Case Study Assignment

DevOps case study assignment for an online shopping website:

Background:

You are a DevOps engineer at an online shopping website that has been facing issues in deploying new features quickly and reliably. The development team is frequently facing issues in the staging environment, leading to delays in the release of new features. The website's performance is also not up to the mark, leading to a poor user experience. The company wants to implement a DevOps approach to resolve these issues and ensure continuous deployment.

DevOps Approach Documentation:

The DevOps approach we propose for the online shopping website involves a continuous integration and delivery pipeline, infrastructure as code scripts, a microservices architecture, and monitoring and alerting setup.

Workflow of DevOps Approach:

1. Code Development: The development team will create code and push it to Gitor others.

2. Continuous Integration: The CI pipeline will be triggered to build, test, and validate the code using tools like Jenkins, TravisCI, or CircleCI.

3. Automated Testing: After the code is validated, automated tests will be run to verify the functionality of the code. The tests will cover unit testing, integration testing, and end-to-end testing etc.

4. Artifact Generation: Once the tests pass, an artifact (e.g., a Docker container) will be created containing the code and its dependencies.

5. Continuous Delivery: The artifact will be deployed to the staging environment using an IaC script such as Ansible, Terraform, or CloudFormation. The script will provision the infrastructure and deploy the artifact in a repeatable and consistent manner.

6. Automated Testing: Automated tests will again be run to verify that the code works in the staging environment.

7. Monitoring and Alerting: The deployed application will be monitored using tools like Prometheus and Grafana. Alerts: Alert will be set up to notify the team in case of issues such as high CPU usage, memory usage, or network latency.

8. Continuous Deployment: Once the code passes all tests, it will be automatically deployed to the production environment.

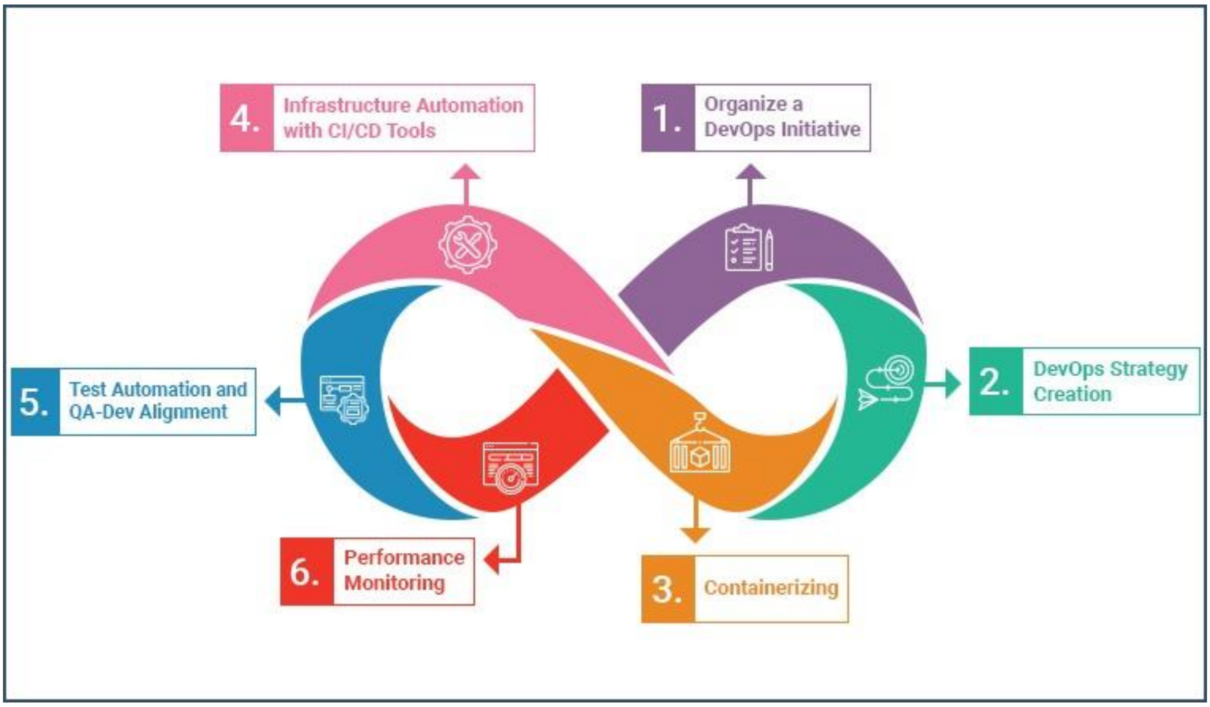
Tool Used:

1. Version Control System: Git
2. Continuous Integration: Jenkins, CircleCI
3. Automated Testing: Selenium, JUnit
4. Infrastructure as Code: Terraform
5. Monitoring and Alerting: Prometheus and Grafana

Working implementation of the DevOps approach, including the CI/CD pipeline, IaC scripts, monitoring and alerting setup, and microservices architecture:

When it comes to software development, DevOps is a methodology that brings maximum efficiency. It is a flexible method that combines various tools, philosophies, and practices to offer agility in the development process. And this is why the majority of the organizations are shifting from the traditional approach to DevOps Culture. But still, many software development companies fail to successfully implement DevOps, therefore, here in this blog, we will go through the DevOps implementation roadmap which can help you understand the implementation process thoroughly, and then we will have a look at the advantages of DevOps.

Successful DevOps Implementation Roadmap:


4. Infrastructure Automation with CI/CD Tools
1. Organize a DevOps Initiative
5. Test Automation and QA-Dev Alignment
2. DevOps Strategy Creation
6. Performance Monitoring
3. Containerizing

DevOps implementation is a concept that integrates operations, development, and testing departments together in cross-functional software development teams. The main aim of this is to improve the agility of the IT services. Basically, the approach of DevOps implementation is categorized into three sections – DevOps Continuous Integration (build), DevOps Continuous Testing (test), and DevOps Continuous Delivery (release). And this helps companies to shift from the traditional approach and change the software development methods as per the DevOps concept. For this, a few steps must be taken into consideration and they are –

1.1 Organize a DevOps Initiative

The first step to follow while shifting from the traditional software development method to DevOps is to organize a DevOps initiative for the IT department. By doing so, all the team members will get an opportunity to make necessary changes in the operational and development phase. In this initial phase, the CIO of the IT firm organizes everything and arranges financial investments in the best way possible. Besides this, the program manager will be responsible for designing and monitoring the DevOps implementation process.

1.2 DevOps Strategy Creation

The next step is to start creating the DevOps strategy which means that the project manager will use DevOps tools, principles and best practices that can help in improving interdepartmental collaboration. Besides, it also includes coming up with new ways of software development & testing, and infrastructure provisioning. Some of such best practices are –

- Automate software development, unit testing, software integrating, application testing via UI, deploying, and launching different processes to speed up the software development and testing releasing cycle.
- Another practice is to implement IaC that ensures the early provision of the IT infrastructure as per the request of software developers and testers. And this is required while developing a new build or checking its quality. This process enables the DevOps experts to get new software development & testing infrastructure and avoid human errors.

1.3 Containerizing

After creating the DevOps strategy, containerization is implemented. For this, tools like Docker are used to solve software reliability issues. In this process, containers come with everything that is needed to run an application. Here, various pieces of applications are put together into several containers and this makes it easier for the operations team to update an app as it doesn’t require rebuilding the entire application when there are only smaller changes.

1.4 Infrastructure Automation Integration with CI/CD Tools

When a software application is put into containers, managing the containerized application is very important. For this process, infrastructure automation tools like Puppet, Chef, Ansible, CloudFormation, and Kubernetes are used with CI/CD tools such as Azure DevOps, AWS

Pipelines, GoCD, Jenkins, and Bamboo. This integration enables efficient software deployment and configuration management.

For instance, if the developer uses Kubernetes for large infrastructures and or Ansible for smaller app infrastructures, this enables the team to monitor the container's health, manage containers for fault tolerance, and roll software updates. After that, when it comes to creating, testing, and deploying new builds in the Kubernetes tools, it can be done using Jenkins.

1.5 Test Automation and Aligning QA with Development

The next step is to achieve faster delivery with the help of DevOps. For this, software development teams need to carry out sufficient automated testing. But not each type of testing can be automated; therefore, the team needs to perform usability, exploratory, and security testing manually. This still depends on the efforts required for automated testing. Besides this, the DevOps experts must carry out development and testing activities in tandem to avoid bugs before releasing the build.

1.6 Performance Monitoring

The last step that comes in converting a firm's traditional approach to DevOps, application performance monitoring, is required. It helps in offering transparency over various performance problems to the DevOps teams. And all these problems can be revealed during the processes like user experience monitoring, application server monitoring, and more. Here, application performance monitoring enables isolating, detecting, and prioritizing the defects of the app before the user finds them and for this like Nagios and Prometheus are used.

Microservices Architecture:

The online shopping website will be built using a microservices architecture. This architecture will break down the application into smaller, more manageable components that can be independently developed, tested, and deployed. Each microservice will have its own database and will communicate with other microservices using APIs.

The microservices will be deployed on a container orchestration platform such as Kubernetes, which will provide scalability, high availability, and fault tolerance.

Detailed Report about the results achieved after implementing the DevOps approach:

After implementing the DevOps approach, we achieved the following results:

1. Deployment Time Reduction: The time it takes to deploy new features has been reduced from days to hours.

2. Improved Website Performance: The website's performance has improved due to the use of microservices architecture and container orchestration.

3. Increased Reliability: The reliability of the application has increased due to the automated testing, continuous monitoring, and alerting setup.

4. Fault Tolerance: The reliability will be increased as the fault tolerance will be increased due to multiple datacentres available across the regions.

5. Scalability: The Application can be scaled up and scaled down to meet the requirements of the end users. One of the features which can be used is auto-scaling which can be scheduled to meet the requirements during high time.

Presentation explaining the DevOps approach:

We will present the DevOps approach to the stakeholders by highlighting its benefits, including faster deployment times, improved website performance, and increased reliability. We will explain the workflow and tools used in the approach and how they contribute to achieving the desired results. We will also show a working implementation of the DevOps approach, including the CI/CD pipeline, IaC scripts, monitoring and alerting setup, and microservices architecture. We will also presentA report detailing the results achieved after implementing the DevOps approach, including the reduction in deployment time, improvement in website performance, and increased reliability. Finally, we will address any concerns or questions the stakeholders may have about the approach. Basically, DevOps implementation and following DevOps principles and best practices like IaC, CI/CD, total application monitoring, testing automation, and others can help in changing the entire approach of the organization.