
AWS
re:Invent

CON214

Fidelity Investments: Two years and 10,000 containers later

Ike Arias
Principal, Container Services
AWS

Niraj Amin
Director, Platform Architecture
Fidelity Investments

Rajarajan Pudupatti SJ
Director, Platform Architecture
Fidelity Investments

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Platform goals

Why have a platform strategy?

Platform architecture

Platform building blocks

Driving adoption

Lessons learned



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Fidelity Investments

1946
Fidelity Management & Research Company Founded

1965
Fidelity Purchases First Computer



1995
Fidelity Becomes an Internet Pioneer with First Mutual Fund to Create a Home Page



2016
Fidelity's First Product Application Deployed to the Cloud

2019
Fidelity Launches Multi-Cloud Hybrid Strategy

Today
2,500 Applications on Public Cloud

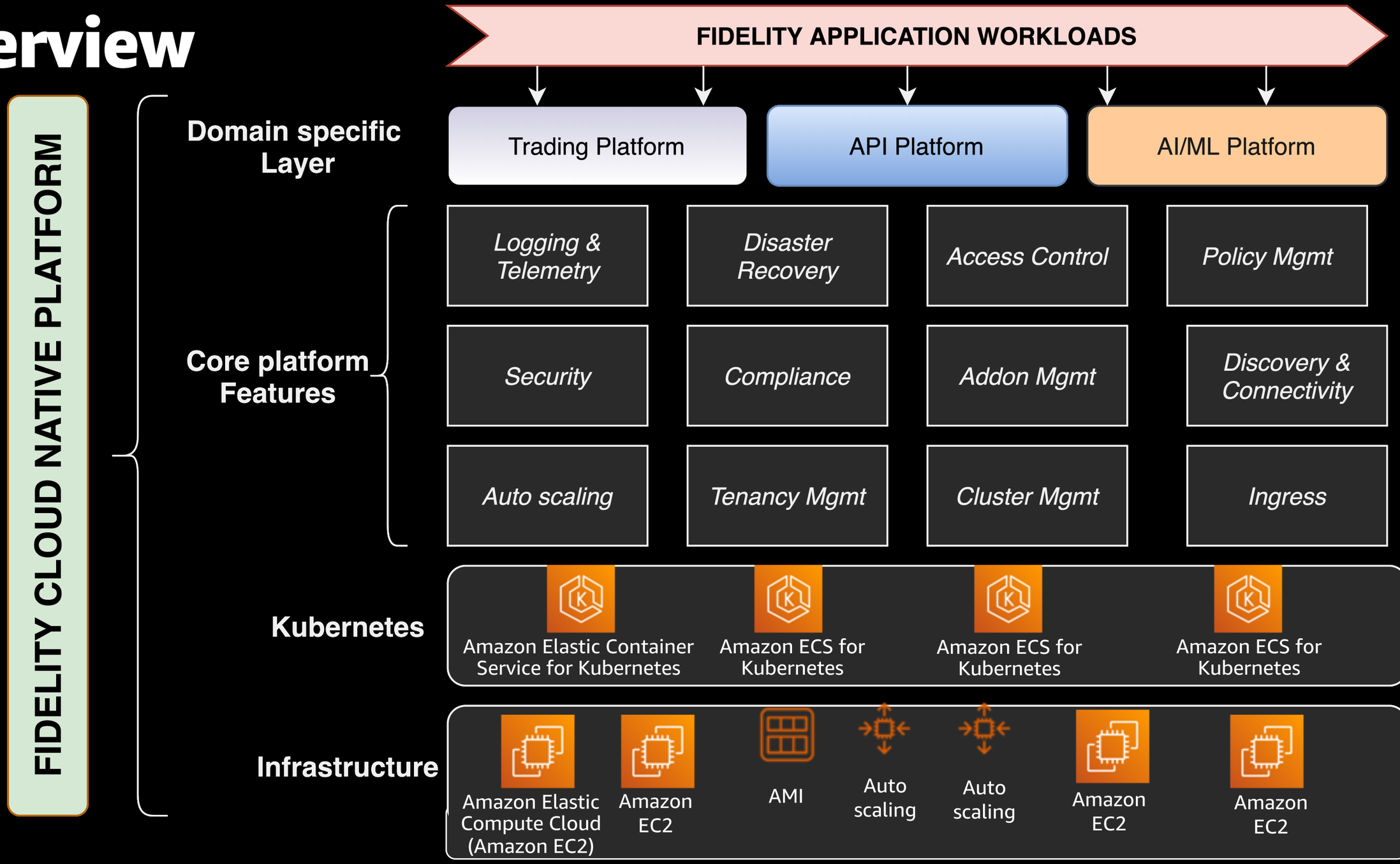
Platform goals

- Scalability and agility
  - Faster time to market, with more focus on business logic
- Innovation
  - Innovative solutions to customer problems leveraging state of the art cloud capabilities
- Reliability
  - Highly-reliable systems with on-demand compute, spread across regions for high availability
- Cost
  - Utilization-based charge back
- Transparency/Traceability
  - Full transparency/traceability on usage, assets, accesses, deployments and issues

Why have a platform strategy?

- Ability to scale to hundreds of clusters **without** compromising compliance and security
- Ability to enforce maintainable standards uniformly
- Ability to pivot
- Rolling out new changes is just a platform upgrade
- Composability and extensibility

Overview


FIDELITY CLOUD NATIVE PLATFORM
FIDELITY APPLICATION WORKLOADS
Domain specific Layer
Trading Platform
API Platform
AI/ML Platform
Core platform Features
Logging & Telemetry
Disaster Recovery
Access Control
Policy Mgmt
Security
Compliance
Addon Mgmt
Discovery & Connectivity
Auto scaling
Tenancy Mgmt
Cluster Mgmt
Ingress
Kubernetes
Amazon Elastic Container Service for Kubernetes
Amazon ECS for Kubernetes
Amazon ECS for Kubernetes
Amazon ECS for Kubernetes
Infrastructure
Amazon Elastic Compute Cloud (Amazon EC2)
Amazon EC2
AMI
Auto scaling
Auto scaling
Amazon EC2
Amazon EC2

Platform building blocks



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Kconnect

https://github.com/fidelity/kconnect/

- Connection manager command-line interface (CLI) for connecting to K8s clusters across different providers
- Simply plug-in your *enterprise credentials* and connect to clusters
- Abstracts away **kubeconfig** management
- Quickly connect to a cluster using meaningful **aliases**
- Easily plugs in to your existing CI/CD processes

Kconnect

https://github.com/fidelity/kconnect/

- kconnect use *eks* --username user --password *******
- kconnect ls
- kconnect to <ALIAS>
- e.g. *kconnect to my-eks-cluster1*

Kconnect





ENTERPRISE CREDENTIALS

kconnect CLI

1

2

3

PROVIDER INTERFACE

eks plugin

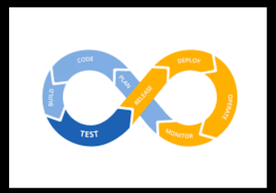
IDENTITY INTERFACE

Ping Identity

okta

Google

Microsoft ADFS



CI/CD





Amazon ECS for Kubernetes



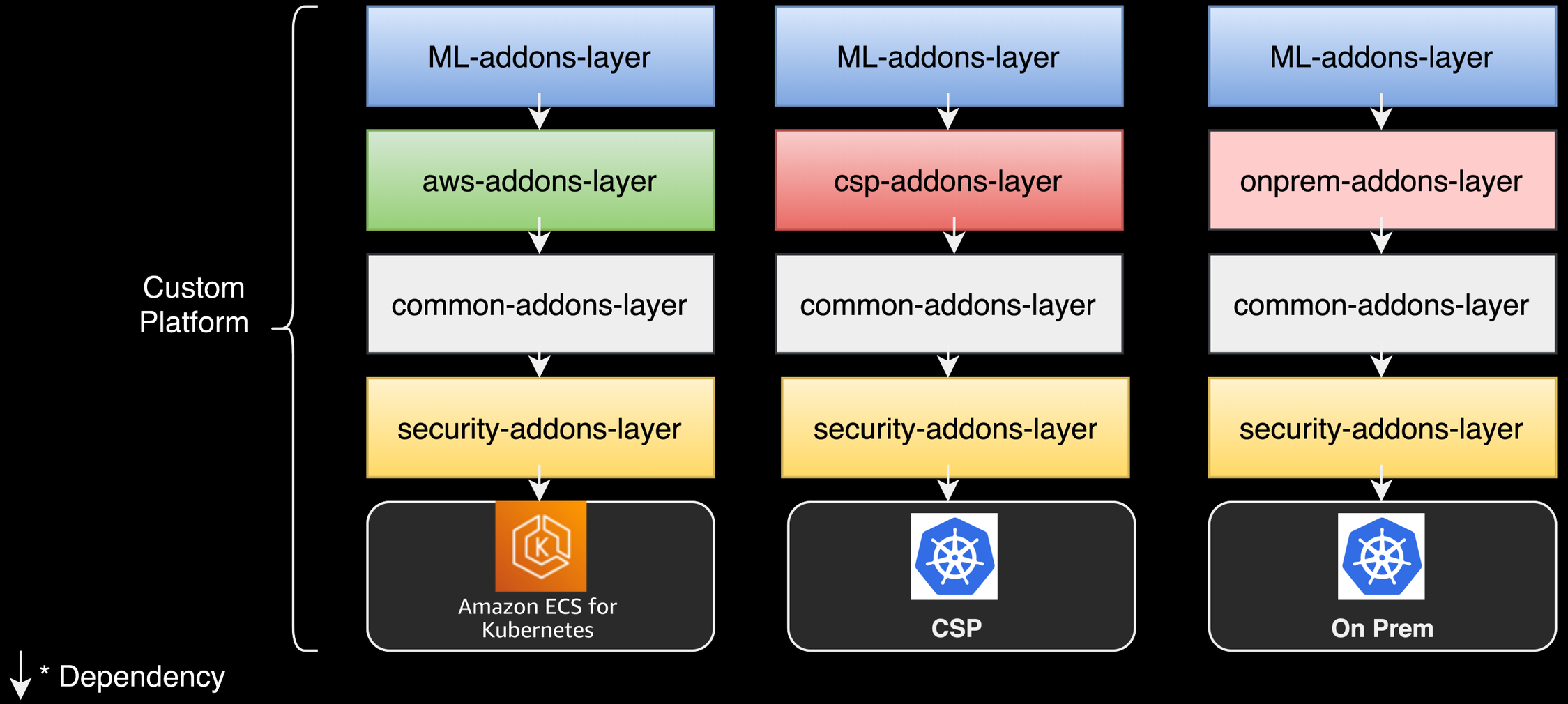
Amazon ECS for Kubernetes

Kraan – Addon Mgmt

https://github.com/fidelity/kraan/

- Open-source project, introduces the concept of “Layers”
- Implemented as a k8s controller
- Builds on top of other projects like fluxcd/toolkit
- Layers are composed of addons, represented as *HelmReleases*
- Layers can have *dependencies* established which will be honored
- Helps you share a layer across multi-cloud platform, e.g. a common security layer

Kraan – Addon Mgmt



Tenancy management

- Cluster admins have full autonomy to manage tenants in a cluster
- ‘Nsgroup’ is a k8s extension to represent a *collection of namespaces*
- Nsgroup belongs to a single tenant identity, e.g. ADGroup, IAM Role etc.
- Teams on-board simply by requesting a Nsgroup to cluster admins
- Nsgroup is bound to resource quotas and limit ranges
- Adding a namespace to an existing Nsgroup will extend existing access into that new namespace
- Nsgroup can be mapped to a CMDB object, if needed, simply by labelling it

Tenancy Mgmt



Amazon ECS for Kubernetes



* CLUSTER ADMIN

kubectl create -f nsgroups.yaml













NSGROUP-Team1

dev-namespace

dev2-namespace

qa-namespace

perf-namespace

NSGROUP-Team2

dev-namespace

dev2-namespace

NSGROUP-Team3

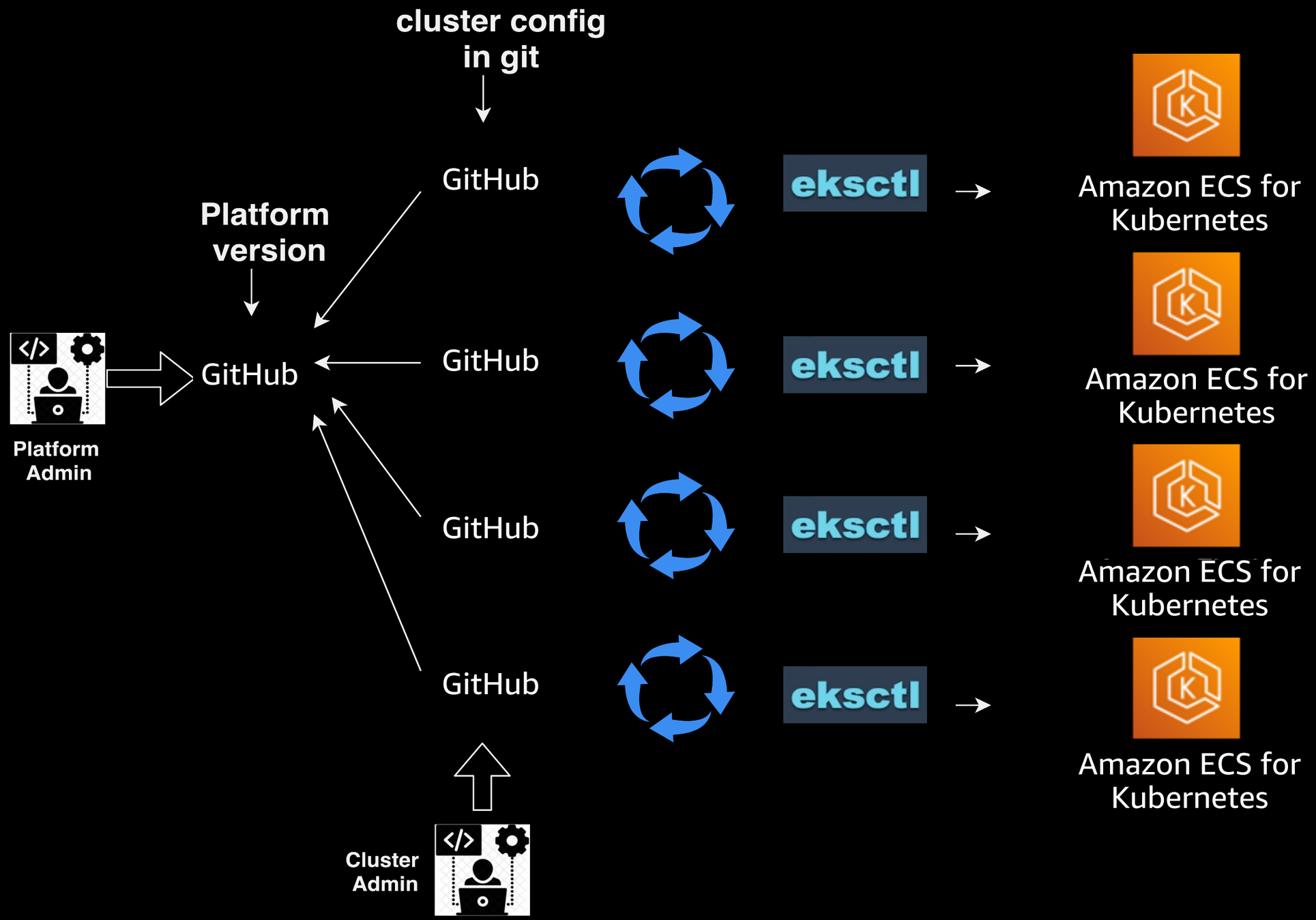
dev-namespace

dev2-namespace

perf-namespace

GitOps for Cluster Mgmt

- Platform is versioned in a git repo
- Each cluster has a git repository where its desired state is recorded
- The cluster git repo ‘points’ to specific version tags of platform repo
- Whenever a git commit to this cluster repo is detected, a backend pipeline runs ‘eksctl’ to bring the cluster to its desired state
- Layers (collections of addons) are applied to the cluster, which is processed by Kraan, which brings the addons to the desired state


cluster config
in git
GitHub
Platform
version
GitHub
Platform
Admin
GitHub
GitHub
GitHub
eksctl
eksctl
eksctl
eksctl
Amazon ECS for
Kubernetes
Amazon ECS for
Kubernetes
Amazon ECS for
Kubernetes
Amazon ECS for
Kubernetes
Cluster
Admin

Platform addons

- RBAC management
- Ingress
- Autoscaling (HPA/VPA)
- Certificate management
- DNS discovery
- Logging and telemetry
- Secret injection
- Policy management (OPA)
- DR/Backup (Velero)
- Compliance addons

Driving adoption

ENABLEMENT/COE



SRE



PLATFORM TEAM



DRIVING ADOPTION

APPLICATION TEAMS



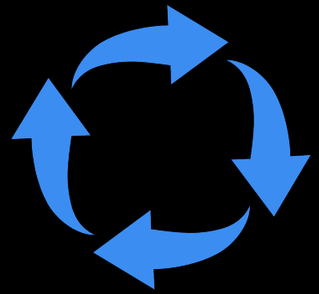






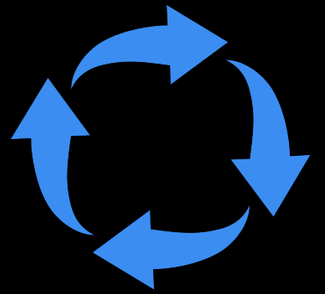


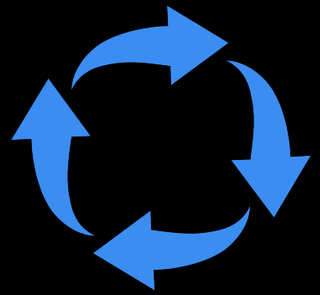




TECH FORUMS



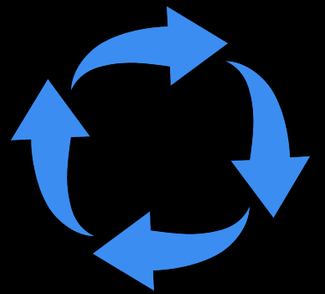




* FEEDBACK LOOP

WORKING GROUP





Lessons learned



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.