

Pipelines

A **Pipeline** is a set of automated processes that allow developers to consistently and reliably build, test, and deploy their code.

Aras provides the following set of Pipelines, if SDE is referencing `Aras.SaaS.DevOpsFramework.Msi` package in the `/AutomatedProcedures/tools/packages.config` :

1. **ContinuousIntegration**: This **Pipeline** runs a Build given a **Commit** in the Git **Repo**. The system triggers it in one of the following ways:
 - Validation for a **Branch**.
 - Validation for a **Pull request**.
 - Request by a DevOps user.Typically, the Continuous Integration (CI) **Pipeline** is not manually initiated; it is automatically triggered by actions such as **Pull requests (PRs)** and mergers governed by **Branch** policies.
2. **DeployToSIT**: This **Pipeline** creates a Deployment **Package** from a Build, deploys it to the SIT environment, and stores a copy in the **Artifacts** storage.
3. **DeleteInnovator**: This **Pipeline** deletes a given test instance in the SIT environment.
4. **GenerateBaseline**: Generating **Baselines** establishes a starting point and facilitates tracking project changes. This **Pipeline** generates a new **Baseline** and stores the **Artifact** in the **Artifact** storage.

Aras provides the following set of Pipelines for Aras Enterprise customers working with containerized deployments:

1. **ContinuousIntegration**: This **Pipeline** runs a Build given a **Commit** in the Git **Repo**. The system triggers it in one of the following ways:
 - Validation for a **Branch**.
 - Validation for a **Pull request**.
 - Request by a DevOps user.Typically, the Continuous Integration (CI) **Pipeline** is not manually initiated; it is



automatically triggered by actions such as **Pull requests (PRs)** and mergers governed by **Branch** policies.

2. **DeployInnovator_SIT**: This **Pipeline** creates a Deployment **Package** from a Build, deploys it to the SIT environment, and stores a copy in the **Artifacts** storage.
3. **DeployInnovator_UAT**: This Pipeline creates a Deployment **Package** from a Build, deploys it to the UAT environment, and stores a copy in the **Artifacts** storage.
4. **DeployInnovator_STG**: This Pipeline creates a Deployment **Package** from a Build, deploys it to the STG environment, and stores a copy in the **Artifacts** storage.
5. **DeployInnovator_PROD**: This Pipeline creates a Deployment Package from a Build, deploys it to the PROD environment, and stores a copy in the Artifacts storage.
6. **DeleteInnovator_SIT**: This **Pipeline** deletes a given test instance in the SIT environment.
7. **DeleteInnovator_UAT**: This **Pipeline** deletes a given test instance in the UAT environment.
8. **DeleteInnovator_STG**: This **Pipeline** deletes a given test instance in the STG environment.
9. **DeleteInnovator_PROD**: This **Pipeline** deletes a given test instance in the PROD environment.
10. **GenerateBaseline (or DownloadBaseline)**: Generating Baselines establishes a starting point and facilitates tracking project changes. This Pipeline generates a new Baseline and stores the Artifact in the Artifact storage.
11. **SeleniumTestsAicd**: This **Pipeline** to validate commit changes with Selenium tests.
12. **ReleaseInnovator**: This **Pipeline** provides a standardized and automated way to deliver fixes and updates to Aras Innovator SaaS environments while keeping release operations simple, predictable, and governed. The pipeline uses a single, consistent workflow to manage deployments to SIT, UAT, STG, and Production, with built-in approvals for controlled promotion.

