Assignment no. 2

Q. Demo DevOps a 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.
With joint development and operations efforts, the team's applications are ready to use more quickly.

- 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. Recovery time is an important issue because you should except some failure.

- Better Communication and Cooperation.
Improved DevOps also development culture. The common teams are happier and more productive. Teams can focus on bringing the product to market on production, and their key performance indicators must be organized accordingly.

- Greater Competencies.
Using an environment avoids a useless task of transferring data between environments. This means

that you do not have to use a development environment, a different testing environment and a third implementation.

- Reduce costs and IT Staff.
All the benefits of DevOps translate into reduced general costs and requirements of IT staff. DevOps development teams require IT staff to be 35 percent less and IT costs 30 percent lower.

2. 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 stakeholders 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 databases compiled resources and your website can only gain by managing them in

your one true source of truth.

- Issue tracking tools →
  These tools increase responsiveness and visibility. All teams should use the same issue tracking tool, unifying internal issue tracking as well as customer generated ones.

- 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 →
  " " " provide an immediate feedback loop by regularly merging code.

- Automated Testing Tools →
  Tools of this sort are tested with verifying code quality before passing the build.

- Deployment Tools →
  In an effective devops environment application deployments are frequent, predictable and reliable. Deployment tools are essential to checking those boxes.

- Database DevOps Tools →
  The " obviously needs to be an honored members of the managed resources family.