Continuous Delivery with Spinnaker on Kubernetes Cluster

Ananda Dwi Rahmawati
ananda@btech.id

OICNDI

November 7th 2020

vmware boer technology onf

About Me

Cloud Engineer [at] Boer Technology

Applied Undergraduate Student [at] UGM

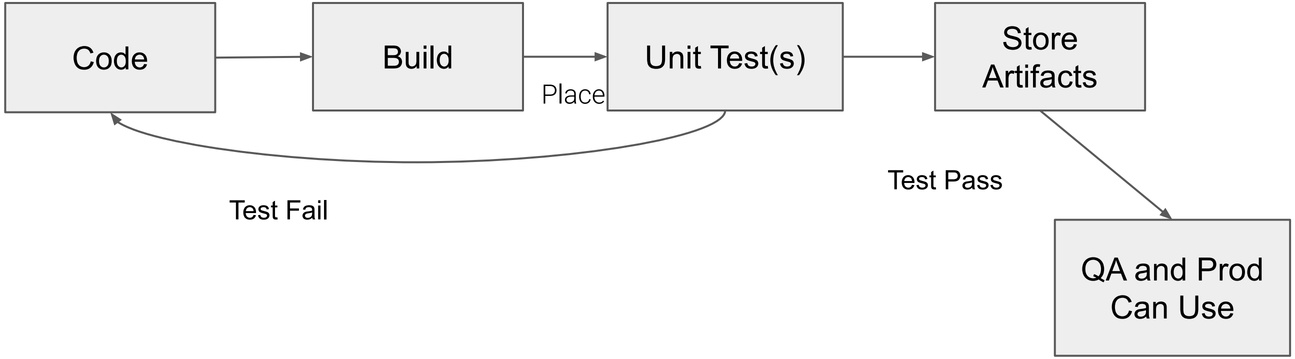
Activist [at] BlankOn

Tech Enthusiast

Keep in touch with me [at] @misskecupbung

Requirement(s)

- Bash Shell
- Cloud Provider
- Docker
- Kubernetes



Code

Build

Place

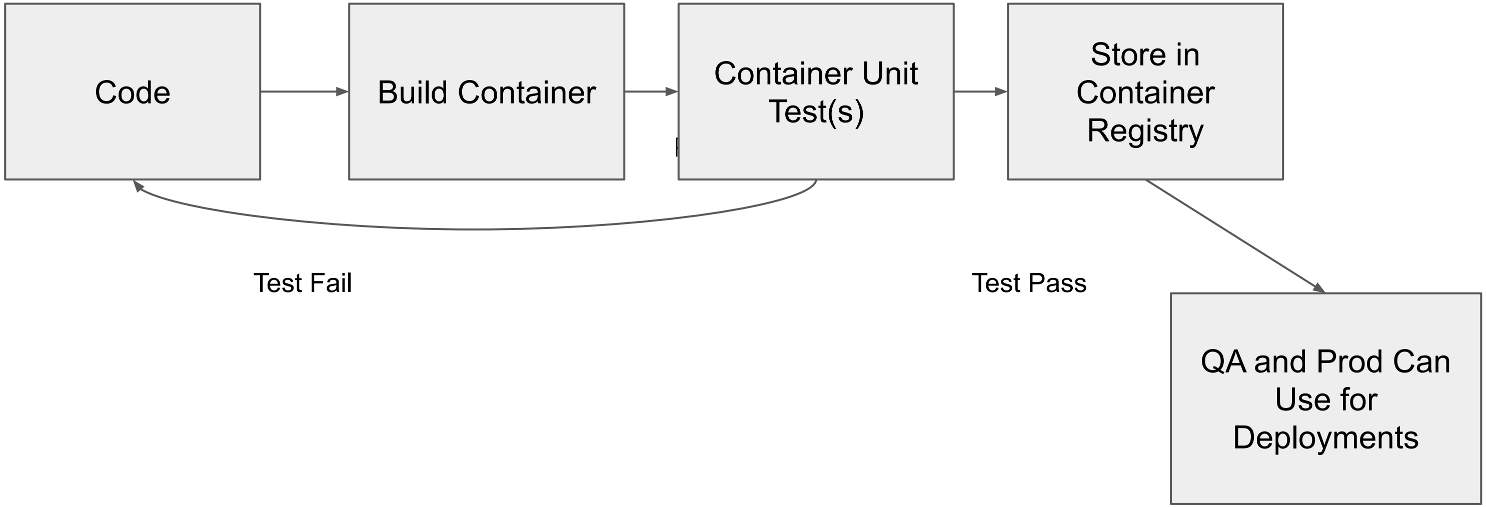
Unit Test(s)

Store Artifacts

Test Fail

Test Pass

QA and Prod Can Use



Code

Build Container

Container Unit Test(s)

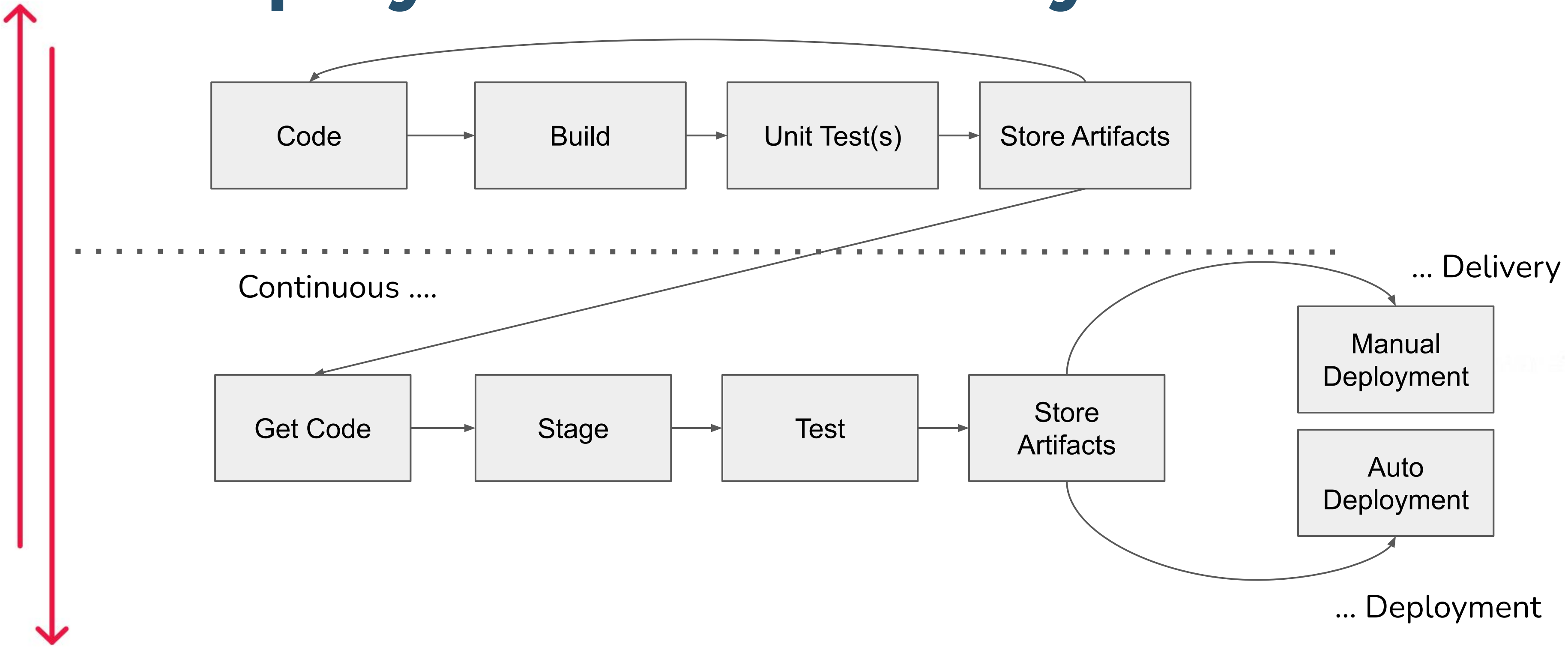
Store in Container Registry

QA and Prod Can Use for Deployments

Test Fail

Test Pass

Deployment or Delivery?



Code

Build

Unit Test(s)

Store Artifacts

Continuous ....

Get Code

Stage

Test

Store Artifacts

... Delivery

Manual Deployment

Auto Deployment

... Deployment

Deployment or Delivery?

- Delivery:
  - Land of Operator
  - Deployment may be automated, but often as a part of idempotent deployment tools (e.g Ansible, Puppet, etc)
- Deployment:
  - Release and idempotently deploy an application
  - Need to support rollback
  - May still make us of deployment tools
  - Containers make this much simpler

Spinnaker

- Pronunciation → spĭn'əkər
- Open Source multi-cloud CD platform | Inventory + Pipelines
- A supplemental sail to the main sail, especially a triangular one, used on yachts for running before the wind
- Initially developed by Netflix's Asgard (2014), Open-Sourced in 2015 | Built for releasing software changes with high velocity, confidence | Designed with pluggability in mind
- Support for all major Cloud Provider (App Engine, GCP, Azure, AWS, DC/OS, Oracle Cloud, Cloud Foundry)

Spinnaker Core Features

- Cluster Management
- Deployment management
- Multi-cloud capable
- Deployments are built-in and no custom scripting in needed

Spinnaker Advantages

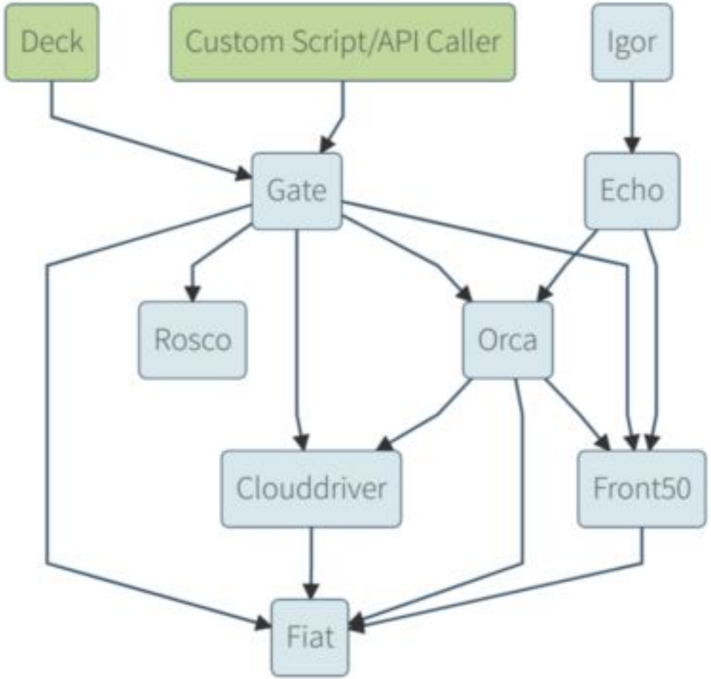
- Multi-Cloud CI/CD
- Variable pipeline type, easy rollback
- Flexible pipeline management system
- Variable deployment strategy (Blue-Green, Rolling Red/Back, Canary)
- Hybrid Cloud (VM/Container)
- CI (Jenkins and more)
- Halyard CLI
- Packer, Helm Packaging, Terraform,
- RBAC
- Notification: email, slack, and even sms
- Safe deployment: judgement
- Chaos monkey built-in
- Community (github, slack, https://community.spinnaker.io)

Spinnaker Architecture

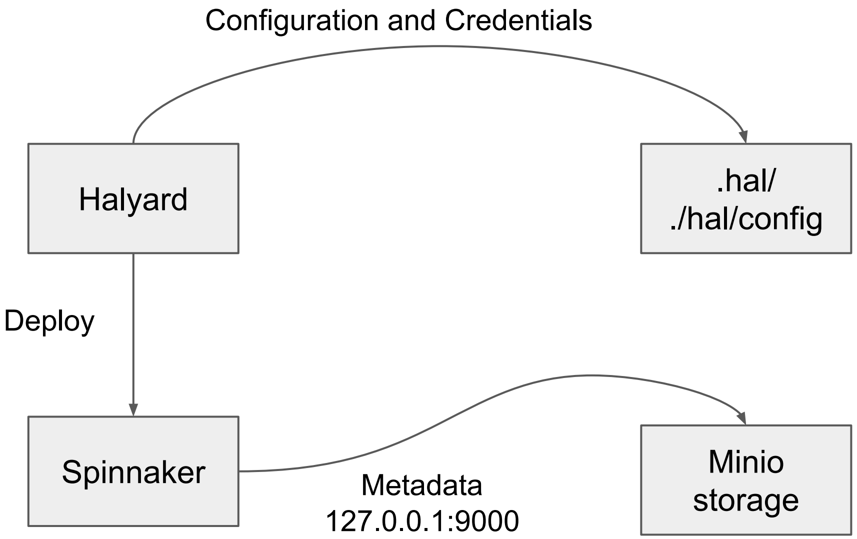
- Deck: Web-Based UI
- Gate: API Gateway
- Orca: Orchestration Engine
- Clouddriver: Indexing/Caching
- Front50: Metadata
- Rosco: Prod VM Images

- Igor: Trigger Pipeline
- Echo: eventing bus
- Flat: Auth service
- Kayenta: Canary Analysis
- Halyard: Configuration service

Spinnaker Architecture



Spinnaker Halyard

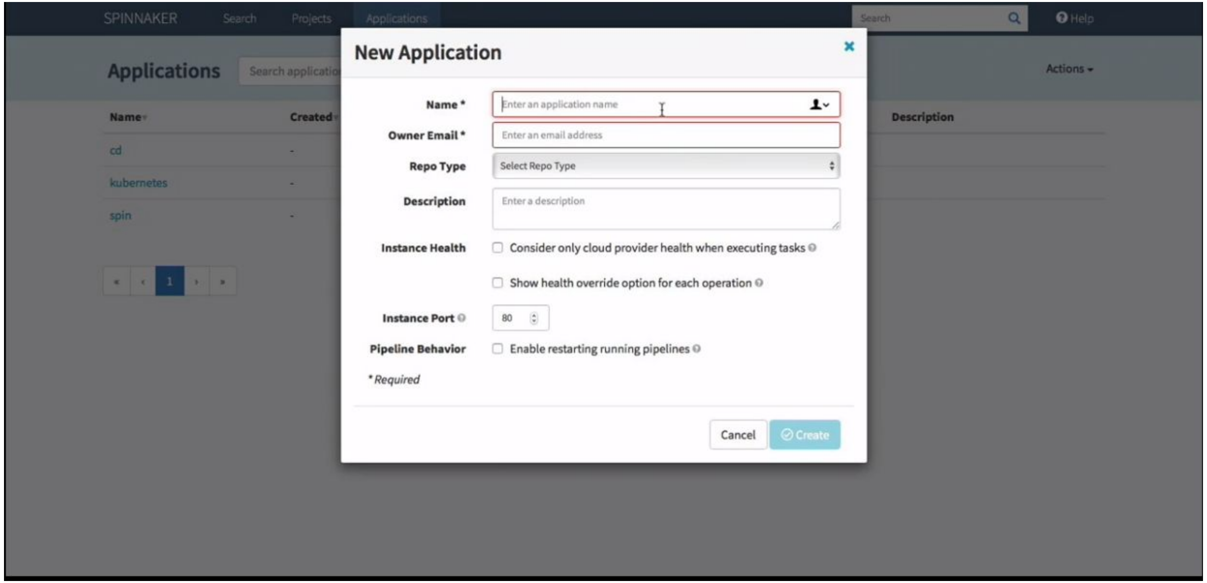


Azure Storage
Google Cloud Storage
Minio
Redis
S3
Oracle Object Storage

Let’s Try

https://s.id/sp-oicndi2020

Create an Application



New Application

Name * Enter an application name

Owner Email * Enter an email address

Repo Type Select Repo Type

Description Enter a description

Instance Health ☐ Consider only cloud provider health when executing tasks

☐ Show health override option for each operation

Instance Port 80

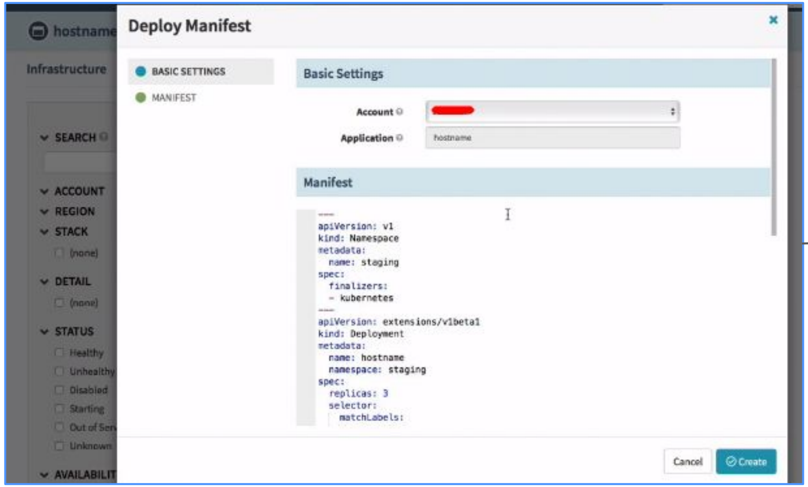
Pipeline Behavior ☐ Enable restarting running pipelines

*Required

Cancel Create

https://github.com/misskecupbung/spinnaker-example

Create a Manifest



Deploy Manifest

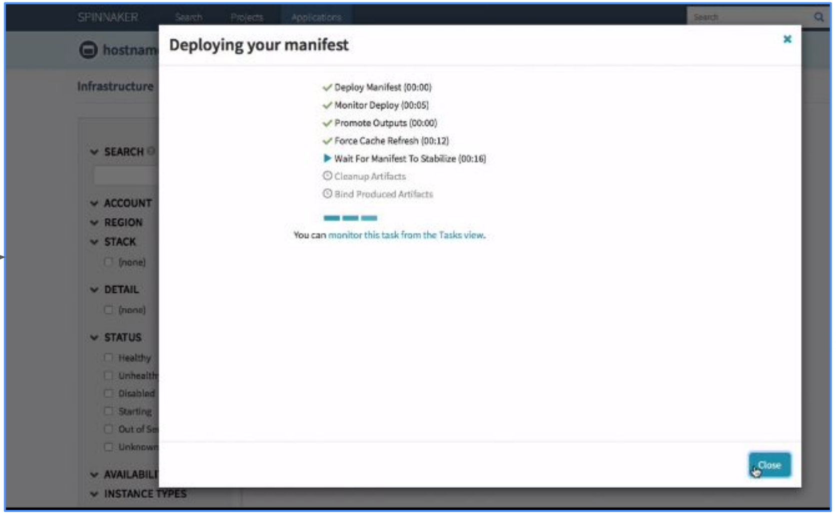
Basic Settings

Account: [redacted]

Application: hostname

Manifest

```
---
apiVersion: v1
kind: Namespace
metadata:
  name: staging
spec:
  finalizers:
  - kubernetes
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: hostname
  namespace: staging
spec:
  replicas: 3
  selector:
    matchLabels:
```



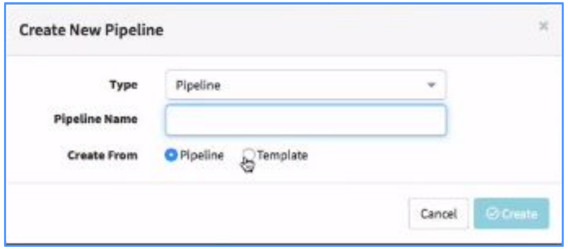
Deploying your manifest

✓ Deploy Manifest (00:00)
✓ Monitor Deploy (00:05)
✓ Promote Outputs (00:00)
✓ Force Cache Refresh (00:12)
▶ Wait For Manifest To Stabilize (00:16)
◷ Cleanup Artifacts
◷ Bind Produced Artifacts

You can monitor this task from the Tasks view.

https://github.com/misskecupbung/spinnaker-example

Create a Pipeline

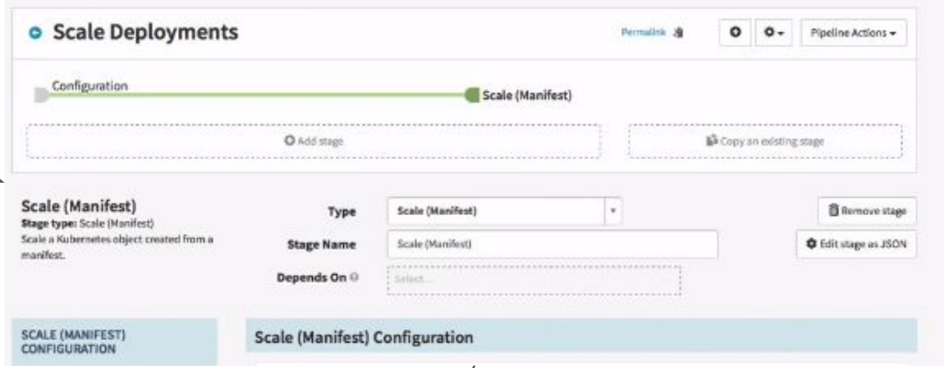


Create New Pipeline ×

Type: Pipeline

Pipeline Name:

Create From: ● Pipeline ○ Template



Scale Deployments

Scale (Manifest)
Stage type: Scale (Manifest)
Scale a Kubernetes object created from a manifest.

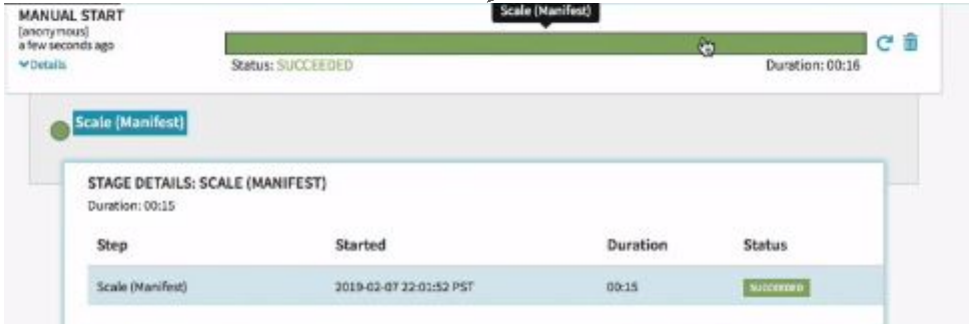
Type: Scale (Manifest)

Stage Name: Scale (Manifest)

Depends On: Select...

SCALE (MANIFEST) CONFIGURATION

Scale (Manifest) Configuration



MANUAL START
[anonymous]
a few seconds ago
Details

Status: SUCCEEDED

Duration: 00:16

Scale (Manifest)

STAGE DETAILS: SCALE (MANIFEST)
Duration: 00:15

Step	Started	Duration	Status
Scale (Manifest)	2019-02-07 22:01:52 PST	00:15	SUCCEEDED

More Explore(s)

- Parallel Action
- Integrating artifacts
- Triggering from webhooks
- Parameterization of manifests
- Rollback and Scale
- Manual Gate
- etc

Resource(s)

- https://spinnaker.io/setup/install/
- https://min.io/download#/linux
- https://kubernetes.io/id/docs/
- https://spinnaker.io/community/releases/versions/1-19-0-changelog
- https://github.com/spinnaker/halyard

Thank you.

Sponsored by:

vmware boer technology ONF

OICNDI

November 7th 2020