# Migrating from ESTA Cloud Pipeline

* [Migrating estaCloudPipeline.json](#MigratingfromESTACloudPipeline-Migratin)
* [Dockerfile Migration](#MigratingfromESTACloudPipeline-Dockerfi)
* [OpenShift Templates to Helm Chart Migration](#MigratingfromESTACloudPipeline-OpenShif)
* [Kafka, APIM, IAM AppRegistration Deployment via ArgoCD](#MigratingfromESTACloudPipeline-Kafka,AP)

## Migrating estaCloudPipeline.json

The migration of projects already using the ESTA Cloud Pipeline with an estaCloudPipeline.json is very easy. The Tekton Control Panel provides a fully automated converter to generate your new estaTektonPipeline.json. Read more about the [ESTA Cloud Pipeline Migration Tool](file:///C:\display\CLEW\Tekton+Control+Panel#TektonControlPanel-ESTACloudPipelineMigrationTool).

## Dockerfile Migration

As opposed to the ESTA Cloud Pipeline, the build process in Tekton differs a bit. While the Cloud Pipeline first pushed the build artifacts (e.g. the app jar) to Artifactory in order to pull it again for building the Docker image, the Docker build process in Tekton has direct access to the build workspace and thus makes that step unnecessary. When using a custom Dockerfile (for Java projects you can omit this and use a standard template suitable for most apps) use the COPY command to copy the build artifacts file from the build output directory into your Docker image.

For a Java/Maven project, replace the ADD <https://bin.sbb.ch/artifactory/...jar> command with COPY target/<app-file-name>\*.jar in your Dockerfile. The value of the currently built version is passed as APP\_RELEASE\_VERSION build arg and can be consumed in the Dockerfile with ARG APP\_RELEASE\_VERSION. An example Dockerfile can be found [here](https://code.sbb.ch/projects/KD_ESTA/repos/esta-tekton-pipeline-json-examples/browse/java/springboot-docker/docker).

Another difference to the ESTA Cloud Pipeline is the context directory for Docker builds. While in the Cloud Pipeline all file references (e.g. ADD or COPY) in the Dockerfile were relative to the directory where the Dockerfile resides, in Tekton the context directory is set to the project root unless you specify it explicitly with the [contextDir property](file:///C:\display\CLEW\ESTA+Tekton+Parameters+-+estaTektonPipeline.json#ESTATektonParametersestaTektonPipeline.json-Dockerconfigurationstructure). Therefore you should review all ADD and COPY commands in your Dockerfile(s).

Additionally, do these steps from  <https://code.sbb.ch/projects/KD_ESTA_BLUEPRINTS/repos/esta-cloud-angular/browse/docker/Dockerfile>

## OpenShift Templates to Helm Chart Migration

Although the ESTA Cloud Pipeline also supports deployments with Helm charts, you may still use the deprecated OpenShift Templates for your deployments. In order to move the deployment to [ArgoCD](file:///C:\display\CLEW\Deployment+with+ArgoCD+and+Helm+Charts) these templates can be converted into Helm charts using a small utility tool. This [guide describes](file:///C:\display\CLEW\OpenShift-Helm-Migration) the migration in detail.

## Kafka, APIM, IAM AppRegistration Deployment via ArgoCD

For the kafkaDeployParameters in estaCloudPipeline.json there's no equivalent in estaTektonPipeline.json because deployment topics are handled by ArgoCD. To manage these resources GitOps-style you can find details linked here: [Kafka, IAM AppRegistrations and APIM Deployments with Argocd](file:///C:\display\CLEW\Kafka,+IAM+AppRegistrations+and+APIM+Deployments+with+Argocd)