Clusters as Cattle

Extending Kubernetes for Multi-Cluster and Multi-Cloud Workloads

Illya Chekrygin
Founding Engineer, Upbound

ichekrygin

illya_chekrygin

Cloud Computing

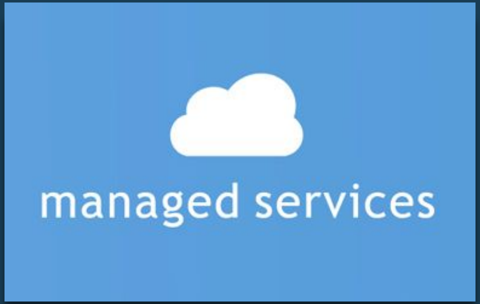
- Predominate
- On demand
- Business oriented
- World-class managed services
- Global scale
- Pay-Per-Use

Cloud Providers

- Competitive
- Open-Source adopters
- Closed-Source offering

Managed Services

- Dependencies
- Worry free (almost)
- Hands off (almost)
- You get an SLA
- For which you Pay



Managed Services Overlap

- Same services
- Different:
  - Provisioning
  - Configuration
  - Scale

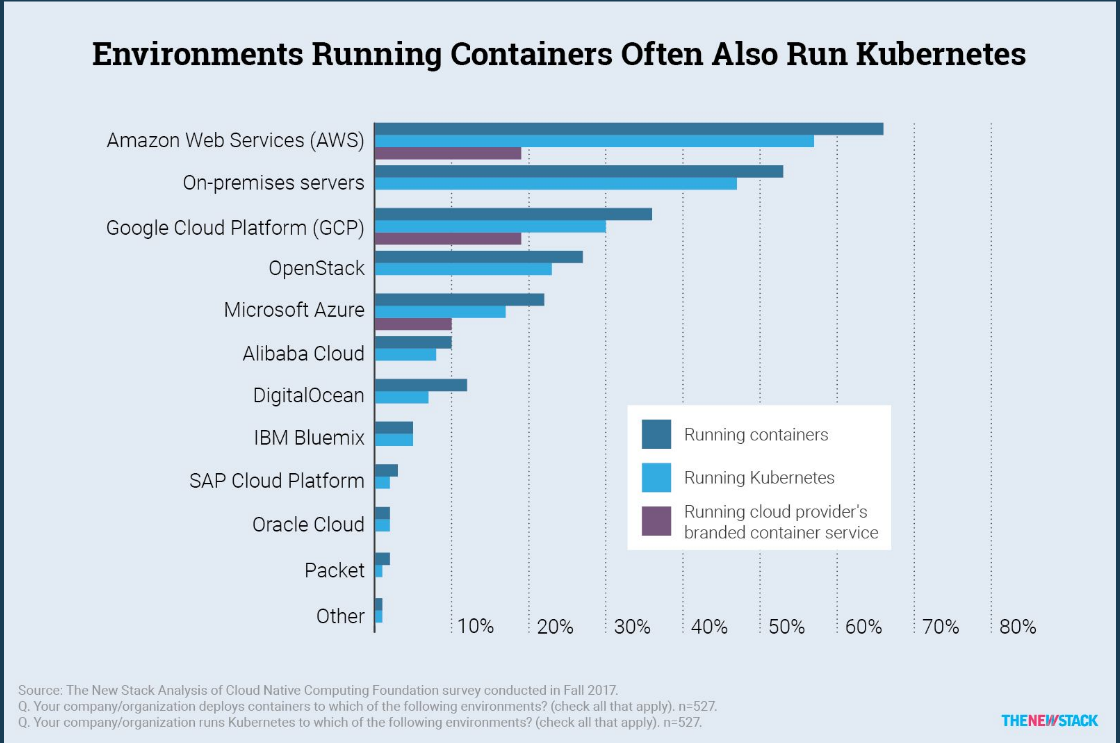
Multicloud

- Is reality
- Needs:
  - Control Plane

Kubernetes

- What is it
  - container platform
  - a microservices platform
  - a portable cloud platform and a lot more.
- Borne at Google
  - Borg / Omega
- Open Source
- Growing community
  - Microsoft, RedHat, IBM, Docker
- Managed by CNCF

Kubernetes

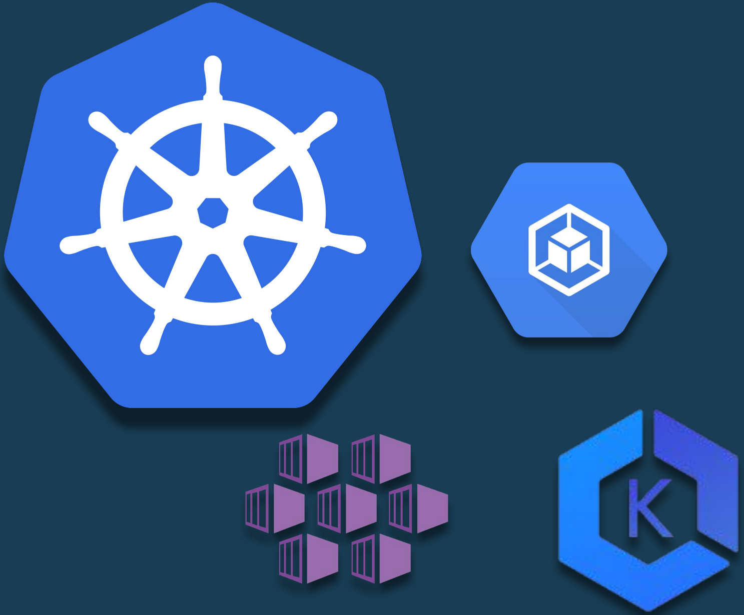


Environments Running Containers Often Also Run Kubernetes

Source: The New Stack Analysis of Cloud Native Computing Foundation survey conducted in Fall 2017.
Q. Your company/organization deploys containers to which of the following environments? (check all that apply). n=527.
Q. Your company/organization runs Kubernetes to which of the following environments? (check all that apply). n=527.

Managed Kubernetes

- Many Choices
- Easy to Provision
- Not Consistent



Kubernetes API

- Declarative Style
- Level-based
- State separation: Desired (Spec) vs. Observed (Status)
- Complete
- Authoritative
- Extensible

Extending Kubernetes

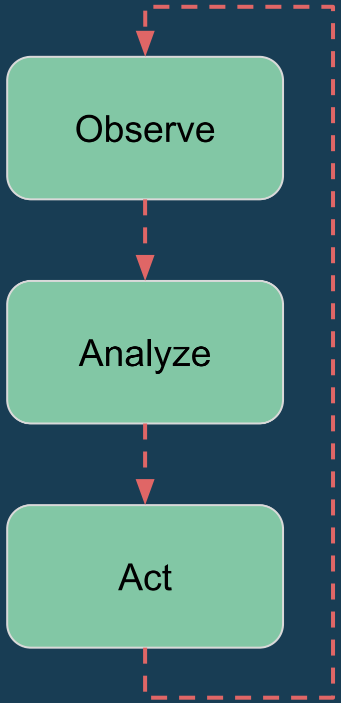
- Controller Pattern
- Custom Resource Definitions
- Operators
  - Deploy + Package
- Frameworks
  - controller-runtime
  - client-go

Extending Kubernetes

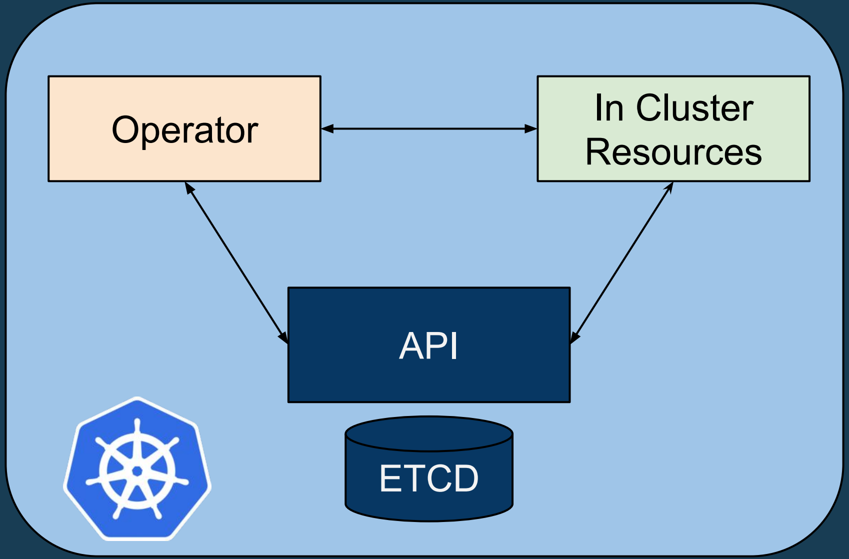
- Controller Pattern
- Custom Resource Definitions
- Operators
    - Deploy + Package
- Frameworks
    - controller-runtime
    - client-go

Controller

- Retrieve
- Process
    - Actual State -> Desired State
    - CRUD
- Update Status



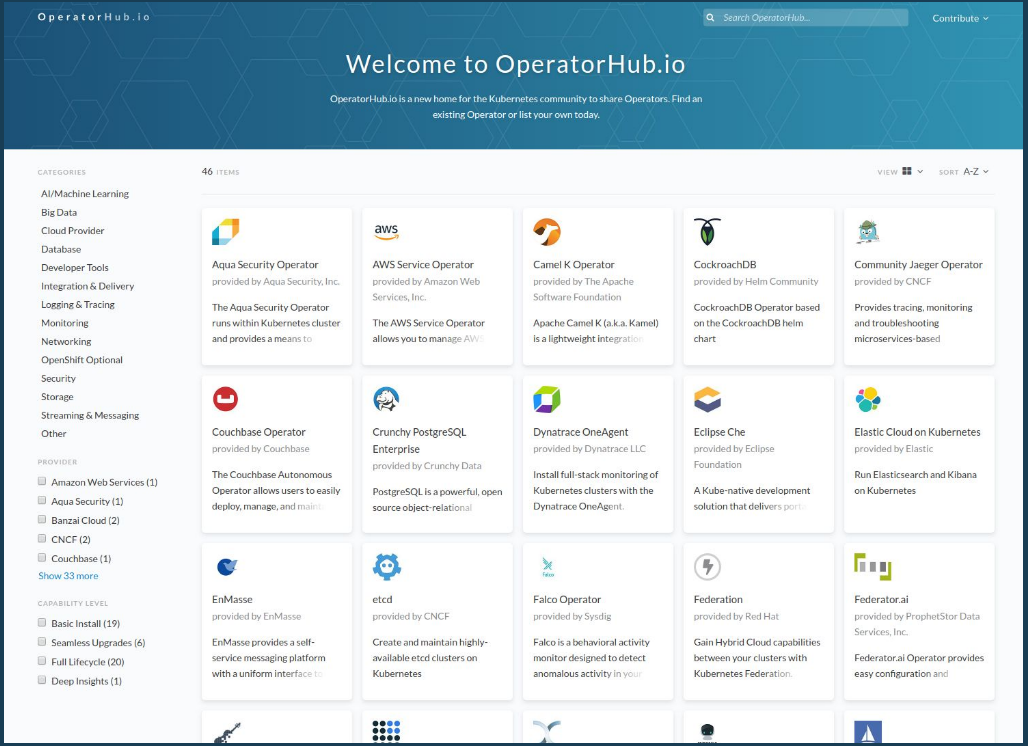
In Cluster Resources



Stateful Applications

```
grtl/mysql-operator
o[illegible]sql-opera[illegible]
p[illegible]/mysql-o[illegible]
b[illegible]ud/mysql-operator
Per[illegible]-Lab/percona-xtradb-cluster-operator
```

Operator Resources



Welcome to OperatorHub.io

OperatorHub.io is a new home for the Kubernetes community to share Operators. Find an existing Operator or list your own today.

Aqua Security Operator
provided by Aqua Security, Inc.
The Aqua Security Operator runs within Kubernetes cluster and provides a means to

AWS Service Operator
provided by Amazon Web Services, Inc.
The AWS Service Operator allows you to manage AWS

Camel K Operator
provided by The Apache Software Foundation
Apache Camel K (a.k.a. Kamel) is a lightweight integration

CockroachDB
provided by Helm Community
CockroachDB Operator based on the CockroachDB helm chart

Community Jaeger Operator
provided by CNCF
Provides tracing, monitoring and troubleshooting microservices-based

Couchbase Operator
provided by Couchbase
The Couchbase Autonomous Operator allows users to easily deploy, manage, and maint

Crunchy PostgreSQL Enterprise
provided by Crunchy Data
PostgreSQL is a powerful, open source object-relational

Dynatrace OneAgent
provided by Dynatrace LLC
Install full-stack monitoring of Kubernetes clusters with the Dynatrace OneAgent.

Eclipse Che
provided by Eclipse Foundation
A Kube-native development solution that delivers port

Elastic Cloud on Kubernetes
provided by Elastic
Run Elasticsearch and Kibana on Kubernetes

EnMasse
provided by EnMasse
EnMasse provides a self-service messaging platform with a uniform interface to

etcd
provided by CNCF
Create and maintain highly-available etcd clusters on Kubernetes

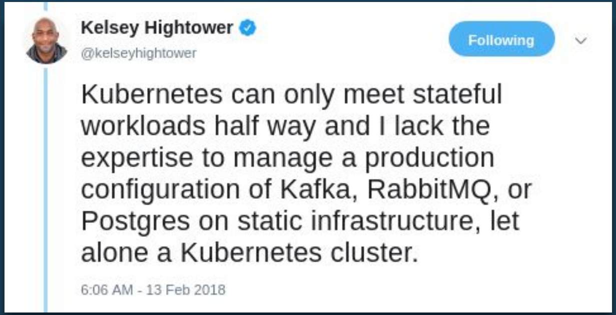
Falco Operator
provided by Sysdig
Falco is a behavioral activity monitor designed to detect anomalous activity in your

Federation
provided by Red Hat
Gain Hybrid Cloud capabilities between your clusters with Kubernetes Federation.

Federator.ai
provided by ProphetStor Data Services, Inc.
Federator.ai Operator provides easy configuration and

Kubernetes Platform Services

- Do not use managed services?
- Pure Play → Portability
- Problems:
  - Maturity
  - Support and SLA
  - Unified Console
  - Domain knowledge



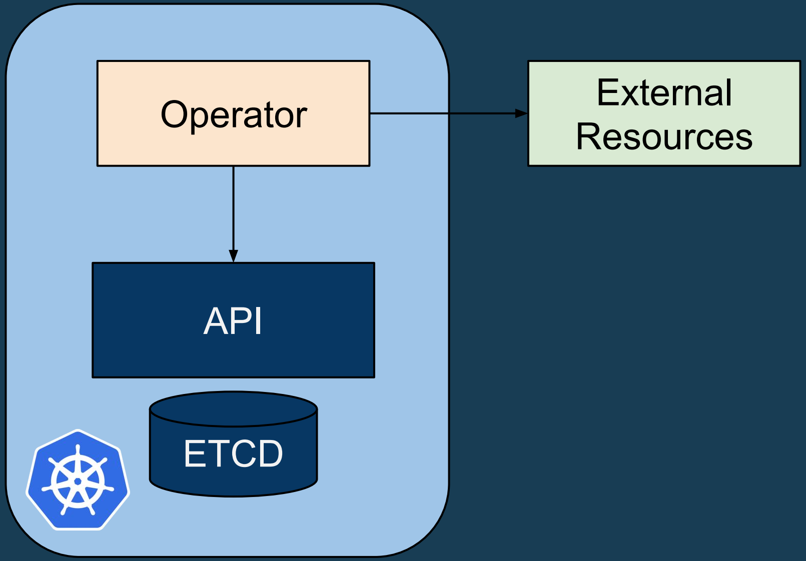
Kelsey Hightower
@kelseyhightower

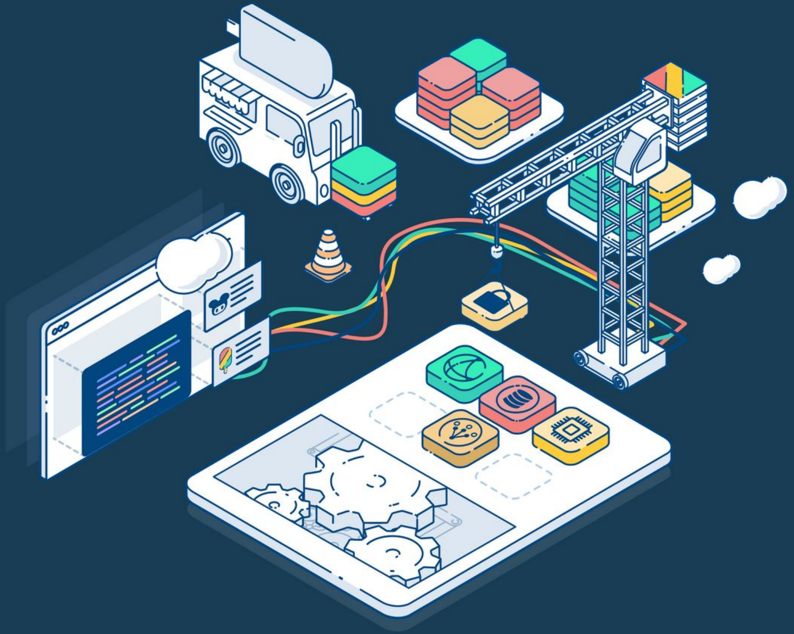
Kubernetes can only meet stateful workloads half way and I lack the expertise to manage a production configuration of Kafka, RabbitMQ, or Postgres on static infrastructure, let alone a Kubernetes cluster.

6:06 AM - 13 Feb 2018

External Resources

- Internal Resources:
  - Kubernetes API + client-go
- External API Resources:
  - AWS API + aws-sdk-go
  - Azure API + azure-sdk-for-go
  - GCP API + google.golang.org/api
- Other External Resources

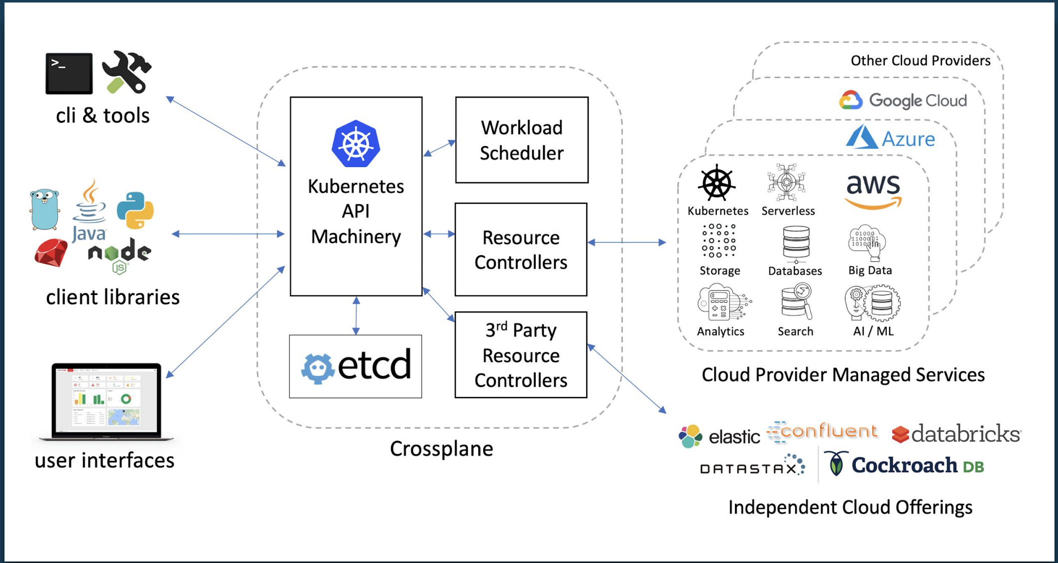




Crossplane

Crossplane

- Declarative API
- Portable Resource Abstractions
- Based on and inspired by Kubernetes
- Separation of Concerns
- Increased reusability

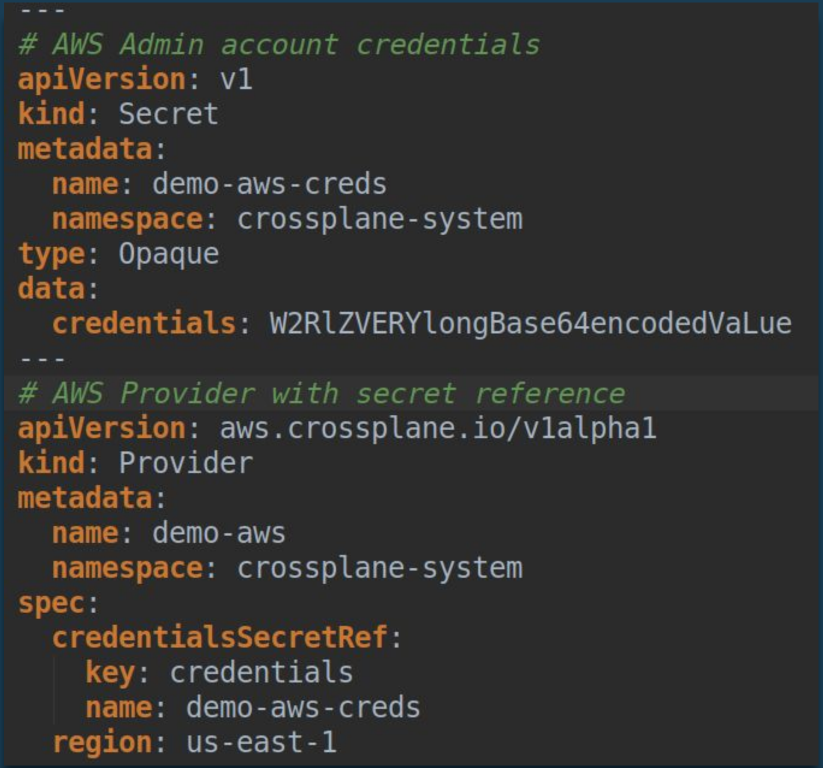


Managed Resources as CRD’s

- Cloud Provider
  - AWS, Azure, GCP (initially)
- Managed Resource
  - Relational Databases
  - Redis Memory Cache
  - EKS, AKS, GKE
  - Buckets
- Resource Classes
- Resource Claims (or Abstract Resources)
  - MySQLInstance, KubernetesCluster

Cloud Provider as a Resource

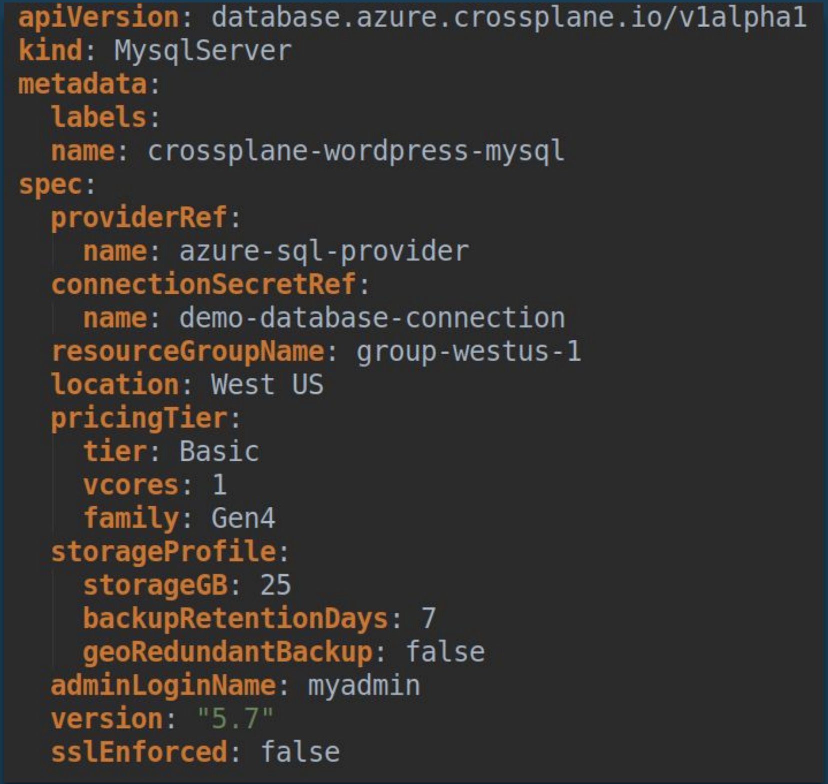
- Secret to store creds
- Provider with secret references
- Controller
  - validation



```yaml
---
# AWS Admin account credentials
apiVersion: v1
kind: Secret
metadata:
  name: demo-aws-creds
  namespace: crossplane-system
type: Opaque
data:
  credentials: W2RlZVERYlongBase64encodedVaLue
---
# AWS Provider with secret reference
apiVersion: aws.crossplane.io/v1alpha1
kind: Provider
metadata:
  name: demo-aws
  namespace: crossplane-system
spec:
  credentialsSecretRef:
    key: credentials
    name: demo-aws-creds
  region: us-east-1
```

Managed Service as a Resource

- Specific Resource
- Strongly Typed
- Provider Reference
- Controller
  - Provision
  - Connection Secret
  - Track Status



```yaml
apiVersion: database.azure.crossplane.io/v1alpha1
kind: MysqlServer
metadata:
  labels:
  name: crossplane-wordpress-mysql
spec:
  providerRef:
    name: azure-sql-provider
  connectionSecretRef:
    name: demo-database-connection
  resourceGroupName: group-westus-1
  location: West US
  pricingTier:
    tier: Basic
    vcores: 1
    family: Gen4
  storageProfile:
    storageGB: 25
    backupRetentionDays: 7
    geoRedundantBackup: false
  adminLoginName: myadmin
  version: "5.7"
  sslEnforced: false
```

Separation of concerns

Application Owner
ns: default

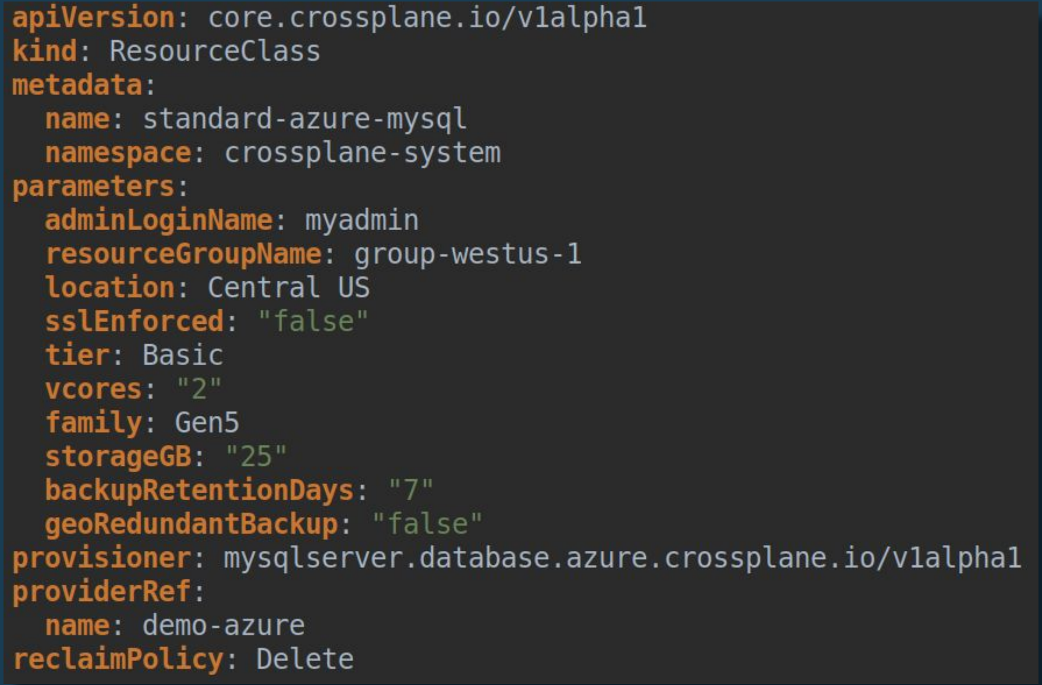
- Resource Claims
- Workloads

Administrator
ns: crossplane-system

- Resource Classes
- Providers
- Concrete Resources

Resource Classes

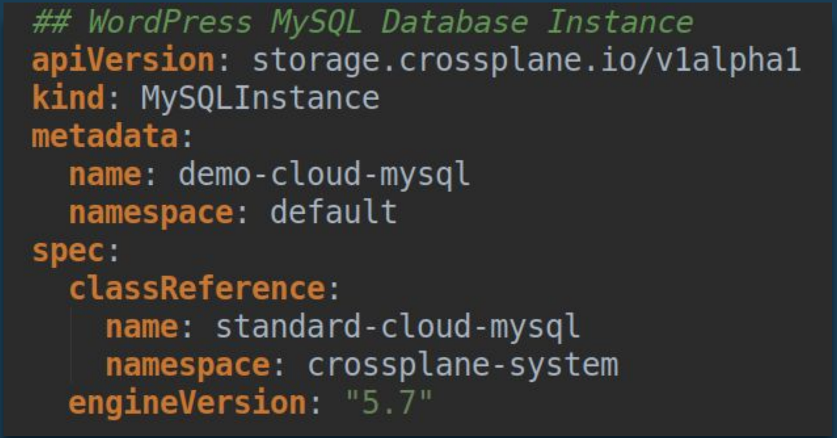
- Provisioner
- Provider Reference
- Properties
- Reclaim Policy



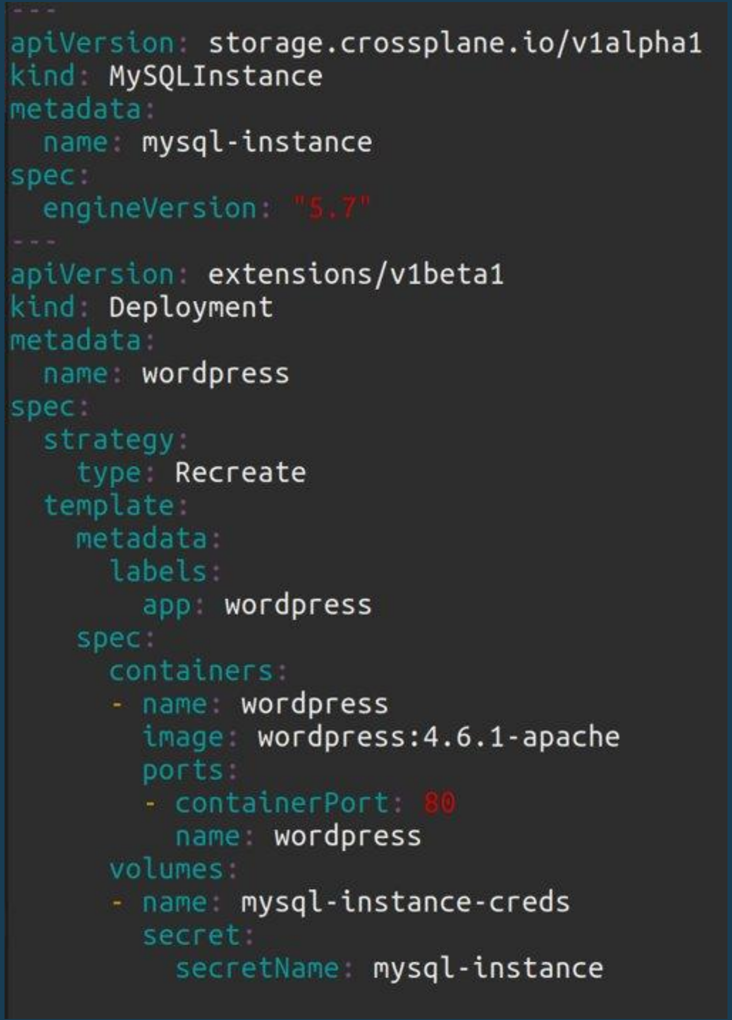
```yaml
apiVersion: core.crossplane.io/v1alpha1
kind: ResourceClass
metadata:
  name: standard-azure-mysql
  namespace: crossplane-system
parameters:
  adminLoginName: myadmin
  resourceGroupName: group-westus-1
  location: Central US
  sslEnforced: "false"
  tier: Basic
  vcores: "2"
  family: Gen5
  storageGB: "25"
  backupRetentionDays: "7"
  geoRedundantBackup: "false"
provisioner: mysqlserver.database.azure.crossplane.io/v1alpha1
providerRef:
  name: demo-azure
reclaimPolicy: Delete
```

Resource Claim - MySQLInstance

- Class Reference
- Additional Specifications
- Controller
    - Provision
    - Secret
    - Status



```yaml
## WordPress MySQL Database Instance
apiVersion: storage.crossplane.io/v1alpha1
kind: MySQLInstance
metadata:
  name: demo-cloud-mysql
  namespace: default
spec:
  classReference:
    name: standard-cloud-mysql
    namespace: crossplane-system
  engineVersion: "5.7"
```



```
---
apiVersion: storage.crossplane.io/v1alpha1
kind: MySQLInstance
metadata:
  name: mysql-instance
spec:
  engineVersion: "5.7"
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: wordpress
spec:
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
    spec:
      containers:
      - name: wordpress
        image: wordpress:4.6.1-apache
        ports:
        - containerPort: 80
          name: wordpress
      volumes:
      - name: mysql-instance-creds
        secret:
          secretName: mysql-instance
```

Application Portability

Workload

- Required Resources
  - Secrets
- Destination Cluster
  - Automatic Scheduling (Dynamic)
  - Designated (Assigned)
- Payload
  - Deployment
  - Service

Crossplane Vision

- Open cloud-computing platform
- Open control plane for open cloud
- More choices
- Extensible
- Inclusive

Crossplane Vision

- Open cloud-computing platform
- Open control plane for open cloud
- More choices
- Extensible
- Inclusive

Thank you

Q & A

github.com/crossplaneio/crossplane

https://crossplane.io/

Crossplane_io

References

- kubernetes/community/api-conventions
- https://coreos.com/operators/
- https://blog.couchbase.com/kubernetes-operators-game-changer/
- https://kubernetes.io/docs/concepts/workloads/controllers/garbage-collection/
- https://kubernetes.io/docs/concepts/extend-kubernetes/extend-cluster/
- https://kubernetes.io/docs/concepts/extend-kubernetes/api-extension/custom-resources/
- https://book.kubebuilder.io/basics/simple_controller.html
- https://github.com/crossplaneio/crossplane/blob/master/design/reconciler-patterns.md
- https://github.com/operator-framework/operator-sdk
- https://github.com/kubernetes-sigs/kubebuilder
- https://github.com/GoogleCloudPlatform/metacontroller
- https://github.com/kubernetes/kubernetes/issues/59850 [Propagation Policy: Foreground]