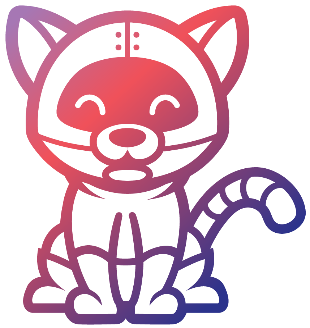
# Core Concepts

* [Tekton Pipelines for CI/CD](#CoreConcepts-TektonPipelinesforCI/CD)
* [Deployment with ArgoCD](#CoreConcepts-DeploymentwithArgoCD)
  + [Individual ArgoCD instances](#CoreConcepts-IndividualArgoCDinstances)
  + [App of Apps Pattern](#CoreConcepts-AppofAppsPattern)
* [ESTA Tekton Pipelines](#CoreConcepts-ESTATektonPipelines)

# Tekton Pipelines for CI/CD

In ESTA Tekton CI/CD we use [Tekton](https://tekton.dev/) or the [OpenShift variant](https://docs.openshift.com/container-platform/4.7/cicd/pipelines/understanding-openshift-pipelines.html) of it in its core. Tekton is available on the dedicated build cluster. For the build and release process ESTA Tekton implements standard pipelines based on Tekton. The Pipelines are started by the ESTA Tekton Controller (see [Architecture Overview](file:///C:\display\CLEW\ESTA+Tekton+Pipeline)). Frontend to the builds is Tekton Control Panel. No direct access or deep understanding of Tekton is necessary for the user to operate the build system.

Tekton however is a very powerful tool and the ESTA Tekton build system also allows modifications or extensions to the standard pipelines we provide. It even supports to create completely custom pipelines which are still started and monitored by the controller. If the standardized build process doesn't cover all your needs, we recommend to learn more about [Tekton Pipelines](https://tekton.dev/docs/pipelines/) and start building your own. In the future it will be possible to overwrite the esta standard pipelines by your own. Of course the Clew team can support you with your needs or feature requests.



# Deployment with ArgoCD

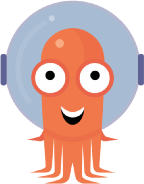
For the deployment of the ESTA Tekton CI/CD system we recommend to use [ArgoCD](https://argoproj.github.io/cd) with [Helm](https://helm.sh/) charts to describe the project deployment. Deployment is not an integrated part of Tekton pipelines and thus ArgoCD is only one option to choose, although the preferred one. The build pipelines can create Helm charts and publish them to a Helm repository in Artifactory. ArgoCD can be configured to apply these Helm charts from Artifactory or directly sync charts from a Git repository.

## Individual ArgoCD instances

As shown in the [Architecture Overview](https://confluence.sbb.ch/display/CLEW/ESTA+Tekton+Pipeline), individual ArgoCD instances are set up on the cluster where your projects are deployed and are just triggered by Tekton to sync after successful builds. The setup and configuration (including access control) is specific to each business unit for maximum flexibility and security.

## App of Apps Pattern

ArgoCD promotes the GitOps approach.  So almost all of its configurations should be stored in Git. We recommend to implement the [app of apps pattern](https://argoproj.github.io/argo-cd/operator-manual/cluster-bootstrapping/) as described in the ArgoCD documentation.



# ESTA Tekton Pipelines

So what are ESTA Tekton Pipelines then?

It's a catalog of standard [Tekton pipelines](https://tekton.dev/docs/pipelines/) and tasks used to build and deploy SBB projects from a estaTektonPipeline.json definition. Each pipeline definition has a corresponding template (\_run.yaml) to start a [PipelineRun](https://tekton.dev/docs/pipelines/pipelineruns/) using a set of parameters. These files are primarily controlled by the ESTA Tekton Controller but can also be applied and run individually. The ESTA Tekton Controller makes sure that all those pipelines with their referenced task definitions are present in the build namespace.

