

Copado Important Considerations while Building a User Story

In this page, we go through some important points while committing changes and making a user story ready for next deployment:

Before committing the changes (this is normally a one-time thing in copado, only necessary at the beginning)

- Ensure that your logged-in **user has the "Copado User Lite" permission set assigned.**
- Ensure that your logged-in **user has a Copado API Key** set (go to App Launcher Account Summary API Key).
- In the user story where commits will be done, **do not use copado credentials that are not owned by the logged-in user.**
 - Ensure that the user story has a credential that is owned by your user. For example, if I login with "Bob" user, the user story in which I will commit the changes must have the developer org credential that is owned by "Bob" (therefore created by "Bob" user).

While committing the changes

- If you need to commit **destructive changes**, do not mix with common changes, instead create a new Copado ticket (by creating a Jira subtask, check [Jira <-> Copado Integration](#)) and add the destructive changes on it.
- If you need to add changes that cannot be committed (i.e **manual tasks**)
 - First check if it's possible to automate them, and note that Copado has "Deployment Task" records where we can insert automations such as Apex Scripts, Custom settings update, and others, that will be automatically run pre or post deployment of the user story. Check [Copado User Story Deployment Tasks documentation](#).
- If you need to commit a **change in a Profile or Permission set**, perform an incremental change and generally avoid the "Full profiles and permission sets" commit, as this is only for very specific scenarios (such as recently created profile/permsets). Check the copado documentation "[Commit incremental Changes in Copado](#)"

After committing changes, and before deployment

In the following order:

- If you have added Apex code, ensure to add the Apex Tests classes, and ensure that the **minimum coverage** is met by running these tests. After running the tests, the coverage will display in the respective field of the user story.
- Run a **validation deployment** (through "Validate Changes" quick action), and make sure this is successfully validated. If there are any conflicts, you need to solve them by going into the generated Promotion record and clicking the available "Review merge conflicts" button. See [Copado Solving Merge Conflicts](#) for more details. Additionally, if the validation fails, ensure to do the proper changes, commit and re-validate following the same steps.
- Create a **Pull Request** from the validation deployment and have an approval of it. Note that even though the pull request can also be created using the "Create Pull Request" quick action in the user story, creating it through the latest Validation Deployment is more clear, because it will only display the changes from the user story and not other user stories (more in [Pull Request Shows Changes Unrelated to the Components Committed on the User Story](#)).

Copado How-Tos Documentation

<https://docs.copado.com/articles/#!/copado-ci-cd-publication/how-tos>