Swiss Cloud Day

ZURICH | 29 SEPTEMBER 2022

MAD204

Making sense of AWS container services

René Penkert (he/him)
Professional Services Consultant
AWS

Sascha Möllering (he/him)
Container Specialist Solution Architect
AWS

Why containers?

Containers are very **lightweight** and smaller than VMs

Orders of magnitude **faster to start** compared with VMs

More **efficient** because multiple applications can share the same OS kernel

Container image **packaging** reduces dependency management overhead

Offers **immutability** and version control

Build once, run anywhere

AWS customers are broad and diverse

AWS serves millions of customers

They come from mixed backgrounds and industries

They want to optimize for different dimensions

1. **Simplicity**
2. **Flexibility**
3. **Agility**
4. **Hybridity**

There is no single containers-based solution that can serve them all

A container service for everyone

AWS services that run containers fit into several categories

- Run traditional long-running applications and provide **flexibility** and **extensibility**
- Run traditional long-running applications and offer **simplicity** and low operational overhead
- Run containers offering a **serverless**, pay-for-use billing model
- Run **event-driven** applications
- Run containers offering a **platform-as-a-service** experience
- Run containers without the need to supply a container image

. . . and more

Services and tools for deploying containers

Compute capacity

AWS Fargate

Amazon EC2

AWS Outposts

AWS Wavelength

Servers

Serverless

Elastic compute

On premises and edge

Services and tools for deploying containers

Container orchestration

Amazon EKS

Amazon ECS

Red Hat OpenShift Service on AWS (ROSA)

Compute capacity

AWS Fargate

Amazon EC2

AWS Outposts

AWS Wavelength

Servers

Serverless

Elastic compute

On premises and edge

Services and tools for deploying containers

Opinionated deployment

AWS App Runner

AWS Elastic Beanstalk

AWS IoT Greengrass

AWS Batch

AWS Amplify

Amazon Lightsail

Docker Compose

AWS Copilot

Container orchestration

Amazon EKS

Amazon ECS

Red Hat OpenShift Service on AWS (ROSA)

Compute capacity

AWS Fargate

Amazon EC2

AWS Outposts

AWS Wavelength

Servers

Serverless

Elastic compute

On premises and edge

AWS Lambda

Services and tools for deploying containers

Opinionated deployment

AWS App Runner

AWS Batch

Docker Compose

Container orchestration

Amazon EKS

Amazon ECS

Compute capacity

AWS Fargate

Serverless

Amazon EC2

Elastic compute

AWS Outposts

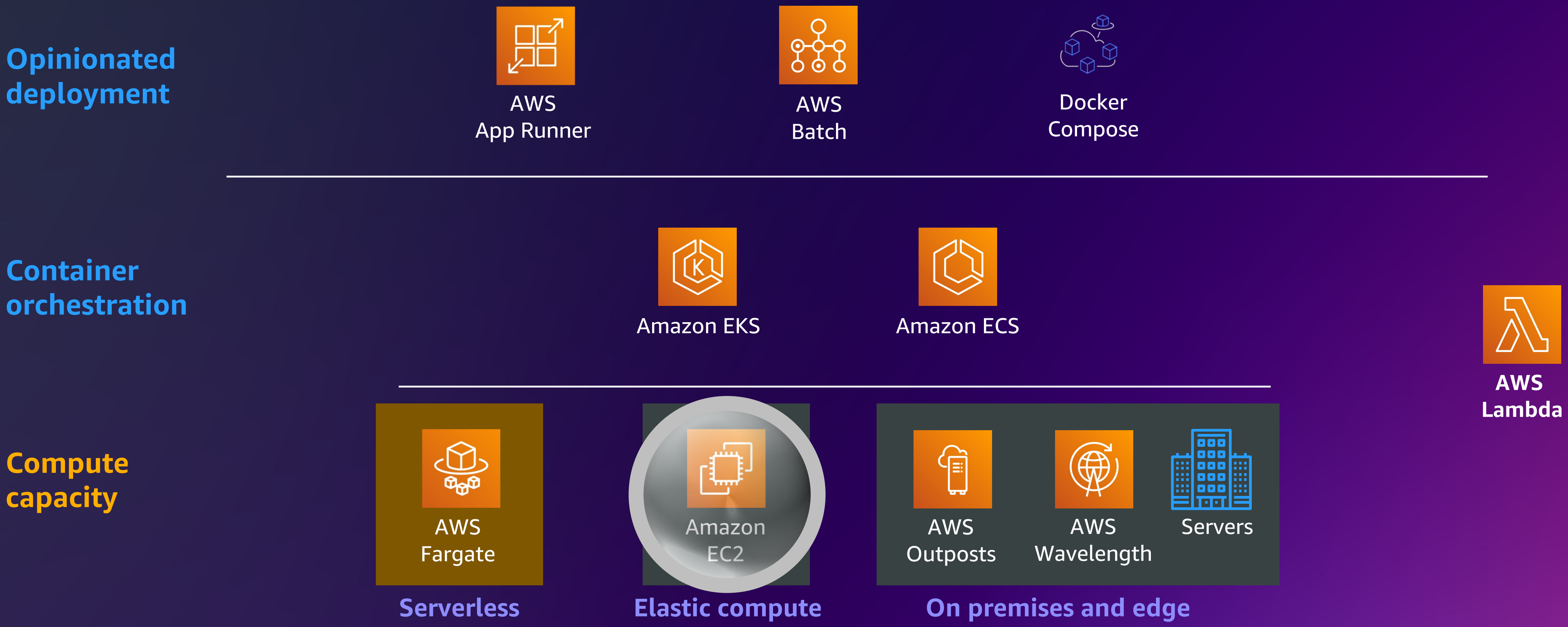
AWS Wavelength

Servers

On premises and edge

AWS Lambda

Closer look at EC2 runtime for Amazon EKS & ECS



Opinionated deployment

AWS App Runner

AWS Batch

Docker Compose

Container orchestration

Amazon EKS

Amazon ECS

AWS Lambda

Compute capacity

AWS Fargate

Serverless

Amazon EC2

Elastic compute

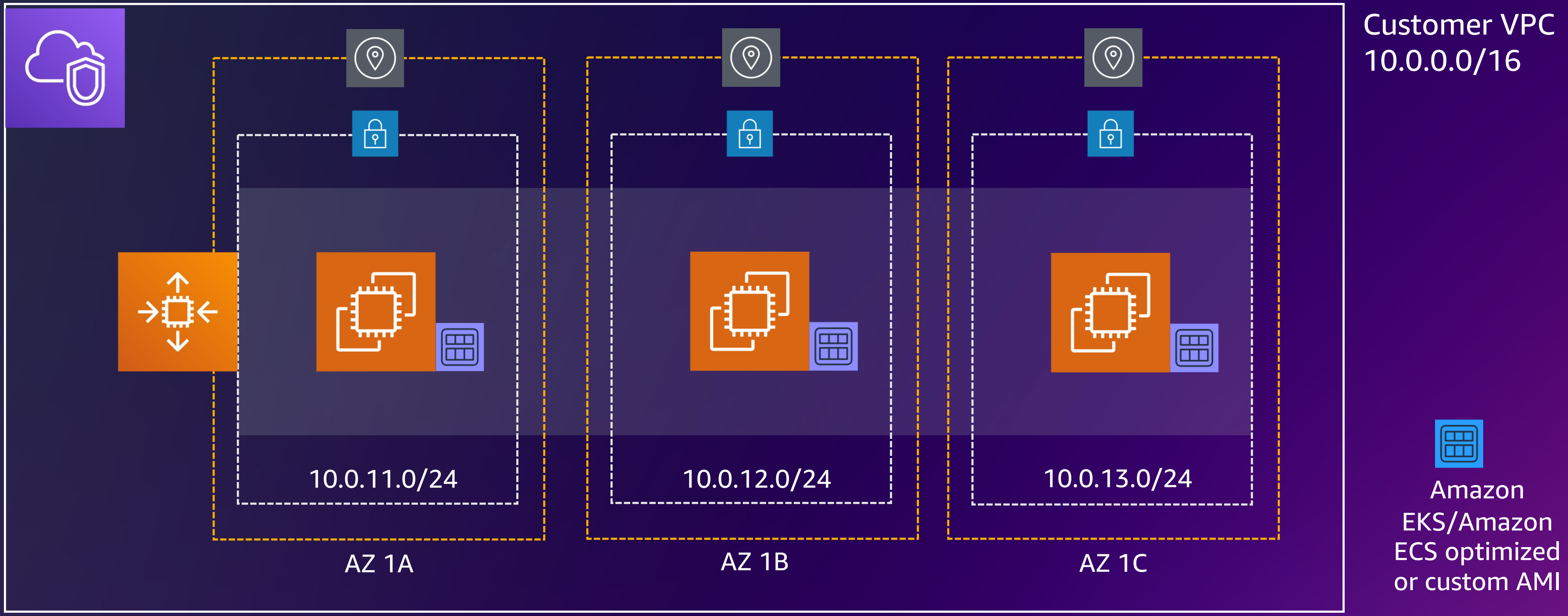
AWS Outposts

AWS Wavelength

Servers

On premises and edge

Amazon ECS and EKS with Amazon EC2

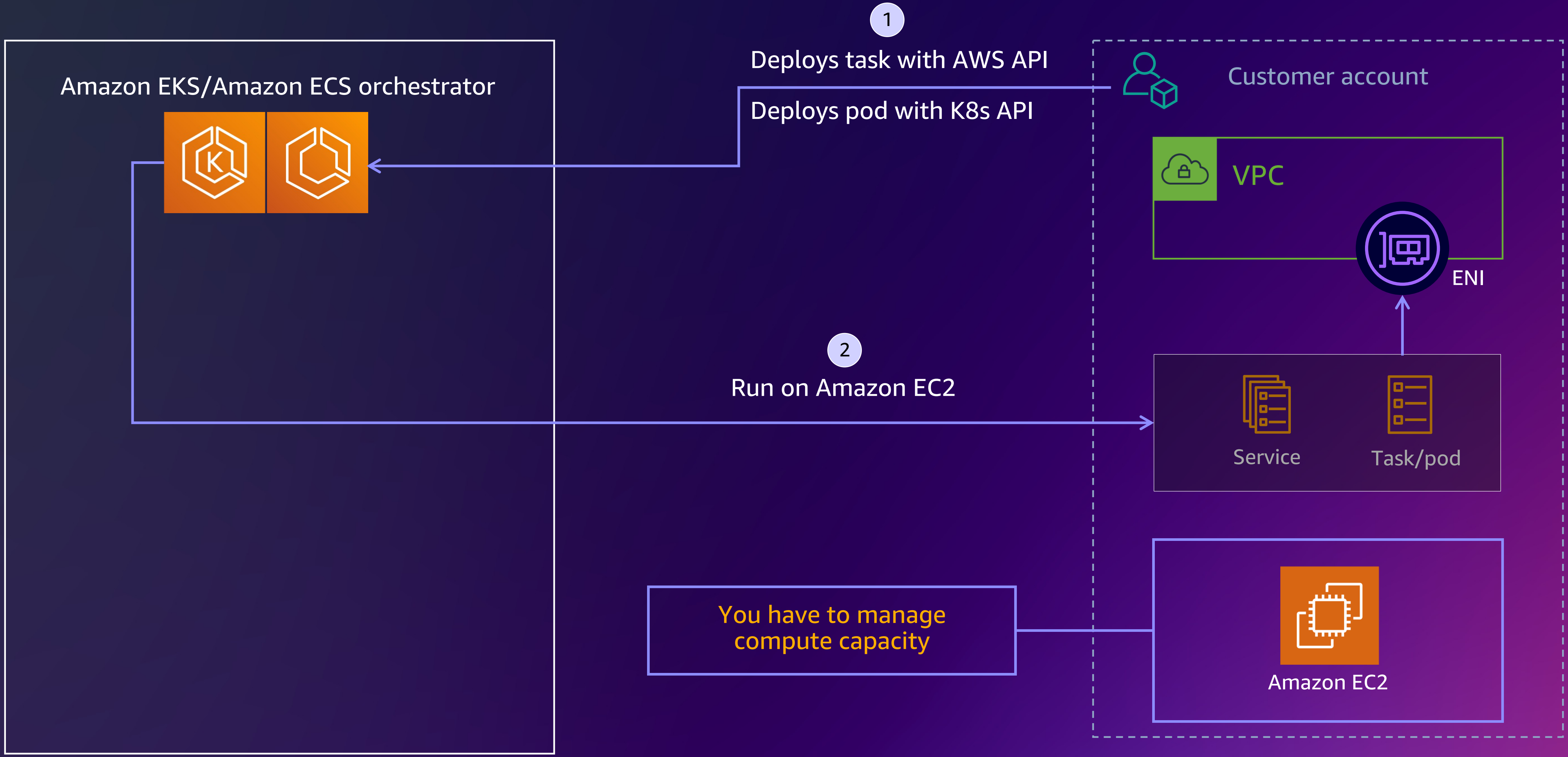


Auto Scaling groups of Amazon EC2 instances that host containerized applications

For Amazon EKS, using **Managed Node Groups** shifts some of the responsibilities to AWS

For Amazon EKS, using **Karpenter** allows you to provision EC2 instances without the use of ASGs

Amazon ECS and EKS with Amazon EC2



1

Deploys task with AWS API

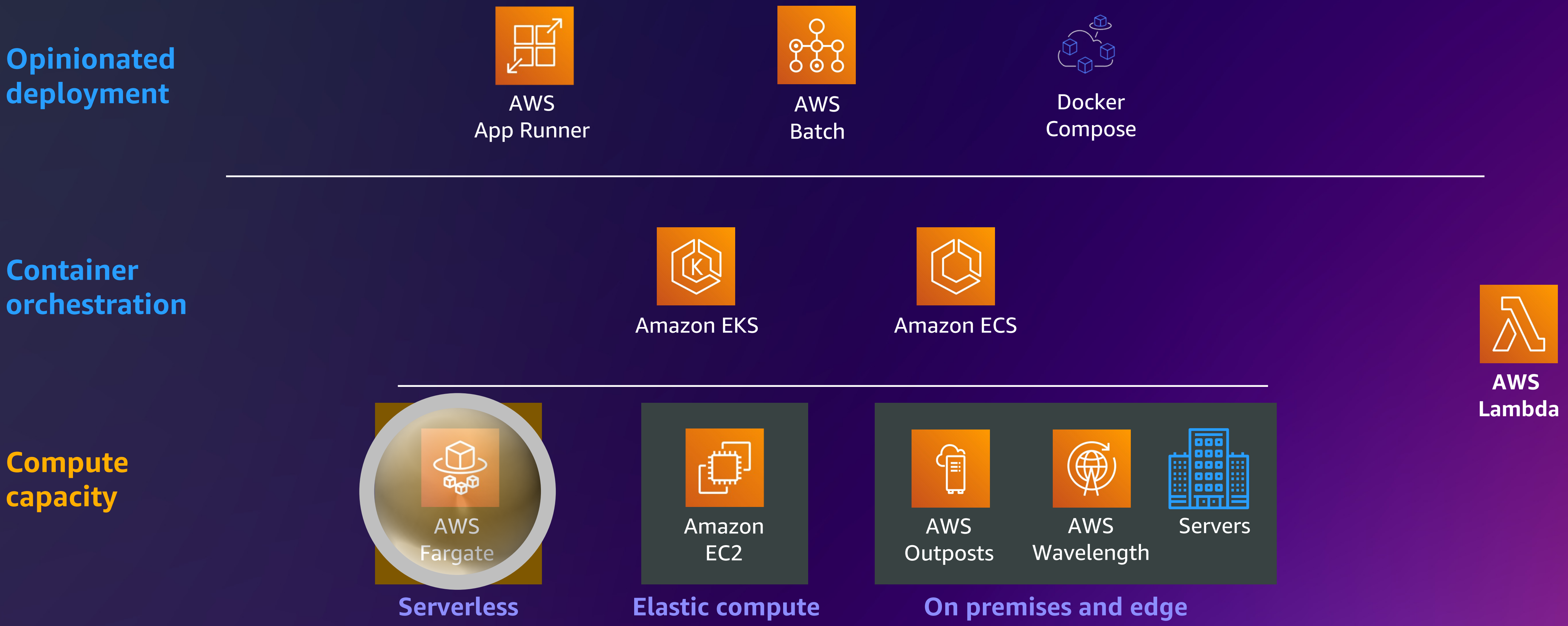
Deploys pod with K8s API

2

Run on Amazon EC2

You have to manage compute capacity

Closer look at EC2 runtime for Amazon EKS & ECS



Opinionated deployment

AWS App Runner

AWS Batch

Docker Compose

Container orchestration

Amazon EKS

Amazon ECS

AWS Lambda

Compute capacity

AWS Fargate

Serverless

Amazon EC2

Elastic compute

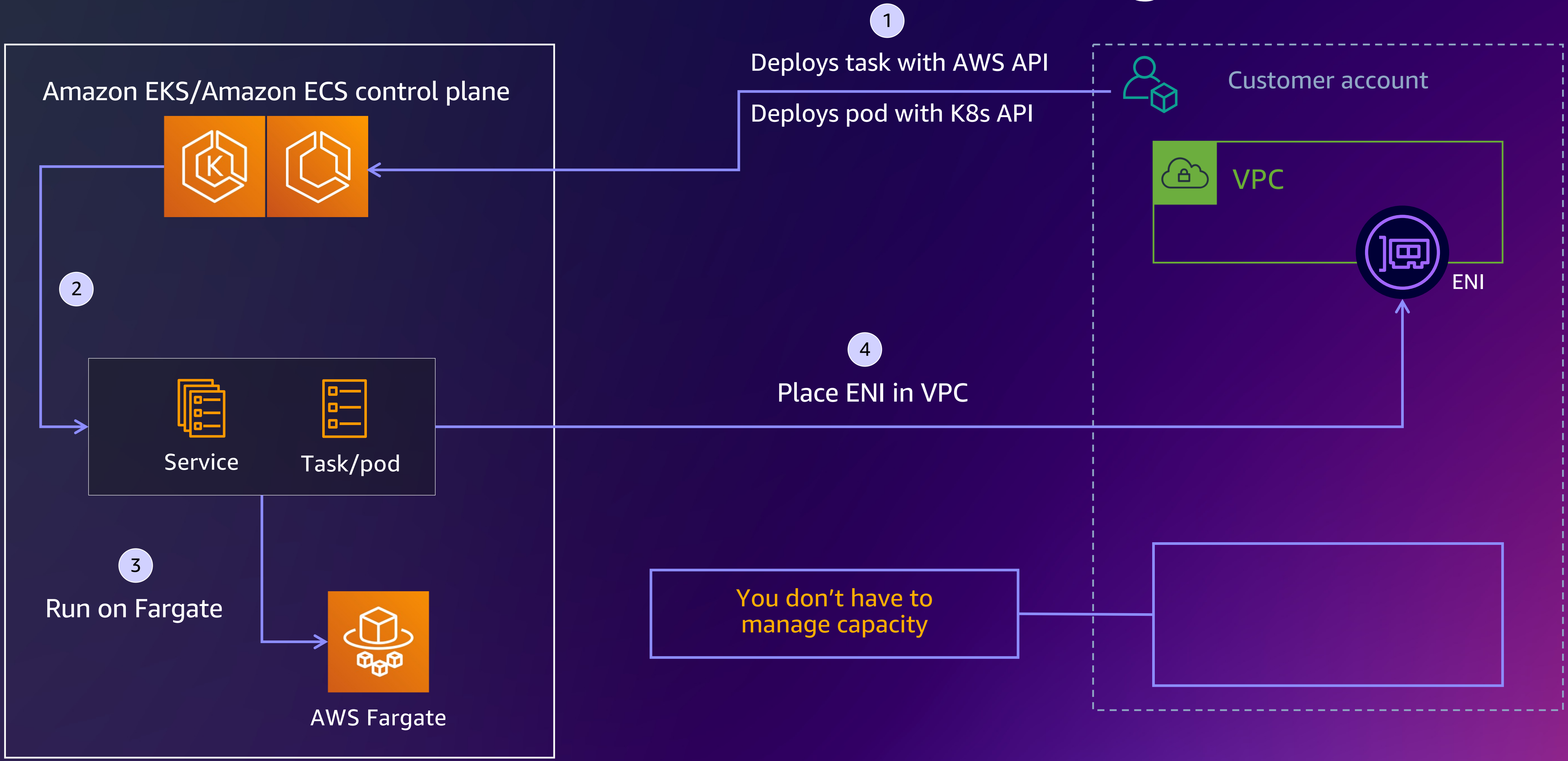
AWS Outposts

AWS Wavelength

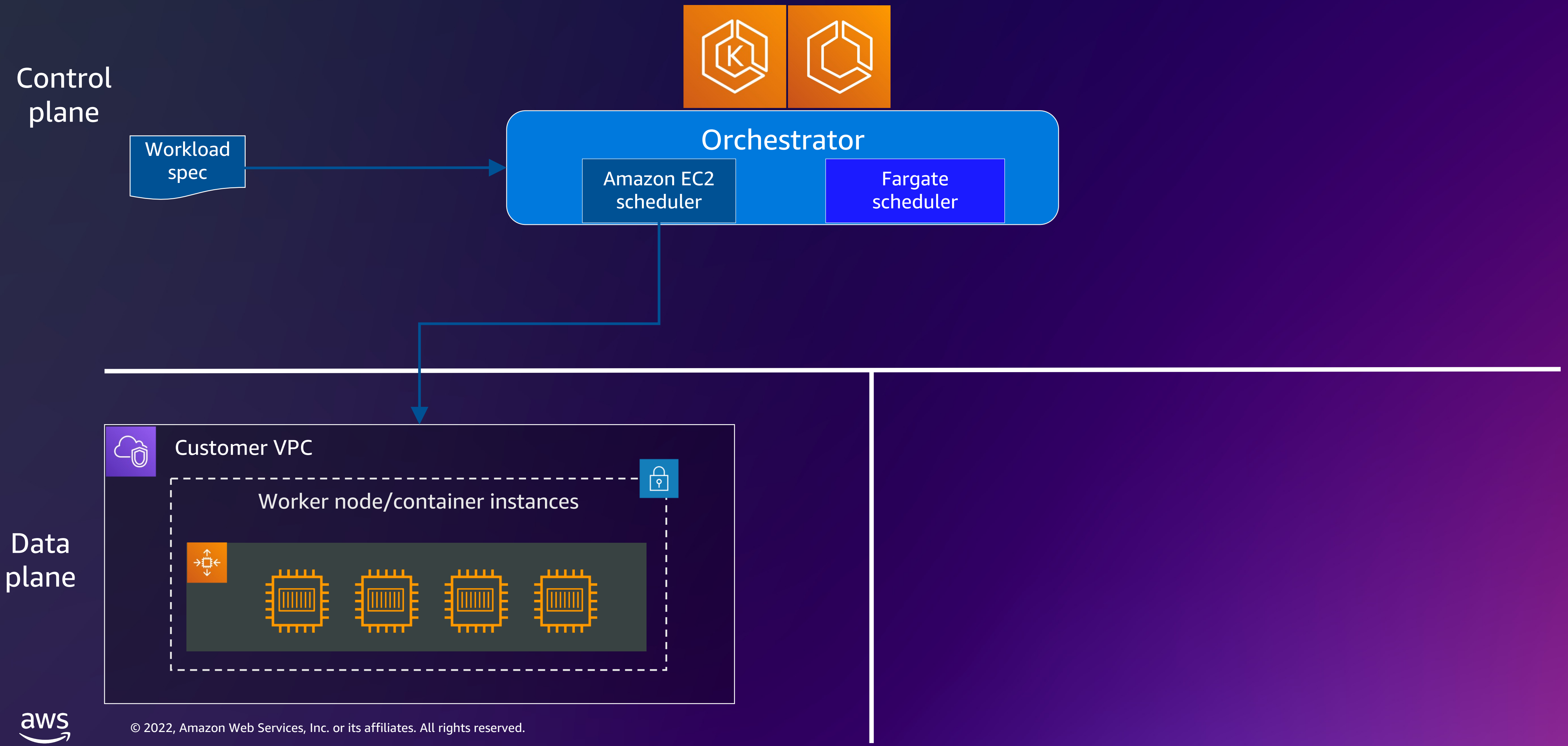
Servers

On premises and edge

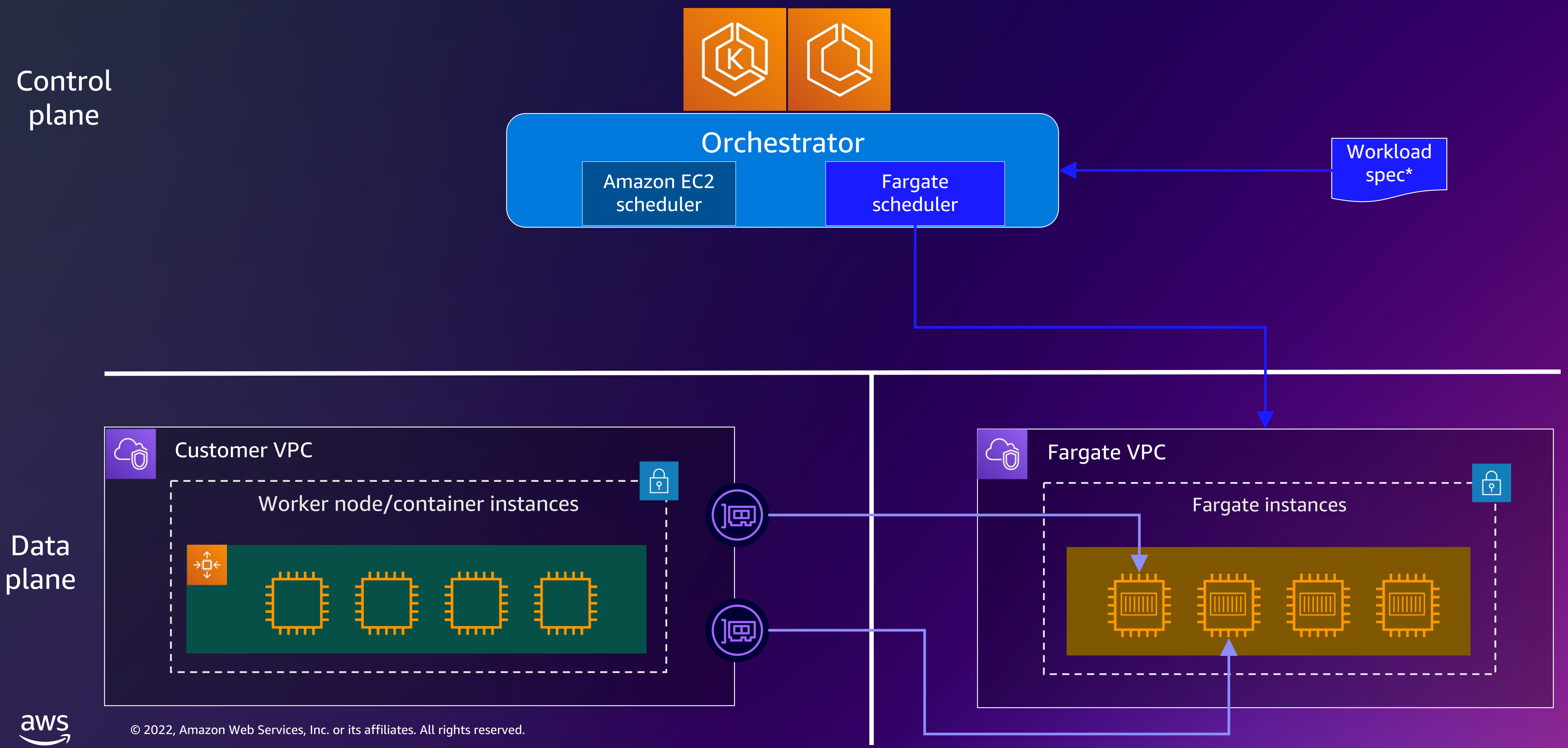
Amazon ECS and EKS with AWS Fargate



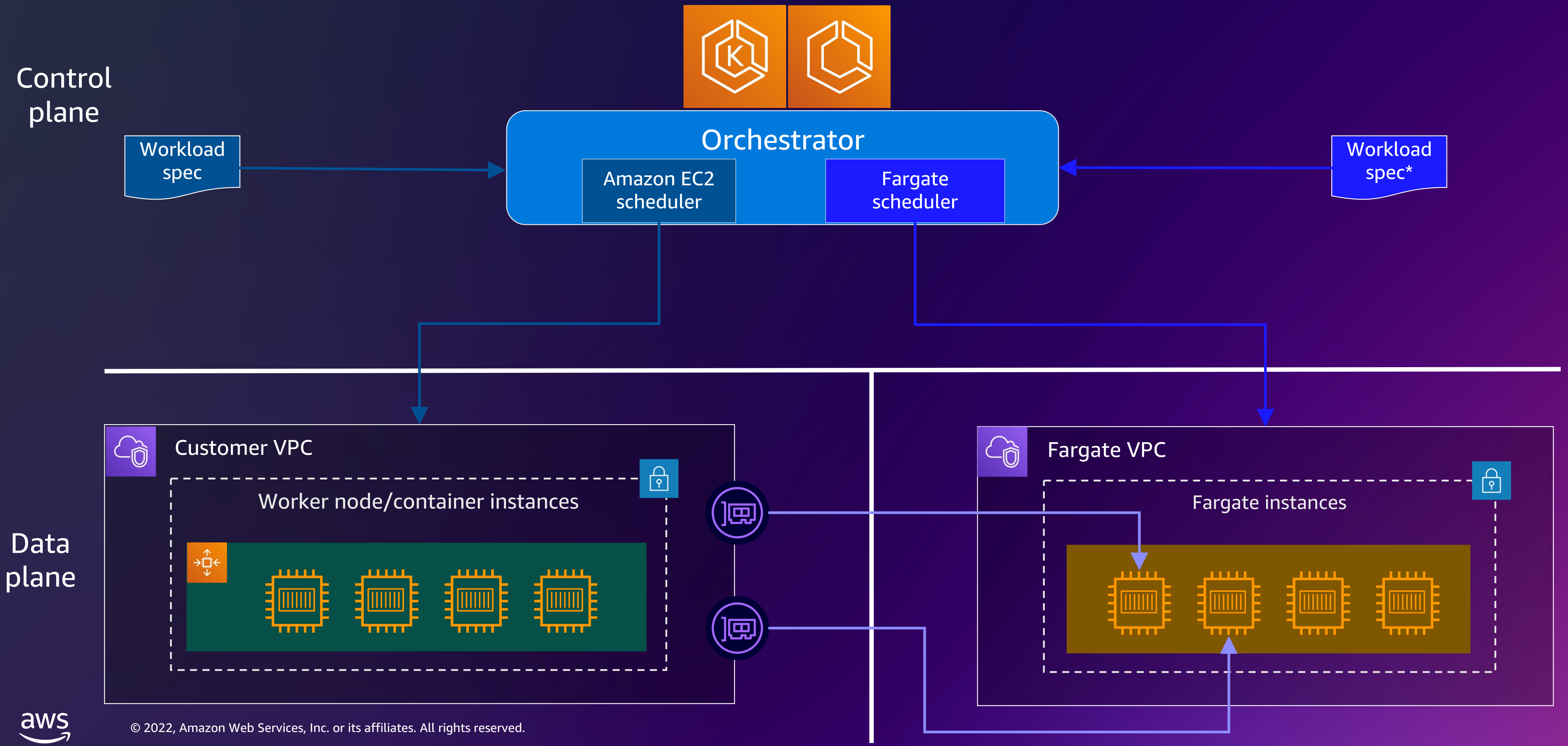
Scheduling workloads on Amazon EC2



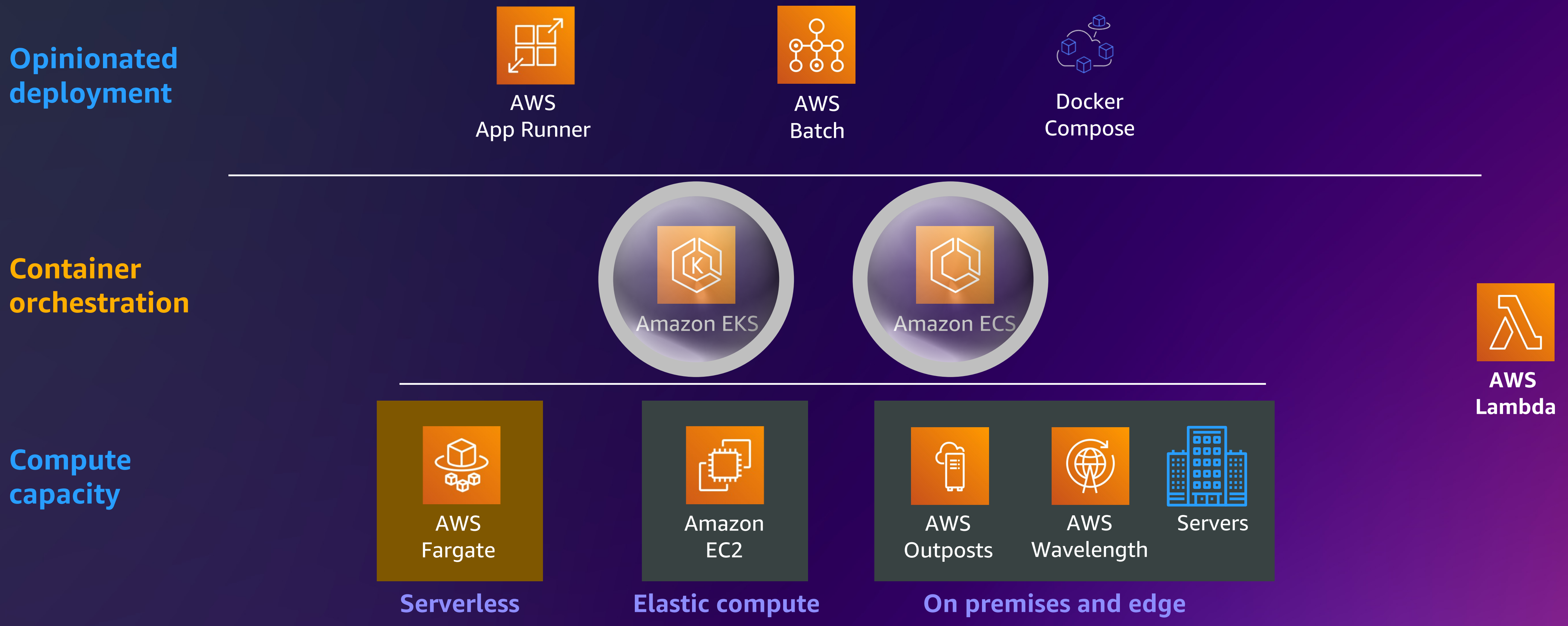
Scheduling workloads on Fargate



Scheduling workloads on Amazon EC2 & Fargate



Container orchestration – EKS or ECS?



Opinionated deployment

AWS App Runner

AWS Batch

Docker Compose

Container orchestration

Amazon EKS

Amazon ECS

AWS Lambda

Compute capacity

AWS Fargate

Serverless

Amazon EC2

Elastic compute

AWS Outposts

AWS Wavelength

Servers

On premises and edge

Amazon EKS or Amazon ECS?

ECS

Powerful simplicity

EKS

Open flexibility

Amazon ECS delivers powerful simplicity

Amazon ECS

AWS opinionated way to run containers at scale

Reduce decisions without sacrificing scale or features

Reduce time to build, deploy, and migrate applications

Run traditional applications while keeping operations overhead to a minimum

Amazon EKS delivers open flexibility

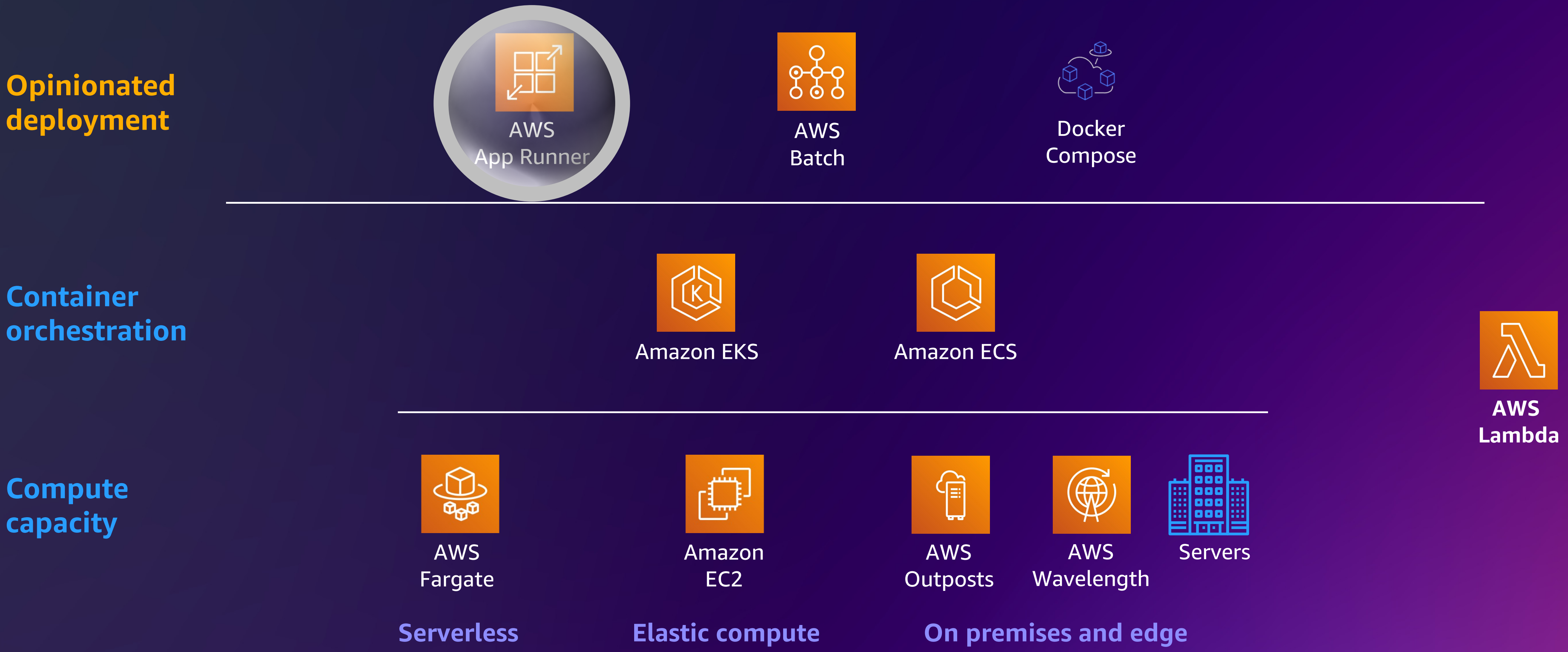
Amazon EKS

Gain agility and efficiency with AWS optimized Kubernetes and standardize operations everywhere

Secure, highly available, with observability across all Kubernetes deployments

Build with choice of solutions from the broader community around Kubernetes

App Runner



Opinionated deployment

AWS App Runner

AWS Batch

Docker Compose

Container orchestration

Amazon EKS

Amazon ECS

Compute capacity

AWS Fargate

Amazon EC2

AWS Outposts

AWS Wavelength

Servers

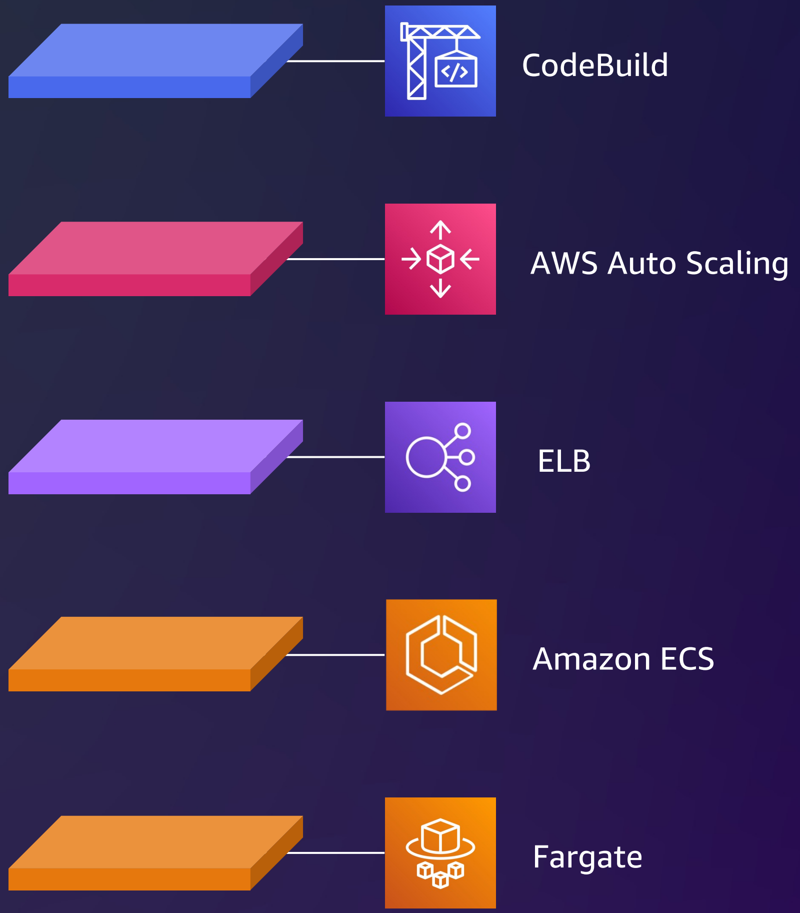
Serverless

Elastic compute

On premises and edge

AWS Lambda

Fully managed service for web applications



CodeBuild

AWS Auto Scaling

ELB

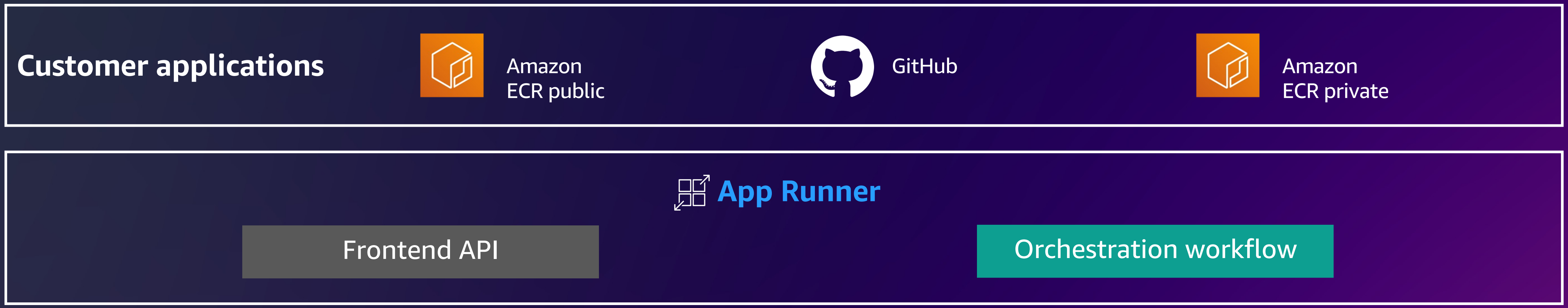
Amazon ECS

Fargate

Web application

Backend server

App Runner building blocks



With App Runner, customers don’t need to manage . . .

- Load balancing and health check (Envoy)
- Safe deployment (blue-green)
- Service and instance metrics (CloudWatch metrics)
- Build (CB)
- Auto scaling (request-based)
- Host patching (Fargate)
- Custom domain (NLB)
- Runtime patching (Amazon ECR)
- Encryption (AWS KMS)
- Hosting (Fargate, Amazon VPC)
- CICD (webhook, CW events)
- Deployment and application logs (CloudWatch logs)

App Runner application properties

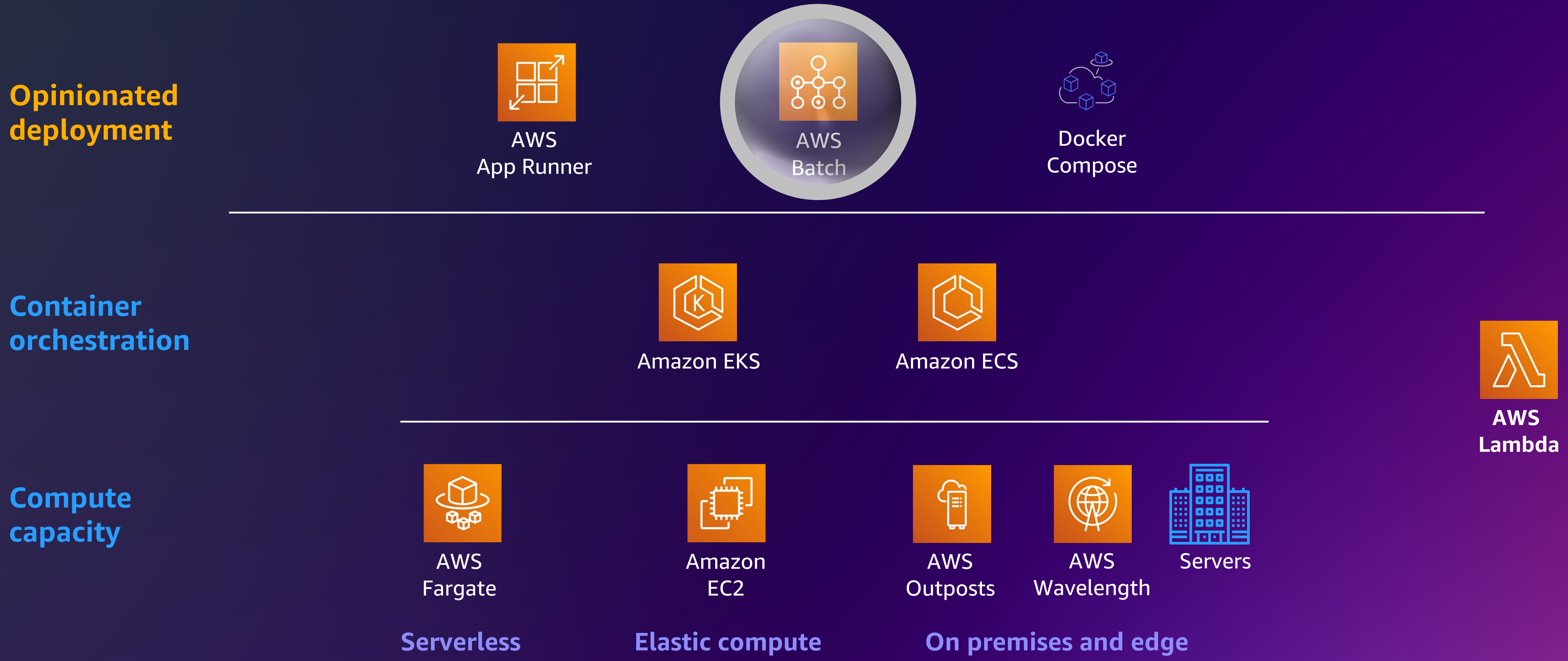
App Runner

For Web applications and API servers

Multi-concurrent, long-running application

Stateless within application with external state so that requests are processed independently

AWS Batch



Opinionated deployment

AWS App Runner

AWS Batch

Docker Compose

Container orchestration

Amazon EKS

Amazon ECS

Compute capacity

AWS Fargate

Amazon EC2

AWS Outposts

AWS Wavelength

Servers

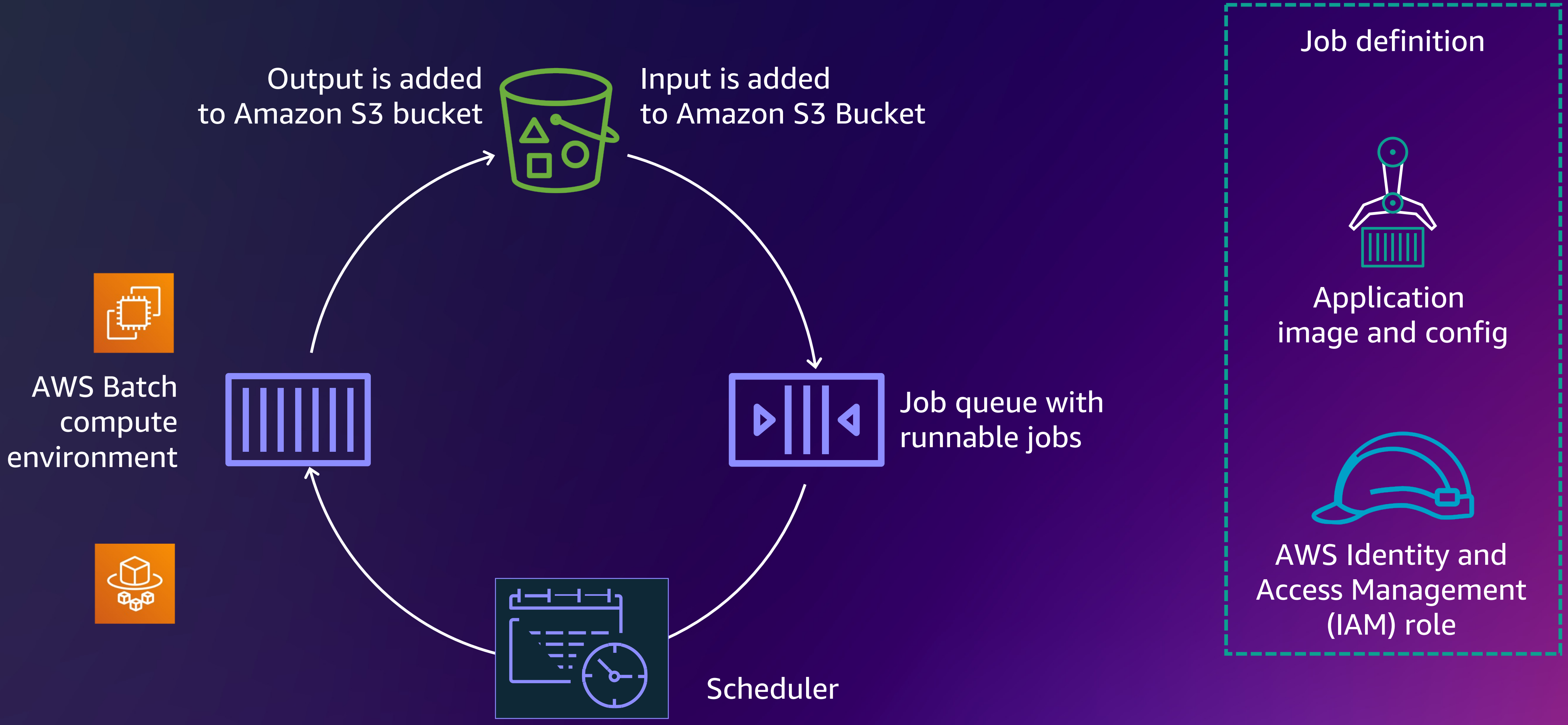
Serverless

Elastic compute

On premises and edge

AWS Lambda

Typical AWS Batch job architecture



Output is added to Amazon S3 bucket

Input is added to Amazon S3 Bucket

Job queue with runnable jobs

Scheduler

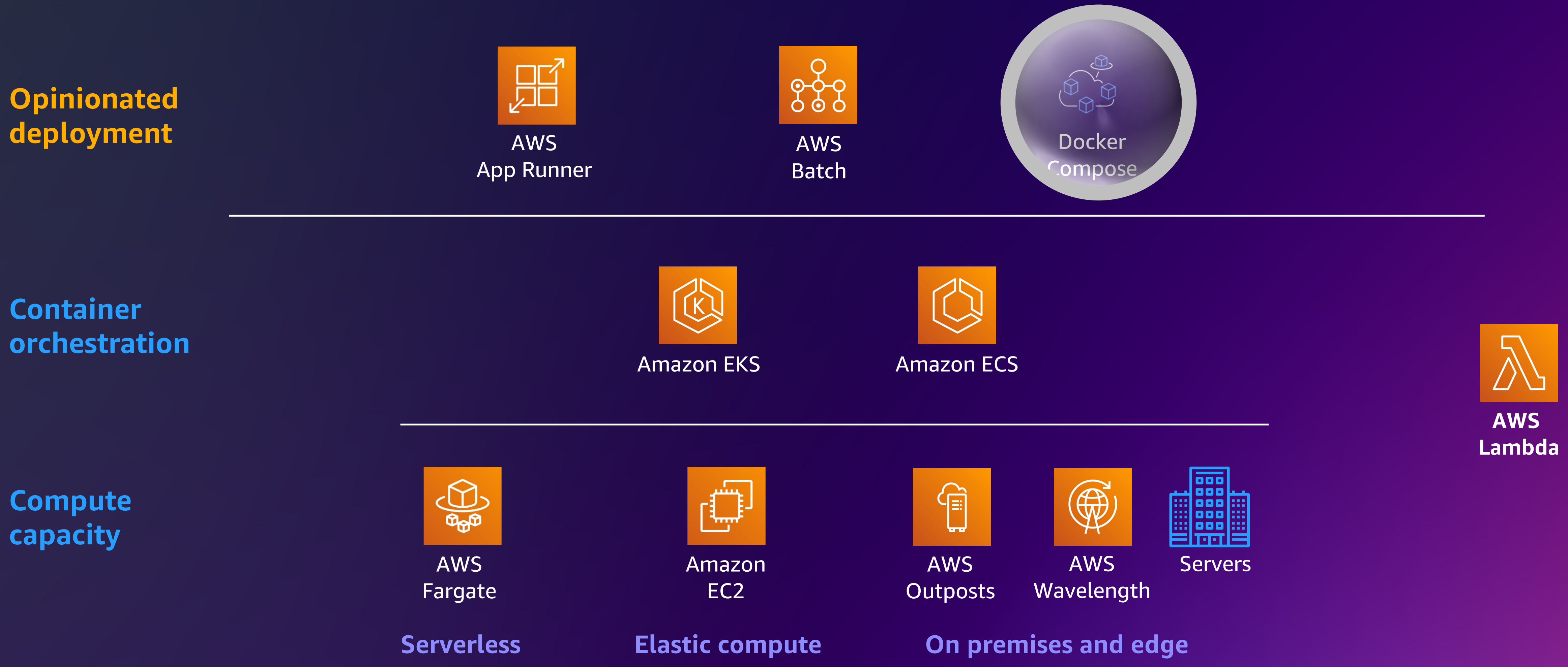
AWS Batch compute environment

Job definition

Application image and config

AWS Identity and Access Management (IAM) role

Docker Compose



Opinionated deployment

AWS App Runner

AWS Batch

Docker Compose

Container orchestration

Amazon EKS

Amazon ECS

AWS Lambda

Compute capacity

AWS Fargate

Amazon EC2

AWS Outposts

AWS Wavelength

Servers

Serverless

Elastic compute

On premises and edge

Docker Compose for Amazon ECS

Amazon ECS + Docker compose

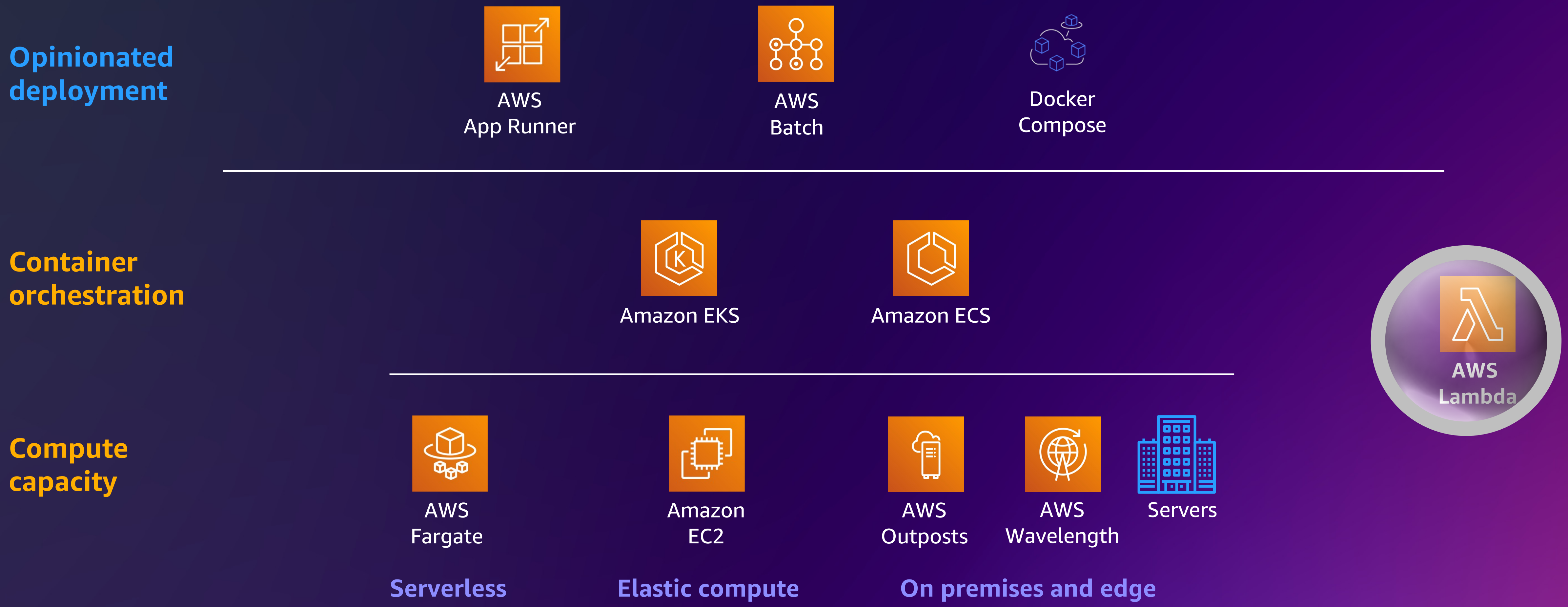
Use Docker Compose to deploy applications to Amazon ECS Fargate

Quickly switch between local development and ECS environments

Opinionated approach to deploying applications to ECS

As easy as docker compose up and docker compose down

Running containers with Lambda



Opinionated deployment

AWS App Runner

AWS Batch

Docker Compose

Container orchestration

Amazon EKS

Amazon ECS

Compute capacity

AWS Fargate

Amazon EC2

AWS Outposts

AWS Wavelength

Servers

Serverless

Elastic compute

On premises and edge

AWS Lambda

Lambda container image support

- Easier **dependency management** and application building with container images

- Use a **consistent set of tools** for containers and Lambda-based applications

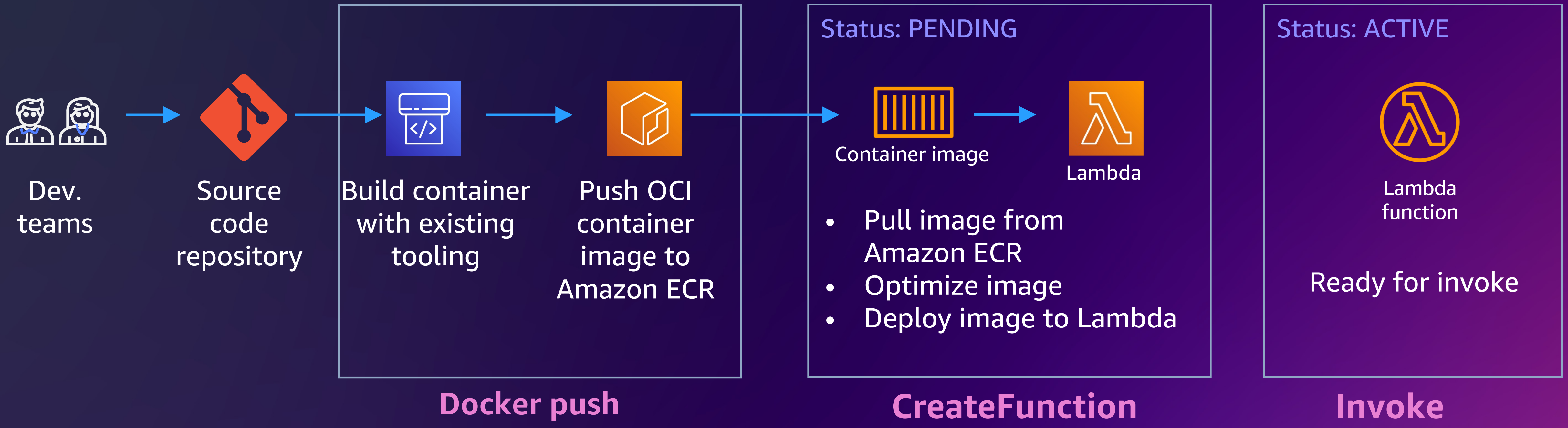
- Deploy large applications with AWS vended or third-party **images of up to 10 GB**

- Benefit from sub-second automatic scaling, high availability, **140 native service integrations**, and pay-for-use billing model

Lambda container image

Simple developer experience

Package and deploy functions as container images



Dev. teams

Source code repository

Build container with existing tooling

Push OCI container image to Amazon ECR

Docker push

Status: PENDING

Container image

Lambda

- Pull image from Amazon ECR
- Optimize image
- Deploy image to Lambda

CreateFunction

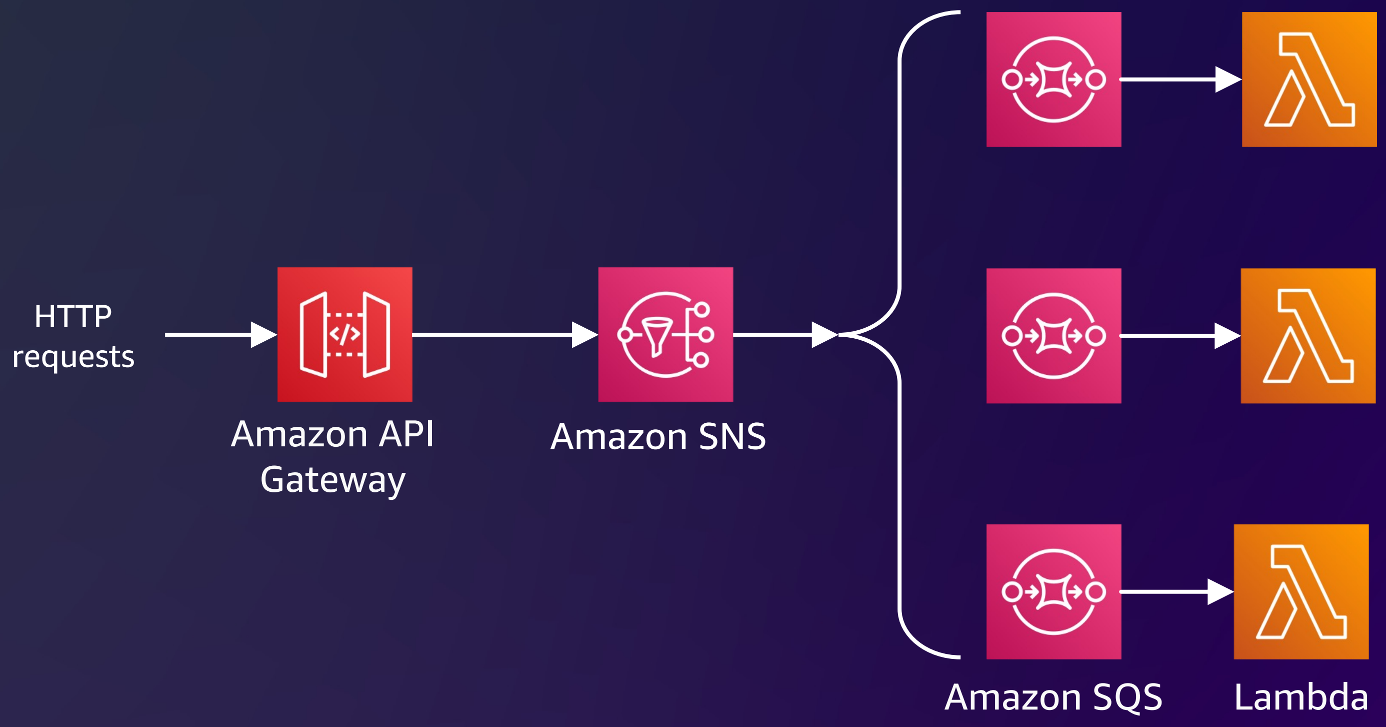
Status: ACTIVE

Lambda function

Ready for invoke

Invoke

Fan-out pattern for message processing



API Gateway can integrate with AWS services directly

Publish notifications directly to Amazon SNS

Use Amazon SNS subscription policies to separate messages into different Amazon SQS queues

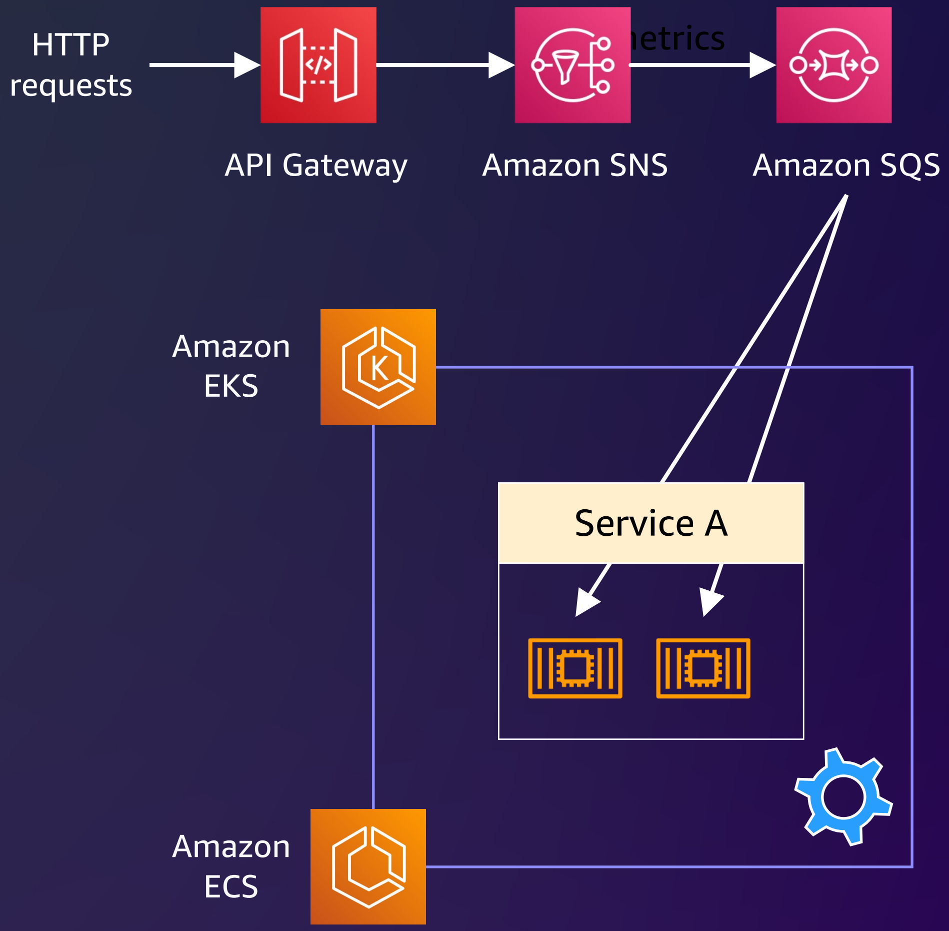
Messages from each queue trigger a Lambda function

Using provisioned concurrency for scale out

No resources are consumed when queues are empty

Event-driven application

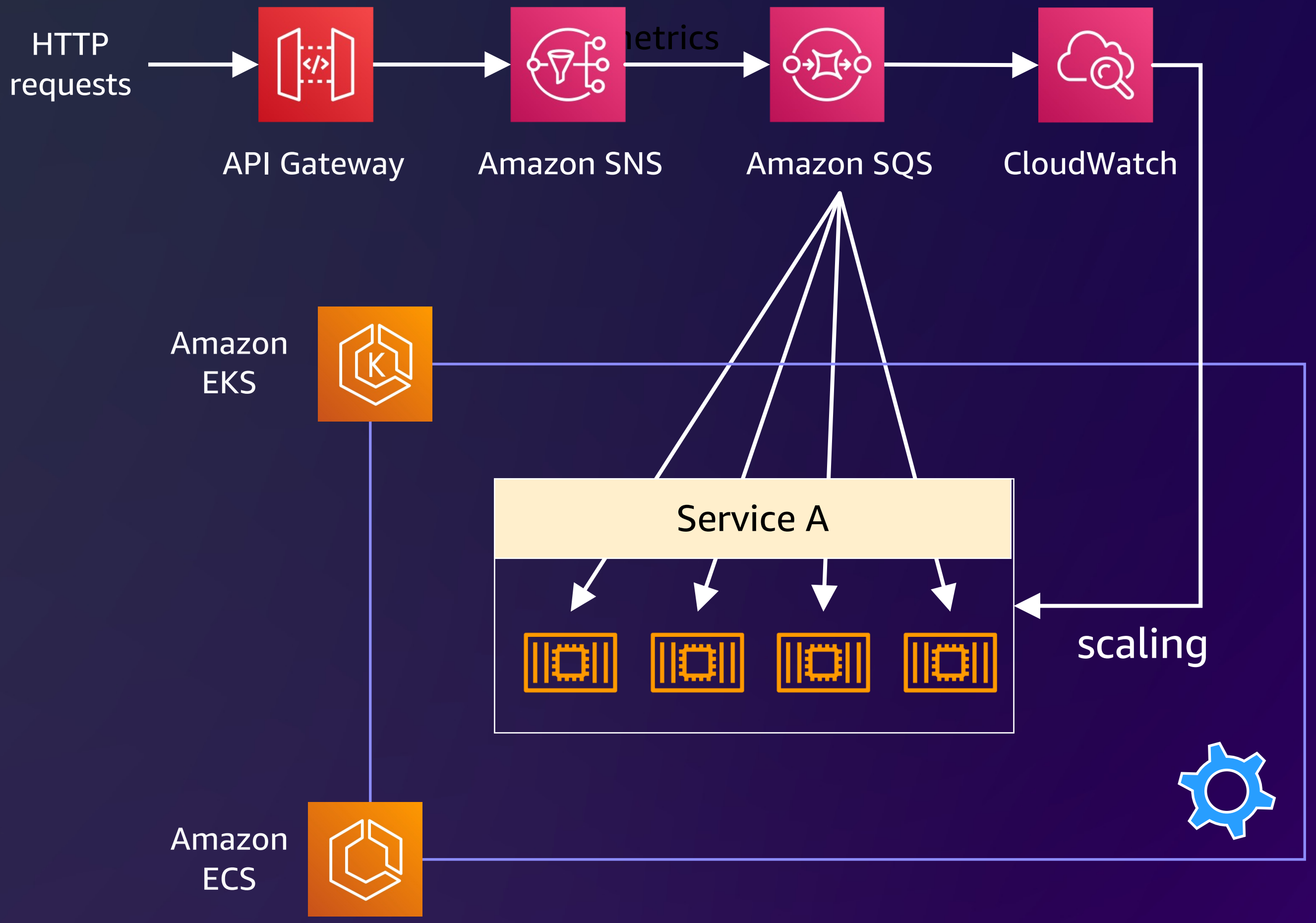
Fan-out pattern for message processing



Long-running Amazon ECS/Amazon EKS workloads pull and process messages from Amazon SQS queue

Traditional long-running application

Fan-out pattern for message processing



Long-running Amazon ECS/Amazon EKS workloads pull and process messages from Amazon SQS queue

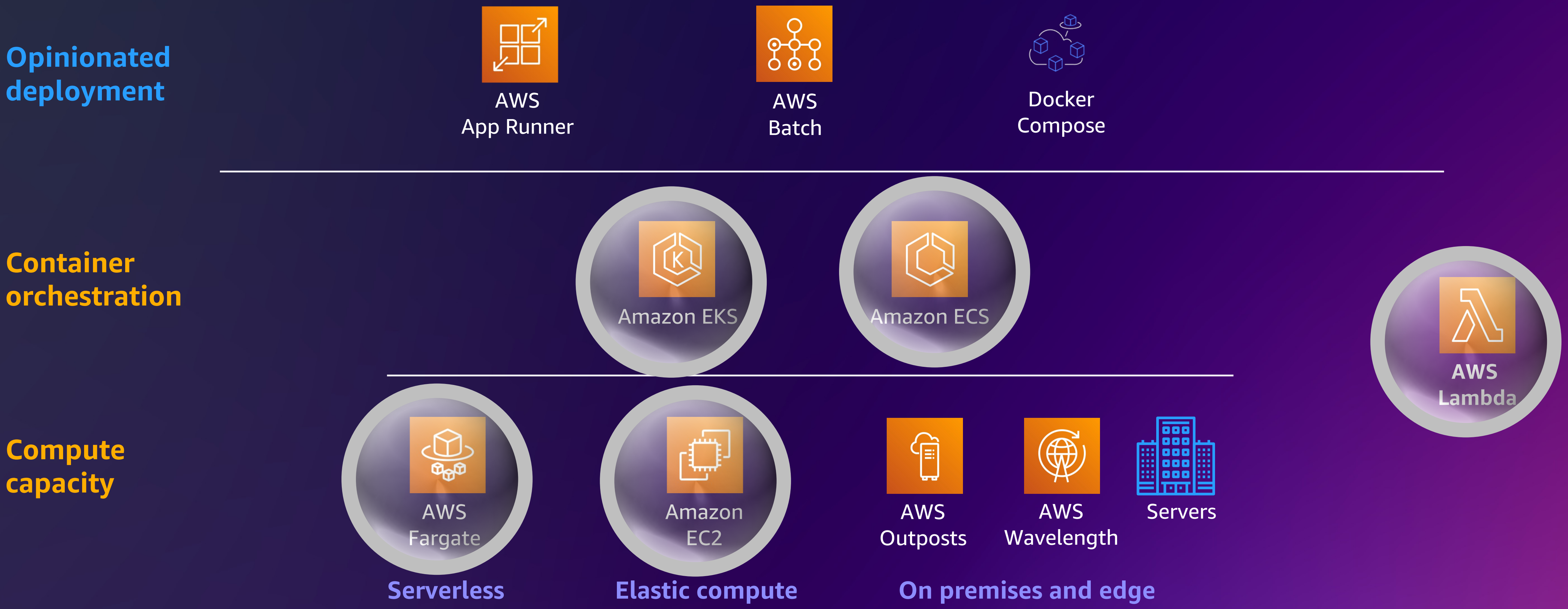
Use CloudWatch metrics for Amazon SQS to scale the workloads horizontally

Even when queues are empty, workloads are running, polling for messages

Compute capacity is always in use, albeit scaled down

Traditional long-running application

Summary – When to choose what?



Opinionated deployment

AWS App Runner

AWS Batch

Docker Compose

Container orchestration

Amazon EKS

Amazon ECS

AWS Lambda

Compute capacity

AWS Fargate

Amazon EC2

AWS Outposts

AWS Wavelength

Servers

Serverless

Elastic compute

On premises and edge

Go serverless when you can

Lambda

Fargate

Serverless-first strategy for building net new and refactoring existing applications

Go serverless when you can

Lambda

Serverless event-driven code execution

- Short-lived
- All language runtimes
- Data source integrations

Fargate

Serverless compute engine for containers

- Long-running
- Bring existing code
- Fully-managed orchestration

Customers choose Lambda when . . .

- developing new applications
- developing event-driven applications
- using managed integrations with other AWS services
- refactoring an existing application

When is Lambda not suitable?

Lambda technical constraints

- single event execution that runs for 15+ minutes
- workloads that benefits from GPUs
- workloads that needs more than 10GB of memory

Customers choose ECS on Fargate for . . .

- migrating existing applications
- deploying traditional long-running applications
- using AWS opinionated abstractions
  over orchestration

- container orchestration with less operational overhead

Customers choose ECS on EC2 for . . .

- migrating existing applications
- deploying traditional long-running applications
- using AWS opinionated abstractions
  over orchestration

- complete control over compute infrastructure

Customers choose Amazon EKS when . . .

- deploying traditional long-running applications
- need flexibility and extensibility with container orchestration
- standardize on Kubernetes APIs instead of AWS APIs
- leverage open-source tools in Kubernetes ecosystem

Criteria for choosing the right container service

AWS services that run container images can be scored against these dimensions

- Workload types
- Compute capacity
- Scalability
- Extensibility
- Operational overhead
- Ease of consumption
- Hybrid support
- Ecosystem

Thank you!

René Penkert

✉ rppenker@amazon.ch
in linkedin.com/in/renpen

Sascha Möllering

✉ smoell@amazon.de

Please complete the session survey