
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

PEX312-R

Enabling application development for hybrid Amazon EKS environments

Andre Boaventura

Principal Partner Solutions Architect
AWS

Sabri Alouache

Partner Solutions Architect
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Business and technical drivers

Benefits and use cases of hybrid Amazon EKS environments

Getting to know the building blocks

Architecting your environment – Demo

FAQ and discussion

Best practices – CI/CD and GitOps to the rescue

Call to action

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

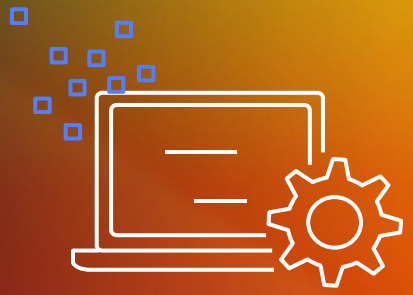
Problem statement: Customer needs



Latency constraints



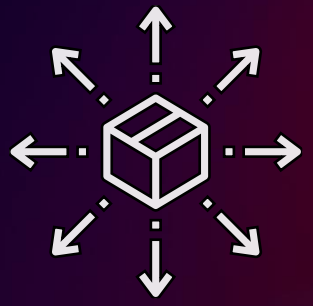
Regulatory requirements



Process data locally



Deploy where customers are



Containerize everywhere



Phased transition to cloud

aws

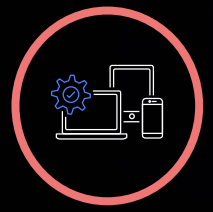
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Benefits of a hybrid Amazon EKS environment

Benefits



Leverage existing infrastructure on-premises



Simplify Kubernetes management on-premises



Reduce and optimize costs



Secure and audit your environment for compliance

Use cases



Visibility

Single plane of glass visibility



Business process

Edge computing and data movement



Control

Control access to resources



Low latency

Low-latency, local data processing/residency

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS Distro



EKS Distro

EKS Distro provides customers **version, patching, and security alignment** with Amazon EKS

- **Kubernetes open-source components used by Amazon EKS and powers EKS Anywhere**
- **Security patches, tooling, and configs that will be adopted by upstream**
- **Backport fixes to versions supported by Amazon EKS**
- **Validated to be reliable and secure**
- **EKS Distro builds are reproducible using the published tools and environment variables**

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS Anywhere

Create and operate Kubernetes in your data center and on your existing hardware using the same consistent EKS experience.

Amazon EKS Anywhere



Helps customers run the same secure, stable, and tested components used to run Amazon EKS



Built on open-source standards cluster lifecycle management tooling



Bundled with components to accelerate readiness of production clusters

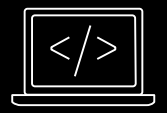


Bring your existing bare metal/ self-managed compute

Deploy and manage the lifecycle of EKS Distro clusters on-premises

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CDK accelerates development teams



Familiar

Your language
Just classes and methods



Tool support

Compatible with testing
Frameworks you're already using



Abstraction

Sane defaults
Reusable classes

```
class UrlShortener extends Stack {
  constructor(scope: App, id: string, props?: UrlShortenerProps) {
    super(scope, id, props);

    const vpc = new ec2.Vpc(this, 'vpc', { maxAzs: 2 });
    const cluster = new ecs.Cluster(this, 'cluster', { vpc: vpc });
    const service = new patterns.NetworkLoadBalancedFargateService(this, 'sample-app', {
      cluster,
      taskImageOptions: {
        image: ecs.ContainerImage.fromAsset('ping'),
      },
      dom
    });
    // Setup AutoScaling policy
    const scaling = service.service.autoScaleTas
    scaling.scaleOnCpuUtilization('CpuScaling',
      targetUtilizationPercent: 50,
      scaleInCooldown: Duration.seconds(60),
      scaleOutCooldown: Duration.seconds(60)
    });
  }
}
```

domainName
domainZone

(property) patterns.NetworkLoadBala
ncedServiceBaseProps.domainName?: s
tring | undefined

The domain name for the service, e.g.
"api.example.com."

@default

- No domain name.

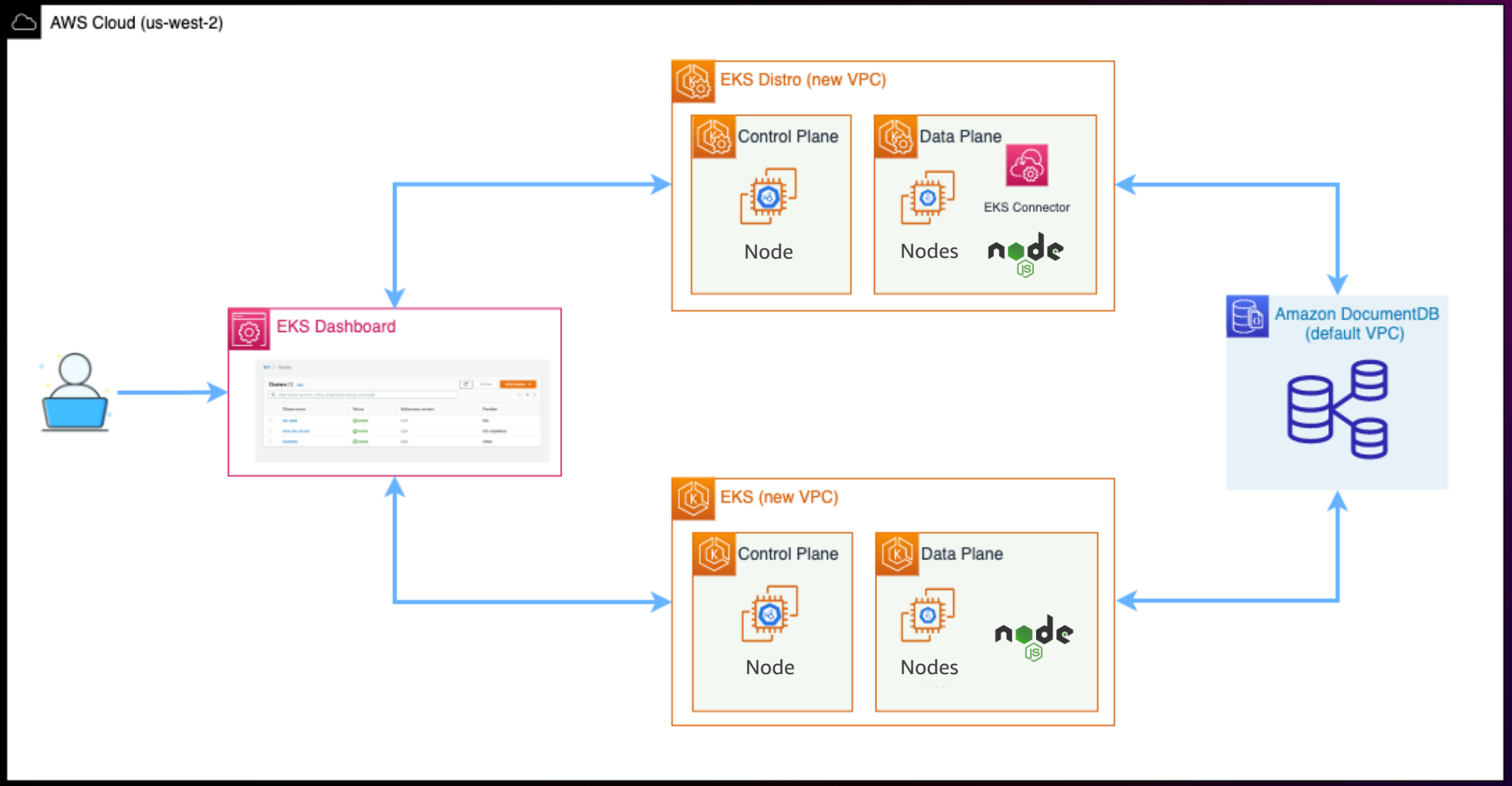






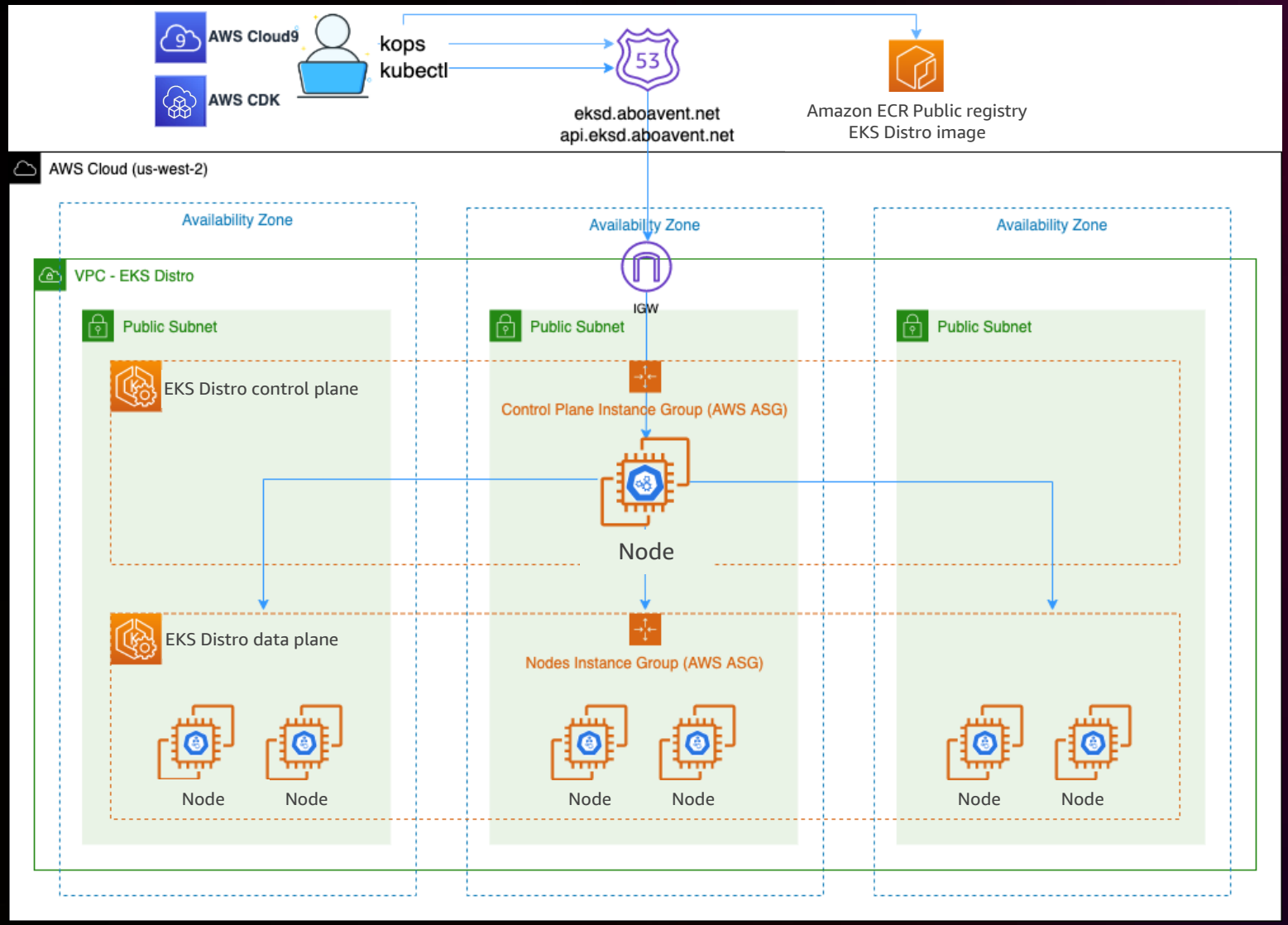

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture overview

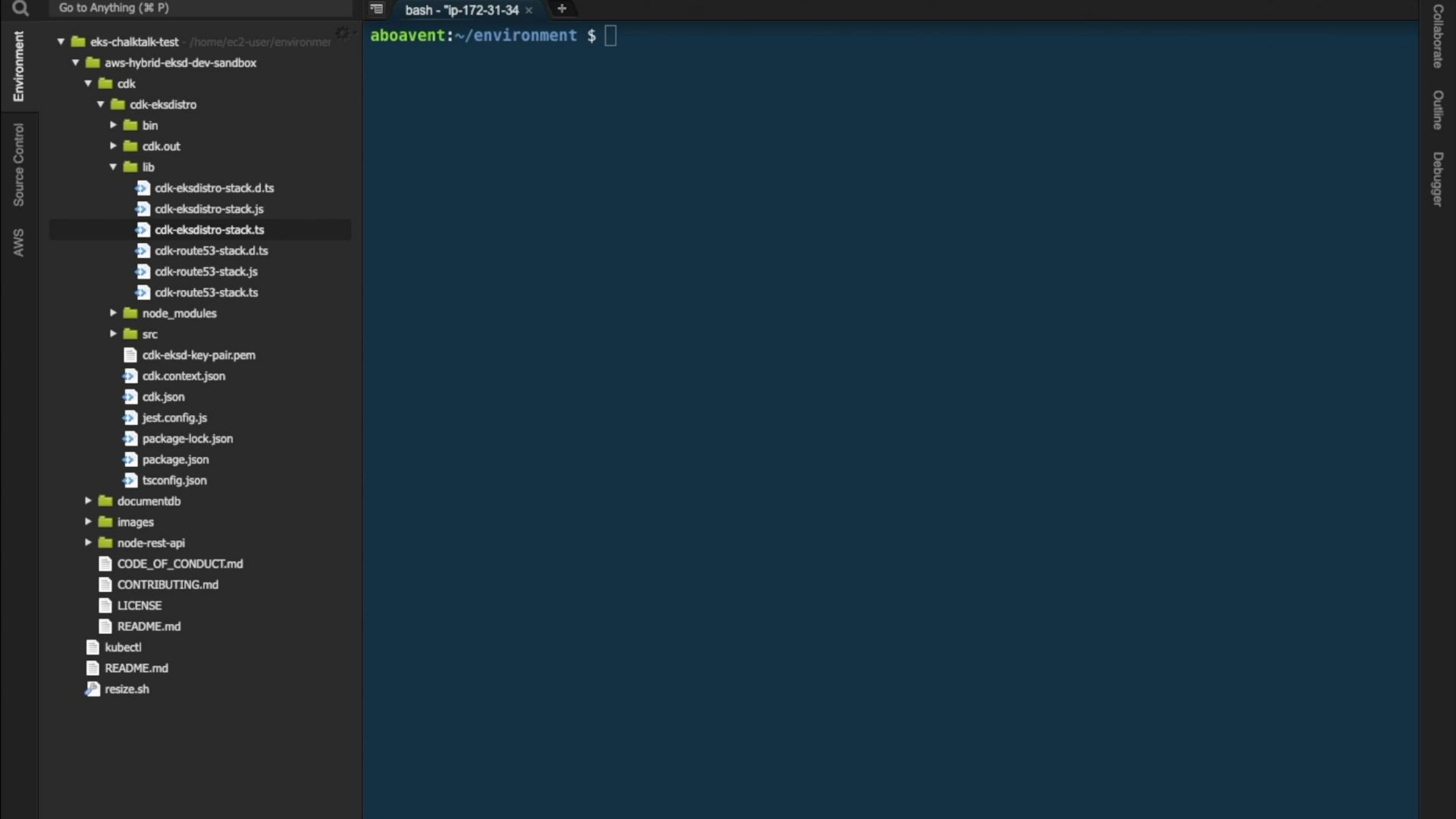


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture deep dive



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.


Go to Anything (⌘ P)
Environment
Source Control
AWS
eks-chalktalk-test - /home/ec2-user/environment
aws-hybrid-eksd-dev-sandbox
cdk
cdk-eksdistro
bin
cdk.out
lib
cdk-eksdistro-stack.d.ts
cdk-eksdistro-stack.js
cdk-eksdistro-stack.ts
cdk-route53-stack.d.ts
cdk-route53-stack.js
cdk-route53-stack.ts
node_modules
src
cdk-eksd-key-pair.pem
cdk.context.json
cdk.json
jest.config.js
package-lock.json
package.json
tsconfig.json
documentdb
images
node-rest-api
CODE_OF_CONDUCT.md
CONTRIBUTING.md
LICENSE
README.md
kubectl
README.md
resize.sh
bash - "ip-172-31-34
aboavent:~/environment $
Collaborate
Outline
Debugger

FAQ and discussion

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

FAQ and discussion

What is the main element when implementing a hybrid environment?

Which approach is best when implementing a hybrid environment – declarative or imperative?

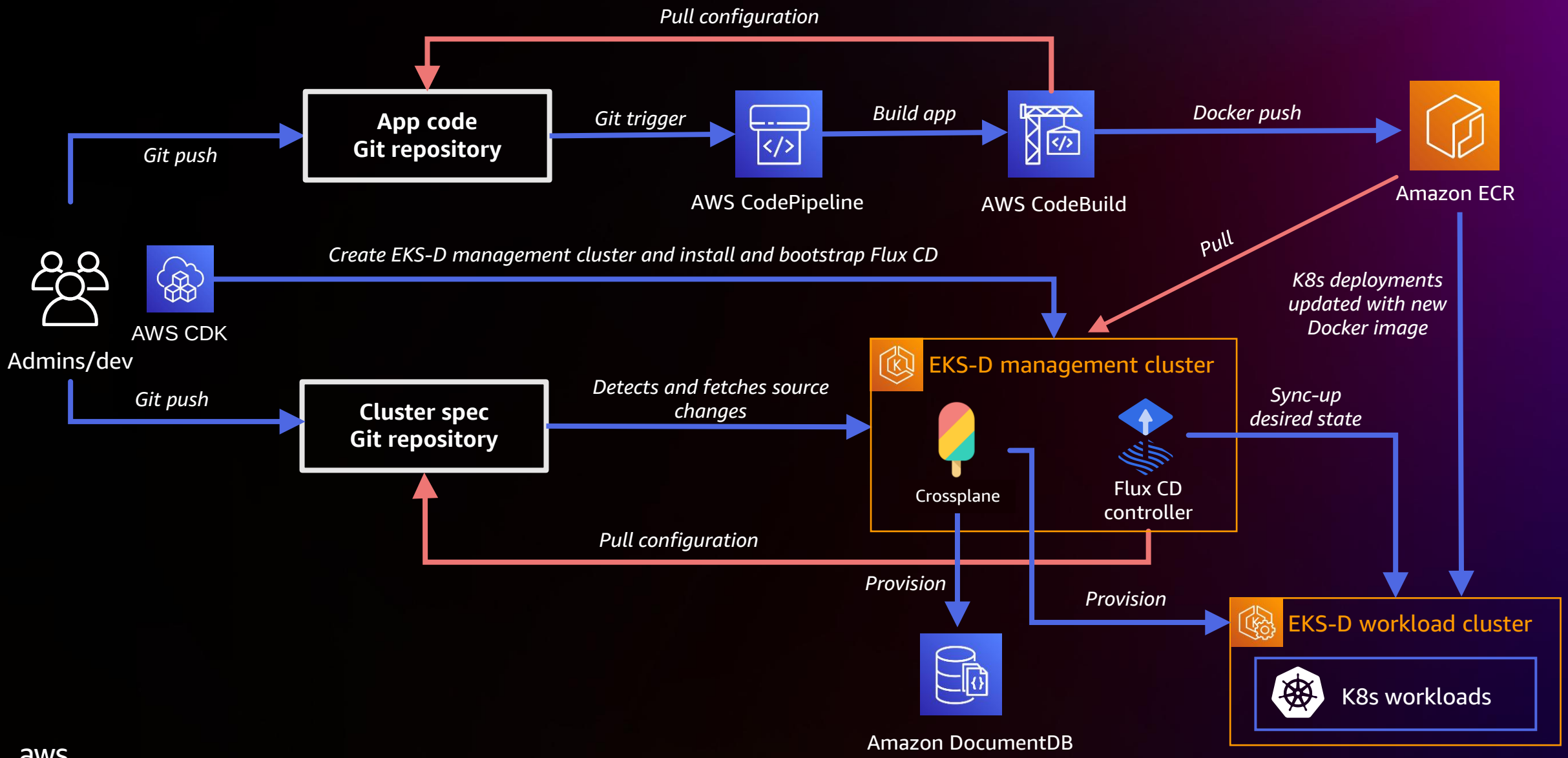
What about cost? Will it cost me more to run a hybrid environment with EKS-D and EKS Anywhere?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Best practices

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

GitOps model for hybrid EKS dev environments



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

GitOps compared to traditional CD systems

Traditional CD systems

- Push on an edge-triggered event
- Deploy only the difference
- The system may drift between deployments
- CD process must have access to deployment environment

GitOps-based systems

- The system continuously polls for changes
- The entire system is declared for any deployment
- The system will correct any drift
- Pipelines run within the system in-scope authorization

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

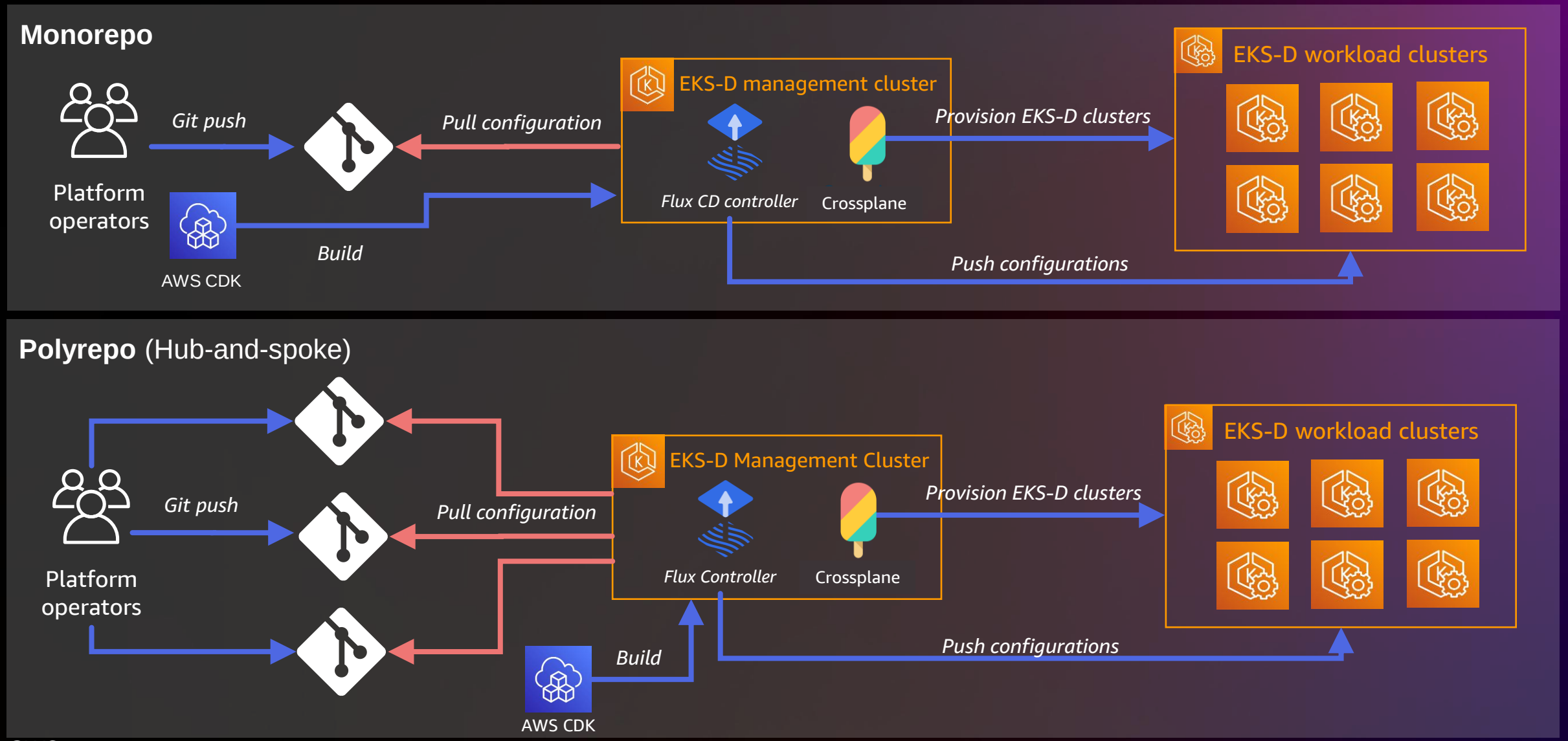
GitOps operators

- Enforce the cluster configuration defined by the platform team in Git
- Prevent, detect, and remediate misconfiguration
- Deploy and manage cluster applications



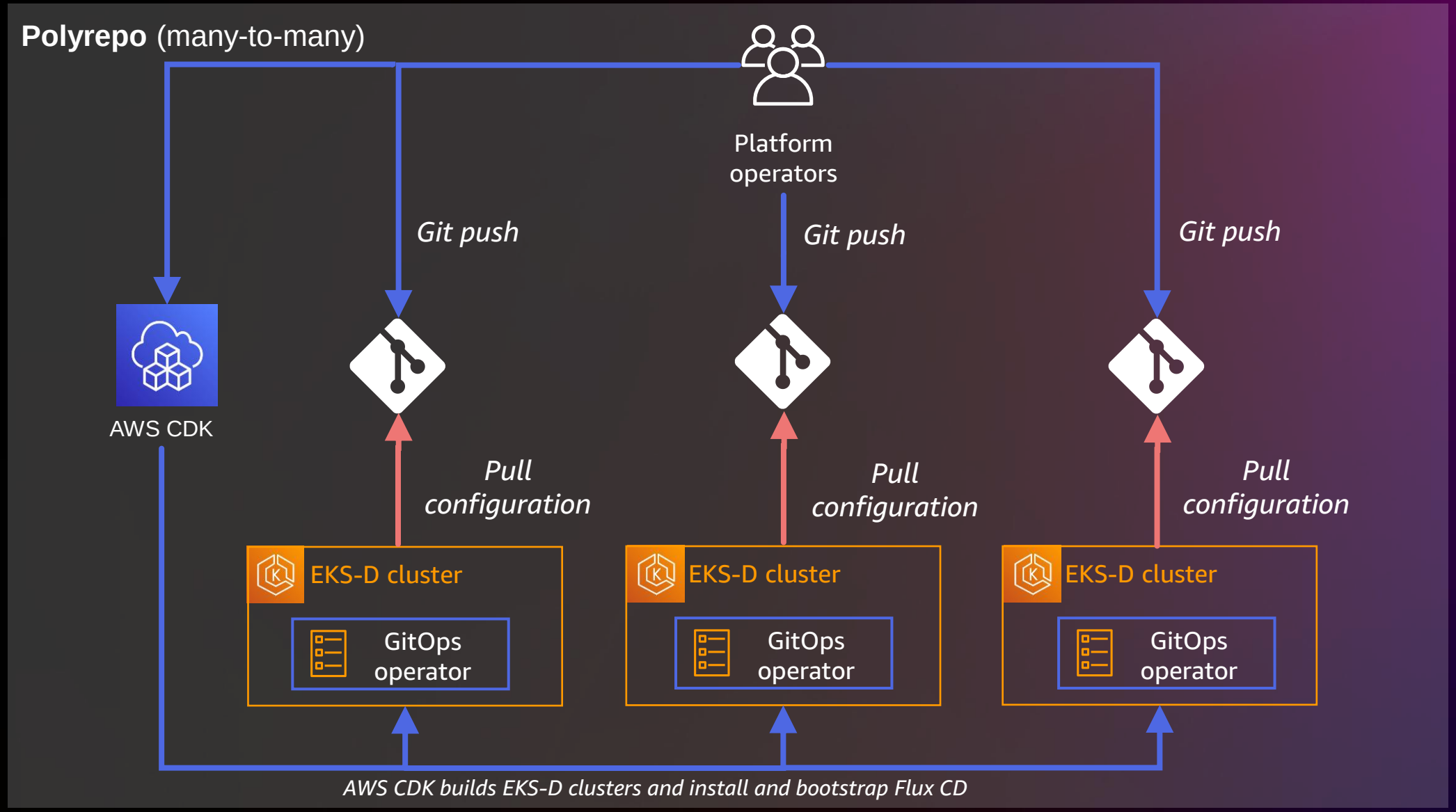
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

GitOps repository structure considerations



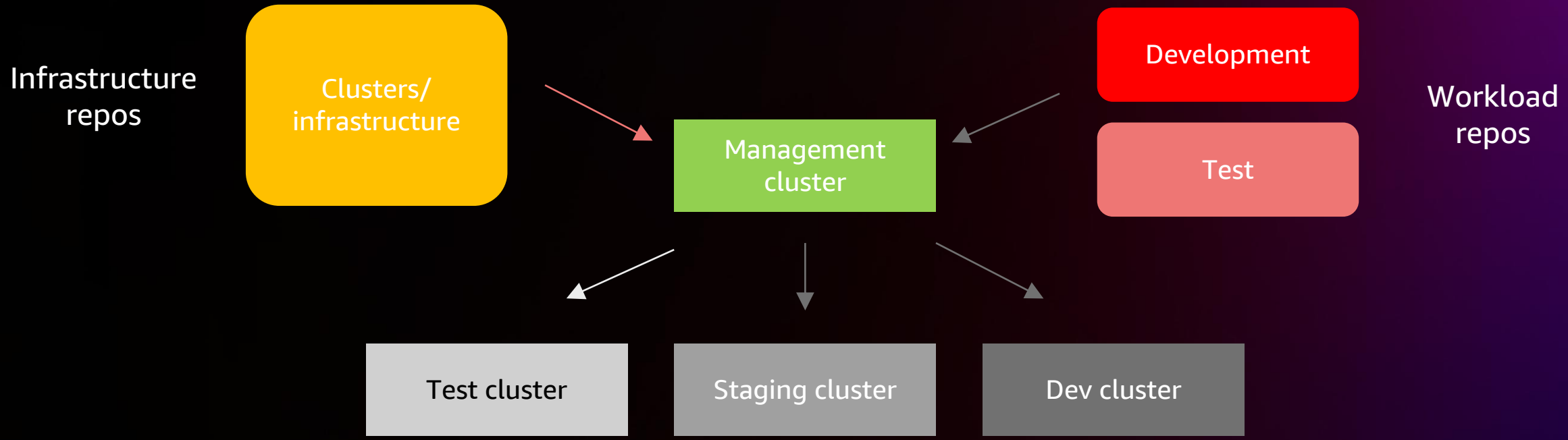
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

GitOps across multiple development clusters



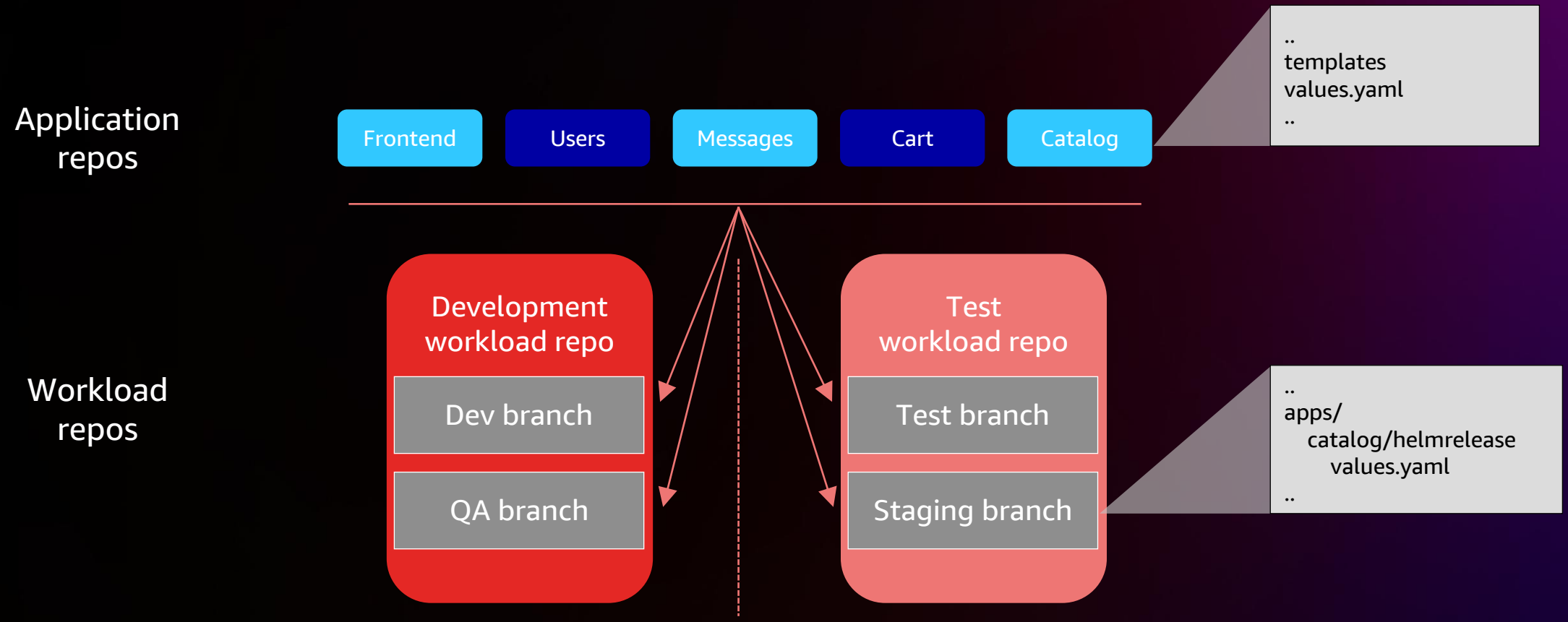
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Getting organized with repositories



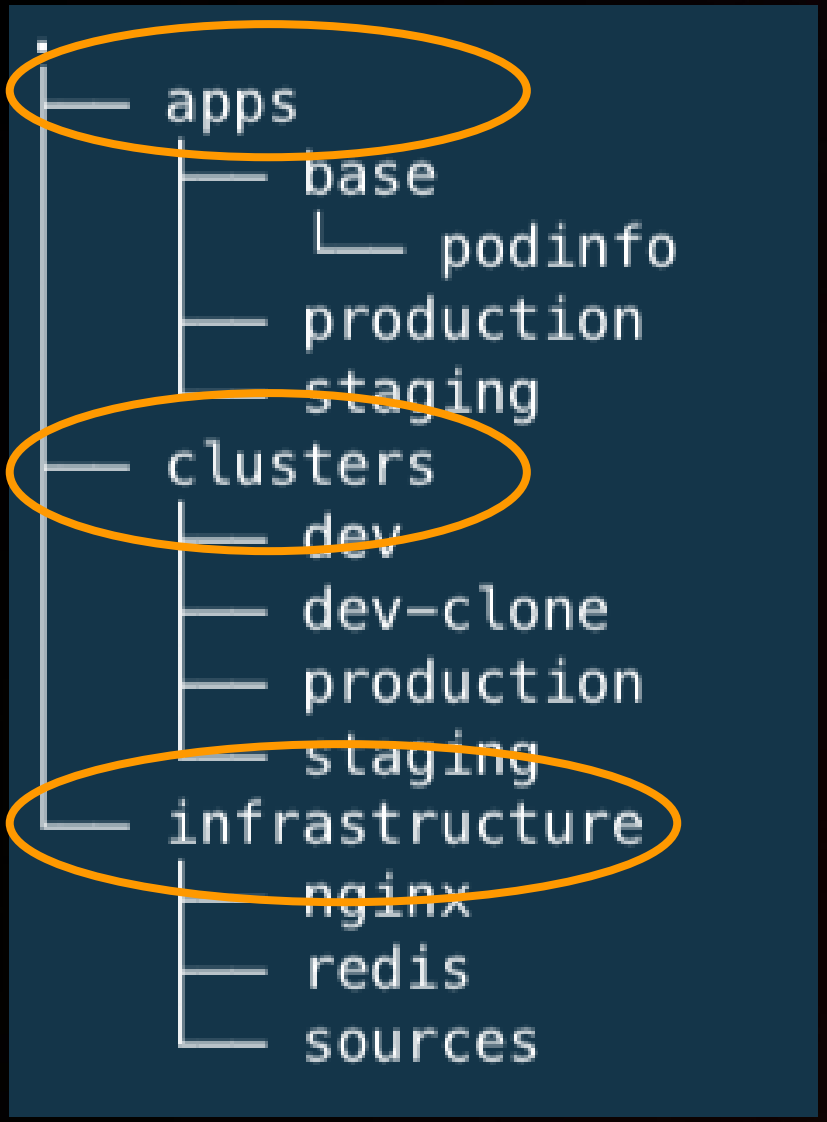
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Where’s the source?



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Directories for cluster repositories



Helm releases with a custom configuration per cluster

Flux configuration per cluster

Common infra tools (NGINX, Redis, etc.) and Helm repository definitions

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

GitOps for hybrid EKS

- GitOps techniques are key enablers of hybrid and cloud software delivery, allowing a seamless experience for building k8s environments with EKS-D, targeting the development of applications which can be deployed in both the cloud and on-premises

- GitOps as an operating model, together with EKS Distro, provides a unique platform that simplifies the deployment of Kubernetes clusters across environments and infrastructure providers

- Adoption path
  - Declarative infrastructure, not reconciled, potentially imperative
  - GitOps for workloads, with automated reconciliation and drift detection
  - Pull-based deployments from source, Helm charts, images, and so on
  - Focus on visibility and socializing changes; approval workflows get refined

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Key takeaways

- Design repositories and integration with GitOps based upon your organization's roles and responsibilities
- Infrastructure teams can focus on application deployment and scaling, and developers can focus on the application code
- Ideally, changes to the system involve a configuration change in one place
- Observability is critical, as always

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Key takeaways

- Predictable and uniform change throughout the system comes from well-organized changes
- Use templates to create modular templates for both workloads and infrastructure (Kustomize, Helm, etc.)
- Leverage Crossplane to implement the “cloud-native” way to do GitOps. Cluster API and AWS Controllers for Kubernetes (ACK) are very flexible options as well
- Configuration drift is detected and response is configurable and programmatic

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Call to action

WHAT'S NEXT

- Check the Amazon EKS Distro (EKS-D) repository
- Check the other session about hybrid use case of Amazon EKS
  - *Application modernization using Amazon EKS Blueprints for AWS CDK*
    - PEX303-R
  - *Learn to deploy containerized applications in hybrid cloud environments*
    - CON304-R
  - *Deploy containerized applications in a hybrid cloud environment*
    - CON302

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Andre Boaventura
aboavent (GitHub)

Sabri Alouache



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.