# Getting Started

* [Prerequisites](#GettingStarted-Prerequisites)
* [Setup your Build Namespace](#GettingStarted-SetupyourBuildNamespace)
* [Enable your Repository for Tekton](#GettingStarted-EnableyourRepositoryforT)
* [Start your first Tekton Build](#GettingStarted-StartyourfirstTektonBuil)
* [What's next?](#GettingStarted-What'snext?)

# Prerequisites

As shown in the [Architecture Overview](file:///C:\display\CLEW\ESTA+Tekton+Pipeline), ESTA Tekton is meant to run in individual namespaces owned by a certain business unit on a dedicated build cluster managed by CLEW. This is to ensure strict access control to your resources and therefore requires specific access credentials to your Bitbucket projects/repositories and Artifactory repositories. Before you start, create or request the following:

* Bitbucket user with admin privileges to your project repositories
* Artifactory user with write privileges to your Docker, Maven, NPM and Helm repositories.
* An [RBT](https://rbt.sbb.ch/) AD group including the users who shall have access to the ESTA Tekton build system

How to get technical users for Bitbucket & Artifactory

If you do not yet have technical user accounts for Bitbucket and Artifactory, you must request them via a [CLEW support ticket](https://flow.sbb.ch/servicedesk/customer/portal/8/create/34). When opening a ticket, select the components "Artifactory" and "Bitbucket". The user name should follow this pattern: esta-build-<your-project/dso-name>. Also provide an email address for the user or a safe group mailbox, since security relevant data are sent over this mailbox.

# Setup your Build Namespace

To setup your ESTA Tekton build namespace, follow these steps:

1. Get technical users on Bitbucket and Artifactory as mentioned in the Prerequisites and grant them the according rights as described [here](file:///C:\display\CLEW\Dedicated+Build+Namespaces#DedicatedBuildNamespaces-PermitTektontoAccessyourResources).  
   If you don't already have such user accounts, please open a [CLEW support ticket](https://flow.sbb.ch/servicedesk/customer/portal/8/create/34) to order them (see box above).
2. Order the Tekton build namespace on the [CLEW Self Service Portal](https://self.sbb-cloud.net/tools/openshift/build-namespace).  
   For project name enter a simple name representing your project or DSO.  
   Once the Tekton build system is ready, you'll receive an email notification with a link to the Tekton Control Panel.
3. When opening the [Tekton Control Panel](file:///C:\display\CLEW\Tekton+Control+Panel) you probably see this warning box:  
     
   Click it and follow the guided steps in the Setup Assistant.  
   When finished, your ESTA Tekton build system is ready.

# Enable your Repository for Tekton

In order to make a certain Bitbucket repository tekton-ready, you need to create a file named estaTektonPipeline.json in the root directory of the **default branch**.

Here's a minimal example for a Java Maven application with a Dockerfile:

**estaTektonPipeline.json**

{

"$schema": "https://clew-resources.sbb-cloud.net/tekton-schema.json",

"productName": "esta-tekton-pipeline-test",

"mvn": {

"parentPom": "pom.xml"

},

"docker": {

"artifactoryDockerRepo": "test.docker"

},

"pipelines": [

{

"name": "continuous",

"triggerType": ["GITEVENT", "USER"],

"branchNamePrefixes": [

"master",

"feature"

]

},

{

"name": "release",

"triggerType": ["GITEVENT"],

"versionTagEventPatterns": [

"^(\\d+\\.)(\\d+\\.)(\\\*|\\d+)"

],

"build": {

"buildDockerImage": true

}

}

]

}

The JSON example in short:

* $schema: is used for validation and also enables auto-completion in IntelliJ or VSCode.
* productName: this is the name of your project/product. This value is used as a default to compose Docker image and Helm chart names.
* mvn: indicates it is a Java Maven project
* docker: defines a Docker image build which pushes the image to the specified repository in Artifactory.
* pipelines: this is a list of configurations to trigger build pipelines on specific events:
  + "continuous": this profile triggers on Git pushes to a branch starting with "feature". It simply runs a Maven build without creating any artifacts.
  + "release": this profile triggers when a Git tag with a semantic version (e.g. 1.2.24) is pushed. It also builds a Docker image and pushes it to Artifactory.
  + "xyz-snapshot-build": you can create as many pipeline profile as you want with your own trigger setup by filtered by branches, event source or version tag pattern.

Build types

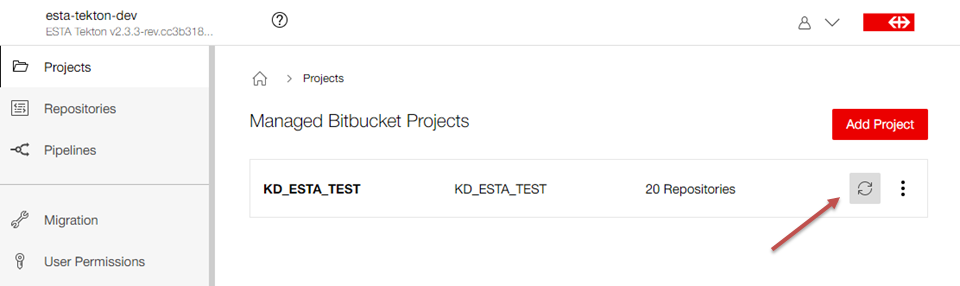
With the ESTA Tekton Pipeline we have more flexible build steps compared to the ESTA Cloud Pipeline. Before you had build types like continuous, release, snapshot and so on, now you can define as many build profiles as you want and even start your own custom pipelines behind it.

When pushing a new tag 1.2.25 to Bitbucket, this example builds the Java application, creates the jar, package it into the Docker image and push it to test.docker.bin.sbb.ch/test/esta-tekton-pipeline-test:1.2.25

For a complete documentation of the estaTektonPipeline.json schema, see [ESTA Tekton Parameters - estaTektonPipeline.json](file:///C:\display\CLEW\ESTA+Tekton+Parameters+-+estaTektonPipeline.json).

The ESTA Tekton Controller will periodically scan all Bitbucket repositories in the configured Bitbucket projects. If it finds a estaTektonPipeline.json file, it will display the Bitbucket repository in the Tekton Control Panel.   
Behind the scenes the ESTA Tekton Controller registers its own url as a webhook in the repository with estaTektonPipeline.json available.

You can enforce a re-scan of your projects by clicking the sync button in the "Projects" view:



Migration

In case you already have a estaCloudPipeline.json you can either let the builds coexist or disable temporarily the ESTA Cloud Pipeline build by commenting the pipeline call in your Jenkinsfile. If you let both builds coexist, be aware, that there might both builds on Tekton and Jenkins triggered at the same time.

# Start your first Tekton Build

Tekton pipelines are either started upon receiving webhook notifications from Bitbucket events like pushing new tags or commits or from the Tekton Control Panel.

**a) Trigger build via Git**

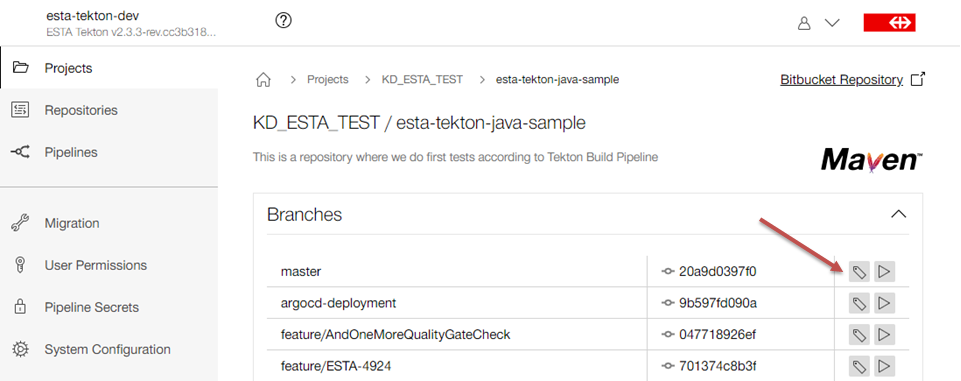
To start a continuous integration build, simply create a new branch named feature/tekton-test and push it to Bitbucket.

The above listed example configuration will now start a Tekton pipeline basically executing mvn clean package with all tests and then stop.  
If build and tests succeed, the pipeline run is successful otherwise failed. This build result is also reported to Bitbucket and displayed when creating pull requests.

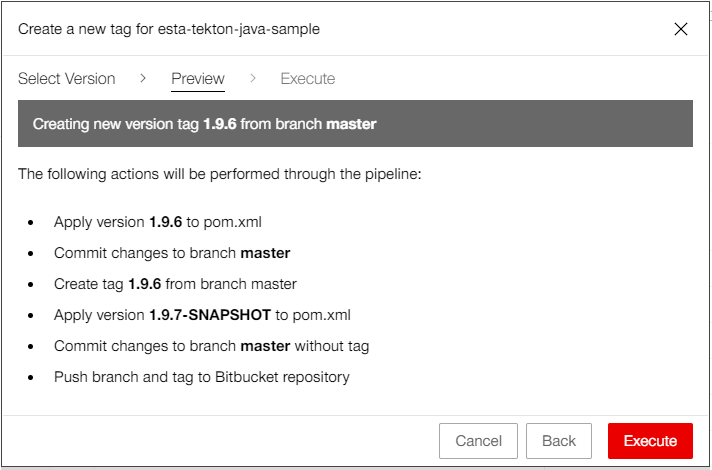
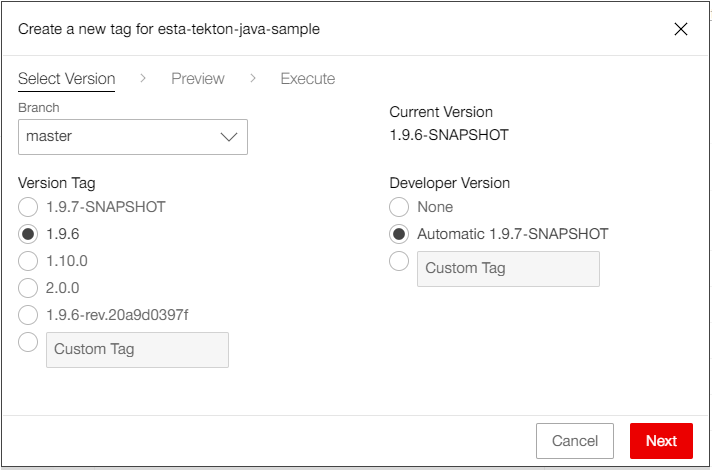
**b) Start Tagging via the Tekton Control Panel**

Esta Tekton wants to enforce to create Git tags for each released version. Although this can be done directly with Git, the ESTA Tekton Tag assistant simplifies creating new releases.

Open the Tekton Control Panel, navigate to the repository you want to create a release for and click the "Start Tagging" button on the desired branch:

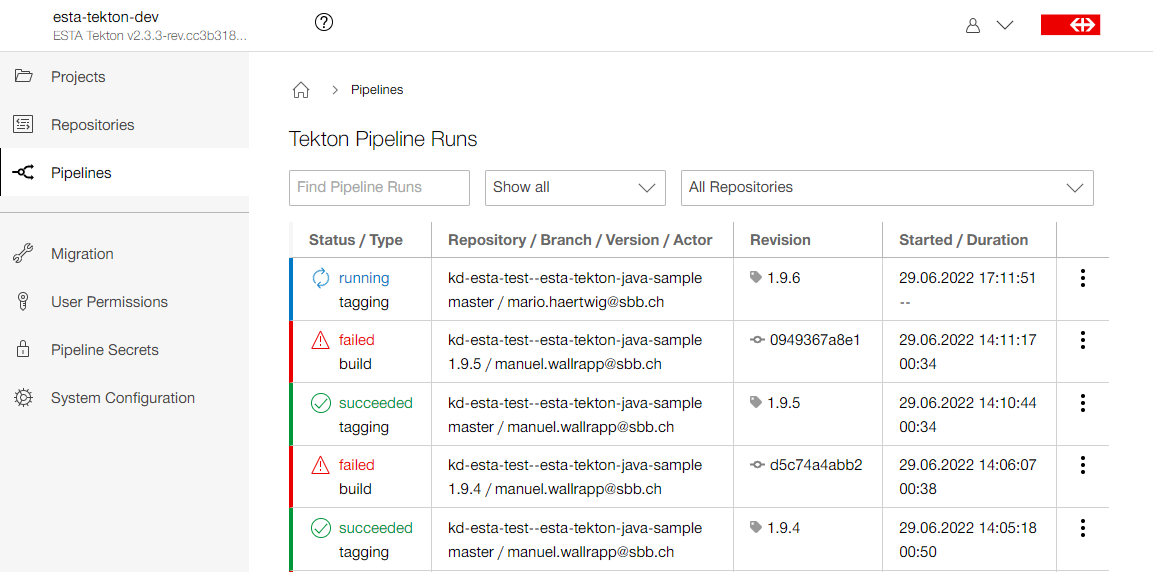


This will open a tagging assistant to guide you through the process. It first reads the current version from the pom.xml in your repository. It then suggests you the next versions by the [semver](https://semver.org/) pattern. Select the desired version or enter a custom tag name. The next step shows a summary of what the tagging pipeline will do. Besides creating a Git tag, it'll first apply the selected version to the project's pom.xml and if selected set the next SNAPSHOT version.



As shown in the [Architecture Overview](file:///C:\display\CLEW\ESTA+Tekton+Pipeline), the Git push executed by the tagging pipeline, triggers a webhook so the Tekton controller will process the Git event according to the settings in estaTektonPipeline.json. In the example showed in the screenshot, the creation of the Git tag 1.5.1 triggers the "release" build pipeline profile. It starts a Tekton pipeline to build the application and pushes the resulting Docker image to Artifactory.

Navigate to the "Pipelines" section in the Tekton Control Panel and watch the progress of the started pipelines there:



# What's next?

Now you have enabled your first repository for Tekton and run a Tekton pipeline. It's time to

* get familiar with the [Tekton Control Panel](file:///C:\display\CLEW\Tekton+Control+Panel)
* dive deeper into the [ESTA Tekton Workflow](file:///C:\display\CLEW\ESTA+Tekton+Workflow)
* learn how to [customize your builds](file:///C:\display\CLEW\ESTA+Tekton+Parameters+-+estaTektonPipeline.json) using the estaTektonPipeline.json file
* find out how to [configure notifications](file:///C:\display\CLEW\ESTA+Tekton+Workflow#ESTATektonWorkflow-Notifications) for build success and failures
* [set up ArgoCD](file:///C:\display\CLEW\Deployment+with+ArgoCD+and+Helm+Charts) to deploy your applications with a single click

Learn more about the concepts and technologies behind ESTA Tekton in [Core Concepts](file:///C:\display\CLEW\Core+Concepts).