COPADO | #1 Native DevOps for Salesforce

Copado Secluded

Trial Setup

Learning Objectives

- ➔ Salesforce Trial Org Creation
- ➔ Sandboxes
- ➔ Install Copado
- ➔ User licenses assignment
- ➔ Org credentials
- ➔ Git Repository
- ➔ Git Snapshots
- ➔ Pipeline Configuration
- ➔ Pipeline Configuration Steps
- ➔ Create a Project.
- ➔ Manage Branches
- ➔ Optional: Enable impersonation right to admin user for Copado support.
- ➔ Conclusion

As you work through this trail, please let us know if you run into any issues. We will be working together through this process so you can get the most out of your Copado Trial experience.

COPADO | #1 Native DevOps for Salesforce

Salesforce Trial Org

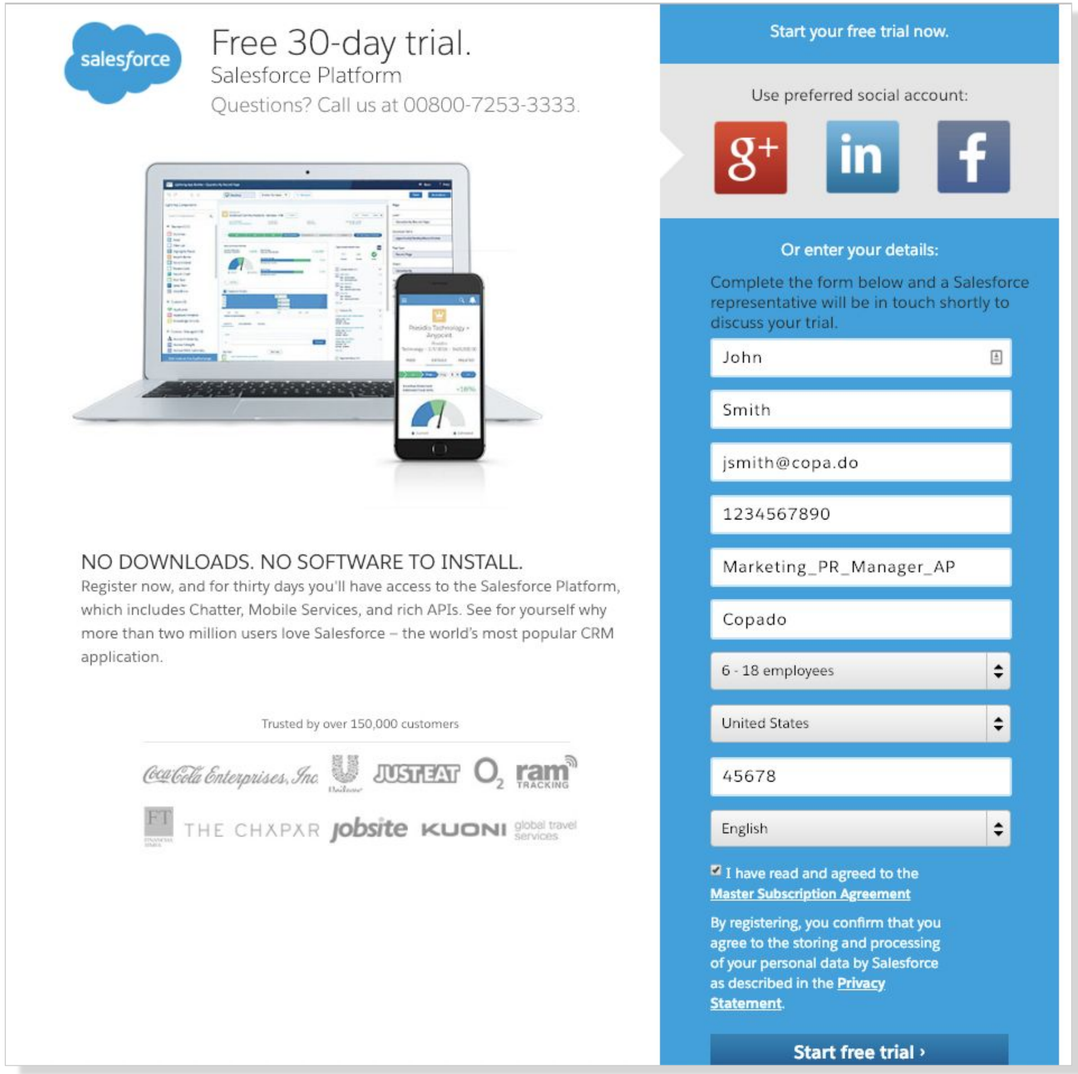
First, we need to create an Enterprise Edition Salesforce Trial Org.

- Please, click on the following link and follow the steps to create a Salesforce Trial Org: https://www.salesforce.com/uk/form/signup/freetrial-platform/

- Please fill out the fields in the form and click “**Start free trial”**

  - Email: in the shape firstname.last+whatever@domain.com based on an email firstname.lastname@domain.com is supported for RFC specification for email such as Gmail

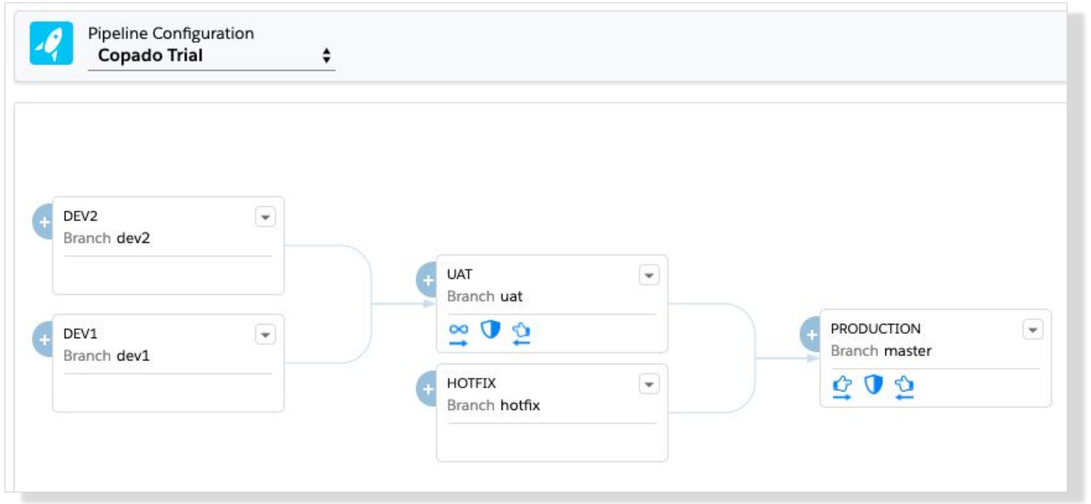
- You will receive an email from Salesforce to set the password for your user shortly.



Sandboxes

Once you have access to the Salesforce Trial Org you created, we will need to enable My Domain and create Sandboxes for each of the environments shown in the Deployment Flow below. This can be accomplished with the following steps:

1. Instructions to enable My Domain can be found here.
    a. Since My Domain must be unique, we recommend a naming convention such as {***YourOrgName***}**CopaTrial** (ex. *NelsonBankCopaTrial*)
2. Navigate to **Setup > Environments > Sandboxes > New**.
3. Name the first Sandbox “Dev1”, provide a description, ensure that you’re cloning from Production and click “Next” under Developer Sandbox Licenses.
    a. Please ensure there aren’t any spaces in the sandbox name.
4. Leave Apex Class empty and click “Create”.
5. Follow the same steps for “Dev2”, “UAT”, and “Hotfix”.



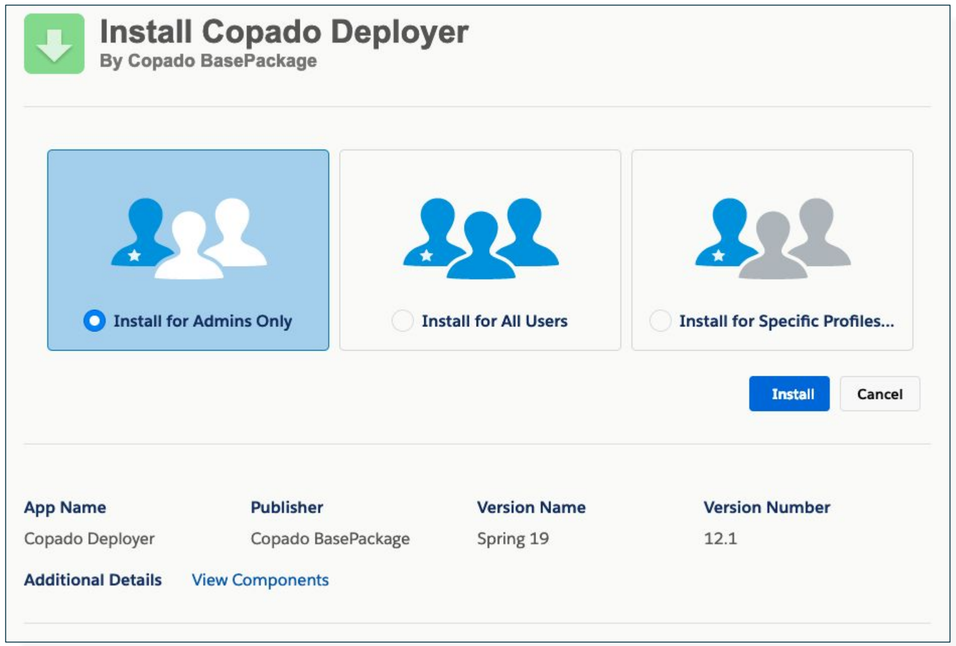
Example of Pipeline

Install Copado

Once we have sandboxes created, we are going to install Copado.

Our recommendation is to “Install for Admins Only”, since this option allows you to grant access to the profiles or permission sets of your choice later on.

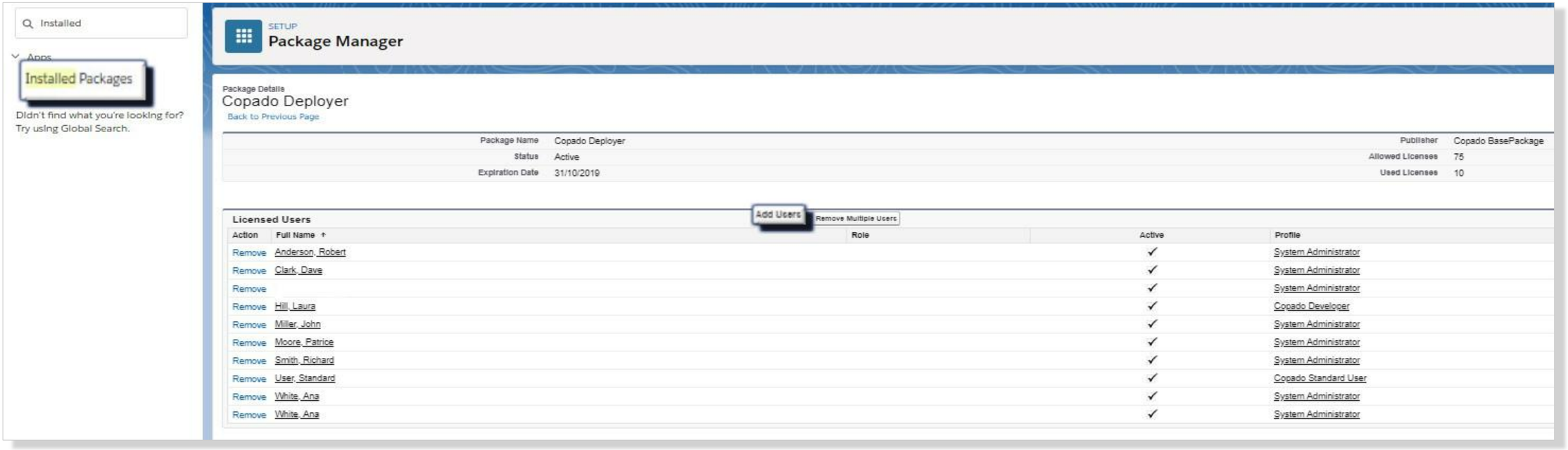
You only need to install Copado in the production org from the following link (make sure you’re logged in only to that Production Environment within your browser session):



[Production/Development installation link]

Install Copado > Managed Package Licenses > Add Users

If you want to add more users to work with Copado, you need to assign them Managed Package licenses to the Copado package. In order to do so, go to: **Setup > Installed Packages > Copado Deployer > Manage Licenses** and click on the “**Add Users**” button.

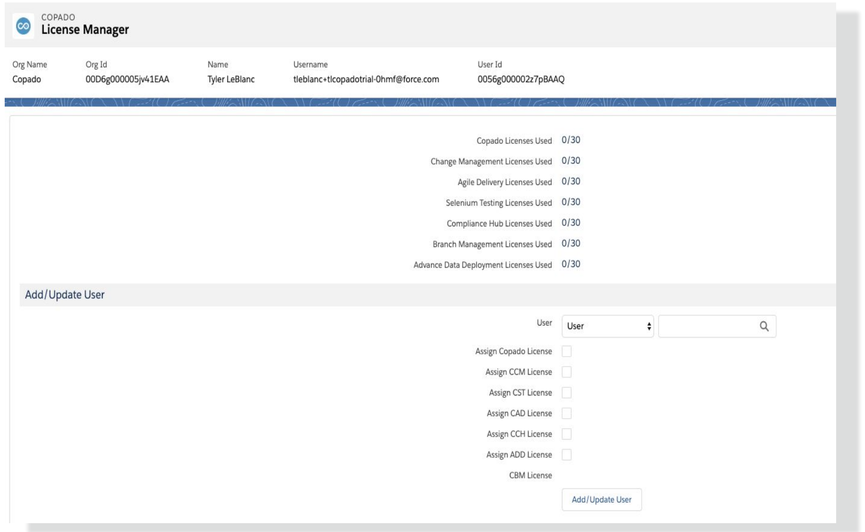


If you don’t have enough licenses to assign, please talk to your Copado representative to assign more seats to your org.

Copado Licenses

1. Go to “**Copado Release Manager**” Application
2. “**Getting Started**” tab
3. **Account Summary** subtab
4. Click on “**Assign Licenses”** and make sure that each Copado user has the appropriate licenses assigned based on the Trial expectations that have been agreed upon with your Copado Account Executive.

   a. At a bare minimum, “**Assign Copado Licenses**” and **“Assign CCM License”** must be selected to test Copado Change Management.
   b. Note that **“CBM Licenses”** are associated automatically with each Deployment Flow Step created from an active Deployment Flow.
   c. **“Assign CST License”** is for Copado Selenium Testing.
   d. **“Assign CCH License”** is for Copado Compliance Hub
   e. **“Assign ADD License”** is for Advanced Data Deployer

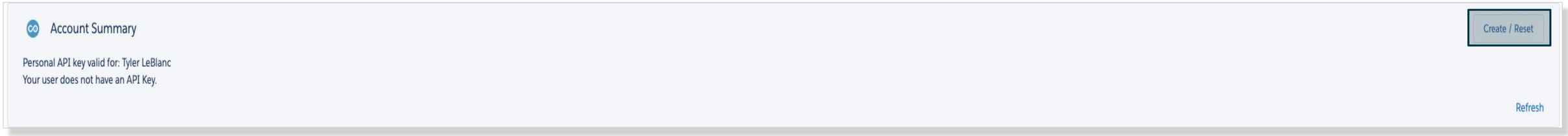


CST and CCH will only be available if you have discussed testing these features with your Copado Account Executive.

Copado Licenses (continued)

1. Go to the **Account Summary** tab
2. Click on the **Create/Reset** button to create the API Key.

    a. This API Key will be used for the Copado Webhooks functionality. These Webhooks will integrate some of the actions done in the repository (Pull Requests, etc.) with Copado.



Lastly, be sure to add the **Copado User** permission set to all users who will be working within the trial.

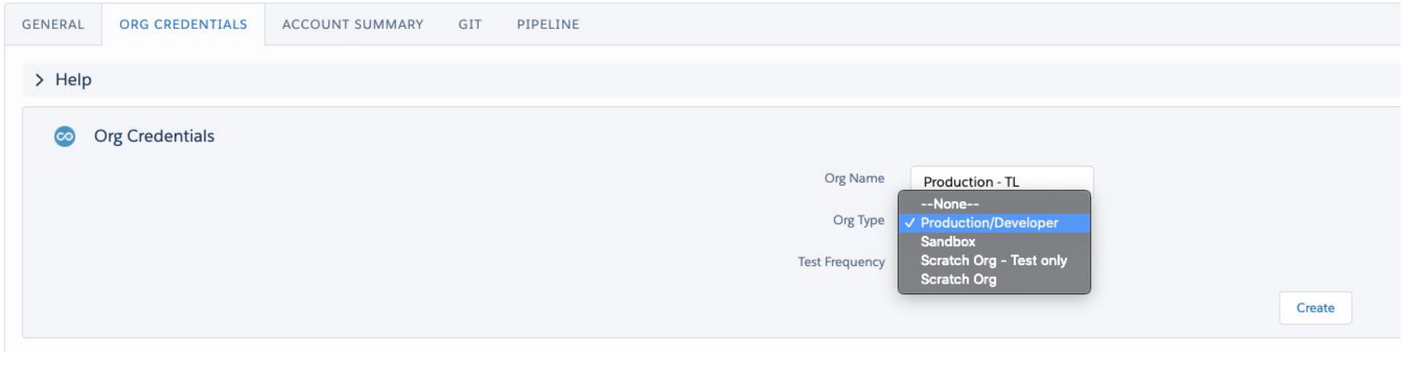
Org Credentials

You have to create an Org Credential for each environment (Production and Sandboxes) which is part of the Deployment Flow (e.g. Dev1, Dev2, Integration, UAT, Hotfix, Production) and for every user using Copado to deploy metadata. These Org Credentials will be used when assigning work to DevOps Resources, so we recommend adding the User's initials to the Org Credential (i.e "Production - TL) to allow clear assignment.

1. From the **Getting Started tab**, click on the **Org Credentials** Subtab.

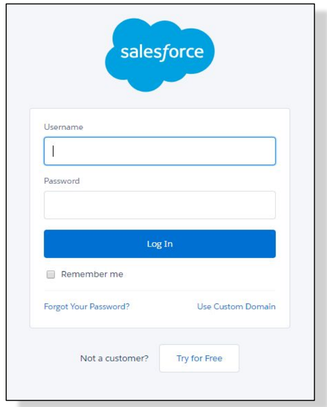
> ***Important**: The first Org Credential created should always be the one representing the primary environment where Copado is installed.*

2. Introduce an Org Name, select the Org Type (Production/Developer, Sandbox) and leave the **Test Frequency** as None.



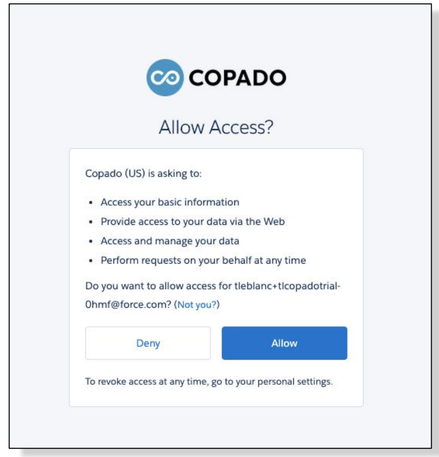
Org Credentials (continued)

3. After the Org Credential record is created, a new row appears in the section below with the Org Credential details



4. Click on the “**Authenticate**” button and login with the credentials of the org to connect it to Copado.

5. Click **Allow**



6. Then Save the record, leave other fields empty

Git Repository

Now, we will integrate Copado with a GIT Repository. This enables your projects to have version control of the Salesforce metadata changes. Version control software allows you to have “versions” of a project and its metadata, which shows the changes that were made to the code and configuration over time. It also enables you to rollback if necessary by undoing the changes made.

- This ability alone – of being able to compare two versions or reverse changes, makes it fairly invaluable when working on medium and large sized projects.
- Git is the Industry standard for software development and is used from open source projects such as Google Android to the Microsoft Windows code base.
- If you do not have a Git Repository, you can open an account in GitHub, Gitlab, Azure Pipelines (formerly Microsoft VSTS) or Bitbucket, where you can create free private repositories.
- Copado also works without version control although you will miss out from advanced functionality.

Git Repository (Continued)

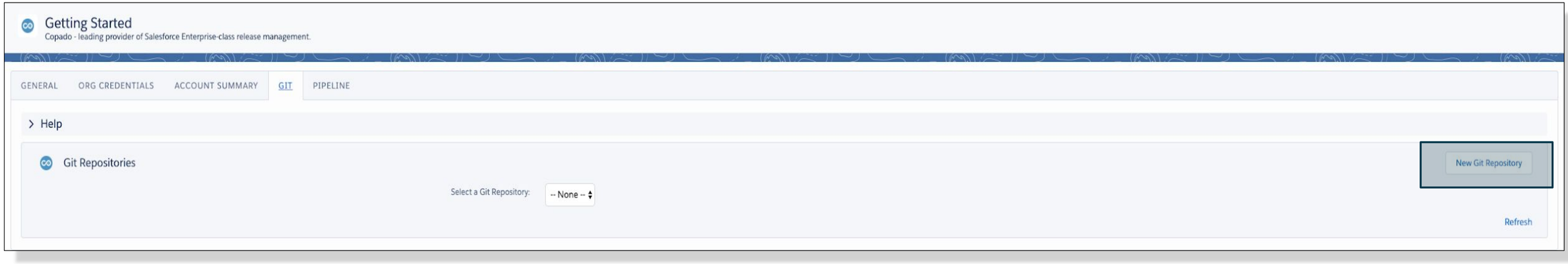
- Setup a new, clean Git Repository with your current Git service provider (GitHub, Gitlab, VSTS Azure, etc.).
- If your Git repository currently sits behind a firewall, please use the following link for connectivity instructions: https://docs.copa.do/copado-getting-started/ip-addresses.
- If the first Git repository option above is not viable, please setup a new free - and private - Git Repository from any of the following providers:
    a. GitHub
    b. GitLab
    c. BitBucket
    d. Azure Pipelines (former Microsoft VSTS)
- Commit this .gitignore file in the master branch to ignore unnecessary files in your Git branches. The gitignore file should be located in the root of the repository (not inside a folder).

All the components that are included in the .gitignore file will not be included in the Copado Release Management Process. Excluding unnecessary metadata will improve the performance of most of the operations such as: Commits, validations, deployments, snapshots,...
If you need to include one of those which are already in the file, comment the line where the component is referenced.

Git Repository (Continued)

1. Go back to **Copado Release Manager** and in the **Getting Started t**ab click on **Git**

2. Click **New** **Git Repository** and create a new record with name “**Copado Trial Repository**”.



Git Repository (Continued)

1. Now, open your newly created Git Repository Record and Select a Git provider from the options in the drop-down menu (Github, Bitbucket, Gitlab, Microsoft Team Service, Azure)

2. Fill in the Branch Base URL, Commit Base URL, Pull Request Base URL, and Tag Base URL based of the format found here

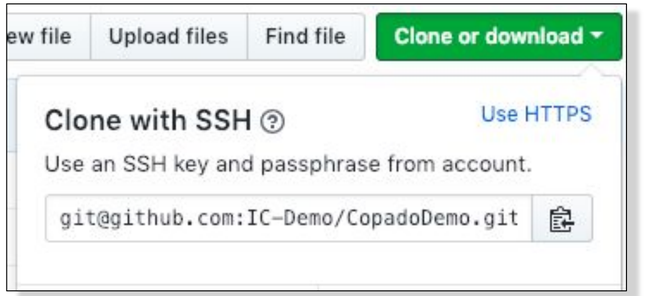
3. Below, you will see the before and after of how your URLs should be configured. Additional information on configuring your Git Repository can be found here.

Before

After

Git Repository (Continued)

Copado can connect to your Repository via SSH or HTTPS authentication. For both of them, your Git Repository URI is required.



Go to git and access the repo. URI is displayed when clicking on the “clone” button, like this:

You can then copy the URI and paste it in the Copado “Git Repository” record, according to the authentication option you plan to use.

1. **SSH**: After the Git Repository record Creation, you must create an SSH API Key in Copado:
    a. Click on the “Create SSH Keys” button in the layout
    b. Scroll down to the “Current Keys” section and click on “View”. Copy the key content and add it to your repository SSH Keys settings.
2. **HTTPS**: Enter the username and password and save the record.

When the connection has been made successfully, the following message appears:

Git Snapshots

1. Go back to the Copado Getting Started tab and click the “Git” subtab. Select the “Copado Trial Repository” from the picklist. If it is not there, click the Refresh link.

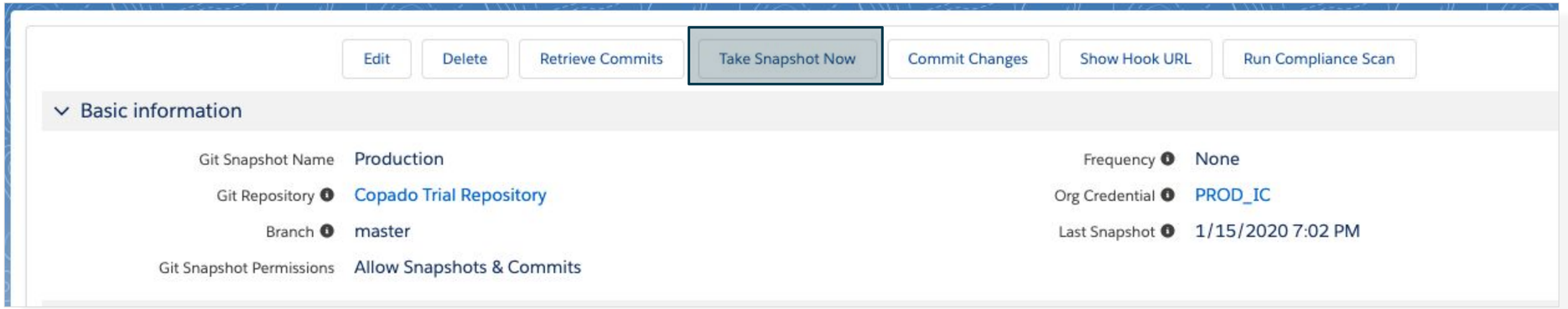
2. Click New Git Snapshot and name it “Production Snapshot”.
3. Type “master” in the branch field.

This is case sensitive so please ensure that it is a keyword match to the branch name in Git.

4. Select the Snapshot permission “Allow Snapshots & Commit”.
5. Leave the Frequency as None.
6. Select the Org Credential that corresponds to the Production environment.
7. Click Create.

Git Snapshots (Continued)

8. Open the Production Snapshot Record.
9. Click ”Take Snapshot Now” to make a full copy of the Production Org metadata in the “master” branch.



10. Once the Snapshot is finished, login to the Git Repository to verify that the metadata has been committed in the master branch. Then create a new branch out of master and name it “uat”. Then create two new branches out of uat, one named “dev1”, and the other “dev2”.
11. Go back to Copado
12. Create 3 more Git Snapshot records for the UAT, Dev1 and Dev2 orgs.
13. For these Git Snapshots set the Snapshot permission to “Allow Commits Only”, type the corresponding branches (uat, dev1, dev2) and select the corresponding Org Credentials.
14. If you created a Hot Fix environment please follow the same steps.

Pipeline Configuration

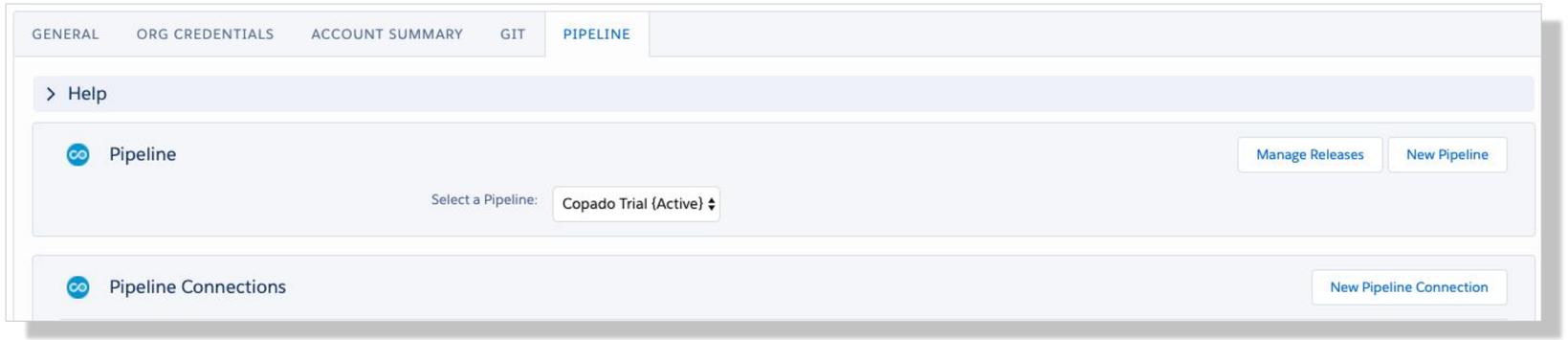
1. Go back to “Getting Started” tab and click on the “Pipeline” subtab.

2. Click on **New Pipeline.**
3. Name it “Copado Trial”
4. Search for the Git Repository we had just created and select it.
5. Type “master” in the “Main Branch” field.

This is case sensitive so please ensure that it is a keyword match to the branch name in Git.

6. Check the Active checkbox.
7. Click **Create**.

Pipeline Configuration (Continued)



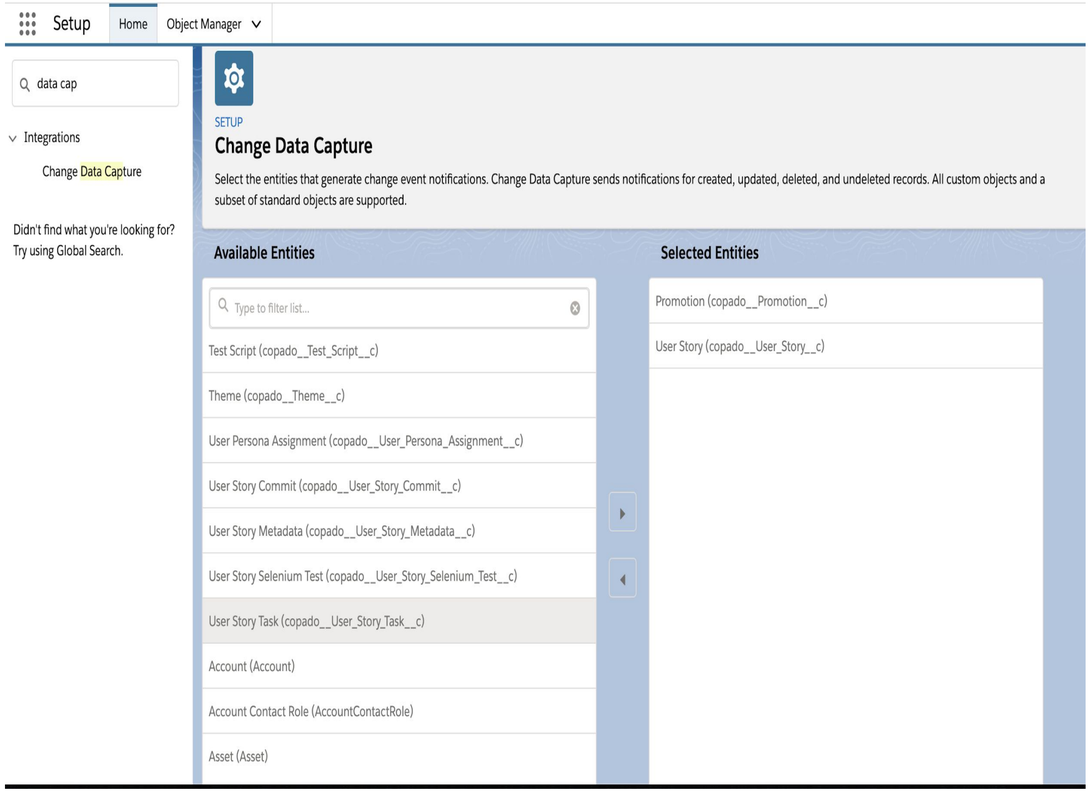
8. Click on **New Pipeline Connection** (make sure your Pipeline is selected). Each step links a source Org to a target Org following the deployment order that you want to achieve (Dev1/Dev2 => UAT => Production).

9. Create 3 different Pipeline Connections as follows:
   a. Source Environment: UAT, Destination Environment: Production, Branch: uat
   b. Source Environment: Dev1, Destination Environment: UAT, Branch: dev1
   c. Source Environment: Dev2, Destination Environment: UAT, Branch: dev2
   d. Source Environment: Hotfix, Destination Environment: Production, Branch: hotfix (Optional)

Change Data Capture

In order for the CI/CD portion to work in Trail 2, we will need to add the Copado User Story Object to the Change Data Capture Setup. To do so, follow the steps below:

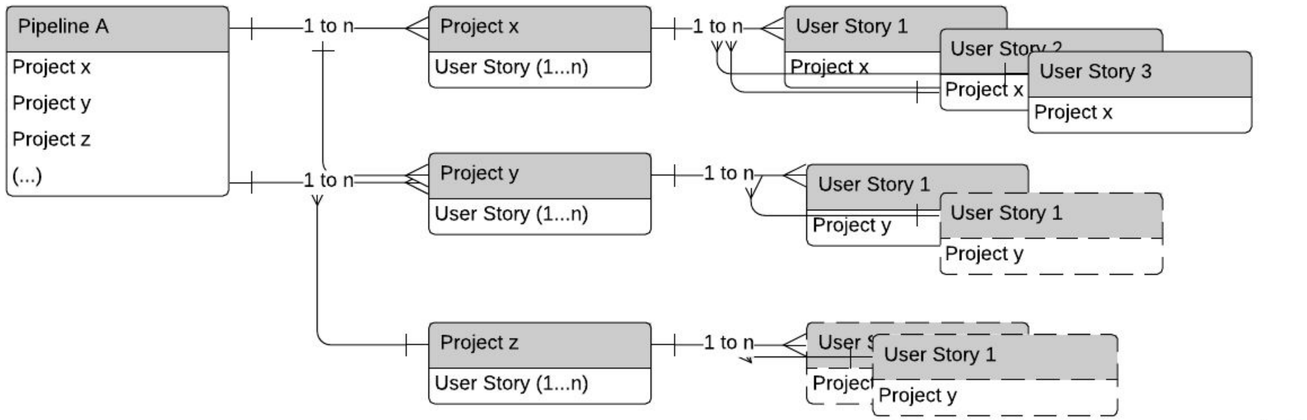
1. Navigate to **Setup**
2. From the search, enter “**Data Capture**”
3. Select **Integration > Change Data Capture**
4. Add to the Selected Entities:
   a. “User Story (copado_User_Story__c)”
   b. “Promotion (copado_Promotion__c)



Create a Project

A User Story in Copado includes detailed information about “what needs to be done.” Within the User Story, all metadata changes are committed to the Git Repository, Quality Gates can be invoked and reviewed, and the User Story related lists provide a comprehensive audit of all actions occurring in a development life cycle path as the User Story moves through the Release Pipeline.

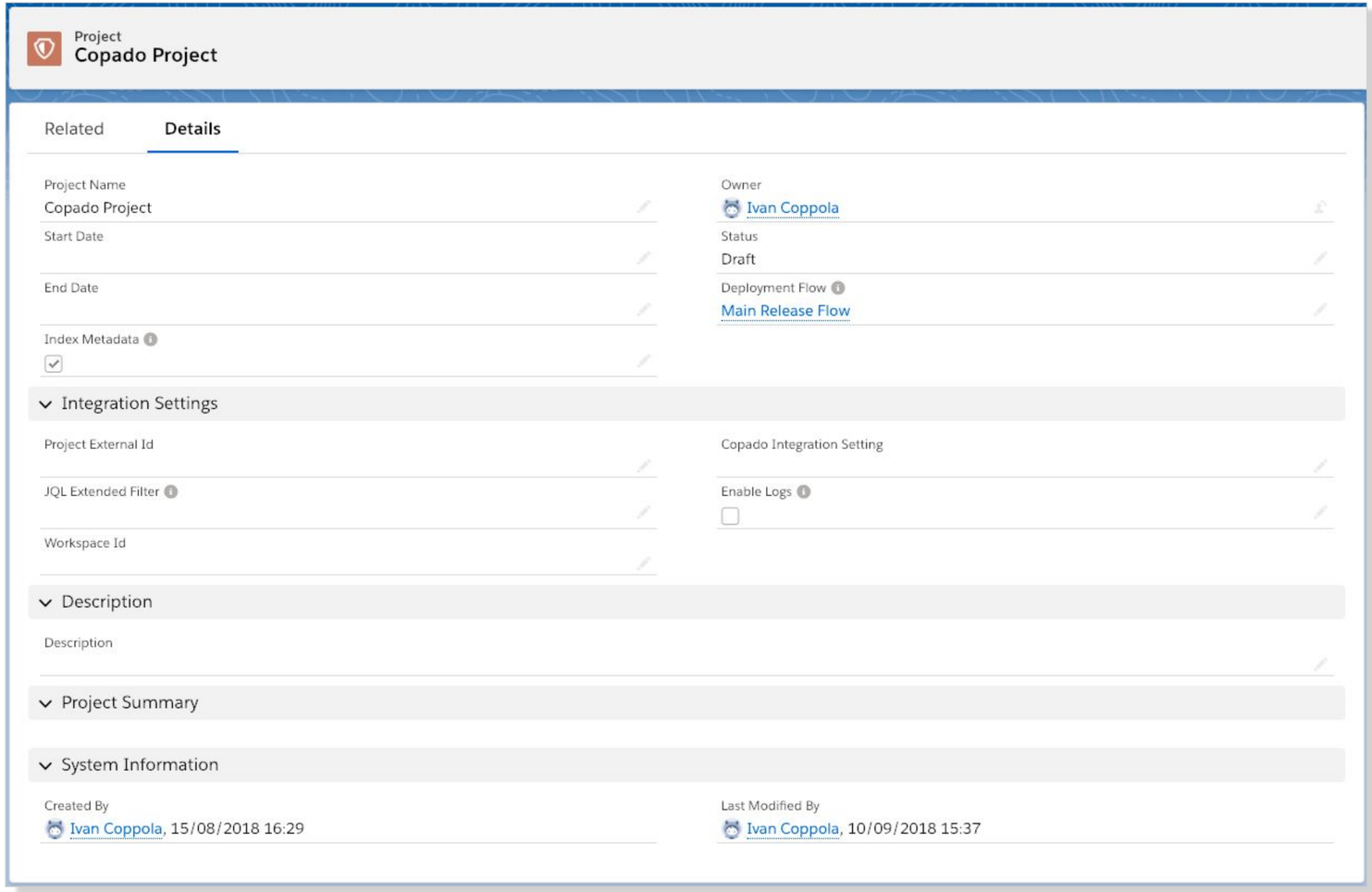
A Project in Copado is the parent object to User Story. A Project can contain one or more User Story, defines the Pipeline for its User Stories, can have Sprints and Releases associated to it and will be the central object for all Copado-ALM Integrations.



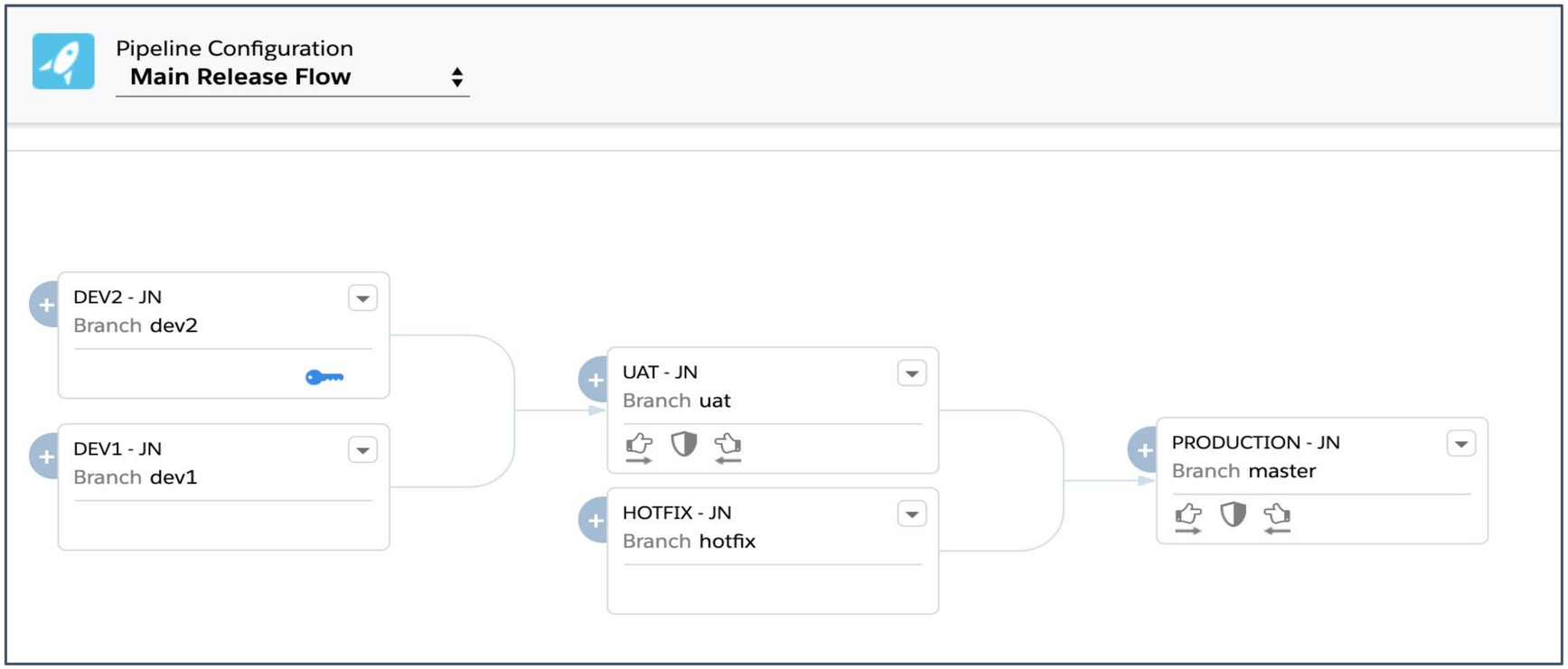
Create a Project

Open the **Copado Developer** application and click on the **Projects** tab and create a new Project record:

- Name: **Copado Trial**
- Index Metadata Checkbox: **true**
- Deployment Flow: Select the Pipeline created during the setup (“Copado Trial”)



Manage Branches



Congratulations! You have just set up Copado to start migrating Metadata from the lower environments all the way to Production.

Assign Static Code Analysis permissions to the System Administrator profile

Go to Setup > Profiles > System Administrator > Object Settings

Assign both the “PMD” and “CodeScan” record types to the following objects. Make sure PMD is selected as **“Default Record Type”**

- Static Code Analysis Results
- Static Code Analysis Settings
- Static Code Analysis Violations

Record Types	Page Layout Assignment	Assigned Record Types	Default Record Type
--Master--	Static Code Analysis Result Layout	☐	☐
CodeScan	CodeScan SCA Layout	☑	☐
PMD	Static Code Analysis Result Layout	☑	☑

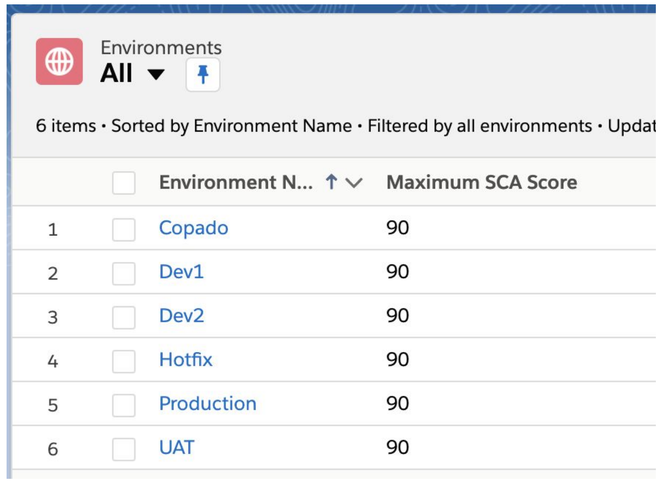
Example on the “Static Code Analysis Result” object

Set “Maximum SCA Score” field on Environments records

In the App Launcher, look for “Environments”.

If you have list view inline editing available, you can add the “Maximum SCA Score” field to the list view and update them all at once. Otherwise, open all the environment records and set the value to 90 and save.

This will define a default value so when running SCA scans, you don’t get an error message.



Optional: Account Login Access

Account login access can be granted in order for Copado to provide you full support during this trial phase.

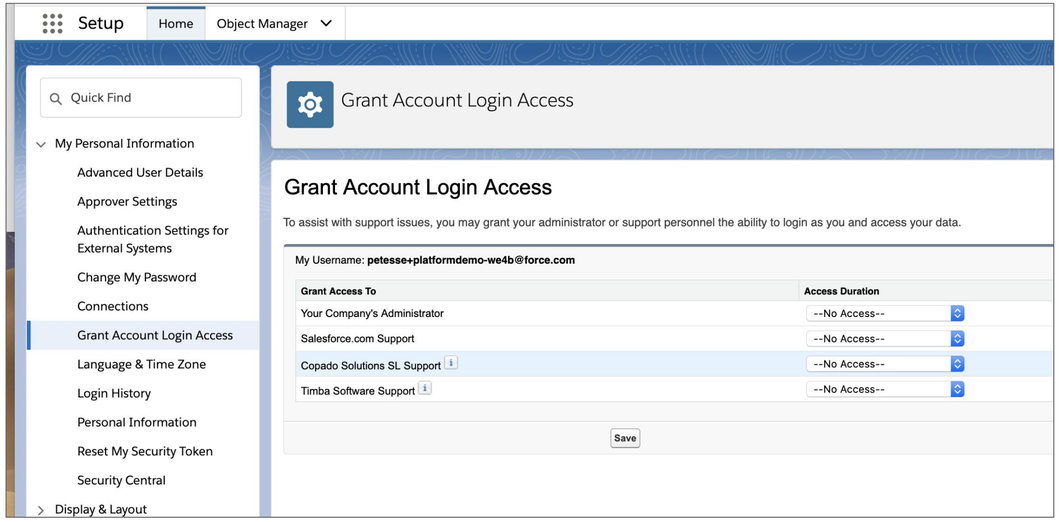
1. **In Lightning**: Go to your **profile picture > settings > personal information > Grant Account login access**

2. **In Classic**: Your **name > my settings > personal > Grant Account Login Access**

3. Click on **Copado Solutions SL. Support**, **Access duration** drop down

4. Select the **duration to 30 days**

5. Click **save**



Conclusion

Copado has User Story records to gather the functional requirements and link them with the metadata components which are modified for each requirement.

User Stories can be imported from other agile applications (like Azure DevOps (VSTS), JIRA, Agile Accelerator, etc.) into Copado so that you can deploy each user story individually or together in a release.

User Stories can be imported from other agile applications (like Azure DevOps (VSTS), JIRA, Agile Accelerator, etc.) into Copado so that you can deploy each user story individually or together in a release. The import of user stories is done via an unmanaged package that you can install from here.

You can also use Copado as your agile application and thus simply create the user story records manually.

Now, we can continue with the Trial Trail “Build, Commit & Deploy Trail” where we will deploy metadata changes using the Copado user stories.