**KubeCon** | **CloudNativeCon**

North America 2017

# Continuous Integration at Scale on Kubernetes

Karthik Gajjala, Cloud Engineering Director, *eBay*

Xiaogang Xin, Cloud Engineer MTS2, *eBay*

# Agenda

- eBay in a Nutshell
- Builds at eBay
- CI as a Service
    - Our Journey
    - CIaaS Architecture
    - CIConfig and CIController
    - Ingress
    - Hibernation
- Learnings
- Future Roadmap

# eBay in a Nutshell

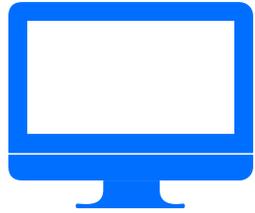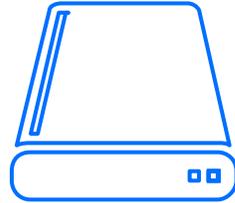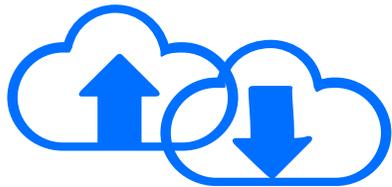| | | | |
|---|---|---|---|
| **168 M** Active Buyers | **$21.7 B** GMV | **$11 B** Mobile | **57%** International |
| **88% GMV** Fixed Price | **81%** Sold are New | **$2.4B** Revenue | **1.1B** Live Listings |

# eBay Infrastructure

**201k**
Managed VMs

**4.5PB**
Managed Storage

**80k**
Managed BMs

**95%**
Traffic on Cloud

**4k**
Applications

**100B**
URLs per Day

# eBay Applications
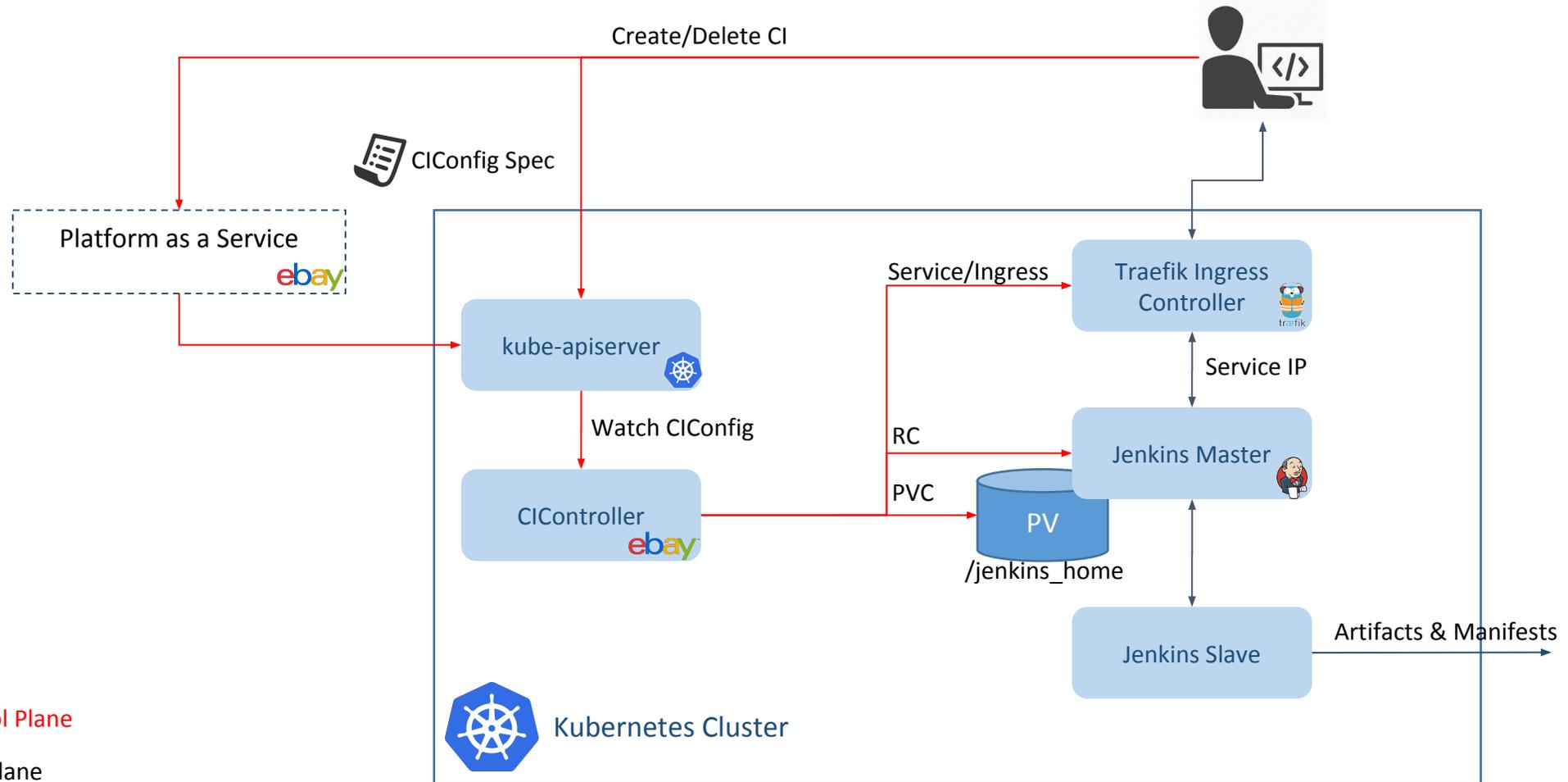


300,000

Builds Per Month

# Our CI Journey

Do your own Jenkins - Self-service VMs and Jenkins configuration

Mesos Cluster - Shared Filer for persistent configurations, Separate Master and Build clusters, PaaS Layer

## Kubernetes

- Separate images for Master and Slave
- Tainted nodes for builder pods (slaves)
- Service and Ingress (Traefik) for Master
- Persistent Volumes (backed by Ceph)
- Highly scalable and incredibly resource efficient

# Architecture



Create/Delete CI

CIConfig Spec

Platform as a Service
ebay

kube-apiserver

Watch CIConfig

CIController
ebay

Service/Ingress

Traefik Ingress Controller
traefik

Service IP

RC

Jenkins Master

PVC

PV
/jenkins_home

Jenkins Slave

Artifacts & Manifests

Kubernetes Cluster

— Control Plane

— Data Plane

# Why CIConfig?

- Model driven is the reason?
  - No, we always think from user's requirement perspective


- What user really want is a CI while not Jenkins
  - A CI is a process from source to manifest
  - Jenkins is only one of the build agents

# CIConfig & Control Loop

## Model driven CI creation & hibernation

```
{
  "kind": "CiConfig",
  "apiVersion": "v1",
  "metadata": { … },
  "spec": {
    "source": { },
    "strategy": {
      "master": { … },
      "env": [ … ],
      "standardBuild": { … }
    },
    "additionalProperties": { }
  }
}
```

```
spec": {
  "source": {
    "git": {
      "uri": "github.corp.ebay.com/ciaas/ciaas.git"
    }
  },
  "strategy": {
    "master": {
      "image": "prod-jenkins-master:stable",
      "volumeSize": 100 GiB
    },
    "env": [ … ],
    "standardBuild": {
      "spec": {
        "stack": {
          "type": "maven",
          "builder": "hub.ebay.io/ciaas/maven_builder:v1.0"
        }
      },
```
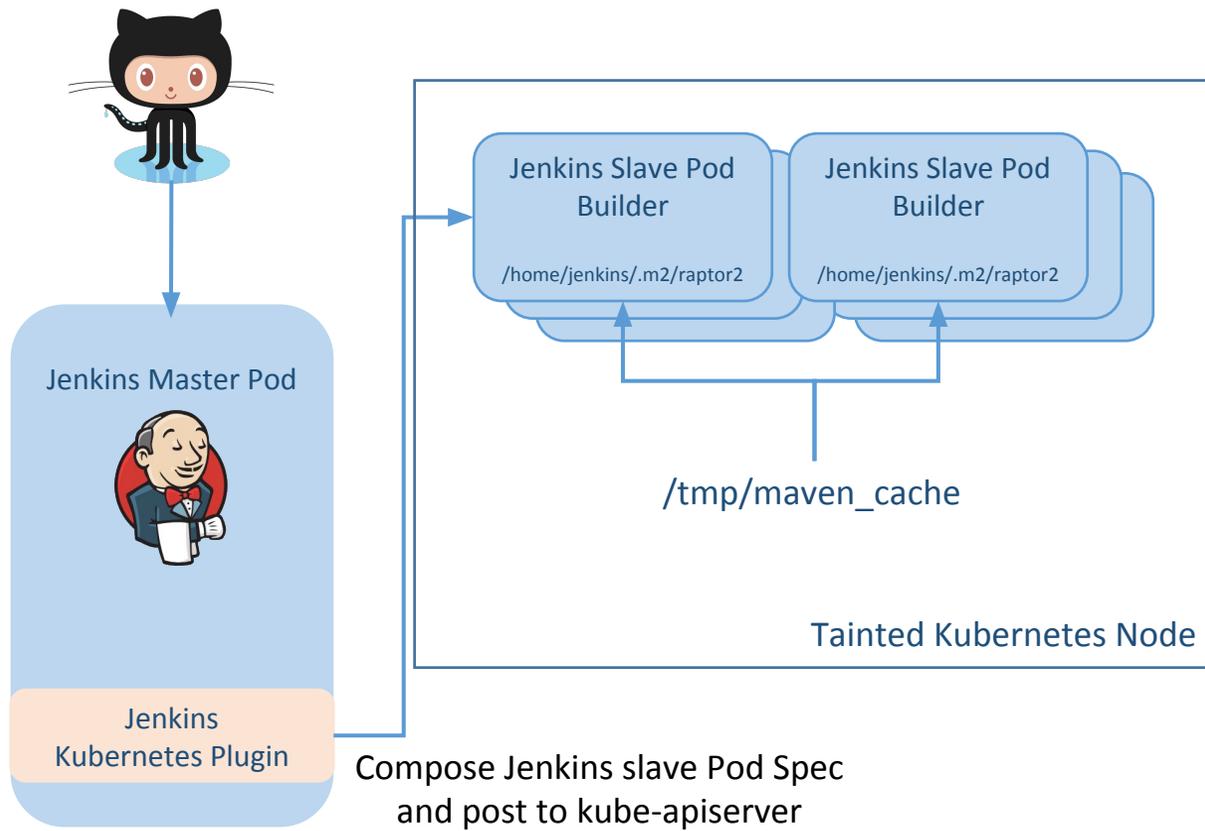
Watch → CIController
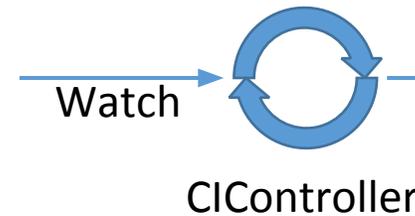
### For Provision CI

- Provision PVC
- Provision Jenkins Master Pod
- Provision Service
- Provision Ingress

### For Hibernate/Re-active CI

- Del/Re-create Jenkins Master Pod
- Del/Re-create Service
- Setup Ingress

# Life of a Build



Jenkins Master Pod

Jenkins Kubernetes Plugin

Compose Jenkins slave Pod Spec
and post to kube-apiserver

Jenkins Slave Pod Builder

/home/jenkins/.m2/raptor2

Jenkins Slave Pod Builder

/home/jenkins/.m2/raptor2

/tmp/maven_cache

Tainted Kubernetes Node

- User has the flexibility to use their own builder
  - Maven and NodeJS has been hardened as standard build
  - Python and Scala is built by users
- Kubernetes plugin compose slave Pod spec
  - Docker image
  - Taint toleration
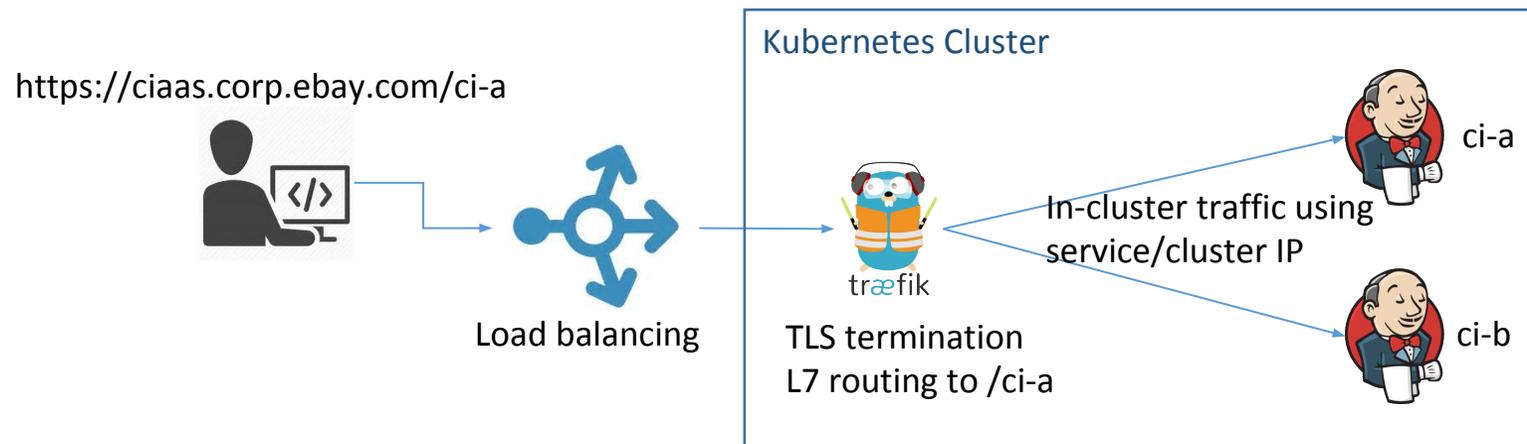  - Volumes
    - Secrets
    - HostPath

# Ingress

## L7 Ingress helps us to solve the following problems

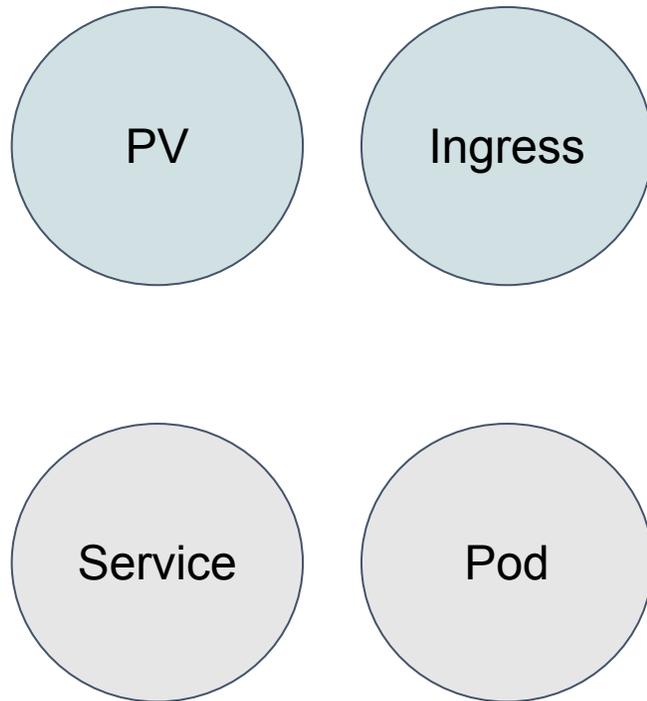| | | |
|---|---|---|
| Every CI instance needs a durable access point. Use load balancer would be a huge waste of VIP. | → | By enabling Ingress L7 routing, we only need one Ingress VIP. |
| Wildcard SSL certificate is not secure enough. It's hard to issue specific SSL certificate for every CI instance. | → | By enabling Ingress, we only need to issue the Ingress certificate. |

https://ciaas.corp.ebay.com/ci-a

Kubernetes Cluster

traefik

ci-a

In-cluster traffic using service/cluster IP

ci-b

Load balancing

TLS termination
L7 routing to /ci-a

# Hibernation

PV

Ingress

Service

Pod

Leave PV and record it in CIConfig status for resurrection

Configure Ingress to route CI request to a re-activate service

- Re-activate service will -
  - Re-create Pod and mount the PV in CIConfig status
  - Re-create Service
  - Re-configure Ingress to route to new Service

# Learnings

- CIConfig as an Abstraction

- Tainted pool of compute nodes for builder pods - hostpath mount

- Pod Templates make it self-service for developers - Polyglot

- Monitoring CIs help catch laggards - Traefik missing events, Pods are not really running

- PVC volume detach and attach is a heavy process

# Future Work

- Span across multiple Kubernetes clusters with Federation

- Object store as persistent store

- Declarative pipelines

- Leverage public clouds