Jenkins Kubernetes Operator

Tomasz Sęk

Senior Cloud Engineer

About

Tomasz Sęk
🐦 @tomaszsek1

Senior Cloud Engineer

- AWS and Azure
- custom Kubernetes distribution
- Golang
- Terraform

History

Jenkins snowflake

No one knew how it worked...

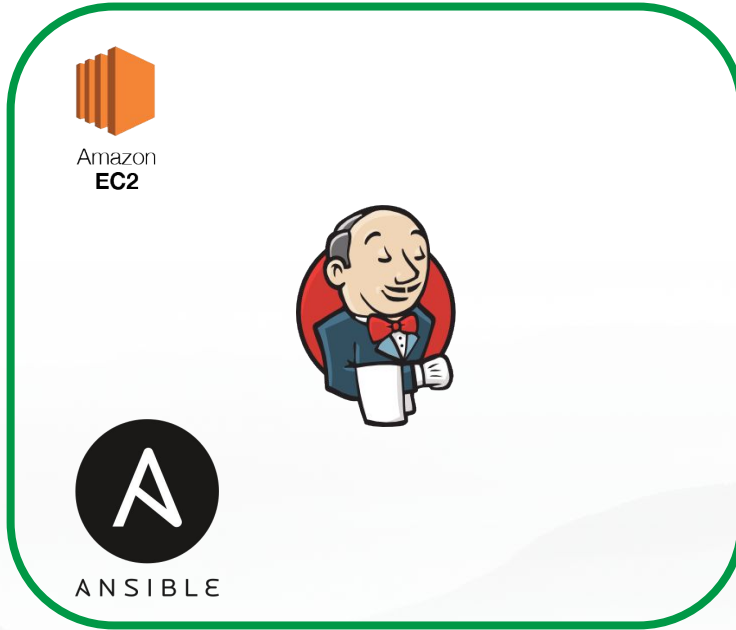# Manual

Software
Developers

Manual

- Jenkins out of date (unsecure)
- cloud infrastructure built by hand
- tons of unnecessary files in the Jenkins home
- no auto-scaling
- extremely high MTTR

Automated

Amazon **EC2**

ANSIBLE

Terraform

Software Developers

Automated

- very hard to write
- even harder to maintain
- changes "by hand" will brake deployment scripts
- significant problems with automatic plugins installation
- very limited immutability
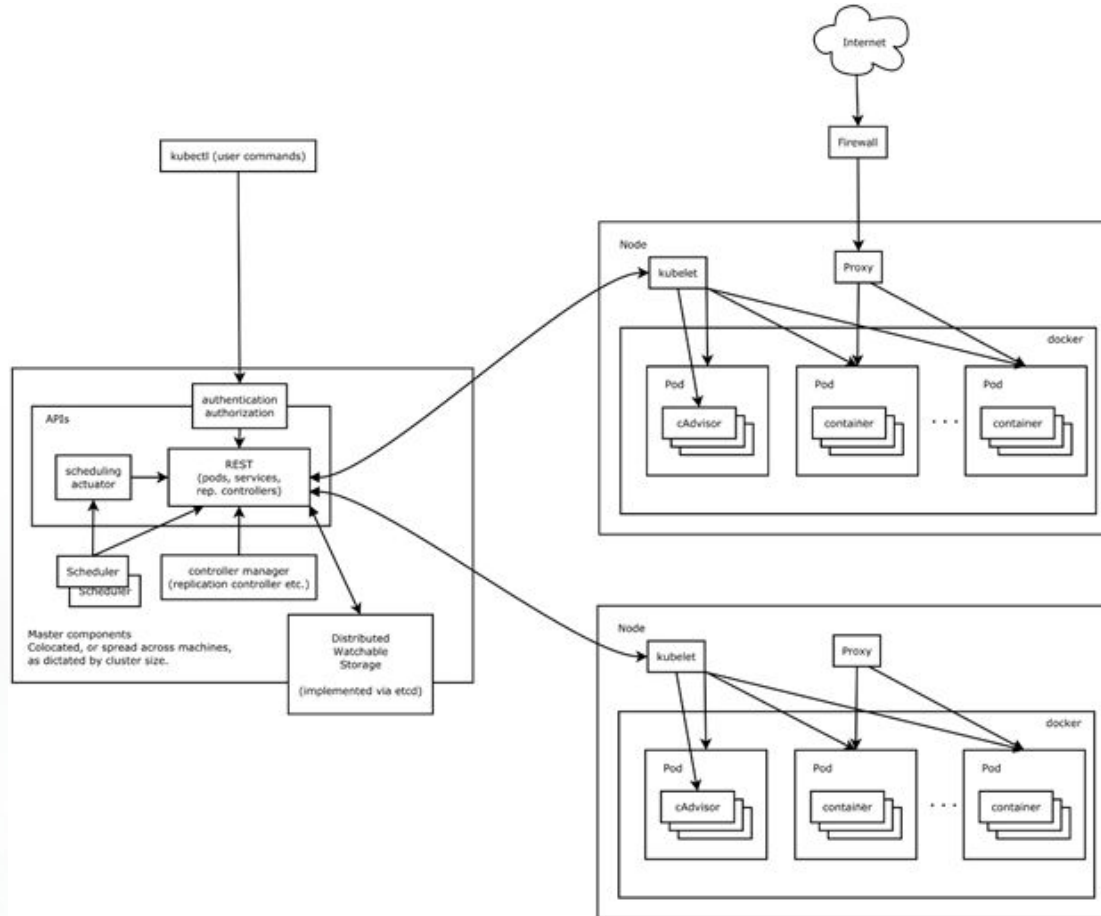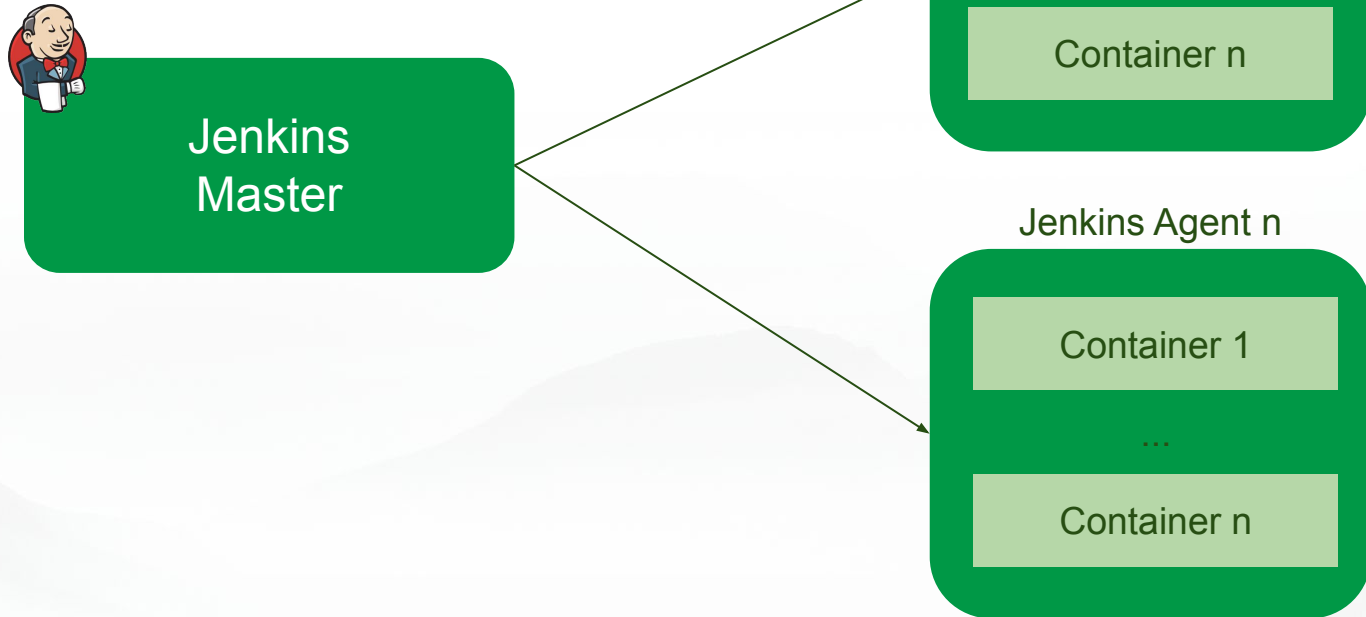- no auto-scaling
- no self-healing
- medium MTTR

Jenkins Kubernetes

1.4

Internet

Firewall

kubectl (user commands)

Node

kubelet

Proxy

docker

Pod

cAdvisor

Pod

container

Pod

container

APIs

authentication
authorization

scheduling
actuator

REST
(pods, services,
rep. controllers)

Scheduler
Scheduler

controller manager
(replication controller etc.)

Distributed
Watchable
Storage

(implemented via etcd)

Master components
Colocated, or spread across machines,
as dictated by cluster size.

Node

kubelet

Proxy

docker

Pod

cAdvisor

Pod

container

Pod

container

Let's talk! **@VirtusLab**

W W W . V I R T U S L A B . C O M

Jenkins & Kubernetes - why?

Jenkins Agent 1

Container 1

...

Container n

Jenkins
Master

Jenkins Agent n

Container 1

...

Container n

Jenkins & plain manifests

- ServiceAccount
- Role
- RoleBinding
- Deployment
- ConfigMap
- Secret
- Service
- PersistentVolumeClaim

```yaml
kind: ServiceAccount
apiVersion: v1
metadata:
 name: jenkins

kind: Deployment
apiVersion: apps/v1
metadata:
 name: jenkins
spec:
 replicas: 1
   spec:
     serviceAccountName: jenkins
     containers:
     - image: jenkins/jenkins:lts
       name: jenkins-master
       volumeMounts:
       - name: config
         mountPath: /etc/jenkins/config
     volumes:
     - name: config
       configMap:
         name: jenkins
```

Jenkins & Helm


```yaml
master:
 componentName: "jenkins-master"
 image: "jenkins/jenkins"
 tag: "lts"
 imagePullPolicy: "Always"
 imagePullSecretName:
 numExecutors: 0
 customJenkinsLabels: []
```

Jenkins & Helm


```
master:
  componentName: "jenkins-master"
  image: "jenkins/jenkins"
  tag: "lts"
  imagePullPolicy: "Always"
  imagePullSecretName:
  numExecutors: 0
  customJenkinsLabels: []
```

"Think twice before using Helm"

Kubernetes

- high cost / high gain
- high immutability
- auto-scaling
- limited self-healing
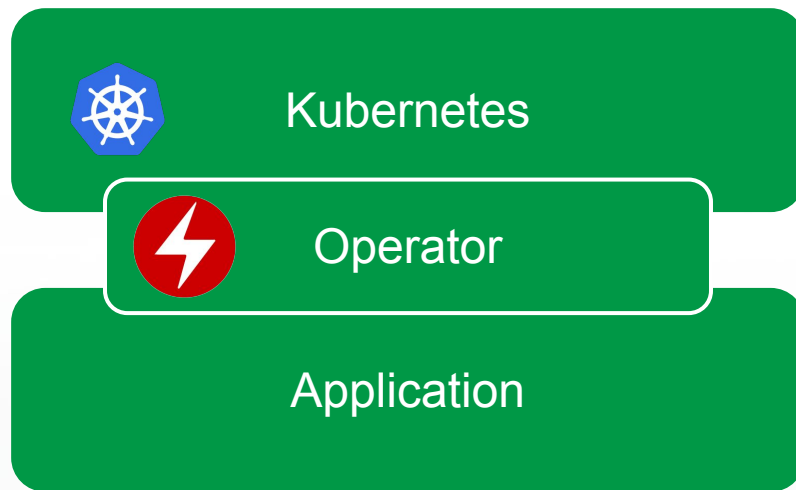- errors not visible for end users
- low MTTR

Jenkins Kubernetes Operator

Operator

Operator

Operator

- Kubernetes native
- separate software
- includes domain or application-specific knowledge
- reliably manages an application
- extends Kubernetes API through a set of CustomResourceDefinitions

Custom Resource Definition

- analogous to a class
- new API
- seamless integration with existing API
- kubectl support

```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
 name: kafkas.kafka.strimzi.io
spec:
 group: kafka.strimzi.io
 version: v1beta1
 scope: Namespaced
 names:
   kind: Kafka
   listKind: KafkaList
   singular: kafka
   plural: kafkas
```
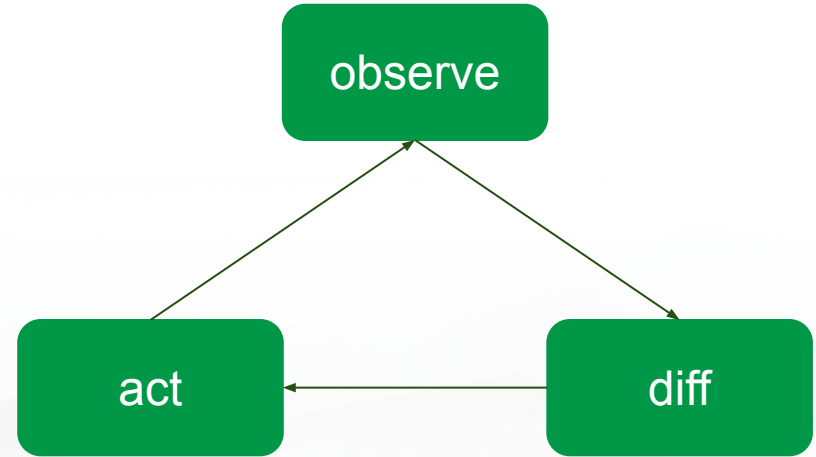
Custom Resource

- analogous to an object
- represents Kafka cluster instance

```
apiVersion: kafka.strimzi.io/v1beta1
kind: Kafka
metadata:
 name: example
spec:
 version: 2.3.0
 replicas: 3
```
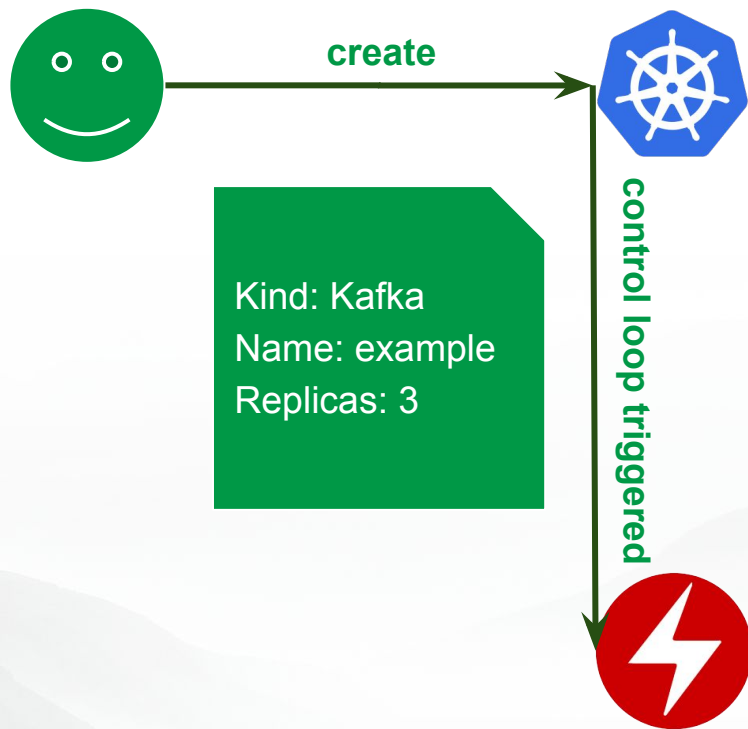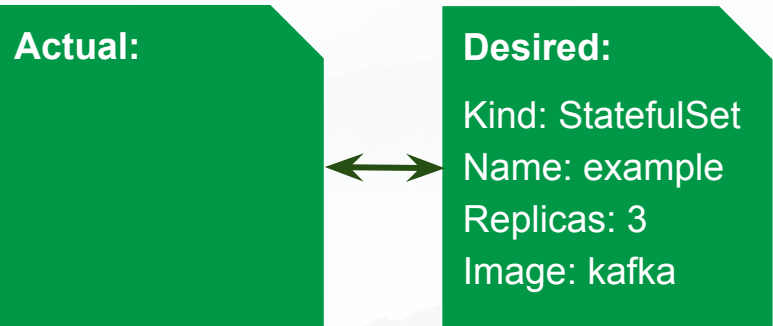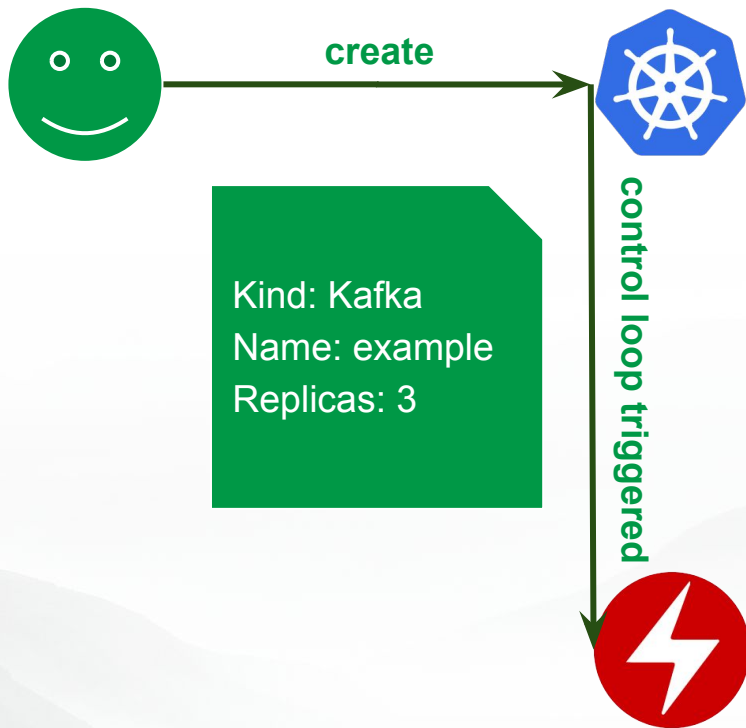
Control loop

- uses a watch operation on a Kubernetes resources
- analyzes difference between actual and desired state
- acts only on changes

Observe

create

control loop triggered

Kind: Kafka
Name: example
Replicas: 3

Diff

create

control loop triggered

Kind: Kafka
Name: example
Replicas: 3

Actual:

Desired:

Kind: StatefulSet
Name: example
Replicas: 3
Image: kafka

Act

Observe

Kafka 1

Kafka 2

Kafka 3

update

control loop triggered

Kind: Kafka
Name: example
Replicas: 2

Diff
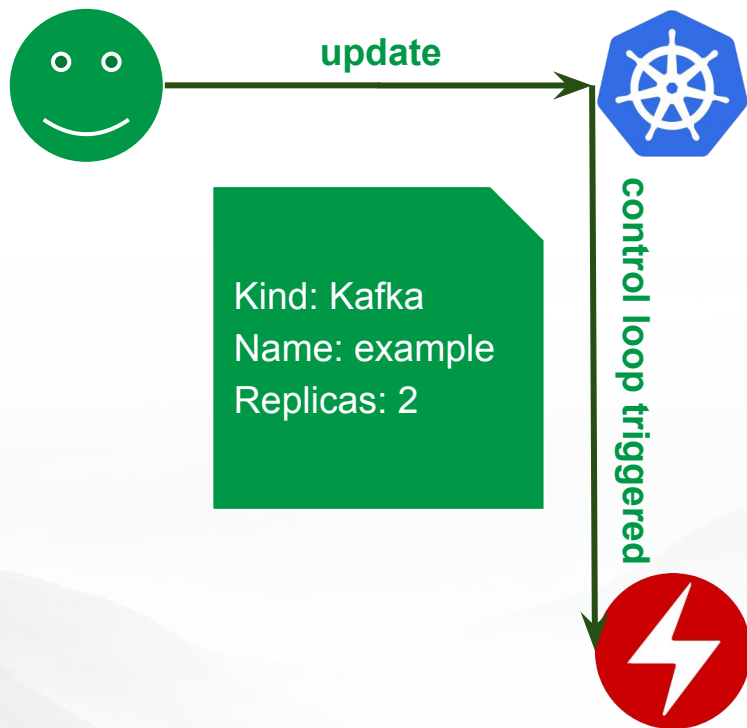
update

control loop triggered
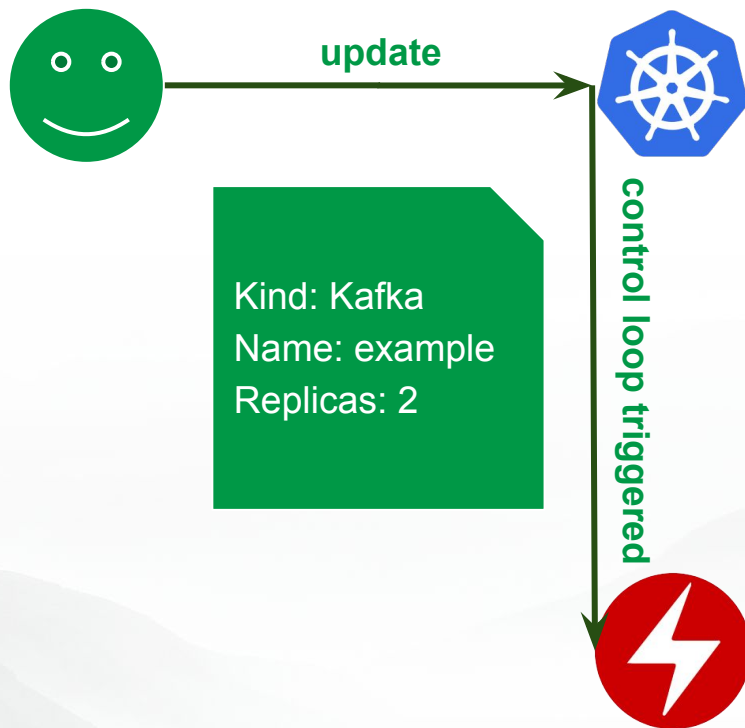
Kind: Kafka
Name: example
Replicas: 2

Kafka 1

Kafka 2

Kafka 3

Actual:

Kind: StatefulSet
Name: example
Replicas: 3
Image: kafka

Desired:

Kind: StatefulSet
Name: example
Replicas: 2
Image: kafka

Operator SDK

- tools for scaffolding and code generation to bootstrap a new project fast
- high level APIs and abstractions to write the operational logic more intuitively
- extensions to cover common operator use cases

Operator SDK

Operator SDK *Build, test, iterate*

Foundation → Test locally — Test with live cluster → Publish → Operator manifest / Operator image
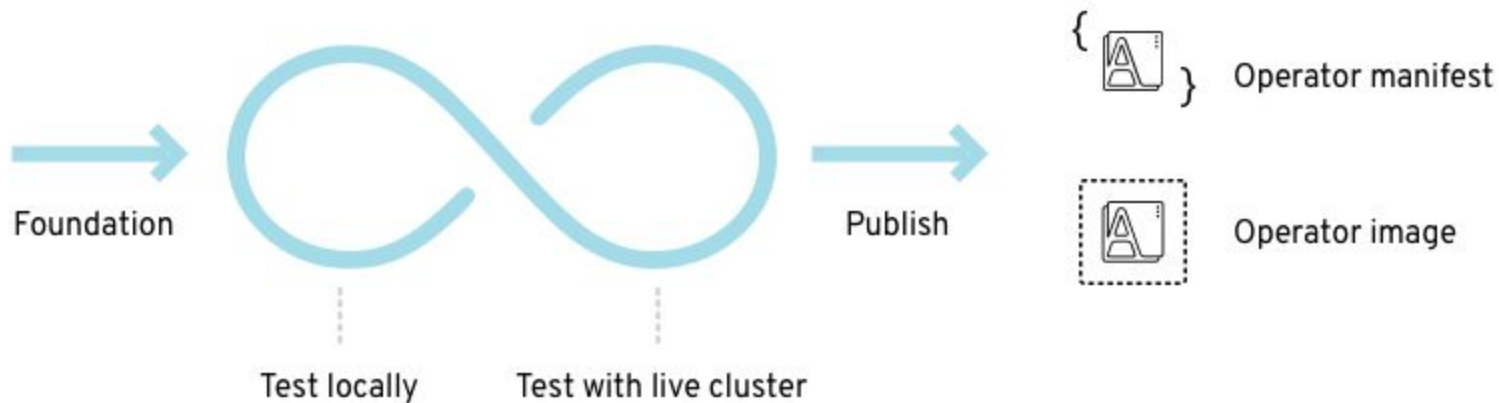
Operator Lifecycle Manager

Problems

- mutable (Jenkins home)
- no configuration as code
- no-autoscaling
- no self-healing
- plugins management
- backup and restore for jobs history and artifacts
- no security by default
- errors not visible for end users
- lack of end to end tests
- orphaned builds (no JNLP connection)

Goals

- run Jenkins on top of Kubernetes
- configuration in one place
- keep Jenkins instance configured
- make Jenkins immutable
- scale Jenkins agents
- make default configuration secure
- make backups of Jenkins builds history and artifacts
- make errors more visible for end users
- reduce Jenkins master unavailability

Operator

Kubernetes

Operator

Jenkins

Let's talk! @VirtusLab
WWW.VIRTUSLAB.COM

Architecture

Custom Resource

Secrets

Config Maps

Services etc

watch

Jenkins Operator

manage, watch

Jenkins Master

Jenkins Agent 1

Container 1

...

Container n

Jenkins Agent n

Container 1

...

Container n
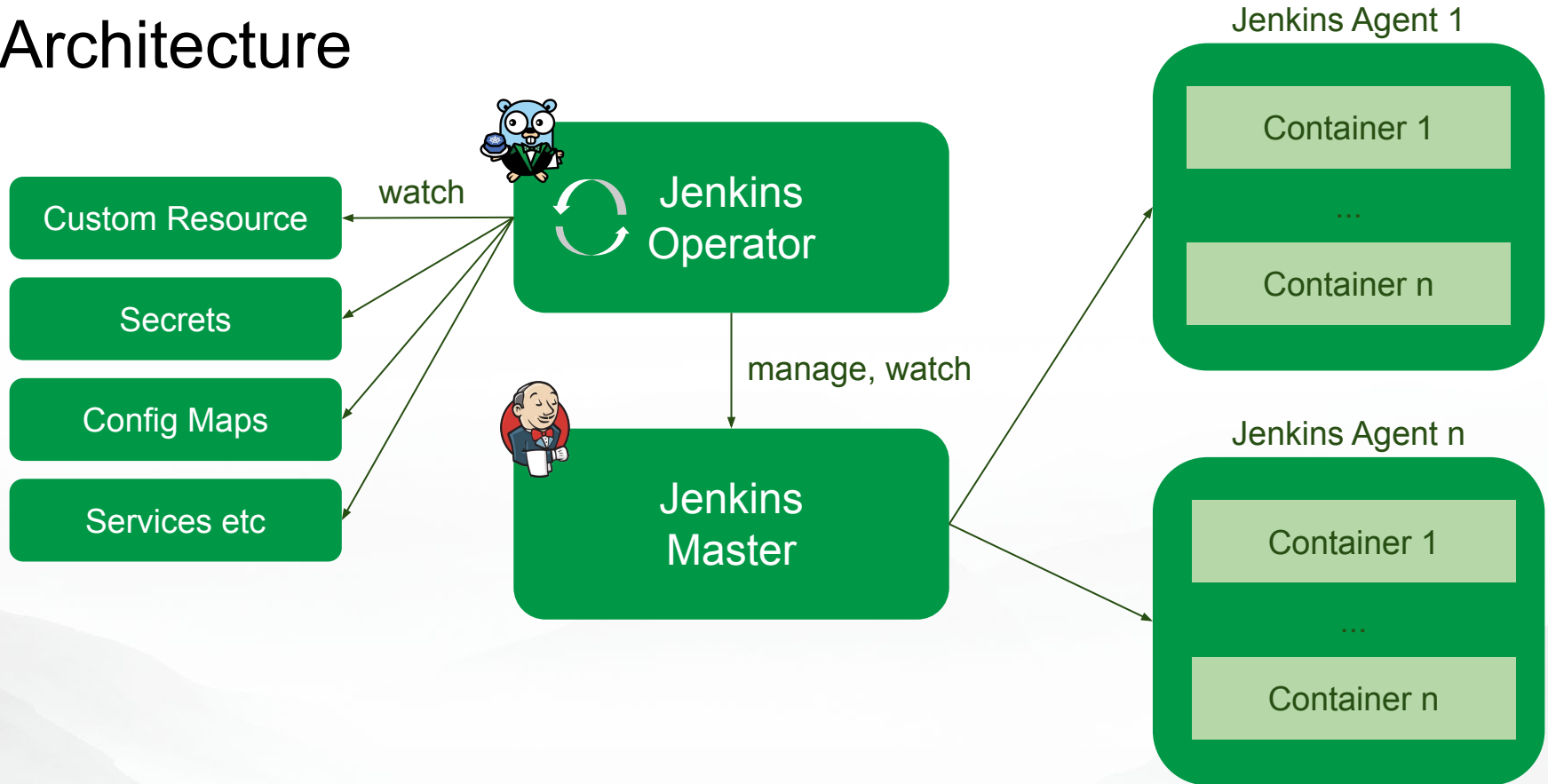
Control loop phases
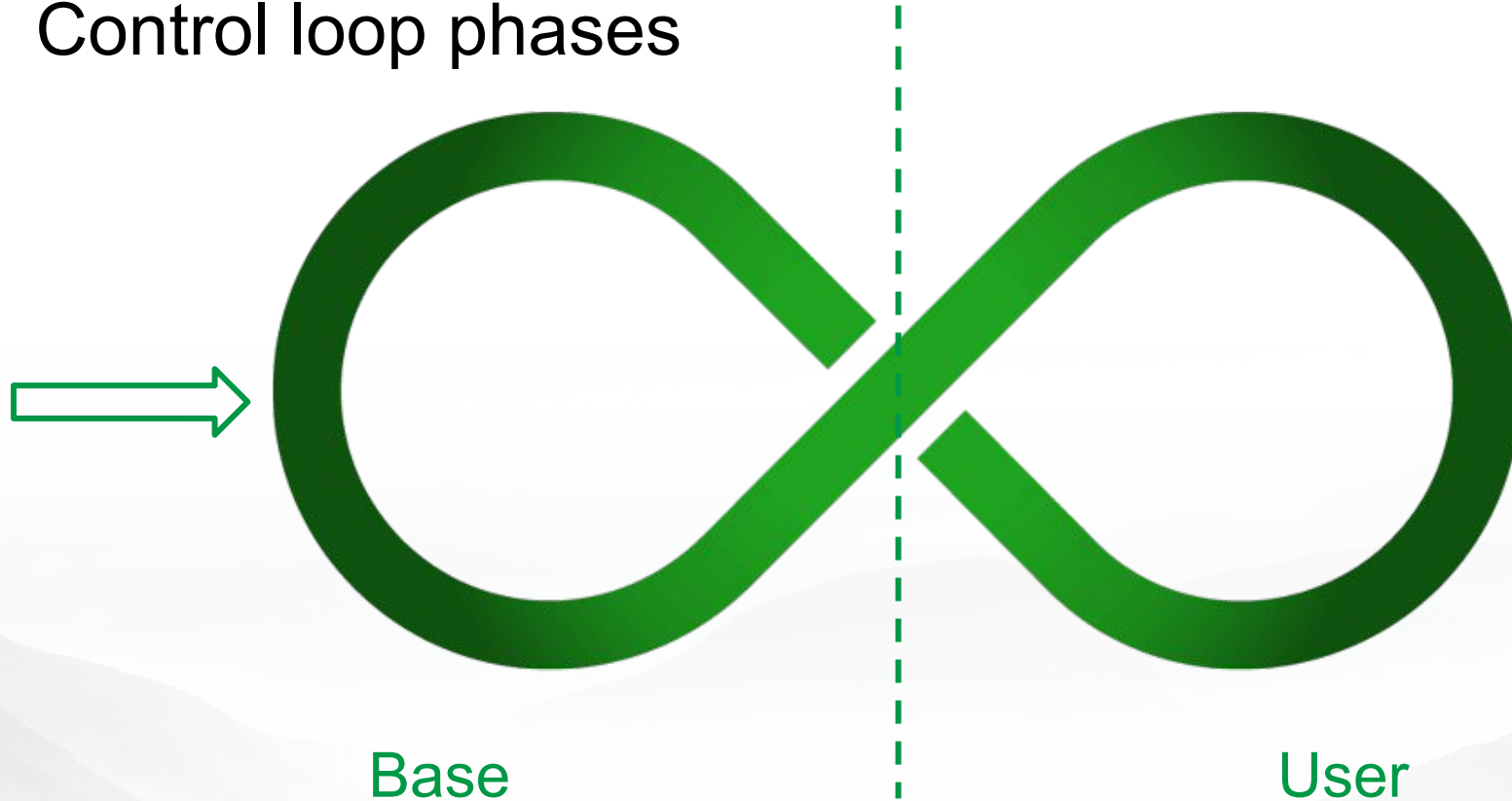
Base

User

Base phase

- set default Jenkins Custom Resource values
- validate
- ensure required resources (services, service accounts etc)
- ensure Jenkins master pod
- detect Jenkins pod start-up issues
- install plugins
- create a Jenkins user for the operator
- create Jenkins API token
- run base configuration (groovy scripts)

User phase

- validate

- ensure and run Jenkins seed jobs

- restore backup

- ensure user configuration (groovy scripts)

- ensure user configuration (Configuration as Code)

- create a backup

- enable backup trigger

Jenkins Custom Resources

```yaml
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
 name: jenkins.jenkins.io
spec:
 group: jenkins.io
 names:
    kind: Jenkins
    listKind: JenkinsList
    plural: jenkins
    singular: jenkins
 scope: Namespaced
 versions:
    - name : v1alpha2
      served: true
      storage: true
    - name : v1alpha1
      served: true
      storage: false
```

Master Pod atributes

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 master:
   containers:
     - name: jenkins-master
       image: jenkins/jenkins:lts
       resources:
         limits:
           cpu: 1500m
           memory: 3Gi
         requests:
           cpu: "1"
           memory: 500Mi
```

Master Pod attributes

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 master:
   containers:
    - name: jenkins-master
      image: jenkins/jenkins:lts
      resources:
        limits:
          cpu: 1500m
          memory: 3Gi
        requests:
          cpu: "1"
          memory: 500Mi
```

Groovy Scripts

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 groovyScripts:
   configurations:
     - name: user-config
   secret:
     name: "user-config-secrets"
```

Groovy Scripts

```yaml
kind: Secret
apiVersion: v1
type: Opaque
metadata:
 name: user-config-secrets
stringData:
 EXECUTORS_AMOUNT: "6"

apiVersion: v1
kind: ConfigMap
metadata:
 name: user-config
data:
 1-configure-executors.groovy: |
   import jenkins.*
   import jenkins.model.*
   import hudson.*
   import hudson.model.*
   Jenkins jenkins = Jenkins.getInstance()
   jenkins.setNumExecutors(Integer.valueOf(secrets["EXECUTORS_AMOUNT"]))
```

Groovy Scripts

```yaml
kind: Secret
apiVersion: v1
type: Opaque
metadata:
 name: user-config-secrets
stringData:
 EXECUTORS_AMOUNT: "6"

apiVersion: v1
kind: ConfigMap
metadata:
 name: user-config
data:
 1-configure-executors.groovy: |
   import jenkins.*
   import jenkins.model.*
   import hudson.*
   import hudson.model.*
   Jenkins jenkins = Jenkins.getInstance()
   jenkins.setNumExecutors(Integer.valueOf(secrets["EXECUTORS_AMOUNT"]))
```

Groovy Scripts

```yaml
kind: Secret
apiVersion: v1
type: Opaque
metadata:
 name: user-config-secrets
stringData:
 EXECUTORS_AMOUNT: "6"

apiVersion: v1
kind: ConfigMap
metadata:
 name: user-config
data:
 1-configure-executors.groovy: |
   import jenkins.*
   import jenkins.model.*
   import hudson.*
   import hudson.model.*
   Jenkins jenkins = Jenkins.getInstance()
   jenkins.setNumExecutors(Integer.valueOf(secrets["EXECUTORS_AMOUNT"]))
```

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 configurationAsCode:
   configurations:
     - name: user-config
   secret:
     name: "user-config-secrets"
```

```yaml
kind: Secret
apiVersion: v1
type: Opaque
metadata:
 name: user-config-secrets
stringData:
 SYSTEM_MESSAGE: "Psss secret value"

apiVersion: v1
kind: ConfigMap
metadata:
 name: user-config
data:
 1-system-message.yaml: |
    jenkins:
      systemMessage: ${SYSTEM_MESSAGE}
```

```yaml
kind: Secret
apiVersion: v1
type: Opaque
metadata:
 name: user-config-secrets
stringData:
 SYSTEM_MESSAGE: "Psss secret value"

apiVersion: v1
kind: ConfigMap
metadata:
 name: user-config
data:
 1-system-message.yaml: |
    jenkins:
      systemMessage: ${SYSTEM_MESSAGE}
```

CasC plugin

```yaml
kind: Secret
apiVersion: v1
type: Opaque
metadata:
 name: user-config-secrets
stringData:
 SYSTEM_MESSAGE: "Psss secret value"

apiVersion: v1
kind: ConfigMap
metadata:
 name: user-config
data:
 1-system-message.yaml: |
   jenkins:
     systemMessage: ${SYSTEM_MESSAGE}
```

Base plugins

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 master:
  basePlugins:
    - name: kubernetes
      version: 1.18.3
    - name: workflow-job
      version: "2.34"
    - name: workflow-aggregator
      version: "2.6"
    - name: git
      version: 3.12.0
    - name: job-dsl
      version: "1.76"
    - name: configuration-as-code
      version: "1.29"
    - name: configuration-as-code-support
      version: "1.19"
    - name: kubernetes-credentials-provider
      version: 0.12.1
```

User plugins

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 master:
   plugins:
     - name: simple-theme-plugin
       version: 0.5.1
```

Let's talk! **@VirtusLab**

WWW.VIRTUSLAB.COM

Seed jobs

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 seedJobs:
   - id: jenkins-operator
     targets: "cicd/jobs/*.jenkins"
     description: "Jenkins Operator repository"
     repositoryBranch: master
     repositoryUrl: https://github.com/jenkinsci/kubernetes-operator.git
```

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 seedJobs:
   - id: jenkins-operator
     targets: "cicd/jobs/*.jenkins"
     description: "Jenkins Operator repository"
     repositoryBranch: master
     repositoryUrl: https://github.com/jenkinsci/kubernetes-operator.git
```

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 seedJobs:
   - id: jenkins-operator
     targets: "cicd/jobs/*.jenkins"
     description: "Jenkins Operator repository"
     repositoryBranch: master
     repositoryUrl: https://github.com/jenkinsci/kubernetes-operator.git
```

```groovy
#!/usr/bin/env groovy
pipelineJob('build-jenkins-operator') {
    displayName('Build jenkins-operator')
    logRotator {
        numToKeep(10)
        daysToKeep(30)
    }
    definition {
        cpsScm {
            scm {
                git {
                    remote {
                        url('https://github.com/jenkinsci/kubernetes-operator.git')
                        credentials('jenkins-operator')
                    }
                    branches('*/master')
                }
            }
            scriptPath('cicd/pipelines/build.jenkins')
        }
    }
}
```

Seed jobs

```groovy
#!/usr/bin/env groovy
pipelineJob('build-jenkins-operator') {
    displayName('Build jenkins-operator')
    logRotator {
        numToKeep(10)
        daysToKeep(30)
    }
    definition {
        cpsScm {
            scm {
                git {
                    remote {
                        url('https://github.com/jenkinsci/kubernetes-operator.git')
                        credentials('jenkins-operator')
                    }
                    branches('*/master')
                }
            }
            scriptPath('cicd/pipelines/build.jenkins')
        }
    }
}
```

Seed jobs

```groovy
#!/usr/bin/env groovy
pipelineJob('build-jenkins-operator') {
    displayName('Build jenkins-operator')
    logRotator {
        numToKeep(10)
        daysToKeep(30)
    }
    definition {
        cpsScm {
            scm {
                git {
                    remote {
                        url('https://github.com/jenkinsci/kubernetes-operator.git')
                        credentials('jenkins-operator')
                    }
                    branches('*/master')
                }
            }
            scriptPath('cicd/pipelines/build.jenkins')
        }
    }
}
```

```groovy
#!/usr/bin/env groovy
def label = "build-jenkins-operator-${UUID.randomUUID().toString()}"
podTemplate(label: label,
        containers: [
                containerTemplate(name: 'go', image: 'golang:1-alpine', command: 'cat', ttyEnabled: true),
        ],
) {
   node(label) {
        stage('Init') {
            timeout(time: 3, unit: 'MINUTES') {
                checkout scm
            }
        }
        stage('Dep') {
            container('go') {
                sh 'make dep'
            }
        }
        stage('Test') {
            container('go') {
                sh 'make test'
            }
        }
        stage('Build') {
            container('go') {
                sh 'make build'
            }
        }
    }
}
```

Let's talk! @VirtusLab

WWW.VIRTUSLAB.COM

```groovy
#!/usr/bin/env groovy
def label = "build-jenkins-operator-${UUID.randomUUID().toString()}"
podTemplate(label: label,
        containers: [
                containerTemplate(name: 'go', image: 'golang:1-alpine', command: 'cat', ttyEnabled: true),
        ],
) {
    node(label) {
        stage('Init') {
            timeout(time: 3, unit: 'MINUTES') {
                checkout scm
            }
        }
        stage('Dep') {
            container('go') {
                sh 'make dep'
            }
        }
        stage('Test') {
            container('go') {
                sh 'make test'
            }
        }
        stage('Build') {
            container('go') {
                sh 'make build'
            }
        }
    }
}
```

Seed jobs

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 master:
   containers:
     - name: backup
       env:
         - name: BACKUP_DIR
           value: /backup
         - name: JENKINS_HOME
           value: /jenkins-home
         - name: BACKUP_COUNT
           value: "20"
       image: virtuslab/jenkins-operator-backup-pvc:v0.0.6
       volumeMounts:
         - mountPath: /jenkins-home
           name: jenkins-home
         - mountPath: /backup
           name: backup
   volumes:
     - name: backup
       persistentVolumeClaim:
         claimName: jenkins-backup
```

Backup & restore

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 master:
   containers:
    - name: backup
      env:
       - name: BACKUP_DIR
         value: /backup
       - name: JENKINS_HOME
         value: /jenkins-home
       - name: BACKUP_COUNT
         value: "20"
      image: virtuslab/jenkins-operator-backup-pvc:v0.0.6
      volumeMounts:
       - mountPath: /jenkins-home
         name: jenkins-home
       - mountPath: /backup
         name: backup
   volumes:
    - name: backup
      persistentVolumeClaim:
        claimName: jenkins-backup
```

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 master:
   containers:
   - name: backup
     env:
     - name: BACKUP_DIR
       value: /backup
     - name: JENKINS_HOME
       value: /jenkins-home
     - name: BACKUP_COUNT
       value: "20"
     image: virtuslab/jenkins-operator-backup-pvc:v0.0.6
     volumeMounts:
     - mountPath: /jenkins-home
       name: jenkins-home
     - mountPath: /backup
       name: backup
   volumes:
   - name: backup
     persistentVolumeClaim:
       claimName: jenkins-backup
```

Backup & restore

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: jenkins-backup
spec:
 accessModes:
    - ReadWriteOnce
 resources:
   requests:
     storage: 20Gi
```

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 backup:
   containerName: backup
   action:
     exec:
       command:
         - /home/user/bin/backup.sh
   interval: 30
   makeBackupBeforePodDeletion: true
 restore:
   containerName: backup
   action:
     exec:
       command:
         - /home/user/bin/restore.sh
```

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 backup:
   containerName: backup
   action:
     exec:
       command:
         - /home/user/bin/backup.sh
   interval: 30
   makeBackupBeforePodDeletion: true
 restore:
   containerName: backup
   action:
     exec:
       command:
         - /home/user/bin/restore.sh
```

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 backup:
   containerName: backup
   action:
     exec:
       command:
         - /home/user/bin/backup.sh
   interval: 30
   makeBackupBeforePodDeletion: true
 restore:
   containerName: backup
   action:
     exec:
       command:
         - /home/user/bin/restore.sh
```

Notifications

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 notifications:
   - loggingLevel: info
     verbose: true
     name: slack
     slack:
       webHookURLSecretKeySelector:
         secret:
           name: user-config-secrets
         key: SLACK_WEBHOOK
```

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 notifications:
   - loggingLevel: info
     verbose: true
     name: slack
     slack:
       webHookURLSecretKeySelector:
         secret:
           name: user-config-secrets
         key: SLACK_WEBHOOK
```

Notifications

Notifications

```yaml
apiVersion: jenkins.io/v1alpha2
kind: Jenkins
metadata:
 name: example
spec:
 notifications:
   - loggingLevel: info
     verbose: true
     name: slack
     slack:
       webHookURLSecretKeySelector:
         secret:
           name: user-config-secrets
         key: SLACK_WEBHOOK
```

Demo

- custom theme (simple-theme-plugin)
- amount of executors - 3 (groovy script)
- system message (CasC plugin)
- backup on PVC
- seed job
- notifications on Slack

Roadmap

- deployment
- refactoring
- improvements

Questions?

Thank you