Red Hat Summit

Connect

aws | Red Hat

Red Hat Summit

Connect

Sviluppare applicazioni moderne su AWS

Leonardo Fenu , Sr. Solutions architect, AWS

aws | Red Hat

What customers ask for?

Build applications,
not infrastructure

Manage infrastructure
to their requirements

Scale quickly
and seamlessly

Security and
isolation by design

Increasing developer productivity
(**innovation**) is the main priority
now for customers

What do our customers want in a modern application?

Improve ROI & Reduce TCO

Scales to millions of users

Increase efficiency of developers

Has global availability

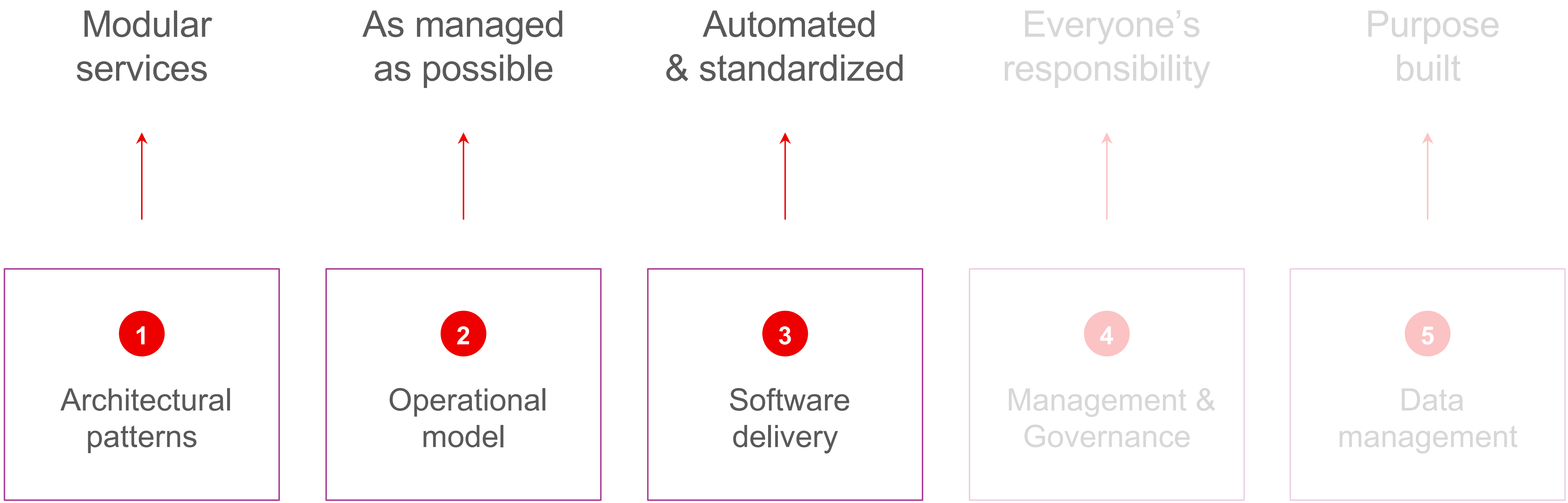
Increase business agility

Responds in milliseconds

Handles petabytes of data

aws | Red Hat

Characteristics of modern applications



Modular services

As managed as possible

Automated & standardized

Everyone’s responsibility

Purpose built

1 Architectural patterns

2 Operational model

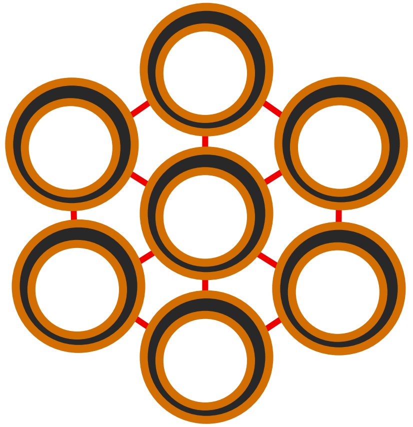
3 Software delivery

4 Management & Governance

5 Data management

What changes do you need to make to adopt these best practices?

Architectural patterns



Microservices

Componentization
Business capabilities
Products not projects
Infrastructure automation

Operational Model

Serverless

provisioning/management
Automatic scaling
Pay for value billing
Availability and resiliency

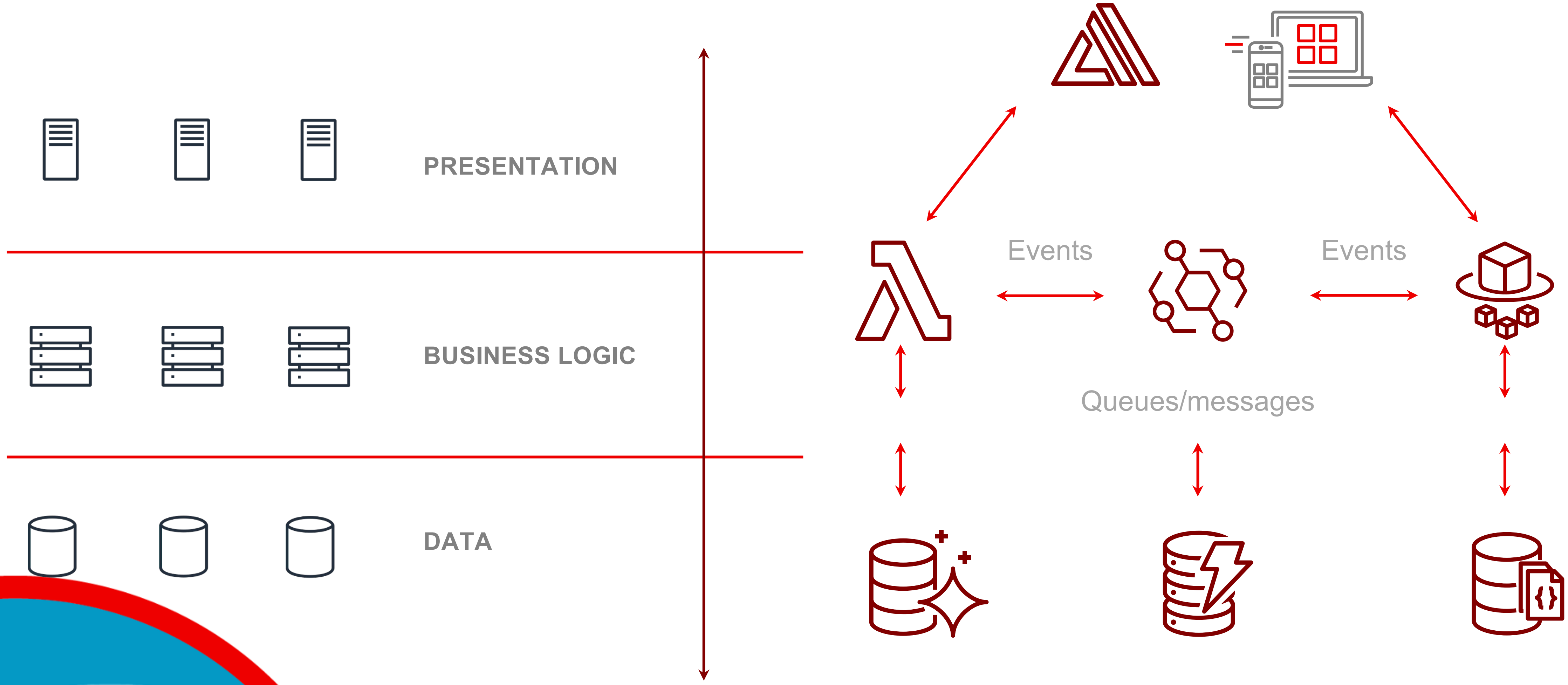
Software Delivery

Velocity

Increase
automation
and iteration

Changes to the architectural patterns

Traditional vs. Modern approach



PRESENTATION

BUSINESS LOGIC

DATA

Events

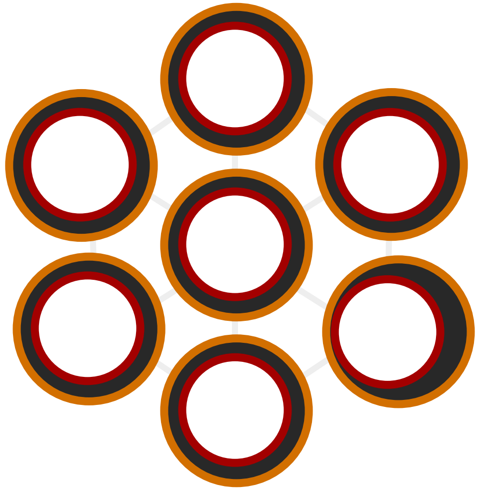
Events

Queues/messages

When the impact of change is small, release velocity can increase

Monolith

Does everything



Microservices

Does one thing

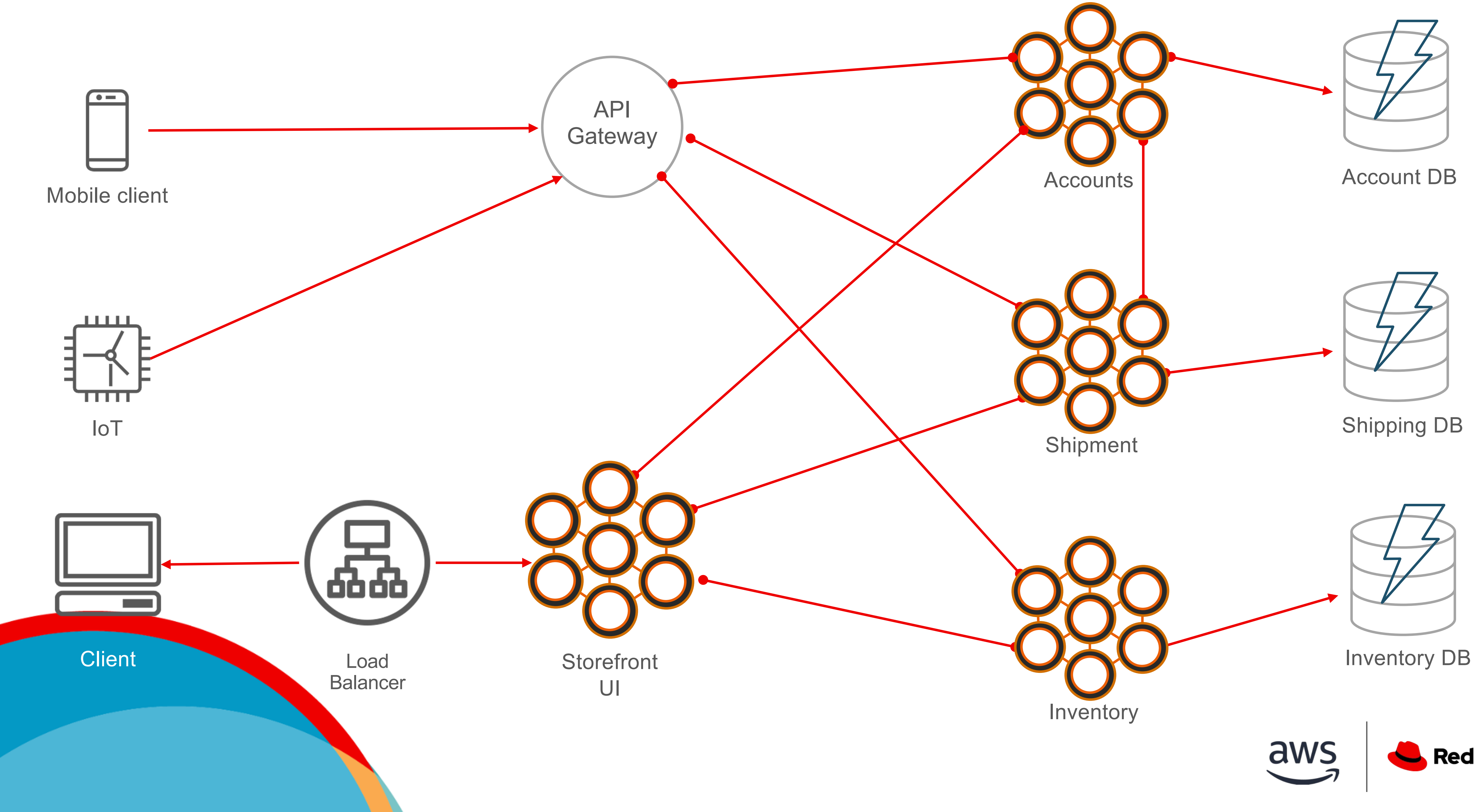
Monoliths are good, but they have some limitations

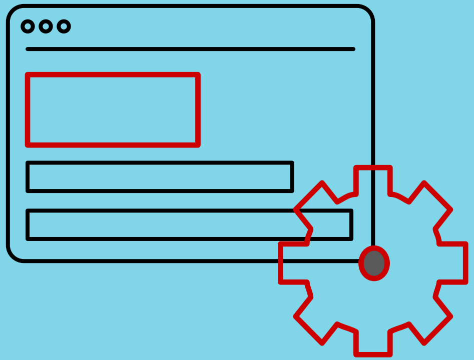
- Coordination overhead
- Poor modularity
- High impact of change
- Poor scalability
- Long build time

Eight reasons to use microservices

- Pick the right tool for the job
- Improve resilience and security
- Lower cost with granular scaling
- Optimize team productivity
- Create new compositions easily
- Experiment and fail safely
- Adopt technology faster
- Deploy features safely and quickly

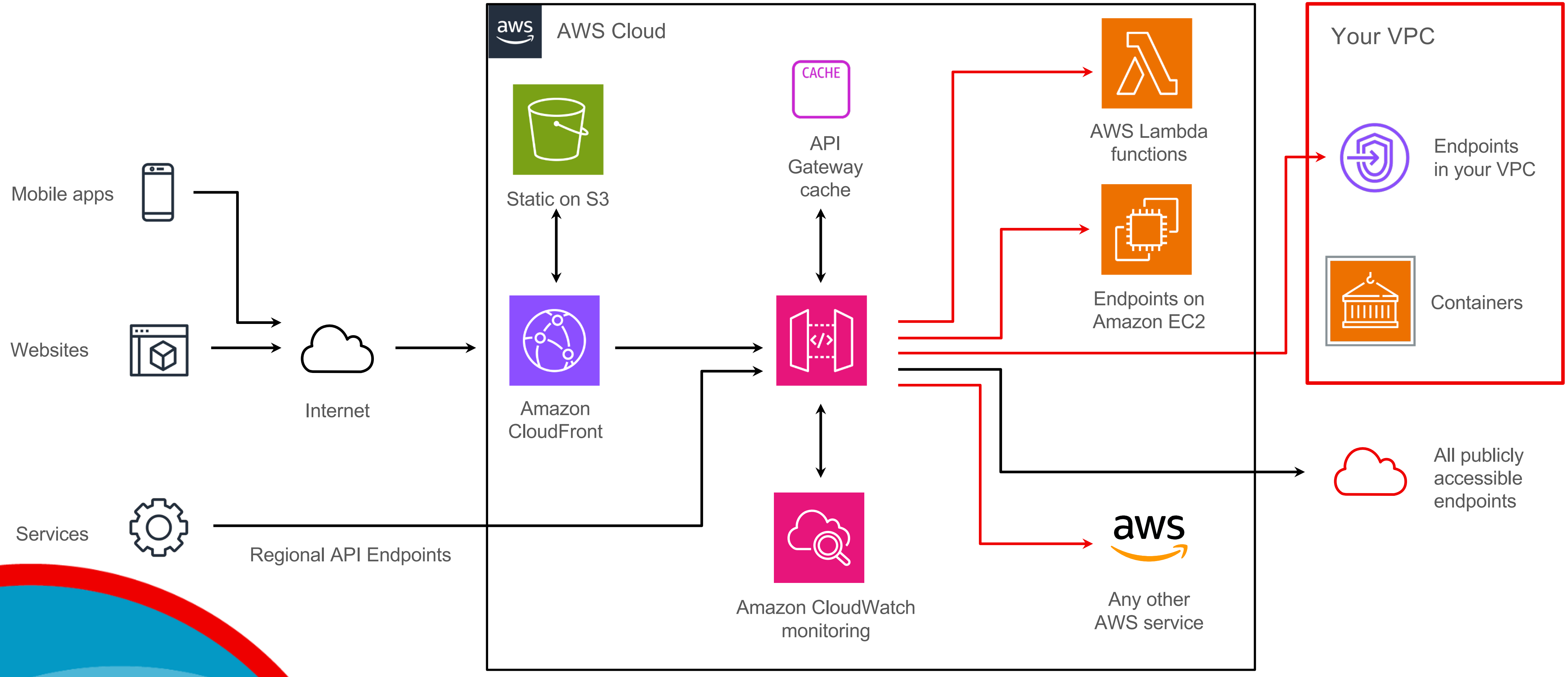
Microservices architectures

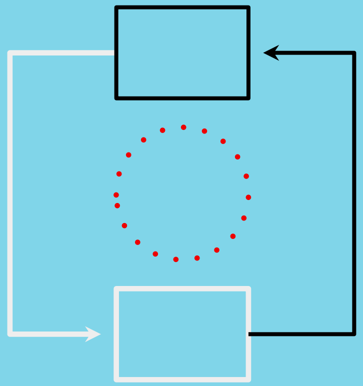




APIs are the front door of microservices

Manage APIs with API Gateway





Event-driven architectures

Decouple state from code using messaging

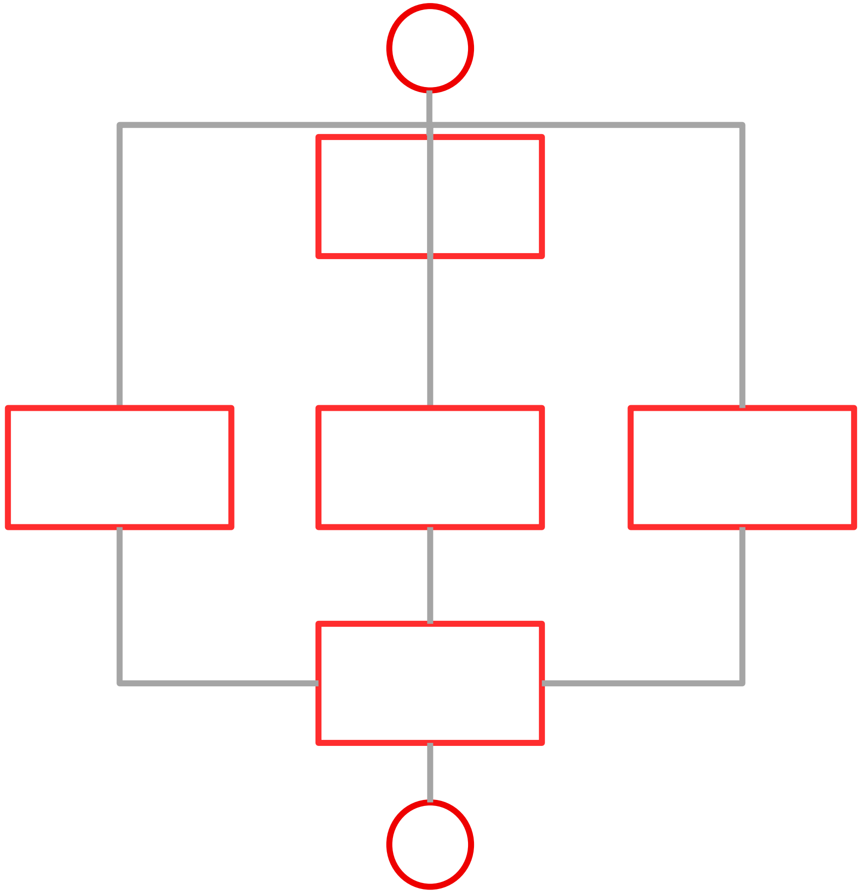
Messagin g

Amazon Simple Queue Service	Amazon Simple Notification Service	Amazon EventBridge
Queues	**Pub/sub**	**Synchronization**
Simple	Simple	Rapid
Fully-managed	Fully-managed	Fully-managed
Any volume	Flexible	Real-time

aws | Red Hat

Build workflows to orchestrate everything

Track status of data and execution



Remove redundant code

Integration options from AWS

Client-to-Service

Amazon API Gateway

Connect
Efficient
Fully-managed
Real-time

Messaging

Amazon SQS

Queues
Simple
Fully-managed
Any volume

Amazon SNS

Pub/sub
Simple
Fully-managed
Flexible

Amazon EventBridge

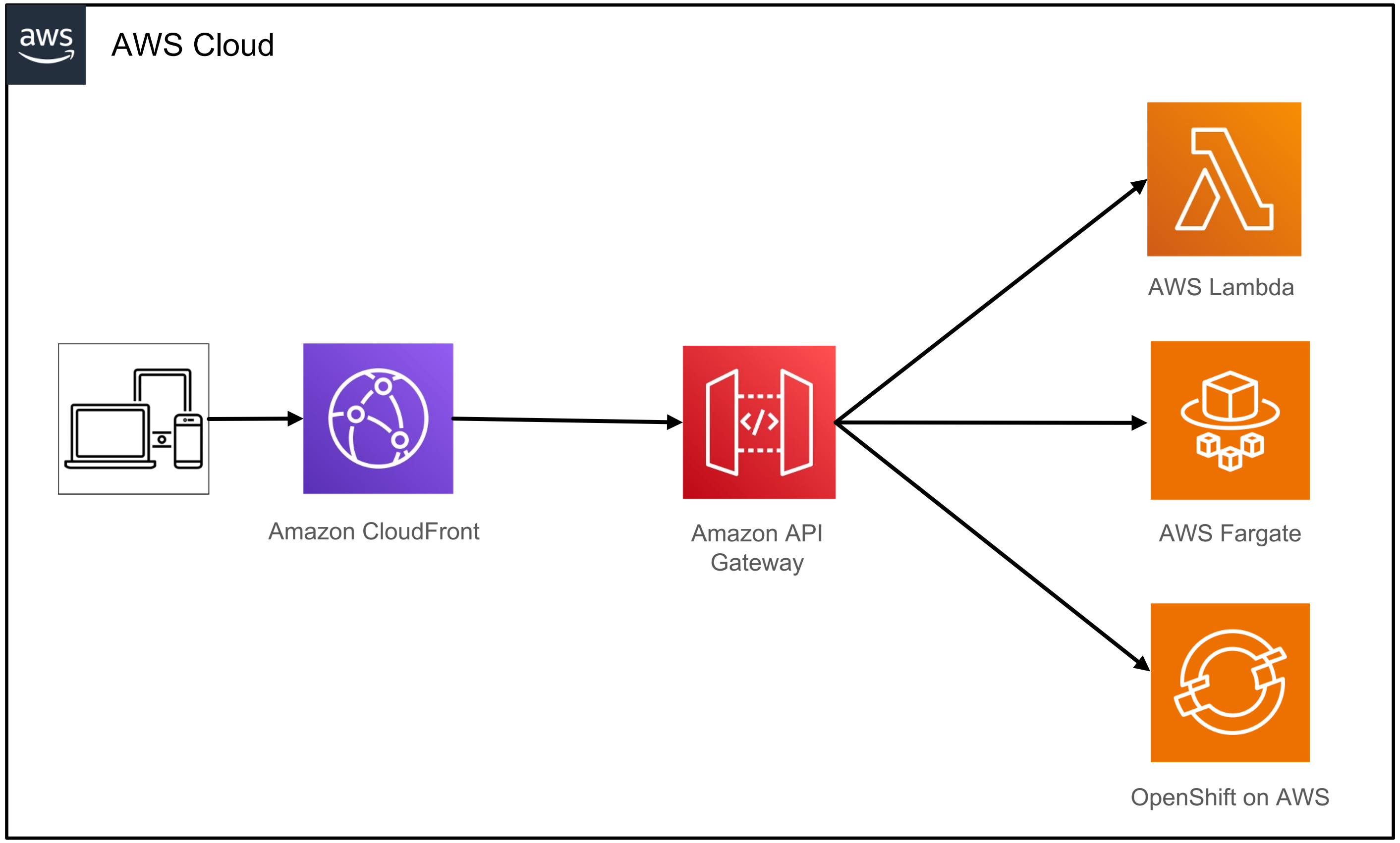
Synchronization
Rapid
Fully-managed
Real-time

Orchestration

AWS Step Functions

Orchestration
Powerful
Fully-managed
Low code

Small pieces, loosely joined

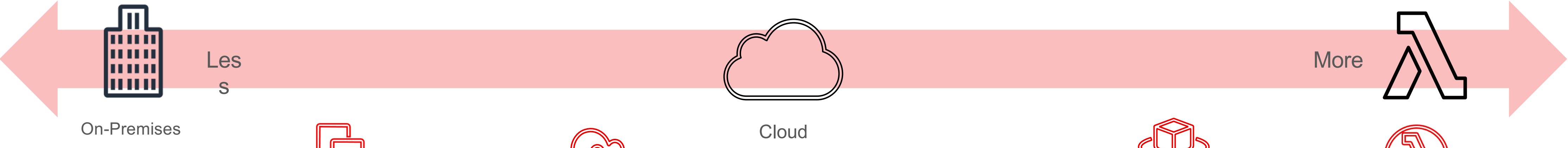


Microservices are better, but they come with challenges

- More moving parts
- Integration becomes a priority
- Deployment, monitoring, and security needs evolve
- Requires organization and culture change

Changes to the operational model

AWS operational responsibility models



Compute	Virtual Machine	EC2	Elastic Beanstalk		Fargate	AWS Lambda
Databases	MySQL	MySQL on EC2	RDS MySQL	RDS Aurora	Aurora Serverless	DynamoDB
Storage	Storage					S3
Messaging	ESBs		Amazon MQ	Kinesis		SQS / SNS

Compute on AWS

+
Opinionated
-

	AWS manages	Customer manages
AWS Lambda Serverless functions	• Data source integrations • Physical hardware, software, networking, and facilities • Provisioning	• Application code
AWS Fargate Serverless containers	• Container orchestration, provisioning • Cluster scaling • Physical hardware, host OS/kernel, networking, and facilities	• Application code • Data source integrations • Security config and updates, network config, management tasks
ECS/EKS/ROSA Container-management as a service	• Container orchestration control plane • Physical hardware software, networking, and facilities	• Application code • Data source integrations • Work clusters • Security config and updates, network config, firewall, management tasks
EC2 Infrastructure-as-a-Service	• Physical hardware software, networking, and facilities	• Application code • Data source integrations • Scaling • Security config and updates, network config, management tasks • Provisioning, managing scaling and patching of servers

Modern Application Computing Services Landscape

	Build your Own Application Platform		Turn-key Application Platform
Application Platform Accelerate and standardize application Management	AWS Proton AWS App Runner EKS Blueprints	AWS X-Ray Cloud Watch Amazon Managed Prometheus	ROS A Red Hat OpenShift Service on AWS
Containers Orchestration Deployment, scheduling, and scaling, containerized applications	Amazon Elastic Container (Amazon ECS)	Amazon Elastic Kubernetes Service (Amazon EKS)	
Containers Infrastructure Registry, Networking, CI/CD	Amazon Elastic Container Registry (Amazon ECR) AWS Cloud Map	AWS App Mesh AWS CodeCatalyst	
Compute	Elastic Compute Cloud (Amazon EC2)	Fargate	

Third-party tooling

AWS and Red Hat Partnership Helps Customers Meet Digital Needs

Red Hat and AWS are industry leaders with extensive experience in **IT infrastructure, hybrid cloud, digital transformation, and open source innovation.**

Through **collaborative engineering** activities, they offer integrated, certified solutions to meet modern, digital business needs.

Consistent, enterprise-grade platforms with advanced security and management features help organizations build IT infrastructure that supports their business efficiently and cost-effectively and adapts on their schedule.

“Given that Red Hat is the world’s leading provider of open-source solutions, our enterprise customers have been passionate about seamlessly running Red Hat Enterprise Linux and various other Red Hat solutions on AWS.”

Andy Jassy | CEO, Amazon

Red Hat and AWS by the numbers

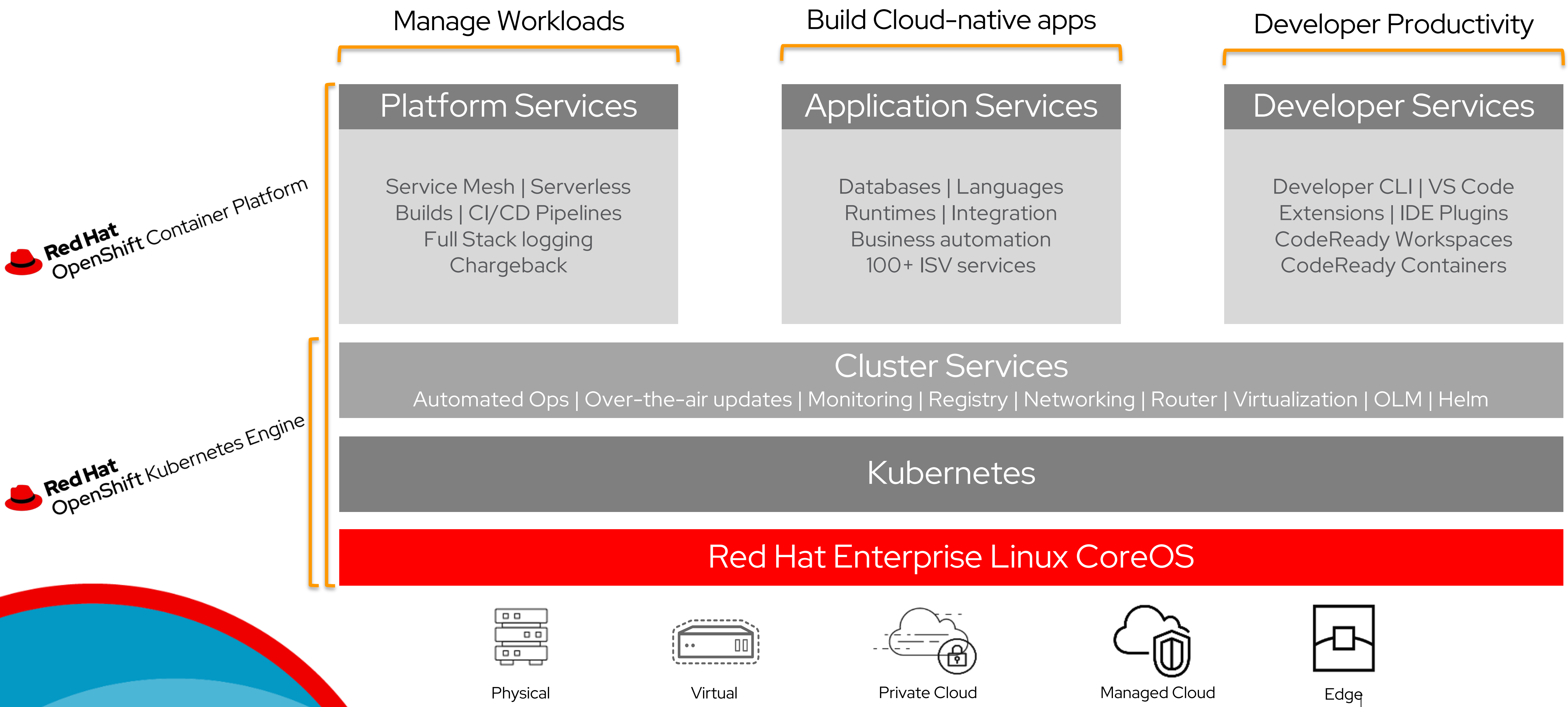
Partners since

2008

>60,000

of AWS customers consume Red Hat products and solutions

What is Red Hat OpenShift? – Opinionated Kubernetes based Platform



Manage Workloads

Build Cloud-native apps

Developer Productivity

Red Hat OpenShift Container Platform

Red Hat OpenShift Kubernetes Engine

Platform Services

Service Mesh | Serverless
Builds | CI/CD Pipelines
Full Stack logging
Chargeback

Application Services

Databases | Languages
Runtimes | Integration
Business automation
100+ ISV services

Developer Services

Developer CLI | VS Code
Extensions | IDE Plugins
CodeReady Workspaces
CodeReady Containers

Cluster Services

Automated Ops | Over-the-air updates | Monitoring | Registry | Networking | Router | Virtualization | OLM | Helm

Kubernetes

Red Hat Enterprise Linux CoreOS

Physical

Virtual

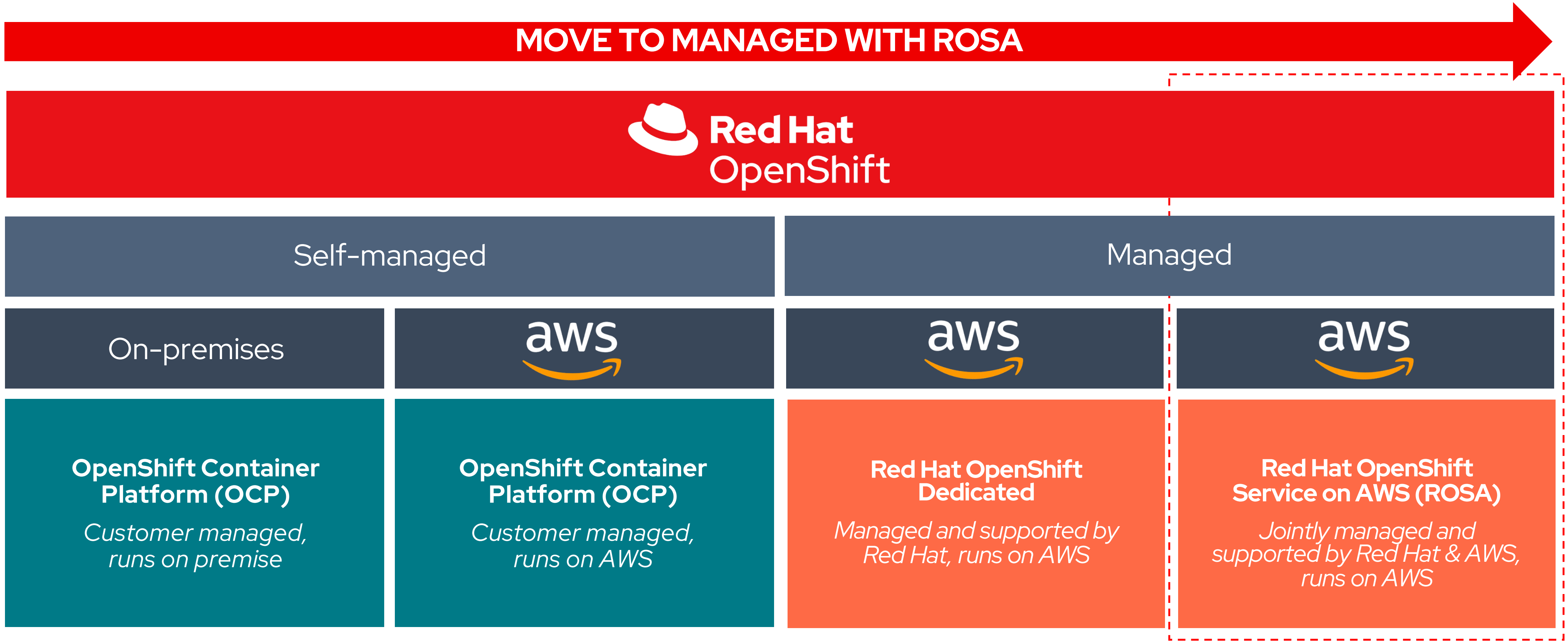
Private Cloud

Managed Cloud

Edge

aws | Red Hat

Openshift managed services evolution on AWS



Self-managed		Managed	
On-premises	aws	aws	aws
OpenShift Container Platform (OCP) *Customer managed, runs on premise*	**OpenShift Container Platform (OCP)** *Customer managed, runs on AWS*	**Red Hat OpenShift Dedicated** *Managed and supported by Red Hat, runs on AWS*	**Red Hat OpenShift Service on AWS (ROSA)** *Jointly managed and supported by Red Hat & AWS, runs on AWS*

Situations we find in customers more frequently than expected

Customers move applications back on-premises from the cloud as the do not get the desired ROIs.

CIOs had bad experiences in migrations and frequently they do not meet deadlines.

Managed Service solves it!

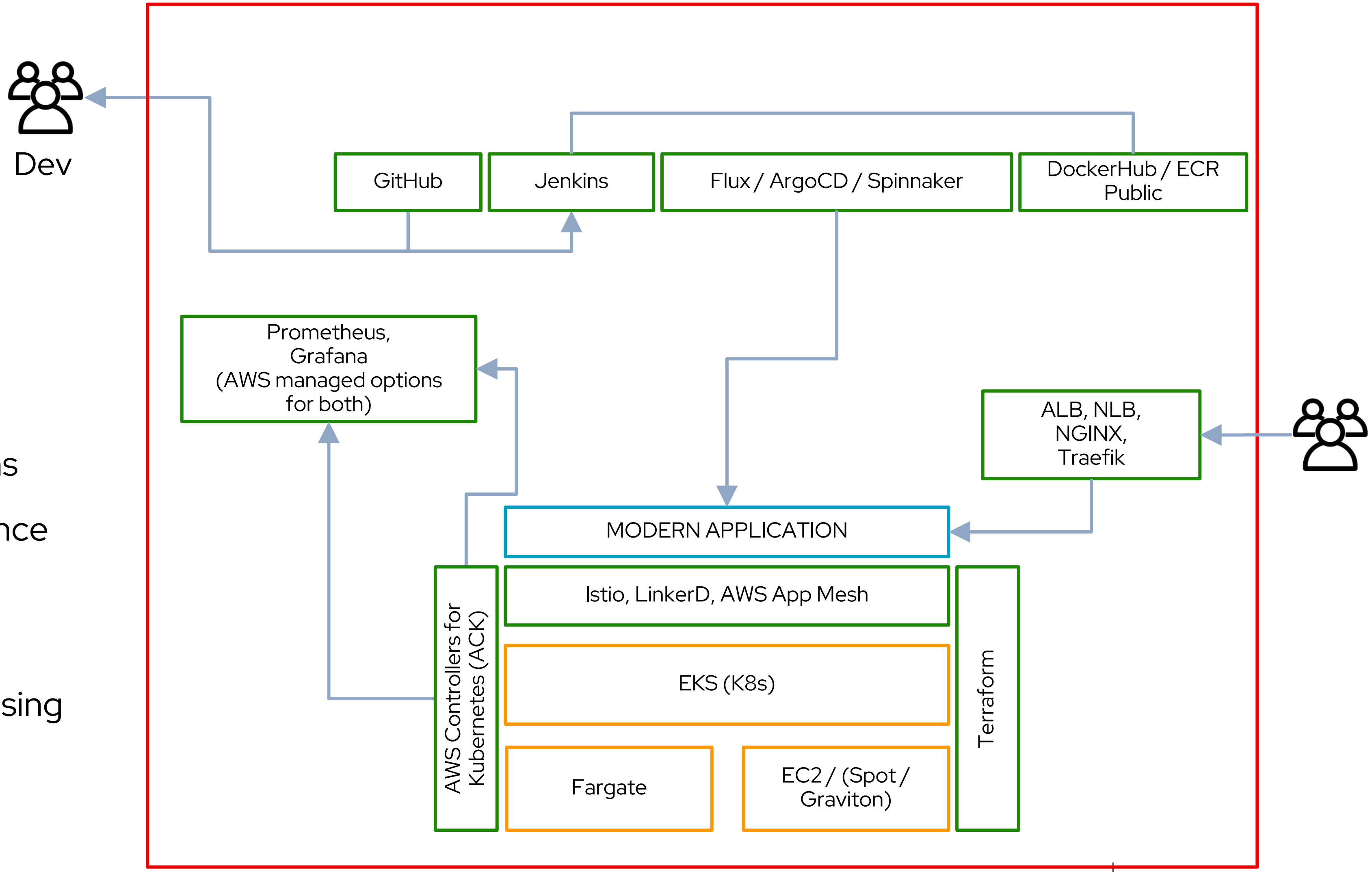
aws | Red Hat

AWS Elastic Kubernetes services: **EKS**

Amazon EKS

The Opensource flexibility

- AWS-optimized managed upstream Kubernetes with four supported versions
- Build your custom platform for compliance and security, with AWS services and community solutions
- Accelerate your containerization and modernization with canonical patterns using AWS Blueprints

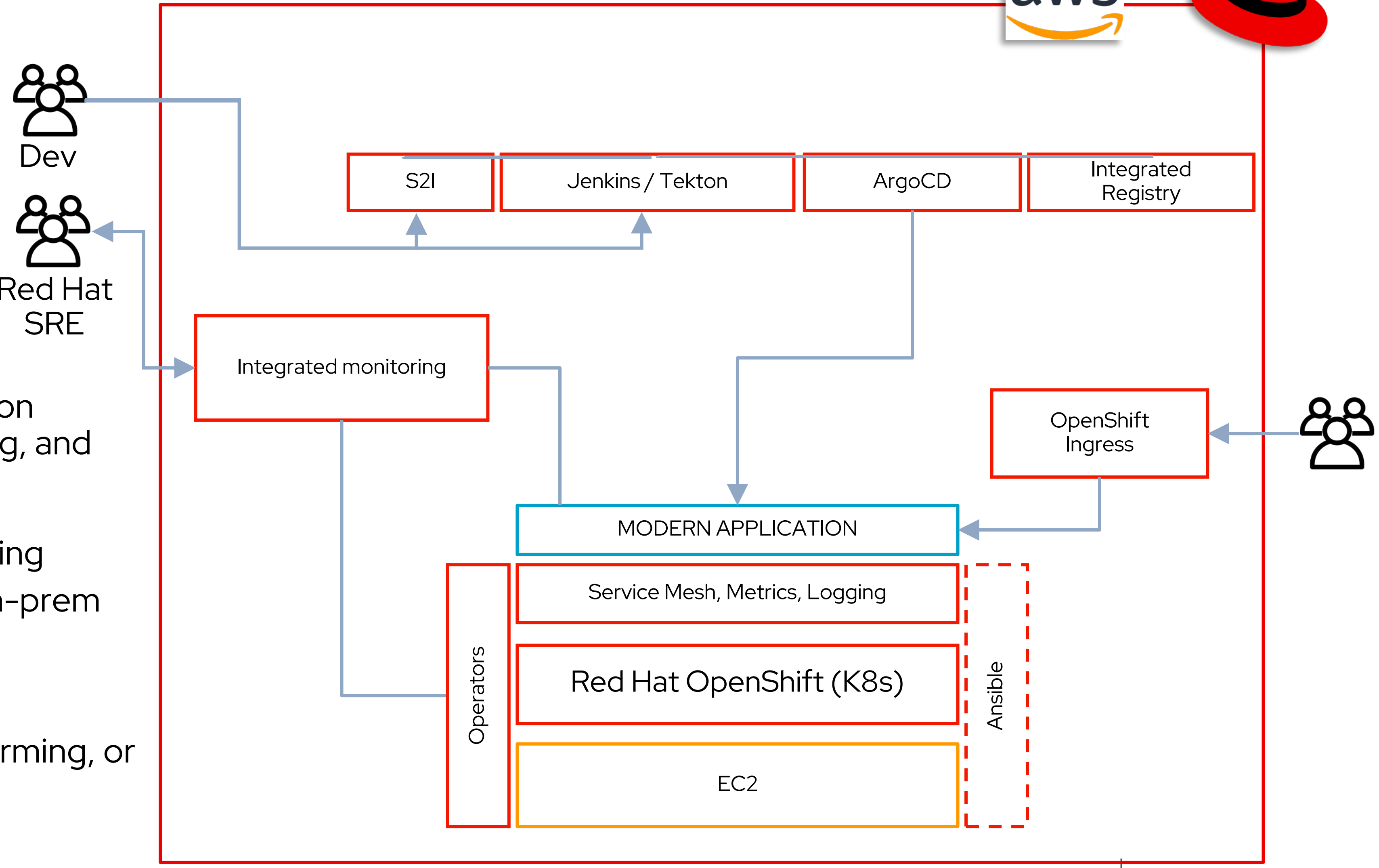

Dev
GitHub
Jenkins
Flux / ArgoCD / Spinnaker
DockerHub / ECR Public
Prometheus, Grafana (AWS managed options for both)
ALB, NLB, NGINX, Traefik
MODERN APPLICATION
Istio, LinkerD, AWS App Mesh
EKS (K8s)
Fargate
EC2 / (Spot / Graviton)
AWS Controllers for Kubernetes (ACK)
Terraform

Red Hat OpenShift on AWS: **ROSA**

Red Hat OpenShift on AWS

Shared Services Platform AKA PaaS

- Integrated Kubernetes based application platform with built-in CI/CD, monitoring, and developer tools.
- Activate ROSA and continue with existing OpenShift skills and processes from on-prem environments to the cloud
- Accelerate application migration and modernization by re-hosting, re-platforming, or re-factoring workloads



*ROSA visual explanation in 12min video

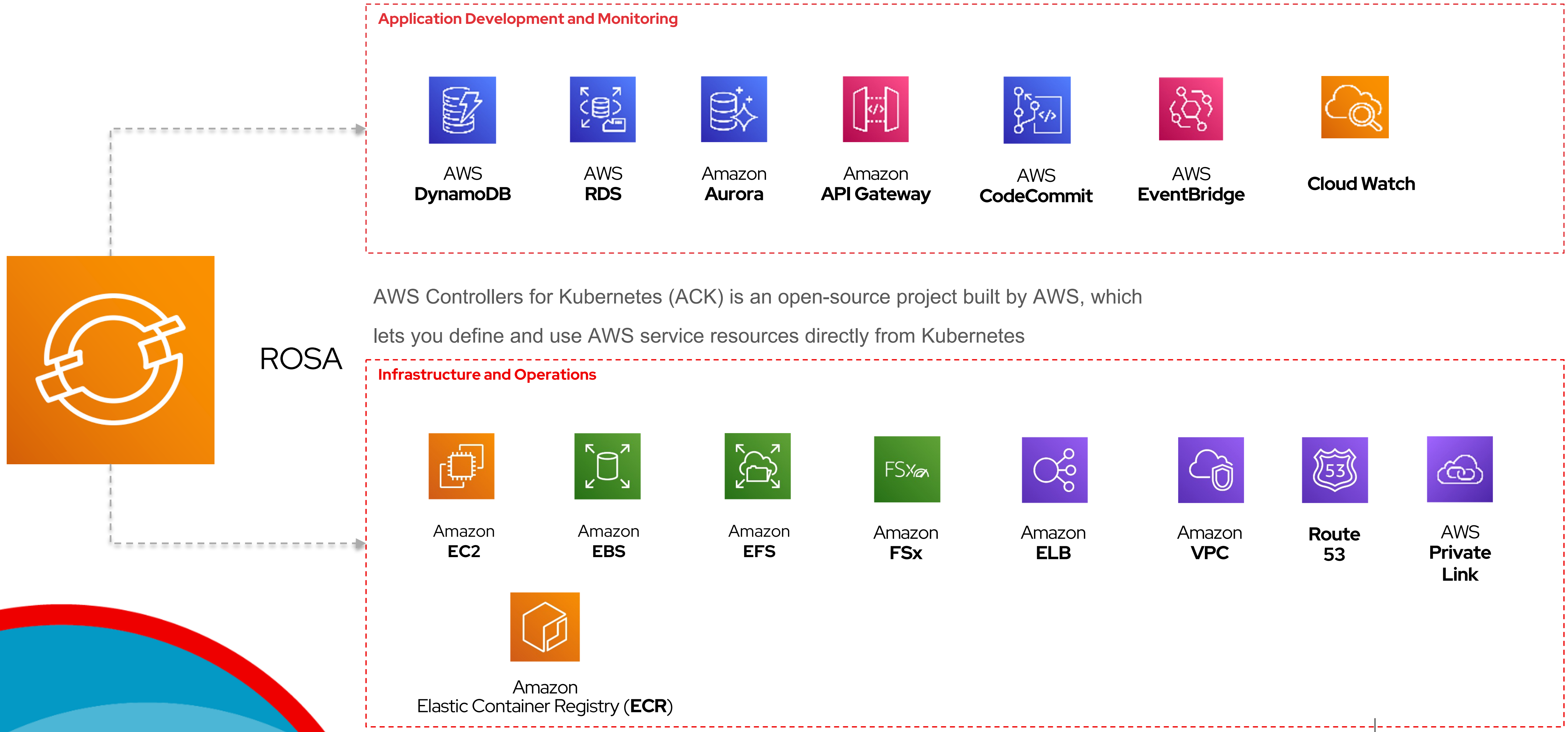
Focus on innovation to add value to business

	On-premises	Cloud	
	OpenShift Container Platform (OCP)	**OpenShift Container Platform (OCP) on AWS**	**Red Hat OpenShift Service on AWS (ROSA)***
Control Plane	Customer	Customer	Red Hat
Compute	Customer	Customer	Red Hat
Data Plane	Customer	Customer	Red Hat
Support	Red Hat	Red Hat	Red Hat aws
Billing	Red Hat	Red Hat	aws

Fully Managed

* Detailed Responsibility Matrix: https://docs.openshift.com/rosa/rosa_policy/rosa-policy-responsibility-matrix.html

Integrated AWS services



ROSA

Application Development and Monitoring

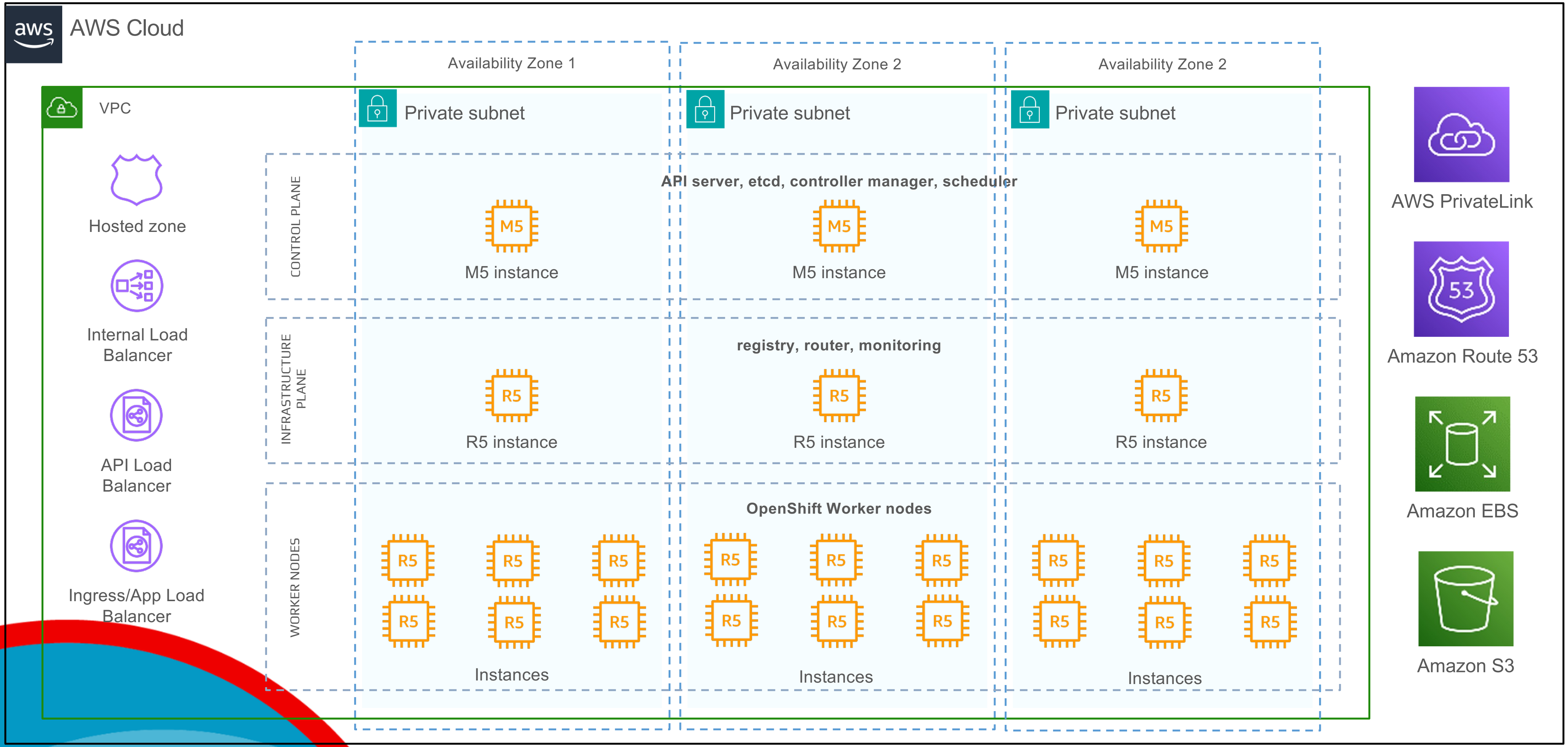
- AWS **DynamoDB**
- AWS **RDS**
- Amazon **Aurora**
- Amazon **API Gateway**
- AWS **CodeCommit**
- AWS **EventBridge**
- **Cloud Watch**

AWS Controllers for Kubernetes (ACK) is an open-source project built by AWS, which lets you define and use AWS service resources directly from Kubernetes

Infrastructure and Operations

- Amazon **EC2**
- Amazon **EBS**
- Amazon **EFS**
- Amazon **FSx**
- Amazon **ELB**
- Amazon **VPC**
- **Route 53**
- AWS **Private Link**
- Amazon Elastic Container Registry (**ECR**)

Rosa Cluster Architecture example



Benefits of ROSA – Turn-Key Application Platform

Developers

Fully managed clusters in minutes to build, deploy, and run applications using built-in developer UI that abstract the complexity of Kubernetes.

Collaborate across teams via shared projects.

Administrators

Standardized and streamlined operations across on-prem and cloud environments.

Built-in monitoring, logging, and networking

Choose platform version upgrades as required for the business.*

Business Leaders

Consolidated billing and cost management across the business.

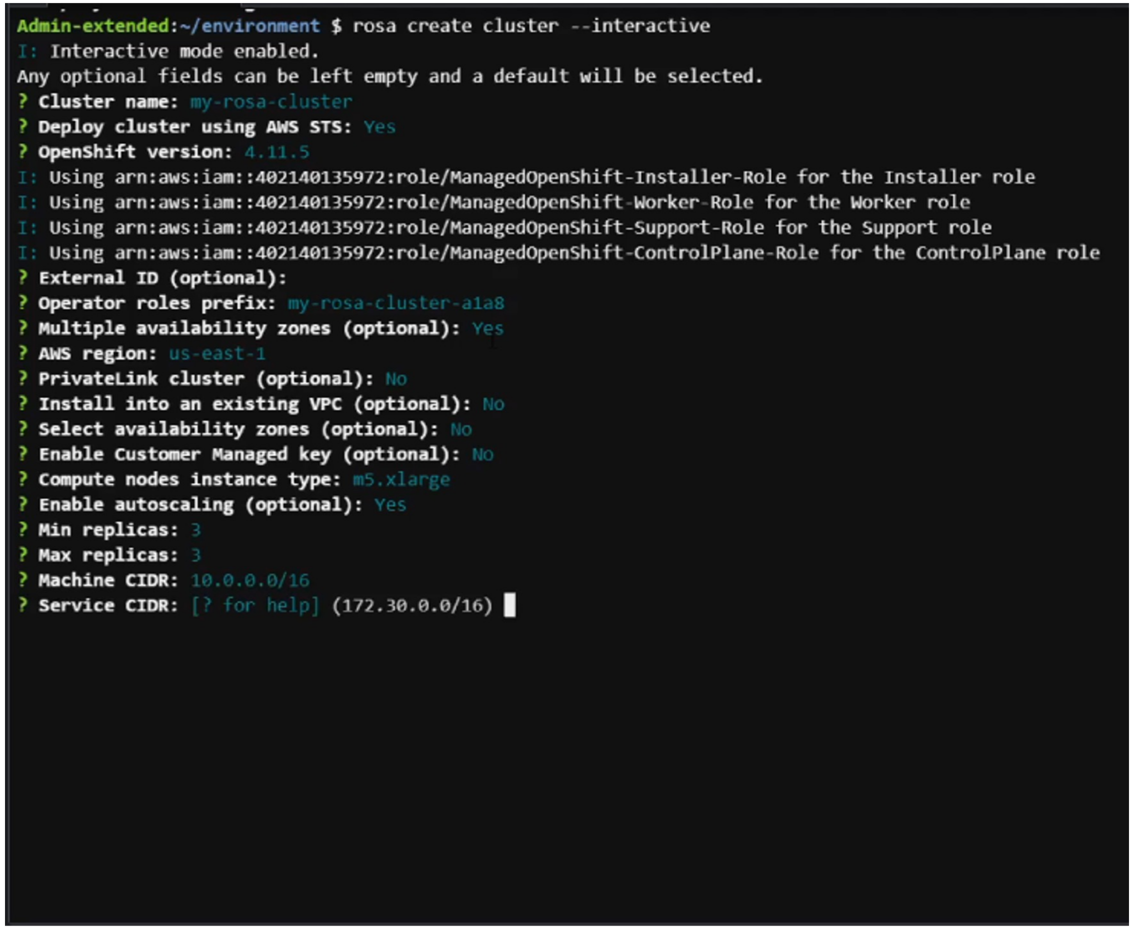
Consumption based pricing for surge and R&D usage.

24x7 full-stack management and support

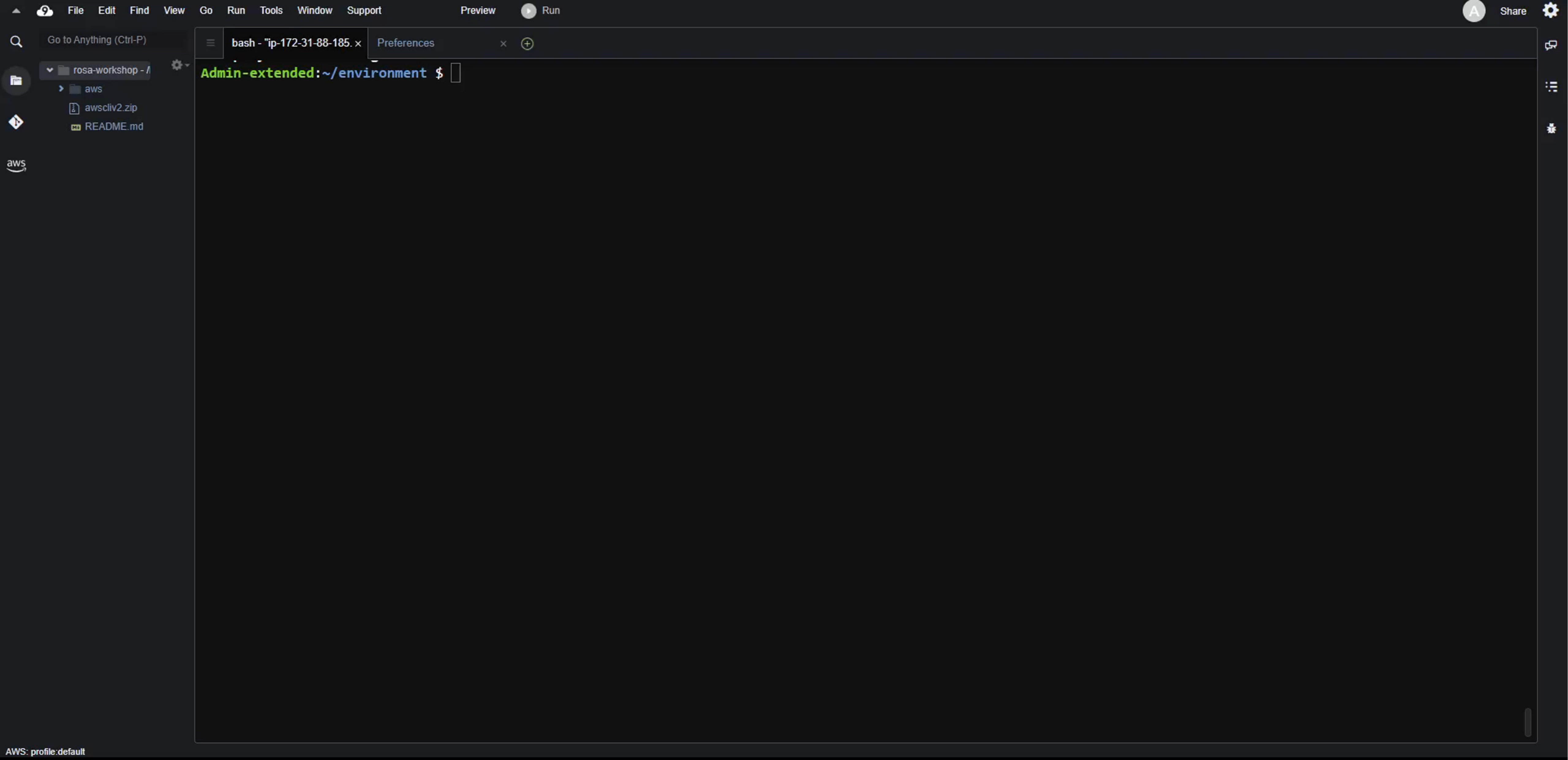
*Security patches are applied automatically – major and minor version upgrades are customer controlled

Quick demo

Create a ROSA cluster: Interactive mode



```
Admin-extended:~/environment $ rosa create cluster --interactive
I: Interactive mode enabled.
Any optional fields can be left empty and a default will be selected.
? Cluster name: my-rosa-cluster
? Deploy cluster using AWS STS: Yes
? OpenShift version: 4.11.5
I: Using arn:aws:iam::402140135972:role/ManagedOpenShift-Installer-Role for the Installer role
I: Using arn:aws:iam::402140135972:role/ManagedOpenShift-Worker-Role for the Worker role
I: Using arn:aws:iam::402140135972:role/ManagedOpenShift-Support-Role for the Support role
I: Using arn:aws:iam::402140135972:role/ManagedOpenShift-ControlPlane-Role for the ControlPlane role
? External ID (optional):
? Operator roles prefix: my-rosa-cluster-a1a8
? Multiple availability zones (optional): Yes
? AWS region: us-east-1
? PrivateLink cluster (optional): No
? Install into an existing VPC (optional): No
? Select availability zones (optional): No
? Enable Customer Managed key (optional): No
? Compute nodes instance type: m5.xlarge
? Enable autoscaling (optional): Yes
? Min replicas: 3
? Max replicas: 3
? Machine CIDR: 10.0.0.0/16
? Service CIDR: [? for help] (172.30.0.0/16)
```



```
Admin-extended:~/environment $
```

Red Hat Summit

Connect

Q&A?

aws

Red Hat

Red Hat Summit

Connect

Grazie!

aws

Red Hat