Assignment

Page No.:

Q7. Why is Scenario?

→ Reason why DevOps major requirement:->

- Shorter Development Cycles, faster Innovation:->

→ When we have a biased response from the development and operations teams, it is often difficult to tell if the application is operational. When development teams simply submit a request, the cycle times are unnecessarily extended.

- Reduce Implementation failure, Reflections and Recovery Time:->

→ The main reason for the failure of the teams in the implementation failure is due to programming defects. With shorter development cycles, DevOps promotes frequent code versions. This, in turn, makes it easy to detect code defects.

- Better Communication and Cooperation:->

→ Improved DevOps software development culture. The common teams are happier and more productive. Culture focuses on performance rather than individual goals. When teams trust each other, they can experiment and innovated more effectively.

- Greater Competencies:->

High efficiency helps accelerate development and makes it less prone to errors. There are ways to automate DevOps tasks.

⇒ DevOps - the amalgamation of development (Dev) and operations (Ops) teams - is an organizational approach that enables faster development of applications and easier maintenance of existing deployments. By enabling organizations to create stronger bonds between Dev, Ops and other stakeholders in the company, DevOps promotes shorter, more controllable iterations through the adoption of best practices, automation and new tools. DevOps is not a technology per se, but it covers everything from the organization to culture, processes and tooling.

Tools in Devops tools :->

- Collaboration Tools :->

→ This type of tool is crucial to helping teams work together more easily, regardless of time zones or locations.

- Planning Tools :->

→ This type of tool is designed to provide transparency to stakeholder and participants. Working together, teams can plan towards common goals, and better understandings of dependencies.

- Source Control Tools:->

-> Tools of this sort make up the building blocks for the entire process ranging across all key assets (whether code, configuration, documentation, database, compiled resources and your web site html - you can only gain by managing them in your one true source of truth.

- Issue tracking Tools:->

-> There tools increase responsiveness and visibility. All teams should use the same issue tracking tool, unifying internal issue tracking as well as customer generator tones.

- Configuration Management Tools:->

Without this type of tool, it would be impossible to enforce desired state norms or achieve any sort of consistency at scale.

- Continuous Integration Tools:->

Continuous integration tools provide an immediate feedback loop by regularly merging code.

- Automated Testing tools:
  - Tools of this sort are tasked with verifying code quality before passing the build.

- Deployment Tools:
  - It is an ideal effective DevOps environment, application deployments are frequent, predictable, and reliable. Deployment tools are essential to checking those boxes.

- Database DevOps Tools:
  - The database, obviously, needs to be an honored member of the managed resources family.