



Kubernetes

The API of Everything

Cloud Native Theater | Cloud Native University by Tibo Beijen

Kubernetes is a ~~container orchestrator~~

Kubernetes is an API

Designed to orchestrate things

Including containers

Controllers

API Machinery

Controllers reconcile

Desired state

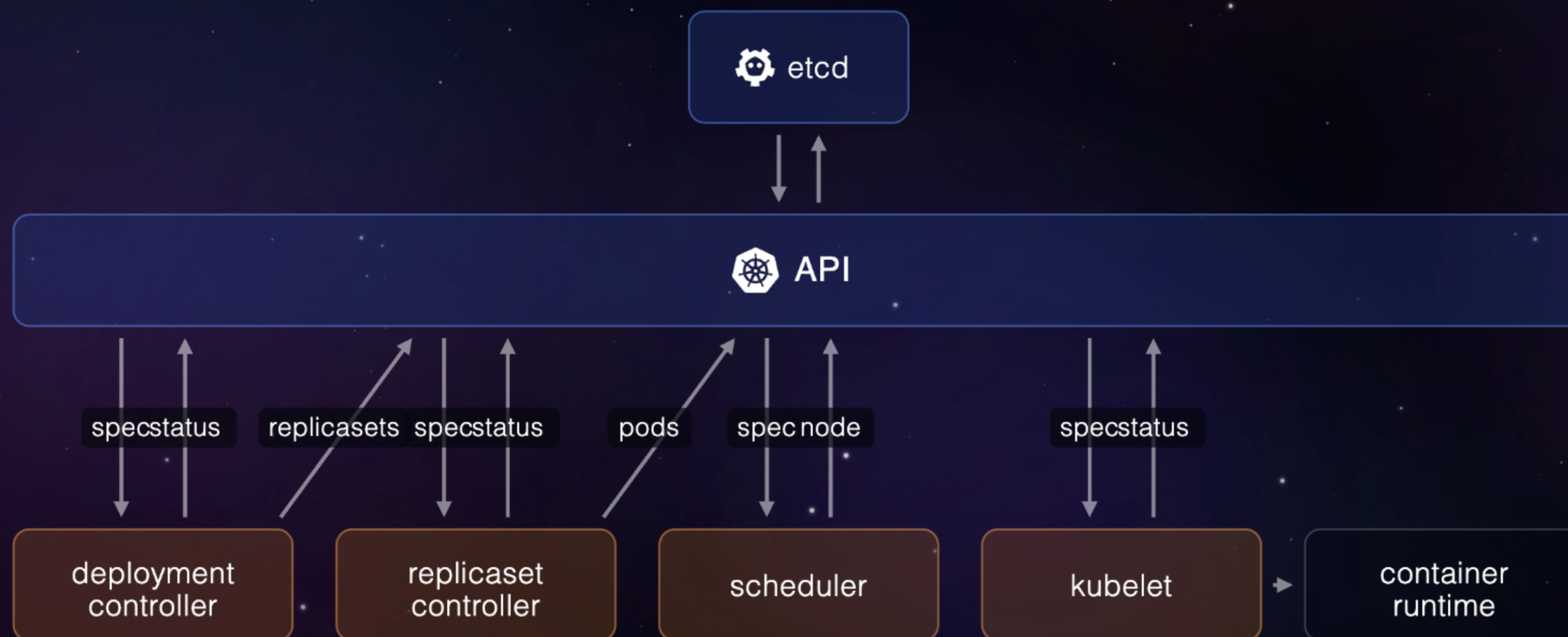
Actual state

NAME	READY	STATUS	RESTARTS	AGE
kube-controller-manager	1/1	Running	0	7d20h

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: coredns
  namespace: kube-system
spec:
  replicas: 2
  spec:
    containers:
      - name: coredns
        image: coredns/coredns:1.14.2
status:
  replicas: 2
```

Orchestration

Controllers independently working together



Extension

API superpower

API → Controllers → Automate

Extended API → More Controllers → Automate even more

Build Platforms, the Universe, and Everything! TM

Operators

CRD

- Custom Resource Definition
- Extending the API



Controller

- Compute workload
- Interacting with Kubernetes API, via Custom Resources (CR)
- Reconciliation loop
- Domain knowledge

Custom Resource Definition

GET /apis/apps/v1/namespaces/kube-system/deployments/coredns

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: coredns
  namespace: kube-system
spec:
  replicas: 2
  selector: {}
  template:
    metadata: {}
    spec:
      containers:
      - name: coredns
        image: coredns/coredns:1.14.2
        volumes: []
status:
  replicas: 2
```

GET /apis/keda.sh/v1alpha1/namespaces/my-app/scaledobjects/workers

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: workers
  namespace: my-app
spec:
  maxReplicaCount: 5
  minReplicaCount: 1
  scaleTargetRef:
    apiVersion: argoproj.io/v1alpha1
    kind: Rollout
    name: workers
  triggers:
  - type: kafka
    metadata:
      topic: my-events
status:
  hpaName: keda-hpa-workers
  originalReplicaCount: 1
```

Operators

Controllers that use Custom Resources

Built-in controller

NAME	READY	STATUS
kube-controller-manager	1/1	Running

KEDA

NAME	READY	STATUS
keda-admission-webhooks-75874c496d-vl2b9	1/1	Running
keda-operator-dd6f88dd5-kgwws	1/1	Running
keda-operator-metrics-apiserver-9858d9f54-qdgck	1/1	Running

Don't see chaos

Scheduling & Orchestration

Crossplane

CNCF GRADUATED

KEDA

CNCF GRADUATED

Knative

CNCF GRADUATED

kubernetes

CNCF GRADUATED

FLUID

CNCF INCUBATING

KARMADA

CNCF INCUBATING

Kubeflow

CNCF INCUBATING

VOLCANO

CNCF INCUBATING

wasmcloud

CNCF INCUBATING

Amazon ECS

MESOS

ARMADA

Azure Service Fabric

backend

capsule

Red Hat CNP

中移智基

Clusternet

Clusterpedia

COZYSTACK

SWARM

DolphinScheduler

ERASER

HAMI

ISSCloud

KCP

kOS

kORDENT

kestra

koordinator

kube-green

kube-rs

KubeAdmiral

KubeWharf

KubeFleet

KubeSlice

HUBESTELLAR

Kured

slurm

Nomad

Open Cluster Management

OPEN FUNCTION

Open Nebula

PREFECT

SERVERLESS DEVS

StackStorm

upbound

Katalyst

Service Mesh

Istio

CNCF GRADUATED

LINKERD

CNCF GRADUATED

Aeraki Mesh

AWS App Mesh

Consul

EaseMesh

Amazon Fargate Service Mesh

GLASNOSTIC

GLOO MESH

greymatter.io

KMESh

Kuma

Worledge

Open Service Mesh

OPENSERGO

Sermant

Service Mesh Interface

SMP

slime

TSB

traefik mesh

API Gateway

EMISSARY INGRESS

CNCF INCUBATING

3SCALE

akana

APIOAK

APISIX

AWAP Cloud

Easegress

saaras.io

envoy gateway

GLOO GATEWAY

GRAVITEE-IO

Hongo

Migress

kgateway

Kong

krakenD

Kuadrant

KubeGateway

KUSK

lunar.dev

Membrane

MuleSoft

ngrok

traefik proxy

Tyk

WSO2 API Microgateway

Remote Procedure Call

gRPC

CNCF INCUBATING

Apache gRPC

Apache Thrift

CloudWeGo

DEBBO

NGO

GoFe.Jay

Connect

SOFARPC

SRPC

TARS

See the pattern



Argo Rollouts

Advanced deployments: Blue-green, Canary

NAME		READY	STATUS	RESTARTS	AGE
argo-rollouts-69ff8d77b5-d2x2w		1/1	Running	0	10h
argo-rollouts-69ff8d77b5-nwg49		1/1	Running	0	14h

```
k -n applications get rollouts.v1alpha1.argoproj.io my-app -o yaml
```

```
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: workers
  namespace: my-app
spec:
  strategy:
    canary:
      canaryService: workers-canary
      stableService: workers-stable
      steps:
        - setWeight: 20
        - pause:
            duration: 30s
      workloadRef:
        name: workers-ref
  status:
    canary:
      weights:
        stable:
          serviceName: workers-stable
          weight: 100
```

See the pattern



Strimzi

Kafka clusters as Kubernetes resources

NAME	READY	STATUS
strimzi-cluster-operator-7f778d4b5d-6sdd2	1/1	Running
strimzi-access-operator-6657746d59-jqhmh	1/1	Running

```
k -n strimzi get kafkas.kafka.strimzi.io prod -o yaml
```

```
apiVersion: kafka.strimzi.io/v1beta2
kind: Kafka
metadata:
  name: prod
  namespace: strimzi
spec:
  kafka:
    version: 3.9.0
    listeners:
      - name: internal
        port: 9092
        type: internal
      - name: saslscram
        port: 9094
        authentication:
          type: scram-sha-512
status:
  kafkaVersion: 3.9.0
  kafkaMetadataState: KRaft
  conditions:
    - type: Ready
      status: "True"
```

See the pattern



Crossplane

Cloud infrastructure from Kubernetes

NAME	READY
crossplane-844d47444b-75vmf	1/1
crossplane-844d47444b-jccxr	1/1
crossplane-rbac-manager-6bdfcc47fd-jgjb5	1/1
provider-aws-iam-8ecb8d0d329b-6696c879d9-flljf	1/1
provider-aws-s3-dd62e217befa-5986db8db8-76vdp	1/1
provider-aws-rds-ba4120f45a6a-7dff7cfbc6-ws7zv	1/1

```
k get roles.iam.aws.upbound.io my-app-irsa -o yaml
```

```
apiVersion: iam.aws.upbound.io/v1beta1
kind: Role
metadata:
  name: my-app-irsa
  namespace: my-app
spec:
  forProvider:
    assumeRolePolicy: '{"Version":"2012-10-17",...}'
status:
  atProvider:
    arn: arn:aws:iam::123456789012:role/my-app-irsa
  conditions:
    - type: Synced
      status: "True"
      reason: ReconcileSuccess
    - type: Ready
      status: "True"
```

See the pattern



Kyverno

Policy as Kubernetes resources

NAME	READY	STATUS
kyverno-admission-controller-59f964868c-jw94c	1/1	Running
kyverno-background-controller-586899bf8-bptsm	1/1	Running
kyverno-cleanup-controller-769797cd47-9d9q5	1/1	Running
kyverno-reports-controller-5dbff45cff-dhkr6	1/1	Running

```
k get clusterpolicies.kyverno.io require-labels -o yaml
```

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-labels
spec:
  validationFailureAction: Enforce
  rules:
    - name: check-team-label
      match:
        any:
          - resources:
              kinds:
                - Deployment
      validate:
        message: "Label 'team' is required."
        pattern:
          metadata:
            labels:
              team: "?*"
  status:
    conditions:
      - type: Ready
        status: "True"
```

See the pattern



External Secrets Operator

Sync secrets from external stores

NAME	READY	STATUS
external-secrets-76887ccc75-8qngv	1/1	Running
external-secrets-cert-controller-5c48bc477f-d2wfw	1/1	Running
external-secrets-webhook-7bb8cc6577-72qcp	1/1	Running

```
k -n my-app get externalsecrets.external-secrets.io my-app -o yaml
```

```
apiVersion: external-secrets.io/v1
kind: ExternalSecret
metadata:
  name: my-app
  namespace: my-app
spec:
  refreshInterval: 1h
  secretStoreRef:
    kind: ClusterSecretStore
    name: aws-params-store
  target:
    name: my-app-secrets
    creationPolicy: Owner
  data:
  - secretKey: api-key
    remoteRef:
      key: /apps/prod/my-app/api-key
status:
  conditions:
  - type: Ready
    status: "True"
    reason: SecretSynced
```

See the pattern



Cluster API

Provision Kubernetes clusters via CRDs

NAME	READY
capi-controller-manager-77b8f4d75c-xk9q2	1/1
capi-kubeadm-bootstrap-controller-manager-6bc4f9b8d6-r8wdt	1/1
capi-kubeadm-control-plane-controller-manager-5f8b9468c4-jm7nz	1/1

```
k get clusters.cluster.x-k8s.io my-cluster -o yaml
```

```
apiVersion: cluster.x-k8s.io/v1beta1
kind: Cluster
metadata:
  name: my-cluster
  namespace: default
spec:
  controlPlaneEndpoint:
    host: my-cluster-api.example.com
    port: 6443
  infrastructureRef:
    apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
    kind: AWSCluster
    name: my-cluster
  controlPlaneRef:
    kind: KubeadmControlPlane
    name: my-cluster-cp
status:
  phase: Provisioned
  infrastructureReady: true
  controlPlaneReady: true
```

Think of Kubernetes

Not as

- Something you install on servers
- That runs containers
- And happens to have an API

But as

- An **API**
- That can be **extended**
by custom resource definitions
- That can orchestrate **everything**
using operators
- And just happens to run on servers

Thank you!

