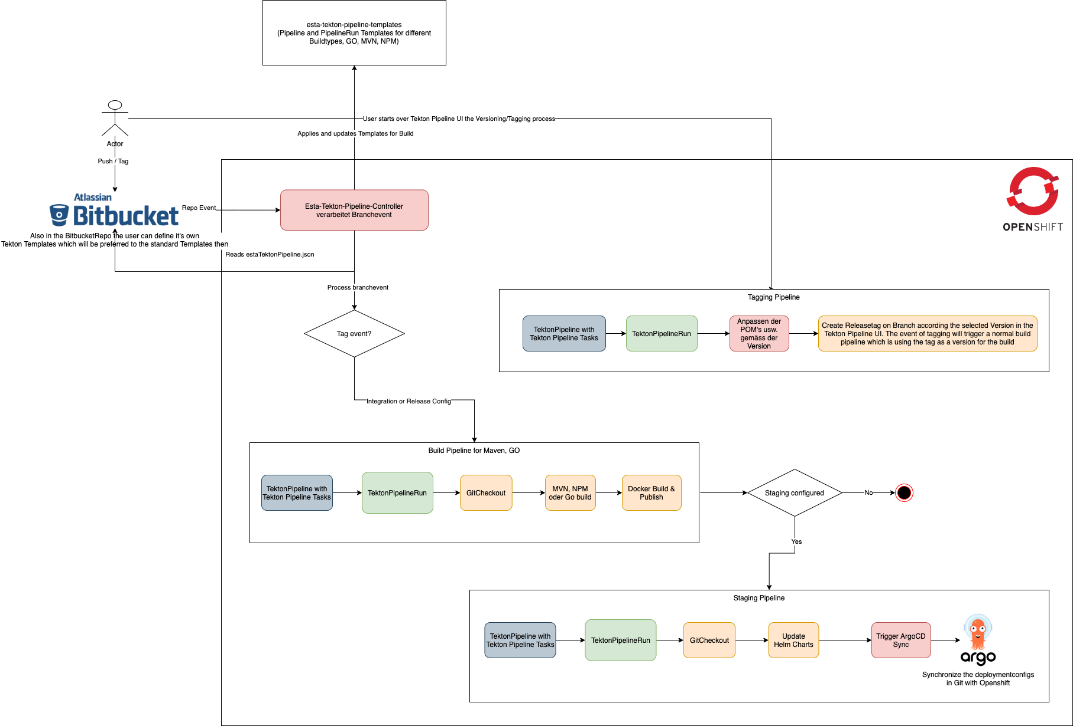
# Esta Tekton Development

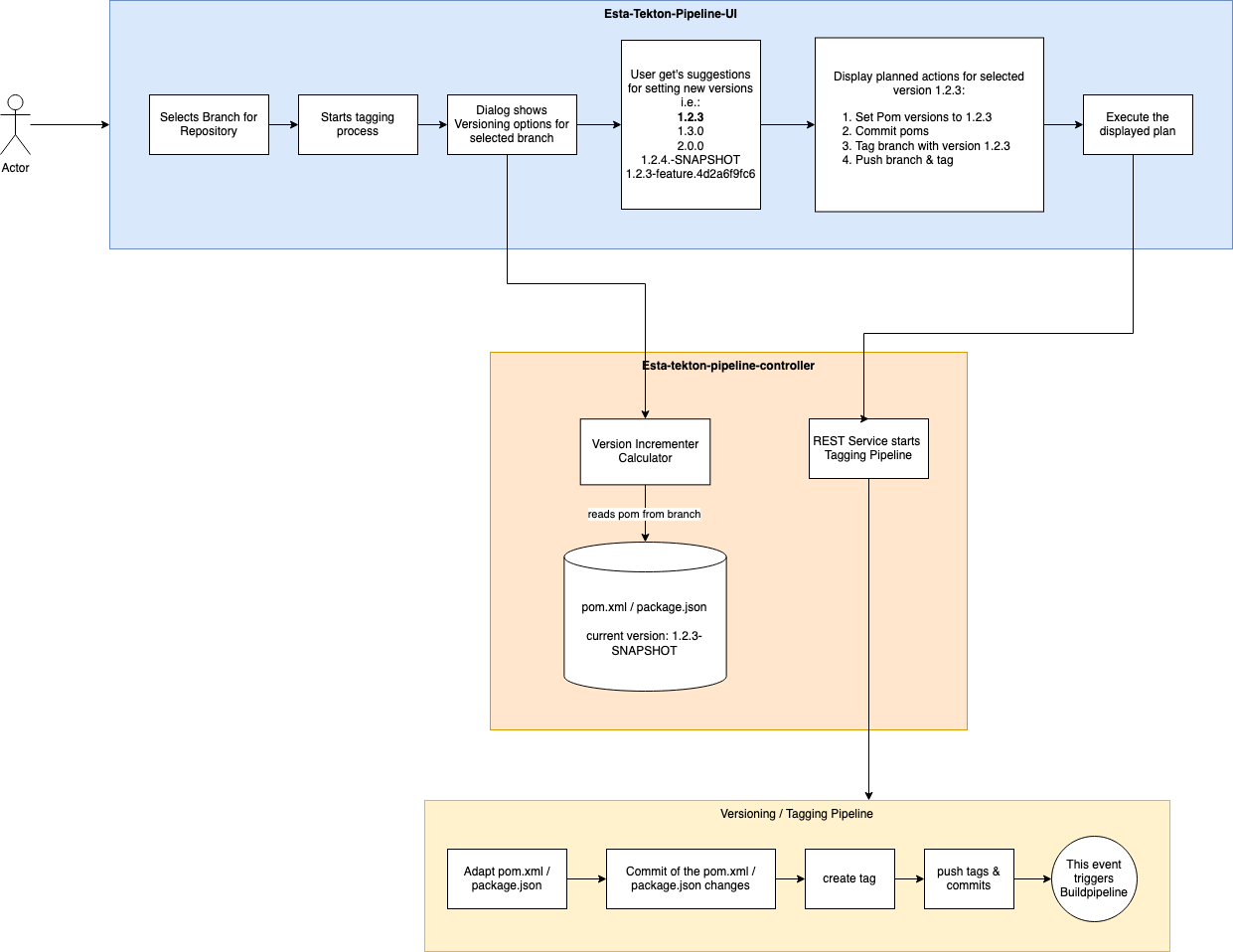
Notizen, Vorgehen, Ideen zur Esta Tekton Pipeline.

* [Gesamtarchitektur Esta Tekton Pipeline](#EstaTektonDevelopment-Gesamtarchitektur)
* [Esta Tekton Tagging Process Detailed](#EstaTektonDevelopment-EstaTektonTagging)
* [Esta Tekton Controller Architektur](#EstaTektonDevelopment-EstaTektonControl)
  + [Start einer Tekton Pipeline](#EstaTektonDevelopment-StarteinerTektonP)
* [Deployment Setup von ESTA Tekton](#EstaTektonDevelopment-DeploymentSetupvo)
* [Ablauf Release/Deployment Prozess mit ESTA Tekton, Helm und ArgoCD](#EstaTektonDevelopment-AblaufRelease/Dep)
* [Source Repositories](#EstaTektonDevelopment-SourceRepositorie)
* [Feature Matrix (Draft)](#EstaTektonDevelopment-FeatureMatrix(Dra)
* [Custom Pipelines (Draft)](#EstaTektonDevelopment-CustomPipelines(D)
  + [Custom Pipeline Templates im Projekt-Repository](#EstaTektonDevelopment-CustomPipelineTem)
  + [Custom Repository mit Pipeline und Task Definitionen](#EstaTektonDevelopment-CustomRepositorym)
  + [Integration in estaTektonPipeline.json](#EstaTektonDevelopment-Integrationinesta)
* [Controller REST API (Draft)](#EstaTektonDevelopment-ControllerRESTAPI)

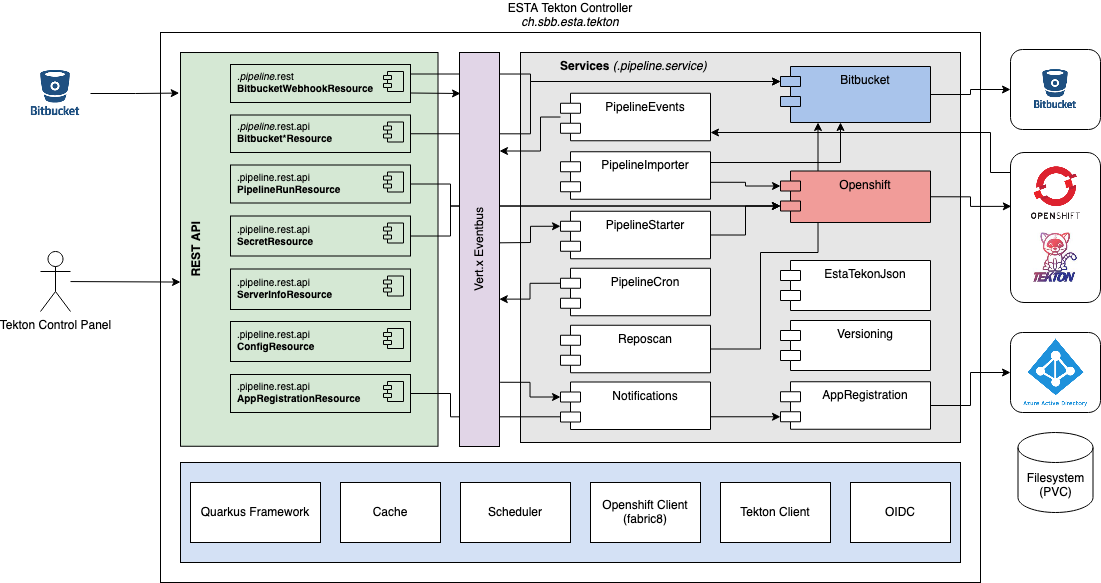
## Gesamtarchitektur Esta Tekton Pipeline



## Esta Tekton Tagging Process Detailed



## Esta Tekton Controller Architektur

******

**REST API** (Klassen)

Definieren und bedienen das REST API. Für den Datenaustausch werden interne Models in DTO (Data Transfer Objects) gemapped und umgekehrt.

**Services** (Module)

Enthalten Service-Klassen, welche die Businesslogik des Controllers implementieren und mit Umsystemen wie Bitbucket oder Openshift/Tekton kommunizieren.

**Eventbus**

Asynchrone Ereignisse werden über den Eventbus gesendet, wo einzelne Service-Klassen Subscriptions registrieren, um die Ereignisse und deren Daten zu empfangen und zu verarbeiten.

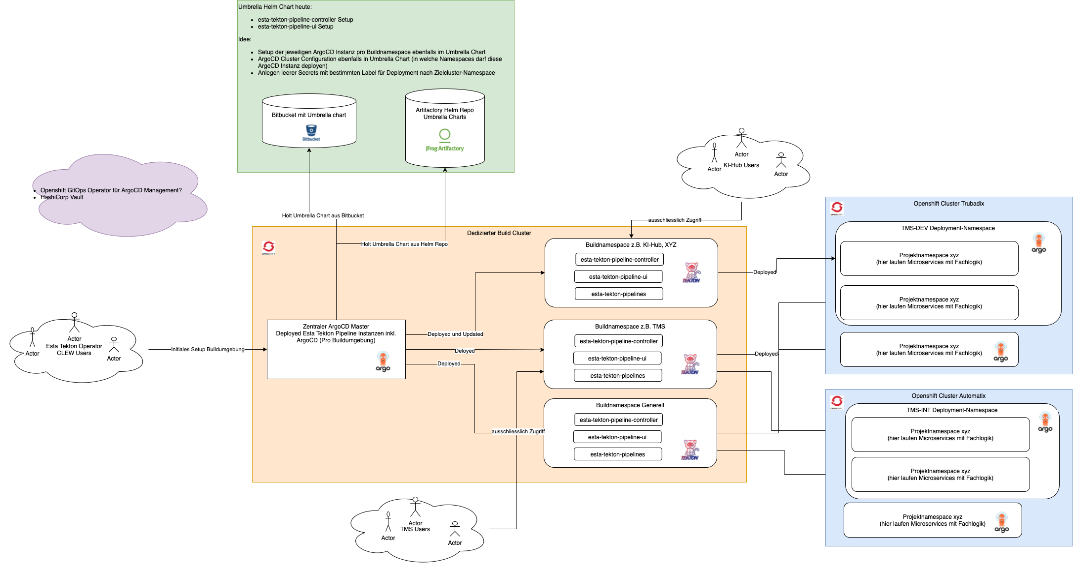
### Start einer Tekton Pipeline

TBD.

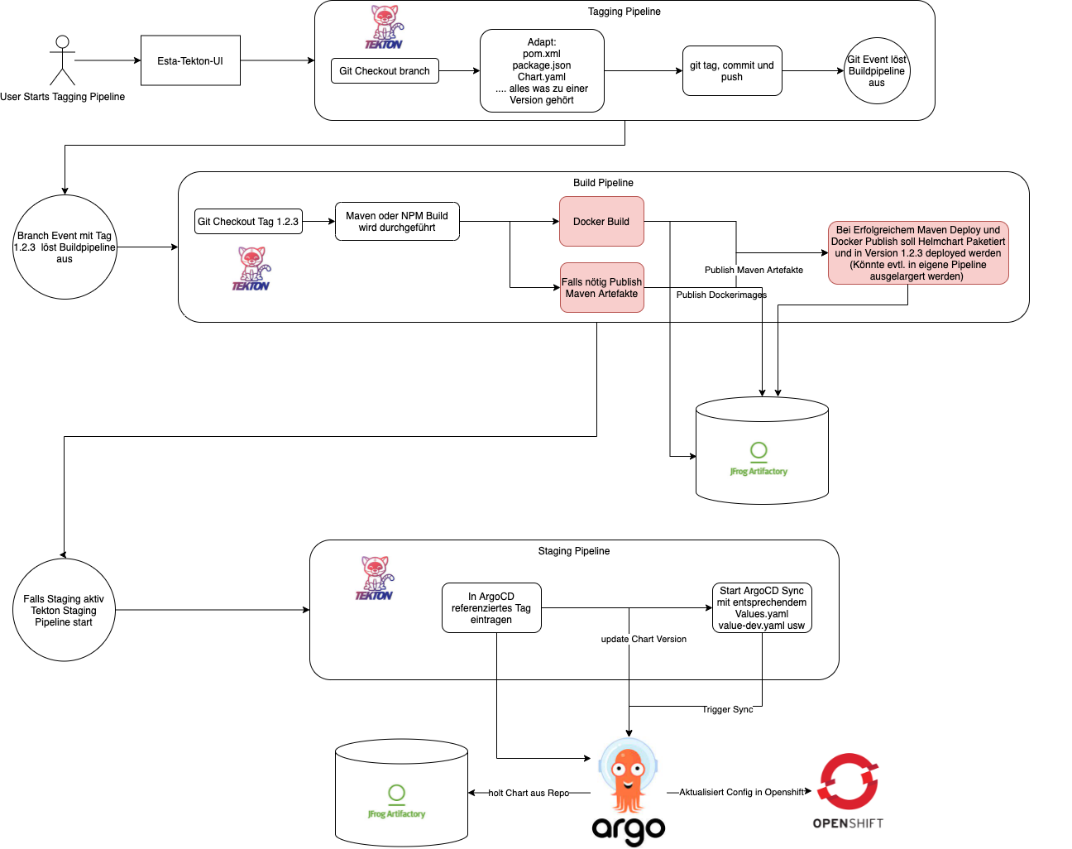
## Deployment Setup von ESTA Tekton

Wir wollen die Möglichkeit bieten, dass jedes Team ihre eigenes ESTA Tekton Build System aufsetzen können. Dies geschieht über ein Umbrella Chart und eine ArgoCD Konfiguration.

Ein mögliches Setup soll anhand dieser Grafik aufgezeigt werden:



## Ablauf Release/Deployment Prozess mit ESTA Tekton, Helm und ArgoCD



## Source Repositories

|  |  |
| --- | --- |
| **Beschreibung** | **Link** |
| ESTA Tekton Controller | <https://code.sbb.ch/projects/KD_ESTA/repos/esta-tekton-pipeline-controller/browse> |
| Tekton Control Panel UI | <https://code.sbb.ch/projects/KD_ESTA/repos/esta-tekton-pipeline-ui/browse> |
| Tekton Pipeline Templates | <https://code.sbb.ch/projects/KD_ESTA/repos/esta-tekton-pipeline-templates/browse> |
| Tekton Builder Images | <https://code.sbb.ch/projects/KD_ESTA/repos/esta-tekton-pipeline-java-builder-image/browse> <https://code.sbb.ch/projects/KD_ESTA/repos/esta-tekton-pipeline-helm-builder-image/browse> <https://code.sbb.ch/projects/KD_ESTA/repos/esta-tekton-pipeline-go-builder-image/browse> <https://code.sbb.ch/projects/KD_ESTA/repos/esta-tekton-pipeline-python-builder-image/browse> |
| Build Namespace Config | <https://code.sbb.ch/projects/KD_ESTA/repos/esta-tekton-pipeline-config/browse> |
| Tekton Test/Dev Projects | <https://code.sbb.ch/projects/KD_ESTA_TEST> |
| Tekton Integrationtests | <https://code.sbb.ch/projects/KD_ESTA/repos/esta-tekton-integrationtests/browse> |
| Tekton Integrationtest Projects | <https://code.sbb.ch/projects/KD_ESTA_TEKTON_IT/> |

## Feature Matrix (Draft)

Auf Basis des estaCloudPipeline.json ist in nachfolgender Matrix sichtbar welche Version der EstaTektonPipeline welche Features supported.

**V1**: Produktiver Pilot (1-5 Projekte builden mit Tekton)   
**V2**: Rampup interner Release für CLEW Projekte  
**Final**: Release für erste public Migrationen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Root Properties** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** | **JSON Path / Bemerkungen** |
| **productName** | Name der Applikation/des Products | :cross: | :tick: | :tick: | :tick: |  | .productName |
| **mainBranch** | Angabe des Haupt Branches | :tick: | :cross: | :cross: | :cross: |  |  |
| **hotfixBranch** | Angeabe des Hotfix Branches | :tick: | :cross: | :cross: | :cross: |  | Wird in Stepkonfig gelöst |
| **versionincrementPosition** | Angabe der Version major.minor.patch welche inkrementiert werden soll | :tick: | :tick: | :tick: | :tick: |  | Wird über Tagging Assistenten im GUI und Auto-releasing Feature gelöst. |
| **advancedMode** | Ermöglicht im Jenkinsgui mehr Parameter die man zum Build setzen kann | :tick: | :cross: | :cross: | :cross: |  |  |
| **estaDebug** | Gibt ein detaillierteres Buildlog aus | :tick: | :cross: | :cross: | :cross: |  | Wird über additional Buildparams gelöst oder aber den Loglevel des Controllers. |
| **keepReleaseBuildLogsOnJenkins** | Property welches bis zu 5 Releasebuildlogs speichert | :tick: | :cross: | :cross: | :cross: |  |  |
| **sendBuildStatusMail** | Per Default wird ein Buildstatus Email gesendet, hier kann das ausgeschalten werden | :tick: | :tick: | :tick: | :tick: | true | .notifications[\*].sendingEnabled |
| **integrationBuildCron** | Hier kann eingestellt werden, ob der Build nächtlich per Cronjob ausgeführt werden soll | :tick: | :tick: | :tick: | :tick: |  | .steps[\*].cron |
| **jdkLabel** | Auswahl des JDK's für den Build (Soll in Zukunft über Maven geschehen und ausschliesslich OpenJdk verwendet werden) | :tick: | :tick: | :tick: | :tick: |  | .mvn.jdkLabel |
| **openshiftTemplateValidation** | Openshift Template Validation prüft jedes Template vor Validierung (Deployment per ArgoCD) | :tick: | :cross: | :cross: | :cross: |  | Tekton baut auf Helm auf. |
| **Konfigstrukturen** |  |  |  |  |  |  |  |
| **docker** | Docker-Build Konfigstrutkur | :tick: | :tick: | :tick: | :tick: |  | .docker |
| **mvn** | Maven-Build Konfigstruktur | :tick: | :tick: | :tick: | :tick: |  | .mvn |
| **npm** | NPM-Build Konfigstruktur | :tick: | :tick: | :tick: | :tick: |  | .npm |
| **go** | Go-Build Konfigstruktur | :tick: | :tick: | :tick: | :tick: |  | .go |
| **stages** | Stages Struktur für das Staging | :tick: | :tick: | :tick: | :tick: |  | .stages |
| **steps** | Steps Struktur für das Setup von verschiedenen Buildtypen | :tick: | :tick: | :tick: | :tick: |  | .steps → .pipelines |
| **helm** | Helm Konfigurationsstruktur | :tick: | :tick: | :tick: | :tick: |  | .helm |
| **credentials** | Credentials Konfigurationsstrutkur um Credentials der Pipeline mitzugeben | :tick: | :tick: | :tick: | :tick: |  | .pipelines[\*].envSecret |
| **notifications** | Notifications Konfigurationsstruktur | :cross: | :tick: | :tick: | :tick: |  | .notifications |
| **teamsNotifications** | Teams Konfigurationsstruktur um Notifications einem Teams Channel mitzugeben | :tick: | :tick: | :tick: | :tick: |  | .notifications[type=teams] |
|  |  |  |  |  |  |  |  |
| **Docker-Build Struktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** | .docker |
| **artifactoryDockerRepo (required)** | Docker Repo von Artifactory | :tick: | :tick: | :tick: | :tick: |  |  |
| **openShiftAppName** | Openshift Applikationsname (neu optional) | :tick: | :tick: | :tick: | :tick: | $.productName |  |
| **dockerFile** | Falls nicht angegeben Selfrunning Jar wird erstellt, sonst Dockerfile | :tick: | :tick: | :tick: | :tick: | "Dockerfile" |  |
| **imageName** | Default: openshiftAppName ; ansonsten Name des Dockerimages | :tick: | :tick: | :tick: | :tick: |  |  |
| **baseImageFrom** | Im Fall des Selfrunningjarbuild ist das das Baseimage welches wir erben | :tick: | :tick: | :tick: | :tick: |  |  |
| **ocBuildVars** | Eigene Umgebungsvariablen für den Build - Beispiel: "ocBuildVars:" "SNAPSHOTVERSION=1234, TIMESTAMP=121342323" | :tick: | :question: | :tick: | :tick: |  |  |
| **newRelic** | New Relic Konfigurationsstruktur für den SelfRunningJar Build | :tick: | :cross: | :question: | :question: |  |  |
| **checkOpenShiftDeploymentReadyEnabled** | Build wird nur erfolgreich sein, wenn das OpenshiftDeployment Ready ist | :tick: | :cross: | :cross: | :cross: |  |  |
| **artifactorySupressProjectSubfolderCreation** | Supresses the subfoldercreation in Artifactory. If set  **true**  your image will be published to i.e. [esta.docker.bin.sbb.ch/yourappname:version](http://esta.docker.bin.sbb.ch/yourappname:version), if  **false**your image will be published to [esta.docker.bin.sbb.ch/](http://esta.docker.bin.sbb.ch/) **esta**/yourappname:version  Evtl. müssen wird das soderso anderse lösen | :tick: | :cross: | :tick: | :tick: |  |  |
|  |  |  |  |  |  |  |  |
| **NewRelic-Configuration Struktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** |  |
| **appName** | Name of the application in New Relic. If not specified openshiftAppName is used. | :tick: | :cross: | :cross: | :cross: |  |  |
| **appLabel** | Application Labels in New Relic in the format "category1:value1, category2:value2". | :tick: | :cross: | :cross: | :cross: |  |  |
| **licenseKey** | New Relic License Key (required) | :tick: | :cross: | :cross: | :cross: |  |  |
| **apiKey** | New Relic API Key. Only needed if recordDeploymentInNewRelic is set to true. | :tick: | :cross: | :cross: | :cross: |  |  |
| **recordDeploymentInNewRelic** | If true, a marker is set in New Relic during deployment. | :tick: | :cross: | :cross: | :cross: |  |  |
|  |  |  |  |  |  |  |  |
| **Mvn Konfigurations-Struktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** | .mvn |
| **parentPom (required)** | Path to parent pom, default: pom.xml | :tick: | :tick: | :tick: | :tick: | "pom.xml" |  |
| **deployableArtifactsPom** | Path to pom which specifies the deployable artifacts, default: pom.xml | :tick: | :tick: | :tick: | :tick: |  |  |
| **artifactoryMavenRepo** | The Maven repository in Artifactory where the Maven artifacts will be published. | :tick: | :tick: | :tick: | :tick: |  |  |
| **mavenLabel** | The used Maven Version can be selected (  (  "Apache Maven 3.5",  "Apache Maven 3.6"  ) |  | :cross: | :question: | :tick: |  | Maven wird erst wenn nötig konfigurierbar. |
|  |  |  |  |  |  |  |  |
| **NPM-Konfigurations-Struktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** | .npm |
| **targetRepo** | Path to target | :tick: | :tick: | :tick: | :tick: |  |  |
| **additionalNpmScripts** | Npm build commands specified in package.json. Comma separated multiple commands can be specified. | :tick: | :tick: | :tick: | :tick: |  |  |
| **packageJson** | Specifies where the PackageJson is located | :tick: | :tick: | :tick: | :tick: | "package.json" |  |
| **publishablePackageJsons** | Specifies where the PackageJson are for publishable projects | :tick: | :tick: | :tick: | :tick: |  |  |
| **installDependencies** | Will excecute npm ci in the background. Default value: true | :tick: | :question: | :question: | :question: |  | Abklären was das genau macht? |
|  |  |  |  |  |  |  |  |
| **Stage Konfigurationsstruktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** | .stages[\*] |
| **stageName** | Name der Stage | :tick: | :tick: | :tick: | :tick: |  |  |
| **isOpenshiftDeploymentEnabled** | Deployment kann hier generell ein oder ausgeschaltet werden - soll über ArgoCDSync gesteuert werden. | :tick: | :cross: | :cross: | :cross: |  |  |
| **openshiftProject** | Openshift Namespace wohin gestaged werden soll | :tick: | :cross: | :cross: | :cross: |  |  |
| **openshiftCluster** | Openshift Cluster:  otc\_obelix,  otc\_test\_04,  aws  or vias | :tick: | :cross: | :cross: | :cross: |  |  |
| **openshiftJenkinsCredentialsId** | The Jenkins CredentialId: 34b64477-3960-4fec-8791-cc859e2abcde | :tick: | :cross: | :cross: | :cross: |  |  |
| **preCiDeploymentJob (preTektonPipeline)** | Jenkins Job welcher vor dem Staging ausgeführt werden soll | :tick: | :cross: | :cross: | :question: |  |  |
| **postCiDeploymentJob (postTektonPipeline)** | Jenkins Job welcher nach dem Staging ausgeführt werden soll | :tick: | :cross: | :cross: | :question: |  |  |
| **preDeployConfirmUserMails** | Ein Bestätigungs Email wird geschickt, in welchem der Benutzer vor einem Deployment das Deployment bestätigen muss. | :tick: | :cross: | :cross: | :question: |  | Das Deployment wird über unterschiedliche ArgoCD Instanzen mit unterschiedlichen Berechtigungen gelöst. |
| **kafkaDeployParameters** | A list of KafkaDeployParameters. Detailed documentation from the Kafka Team, see   [here](https://confluence.sbb.ch/x/2wBSSw) . | :tick: | :cross: | :cross: | :question: |  | Soll nur noch per Helm/ArgoCD gelöst werden  Wir würden allerdings die Tasks/Pipelines bereitstellen. |
| **api** | A list of APIM deployment parameters. For a more detailed documentation, see  [here](https://confluence.sbb.ch/x/fIeRSQ). | :tick: | :cross: | :cross: | :question: |  | Soll nur noch per Helm/ArgoCD gelöst werden  Wir würden allerdings die Tasks/Pipelines bereitstellen. |
| **instana** | Instana release parameters. Also check [Pipeline Helper#cloud\_createInstanaRelease-CreateanInstanarelease](https://confluence.sbb.ch/display/CLEW/Pipeline+Helper#PipelineHelper-cloud_createInstanaRelease-CreateanInstanarelease) | :tick: | :cross: | :cross: | :question: |  | Soll nur noch per Helm/ArgoCD gelöst werden  Wir würden allerdings die Tasks/Pipelines bereitstellen. |
| **appRegistration** | A list of "Apply Azure AD App-Registration" parameters. | :tick: | :cross: | :cross: | :question: |  | Soll nur noch per Helm/ArgoCD gelöst werden. Wir würden allerdings die Tasks/Pipelines bereitstellen. |
| **argoCD** | ArgoCD Konfiguration für diese Stage | :cross: | :tick: | :tick: | :tick: |  |  |
|  |  |  |  |  |  |  |  |
| **Step Konfigurationsstruktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** | .steps[\*] |
| **name (required)** | Eindeutiger Name des Steps | :cross: | :tick: | :tick: | :tick: |  |  |
| **stepType (required)** | Es gibt unterschiedliche Buildtypen (release, Continuous, integration, custom) | :tick: | :cross: | :cross: | :cross: |  |  |
| **isBuildEnabled (required)** | Ob diese Buildkonfiguration generell aktiv ist | :tick: | :tick: | :tick: | :tick: | false | .steps[\*].enabled |
| **failOnXrayScan (deprecated)** | Should the build fail if the XRayscan finds violations? (true / false) | :tick: | :cross: | :cross: | :cross: |  |  |
| **dependencyCheckEnabled** | Do you want to run the OWASP dependency check? (true / false) | :tick: | :cross: | :tick: | :tick: |  | **Qualityscan Konfigstruktur umbauen** |
| **dependencyCheckAdditionalParams** | Define  [additional scan params](https://jeremylong.github.io/DependencyCheck/dependency-check-cli/arguments.html) | :tick: | :cross: | :tick: | :tick: |  |  |
| **sonarScanEnabled** | Do you want the sonar scan done? (true / false) | :tick: | :cross: | :tick: | :tick: |  | Quality Scans prio? |
| **failOnQualityGateFailure** | If enabled the build waits for the result of the Quality Gate analysis and fails the build if it has not been passed. | :tick: | :cross: | :tick: | :tick: |  | Quality Scans? Wie oft benutzt? |
| **additionalBuildParams** | Additional build parameters such as -DskipITs which currently only affect Maven and NPM builds. | :tick: | :tick: | :tick: | :tick: |  |  |
| **additionalBuildParamsForSonarScan** | Additional build parameters such as -DskipTests=true applied for SonarQube scans for all build types. Default value is empty. | :tick: | :cross: | :tick: | :tick: |  | Was machen wir mit den ganzen Qualityscanner? V2 wird sicher ein Teil implementiert. |
| **additionalDockerImageTags** | Additional tags the Docker image shall be pushed to (comma separated) | :cross: | :tick: | :tick: | :tick: |  |  |
| **triggerBuildOnCommitEnabled** | Should a Jenkins build be executed at a Bitbucket event on the mainBranch? (true | false) | :tick: | :cross: | :cross: | :cross: |  |  |
| **triggerType** | We can have here different Triggertypes like (GITEVENT, USER/MANUAL,CRON) | :cross: | :tick: | :tick: | :tick: |  |  |
| **staging (simple, bluegreen, custombranch, subsequent)** | Das Stagingproperty wird benötigt um die verschiedenen Stagingtypen zu unterstützen. | :tick: | :cross: | :cross: | :cross: |  | ArgoCD, ArgoRollouts |
| **stages** | Different stages to trigger deployment in ArgoCd | :cross: | :tick: | :tick: | :tick: |  |  |
| **deployArtifacts** | Should the generated artifacts (Mvn, NPM, Go) be deployed to Artifactory? Default is false, since this is should needed in case of library builds. | :tick: | :cross: | :cross: | :cross: |  | Wird mit buildDockerImage und packageAndDeployHelmChart ersetzt. |
| **buildDockerImage** | If true, DockerImage will be built and deployed to Artifactory | :cross: | :tick: | :tick: | :tick: | false |  |
| **packageAndDeployHelmChart** | If true, HelmChart will be packaged and deployed to Artifactory | :cross: | :tick: | :tick: | :tick: | false |  |
| **junitTestReportPath** | With that property you can declare, where the surfire-reports of this build will be found. The report will be shown in Jenkins afterwards.  e. g. : "junitTestReportPath": "\*\*/surefire-reports/\*.xml", | :tick: | :cross: | :question: |  |  |  |
| **hotfixCreationEnabled (Release Build only)** | Falls true wird am Ende des builds ein hotfix Branch erstellt wird. | :tick: | :tick: | :question: |  |  | Statt Flag ein Pattern für den zu kreierenden Hotfix Branch angeben. |
| **triggerIntegrationBuildAfterReleaseEnabled (Release Build only)** | Falls true wird am Ende des Builds ein Integration Build getriggered. | :tick: | :cross: | :question: |  |  |  |
| **branchNamePrefixes (Continuous Build only)** | Branchname Filter Liste auf welchen dieser Buildtyp hören soll. | :tick: | :tick: | :tick: | :tick: |  |  |
| **versionTagEventPatterns** | Regular expression to match Git tags in order to execute the step. A regex like ^(\d+\.)?(\d+\.)?(\\*|\d+)$ would match a version like 10.5.100 | :cross: | :tick: | :tick: | :tick: |  |  |
| **preCiStepJob (preTektonPipeline)** | Jenkinsjob welcher vor dem Build ausgeführt wird. | :tick: | :cross: | :tick: | :tick: |  | Kann mit Custom Tasks realisiert werden |
| **postCiStepJob (postTektonPipeline)** | Jenkinsjob welcher nach dem Build ausgeführt wird. | :tick: | :cross: | :tick: | :tick: |  | Kann mit Custom Tasks realisiert werden |
| **isS2IBuildActive** | Source to Image Build aktiv | :tick: | :cross: | :cross: | :cross: |  |  |
|  |  |  |  |  |  |  |  |
| **Helm Konfigurationsstruktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** | .helm |
| **release** | Helm release name to deploy. (mandatory) | :tick: | :tick: | :tick: | :tick: |  |  |
| **chartRootFolder** | Relative path to the Helm Chart root folder. (Default: ./charts) | :tick: | :tick: | :tick: | :tick: |  |  |
| **chart** | Relative path to the chart within chartRootFolder. If not specified the productName is used | :tick: | :tick: | :tick: | :tick: |  |  |
| **versionProperty** | Name of the property/field inside the chartFile which holds the version string. | :cross: | :cross: | :tick: | :tick: |  |  |
| **repository** | The name of the Helm repository in Artifactory. If specified the chart is packaged and uploaded to this repository. Make sure that you do not specify a virtual repository. | :tick: | :tick: | :tick: | :tick: |  |  |
| **timeoutInSeconds** | Time to wait for the deployment to be completed. Defaults to 600s. | :tick: | :cross: | :cross: | :cross: |  |  |
|  |  |  |  |  |  |  |  |
| **Go Konfigurationsstruktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** | .go |
| **targetRepo** | Target Artifactory repository name. | :tick: | :tick: | :tick: | :tick: |  |  |
| **buildPackages** |  | :cross: | :tick: | :tick: | :tick: |  |  |
| **useMage** |  | :tick: | :tick: | :tick: | :tick: |  |  |
| **additionalGoParams** |  | :tick: | :tick: | :tick: | :tick: |  |  |
| **analysisExclusions** |  | :tick: | :cross: | :question: | :tick: |  |  |
|  |  |  |  |  |  |  |  |
| **ArgoCD Konfigurationsstruktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** | .stages[\*].argoCD |
| **argoCdAppName** | The app name which will be referenced in ArgoCd. Default value is the top-level productName property. | :cross: | :tick: | :tick: | :tick: | $.productName |  |
| **chartRepository** | The repository where your helm charts for deployments are. Default is the current repository. | :cross: | :tick: | :tick: | :tick: |  |  |
| **chartBranchOrTag** | Git branch where ArgoCD should fetch the charts from. | :cross: | :tick: | :tick: | :tick: | "master" |  |
| **autoSync** | If true, the deployment pipeline will trigger an ArgoCD sync. | :cross: | :tick: | :tick: | :tick: | false |  |
|  |  |  |  |  |  |  |  |
| **Credentials** **Konfigurationsstruktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** | Credentials werden in OpenShift Secrets gespeichert |
| **id** | Id of Jenkins secret to lookup | :tick: | :cross: |  |  |  |  |
| **type** | Jenkins credentials type, possible values: string, usernamePassword | :tick: | :cross: |  |  |  |  |
| **variable** | Name of the environment variable containing the secret to set | :tick: | :cross: |  |  |  |  |
| **usernameVariable** | Name of the environment variable containing the username to set | :tick: | :cross: |  |  |  |  |
| **passwordVariable** | Name of the environment variable containing the password to set | :tick: | :cross: |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **TeamsNotification** **Konfigurationsstruktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** | .notifications[type=teams] |
| **url** | URL from the Microsoft Teams connector configuration. Required. | :tick: | :cross: |  |  |  | .notifications[\*].webhookUrl |
| **startNotification** | Notify build start | :tick: | :cross: |  |  |  |  |
| **notifySuccess** | Notify a successful build | :tick: | :cross: |  |  |  |  |
| **notifyAborted** | Notify an aborted build | :tick: | :cross: |  |  |  |  |
| **notifyNotBuilt** | Notify a build that was not built | :tick: | :cross: |  |  |  |  |
| **notifyUnstable** | Notify an unstable build | :tick: | :cross: |  |  |  |  |
| **notifyFailure** | Notify a build failure | :tick: | :cross: |  |  |  |  |
| **notifyBackToNormal** | Notify back to normal status | :tick: | :cross: |  |  |  |  |
| **notifyRepeatedFailure** | Notify repeated failure | :tick: | :cross: |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **APIM** **Konfigurationsstruktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** |  |
| **instance** | Instance of APIM platform. Values: int / prod (dev is only not open for the public) | :tick: | :cross: |  |  |  |  |
| **clientSecret** | Secret of API Client | :tick: | :cross: |  |  |  |  |
| **clientId** | ID of API Client with permissions on the API (Client Credentials, Service Owner Role of API) | :tick: | :cross: |  |  |  |  |
| **clientJenkinsId** | Secret ID of API Client with permissions on the API (Client Credentials, Service Owner Role of API) | :tick: | :cross: |  |  |  |  |
| **classifier** | ConfigClassifier For ApimConfigurationFiles | :tick: | :cross: |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Instana** **Konfigurationsstruktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** |  |
| **apiTokenId** | Instana API Token ID. Only used if recordDeployment is true. Default value: null | :tick: | :cross: | :question: |  |  |  |
| **environment** | Instana environment to use, enum of test|prod, default: test | :tick: | :cross: | :question: |  |  |  |
| **recordDeployment** | If true a new deployment is recorded in Instana. Also refer to  <https://www.instana.com/docs/pipeline_feedback/configuration-and-api/>). Default value: true | :tick: | :cross: | :question: |  |  |  |
|  |  |  |  |  |  |  |  |
| **Kafka** **Konfigurationsstruktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** |  |
| **type** | either topics or users | :tick: | :cross: | :question: |  | Evtl. Gitops mit Charts lösen / Mit Kafkateam besprechen |  |
| **environmentId** | Kafka environment (test, inte, prod) | :tick: | :cross: | :question: |  |  |  |
| **clientId** | Id of the Client API. Further information, see  [here](https://confluence.sbb.ch/x/DgzFUQ) . | :tick: | :cross: | :question: |  |  |  |
| **clientSecret** | Secret of the API client | :tick: | :cross: | :question: |  |  |  |
| **fileGlob** | Glob pointing to the kafka-resources. Default value: kafka/{type}/{environmentId}/\* | :tick: | :cross: | :question: |  |  |  |
| **kafkaAutomationUrl** | kafkaAutomation URL. Default value:  <https://kafka-automation-prod.app.obelix.otc.sbb.ch> | :tick: | :cross: | :question: |  |  |  |
| **ssoUrl** | sso URL. Default value:  <https://sso.sbb.ch/auth/realms/SBB_Public/protocol/openid-connect/token> | :tick: | :cross: | :question: |  |  |  |
|  |  |  |  |  |  |  |  |
| **Azure AD** **Konfigurationsstruktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** |  |
| **instance** | Instance resp. stage of Azure AD API. Possible values:  dev, int ,  prod | :tick: | :cross: | :tick: | :tick: | Soll das mit Plugins gelöst oder fix implementiert werden?  Kann mit Helmcharts ein REST call ausgeführt werden? |  |
| **appRegistrationConfigPath** | Path to the config file (from the repository root): e.g. "app-registration/esta-iam-blueprints-app-registrations.yml" | :tick: | :cross: | :tick: | :tick: |  |  |
| **clientId** | Id of the client that has been registered in the  [SBB developer portal](https://developer.sbb.ch/apis)  to access the Azure AD API (Client Credentials flow). | :tick: | :cross: | :tick: | :tick: |  |  |
| **clientSecret or  jenkinsCredentialsId** | One of the two parameters must be given: either the client secret or (preferably!) the jenkinsCredentialsId referencing the secret. | :tick: | :cross: | :tick: | :tick: |  |  |
|  |  |  |  |  |  |  |  |
| **CiJobParameters** **Konfigurationsstruktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** |  |
| **upstreamGitRepoUrl** | SSH url of the upstream repository | :tick: | :cross: |  |  |  |  |
| **upstreamGitRepoBranch** | Branch of the upstream repository | :tick: | :cross: |  |  |  |  |
| **upstreamGitRepoCommit** | Commit id of the upstream repository | :tick: | :cross: |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Notifications Konfigurationsstruktur** | **Feature Beschreibung** | **EstaCloud Pipeline** | **V1** | **V2** | **Final** | **Default** | .notifications[\*] |
| **type (required)** | Type/delivery of notification messages ["EMAIL","TEAMS"] | :cross: | :tick: | :tick: | :tick: | "email" |  |
| **sendingEnabled** | Flag to enable/disable sending of notifications | :cross: | :tick: | :tick: | :tick: | true |  |
| **eventTypes** | Filter for events that trigger this notification ("START","SUCCESS","FAILURE","ABORTED"). If empty, all events will trigger notifications. | :cross: | :tick: | :tick: | :tick: |  |  |
| **pipelineTypes** | Filter for pipeline types trigger this notification ("TAGGING","BUILD","STAGING","CUSTOM"). If empty, all pipeline types will trigger notifications. | :cross: | :tick: | :tick: | :tick: |  |  |
| **title** | The message title. Default title is taken from the global config map of the ESTA Tekton Controller | :cross: | :cross: | :question: | :tick: |  |  |
| **message** | The message body. Default value is taken from the global config map of the ESTA Tekton Controller | :cross: | :cross: | :question: | :tick: |  |  |
| **recipients** | List of recipient email addresses for "email" type notifications. The String "$actor" will be replaced with the email address of user who started the pipeline. | :cross: | :tick: | :tick: | :tick: |  |  |
| **webhookUrl** | Webhook URL for "teams" type notifications | :cross: | :tick: | :tick: | :tick: |  |  |
| **Nicht konfigurative Features** |  |  |  |  |  |  |  |
| Support von Conventional commits | <https://www.conventionalcommits.org/en/v1.0.0/> | :cross: | :cross: | :question: |  |  |  |

## Custom Pipelines (Draft)

Es versteht sich, dass die Standard Pipelines für Maven, NPM, Go und Docker Builds, welche von ESTA Tekton angeboten werden, nicht für alle Projekte und Bedürfnisse genügen. Einer der Vorteile von Tekton ist, dass man einfach eigene Pipelines erstelllen kann und dabei auf bestehende Tasks von ESTA Tekton oder aus dem öffentlichen [Tekton Katalog](https://hub.tekton.dev/) zugreifen kann. Die Unterstützung für Custom Pipelines in ESTA Tekton soll daher möglichst einfach umgesetzt werden und in einer Form, wo die Funktionalität des ESTA Tekton Controllers für das Management bestmöglichst genutzt werden kann. So sehen wir zwei Stufen für die Definition und Anwendung von Custom Pipelines in Repos resp. Projekten vor:

### Custom Pipeline Templates im Projekt-Repository

Templates für Custom Pipelines resp. PipelineRuns werden direkt im Repository gehalten, wo auch das estaTektonPipeline.json liegt und von selbigem referenziert.

Als Konvention gilt, dass die PipelineRun Templates im Ordner "tekon" gespeichert und "self-contained" sind. Das bedeutet, sie werden vom ESTA Tekton Controller direkt via oc process -f <file> verarbeitet und nicht im OpenShift Namespace applied. Sie enthalten vorzugsweise die [Pipeline Definition](https://tekton.dev/docs/pipelines/pipelineruns/#specifying-the-target-pipeline) inline mit pipelineSpec, wie in [diesem Beispiel](https://code.sbb.ch/projects/KD_ESTA/repos/esta-tekton-pipeline-builder-image/browse/tekton/esta-builder-image-build-pipeline-run.yaml) gezeigt. Damit wird gewährleistet, dass die Custom Pipeline in der richtigen Version aus dem entsprechenden Branch/Tag/Commit gestartet wird.

### Custom Repository mit Pipeline und Task Definitionen

Für grössere Projekte können Tekton Tasks und Pipelines in einem eigenen Repository verwaltet werden. Diese werden dann vom Controller in den entsprechenden OpenShift Namespace importiert, analog zu den Standard ESTA Tekton Pipelines. Sie sind dann also global und in der im Controller konfigurierten Version vorhanden und werden über ihre eindeutigen Namen gestartet oder referenziert. Über diesen Mechanismus können die Tasks und Pipelines von ESTA Tekton ergänzt oder ggf. sogar überschrieben werden.

### Integration in estaTektonPipeline.json

Die Verwendung von Custom Pipelines wird über die steps Definition im estaTektonPipeline.json gesteuert und zwar über das Propertry customPipelines mit einer Liste von Pipeline Spezifikationen.

Eine Custom Pipeline wird dabei über eine der folgenden Properties angegeben:

"pipelineTemplate": "<Name oder Datei des PipelineRun Templates>"   
"pipelineRef": "<Name der Pipeline>"

Mit pipelineTemplate wird auf ein [OpenShift Template](https://docs.openshift.com/container-platform/4.7/openshift_images/using-templates.html) verwiesen, welches einen PipelineRun und allfällige zusäzliche Resourcen wie PVCs erstellt. Endet das Property mit .yaml wird die enstprechende Datei im Ordner "tekton" des Repositors verwendet. Ansonsten wird der Wert als Name für ein im Namespace vorhandenes Template betrachtet.

Ist stattdessen pipelineRef in der Step Konfiguration angegeben, wird dies als Referenz auf eine im Namespace vorhandene [Pipeline](https://tekton.dev/docs/pipelines/pipelines/) interpretiert und über ein Standard PipelineRun Template gestartet.

**Pipeline Parameter**

Zusätzliche Parameter, welche an die PipelineRuns übergeben werden, können in der Custom Pipeline Konfiguration im estaTektonPipeline.json definiert werden:

"params": {  
 "VERSION\_TAG": "snapshot.${DATE}",  
 "IMAGE\_LABELS": "buildtype=snapshot"

}

Die Parameter Values können interpolierte Strings enthalten, welche mit vom Controller global definierten Parametern (${PARAM\_NAME}) oder Werten aus dem estaTektonPipeline.json (${/json/pointer}) ersetzt werden.

**Beispielkonfiguration eines Steps mit Custom Pipeline**

Eine Beispielkonfiguration eines Buildsteps könnte so aussehen. Wie man sieht, können auch mehrere Pipelines definiert werden, welche vom Controller parallel gestartet werden:

{  
 "name": "snapshot",  
 "enabled": true,  
 "buildDockerImage": true,  
 "triggerType": [  
 "GITEVENT"  
 ],  
 "customPipelines": [  
 {  
 "pipelineTemplate": "custom-build-pipeline-run-template.yaml",  
 "params": {  
 "IMAGE\_TAG": "snapshot.${DATE}",  
 "APP\_NAME": "${/stages/1/argoCD/argoCdAppName}"

}  
 }  
 ]  
}

## Controller REST API (Draft)

|  |  |  |  |
| --- | --- | --- | --- |
| **Route** | **Methode** | **Beschreibung** | **Payload** |
| /api/bitbucket/projects | GET | Listet alle Bitbucket Projekte, welche im Controller registriert sind |  |
| /api/bitbucket/projects/:pid | GET | Details zu einem Bitbucket Projekt |  |
| /api/bitbucket/projects/:id/repos | GET | Listet alle Tekton-enabled Bitbucket Repositories im Projekt |  |
| /api/bitbucket/projects/:id/repos/:rid | GET | Details zu einem Bitbucket Repository inkl. Branches und estaTektonPipeline.json |  |
| /api/bitbucket/projects/:id/repos/:rid/branches | GET | Gibt die Branches zu einem Repository zurück |  |
| /api/bitbucket/projects/:id/repos/:rid/branches/:branch/build | POST | Startet einen Pipeline Run für den gegebenen Branch | {} |
| /api/bitbucket/projects/:id/repos/:rid/branches/:branch/tag | POST | Startet einen Tagging Pipeline Run ür den gegebenen Branch | {"versionTag":"1.2.3.0"} |
| /api/pipelinerun?repository=kd-esta--foo&type=\* | GET | Listet alle Pipeline Runs welchen den angegegebenen Selektoren entsprchen |  |
| /api/pipelinerun/:id | GET | Details zu einem Pipeline Run |  |
| /api/pipelinerun/:id | DELETE | Löschen eines Pipeline Runs mit allen verknüpften Resourcen |  |
| /api/pipelinerun/:id/logs[/stream] | GET | Logs eines Pipeline Runs |  |
| /api/pipelinerun/:id/rerun | POST | Neustart des Pipeline Runs |  |
| /api/pipelinerun/:id/cancel | POST | Abbrechen eines laufenden Pipeline Runs |  |
| /api/pipelinerun/:id/delete | DELETE | Stop und Löschen eines Pipeline Runs |  |

Die Aktuelle Version des APIs wird mit Swagger/OpenAPI dokumentiert: <https://esta-tekton-controller-esta-tekton-dev.sbb-cloud.net/swaggerui/>