Kubernetes Deployment and Management: From Platform to API

Yuriy Bezsonov

Senior Solutions Architect
AWS

Agenda for today

Application Delivery Streams

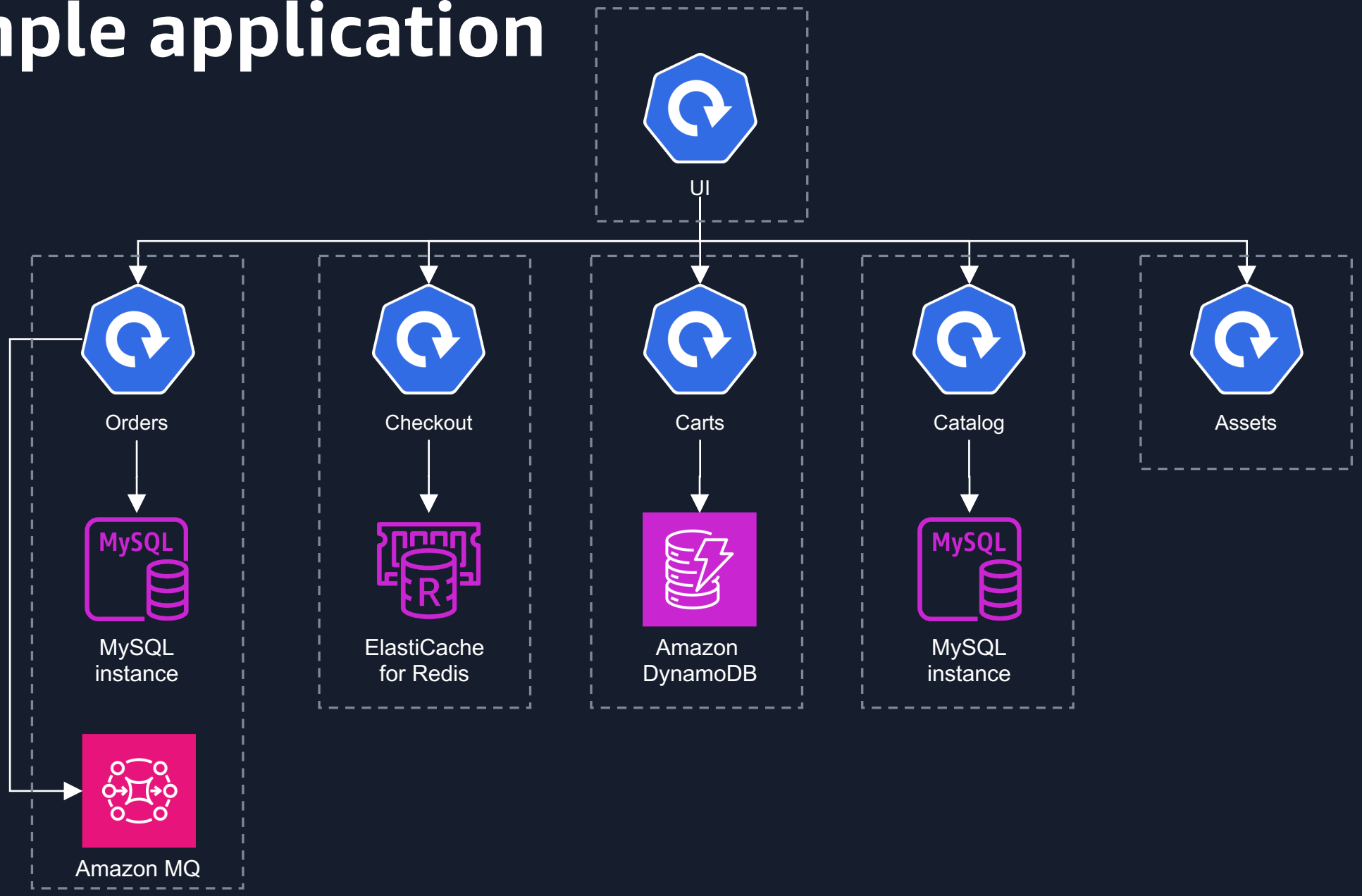
Infrastructure Controllers

Compositions in Kubernetes

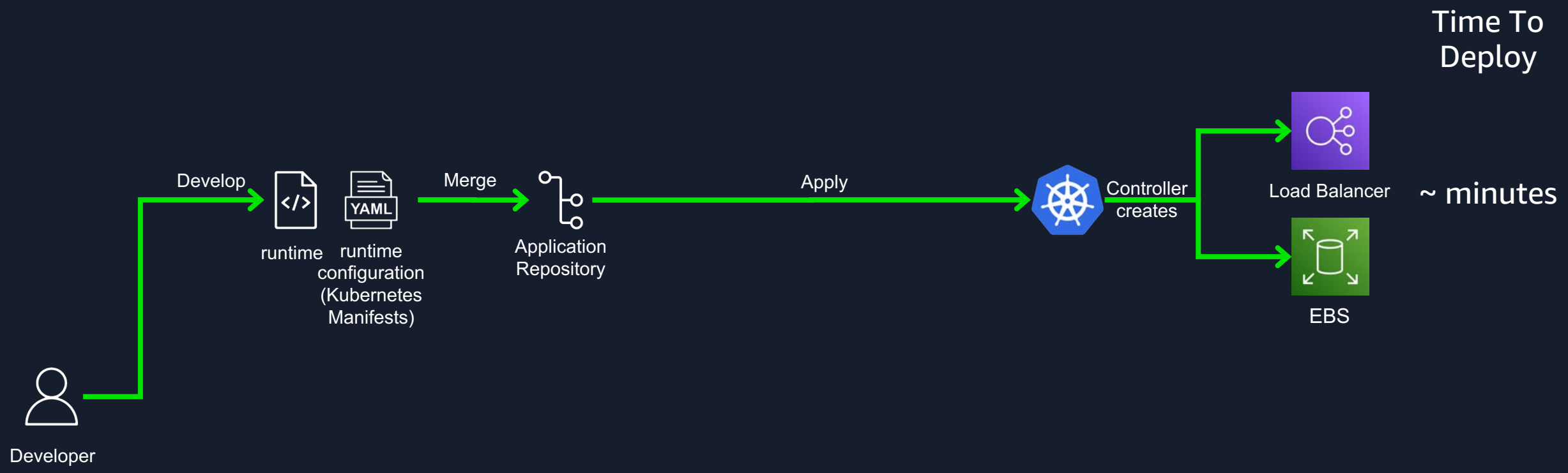
Solution Overview

App Delivery Streams

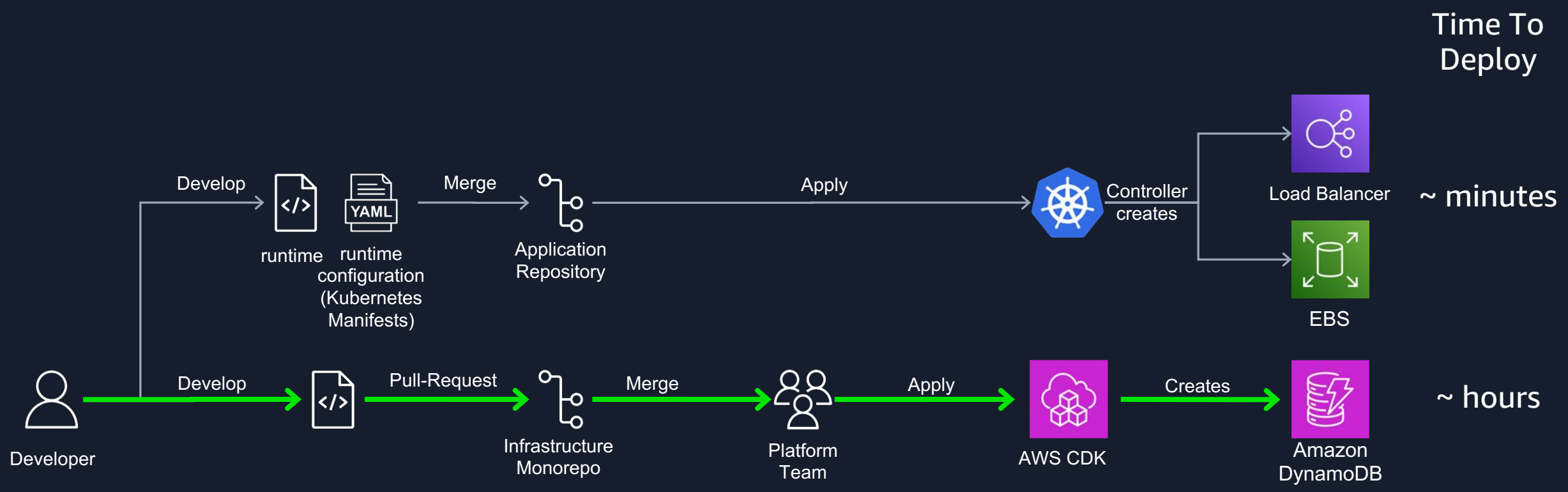
Sample application



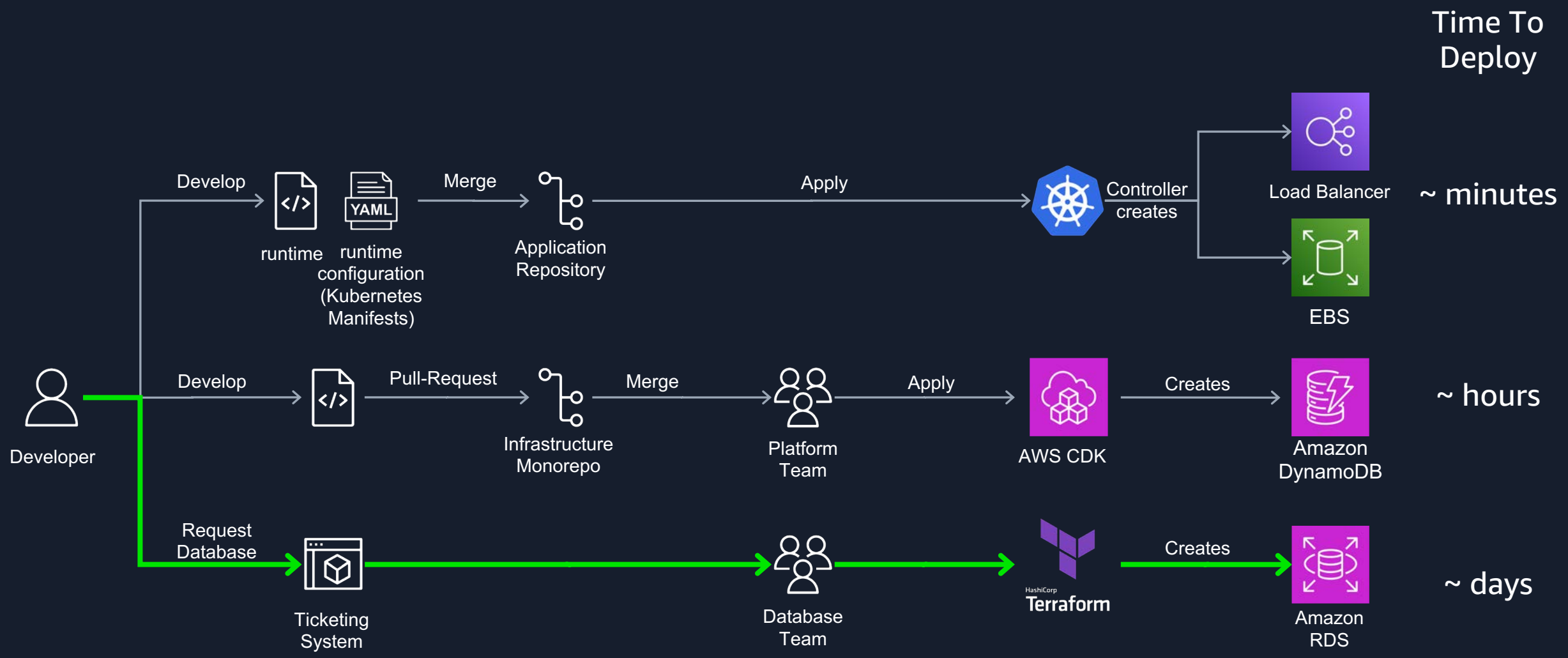
Kubernetes as a Platform



Backing Service – Infrastructure-as-Code



Backing Service – Ticketing System



What can we do to shorten deployment time?

Requirements

Platform Builders

I would like to standardize the deployment process for application teams while enforcing organizational standards.

App Developers

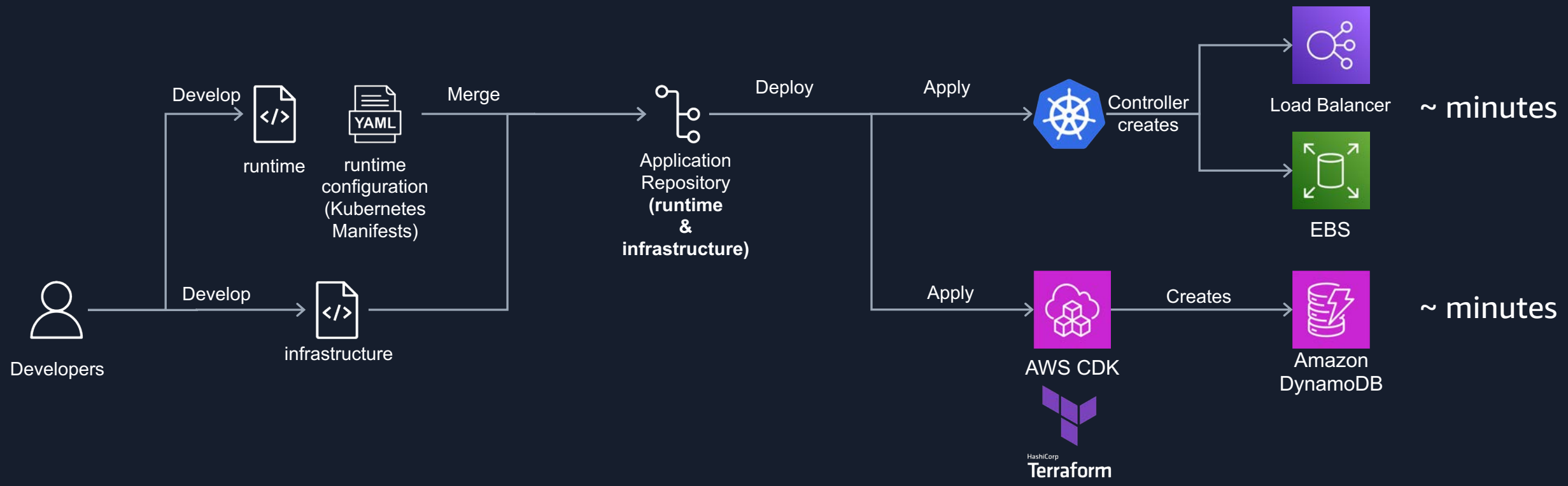
I would like to have a full-ownership of my application and its backing services deployment lifecycle.

Infrastructure and runtime code live in the same package

Best practices for developing and deploying cloud infrastructure with the AWS CDK

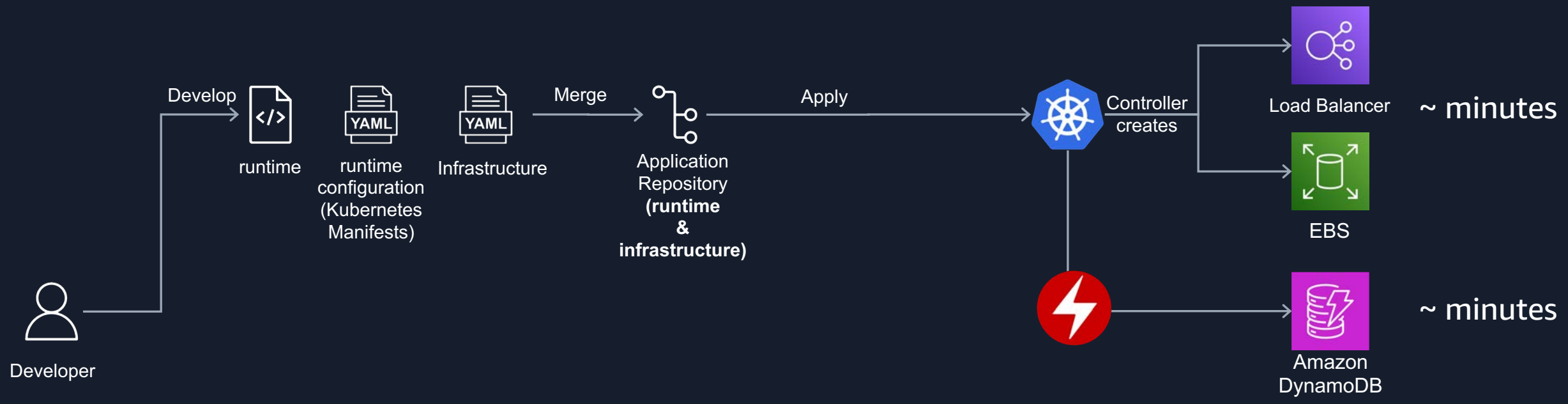
https://docs.aws.amazon.com/cdk/v2/guide/best-practices.html

Kubernetes as a Platform with IaC



Deployment time: from hours to minutes

Kubernetes as an operator/controller



Infrastructure Controllers

Infrastructure Controllers

Manage cloud services using Kubernetes API

A single API for Kubernetes and AWS Services

Create your own platform API

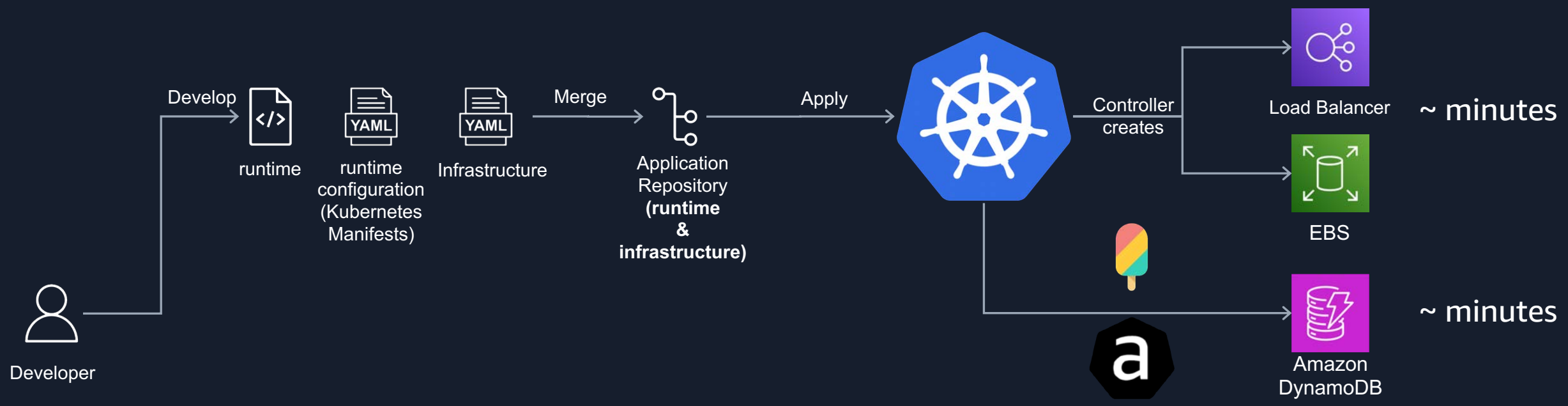
Declarative infrastructure configuration

Infrastructure Controllers – toolset

AWS Controllers for Kubernetes
Manage AWS services using Kubernetes

Control Plane Framework
Orchestrate applications and infrastructure

Kubernetes as an API

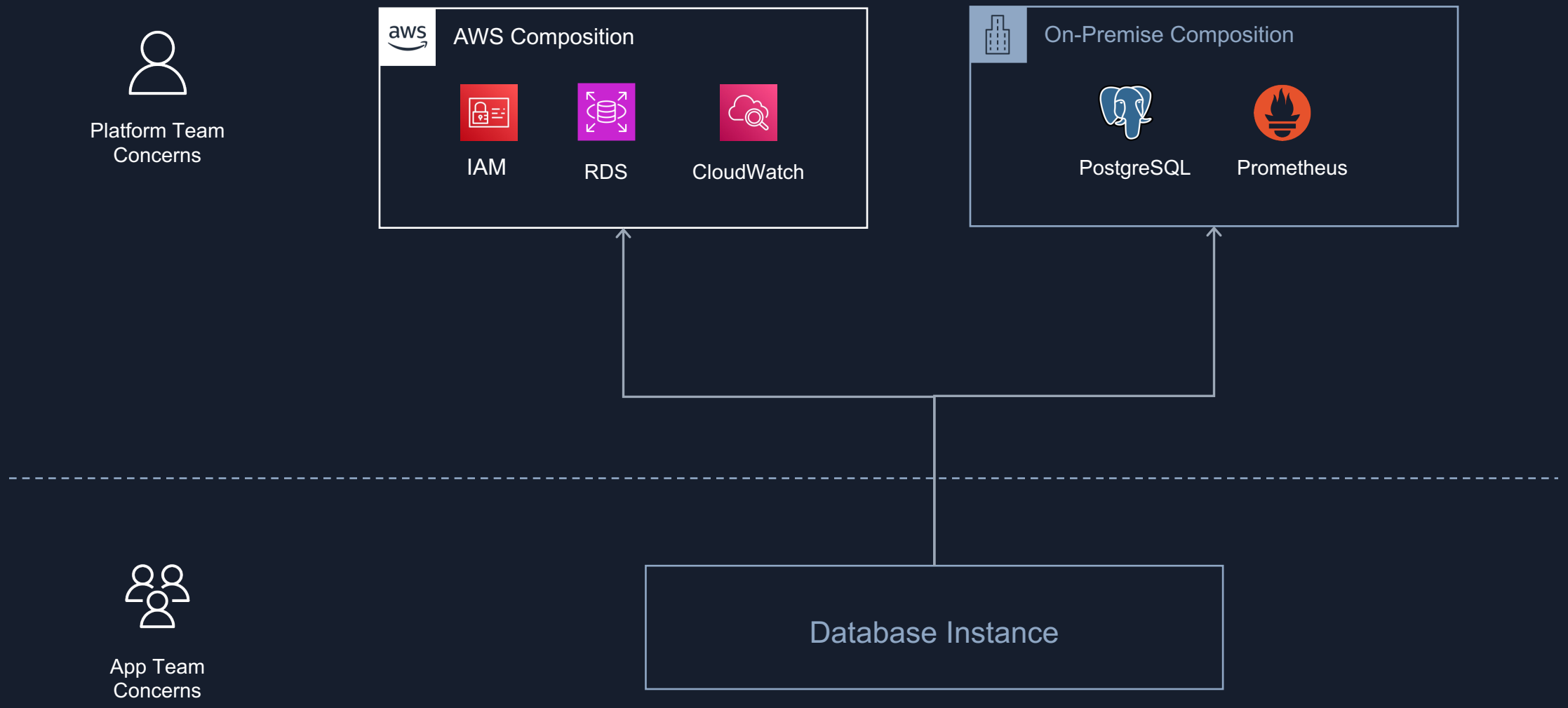


Now I have to configure my infrastructure too?

Any developer...

Team concerns – Application teams

Team concerns – Platform teams



Compositions in IaC

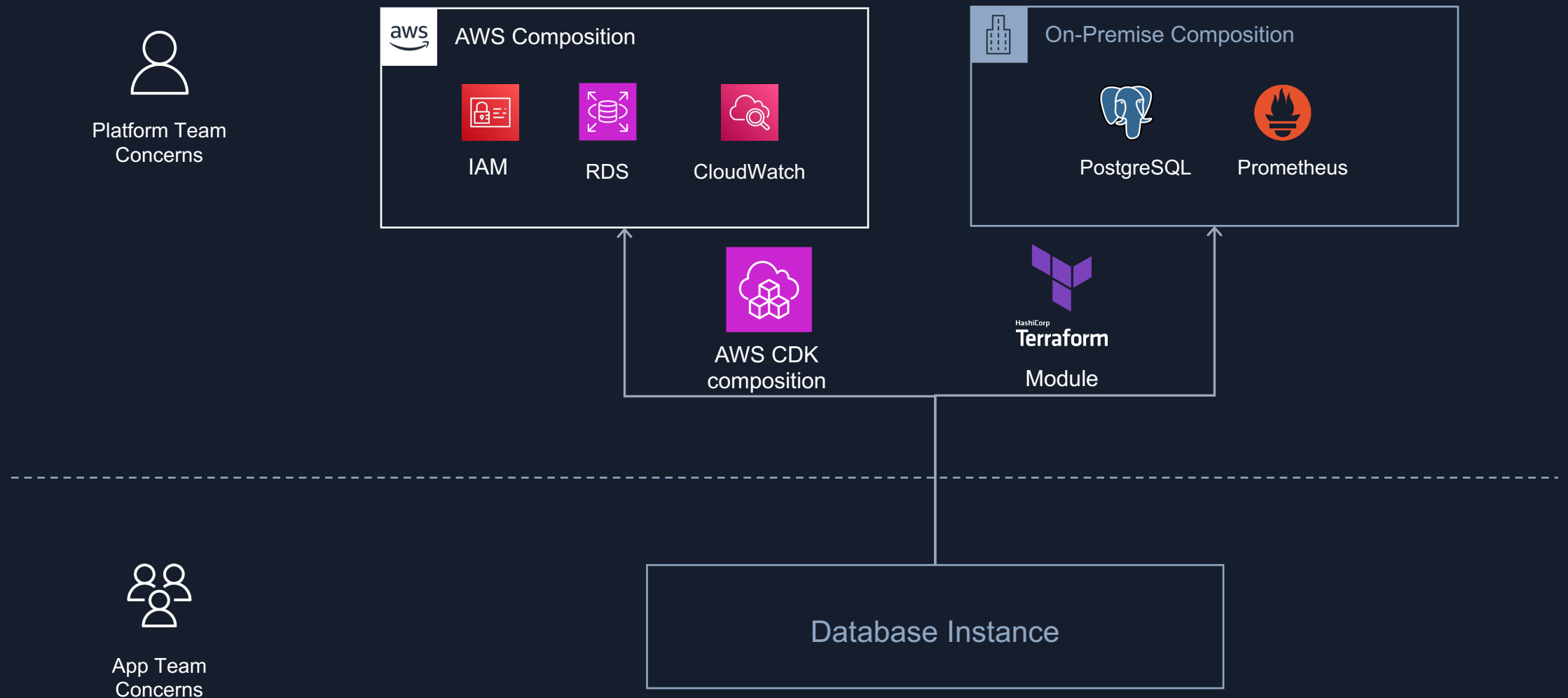
AWS CDK

"Composition is the key pattern for defining higher-level abstractions through constructs."

HashiCorp
Terraform

"Modules are containers for multiple resources that are used together"

Compositions with IaC



Compositions IaC

AWS CDK

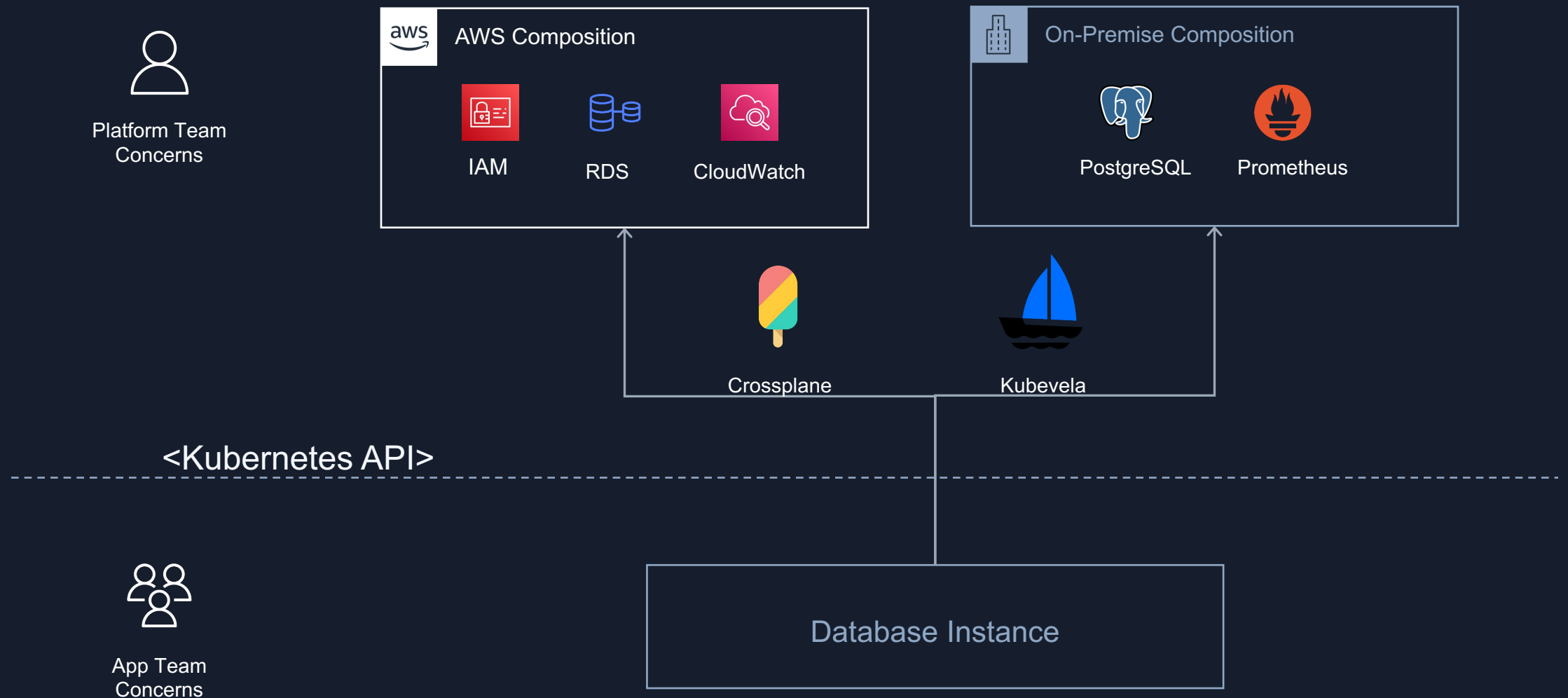
"Composition is the key pattern for defining higher-level abstractions through constructs."

HashiCorp
Terraform

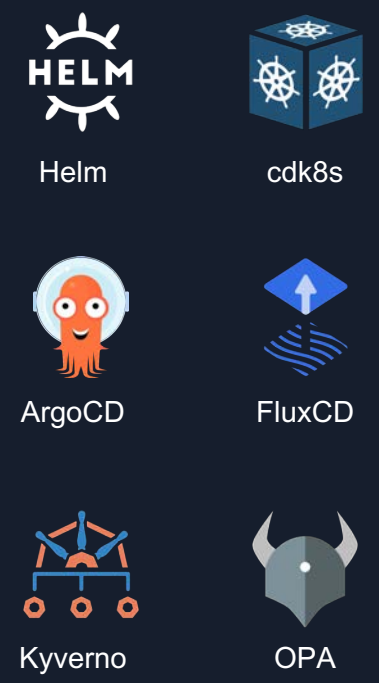
"Modules are containers for multiple resources that are used together"

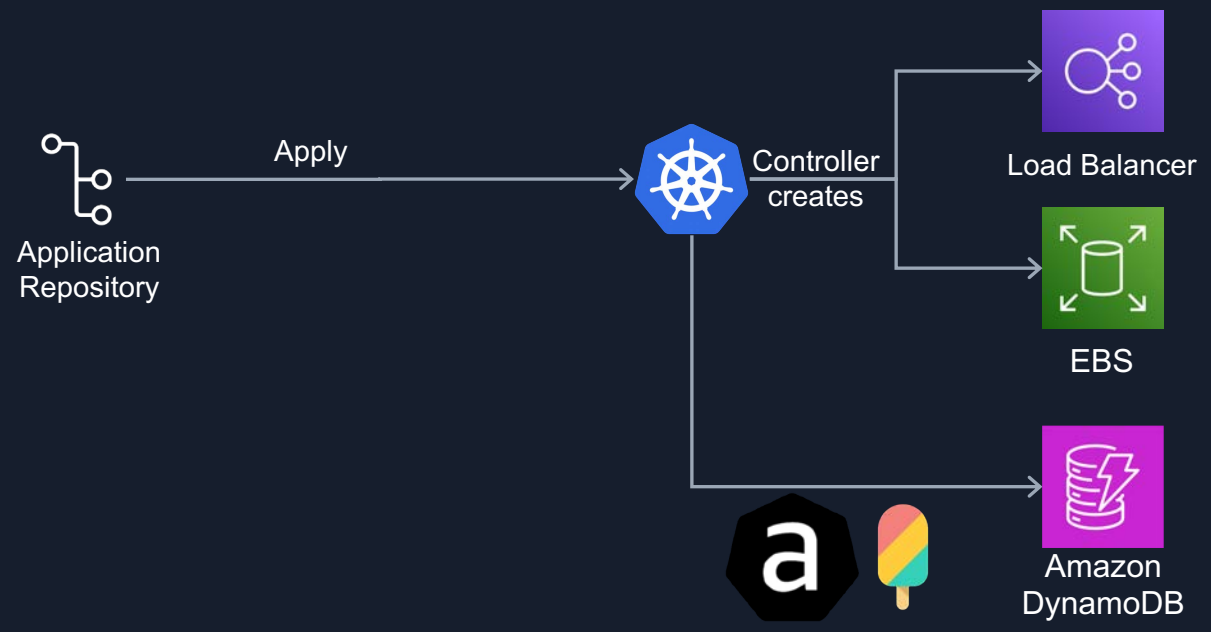
?

Compositions in Kubernetes



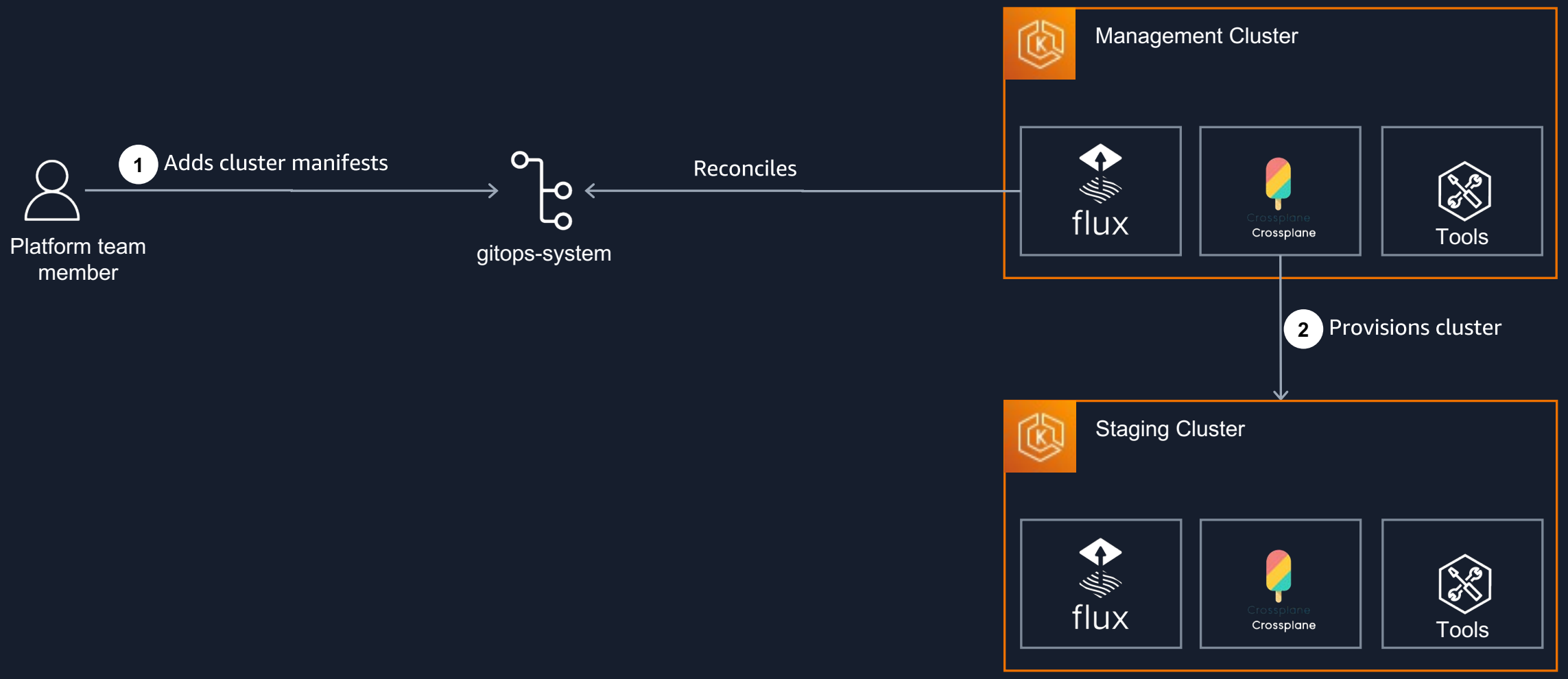
Leveraging ecosystem tooling



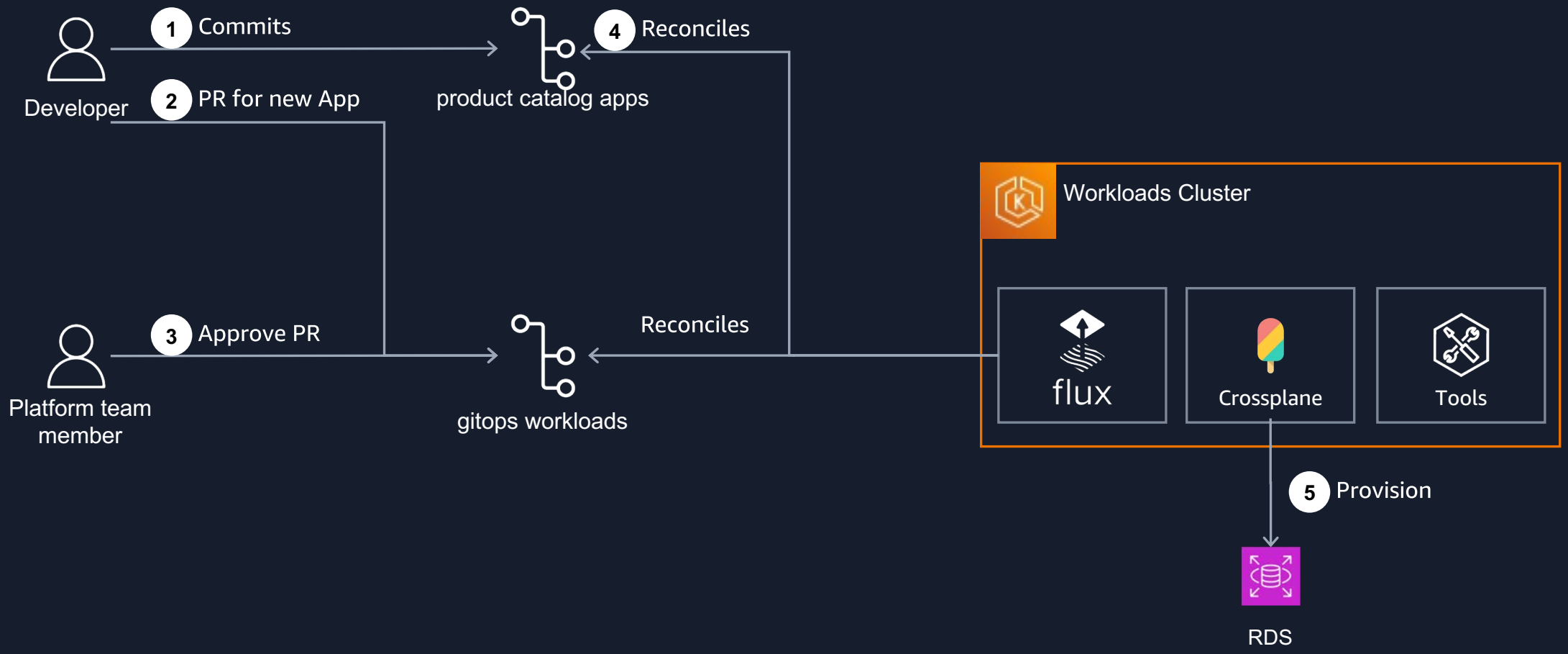


Solution Overview

Amazon EKS Cluster Provisioning



Application onboarding process



Solution

HTTPS://AWS.AMAZON.COM/BLOGS/CONTAINERS/PART-1-BUILD-MULTI-CLUSTER-GITOPS-USING-AMAZON-EKS-FLUX-CD-AND-CROSSPLANE/

Containers

Part 1: Multi-Cluster GitOps using Amazon EKS, Flux, and Crossplane

by Islam Mahgoub, Mike Rizzo, Nicholas Thomson, and Sheetal Joshi | on 07 APR 2023 | in Amazon Elastic Kubernetes Service, Containers, Technical How-To | Permalink | Share

Introduction

GitOps is a way of managing application and infrastructure deployment so that the whole system is described declaratively in a Git repository. It's an operational model that offers you the ability to manage the state of multiple Kubernetes clusters using the best practices of version control, immutable artifacts, and automation. Organizations have adopted GitOps to improve productivity, developer experience, stability, reliability, consistency, standardization, and security guarantees. Refer to the Guide to GitOps for more details about GitOps principles, patterns, and benefits.

Many AWS customers use multiple Amazon Elastic Kubernetes Service (Amazon EKS) clusters to segregate workloads that belong to different lines of business within their organizations or environments, such as production and staging, to comply with governance rules related to division of responsibilities. Platform teams in these organizations face the challenge of managing the lifecycles of these clusters in a consistent manner. Customers that adopt Amazon EKS also make use of other services, such as messaging services, relational databases, key-value stores, etc., in conjunction with their containerized workloads. It's typical for an application running on an Amazon EKS cluster to interact with managed services such as Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, and Amazon Simple Queue Service (Amazon SQS). Ideally, the lifecycles of these managed resources, including the Amazon EKS cluster, should be managed using the same GitOps declarative model used for managing applications.

Takeaways

- Packaging runtime and infrastructure can define clear responsibilities
- Infrastructure controllers extends Kubernetes capabilities for provisioning AWS Services
- Kubernetes as an API enable standardization
- Integrate with Kubernetes ecosystem
- Using GitOps for AWS managed services

Thank you!

Yuriy Bezsonov

Senior Solutions Architect,

AWS

https://www.linkedin.com/in/yuriybezsonov/

https://pulse.aws/survey/C1WYN8EP