

USING SPINNAKER WITH KUBERNETES FOR CONTINUOUS DELIVERY



In this blog post, we are going to talk about doing continuous delivery with two really cool platforms, Spinnaker and Kubernetes. We will:

- Briefly go over both platforms
- Dig into the reason to use both together for CD
- Look at some use cases

Describing Spinnaker and Kubernetes

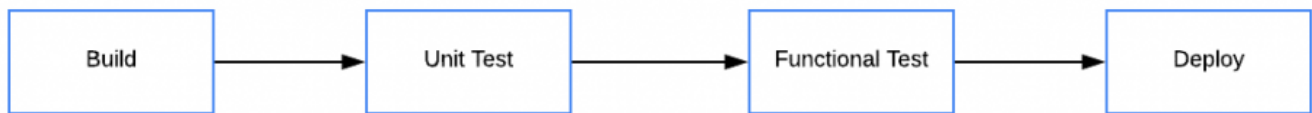
As a refresher,

- [Spinnaker](#) enables teams to automate deployments across multiple cloud accounts and regions, and even across multiple cloud platforms, into pipelines that are run whenever a new version is released.
- [Kubernetes](#) is a container/microservice platform that orchestrates computing, networking, and storage infrastructure workloads developed by google.

With these two platforms you can [build a solid continuous delivery \(CD\) pipeline](#) with best practices already built into it.

Continuous Delivery with Spinnaker

Code changes should naturally go through a spinnaker pipeline that looks like the following diagram



In the simplified pipeline above, we do not have any complicated logic in it so it can be achieved with tools like Jenkins 2.0 which has its own CD extension since it was not originally built for CD. However, Spinnaker was built to combine CI with CD in one platform and on top of that it has the added benefit of managing your deployed cloud applications. With Spinnaker in the mix, you are guaranteed to have a safe deployment every time.

Leveraging Kubernetes

Spinnaker was built to be multi-cloud, so there is a strong case for using it to deploy to cloud services like AWS and GCP but the way Kubernetes was built makes it a perfect fit for Spinnaker. K8s was built with the following in mind:

- Fast deployment
- Declarative
- Multi cloud
- Rollout is according to policy define

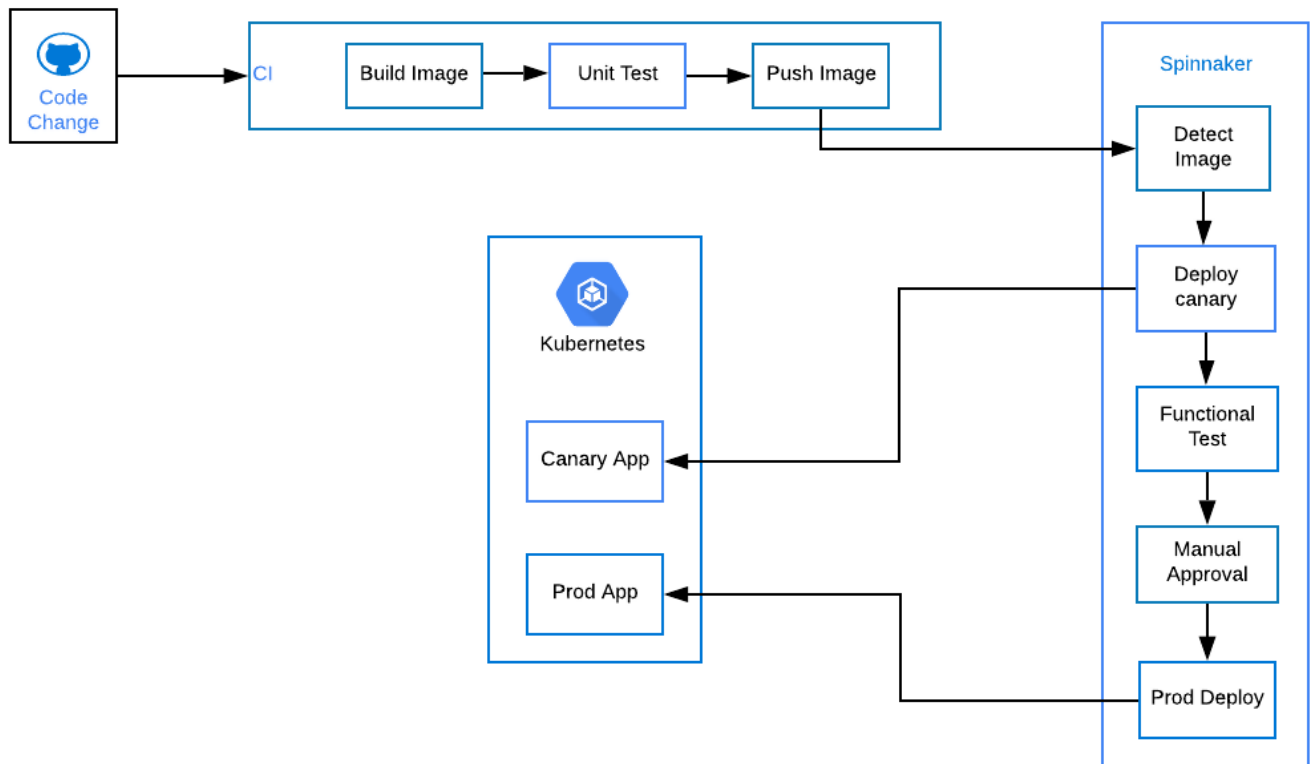
Robust CD Pipeline

By combining both of these platforms, we can build a fully automated CD pipeline that allows us to deploy applications with best practices built into it. By leveraging Spinnaker and k8s, we get:

- Seamless [resource mapping](#) especially with the v2 of the k8s.
- Advanced deployment strategy like [Canary](#).
- Deployment rollback.
- Spinnaker first class support for k8s unlike tools like Jenkins.
- Spinnaker will version deployed manifest to k8s to use for rollbacks.
- Spinnaker will monitor the health of your application in k8s. It allows not only to deploy to but gives a good view of the health of the k8s application and cluster.
- Integration for CI tools like Jenkins to pull artifacts.

Use cases

There are a lot of use cases. A simple use case could be detecting a container image push, performing some tests and deploying that image to k8s if test pass. The diagram below shows what the flow looks like.



Another use case is to use spinnaker to build an AMI in one of the stages with the help of packer, then deploying that AMI to production after test passes with the benefit of seeing the state of our deployed k8s application. One other way people use both together is to deploy spinnaker itself on k8s as a deployment, and using that deployed spinnaker cluster to deploy to the host k8s cluster.

Spinnaker & K8s for CD best practices

We discussed how to integrate both and how some of their features complement each other. To give it a try, Installation of both [Spinnaker](#) and [Kubernetes](#) is very well documented and that includes integration.

Additional resources

For more on Kubernetes, explore these resources:

- [Kubernetes Guide](#), with 20+ articles and tutorials
- [BMC DevOps Blog](#)
- [Bring Kubernetes to the Serverless Party](#)
- [How eBay is Reinventing Their IT with Kubernetes & Replatforming Program](#)