

AWS Developer: Deployment and Security

DEPLOYING AND SECURITY ON AWS



Ryan Lewis

CLOUD ENGINEER

@ryanmurakami ryanlewis.dev

Better deployment

Less errors

Better security

Safer applications

TRAINING & CERTIFICATION

- AWS Certification >
- AWS Certified Cloud Practitioner >
- AWS Certified Solutions Architect - Associate >
- AWS Certified Solutions Architect - Professional >
- AWS Certified Developer - Associate** >
- AWS Certified SysOps Administrator - Associate >
- AWS Certified DevOps Engineer - Professional >
- AWS Certified Advanced Networking - Specialty >
- AWS Certified Big Data - Specialty >

AWS Certified Developer - Associate

The AWS Certified Developer – Associate exam validates technical expertise in developing and maintaining applications on the AWS platform. Exam concepts you should understand for this exam include:

- Picking the right AWS services for the application
- Leveraging AWS SDKs to interact with AWS services from your application
- Writing code that optimizes performance of AWS services used by your application
- Code-level application security (IAM roles, credentials, encryption, etc.)

Candidate Overview

Eligible candidates for this exam have:

- One or more years of hands-on experience designing and maintaining an AWS-based application
- In-depth knowledge of at least one high-level programming language
- Understanding of core AWS services, uses, and basic architecture best practices
- Proficiency in designing, developing, and deploying cloud-based solutions using

Schedule an exam and get started with
AWS Certification

[Go to aws.training](#)

[Download the exam guide »](#)

[Download sample questions »](#)

MORE RESOURCES

[Exam Preparation Resource Guide](#)

Find resources and tips to help you prepare for your AWS Certification.

AWS Certified Developer Path at Pluralsight

The screenshot shows the course page for "AWS Developer: Getting Started" by Ryan Lewis. The course is part of the AWS Certified Developer - Associate Path. It has a rating of 4.5 stars (267 reviews) and a duration of 4h 6m. The course info section indicates it is a Beginner level course. A large orange starburst graphic with the text "This Course!" is overlaid on the bottom right of the screenshot.

AWS Developer: Getting Started

The screenshot shows the course page for "AWS Developer: Designing and Developing" by Ryan Lewis. The course is part of the AWS Certified Developer - Associate Path. It has a rating of 4.5 stars (39 reviews) and a duration of 4h 20m. The course info section indicates it is an Intermediate level course.

AWS Developer: Designing and Developing

The screenshot shows the course page for "AWS Developer: Deployment and Security" by Ryan Lewis. The course is part of the AWS Certified Developer - Associate Path. It has a rating of 4.5 stars (22 reviews) and a duration of 1h 36m. The course info section indicates it is an Intermediate level course.

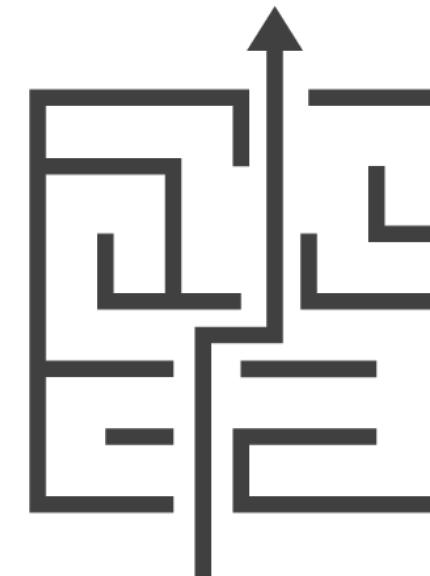
AWS Developer: Deployment and Security

The screenshot shows the course page for "AWS Developer: Serverless Architecture and Monitoring" by Ryan Lewis. The course is part of the AWS Certified Developer - Associate Path. It has a rating of 4.5 stars (22 reviews) and a duration of 1h 36m. The course info section indicates it is an Intermediate level course.

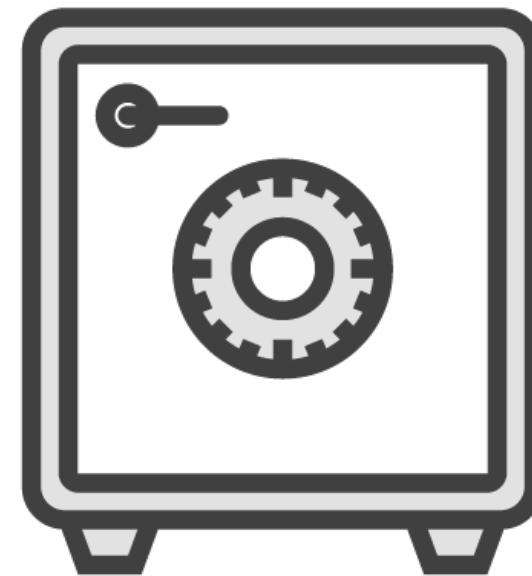
AWS Developer: Serverless Architecture and Monitoring

Module Overview

Deploying and Security on AWS



Deployment
Strategies



Security
Strategies

Module Overview

Deploying Applications to AWS

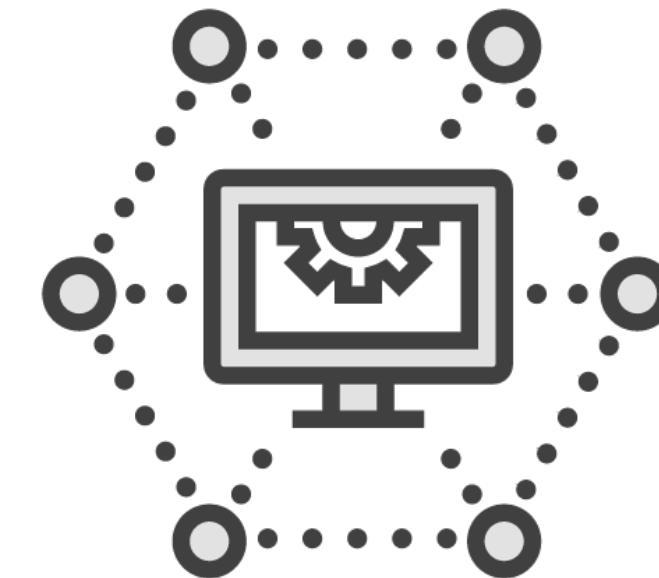


Resource Creation
with CloudFormation

App Deployment
with Elastic Beanstalk

Module Overview

Coordinating Services in AWS



Service Coordination
with Simple Workflow

Module Overview

Securing Infrastructure on AWS



Infrastructure Security
with Virtual Private Cloud



Account Logging
with CloudTrail

Module Overview

Managing Access to AWS



User Access Management
with Identity and Access Management (IAM)

AWS Developer: Getting Started

Services include:

CloudFormation

Elastic Beanstalk

VPC

IAM

- For beginners to AWS
- Aims for familiarity with Services

AWS Developer: Deployment and Security

Services include:

CloudFormation

Elastic Beanstalk

Simple Workflow

VPC

CloudTrail

IAM

- For regular users of AWS
- Deep dive and best practices

Let's get started!

Deployment Strategies for AWS

EC2 Deployment Strategies

Manual
Upload

Source
Control

Elastic
Beanstalk

Cloud
Formation

AMI

S3

Previously on AWS Developer...

Deploying
with scp

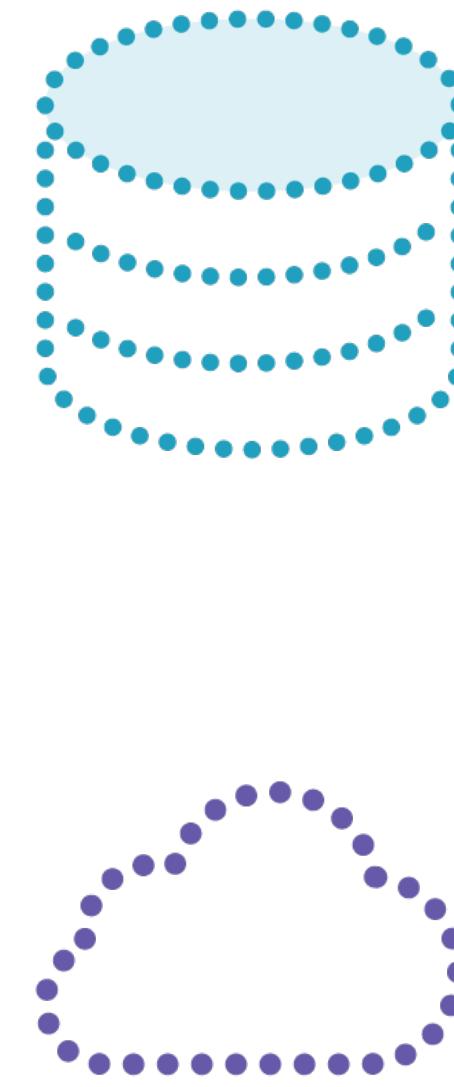
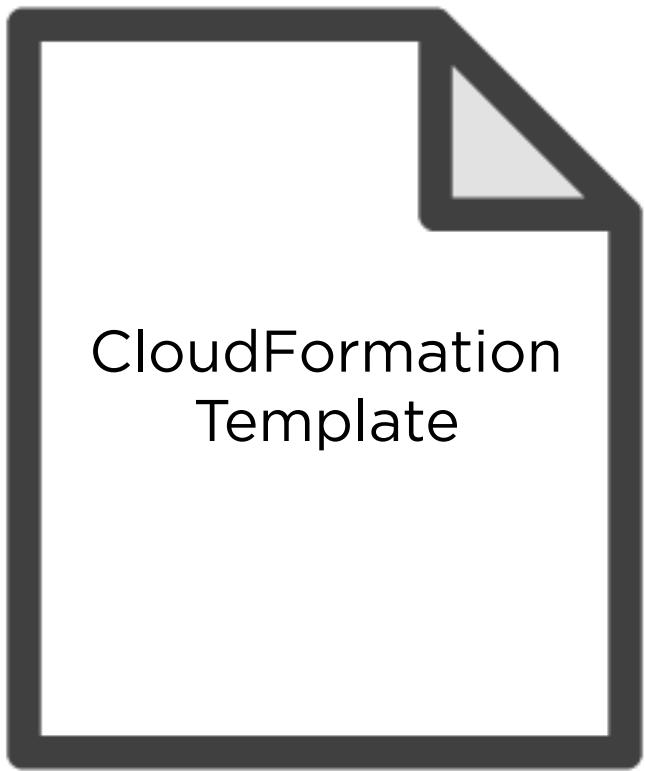
Deploying
with git

Deploying
with a whole
bunch of code

Deployment Method #1

CloudFormation

How CloudFormation Works

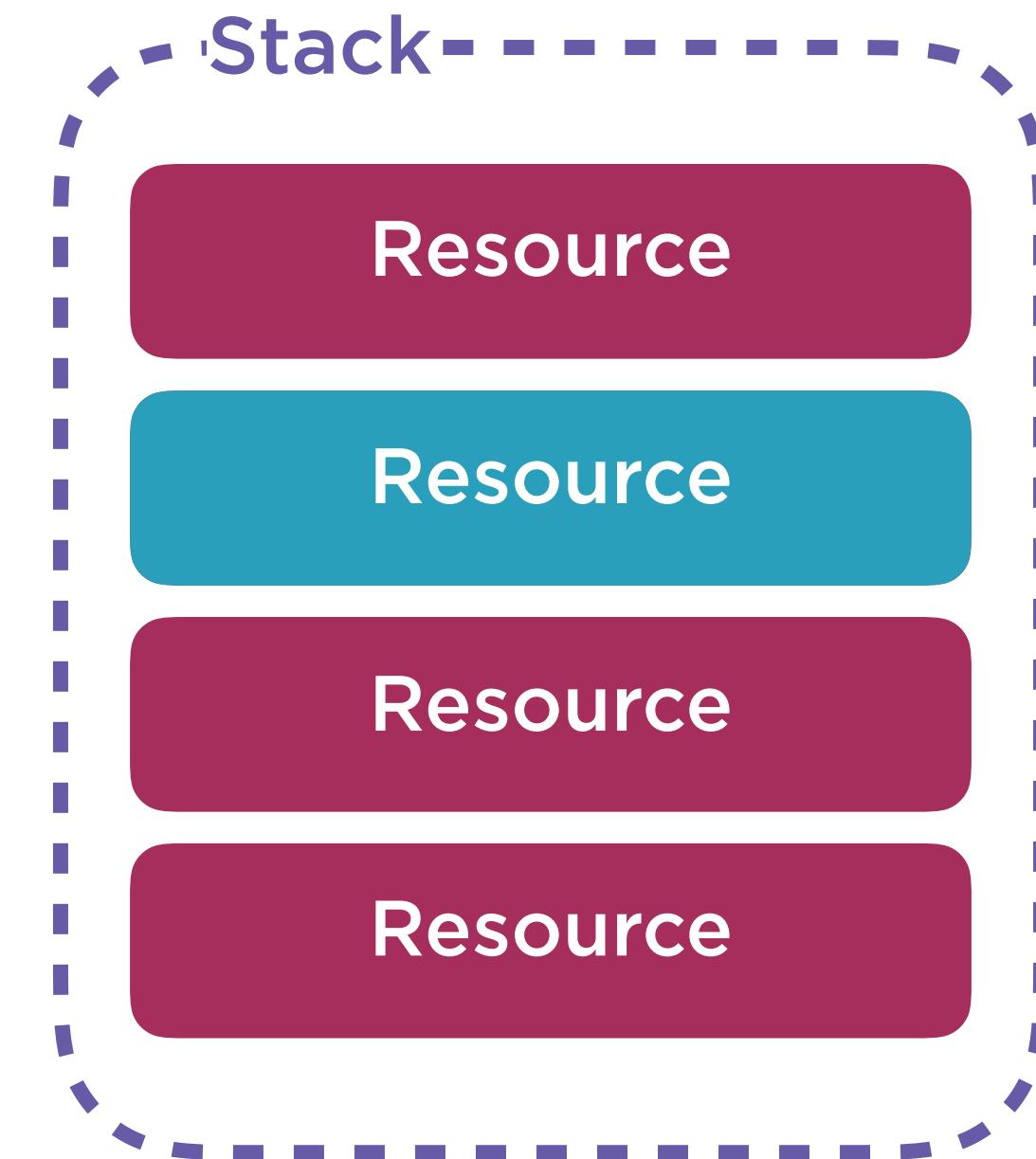


What Can You Create with CloudFormation?

EC2 Instance
DynamoDB Table
RDS Database
VPC
Subnet
IAM User
Elasticache Cluster
S3 Bucket
IAM Role
CloudWatch Log Group
SNS Topic
SQS Queue
Elastic Beanstalk App

Elastic Load Balancer
Security Group
IAM Group
CloudFront Distribution
Route 53 Hosted Zone
API Gateway
Lambda Function
CloudWatch Alarm
SNS Subscription
IAM Profile
Route 53 Domain
Auto Scaling Group
And Much, Much More

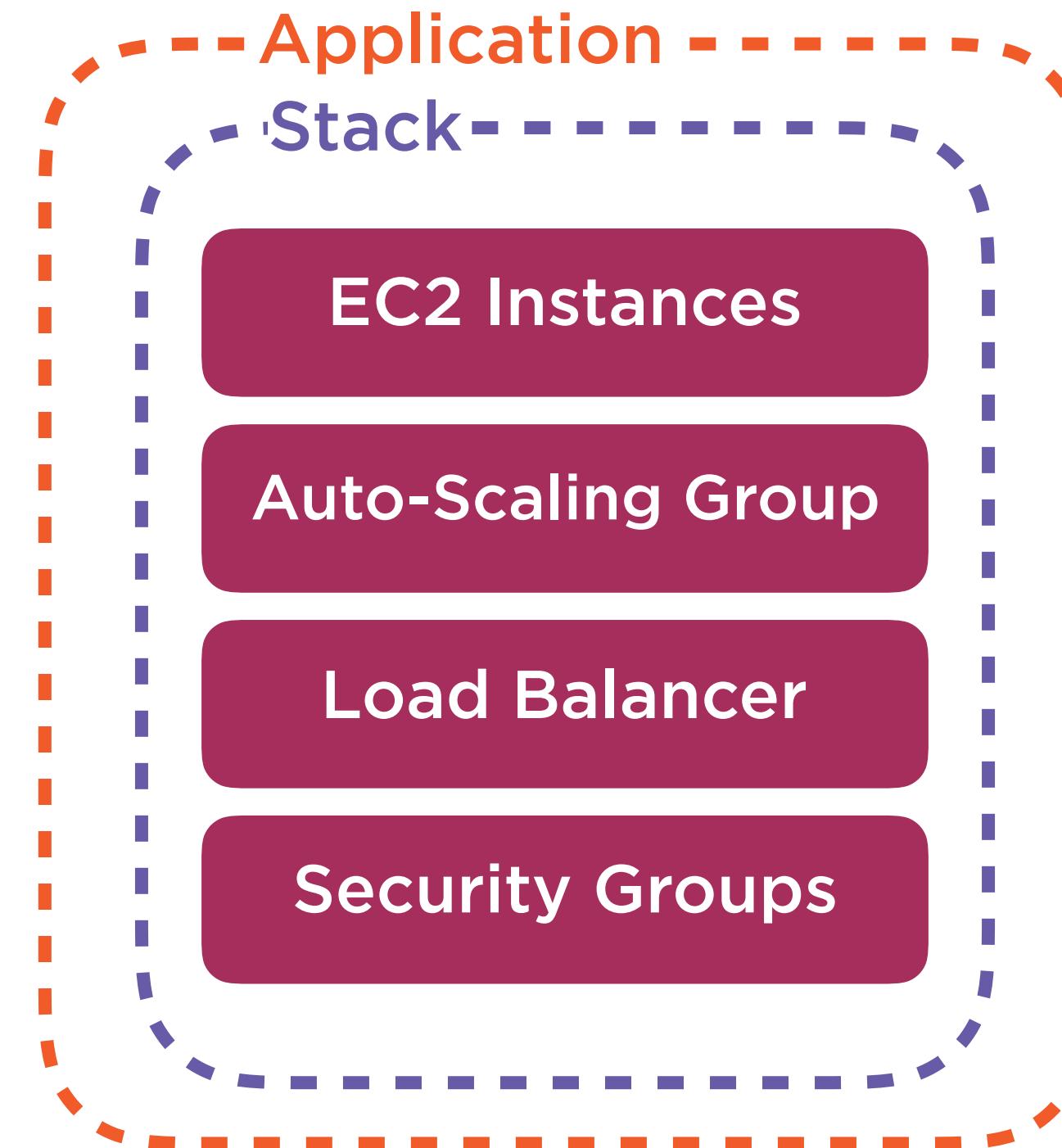
CloudFormation Stack



Deployment Method #2

Elastic Beanstalk

Elastic Beanstalk Application



Elastic Beanstalk works best
with one application

What about
containers?

Deployment Method #3

Elastic Container Service

*Not covered in this course

Pluralsight Courses on Elastic Container Service

Docker in Production Using AWS
by Justin Menga

Using Docker on AWS
by David Clinton

Using Docker with AWS Elastic Beanstalk
by David Clinton

What Does Security in AWS Mean?

Developer



Security = Safe Code

AWS



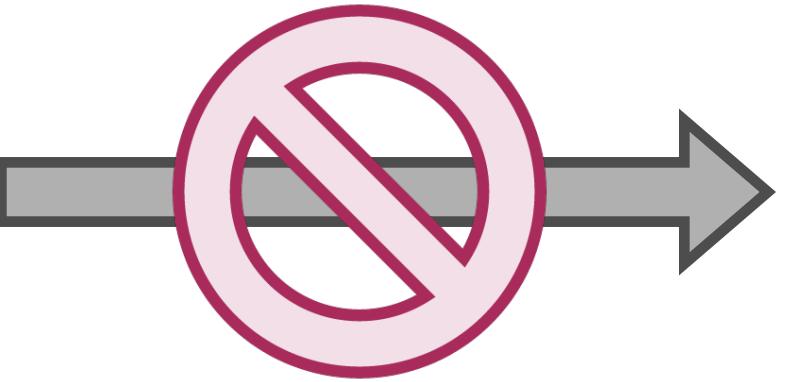
Security = ?

AWS Security is...

...not your application code

AWS Security is...

...managing access to resources



AWS
Resource

AWS Security is...

...patching software vulnerabilities

**Operating System
Version X.X**

EC2 Instance

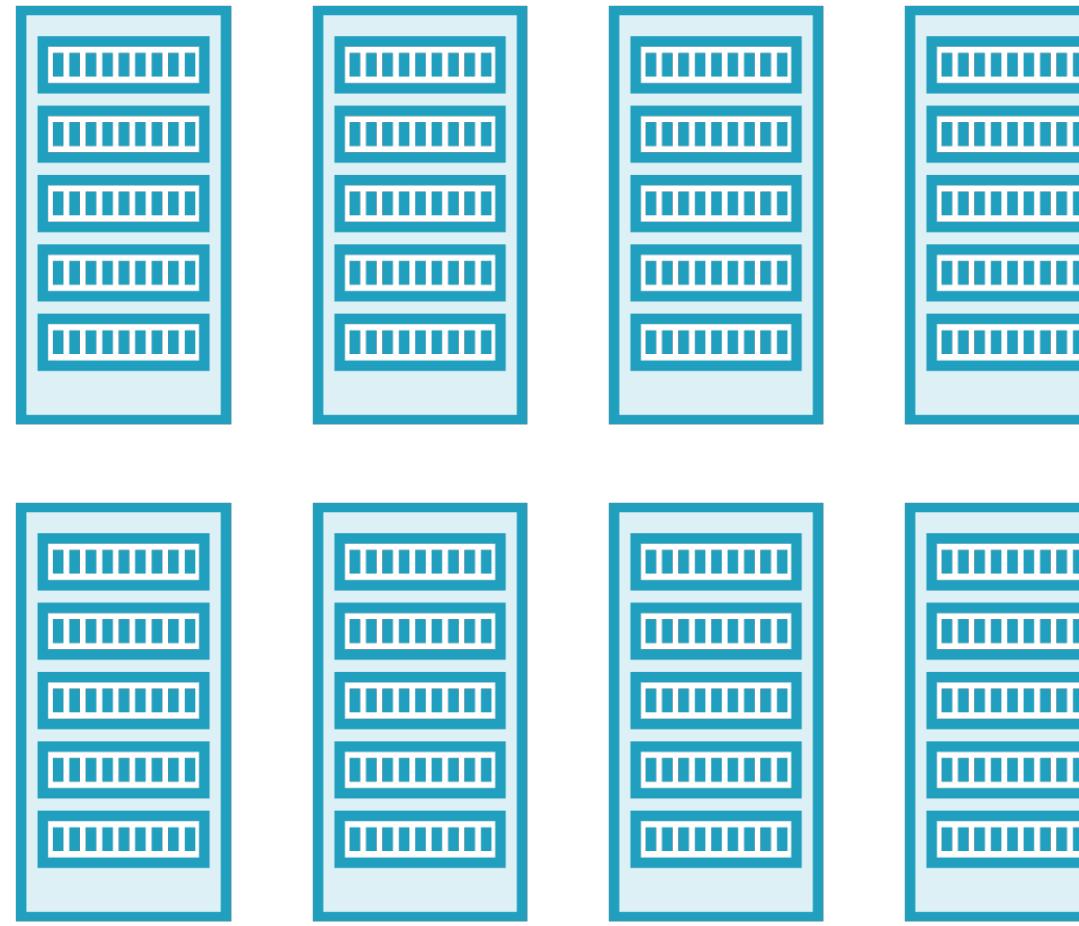
**DB Engine
Version X.X**

RDS Cluster

AWS Security is...

...configuring permissions

AWS Account



AWS
Security Tools + You = Awesome



The AWS Shared Security Model

AWS Shared Security Model

AWS is responsible for part of the security of your account. You are responsible for the rest, with help from tools and services provided by AWS.

Three Types of Services in AWS

?

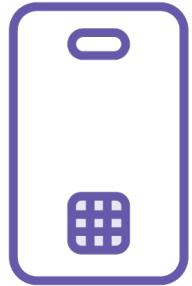
?

?

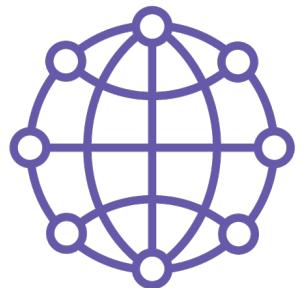
AWS Basic Security Responsibilities



Physical facilities maintenance



Physical facilities security



Network infrastructure



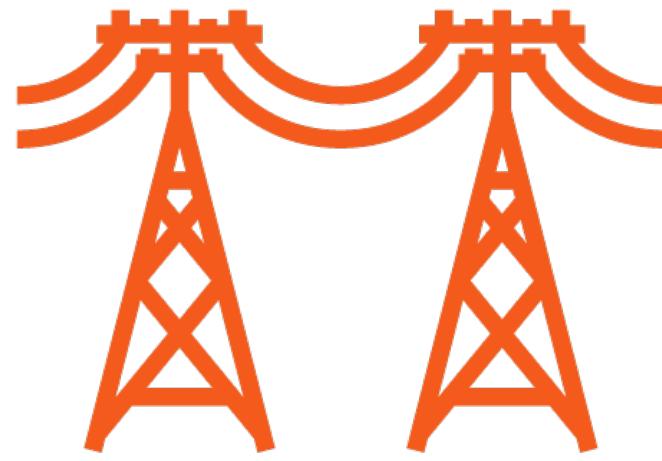
Virtualization infrastructure

Three Types of Services in AWS

Infrastructure
Services

Container
Services

Abstracted
Services



Infrastructure Services

Basic services

Customer responsible for more security

Example services:

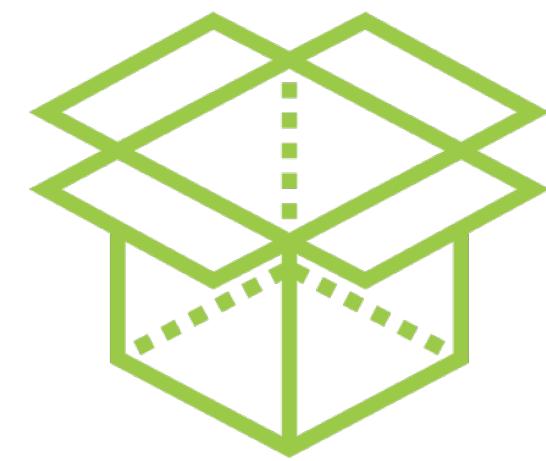
- Elastic Cloud Compute (EC2)
- VPC
- Elastic Block Storage (EBS)

Managed services

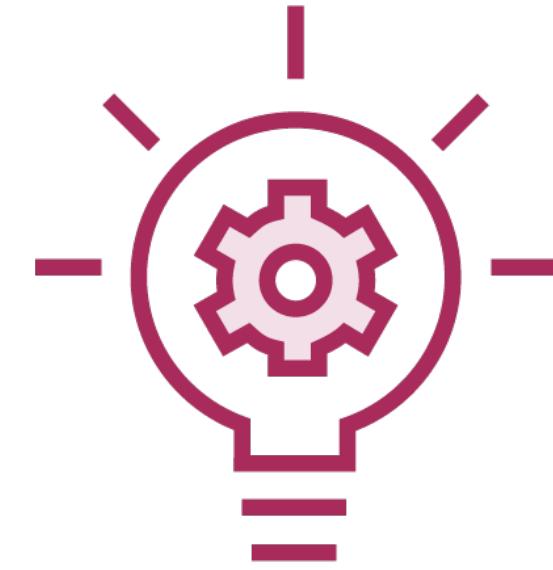
AWS handles more security responsibilities

Example services:

- RDS
- Elastic Beanstalk



**Container
Services**



Abstracted
Services

Near-zero configuration services
AWS owns nearly all security responsibilities

Example services:

- DynamoDB
- Simple Queue Service (SQS)

You have the most control over security

AWS handles basic security responsibilities

User manages:

Software (OS and more)

Data (at rest, incoming, outgoing)

Credentials and permissions

Infrastructure Services

Tools to manage security

VPC

Network ACLs

Security Groups

IAM

Provided software

Amazon
Machine Images
(AMI)

AWS handles basic security responsibilities

AWS handles OS and software and updates

User manages:

Access

AWS handles (basically) all security responsibilities

AWS handles server-side data encryption

User manages:

IAM permissions

AWS Security by Service Type

Infrastructure Services

User responsible
for most
security

Container Services

AWS
responsible for
more security

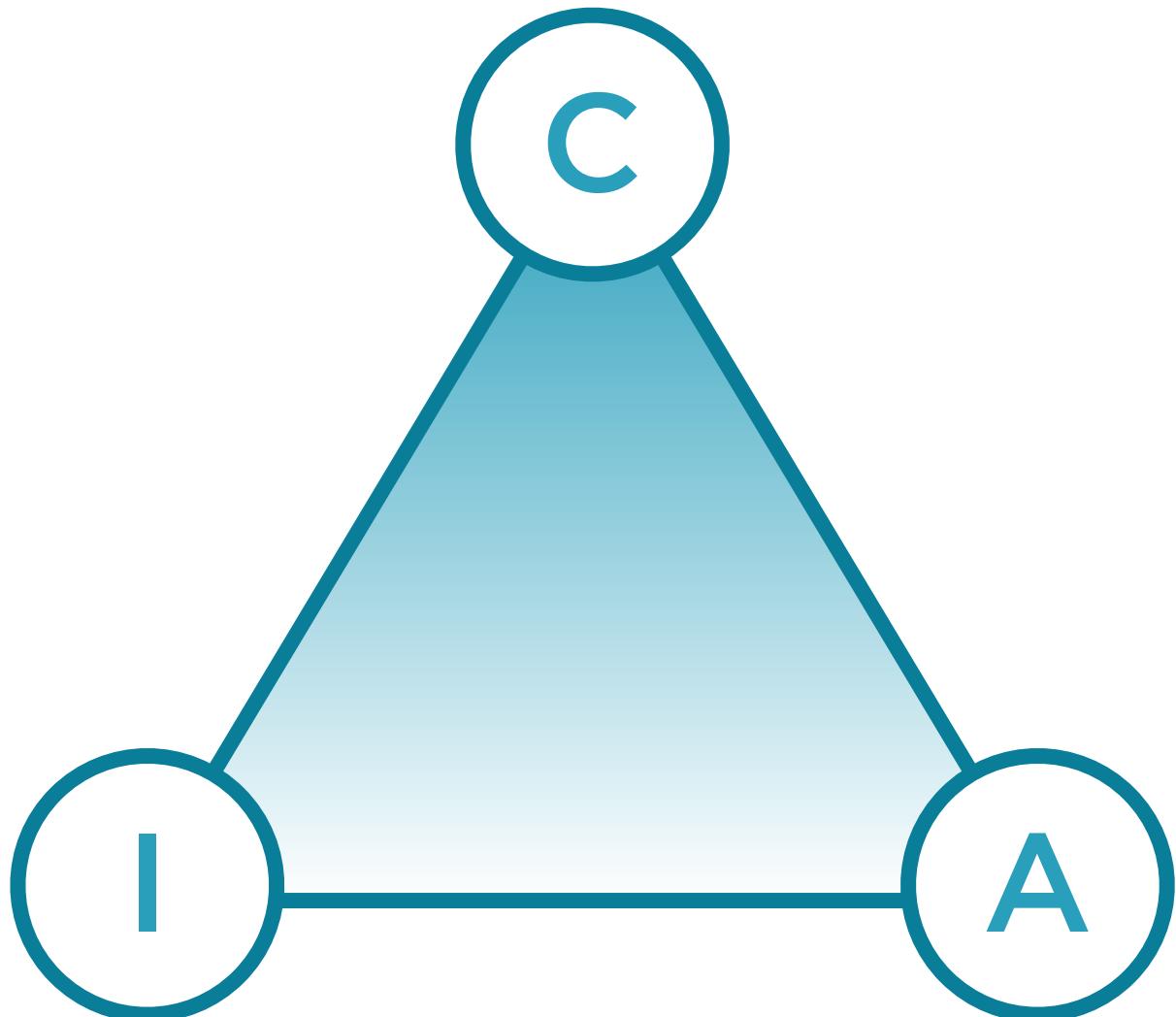
Abstracted Services

AWS
responsible for
all security

Essential Security Concepts in AWS

AWS wants you to be
thinking about security

CIA Triad

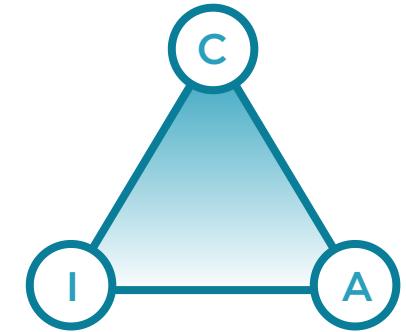


Confidentiality

Integrity

Availability

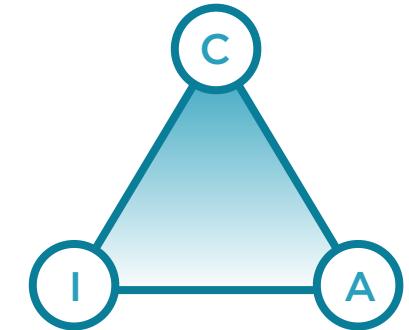
CIA Triad



Confidentiality

Managing access to data; ensuring that bad actors do not get access to it.

CIA Triad



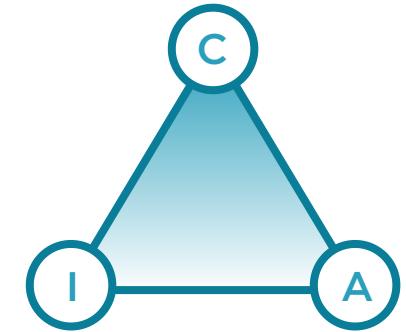
Managing Confidentiality

IAM Roles

Network ACLs

Security Groups

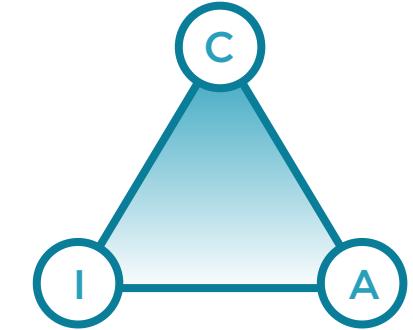
CIA Triad



Integrity

Keeping data safe from modification and deletion.

CIA Triad

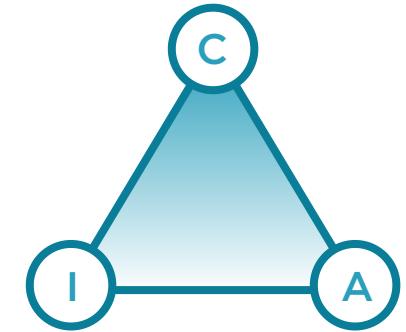


Managing Integrity

Keep backups

Verify against hashes

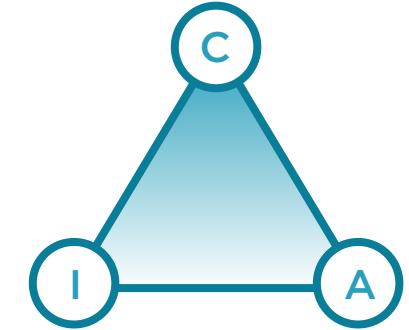
CIA Triad



Availability

Ensuring data is available to authorized users.

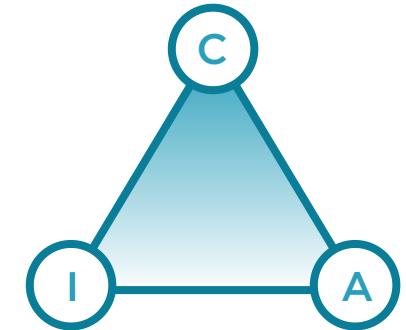
CIA Triad



Managing Availability

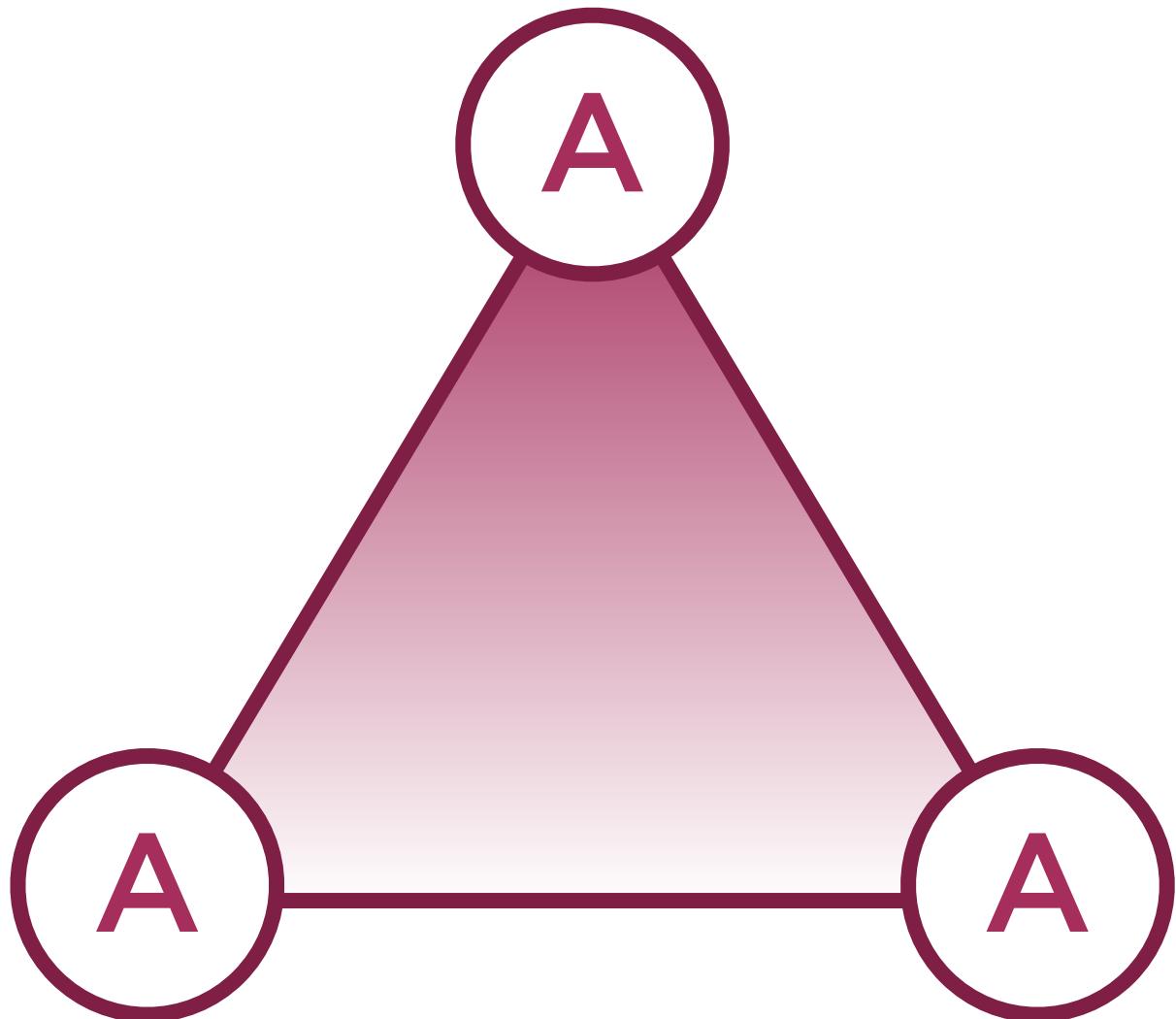
Keep backups across AZs

CIA Triad



CIA is your responsibility for infrastructure services and container services

AAA

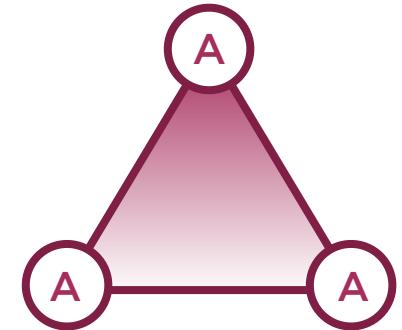


Authentication

Authorization

Accounting

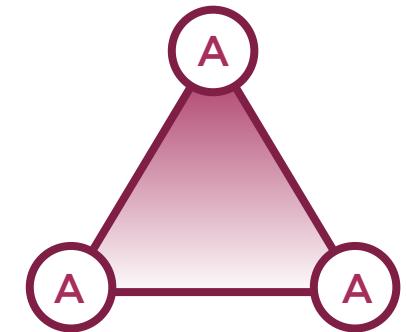
AAA



Authentication

Determining if a user is who they say they are.

AAA

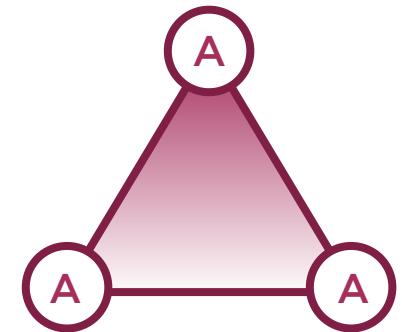


Managing Authentication

Username/Password

Access key

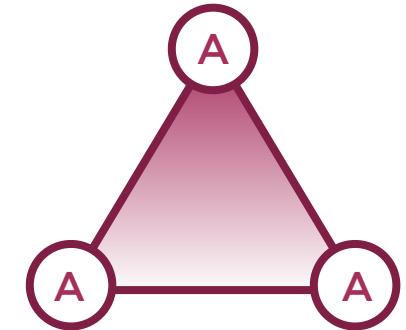
AAA



Authorization

Determining if the user can do what they're asking to do.

AAA

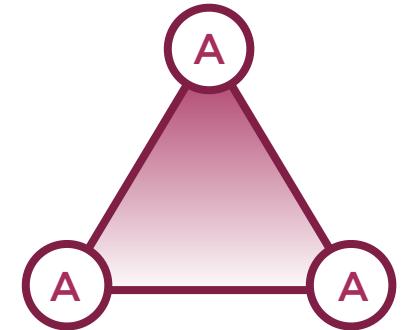


Managing Authorization

IAM User

IAM Groups

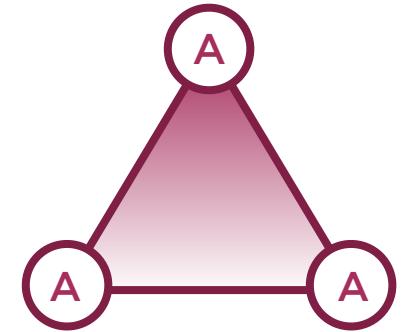
AAA



Accounting

Logging actions taken by users.

AAA

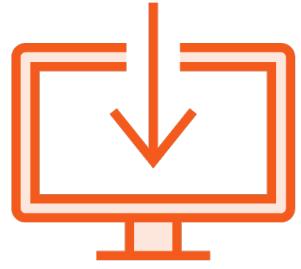


Managing Accounting

CloudTrail

What You Need for This Course

AWS CLI



Download and install AWS CLI



Configure with a user's access key and secret



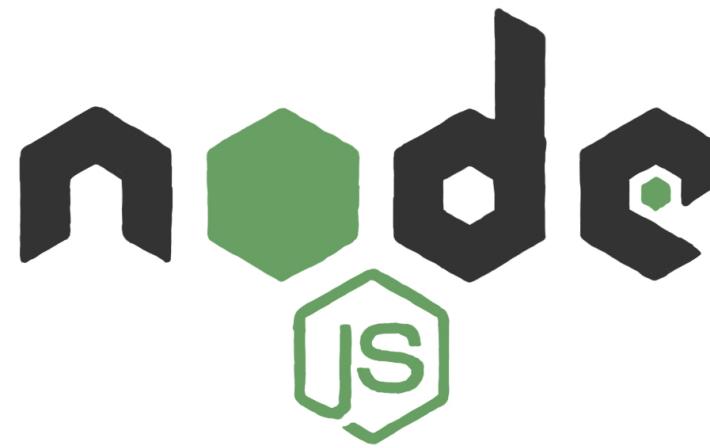
Ensure permissions for all services used in course

Local AWS credentials are located at:

OSX & Linux - `~/.aws/credentials`

Windows - `C:\Users\<you>\.aws\credentials`

Node.js Platform



Download at nodejs.org

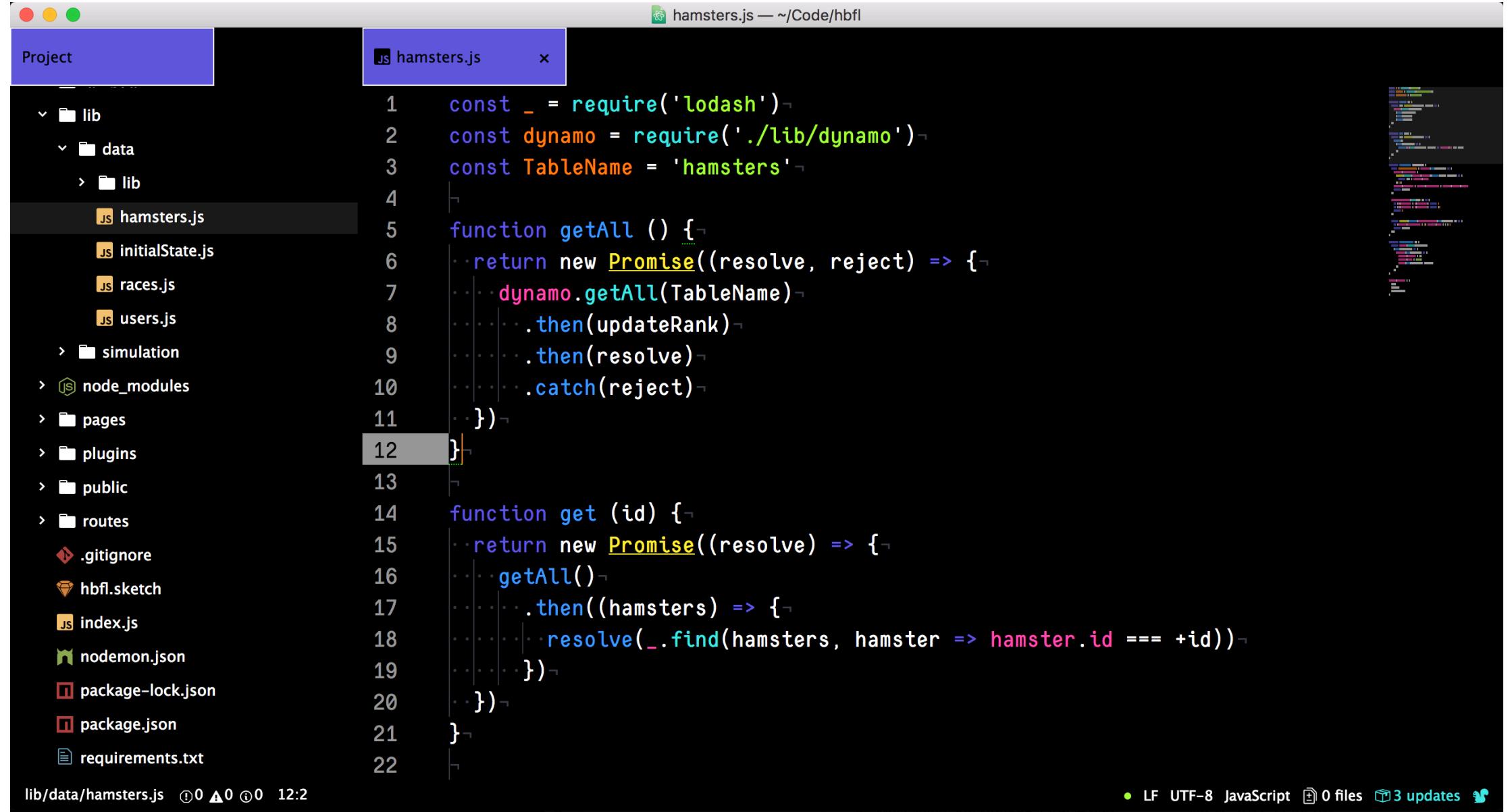
Version 12+

Code Editor

Try Atom!

Download at
atom.io

Try out
my syntax theme:
no-compromise



The screenshot shows the Atom code editor interface. The left sidebar is titled 'Project' and lists the project structure. The main editor tab is titled 'hamsters.js' and contains the following JavaScript code:

```
1  const _ = require('lodash')
2  const dynamo = require('./lib/dynamo')
3  const TableName = 'hamsters'
4
5  function getAll () {
6    return new Promise((resolve, reject) => {
7      dynamo.getAll(TableName)
8        .then(updateRank)
9        .then(resolve)
10       .catch(reject)
11    })
12  }
13
14  function get (id) {
15    return new Promise((resolve) => {
16      getAll()
17        .then((hamsters) => {
18          resolve(_.find(hamsters, hamster => hamster.id === +id))
19        })
20    })
21  }
22
```

At the bottom of the editor, it shows 'lib/data/hamsters.js ①0 ▲0 ①0 12:2'. On the right side of the editor, there are several floating panels for file navigation, search, and other editor settings.

AWS CLI

Node.js

Code Editor

Command Line

Taking Hamster Security Seriously

Module Demo

Deploying Applications to AWS

Deploy to EC2 with CloudFormation

Deploy with Elastic Beanstalk

Module Demo

Coordinating Services in AWS

Build a race results processor with Simple Workflow

Module Demo

Securing Infrastructure on AWS

Deploy to EC2 inside a secure VPC

Module Demo

Managing Access to AWS

No application!

No JS? No Problem

Conclusion

Summary

The evolution of deployment

What AWS thinks about security

Generously sharing security with you

CIA or AAA? Why not both!

Get ready for JSON overload

Who could say no to hamsters?

Up Next

Application Deployment

with CloudFormation
and Elastic Beanstalk