CON421-R

Amazon EKS under the hood

Eswar Bala
Sr. Software Development Manager
Amazon Web Services

Richard Sostheim
Principal Engineer
Amazon Web Services

Ahmed El Baz
Software Engineer
Snap Inc

re:Invent

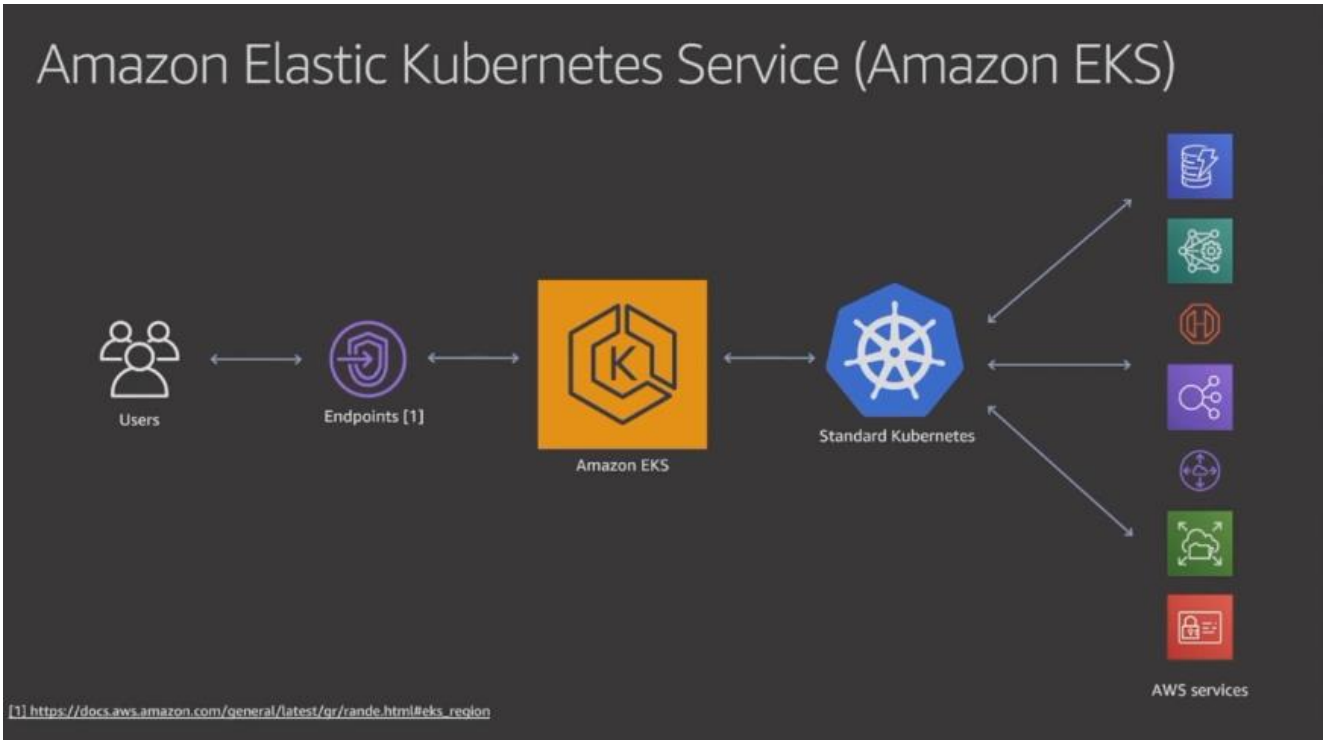
aws

Kubernetes allows you to run containerized workloads and services using declarative configuration and automation. Amazon Elastic Kubernetes Service (Amazon EKS) is a managed service that makes it easy to run Kubernetes on AWS without needing to stand up or maintain your own Kubernetes clusters. Amazon EKS takes care of the undifferentiated heavy lifting around securing, patching, qualifying, and upgrading Kubernetes clusters. Join us for a look under the hood at how Amazon EKS manages Kubernetes at scale. We also discuss some of the key design decisions in building out the infrastructure to manage one of the industry's fastest-growing open-source projects.

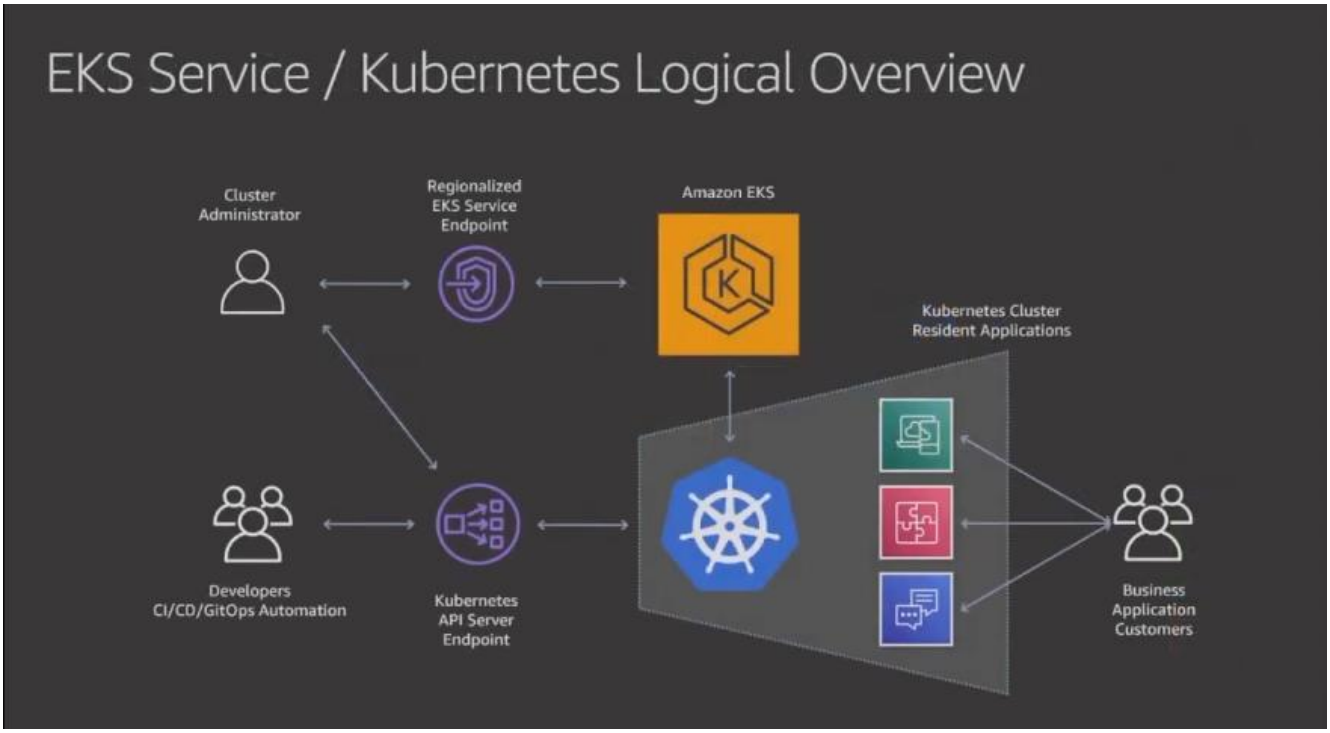
Agenda

- Amazon EKS architectural overview
- Amazon EKS under the hood
- Amazon EKS operations
- Amazon EKS enhancements
- Snap Service Mesh

Amazon EKS architectural overview

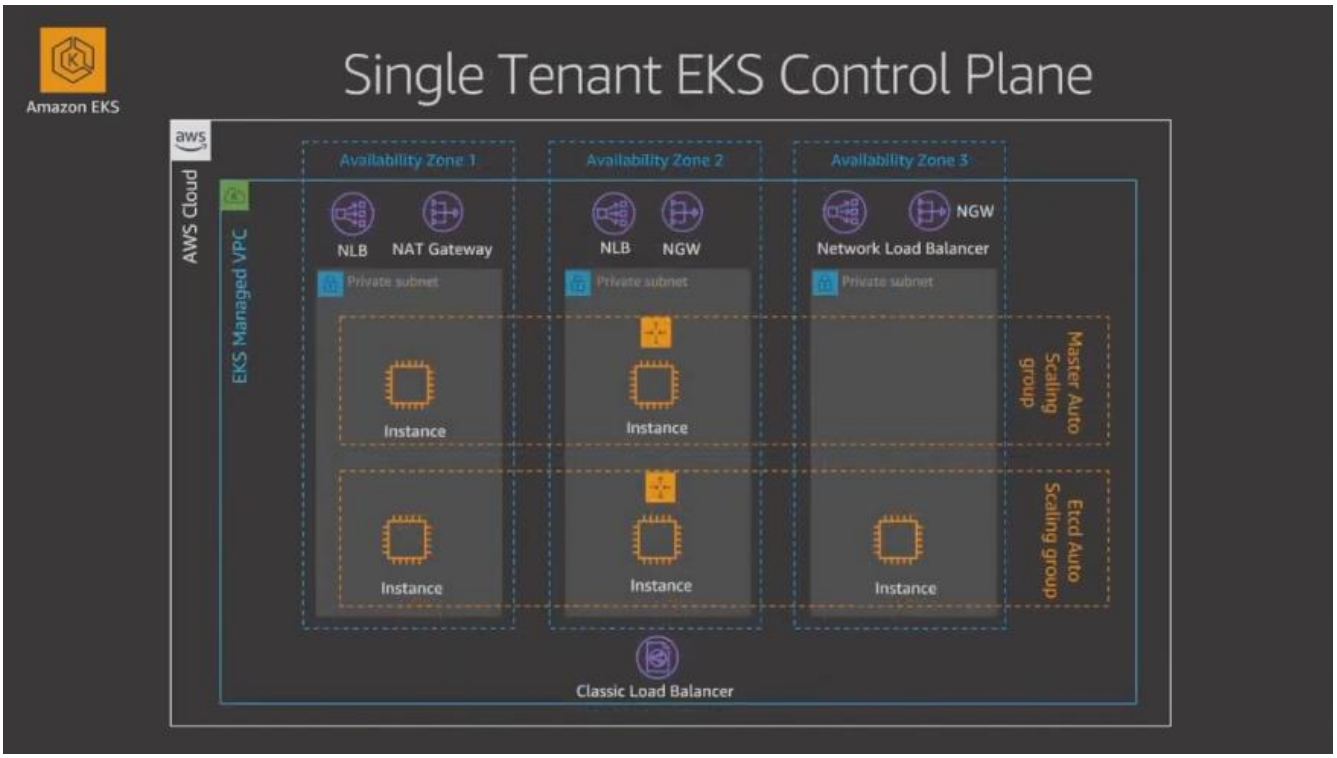


EKS is a regional service with regional endpoints that customers can connect to as their control plane. AWS will create a cluster for you to deploy and run your apps in.



The cluster admins tend to use ***kubectl*** while the developers tend to deploy their apps via CI/CD pipelines, the business end users tend to use the application endpoint URLs to access and use the apps.

Customers can pick the nodes to deploy into the Customer Managed VPC that connects with the EKS Managed Control Plane created for the cluster. Instances are spread over a minimum of 3 AZs.



EKS Cellular Architecture

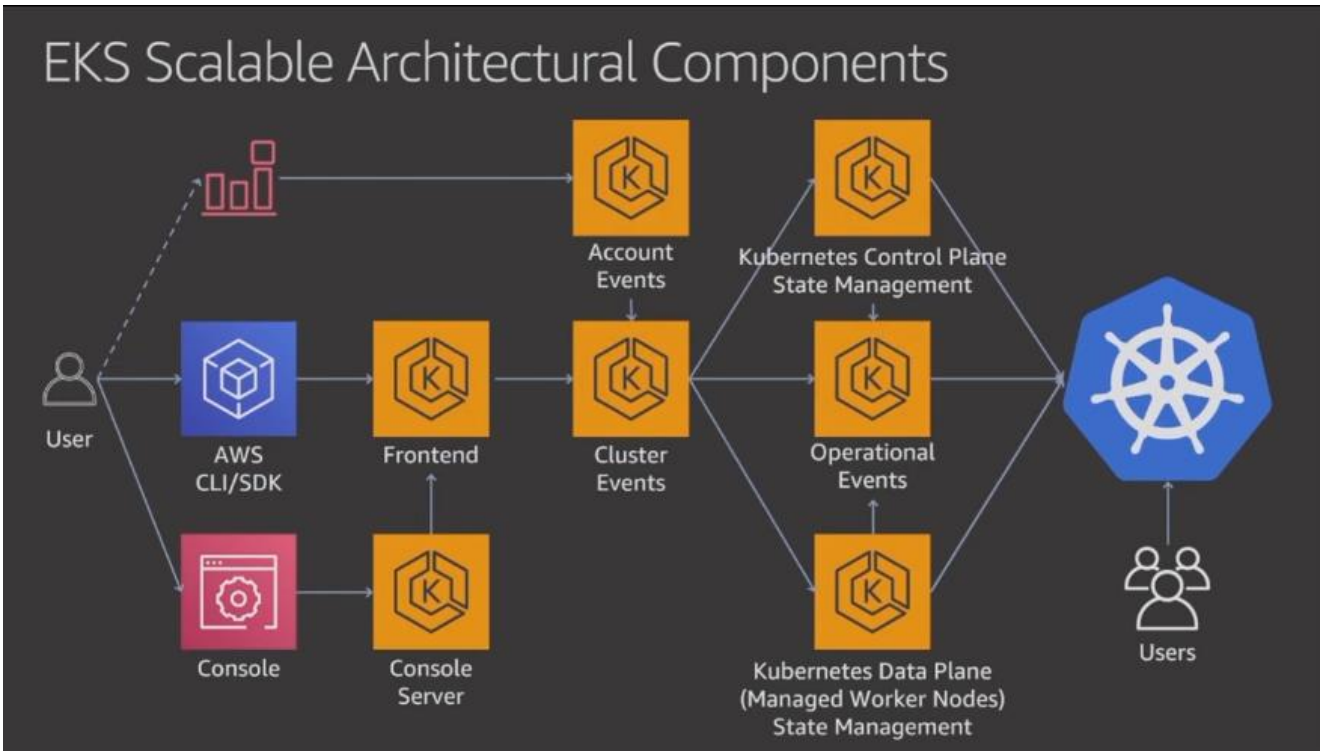
EKS Service Failure Domains – isolated failure domains designed to limit the blast radius of events

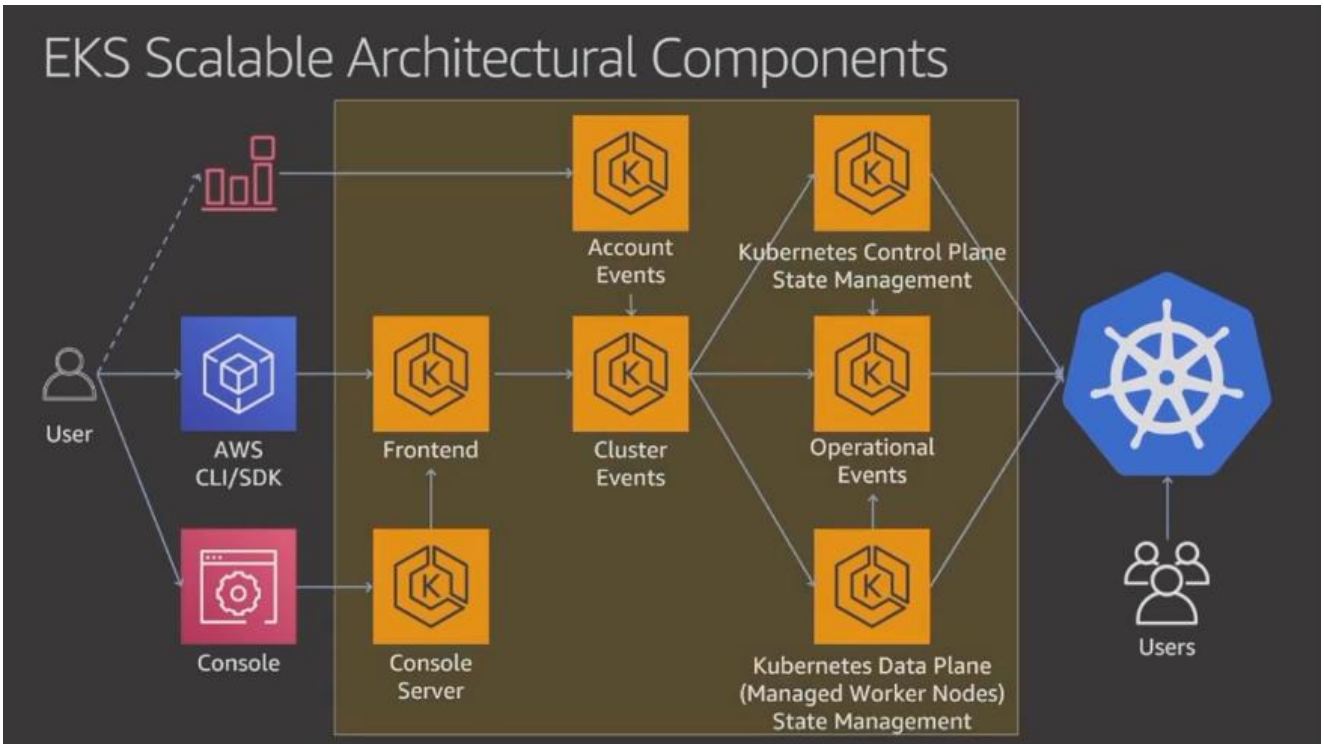
- Region - top level of isolation
  - Force majeure, hurricane, asteroid (space junk), earthquake, other significant event
- Availability Zone – subdivides region geographically
  - Localized event, natural disasters, lightning, tornado, power grid failure, civil unrest
- AWS Account – subdivides region by resource ownership
  - Security isolation, limit management, load partitioning (shard)

1 cell = 1 AWS account

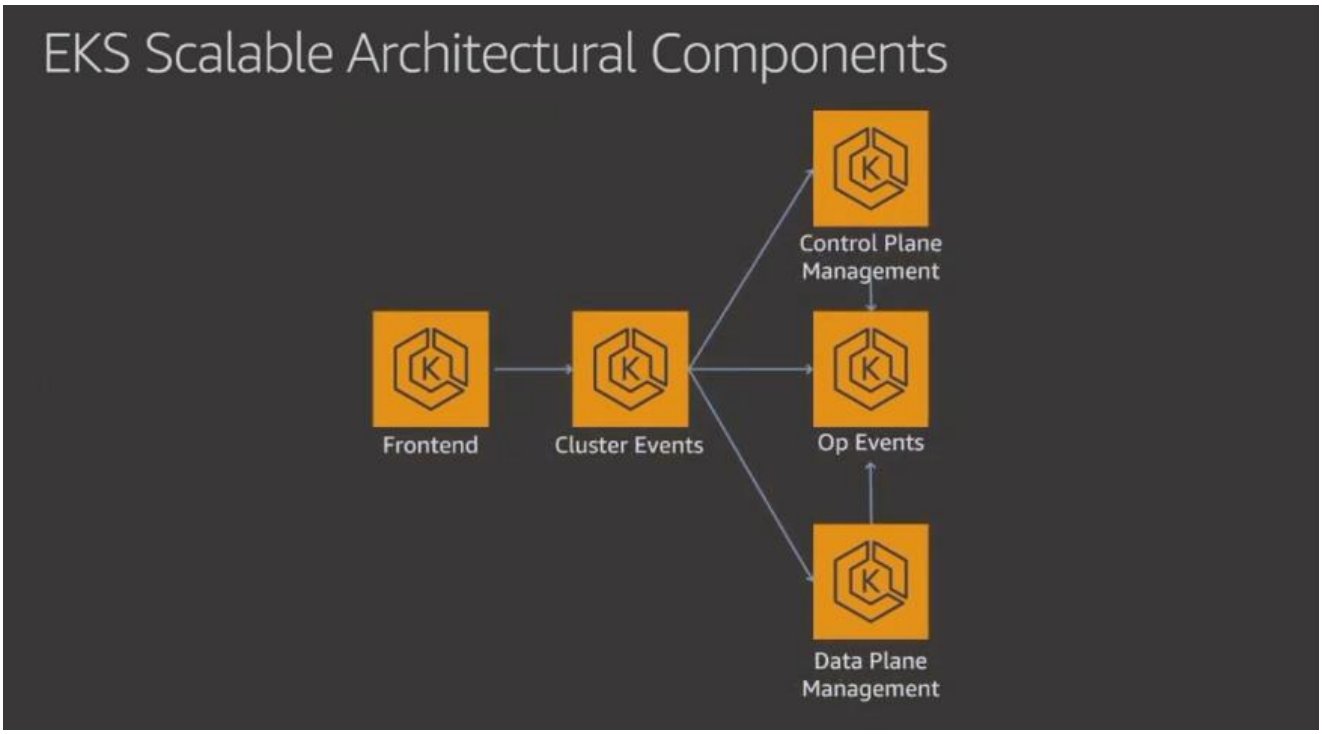
EKS Logical Single Highly Available Service

You can access your EKS interface in one of 2 ways

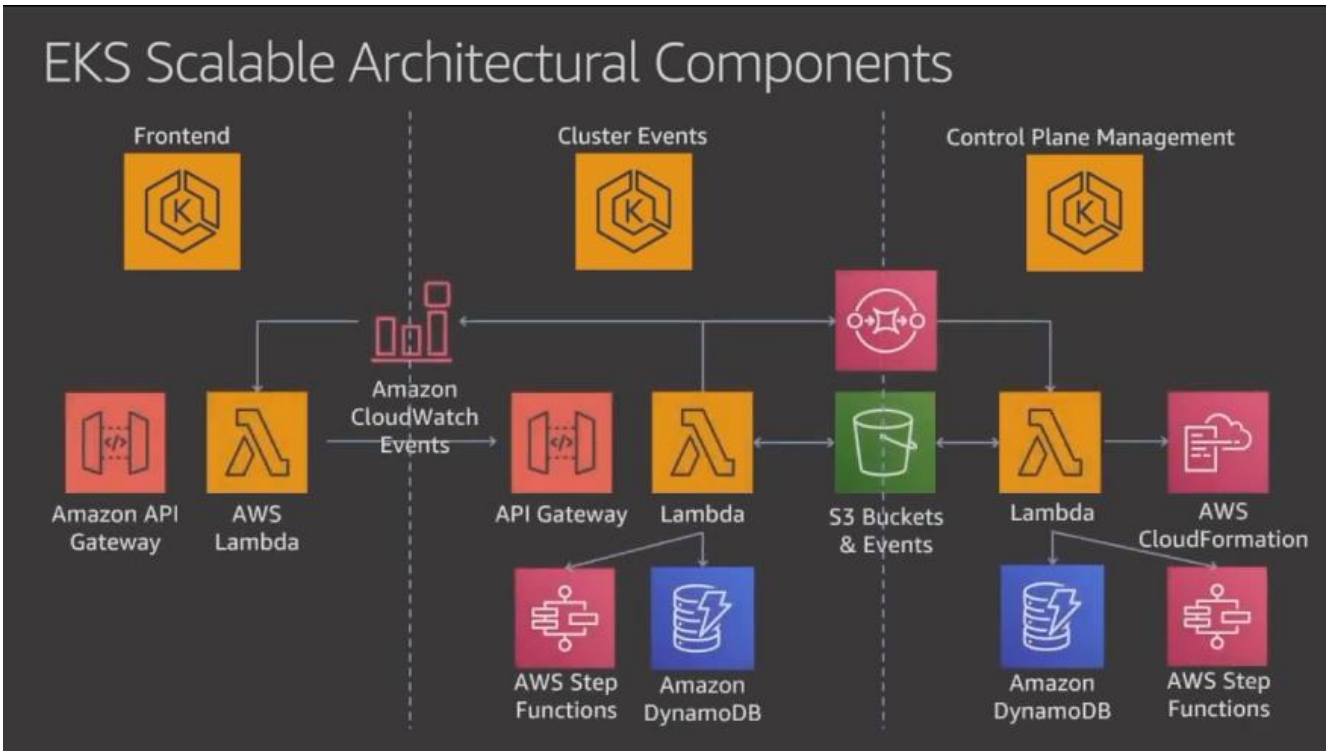




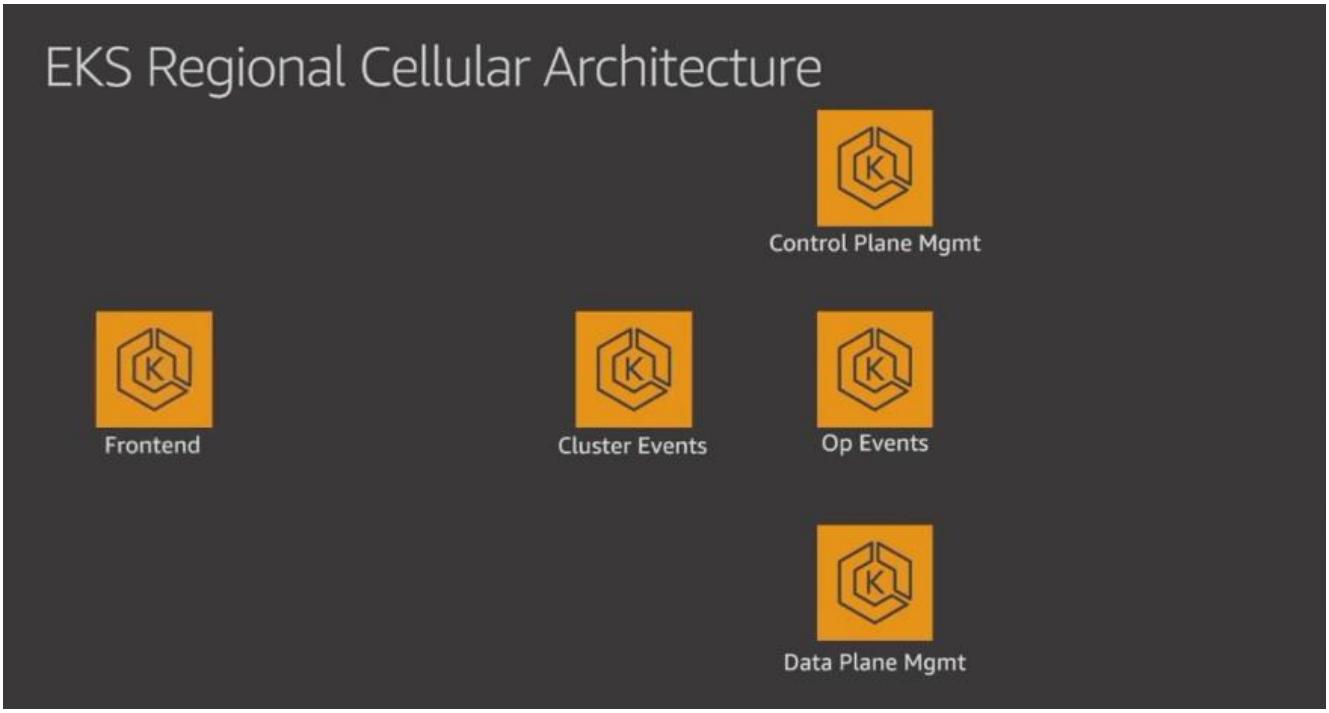
We added the event stream interface too. We are also depicting the microservice pattern used in the implementation of the EKS service for your K8s cluster and worker nodes

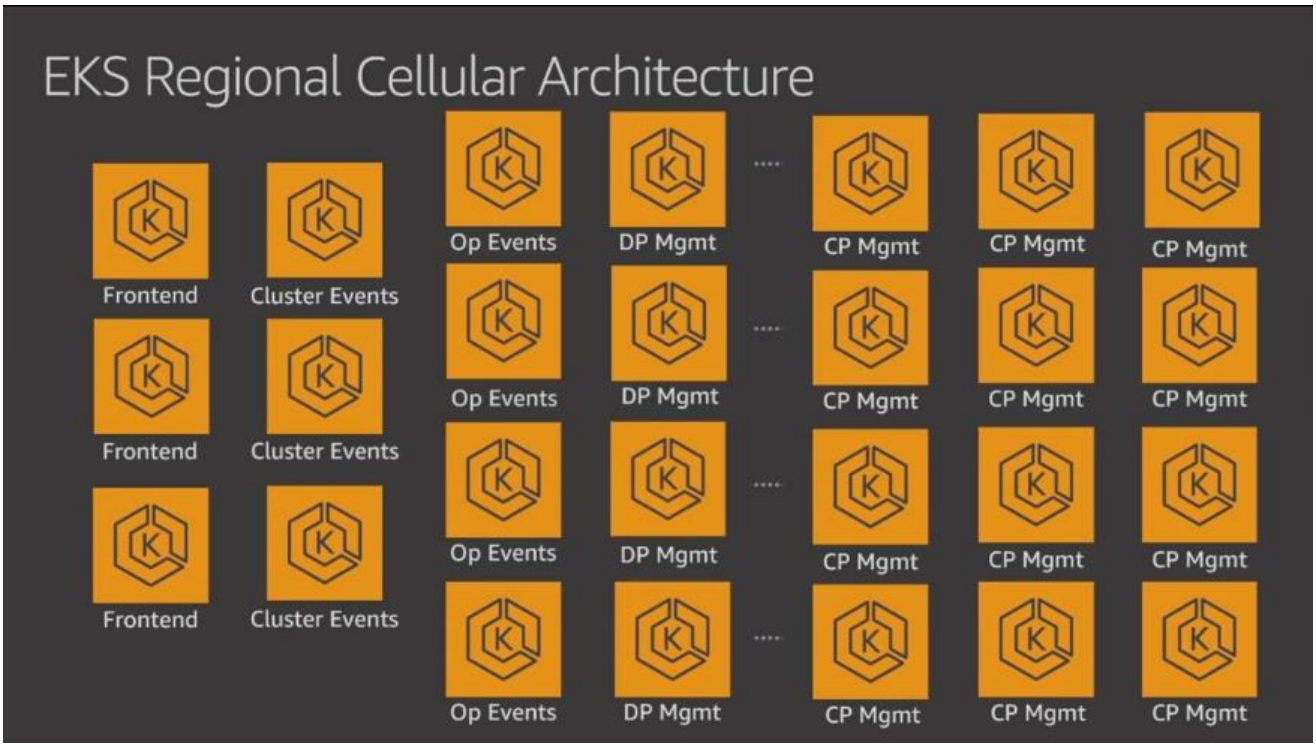


These are some of the 3 major components that manage state for your EKS cluster

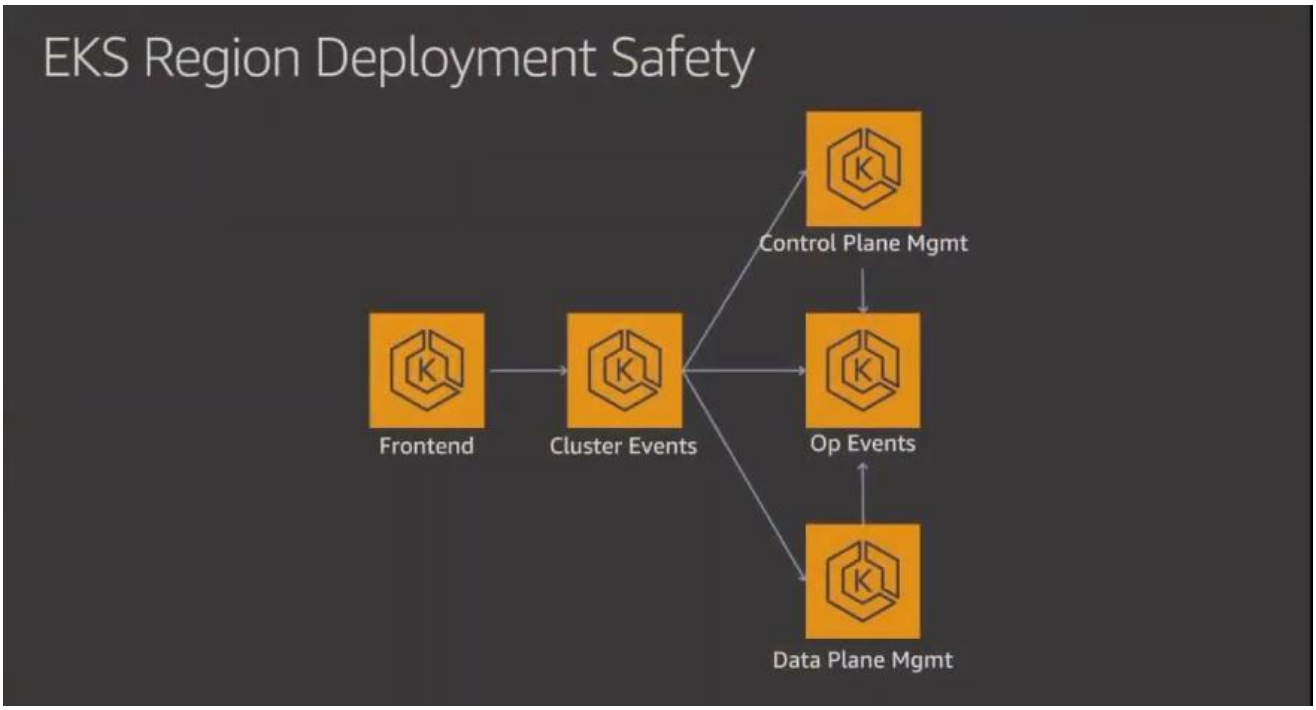


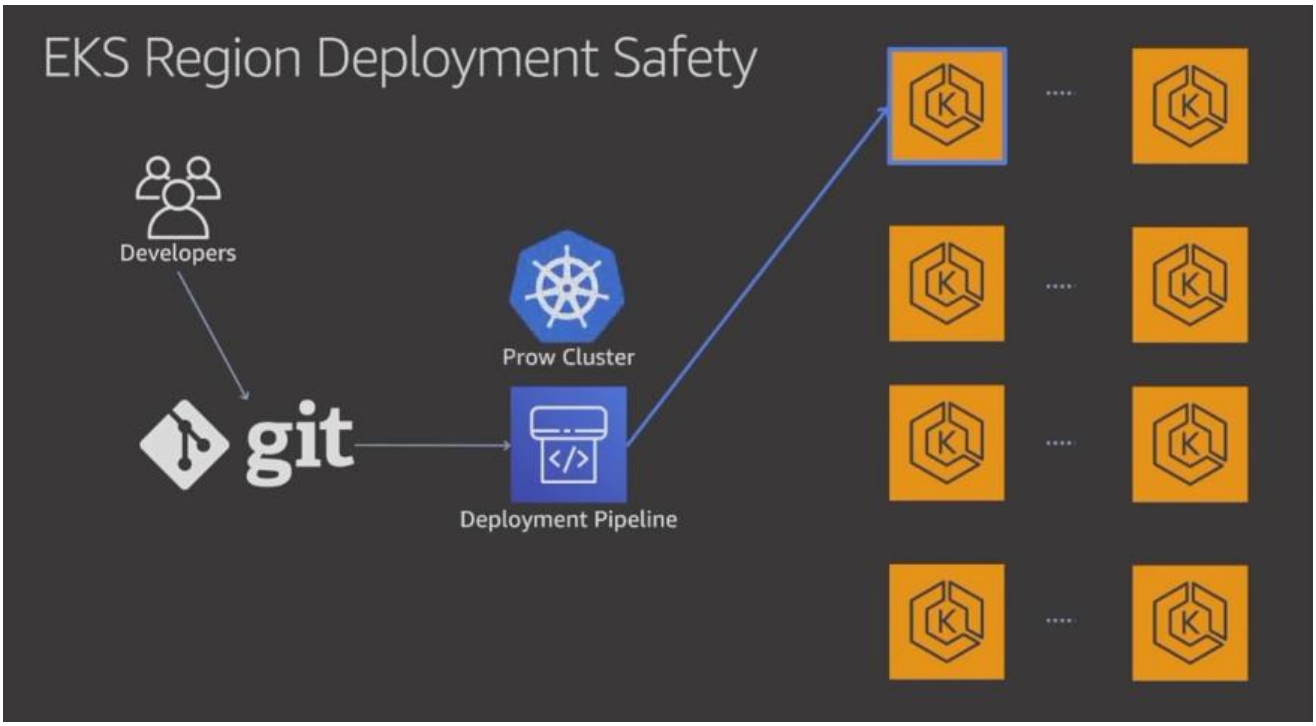
We use other AWS services within EKS. Note that EKS is based on Lambda and not on K8s.

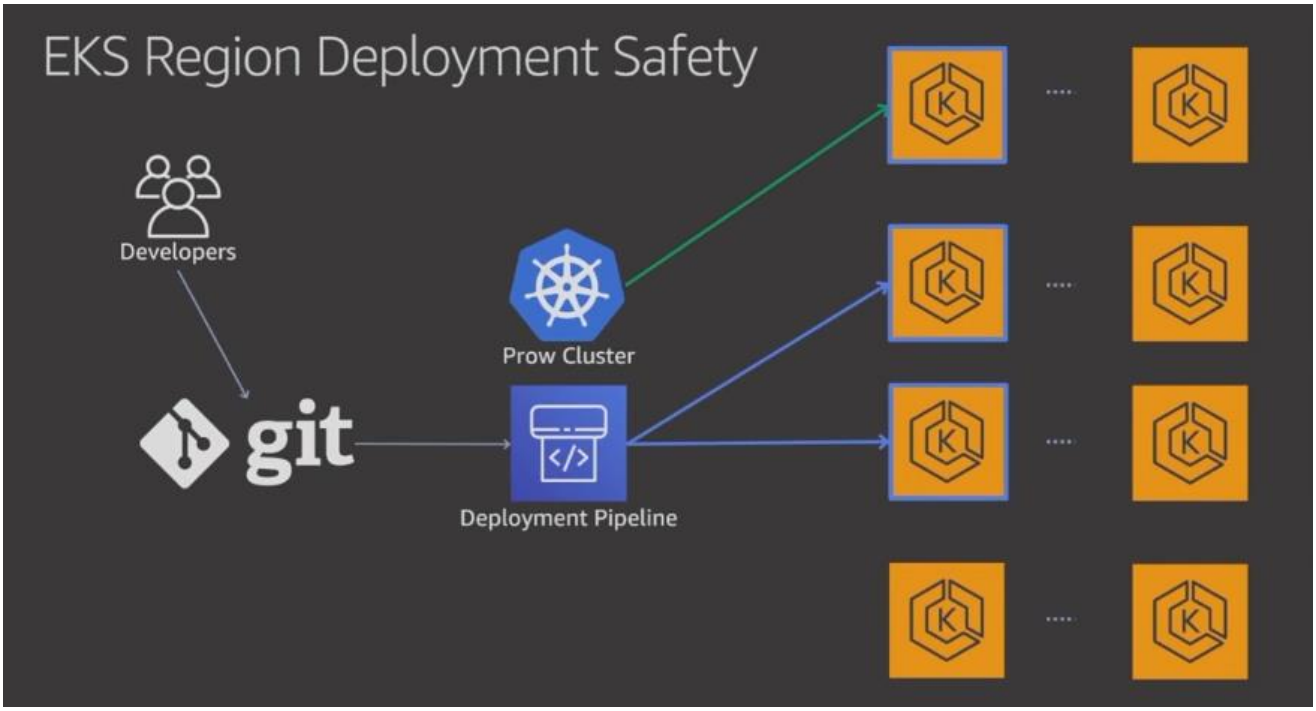


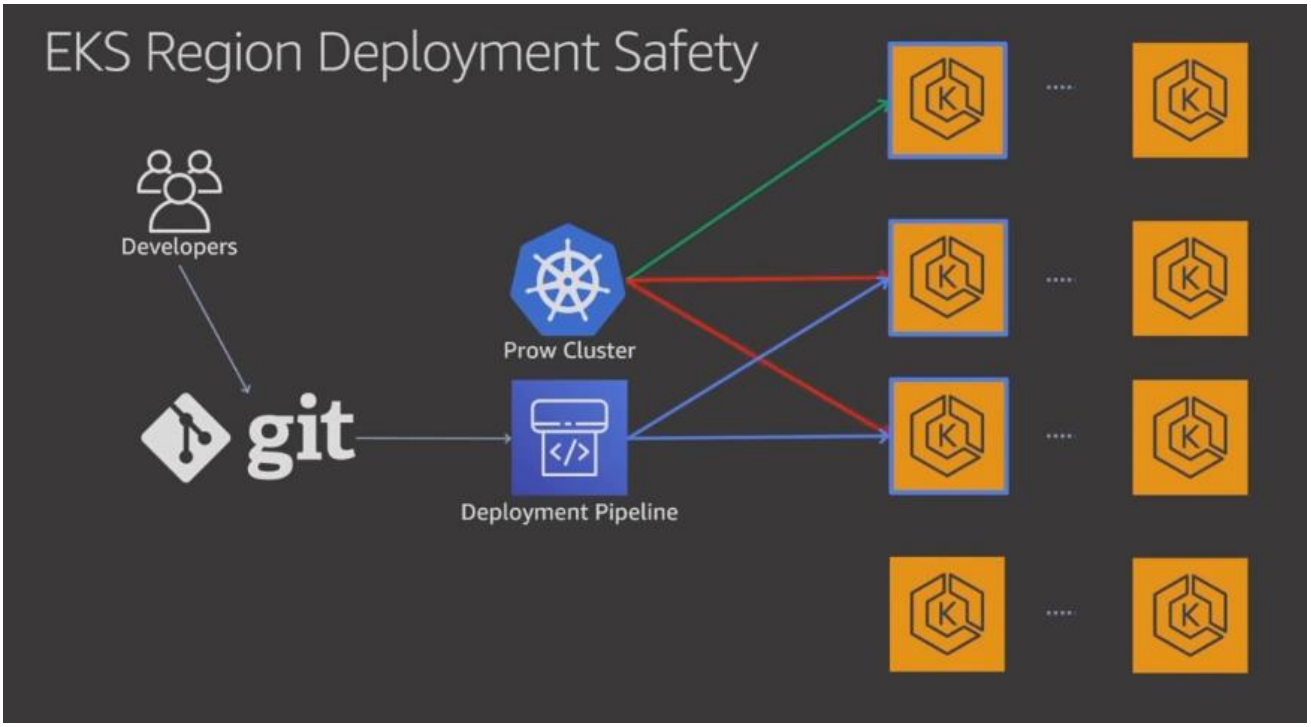


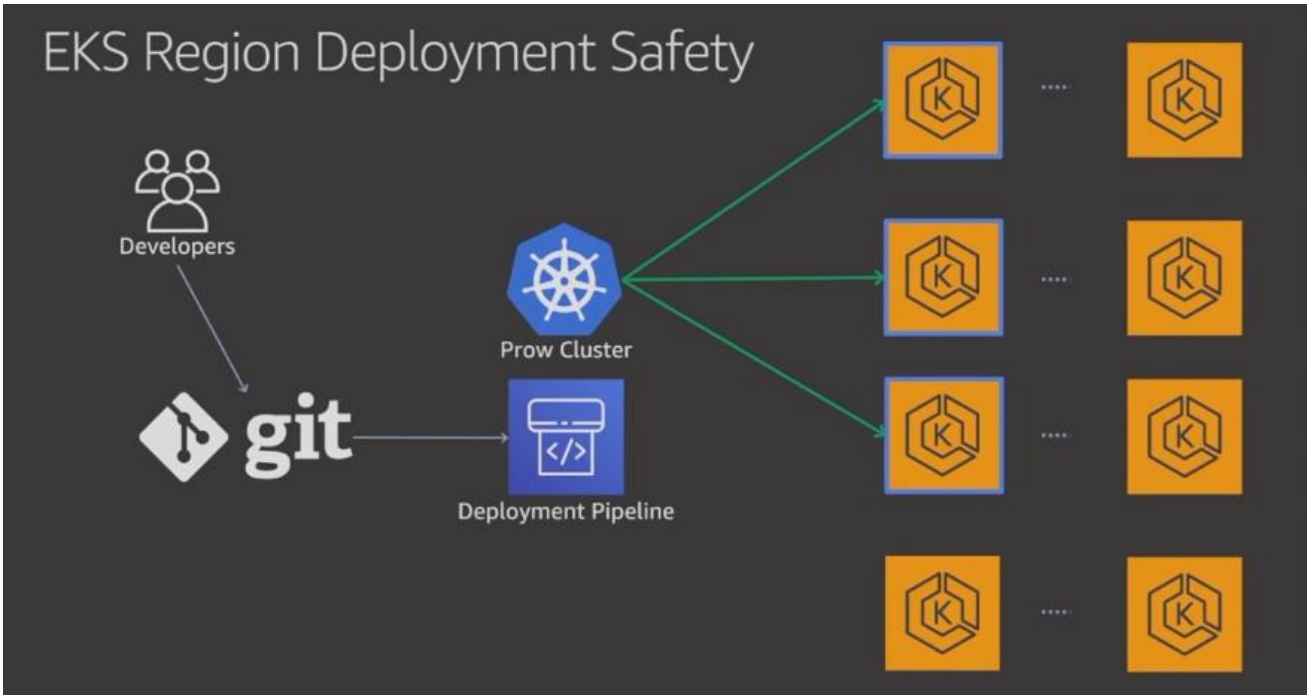
We then scale the components to meet the demand of the EKS global service in every region

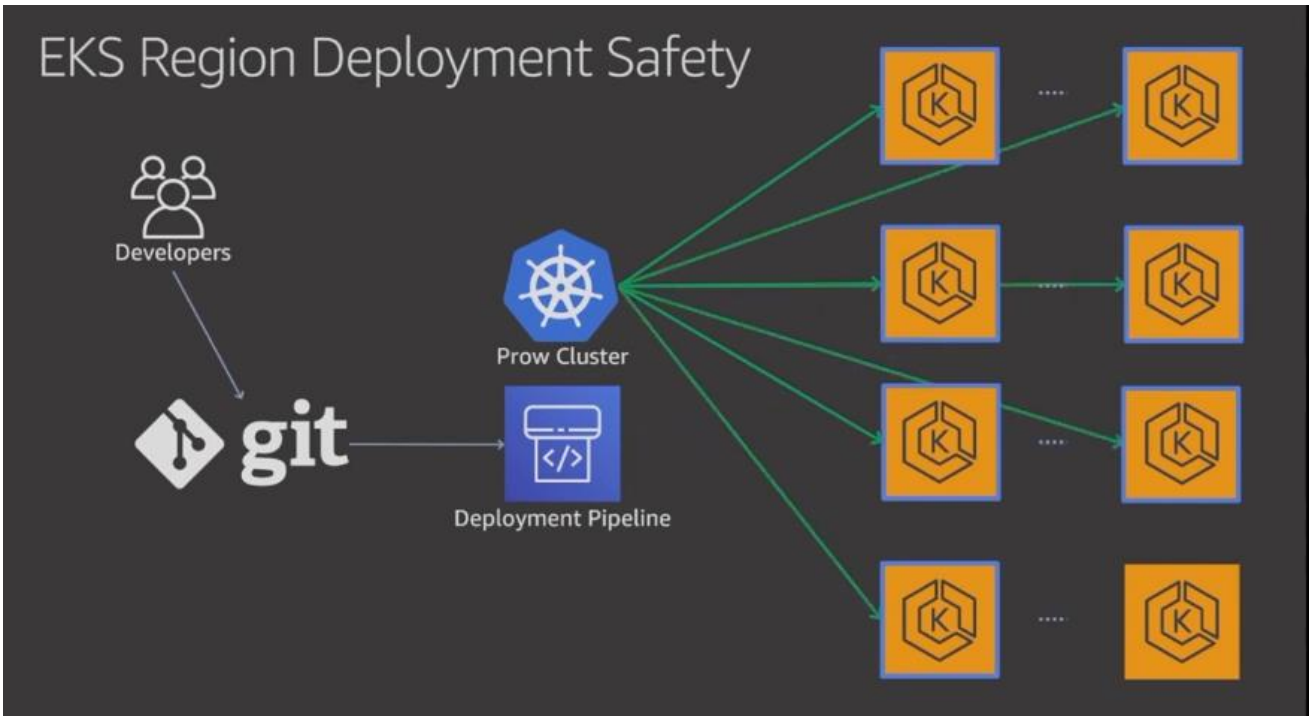


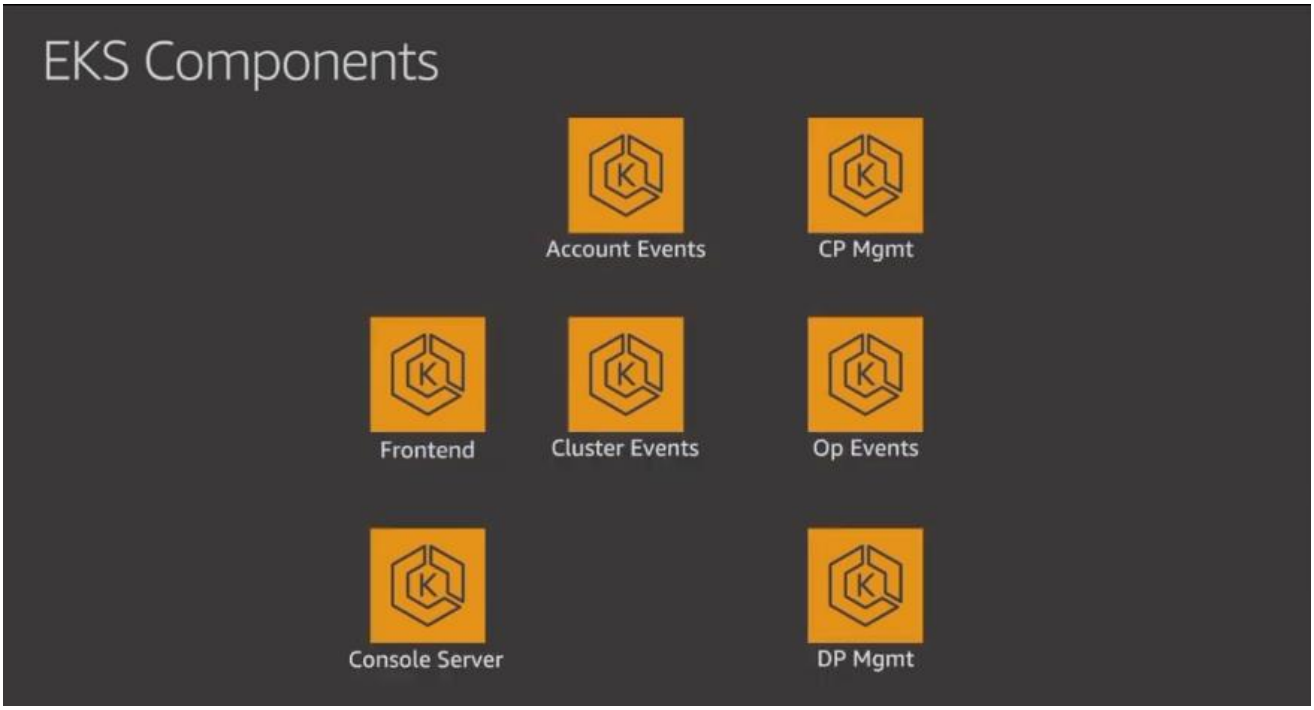

EKS Region Deployment Safety
Developers
git
Prow Cluster
Deployment Pipeline

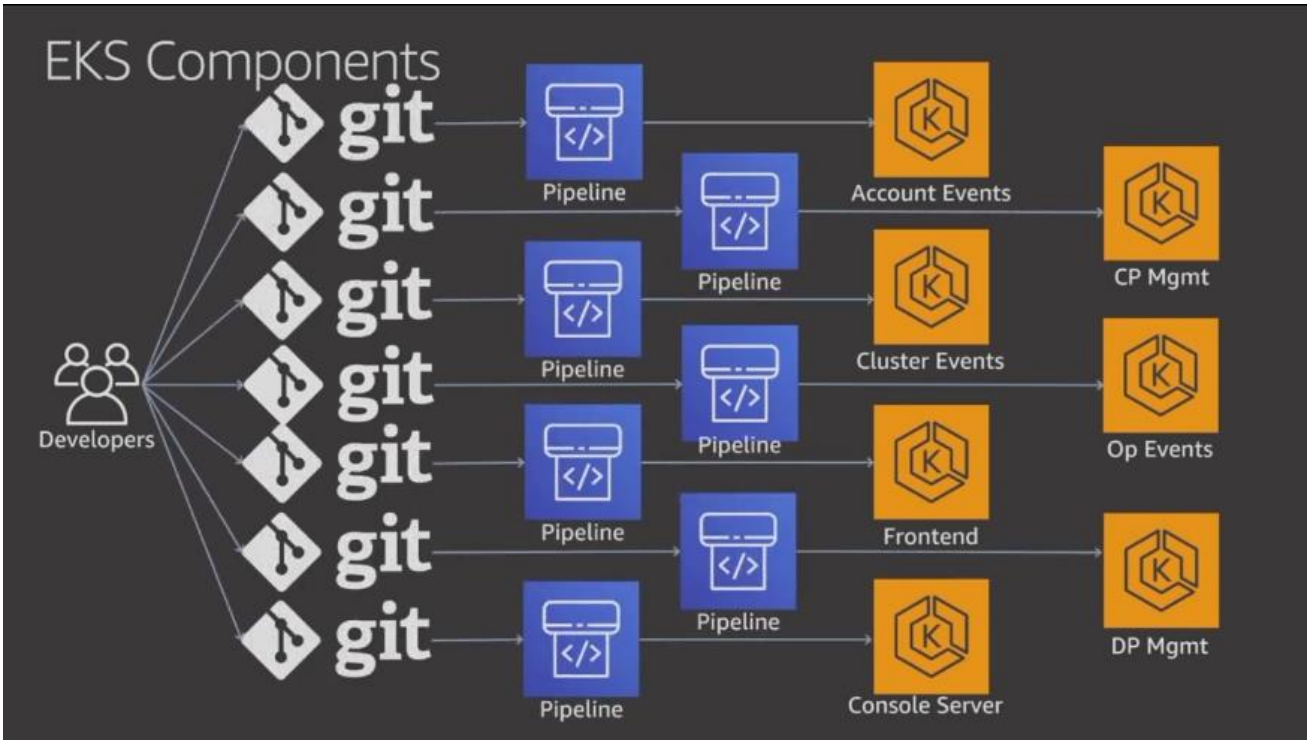

EKS Region Deployment Safety
Developers
git
Prow Cluster
Deployment Pipeline


EKS Region Deployment Safety
Developers
git
Prow Cluster
Deployment Pipeline


EKS Region Deployment Safety
Developers
git
Prow Cluster
Deployment Pipeline


EKS Region Deployment Safety
Developers
git
Prow Cluster
Deployment Pipeline


EKS Components
Account Events
CP Mgmt
Frontend
Cluster Events
Op Events
Console Server
DP Mgmt



EKS Enhancements: What we've been up to

The year in review

Security & Reliability
ISO, SOC 123, and PCI compliance
99.9% Service Level Agreement
Cluster creation limit raised to 50 per region
API Server Endpoint Access Control
Control Plane Logs in Amazon CloudWatch
AWS IAM authenticator integration
EKS v1.10 and 1.11 end of life
Amazon ECR PrivateLink support
Kubernetes pod security policies
AWS IAM for Service Accounts
Cluster tagging

Regions & Versions
Seoul, Mumbai, London, Paris, Ohio, Frankfurt, Singapore, Sydney, Tokyo, Hong Kong, São Paulo, Bahrain
Support for Kubernetes versions 1.11, 1.12, 1.13, and 1.14

Nodes
Windows Node Support (GA)
Managed Node Groups
A1 (ARM) instance support (preview)
EKS-Optimized AMI AWS Systems Manager parameters

Storage & Networking
Alpha CSI Driver for Amazon FSx for Lustre
Beta CSI Drivers for Amazon EBS and Amazon EFS
Support for Public IP Addresses Within Cluster VPCs
AWS ALB Ingress Controller
Amazon VPC CNI plugin v1.3, 1.4, 1.5

Tooling
AWS App Mesh controller
Managed Cluster Version Updates
CloudWatch Container Insights
eksctl as the official EKS CLI
AWS Node Termination Handler
Mixed instance policy support and GPU-provider for Cluster Autoscaler

Machine Learning
Deep Learning Benchmark Utility
AWS in official Kubeflow documentation
Support for P3dn and G4dn instances
Escalator autoscaler one-click capacity

All since re:Invent 2018

AWS IAM Roles for Service Accounts

Secure

- IAM policy restrictions can restrict roles to Service Accounts or Namespaces
- Enables isolated AWS permissions per Service Account
- Credentials are automatically rotated
- The cluster's signing key is automatically rotated

Easy Integration

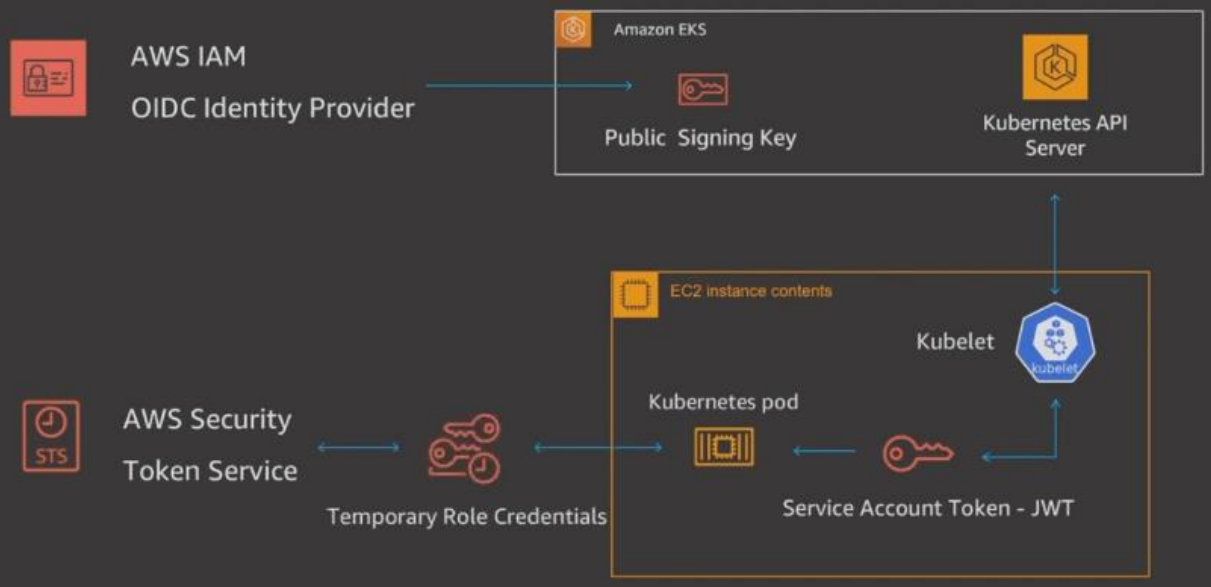
- Annotate the Service Account
- Built into the default credential chains in the AWS SDKs and CLI

Auditable

- Service Account names are logged in AWS CloudTrail

AWS IAM Roles for Service Accounts

AWS IAM Roles for Service Accounts



p0
Security

p1
Reliability

Investments in security and reliability

- Cellular Architecture
- Version qualification and release
- Security Patching
- Operations tooling

EKS Enhancements: Things you're gonna love

AWS Fargate for Amazon EKS

Fargate is a serverless compute platform for containers on AWS

The differences between using EKS and ECS with Fargate are driven by the orchestration system.

AWS Fargate for Amazon EKS

Bring existing pods

You don't need to change your existing pods.

Fargate works with existing workflows and services that run on Kubernetes.

Production Ready

Launch pods quickly. Easily run pods across multiple AZs for high availability.

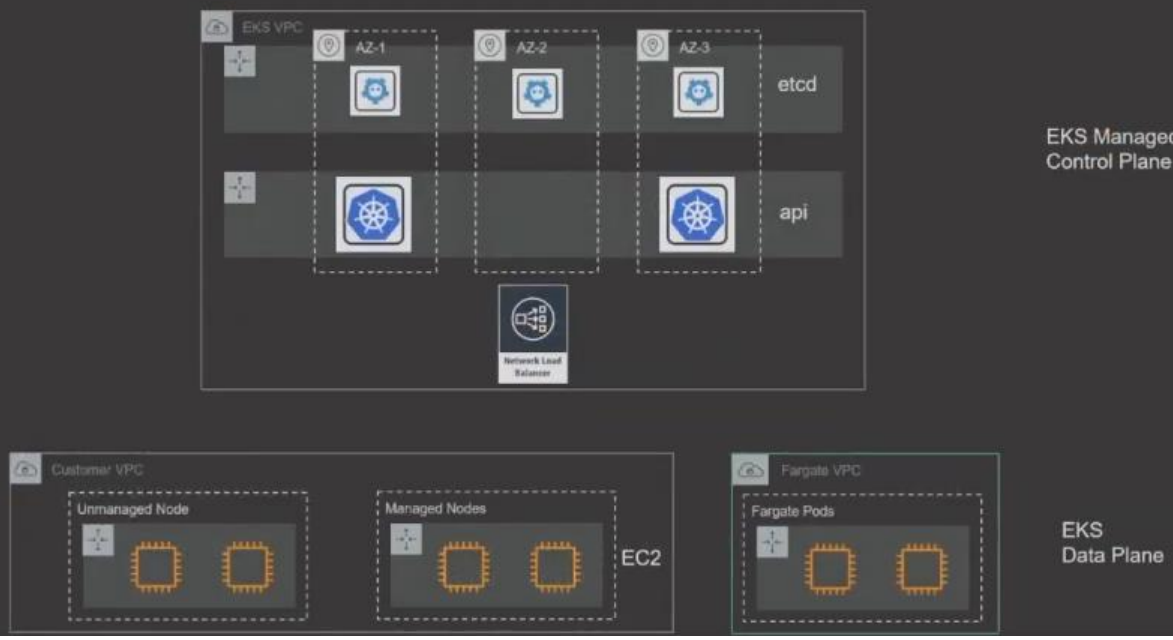
Each pod runs in an isolated VM compute environment.

Right-Sized and Integrated

Only pay for the resources you need to run your pods.

Includes native AWS integrations for networking and security.

EKS Cluster Architecture



EKS Fargate profile template


```
{
    "status": "ACTIVE",
    "subnets": [
        "subnet-0de8355bc4ds45af3",
        "subnet-0det555b36hdy67d3"
    ],
    "clusterName": "FargateCluster",
    "fargateProfileArn": "arn:aws:eks:us-west-2:123456789:fargateprofile/FargateCluster/FargateProfileCatchAll/4cg3303c-539e-a202-5b75-bb1dd3dd0590",
    "selectors": [
        {
            "namespace": "default"
        },
        {
            "namespace": "kube-system"
        },
        {
            "labels": {
                "foo": "bar"
            },
            "namespace": "mynamespace"
        }
    ],
    "fargateProfileName": "FargateProfileCatchAll",
    "podExecutionRole": "arn:aws:iam::123456789:role/FargateCluster-SERVICE-ROLE-AWSServiceRoleFargateCluster-1PLJY3220ID6I",
    "createdAt": 1573039680.227
} I
```

EKS Fargate profile template

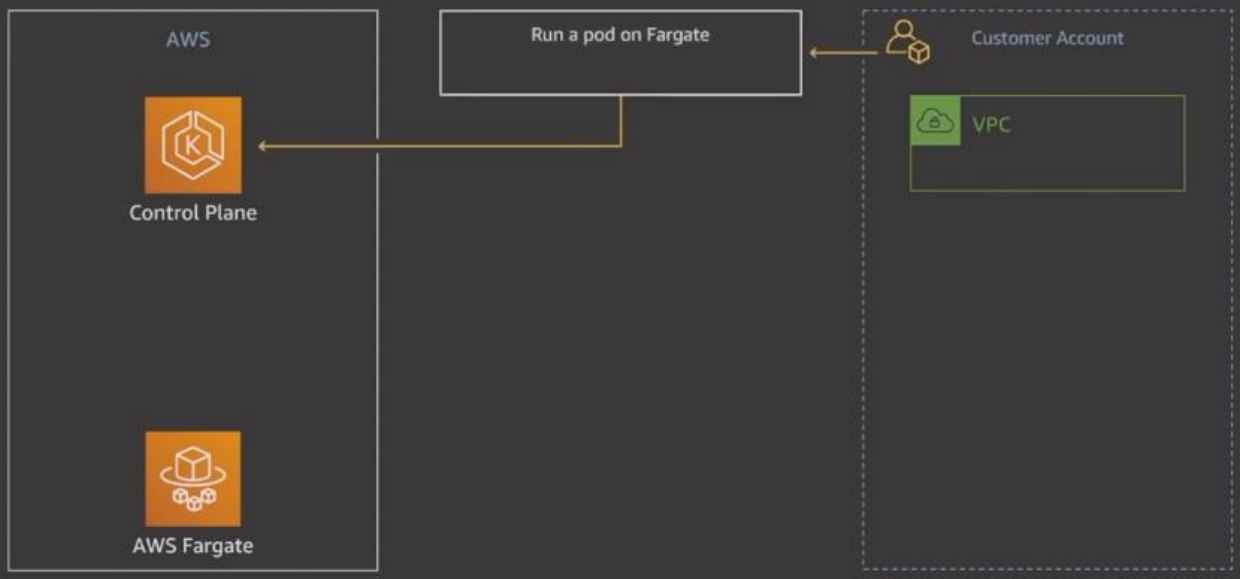

```
{
    "status": "ACTIVE",
    "subnets": [
        "subnet-0de8355bc4ds45af3",
        "subnet-0det555b36hdy67d3"
    ],
    "clusterName": "FargateCluster",
    "fargateProfileArn": "arn:aws:eks:us-west-2:123456789:fargateprofile/FargateCluster/FargateProfileCatchAll/4cg3303c-539e-a202-5b75-bb1dd3dd0590",
    "selectors": [
        {
            "namespace": "default"
        },
        {
            "namespace": "kube-system"
        },
        {
            "labels": {
                "foo": "bar"
            },
            "namespace": "mynamespace"
        }
    ],
    "fargateProfileName": "FargateProfileCatchAll",
    "podExecutionRole": "arn:aws:iam::123456789:role/FargateCluster-SERVICE-ROLE-AWSServiceRoleFargateCluster-1PLJY3220ID6I",
    "createdAt": 1573039680.227
}
```

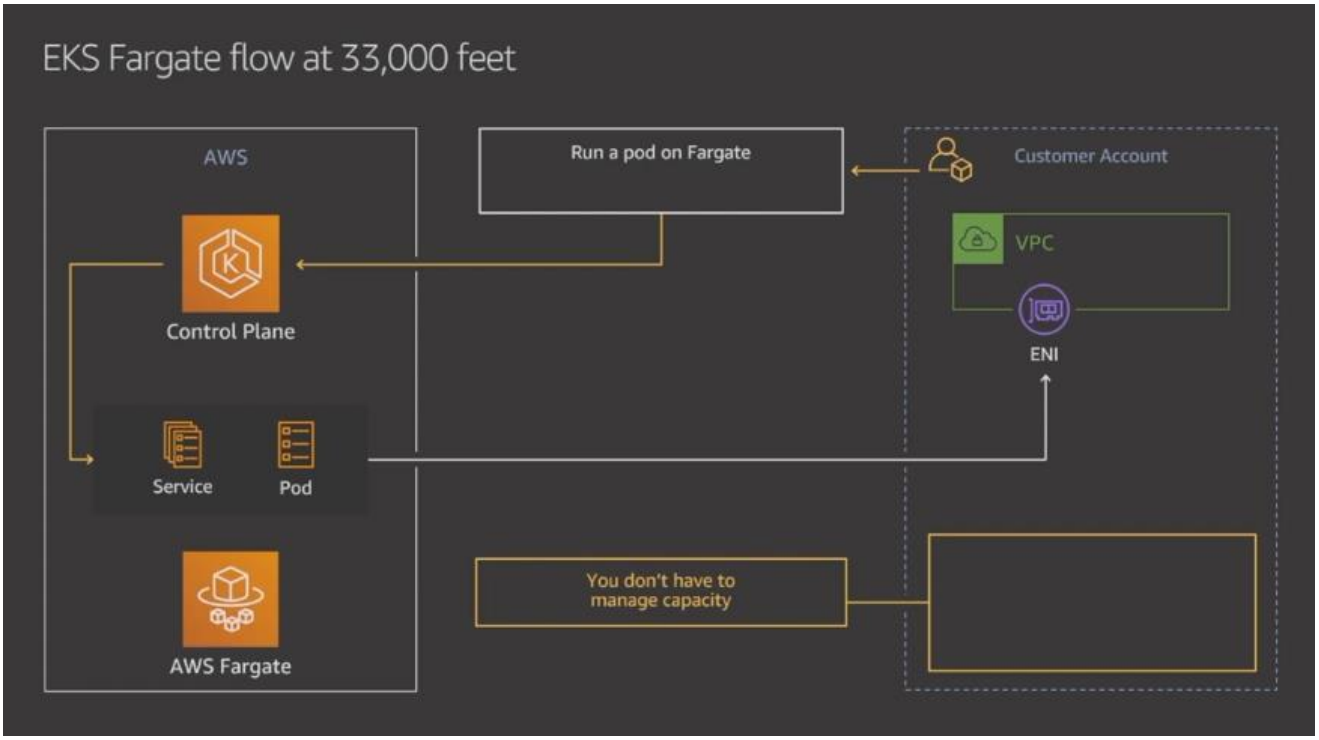

Subnets to launch the pods in

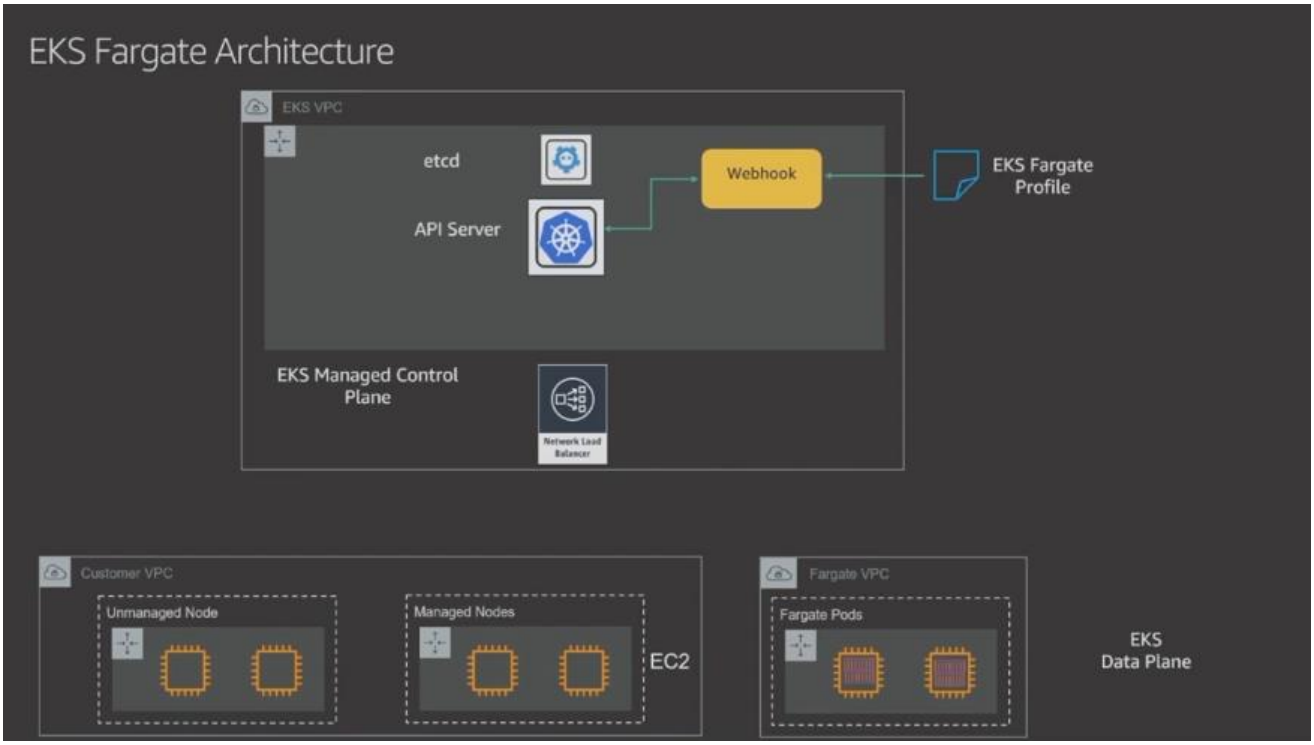
Selection criteria into Fargate

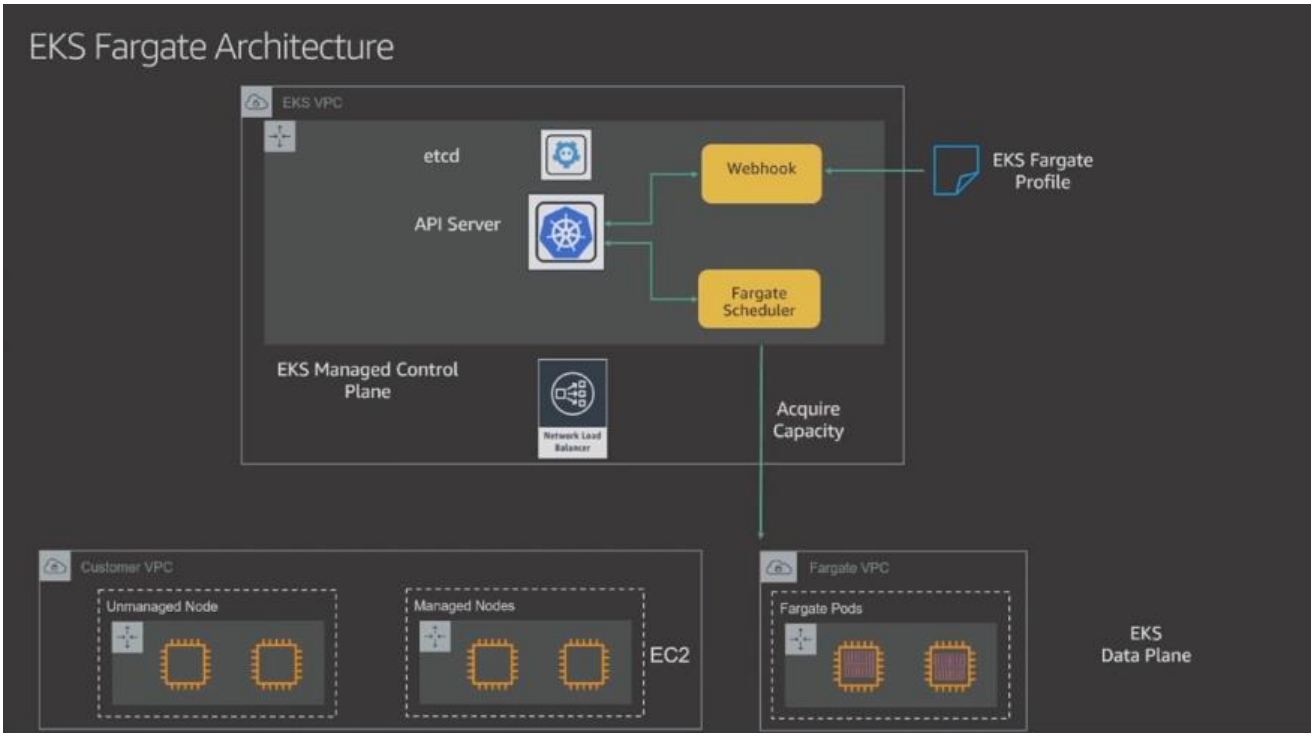
IAM Role to be associated to the kubelet

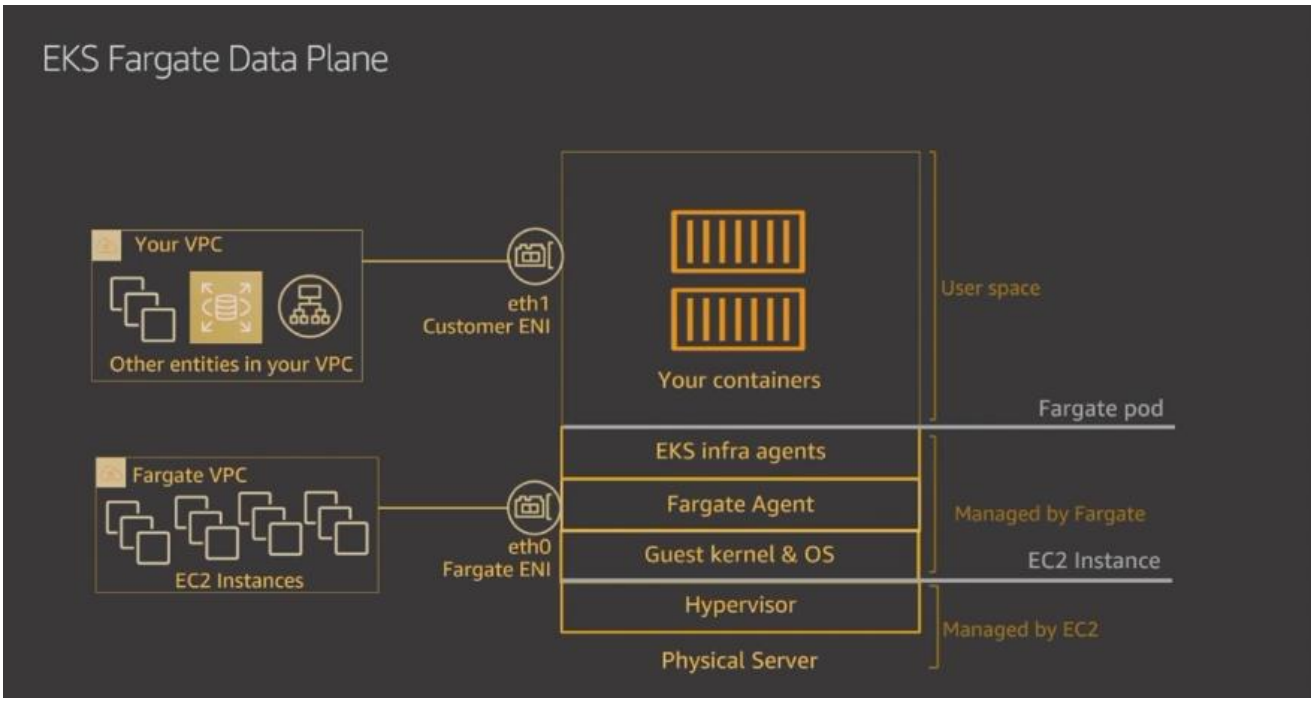
EKS Fargate flow at 33,000 feet




EKS Fargate flow at 33,000 feet
AWS
Control Plane
Service
Pod
AWS Fargate
Run a pod on Fargate
Customer Account
VPC
ENI
You don't have to manage capacity


EKS Fargate Architecture
EKS VPC
etcd
API Server
Webhook
EKS Fargate Profile
EKS Managed Control Plane
Customer VPC
Unmanaged Node
Managed Nodes
EC2
Fargate VPC
Fargate Pods
EKS Data Plane


EKS Fargate Architecture
EKS VPC
etcd
API Server
Webhook
Fargate Scheduler
EKS Fargate Profile
EKS Managed Control Plane
Network Load Balancer
Acquire Capacity
Customer VPC
Unmanaged Node
Managed Nodes
EC2
Fargate VPC
Fargate Pods
EKS Data Plane


EKS Fargate Data Plane
Your VPC
Other entities in your VPC
eth1
Customer ENI
Your containers
User space
Fargate pod
Fargate VPC
EC2 Instances
eth0
Fargate ENI
EKS infra agents
Fargate Agent
Guest kernel & OS
Managed by Fargate
EC2 Instance
Hypervisor
Managed by EC2
Physical Server

Recap: EKS Fargate UX changes

Things you no longer need to do

- Manage Kubernetes worker nodes
- Pay for unused capacity
- Use K8s Cluster Autoscaler (CA)

Things you get out of the box

- VM isolation at pod level
- Pod level billing
- Easy chargeback in multi tenant scenarios

Things you can't do
(for now)

- Deploy Daemonsets
- Use service type LoadBalancer (CLB/NLB)
- Running privileged containers
- Run stateful workloads

EKS Fargate Availability

Available today for all new 1.14 clusters

- Create a new cluster
- Update a 1.13 cluster to 1.14

Use EKS with Fargate in

- Virginia (us-east-1)
- Ohio (us-east-2)
- Dublin (eu-west-1)
- Tokyo (ap-northeast-1)

EKS Enhancements: What's Next?

Our vision for EKS

Globally available | Easy to use | Production ready | Cost-effective | High-performance

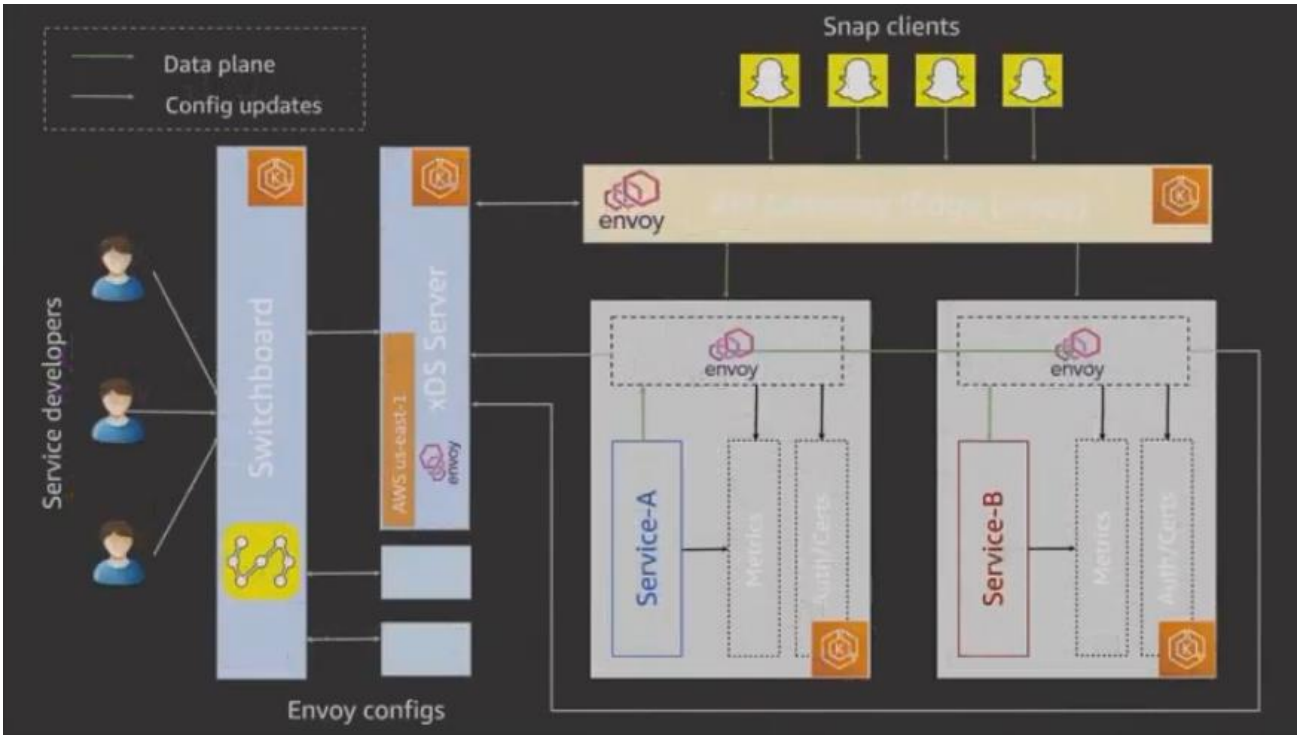
All the building blocks of Kubernetes in one place

Snap Service Mesh on EKS

Snap service mesh …

Infrastructure layer providing foundation for SOA enables core capabilities by default at the platform level

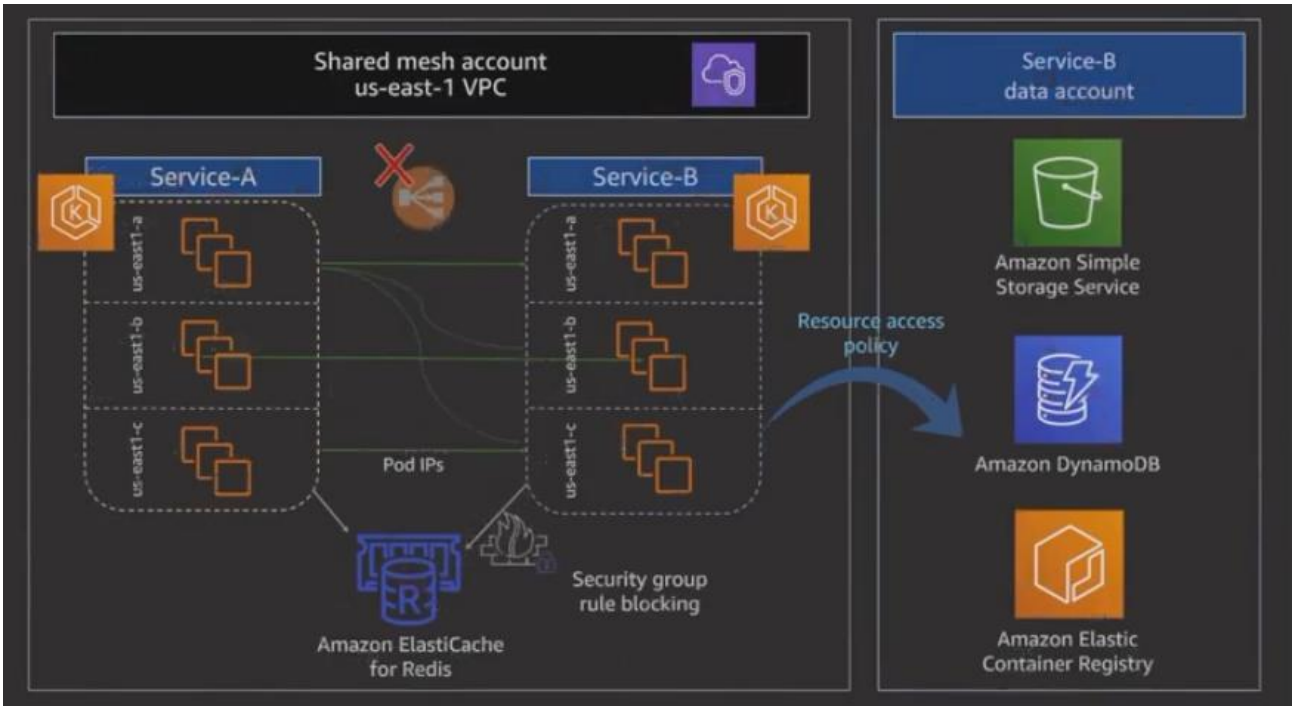
- Security by default
- Standardized traffic management and routing policies:
  - Service discovery—Just call `<service>.snap`
  - Zonal affinity and regional proximity to favor closest endpoints
  - Traffic splitting, mirroring, and failover
  - Automatic resilience and circuit breakers
- Observability by default



This is a high-level design of our service mesh architecture

Architectural design choices for AWS

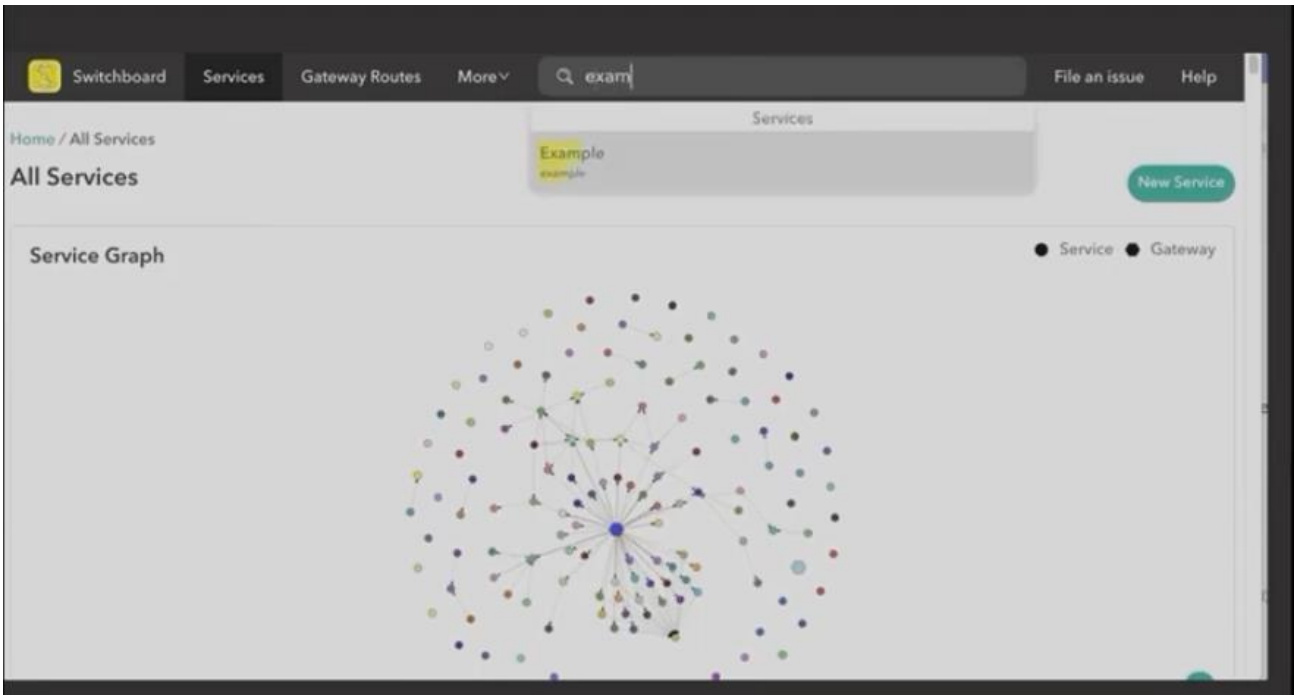
- Accounts
  - One shared account for compute and network
  - Service data is isolated into separate accounts
- Compute: Amazon EKS
  - One EKS cluster per group of correlated services
  - ~300 EKS clusters in 4 mesh regions (as large as ~3K nodes)
- Network: > 4M QPS in AWS Regions
  - One VPC/Region, with subnets in 3 AZs
  - Security perimeter at the edge
  - Network-level protection: Security groups, network ACLs, resource access policy

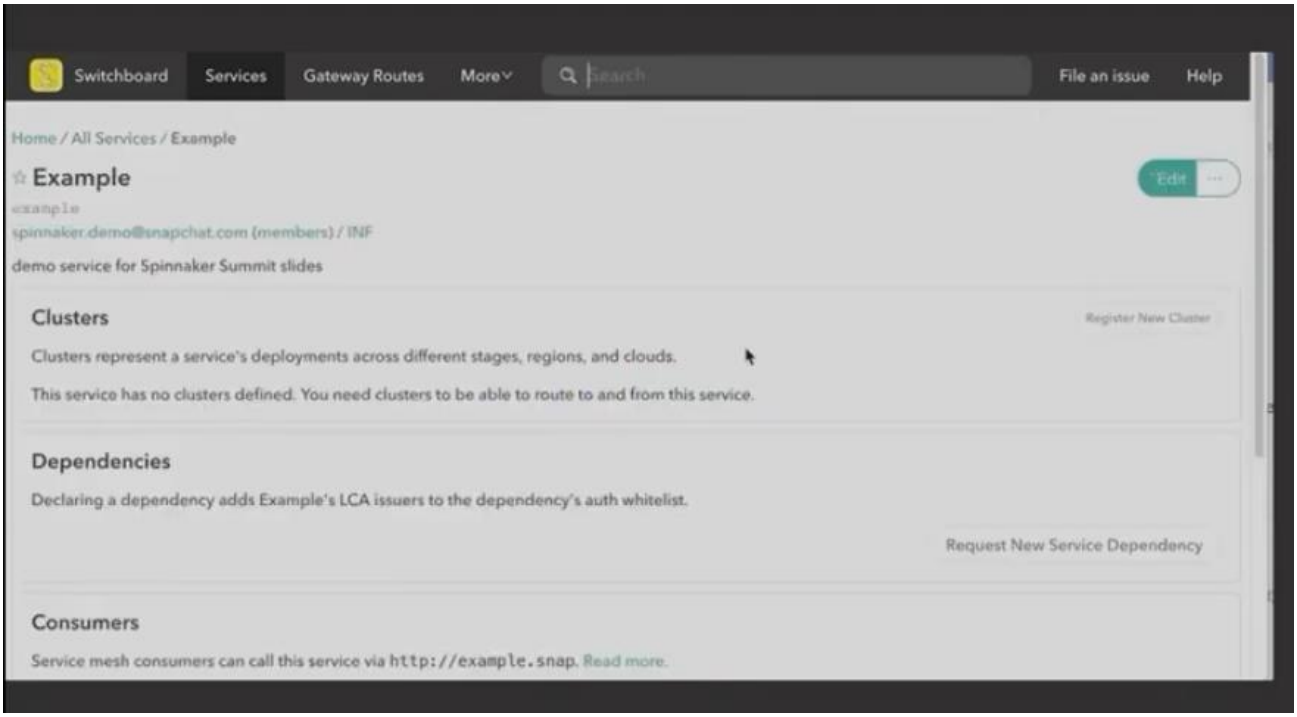


This illustrates our AWS Service Mesh architecture.

Tooling for common service requirements

- Resource management:
  - Automate Amazon EKS cluster provisioning, and version upgrades
  - Standardize cluster add-ons: Cluster Auto Scaler, CoreDNS, and CNI
  - Per-service AWS Identity and Access Management (IAM) roles and granular access controls


Switchboard
Services
Gateway Routes
More
exam
File an issue
Help
Services
Example
example
Home / All Services
All Services
New Service
Service Graph
Service
Gateway


Switchboard
Services
Gateway Routes
More
Search
File an issue
Help
Home / All Services / Example
Example
Edit
example
spinnaker.demo@snapchat.com (members) / INF
demo service for Spinnaker Summit slides
Clusters
Register New Cluster
Clusters represent a service's deployments across different stages, regions, and clouds.
This service has no clusters defined. You need clusters to be able to route to and from this service.
Dependencies
Declaring a dependency adds Example's LCA issuers to the dependency's auth whitelist.
Request New Service Dependency
Consumers
Service mesh consumers can call this service via http://example.snap. Read more.

Switchboard | Services | Gateway Routes | More | Search | File an issue | Help

Dependencies

Declaring a dependency adds Example's LCA issuers to the dependency's auth whitelist.

Request New Service Dependency

Consumers

Service mesh consumers can call this service via `http://example.snap`. Read more.

This service has no consumers yet.

Cloud Resources

New Kubernetes Cluster

Kubernetes clusters associated with this service.

example_ex2_us-west-2_staging AWS us-west-2

example_testing_us-west-2_staging AWS us-west-2

Switchboard | Services | Gateway Routes | More | Search | File an issue | Help

Home / All Services / Example / Create New Cloud Resource

Resource Type

Kubernetes Cluster

Cloud Provider

AWS

Stage

An EKS cluster's Stage is associated with a set of IAM roles, which can be granted permission to assume data account roles. Only **Production** clusters can assume roles under PROD data accounts.

- Production
- Staging

Region

The region to create your kubernetes cluster in

Select region

Minimum Node Pool Size

Switchboard Services Gateway Routes More Search File an issue Help

under PROD data accounts.

Production

Staging

Region

The region to create your kubernetes cluster in

Select region

us-east-1

us-west-2

Maximum Node Pool Size

The maximum number of nodes you want your node pool to be scaled up to

8

Node Instance Type

The EC2 instance type you want to use

Switchboard Services Gateway Routes More Search File an issue Help

under PROD data accounts.

Production

Staging

Region

The region to create your kubernetes cluster in

us-west-2

Minimum Node Pool Size

The minimum number of nodes you want running your pods

1

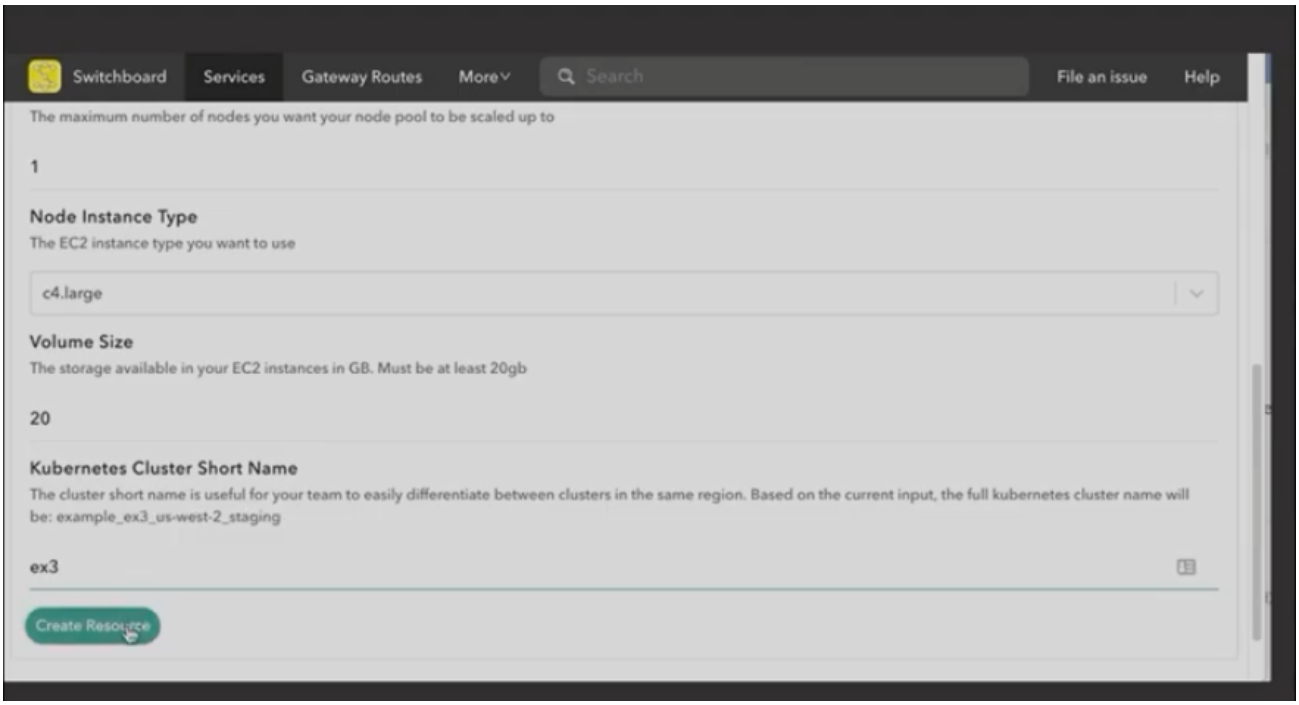
Maximum Node Pool Size

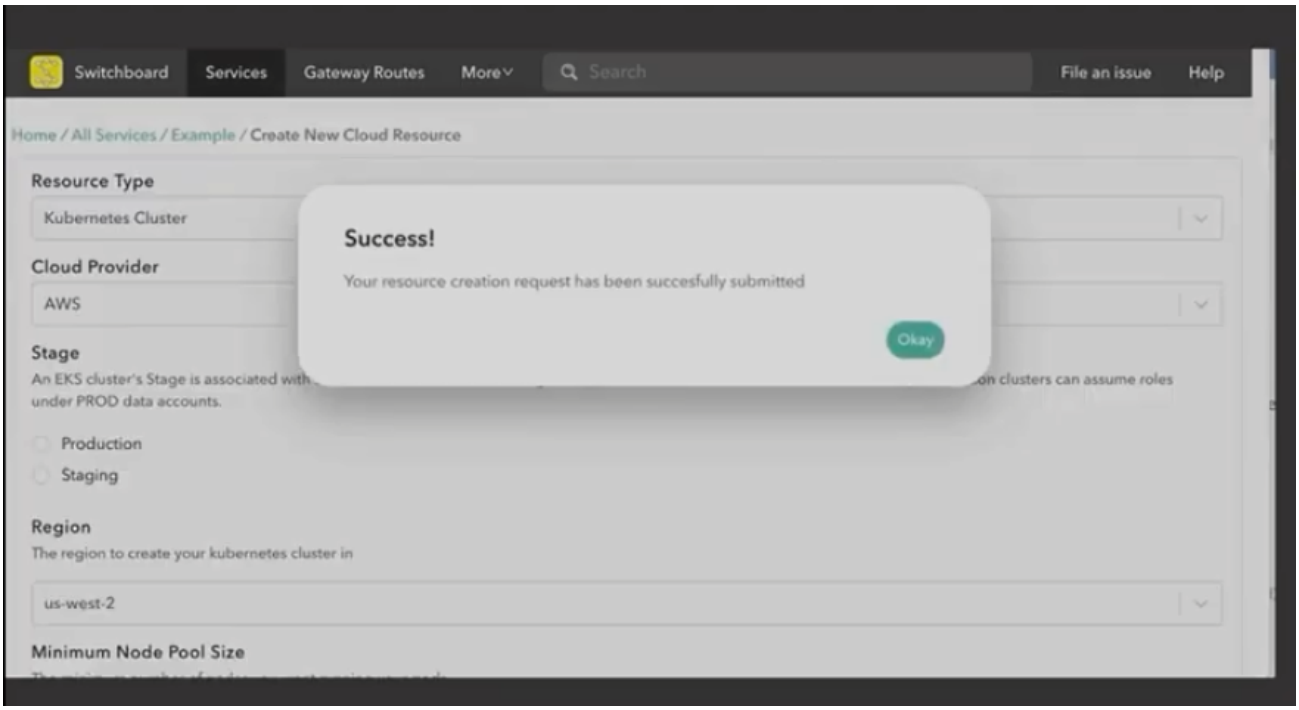
The maximum number of nodes you want your node pool to be scaled up to

1

Node Instance Type

The EC2 instance type you want to use





Tooling for common service requirements

- Standardize service deployments
  - Injection and upgrades of common sidecar containers
  - Default best practices through Spinnaker pipelines:
    - Uniform pod distribution per zone
    - Safe rollouts with integrated health checks

Switchboard | Services | Gateway Routes | More | Search | File an issue | Help

Home / All Services / Example / Cluster: default.aws-us-west-2.STAGING

default.aws-us-west-2.STAGING

in service Example

spinnaker.dev@snapchat.com (members) / INF

Staging | AWS | us-west-2 | Security Updates Needed

Edit

Delete

Security Warnings

Security features may need to be released in an opt-in way to ensure changes don't break running services. Each security warning can be fixed by opting into these features via options available in the edit cluster form.

Missing TLS Verification - Mesh CA

This cluster's consumers will not validate TLS certificates for this mesh cluster. Ensure the auth-sidecar is configured correctly and enable the option to load the mesh CA root certificates.

Missing Header Scrubbing

This cluster is vulnerable to spoofed headers from other services in the mesh. Ensure the envoy sidecar is up to date and enable the pre/post auth header scrubbing filter.

Missing SDS

This cluster is not configured to use the Secret Discovery Service (SDS). Without SDS enabled, the TLS certificates for this cluster will not rotate properly and calling services cannot fully validate certificates of this cluster.

Allowed Duplicate LCA Audiences

This cluster is configured to allow other clusters to use the same LCA audience. When other services use the same LCA audience, they could potentially obtain LCA tokens that are valid to authenticate to this service. Please enable the duplicate audiences check for this cluster.

Incoming Calls

Other services that call `http://example.snap` in aws-us-west-2 will be directed to `https://<HOST-IP>:8080`. Read more about how to call this service from different clients.

Switchboard | Services | Gateway Routes | More | Search | File an issue | Help

Home / All Services / Example / Cluster: default.aws-us-west-2.STAGING / SpinnakerDeployment

Pipeline | Create | Trigger

Pipeline Name

example-default.aws-us-west-2.STAGING

Spinnaker Application

If you don't have a Spinnaker application yet please head over to Spinnaker portal and follow instructions provided here.

Select...

Spinnaker Pipeline

The spinnaker pipeline associated with the switchboard cluster.

Switchboard | Services | Gateway Routes | More | Search | File an issue | Help

Home / All Services / Example / Cluster: default.aws-us-west-2.STAGING / SpinnakerDeployment

Pipeline | Create | Trigger

Pipeline Name

example-default.aws-us-west-2.STAGING

Spinnaker Application

If you don't have a Spinnaker application yet please head over to Spinnaker portal and follow instructions provided here.

spinnakersummit

Kubernetes Cluster

The name of the Kubernetes cluster in AWS (EKS) where the service is deployed.

example_ex3_us-west-2_staging

Advance Options

- [ ] Canary Analysis
  Add support to run canary analysis using Envoy metrics before the rollpout.
- [ ] Zonal Rollout
  Add support for zonal affinity and replicate service in all available zones in the region.
- [ ] Health Checks
  Add support to run health checks using Envoy metrics after each deployment step.

Spinnaker Pipeline

The spinnaker pipeline associated with the switchboard cluster.

Switchboard | Services | Gateway Routes | More | Search | File an issue | Help

Spinnaker Application

If you don't have a Spinnaker application yet please head over to Spinnaker portal and follow instructions provided here.

spinnakersummit

Kubernetes Cluster

The name of the Kubernetes cluster in AWS (EKS) where the service is deployed.

example_ex3_us-west-2_staging

Advance Options

- [x] Canary Analysis
  Add support to run canary analysis using Envoy metrics before the rollpout.
- [x] Zonal Rollout
  Add support for zonal affinity and replicate service in all available zones in the region.
- [x] Health Checks
  Add support to run health checks using Envoy metrics after each deployment step.

Spinnaker Pipeline

The spinnaker pipeline associated with the switchboard cluster.

Create Pipeline

Looking ahead—Amazon EKS features to consume

- IAM roles for service accounts:
  - Least privilege: Scope permissions at the pod level instead of worker nodes
  - Access isolation between pods
- Managed worker node groups:
  - Node draining and graceful node shutdown
  - Integrated cluster Auto-Scaling (with multi-AZ node group)
  - Simplified cluster upgrade experience
- Managed cluster add-ons
  - Metrics server
  - CoreDNS auto-scaling

More Information at re:Invent

Related breakouts

CON203 - Getting started with Kubernetes on AWS
CON205 - Deploying applications using Amazon EKS
CON206 - Management and operations for Amazon EKS
CON212 - Running Kubernetes at Amazon scale using Amazon EKS
CON306 - Building ML infrastructure on Amazon EKS with Kubeflow
CON310 - Achieving zero-downtime deployments with Amazon EKS
CON316 - Adopting CSI for stateful workloads on Amazon EKS
CON317 - Securing your Amazon EKS cluster
CON327 - Oversubscription at scale: Running tons of containers with Kubernetes
CON334 - Running high-security workloads on Amazon EKS
CON411 - Advanced network resource management on Amazon EKS
CON413 - Move your machine learning workloads to Amazon EKS

Thank you!

Eswar Bala
Sr. Software Development Manager
Amazon Web Services
Twitter: @bala_eswar

Richard Sostheim
Principal Engineer
Amazon Web Services

Ahmed El Baz
Software Engineer
Snap Inc

re:Invent

aws