GitOpsifying Cloud Infra with Crossplane

About Me

About Me

Cloud Architect at Accenture Baltics

accenture

Specialized

gitops aws DEV OPS kubernetes

Certifications

kubernetes AND CLOUD NATIVE ASSOCIATE

CERTIFIED kubernetes ADMINISTRATOR

CERTIFIED kubernetes APPLICATION DEVELOPER

aws certified Solutions Architect ASSOCIATE

Experience

13
Years Of Experience

5
Companies

22
Projects

Agenda

- Problem
- Solution
  - Kubernetes
  - Argo CD
  - Crossplane
- Solution Diagram
- What Why How Crossplane?
- Crossplane components overview.
- Demo

Problem

The problem is to achieve **true GitOps** for both **infrastructure** and **applications** while minimizing the use of **multiple tools** and **languages** and reducing **complexity**.

To have True GitOps you will face challenges:

- Multiple Tools
- Multiple Languages
- Complexity

Infrastructure Automation History

Name	Tools
Manual	
Scripting	(Powershell , Bash)
Configuration Management	(Puppet ,Chef ,Ansible)
IAC	
Declarative	(CloudFormation ,Terraform)
Componentized	(CDK , Pulumi , CDKTF)
Central control plane	(Crossplane)

solution

- Kubernetes and Argo CD, along with Crossplane, are solutions that help to solve the problem statement by providing a more streamlined and automated approach to managing infrastructure and applications using GitOps principles.

Kubernetes

Kubernetes provides a platform for deploying, scaling, and managing containerized applications, making it easier to manage infrastructure as code.

Argo CD

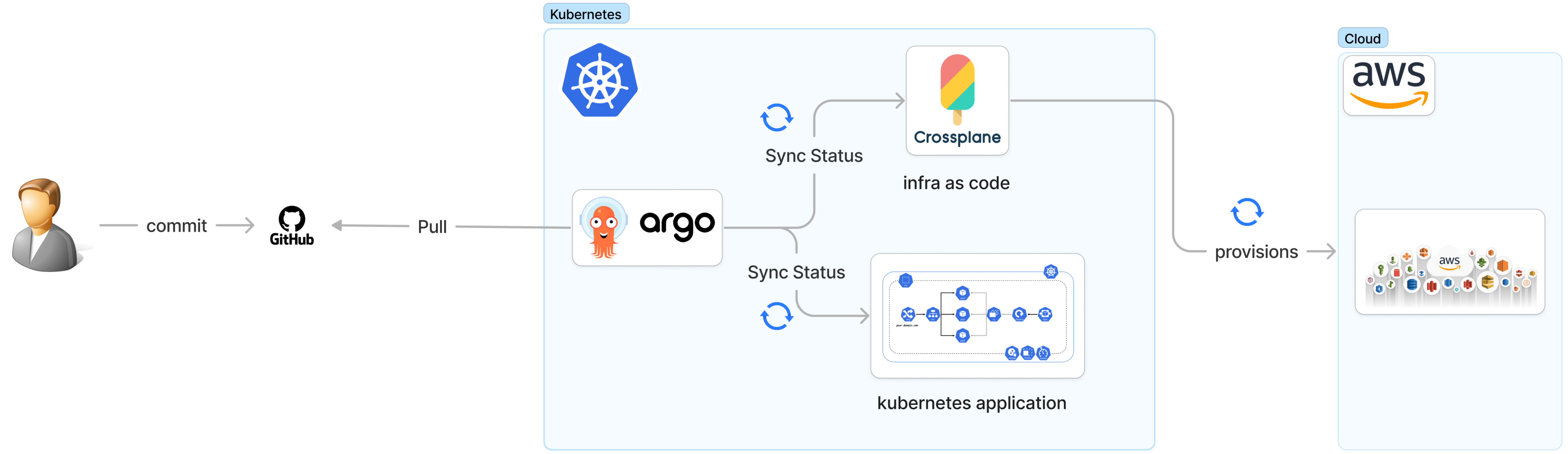
Argo CD is a GitOps-based continuous delivery tool that automates the deployment of applications to Kubernetes clusters, helping to ensure that the desired state of the infrastructure is always in sync with the state described in Git.

Crossplane

Crossplane provides a unified control plane that makes it easier to manage multiple cloud resources using GitOps principles, by allowing administrators to manage infrastructure as code.

By combining these tools, it becomes possible to achieve true GitOps for both infrastructure and applications, reducing complexity and minimizing the need to use multiple tools and languages.

Solution Diagram



What is Crossplane?

Crossplane is an open-source multi-cloud control plane that enables users to manage infrastructure as code and automate provisioning, scaling, and management of cloud resources. It provides a common way to provision and manage resources across different cloud providers and can be integrated with Kubernetes.

Why use Crossplane?

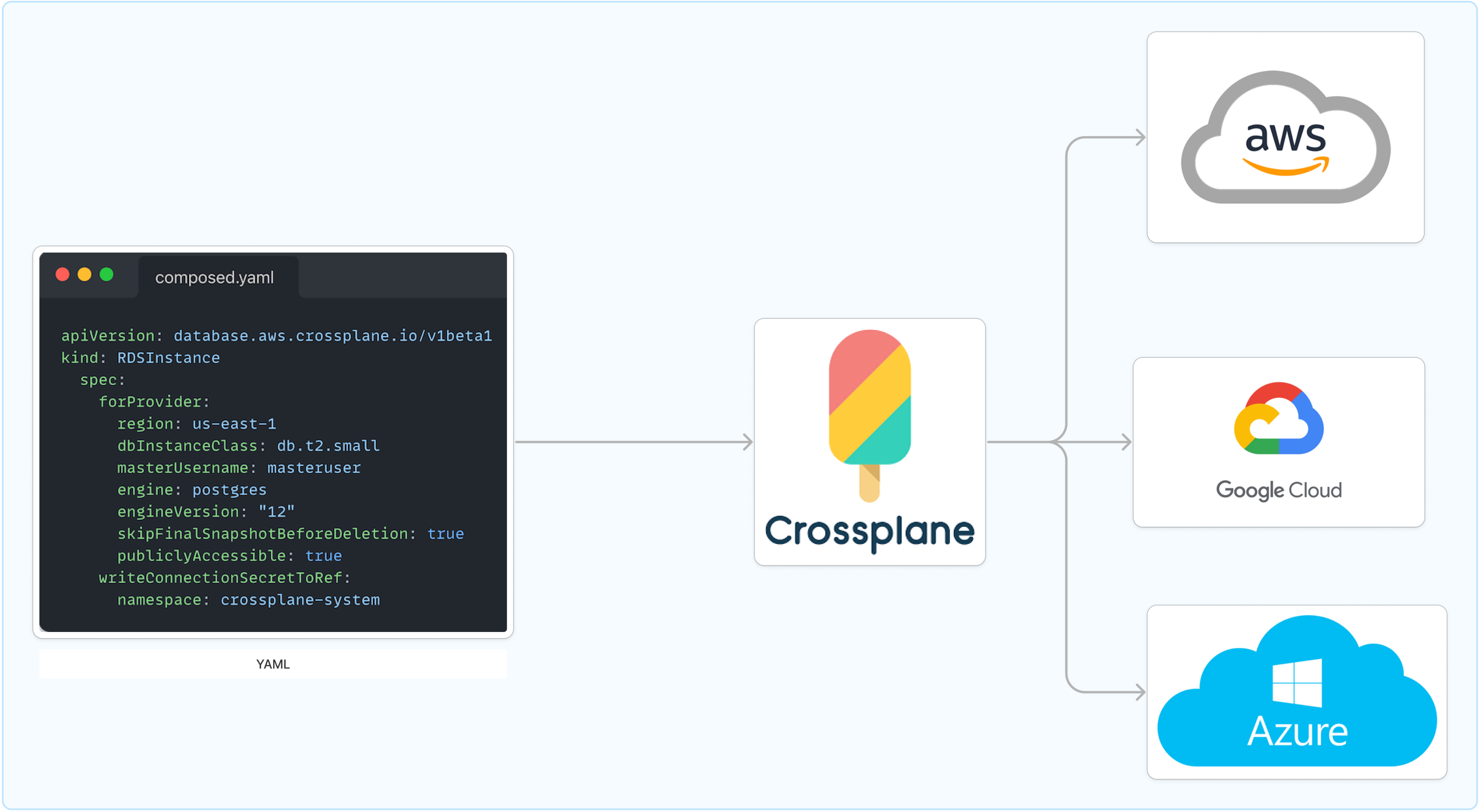
- Declarative configuration
- One source of truth for infrastructure configuration and setup
- Built with high levels of extensibility
- Unify application and infrastructure configuration and deployment
- Automate operational tasks with reconciling controllers
- A strong separation of concerns.

How Crossplane works?

- Crossplane uses a Kubernetes-native API to provision and manage cloud resources, making it easy to integrate with existing Kubernetes workflows and tools.
- Crossplane uses a Custom Resource Definition (CRD) to define the desired state of cloud resources, which can be versioned and tracked in Git.

How Crossplane works?

- Crossplane uses controllers that watch for changes to the CRDs and automatically provision and manage the resources to match the desired state.
- Crossplane can be integrated with other Kubernetes tools such as Helm, Kustomize, and Kubernetes Operators to provide a powerful and flexible multi-cloud infrastructure management solution.



```yaml
apiVersion: database.aws.crossplane.io/v1beta1
kind: RDSInstance
  spec:
    forProvider:
      region: us-east-1
      dbInstanceClass: db.t2.small
      masterUsername: masteruser
      engine: postgres
      engineVersion: "12"
      skipFinalSnapshotBeforeDeletion: true
      publiclyAccessible: true
    writeConnectionSecretToRef:
      namespace: crossplane-system
```

Crossplane components overview

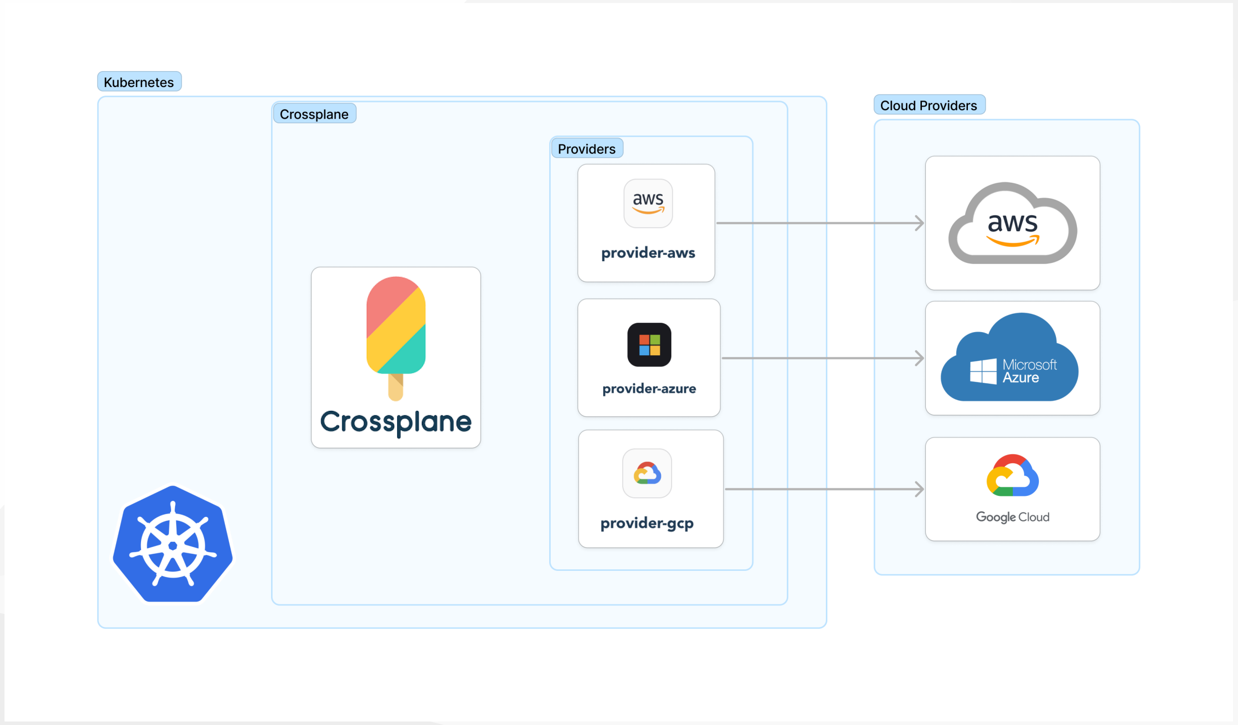
Component	Abbreviation	Scope
Provider		cluster
ProviderConfig	PC	cluster
Managed Resource	MR	cluster
Composition		cluster
Composite Resources	XR	cluster
Composite Resource Definitions	XRD	cluster
Claims	XC	namespace

Providers

Creates new Kubernetes Custom Resource Definitions for an external service.

ProviderConfig

Applies settings for a Provider.

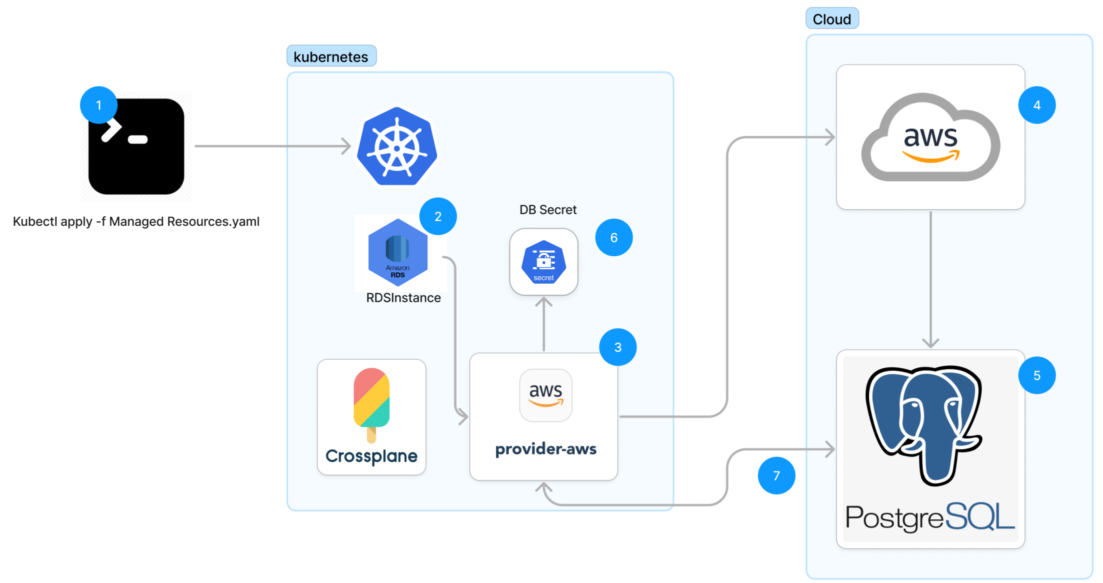


```yaml
apiVersion: pkg.crossplane.io/v1
kind: Provider
metadata:
  name: provider-aws
spec:
  package: "xpkg.upbound.io/crossplane-contrib/provider-aws:v0.33.0"
```

```yaml
apiVersion: aws.crossplane.io/v1beta1
kind: ProviderConfig
metadata:
  name: aws-provider
spec:
  credentials:
    source: Secret
    secretRef:
      namespace: crossplane-system
      name: aws-creds
      key: creds
```

Managed Resources

A provider resource created and managed by Crossplane inside the Kubernetes cluster.



```yaml
apiVersion: database.aws.crossplane.io/v1beta1
kind: RDSInstance
metadata:
  name: rdspostgresql
spec:
  forProvider:
    region: us-east-1
    dbInstanceClass: db.t2.small
    masterUsername: masteruser
    allocatedStorage: 20
    engine: postgres
    engineVersion: "12"
    skipFinalSnapshotBeforeDeletion: true
  writeConnectionSecretToRef:
    namespace: crossplane-system
    name: aws-rdspostgresql-conn
```

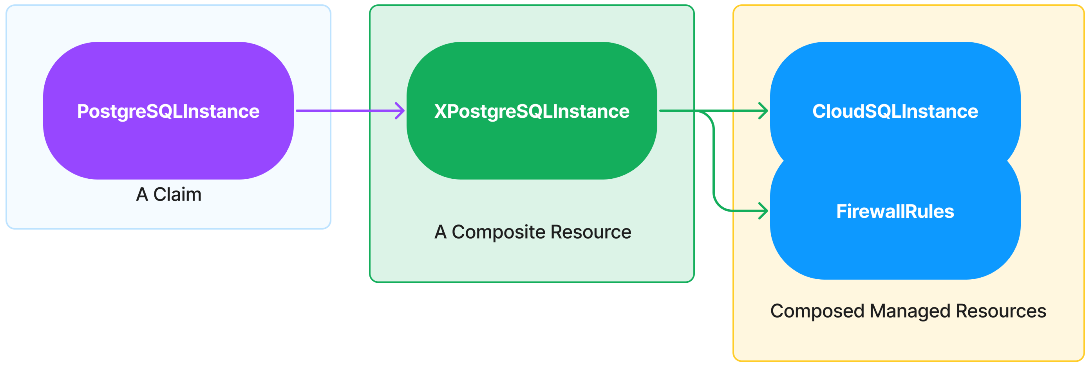
Composition

A template for creating multiple managed resources at once.

```yaml
apiVersion: apiextensions.crossplane.io/v1
kind: Composition
metadata:
  name: example
  labels:
    crossplane.io/xrd: xpostgresqlinstances.database.example.org
    provider: gcp
spec:
  writeConnectionSecretsToNamespace: crossplane-system
  compositeTypeRef:
    apiVersion: database.example.org/v1alpha1
    kind: XPostgreSQLInstance
  resources:
  - name: cloudsqlinstance
    base:
      apiVersion: database.gcp.crossplane.io/v1beta1
      kind: CloudSQLInstance
      spec:
        forProvider:
          databaseVersion: POSTGRES_12
          region: us-central1
          settings:
            tier: db-custom-1-3840
            dataDiskType: PD_SSD
            ipConfiguration:
              ipv4Enabled: true
              authorizedNetworks:
                - value: "0.0.0.0/0"
    patches:
    - type: FromCompositeFieldPath
      fromFieldPath: spec.parameters.storageGB
      toFieldPath: spec.forProvider.settings.dataDiskSizeGb
```

Composite Resources

Uses a Composition template to create multiple managed resources as a single Kubernetes object.



Composite Resource Definitions

Defines the API schema for Composite Resources and Claims

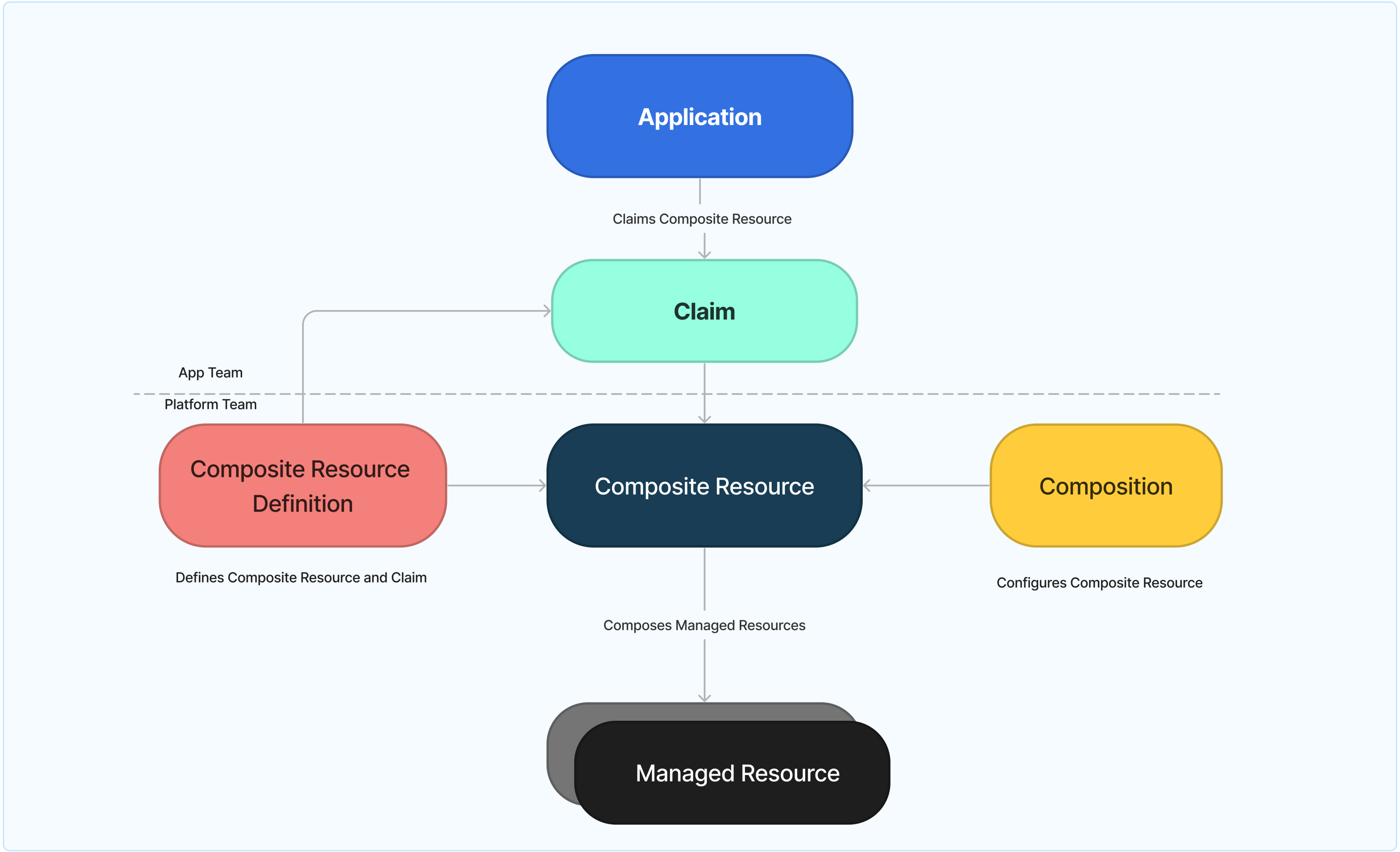
```yaml
apiVersion: apiextensions.crossplane.io/v1
kind: CompositeResourceDefinition
metadata:
  name: xpostgresqlinstances.database.example.org
spec:
  group: database.example.org
  names:
    kind: XPostgreSQLInstance
    plural: xpostgresqlinstances
  claimNames:
    kind: PostgreSQLInstance
    plural: 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
```

Claims

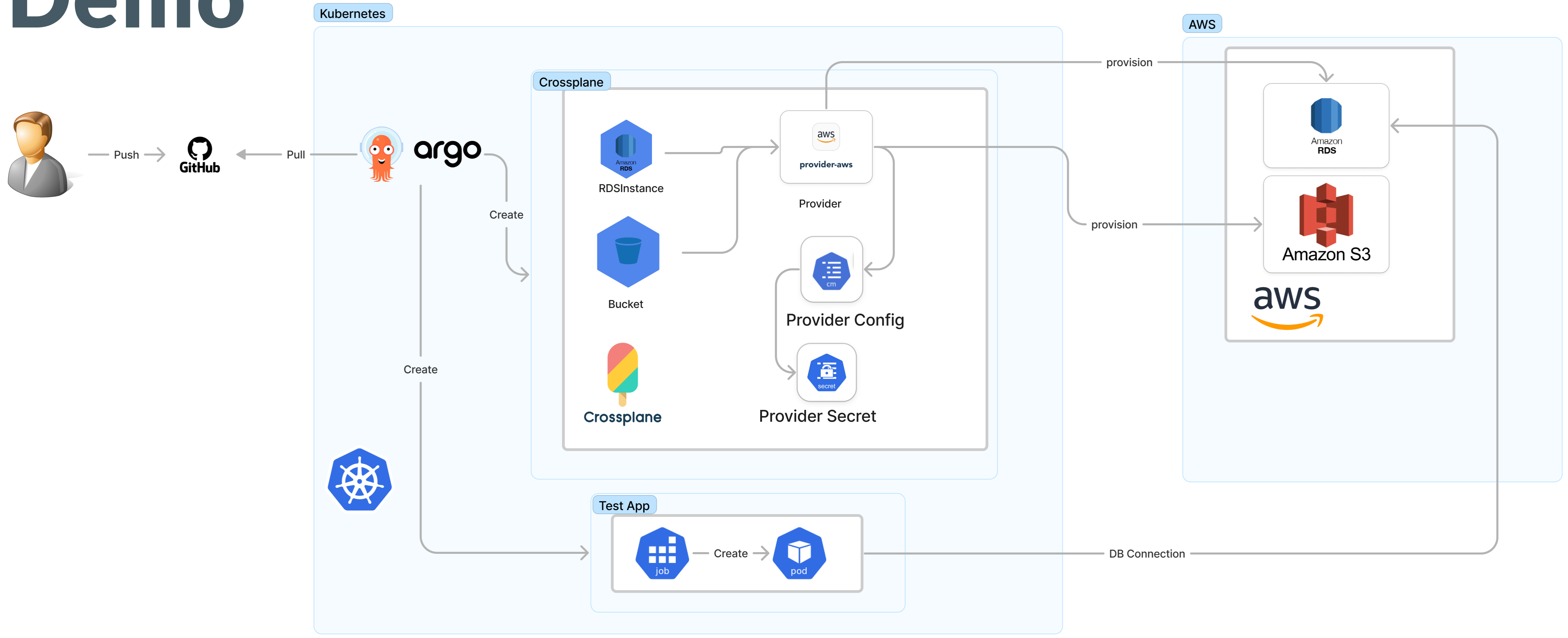
Like a Composite Resource Uses a Composition template to create multiple managed resources as a single Kubernetes object., but namespace scoped.

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

How Composite Resources Works


Application
Claims Composite Resource
Claim
App Team
Platform Team
Composite Resource Definition
Defines Composite Resource and Claim
Composite Resource
Composition
Configures Composite Resource
Composes Managed Resources
Managed Resource

Demo



Kubernetes
Push
GitHub
Pull
argo
Create
Create
Crossplane
RDSInstance
Bucket
Crossplane
aws
provider-aws
Provider
cm
Provider Config
secret
Provider Secret
Test App
job
Create
pod
provision
provision
DB Connection
AWS
Amazon RDS
Amazon S3
aws

Q
A

You Can Find Me https://www.linkedin.com/in/Ebrahim-Ramadan