**DevOps Setup**

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Description | Author |
| 1.0 | 30-10-2023 | Initial release | Santosh Kumar Kar |
| 1.1 |  |  |  |

Document Review History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Reviewer | Review Status | Review Date | Approval Date |
| 1.0 | Suja Anian | Pending |  |  |
| 1.0 | Sankar Rao | Pending |  |  |

[**1.** **Introduction** 3](#_Toc149727568)

[Purpose 3](#_Toc149727569)

[**2.** **Environment Overview** 3](#_Toc149727570)

[**3.** **Tools and Technologies** 4](#_Toc149727571)

[SonarQube: 4](#_Toc149727572)

[Jenkins: 4](#_Toc149727573)

[Snyk: 4](#_Toc149727574)

[Verdaccio (Node.js Artifactory Repository): 4](#_Toc149727575)

[BaGet (.NET Artifactory Repository): 4](#_Toc149727576)

[Pulumi (Infrastructure as Code): 5](#_Toc149727577)

[**4.** **CI/CD Pipeline** 5](#_Toc149727578)

[Identify the Version Control System (VCS): 5](#_Toc149727579)

[Implement Branching Strategy: 5](#_Toc149727580)

[Continuous Integration: 5](#_Toc149727581)

[Build the application: 5](#_Toc149727582)

[Continuous testing and security automation: 5](#_Toc149727583)

[Build the artifact: 6](#_Toc149727584)

[5. Integrate with the Code Repo - Pending 6](#_Toc149727585)

[6. Continuous Delivery 6](#_Toc149727586)

[7. Monitoring and Alerting 7](#_Toc149727587)

[8. Security 7](#_Toc149727588)

[9. Backups and Disaster Recovery 7](#_Toc149727589)

[10. Future Scope 7](#_Toc149727590)

[1. Integrate other testing in the CI/CD 7](#_Toc149727591)

[2. Work on the licensing parts 7](#_Toc149727592)

[3. Monitoring & Logging 7](#_Toc149727593)

[4. Feedback and Notification 7](#_Toc149727594)

[11. Timeline 8](#_Toc149727595)

# **Introduction**

## Purpose

This document serves as a comprehensive guide outlining the steps, tools, and processes involved in establishing a robust CI/CD pipeline. It aims to provide a clear understanding for all stakeholders, ensuring effective collaboration and informed decision-making throughout the development and deployment lifecycle.

A screenshot of a computer

Description automatically generated

# **Environment Overview**

1. **Node.js** is Utilized for building scalable and efficient server-side applications.
2. **C#.NET** is used for developing robust and high-performance applications.
3. AWS Code Commit is used as the central repository for version control.
4. Amazon EC2 Instances for the infra setup.
5. Amazon S3 to store the artifacts.
6. Dedicated Build Servers: Configured to efficiently handle continuous integration tasks, ensuring smooth and automated build processes.

# **Tools and Technologies**

## SonarQube:

* *Purpose:* SonarQube is used for static code analysis to identify code quality issues, security vulnerabilities, and code smells. It provides detailed reports to help improve code maintainability and reliability.
* *Application:* It's applied to perform continuous code quality assessments, ensuring that our codebase meets established standards.

## Jenkins:

* *Purpose:* Jenkins is our primary tool for continuous integration and continuous deployment (CI/CD). It automates the build, test, and deployment processes, enabling rapid and reliable software delivery.

*Application:* Jenkins pipelines are configured to handle various stages of development, from code commits to deployment, ensuring seamless integration and release workflows.

## Snyk:

* *Purpose:* Snyk specializes in identifying and mitigating security vulnerabilities in open-source dependencies. It actively scans our codebase and its dependencies to provide timely security alerts and suggested fixes.
* *Application:* Snyk is integrated into our development process to proactively address security concerns and ensure that our applications are built on a foundation of secure dependencies.

## Verdaccio (Node.js Artifactory Repository):

* *Purpose:* Verdaccio serves as a private npm registry for Node.js packages. It allows us to host, manage, and distribute internal packages, enhancing control and security over our Node.js dependencies.
* *Application:* It's used to store and manage proprietary packages and third-party packages that may not be publicly available or require additional customization.

## Amazon S3

* S3 will be used to store the artifacts generated from the node.js and .net code. This will be used by verdaccio to push and pull the npm packages.

## BaGet (.NET Artifactory Repository):

* *Purpose:* BaGet serves as our private NuGet registry for .NET packages. It enables us to maintain a secure and controlled environment for managing internal .NET libraries and packages.
* *Application:* BaGet is instrumental in storing and managing .NET packages, ensuring that our applications have access to the specific versions they require.

## Pulumi (Infrastructure as Code):

* *Purpose:* Pulumi facilitates the definition and deployment of cloud infrastructure through code. It allows us to manage infrastructure resources in a programmatic and version-controlled manner.
* *Application:* We utilize Pulumi to define our cloud infrastructure components, enabling consistent and repeatable deployment processes across different environments.

# **CI/CD Pipeline**

## Identify the Version Control System (VCS):

Status: Done

We are already using the codebase AWS Code Commit

## Implement Branching Strategy:

#### Set up the main branches:

master - Done

develop - Pending

#### Developers: - Pending

#### New Feature:

1. Create feature branches from develop for new features.
2. When features are ready, they are merged back into develop.

#### Releases:

1. Create a release branch from develop.
2. Do any necessary final testing.
3. Merge it into both master and develop.

#### Hotfixes:

1. Create a branch from master.
2. Fix the issue.
3. Merge it back into develop.
4. Merge to master.

## Continuous Integration:

### Build the application:

1. Node JS
2. .NET

### Continuous testing and security automation:

1. Setup Jenkin Server
   1. For NodeJS - Done
   2. For .NET - Done
2. Identify Code Coverage Tool
   1. For NodeJS - Done with ***Jest***
   2. For .NET - Done with Coverlet
3. Setup SonarQube for Static Code Analysis
   1. For NodeJS - Done
   2. For .NET - Done
4. Identify Code Security Scanning

For NodeJS, C# - Done with Snyk

1. Generate Reports and integrate with SonarQube

Snyk Sarif will be used to integrate with SonarQube and show the report in dashboard. - Pending

### Build the artifact:

1. Identify Artifact Repository
   * 1. For Node.JS - Done (Verdaccio for free)
     2. For .NET - Pending (***BaGet?***)
2. Setup Verdaccio in AWS - Pending
   * 1. Setup in EC2 with Docker
3. Setup **BaGet** in AWS - Pending
   * 1. Setup in EC2 with Docker

## Integrate with the Code Repo - Pending

1. **Steps**
   * 1. Developers should raise the PR
     2. When approval is done the PR, code must merge with the Development branch.
     3. Then the Build pipeline must be triggered.
2. **Setup Trigger:**
3. *AWS* *Code Commit* webhooks
4. Any other alternate tools

## Continuous Delivery

- Pending

#### Setup the Deploy Pipeline:

1. Run the Unit Tests
2. Integration Testing
3. Security Scanning
4. and any other vulnerability scanning

#### Setup the infrastructure with Pulumi (to be discussed)

#### Deploy the code

Future Tasks

## Monitoring and Alerting

## Security

## Backups and Disaster Recovery

## Future Scope

### Integrate other testing in the CI/CD

* Integration Testing
* Load Testing

### Work on the licensing parts

### Monitoring & Logging

### Feedback and Notification

## Timeline

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Resource** | **Activities** | **Target Start date** | **Target End Date** | **Status** |
| 1 | Ganesh | Created an EC2 instance (t2.micro) | 11 October | 11 October | Done |
| 2 | Ganesh | Set up done for Jenkins and SonarQube using Docker images | 12 October | 16 October | Done |
| 3 | Ganesh | Integrate SonarQube with Jenkins for code scan. | 17 October | 18 October | Done |
| 4 | Ganesh | Integrate Jest for Node | 17 October | 20 October | Done |
| 5 | Ganesh,  Mansingh | Integrate Coverlet for .Net | 23 October | 26 October | Done |
| 6 | Ganesh,  Mansingh | Integrate Snyk for Security & Vulnerability Scanning | 27 October | 29 October | Done |
| 7 | Ganesh,  Mansingh | Storing artifact for NodeJS | 1 November | 3 November | Pending |
| 8 | Ganesh,  Mansingh | Storing artifact for C# | 6 November | 9 November | Pending |
| 9 | N/A | Working on Branching Strategy | 10 November | 10 November | Pending |
| 9 | Ganesh,  Mansingh | Setting up AWS Codecommit webhook to trigger the pipleline on code | 13 November | 16 November | Pending |
| 10 | Ganesh,  Mansingh | Deploy the code through pipeline | 20 November | 24 November | Pending |