



Modern Applications at Scale with a Shared Services Platform

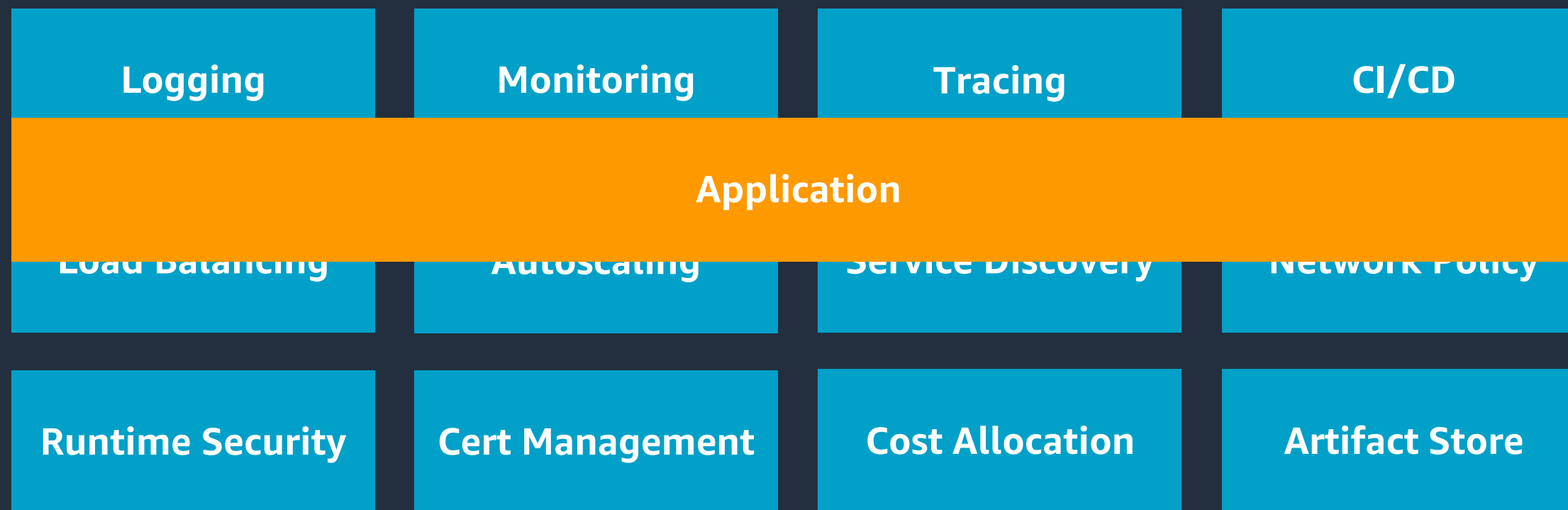
Kevin Coleman
Principal Container Specialist

Agenda

- Problem
- Shared Services Platform?
- Customer Benefits
- How AWS can help
- Next Steps

Components of a modern application

Day two operational considerations



Modern application services at AWS

200+ AWS services and counting

All services

Compute

EC2
Lightsail [↗](#)
Lambda
Batch
Elastic Beanstalk
Serverless Application Rep...
AWS Outposts
EC2 Image Builder
AWS App Runner

Containers

Elastic Container Registry
Elastic Container Service
Elastic Kubernetes Service
Red Hat OpenShift Service ...

Storage

S3
EFS
FSx
S3 Glacier
Storage Gateway
AWS Backup

Database

RDS
DynamoDB
ElastiCache
Neptune
Amazon QLDB
Amazon DocumentDB
Amazon Keyspaces
Amazon Timestream

Migration & Transfer

AWS Migration Hub
AWS Application Migration ...

Customer Enablement

AWS IQ [↗](#)
Support
Managed Services
Activate for Startups

Robotics

AWS RoboMaker

Blockchain

Amazon Managed Blockchain

Satellite

Ground Station

Quantum Technologies

Amazon Braket

Management & Governance

AWS Organizations
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
AWS AppConfig
Trusted Advisor
Control Tower
AWS License Manager
AWS Well-Architected Tool
Personal Health Dashbo... [↗](#)
AWS Chatbot
Launch Wizard
AWS Compute Optimizer
Resource Groups & Tag Editor

Machine Learning

Amazon SageMaker
Amazon Augmented AI
Amazon CodeGuru
Amazon DevOps Guru
Amazon Comprehend
Amazon Forecast
Amazon Fraud Detector
Amazon Kendra
Amazon Lex
Amazon Personalize
Amazon Polly
Amazon Rekognition
Amazon Textract
Amazon Transcribe
Amazon Translate
AWS DeepComposer
AWS DeepLens
AWS DeepRacer
AWS Panorama
Amazon Monitron
Amazon HealthLake
Amazon Lookout for Vision
Amazon Lookout for Equip...
Amazon Lookout for Metrics

Analytics

Athena
Amazon Redshift
EMR
CloudSearch
Elasticsearch Service
Kinesis
QuickSight [↗](#)
Data Pipeline
AWS Data Exchange
AWS Glue
AWS Lake Formation

AWS Cost Management

AWS Cost Explorer
AWS Budgets
AWS Marketplace Subscript...
AWS Application Cost Profiler

Front-end Web & Mobile

AWS Amplify
Mobile Hub
AWS AppSync
Device Farm
Amazon Location Service

AR & VR

Amazon Sumerian

Application Integration

Step Functions
Amazon AppFlow
Amazon EventBridge
Amazon MQ
Simple Notification Service
Simple Queue Service
SWF
Managed Apache Airflow

Business Applications

Amazon Connect
Amazon Pinpoint
Amazon Honeycode
Amazon Chime [↗](#)
Amazon Simple Email Service
Amazon WorkDocs
Amazon WorkMail
Alexa for Business

End User Computing

WorkSpaces

AWS customers use open source tools

The image displays a comprehensive grid of open source tools used by AWS customers, organized into several functional categories:

- App Definition and Development:** Includes Database (e.g., KV, Vitess, Cockroach Labs), Streaming & Messaging (e.g., cloudevents, NATS), Application Definition & Image Build (e.g., HELM, Buildpacks.io), and Continuous Integration & Delivery (e.g., argo, Jenkins, CircleCI).
- Orchestration & Management:** Includes Scheduling & Orchestration (e.g., kubernetes), Coordination & Service Discovery (e.g., CoreDNS, etcd), Remote Procedure Call (e.g., gRPC), Service Proxy (e.g., envoy), API Gateway (e.g., Kong, Apigee), and Service Mesh (e.g., Istio).
- Runtime:** Includes Cloud Native Storage (e.g., Rook, MinIO), Container Runtime (e.g., cri-o, containerd), and Cloud Native Network (e.g., CNM, Cilium).
- Provisioning:** Includes Automation & Configuration (e.g., KubeEdge, Ansible), Container Registry (e.g., Harbor, Dragonfly), Security & Compliance (e.g., Falco, Clair), and Key Management (e.g., Spiffe, SPIRE).
- Platform:** Includes Certified Kubernetes - Distribution (e.g., AWS, Alibaba Cloud), Certified Kubernetes - Hosted (e.g., AWS EKS, Azure AKS), Certified Kubernetes - Installer (e.g., Kubeadm), and PaaS/Container Service (e.g., Flynn, Heroku).
- Observability and Analysis:** Includes Monitoring (e.g., Prometheus, Thanos), Logging (e.g., Fluentd, ELK), Tracing (e.g., Jaeger, OpenTracing), Chaos Engineering (e.g., Gremlin), and Serverless (e.g., AWS Lambda).

Components of modern applications...at scale

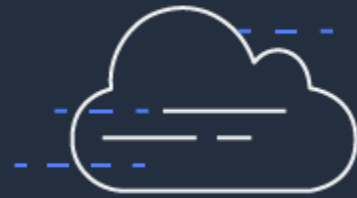
Day two operational considerations



What we hearing from customers



We want developers to focus on applications not infrastructure



We want to standardize on modern application tools and best practices



We need to implement guardrails and establish controls



We want to optimize our cloud resources and spend

Shared Services Platform



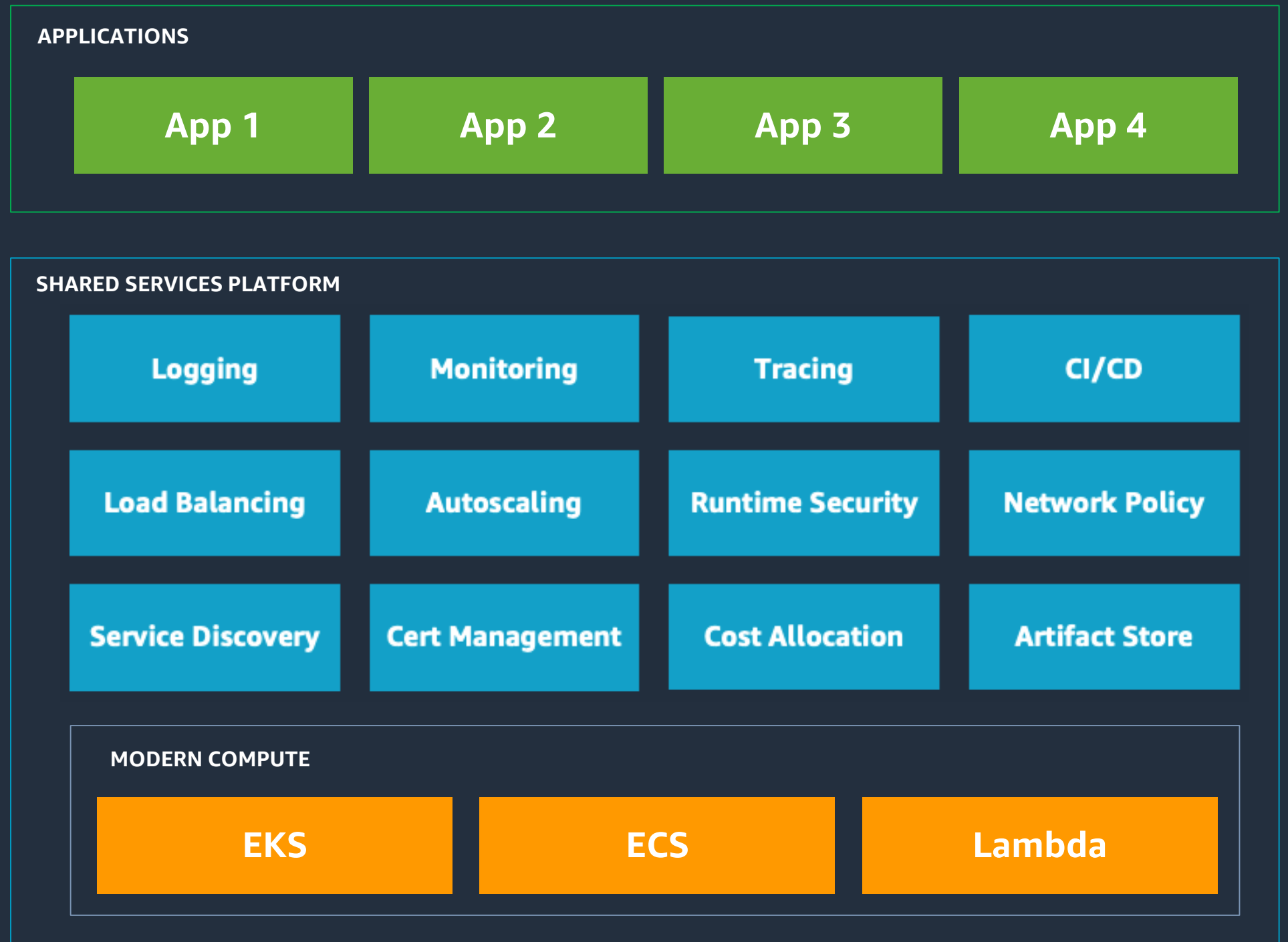


Shared Services Platform

An internal development platform that allows multiple teams to run applications on shared infrastructure that is managed, secured, and governed by a central platform team.

Platform ops

Platform is centrally managed



Separation of concerns in software delivery



Platform Engineers

Build tools that provision, manage and secure the cloud computing infrastructure.

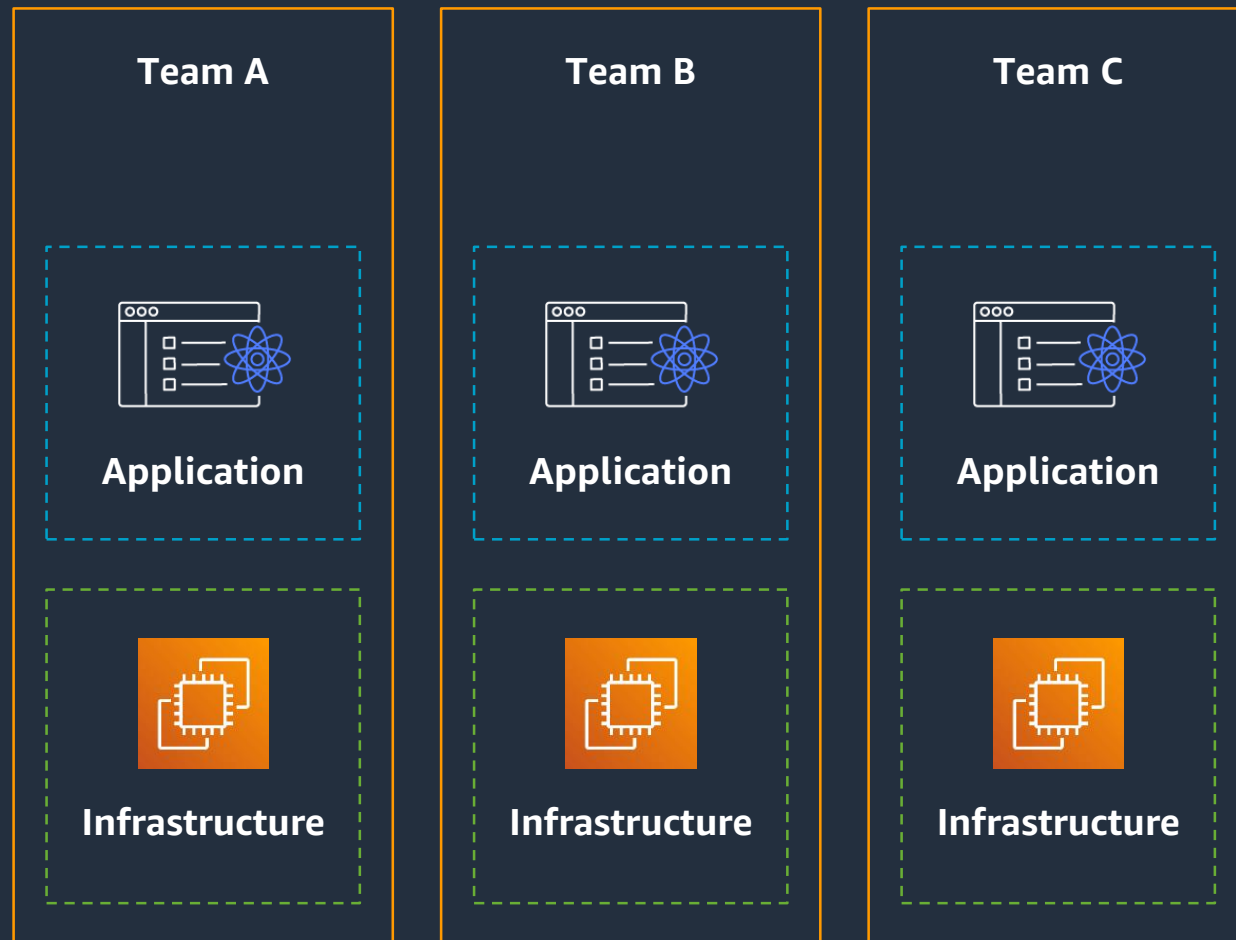


Application Engineers

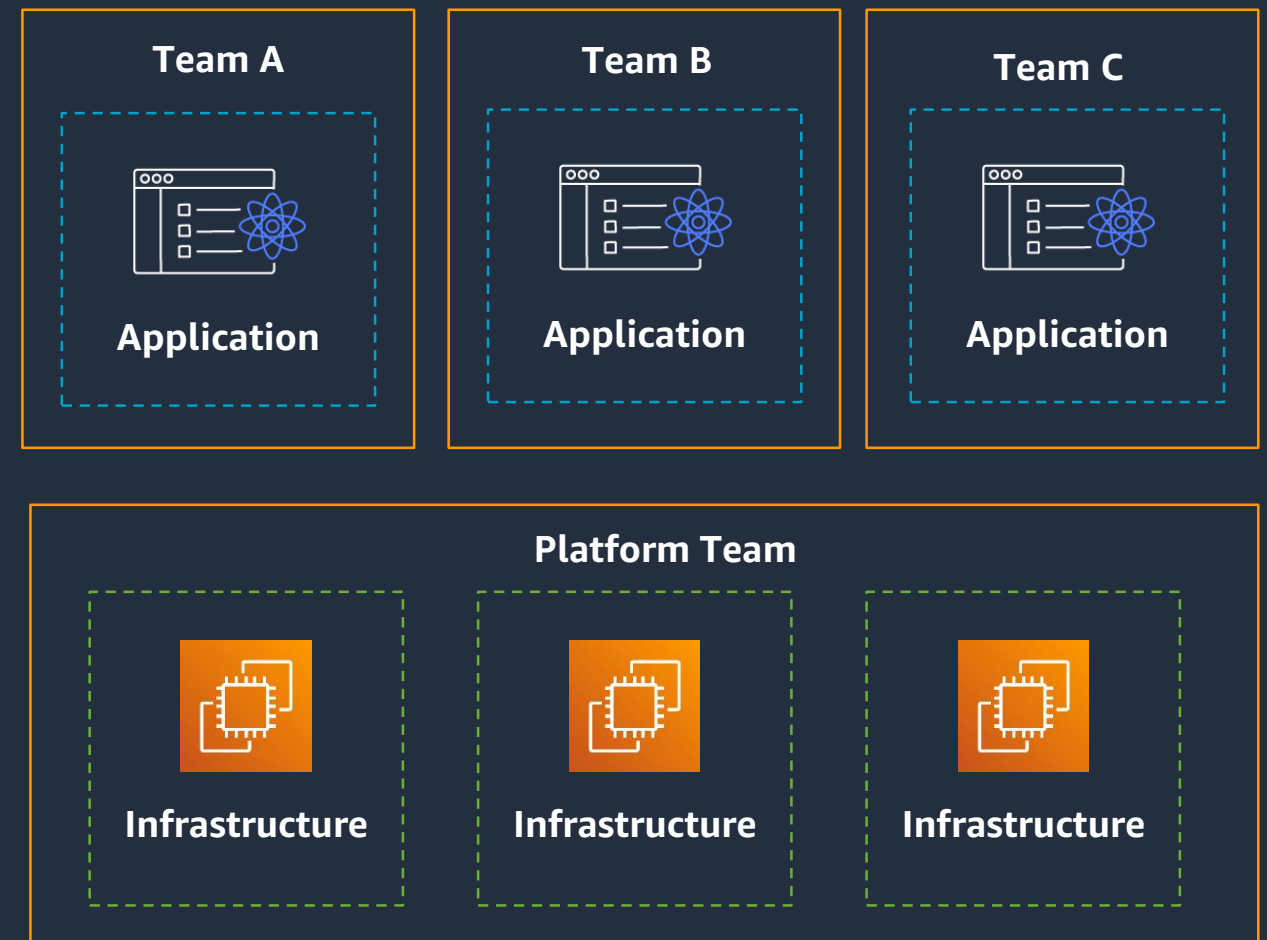
Build applications that deliver business value to customers and internal users.

Platform operations

DevOps



PlatformOps



Platform implementations vary



Developer Experience

Command Line Tools, GUIs,
Pipelines, Manifests



Operating Model

Shared infrastructure v
dedicated infrastructure



Workload Types

Backend Services, Batch
Jobs, Machine Learning

Customer Benefits



Motivations for building an SSP



Velocity

Increase developer productivity and the pace of software delivery.

Automate infrastructure provisioning and deployment.

Decrease time to market for new products and services.



Guardrails

Ensure engineering teams are deploying the right infrastructure for their workloads.

Include security and operational best practices are including in all deployments.



Efficiency

Increase resource utilization by sharing resources among teams.

Reduce operational overhead by centralizing platform management

SSPs enable modern applications at scale

- **Developer self service** – Developers don't need to ask permission to onboard new workloads or wait for other teams.
- **Batteries Included** – All workloads running on the platform inherit best practices and are complete with operational tools.
- **Avoid redundant efforts** – Application teams don't have to re-invent the wheel each time they want to deploy a new service.
- **Economies of scale** - Once a platform is in place, many teams can leverage the same foundation to deploy their workloads.

Customers



Success stories



EKS is one of our strategic directions right now, migrating this account. Requires a very **high secure way** for our developer teams and our devops teams to access our EKS cluster environment.

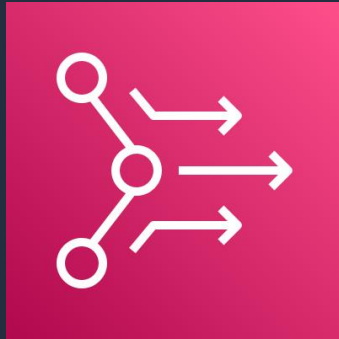


By using Amazon ECS, AQR developers cut the time for building and launching a new app from **6 weeks to 30 minutes**, and automatic security and compliance frameworks have simplified development.

How AWS can help?



AWS services



AWS Proton

Automated management
for container and serverless
deployments



AWS App Runner

Production web
applications at scale made
easy for developers



AWS Copilot CLI

Build, release and operate
production ready
containerized applications

AWS customer enablement

Accelerate time to market for your platform



Actions we recommend

1

Determine if your company could benefit from an SSP

2

Learn about AWS customer SSP implementations

3

Engage your AWS account team, Compute Specialist teams, and partners

4

Schedule an EBC