

Introduction to DevOps...

Page No.	
Date	

Q8] What is DevOps? Explain role of devOps engineer & developer responsibility in devOps culture.

-
- DevOps is combination of Developer & Operations.
 - It is built to make faster & more reliable software delivery.
 - It enables collaboration between software developer, QA engineers as System administration (SA) throughout the entire project lifecycle from design & coding to testing, deployment & maintenance.
 - DevOps uses automation, continuous integration & continuous delivery (CI/CD) to streamline workflows.
 - Goal of DevOps is to increase speed, quality & efficiency of delivering applications & services.

~~Role of DevOps Engineer~~ →

Developer Responsibilities in DevOps Culture

- 1) Process & Tool Integration → Introduce process, tools & automation methods to improve collaboration b/w development & operations teams
- 2) Documentation → prepare tech stack, architect diagram
- 3) System Analysis → Evaluate current infrastructure

- L-17/18
- importance of software testing
- 1) Development & Coding → write code, automation testing, buildability, manage repositories.
 - 2) Project planning → meetings, costs, upcoming projects.
 - 3) Testing & Quality Assurance → perform automated & manual testing of code.
 - 4) Maintenance & Troubleshooting →
 - 5) Performance optimization.



Roles of Developers

- 1) Process & Tool Management → choose and manage tools for the whole software process (coding → testing → deploying → maintenance) including CI/CD, automation
- 2) Automation → Automate tasks like code deployment, server setup & testing to save time & avoid mistakes
- 3) Infrastructure management → Take care of servers, cloud setups & networks so they can handle continuous updates
- 4) Coding & Scripting → write programs & scripts to build & maintain automation tools

5) Collaboration → helps developer and operation teams to work together smoothly

6) Problem Solving → fix complex issues & think for new solutions

7) Security → Add security checks in the development process & make sure everyone follows the standards

Q] Explain devops culture & Explain benefits of devops. ~~culture~~:

- Devops is the modern way of working that removes the gap between developer & operations
- It focus on teamwork, trust & continuous improvement instead of working separate
- The main idea behind this is "You build it, you run it" - developers ~~are~~ are responsible for running & maintaining what they build.
- Team gets freedom to make decisions without slow approval processes.
- Fast feedback is valued so that mistakes can be fixed quickly & improvement can happen continuously.

Key points in DevOps' Culture

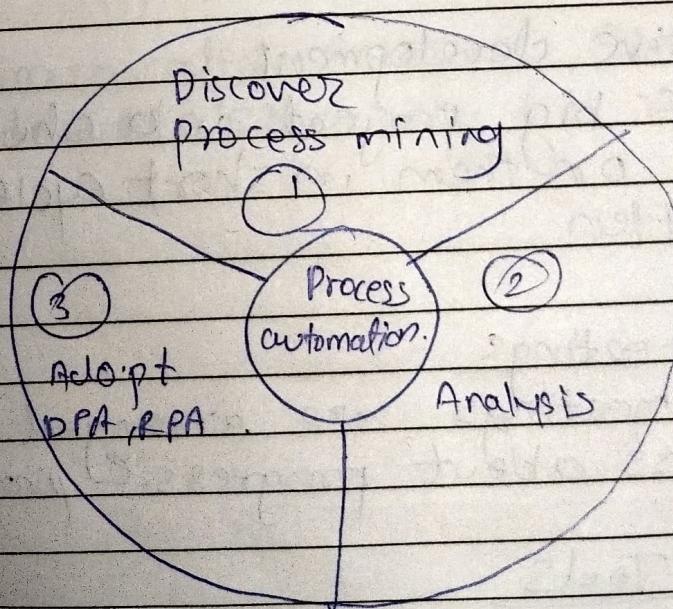
- 1) Shared responsibility
- 2) More Transparency
- 3) Team Autonomy
- 4) fast feedback

Benefits of Devops

- 1) faster & Better Releases → Work is delivered quicker with fewer errors
- 2) Improved collaboration & Trust → Team works as one unit with same goal
- 3) Faster problem Solving → Issues are spotted and fixed very quickly
- 4) Better handling of unplanned work. Team can deal with unexpected task.
- 5) Smarter decision making → Data is used for decisions without slow approval steps

Q] Explain process automation.

- process automation uses a technique to automate complex business processes
- It has three primary functions : automation processes, centralizing information & reducing the need for human intervention in tasks .
- The main goal of process automation is to reduce errors, prevent data loss & increase the processing speed of various business activities
- A most common process automation is business process automation (BPA) which uses software to automate repeatable, multi-step business transaction.



Project No.	
Date	

- An example of process automation is - washing car -
 - choose wash type
 - payment request
 - Takes payment
 - Approve transaction
 - car washing / rinsing / drying
 - last mirror touch
 - exit .

Q) What is agile ? Explain agile's best practices .

- Agile is a flexible way to developing software & managing projects in small part : work is delivered in small parts , plans are adjusted often , and teams quickly adapt changes

Agile Best practices →

1) Iterative development

- Break big project into chunks - work on them in short cycle & test often

2) Daily meetings

- Daily meetings are arranged to discuss about progress & problem

3) Right Tools

- use project management tools to organize

Project	
Date	

tasks & collaborate better

Agile best practices - Scrum project management
 → Scrum is known to be dominant Agile framework, According to survey 58% organization uses this Scrum, & 18% organization works with the combination.

Scrum practices →

- 1) Stand-ups → quick 15-min daily meetings to update everyone
- 2) Product Backlog & Vision → list all needed features and set a shared goal
- 3) Burndown Charts - Show remaining work vs completed work visually
- 4) Communication Rules → Clear communication guidelines, especially for remote teams

Q] Reason for adopting DevOps

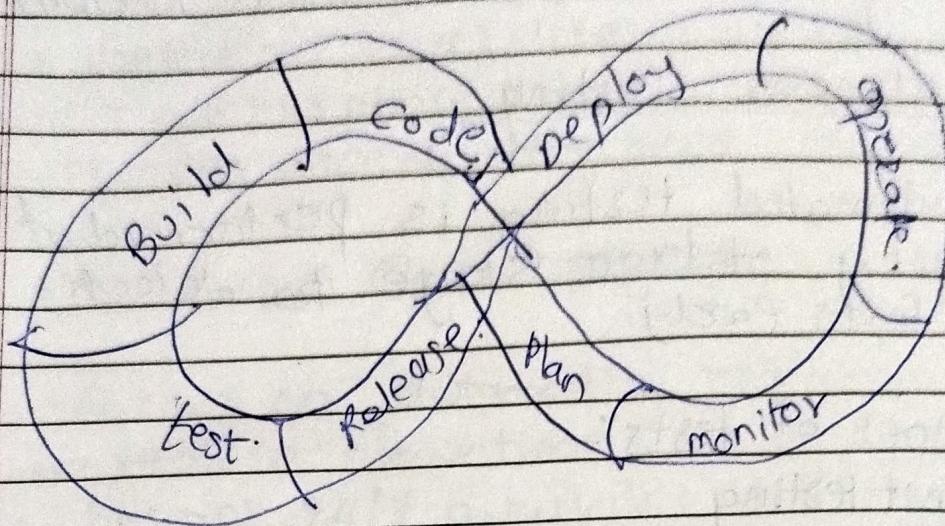
→ Reason for adopting DevOps.

- 1) Better planning
- 2) Fast delivery
- 3) Built-in security
- 4) Better customer experience
- 5) Improved Teamwork
- 6) More Time for innovation
- 7) Better decisions
- 8) High Trust.

Q] What and who are involved in DevOps.

- 1) DevOps Evangelist → Sr. lead
- 2) DevOps engg → Design, use automation, ensure security
- 3) Release manager → Manage process from planning to delivery
- 4) DevOps architect → Create CI/CD pipeline for fast delivery
- 5) Software dev/tester → Works on features, monitor performance
- 6) Security Engg → Add security into CI/CD pipeline
- 7) Automation Engineer → Automates processes to save time.
- 8) Quality Assurance → Quality check.

Q] Explain DevOps pipeline phases.



1) Continuous Development →

- Involves planning the project defining requirements & writing the code.
- Project scope, features, goals & timelines are decided.
- Developer writes code using programming language.
- Version controls like git, github are used to track and manage code changes.
- Code is broken into small, manageable commits to make integration & testing easier.

2) Continuous Integration (CI)

- 1) merges codes developed by multiple developers into a ~~single~~ single repository.

Ensuring that code works well with existing systems doesn't break functionality.

3) continuous Testing

1) Automated testing is performed at every stage to detect defects early.

2) Types of tests:

unit testing

Integration testing

Regression Testing

Performance Testing.

3) Test results are automatically sent to the development team

4) Continuous Deployment.

Automatically deploying to production without manual approval.

This is done by predefined tests.

Once the ~~set~~ tests are passed the system pushes the updates directly to software's users.

5) Continuous monitoring

Applications are monitored in real time for performance, security

2) Availability →

- Detects issues like high CPU usage, slow response time
- Issues are directly sent to devOps team.

6) Continuous Operation →

- Focus on automatic release & update to reduce downtime.
- Ensures 24/7 availability of application.
- Enables blue-green deployment or rolling updates so that updates happens without downtime.

7) Continuous feedback →

- Collects feedback from:
 - monitoring tools
 - user analytics
 - direct customer input.
- Feedback is analyzed to identify bugs & system satisfaction.
- Helps in improvement.

Q3 Explain monitoring Best practices.

- Monitoring is process of gathering and analysing data related to a critical system, services or app to run as expected.

Key best practices

1) Define monitoring needs

- Identify why you need monitoring.
- Set clear goals by discussing with security, legal & business stakeholders.

2) List what to log & How to monitor

- Decide what data to log
- Use alerts for unusual activities.
- Establish anomaly detection

3) Identify Assets & Events to monitor

- Focus on critical systems and apps
- Classify data based on security

4) Choose right solution

- Pick tools that are scalable & automated.

- Tune alerts regularly as threats evolve.

5) Design for security

- Mask sensitive data in log.

- Enforce role-based access control,
- Check log for integrity & apply encryptor,

6) Adopt organization.

- Define logging requirements for all system.
- Ensure consistency with company-wide policies

7) Establish an Incident Response plan.

- Have a tested incident response procedure

Trigger

- Trigger alerts quickly & take automated action when needed.