*SELF INTRO*

My Self Ramakrishna Reddy I have total 2+ years experience as a DevOps engineer in Microland. My roles is to automate the Build and Deployment's for the applications which are running in the diff technologies like **SpringBoot, Microservices , Angler, .Net** as part of this i am involved like different different CICD process, so that application teams follow to build and deliver their application in to respective environments. And I am familiar with Jenkins to create CI/CD jobs Setting up the Jenkins servers and master and slave configuration and managing the CI/CD jobs, Managing Jenkins plugins like adding the additional plugins, and integrating the other **DevOps tools like Git, Maven, Tomcat Apache, SonarQube, Nexus Docker and other …,** And we are using the **Git** as a VersionControlSystem enabling the branching Strategies like **Dev, Qa, Prod, Pre-prod Branches** we create. Manging git repositories using various GIT commands like **push, pull, meagre, clone, branch, commit.** Manging git user access nothing but giving access to the specific Repositories or Specific branches to secure our code. And we are using **Maven as Build Tool, SonarQube for the Quality analysis**, then to **publish the artifact to the nexus repository** and i have knowledge on **Docker** to create the docker files and building the images and i am familiar with **Linux commands** and i have Knowledge on **Aws Services** **like Ec2, Vpc, Lodebalancer, S3.**

**As part this my daily responsibilities are, when i am login to the system, firstly i can check the mails, whether any ticket is generated regarding the daily tasks.**

**for example the ticket is regarding Git, & i will execute the git related commands and all. to creating the user account's, creating the new branches & etc..**

**And i Monitor the CI jobs, that means in our organization some nightly jobs are scheduled. So every, next day morning we have to see the status of the nightly job, whether the nightly job is successful/not. if it is failed, it is my responsibility to escalate that failure to the respective dev-team, they look into that.**

**so here what happend, devlopers will fix the issue, so here my responsibility is to monitor the ci jobs make sure everything green.**

**And in the last 3 months I involved in the migration part**

**so previously they maintained all the data in digital-ocean. so now they want to migrate digital-ocean to aws-cloud. actualy they given 2 ansible playbooks, one is host-create-file & another one is host-transfer-file.**

**so here in the host-create-file, i mention all the aws details, like IAM information, aws credentials and what type of instances we are creating & instancing type, and how many instances we have to create & all, all these details we mentioned in the ansible host-create-file.**

**Then after i run the host-create-file, so that automatically empty instances are created in aws-cloud. so here, up now we created the empty instances in the aws cloud, here there is no data present in these instances what ever we created. so here the data present in the digital-ocean.**

**and here we have another playbook called host-transfer-file, in the host transfer file i mention, all the digital-ocean server IP(source )address are mention, and also we mentioned the server-ip address what we created in the aws-cloud. after that we can run the host-transfer-file, automatically what ever the data present in the digital-ocean, migrate to aws-cloud. in this process all the keys(pem-files) are generated. and we can store these keys in s3 bucket for backup, why because for the future purpose when we want those key's we can easily retrieve those pem files and easily use those pemfiles.**

**so that, to process this requirement we are maintaining the vpc for security purpose, instances and s3 buckets.**

**=================================================================**

And coming to **CI(continuous integration),** In a SoftwareDevoplopment, developer’s integrate their work frequently, it may leading to multiple integrations per a day. So lets Assume I am a **team x** and my team got the requirement called Feature. Which are going to deliver in the next release so here the team will do basically checkout the Base-Code in our local machine. At this particular point once we checkout the code, the base-code which is present in the **Centralized/Distributed Repository** and the base-code which is present in our local Machine, both are at the same level.

Development team will do the related changes in the software, what ever the requirement we get, once our changes will be done, then they create team specific software build and deploy that and verify the integration server if the integration is working fine then they will meagre the changes to the basecode & if any failure then development team will do the changes again, and we will test the changes again ….., it is a cycle. We will test the local changes until & unless all the test cases are passed like Regration test, Functional test, System-test…., etc.

Once all the test cases are passed and we assumed that our changes are ready to frize. And meagre changes in to the Basecode.

**CD(Continuous Delivery)** the base code is called production ready code, when ever we trigger a new build for Basecode that must be standard which we can deploy for the customer.