# Deployment with ArgoCD and Helm Charts

* [Setup ArgoCD for your Deployments](#DeploymentwithArgoCDandHelmCharts-Setup)
  + [Enable ArgoCD Sync via Tekton Pipelines](#DeploymentwithArgoCDandHelmCharts-Enabl)
  + [Connect to different ArgoCD instances from Tekton Pipelines](#DeploymentwithArgoCDandHelmCharts-Conne)
* [Create Helm Charts](#DeploymentwithArgoCDandHelmCharts-Creat)
  + [Umbrella Charts](#DeploymentwithArgoCDandHelmCharts-Umbre)
  + [Best practice suggestions](#DeploymentwithArgoCDandHelmCharts-Bestp)
  + [Migrate OpenShift Templates to Helm Charts](#DeploymentwithArgoCDandHelmCharts-Migra)
* [Manage ArgoCD Deployments in Git (app of apps)](#DeploymentwithArgoCDandHelmCharts-Manag)
* [Update ArgoCD Values from Tekton Pipelines](#DeploymentwithArgoCDandHelmCharts-Updat)

Following the GitOps pattern, deployments to OpenShift shall be defined as Helm charts and applied using ArgoCD.

Note that ArgoCD only uses the templating functions of Helm charts but does not actually perform a Helm deployment.

# Setup ArgoCD for your Deployments

Follow the [ArgoCD Setup and Configuration](file:///C:\display\CLEW\ArgoCD+Setup+and+Configuration) guide to get your own ArgoCD instance up and running.

## Enable ArgoCD Sync via Tekton Pipelines

In order to allow Tekton to trigger app syncs in ArgoCD, a specific user *tektoncontroller* should be created in ArgoCD and its credentials added to your Tekton build namespace.

* When using the [CLEW ArgoCD chart](https://code.sbb.ch/projects/OM_PAAS/repos/helm-argocd/browse), the user tektoncontroller is already configured.
* If you're not using this chart, you have to create the user tektoncontroller manually:

oc -n argocd patch configmap argocd-cm -p '{ "data": { "accounts.tektoncontroller": "apiKey" } }'

Afterwards, you just have to generate a token for the user:

1. Generate an access token for this user via the ArgoCD web UI:
   1. navigate to Settings > Accounts > tektoncontroller
   2. in the section Tokens click "Generate New"
   3. copy the new token
2. Enter the ArgoCD host name (without https://) and the generated access token to the secret with the name argocd-env-secret.
   1. in the [Tekton Control Panel](file:///C:\display\CLEW\Tekton+Control+Panel), navigate to Pipeline Secrets
   2. edit the secret argocd-env-secret and save the changes

## Connect to different ArgoCD instances from Tekton Pipelines

By default, Tekton pipelines use the argocd-env-secret to connect to ArgoCD but you can create additional secrets with the keys ARGOCD\_SERVER and ARGOCD\_AUTH\_TOKEN and reference the secret name in the [stage config](file:///C:\display\CLEW\ESTA+Tekton+Parameters+-+estaTektonPipeline.json) of your estaTektonPipeline.json:

"stages": [  
 {  
 "stageName": "dev",  
 "argoCD": {  
 "argoCdAppName": "<argocd-appname>",  
 "autoSync": true,  
 "argoCdSecret": "argocd-env-secret-<some-name>"

}  
 }  
],

You can manage these secrets directly in the Pipeline Secrets section of your [Tekton Control Panel](file:///C:\display\CLEW\Tekton+Control+Panel).

# Create Helm Charts

If your start from scratch, have a look at the [esta-argocd blueprint](https://code.sbb.ch/projects/KD_ESTA_BLUEPRINTS/repos/esta-argocd/browse/applications) providing a sample Helm chart with the minimal required resources to get an application running on OpenShift.

Another way is to use the [Helm CLI](https://helm.sh/docs/intro/install/) to create a new chart:

$ helm create <chart-name>.

The generated chart, however, might be too complex and includes resources like HorizontalPodAutoscaler and Ingress which are not needed in our environments.

## Umbrella Charts

Helm charts have the ability to include other charts, referred to as subcharts, via their [dependencies](https://helm.sh/docs/helm/helm_dependency/) section. When a chart is created for the purpose of grouping together related subcharts/services, such as to compose a whole set of micro-service deployments, we call this an umbrella chart. Use this technique to orchestrate complex deployments with cross-dependencies.

In order to use a Helm chart of an application/service as a dependency in an umbrella helm chart, the former needs to packaged and published to a Helm chart repository. Tekton build pipelines will do that for you and can push versioned charts to Artifactory (bin.sbb.ch/artifactory/xxxx.helm). Enable the packageAndDeployHelmChart flag in your estaTektonPipeline.json [build pipeline config](https://confluence.sbb.ch/display/CLEW/ESTA+Tekton+Parameters+-+estaTektonPipeline.json#ESTATektonParametersestaTektonPipeline.json-Buildpipelineconfigurationstructure) and define the repository name using the chartRepository property in the [Helm configuration structure](https://confluence.sbb.ch/display/CLEW/ESTA+Tekton+Parameters+-+estaTektonPipeline.json#ESTATektonParametersestaTektonPipeline.json-Helmconfigurationstructure).

## Best practice suggestions

Besides the [Best Practices](https://helm.sh/docs/chart_best_practices/templates/) suggested by Helm, we'd like to highlight the following:

* Create a values-xxx.yaml file for each stage/environment (e.g. dev, test, int. prod)
* Read the deployment version of the app (→ docker image tags, etc.) from .Values to maintain them individually for each stage

## Migrate OpenShift Templates to Helm Charts

ESTA provides a tool to help you migrate existing OpenShift templates into Helm charts.  
See the [OpenShift-Helm-Migration](file:///C:\display\CLEW\OpenShift-Helm-Migration) section for details.

# Manage ArgoCD Deployments in Git (app of apps)

Although new applications can easily be added to ArgoCD via the UI or as Kubernetes resources (kind: Application), they're only persisted in OpenShift and may be lost during migration or due to accidental deletion. Therefore we recommend the declarative approach to manage them in Git and let ArgoCD sync its own configuration.

Read more about the **app of apps pattern** in the [ArgoCD docs](https://argo-cd.readthedocs.io/en/stable/operator-manual/cluster-bootstrapping/#app-of-apps-pattern) or use fork the [ESTA ArgoCD Blueprint](https://code.sbb.ch/projects/KD_ESTA_BLUEPRINTS/repos/esta-argocd/browse).

# Update ArgoCD Values from Tekton Pipelines

See the Staging Section of [ESTA Tekton Parameters](file:///C:\display\CLEW\ESTA+Tekton+Parameters+-+estaTektonPipeline.json) to learn how to update deployment versions in ArgoCD after successful builds in ESTA Tekton.