TEKTON

Tekton is a powerful and flexible open-source framework for creating CI/CD systems, allowing developers to build, test, and deploy across cloud providers and on-premise systems.

**Note**: Tekton can either be used as a standalone tool or can be integrated/combined with other CI/CD tools like Jenkins or argoCD to create powerful CI\CD solutions.

Tekton vs Jenkins

|  |  |
| --- | --- |
| **Tekton** | **Jenkin** |
|  |  |
| 1. Tekton is serverless and distributed, and there is no central dependency for execution. | 1. Jenkins has a control node. Jenkins executes pipelines and steps centrally or orchestrates jobs running in other nodes. Requires dedicated server and JVM to run continuously. |
| 1. Everything is container based. Runs in form of CRD, Pods. | 1. The containers are launched by the control node through the pipeline |
| 1. Extensibility achieved by tasks in Tekton Hub, or by creating custom tasks and scripts. | 3. Extensibility is achieved using plugins. |
| 1. Kubernetes native | 4. Was not created using cloud in view. |
| 5. YAML based so easier for Kubernetes developers. | 1. Groovy language used. So, developer may need to learn groovy basic. |
| 6. Easy Installation. | 1. Complex Installation. |
| 7. Fairly new and still under development. | 1. Highly used and is highly stable. |
| 8. Flexible, modular, scalable. | 1. Large dependency on availability of slaves. |
| 9. Easier reusability of yaml files. | 1. Each project will need a separate jenkinfiles. Although common code can be copied for similar projects. |
|  |  |

PROS of tekton:

* Pipeline configuration is stored separately from code. So, tasks can be reused by several projects.
* As developer writes the task script, it provides developer freedom to utilize any tool/tech. We can say it is a DIY CI\CD.
* PipelineRun and Taskrun can be created using UI via dashboard.
* Dashboard can be used to check, delete resources/component.
* Tekton can easily be integrated with other tools like argoCD to create better CI/CD system. Tekton plugins are also present in Jenkins.
* Jenkin X internally uses tekton. So, it is being highly used for modern CI tools.

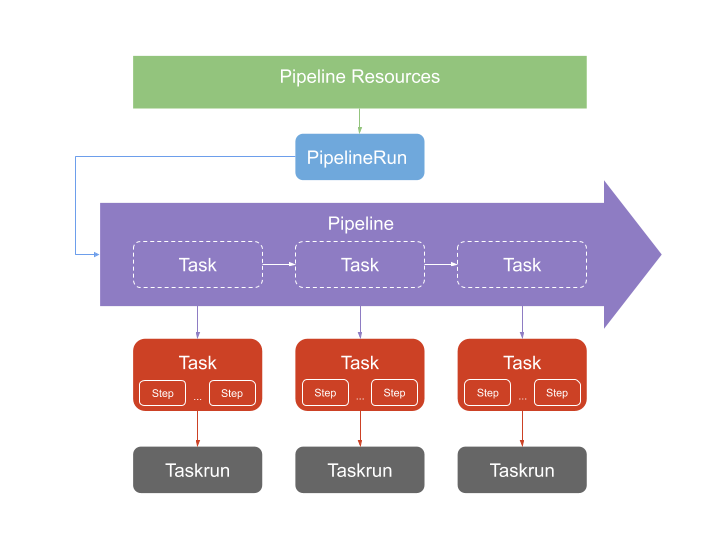
CONS of tekton:

* Manual modification and application of yaml files.
* Requires knowledge of Kubernetes, scripting. This is essential for creating custom tasks.
* There is an initial learning curve required at start to understand basic and usecase of tekton.

Tekton components

The below diagram shows the basic required component of a tekton pipeline.

For detail regarding these, check link: <https://tekton.dev/docs/getting-started/>



Optional component includes dashboard, CLI, catalog, eventlistener, trigger.

Scenarios covered till now:

1. Create webhook and eventlistener to automate pipelineRun.
2. Clone Git repo.
3. Use dockerfile to build image.
4. Upload to docker Hub.
5. Modify deployment yaml to include latest image digest.
6. Use deployment yaml to deploy application.
7. Try using multiple credentials in pipeline (using workspace).
8. Try multi-branch build.
9. Try deploying to a different cluster.
10. Use multiple tools in a step (by installing tools as required.).
11. Try using webhook payload for getting dynamic values (like refs, project name, commit id) while creating pipelineRun.
12. While deployment, add check to confirm if namespace is already existing before executing command to create namespace.

Example code for above scenarios stored at below location: <https://github.com/Priyakr2205/TektonTrail>

Scenario/features remaining to check:

1. Try staging deployment.
2. Check how to do integration testing / verify deployment is successful.
3. Try webhook with secret.
4. Try ‘when’ condition in Task.
5. Check if multiple images can be used in a single task.
6. Add conditions to trigger specific task based on some filter like branch name.
7. Check versality of tekton dashboard (how user-friendly it is). Check addition of extensions.
8. Check how to add test stage before build.
9. How to add cloudevent to notify success or failure. Integration with slack/team or email.
10. Auto cleanup of pods deployed by pipeline (This may require creation of another pipeline for deleting pod. Currently, pods need to be removed manually.)
11. Study about tekton HA feature.

Help Links:

Tekton doc: <https://tekton.dev/docs/getting-started/>

tekton yq help doc: <https://mikefarah.gitbook.io/yq/v/v4.x/upgrading-from-v3#updating-writing-documents>

github token: <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token#creating-a-token>

tekton trigger and eventlistner: <https://github.com/tektoncd/triggers/tree/main/docs/getting-started>

tekton install: <https://github.com/tektoncd/dashboard>

namespace: <https://www.codegrepper.com/code-examples/shell/write+shell+script+to+check+whether+a+namespace+exists+in+kubernetes>

getting-started: <https://medium.com/hiredscore-engineering/getting-started-with-tekton-pipelines-part-1-container-image-build-and-push-5de1f96b0515>

tekton vs Jenkins: <https://www.peerspot.com/questions/how-does-tekton-compare-with-jenkins>

tekton vs Jenkins: <https://docs.openshift.com/container-platform/4.8/cicd/jenkins-tekton/migrating-from-jenkins-to-tekton.html#:~:text=Jenkins%20executes%20pipelines%20and%20steps,control%20node%20through%20the%20pipeline>.

Jenkin vs tekton: <https://medium.com/@sequeira.i/moving-from-jenkins-to-tekton-2d0b7b3b3067>

finally task in tekton: <https://developer.ibm.com/blogs/add-finally-to-tekton-pipelines/>

tekton vs jenkin: <https://www.peerspot.com/questions/how-does-tekton-compare-with-jenkins>

tekton trigger: <https://tekton.dev/vault/triggers-main/>

Video:

<https://www.youtube.com/watch?v=TWxKD9dLpmk>

<https://morioh.com/p/2ed3eb3d185c>

<https://www.youtube.com/watch?v=CnVCgMRE4xI>

<https://blog.sebastian-daschner.com/entries/cloud-native-ci-cd-tekton-argocd-video-course>