AWS App Runner Deep Dive

Archana Srikanta
Principal Software Engineer
Amazon Web Services

Agenda

What is App Runner

Applications that are best suited for App Runner

Why we built App Runner

Understanding the evolution of compute services on AWS

How to use App Runner

Walk through of the App Runner experience with a demo

Features

Autoscaling, Deployments, Observability with demos

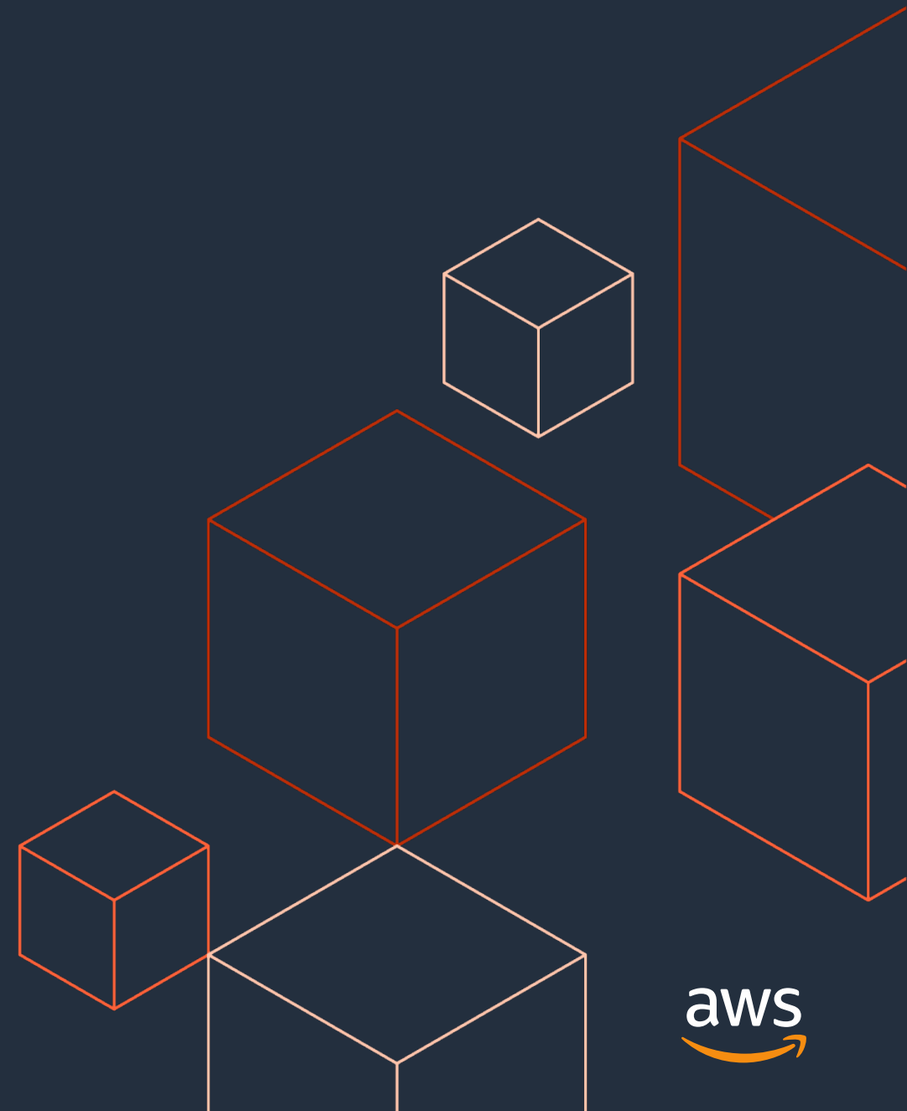
Pricing

Pricing dimensions with an example

aws

What

is App Runner



What is App Runner

AWS App Runner provides a highly abstracted and simple managed experience for running web applications and API hosting services

aws

Application properties

Web Applications & API servers

Applications that serve HTTP based requests

Multi-concurrent

The application is long-running.

A single instance of the application may serve many requests during its lifetime.

Multiple requests maybe handled simultaneously.

Stateless

Requests are processed independently and do not depend on local state.

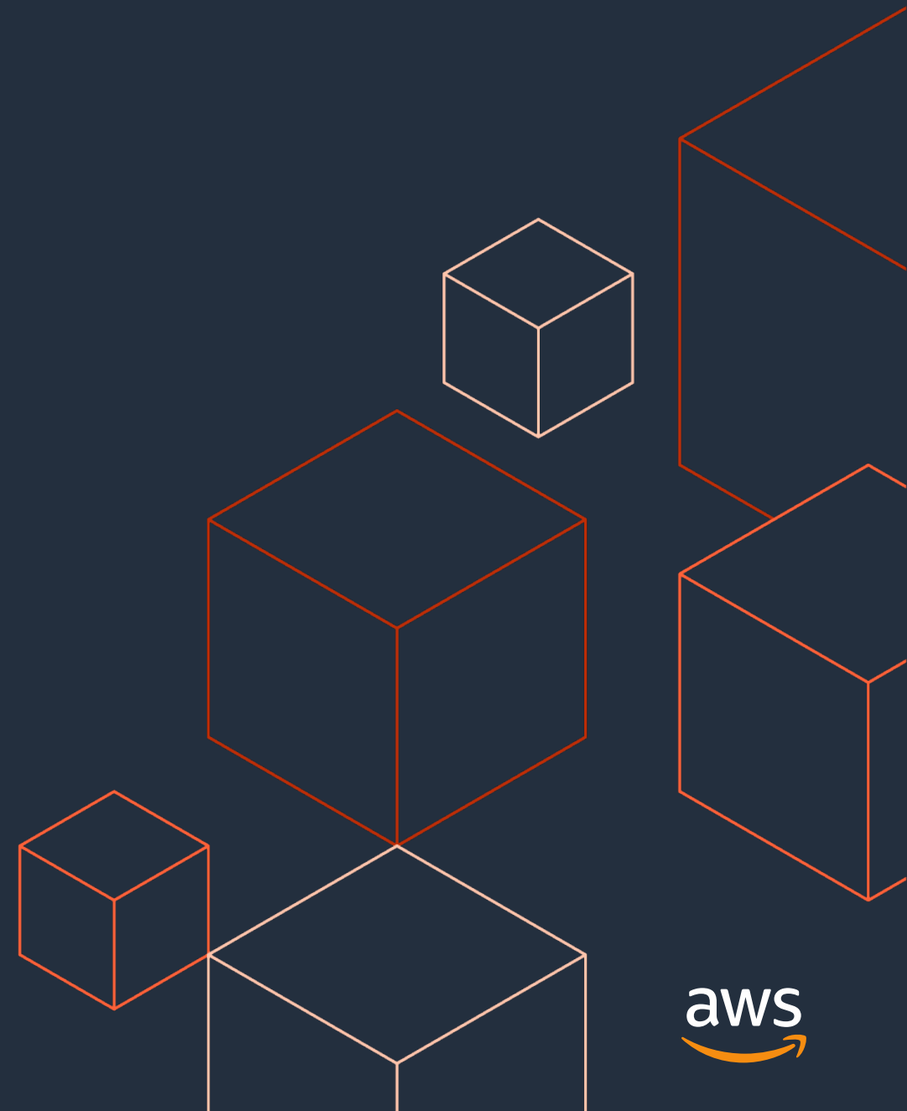
State maybe stored external to the application instance (eg: a DynamoDB table)

No background processing

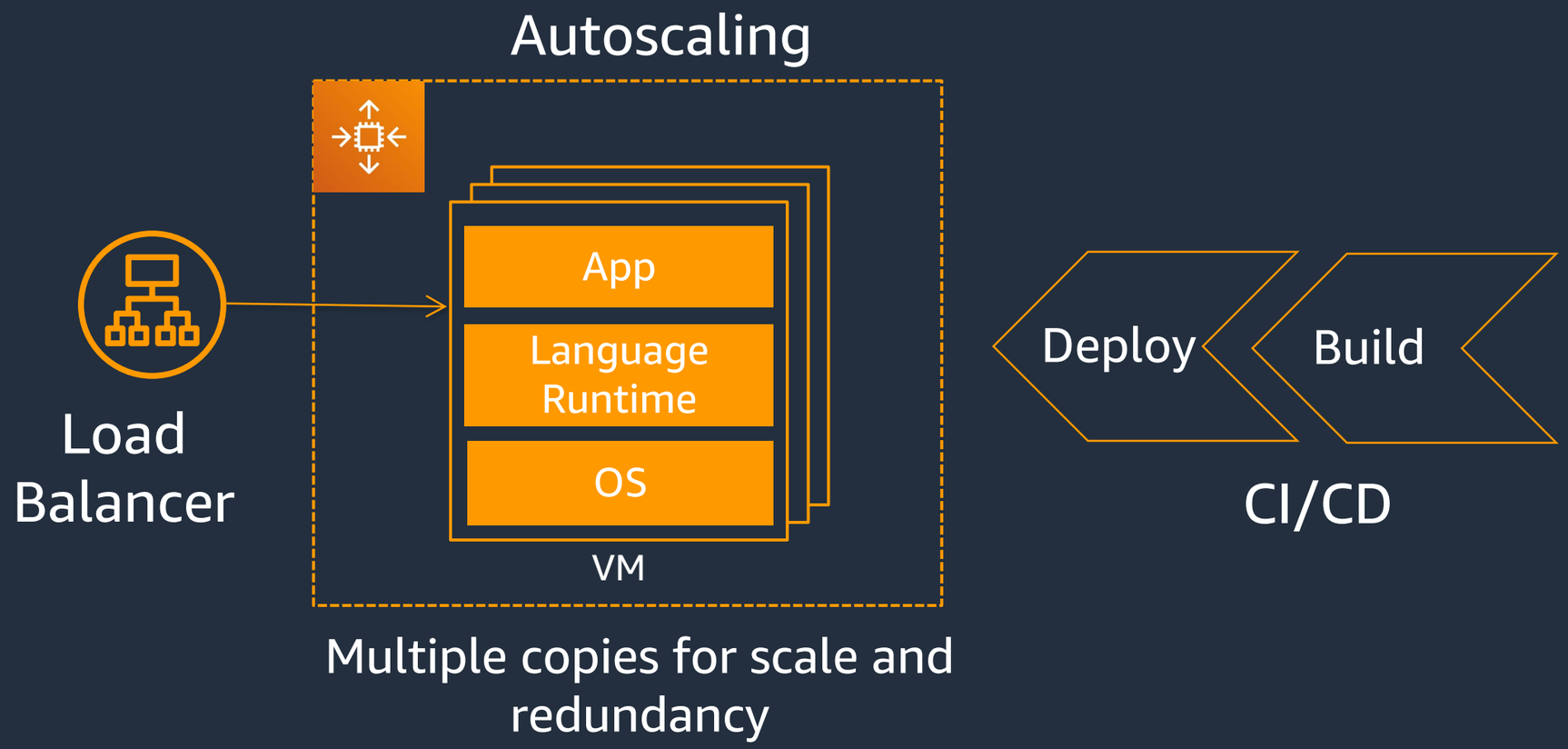
Any processing outside the context of a request must be limited

aws

Why
we built App Runner



Typical application architecture

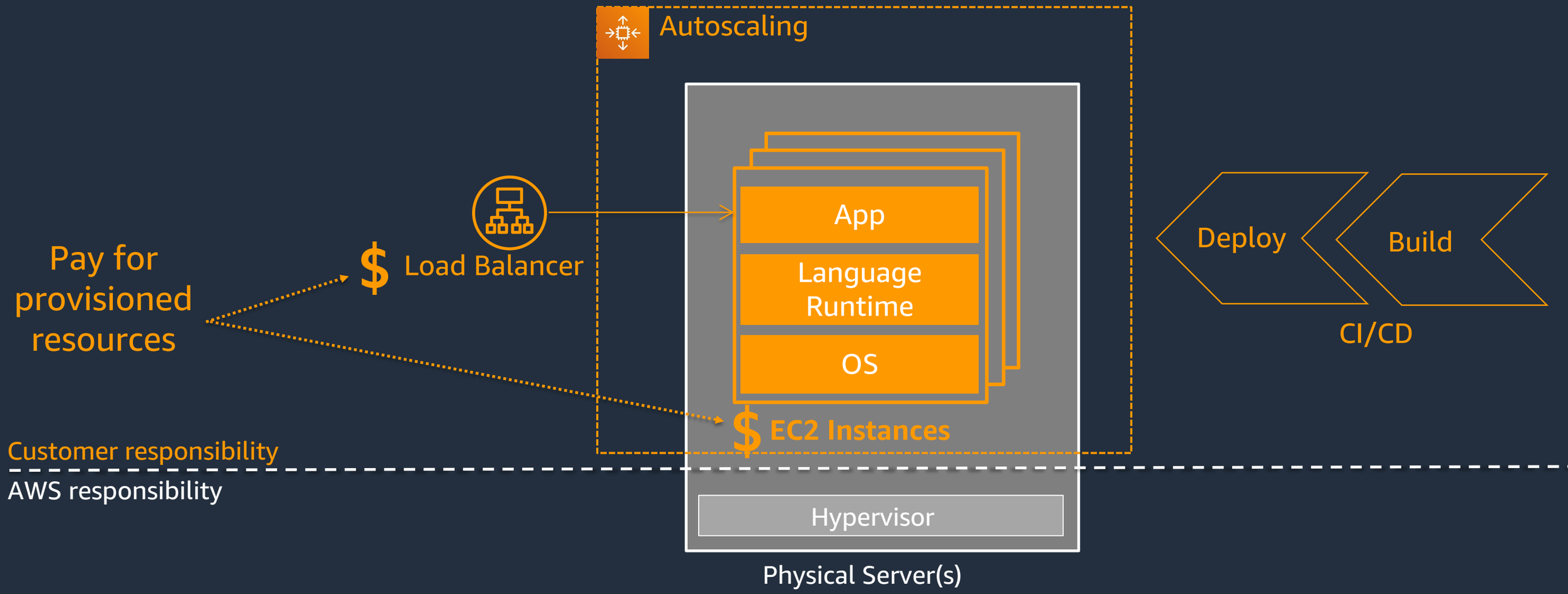


Shared responsibility on AWS

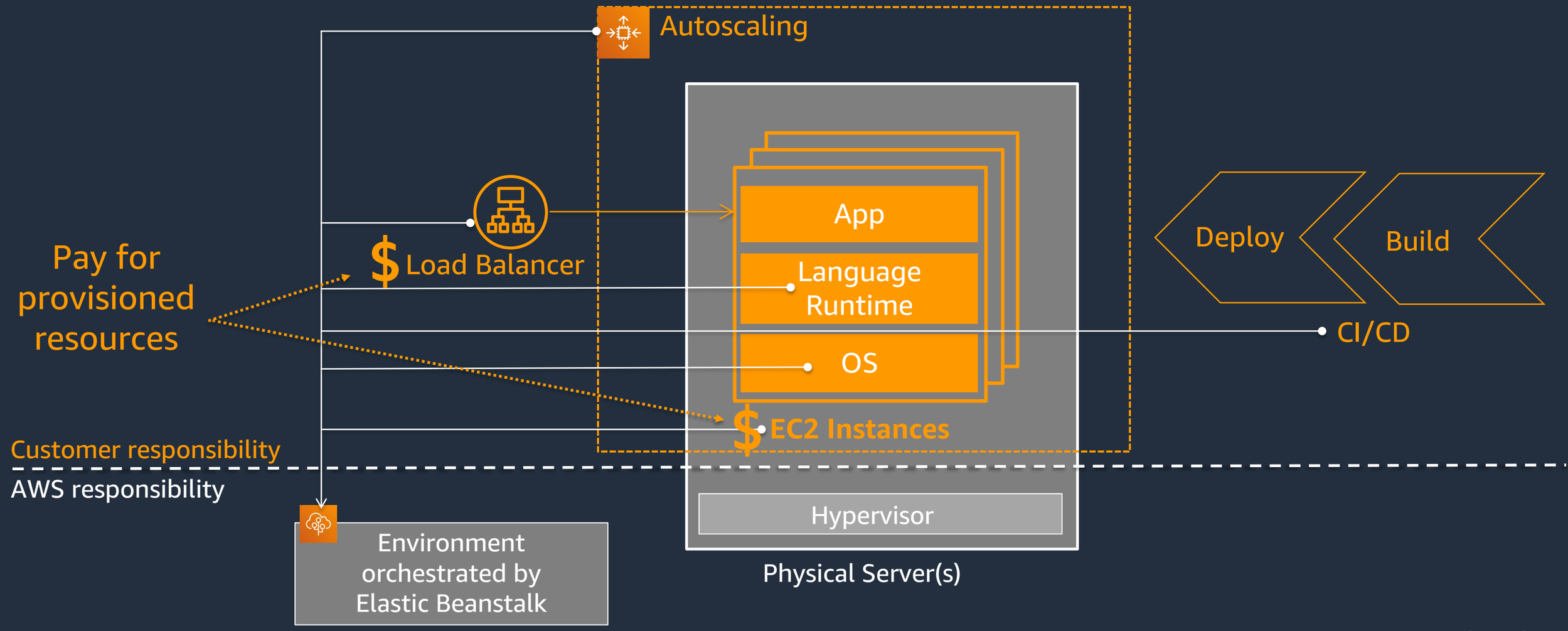
Application security & operational stability is a shared responsibility between AWS and the customer

aws

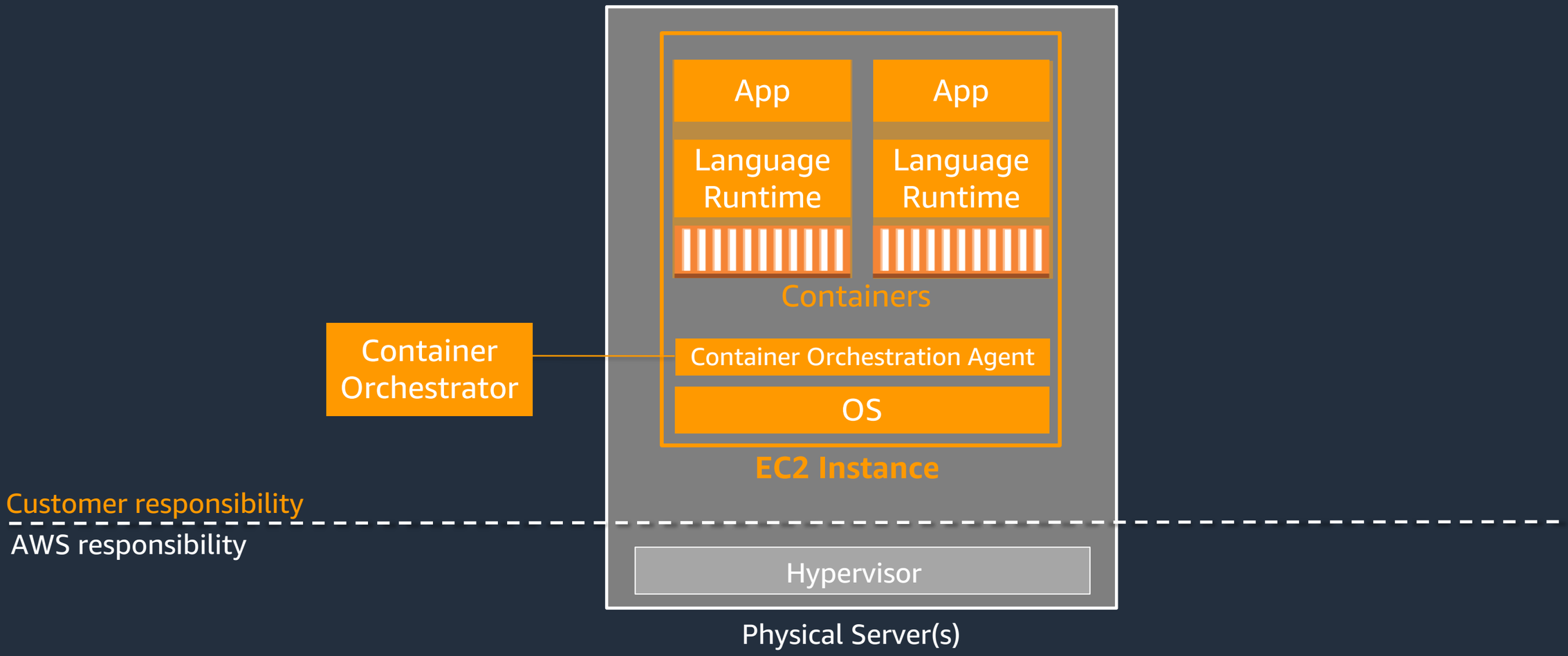
Shared responsibility on Amazon EC2 (released 2006)



Shared responsibility on AWS Elastic Beanstalk (released 2011)

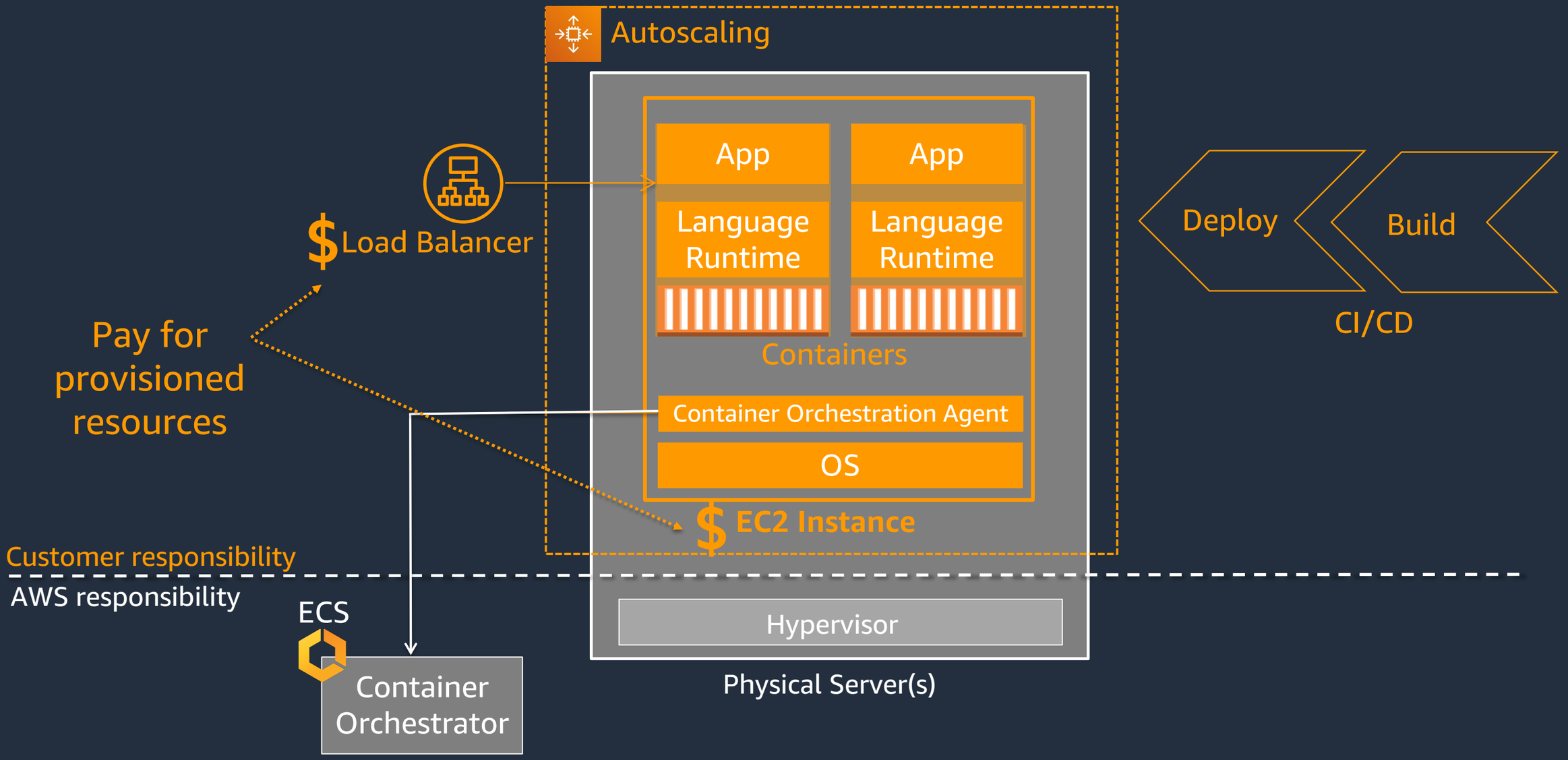


Containers became popular

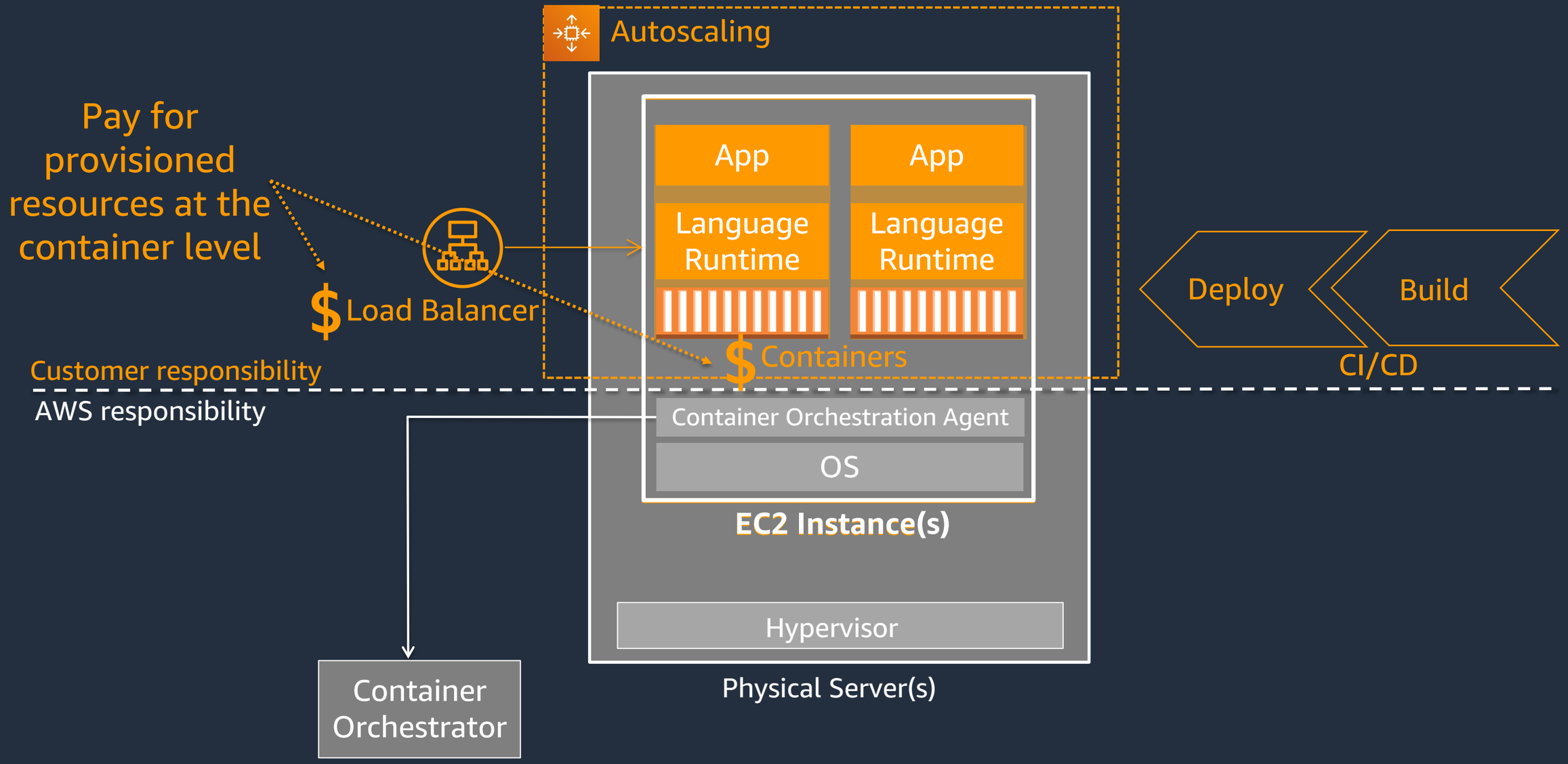


aws

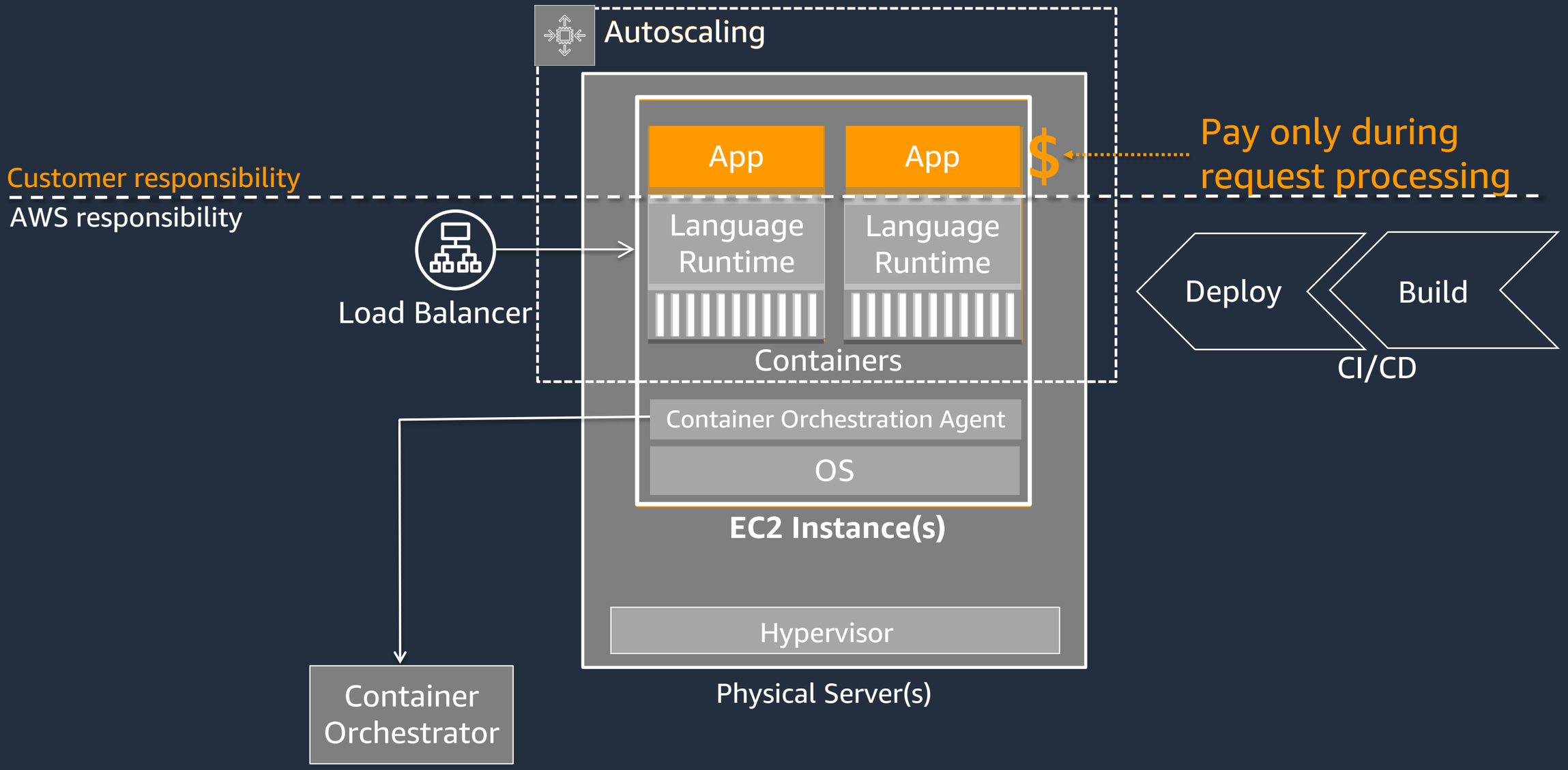
Shared responsibility on Amazon ECS (released 2015)



Shared responsibility on AWS Fargate (released 2017)

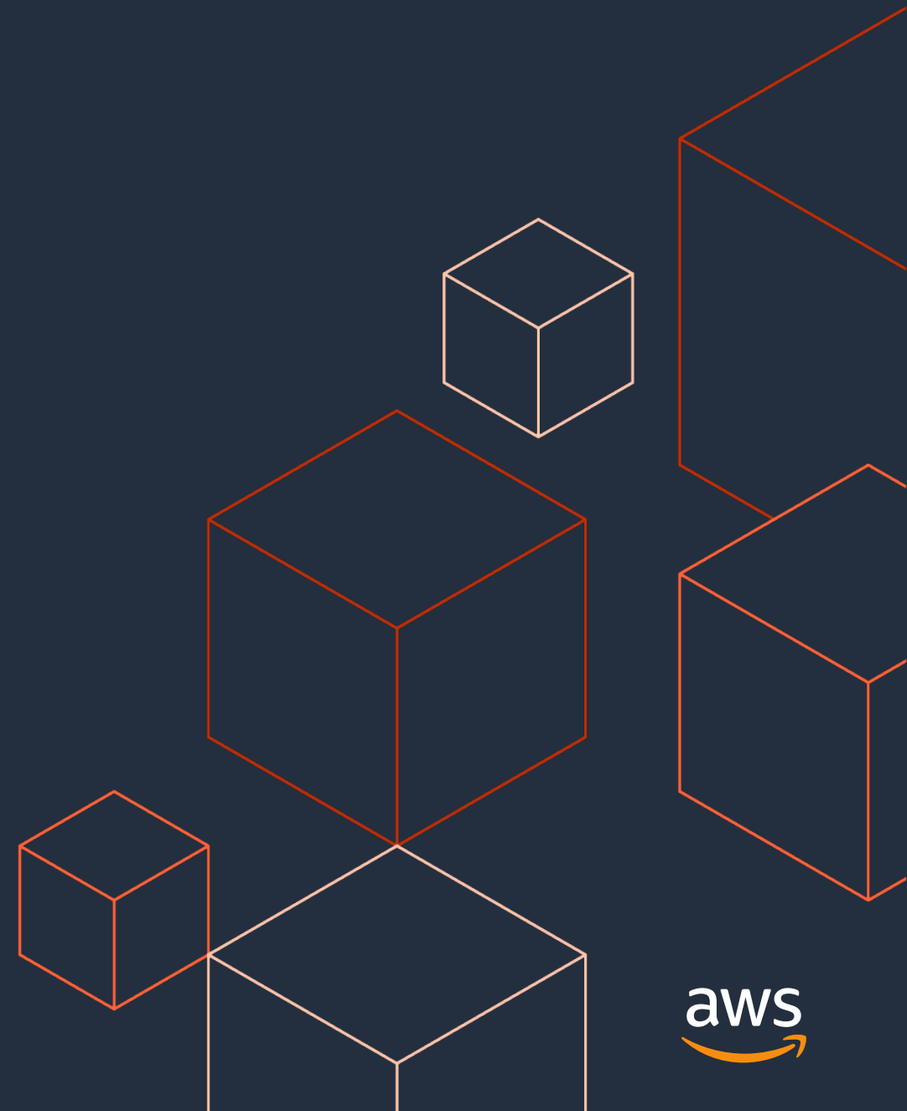


Shared responsibility on AWS App Runner (released 2021)

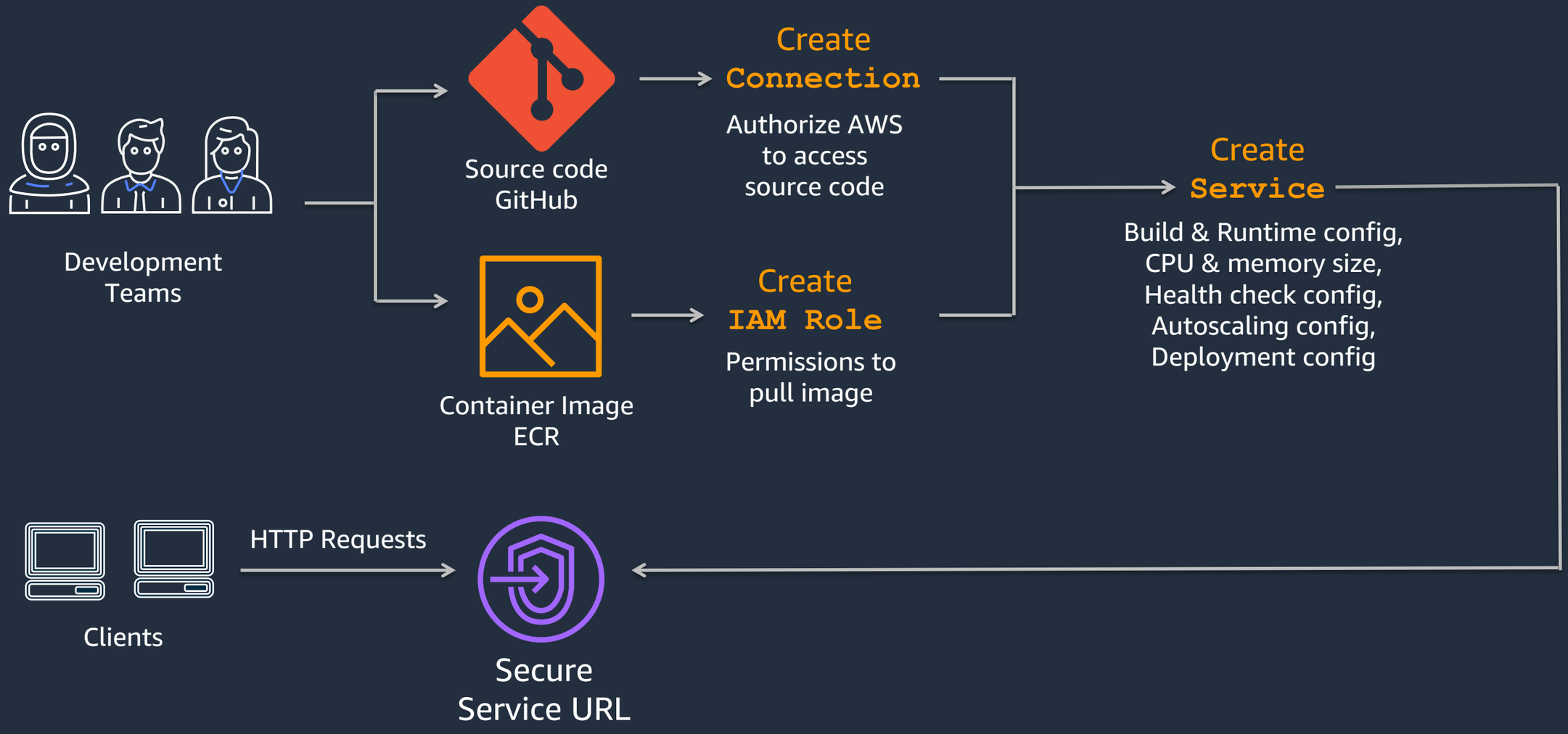


How

to use App Runner

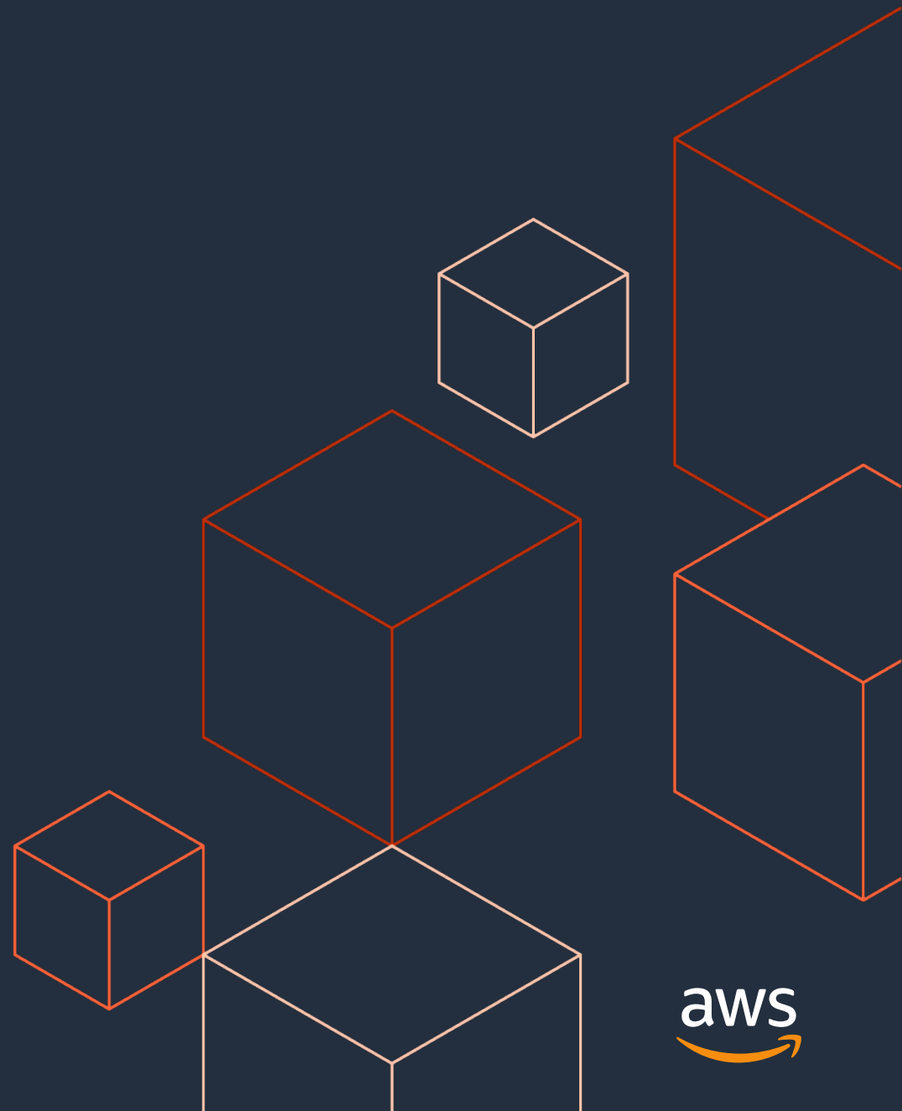


Experience



Demo 1

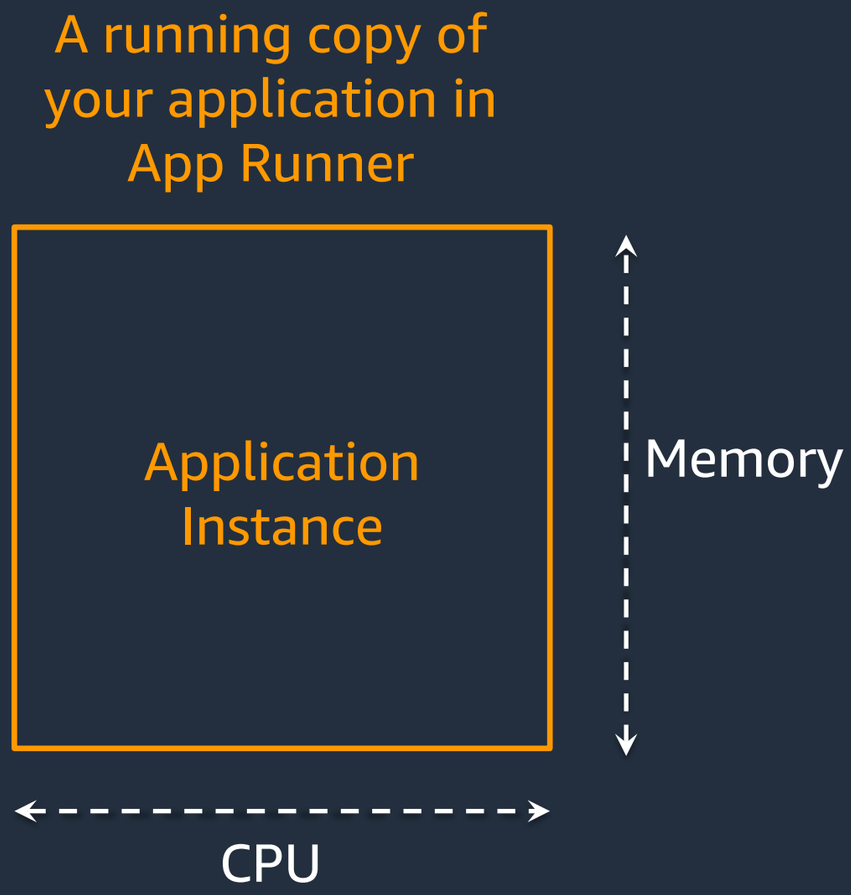
Service Creation



Compute

Max Concurrency

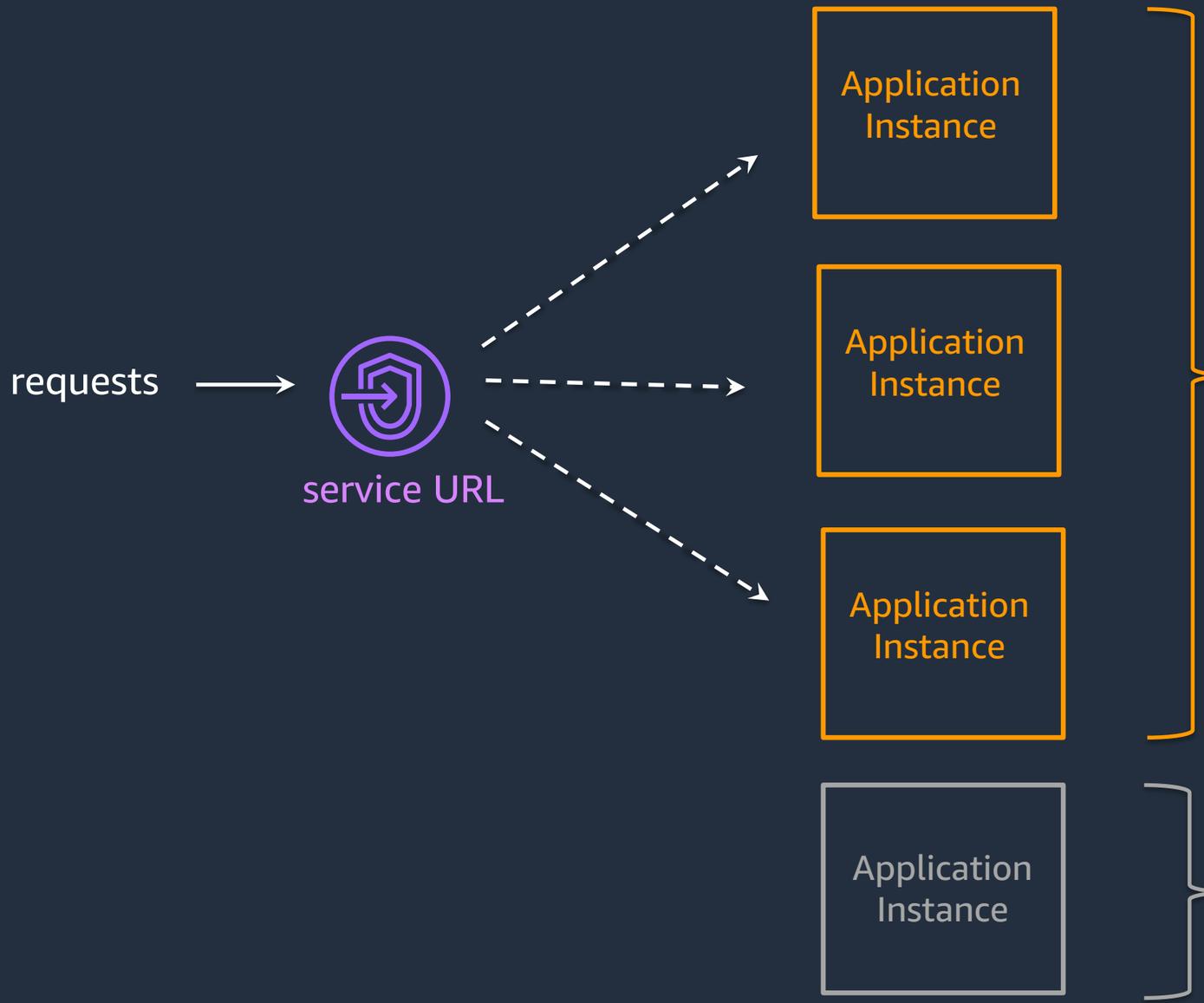
The maximum number of simultaneous requests a single application instance can handle



CPU	Memory
1 vCPU	2GB, 3GB, 4GB
2 vCPU	4GB

aws

Application instances



Active Instances

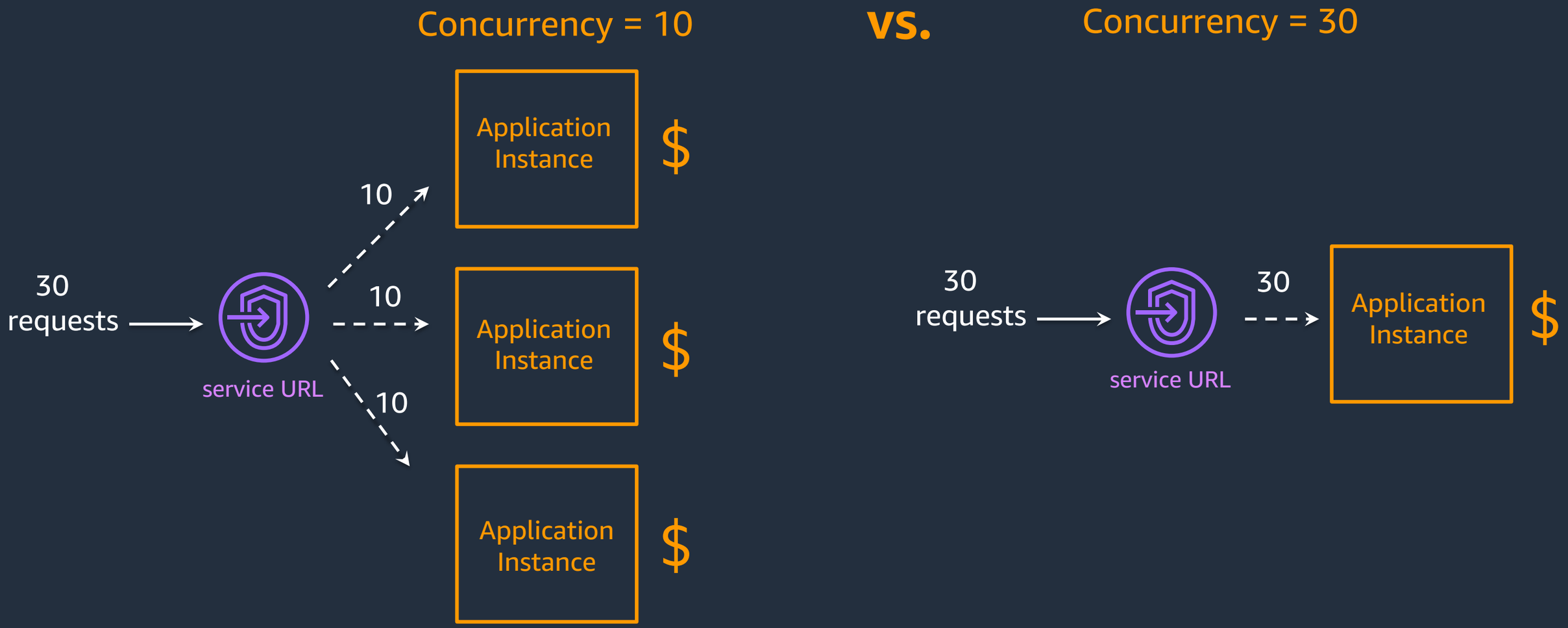
An instance is `Active` if it is processing requests.

Idle Instances

An instance is `Idle` if it is not processing any requests.
It may be resource throttled in this state

aws

Setting concurrency



aws

Networking & Storage

Networking

- Ingress: The service URL is an internet-facing endpoint
- Egress: Application instances have outbound access to the internet
- No need to setup VPC, NAT Gateway, etc.
- On the roadmap to come:
    - Ingress: Private endpoints in VPC
    - Egress: Outbound access to VPC resources

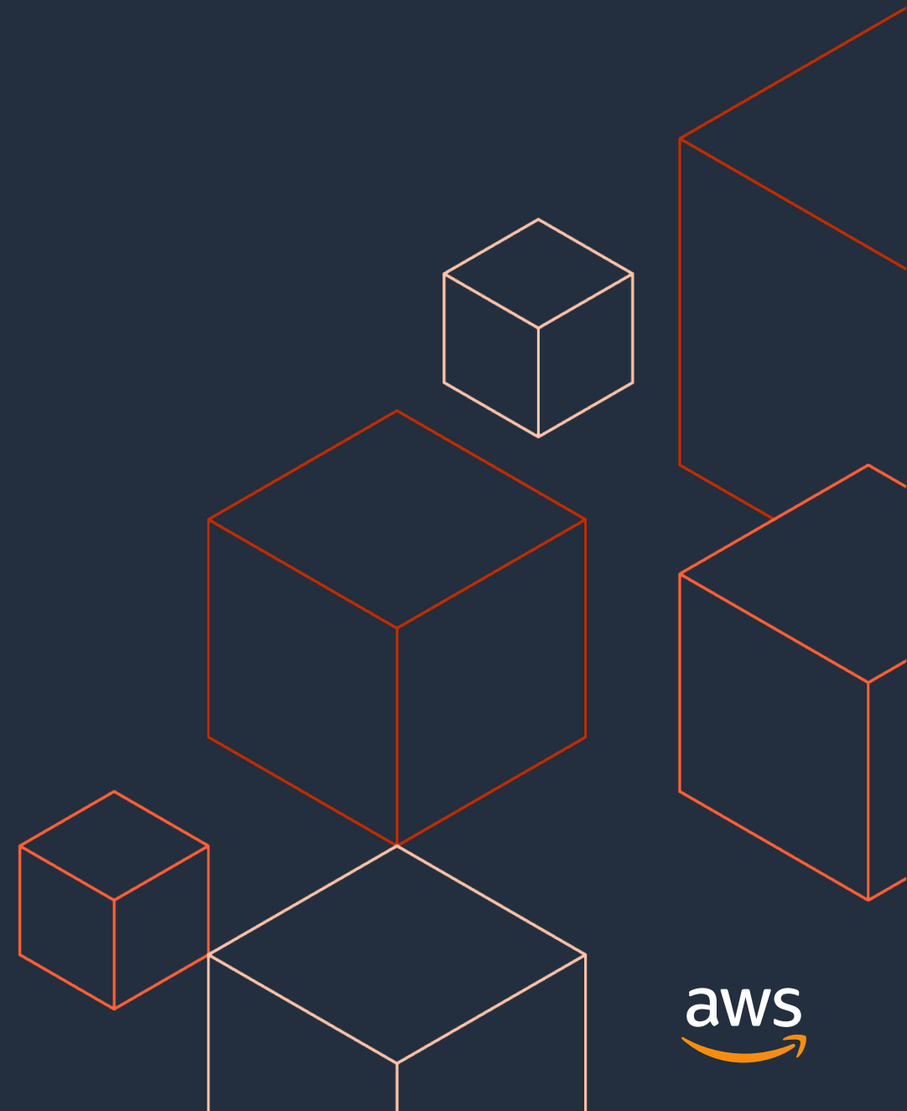
Storage

- 3GiB disk space available to each application instance
- This includes space consumed by the container image
- Storage is ephemeral

aws

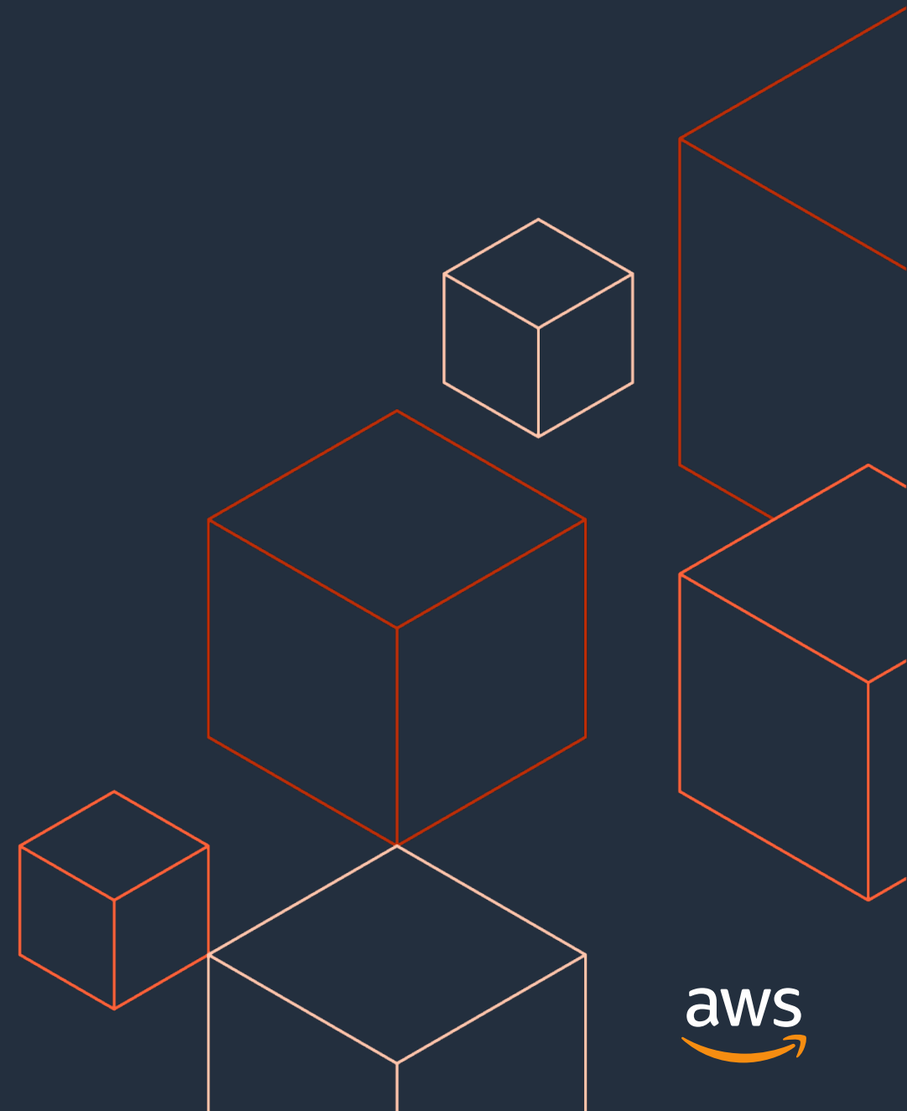
Demo 1 contd.

Service Creation



Features

Autoscaling
Deployments
Observability
Others



Autoscaling Controls

`Max Concurrency`

The maximum number of simultaneous requests a single application instance can handle

`Minimum Provisioned Instances`

Minimum provisioned instances to avoid cold start latencies

`Maximum Instances`

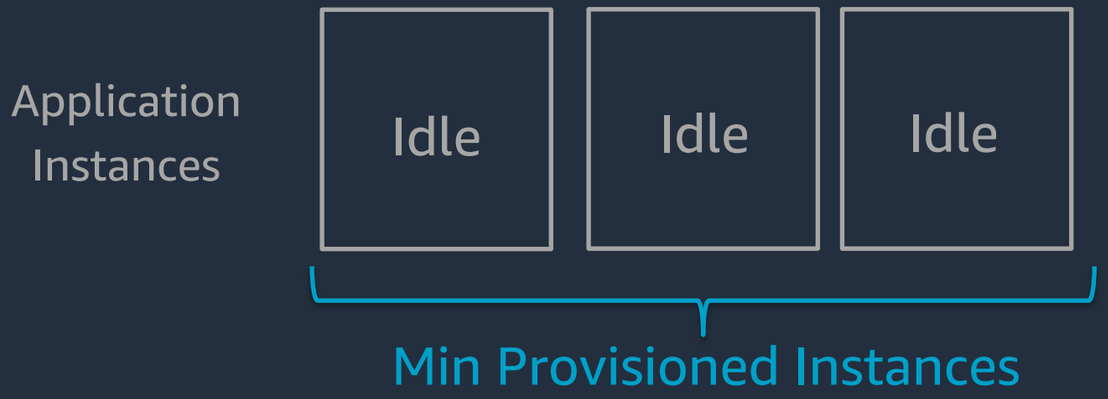
Upper bound on the number of instances launched to control cost

aws

Autoscaling Example

`concurrency = 30; min=3; max=5`

No load

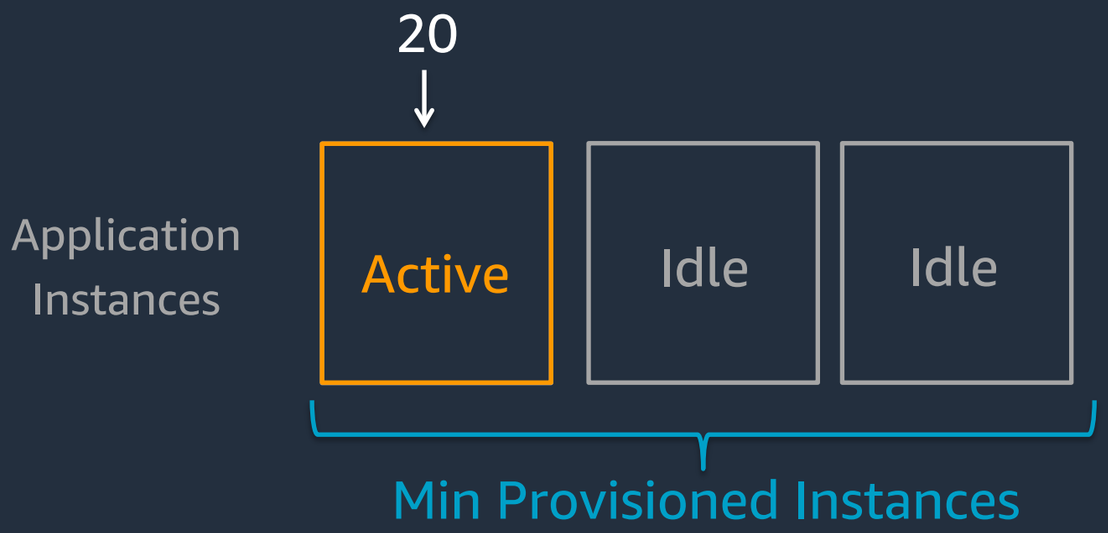


aws

Autoscaling Example

`concurrency = 30; min=3; max=5`

Load: 20 concurrent requests



aws

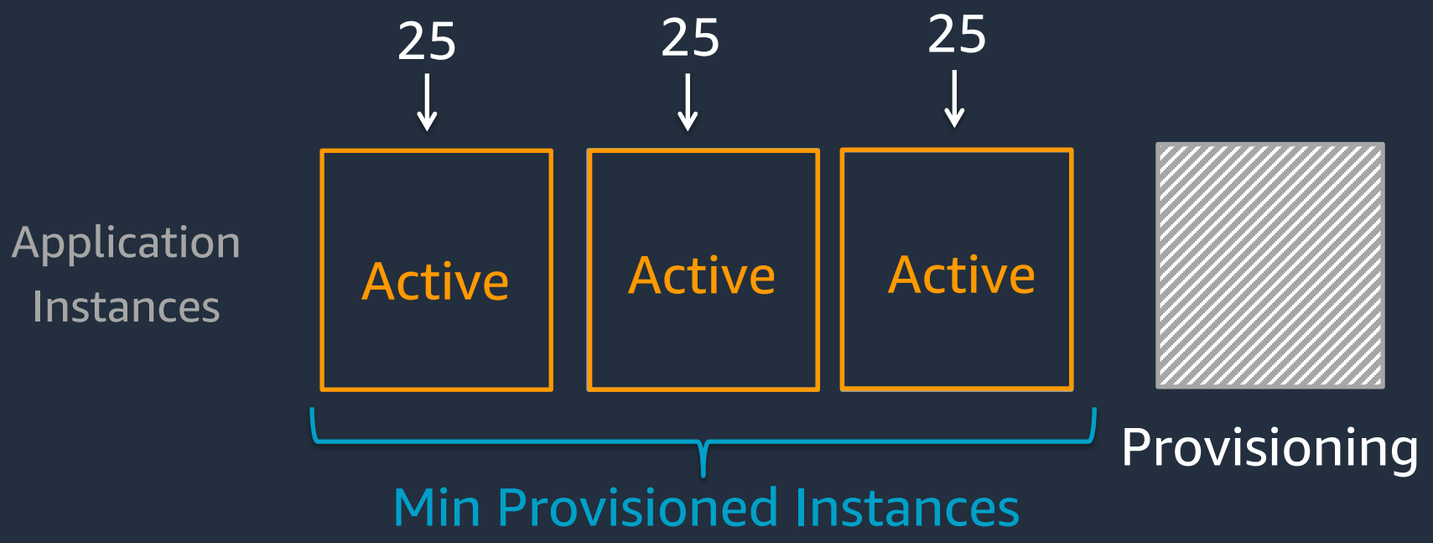
Demo 2

Autoscaling @ 20 concurrent requests

Autoscaling Example

`concurrency = 30; min=3; max=5`

Load: 75 concurrent requests

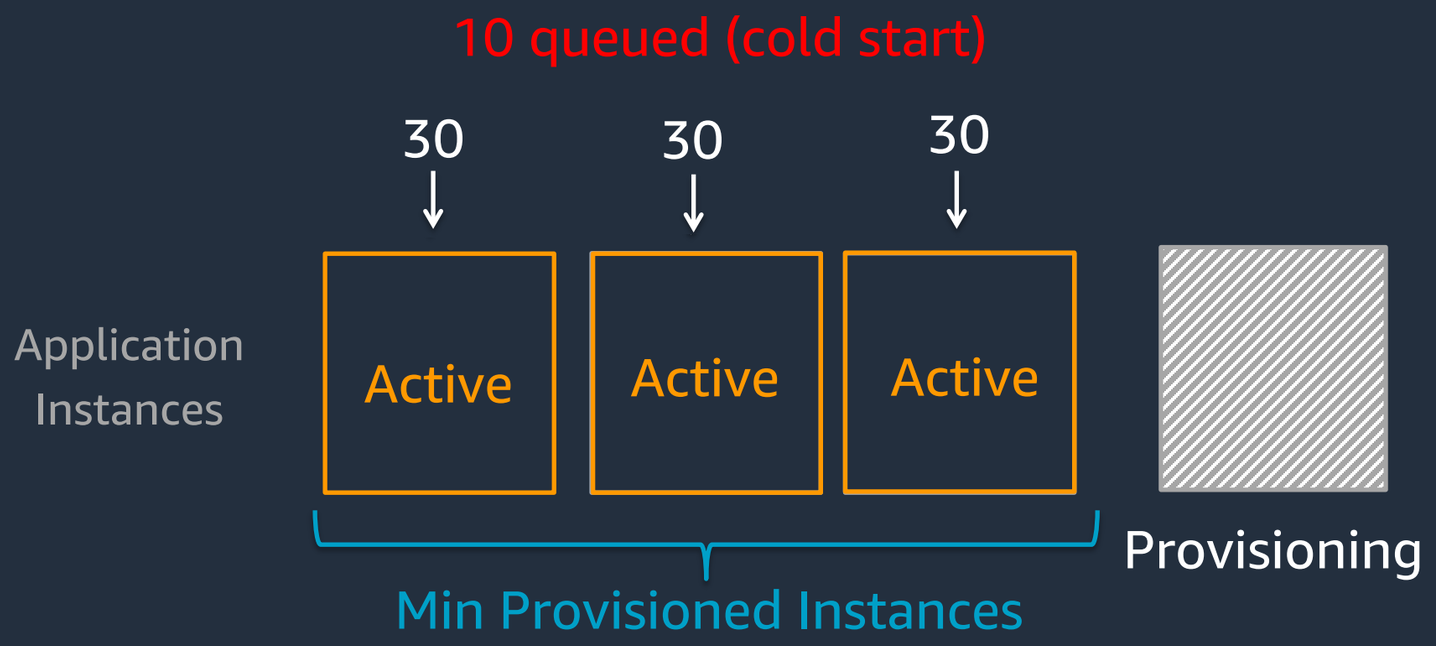


aws

Autoscaling Example

`concurrency = 30; min=3; max=5`

Load: 100 concurrent requests

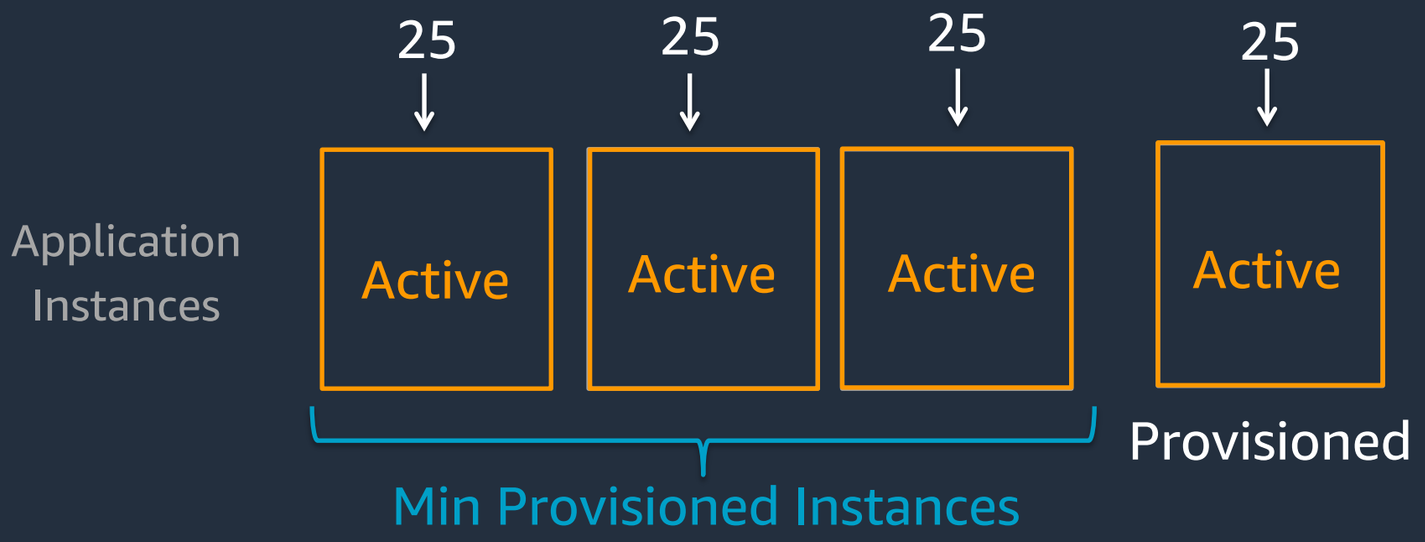


aws

Autoscaling Example

`concurrency = 30; min=3; max=5`

Load: 100 concurrent requests



aws

Demo 2 contd.

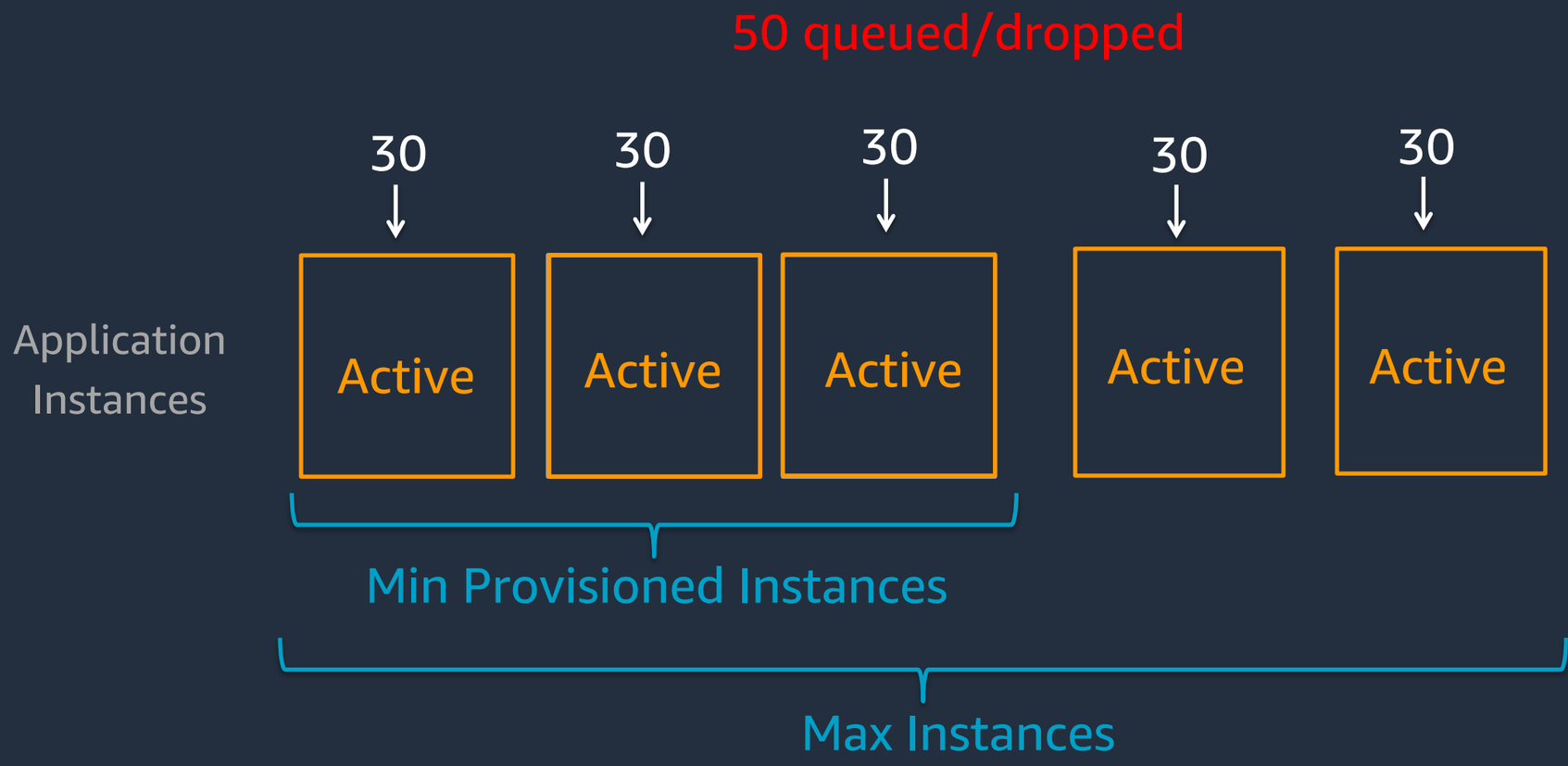
Autoscaling @ 100 concurrent requests

aws

Autoscaling Example

`concurrency = 30; min=3; max=5`

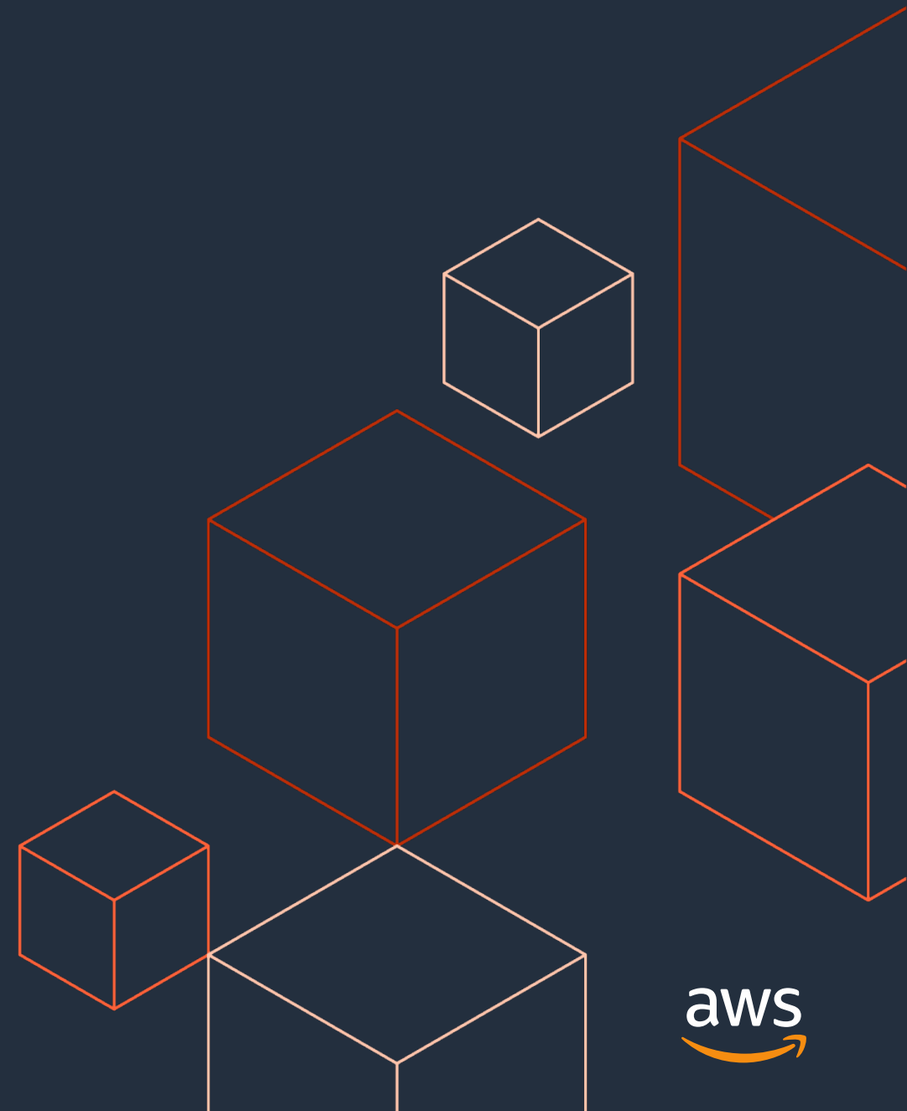
Load: 200 concurrent requests



aws

Features

Autoscaling
Deployments
Observability
Others



aws

Deployment features

Auto deploy on GitHub code commit or ECR image push
or
`StartDeployment` API to trigger a manual deployment

Blue-green deployment, no downtime

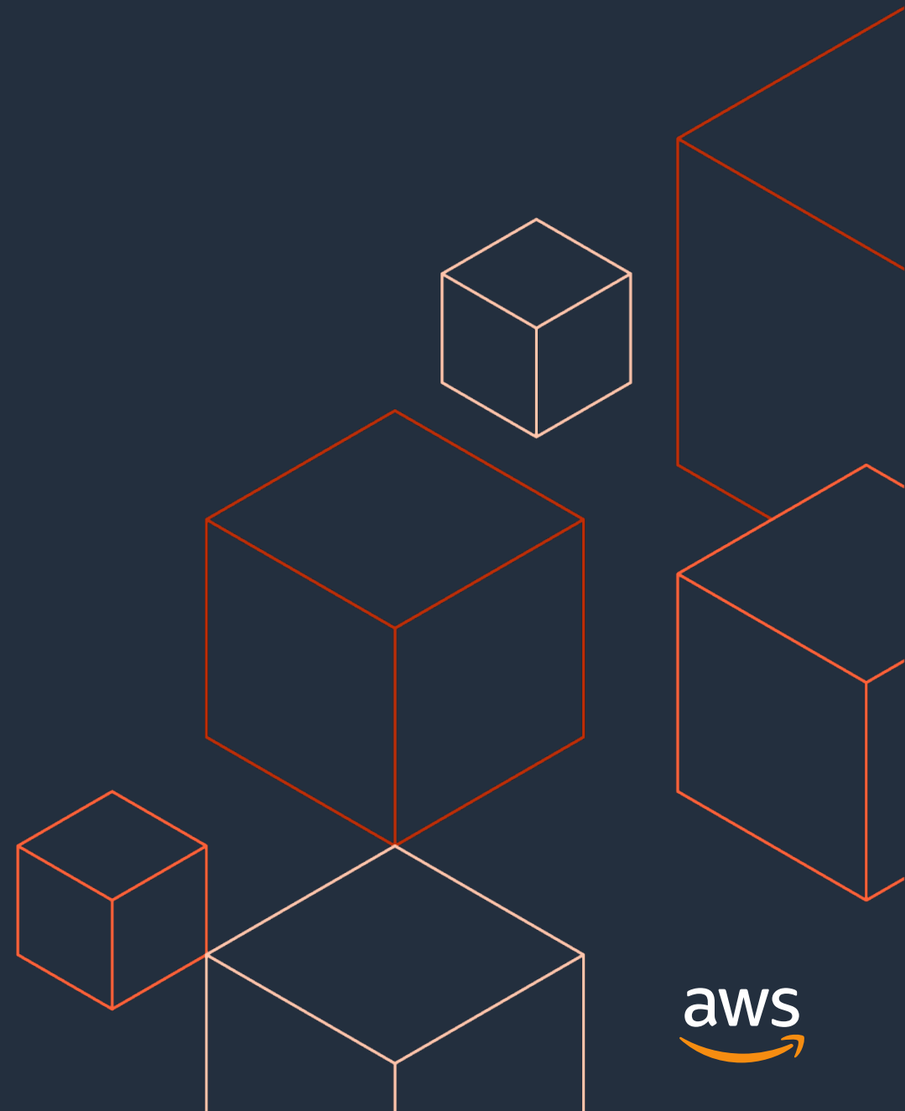
Automatic rollback if health-check fails

---

aws

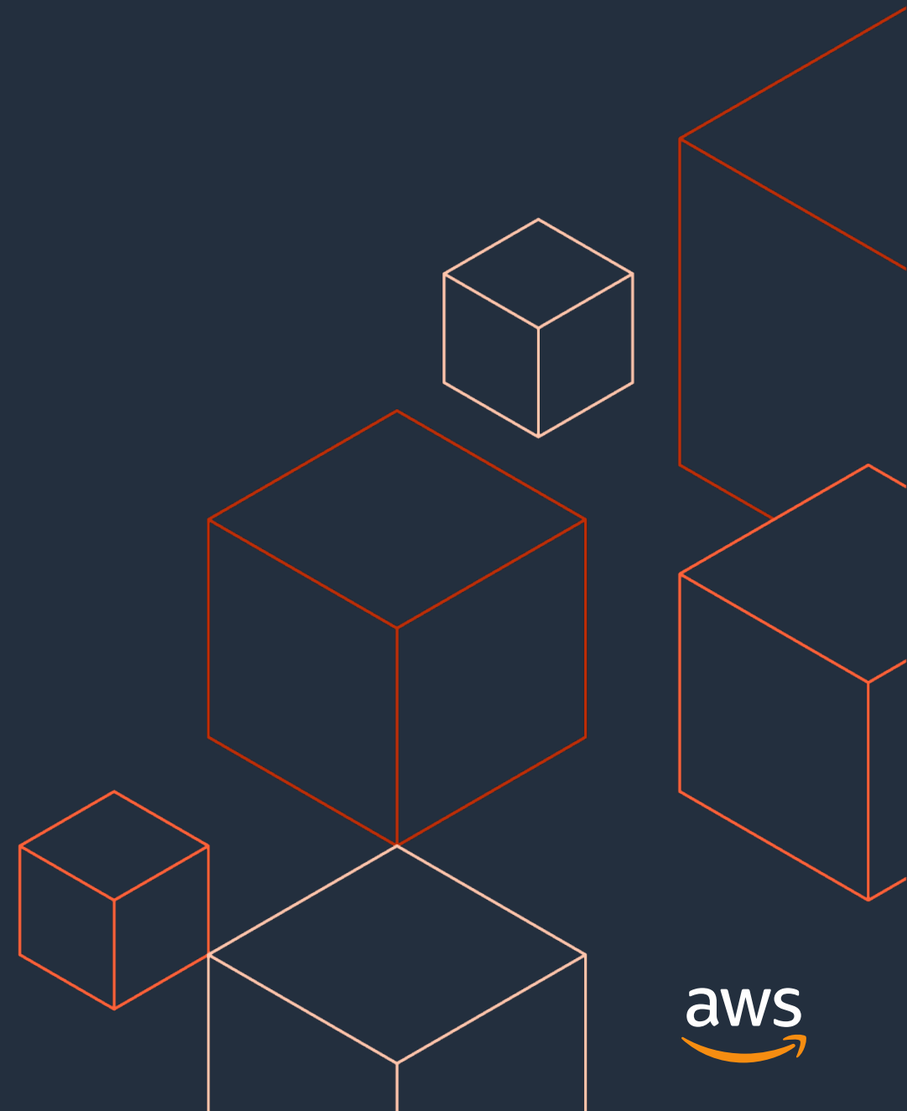
Demo 3

Auto Deploy



Features

Autoscaling
Deployments
Observability
Others



Logging

Logs delivered to AWS Cloudwatch

Three types of logs:

Deployment Logs

Build output

Application Logs

Logs emitted by your web application

Event Logs

Log of lifecycle operations executed by App Runner during service creation & deployments

aws

Metrics

Metrics delivered to AWS Cloudwatch

Request Metrics

Request count, latency, HTTP error codes

Active Instances

Number of application instances actively processing requests

aws

Events

Service Lifecycle Events

Service & Deployment status change events

Delivered to Amazon Event Bridge

Can be used to initiate rule-based automated actions in targets like AWS Lambda, Amazon ECS, Amazon SNS, etc

---

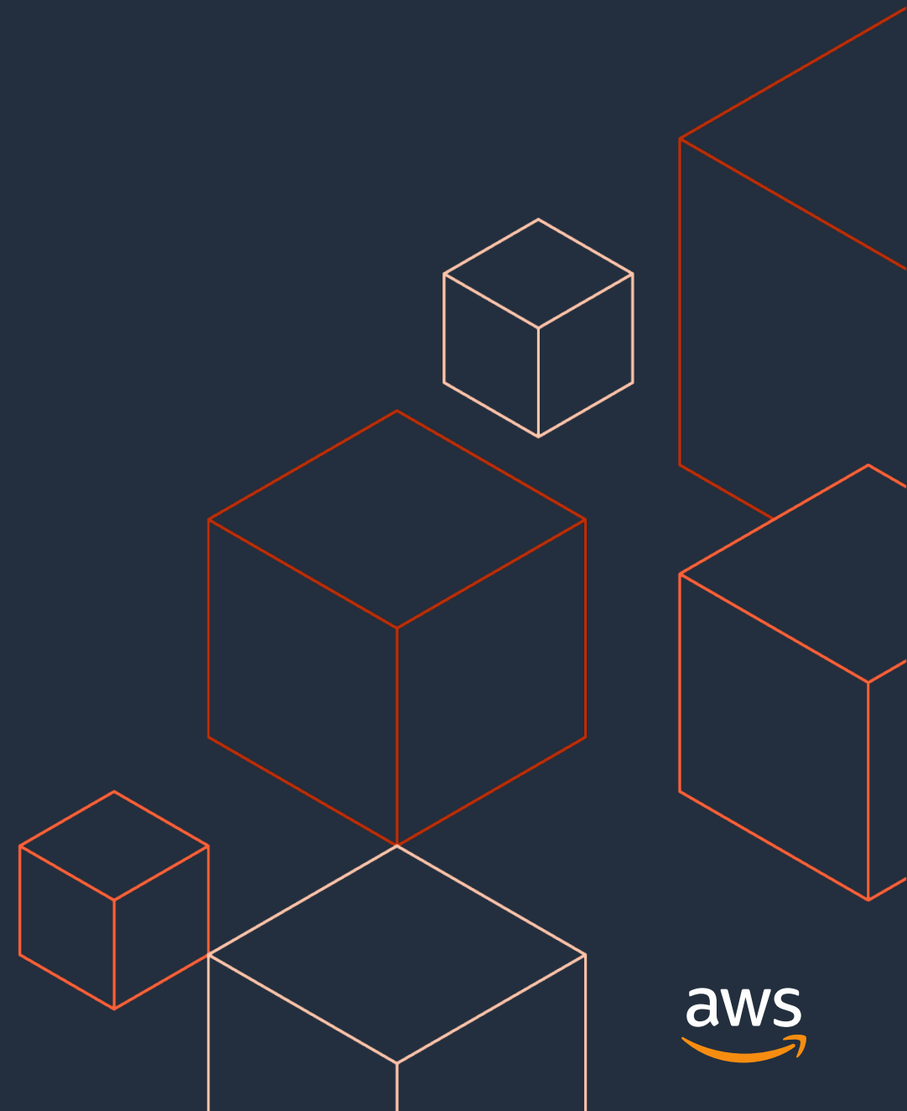
API Actions

API calls logged to AWS CloudTrail for auditing

aws

Features

Autoscaling
Deployments
Observability
Others



Other Features

Custom Domain Names

Associate custom domain names with your service

App Runner will create and manage certificates behind the scenes

Data Encryption

Source code and images pulled by App Runner are encrypted at rest using AWS managed keys (default) or customer provided keys

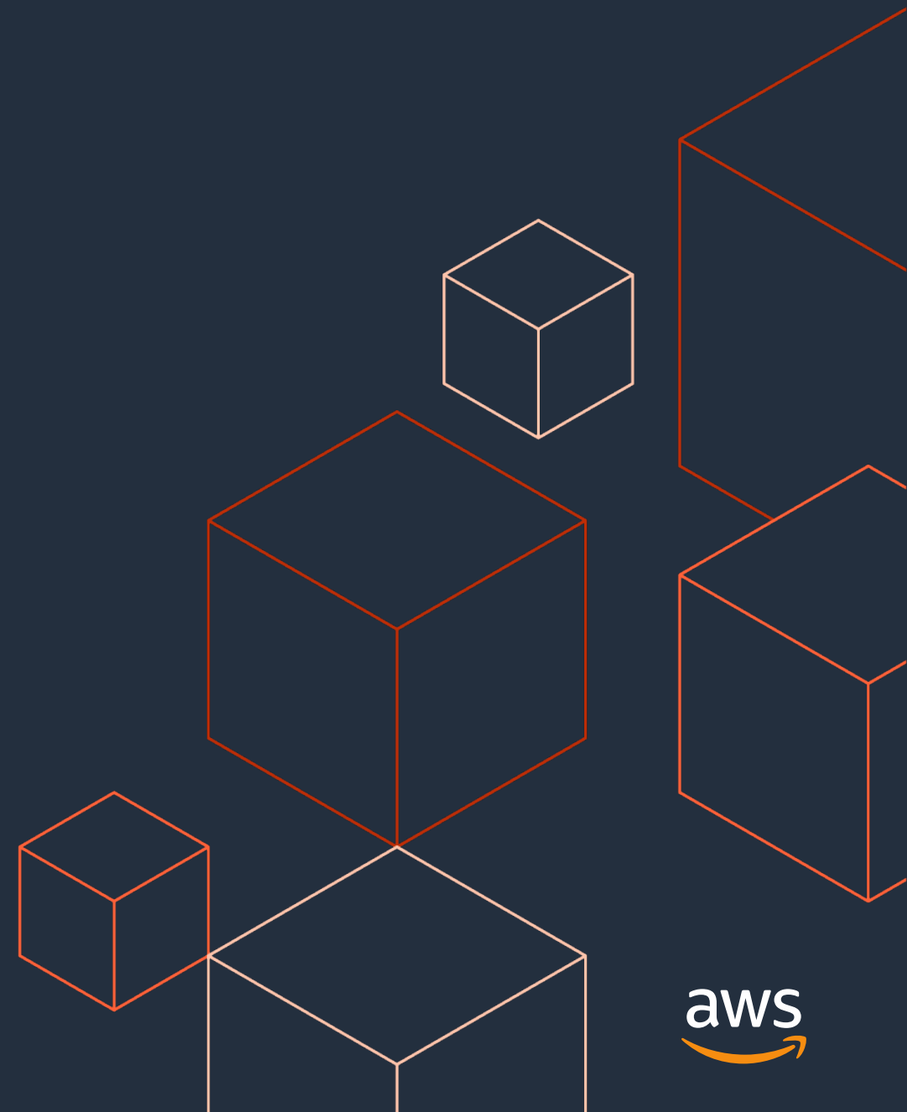
Pause/Resume Service

Services can be temporarily disabled using the Pause/Resume feature

While paused, App Runner reduces the instances behind the service to zero

aws

Pricing



Pricing Dimensions

CPU & Memory

Pay for CPU and memory of active instances

When idle, only pay for memory of min provisioned instances

Build Fee

$0.005 / build-minute

Only applies when starting with source code. Does not apply if starting with a container image

Automated Deployment Fee

$1 / service, per month

Only applies if automated deployments are enabled

AWS Data Transfer

Standard AWS Data Transfer fees apply to application traffic

aws

CPU/Memory Pricing Example

`concurrency = 30; min=3; max=5`

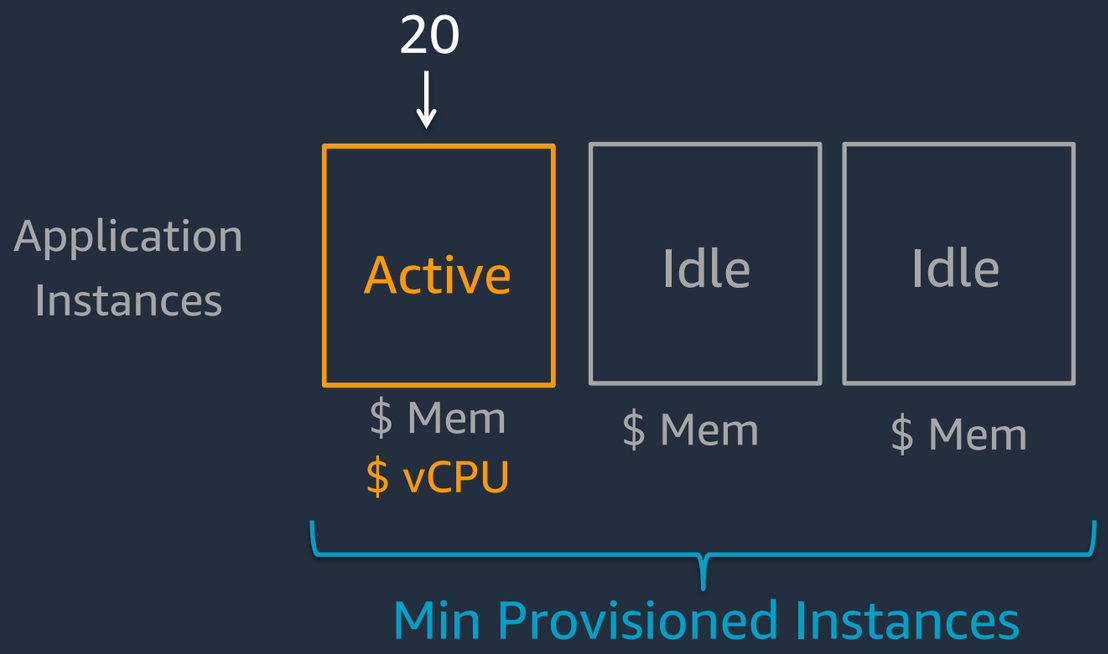
No Load

aws

CPU/Memory Pricing Example

`concurrency = 30; min=3; max=5`

Load: 20 concurrent requests

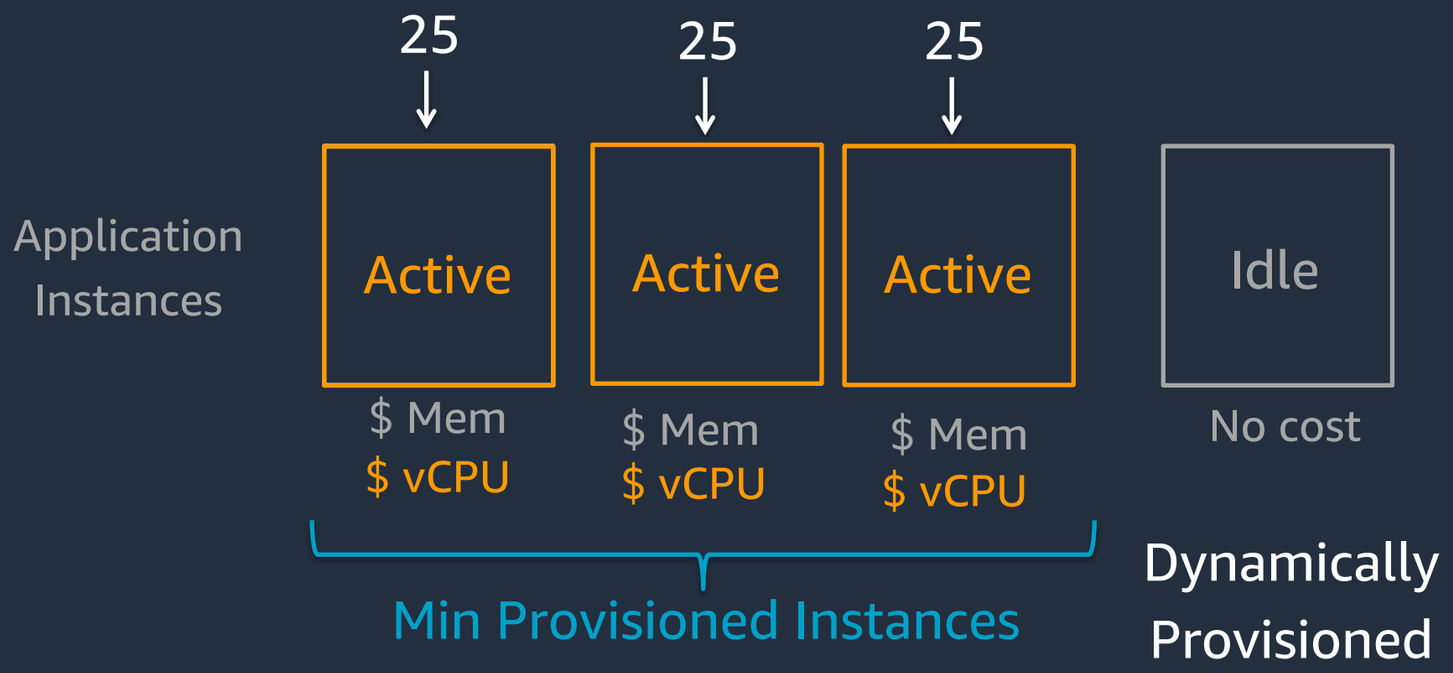


aws

CPU/Memory Pricing Example

`concurrency = 30; min=3; max=5`

Load: 75 concurrent requests

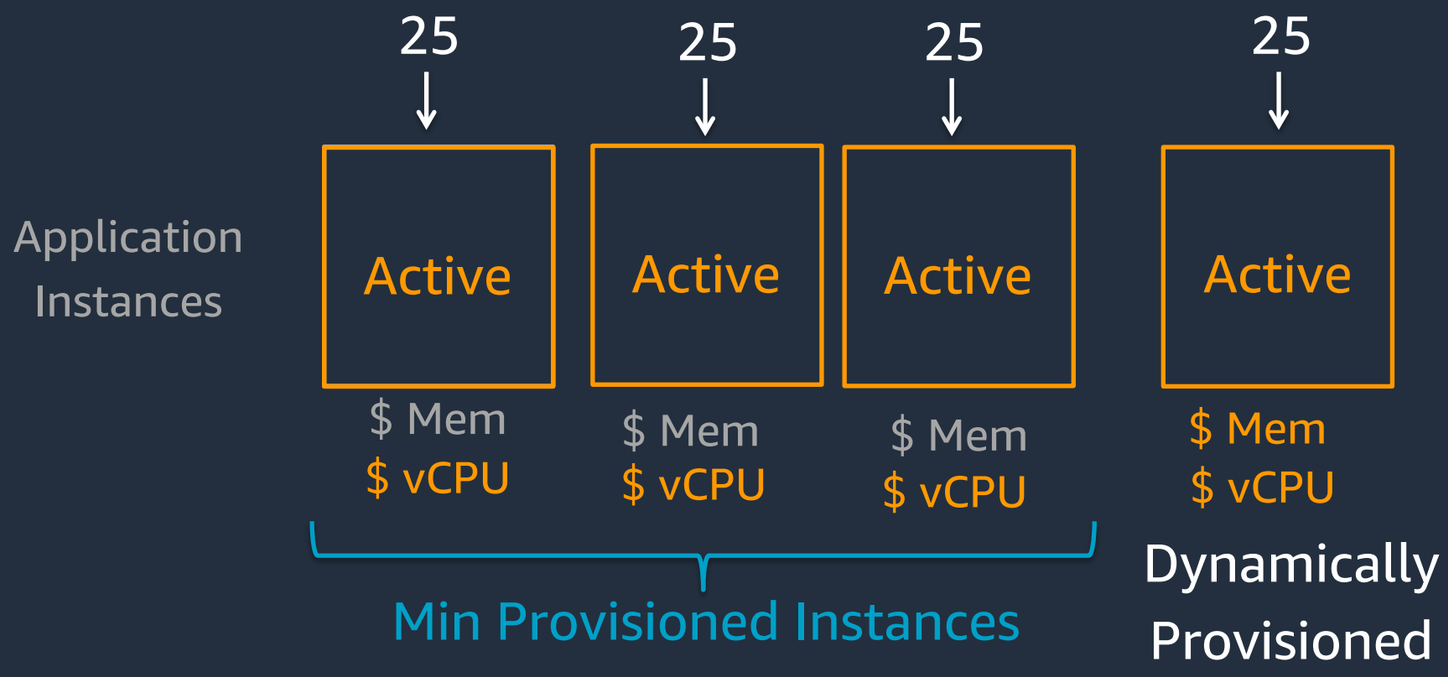


aws

CPU/Memory Pricing Example

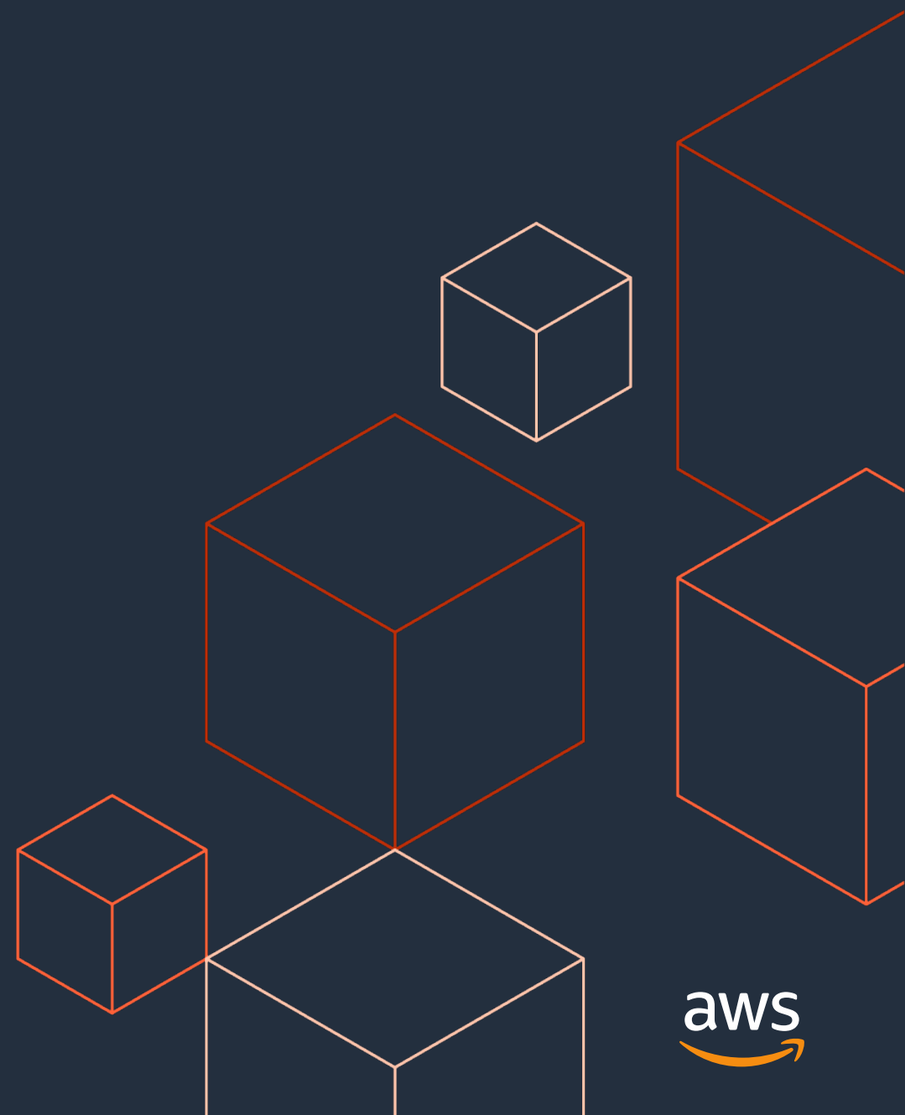
`concurrency = 30; min=3; max=5`

Load: 100 concurrent requests



aws

Take Aways



Take Aways

App Runner is:

- A simplified experience for running HTTP request/reply services
- Containers, language runtime, load-balancing, autoscaling, CI/CD infrastructure are abstracted away and managed behind the scenes
- Pay by use, rather than provisioned resources

Check out the public roadmap @ https://github.com/aws/apprunner-roadmap

Get started with your app! https://aws.amazon.com/apprunner/

aws

Thank you!

@ArchanaSrikanta