MySQL, MongoDB. We have a seperate DBAdmin to manage them**

**We develop Jenkins CICD pipeline to deploy the applications into VM as welll as EKS. Our tools stack is**

**GitHub/Bitbucket --> GitHub enterprise into our own infra, https://github.joindevops.com**

**Jenkins --> CICD**

**SonarQube --> Scanning**

**Unit testing**

**Maven, NPM, pip as build tools**

**Nexus for artifact upload**

**Docker for imaging**

**Helm charts for deployment**

**We follow industry best practices like**

**shift-left**

**build once run any where -->**

**jenkins shared libraries, pipeline as a code**

**We have centralised pipelines, our developers can just call those pipelines to deploy applications, these are completely automated DEV to PROD. We let our developers to completely concentrate on development instead of deployment.**

**clone**

**build**

**unit testing**

**scanning --> code coverage 80%**

**build**

**artifact upload to nexus**

**build docker image**

**upload image nexus/ECR**

**deployment using helm**

**How to promote app to PROD**

**-----------------------------**

**Developers raise JIRA ticket mentioning their details**

**Project code**

**application**

**version**

**deployment date**

**approver**

**sonar scan results**

**SAST, imaging scanning results**

**what happened to testing**

**Change management**

**-----------------------------**

**Support team will raise CR, they will mention the JIRA ticket. same details as in JIRA and approval from client.**

**Once CR is approved. Then our team will also approve JIRA**

**We use JIRA for project management, we get tickets from our team lead or developers.**

**master and feature**

**We help them to onboard their project in all tool stack.**

**support team, they will trigger deployment**

**our pipeline to PROD will be called and deployed..**

**main --> feature**

**feature --> PR**

**merge vs rebase**

**a merge commit ID is created into main branch...**

**if commit ID is changed, are you 100% sure code is not changed --> no**

**if commit ID is not changed, are you 100% sure code is not changed --> yes**

**merge strategy**

**merge**

**rebase**

**squash**

**feature --> 20 commits**

**merge --> main branch will have a commit called merge commit**

**rebase --> commit id will be rewritten and pushed to main branch**

**fast-forward merge --> commit ID will never change even if you merge PR from feature to main branch.**

**main --> catalogue-deploy**

**PROD deploy**

**we have webhooks configured from github/bitbucket to jenkins, whenever there is a new commit into feature our developement pipeline will be automatically triggered.**

**new commit push is the event**

**main --> merge PR event**

**what jenkins pipeline you can create --> create JIRA ticket automatically**

**you can trigger pipeline automatically**

**SRE**

**--------**

**jenkins, git, sonar, jira, cr tool**

**authentication**

**authorisation**

**monitor**

**upgrade**

**backups**

**rate limiting**