

Building Internal Developer Platforms with Kubernetes Operators

Conf42 Cloud Native 2024



George Hantzaras
Engineering Director, MongoDB



Dan Mckean
Sr Product Manager, MongoDB



Agenda

- Internal Developer Platforms
- The Rise of Kubernetes
- Extending Kubernetes
- Example IDP Architecture
- Demo



Internal Developer Platforms and Platform Engineering



Platform:

A foundation that developers can use to build software applications. It provides a set of tools and services that make it easier to develop, deploy, and manage applications.

Internal Developer Platforms (IDPs) try to go further:

- Reduce the cognitive load
- Provide a consistent experience
- User friendly design

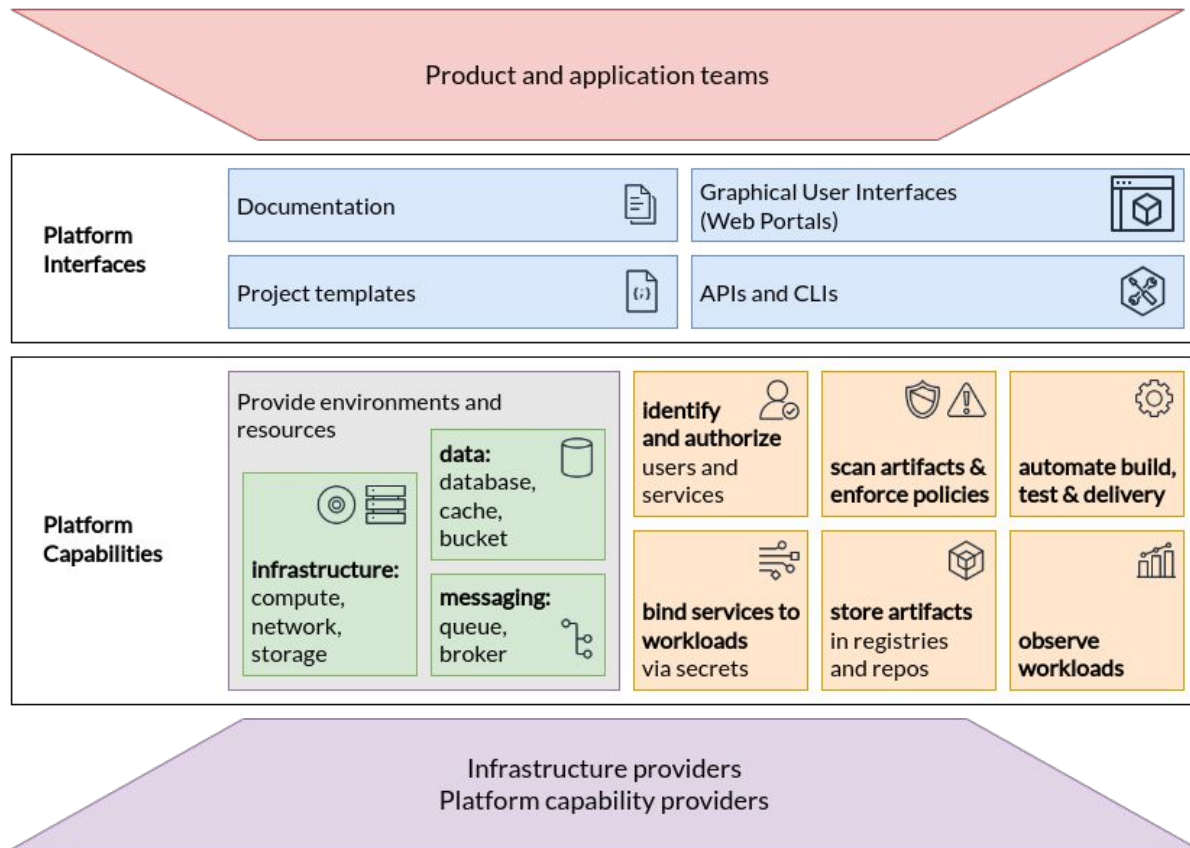


Platform Capabilities

Platforms bridge the gap between underlying capability providers and platform users.

Key components:

- Platform Capabilities
- Platform Interfaces





The Rise of Kubernetes in Platform Engineering



Kubernetes is winning as the Platform of Platforms 🏆

But why...?

- Containerization
- Scalability
- Orchestration
- Flexibility
- Declarative Management

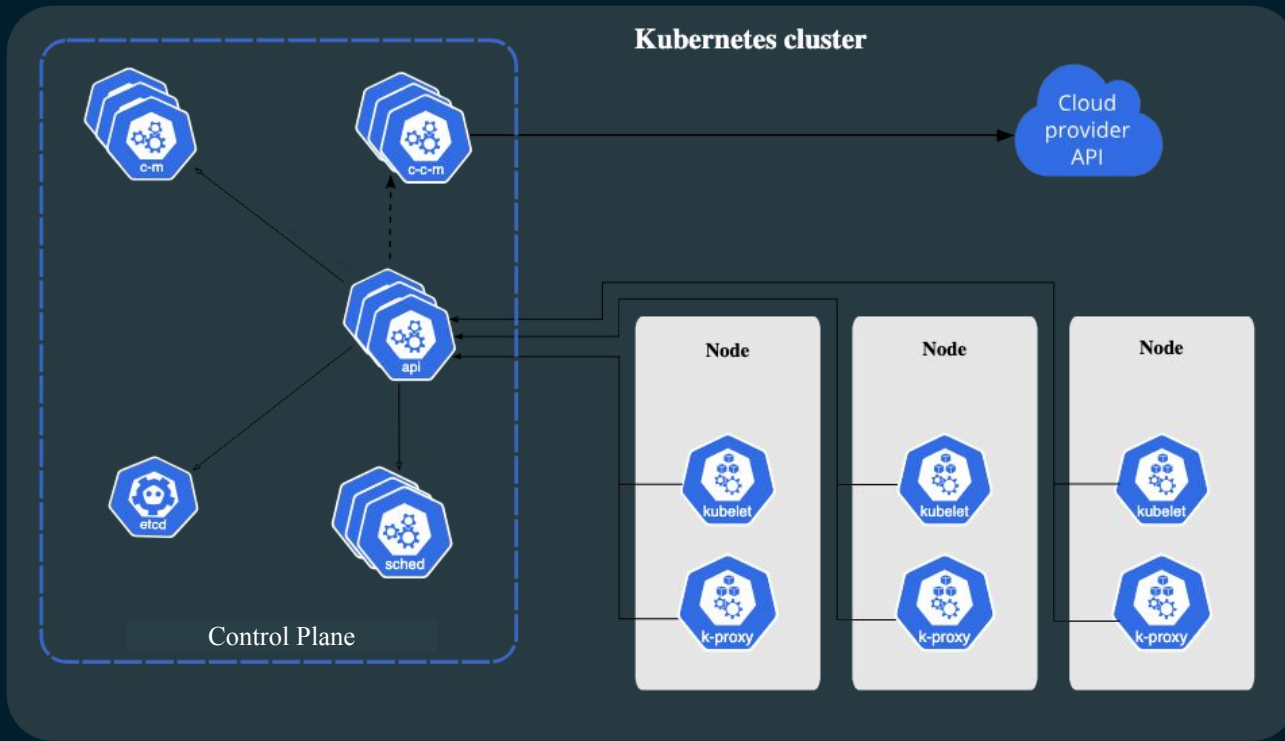


Key Developments:

- From Orchestration to Platform
- A Thriving Ecosystem
- Standardization and Abstraction
- Extensible Interfaces
 - Container Storage Interface (CSI)
 - Container Runtime Interface (CRI)
 - Container Network Interface (CNI)



Kubernetes control plane components





The Controller Manager and the Control Loop

Controller manager =

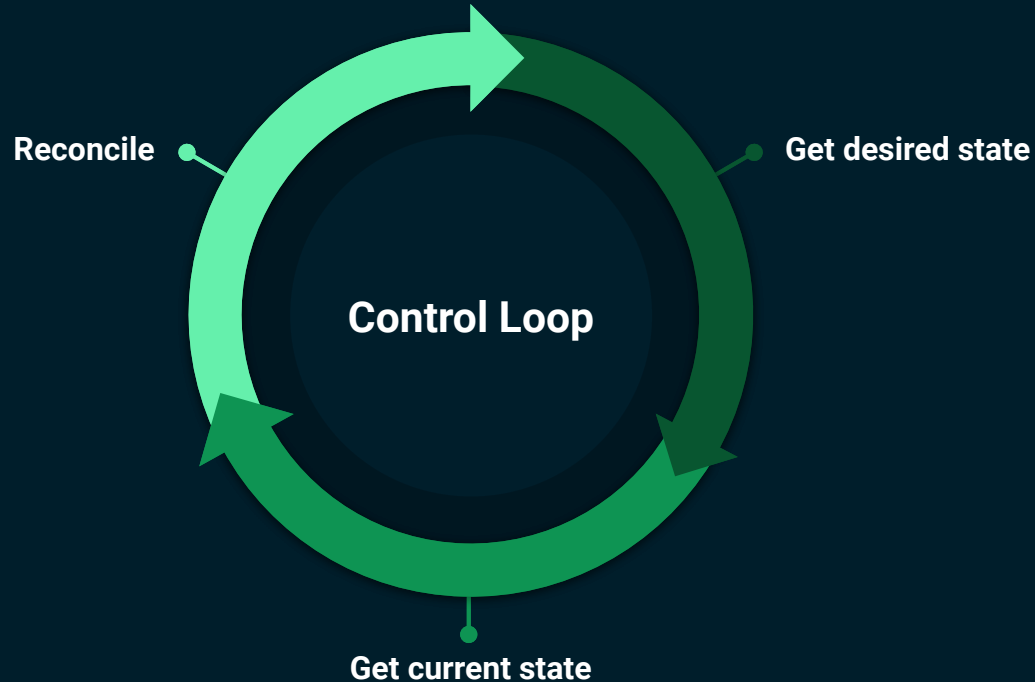
A collection of controllers that together ensure that the state of the cluster matches the desired state.

For example:

- Replication Controller
- Endpoints Controller
- Namespace Controller



The Controller Manager and the Control Loop





Extending Kubernetes



Custom Controllers

Custom controllers =

User-defined programs that extend the capabilities of Kubernetes.

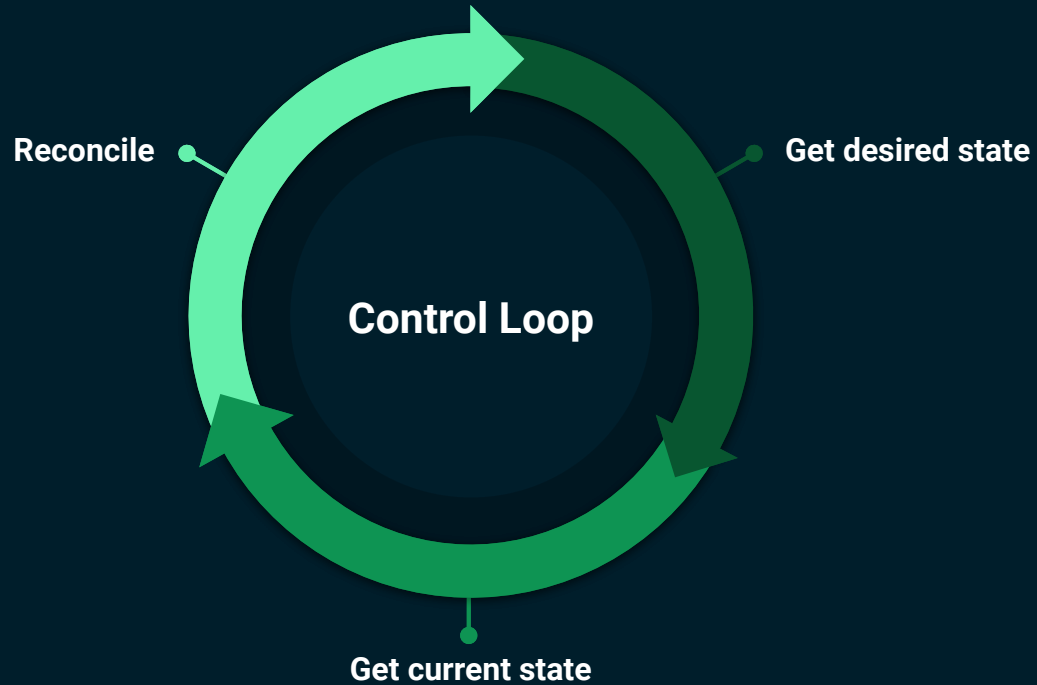
Custom controllers use Custom Resource Definitions (CRDs)

CRD =

A YAML-defined schema for a custom resource



Custom Controllers





Custom Controllers

Benefits of custom controllers:

- Extend native Kubernetes functionality
- Enable declarative management of custom resources
- Automation



Kubernetes Operators

Operators package up custom controllers and more, to make it easier to deploy and manage.

Operators can include the following:

- CRD definitions
- Controller code
- Deployment manifests
- Service manifests
- Helm charts
- Documentation and monitoring tools



Why use Operators

Benefits:

- Simplified application management
- Declarative configuration of an application
- Reduce errors
- Standardized packaging
- Domain-specific expertise
- Rich ecosystem



Two types of Operator

- Internal Operators
 - Managing resources running within the Kubernetes cluster
- External Operators
 - Managing resources outside of the Kubernetes cluster
- Both share the same benefits
 - Simplified application management
 - Declarative configurations
 - Reduced errors
 - Standardized packaging



Demo



Our demo

- Database-as-a-Service capability in an IDP
 - Using an external database service - MongoDB Atlas
 - The Atlas Kubernetes Operator
- A GitOps interface for our users
 - Using the ArgoCD operator
 - Providing a self-service mechanism for our IDP

Thank you!

To find out more about the Atlas Operator (including access to the repo containing all the information of our demo) please go to →

George Peter Hantzaras

Director of Engineering, Kubernetes

Dan Mckean

Senior Product Manager, Kubernetes



linktr.ee/dbaas_on_idp