

# Crossplane

Introduction & Deep Dive

<https://crossplane.io>

*Christopher Haar, DKB*  
*Jared Watts, Upbound*





# What is Crossplane?

- **Framework** for building cloud native **control planes**
  - No need to write any code
- Cloud providers have been managing their infrastructure with control planes for years
  - Crossplane helps you build your own - with your own **opinions**
- Extensible backend to manage **any infrastructure** in **any environment**
- Configurable frontend to expose **declarative APIs** (abstractions) for developer **self-service**



# CNCF Project for the Community

- Crossplane is a neutral place for vendors and individuals to come together in enabling control planes
- Launched in Dec 2018 by creators of CNCF graduated Rook project
  - Accepted into Sandbox in June 2020
  - First major “stable” milestone [v1.0 released](#) in Dec 2020
  - Moved to Incubation September 2021
  - [v1.12](#) releasing next week
  - Progressing towards [Graduation](#) - we need your help [adopters](#)!

# Project and Community Stats



**8,000+**

Stars

**70,000+**

Contributions

**1,500+**

Contributors



**6,500+**

Followers



**8,500+**

Members



**39M+**

Pulls

# The Basics

Managed Resources

# Managed Resources Example: AWS

Networking

Databases

Kubernetes Clusters

IAM

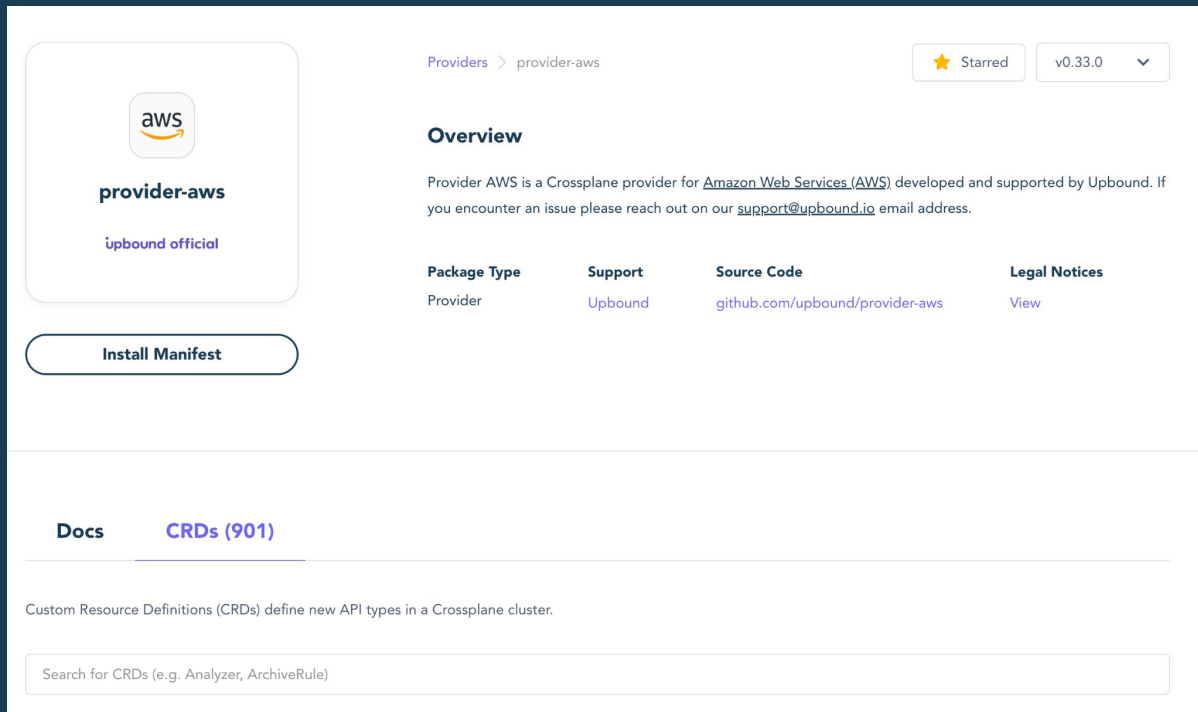
VMs

Message Queues

Caches

Certificates

...and much more...



The screenshot shows the Upbound Marketplace page for the **provider-aws**. On the left, there is a card with the AWS logo, the text **provider-aws**, *upbound official*, and an **Install Manifest** button. The main content area includes a breadcrumb [Providers](#) > **provider-aws**, a **Starred** button, and a version dropdown set to **v0.33.0**. Below this is an **Overview** section stating that Provider AWS is a Crossplane provider for Amazon Web Services (AWS), developed and supported by Upbound, with contact information [support@upbound.io](mailto:support@upbound.io). A table below the overview lists details: **Package Type** is Provider, **Support** is Upbound, **Source Code** is [github.com/upbound/provider-aws](https://github.com/upbound/provider-aws), and **Legal Notices** has a [View](#) link. At the bottom, there is a **Docs** section with a link to **CRDs (901)** and a search bar for Custom Resource Definitions (CRDs) with the placeholder text "Search for CRDs (e.g. Analyzer, ArchiveRule)".

Package Type	Support	Source Code	Legal Notices
Provider	Upbound	<a href="https://github.com/upbound/provider-aws">github.com/upbound/provider-aws</a>	<a href="#">View</a>


# Managed Resources

```
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	Amazon Resource Name (ARN)	Creation date
EU (Ireland) eu-west-1	 arn:aws:s3:::crossplane-deepdive-demo-bucket	April 21, 2023, 15:00:23 (UTC+01:00)

## Tags (1)

Track storage cost or other criteria by tagging your bucket. [Learn more](#) 

Key	Value
Name	CrossplaneDeepDiveDemoBucket

# Managed Resources

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

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

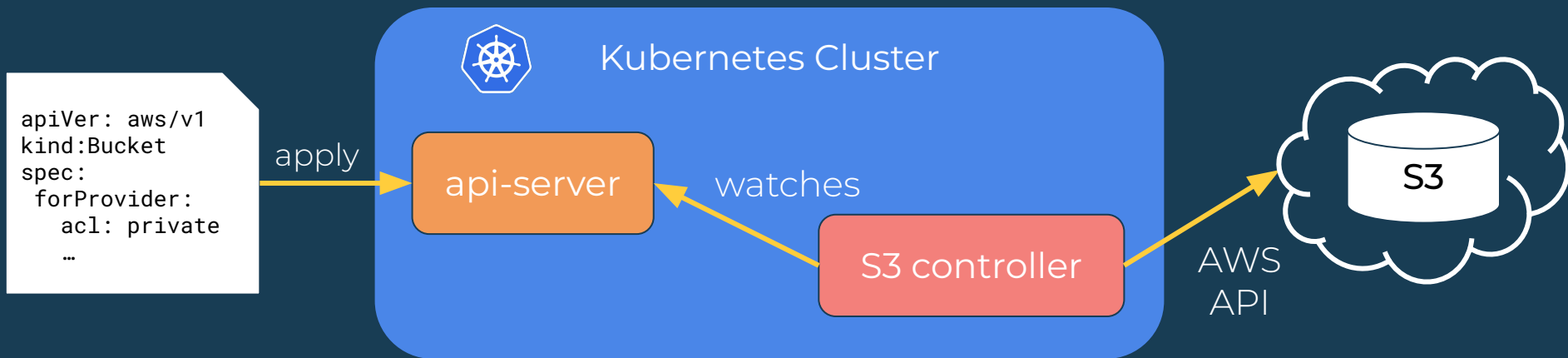
```
Events:
  Type      Age      From                                Message
  ----      -
  Normal    6m8s    bucket.s3.aws.crossplane.io    Successfully created external resource
```

Managed Resources  
Generate K8s Events



# Managed Resource Reconciliation

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



# 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

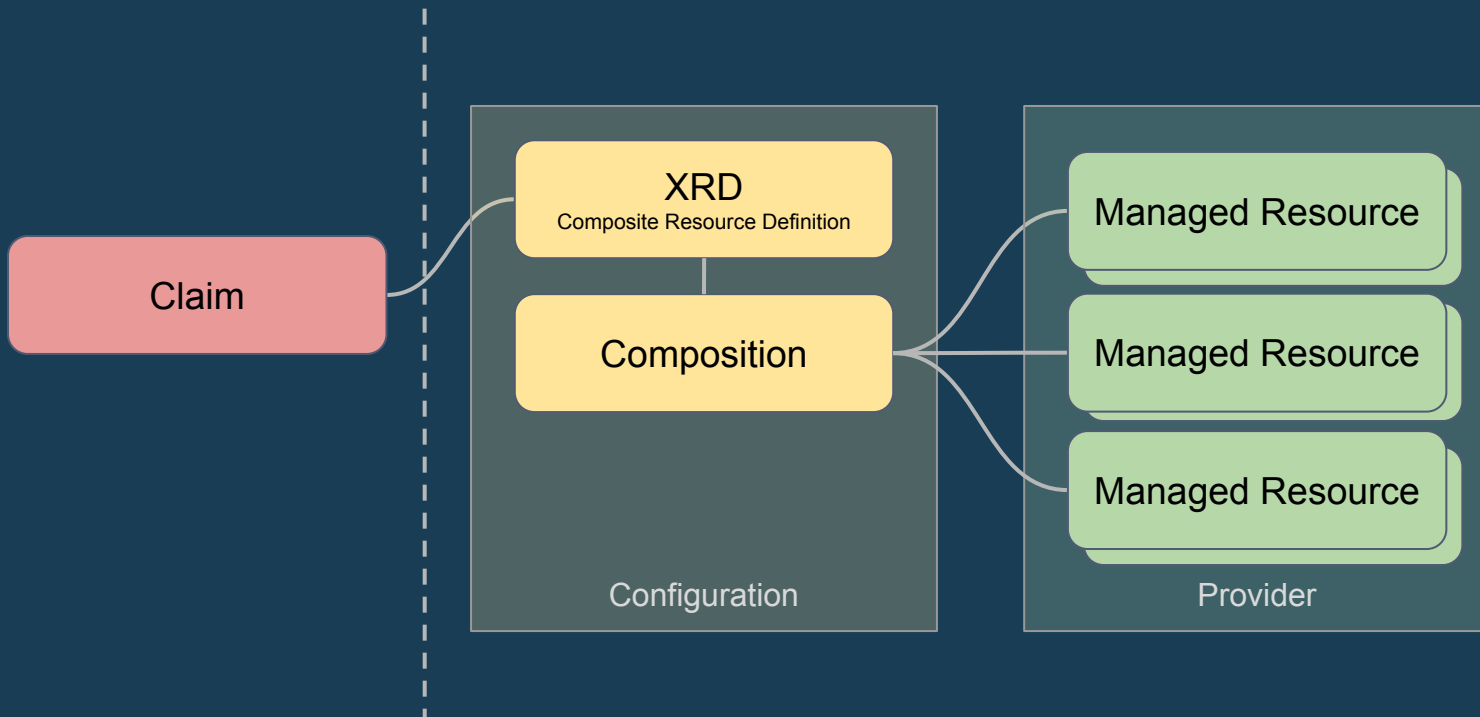
# Building Your Control Plane

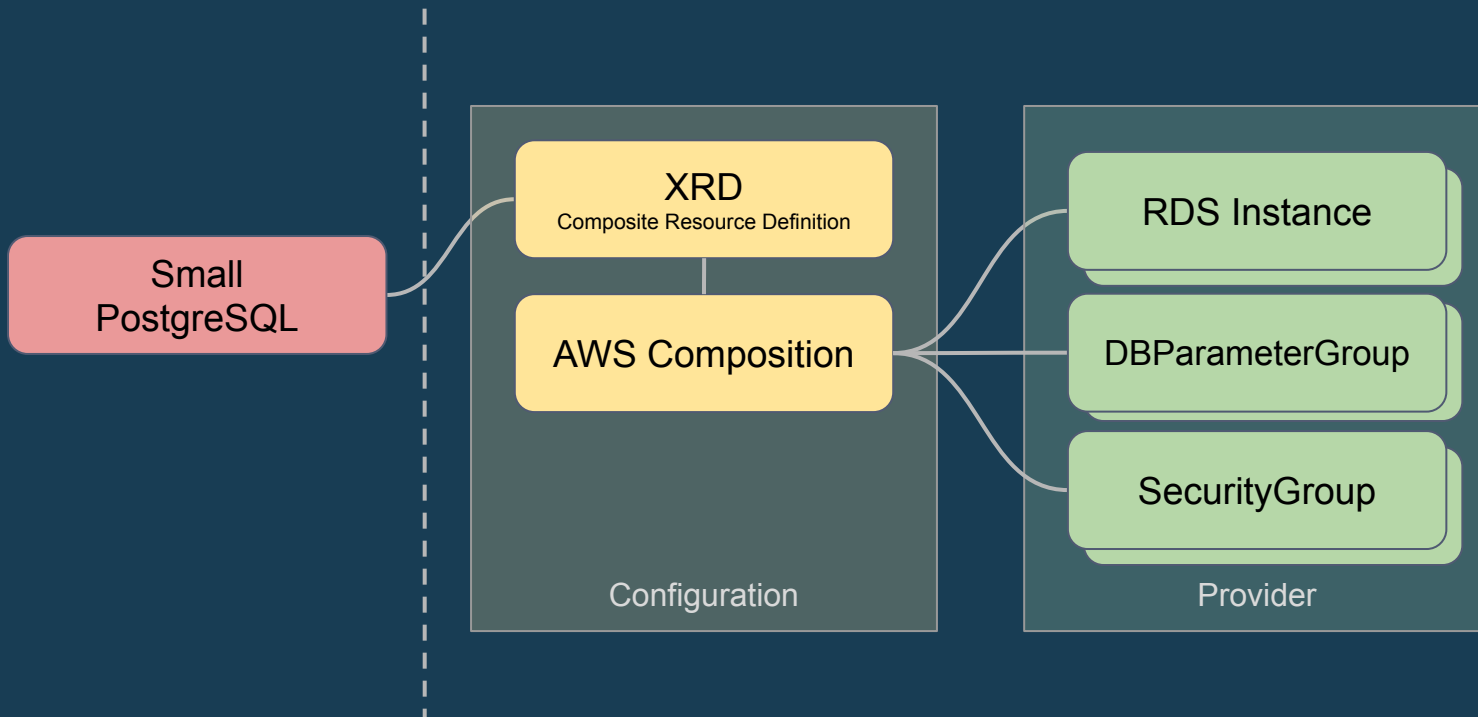
Composition



# 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**





# Composite Resources

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

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

Custom API Group

Standard openAPIV3 Schema

# Compositions

```
apiVersion: apiextensions.crossplane.io/v1
kind: Composition
metadata:
  name: xpostgresinstances.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 a 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



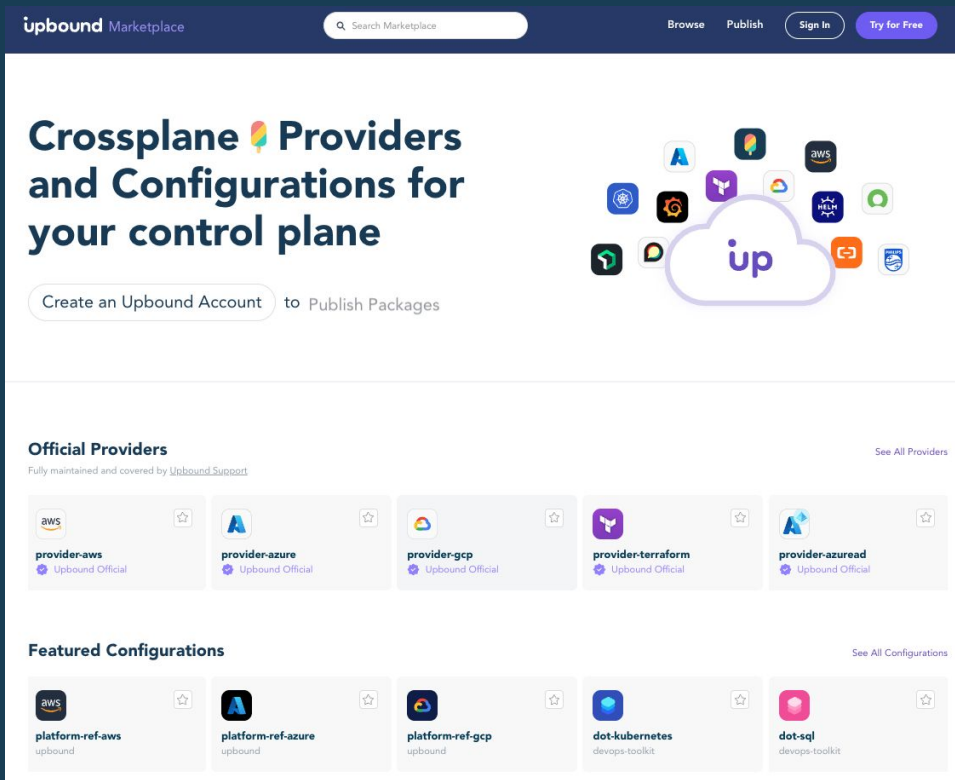
# Current Extension Points

- 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



# Marketplace for all Extensions



The screenshot shows the Upbound Marketplace homepage. At the top, there's a navigation bar with the 'upbound Marketplace' logo, a search bar, and links for 'Browse', 'Publish', 'Sign In', and 'Try for Free'. The main heading reads 'Crossplane Providers and Configurations for your control plane'. Below this is a button that says 'Create an Upbound Account to Publish Packages'. To the right is a cloud icon with 'up' inside, surrounded by various provider logos like AWS, Azure, GCP, and Kubernetes. The page is divided into two main sections: 'Official Providers' and 'Featured Configurations'. The 'Official Providers' section lists five providers: provider-aws, provider-azure, provider-gcp, provider-terraform, and provider-azuread, all marked as 'Upbound Official'. The 'Featured Configurations' section lists five configurations: platform-ref-aws, platform-ref-azure, platform-ref-gcp, dot-kubernetes, and dot-sql, with their respective maintainers.

**Crossplane Providers and Configurations for your control plane**

Create an Upbound Account to Publish Packages

**Official Providers**  
Fully maintained and covered by [Upbound Support](#) [See All Providers](#)

- provider-aws (Upbound Official)
- provider-azure (Upbound Official)
- provider-gcp (Upbound Official)
- provider-terraform (Upbound Official)
- provider-azuread (Upbound Official)

**Featured Configurations** [See All Configurations](#)

- platform-ref-aws (upbound)
- platform-ref-azure (upbound)
- platform-ref-gcp (upbound)
- dot-kubernetes (devops-toolkit)
- dot-sql (devops-toolkit)

Discover and share Crossplane extensions

Open to everyone

<https://marketplace.upbound.io>

# Composition Functions

```
apiVersion: database.example.org/v1alpha1
kind: AcmeCoDatabase
metadata:
  name: example-db
spec:
  parameters:
    storageGB: 20
  compositionRef:
    name: example
```

```
apiVersion: apiextensions.crossplane.io/v1
kind: Composition
metadata:
  name: example
spec:
  compositeTypeRef:
    apiVersion: database.example.org/v1alpha1
    kind: AcmeCoDatabase
  resources:
    - name: cloudsqlinstance
      base:
        apiVersion: database.gcp.crossplane.io/v1beta1
        kind: CloudSQLInstance
        spec:
          forProvider:
            databaseVersion: POSTGRES_9_6
            region: us-central1
            settings:
              tier: db-custom-1-3840
              dataDiskType: PD_SSD
      patches:
        - type: FromCompositeFieldPath
          fromFieldPath: spec.parameters.storageGB
          toFieldPath: spec.forProvider.settings.dataDiskSizeMb
          transforms:
            - type: math
              math:
                multiply: 1024
```





# Limitations of Composition

- No iteration. No conditionals. No templates.
- No advanced logic other than simple transforms
- List of resources is static
- No ability to call external APIs to get values
- ...
- It is not a programming language
- We didn't want to grow a DSL expressed in YAML
  - Need to reinvent a lot of wheels - testing, linting, etc
  - Infra DSLs tend to grow organically, but not cohesively
  - We're not language designers

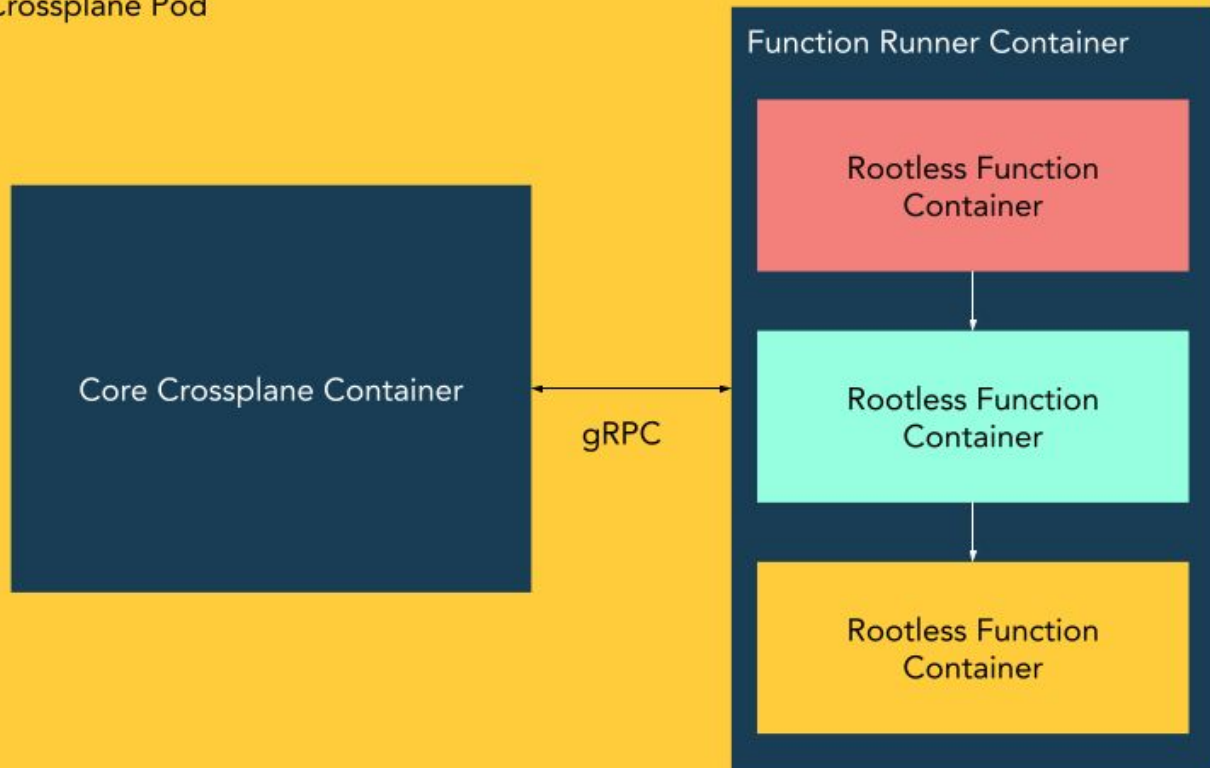


# Composition Functions

- Released as alpha in [v1.11.0](#)
- Run a pipeline of simple containerized functions
- Any logic you want to include for your use case
- No language constraints or limitations
  - Any container that can receive and return a FunctionIO object
- Additive to the existing composition mechanisms
- ...
- All the things you couldn't do with Composition!
- Sweet spot between “no code” and building an entire controller
  - Focus on your platform's unique needs - Crossplane still does the heavy lifting of CRUD-ing resources, finalizers, owner refs, etc

```
apiVersion: apiextensions.crossplane.io/v2alpha1
kind: Composition
metadata:
  name: example
spec:
  compositeTypeRef:
    apiVersion: database.example.org/v1alpha1
    kind: XPostgreSQLInstance
  functions:
  - name: my-cool-function
    type: Container
    container:
      image: xkpg.io/my-cool-function:0.1.0
```

## Crossplane Pod



# Composition Functions DEMO

# Observe Only Resources



## Observe Only Resources to Alpha

- providers will **observe** existing resources **without taking ownership** of it.
- Enable by passing the `--enable-management-policies` to the ControllerConfig for providers
- **CRD ValidationRules** - create managed resources without providing all required fields

Status

[alpha](#)

Issue

[#1722](#)

Design

[#3531](#)



# Observe Only Resources to Alpha

## API Changes:

- new **spec.managementPolicy**
  - FullControl
  - OrphanOnDelete
  - ObserveOnly
- full state under the **status.atProvider**
- deprecation of **spec.deletionPolicy**

**Status**  
alpha

**Issue**  
[#1722](#)

**Design**  
[#3531](#)

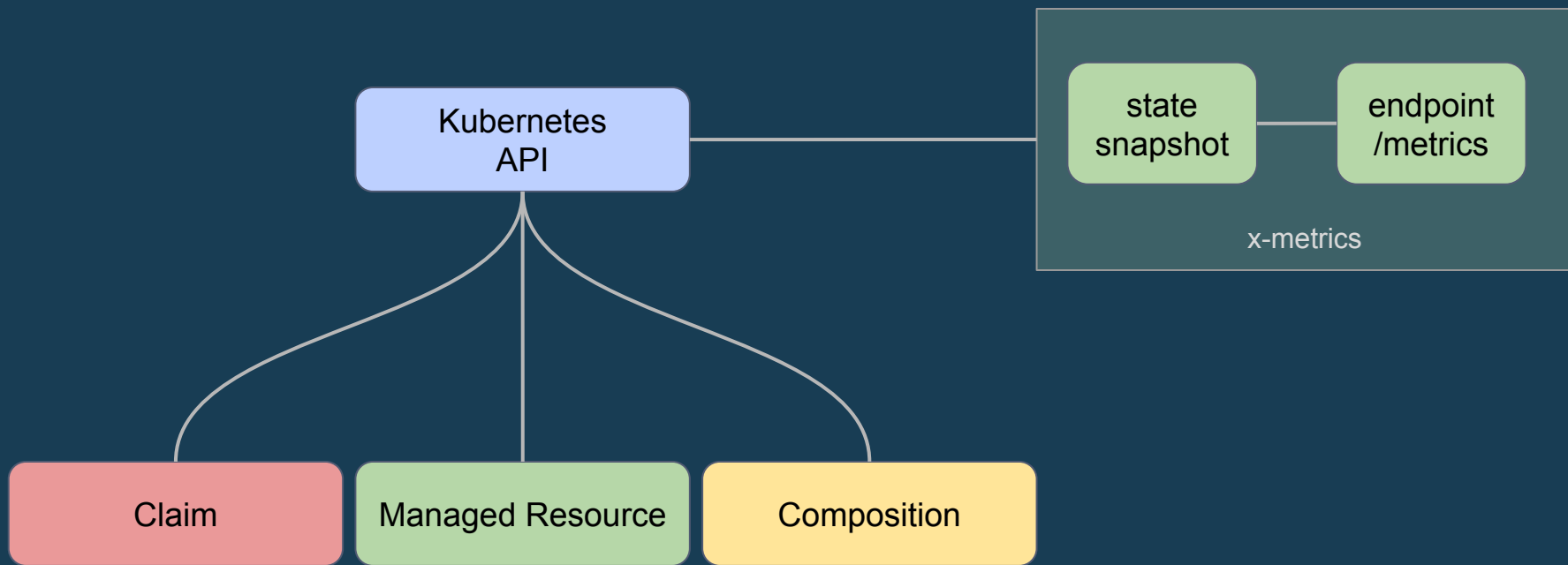


```
apiVersion: ec2.aws.crossplane.io/v1beta1
kind: VPC
metadata:
  annotations:
    crossplane.io/external-name: vpc-12345678
  name: observe-vpc
spec:
  managementPolicy: ObserveOnly
  forProvider:
    region: us-east-1
```

```
apiVersion: ec2.aws.crossplane.io/v1beta1
kind: VPC
metadata:
  annotations:
    crossplane.io/external-name: vpc-12345678
  name: observe-vpc
spec:
  managementPolicy: ObserveOnly
  forProvider:
    region: us-east-1
status:
  atProvider:
    cidrBlock: 172.16.0.0/16
    enableDnsHostNames: false
    enableDnsSupport: true
    instanceTenancy: default
    region: us-east-1
    tags:
      - key: managed-by
        value: crossplane
  conditions:
    - lastTransitionTime: "2023-04-21T14:00:00Z"
      reason: ReconcileSuccess
      status: "True"
      type: Synced
```

# Observe Only Resources DEMO

# Metrics



```
apiVersion: metrics.crossplane.io/v1
kind: ClusterMetric
metadata:
  name: clustermetric-sample
spec:
  matchName: "vpcs.ec2.aws.upbound.io"
```

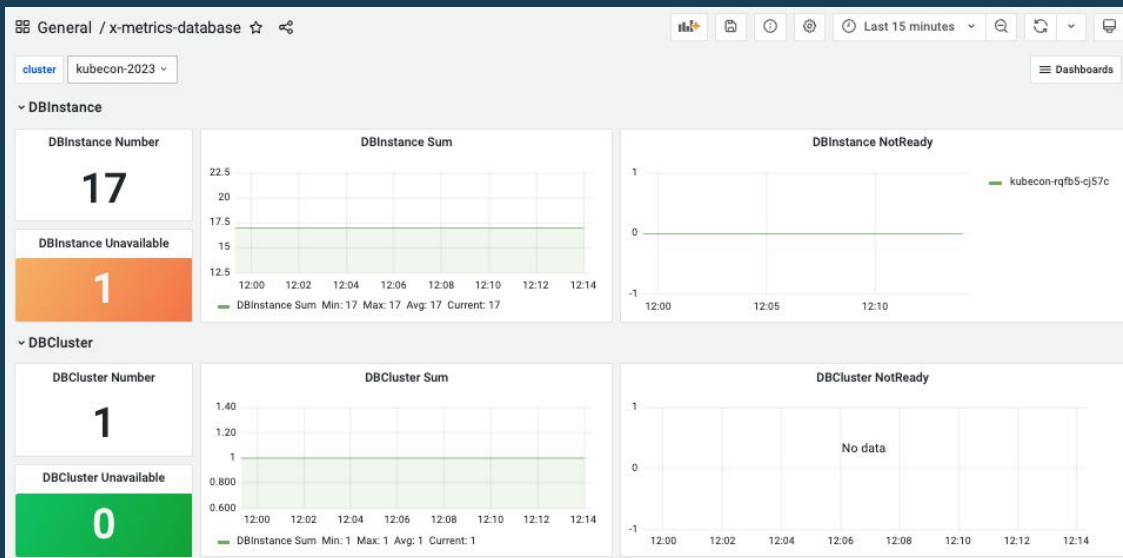
Metrics browser > rds\_

Query type Range

+ Add query ⌚ Query

- 🔗 rds\_aws\_crossplane\_io\_DBInstance\_v1alpha1
- 🔗 rds\_aws\_crossplane\_io\_DBInstance\_v1alpha1\_created
- 🔗 rds\_aws\_crossplane\_io\_DBInstance\_v1alpha1\_info
- 🔗 rds\_aws\_crossplane\_io\_DBInstance\_v1alpha1\_labels
- 🔗 rds\_aws\_crossplane\_io\_DBInstance\_v1alpha1\_ready
- 🔗 rds\_aws\_crossplane\_io\_DBInstance\_v1alpha1\_ready\_ti...
- 🔗 rds\_aws\_crossplane\_io\_DBInstance\_v1alpha1\_synced
- 🔗 rds\_aws\_crossplane\_io\_DBInstance\_v1alpha1\_synced\_t...
- 🔗 rds\_aws\_crossplane\_io\_DBParameterGroup\_v1alpha1
- 🔗 rds\_aws\_crossplane\_io\_DBParameterGroup\_v1alpha1\_cr...
- 🔗 rds\_aws\_crossplane\_io\_DBParameterGroup\_v1alpha1\_in...
- 🔗 rds\_aws\_crossplane\_io\_DBParameterGroup\_v1alpha1\_la...

rds\_aws\_crossplane\_io\_DBInstance\_v1alpha1 X  
1 : gauge  
A metrics series for each object



# Metrics DEMO



# Community is everything



## Get Involved

- Website: <https://crossplane.io/>
- Docs: <https://crossplane.io/docs>
- GitHub: <https://github.com/crossplane/crossplane>
- Slack: <https://slack.crossplane.io/>
- Blog: <https://blog.crossplane.io/>
- Twitter: [https://twitter.com/crossplane\\_io](https://twitter.com/crossplane_io)
- Youtube: [Crossplane Youtube](#)



## Calling all Crossplane Adopters!

🚧 We'd love to hear about your adoption of Crossplane, please share your story in [ADOPTERS.md](#) in the crossplane/crossplane repo 🚧

