**CS691 – Jenkins Progress Report**

**Team: Team 2**

**Jenkins Admin: Kuldeep Saindane**

**Sprint1: Planned and Completed Tasks**

|  |  |
| --- | --- |
| **Planned Tasks** | **Completed Tasks** |
| **Configuring Required Plugins in Jenkins**   * Install essential plugins like AWS Credentials Plugin for securely managing AWS authentication * Install Slack Notification Plugin for real-time alerts * Install GitHub Integration Plugin for source code automation | * The AWS Credentials Plugin was successfully installed to securely manage AWS authentication. * The Slack Notification Plugin was successfully installed to provide real-time alerts for Jenkins builds. * The GitHub Integration Plugin was successfully installed to automate source code integration with Jenkins. |
| **Integrating and Automating DEV**   * Connect DEV pipeline to the QA instance * Set up GitHub webhook for auto code pulling and deployment from the DEV branch | * The DEV pipeline was successfully connected to the QA instance for automated deployments and testing. * A GitHub webhook was successfully set up to automatically pull code and trigger deployments from the DEV branch. |
| **Jenkins Security**   * Update Jenkins plugins to the latest versions * Identify and fix pipeline scripts vulnerabilities | * Jenkins plugins were successfully updated to the latest versions to ensure optimal functionality and security. * Vulnerabilities in the pipeline scripts were identified and successfully fixed, enhancing security. |
| **Implementing Slack Notifications**   * Integrate Jenkins with Slack to send automatic notifications for build start, success, or failure * Ensure visibility of notifications in the Team2 Slack channel | * Jenkins was integrated with Slack to send automatic notifications for build start, success, and failure. * Visibility of Jenkins notifications was successfully ensured in the Team2 Slack channel. |
| **Jenkins Documentation**   * Document Jenkins setup and plugin usage * Share updated CI/CD process with the team | * Jenkins setup and plugin usage were thoroughly documented for future reference and team understanding. * The updated CI/CD process was successfully shared with the team, ensuring alignment and clarity. |

**Sprint2: Planned and Completed Tasks**

|  |  |
| --- | --- |
| **Planned Tasks** | **Completed Tasks** |
| **Pipeline Optimization**   * Implement deployment stages in Dev and QA pipelines. And try to create a separate pipeline. * Configure conditional triggers   **Automating QA**  Set up GitHub webhook for auto code pulling and deployment to the QA branch.  **Integrate Jenkins with Docker for containerized build environments**  **Performance Tuning**  Identify and fix performance bottlenecks in Jenkins builds (e.g., slow builds, unnecessary steps)  **Jenkins Documentation**  Update Jenkins documentation to reflect Sprint 2 changes and improvements | * Deployment stages were successfully implemented in both the Dev and QA pipelines to streamline the build and release process. * Separate pipelines were created for improved modularity and management. * Setup the Github webhook for auto code pulling and deployment to the Dev, and pending for QA branch. * Jenkins was successfully integrated with Docker to enable containerized build environments, enhancing build consistency and reducing environment-related issues. * Optimizations were implemented by refining build scripts and removing unnecessary stages, resulting in faster and more efficient builds. * Jenkins documentation was updated to reflect all changes and improvements made during Sprint 2. * The team was informed of the enhancements, ensuring clarity and alignment with the updated CI/CD workflow. |

**Sprint3: Planned and Completed Tasks**

|  |  |
| --- | --- |
| **Planned Tasks** | **Completed Tasks** |
| **Set Up Branch-Based Build Control**   * Restrict Jenkins job triggers to specific branches such as Dev and QA for controlled CI/CD execution.   **Enhance Slack Notifications with Commit Messages**   * Include the latest Git commit message in Slack notifications to provide better context for triggered builds.   **Integrate Blue Ocean Plugin**   * Enhance the Jenkins user experience with a visual pipeline interface for better visualization and debugging.   **Configure Jenkins Backup**   * Ensure periodic backups of Jenkins jobs, configurations, and settings to prevent data loss and support disaster recovery using ThinBackup Plugin.   **Conduct Team Training Session**   * Organize a walkthrough session of the Jenkins setup to improve team understanding and readiness. * Gather feedback and make necessary adjustments.   **Finalize Jenkins Documentation**   * Complete and refine all Jenkins-related documentation, including CI/CD workflows, plugin configurations, and security practices. | * Jenkins job triggers were successfully restricted to the Dev and QA branches to ensure controlled and secure CI/CD execution. * Slack notifications were enhanced to include the latest Git commit messages, providing greater visibility into build triggers and facilitating quick context sharing with the team. * The Blue Ocean Plugin was successfully integrated into Jenkins, offering an improved UI and enhanced visualization of pipeline executions, which simplified debugging and monitoring. * The ThinBackup Plugin was installed and configured to perform scheduled backups of Jenkins jobs and configurations, enhancing reliability and disaster recovery readiness. * A comprehensive training session was conducted to walk the team through the Jenkins setup, including pipelines, plugins, and build triggers. * Team feedback was gathered and used to fine-tune certain configurations. * All Jenkins-related documentation was finalized and shared with the team, including detailed notes on CI/CD workflows, plugin use, backup configuration, and security measures. |