KubeCon | CloudNativeCon China 2018

ebay

Model and Operate Datacenter by Kubernetes at eBay

辛肖刚, Cloud Engineering Manager, ebay
梅岑恺, Senior Operation Manager, ebay

Agenda

- About ebay
- Our fleet
- Kubernetes makes magic at ebay
- Model + Controller
- How we model our datacenter
- Operation in large scale
- Q&A

ebay

177M

Active buyers worldwide

$2.6B

Reported revenue

1.1B

Live listings

81%

Sold are new

62%

International revenue

$22.7B

Amount of eBay Inc. GMV

88%

Fixed price

$11B

Mobile

The Fashion category has accounted for **26%** of total Retail Revival GMV.

KubeCon | CloudNativeCon

China 2018

ebay

Our fleet

3 US Data Centers	15 POPs	200K+ Managed Vms
100K Managed BMs	4K Applications	4.5PB Managed Storage

ebay

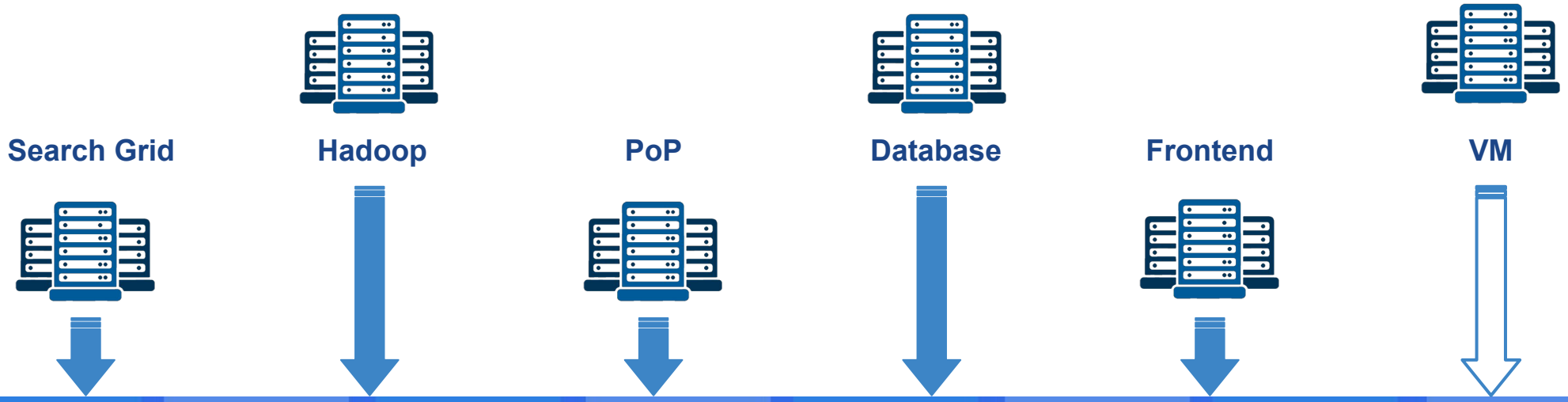
All of us know that...

It’s not easy to manage fleet and infrastructure at scale

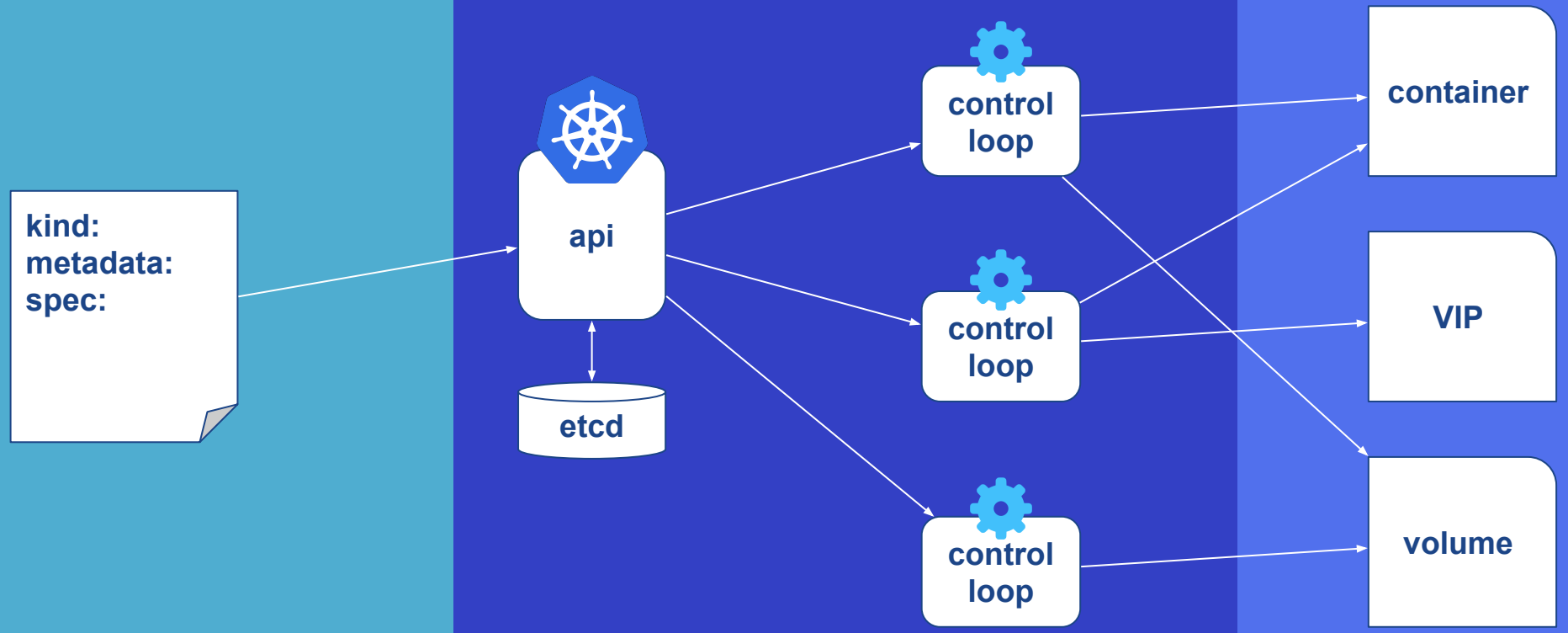

Infrastructure should play magic

Way to Kubernetes

Kubernetes plays magic



Kubernetes Core concept of Kubernetes - Declarative magic



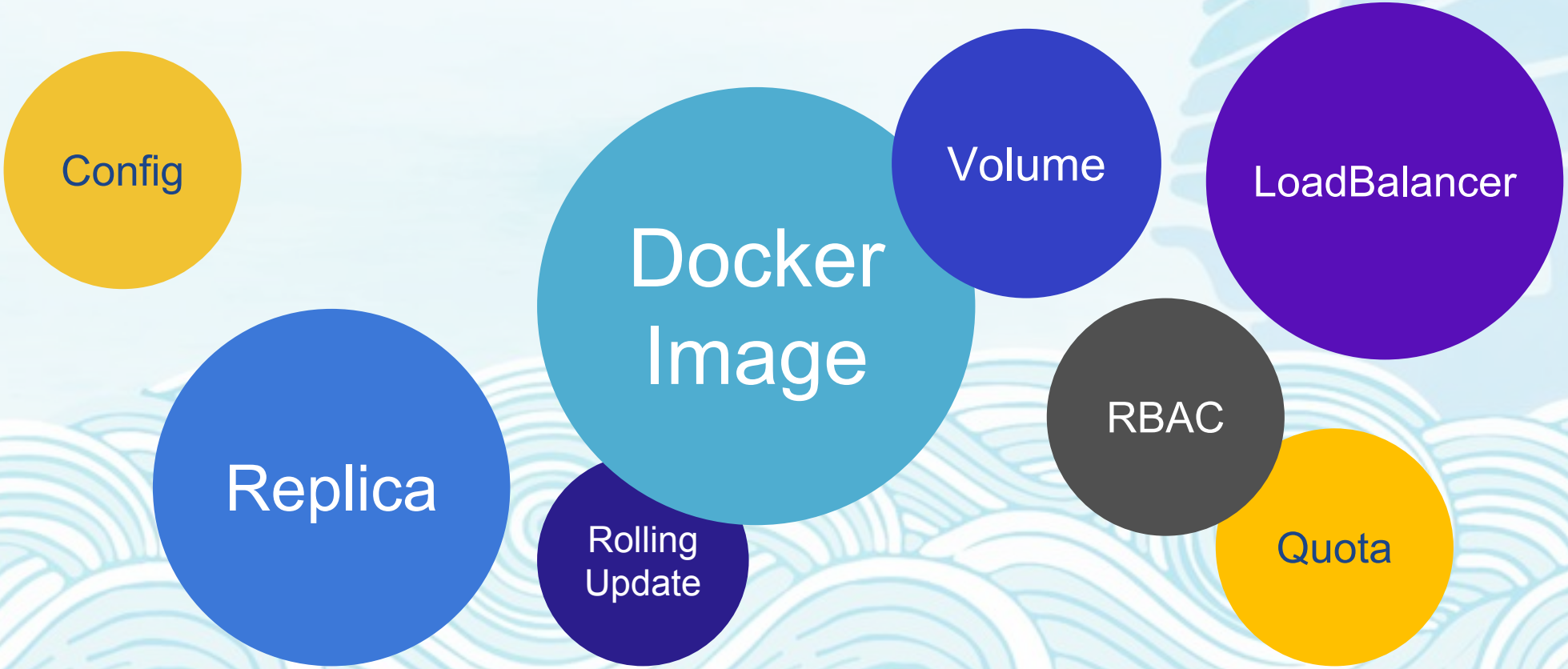
WISB: What it should be

Converge & Reconcile

WIRI: What it really is

Kubernetes models applications

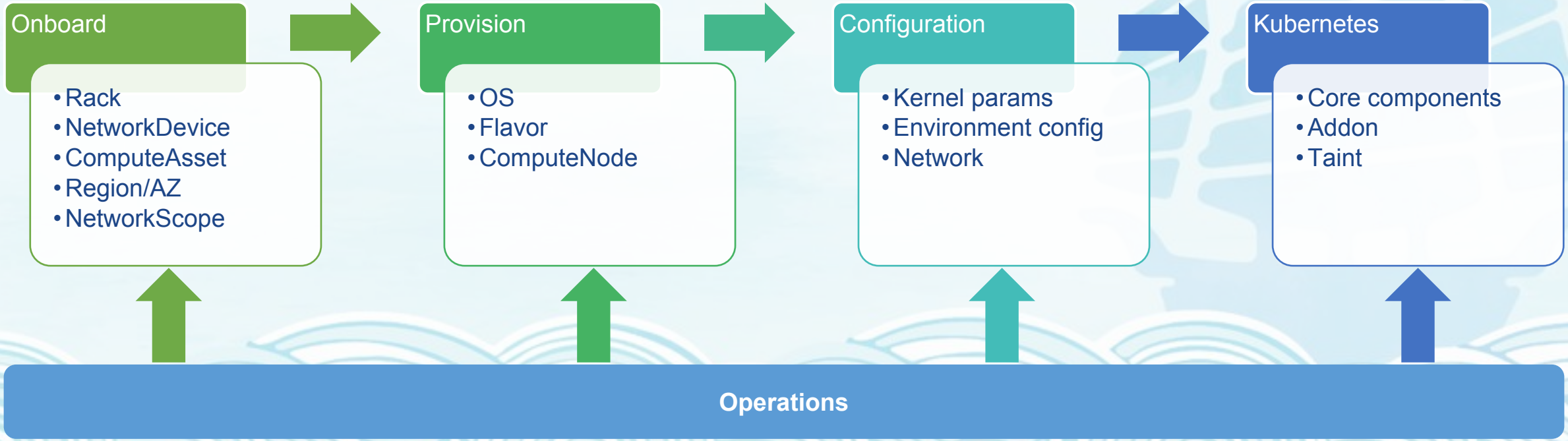
What is an application looks like?



How about Kubernetes itself?

How about the fleet Kubernetes running on?

Our thinking of datacenter modeling by extending Kubernetes



Let's model a datacenter running Kubernetes

Onboard → Provision → Configuration → Kubernetes

You need onboard something from nothing!

```yaml
apiVersion: infra.tess.io/v1alpha1
kind: GeographicalRegion
metadata:
  name: us-south
spec:
```

```yaml
apiVersion: infra.tess.io/v1alpha1
kind: AvailabilityZone
metadata:
  name: slc01
region:
  name: us-central
```

```yaml
apiVersion: network.tess.io/v1alpha1
kind: L2Domain
metadata:
  name: rnb10-ra082
spec:
  availabilityZone: slc01
  networkZone:
    name: production
status:
  phase: ready
```

```yaml
apiVersion: network.tess.io/v1alpha1
kind: SubNetwork
metadata:
  name: 10-242-158-0--24
spec:
  availabilityZone: slc01
  ipCidrRange: 10.242.158.0/24
  l2Domain:
    name: rnb10-ra065
  network:
    name: ebay
  networkZone:
    name: production
```

```yaml
apiVersion: infra.tess.io/v1alpha1
kind: Rack
metadata:
  name: rno--mcc10--01-1110--33--06
position:
  room: RNO:MCC10:01-1110:33
  row: RNO:MCC10:01-1110
```

```yaml
apiVersion: infra.tess.io/v1alpha1
kind: ComputeAsset
metadata:
  name: asset00538893
manufacturer: hyve
rack:
  name: RNO:MCC10:01-1110:33:06
sku:
  name: BD3G6
devices:
- Labels: null
  function: network
  index: 1
  l2Domain:
    name: rnb04-ra020
  macAddress: 4C:38:D5:04:AC:85
  networkSwitch:
    name: rnb04-ra020
  speed: 10000Mb/s
  type: Provision
- Labels: null
  function: network
  ipAddr: 10.24.113.131
  l2Domain:
    name: rnb10-ra082
  macAddress: 4C:38:D5:04:AC:84
  networkSwitch:
    name: rnb10-ra082
  type: Management
```

Let’s model a datacenter running Kubernetes

After you define your fleet, you want a accessible compute node: Asset + Flavor + OS = ComputeNode

```yaml
apiVersion: compute.tess.io/v1alpha1
kind: OSImage
metadata:
  name: centos-atomic
spec:
  kernel_version: "3.10"
  os_version: "7.5"
  ostree_repo_url: http://ostree.ebay.com/atomic-ostree/centos/7.5.1804/docker-18.03.1-ce/ostree-3-10-3/
  image_id: 42feb598-48ab-45c5-a79b-2c9bd6a53232
```

Let's model a datacenter running Kubernetes

Onboard → Provision → Configuration → Kubernetes

After you define your fleet, you want a accessible compute node: Asset + Flavor + OS = ComputeNode

```yaml
apiVersion: infra.tess.io/v1alpha1
kind: ComputeFlavor
metadata:
  name: p3g6-minion-nudata
spec:
  cpu:
  - spec:
      cores: 48
      frequency: "0"
  memory:
  - DRAMType: ""
    memoryModuleFormat: ""
    spec:
      frequency: "0"
      size: 384G
```

```yaml
storage:
- disks:
  - disk:
      blockDevice:
        device:
          Labels: null
          function: disk
          name: sda
        partitions:
        - blockDevice:
            device:
              Labels: null
            size: "0"
          bootPartition: true
          fileSystem:
            fstype: biosboot
          name: biosboot
          size: 1M
          type: gpt
        - blockDevice:
            device:
              Labels: null
            size: "0"
          bootPartition: true
          fileSystem:
            fstype: ext4
          name: /boot
          primary: true
          size: 300M
          type: gpt
        - blockDevice:
            device:
              Labels: null
            size: 85528M
          name: pv.01
          primary: true
          type: gpt
      ......
```

```yaml
lvms:
- lvmDevice:
    blockDevice:
      device:
        Labels: null
      size: "0"
    lvmLogicalVolume: {}
  spec:
    lvs:
    - volumeGroupName: vg00
      volumeSpec:
        namePattern: root
        size: 151552M
    pvs:
    - volumeGroupName: vg00
      volumeSpec:
      - name: pv.01
      - name: pv.02
      - name: pv.03
```

Let’s model a datacenter running Kubernetes

After you define your fleet, you want a accessible compute node: Asset + Flavor + OS = ComputeNode

```yaml
apiVersion: compute.tess.io/v1alpha1
kind: ComputeNode
metadata:
  name: tess-node-zzq4c
spec:
  assetName: asset00538893
  flavor: bd3g6-minion-hadoop
  livenessProbes:
  - failureThreshold: 3
    initialDelaySeconds: 600
    periodSeconds: 3
    tcpSocket:
      port: "22"
    timeoutSeconds: 3
  osImage:
    name: centos-atomic-hadoop
  provider: foreman
```

Let's model a datacenter running Kubernetes

Onboard → Provision → Configuration → Kubernetes

You have your compute node now, all you need is to configure it by a configuration management orchestration. We use **SaltStack**.

```yaml
apiVersion: salt.tess.io/v1alpha1
kind: SaltMaster
metadata:
  name: salt-master-test
  namespace: default
spec:
  computeNodeRef: tess-node-e2e-00489302
  pillars:
  - name: group
    secretRef:
      name: pillarsecret
  saltEnvironments:
  - environment: other0.28
    gitRepo:
      directory: salt/other/other0.28
      path: /srv/salt/other0.28
      repository: https://git.ebay.com/tess/tessops.git
    name: saltStateother0.28
    pillars:
    - environment: other0.28
      gitRepo:
        directory: pillar/other/other0.28
        path: /srv/pillar/other0.28
        repository: https://git.ebay.com/tess/tessops.git
      name: pillarv28
  - environment: hrtv0.28
    gitRepo:
      directory: salt/hrt/hrtv0.28
      path: /srv/salt/hrtv0.28
      repository: https://git.ebay.com/tess/tessops.git
    name: saltStatehrtv0.28
    pillars:
    - environment: hrtv0.28
      gitRepo:
        directory: pillar/hrt/hrtv0.28
        path: /srv/pillar/hrtv0.28
        repository: https://git.ebay.com/tess/tessops.git
      name: pillarhrtv0.28
```

```yaml
apiVersion: salt.tess.io/v1alpha1
kind: SaltMinion
metadata:
  name: salt-minion-1
  namespace: default
spec:
  computeNodeRef: tess-node-minion01
  grains:
  - configMapName: minion1-grains
    name: group
  - name: defaultGrain
    secretRef:
      name: secretgrain
  saltMaster: salt-master-test
```

```yaml
apiVersion: salt.tess.io/v1alpha1
kind: SaltDeployment
metadata:
  namespace: 99
  name: sd-kubernetes-master-99
spec:
  minionsConfig:
    selector:
      matchLabels:
        k8s.tess.io/cluster: "99"
        k8s.tess.io/role: master
    strategy:
      name: rack-by-rack
    template:
      spec:
        grains:
        - secretRef:
            name: mastergrain
        - secretRef:
            name: salt-minion-grain
        roles:
        - kubernetes-master
        saltEnvironments:
        - gitRepo:
            directory: kubernetes/salt/99
            repository: https://git.ebay.com/tess/tessops.git
            revision: ab136e189b1081ebf2769a949bf6b1cd80a38c6a
        targetPillar:
        - pillarv2
        ...
  applicationInstances:
  - etcd
  - apiserver
  - contoller-manager
  - scheduler
status:
  minionsStatus:
    desiredBuckets: 4
    lastProbeTime: 2018-08-31T22:45:10Z
    lastTransitionTime: 2018-08-31T18:44:28Z
    transactionName: sd-kubernetes-master-5l9hg
    updatedBuckets: 2
  pause: true
```

Let’s model a datacenter running Kubernetes

Onboard → Provision → Configuration → Kubernetes

It’s time to spin up a Kubernetes cluster!

```yaml
apiVersion: k8s.tess.io/v1alpha1
kind: K8sCluster
metadata:
  labels:
    az: phx02
    realm: production
    region: us-central-1
  name: "21"
spec:
  availabilityZone:
    name: phx02
  description: Tess production cluster in phx02
  networkZone:
    name: production
  provider: c3
status:
  loadBalancer:
    dns: api.system.svc.21.tess.io
    ip: 10.137.209.14
```

Easy operation

How to flex up some nodes?

KubeCon | CloudNativeCon China 2018

Step 1. Find some assets not used

```
xnxin@LM-SHC-16501473:~$ tctl get computeassets
NAME            AGE
asset00489020   276d
asset00489021   276d
asset00489022   276d
asset00489023   276d
asset00489024   276d
asset00489025   276d
asset00489026   276d
asset00489027   276d
asset00489028   276d
asset00489029   276d
asset00489030   276d
asset00489031   276d
asset00489032   276d
asset00489033   276d
```

Step 2. Create a ComputeNode

```
xnxin@LM-SHC-16501473:~$ cat computenode.yaml
kind: ComputeNode
apiVersion: tess.io/v1
metadata:
  name: kubernetes-master-6
  labels:
    role: master
spec:
  selector:
    role: master
    tess.io/etcd-storage-provider: host
  assetName: asset00489191
  flavor: p2bg5-master-std
  osImage: centos-atomic
  provider: foreman
  livenessProbes:
  - Exec:
    HTTPGet:
    TCPSocket:
      Port: '22'
      Host: ''
    InitialDelaySeconds: 600
    TimeoutSeconds: 3
    PeriodSeconds: 0
    SuccessThreshold: 0
    FailureThreshold: 3
xnxin@LM-SHC-16501473:~$ kubectl create -f computenode.yaml
```

Step 3. Relax and have a cup of coffee

ebay

What if salt master down?

Step 1. Find some compute nodes

```
xnxin@LM-SHC-16501473:~$ tctl get computenodes
NAME                          AGE
tess-node-00t2f               260d
tess-node-0996h               260d
tess-node-09dkn               271d
tess-node-09q7j               270d
tess-node-0bdgn               260d
tess-node-0n080               255d
tess-node-0rl46               270d
tess-node-0w8xh               260d
tess-node-0wmt3               260d
tess-node-0x3t1               270d
tess-node-0zr5z               208d
tess-node-11jjm               260d
tess-node-1bk95               271d
tess-node-1k37v               259d
```

Step 2. Create a SaltMaster

```
xnxin@LM-SHC-16501473:~$ cat saltmaster.yaml
apiVersion: salt.tess.io/v1alpha1
kind: SaltMaster
metadata:
  name: salt-master-test
spec:
  computeNodeRef: tess-node-09dkn
  pillars:
  - name: group
    secretRef:
      name: pillarsecret
  saltEnvironments:
  - environment: other0.28
    gitRepo:
      directory: salt/other/other0.28
      path: /srv/salt/other0.28
      repository: https://git.ebay.com/tess/tessops.git
    name: saltStateother0.28
    pillars:
    - environment: other0.28
      gitRepo:
        directory: pillar/other/other0.28
        path: /srv/pillar/other0.28
        repository: https://git.ebay.com/tess/tessops.git
      name: pillarv28
xnxin@LM-SHC-16501473:~$ kubectl create -f saltmaster.yaml
```

Step 3. Relax and have a cup of coffee

How to upgrade a cluster?

KubeCon | CloudNativeCon China 2018

Upgrade Kubernetes core components

```yaml
apiVersion: k8s.tess.io/v1alpha1
kind: K8sDeployment
metadata:
  namespace: default
  name: k8s-99
  labels:
    k8s.tess.io/cluster: 99
spec:
  version: release-0.33.0
  repository: https://git.ebay.com/tess/tessops.git
  saltDeployments:
  - name: sd-kubernetes-master-99
    type: kubernetes-master
    saltEnironments:
    - name: kubernetes
    - name: hrt
      revision: ab136e189b1081ebf2769a949bf6b1cd80a38c6a
    grains:
    - name: mastergrain
    roles:
    - kubernetes-master
  - name: sd-minion-nudata-99
    deploymentStrategy: rack-by-rack
    type: kubernetes-node
    saltEnironments:
    - name: kubernetes
    - name: hrt
      revision: ab136e189b1081ebf2769a949bf6b1cd80a38c6a
    grains:
    - name: miniongrain
    - name: salt-minion-grain-nudata
    roles:
    - kubernetes-pool
  - name: sd-minion-gpu-99
    ......
status:
  .....
```

Upgrade addons

```yaml
apiVersion: k8s.tess.io/v1alpha1
kind: K8sDeployment
metadata:
  namespace: default
  name: k8s-addon-release-0.33.0
  labels:
    k8s.tess.io/cluster: 99
spec:
  version: release-0.33.0
  repository: https://git.ebay.com/tess/tessops.git
  saltDeployments:
  - name: sd-salt-master-99
    saltEnironments:
    - name: addon
      secretPillars:
      - name: kube2udns-99
      - name: models-99
    applicationInstances:
    - etcd
    - apiserver
    - kubelet
    - kube-proxy
    - models
    - kube2udns
status:
  version: release-0.33.0
  saltDeployments:
  - name: sd-salt-master-99
    updated: true
    finished: true
```

ebay

Recap

- Kubernetes is amazing on its simple architecture
- Model + Controller is the key concept of Kubernetes
- It's easy to extend Kubernetes API and write your controller based on list/watch
- ebay uses Kubernetes to model and operate it's datacenter

KafkaCluster, HadoopCluster, MongoDB, ESCluster ……	Application Service
K8sCluster, K8sAddons, K8sDeployment	Infrastructure Service
SaltMaster, SaltMinion, SaltDeployment	Configuration Management
Region, AvailabilityZone, NetworkZone, L2Domain Rack, NetworkDevice, ComputeAsset	Fleet (Compute, Network, Storage)

KubeCon | CloudNativeCon China 2018

We are hiring!

xnxin@ebay.com
cmei@ebay.com

KubeCon | CloudNativeCon

China 2018

ebay

Q&A