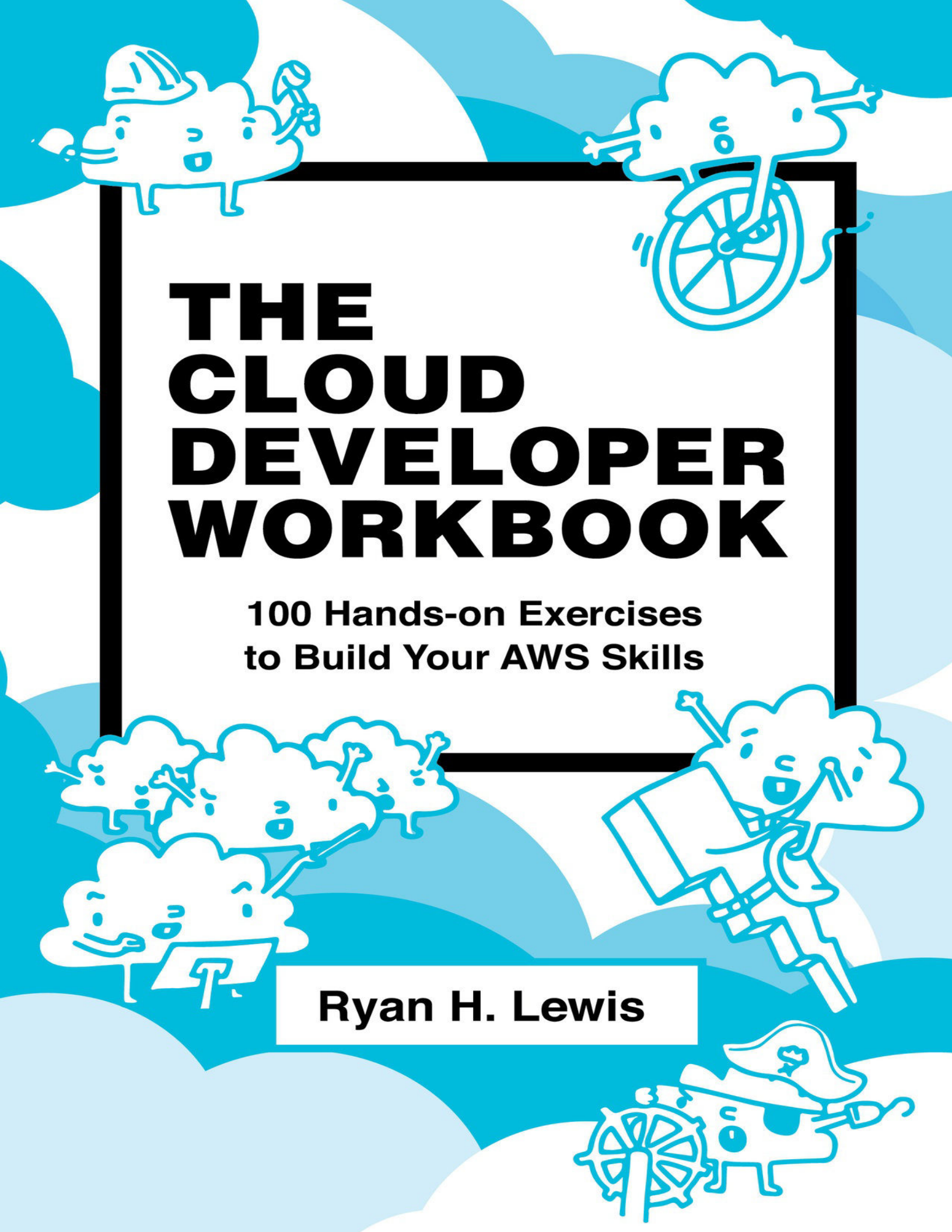
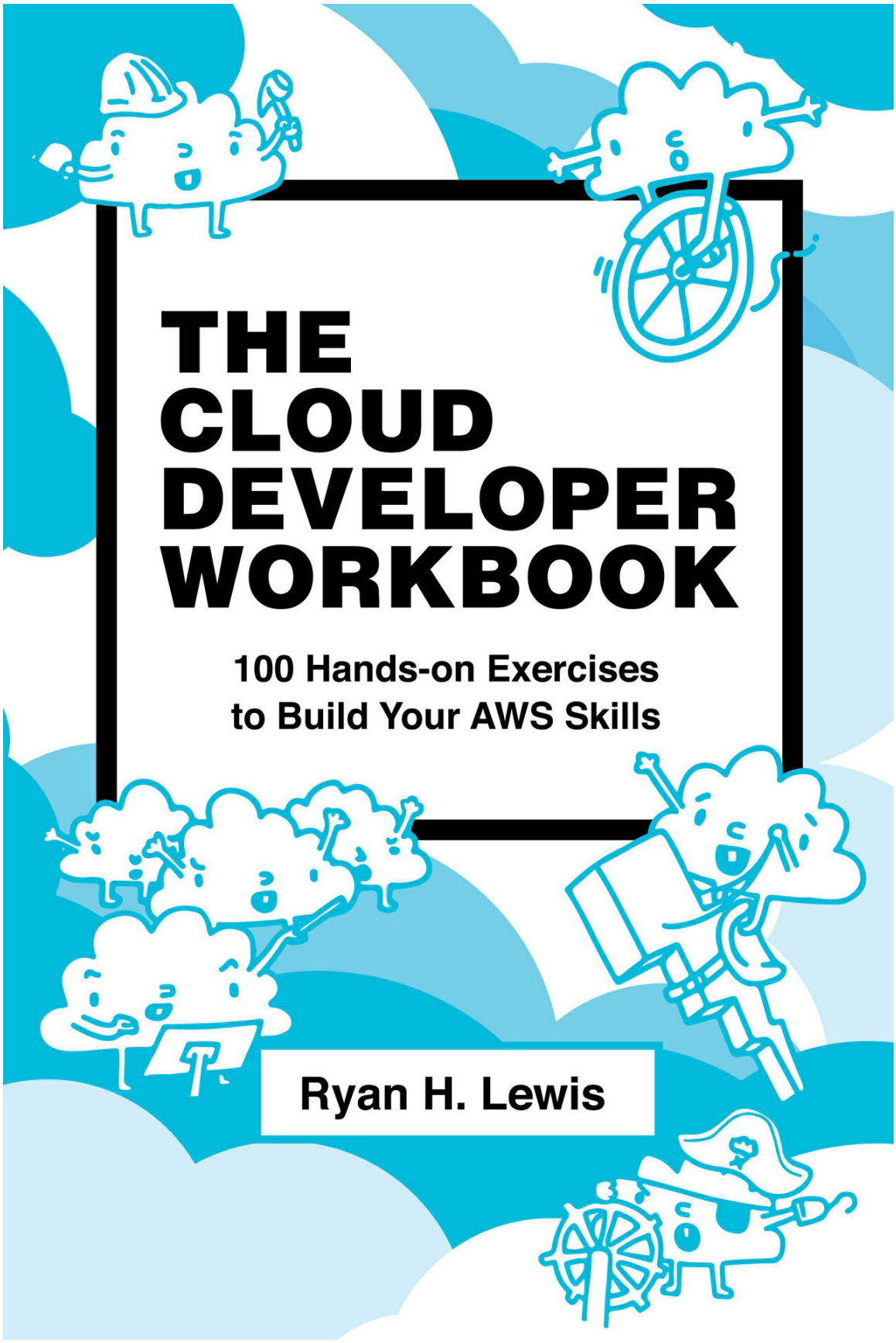
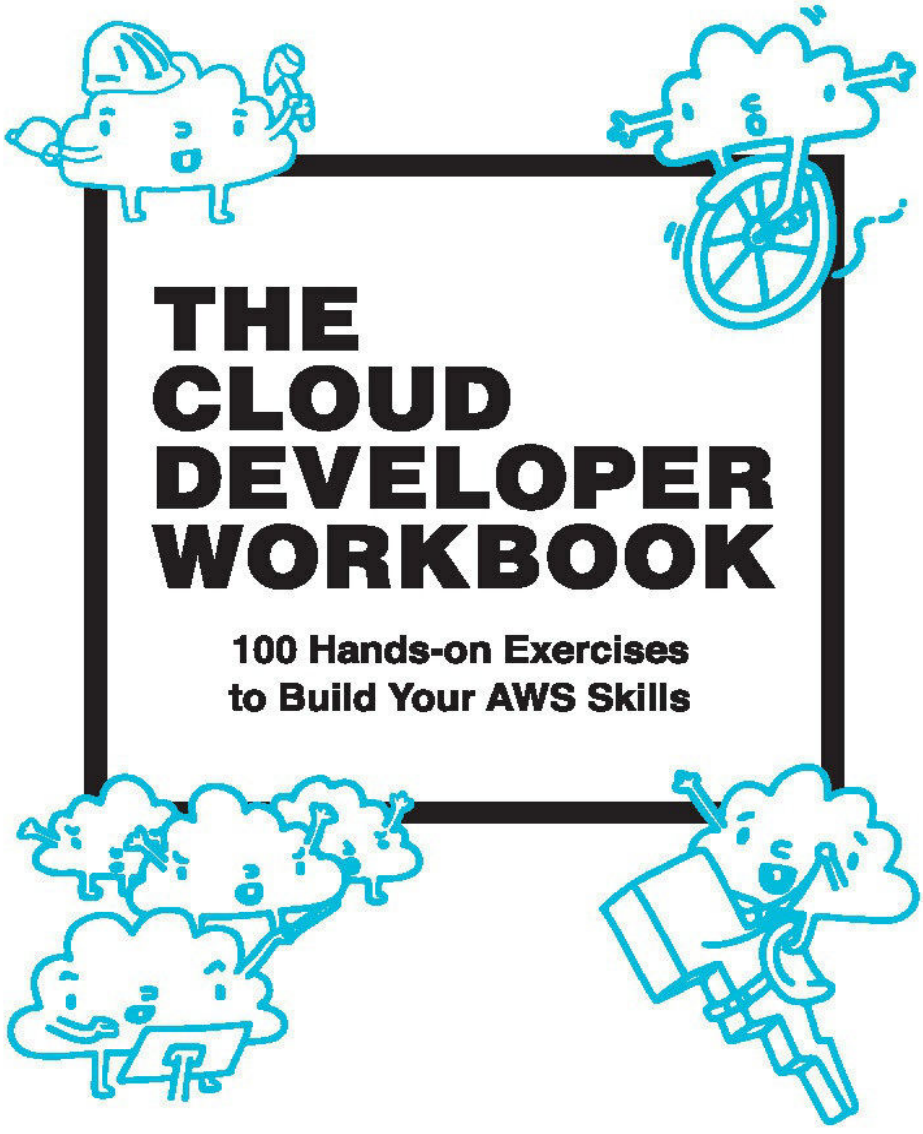


THE CLOUD DEVELOPER WORKBOOK

100 Hands-on Exercises to Build Your AWS Skills

Ryan H. Lewis


THE
CLOUD
DEVELOPER
WORKBOOK
100 Hands-on Exercises
to Build Your AWS Skills
Ryan H. Lewis



THE CLOUD DEVELOPER WORKBOOK

100 Hands-on Exercises to Build Your AWS Skills

Ryan H. Lewis

The Cloud Developer Workbook

Version 1.0 – February 2021
Interior Illustrations and Book Text: Ryan H. Lewis
Editor: Jo Finchen-Parsons
Proofreader: Whitney Stovall
Cover Design and Interior Layout: Sherry Anne Luna

ISBN: 9798700306485

Table of Contents

How to Use This Workbook

This workbook includes 100 exercises for increasing your knowledge of Amazon Web Services (AWS). Each exercise will ideally take you less than an hour to complete if you do everything on the page. Although the exercises build on each other, feel free to jump around and cover the topics you want. I suggest doing one exercise a day to allow the new information to settle before jumping into the next exercise. The most important thing is to try to tackle every single exercise in this workbook. AWS is nebulous, but you will have a comprehensive understanding of what AWS is and what you can do with it by completing everything in this workbook. And you'll also have a lot of fresh hands-on cloud development experience!

There are four sections in each exercise:

1. Each exercise begins with explaining what you'll be focusing on during that exercise in the **Introduction** .
2. The **Study** section contains articles and documentation to read to understand the concept of the exercise better. Reading about services helps you get more comfortable with technical documentation and makes it easier to learn on your own in the future.
3. The **Tasks** section gives you a step-by-step set of tasks for you to complete for that exercise. When necessary, I'll include links to other guides that will help you. The point of this section is for you to get hands-on development experience with AWS and cement the things you've read.
4. The **Consider** section contains some questions to answer on your own or with a study group. Active thinking will help you better understand what you've learned and done in the exercise.

If you have questions or feedback, I'd love to hear from you. Email me at ryan@ryanlewis.dev or @ryanmurakami on Twitter.

Best of luck on your journey!

— Ryan H. Lewis

100 Exercises


Getting Started

Exercise 1: Setting Up Your Account

Introduction

To start, you're going to make a new AWS account. You may already have an AWS account, but to save you some money and get free resource hours, starting a new account will enroll you in the free tier.

Study

- *AWS Free Tier*
  - https://ryanlewis.link/aws-free-tier

Tasks

- Set up a new AWS account
  - Open a new AWS account
  - Activate Multi-Factor Authentication for the root account
  - Set a password policy for your account in Identity and Access Management (IAM)

Consider

- Why should your root account be extra secure?
- What do you hope to achieve by completing these 100 exercises?
- What do you want to do with AWS?

Exercise 2: Your First IAM User

Introduction

Now you're going to set up the method with which you'll access AWS from now on. Accessing your account with the root user is risky because of elevated permissions, so the best practice is to use an IAM user instead.

Study

- *Policies and permissions in IAM*
  - https://ryanlewis.link/IAM-policies-and-permissions

Tasks

- Create a new IAM user and give it permissions
  - Create a group with the name "Admins"
  - Create a user with your name
    - Select the "Password for Authentication" method
  - Add the "AdministratorAccess" policy to the group
  - Add the user you created to the group "Admins"

Consider

- Why are groups used with users?
- When should you add policies to a user directly?
- When should you add policies to a group?
- If you delete a group, what happens to the users assigned to that group?

Exercise 3: Setting Up the AWS CLI

Introduction

Using the AWS Management Console is great for experimentation and discovery, but the AWS Command Line Interface (CLI) is less error-prone and easier to automate. In this exercise, you're going to configure your AWS CLI to connect to your account.

Study

- *What is the AWS Command Line Interface?*
  - https://ryanlewis.link/aws-cli
- *Where's My Secret Access Key?*
  - https://ryanlewis.link/aws-secret-access-key

Tasks

- Configure the AWS CLI
  - Create an access key and secret access key for your user
    - https://ryanlewis.link/aws-create-access-key
  - Install the AWS CLI v2
    - https://ryanlewis.link/aws-cli-v2-install
  - Run aws configure in your command line and enter your access key and secret access key
    - https://ryanlewis.link/aws-cli-configure
  - Try out commands from AWS CLI docs
    - https://ryanlewis.link/aws-cli-command-reference
  - Suggestion: View any running EC2 instances
    - https://ryanlewis.link/aws-describe-instances

Consider

- How often should you rotate (delete and create new) your AWS access key?
- When would you use the AWS CLI versus the AWS Management Console?
- How can you retrieve a secret access key?

Exercise 4: Setting Up a Budget

Introduction

Let’s set up a budget in AWS Budgets to ensure you don’t receive any surprise bills in the future. Even though your account is in the free tier, some services only give you a certain amount of free resources before you may be charged.

Study

- *Managing your costs with AWS Budgets*
  - https://ryanlewis.link/aws-budget-managing-costs
- *Control your AWS Costs*
  - https://ryanlewis.link/AWS-control-costs

Tasks

- Create a budget in AWS Budgets
  - Create a cost budget
  - Make the budget monthly, with a fixed cost you are comfortable with (I usually put $5)
  - Set the threshold to "Forecasted Cost" at 100%
  - Use an email address you often use so you'll see the budget alert email

Consider

- What's the difference between an actual cost threshold and a forecasted cost threshold in AWS Budgets?
- What are some of the vectors with which you can segment budgets?


Elastic Compute Cloud

Exercise 5: Creating an EC2 Instance

Introduction

You need a place in the cloud to run your code, and EC2 is that place. An instance in EC2 is a virtual server that can run an operating system and any application code you want. In this exercise, you're going to create an EC2 instance.

Study

- *Amazon EC2 Instance Types*
  - https://ryanlewis.link/aws-EC2-instance-types
- *Amazon Linux 2*
  - https://ryanlewis.link/aws-linux-2
- *Key Pair documentation*
  - https://ryanlewis.link/aws-ec2-key-pairs-linux
  - https://ryanlewis.link/aws-ec2-key-pairs-windows

Tasks

- Create an EC2 instance
  - Instance type: t3.micro
  - Amazon Machine Image (AMI): Amazon Linux 2
  - Default options for everything else
  - Create and save the key pair

Consider

- What tasks are best suited for each instance type category?
- What product domain do you work in, and what instance type

would suit those types of applications the best?

- What other categories of AMIs are there? Tip: Try looking in the AWS Marketplace.

Exercise 6: EC2 Security Groups

Introduction

Now you're going to attach an EC2 security group to your instance. EC2 security groups are mini firewalls that protect your EC2 instances by blocking or allowing traffic from sources and ports.

Study

- *Working with security groups*
  - https://ryanlewis.link/aws-ec2-security-windows
- *Amazon EC2 security groups for Linux instances*
  - https://ryanlewis.link/aws-ec2-security-linux

Tasks

- Create a new EC2 security group
  - Add an incoming rule to the security group for SSH access (port 22) from your IP address
  - Add an incoming rule for port 3000 from anywhere
- Attach the security group to your EC2 instance

Consider

- What does an EC2 security group protect the instance from?
- What is a VPC security group?
- How many security groups can be attached to an EC2 instance?

Exercise 7: Prepping the EC2 Instance

Introduction

Accessing your EC2 instances with SSH is vital for deployment, retrieving logs, and troubleshooting. In this exercise, you're going to use SSH to log in to your instance and install dependencies for your application.

Study

- *Connecting to your Linux instances using SSH*
  - https://ryanlewis.link/aws-ssh-linux-connect
- *Bonus points for reading the Linux SSH Best Practices guide*
  - https://ryanlewis.link/aws-ssh-linux-guide

Tasks

- Install dependencies on your EC2 instance
  - SSH into your EC2 instance
    - Reference the keypair when connecting
    - E.g. ssh -i ~/.ssh/my_keypair ec2-user@<public_ip_of_instance>
- Install any dependencies needed for your web application

Consider

- What are the security risks of leaving the SSH port open on your instance?
- If your application was having issues, how could you troubleshoot it using SSH?

Exercise 8: Deploying to an EC2 Instance

Introduction

It's time to deploy your application to the EC2 instance. While there are many ways to deploy to AWS, merely copying your code over is the most straightforward. We'll try other methods in later exercises.

Study

- *Transferring files to EC2*
  - https://ryanlewis.link/aws-ec2-file-transfer
- *Deploying with AWS CodeDeploy*
  - https://ryanlewis.link/aws-code-deploy

Tasks

- Deploy an application to EC2
  - Prepare your web application (or EC2 Field Day from me)
  - Use SCP to copy the application to your instance
    - https://ryanlewis.link/scp-file-transfer-linux
    - SSH & SCP on Windows: https://ryanlewis.link/scp-ssh-file-transfer-windows
  - SSH into your instance and start the application

Consider

- How could you automate the deployment process that you followed in this exercise?
- What deployment tools are you aware of? Do they support deployment to AWS?

Exercise 9: More Deploying to EC2 Instances

Introduction

Let's try deploying your code differently. It's essential to consider security and ease of use when picking a deployment method. One size doesn't fit all, and it matters most what works best for your team.

Study

- *Running commands on your Linux instance at launch*
  - https://ryanlewis.link/aws-run-commands-linux
- *Running commands on your Windows instance at launch*
  - https://ryanlewis.link/aws-run-commands-windows

Tasks

- Deploy an application from GitHub to EC2
  - Find a web application on GitHub
    - E.g. https://ryanlewis.link/github-ec2-field-day-demo
  - Write a shell script to do the following:
    - Clone the app from GitHub
    - Install any dependencies
    - Start the app
  - Create a new EC2 instance and add the script to the user data section
  - Once the instance has started, try verifying that everything worked by accessing the EC2 instance with its public IP address

Consider

- What kind of commands would you want to add to the user data for instances?
- Does user data solve all your needs for deployment? What is missing?

Exercise 10: Creating an Elastic IP

Introduction

Public IP addresses are needed to access your EC2 instance from the Internet. Elastic IPs don't disappear until you release them, so they enable you to maintain a consistent address even if the EC2 instance changes. In this exercise, you'll attach an Elastic IP to your instance and view your application.

Study

- *EC2 Elastic IP Addresses*
  - https://ryanlewis.link/aws-ec2-elastic-ip-address
- *Amazon EC2 Instance IP Addressing*
  - https://ryanlewis.link/aws-ec2-instance-ip-address
- Understanding Elastic IP Charges
  - https://ryanlewis.link/aws-ec2-elastic-ip-charges

Tasks

- Attach an Elastic IP address to your EC2 instance
  - Create a new Elastic IP
  - Associate it with your EC2 instance
  - Make sure your app is running on your instance
  - Use the Elastic IP + Port to view your app in a browser
    - E.g. 127.0.0.0:3000
    - If that address times out, double-check the security group associated with the instance. Make sure it gives access to incoming connections on the right port to your IP address (or anywhere).

Consider

- What is the difference between a VPC Elastic IP address and an EC2 Elastic IP address?
- What is the cost of an Elastic IP address? Is there an additional cost if it is associated with an instance?

Exercise 11: Creating a Load Balancer

Introduction

Load balancing keeps your applications from becoming unavailable, and it efficiently balances requests across your instances. In this exercise, you're going to create a load balancer to route requests to your application.

Study

- *What is Elastic Load Balancing?*
  - https://ryanlewis.link/aws-elastic-load-balancing
- *What is an Application Load Balancer?*
  - https://ryanlewis.link/aws-application-load-balancer

Tasks

- Attach a load balancer to your EC2 instance
  - Create a target group
    - Listen on port 3000
    - Register your instance to the target group
  - Create a Load Balancer
    - Application load balancer type
    - Select the target group you created
  - Try the load balancer DNS name in your browser

Consider

- What are some of the differences between an Application Load Balancer and a Network Load Balancer?

- How does a Load Balancer scale as requests increase?

Exercise 12: Securing your EC2 Instance

Introduction

In this exercise, you're going to update your EC2 security group so that only the load balancer can access your instance. This access method is a best practice because it blocks attackers from accessing your instances directly.

Study

- *Principle of Least Privilege*
  - https://ryanlewis.link/principle-of-least-privilege
- *Security in Amazon EC2*
  - https://ryanlewis.link/aws-ec2-security

Tasks

- Configure your EC2 instances to only allow traffic from the load balancer
  - Change the EC2 security group ingress rules
    - Allow Port 3000 connections from the load balancer's security group
    - Remove the rule for SSH connections
  - Try accessing the app again with the load balancer DNS name

Consider

- How can an EC2 instance still be insecure if it has restrictive security group rules?
- How many of the OWASP Top 10 risks can be mitigated by EC2 security groups?

Exercise 13: Creating an Amazon Machine Image

Introduction

Now we're going to save the work you've done on EC2 as an Amazon Machine Image. AMIs are the best way to store EC2 state for security or sharing purposes. We'll be removing the resources you've created to avoid any future charges.

Study

- *Amazon Machine Images (AMI)*
  - https://ryanlewis.link/aws-ami

Tasks

- Save your work with an AMI and clean up
  - Create an Amazon Machine Image (AMI) from your EC2 instance
    - This will use storage from EBS, but the free tier will cover it for the first 12 months of your account
  - Remove all other resources created until now
    - Delete the target group and Load Balancer
    - Terminate the EC2 instance
    - Delete the EC2 security group
    - Release the Elastic IP address

Consider

- How can you use AMIs to share code or applications between projects and teams?

- Does your place of work have a centralized team that would manage AMIs?


Well-Architected
Framework

Exercise 14: The Well-Architected Framework

Introduction

The exercises leading up to this one have built your foundational knowledge of AWS while you focused on building and deploying an application. Now you're going to learn how to develop the right way in the cloud by studying AWS's battle-tested best practices in the "Well-Architected Framework".

Study

- *The AWS Well-Architected Framework – Introduction section*
  - https://ryanlewis.link/aws-waf
  - Or pages 1–6 at: https://ryanlewis.link/aws-waf-whitepaper

Tasks

- Read the white paper above

Consider

- Why does AWS produce content like the "Well-Architected Framework"?
- Which pillar do you know the most about already? Which do you know the least?
- Of the General Design Principles, which are not relevant to cloud developers?

Exercise 15: The Operational Excellence Pillar

Introduction

Continually analyzing and improving your cloud development practices ensures you spend less and less time on maintaining cloud hygiene. The Operational Excellence Pillar gives you concrete best practices for achieving that.

Study

- *The AWS Well-Architected Framework – Operational Excellence Pillar*
  - https://ryanlewis.link/aws-waf-operational-excellence
  - Or pages 6–15 at: https://ryanlewis.link/aws-waf-whitepaper
- *Bonus: Operational Excellence Pillar White Paper (48 pages)*
  - https://ryanlewis.link/aws-waf-operational-excellence-whitepaper

Tasks

- Read the white papers above

Consider

- What services or tools does AWS provide to support the Operational Excellence Pillar principles?
- Are there tools you use when developing software that support these same Operational Excellence principles outside of AWS?

Exercise 16: The Security Pillar

Introduction

You do not want to be responsible for a security breach in your cloud application, especially when it's something that AWS has already warned you not to do. Reading up on the Security Pillar will ensure that you utilize all the security best practices that AWS has to offer.

Study

- *The AWS Well-Architected Framework – Security Pillar*
  - https://ryanlewis.link/aws-waf-security
  - Or pages 15–22 at: https://ryanlewis.link/aws-waf-whitepaper
- *Bonus: Security Pillar White Paper (46 pages)*
  - https://ryanlewis.link/aws-waf-security-whitepaper

Tasks

- Read the white papers above

Consider

- How should you manage user access to AWS resources?
- What services or tools does AWS provide to support the Security Pillar principles?
- Are there any security concerns you already focus on that AWS takes care of on its own?

Exercise 17: The Reliability Pillar

Introduction

Reliability is a big deal. It is why so many developers and companies choose AWS. The Reliability Pillar highlights how to scale instances, deploy efficiently, and instantly provision resources.

Study

- *The AWS Well-Architected Framework – Reliability Pillar*
  - https://ryanlewis.link/aws-waf-reliability
  - Or pages 22–28 at: https://ryanlewis.link/aws-waf-whitepaper
- Bonus: Reliability Pillar White Paper (74 pages)
  - https://ryanlewis.link/aws-waf-reliability-whitepaper

Tasks

- Read the white papers above

Consider

- Why is scaling horizontally a better practice than scaling vertically?
- What services or tools does AWS provide to support the Reliability Pillar principles?

Exercise 18: The Performance Efficiency Pillar

Introduction

The Performance Efficiency Pillar explains how to choose the right cloud tool for the job. With over 175 services, you want to at least be aware of what each service does so you don't end up rebuilding something that AWS already offers.

Study

- *The AWS Well-Architected Framework – Performance Efficiency Pillar*
  - https://ryanlewis.link/aws-waf-efficiency
  - Or pages 28–36 at: https://ryanlewis.link/aws-waf-whitepaper
- *Bonus: Performance Efficiency Pillar White Paper (43 pages)*
  - https://ryanlewis.link/aws-waf-efficiency-whitepaper

Tasks

- Read the white papers above

Consider

- What aspects would you consider when choosing a Compute solution in AWS?
- What services or tools does AWS provide to support the Performance Efficiency Pillar principles?

Exercise 19: The Cost Optimization Pillar

Introduction

Even if paying the AWS bill doesn't fall on your shoulders, there are some really great best practices that the Cost Optimization Pillar can offer. By combining those best practices with the Performance Efficiency Pillar, you'll always be able to create the highest performing applications for the lowest cost.

Study

- *The AWS Well-Architected Framework – Cost Optimization Pillar*
  - https://ryanlewis.link/aws-waf-optimization
  - Or pages 36–43 at: https://ryanlewis.link/aws-waf-whitepaper
- *Bonus: Cost Optimization Pillar White Paper (41 pages)*
  - https://ryanlewis.link/aws-waf-optimization-whitepaper

Tasks

- Read the white papers above

Consider

- How can you save money on AWS resources when your teams aren't working?
- What services or tools does AWS provide to support the Cost Optimization Pillar principles?


Simple Storage Service

Exercise 20: Creating an S3 Bucket

Introduction

In this exercise, you'll take your first steps into S3, AWS's oldest service, by creating a bucket and uploading some files. Using this simple form of storage is the backbone of nearly every cloud application.

Study

- *What is Amazon S3?*
  - https://ryanlewis.link/aws-s3
- *Working with Amazon S3 Buckets*
  - https://ryanlewis.link/aws-s3-buckets

Tasks

- Create an S3 bucket and upload files
  - Create an S3 bucket
    - Uncheck "Block All Public Access"
  - Upload some files
    - Use all default options

Consider

- Why shouldn't S3 buckets be public?
- What file types can be stored in an S3 bucket?

Exercise 21: Accessing S3 Objects

Introduction

Choosing who accesses each object in S3 is up to the developer. Now, you're going to make some configuration changes to increase that access. Understanding the access properties of S3 is vital to avoiding a costly data leak for your company.

Study

- *Setting S3 Object Permissions*
  - https://ryanlewis.link/aws-s3-object-permissions
- *Managing Access with ACLs*
  - https://ryanlewis.link/aws-s3-acls

Tasks

- Make an S3 object readable by everyone
  - Select one of the S3 objects you uploaded
  - Click "Edit" in the "Access control list (ACL)" panel
    - Click the checkbox next to "Everyone (public access)" and under "Objects"
    - Click the acknowledgment checkbox then "Save changes"
  - Click the "Object URL" link at the top of the object details page

Consider

- When should S3 objects be made public (readable by everyone)?
- What are other ways you could let the public access files in S3 without making the object public?

Exercise 22: S3 Lifecycle Rules

Introduction

Managing an assortment of S3 objects on your own is the quickest way to drive yourself crazy. We're going to set up an S3 lifecycle rule to automatically delete objects after a specified amount of time since creation. You can use lifecycle rules to automate your S3 object management in many different ways.

Study

- *S3 Object Lifecycle Management*
  - https://ryanlewis.link/aws-s3-object-lifecycle-mgmt
- *Transitioning Objects using Amazon S3 Lifecycle*
  - https://ryanlewis.link/aws-s3-transitioning-objects

Tasks

- Create an S3 lifecycle rule for your bucket
  - Go to the "Management" tab for the bucket you created
  - Create a new lifecycle rule
    - In the "Lifecycle rules" section, click "Create lifecycle rule"
    - Give it a name and select the radio button for "all objects"
    - Pick a lifecycle rule, such as "Expire current versions of objects"
    - Set the settings for the rule
    - Click "Create rule"

Consider

- What kinds of objects would you want to configure a lifecycle rule

for?

- Which lifecycle rule could you see yourself using most often?

Exercise 23: S3 Storage Classes

Introduction

Storing all your objects in the Standard S3 storage class can get expensive. In this exercise, you'll learn about the different S3 storage classes and move some of your objects to a cheaper option.

Study

- *Amazon S3 Storage Classes*
  - https://ryanlewis.link/aws-s3-storage-classes
- *S3 Intelligent Tiering*
  - https://ryanlewis.link/aws-s3-intelligent-tiering

Tasks

- Move an object to a cheaper S3 storage class
  - Change the storage class for an object
    - Click on an object
    - Click "Edit" in the "Storage class" section
    - Check it from "Standard" to "Standard-IA"
    - Click "Save changes"

Consider

- What types of files would be best to store in S3 Glacier?
- How does S3 Intelligent Tiering work?

Exercise 24: S3 Website Hosting

Introduction

Now you are going to add configuration to your bucket to make it an S3 static website. Whether you're serving HTML or client-side assets, this is one of the quickest ways to get files to your end-users.

Study

- *Hosting a Static Website on Amazon S3*
  - https://ryanlewis.link/aws-s3-static-website
- *Configuring an Amazon S3 Bucket for Static Website Hosting*
  - https://ryanlewis.link/aws-s3-bucket-configure

Tasks

- Set up an S3 static website
  - Go to your bucket in the AWS Management Console
  - Click on the "Properties" tab
  - At the bottom, click "Edit" in the "Static website hosting" panel
  - Click the "Enable" radio button
  - Enter the key for one of your objects for the "Index document" and "Error document"
  - Click "Save changes"
  - Use the endpoint link in the "Static website hosting" panel to test it

Consider

- What kinds of objects are best served with an S3 static site?

- What is different about an S3 static website versus an S3 public bucket?

Exercise 25: S3 Bucket Policies

Introduction

Set a bucket policy to avoid having to configure each object in an S3 bucket individually. A single purpose bucket can save you a lot of time and prevent mistakes with uneven object configuration. In this exercise, you'll configure a read-only public policy for your bucket.

Study

- *S3 Bucket Policy Examples*
  - https://ryanlewis.link/aws-s3-bucket-policy-examples
- *How to use S3 Bucket Policies to Secure Your Data*
  - https://ryanlewis.link/aws-s3-bucket-security

Tasks

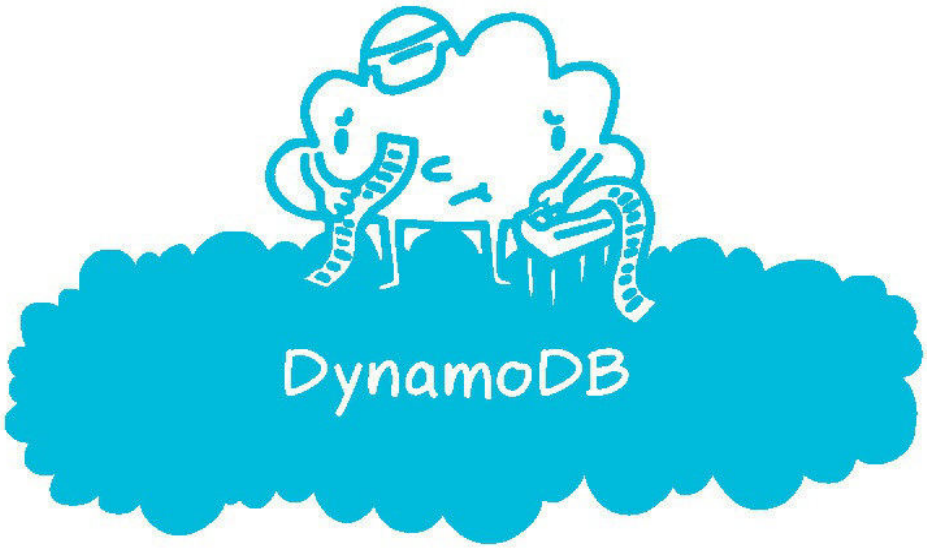
- Set a read-only public bucket policy in your S3 bucket
  - Copy the "Read-Only..." policy from the bucket policy examples
    - https://ryanlewis.link/aws-s3-bucket-policies-read-only
  - Go to your bucket details page and the "Permissions" tab
  - Click "Edit" in the "Bucket policy" panel
  - Copy the policy into the policy section
  - Click "Save changes"
- All objects in the bucket should now be public
  - Try testing it by clicking on all their URLs

Consider

- What's the difference between setting an S3 bucket policy and

setting an S3 object ACL?

- What are some ways to use a bucket policy other than giving access to everyone?


DynamoDB

Exercise 26: Creating a DynamoDB Table

Introduction

Using DynamoDB for data storage is an easy method to quickly have a scalable database that's easy to work with. Getting hands-on experience with it means you'll be ready to use it in your applications. In this exercise, you'll create a DynamoDB table and create some items.

Study

- *What is Amazon DynamoDB?*
  - https://ryanlewis.link/aws-dynamodb
- *Working with Tables and Data in DynamoDB*
  - https://ryanlewis.link/aws-dynamodb-tables-data

Tasks

- Create a DynamoDB table and some items
  - Create a DynamoDB table
    - Go to the DynamoDB dashboard
    - Click "Create table"
    - Enter a name for the table
    - Enter "id" for the Partition key with the type of "String"
    - Use the "Default settings"
    - Click "Create table"
  - Once it's done creating, click into the table and create some items on the "Items" tab
    - Use multiple properties for the items you create. We'll use them later

Consider

- What's the difference between DynamoDB tables and S3 buckets?
- What kind of databases have you used before? How does DynamoDB differ?

Exercise 27: DynamoDB Table Capacity

Introduction

DynamoDB storage capacity isn't something you have to worry about, but you do have to manage concurrent reads and writes by configuring your table's capacity. In this exercise, you'll configure that capacity, increasing it to handle more concurrent reads and writes. This is what you would want to do for a production application.

Study

- *DynamoDB Read/Write Capacity Mode*
  - https://ryanlewis.link/aws-dynamodb-read-write
- *Considerations When Changing Read/Write Capacity Mode*
  - https://ryanlewis.link/aws-dynamodb-read-write-considerations

Tasks

- Increase the provisioned capacity for your DynamoDB table
  - Go to your table dashboard
  - Click the "Actions" dropdown and click "Edit capacity"
  - In the "Table capacity" section, you can modify both read and write capacity units
  - It's currently auto scaled, so set a different minimum and maximum for both read and write
  - Click "Save Changes"

Consider

- How does auto scaling work with DynamoDB table capacity?
- What are the differences between on-demand capacity and

provisioned capacity?

Exercise 28: DynamoDB Indexes

Introduction

Creating an index for your DynamoDB table will increase the ease and speed with which you can access your data. Matching your access methods to indexes in DynamoDB is the secret to a performant datastore. In this exercise, you'll create a global secondary index for your table.

Study

- *DynamoDB Secondary Indexes*
  - https://ryanlewis.link/dynamodb-secondary-indexes
- *Using Global Secondary Indexes in DynamoDB*
  - https://ryanlewis.link/aws-dynamodb-gsi
- *How to Design Global Secondary Indexes*
  - https://ryanlewis.link/aws-dynamodb-gsi-design

Tasks

- Create a global secondary index for your DynamoDB table
  - Go to your table dashboard
  - Select the "Indexes" tab
  - Click "Create index"
  - Enter one of the additional fields you entered for your items in the "Partition key" input box
  - Give the index a name
  - Click "Create index"

Consider

- When would you want to use a global secondary index?
- What is the main difference between a local and global secondary index?

Exercise 29: The DynamoDB SDK

Introduction

One of the awesome features of DynamoDB is that you access your data through a convenient REST API. By writing a program using the DynamoDB SDK, you'll experience how easy it will be to interact with your data in real world applications.

Study

- *Programming with DynamoDB and the AWS SDKs*
  - https://ryanlewis.link/aws-dynamodb-programming
- *JavaScript AWS SDK DynamoDB Documentation*
  - https://ryanlewis.link/aws-dynamodb-javascript

Tasks

- Write a program to interact with your DynamoDB data
  - Use a language you're comfortable with that has a supported AWS SDK
    - https://ryanlewis.link/aws-dynamodb-tools
  - Write a small script that uses the AWS SDK to access your DynamoDB table
  - Try performing some of these operations
    - Put a record
    - Get a record
    - Query and scan

Consider

- How is accessing DynamoDB in code different from working with

other databases?

- Which SDK method seems the easiest to use?

Exercise 30: Modeling DynamoDB Tables

Introduction

With DynamoDB tables, you model how you expect to access the data instead of normalizing the data like in relational databases. Effective modeling results in quicker database transactions and cleaner code. In this exercise, you'll read some DynamoDB data modeling examples and then try your hand at modeling.

Study

- *Best Practices for Modeling Relational Data in DynamoDB*
  - https://ryanlewis.link/aws-dynamodb-best-practices
- *Examples of Modeling Relational Data in DynamoDB*
  - https://ryanlewis.link/aws-dynamodb-best-practices-examples

Tasks

- Model a DynamoDB table
  - Use some of the techniques in the Study articles
  - Draw a model for data you'd put in DynamoDB
  - Following the model, create a table and add some items
- Delete your DynamoDB table after this exercise

Consider

- What is the main difference between modeling data in DynamoDB and Relational databases?
- How can modeling data improve how you use DynamoDB?


Relational Database
Service

Exercise 31: RDS Engine Types

Introduction

Since you've learned about non-relational data with DynamoDB, it's also essential to get experience with managed relational databases in Amazon Relational Database Service. It's unlikely you'll encounter an application that doesn't need a relational database, so understanding how AWS can help you is essential.

Study

- *What is Amazon Relational Database Service?*
  - https://ryanlewis.link/aws-rds
- *Best Practices for Amazon RDS*
  - https://ryanlewis.link/aws-rds-best-practices
- *MySQL on Amazon RDS*
  - https://ryanlewis.link/aws-rds-mysql
- *PostgreSQL on Amazon RDS*
  - https://ryanlewis.link/aws-rds-postgresql

Tasks

- Read the links above around the engine types available in RDS

Consider

- Why does AWS provide hosted solutions for these open-source database engines?
- Why doesn't AWS just make all their own DB engines?

Exercise 32: Creating an RDS Database

Introduction

Now, you'll take what you learned from Exercise 31 about RDS and actually create a database instance, and choose which engine type to use. Setting up a database instance with the right configuration has a massive impact on your application performance.

Study

- *Creating an Amazon RDS DB Instance*
  - https://ryanlewis.link/aws-rds-db-instance
- *Maintaining a DB Instance*
  - https://ryanlewis.link/aws-rds-db-instance-maintain

Tasks

- Create an RDS DB instance
  - Choose a database engine
  - Create an RDS instance with that engine
    - Use "Easy create"
    - Select "Free tier"

Consider

- What kind of database maintenance does AWS do for you with RDS?
- What should you consider when selecting an RDS DB Instance type?

Exercise 33: RDS Accessibility

Introduction

Now you're going to configure your database to connect to it from your local computer. This configuration change will let you connect to it with database clients you may be familiar with and see how accessing a managed database in RDS works mostly the same as any other method. The process listed is not secure for production applications, so make sure to follow AWS best practices in those cases.

Study

- *Security in Amazon RDS*
  - https://ryanlewis.link/aws-rds-security
- *Best Practices for Securing Data in Amazon RDS*
  - https://ryanlewis.link/aws-rds-security-best-practices

Tasks

- Configure your RDS DB for public access and connect to it
  - Configure your RDS DB to let you access it
    - Add an Ingress rule to the RDS DB security group
      - Select the port the database engine uses
      - Select "My IP" for source
    - Edit the DB instance to set "Publicly Accessible" to "True"
  - Try connecting to the DB using a local client

Consider

- How would you secure a DB instance in RDS?
- What does setting a DB instance to be publicly accessible

actually do?

Exercise 34: RDS Instance Types

Introduction

There are many different instance type options for your RDS databases, and choosing the right one can make a considerable cost and performance difference. For this exercise, you'll make a configuration change to your database instance type.

Study

- *Amazon RDS Instance Types*
  - https://ryanlewis.link/aws-rds-instance-types
- *DB Instance Classes*
  - https://ryanlewis.link/aws-rds-instance-types-db

Tasks

- Change the instance type of your RDS database
  - Edit your DB instance
  - Choose a larger or smaller instance type
  - Make the change effective immediately

Consider

- What does the instance type affect in the operation of your database?
- What instance type is best for your use case and why?

Exercise 35: RDS Backups

Introduction

Now you're going to restore your database from a backup. RDS enables automated backups by default, but it's good to run through the restoration process so that you're ready in the event you need to recover from a catastrophic failure.

Study

- *Working with RDS Backups*
  - https://ryanlewis.link/aws-rds-backups
- *Amazon RDS Backup and Restore*
  - https://ryanlewis.link/aws-rds-backups-restore

Tasks

- Restore your database from a backup
  - Automated backups are enabled by default
  - Find a recent backup
    - If there are none yet, create a manual backup first
  - Restore your DB from that backup
  - Try accessing the restored DB
- Terminate any RDS DB instances after this
  - Make sure to delete any DB Snapshots if they exist

Consider

- How often do you need your database backed up?
- Why does RDS create a new DB instance from a backup instead

of restoring it in place?


Regions & Availability
Zones

Exercise 36: AWS Regions

Introduction

The AWS Infrastructure doesn't just impact how pricing works, it also affects how you develop. For this exercise, you'll become familiar with changing and working in different regions, so that you don't make mistakes in the future, such as provisioning resources in the wrong region.

Study

- *AWS Regions and Availability Zones*
  - https://ryanlewis.link/aws-global-infrastructure-regions
- *AWS Regional Services*
  - https://ryanlewis.link/aws-global-infrastructure-services

Tasks

- Explore switching regions in the AWS Management Console
  - Are there any regions you're restricted from using?
- Not all services are regional
  - Find some of the services that are global

Consider

- What are some of the differences between regions besides physical location?
- Which region is closest to you or your users?

Exercise 37: AWS Availability Zones

Introduction

Availability zones are the physical data centers where AWS resources are actually provisioned. Being able to work between them and utilize that global infrastructure to achieve reliability is one of the skills expected from any cloud developer. In this exercise, you can try modifying the availability zones of any resources you still have running.

Study

- *Finding Availability Zone Options*
  - https://ryanlewis.link/aws-availability-zone-options
- *Choosing Regions and Availability Zones*
  - https://ryanlewis.link/aws-regions-availability-zones

Tasks

- Try changing the deployment availability zone for any RDS DB instances or EC2 instances you have running

Consider

- How are availability zones connected to each other?
- Why do some services require multiple availability zones to be selected?


Pricing

Exercise 38: The AWS Pricing Model

Introduction

You may not have to worry about the AWS bill on larger teams, but understanding how pricing works and getting the most performance for the least cost is a valuable skill. In this exercise, you'll take a look at the billing dashboard for your account and see if you have any existing charges to analyze.

Study

- *AWS Pricing*
  - https://ryanlewis.link/aws-pricing
- *Amazon EC2 Pricing*
  - https://ryanlewis.link/aws-ec2-pricing
- *Amazon S3 Pricing*
  - https://ryanlewis.link/aws-s3-pricing
- *How AWS Pricing Works White Paper*
  - https://ryanlewis.link/aws-pricing-whitepaper

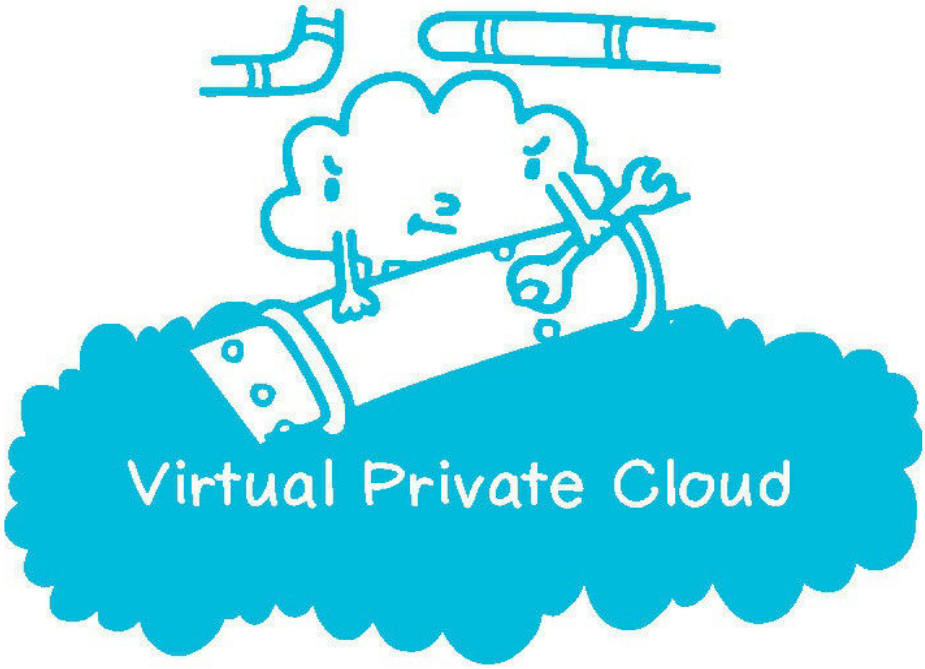
Tasks

- Take a look at the billing dashboard in your account
  - Have any charges accrued?
  - Look at your free tier usage so far
  - Check out the cost explorer if you have charges

Consider

- What's the most expensive AWS service you've found?

- Why is the pricing for services different for each region?


Virtual Private Cloud

Exercise 39: Creating a VPC

Introduction

VPCs act as the foundation for most of your resources. Setting up a VPC with the correct configuration is essential for your application security and performance. In this exercise, you will create and configure a new VPC for you to use.

Study

- *What is Amazon VPC?*
  - https://ryanlewis.link/aws-vpc
- *Examples for VPC*
  - https://ryanlewis.link/aws-vpc-examples

Tasks

- Create a new VPC
  - Use the CIDR Block: 10.0.0.0/16

Consider

- How does VPC help protect your AWS resources?
- If VPC is the networking for EC2, why do other services in AWS use it as well?

Exercise 40: Creating VPC Subnets

Introduction

In this exercise, you will set up subnets for your VPC. These are necessary to provision your resources. You will also set them up so that you can later change to a public/private subnet configuration, which is the most secure way to configure your VPC.

Study

- *VPCs and Subnets*
  - https://ryanlewis.link/aws-vpc-subnets
- *Default VPC and Default Subnets*
  - https://ryanlewis.link/aws-vpc-subnets-default

Tasks

- Create two subnets
  - Use the VPC created in Exercise 39
  - Name one "public" with CIDR 10.0.0.0/24
  - Name the other "private" with CIDR 10.0.1.0/24

Consider

- Why would you want to create resources in different subnets?
- What configuration options do VPCs have that subnets do not?

Exercise 41: Creating an Internet Gateway

Introduction

In this exercise, you're going to add an Internet Gateway to your VPC. Internet Gateways allow the resources in a VPC to connect to the outside world, which is required by pretty much any web application.

Study

- *Internet Gateways*
  - https://ryanlewis.link/aws-vpc-internet-gateways
- *Create and Attach an Internet Gateway to a VPC*
  - https://ryanlewis.link/aws-vpc-internet-gateways-create

Tasks

- Create a new Internet Gateway
  - Attach it to your VPC

Consider

- What exactly is an Internet Gateway?
- Why are Internet Gateways used to give AWS resources access to the Internet?

Exercise 42: Creating a Route Table

Introduction

VPCs are the plumbing for your AWS resources, which explains why there are quite a few steps to set them up. Route tables define how to route outgoing network traffic in your VPC, and configuring it the right way is necessary to have a working VPC. In this exercise, you're going to set up a route table to send all outgoing, non-local traffic to the Internet Gateway.

Study

- *VPC Route Tables*
  - https://ryanlewis.link/aws-vpc-route-tables
- *Working with Route Tables*
  - https://ryanlewis.link/aws-vpc-route-tables-working

Tasks

- Create and configure a route table
  - Create a new route table
    - There will be an existing route for local, VPC traffic
    - Add a new route that sends 0.0.0.0/0 traffic to the Internet Gateway
  - Set it as the main route table for your VPC

Consider

- How do route tables increase the security of your VPC?
- What are some other types of routes you may want to add to your route table?

Exercise 43 : Public & Private Subnets

Introduction

Internet Gateways provide two-way traffic into your VPC and subnets. To have a more secure subnet, use NAT Gateways to allow outgoing traffic and block incoming traffic. In this exercise, you'll complete the secure public/private subnet configuration by creating a NAT Gateway and configuring a new route table to route traffic to it for your private subnet.

Study

- *VPC with Public and Private Subnets*
  - https://ryanlewis.link/aws-vpc-subnets-public-private

Tasks

- Configure your VPC for public/private subnets
  - Create a NAT Gateway
    - Provision it in your public subnet
  - Create and configure a route table
    - There will be an existing route for local, VPC traffic
    - Add a route sending 0.0.0.0/0 traffic to the NAT Gateway
  - Associate the route table with the "private" subnet

Consider

- Why shouldn't you provision all resources in a "public" subnet?
- Of the resources and services you may use, which ones should be provisioned in which type of subnet (public or private)?

Exercise 44: VPC Security

Introduction

Network ACLs are the final piece of VPC configuration used to secure your AWS resources. Luckily, they are created by default with a VPC and don't require any additional configuration, so in this exercise, you can just learn more about them and take a look at how they're configured in your VPC.

Study

- *Security Best Practices for your VPC*
  - https://ryanlewis.link/aws-vpc-security-best-practices
- *Network ACLs*
  - https://ryanlewis.link/aws-vpc-network-acls

Tasks

- Look at the network ACL assigned to your VPC
  - Look at the inbound and outbound rules

Consider

- What type of security does Network ACLs provide to your application? When would they be useful?
- When would the security measures in the "*Best Practices* " article not protect your application?


Route 53

Exercise 45: Amazon Route 53

Introduction

Route 53 requires a degree of DNS knowledge that, unfortunately, isn't always spread evenly across team members. In this exercise, you'll learn more about DNS and how Route 53 fulfills those functions in AWS. Routing is essential for all cloud applications, and Route 53 has some neat features that you can use to enhance your application capabilities.

Study

- *What is Amazon Route 53?*
  - https://ryanlewis.link/aws-route53
- *Amazon Route 53 Concepts*
  - https://ryanlewis.link/aws-route53-concepts
- *DNS Wikipedia*
  - https://ryanlewis.link/domain-name-system
- *How internet traffic is routed to your website or web application*
  - https://ryanlewis.link/aws-route53-traffic

Tasks

- Read the links above and make sure you understand the function of DNS

Consider

- Why is Route 53 considered one of the core AWS services?
- What would you use Route 53 for?

Exercise 46: Creating a Hosted Zone

Introduction

Hosted zones are how you organize your domains in Route 53, and the best part is that they don't require you to own the domain before configuring them. This lets you experiment with configurations and learn without having to pay for a domain. In this exercise, you'll create your own hosted zone for a domain name you may or may not own.

Study

- *Working with Public Hosted Zones*
  - https://ryanlewis.link/aws-route53-public-hosted-zones

Tasks

- Create a Route 53 hosted zone
  - It can be for a real or fake domain name
  - Make it "public"

Consider

- What kind of domain names can you use with Route 53?
- How would you go about getting a real URL to use with Route 53?

Exercise 47: Route 53 Record

Introduction

Users can't reach your web application without adequately configured DNS records. So in this exercise, you'll create a simple routing record in your Route 53 hosted zone. There are also many other different types of records available that you'll look into later.

Study

- *Working with Records*
  - https://ryanlewis.link/aws-route53-records
- *Supported DNS Record Types*
  - https://ryanlewis.link/aws-route53-records-dns

Tasks

- Create a new record in your hosted zone
  - Choose simple routing
  - Add an IP address as an A record

Consider

- What kind of record can forward traffic to a different URL?
- What's the difference between an A record, CNAME record, and MX record?

Exercise 48: Route 53 Routing Policies

Introduction

Simple routing records are not usually intelligent enough for modern web application routing, so Route 53 provides many other great options. In this exercise, you'll create a new record with a different routing policy to better understand how they're configured differently than a simple routing policy.

Study

- *Choosing a Routing Policy*
  - https://ryanlewis.link/aws-route53-routing-policy

Tasks

- Create a new record
  - Choose a different routing policy, such as geolocation
  - Try different settings to see what options there are

Consider

- Which type of routing policy would you be most likely to use?
- What are some of the differences between a failover routing policy and a Load Balancer?

Exercise 49: Route 53 Alias Records

Introduction

Alias records are one of the features of Route 53 that make it so unique and easy to use. IP addresses for AWS resources can be unknown or randomly changing, so setting up an alias record saves you hours of wasted DNS management. In this exercise, you'll see how to configure a simple record to point to an AWS resource via an alias.

Study

- *Choosing between Alias and Non-Alias Records*
  - https://ryanlewis.link/aws-route53-records-alias-nonalias

Tasks

- Create a record that routes to an alias
  - Create a new simple record
    - Choose an alias target
    - If you don't see any options, you may need to create a resource to route to

Consider

- What can you do with alias records that you can't do with non-alias records?
- How do alias records help teams develop more quickly?

Exercise 50: Creating a Private Hosted Zone

Introduction

When accessing AWS resources internally, hard-coded IP addresses can become a configuration nightmare. Provisioning a private hosted zone and using it across all your AWS resources can save you a lot of time and complexity. In this exercise, you'll set one up and see how to connect it to your VPC.

Study

- *Working with Private Hosted Zones*
  - https://ryanlewis.link/aws-route53-private-hosted-zones

Tasks

- Create a private hosted zone
  - Create a hosted zone
    - Select the type "Private"
    - Associate it with your VPC

Consider

- What use cases can you think of that would benefit from a private hosted zone?
- How do VPCs perform DNS Resolution for URLs?


CACHE
LAMBDA
ROL
EC2
CLOUDY
S3
Identity & Access
Management

Exercise 51: IAM User Management

Introduction

One of the most significant deficiencies for aspiring cloud developers is a lack of understanding of how IAM works and affects AWS resources. It can't be overstated how much you'll interact with IAM when working in the cloud. In this exercise, you'll create some users in IAM.

Study

- *Managing IAM Users*
  - https://ryanlewis.link/IAM-identity-and-access
- *IAM Users*
  - https://ryanlewis.link/IAM-user-guide

Tasks

- Create a few users in IAM
  - Try using the AWS CLI
  - Generate access key, but there's no need to save it

Consider

- What is the difference between an IAM user and the root account?
- How many different ways can AWS authenticate a user?

Exercise 52: IAM Policies

Introduction

Well written policies with explicit permissions to resources will supercharge the security of your application and AWS resources. In this exercise, try creating a policy or two using the visual editor and also by hand with JSON.

Study

- *Managed Policies and Inline Policies*
  - https://ryanlewis.link/IAM-managed-and-inline-policies
- *Creating IAM Policies*
  - https://ryanlewis.link/IAM-creating-policies

Tasks

- Create IAM Policies
  - Create a new policy
    - Use the Visual Editor to add permissions
    - Add 2 or 3 actions to an existing resource
  - Create another new policy
    - Use the "JSON" tab

Consider

- What types of values does each property on a policy expect?
- What are the different principals allowed in a policy?

Exercise 53: IAM Roles

Introduction

In this exercise, you will create a role, which is what is used to give permissions to AWS resources. Misconfigured roles are one of the most common issues when troubleshooting resource interaction bugs.

Study

- *IAM Roles*
  - https://ryanlewis.link/IAM-roles
- *Using Instance Profiles*
  - https://ryanlewis.link/IAM-instance-profiles

Tasks

- Create and use an IAM role
  - Create a new IAM role
    - This will be used with an EC2 instance
    - Attach 2 or 3 policies, including one you made
- Try launching an EC2 instance and attaching the role
  - Use the AMI you created in an earlier exercise

Consider

- What are the differences between users and roles?
- Why do EC2 instances need an instance profile?

Exercise 54: Services and IAM

Introduction

In this exercise, you'll create a service user and configure the application to use the user's access key. This method is used to allow access to your AWS account from an external application. You will often encounter this typical pattern in the wild; it's a reliable way to manage third-party service application permissions.

Study

- *Understanding How IAM Works*
  - https://ryanlewis.link/IAM-intro-structure
- *Managing Access Keys for IAM Users*
  - https://ryanlewis.link/IAM-access-keys

Tasks

- Create a service user to give permissions to an application
  - Create a new IAM user
    - Name it "service"
    - Save the access key
  - Add permissions to the user to perform an action
  - Write a simple program that uses the AWS SDK
    - Configure the access key in your local AWS CLI
    - Perform the action that the service user has permissions for in code
  - Try running the program and ensure it functions correctly

Consider

- Are there other ways you can give a service permission to access your AWS resources?
- What services or applications have you used that might need AWS access managed by IAM?

Exercise 55: IAM Access Advisor

Introduction

Even if the responsibility doesn't fall to you, understanding how to audit your users' access to your resources is vital for overall account security. In this exercise, you'll take a look at how to view recently used permissions with IAM Access Advisor.

Study

- *Refining Permissions in AWS Using Last Access Information*
  - https://ryanlewis.link/IAM-last-access-info
- *Viewing Last Accessed Information for IAM*
  - https://ryanlewis.link/IAM-last-access-info-view
- *Example Scenarios for Using Last Accessed Information*
  - https://ryanlewis.link/IAM-last-access-info-example

Tasks

- Use IAM Access Advisor to review permissions
  - Access the user you're logging in as in the IAM dashboard
  - Look at the "Access Advisor" tab
  - Observe what permissions were used most recently and when

Consider

- How does IAM Access Advisor help you apply the Principle of Least Privilege?
- What other services could you use to get more granular information on permissions than IAM Access Advisor?


Serverless

Exercise 56: Serverless on AWS

Introduction

In this exercise, you'll take a look at the Serverless Application Repository in AWS and see what kinds of applications are available. As an architecture, serverless can be a little daunting to get started with, so having a head start with pre-built applications available in SAR is a huge help.

Study

- *Serverless on AWS*
  - https://ryanlewis.link/AWS-serverless
- *Serverless on AWS Deep Dive*
  - https://ryanlewis.link/AWS-serverless-deep-dive
- *What is the Serverless Application Repository?*
  - https://ryanlewis.link/AWS-serverless-app-repository

Tasks

- Investigate the Serverless Application Repository
  - Look at the available public applications
  - Try launching one

Consider

- What does the word "Serverless" really mean?
- What are some of the other services in AWS that fit in serverless architecture?

Exercise 57: Creating a Lambda Function

Introduction

When AWS released Lambda, serverless really came into its own. You'll use Lambda a ton in AWS for all sorts of things. In this exercise, you'll create a Lambda function and see how to test it through the AWS Management Console.

Study

- *What is AWS Lambda?*
  - https://ryanlewis.link/AWS-lambda
- *AWS Lambda Features*
  - https://ryanlewis.link/AWS-lambda-features

Tasks

- Create and test a Lambda function
  - Create a new Lambda function
    - Choose a Runtime language you're familiar with
      - Node.js lets you modify the function code through the AWS Management Console
    - Use the default example, or just have the function return a simple response
  - Test the function execution through the function dashboard

Consider

- When would you want to use Lambda instead of EC2 for running code?
- How does scaling with Lambda functions work?

Exercise 58: Lambda Roles

Introduction

Lambda functions are often the most useful when they're invoked by and interact with other AWS resources. To do this, they need the correct permissions in their role, which is what you'll configure in this exercise.

Study

- *AWS Lambda Execution Environment*
  - https://ryanlewis.link/AWS-lambda-execution-environment
- *AWS Lambda Execution Role*
  - https://ryanlewis.link/AWS-lambda-execution-role

Tasks

- Modify your Lambda function to interact with another AWS service
  - Create a new IAM role for the Lambda function
    - Give it permissions to perform an action on a resource
    - E.g. Pull records from a DynamoDB table
    - Change the attached role to the new one
  - Write code in your function to perform the action
    - This will likely use the AWS SDK in your function
    - Many runtimes already include the AWS SDK as a dependency
  - Test the function execution

Consider

- What is the main difference in JSON between a role for an EC2 instance versus a role for a Lambda function?
- For a Lambda in a VPC, what manages the Lambda function's access to the public Internet?

Exercise 59: Lambda Triggers

Introduction

Seeing a web request go from API Gateway to Lambda, back to API Gateway, and back to the client is a feeling that will make every cloud developer warm and fuzzy. It's amazing being able to build this typical flow entirely on serverless technology. In this exercise, you'll build out this flow and see it all work together.

Study

- *Using AWS Lambda with Amazon API Gateway*
  - https://ryanlewis.link/AWS-lambda-api-gateway
- *Build an API Gateway REST API with Lambda integration*
  - https://ryanlewis.link/AWS-lambda-api-gateway-rest

Tasks

- Create and configure a Lambda Trigger
  - Create a REST API in API Gateway
  - Create an endpoint and method that triggers your Lambda function
    - Modify your Lambda code to return the correct response
    - It should return an object with the appropriate properties

Consider

- What are other useful triggers for Lambda functions?
- Why did AWS adopt an event-based model for Lambda invocation?

Exercise 60: Lambda Logging

Introduction

Because you don't have log files lying around for your Lambda functions, it is more important than ever to log to a collection service like CloudWatch properly. In this exercise, you'll give your Lambda permissions to log to CloudWatch and ensure the process is working correctly.

Study

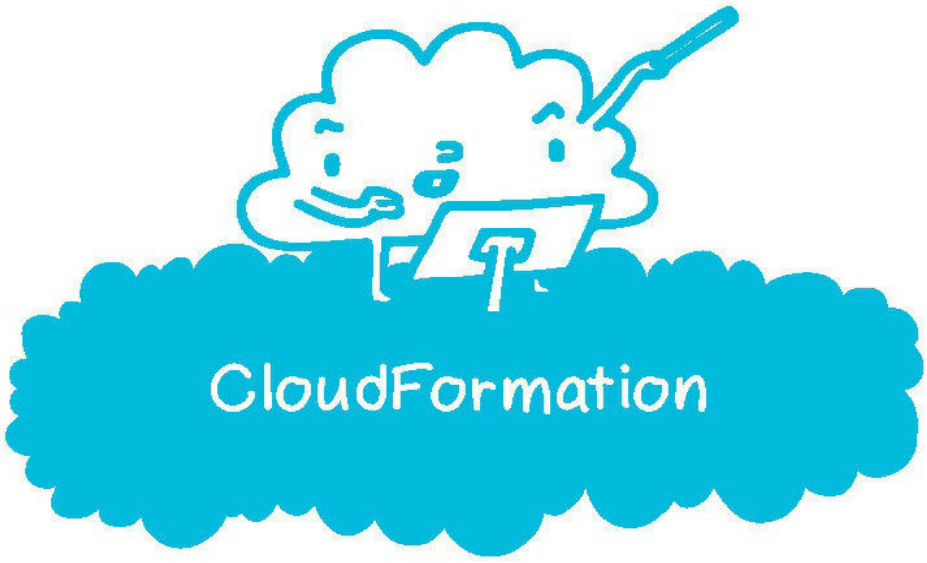
- *AWS Lambda Function Logging in Node.js*
  - https://ryanlewis.link/aws-lambda-nodejs
- *Accessing Amazon CloudWatch logs for AWS Lambda*
  - https://ryanlewis.link/aws-lambda-cloudwatch

Tasks

- Configure your Lambda function to log to CloudWatch Logs
  - Modify your Lambda role to allow for logging to CloudWatch
    - Add the "AWSLambdaBasicExecutionRole" policy to the role
  - Add a log statement to your function code
  - Execute the function and view the logs in CloudWatch Logs
    - Logs can sometimes take up to 30 seconds to appear

Consider

- What other methods, besides logging, does AWS provide to monitor Lambda function execution?
- What are some of the benefits of storing your logs in CloudWatch?


CloudFormation

Exercise 61: AWS CloudFormation

Introduction

Creating resources by hand is okay for experimentation, but learning how to best utilize infrastructure as code is the only way to build maintainable AWS infrastructure. CloudFormation is a built-in AWS service to provision resources from a static template. In this exercise, you'll create a stack using a sample template.

Study

- *What is AWS CloudFormation?*
  - https://ryanlewis.link/AWS-cloud-formation
- *Getting Started with AWS CloudFormation*
  - https://ryanlewis.link/AWS-cloud-formation-user-guide
- *Sample CloudFormation Templates*
  - https://ryanlewis.link/aws-cloud-formation-templates

Tasks

- Create a CloudFormation stack
  - Create a new CloudFormation stack in the CloudFormation dashboard
    - Select a sample template

Consider

- Why should you use CloudFormation instead of configuring infrastructure manually?
- How does CloudFormation keep your resources grouped?

Exercise 62: CloudFormation Templates

Introduction

Writing CloudFormation templates can be pretty confusing initially. Building them from nothing will help you get familiar with the structure and the CloudFormation resource documentation. In this exercise, you'll create a CloudFormation template from scratch.

Study

- *Working with AWS CloudFormation Templates*
  - https://ryanlewis.link/aws-cloud-formation-template-guide
- *CloudFormation Template Reference*
  - https://ryanlewis.link/aws-cloud-formation-template-references

Tasks

- Create a CloudFormation template from scratch
  - Choose JSON or YAML
  - Define a few different types of resources in your template
    - Use the CloudFormation resource documentation
      - https://ryanlewis.link/aws-cf-resource-docs
    - Make sure you're including any required properties for the resource

Consider

- What kind of AWS resources can be created with a CloudFormation template?
- What side effects can occur when keeping your infrastructure in code?

Exercise 63: CloudFormation Stacks

Introduction

In this exercise, you will deploy the template you created in the previous exercise. Don't be discouraged if there are errors. Becoming an expert at CloudFormation template writing takes time, but it will be well worth it by making your infrastructure more reproducible and maintainable.

Study

- *Working with Stacks*
  - https://ryanlewis.link/aws-cloud-formation-stacks
- *AWS CloudFormation Best Practices*
  - https://ryanlewis.link/aws-cloud-formation-best-practices

Tasks

- Deploy your CloudFormation template
  - Deploy the template
    - Use either the AWS Management Console or the AWS CLI
  - Watch the "Events" tab for the created stack in the CloudFormation dashboard

Consider

- When organizing resources into stacks, what are some of the things you should consider for separating them?
- What are some of the other tabs provided in the stack section of the CloudFormation dashboard?

Exercise 64: CloudFormation Parameters

Introduction

With parameters, your CloudFormation templates can be used for more than one application or team. In this exercise, you'll modify your template to accept and use parameters and then learn how to deploy the template with this new option.

Study

- *CloudFormation Parameters*
  - https://ryanlewis.link/aws-cloud-formation-parameters

Tasks

- Use parameters in a CloudFormation template
  - Modify your template for parameters
    - Accept one or more parameters
    - Experiment with AWS-specific parameter types
      - https://ryanlewis.link/aws-cf-params-amazon
    - Use the parameter values with properties in your resources
  - Deploy the modified template
    - Update your existing stack with the AWS CLI or AWS Management Console
    - Provide real values for each parameter when deploying

Consider

- What does using parameters with your templates allow you to do?
- When would you want to use an AWS-specific parameter type?

Exercise 65: CloudFormation Functions

Introduction

You'll often find yourself wanting to add a little logic to your CloudFormation templates, beyond user input values. CloudFormation functions give you the flexibility to dynamically build values for your resources. In this exercise, you'll modify your template to use one or two of these intrinsic functions.

Study

- *CloudFormation Intrinsic Function Reference*
  - https://ryanlewis.link/aws-cloud-formation-intrinsic-function

Tasks

- Use CloudFormation functions in a template
  - Add some intrinsic functions to your template
    - E.g. "GetAtt" or "Join"
    - Use the function reference above
  - Try using parameters references in with a function
  - Deploy the modified template

Consider

- How often should you use functions in your templates?
- Are there any functions missing from CloudFormation that you wish were there?

Exercise 66: CloudFormation Pseudo Parameters

Introduction

One of the last remaining CloudFormation template features is one of my favorites. You'll use pseudo parameters quite a bit when needing to access things like your account ID or region in a template. Modify your template to use some of these and take a look at which AWS values you can access with pseudo parameters.

Study

- *Pseudo Parameters Reference*
  - https://ryanlewis.link/aws-cloud-formation-parameters-pseudo

Tasks

- Use pseudo parameters in a CloudFormation template
  - Modify your template to reference pseudo parameters
    - E.g. AccountId or region
    - You may need to add another resource
  - Deploy the template

Consider

- How can pseudo parameters enable your templates to be more reusable?
- Why did AWS create pseudo parameters when parameters already existed?

Exercise 67: AWS Serverless Application Model

Introduction

When building CloudFormation templates that create serverless resources, such as Lambda functions and API Gateway APIs, you can avoid a lot of boilerplate using the Serverless Application Model (SAM). You'll use that in this exercise to deploy a quick function and API using the SAM CLI.

Study

- *What is the AWS Serverless Application Model (AWS SAM)?*
  - https://ryanlewis.link/aws-serverless-sam-guide
- *AWS Serverless Application Model*
  - https://ryanlewis.link/aws-serverless-sam
- *Getting Started with AWS SAM*
  - https://ryanlewis.link/aws-serverless-sam-getting-started

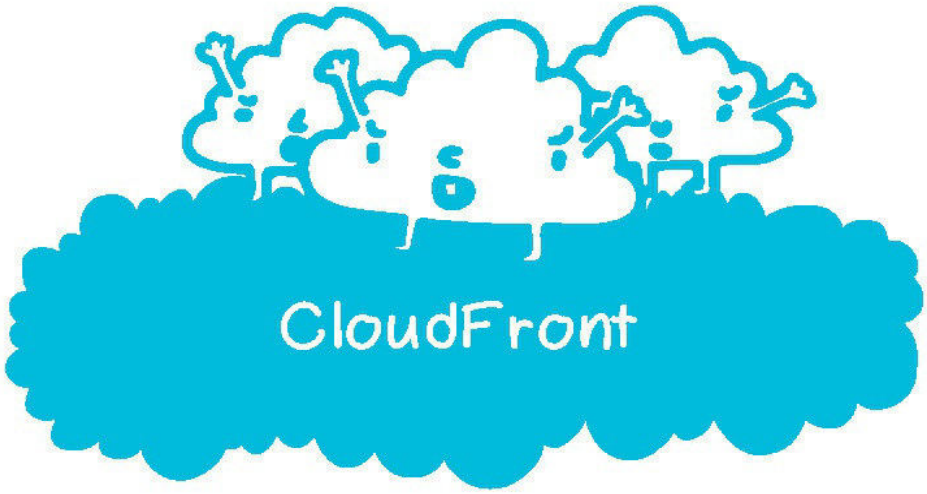
Tasks

- Use the Serverless Application Model to provision resources
  - Write a simple SAM template
    - Include one function and one API
  - Install the SAM CLI
  - Deploy the template with the SAM CLI

Consider

- What benefits does using SAM give you over regular CloudFormation?

- What resource types can you create with a SAM template?


CloudFront

Exercise 68: CloudFront Distributions

Introduction

Speed on the web is a requirement for any application, and nothing is faster for static file delivery from AWS than CloudFront. This CDN service is great to utilize anytime you are serving files. In this exercise, you'll set up a new distribution in front of an S3 bucket and try accessing those files.

Study

- *What is Amazon CloudFront?*
  - https://ryanlewis.link/aws-cloudfront
- *Working with CloudFront Distributions*
  - https://ryanlewis.link/aws-cloudfront-distributions

Tasks

- Use a CloudFront distribution to serve S3 objects
  - Create a new CloudFront web distribution
    - Set the Origin to the S3 bucket you created
- Access a file in the bucket using the CloudFront URL
  - It can take several minutes for the distribution to finish deploying

Consider

- What edge locations would you want to use with CloudFront?
- What types of origins can you configure with a distribution?

Exercise 69: Managing CloudFront

Introduction

Sometimes files cached in CloudFront distributions go stale, and rather than waiting for them to update, you can manually invalidate those files. This process is occasionally necessary when deploying new files to a CloudFront origin. In this exercise, you'll invalidate a file after it has been cached in your CloudFront distribution.

Study

- *Invalidating Files in CloudFront*
  - https://ryanlewis.link/aws-cloudfront-invalidating-files

Tasks

- Update a file in S3 and invalidate the cached version
  - View a file with the CloudFront URL
  - Change the file in the S3 bucket
    - Change the content or name another object to the same name
  - Create a new invalidation for the file in the CloudFront distribution
    - This step can take several minutes to complete
  - View the file again with the CloudFront URL

Consider

- If you don't invalidate files with CloudFront, when would cached files update to a changed version from the origin?
- Is it better to write invalidations for single files or folders? Why?


Elastic Beanstalk

Exercise 70: AWS Elastic Beanstalk

Introduction

To fully deploy an application into the cloud, there are a lot of steps. AWS created services like Elastic Beanstalk to make the process easier for developers. In this exercise, you'll read up on how Elastic Beanstalk works and what kinds of actions it simplifies and takes care of.

Study

- *AWS Elastic Beanstalk*
  - https://ryanlewis.link/aws-elastic-beanstalk
- *What is AWS Elastic Beanstalk?*
  - https://ryanlewis.link/aws-elastic-beanstalk-guide
- *Elastic Beanstalk Concepts*
  - https://ryanlewis.link/aws-elastic-beanstalk-concepts

Tasks

- Read the links above to understand what functions that Elastic Beanstalk manages

Consider

- What platform would you use for your applications with Elastic Beanstalk?
- What deployment and resource provisioning steps does Elastic Beanstalk automate for you?

Exercise 71: Creating an Elastic Beanstalk Application

Introduction

Now you'll create a new Elastic Beanstalk application and deploy a sample application to it. By grouping your deployment resources as applications, Elastic Beanstalk makes it much easier to utilize multiple environments for testing, verification, or blue/green deployments.

Study

- *Elastic Beanstalk Supported Platforms*
  - https://ryanlewis.link/aws-elastic-beanstalk-platforms
- *Deploying Node.js Applications to Elastic Beanstalk*
  - https://ryanlewis.link/aws-elastic-beanstalk-nodejs
- *Creating and Deploying Java Applications on Elastic Beanstalk*
  - https://ryanlewis.link/aws-elastic-beanstalk-java

Tasks

- Deploy a sample application as an Elastic Beanstalk application
  - Create a new application
    - Select the platform of your choice
    - Choose the sample application
    - The new application wizard also creates an environment for you
- When it's finished, try visiting the application via the URL

Consider

- What should you consider when choosing a platform for an Elastic Beanstalk application?
- How can you use the separation of application and environment to test your application?

Exercise 72: Deploying to Elastic Beanstalk

Introduction

Elastic Beanstalk is very convenient, but there are still some rough edges that you might encounter when deploying a real-world application. Knowing the quirks of Elastic Beanstalk and how to troubleshoot an application is a valuable skill to have. In this exercise, you'll deploy a real application to Elastic Beanstalk.

Study

- *Deploying Applications to Elastic Beanstalk Environments*
  - https://ryanlewis.link/aws-elastic-beanstalk-app-deploy
- *Deploy an Elastic Beanstalk Application from the Command Line*
  - https://ryanlewis.link/aws-elastic-beanstalk-cli-deploy
- *Managing Elastic Beanstalk Application Versions*
  - https://ryanlewis.link/aws-elastic-beanstalk-application-versions

Tasks

- Deploy a real application to an Elastic Beanstalk environment
  - Create or find an existing web application that matches the platform you chose
    - You can use my app: EC2 Field Day
      - https://ryanlewis.link/github-ec2-field-day-demo
  - Deploy the application code to your Elastic Beanstalk environment
    - Try using the AWS Management Console first, then the AWS CLI
  - Once it's deployed, try accessing the app with the EB URL

Consider

- How could you automate the Elastic Beanstalk deployment process?
- How does keeping EB Application Versions around help your application be more reliable?

Exercise 73: Elastic Beanstalk Configuration

Introduction

There are many configuration options available for your Elastic Beanstalk environment, from scaling and instance type configuration to load balancing and VPCs. In this exercise, you're going to get some hands-on experience making some configuration changes to your Elastic Beanstalk environment.

Study

- *Configuring Elastic Beanstalk Environments*
  - https://ryanlewis.link/aws-elastic-beanstalk-configure
- *Customizing your Elastic Beanstalk environment using .ebextensions*
  - https://ryanlewis.link/aws-elastic-beanstalk-customize

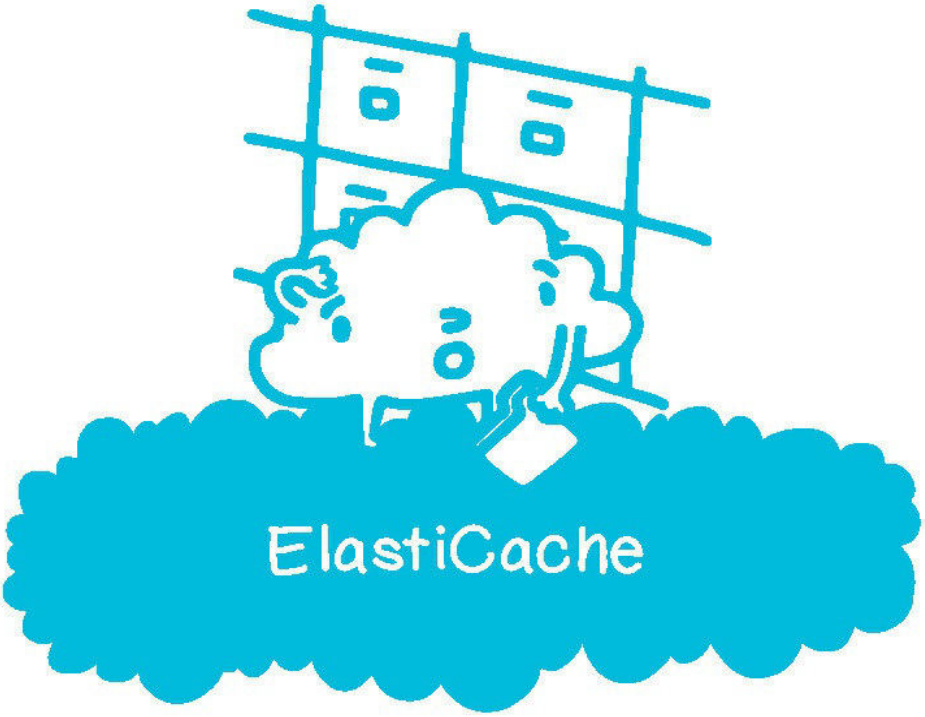
Tasks

- Modify the configuration of your Elastic Beanstalk environment
  - Go to the configuration page of your environment
  - Change one of the configuration properties
    - Example changes:
      - Increase the maximum instances count
      - Modify the instance type
      - Change the scaling rules

Consider

- What types of properties can you configure using an .ebextensions file?

- Do you think Elastic Beanstalk would save you time with deployment and resource provisioning?


ElastiCache

Exercise 74: Amazon ElastiCache

Introduction

While CloudFront is used to speed up web applications over the Internet, ElastiCache is a service that speeds up the internal functionality of applications. Using a managed cache service such as ElastiCache, you can cache session data, database information, and rendered web pages without much administrative overhead. In this exercise, you'll read up on how ElastiCache works and the different engine options that are supported.

Study

- *Amazon ElastiCache Overview*
  - https://ryanlewis.link/aws-elasticache
- *Common ElastiCache Use Cases and How ElastiCache Can Help*
  - https://ryanlewis.link/aws-elasticache-cases
- *Redis vs. Memcached*
  - https://ryanlewis.link/aws-elasticache-redis-vs-memcached
- *Comparing Memcached and Redis*
  - https://ryanlewis.link/aws-elasticache-redis-vs-memcached-compare
- *Amazon ElastiCache for Redis Overview*
  - https://ryanlewis.link/aws-elasticache-redis
- *Amazon ElastiCache for Memcached Overview*
  - https://ryanlewis.link/aws-elasticache-memcached

Tasks

- Read the links above and understand the differences between

Redis and Memcached

Consider

- Which ElastiCache engine will suit your needs the best?
- What parts or functions of your application could you see being enhanced with ElastiCache?

Exercise 75: Redis & ElastiCache

Introduction

Redis is the best pick of ElastiCache engines for most people due to its popularity and ease of use. Configuring a cluster is the first step to improving your application's speed. In this exercise, you'll create a simple Redis cluster with no replicas.

Study

- *What is Amazon ElastiCache for Redis?*
  - https://ryanlewis.link/aws-elasticache-redis-what-is
- *Creating an ElastiCache Cluster*
  - https://ryanlewis.link/aws-elasticache-cluster

Tasks

- Create an ElastiCache Redis cluster
  - Create a new ElastiCache Redis cluster
    - Choose a small node type
      - E.g. t2.micro
    - No replicas
    - Default values for the other options

Consider

- Why does ElastiCache deploy as clusters, even with only one node?
- What are the auto scaling options available for ElastiCache with Redis?

Exercise 76: Connecting to ElastiCache

Introduction

One benefit of Redis's ongoing popularity is that it is widely supported by the open-source community and has libraries available in almost every language. Seeing how you interact with your cache in this exercise will help you understand the right way to improve your application speed.

Study

- *Connecting to a Redis Cluster's Nodes*
  - https://ryanlewis.link/aws-elasticache-redis-cluster-node
- *Access Patterns for Accessing an ElastiCache Cluster in an Amazon VPC*
  - https://ryanlewis.link/aws-cluster-access-patterns

Tasks

- Connect to a Redis cluster in ElastiCache
  - Modify your application code
    - This is the application code running in Elastic Beanstalk
    - Put a value in the Redis cluster cache
    - You may need a third-party library for connecting to Redis
  - Modify the ElastiCache security group
    - Allow incoming connections from port 6379 (the default Redis port)
    - Source: The security group of the EC2 instances in your Elastic Beanstalk environment (you'll need to look this up)
  - Deploy the modified application

- Deploy to Elastic Beanstalk
- Verify the ElastiCache connection works
- You can monitor active connections and records in the ElastiCache dashboard

Consider

- Why does AWS make it challenging to access ElastiCache clusters outside of AWS?
- In a public/private subnet-type VPC, where should you provision an ElastiCache cluster?


Messaging

Exercise 77: AWS Messaging

Introduction

It is important to decouple components as much as you can to make your cloud applications more resilient. Messaging makes that possible, and AWS has quite a few options for managed messaging services. In this exercise, you'll take a look at the available messaging services and see how they compare.

Study

- *AWS Messaging Overview*
  - https://ryanlewis.link/aws-messaging
- *Modern Messaging for Application Architecture*
  - https://ryanlewis.link/messaging-app-architecture
- *Amazon SQS Overview*
  - https://ryanlewis.link/aws-sqs
- *Amazon SNS Overview*
  - https://ryanlewis.link/aws-sns
- *Amazon Kinesis Data Streams Overview*
  - https://ryanlewis.link/aws-kinesis-data-streams

Tasks

- Read the links and make sure you know all the different messaging offerings in AWS

Consider

- What are some of the differences between AWS's messaging offerings?

- Which messaging service would you use the most?

Exercise 78: SQS Queues

Introduction

Simple Queue Service is a basic queuing service that is good for decoupling processing jobs and works with multiple consumers. While not the best for real-time interactions, its simplicity and ease of use with Lambda makes it a tool I use often. In this exercise, you'll create a queue in SQS.

Study

- *How Amazon SQS Works*
  - https://ryanlewis.link/aws-sqs-guide
- *Creating an Amazon SQS Queue*
  - https://ryanlewis.link/aws-sqs-queue

Tasks

- Create a new SQS queue
  - Standard queue
  - Leave other defaults

Consider

- What are some use cases you could use Amazon SQS for?
- How does Amazon SQS help you decouple components of your application?

Exercise 79: SQS Messages

Introduction

Interacting with an SQS queue is simple because there are no ongoing connections needed and your sends and receives are just API calls. In this exercise, you'll get hands-on experience working with an SQS queue in your application.

Study

- *Configuring an SQS Queue Access Policy*
  - https://ryanlewis.link/aws-sqs-queue-access
- *Configuring SQS Queue Parameters*
  - https://ryanlewis.link/aws-sqs-queue-parameters
- *Amazon SQS Visibility Timeout*
  - https://ryanlewis.link/aws-sqs-visibility-timeout

Tasks

- Send and receive messages to an SQS queue
  - Modify your application code or write a new application
- Send SQS messages and receive SQS messages
    - Use the AWS SDK to send and receive messages
  - Update the role used by the EC2 instances in your Elastic Beanstalk environment
    - Add the ability to send and receive messages with the SQS queue
  - Deploy your application
  - Test the new application functionality

Consider

- What type of data can be sent as a message in an SQS queue?
- When would you need to modify the default length of a visibility timeout?

Exercise 80: Kinesis Data Streams

Introduction

For real-time messaging, there is nothing better in AWS than Kinesis Data Streams. From chat applications to decoupled microservices, you'll never run out of tasks for Kinesis Data Streams. In this exercise, you'll set up a new stream with one shard.

Study

- *What is Amazon Kinesis Data Streams?*
  - https://ryanlewis.link/aws-kinesis-data-streams-guide
- *Creating a Kinesis Data Stream*
  - https://ryanlewis.link/aws-kinesis-data-streams-create
- *Amazon Kinesis Data Streams Terminology and Concepts*
  - https://ryanlewis.link/aws-kinesis-data-streams-concepts

Tasks

- Create a new Kinesis Data Stream
  - One open shard

Consider

- How do Kinesis Data Streams differ from SQS queues?
- When would you use Kinesis Data Streams?

Exercise 81: Consuming Kinesis Data

Introduction

Although I prefer using Lambda as a consumer for Kinesis Data Streams, AWS also provides the Kinesis Client Library to simplify writing your own consumers. In this exercise, you'll get some hands-on experience writing a consumer application.

Study

- *Reading Data from Amazon Kinesis Data Streams*
  - https://ryanlewis.link/aws-kinesis-data-streams-data
- *Using the Kinesis Client Library*
  - https://ryanlewis.link/aws-kinesis-data-streams-client-library

Tasks

- Build a Kinesis Consumer Application
  - Write a small application
  - Use the Kinesis Client Library for your language
  - Connect to your stream
  - Try sending some data through your stream and log it from your consumer

Consider

- Why would it help to have multiple consumers connected to the same stream?
- What does the Kinesis Client Library simplify for you?

Exercise 82: SNS Topics

Introduction

Simple Notification Service is the best way to configure notifications for your applications. It also provides different functionality than the other messaging services available due to its push model. In this exercise, you'll create a new topic that can be used to send notifications to.

Study

- *What is Amazon SNS?*
  - https://ryanlewis.link/aws-sns-guide
- *Creating an Amazon SNS Topic*
  - https://ryanlewis.link/aws-sns-create-topic

Tasks

- Create a new SNS topic
  - Standard type
  - No other options needed

Consider

- How does SNS compare to SQS or Kinesis Data Streams?
- When would you use SNS?

Exercise 83: SNS Subscriptions

Introduction

SNS is an extremely versatile service because of the wide variety of protocols available to send notifications. From standard email or SMS notifications to Lambda or HTTP endpoints, SNS is excellent for human or inter-application notification. In this exercise, you'll create a few subscriptions to your topic and try publishing messages to trigger notifications.

Study

- *Subscribing to an Amazon SNS Topic*
  - https://ryanlewis.link/aws-sns-subscribe

Tasks

- Create an SNS subscription and publish messages
  - Subscribe to your topic
    - Create a few subscriptions to your topic
    - Use different protocols for each
  - Publish a message to your topic
    - See what the notifications look like through each protocol

Consider

- What types of protocols are available to be notified by SNS?
- How could SNS be used to decouple components in your applications?

Exercise 84: Amazon MQ

Introduction

Message brokers were a niche tool for large distributed applications, but I've found they're gaining popularity in many smaller applications as well. Amazon MQ follows the ElastiCache pattern and provides a managed solution for popular open-source engines. In this exercise, you'll read up on what Amazon MQ is and how to connect to a broker.

Study

- *What is Amazon MQ?*
  - https://ryanlewis.link/aws-mq
- *How Amazon MQ works*
  - https://ryanlewis.link/aws-mq-guide
- *Creating and Connecting to a RabbitMQ Broker*
  - https://ryanlewis.link/aws-mq-rabbitmq-broker

Tasks

- Read the links above and understand how Amazon MQ differs from other messaging offerings

Consider

- How could you use a message broker in your applications?
- What does Amazon MQ provide that SQS doesn't?


Containers

Exercise 85: Container Concepts

Introduction

One of the most popular strategies for application deployment and management is to use containers. Working with containers can accelerate and simplify your development workflow in many ways. In this exercise, take a look at the AWS services for containers and how each one works.

Study

- *Containers on AWS*
  - https://ryanlewis.link/aws-containers
- *Getting Started with Containers*
  - https://ryanlewis.link/aws-containers-guide

Tasks

- Investigate each AWS service that works with containers
  - What roles do each play?
  - View the list in the middle of this page
    - https://ryanlewis.link/aws-containers

Consider

- How could you use containers for your applications?
- Why is AWS creating so many services for container management?

Exercise 86: Elastic Container Registry

Introduction

In this exercise, you'll create a Docker image and push it to Elastic Container Registry (ECR). ECR is where images must be stored to be deployed to ECS. This service's built-in security and simplicity make it the best option for working with containers in AWS.

Study

- *What is Amazon Elastic Container Registry?*
  - https://ryanlewis.link/aws-ecr
- *Getting Started with Amazon ECR Using the AWS CLI*
  - https://ryanlewis.link/aws-ecr-cli
- *Docker Basics for Amazon ECS*
  - https://ryanlewis.link/aws-ecr-docker-basics

Tasks

- Create and push a Docker image to ECR
  - Install Docker locally
  - Build an image with an application (e.g. EC2 Field Day)
  - Create a new repository in ECR
  - Log in to ECR in your command line
  - Push the image to the ECR repository with Docker

Consider

- Why are cloud services like ECR used to host container images?
- What features does ECR provide that makes image management

easier?

Exercise 87: Elastic Container Service

Introduction

There are many new terms and concepts with Elastic Container Service, such as clusters, services, and tasks. Understanding them is essential to working effectively in ECS. In this exercise, you'll read up on ECS, the terms, and how to work with it.

Study

- *What is Amazon Elastic Container Service?*
  - https://ryanlewis.link/aws-ecr-guide
- *Getting Started with Amazon ECS*
  - https://ryanlewis.link/aws-ecs-guide
- *Common Use Cases in Amazon ECS*
  - https://ryanlewis.link/aws-ecs-cases

Tasks

- Read the links above and understand what each ECS component does

Consider

- Why would you want to use ECS instead of deploying applications on your EC2 instances?
- How does ECS simplify management and monitoring of tasks when scaling?

Exercise 88: Launching a Task

Introduction

In this exercise, you'll use the ECS wizard to deploy the image you pushed to ECR. The wizard will create all the components needed to get your application running, which is very helpful considering all the steps required to set up a service in ECS manually.

Study

- *Amazon ECS Task Definitions*
  - https://ryanlewis.link/aws-ecs-tasks
- *Getting Started with Amazon ECS using Fargate*
  - https://ryanlewis.link/aws-ecs-fargate

Tasks

- Deploy your Docker image to ECS
  - Use the Getting Started wizard in ECS
  - Launch a cluster, service, and task
  - Use the image you pushed to ECR

Consider

- How does Fargate work, and how does it simplify the deployment of your images?
- What is a downside to using ECS and containers for your applications?

Exercise 89: Auto Scaling Tasks

Introduction

Cloud applications would not be exceptional without the ability to load balance and auto scale. In this exercise, you'll create a scaling policy for your ECS service. Because ECS is a newer service than EC2, the built-in scaling functionality is more robust and easier to use.

Study

- *Service Auto Scaling*
  - https://ryanlewis.link/aws-ecs-service-auto-scaling
- *Service Load Balancing*
  - https://ryanlewis.link/aws-ecs-service-load-balancing

Tasks

- Configure your service to scale automatically
  - Set a scaling policy
  - Cause the service to scale by exceeding the policy metric threshold

Consider

- How does auto scaling with ECS differ from auto scaling with Elastic Beanstalk?
- How would you find the best metric to use with a scaling policy for your service?

Exercise 90: AWS Fargate

Introduction

Fargate is one of AWS's newest innovations in the serverless space and makes it super simple to provision compute resources for your ECS tasks. You used Fargate when setting up your ECS service, so here you'll read more on Fargate and look at what the ECS dashboard provides in this exercise.

Study

- *What is AWS Fargate?*
  - https://ryanlewis.link/aws-ecs-fargate-guide
- *Amazon ECS on AWS Fargate*
  - https://ryanlewis.link/aws-ecs-fargate-dev-guide
- *Amazon ECS Launch Types*
  - https://ryanlewis.link/aws-ecs-launch-types
- Amazon ECS CloudWatch Metrics
  - https://ryanlewis.link/aws-ecs-cloudwatch-metrics

Tasks

- Investigate the metrics available in the ECS cluster dashboard

Consider

- How does AWS Fargate compare to EC2 instances as a launch type for ECS?
- How could you use the metrics available in the ECS dashboard to monitor your applications?

Exercise 91: Elastic Kubernetes Service

Introduction

Kubernetes can be a complicated technology to understand fully, especially when coming from the software development world. Still, its rising popularity in the application deployment and management space makes it necessary to have it on your radar. In this exercise, you'll read up on what Kubernetes is, what it's used for, and how to use it on AWS.

Study

- *What is Amazon EKS?*
  - https://ryanlewis.link/aws-eks
- *Kubernetes at Wikipedia*
  - https://ryanlewis.link/kubernetes
- *What is Kubernetes?*
  - https://ryanlewis.link/kubernetes-overview

Task

- Find other videos and explanations of what Kubernetes is and how it works

Consider

- How does Kubernetes differ from Docker?
- Is Kubernetes being used at your company or in your organization?


Certification

Exercise 92: AWS Cloud Practitioner Exam

Introduction

I've found that AWS Certification exams are a great way to help you personally measure your cloud development knowledge, even if certifications aren't as popular in software development as they used to be. The Cloud Practitioner exam is the most straightforward exam available. You should feel confident in your abilities to pass it at this point, so take a look at the exam materials and prep in this exercise.

Study

- *AWS Certified Cloud Practitioner Exam Guide*
  - https://ryanlewis.link/aws-ccp-exam-guide

Tasks

- Review the Cloud Practitioner exam prep materials
  - Make a list of the exam contents and domain concepts
    - Find that in the exam guide above
  - Rate each on 1–5 as to how well you understand each one
  - Download all materials for the exam on AWS's prep site
    - https://ryanlewis.link/aws-certification-prep

Consider

- What does the Cloud Practitioner Certificate demonstrate your knowledge of?
- How could the Cloud Practitioner Certificate help you get a new job or a raise?

Exercise 93: Studying for the Exam

Introduction

Although you can't be sure of the exam's exact questions, studying the topics you aren't familiar with might just make the difference between a pass or fail. In this exercise, use your topic ratings from the last exercise to do some studying.

Study

- Independent research

Tasks

- Study for the Cloud Practitioner exam
  - Research the topics on the exam that you rated the lowest from the previous exercise
  - Try to study and feel good about each one before proceeding

Consider

- What exam topic is the most difficult for you?
- Why does AWS offer a foundational certification like the Cloud Practitioner?

Exercise 94: AWS White Papers

Introduction

The AWS white papers recommended for the Cloud Practitioner exam are pretty basic, but because of the breadth of AWS, there may be a few things you've missed in there. In this exercise, look through each white paper below and read about any topic or service you are unaware of.

Study

- Read the below white papers

Tasks

- Download the AWS white papers for the Cloud Practitioner Exam
  - Skim through each, reading only what you don't know
  - *Overview of Amazon Web Services*
    - http://ryanlewis.link/aws-overview-whitepaper
  - *How AWS Pricing Works*
    - https://ryanlewis.link/aws-pricing-whitepaper
- *Review the different AWS Support Plans*
  - https://ryanlewis.link/aws-support-plans

Consider

- Did you learn anything new from reading through the "*Overview* " white paper?
- What do the AWS support plans tell you about getting support from AWS?

Exercise 95: AWS Practice Exams

Introduction

In this exercise, you'll answer the Cloud Practitioner Exam's sample questions and potentially take a practice exam. Although they aren't as long as the real thing, the AWS practice exams show you how the actual exam questions will be structured. Because the structure of the questions in AWS exams can contain nuance, this will help you avoid getting stuck when you're attempting the real exam.

Study

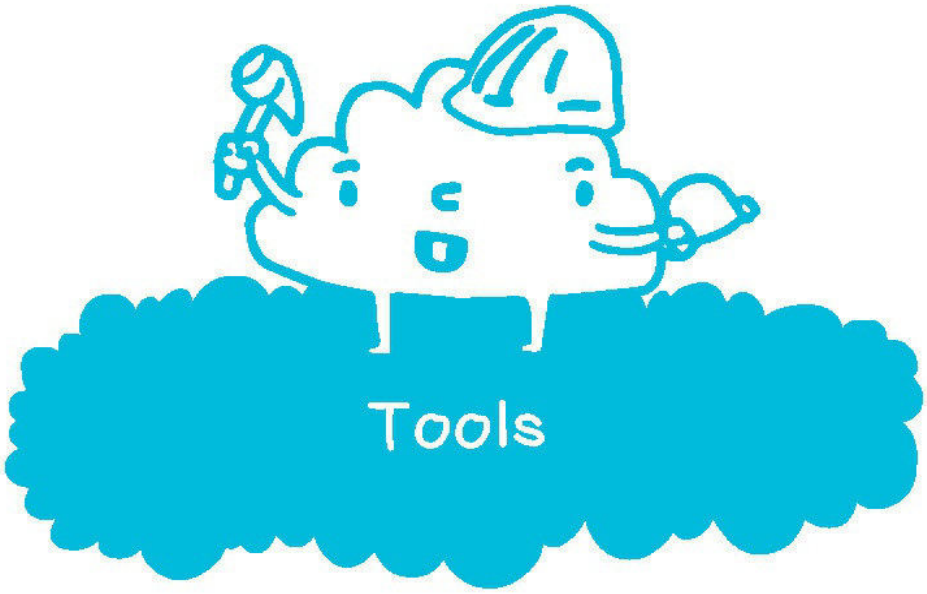
- Research any of the questions you missed from the sample questions or practice exam

Tasks

- *Answer the sample questions*
  - https://ryanlewis.link/aws-ccp-sample-questions
  - Don't look up any of the answers
  - Score yourself
- Take a practice exam
  - These are online and cost money
  - Copy each question to study later

Consider

- Do you feel more prepared to take the exam now than before you reviewed the exam materials?
- Do you think a multiple-choice exam is a good way to measure AWS knowledge?


Tools

Exercise 96: Serverless Framework

Introduction

These last few exercises will introduce you to some additional tools to make your AWS work easier.

Are you working with Lambda in AWS? Then the Serverless Framework is about to be your new best friend. It simplifies the development, deployment, and management of serverless applications. I use it for everything I can. In this exercise, you'll use it to develop and deploy a simple application.

Study

- *Serverless Framework*
  - https://ryanlewis.link/serverless
- *Getting Started*
  - https://ryanlewis.link/serverless-guide
- *Hello World Serverless Example*
  - https://ryanlewis.link/serverless-hello-world

Tasks

- Try out the Serverless Framework
  - Install the Serverless Framework
  - Create a new serverless project
  - Write and configure a Lambda function
  - Configure an API
  - Deploy the application to AWS with the Serverless Framework

Consider

- Why was the Serverless Framework created?
- When would you want to use the Serverless Framework in your workflow?

Exercise 97: Terraform

Introduction

Terraform is an alternative to provisioning AWS resources with CloudFormation. It also follows the “Infrastructure as Code” principle. And you can use the same Terraform format for multi-cloud projects since it works with multiple cloud providers. In this exercise, you’ll convert your CloudFormation template into Terraform configuration and deploy it.

Study

- *Terraform AWS Getting Started*
  - https://ryanlewis.link/terraform
- *Write Terraform Configuration*
  - https://ryanlewis.link/terraform-configuration

Tasks

- Try out Terraform
  - Download the Terraform CLI
  - Rebuild the CloudFormation template you wrote previously as Terraform configuration
  - Initialize Terraform
  - Deploy to AWS

Consider

- What are some of the differences between Terraform and CloudFormation?
- Do you prefer the Terraform configuration format or CloudFormation templates?

Exercise 98: AWS Developer Tools

Introduction

AWS provides developer tools that streamline the cloud development process. It is a big focus as they continue to add new services and improve existing ones. In this exercise, take a look at AWS CodeCommit for your source control needs.

Study

- *Developer Tools on AWS*
  - https://ryanlewis.link/aws-developer-tools
- *What is AWS CodeCommit?*
  - https://ryanlewis.link/aws-code-commit
- *Connect to an AWS CodeCommit Repository*
  - https://ryanlewis.link/aws-code-commit-repository

Tasks

- Use AWS CodeCommit
  - Create a new CodeCommit repository
  - Configure your local environment to be able to push commits to it
  - Commit a code project and push it up

Consider

- How does AWS CodeCommit compare to something like GitHub?
- What are some reasons you would want to use AWS CodeCommit?

Exercise 99: AWS CodePipeline

Introduction

Setting up a pipeline in CodePipeline automates your deployment workflow and can shorten the time to get your code into production. In this exercise, you'll create a new pipeline connected to the CodeCommit repository from the previous exercise and see how the build process works.

Study

- *What is AWS CodePipeline?*
  - https://ryanlewis.link/aws-code-pipeline
- *Getting Started with CodePipeline*
  - https://ryanlewis.link/aws-code-pipeline-guide

Tasks

- Use AWS CodePipeline
  - Create a new pipeline
  - Connect the Source to the CodeCommit repo you created
  - Configure a build step
  - Commit to your CodeCommit repo and watch the pipeline run

Consider

- What are some things you might use AWS CodePipeline for?
- If you've used CI/CD tools before, how does CodePipeline compare?

Exercise 100: AWS Amplify

Introduction

Amplify is another example of AWS's commitment to improving the cloud development experience, and there are many different ways to use it. In this exercise, you'll try out the Amplify CLI and see how it can make it easier to deploy front-end or serverless projects.

Study

- *AWS Amplify*
  - https://ryanlewis.link/aws-amplify
- *AWS Amplify Getting Started*
  - https://ryanlewis.link/amplify

Tasks

- Try out AWS Amplify
  - Install the Amplify CLI
  - Initialize Amplify in a serverless or front-end project
  - Deploy to AWS using Amplify
  - Explore the Amplify dashboard for your project

Consider

- How does AWS Amplify differ from the Serverless Framework?
- Would Amplify make any AWS tasks you'd need to perform easier?

What's Next?

Way to go! You completed 100 cloud exercises!

That’s awesome!

So, what now? You probably saw plenty of things while you were learning that you wanted to pursue further.

Here's what I suggest:

Get more Experience

You have a great understanding of many parts of AWS. Pick some services you want to go deeper with, and do some hands-on projects with them. Struggling to get real applications to work has always become the most critical learning moments for me, and I'll bet they'll help you too.

Go for an Exam

If you tried for the Cloud Practitioner exam already, pick another one. My suggestion is to go for the AWS Certified Developer – Associate certification. It's a logical next step after the Cloud Practitioner exam and an accurate measure of a cloud developer’s abilities.

Ask for More AWS Responsibilities at Work

If you work with a team that does things in AWS, try to do more with it at work. There is no substitute for real-world experience with AWS. That is exactly how I learned!

Thank you for using this workbook!

If you have any suggestions or questions, I would love to hear from you! Shoot me an email: ryan@ryanlewis.dev

Join my newsletter at https://ryanlewis.dev to be notified when I release new courses and books.

This is just the beginning of your journey with AWS. Good luck in the future!

— Ryan H. Lewis

Appendix I: Resources Created in This Workbook

This list contains every resource created in this workbook and which exercise it was created in. Although some of these resources do not cost anything, it's best to make sure each resource is deleted to avoid any unintended charges.

- IAM – User [Ex. 2]
- IAM – Group "Admins" [Ex. 2]
- IAM – Policy [Ex. 52]
- IAM – Role [Ex. 53 & 58]
- Budgets – Cost Budget [Ex. 4]
  - It might be better to leave this, to alert you of any charges
- EC2 – Instance [Ex. 5]
- EC2 – Keypair [Ex. 5]
- EC2 – Security Group [Ex. 6]
- EC2 – Elastic IP [Ex. 10]
- EC2 – Target Group [Ex. 11]
- EC2 – Load Balancer [Ex. 11]
- EC2 – AMI [Ex. 13]
- EC2 – Instance Profile [Ex. 53]
  - This is only viewable and removable from the CLI
- S3 – Bucket & Objects [Ex. 20]
- DynamoDB – Table [Ex. 26 & 30]
- RDS – Database [Ex. 32 & 35]
- VPC – VPC [Ex. 39]
- VPC – Subnets [Ex. 40]

- VPC – Internet Gateway [Ex. 41]
- VPC – Route Table [Ex. 42 & 43]
- VPC – NAT Gateway [Ex. 43]
- Route 53 – Hosted Zone [Ex. 46 & 50]
- Route 53 – Record [Ex. 47, 48 & 49]
- Lambda – Function [Ex. 56, 57 & 96]
- API Gateway – API [Ex. 59 & 96]
- CloudWatch – Logs Group [Ex. 60]
- CloudFormation – Stack [Ex. 61, 63, 67 & 96]
- CloudFront – Distribution [Ex. 68]
- Elastic Beanstalk – Application & Environment [Ex. 71]
- ElastiCache – Redis Cluster [Ex. 75]
- SQS – Queue [Ex. 78]
- Kinesis – Data Stream [Ex. 80]
- SNS – Topic [Ex. 82]
- SNS – Subscription [Ex. 83]
- ECR – Repository [Ex. 86]
- ECS – Cluster [Ex. 88]
- ECS – Service [Ex. 88]
- ECS – Task [Ex. 88]
- Terraform – Various resources [Ex. 97]
  - Remove using the Terraform CLI
- CodeCommit – Repository [Ex. 98]
- CodePipeline – Pipeline [Ex. 99]
- CodeBuild – Project [Ex. 99]
- Amplify – App [Ex. 100]

Appendix II: AWS Acronym Decoder

A	Address
ACL	Access Control List
AMI	Amazon Machine Image
API	Application Programming Interface
ASG	Auto Scaling Group
AWS	Amazon Web Services
AZ	Availability Zone
CDN	Content Delivery Network
CI/CD	Continuous Integration and Continuous Deployment
CIDR	Classless Inter-Domain Routing
CLI	Command Line Interface
CNAME	Canonical Name
DB	Database
DNS	Domain Name System
EB	Elastic Beanstalk
EBS	Elastic Block Store
EC2	Elastic Compute Cloud
ECR	Elastic Container Registry
ECS	EC2 Container Service
EKS	Elastic Kubernetes Service
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
IAM	Identity & Access Management
IP	Internet Protocol

JSON	JavaScript Object Notation
MQ	Managed Message Broker Service
MX	Mail Exchanger
NAT	Network Address Translation
RDS	Relational Database Service
REST	Representational State Transfer
S3	Simple Storage Service
SAM	Serverless Application Model
SAR	Serverless Application Repository
SDK	Software Development Kit
SNS	Simple Notification Service
SQS	Simple Queue Service
SSH	Secure Socket Shell
URL	Universal Resource Locator
VPC	Virtual Private Cloud
YAML	Yet Another Markup Language