# Tekton Control Panel

* [Authentication](#TektonControlPanel-Authentication)
* [Projects and Repositories](#TektonControlPanel-ProjectsandRepositor)
  + [Scan for Repositories](#TektonControlPanel-ScanforRepositories)
  + [Repositories Overview](#TektonControlPanel-RepositoriesOverview)
  + [Repository Details](#TektonControlPanel-RepositoryDetails)
* [Start Builds and Releases](#TektonControlPanel-StartBuildsandReleas)
  + [Start a Branch Builds](#TektonControlPanel-StartaBranchBuilds)
  + [Create a new Release](#TektonControlPanel-CreateanewRelease)
  + [Parametrized Builds (from Version 0.25.x)](#TektonControlPanel-ParametrizedBuilds(f)
* [Monitor Pipeline Runs](#TektonControlPanel-MonitorPipelineRuns)
  + [Pipeline Run Details](#TektonControlPanel-PipelineRunDetails)
* [ESTA Cloud Pipeline Migration Tool](#TektonControlPanel-ESTACloudPipelineMig)
* [User Permissions](#TektonControlPanel-UserPermissions)
* [Pipeline Secrets](#TektonControlPanel-PipelineSecrets)
* [System Configuration](#TektonControlPanel-SystemConfiguration)

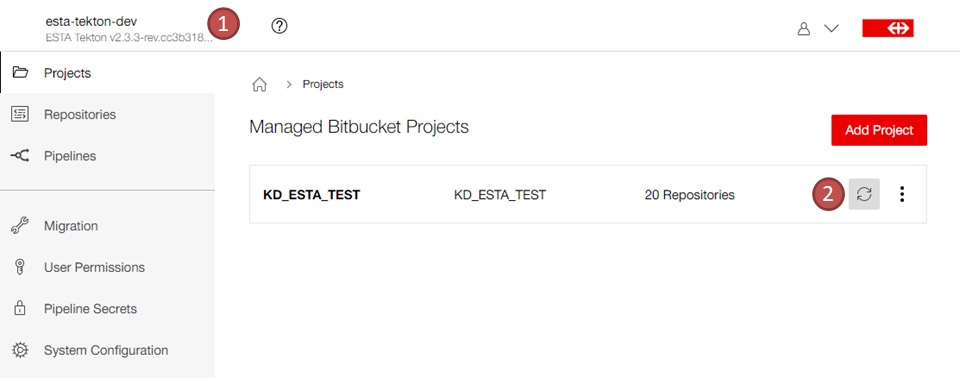
The Esta Tekton Control Panel is a UI to manage pipeline runs for your tekton-enabled repositories.   
If not configured otherwise, the URL looks as follows: https://tekton-control-panel-<your-build-namespace>.sbb-cloud.net and is provided after the setup of the Tekton build namespace.

## Authentication

When opening the Tekton Control Panel, you'll be automatically authenticated with Azure AD ([login.microsoftonline.com](http://login.microsoftonline.com)).  
You can individually give access to your Tekton Control Panel through RBT groups.

## Projects and Repositories

Your ESTA Tekton system is configured to scan Bitbucket projects for repositories with a estaTektonPipeline.json file.  
The projects registered in your build namespace are listed here. The control panel also displays the build namespace name (1).

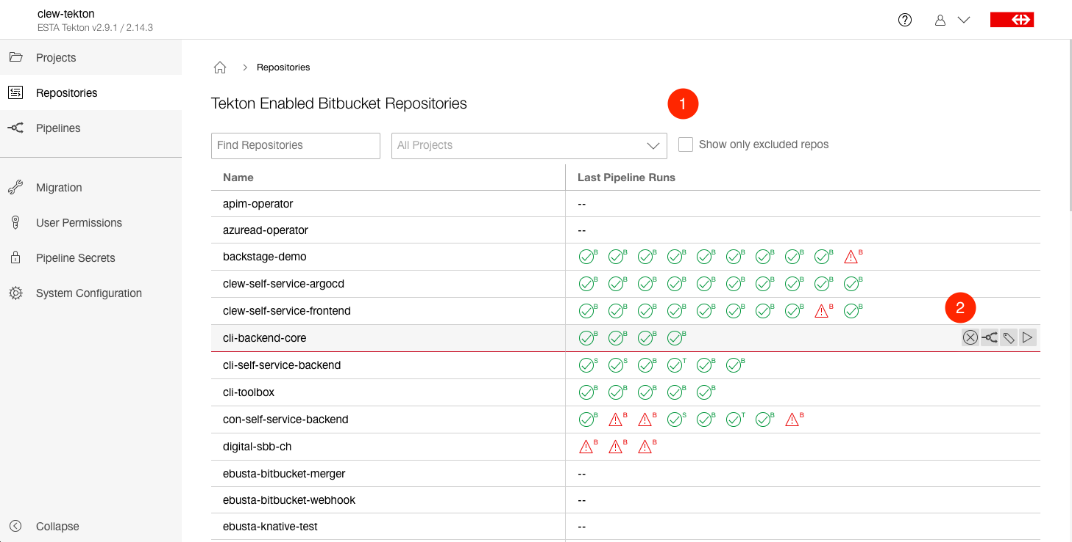


### Scan for Repositories

You can enforce a re-scan of your projects by clicking the sync button (2). This might detect new repositories which are now tekton-enabled.

### Repositories Overview

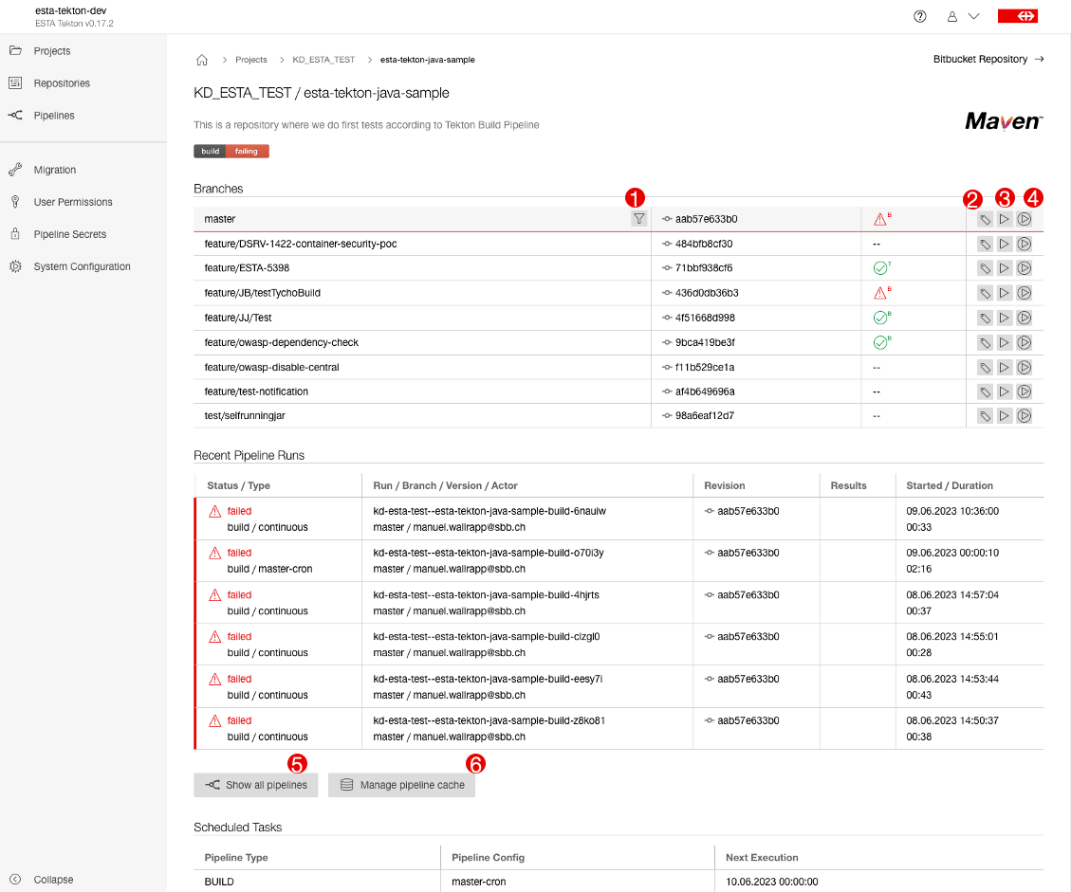
When clicking a project in the list or by selecting the Repositories menu on the left, all tekton-enabled repositories are listed.  
The list can be filtered using the selectors above the table (1). Some actions can directly be chosen from the options menu (2) in each row.



1. You can filter repository by name or project. In case you filtered out some repositories by regular expression from building and listing, you can select the checkbox to show the filtered out repositories.
2. This repository actions can do:
   1. Filter Repository from the list, beeing built, or listed
   2. Show Pipelineruns
   3. Create a tag
   4. Run a build on branch or tag

### Repository Details

Click a row in the listing to open the repository details view with more information and actions to execute on that repository.



The view lists all branches of the particular repository and the most recent pipeline runs.

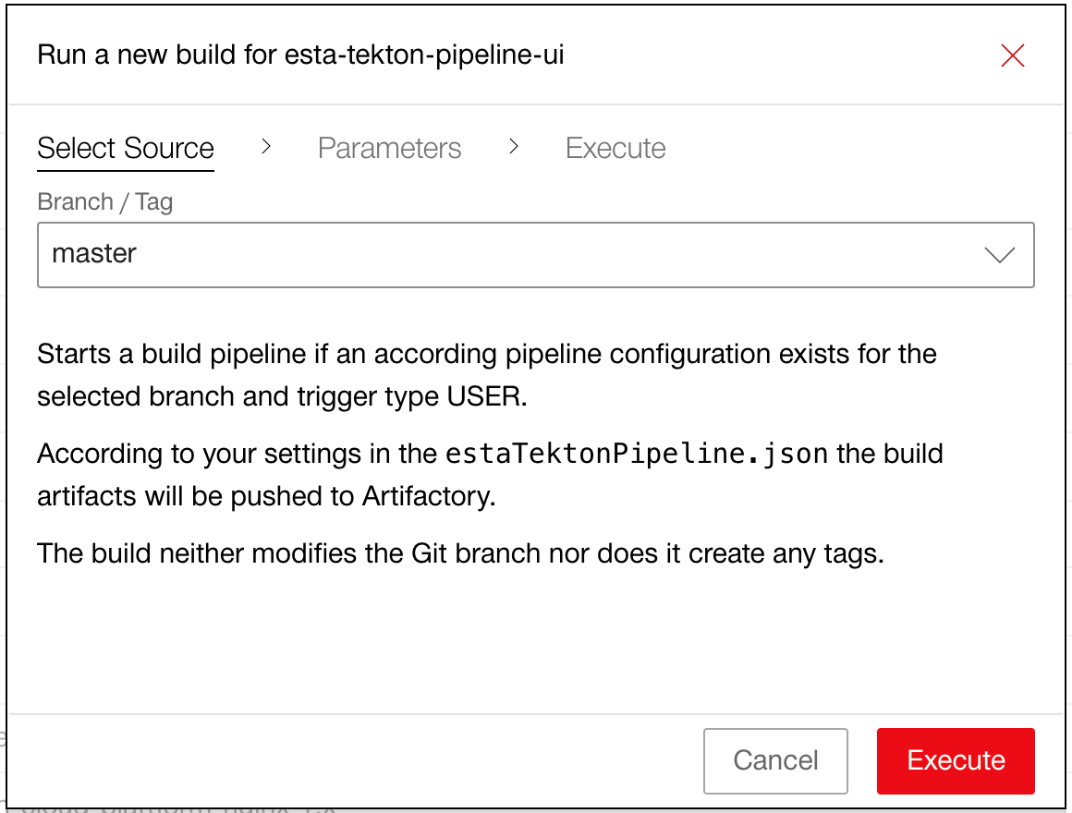
These actions are available for the selected repository:

* (1) Filter the recent pipeline runs to match the selected branch
* (2) Open the tagging assistant to create a release from the according branch
* (3) Start a build for the selected branch with the dialog
* (4) Start a build for the selected branch without the dialog
* (5) Switch to the "Pipelines" view with this repository selected
* (6) Manage the pipeline cache, here you can delete the cache if necessary

## Start Builds and Releases

### Start a Branch Builds

Build pipelines can be started on a specific branch using the "Run build" button.  
From the dialog, simply select a branch (or a tag) as source and press the "Execute" button to start.

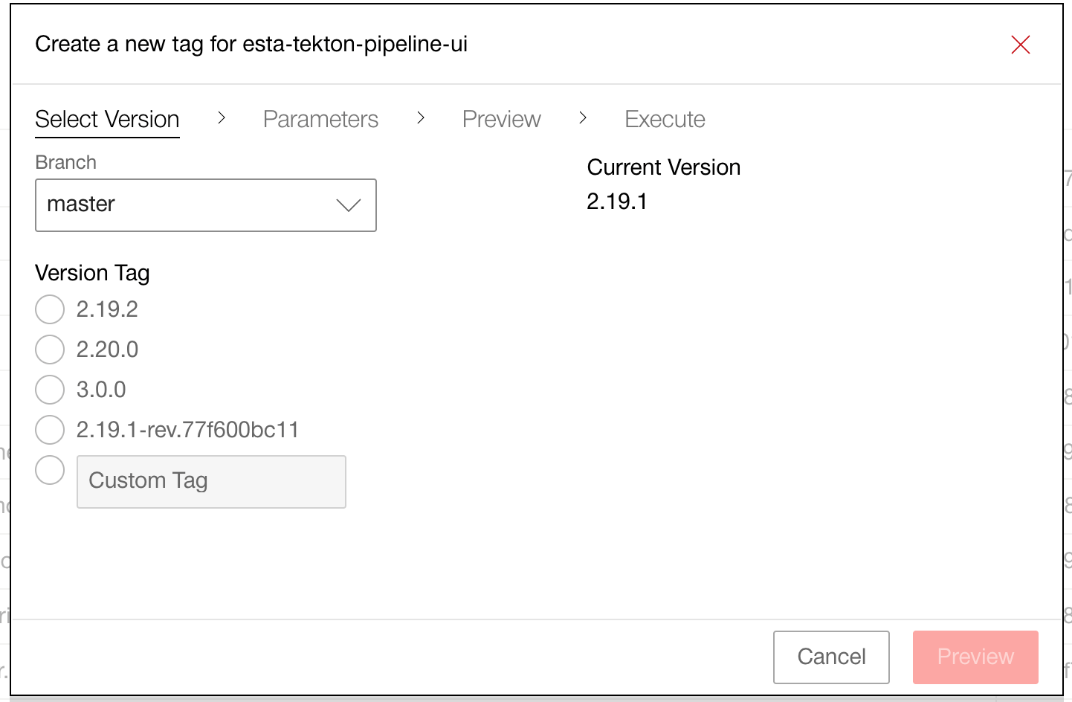
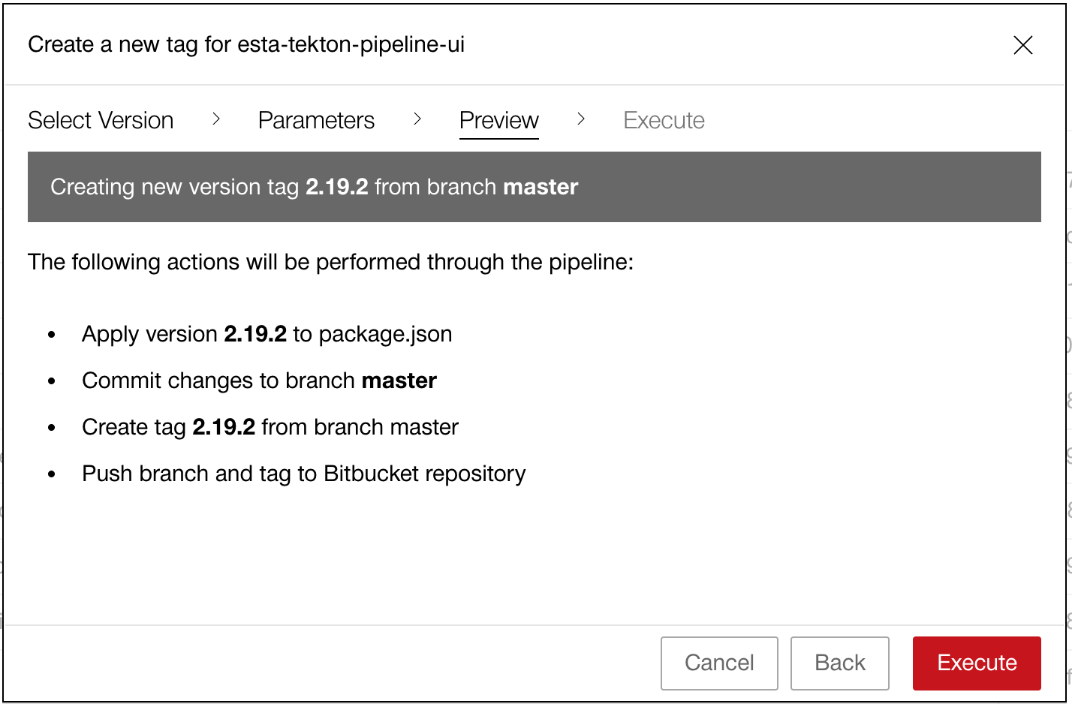


Pipeline configuration required!

A build pipeline is only started if there's a [pipeline configuration](file:///C:\display\CLEW\ESTA+Tekton+Parameters+-+estaTektonPipeline.json#ESTATektonParametersestaTektonPipeline.json-Pipelineconfigurationstructure) in your estaTektonPipeline.json that matches the selected branch (→ branchNamePrefixes) or tag (→ versionTagEventPatterns) and includes the triggerType "USER". Otherwise the execution aborts with the error message **No matching pipeline config for the given PipelineEvent.**

### Create a new Release

As described in the [ESTA Tekton Workflow](file:///C:\display\CLEW\ESTA+Tekton+Workflow#ESTATektonWorkflow-ReleaseProcess) section, releases are always based on a Git tag. The Tekton Control Panel includes a guided process to create such a release tag. Click the "Create new tag" button on the desired branch to create a release from.

The tagging assistant will first read the current version (1) from meta files like pom.xml or package.json.  
It then suggests possible version tags (2) based on the semver pattern. There's also the option to enter a custom tag. If the custom tag already exists, you will not be able to continue the tagging.  
For Maven-based projects, the assistant also suggests the next development version (3) to be set to the pom.xml file in the selected branch.  
Click "Next" to see a summary of steps the tagging pipeline will take. Press the "Execute" button to start the tagging pipeline.

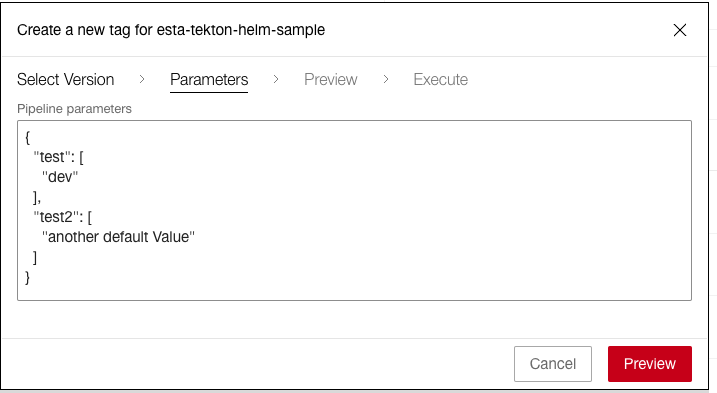
If configured accordingly in estaTektonPipeline.json (→ versionTagEventPatterns), the creation of a Git tag will itself trigger a build pipeline to build the release artifacts.

### Parametrized Builds (from Version 0.25.x)

Parametrized builds are now supported. You can start a Tagging Pipeline with Pipeline Parameters, which will be then propagated to each pipeline in the chain. From Tagging → Build → Staging.

Also if you build a tag, the controller try to find a previous tagging pipeline and read the parameters from there to display them in the UI. You can then edit the parameters for executing the build. The Staging Pipeline then reads the parameters from the preceding Build Pipeline.

In the screenshot you see an example from a tagging pipeline with Pipeline Parameters configured. At the moment you need to edit the parameters as Json, soon we will deliver an UI for it.



To see the "Parameters" step, you need configure the parameters block in your estaTektonPipeline.json. Here you can find an example.

"parameters": [

{

"name": "test",

"type": "String",

"values": ["dev"],

"label": "Test Parameter",

"description": "test parameter",

"options": ["dev", "test", "prod"]

},

{

"name": "test2",

"type": "String",

"values": ["another default Value"],

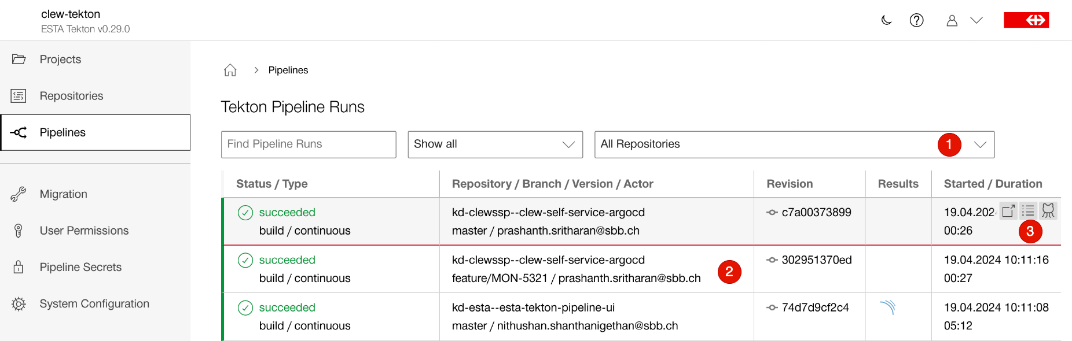
"description": "test parameter"

}

],

## Monitor Pipeline Runs

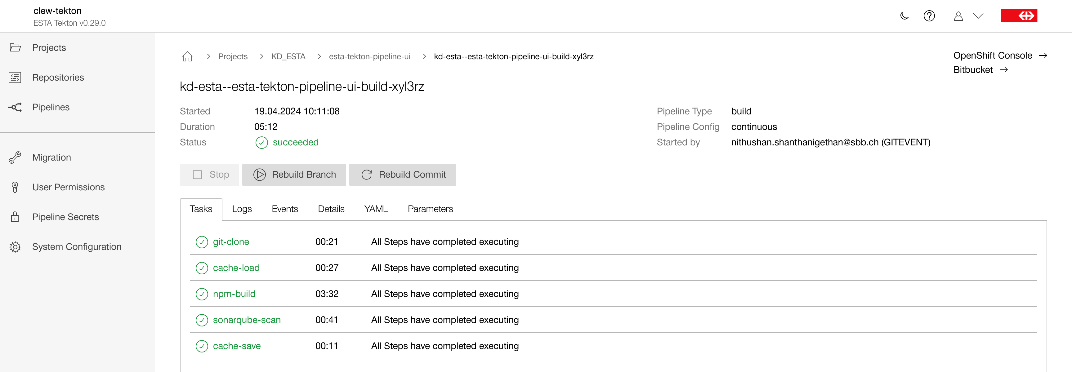
The "Pipelines" section in the main menu opens a list of current and past pipeline runs on your build namespace.

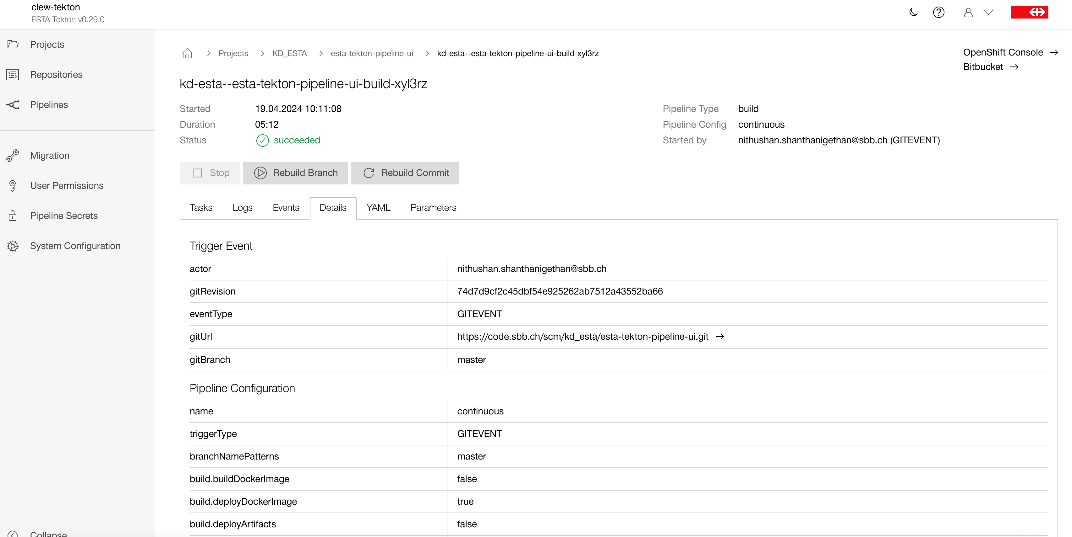


The listing can be filtered by status, project/repository and text search (1) and is sorted in descending order showing the most recent pipeline run on top.  
It shows the most relevant attributes of pipeline runs including the Git source (2). This can either be a commit, a tag or a branch.  
Hover on a row to choose an action like opening the logs or the openshift console (3).

### Pipeline Run Details

All relevant information regarding a Tekton pipeline run are displayed in the Tekton Control Panel. You can watch the progress of a running pipeline including a live stream of the logs and events, the yaml configuration of the pipeline as well as all pipeline parameters. When finished, the logs are persisted for later investigation or reference.

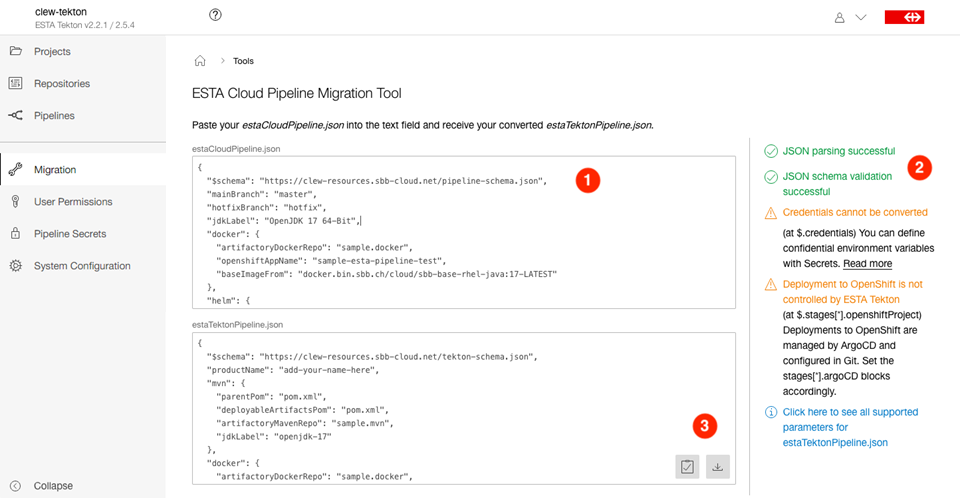




This view also provides the action to perform a stop on the given pipeline run. It will stop all processes and set the status to cancelled.

## ESTA Cloud Pipeline Migration Tool

When migrating a project previously built with the ESTA Cloud Pipeline to ESTA Tekton, the control panel provides an automatic conversion for existing estaCloudPipeline.json file to estaTektonPipeline.json.

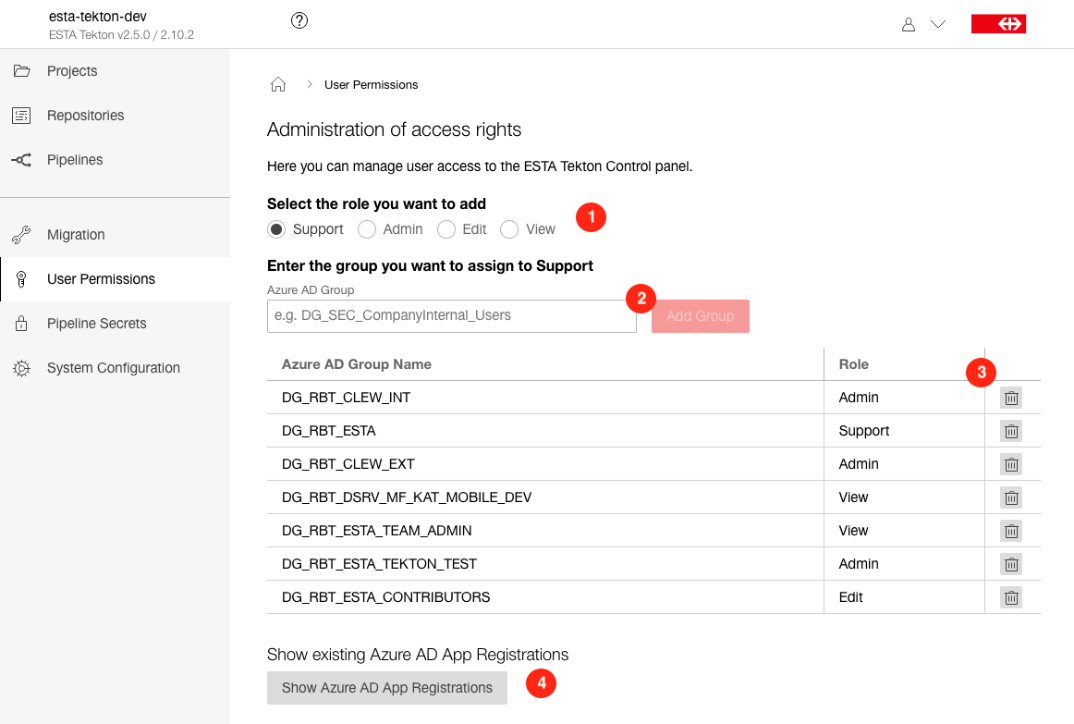


Go to the Tools section in the main menu and simply copy and paste your existing estaCloudPipeline.json into the input field (1) and watch the conversion process (2). When successful, the resulting estaTektonPipeline.json can be copied or downloaded (3).

## User Permissions

If you setup a new ESTA Tekton Namespace, you usually want to give access rights to a certain users.

User permissions can be granted to Azure AD groups and thus also to RBT groups. We suggest to create designated RBT groups to control access to your Tekton build system.



1. Select the role you want to assign a certain group to
2. You start typing or paste your Azure AG group name. After you typed 5 letters it shows possible options of groups to select. Select the group you want to assign the selected role.
3. You can delete an existing group-role assignment to remove access to ESTA Tekton
4. You can display the underlying Azure AD App Registration with all group-role assignments and other details

ESTA Tekton provides four roles to manage different levels of access to the build system:

* **Admin**: All access, including system configuration and user permission management
* **Edit**: Can start, stop, re-run and delete pipelines
* **View**: Read-only view to Tekton enabled repositories and pipelines
* **Support**: Required access to give technical support about ESTA Tekton

Azure Ad Group Openshift Sync

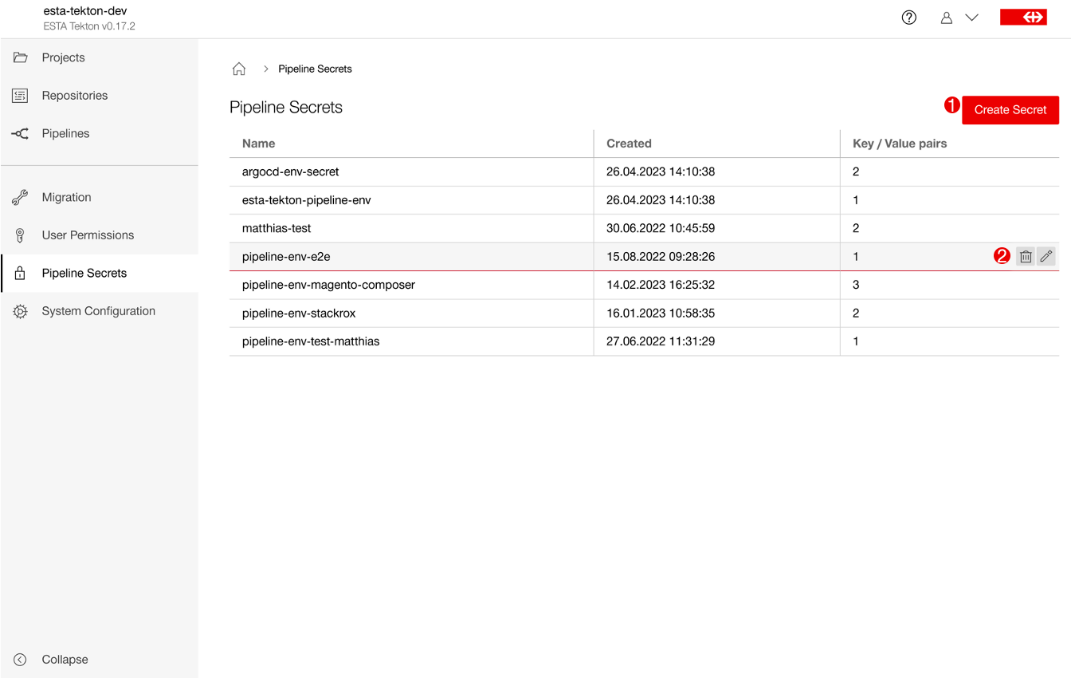
From Release 0.12.0 on the setup Azure Ad Groups will be synced towards your Openshift Tekton namespace (Apart from the Groups DG IT (1 Intern) and DG IT (2 Extern)). You will then have the same access rights in the Openshift Console as you have in the Esta-Tekton-Ui.

Let us support you!

Please assign the DG\_RBT\_ESTA group to either the Support or the Admin role. This helps us (the ESTA team) to help you in case you get stuck with Tekton.

## Pipeline Secrets

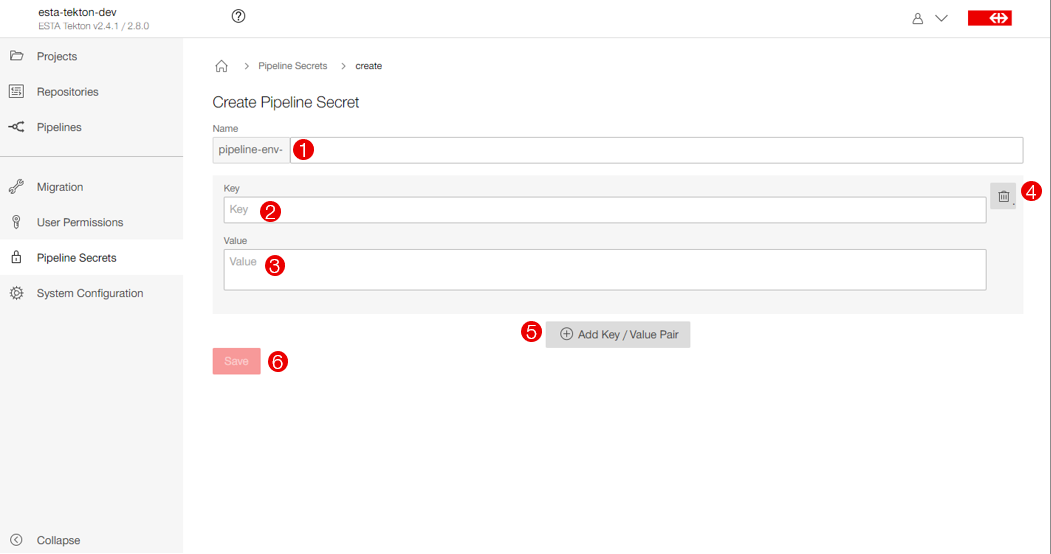
If you want to configure your project specific secrets you can view, edit or create them for your build namespace.



Here you see an overview with general informations about each secret.

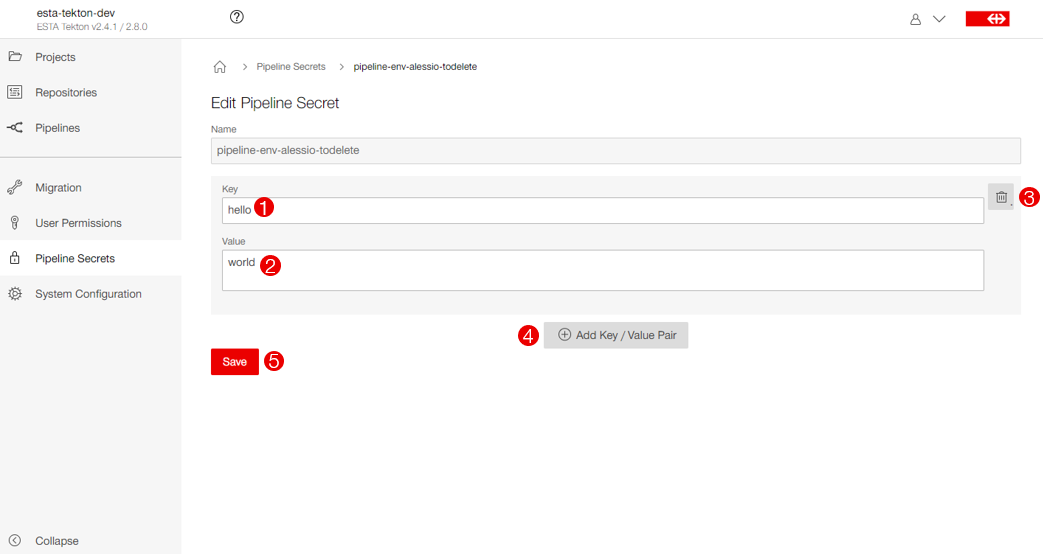
1. If you want to create a new secret, you click on the "Create Secret" button in the upper-right corner.
2. If you want to edit or delete a secret, you can hover on a secret and select either "Delete" or "Edit Secret"

**Create new secret**



1. You have to define the name of the new pipeline secret.
2. You can add a key to the new secret.
3. You can give each key its value.
4. If you don't need a Key / Value Pair you can delete it by clicking on the trash icon.
5. If you need more Key / Value Pair you can add new ones by clicking on "Add Key / Value Pair".
6. Click on the "Save" button in order to create the secret.

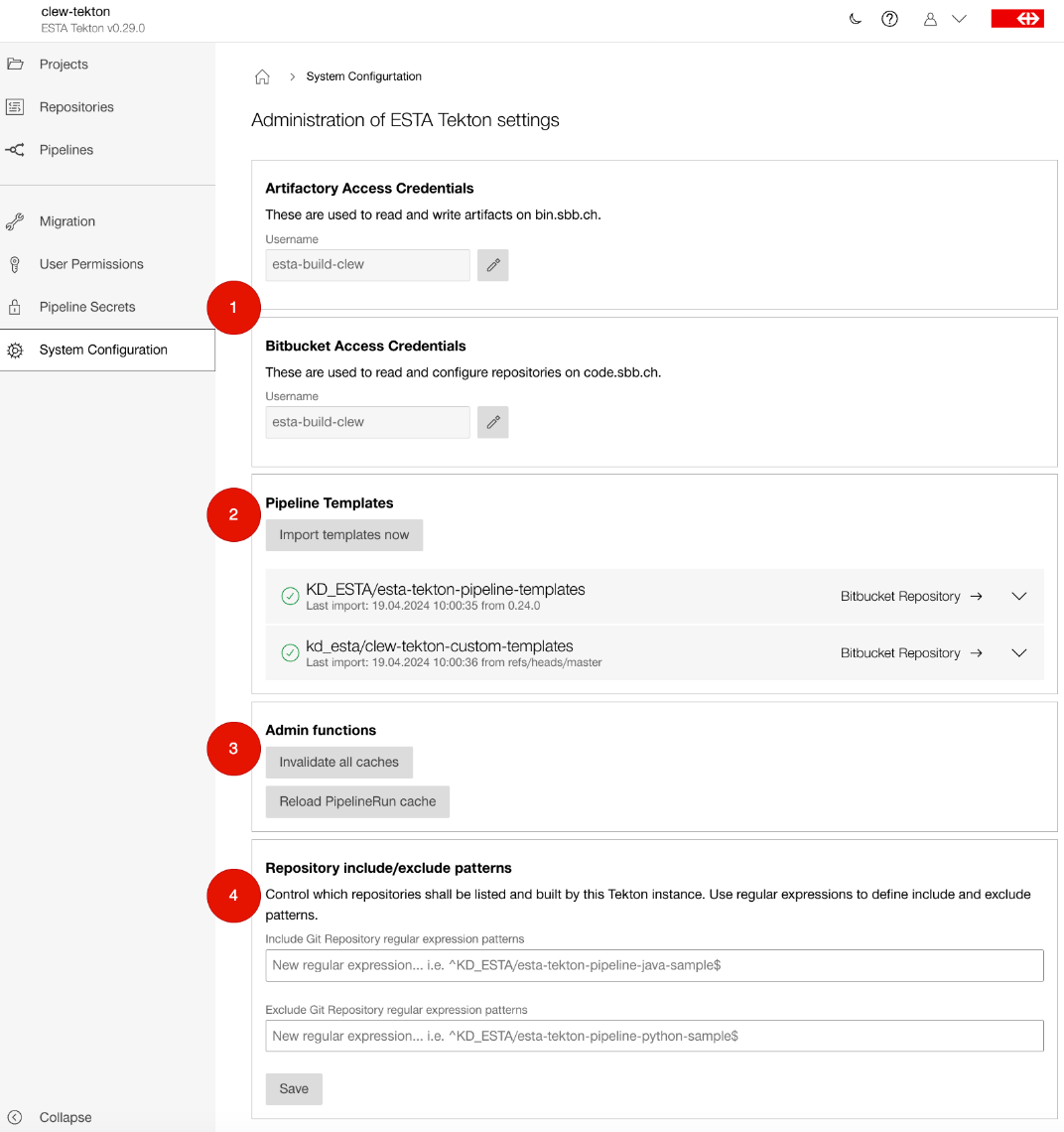
**Edit secret**



1. You can edit the existing Key.
2. You can edit the value given to the key.
3. If you no longer need the Key / Value Pair you can delete it by clicking on the trash icon.
4. If you need a new Key / Value Pair you can add one by clicking on "Add Key / Value Pair".
5. You must click on the "Save" button to save your changes.

You can't edit the name of the secret.

## System Configuration



1. You can edit here your Bitbucket and Artifactory access credentials
2. You can force the import of the template repos, for example if you made a change in your custom templates. If you open the Accordion, you can see each status of the imported template.
3. You can invalidate the caches in case you face some weird behavior, which might have to do with caches.
4. Here you can exclude or include repositories by regex patterns. After you entered your value you need to hit Enter to create a 'chip' and save it. The Caches for the repositories will be refreshed with the new filter settings.