

Continuous Delivery using Spinnaker on Kubernetes Multi Cluster

김진웅
SK주식회사 C&C

List of Contents

- Spinnaker?
- Architecture
- Versus
- Concept
- Pipeline
- Demo
- Conclusion
- Q&A

김진웅

@Samsung SDS

'06-'09 System Engineer

'10-'16 Cloud Infra Engineer

@SK C&C

'17 OpenStack Migration Platform

GPU Cloud Platform (w/ K8s)

'18 Serverless Platform Development, IaC Platform (w/ K8s)

Blog : ddiwoong.github.io

Facebook : [ddiwoong](#) (Welcome Friend!!)

Email : ddiwoong@gmail.com

발음?
[spinəkər]



Spinnaker?

– 오픈소스 멀티클라우드 Continuous Delivery/Deployment 플랫폼

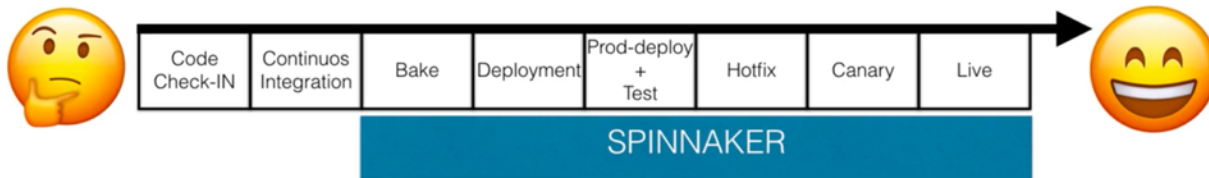


- Netflix's Asgard (2014)로 시작, 2015년에 오픈소스 화
- 빠른 속도와 신뢰로 소프트웨어 변경을 배포하기 위해 빌드
- 디자인 단계에서부터 연결성(pluggability)을 고려
- 대부분의 클라우드 공급자를 지원함
(OpenStack, GCP, Azure, AWS, Kubernetes, Oracle, DC/OS, Cloud Foundry)
- Netflix, Google, MicroSoft, Veritas에서 주로 Contribution

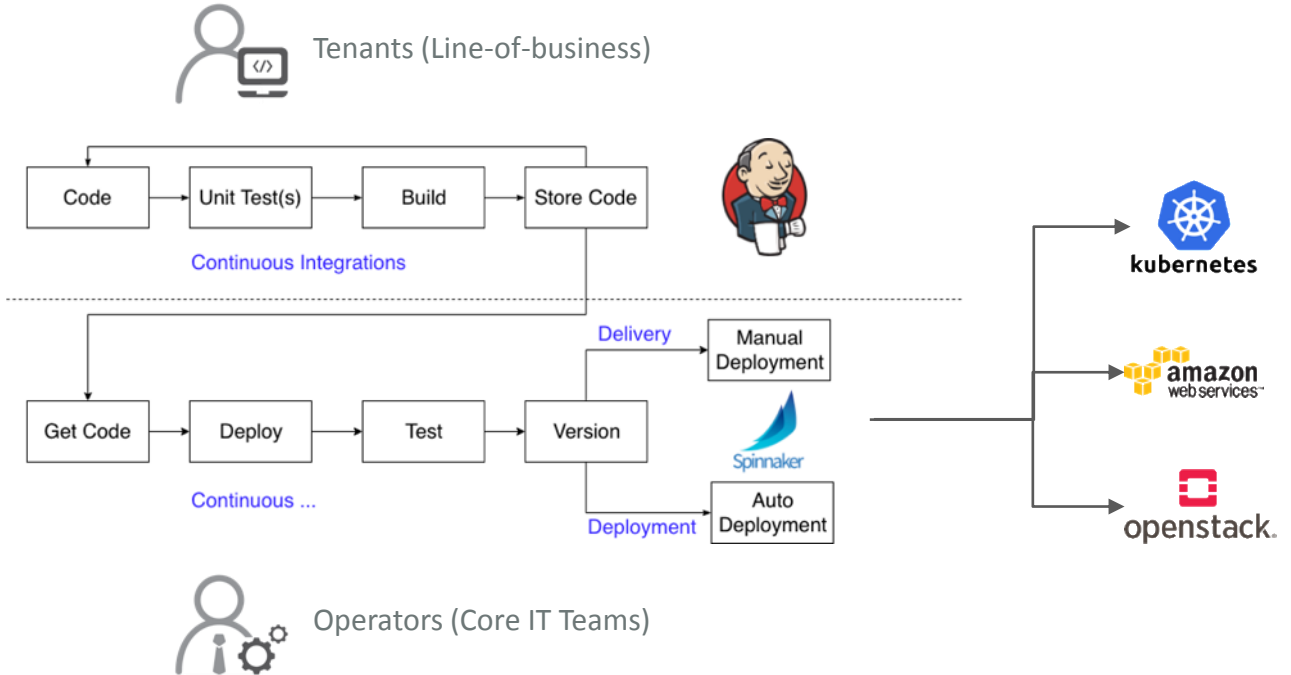
주요 기능

- 클러스터 관리
- 배포 관리
- 멀티 클라우드 가능

기본적으로 클라우드 API연동방식으로 배포 기능이 제공되며, 사용자 스크립트 작성을 최소화



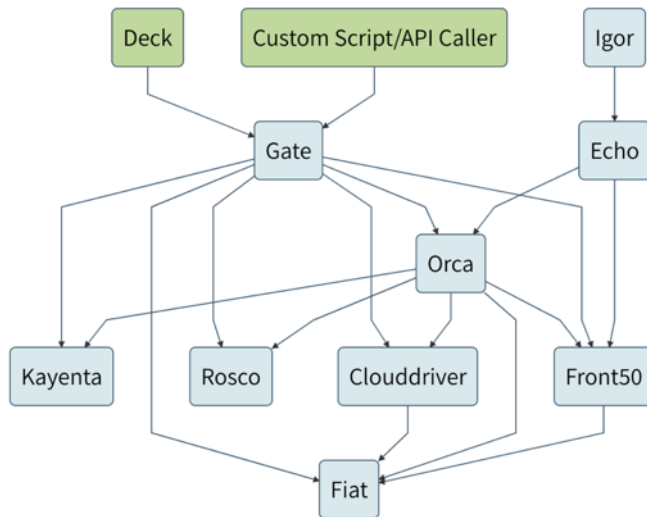
Basic CI/CD



특장점 (TL;DR)

- Multi-Cloud Continuous Delivery/Deployment Platform
- Variable pipeline type, Easy Rollback
- Flexible pipeline management system
- Variable Deployment Strategy (Blue-Green, Rolling Red/Black, Canary)
- Community (github, slack, <https://community.spinnaker.io/>)
- Hybrid Cloud(VM, Container)
- Continuous Integration (Jenkins, Travis CI)
- Halyard CLI (configuring, installing, and updating Spinnaker)
- VM(Packer), Helm Packaging
- RBAC(Role Based Access Control)
- Notification - Email, Slack, Hipchat, SMS
- Safe Deployment - Judgement
- Chaos Monkey Built-in

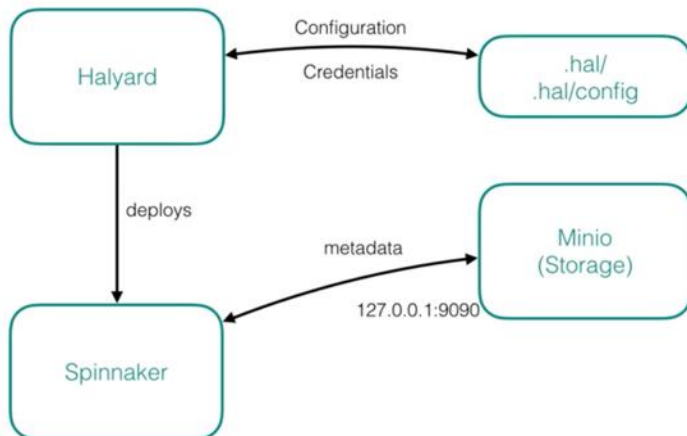
기능별 세분화



- Deck : Browse-based UI
- Gate : API Gateway
- Orca : Orchestration engine
- Clouddriver : Cloud Provider API Connect, Caching, Indexing
- Front50 : applications, pipelines, projects and notifications
- Rosco : Machine Image (w/ Packer)
- Igor : CI Integration (Travis CI, Jenkins)
- Echo : Eventing Bus (Slack, email, SMS)
- Fiat : authorization service
- Kayenta : automated canary analysis
- **Halyard** : Spinnaker Configuration (install, provider, update)
- Other Dependencies
 - S3(or Minio) : To save artifacts
 - Redis : Job/History storage

Halyard

Halyard는 Spinnaker를 설치, 업데이트 및 설정하기 위한 툴




- On Debian/Ubuntu or macOS
- **On Docker**
- **S3 or Compatible S3**
- **Minio**
- Azure Storage
- Google Cloud Storage
- Redis(not recommended)

- **On Kubernetes**
- Local Installation of Debian Packages
- Git Installation (for contributing)

클라우드 공급자(Cloud Provider)

- Google App Engine
- AWS
- Azure
- DC/OS
- Google Compute Engine
- Kubernetes (legacy)
- **Kubernetes v2 (manifest based)**
- Openstack
- Oracle Cloud



```
hal config provider kubernetes account add my-k8s-v2-account \  
  --provider-version v2 \  
  --context $(kubectl config current-context)
```

Kubernetes V1 vs V2

	V1 (Legacy)	V2 (Manifest)
Autonomy	Little	As you want
Deploy	Cluster, Server Groups, Load Balancers	Manifests File
Deprecated	Will Soon	Alpha(1.8) , Beta(1.9~)
Container Registry	GCR, DockerHub, Quay, ECR, JFrog	Any Registry
Manifests (Artifacts)	N/A	Base64 Bitbucket Custom GCS GitHub Gitlab S3

Jenkins vs Spinnaker

Jenkins	Spinnaker
Powerful Build No Deployment tool (Plug-in) Many Scripts Many Plug-ins	Resource Integration Directly Use Cloud API Few Scripts No CI Tools(CI is Backend)

Native Kubernetes vs Spinnaker

Native Kubernetes	Spinnaker
Replica Rollout Slow Rollout Linear Rollout No Approval	Percent Rollout Fast Rollback High Manage cost(infra) Approval(Judgement)

용어

Spinnaker	Kubernetes	비고
Cluster	Deployment	Logical Server Groups
Server-Group	<u>Workloads</u>	Artifact + Configuration <u>CRDs - Custom Build</u>
Load Balancer	Services	LoadBalancer(k8s) 미지원
Firewall	Network Policy	Network Policy 미지원
Applications	Cluster	
Account	Cluster Credentials	

Cluster

- Kubernetes에서의 Cluster가 아님
- Logical Server Groups

Artifacts

<https://www.spinnaker.io/reference/artifacts/>

- Docker 이미지
- GitHub에 저장된 파일
- Amazon Machine Image (AMI)
- a binary blob in S3, Google Cloud Storage, Google Pub/Sub, etc.

Server Group

- 기본 리소스: 배포할 수 있는 artifacts 와 인스턴스(pod) 수, Auto-Scaling, metadata 등 기본 Configuration
- 로드밸런서나 방화벽도 선택적으로 연결
- 배포된 vm이나 pod 형태로 배포된 애플리케이션의 집합체

Provider

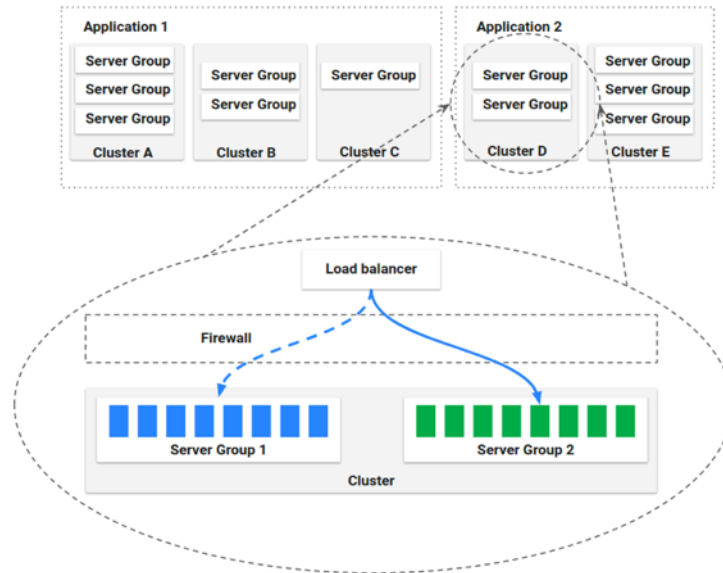
- IaaS - AWS, GCP, Azure, Oracle, Openstack, , Cloud Foundry(alpha)
- PaaS - Google App Engine
- Orchestrator - K8s, DC/OS
- Docker v2 Registry

Account

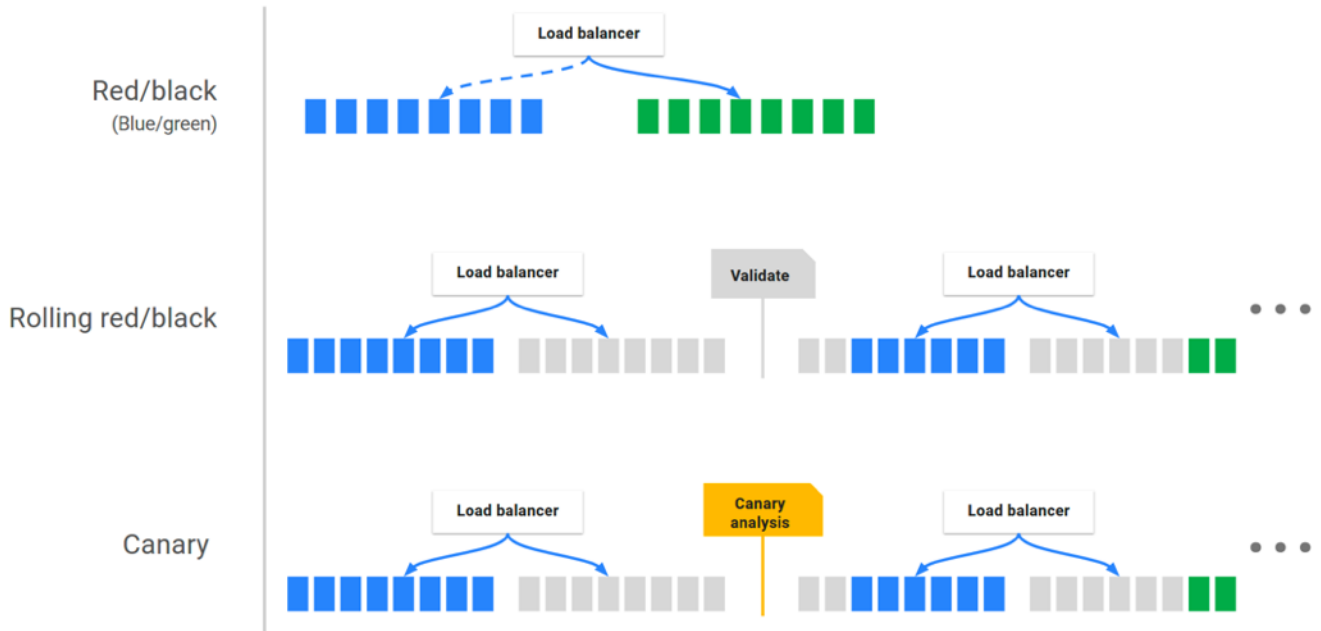
- Provider에 인증하기 위한 Spinnaker에서만 사용하는 Name
- Provider별 Multiple Account 가능
 - Dev/Staging/Prod

애플리케이션 관리

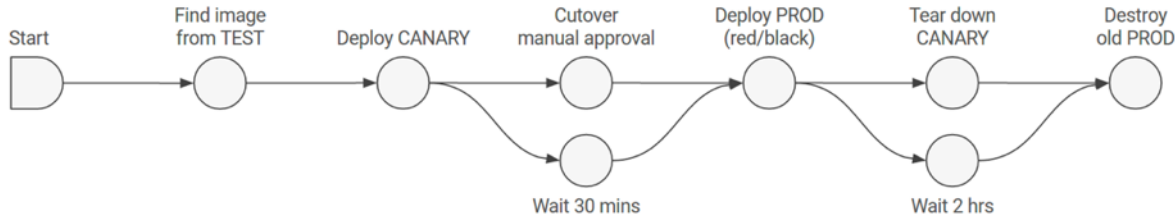
- Pipeline
- A Set of Clusters, Server Group (included firewall, loadbalancer)
- Canary Config



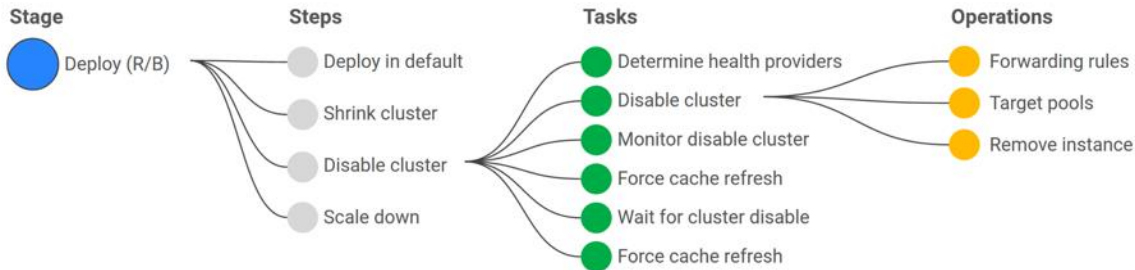
배포 전략



Kubernetes 리소스 배포하기

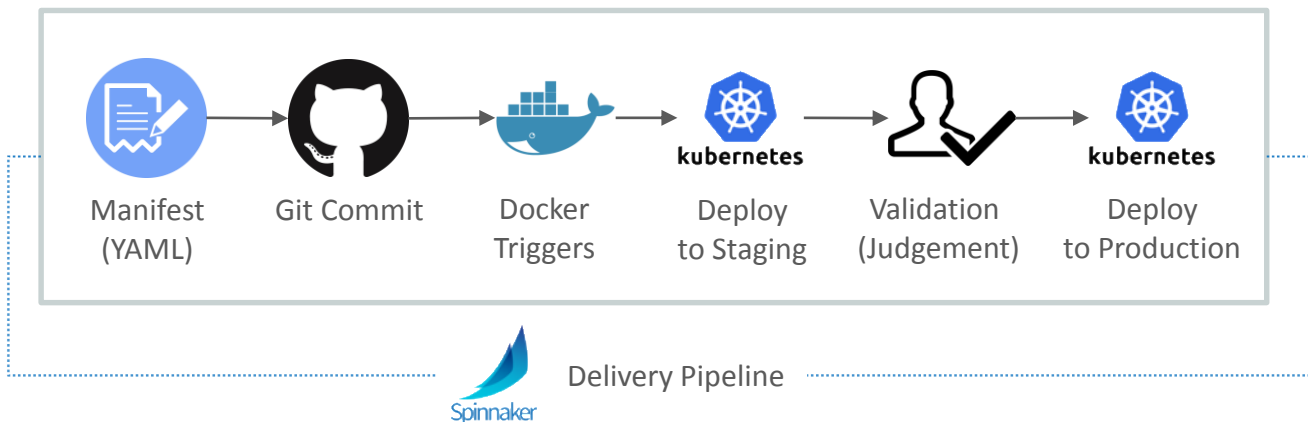


Stage (Pipeline 기본 구성요소)

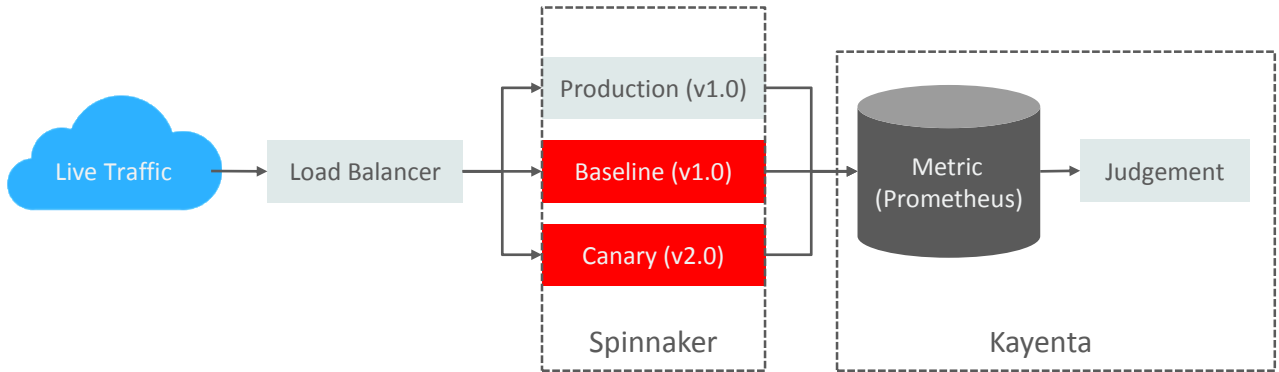


커밋에서 배포까지(Manifest Based)

<https://www.spinnaker.io/guides/tutorials/codelabs/kubernetes-v2-source-to-prod/>



카나리(canary) 배포 응용



Spinnaker 활용하면...

- 배포를 위한 싱글 소스 관리
- 추적성 향상
- 검증된 코드/이미지

추가적인 툴과의 통합

CI/CD	Artifacts	Monitoring	Notifications	Security	Bakery	Triggers	Canary
Jenkins Traivs	Base64 Bitbucket Custom GCS GitHub Gitlab S3	Prometheus Stackdriver Datadog	Slack Email HipChat SMS	Google Groups Github Teams LDAP SAML	Helm Packer	Pub/Sub Github Webhhok Docker Build	Scope Judge Metrics

Spinnaker 대규모 클러스터 적용

- Install, Config, Update with Halyard
- Spinnaker + Jenkins + Packer + Helm
- Redis Tuning
- HPA (Horizontal Pod Autoscaler)
- Monitoring / Logging
 - Datadog, Prometheus, Stackdriver
 - Node logging agent

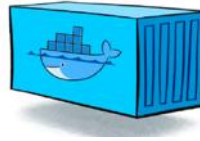
목표

Business-Driven Golden Images

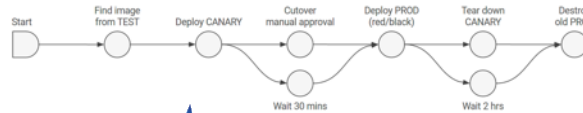
Container Registry



Application Services Team



DevOps Team



Delivery Pipeline



질문 받습니다!

개발자를 모집합니다. 함께 클라우드를
변화시키실 분은 연락주세요!