The Magic of Backing Service Provisioning and Consumption

With Crossplane and ServiceBindings

Timo Salm
Senior Lead Tanzu DevX Solution Engineer - EMEA

May 2024

About me

Timo Salm

Senior Lead Tanzu DevX Solution Engineer – EMEA

https://tanzu.vmware.com

- X (Twitter): @salmto
- LinkedIn: https://linkedin.com/in/timosalm
- GitHub: https://github.com/timosalm

Cloud Native Applications

Ship faster, reduce risk, and grow your business

“Cloud native is structuring teams, culture, and technology to utilize automation and architectures to manage complexity and unlock velocity.”

Joe Beda, Co-Founder Kubernetes

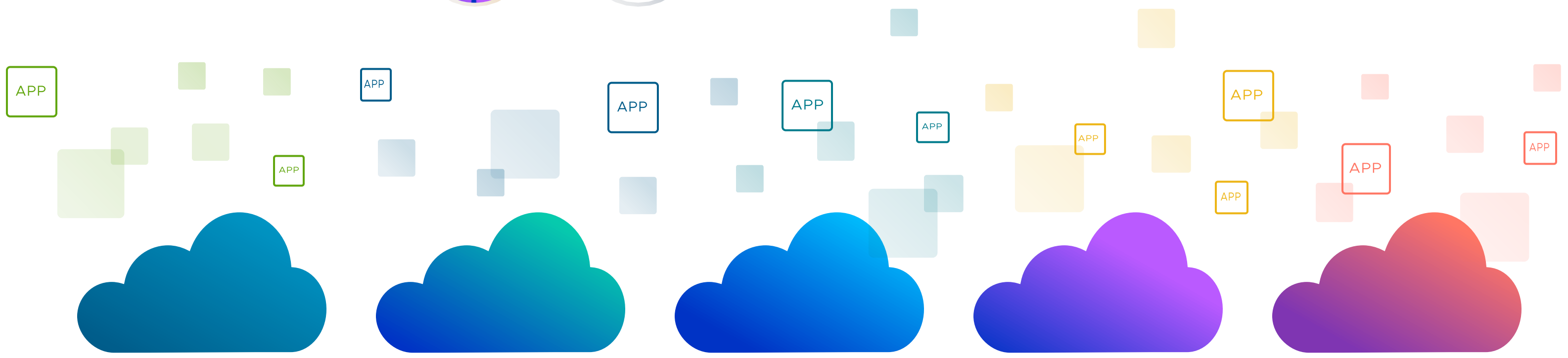
How do you build cloud native applications?

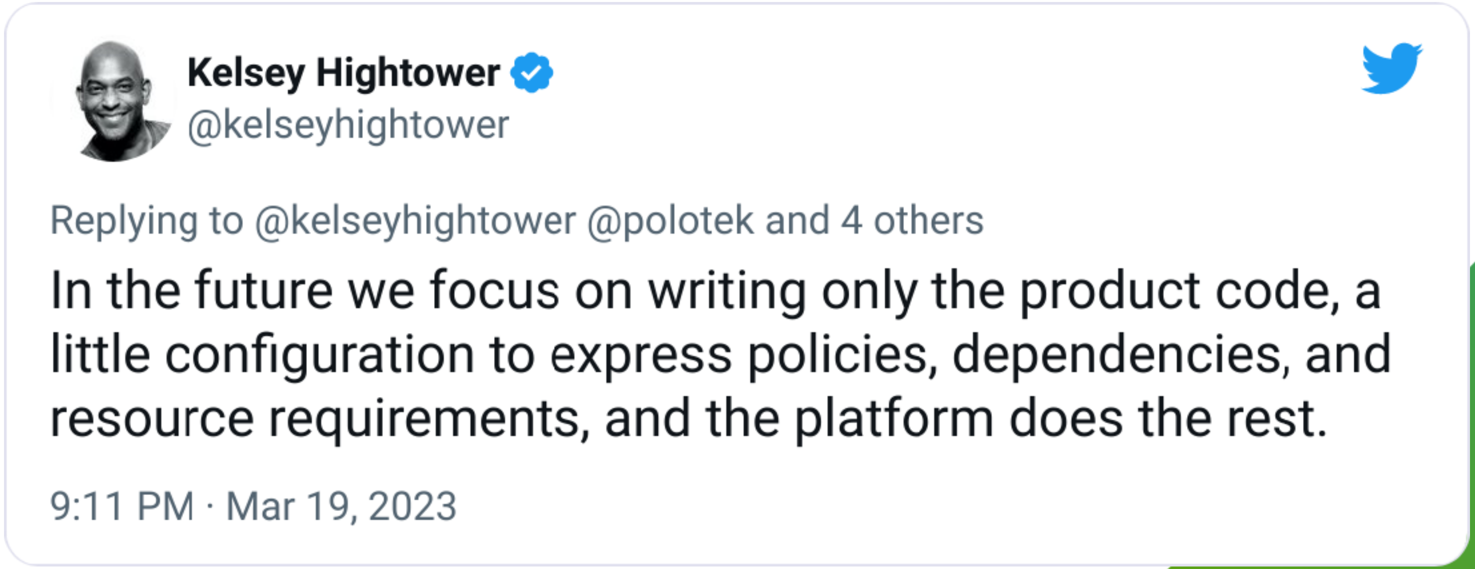
- Agile
- DevSecOps
- Continuous Delivery
- Microservices/Modulith
- Containers

Navigating the Complexity of Cloud Native Applications and Infrastructure

Many applications, teams, processes, a fragmented ecosystem of tools, …

- Container and K8s Runtime
- Migration
- CI/CD
- Observability
- Build Automation
- OSS Packages
- Catalog
- Lifecycle Mgmt
- Run-Time Security
- Capacity Mgmt
- Elastic Scale Out
- Regulatory Config
- Cloud and K8s Cost Optimization
- Access Control
- Secrets Mgmt
- Infra-as-Code
- Secure Supply Chain
- Policies and Guardrails
- Connectivity





Kelsey Hightower
@kelseyhightower

Replying to @kelseyhightower @polotek and 4 others

In the future we focus on writing only the product code, a little configuration to express policies, dependencies, and resource requirements, and the platform does the rest.

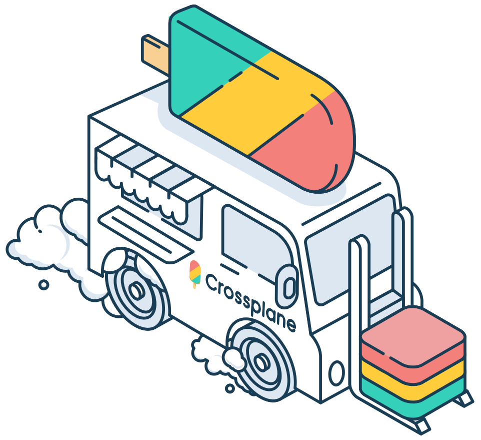
9:11 PM · Mar 19, 2023

Provisioning of Backing Services

With Crossplane

The ability to efficiently provision and manage backing services is critical for cloud native applications!

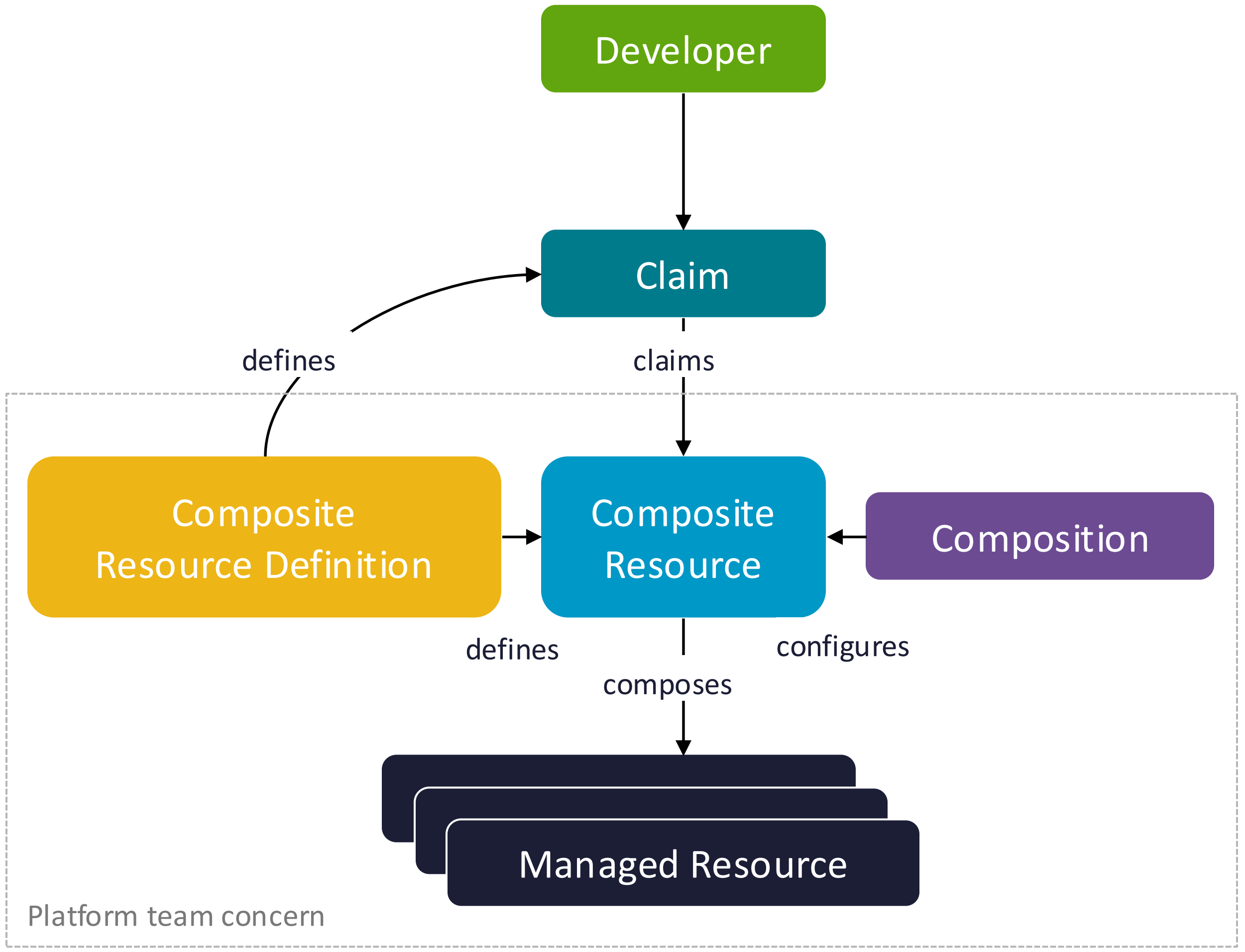
Crossplane enables developers and operators to define, provision, and manage these services with ease.



- Crossplane is an open-source, CNCF project built on the foundation of Kubernetes to orchestrate anything
- There is huge focus on the provisioning and management of external, non-Kubernetes resources like AWS, Azure or GCP services
- Allows platform teams to build abstractions with custom Kubernetes APIs for the consumption of resources with separation of concerns

Provisioning of Backing Services

Crossplane Concepts



Crossplane introduces multiple building blocks that allow for the separation of concern between different personas.

- **Providers** are packages that enable Crossplane to provision infrastructure on an external service
- **Managed Resources** are Kubernetes custom resources that represent infrastructure primitives
- A **composite resource** is a special kind of custom resource that is defined by a **CompositeResourceDefinition (XRD)**. It composes one or more managed resources into a higher-level infrastructure unit. The **Composition** template defines how to create resources.
- **Claims** are the primary way developers interact with Crossplane. They look like composite resources but are namespace-scoped instead of cluster-scoped.
- **Configuration Packages** are a collection of Compositions, Composite Resource Definitions, and any required Providers or Functions.

Diagram source: https://docs.crossplane.io/v1.15/concepts/claims/

Consumption of Backing Services

The Service Binding Specification for Kubernetes and its reference implementation make it as easy as possible to consume those dynamically provisioned backing services.

It automatically injects credentials that are required for the connection to the backing service into the containers of the running application.

The Spring Cloud Bindings library exposes a rich Java language binding for the Kubernetes Service Binding Specification.

It configures Spring Boot application configuration properties appropriate for the type of binding encountered.

The Paketo Spring Boot Cloud Native Buildpack adds Spring Cloud Bindings to the application classpath by default.

https://github.com/timosalm/
crossplane-and-servicebindings

Thank You

- X (Twitter): @salmto
- LinkedIn: https://linkedin.com/in/timosalm
- GitHub: https://github.com/timosalm