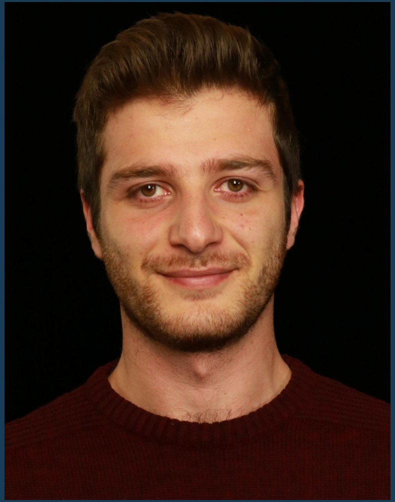
Crossplane 101

The cloud native control plane framework

Crossplane

Introduction

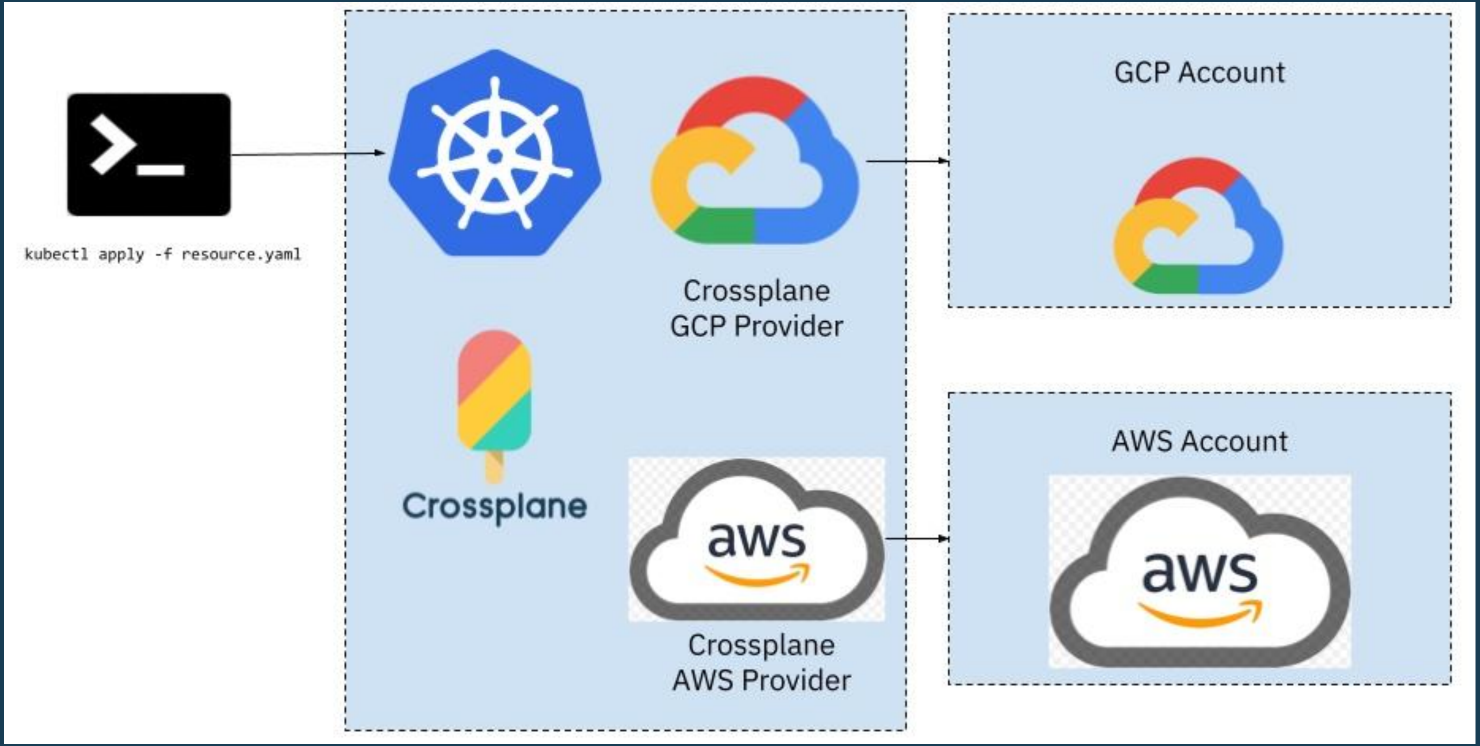


Muvaffak Onus (@muvaf)
Staff Software Engineer at *Upbound*
Maintainer at *Crossplane*

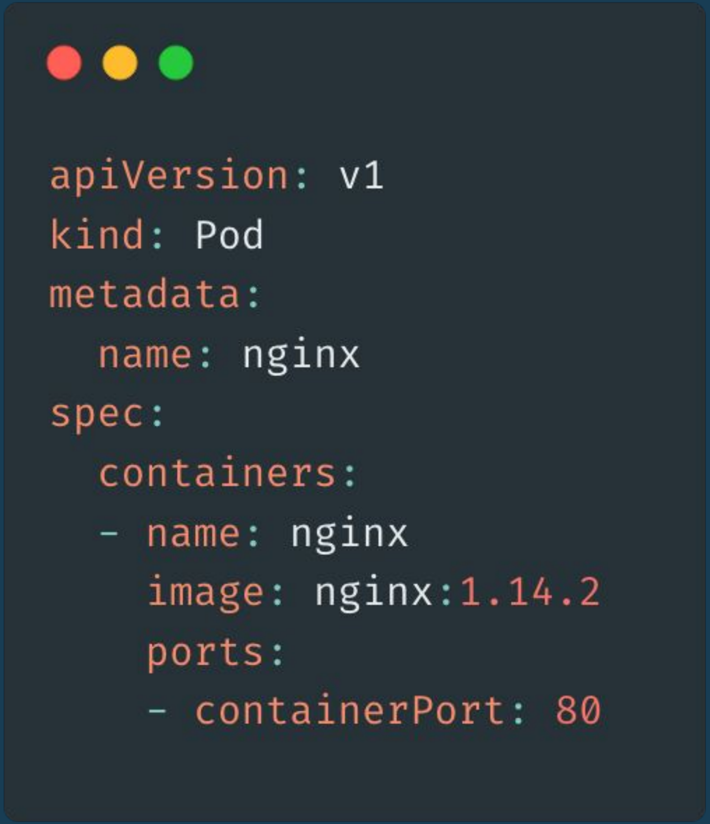
Hasan Türken (@turkenh)
Principal Software Engineer at *Upbound*
Maintainer at *Crossplane*

Crossplane

What is Crossplane?

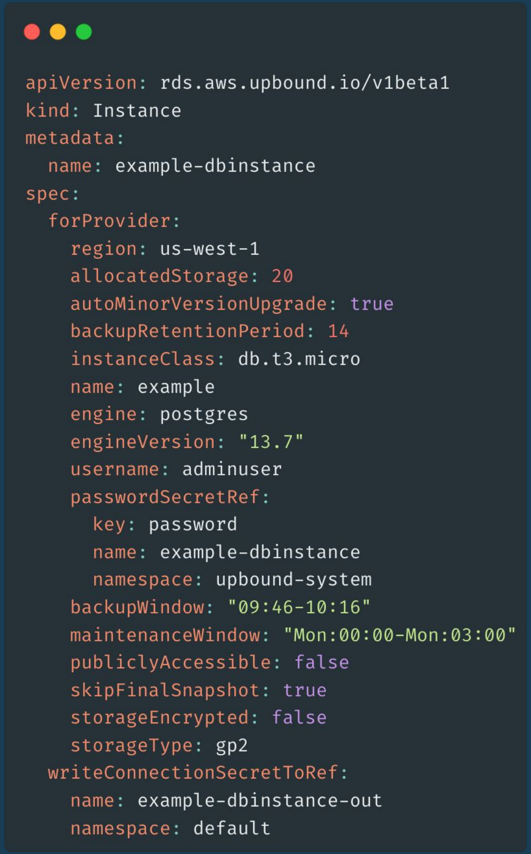

```
kubectl apply -f resource.yaml
```

What is Crossplane?



```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```

What is Crossplane?



```yaml
apiVersion: rds.aws.upbound.io/v1beta1
kind: Instance
metadata:
  name: example-dbinstance
spec:
  forProvider:
    region: us-west-1
    allocatedStorage: 20
    autoMinorVersionUpgrade: true
    backupRetentionPeriod: 14
    instanceClass: db.t3.micro
    name: example
    engine: postgres
    engineVersion: "13.7"
    username: adminuser
    passwordSecretRef:
      key: password
      name: example-dbinstance
      namespace: upbound-system
    backupWindow: "09:46-10:16"
    maintenanceWindow: "Mon:00:00-Mon:03:00"
    publiclyAccessible: false
    skipFinalSnapshot: true
    storageEncrypted: false
    storageType: gp2
  writeConnectionSecretToRef:
    name: example-dbinstance-out
    namespace: default
```

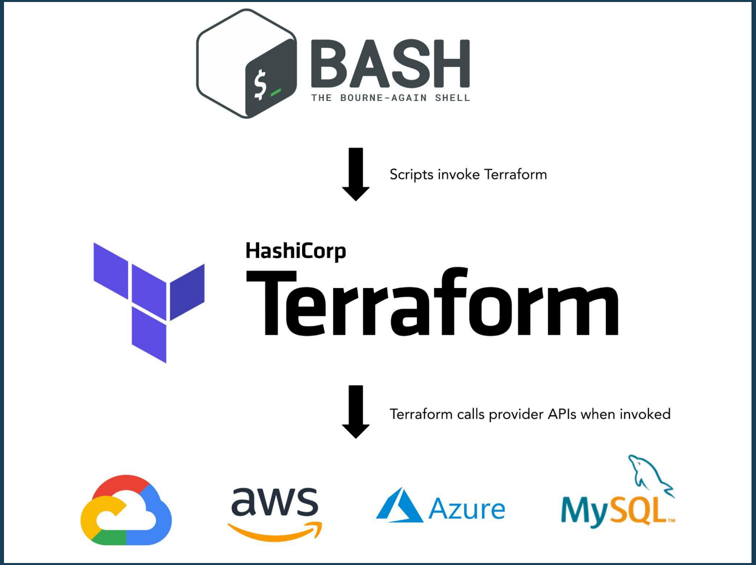
Crossplane

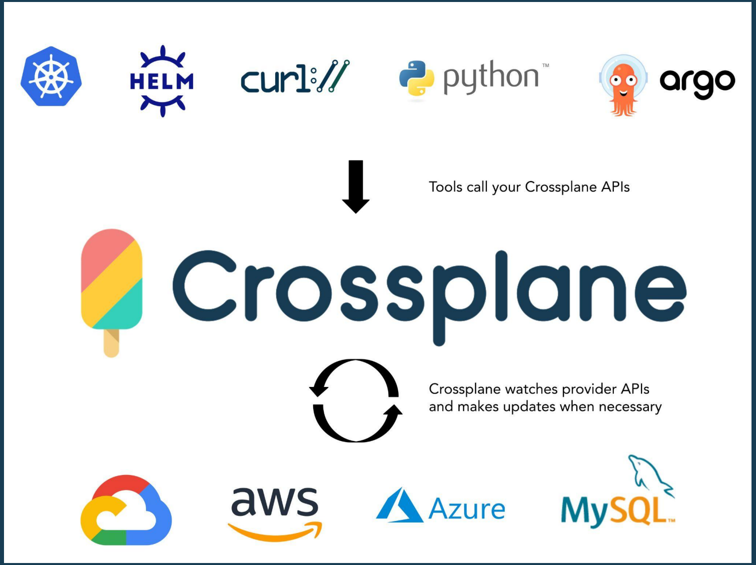
What is Crossplane?

```yaml
apiVersion: database.example.org/v1alpha1
kind: PostgreSQLInstance
metadata:
  name: my-db
spec:
  storageGB: 20
  compositionSelector:
    matchLabels:
      provider: aws
  writeConnectionSecretToRef:
    name: db-creds
```

```yaml
apiVersion: rds.aws.upbound.io/v1beta1
kind: Instance
metadata:
  name: example-dbinstance
spec:
  forProvider:
    region: us-west-1
    allocatedStorage: 20
    autoMinorVersionUpgrade: true
    backupRetentionPeriod: 14
    instanceClass: db.t3.micro
    name: example
    engine: postgres
    engineVersion: "13.7"
    username: adminuser
    passwordSecretRef:
      key: password
      name: example-dbinstance
      namespace: upbound-system
    backupWindow: "09:46-10:16"
    maintenanceWindow: "Mon:00:00-Mon:03:00"
    publiclyAccessible: false
    skipFinalSnapshot: true
    storageEncrypted: false
    storageType: gp2
  writeConnectionSecretToRef:
    name: example-dbinstance-out
    namespace: default
```

Why?


BASH
THE BOURNE-AGAIN SHELL
Scripts invoke Terraform
HashiCorp
Terraform
Terraform calls provider APIs when invoked
aws
Azure
MySQL


HELM
curl://
python
argo
Tools call your Crossplane APIs
Crossplane
Crossplane watches provider APIs
and makes updates when necessary
aws
Azure
MySQL

Why?

	Terraform	Crossplane
Environment	Command line execution	Always-on REST API
State	Single source of truth	State file per environment
Interaction	Only desired state is first-class	Every knob is transparent - desired, observed, defaults
Abstraction Model	Static resource abstraction	Dynamic composition with interfaces
Runtime	Linux experience	Native integration with Kubernetes ecosystem
Cloud Creds	All users need	Only during the setup
Ownership	Hashicorp	CNCF

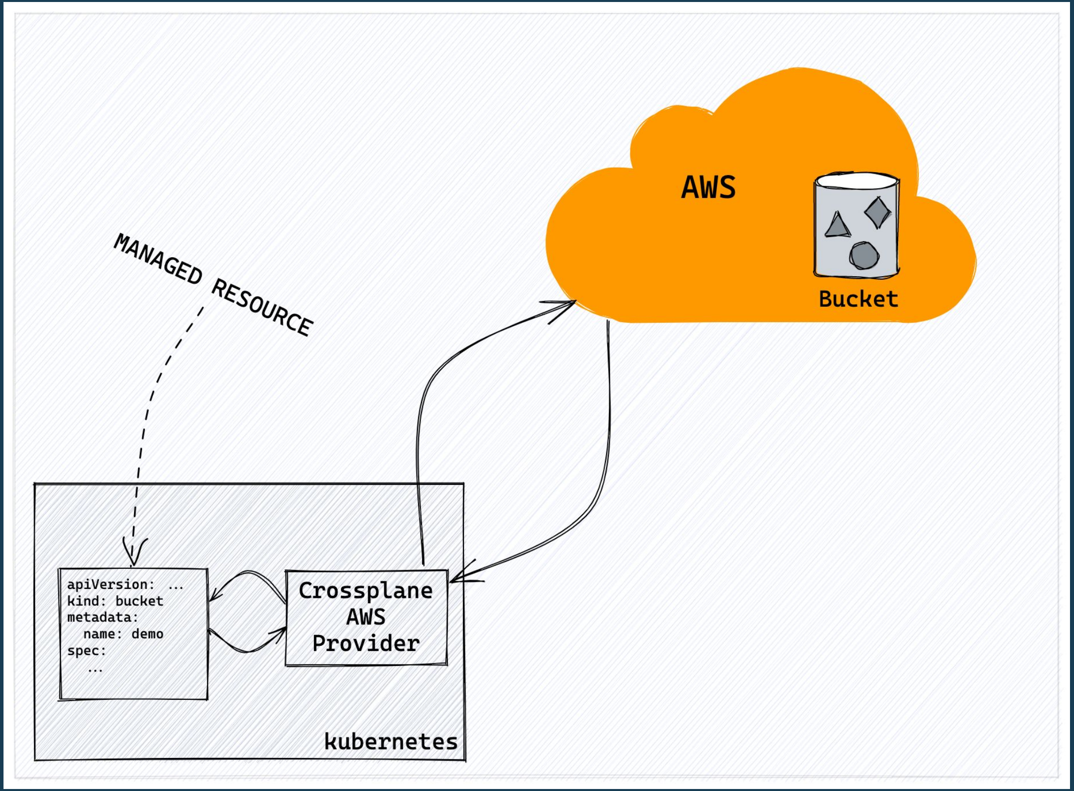
Crossplane

Demo!

Tutorial Link

Concepts: Managed resources (MR)

Crossplane's representation
of a resource
in an external system
(most commonly a cloud provider).



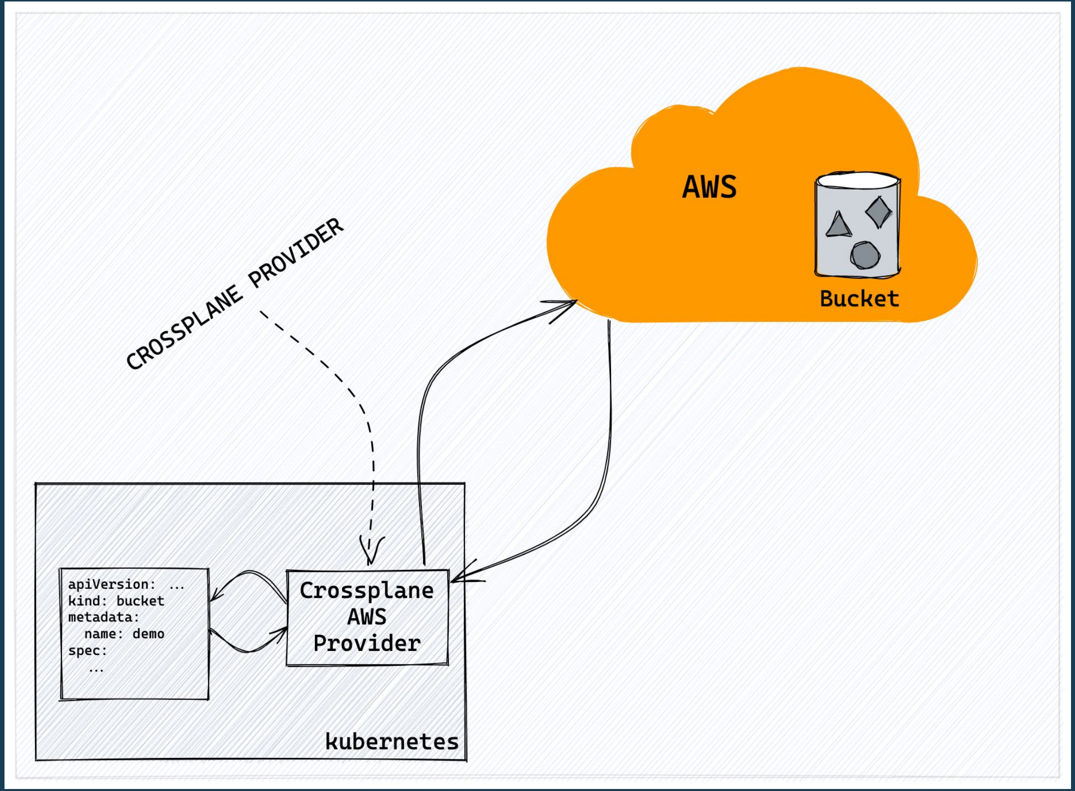
```
apiVersion: ...
kind: bucket
metadata:
  name: demo
spec:
  ...
```

Crossplane

Concepts: Providers

Crossplane packages
that bundle
a set of Managed Resources
and
their respective controllers

More providers?
https://marketplace.upbound.io

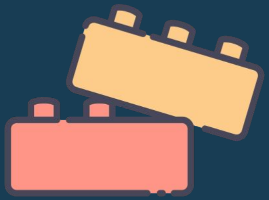


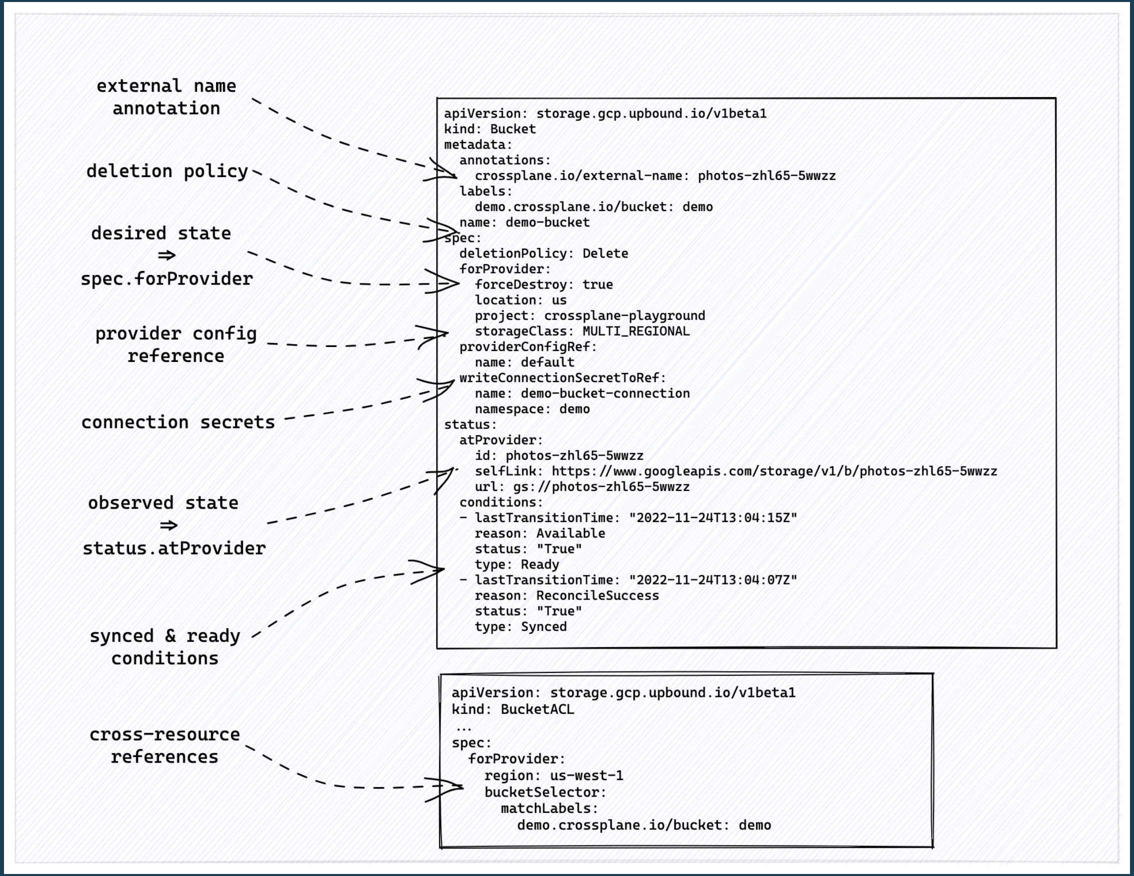
```
apiVersion: ...
kind: bucket
metadata:
  name: demo
spec:
  ...
```

Crossplane

Concepts: Crossplane Resource Model (XRM)

Opinionated and Consistent
API Definition
for
Managed Resources




external name annotation
deletion policy
desired state ⇒ spec.forProvider
provider config reference
connection secrets
observed state ⇒ status.atProvider
synced & ready conditions
cross-resource references

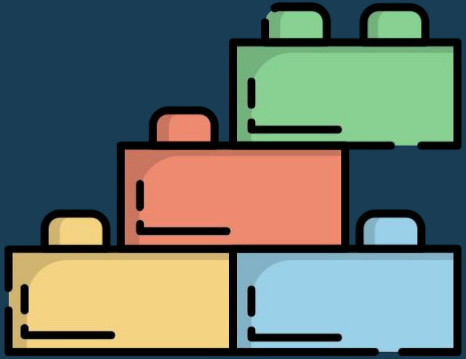
```
apiVersion: storage.gcp.upbound.io/v1beta1
kind: Bucket
metadata:
  annotations:
    crossplane.io/external-name: photos-zhl65-5wwzz
  labels:
    demo.crossplane.io/bucket: demo
  name: demo-bucket
spec:
  deletionPolicy: Delete
  forProvider:
    forceDestroy: true
    location: us
    project: crossplane-playground
    storageClass: MULTI_REGIONAL
  providerConfigRef:
    name: default
  writeConnectionSecretToRef:
    name: demo-bucket-connection
    namespace: demo
status:
  atProvider:
    id: photos-zhl65-5wwzz
    selfLink: https://www.googleapis.com/storage/v1/b/photos-zhl65-5wwzz
    url: gs://photos-zhl65-5wwzz
  conditions:
  - lastTransitionTime: "2022-11-24T13:04:15Z"
    reason: Available
    status: "True"
    type: Ready
  - lastTransitionTime: "2022-11-24T13:04:07Z"
    reason: ReconcileSuccess
    status: "True"
    type: Synced
```

```
apiVersion: storage.gcp.upbound.io/v1beta1
kind: BucketACL
...
spec:
  forProvider:
    region: us-west-1
    bucketSelector:
      matchLabels:
        demo.crossplane.io/bucket: demo
```

Crossplane

Concepts: Compositions

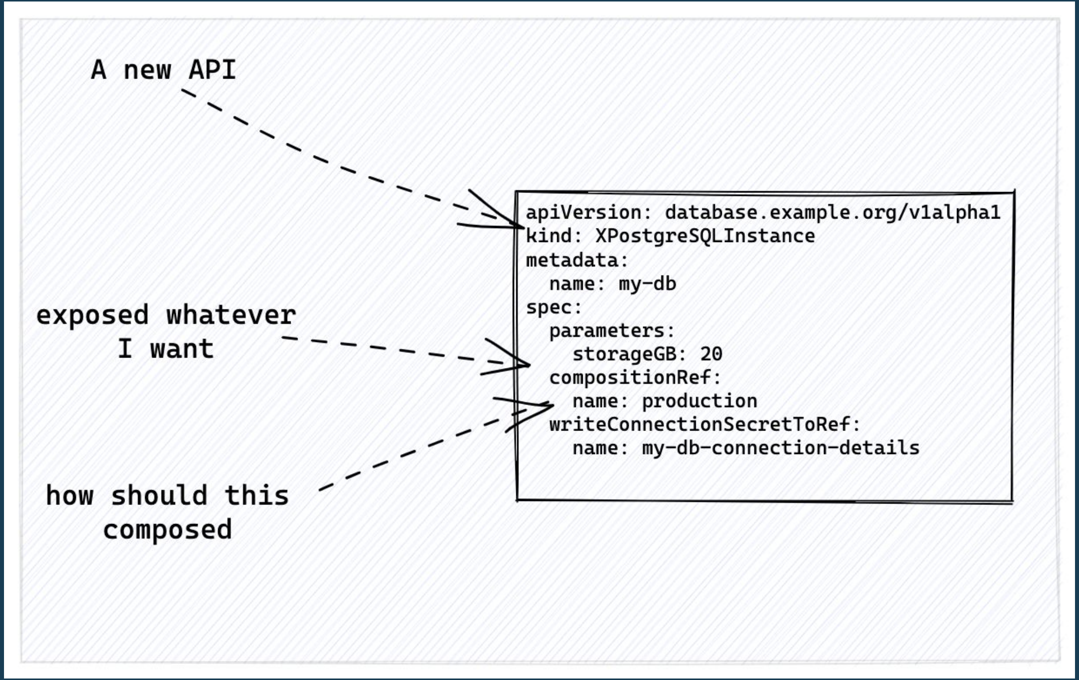
Composing new
Resources (APIs)
using
Managed Resources.



Crossplane

Composite Resources (XRs)

Opinionated
Kubernetes Custom Resources
Composed
of
Managed Resources

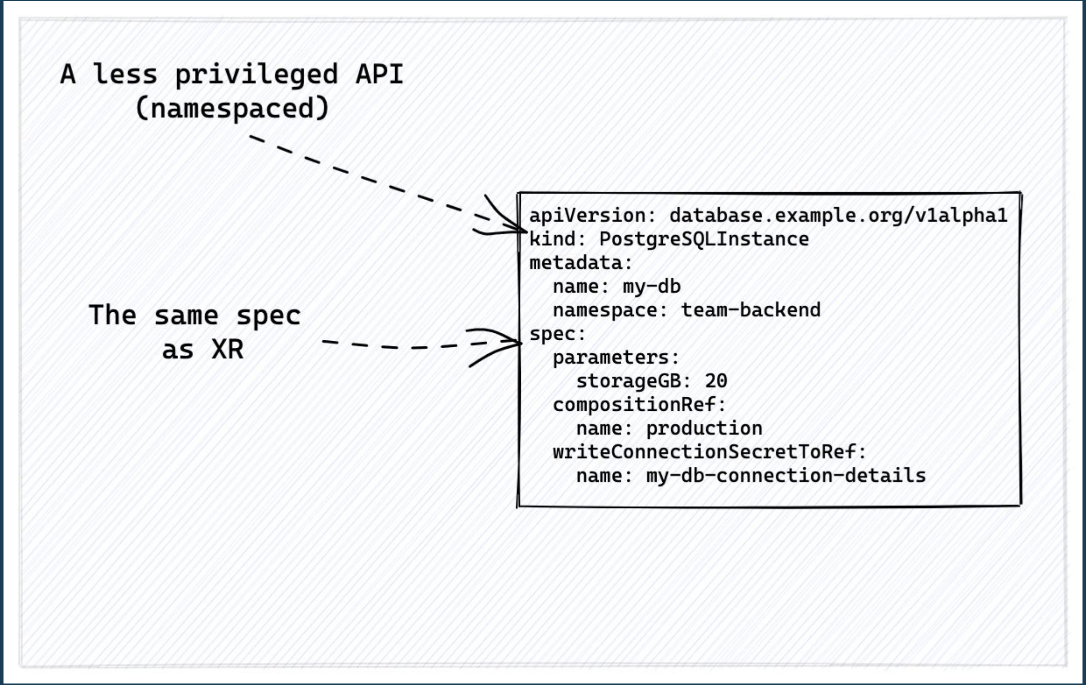


```
apiVersion: database.example.org/v1alpha1
kind: XPostgreSQLInstance
metadata:
  name: my-db
spec:
  parameters:
    storageGB: 20
  compositionRef:
    name: production
  writeConnectionSecretToRef:
    name: my-db-connection-details
```

Crossplane

Claims (XRCs)

The Way
How Consumers
Provision and Manage
Composite Resources (XRs)

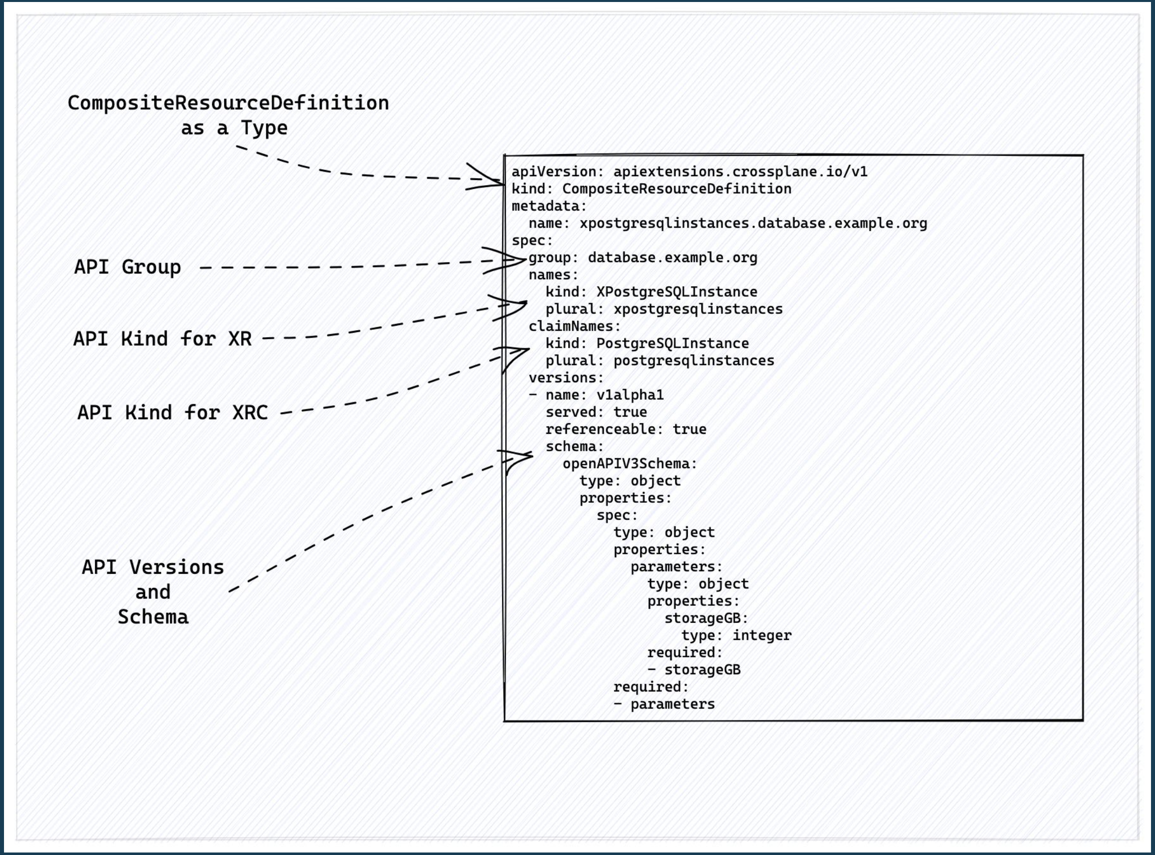


```yaml
apiVersion: database.example.org/v1alpha1
kind: PostgreSQLInstance
metadata:
  name: my-db
  namespace: team-backend
spec:
  parameters:
    storageGB: 20
  compositionRef:
    name: production
  writeConnectionSecretToRef:
    name: my-db-connection-details
```

Crossplane

Composite Resource Definitions (XRDs)

Defines the API
for
Composite Resource

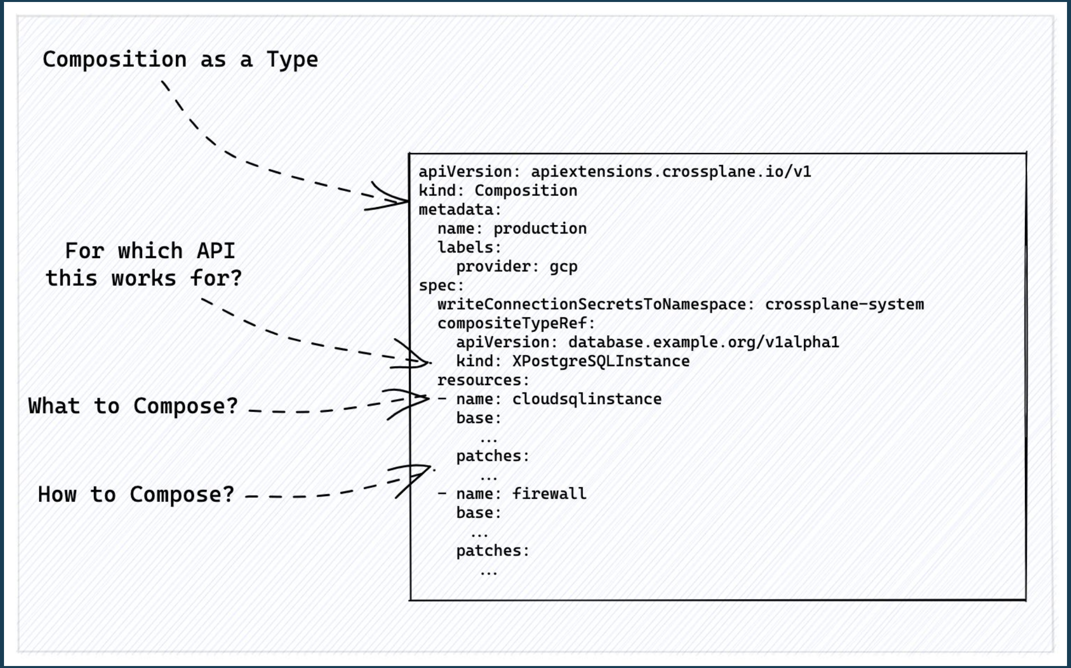


```yaml
apiVersion: apiextensions.crossplane.io/v1
kind: CompositeResourceDefinition
metadata:
  name: xpostgresqlinstances.database.example.org
spec:
  group: database.example.org
  names:
    kind: XPostgreSQLInstance
    plural: xpostgresqlinstances
  claimNames:
    kind: PostgreSQLInstance
    plural: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

Crossplane

Composition (Type)

What and How
to Compose
Managed Resources



```
apiVersion: apiextensions.crossplane.io/v1
kind: Composition
metadata:
  name: production
  labels:
    provider: gcp
spec:
  writeConnectionSecretsToNamespace: crossplane-system
  compositeTypeRef:
    apiVersion: database.example.org/v1alpha1
    kind: XPostgreSQLInstance
  resources:
  - name: cloudsqlinstance
    base:
      ...
    patches:
      ...
  - name: firewall
    base:
     ...
    patches:
      ...
```

Crossplane

Composition Demo

Step 1: Define a new API with Compositions
(Platform Builder/Operator)

Step 2: Consume the new API with a Claim
(Platform Consumer)

Crossplane