Copado - Version Control and Deployment

Table of Contents
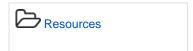
Summary

📂 Resources

Summary

Azure Git is used as Version Control solution for the AbbVie Copado Implementation. This Azure Git is managed by Enterprise BTS Team.

Deployment of the changes to different Salesforce Orgs is handled by Copado. Developers use Copado User Stories to commit/check-in their developed changes to Azure Git, Release Management team will collate the User stories and create Deployment record to deploy the changes to the next Salesforce Org and commit to the associated Azure Git Branch.

Version Control

Version Control - Git Repository in Copado

A **Git Repository** record in Copado stores the relationship between Copado and the version control system to be used (Azure Git). To create a **Git Repository** record in Copado follow the steps below:

1. Navigate to the **Git Repositories** tab.
2. Click on **New.**
3. Give your repository a name.
4. Select a source format.
5. Choose **SSH** or **Https** as authentication type.
6. Fill in the **URI** field and any other fields that may be required depending on the authentication type selected in the previous step.
7. Select a Git provider. You can choose from GitLab, GitHub, Bitbucket or Microsoft Team Service and Copado will populate the URL fields based on the provider selected. If you are working with a different provider, select **Other**: In this case you will need to manually input the URL fields.
8. Click on **Save:**

   blocked URL

Creating and Executing a Deployment

A deployment in Copado is the process of moving one or multiple changes from one environment to another, following a designated **Pipeline**.

Deployment can be created Manually or using Automated Process. In the Automated Process the Pipeline behavior is updated and the User stories matching the criteria are added to automatically Deployment and executed.

Manually, deployments are created using multiple ways, mentioned here is the safe and prescribed method of deployment in case of Manual execution

1. Navigate to the **Selected User Stories** section in the Promotion record.
2. Select the user stories you want to deploy from the list of available user stories.
3. Click on **Create Deployment**:
   - blocked URL
4. A pop-up window will appear next. Here you can choose the test level you want to apply to your deployment.
   a. If you just want to create a validation deployment, flag the **Validation Only** checkbox. Copado will create a promotion branch in your repository, however the deployment step will be flagged as **Check Only**. As a result, no changes will be merged towards the destination branch, and it will not lead to any updates in the environment of the user stories.
   b. If desired, select the **Attach Deployment File** checkbox. See the *Troubleshooting a Deployment* section below for more information about this checkbox:
      i. blocked URL
5. Then, click on **Submit**.
6. When you submit the deployment, the following actions are triggered:

   a. A new promotion branch is created out of the destination branch with the name promotion/{promotion.name-deployment.name}. Once created, Copado merges every feature branch for each of the user stories promoted.
   b. Copado creates a single **Git Promotion** step in the deployment.
   c. If you have specified any manual steps on the user story, these will be added to the deployment.

Once the deployment is completed successfully, the following will happen:

1. The promotion branch will be automatically merged into the destination branch, which is the branch related to the destination org in the pipeline. If the name of the deployment contains:tag=x.y.z, Copado will create and push a Tag x.y.z. Tags cannot be duplicated, therefore the tag is typically added when deploying to production and merging into master.
2. The user stories in the promotion to which the deployment is linked will be moved to the next environment in the pipeline.
3. The checkboxes **Ready to Promote** and **Promote and Deploy** will be unchecked.