



AWS Introduction

Building / Deploying Applications in the Cloud

Christoph Schnidrig, Manager Solutions Architecture

Christian Volkmer, Solutions Architect

08.01.2021

About us...



Agenda

12:15 - 13:00h Theorie Teil 1

- AWS Einführung
- AWS Global Infrastructure
- AWS Service Portfolio
- Example Architecture: Grails App on AWS

Pause 13:00 - 13:10h

13:10 - 14:10h Hands-On Lab 1

Pause 14:10 - 14:20h

14:20 - 15:00h Hands-On Lab 2

Our mission

we want to be earth's most
customer centric company

Our culture of innovation

Customer obsession

"Start every process with the customer and work backwards."

Long term thinking

"Be stubborn on the vision but flexible on the details."

If you want to be inventive, you have to be willing to fail

"We are willing to go down on a bunch of dark alleys and occasionally we find something that really works."

You have to be willing to be misunderstood for a long time

"We are very comfortable being misunderstood."

Two pizza teams

- » Decompose to primitive problems
- » Single-threaded focus
- » Enables self-directed teams
- » Fosters ownership and autonomy



Innovation has Manifested Across Many Domains...



Drone Development



Video Streaming



Kindle Reader



Home Entertainment



Delivery



Advanced Shopping



Home Automation



Cloud Computing

AWS Global Infrastructure

24 geographical regions, 1 local region, 77 availability zones, 200+ POPs

Region & Number of Availability Zones (AZs)

GovCloud (US)
US-East (3), US-West (3)

US West
Oregon (4)
Northern California (3)

US East
N. Virginia (6), Ohio (3)

Canada
Central (3)

South America
São Paulo (3)

Africa
Cape Town (3)

Europe
Frankfurt (3), Paris (3),
Ireland (3), Stockholm (3),
London (3), Milan (3)

Middle East
Bahrain (3)

Asia Pacific
Singapore (3), Sydney (3),
Tokyo (4), Osaka-Local (1)*
Seoul (3), Mumbai (3),
Hong Kong (3)

China
Beijing (2), Ningxia (3)



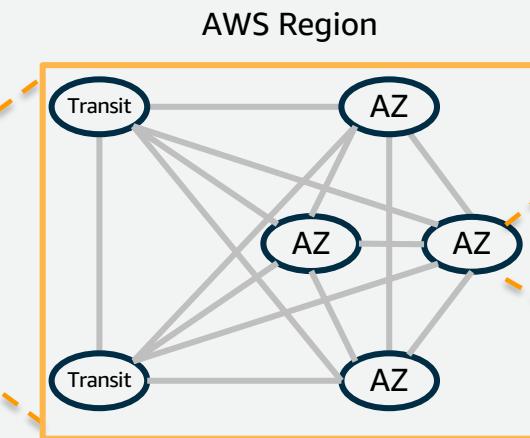
Announced Regions

6 Regions and 18 AZs in Indonesia, Japan, India, Australia, Switzerland, and Spain

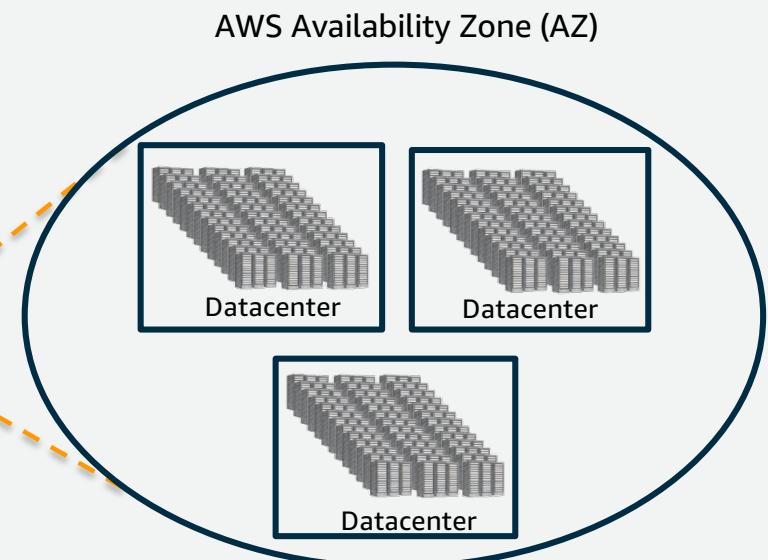
<https://aws.amazon.com/about-aws/global-infrastructure/>

AWS Region design

AWS Regions are comprised of multiple AZs for **high availability, high scalability**, and **high fault tolerance**. Applications and data are replicated in real time and consistent in the different AZs.

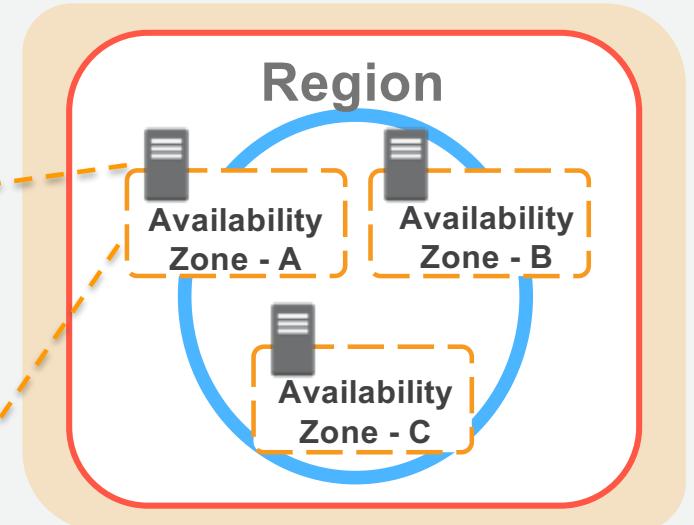
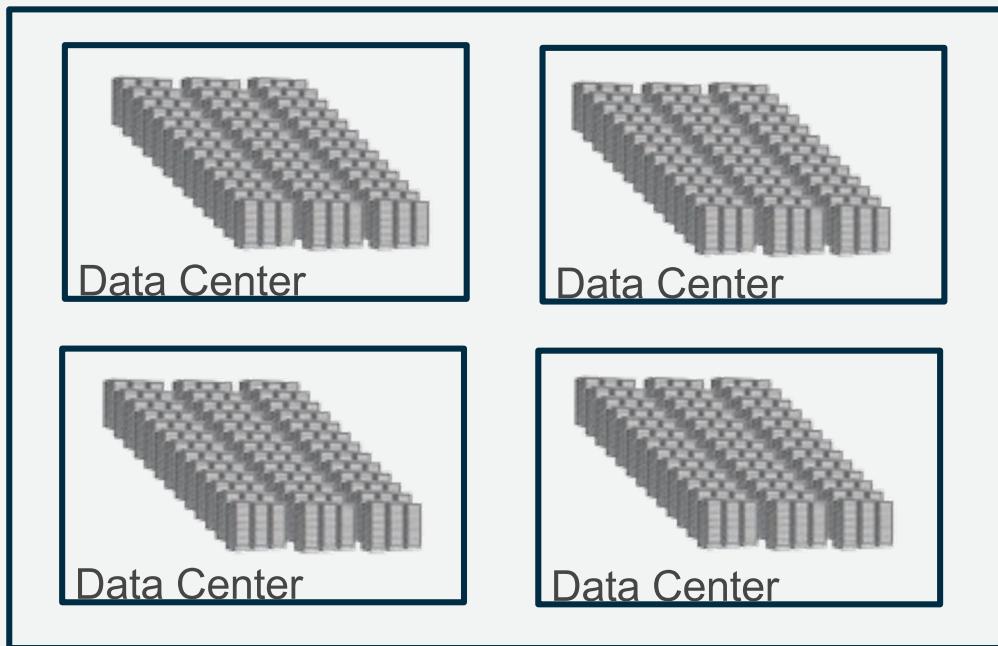


A Region is a physical location in the world where we have multiple **Availability Zones**.



Availability Zones consist of one or more discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities.

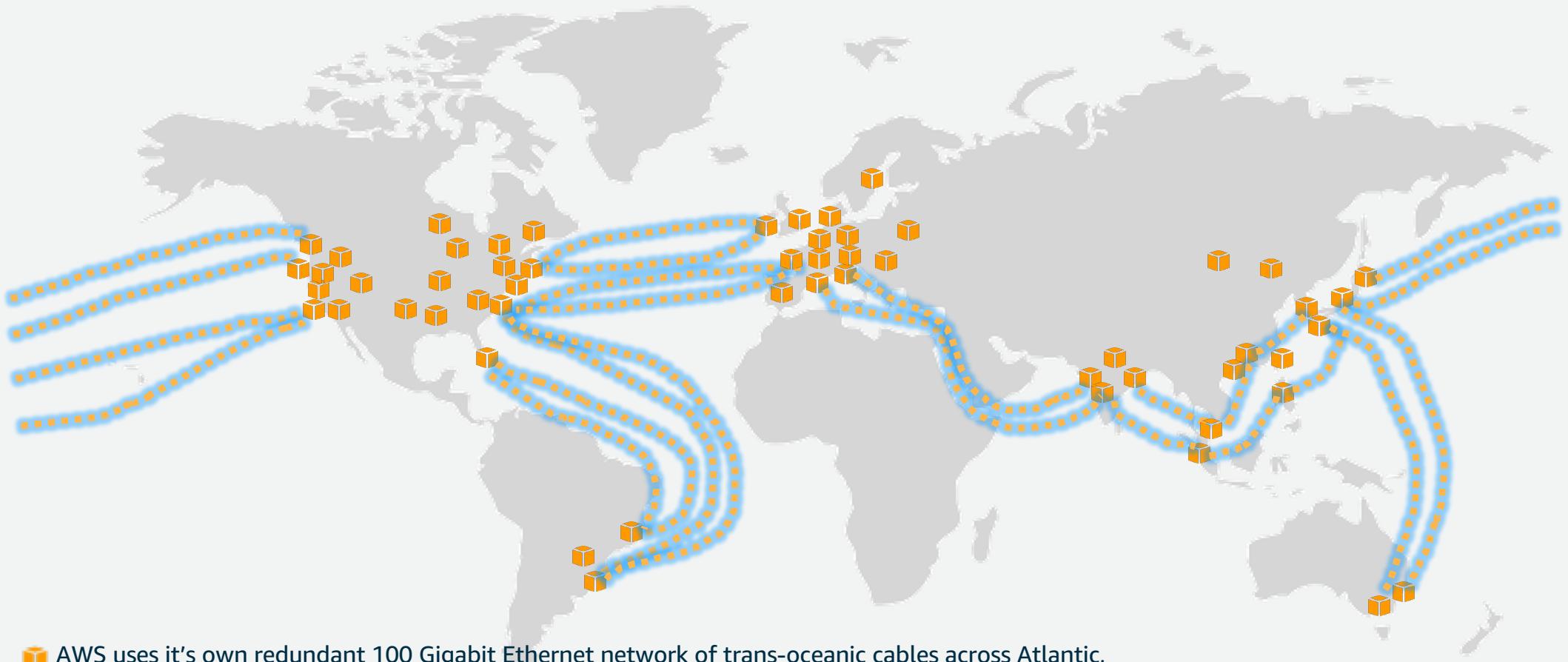
Example AWS Availability Zone



- All regions have 2 or more AZs
- Each AZ is 1 or more DC
 - No data center is in two AZs
 - Some AZs have as many as 6 DCs
- DCs in AZ less than 0.25 ms apart

<https://www.youtube.com/watch?v=uj7Ting6Ckk&t=6s>

The AWS Network



- AWS uses its own redundant 100 Gigabit Ethernet network of trans-oceanic cables across Atlantic, Pacific and Indian Oceans and the Mediterranean, Red and South China Seas.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



AWS custom hardware



Routers



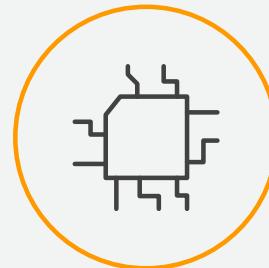
Load
balancers



Custom
compute
servers and
semiconductors

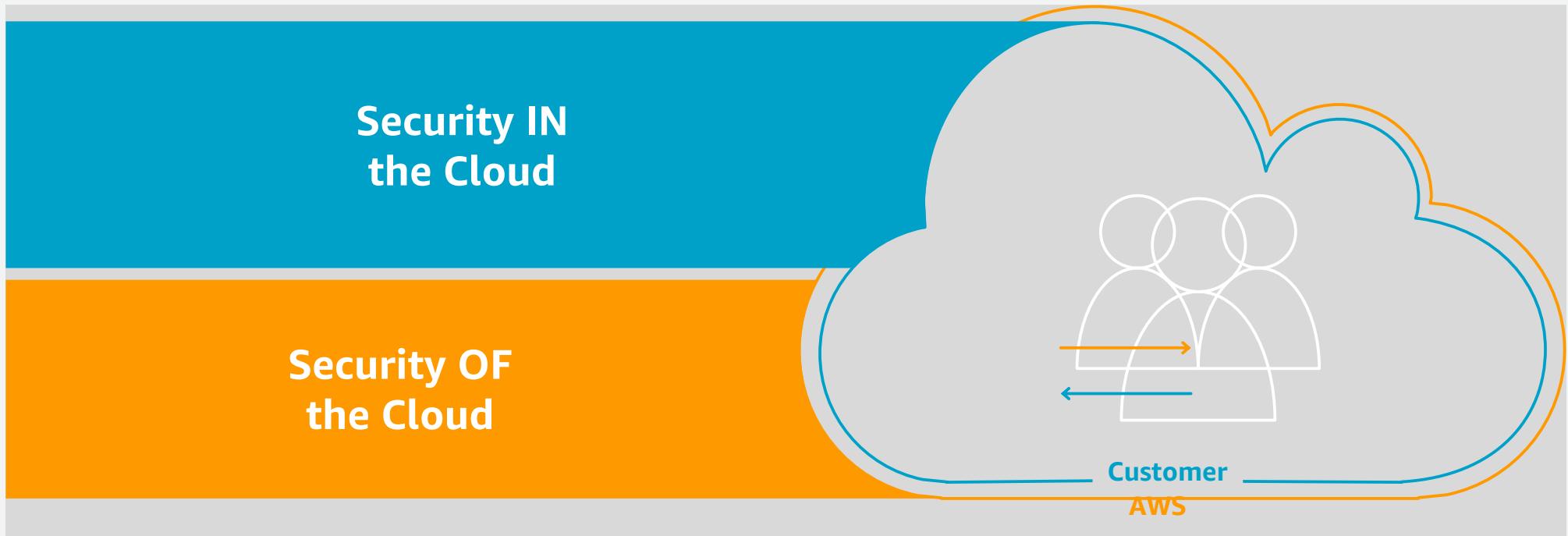


Custom
software

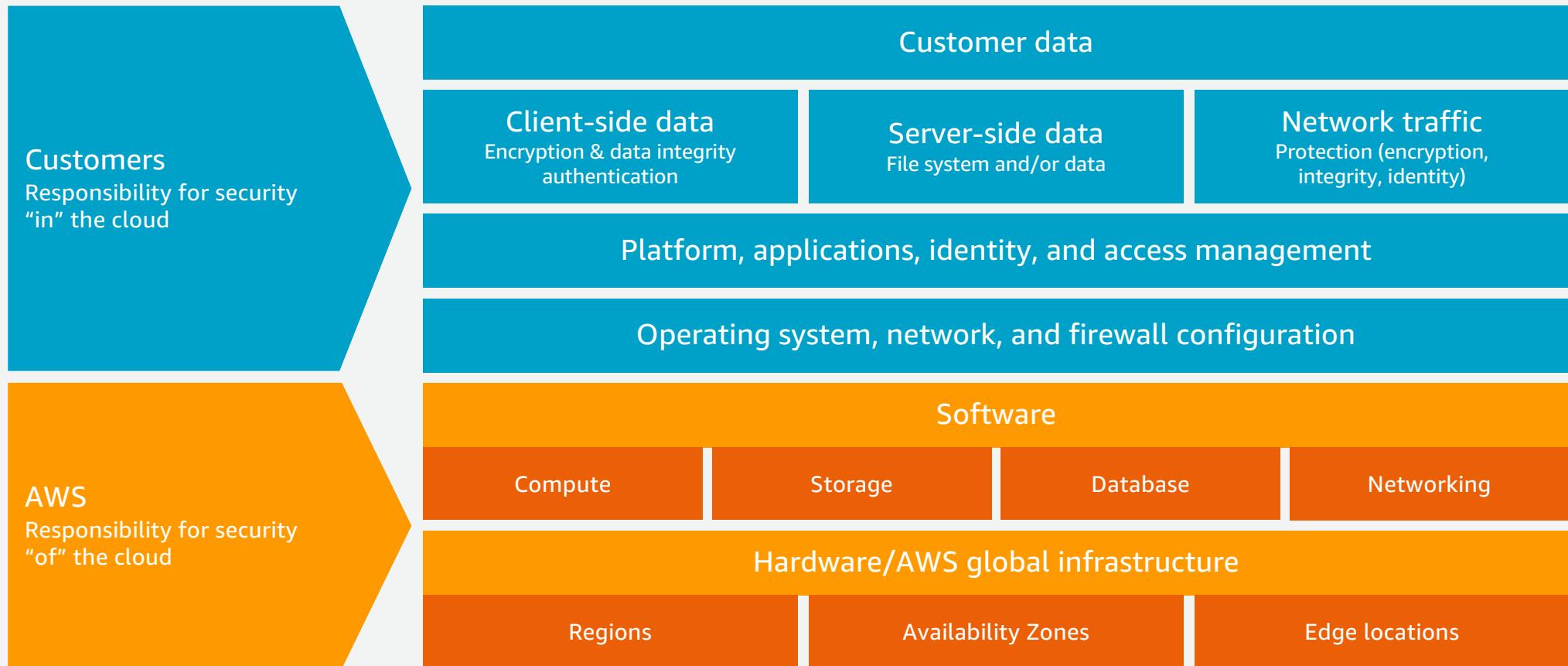


Silicon

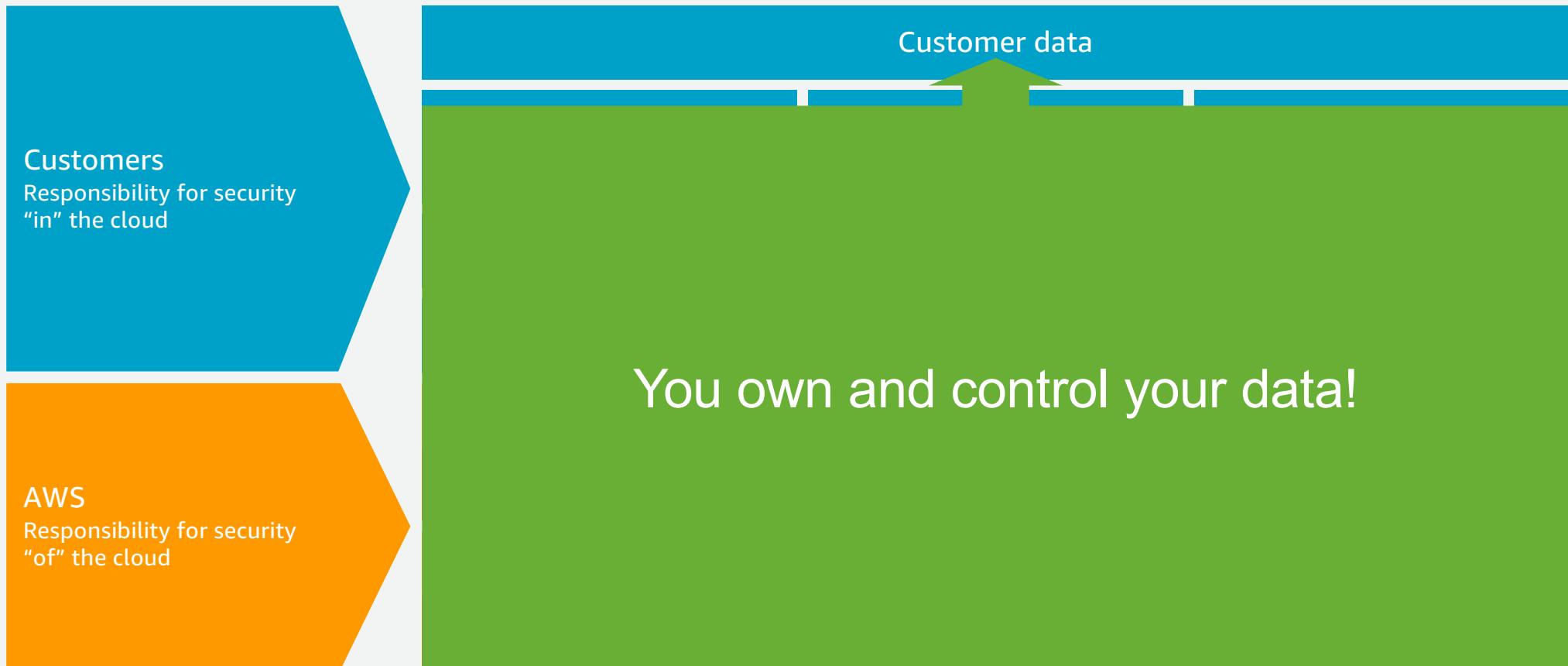
Shared responsibility model



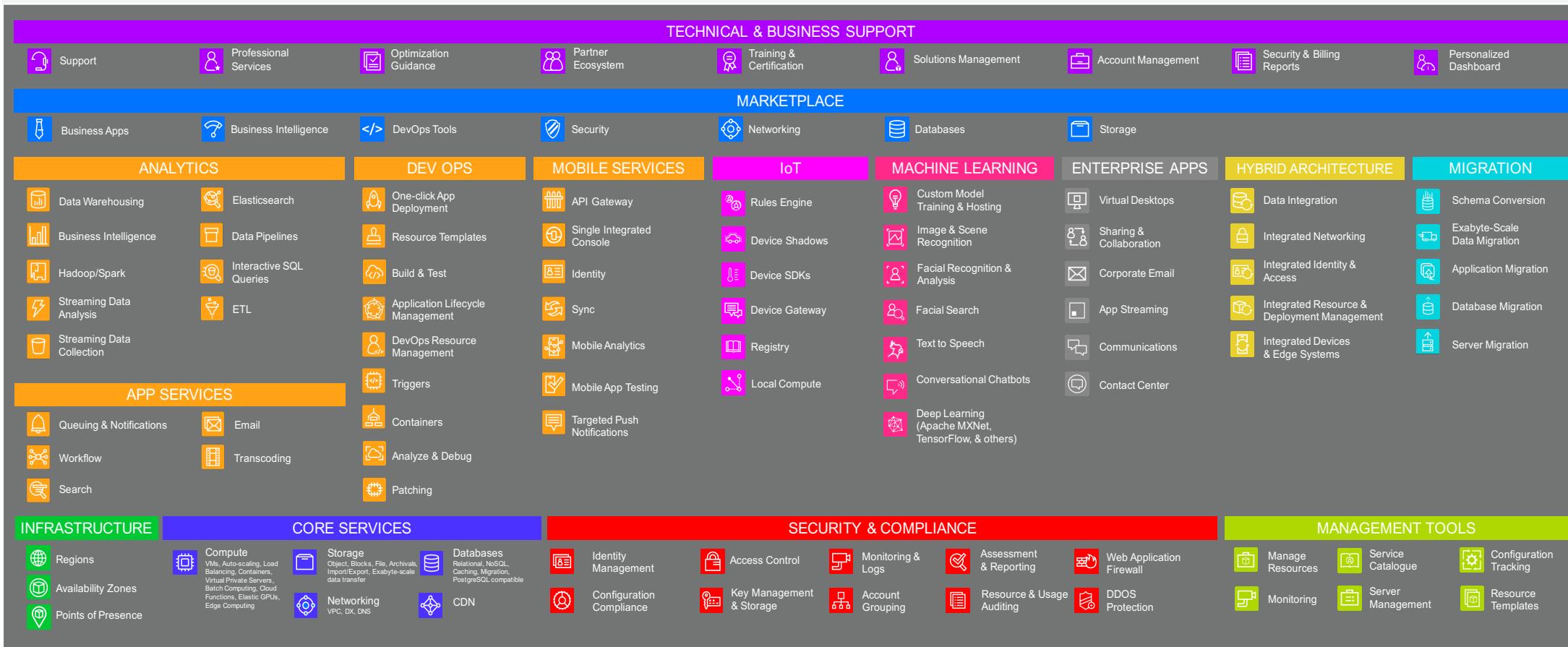
Understanding the AWS Shared Responsibility Model



Understanding the AWS Shared Responsibility Model

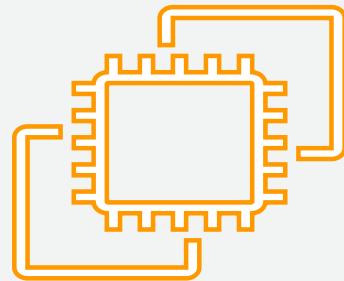


The AWS Platform





Amazon Compute Options



Amazon EC2

Virtual server
instances
in the cloud



Amazon ECS, EKS, and Fargate

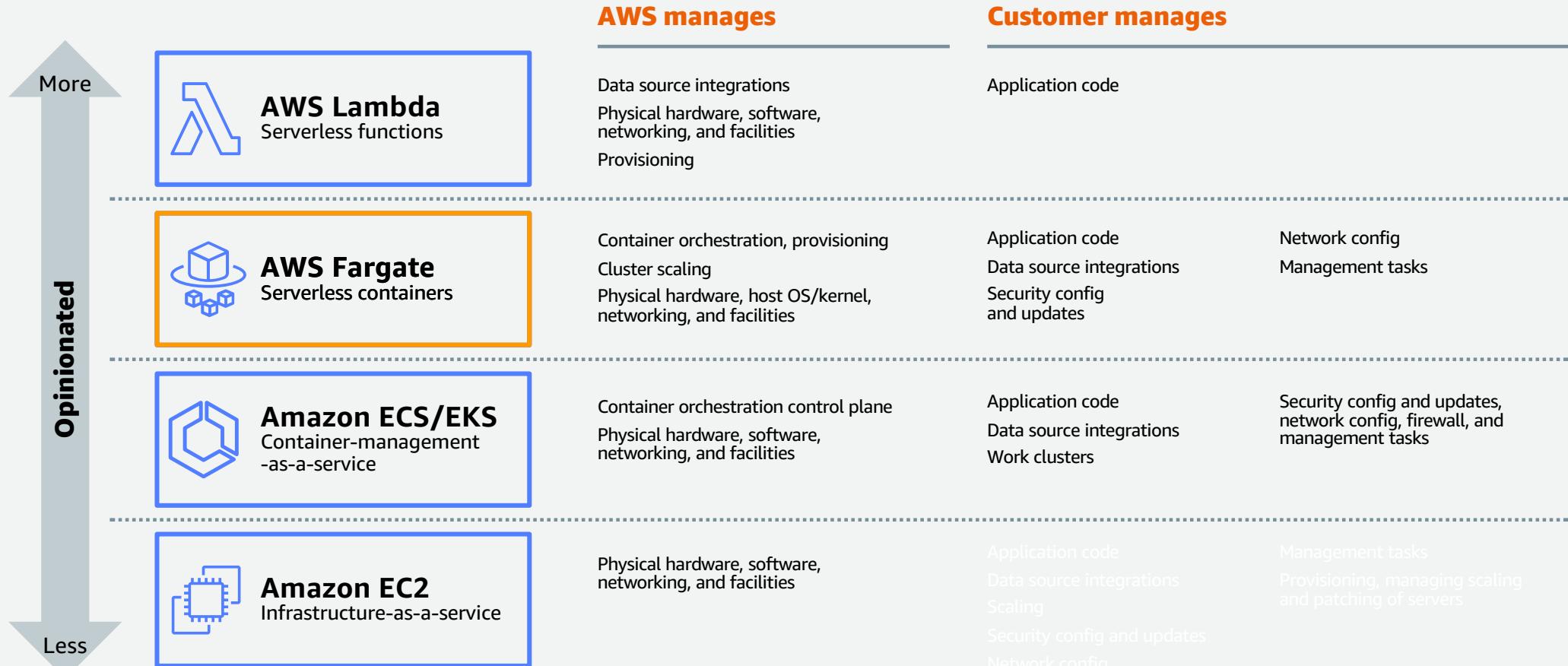
Managed Container
Services



AWS Lambda

Serverless compute
for stateless code
execution in response
to triggers (events)

AWS Compute Options



Deploying to AWS managed container services

Managed container services

1

Choose your orchestration tool

ECS

EKS

2

Choose your launch type



EC2



Fargate



EC2



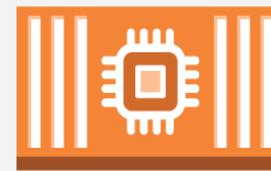
Fargate



Fargate launch type: You're responsible for ...



**Container
instances**



Services



Tasks

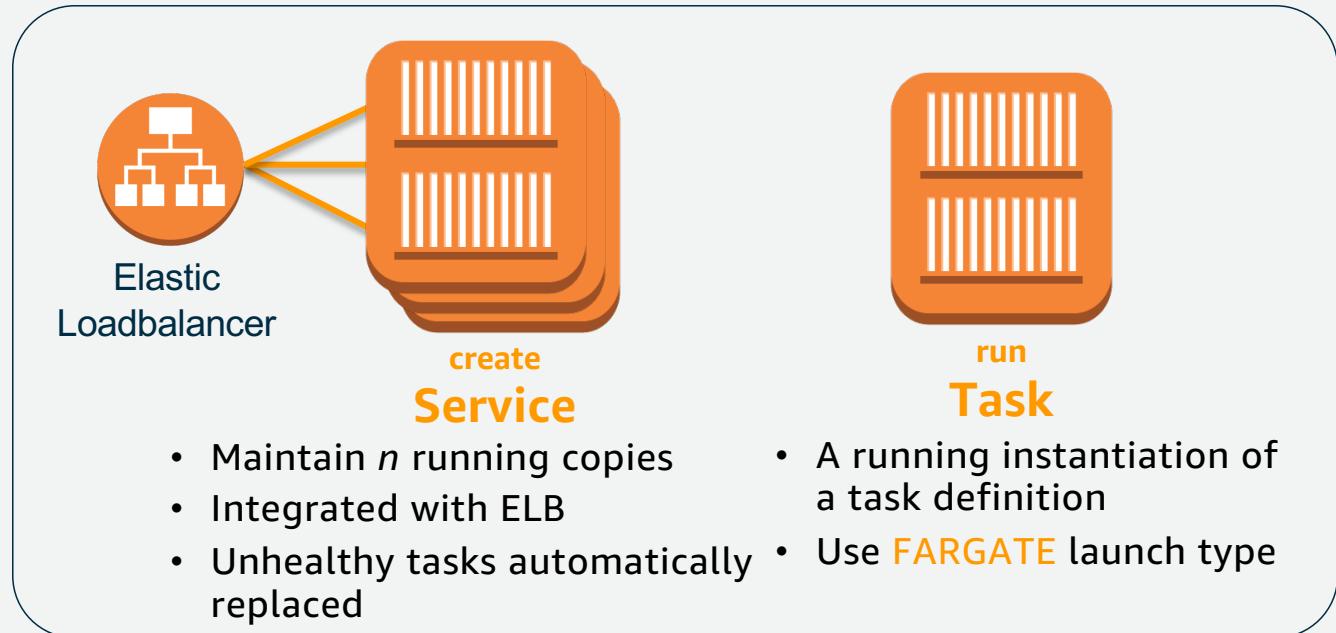


Fargate compute constructs

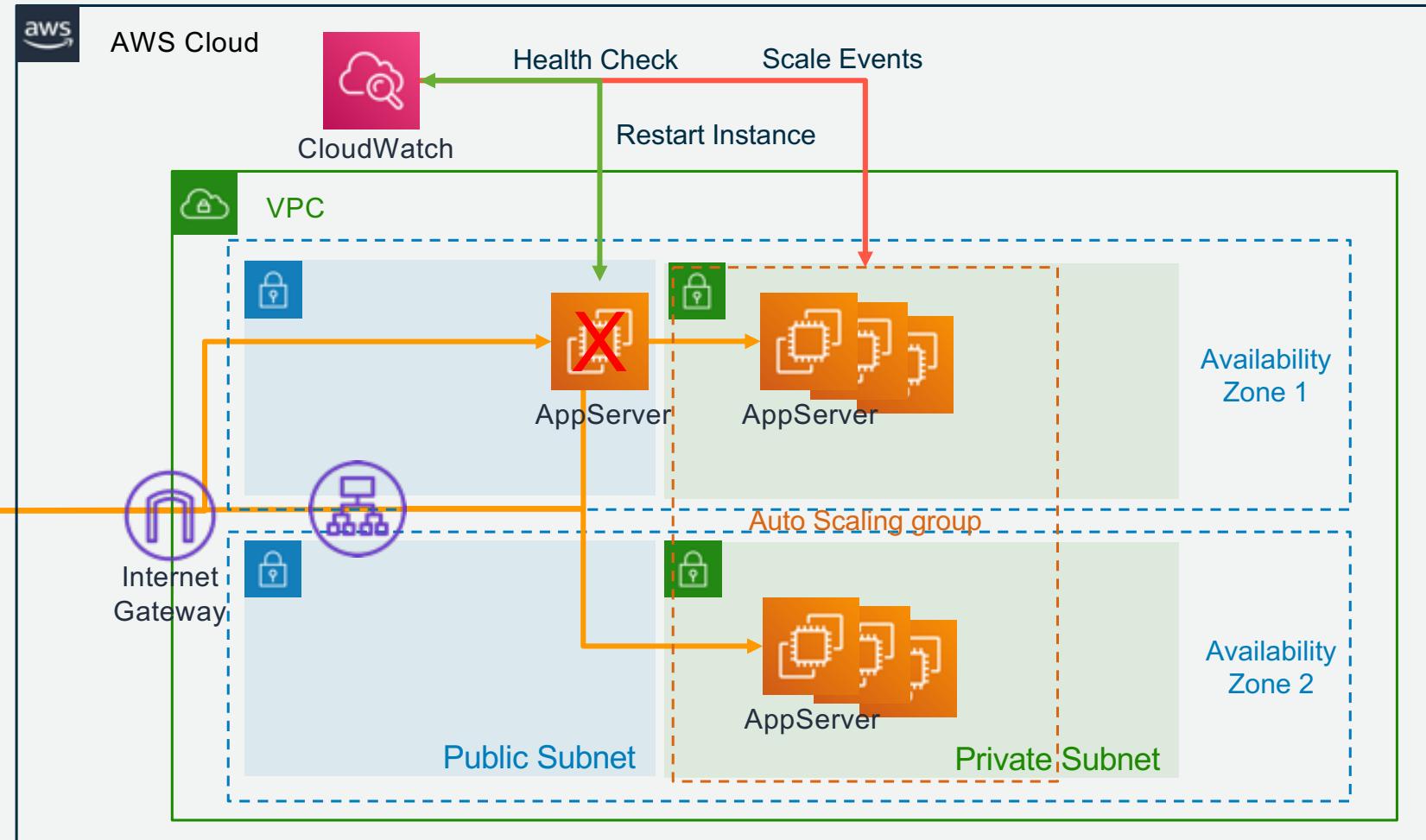


register
Task definition

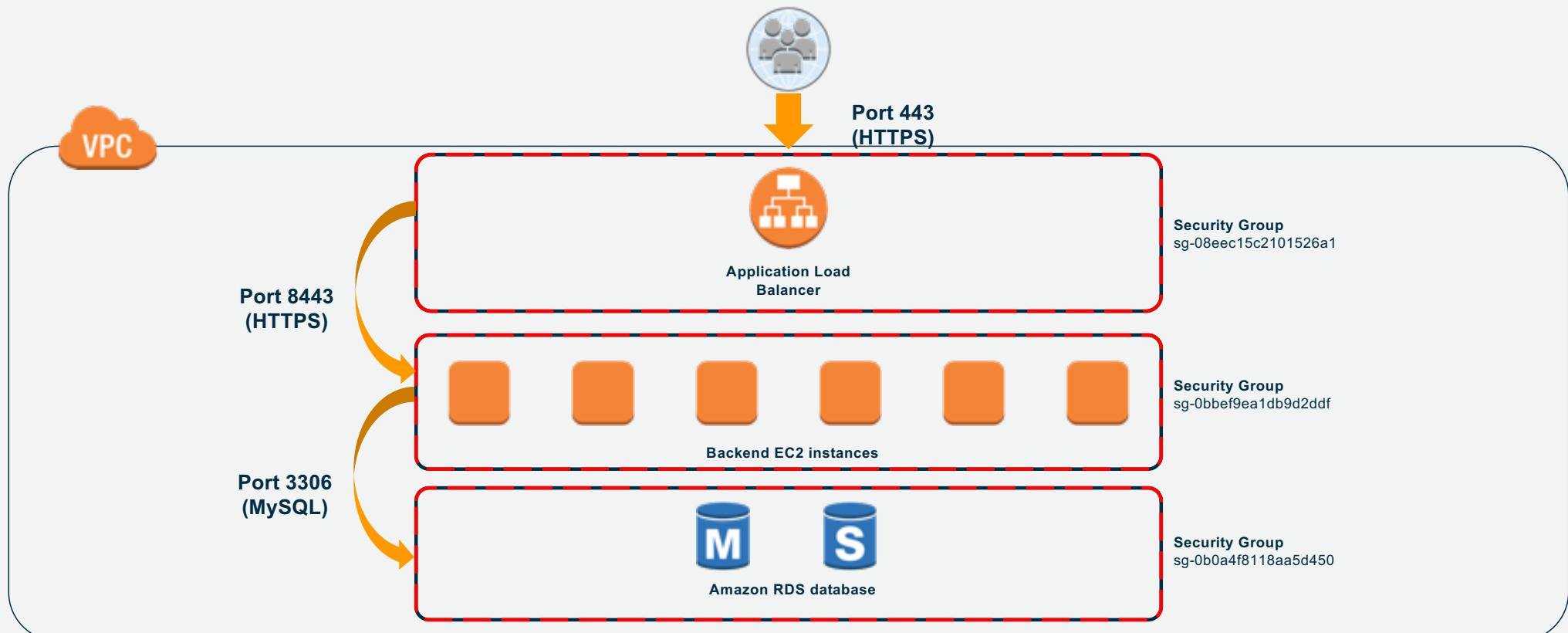
Define application containers:
Image, CPU, and memory
requirements, etc.



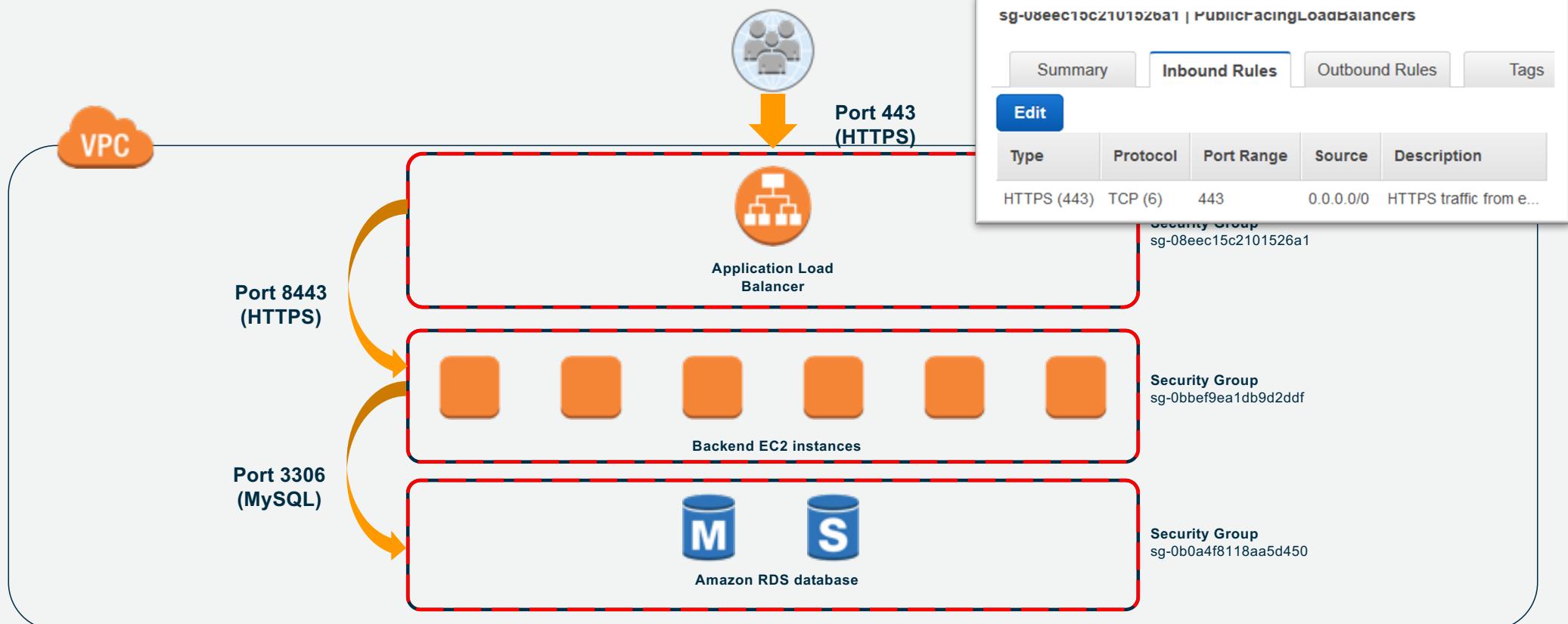
Grails App on AWS



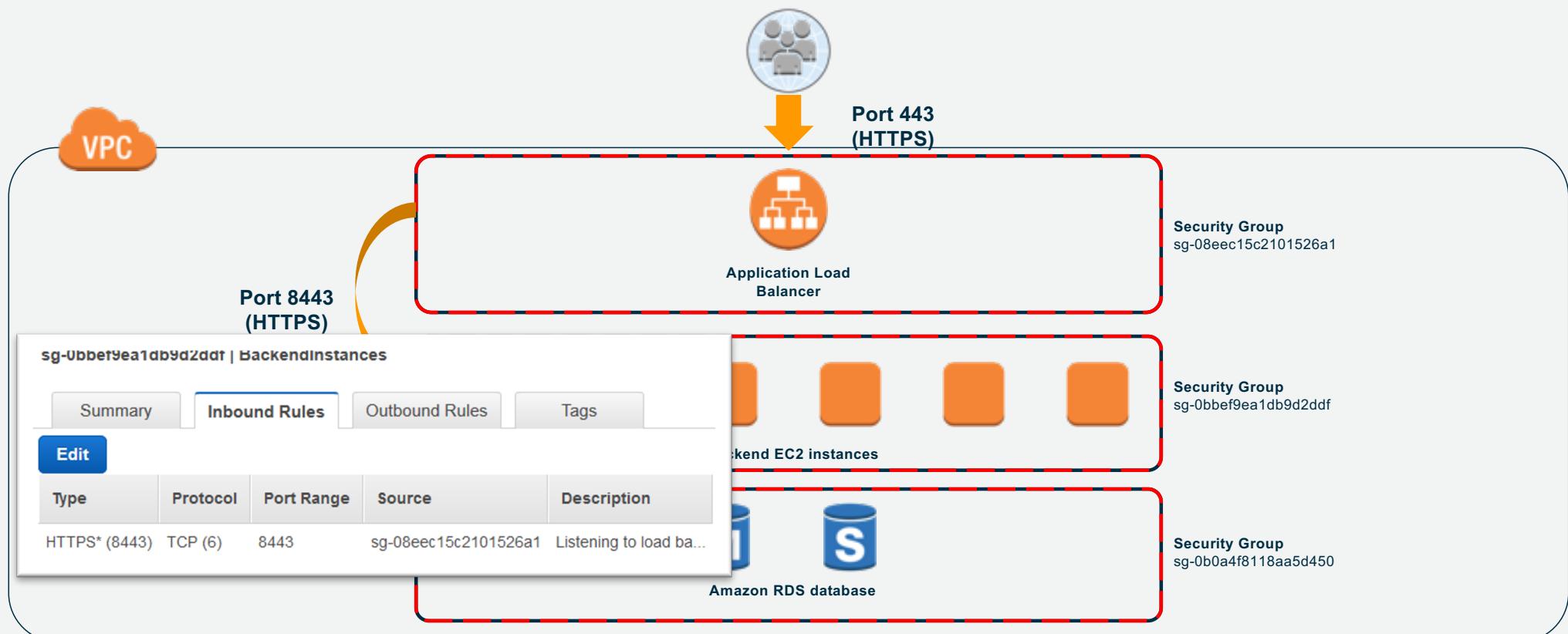
Security Groups: Stateful network firewalls



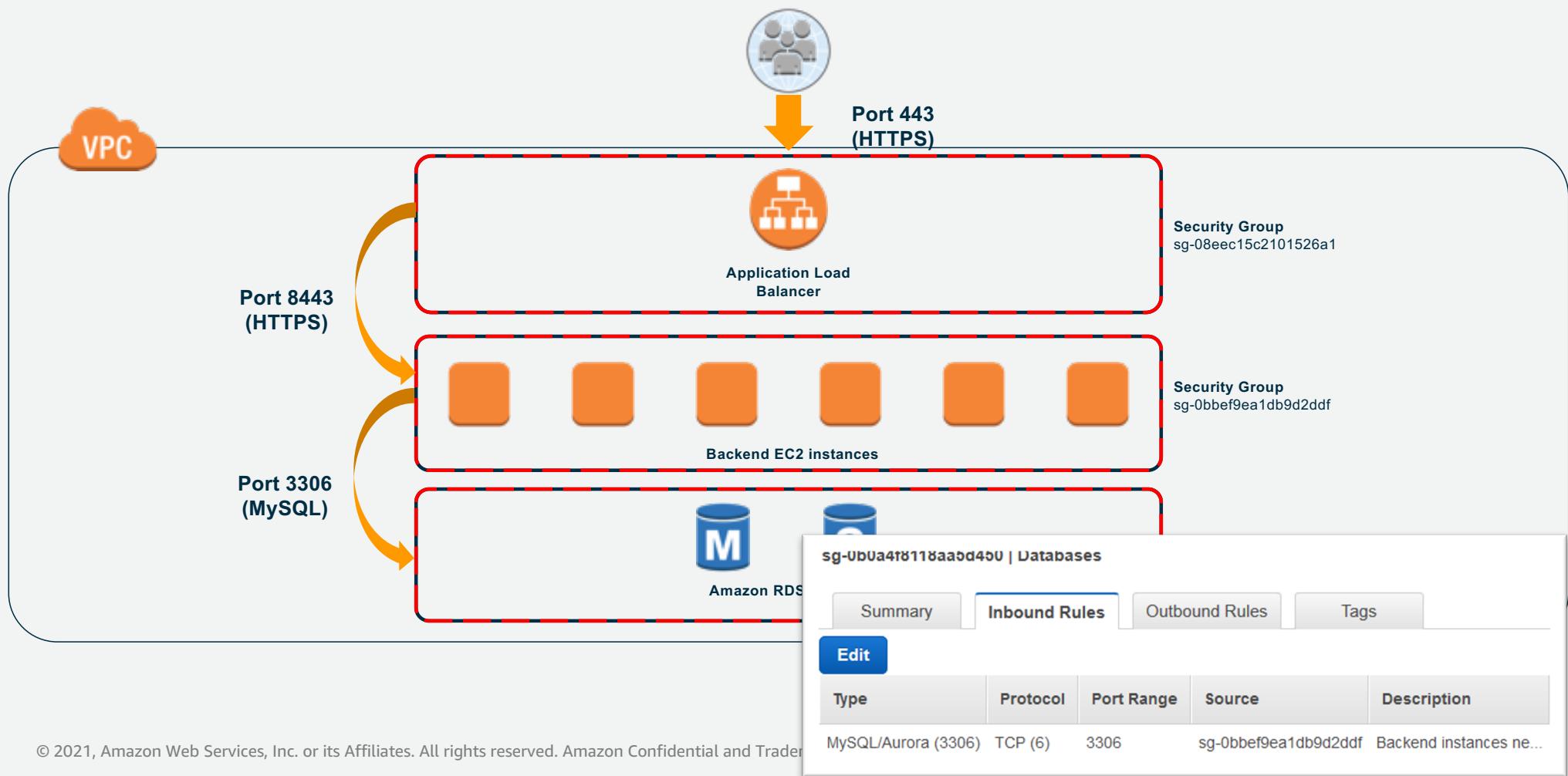
Security Groups: Stateful network firewalls



Security Groups: Stateful network firewalls



Security Groups: Stateful network firewalls

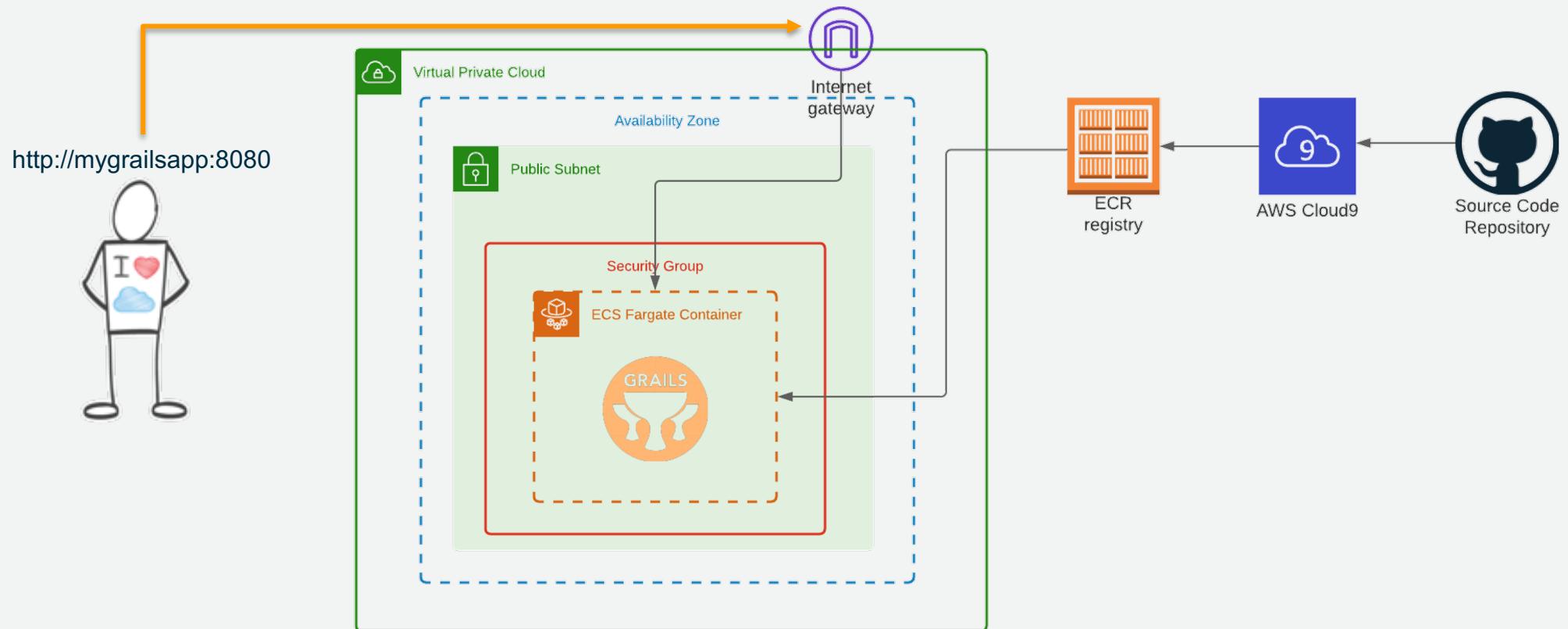


Hands-On Lab

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



Goal: Deploy your Grails Application in a Container on ECS

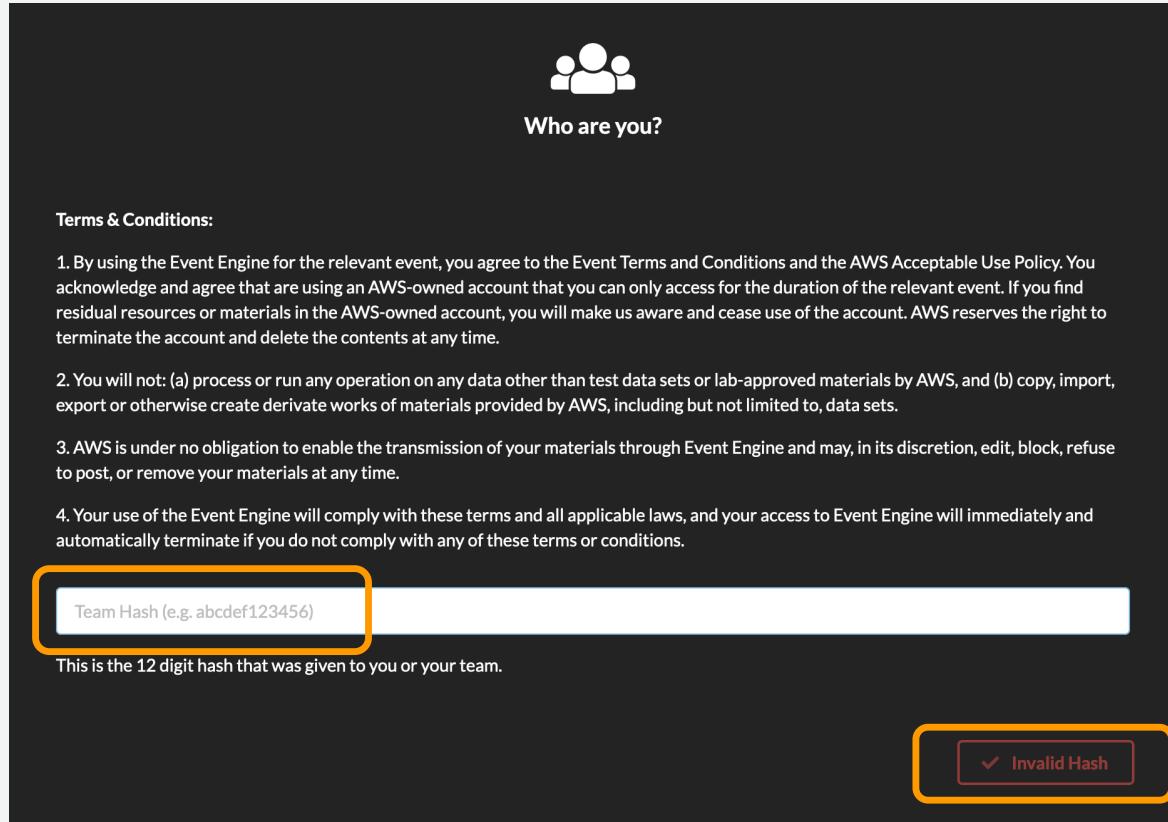


<https://github.com/cvolkmer/fhnw-web-deployment#lab-1>

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



Login to AWS Event Engine – <https://dashboard.eventengine.run>



Review Team Dashboard

Team Dashboard

 Event

 AWS Console  SSH Key

Event: FHNW Deployment workshop
Team Name: (Team Name Not Set Yet)

Event ID: 14
Team ID: ccc

Open AWS Console

AWS Console Login

Remember to only use "eu-central-1" as your region, unless otherwise directed by the event operator.

Login Link

 Open AWS Console  Copy Login Link

Credentials / CLI Snippets

Mac / Linux Windows

Mac or Linux 

```
export AWS_DEFAULT_REGION=eu-central-1
export AWS_ACCESS_KEY_ID=
export AWS_SECRET_ACCESS_
export AWS_SESSION_TOKEN=
```

How do I use the AWS CLI?

Checkout the AWS CLI documentation here: <https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html>

OK

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark

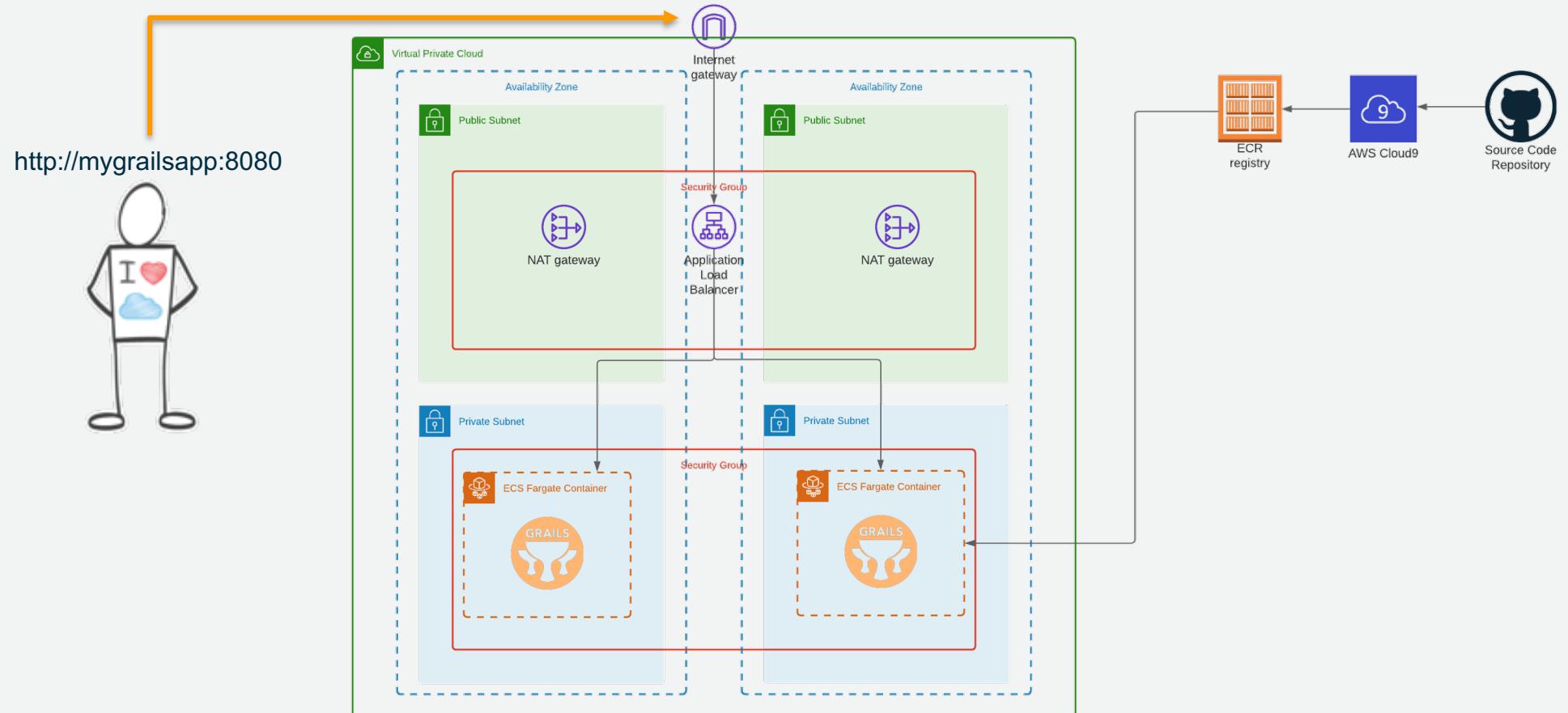


Hands-On Session / Tasks

1. Login to Event Engine
2. Open AWS Console
3. Deploy base infrastructure via CloudFormation Template
4. Explore Cloud9 and build / push Docker Image
5. Deploy Docker Image
6. [Optional]: Make your setup highly available behind a Load Balancer

Full Lab walkthrough:
<https://github.com/cvolkmer/fhnw-web-deployment>

Optional: Make your Grails Application highly available



<https://github.com/cvolkmer/fhnw-web-deployment#lab-2>

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



Develop and Deploy

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark



AWS has six families of developer tools

CI/CD



Infrastructure as Code



SDKs and CLIs



IDEs



Application Modernization

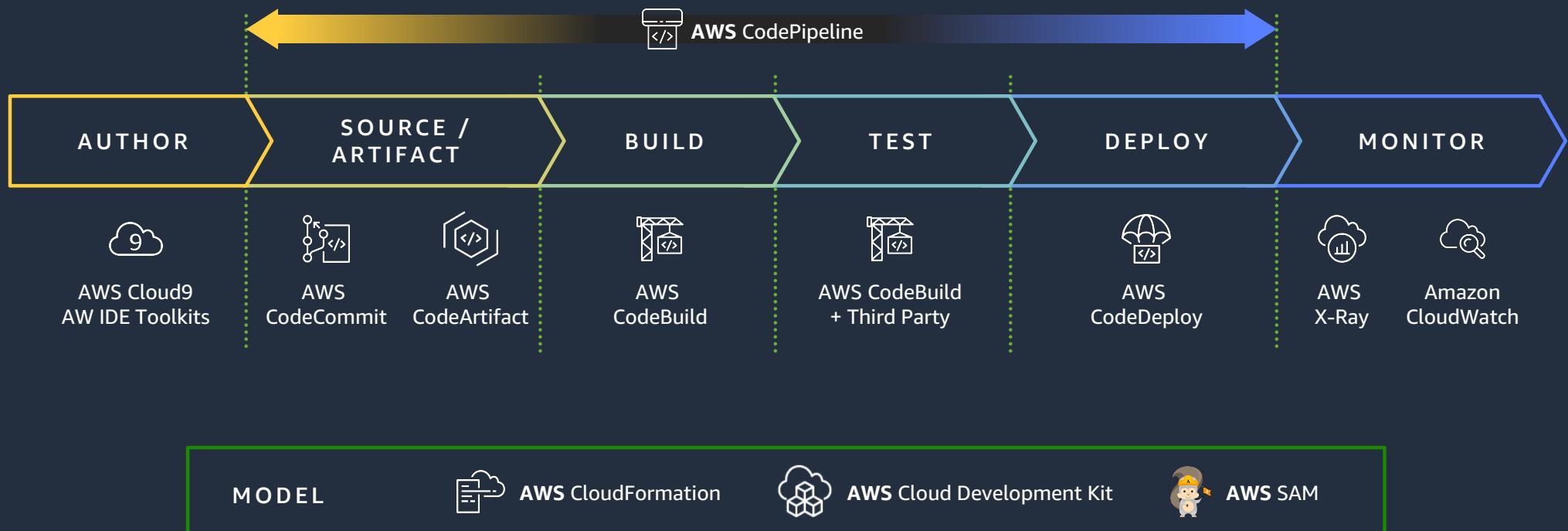


Next Generation DevOps



Development teams can find the right tool for the job

AWS Developer Tools for modern software delivery help automate your workflows with standardized tools



Guidance from Amazon's leading engineers on how to build modern cloud applications

The Amazon Builders' Library

How Amazon builds and operates software



Architecture, software delivery, and operations

By Amazon's senior technical executives and engineers

Real-world practices with detailed explanations

Content available for free on the website

Example Builders' Library content

ARCHITECTURE

LEVEL 300



Leader election in distributed systems

Author: Marc Brooker

Improving efficiency, reducing coordination, and simplifying architectures by using leader election.

^

🔗

SOFTWARE DELIVERY AND OPERATIONS

LEVEL 300



Going faster with continuous delivery

Author: Mark Mansour

Automating the software testing and deployment process for speed and reliability

^

🔗

SOFTWARE DELIVERY AND OPERATIONS

LEVEL 400



Implementing health checks

Author: David Yanacek

Automatically detecting and mitigating server failures without unintended consequences from fleet-wide false positives.

^

🔗

ARCHITECTURE

LEVEL 400



Workload isolation using shuffle-sharding

Author: Colm MacCarthaigh

Shuffle Sharding is one of our core techniques for drastically limiting the scope of impact of operational issues

^

🔗

Shorten the learning curve with AWS Cloud Development Kit (CDK)

Brings cloud infrastructure to developers in ways they can understand

- Build cloud infrastructure with the languages they already know
- Use their existing tools and workflows
- Helpful abstractions that remove the need to learn the details
- Vibrant and fast-growing community of developers



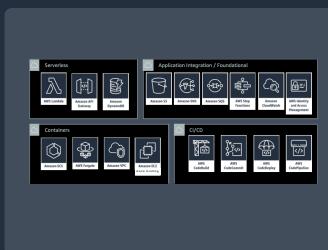
Coming soon

AWS CDK: Major Components

A multi-language software development framework for modeling cloud infrastructure as reusable components in popular programming languages

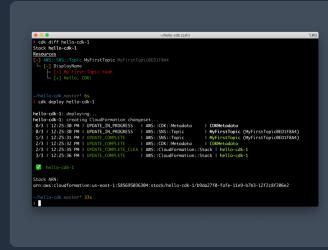
Core Framework

Model and share reusable infrastructure components that meet your organization's security, compliance, and governance requirements



Command Line Interface

Leverage the CDK CLI to interact with your CDK applications: synthesize a CFN template, show differences between the running stack and proposed changes, deploy multiple stacks across multiple environments, and more...

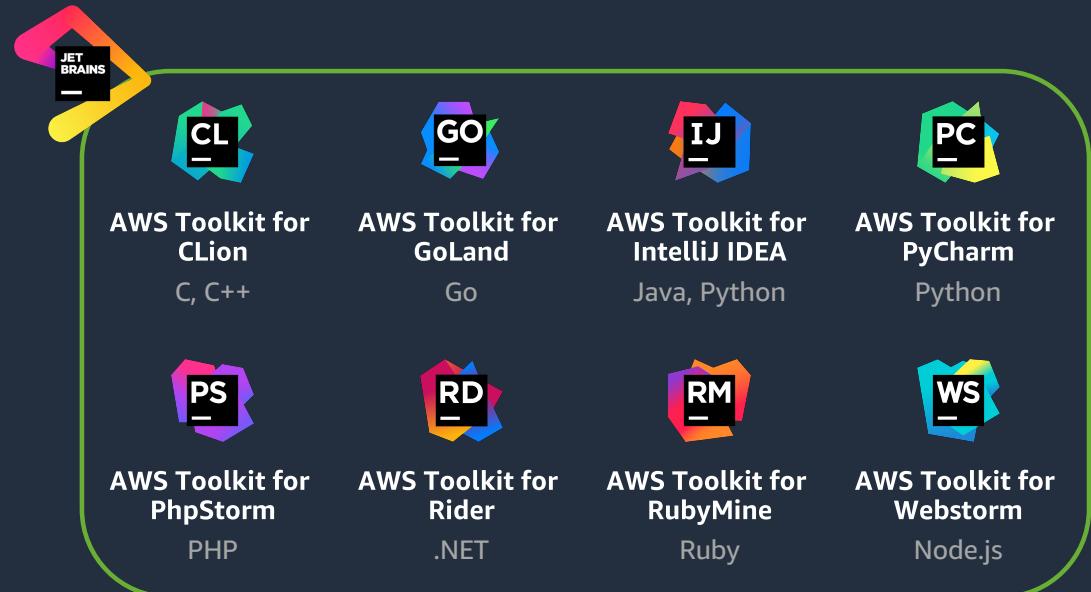


AWS Construct Library

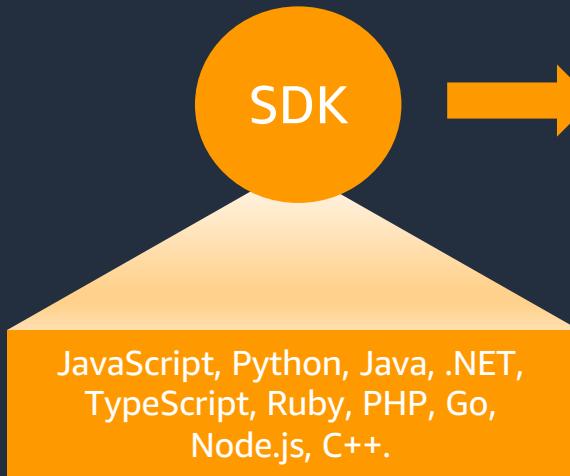
Utilize a pre-configured library of constructs or create your own customer parameters, enabling you to define your applications' infrastructure at a high level

Support for local development

Build, debug, and deploy applications on AWS with IDE Toolkits for popular desktop IDEs



AWS SDKs simplifies the use of AWS services



- **SDK Libraries:** Higher level abstractions saving developers time to concentrate on logic rather than low level API calls.
- Best practices by default (e.g., retries, credential handling)

AWS CodeStar Demo

© 2021, Amazon Web Services, Inc. or its Affiliates.



Helpful Resources

Getting started:

https://aws.amazon.com/getting-started/?nc1=h_ls

AWS Free Tier:

<https://aws.amazon.com/free/>

Online Training (partially free):

<https://www.aws.training/LearningLibrary>

Homegate Serverless Development:

https://youtu.be/bmd_rGbNVh8