ALM-MS

Services:

1. Azure DevOps/TFS
   1. Azure Boards – Requirement and Project Management
   2. Azure repos – Source Control TFVC and Git
   3. Azure Pipelines (CI/CD) – Automated Builds and Releases
   4. Azure Test Plans

TFS/Azure DevOps

Consultation on Azure DevOps/TFS

|  |  |
| --- | --- |
| TFS/Azure DevOps Features | Activities/Actions |
| Boards | * Project Planning – Areas and Iterations * Access Management – User Permissions and Access Groups * Backlog and Task Management – Sprint Planning, WI Import/Export * Project Monitoring and Reporting – Dashboard, Queries, Charts, Widgets. * WIKI - collaborations |
| Azure Pipelines (CI/CD) | * Agent configuration * Continuous Integration - Automatic building and testing whenever new code is committed. * Multi languages support (Java,Python,Dotnet,PHP ,etc) * Scheduled Builds - Triggers CI on timely intervals. * YAML-Based Pipelines- supports YAML syntax * Integration with Third-Party Services -Easily integrates with open source services. * Package Management- Artifacts * Multi-stage deployments -On demand multiple release environments can be created * Parallel Execution: Execution of multiple pipelines at same time |
| Azure Repositories | * Git and TFVC support version control system. * Branching and Merging strategy- Facilitates branching and merging for new feature implementation * Pull Requests and Code Reviews- Supports newly developed code to be reviewed before merging. * Branch Policies -Enables restrictions to the target branches to maintain its integrity. * Repository Permission - Features the settings to control the unnecessary operations on repository * File history and rollback - Provides detailed track of every commit and can rollback them |

|  |  |
| --- | --- |
| SonarQube | Activities/Actions |
| SonarQube | * Static Code Analysis – Identifies detailed coding issues without executing it. * Customizable Quality Profile-: According to languages ,coding standards can be custom define. * Quality Gates - Minimum set of rules ,a code must satisfy for better a performance. * Branch Analysis - Can Implement independent branch analysis * Integration with CI/CD Tools:Easy integration with pipelines * Visualization and Reporting- Displays result of analysis on Dashboard. * Multi-Language Support-Java ,C#,Python,Angular,PHP,Java Scripts and etc. |

|  |  |
| --- | --- |
| Azure DevSecOps Features | Activities/Actions |
| Trivy and Snyk | * **Vulnerability Scanning- S**cans your dependencies, including open source libraries, for known vulnerabilities and provide suggestions. * **Developer Friendly- Has a developer friendly dashboard to display results.** * **Support for Multiple Package Managers-**supports multiple package managers, including APT, Yum, and Apk, making it versatile for different container images. * **Integration with CI/CD- Easily integrates with CI/CD** |

|  |  |
| --- | --- |
| Containerization | Activities/Actions |
| Docker | * Configuring Docker and Docker Compose. * Building Docker images using Dockerfiles. * Pushing and pulling Docker images from Docker Hub or private registries. * Mounting volumes in containers. * Integrating Docker into Azure pipelines . * Automated testing and deployment using Docker. * Deploying and managing containerized applications at scale. * Deploying Docker containers on cloud platforms (AWS, Azure, Google Cloud). |

|  |  |
| --- | --- |
| Azure Test Plans | Activities/Actions |
|  | * Test planning * Test authoring * Test execution * Test reporting (charts, widgets, email) * Exploratory testing * Automation integration with Azure pipelines /Test plans (Selenium Java/C Sharp) |

* Requirement management: Azure Boards
* Project management: Azure Boards
* Source code repository: Azure Repos(Git)
* Unit Testing: NUnit
* Test Case Management: Azure Test Plan
* Continuous Integration(CI): Azure Pipelines
* Continuous Delivery/Deployment(CD): Azure Pipelines
* Package Management: Azure Artifact
* Static Code analysis and coverage tools: SonarQube

1. ALM – Capabilities/ Consultancy

* POCs related to ALM tools and its integrations.
* Azure DevOps Server/Services/TFS
* Azure Test Plans/TFS Test Hub:
  + Test Case Management
  + Defect Management
* DevSecOps integration – Snyk, Trivy
* SonarQube
  + Servers/SonarCloud: Central, standalone
  + Languages:
  + Integrations: Azure Pipelines CI
  + Code Coverage:
* Docker
* Kubernetes

1. Proposal - TFS/Azure DevOps Server/services and integrations with respect to requirements.
2. Training – TFS/Azure DevOps Server/services: Boards, Repos, Pipelines, Test Plans
3. Documentation-
   1. User-guide documents
   2. Best Practices

**You can raise ALM Helpdesk tickets in case of assistance, guidance for any of the below services Provided by ALM team:**

PPT slide explaining ALM Helpdesk