**Self-Related Questions**

1. **Tell me about your self**
2. **What technology you have used in the previous projects**

• My name is John. I have 10 years of experience in IT industry and around 3 years of experience in DevOps and AWS(cloud).

• Currently I am working in xyz organization as DevOps engineer.

• As part of my role, I am responsible to setup and manage DevOps CI/CD pipelines.

• I have used various DevOps tools like Git, Bitbucket (for version control system), Jenkins (for continuous integration), Maven(as a build tool), SonarQube(for code analysis) Jfrog(for artifact repository), Ansible (for continuous Deployment Configuration Management), Docker(containerization) and Kubernetes (for container orchestration).

• I am also working on AWS services.

• This Devops environment is running on AWS cloud which was setup by me according to our DevOps Architect plan.

• I am also support the project as it deploy it AWS Env and local Linux server.

1. **Tell about yourself more**

**Tell me more details about yourself:**

**Git:-**

• Enable branching strategies

• Managing git repositories using various git commands like push, pull, merge, clone, brnach, rebase etc..

• Managing git user access

**Jenkins:**

• Setting Jenkins server

• Master and slave configurations

• Managing CI/CD jobs

• Managing Jenkins plugins

• Integration other DevOps tools like git,Maven,Ansible,SonarQube,Nexus etc..

**Ansible:**

• Setting up Ansible control server

• Writing playbooks and roles

• Managing Ansible hosts

**Docker:**

• Setup Docker environment

• Writing Dockerfiles

• Creating custom images

• Setting up Private Docker registry

• Creating and managing containers

**Kubernetes:**

• Setup Kubernetes environment on AWS or on-premises

• Writing deployment files and service files

• Managing containers on pods

• Managing Kubernetes nodes

1. **What is your team size?**
2. **What is your role and responsible in your Team.**

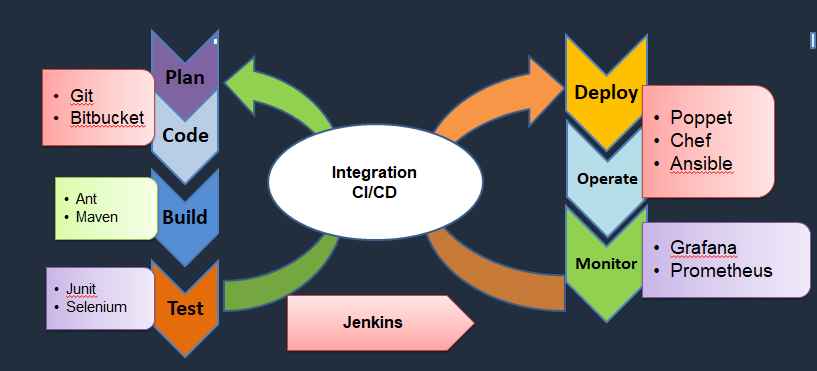
* Test, build, design, deployment, and ability to maintain continuous integration and continuous delivery process using tools like Jenkins, maven Git, etc.
* Must know how to choose the best tools and technologies which best fits the business needs.
* Ability to automate test and deploy the code and monitor.
* Work in close coordination with the development and operations team such that the application is in line with performance according to the customer's expectation.

1. **How good are you with programming?**
2. **How quick can you learn**
3. **What is your Day to Day life in your Organization?**

* Make sure that the pipeline is running smoothly
* Interaction with the other teams
* Work on automation backlog
* Infrastructure management
* Dealing with legacy stuff
* Based on ticketing tool work on prior based incident
* Training and Self Development
* Documentation

1. **Tell me most challenging activity which you faced in DevOps**
2. **What did you achieve new on your current or previous organization as DevOps engineer?**
3. **What are the different phases in DevOps?**

The various phases of the DevOps lifecycle are as follows:



* **Plan** - Initially, there should be a plan for the type of application that needs to be developed. Getting a rough picture of the development process is always a good idea.
* **Code** - The application is coded as per the end-user requirements.
* **Build** - Build the application by integrating various codes formed in the previous steps.
* **Test** - This is the most crucial step of the application development. Test the application and rebuild, if necessary.
* **Integrate** - Multiple codes from different programmers are integrated into one.
* **Deploy** - Code is deployed into a cloud environment for further usage. It is ensured that any new changes do not affect the functioning of a high traffic website.
* **Operate** - Operations are performed on the code if required.
* **Monitor** - Application performance is monitored. Changes are made to meet the end-user requirements.

1. **What is Continuous Integration?**

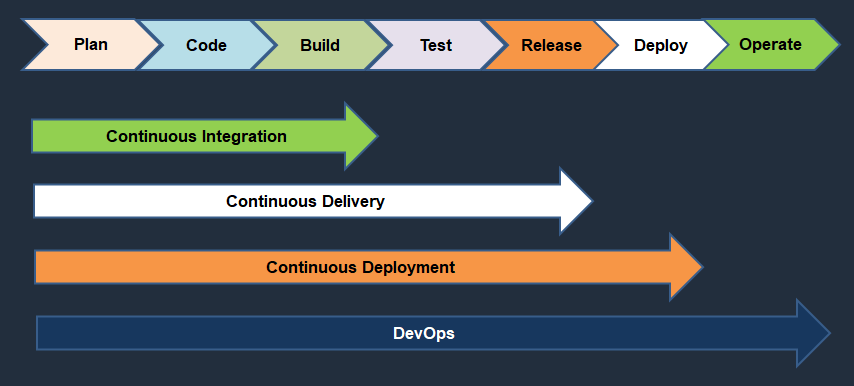
* Developers will push their code several times in a day to a central repository, every time there is code change it should be pulled, built, tested and notified.
* CI encourages developers to share their code and unit tests by merging their changes into a shared version control repository after every small task completion. Committing code triggers an automated build system to grab the latest code from the shared repository and to build, test, and validate the full main or trunk branch.

1. **What is continuous Delivery?**

* The process of CI, the artifact has been generated.
* The artifact should be delivering to all servers in different environments like Dev, QA, Staging.
* It should be automatically delivered to QA servers where testers will do functional tests, load tests etc.
* After it passes the QA tests it should automatically deliver the code to staging area where customer or client or users check the changes and approval to deploy it to production.

1. **What is Continuous Deployment?**

* If the approval is manual then code delivery is continuous Delivery but if the approval process becomes automated then after staging, the code change is done directly to Production systems.
* This is called as Continuous Deployment.



1. **Tell me about your DevOps Lifecycle process in your project.**
2. **What is Blue/Green Deployment Pattern?**

* Blue green deployment is an application release model that gradually transferred user traffic from a previous version of an application to identical new release which is running in production.
* **Older version**: Blue Environment
* **New Version**: Green Environment
* Once production traffic is fully transferred from blue to green, blue can be standing by in case of rollback.

1. **How does continuous monitoring help you maintain the entire architecture of the system?**

Continuous monitoring in DevOps is process of detecting, identifying and reporting any faults or threats in the entire infrastructure of the system.

* Ensures that all infra, applications and resources are running fine on the servers.
* Monitoring the status of CPU, Load Average, Disk Usage and Memory usage
* Make sure the application running on the server is running smoothly or not.