**Devops questions:**

What is DevOps? Why is it important?

* Devops bridges the gap between Dev team and Operational team.
* Breaks the silos between the Dev team and Operational team.
* Faster the development process.
* Helps in streamlining the software development.
* Culture approach which varies from Organisation to Organisation
* Ensures the Quality with continous monitoring and testing.
* Continous monitoring helps in running the applications smoothly.
* Automating the tasks helps in minimizing the human errors.
* To deliver the project/product to the client on time ensuring the quality with continous testing and monitoring.

How is DevOps different from traditional IT operations?

**DEVOPS:**

* Dev team and Operational team collaborate.
* Automated, fast and efficient.
* Releases are frequent and small and incremental.
* Continous testing throughout the Devops process.
* Development process is faster.
* Due to Continous monitoring the issues will be detected easily.

**TRADITIONAL IT OPERATIONS:**

* Teams will not collaborate.
* Development process will be slow.
* Releases are slow and large.
* Testing will be manual.
* Development process is slow with large life cycles.
* Fixes the issues after the deployment.

What are the key principles of DevOps?

* Communication and Collaboration:
  + Devops promotes the culture of the collaboration between the Dev team and Operational team.
  + Breakdown the silos leads to faster and more efficient software delivery.
* Automation:
  + Automating the tasks helps to reduce the errors and increases the efficiency.
  + Tools like Jenkins,Docker help to automate the tasks.
* Continous Integration:
  + Developers frequently merge their code into the remote repository.
* Continuous Delivery (CD) & Continuous Deployment:
  + Ensures that the software is in a deployable state with the automated testing.
  + Deploys the changes automatically.
* Infrastructure as a Code:
  + Managing the Infrastructure using code rather than manual configurations.
  + Ensures consistency,scalability.
* Monitoring and Feedback:
  + Real time monitoring of applications and infrastructure to detect issues early.
  + Improves reliability and scalability.

What are the benefits of implementing DevOps in an organization?

* Faster Software Delivery:
  + Automates the software development life cycles and reduces the delivery time.
  + CI/CD ensures the faster releases.
* Improved communication and Collaboration:
  + Breaks the silos between development and operations team.
  + Encourages the culture of shared responsibility.
  + Miscommunication will be reduced due to the collaboration.
* Higher Software Quality and Reliability:
  + Automation issues were fixed early.
  + Continous monitoring ensures stable and the reliable applications.
* Increased Efficiency and Productivity:
  + Automation reduces the repetitive tasks and manual tasks helps to increase the efficiency.
* Cost Optimization:
  + Reduces the Operational costs by minimizing effort.
  + Efficient resource utilization reduces waste and infrastructure costs.
  + Reduces the time and improves the overall quality.
* Scalability and Reliability:
  + Cloud-native Devops enables the scaling up or down as needed.
  + Containerization makes the application portables
  + Helps to meet the market trends easily.
* Enhanced Security:
  + Highly Secured.
  + Automated security testing helps to identify the issues early.

What are the key concepts of Devops-Life Cycle?

* Plan:
  + In this phase, teams define the requirements, objectives and workflows.
  + Uses Agile methodologies in the sprint planning.
  + Tools: Git,jira.
* Develop:
  + Developers write, review, and code.
  + Uses Git - repositories for collaboration.
  + Tools: Git,Github,Gitlab.
* Build:
  + The source code is compiled and built into executable applications.
  + Dependencies and libraries are installed.
  + Tools: Gradle,Maven.
* Continous Integration:
  + Developers merges the code changes into the remote repository.
  + Automation builds and tests run to detect issues early.
  + Tools: Jenkins.
* Continous Testing:
  + Automated testing is performed to ensure the software quality.
  + Includes the Unit,Integration,Performance and the Security Testing.
  + Tools: Selenium.
* Release and Deploy:
  + The Software is deployed to the production environments.
  + Continous Deployment ensures faster deployments.
  + Tools: Jenkins.
* Operate and Monitor:
  + Ensures the performance monitoring, logging and issue detection in the real time.
  + Ensures proactive resolution and performance optimization.
  + Tools: Ansible.
* Security:
  + Security checks are integrated into the pipeline.

Explain the difference between Agile and DevOps.

| **Agile** | **Devops** |
| --- | --- |
| A s/w dev approach that focuses on incremental, iterative and customer driven development. | A s/w dev and IT approach that focuses on automation, ci/cd and fasters the s/w development. |
| Improves flexibility, satisfaction and customer satisfaction. | Improves automation, ci/cd and fasters the s/w development. |
| Primarily focuses on s/w dev process. | Covers entire s/w lifecycle. |
| Key practices: Sprints, Scrum and user stories | CI/CD, IaC, Automated monitoring, Security. |
| Team Structure: Cross functional dev team | Collab b/w dev, ops,security and QA. |
| Short development cycles(2-4) weeks | Using CI/CD has multiple releases per day. |
| Limited automation | Extensive Automation. |
| Manual and Automated testing during the sprint. | Continous testing throughout development and deployment lifecycle |

Name some popular DevOps tools and their use cases?

* **Git:** De - Centralized Version Control System used for code management.
* **GitHub / GitLab:** Web based git - repositories with CI/CD and collaboration features.
* **Jenkins:** open source automation tool used for building, testing and deploying applications.
* **Docker:** enable containerization, allows the developers to package their application into containers, this helps to maintain consistency among the different phases.
* **Kubernetes:** manages the containerized applications at scale. It automates the deployment,scaling, operations of the containerization applications of the clusters of the hosts, maintains the high availability and efficiency of the resources.
* **Ansible:** provides agentless configuration management and orchestration. It uses YAML playbooks to automate system configurations, software provisioning and application deployments.

**What is Iac?Importance of IaC and its types?**

* Devops Practice used to automate the process of provisioning, configuration and infrastructure management by using instead of manual process.
* It allows the team to define the Iac in Declarative or Imperative scripts to ensure scalability, consistency and repeatability.
* Importance:
  + Automation: Eliminates the manual process of provisioning.
  + Consistency: Helps to maintain the uniform environment
  + Scalability: Enables the faster deployments of Infrastructure of the code at a time.
  + Version control: Infrastructure as a Code can be stored in Git/Github.
  + Faster Deployments: The infrastructure will be deployed from the cloud easily.
  + Cost efficiency: Resources will be utilized efficiently.
  + Disaster failure: Helps to setup the infrastructure immediately in case of failure
* Types:
  + Declarative(What to achieve):
    - Specifies the infrastructure to reach the desired state.
    - Automatically make changes to achieve the desired state.
    - Ex: Terraform
  + Imperative(how to achieve):
    - Step by Step instructions to configure the Infrastructure.
    - More control but hard to maintain
    - Ex: Ansible