



2016中国开源年会

China Open Source Conference 2016

时间：2016年10月15日-16日

地点：北京航空航天大学

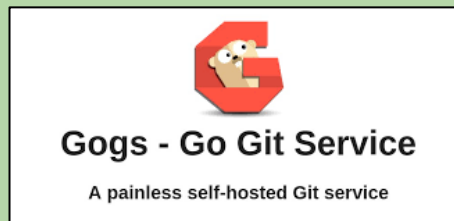


hand-on: Docker-image-based Java App CI

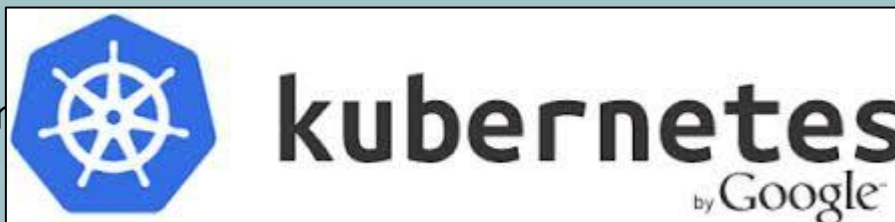
Tang feixiong <fxtang@qingyuanos.com>

PaaS:

- Tenant
- CI/CD
- DevOps
- ...



Container Orchestration



Container Provisioning



Openshift Application template:

Build:

- BuildConfig:

Source: git-hosted, local-workspace-archive

Docker-build-context: git-hosted, local-archive

Hook: git-hooked

Output: Docker image, CD-ready

ImageStream

- Build:

Continuously

Deploy:

- DeploymentConfig

Kubernetes deployment

<https://blog.openshift.com/part-2-creating-a-template-a-technical-walkthrough/>

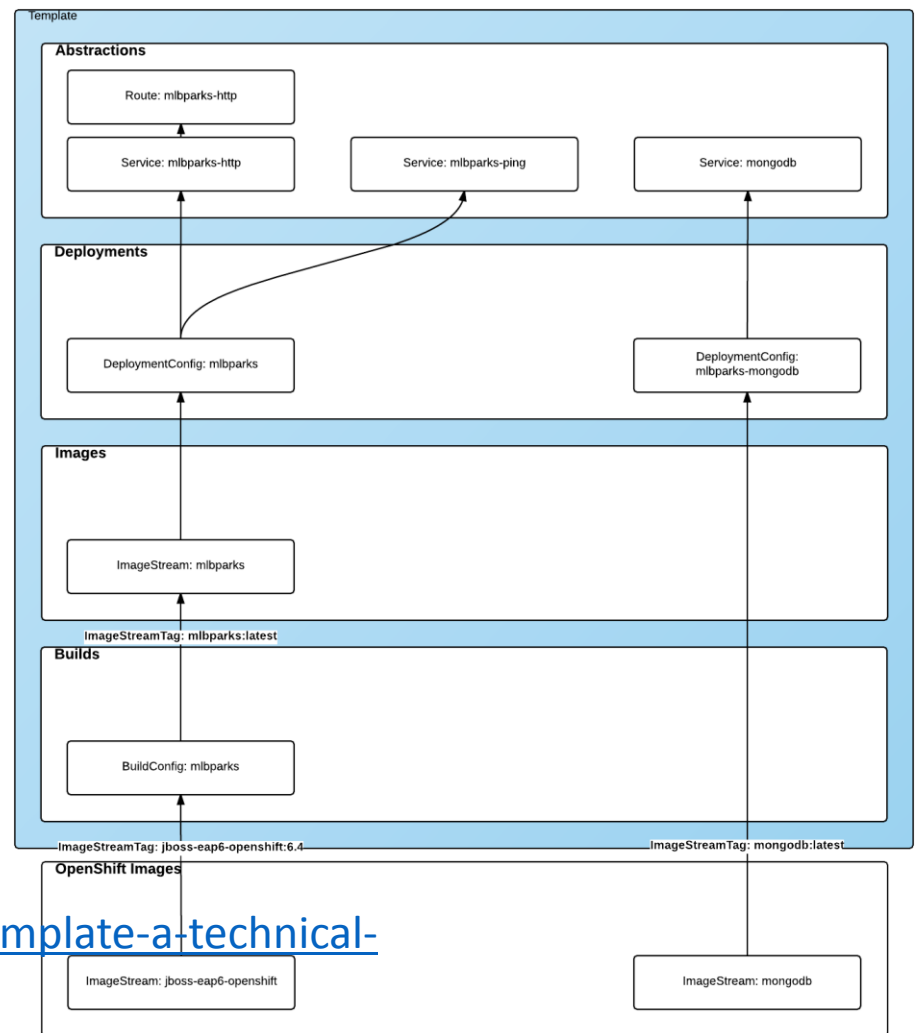
Deploy

Continuously

Staging/Production

- Service: Kubernetes service

- Route: Kubernetes ingress



Layer 1: Builds

```
{
  "kind": "BuildConfig",
  "apiVersion": "v1",
  "metadata": {
    "name": "mlbparks",
    "labels": {
      "application": "mlbparks"
    }
  },
  "spec": {
    "source": {
      "type": "Git",
      "git": {
        "uri": "https://github.com/jorgemoralespou/openshift3mlbparks.git",
        "ref": "master"
      },
      "contextDir": ""
    },
    "strategy": {
      "type": "Source",
      "sourceStrategy": {
        "from": {
          "kind": "ImageStreamTag",
          "namespace": "openshift",
          "name": "jboss-eap6-openshift:6.4"
        }
      }
    }
  },
}
```

```
  "output": {
    "to": {
      "kind": "ImageStreamTag",
      "name": "mlbparks:latest"
    }
  },
  "triggers": [
    {
      "type": "GitHub",
      "generic": {
        "secret": "secret"
      }
    },
    {
      "type": "Generic",
      "github": {
        "secret": "secret"
      }
    },
    {
      "type": "ImageChange",
      "imageChange": {}
    }
  ]
}
```

Layer 2: Images

```
{
  "kind": "ImageStream",
  "apiVersion": "v1",
  "metadata": {
    "name": "mlbparks",           (1)
    "labels": {
      "application": "mlbparks"   (2)
    }
  },
  "spec": {                       (3)
    "dockerImageRepository": "",  (4)
    "tags": [                     (5)
      {
        "name": "latest"
      }
    ]
  }
}
```

Layer 3: Deployments

```
{
  "kind": "DeploymentConfig",
  "apiVersion": "v1",
  "metadata": {
    "name": "mlbparks",           (1)
    "labels": {                   (2)
      "deploymentConfig": "mlbparks",
      "application": "mlbparks"
    }
  },
  "spec": {                       (3)
    "replicas": 1,                (4)
    "selector": {
      "deploymentConfig": "mlbparks" (5)
    },
    "strategy": {
      "type": "Recreate"           (6)
    },
    "template": {                 (7)
      "metadata": {
        "labels": {               (8)
          "deploymentConfig": "mlbparks",
          "application": "mlbparks"
        },
        "name": "mlbparks"        (9)
      },
      "spec": {                   (10)
        "containers": [
          {
```



Layer 3: Deployments - 1

```
"name": "mlbparks",           (11)
"image": "mlbparks",          (12)
"imagePullPolicy": "Always",  (13)
"env": [                      (14)
  {
    "name": "OPENSIFT_DNS_PING_SERVICE_NAME",
    "value": "mlbparks-ping"
  },
  {
    "name": "OPENSIFT_DNS_PING_SERVICE_PORT",
    "value": "8888"
  },
  {
    "name": "MONGODB_USER",
    "value": "user"
  },
  {
    "name": "MONGODB_PASSWORD",
    "value": "password"
  },
  {
    "name": "MONGODB_DATABASE",
    "value": "database"
  }
],
"ports": [                    (15)
  {
    "name": "mlbparks-http",
    "containerPort": 8080,
    "protocol": "TCP"
```

```
},
{
  "name": "mlbparks-ping",
  "containerPort": 8888,
  "protocol": "TCP"
},
],
"readinessProbe": {          (16)
  "exec": {
    "command": [
      "/bin/bash",
      "-c",
      "/opt/eap/bin/readinessProbe.sh"
    ]
  }
},
"resources": {},
"terminationMessagePath":
"/dev/termination-log",
"securityContext": {         (17)
  "capabilities": {},
  "privileged": false
},
],
"restartPolicy": "Always",
"dnsPolicy": "ClusterFirst"
},
"triggers": [                (18)
{
```



Layer 3: Deployments - 2

```

    "type": "ImageChange",
    "imageChangeParams": {
      "automatic": true,
      "containerNames": [
        "mlbparks"
      ],
      "from": {
        "kind": "ImageStreamTag",
        "name": "mlbparks:latest"
      }
    },
  },
  {
    "type": "ConfigChange"
  }
]
}
```

Layer 4: Abstractions - Service

```

{
  "kind": "Service",
  "apiVersion": "v1",
  "metadata": {
    "name": "mlbparks-ping",
    "labels": {
      "application": "mlbparks"
    },
    "annotations": {
      "description": "Ping service for clustered applications"
    }
  },
  "spec": {
    "ports": [
      {
        "port": 8888,
        "targetPort": 8888
      }
    ],
    "selector": {
      "deploymentConfig": "mlbparks"
    }
  }
}
```


Layer 4: Abstractions - Route

```
{
  "kind": "Route",
  "apiVersion": "v1",
  "metadata": {
    "name": "mlbparks-http-route",          (1)
    "labels": {
      "application": "mlbparks"              (2)
    },
    "annotations": {
      "description": "Route for application's http"
    }
  },
  "spec": {
    "host": "mlbparks.cloudapps.example.com", (3)
    "to": {                                   (4)
      "kind": "Service",
      "name": "mlbparks-http"
    }
  }
}
```

Kubernetes Ingress

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: test
spec:
  rules:
    - host: foo.bar.com
      http:
        paths:
          - path: /foo
            backend:
              serviceName: fooSvc
              servicePort: 80
    - host: bar.baz.com
      http:
        paths:
          - path: /bar
            backend:
              serviceName: barSvc
              servicePort: 80
```



← → ↻ <https://console.cloud.google.com/kubernetes/list?project>

应用 ftp://e4ftl01u.ecs.n... ftp://e4ftl01.cr.usgs... / 的索引 / all

Google Cloud Platform My Cloud Project of Kube...

Container Engine

Container clusters

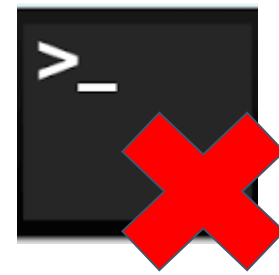
Container Registry

Container clusters

<input type="checkbox"/>	Name	Zone
<input checked="" type="checkbox"/>	cluster-1	asia-east1-a

avid-elevator-296 × +

```
tangfx128@avid-elevator-296:~$ docker run -ti --rm alpine ash
/ # ls
bin      dev      etc      home     lib      linuxrc  media
/ #
```





gRPC: <https://github.com/grpc/grpc-go>

protobuf: <https://github.com/golang/protobuf>

gRPC for Golang: <https://github.com/grpc/grpc-go>

Protobuf for Golang:

<https://github.com/golang/protobuf>

except kube-
APIServer



kubernetes



HTTP/2



protobuf

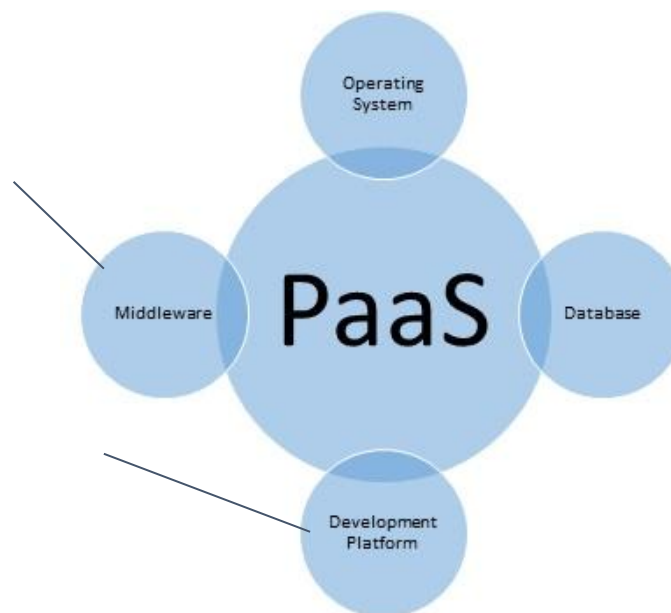
HTTP://

JSON Gateway

Using Kubernetes Charts to catalog application CI/CD template

Templated Micro-service
Store

Continuous
Integration/Distribution



charts:

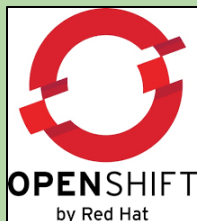
- spec:

<https://github.com/kubernetes/helm/blob/master/docs/charts.md>

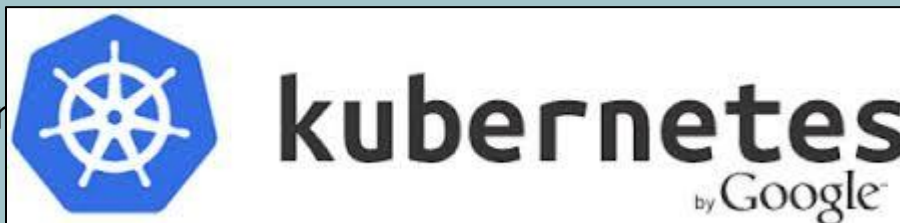
- architect: >

<https://github.com/kubernetes/helm/blob/master/docs/architecture.md#components>

This demonstration



Container Orchestration



Container Provisioning





fxtang@qingyuanos.com

<https://github.com/tangfeixiong/go-to-cloud-1>



Content:

- spring-boot app CI template
- gPRC charts template server
- build and ship

Feature:

- Golang 1.6
- based on more Kubernetes/Openshift community source code
- import "k8s.io/kubernetes/pkg/api"
- import "k8s.io/kubernetes/pkg/client/unversioned"
- import "k8s.io/kubernetes/pkg/runtime"
- import "github.com/openshift/origin/pkg/build/api"
- import

- github project

- API/gRPC protobuf
- Golang gRPC server stub
- Java gRPC client stub

Follow up in future:

- complete CI/CD
- canary distribution
- co-op with Openshift template
- rolling update
- blue-green deployment