MIRANTIS | LABS

Tech Talks

# Crossplane

## Unifying Cloud Service Provisioning and Management

# Featured Presenters

**Martin Nirtl**

*Solutions Architect*

*Mirantis*

linkedin.com/in/martinnirtl

github.com/martinnirtl

**David Murphy**

*Principal Solutions Engineer*

*Upbound*

linkedin.com/in/murph83

# Agenda

- What is Crossplane?
- What problem is solved with Crossplane?
- What are the benefits of using Crossplane?
- How does Crossplane work? Core concepts?
- Example & Demo
- Q&A

“The delta between Kubernetes and a developer friendly PaaS is where the next layer of value is and where things tend to get opinionated - a requirement for reliable end to end workflows.”

*Quote from Kelsey Hightower*

# What is it?

- Open source, cloud native control plane framework
    - Backed by Upbound
- Built on top of Kubernetes
- Extensible by design
    - Providers for **everything**
- Orchestrate applications and infrastructure **anywhere**

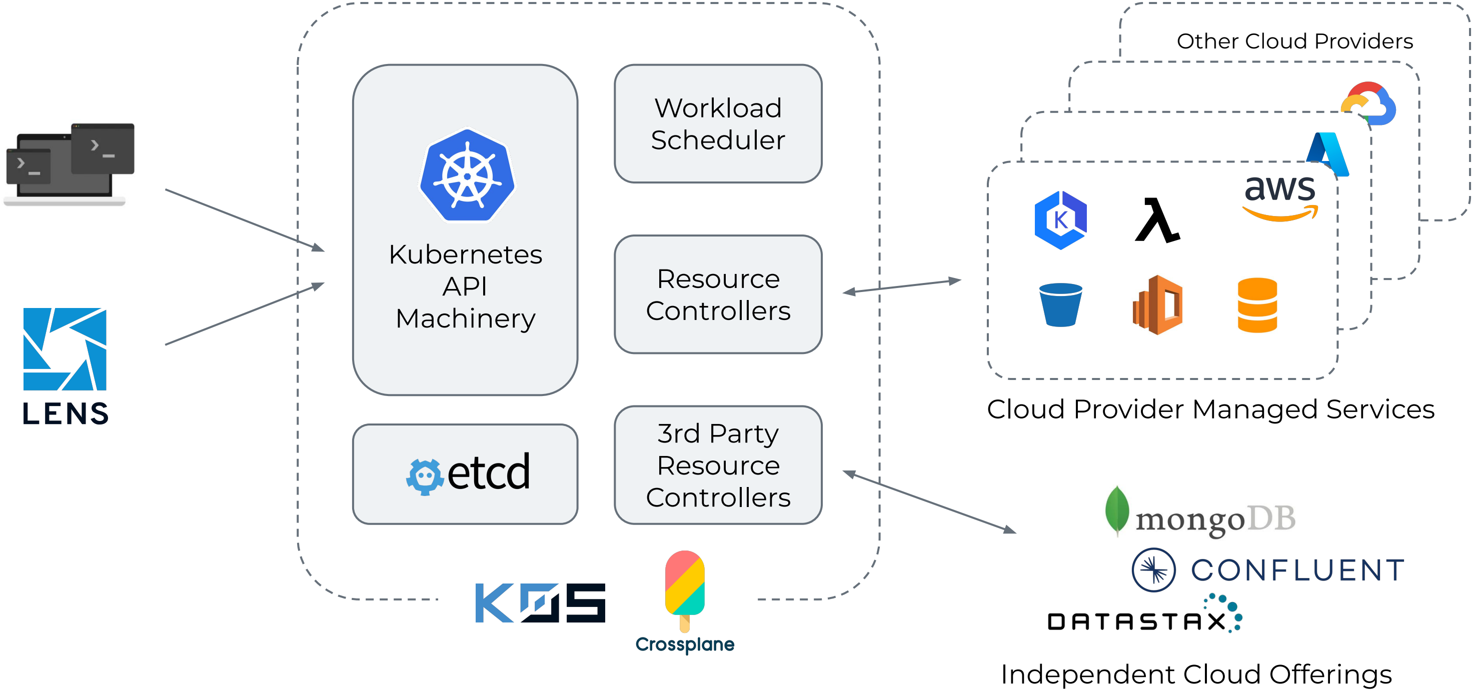
# What problem is solved with it?

- Manage APIs per cloud provider
- Many different tools, no real standardisation
- Operational efforts per cloud/provider
  - Experts needed
- No guided self-service and central governance

# What are the benefits of using it?

- A control plane to rule them all
  - Single or multi-tenancy
- Build your own Integrated Developer Platform (IDP) API
- Abstract resource templates tailored to your engineers' needs
- Kubernetes as the common ground

# How does it work?



# The Basics

## Managed Resources

# Managed Resources Example: AWS

Networking

Databases

Kubernetes Clusters

IAM

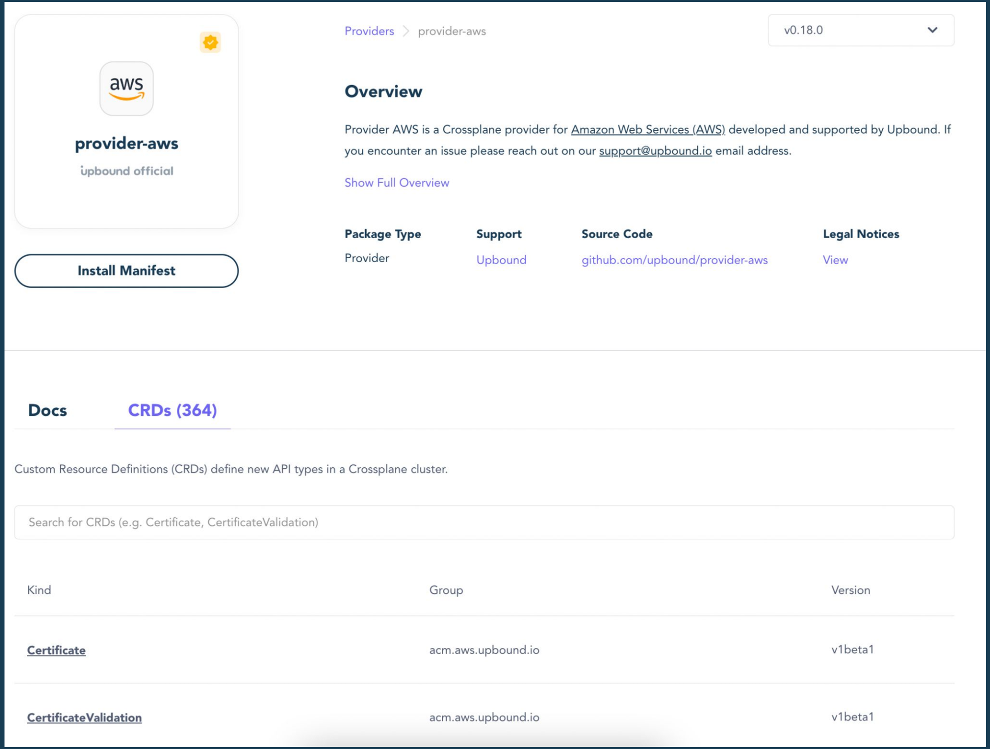
VMs

Message Queues

Caches

Certificates

...and much more...



provider-aws

upbound official

### Overview

Provider AWS is a Crossplane provider for Amazon Web Services (AWS) developed and supported by Upbound. If you encounter an issue please reach out on our support@upbound.io email address.

| Package Type | Support | Source Code | Legal Notices |
|---|---|---|---|
| Provider | Upbound | github.com/upbound/provider-aws | View |

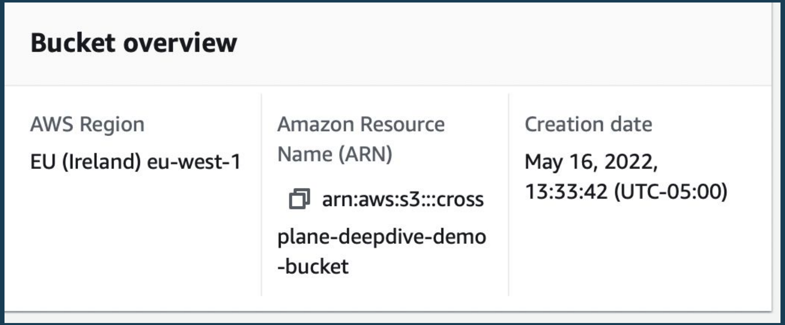
Custom Resource Definitions (CRDs) define new API types in a Crossplane cluster.

| Kind | Group | Version |
|---|---|---|
| Certificate | acm.aws.upbound.io | v1beta1 |
| CertificateValidation | acm.aws.upbound.io | v1beta1 |

https://marketplace.upbound.io/providers/upbound/provider-aws/latest

# Managed Resources

```yaml
apiVersion: s3.aws.crossplane.io/v1beta1
kind: Bucket
metadata:
  name: crossplane-deepdive-demo-bucket
spec:
  forProvider:
    acl: private
    locationConstraint: eu-west-1
    paymentConfiguration:
      payer: BucketOwner
    versioningConfiguration:
      status: Enabled
    tagging:
      tagSet:
      - key: Name
        value: CrossplaneDeepDiveDemoBucket
```

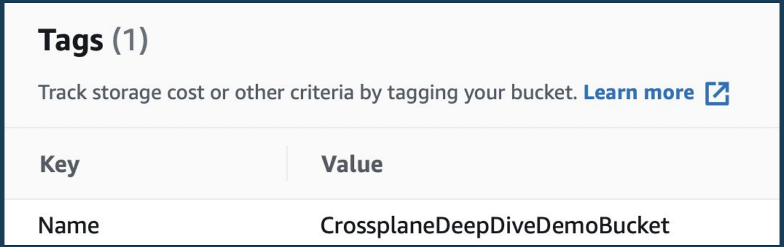


**Bucket overview**

AWS Region
EU (Ireland) eu-west-1

Amazon Resource Name (ARN)
arn:aws:s3:::crossplane-deepdive-demo-bucket

Creation date
May 16, 2022, 13:33:42 (UTC-05:00)



**Tags** (1)

Track storage cost or other criteria by tagging your bucket. **Learn more**

| Key | Value |
|---|---|
| Name | CrossplaneDeepDiveDemoBucket |

# Managed Resources

```
Status:
  At Provider:
    Arn:  arn:aws:s3:::crossplane-deepdive-demo-bucket

Events:
  Type    Reason                   Age   From
Message
  ----    ------                   ----  ----
-------
  Normal  CreatedExternalResource  6m8s  managed/bucket.s3.aws.crossplane.io
Successfully requested creation of external resource
```

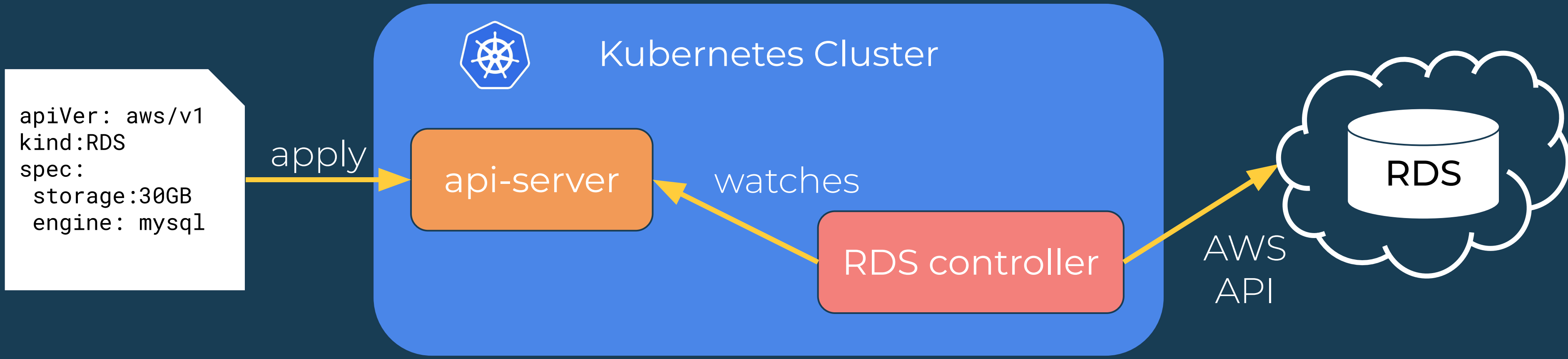
Status contains values returned from the remote API and the condition of the resources.

Managed Resources Generate K8s Events

Crossplane

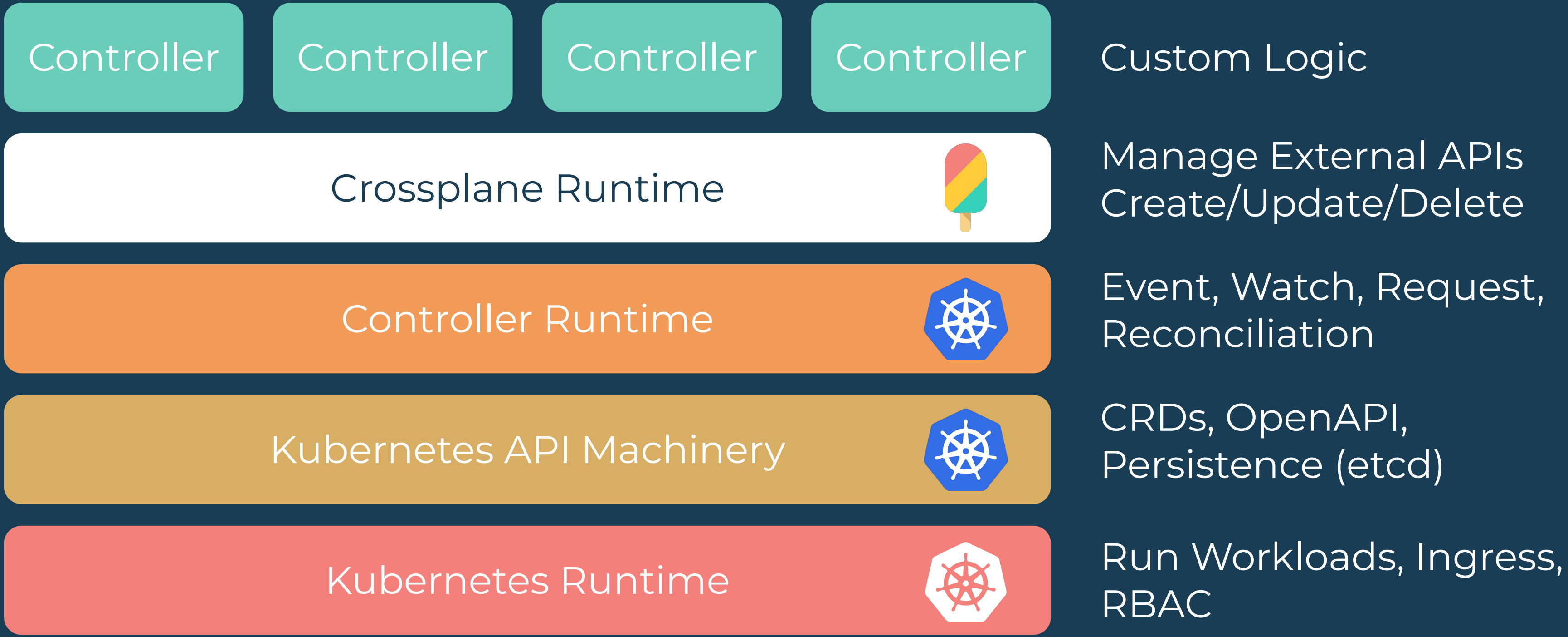
# Managed Resource Reconciliation

- Controllers reconcile Managed Resources with cloud provider and on-prem APIs (e.g., GCP, AWS, or any API)



```
apiVer: aws/v1
kind:RDS
spec:
 storage:30GB
 engine: mysql
```

# Control Plane Internal Stack



Controller | Controller | Controller | Controller — Custom Logic

Crossplane Runtime — Manage External APIs Create/Update/Delete

Controller Runtime — Event, Watch, Request, Reconciliation

Kubernetes API Machinery — CRDs, OpenAPI, Persistence (etcd)

Kubernetes Runtime — Run Workloads, Ingress, RBAC

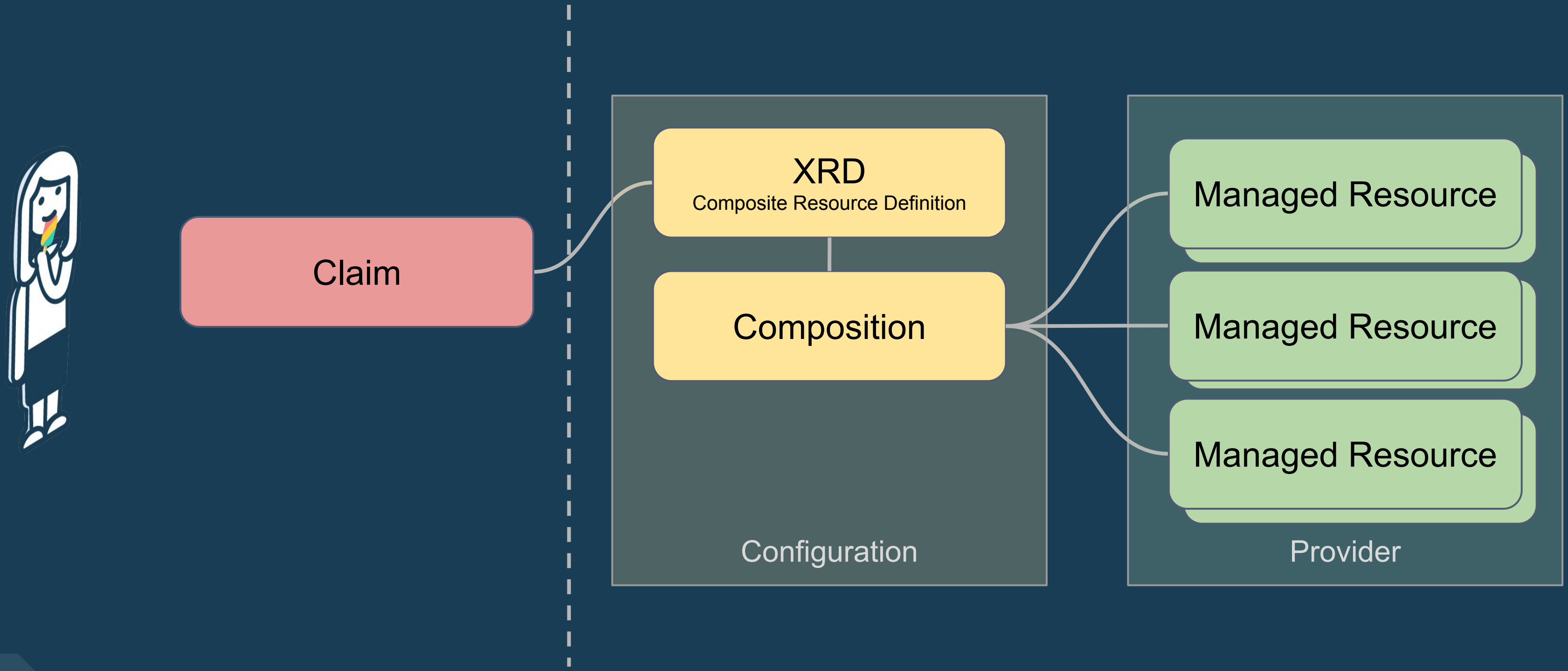
# Building Your Control Plane

Composition

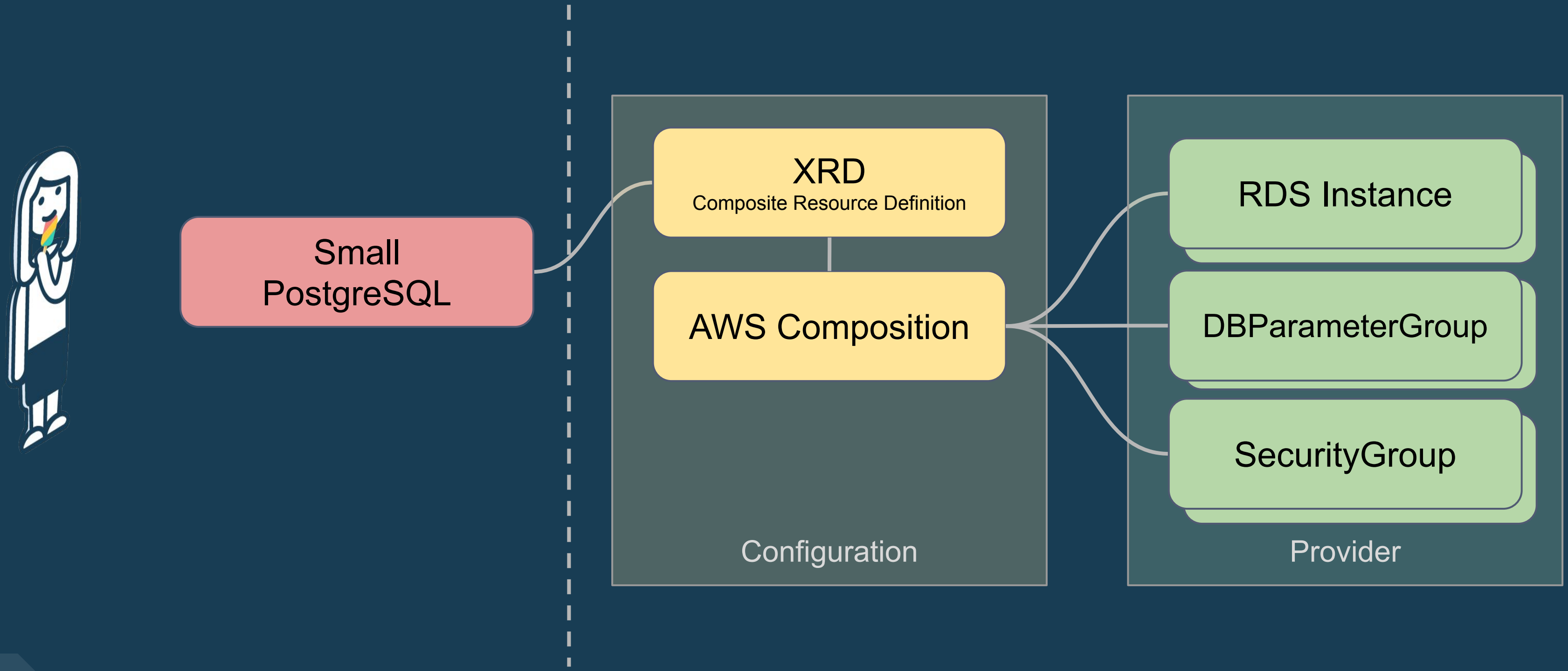
Crossplane

# Build your own Platform API

- Assemble granular resources. E.g. from multiple clouds.
- Expose as higher level self-service API for your app teams
  - **Compose** GKE, NodePool, Network, Subnetwork
  - **Offer** as a single Cluster resource (API) with limited config for developers to self-service
- Hide infrastructure complexity and include policy guardrails
- All with K8s API - compatible with kubectl, GitOps, etc.
- **No code** required, it's all **declarative**


Claim
XRD
Composite Resource Definition
Composition
Configuration
Managed Resource
Managed Resource
Managed Resource
Provider

Crossplane


Small PostgreSQL
XRD
Composite Resource Definition
AWS Composition
Configuration
RDS Instance
DBParameterGroup
SecurityGroup
Provider

Crossplane

# Composite Resources

```
apiVersion: apiextensions.crossplane.io/v1
kind: CompositeResourceDefinition
metadata:
  name: xpostgresqlinstances.database.example.org
spec:
  group: database.example.org
  names:
    kind: XPostgreSQLInstance
    plural: xpostgresqlinstances
  versions:
  - name: v1alpha1
    served: true
    referenceable: true
    schema:
      openAPIV3Schema:
        type: object
        properties:
```

First we create Composite Resource Definition (XRD) to declare our custom platform API

Custom API Group

Standard openAPIV3 Schema

# Compositions

```yaml
apiVersion: apiextensions.crossplane.io/v1
kind: Composition
metadata:
  name: xpostgresqlinstances.aws.database.example.org
spec:
  writeConnectionSecretsToNamespace: crossplane-system
  compositeTypeRef:
    apiVersion: database.example.org/v1alpha1
    kind: XPostgreSQLInstance
  resources:
    - name: parametergroup
      base:
        apiVersion: rds.aws.crossplane.io/v1alpha1
        kind: DBParameterGroup
```

Then we define Composition which implements XRD

XRD reference

List of Managed Resources to Compose

# Patches

Patches enable propagation of data from Composite Resource (XR) down to composed Managed Resources (MR)

```
patches:
  - fromFieldPath: "spec.nodes.count"
    toFieldPath: "spec.forProvider.scalingConfig.desiredSize"
  - fromFieldPath: "spec.nodes.size"
    toFieldPath: "spec.forProvider.instanceTypes[0]"
    transforms:
      - type: map
        map:
          small: t3.small
          medium: t3.medium
          large: t3.large
```

Copy of value from XR spec down to MR spec

Map transform to manipulate the config data

# Extending Crossplane

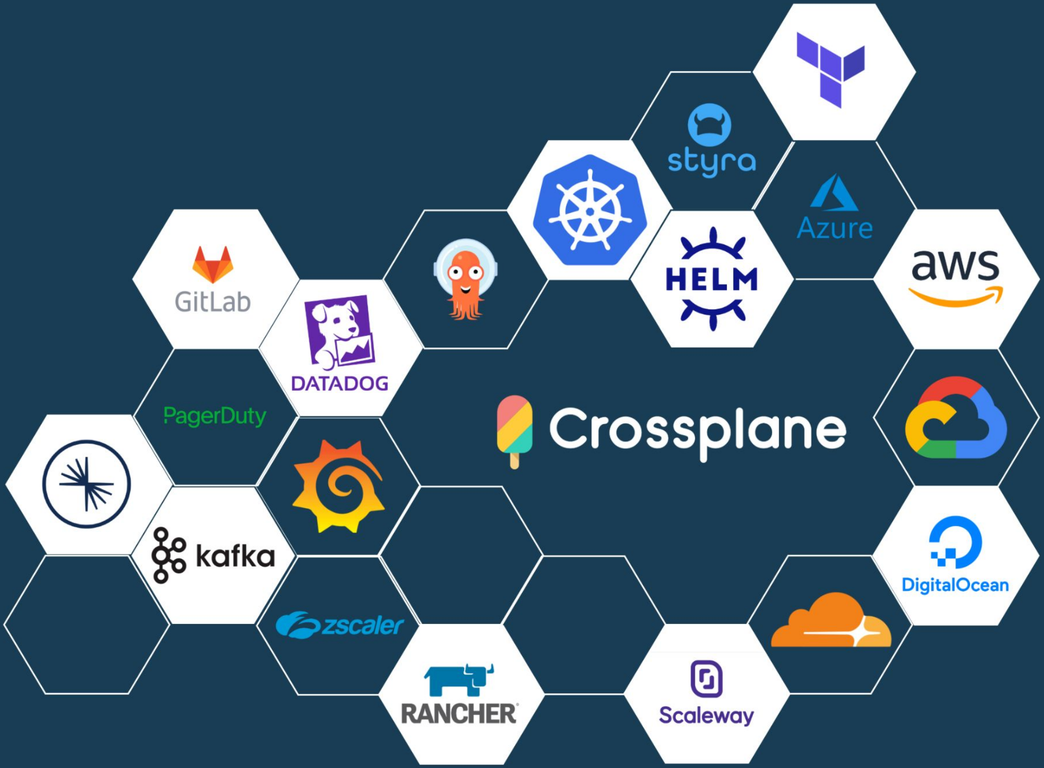
Providers & Configurations

Crossplane

# Current Extension Points

- Crossplane is a highly extensible framework
- **Providers**
  - You can build a provider to manage **anything** with an API
  - CRUD operations for cloud resources, on-prem services, etc.
- **Configurations**
  - Compose resources from providers
  - Define your control plane's declarative APIs and abstractions
  - These are what your devs see - it's how they consume the offerings of your control plane
- Both are Crossplane packages / opinionated OCI Images.

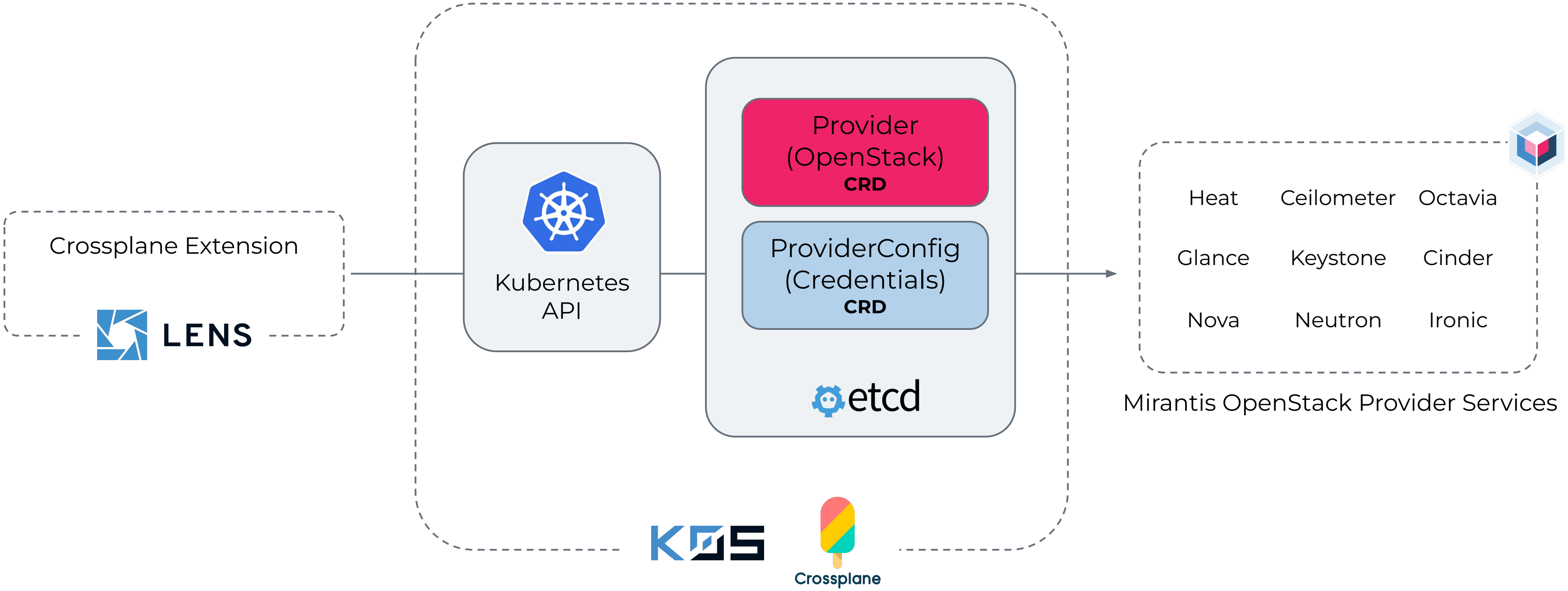
# Crossplane Provider Ecosystem



Crossplane

# Demo

# What you will see in a minute



Crossplane Extension

LENS

Kubernetes API

Provider (OpenStack) **CRD**

ProviderConfig (Credentials) **CRD**

etcd

K0S

Crossplane

Heat Ceilometer Octavia

Glance Keystone Cinder

Nova Neutron Ironic

Mirantis OpenStack Provider Services

# Summary

- Single cloud control plane
- Build your own extensible and opinionated IDP API
- Standardise processes and workflows
- Abstract resource templates tailored to your engineers' needs

# Links

- k0s
  [https://github.com/k0sproject/k0s](https://github.com/k0sproject/k0s)
- Crossplane
  [https://github.com/crossplane/crossplane](https://github.com/crossplane/crossplane)
- Lens Crossplane Extension
  [https://github.com/mirantis-field/lens-crossplane-extension](https://github.com/mirantis-field/lens-crossplane-extension)

MIRANTIS

# Q&A

- k0s
  https://github.com/k0sproject/k0s
- Crossplane
  https://github.com/crossplane/crossplane
- Lens Crossplane Extension
  https://github.com/mirantis-field/lens-crossplane-extension

# Join us for Upcoming Webinars

MIRANTIS | LABS
TECH TALK
July 11, 2023
Revolutionizing Kubernetes: Harness the Power of k0smotron for Scalable Control Planes

## Revolutionizing Kubernetes: Harness the Power of k0smotron for Scalable Control Planes

- July 11, 2023 at 1 pm PST
- Featured presenters: Julian Hennig, Senior Solution Architect, Mirantis

## Edge and Far Edge Clouds with k0s

- July 25, 2023 at 10 am GMT / 11 am CEST
- Featured presenters: Roman Kuzmin, DevOps Engineer, Mirantis