AWS Certified Solutions Architect - Associate (SAA-C01)

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Go To Part 2

Exam Preparation

Linux Academy

Course Navigation

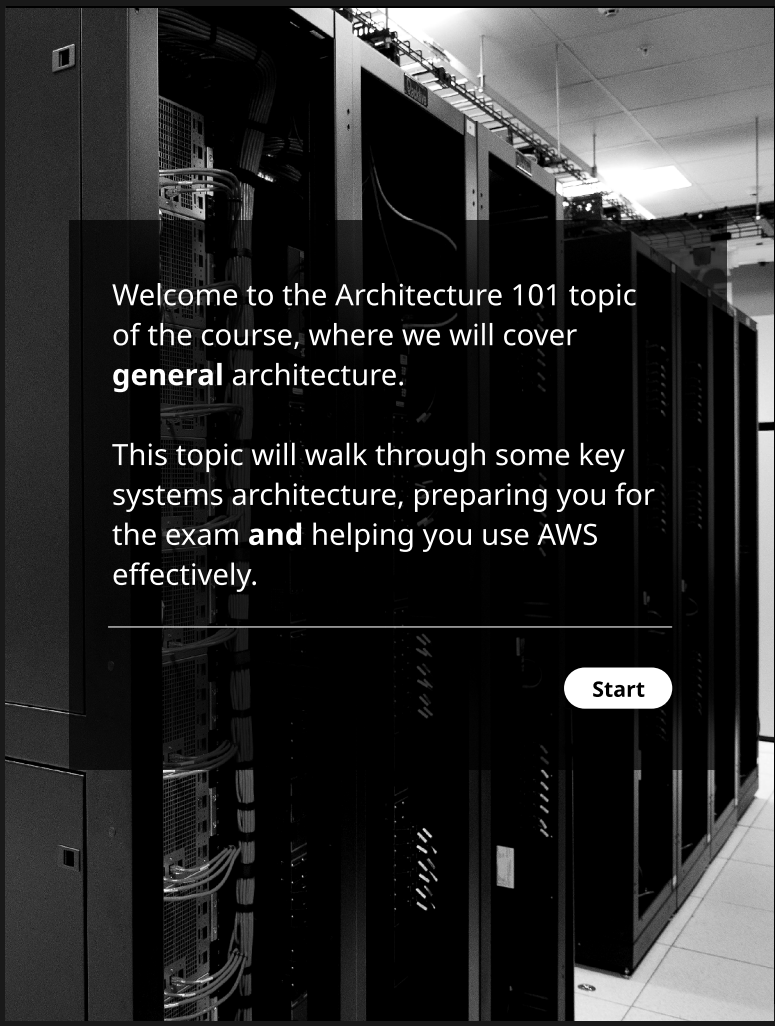
AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



Welcome to the Architecture 101 topic of the course, where we will cover **general** architecture.

This topic will walk through some key systems architecture, preparing you for the exam **and** helping you use AWS effectively.

Start

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Access Management Basics

1. **Principal** — A person or application that can make an **authenticated** or **anonymous** request to perform an action on a system

2. **Authentication** — The process of authenticating a principal against an identity. This could be via username and password or API keys.

3. **Identity** — Objects that require **authentication** and are **authorized** to access **resources**

4. **Authorization** — The process of checking and **allowing** or **denying** access to a resource for an identity

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

AWS Architecture 101

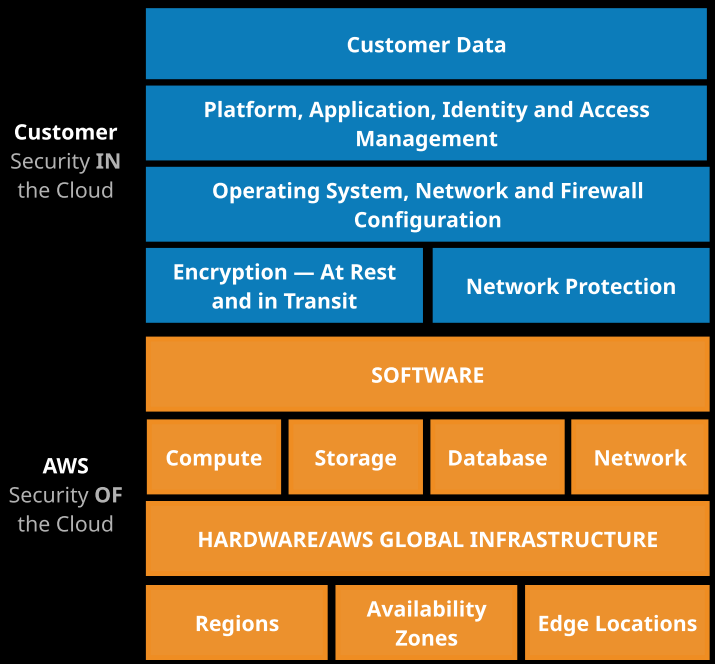
Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Shared Responsibility Model



Back
Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

You

aws

Provider

- Data
- Application
- Runtime
- Operating System (OS)
- Virtualization
- Host/Servers
- Network and Storage
- Data Center

IaaS | PaaS | SaaS

Service models define how a service or product is delivered, **how you pay,** and **what you receive**. They also define which part of the product **you manage** and accept the risks for, as well as which part the **vendor is responsible** for.

Back | Next

Go to Part 2

Back to Main

AWS and SA Fundamentals

Architecture 101

AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

You

Provider

- Data
- Application
- Runtime
- Operating System (OS)
- Virtualization
- Host/Servers
- Network and Storage
- Data Center

IaaS | PaaS | SaaS

Service models define how a service or product is delivered, **how you pay,** and **what you receive**. They also define which part of the product **you manage** and accept the risks for, as well as which part the **vendor is responsible** for.

Back | Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

You

Provider

- Data
- Application
- Runtime
- Operating System (OS)
- Virtualization
- Host/Servers
- Network and Storage
- Data Center

IaaS | PaaS | SaaS

Service models define how a service or product is delivered, **how you pay,** and **what you receive**. They also define which part of the product **you manage** and accept the risks for, as well as which part the **vendor is responsible** for.

Back | Next

Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

You

Provider

- Data
- Application
- Runtime
- Operating System (OS)
- Virtualization
- Host/Servers
- Network and Storage
- Data Center

IaaS | PaaS | SaaS

Service models define how a service or product is delivered, **how you pay,** and **what you receive**. They also define which part of the product **you manage** and accept the risks for, as well as which part the **vendor is responsible** for.

Back | Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

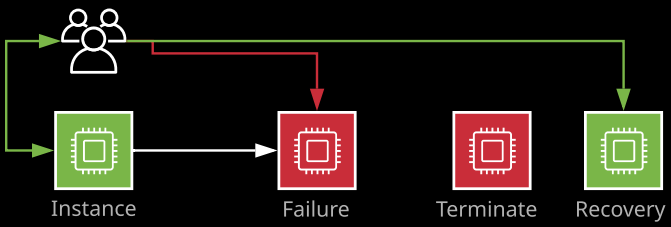
Compute
Section 3

Networking
Section 4

1

High Availability

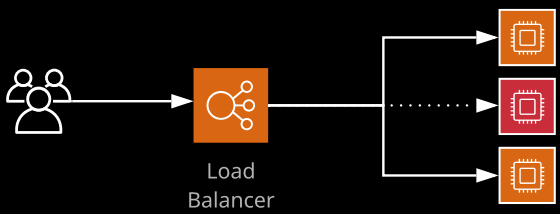
Hardware, software, and configuration allowing a system to **recover quickly** in the event of a failure



2

Fault Tolerance

System designed to **operate through a failure** with **no user impact**. More expensive and complex to achieve.



AWS and SA Fundamentals

Architecture 101

AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

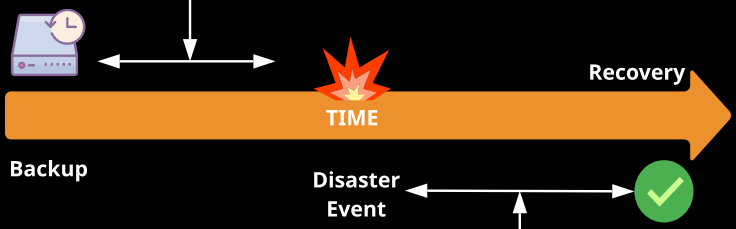
Compute
Section 3

Networking
Section 4

1

Recovery Point Objective (RPO)

How much a business can tolerate to lose, expressed in **time**. The maximum time between a failure and the last successful backup.



2

Recovery Time Objective (RTO)

The maximum amount of time a system can be **down**. How long a solution takes to **recover**.

Back

Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

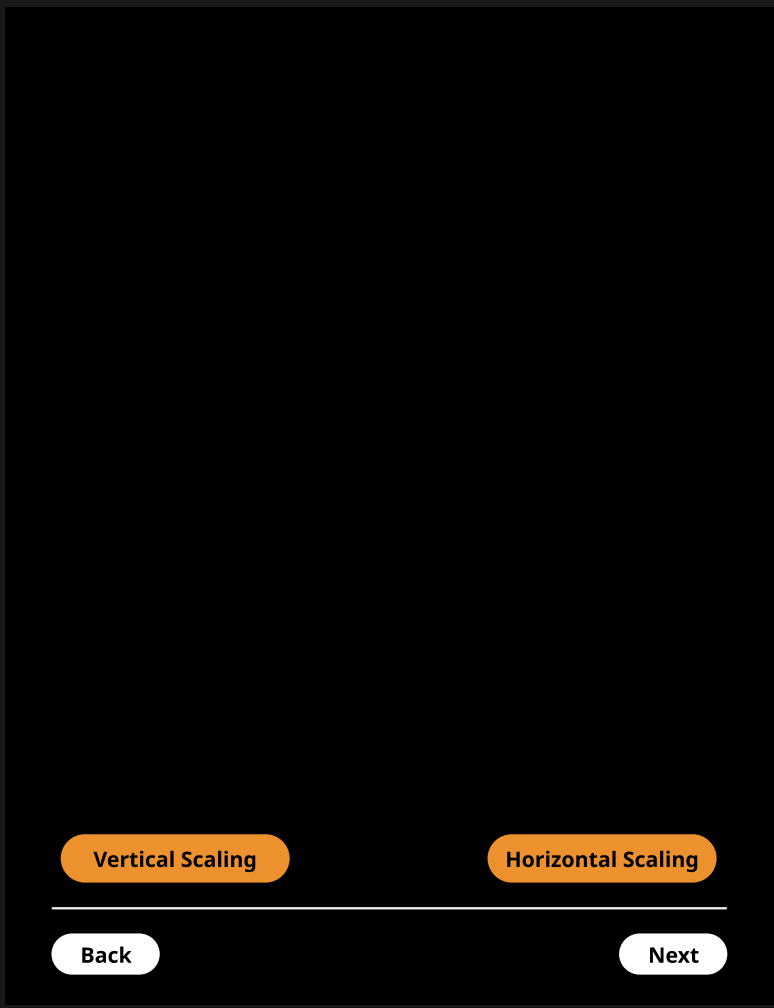
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



Go to Part 2

Back to Main

AWS and SA Fundamentals

Architecture 101

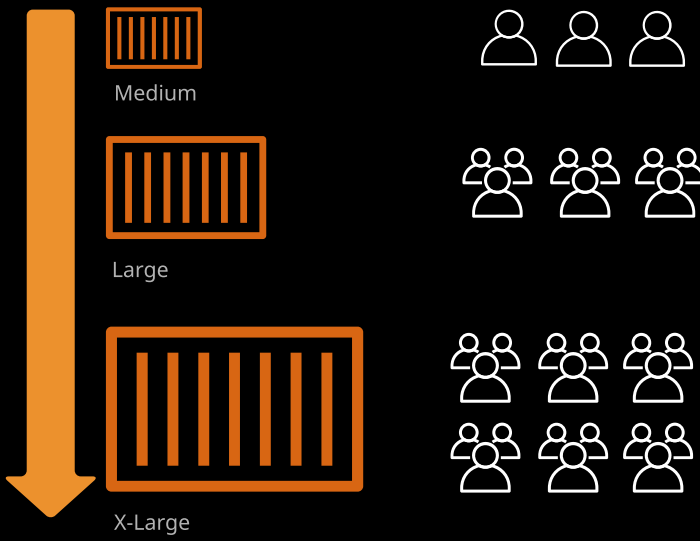
AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



Vertical scaling is achieved by adding additional resources in the form of CPU or memory to an existing machine. By doing so, the machine is able to service additional customers or perform compute tasks quicker. Eventually, **maximum machine sizes** will constrain your ability to scale — either **technically** or from a **cost** perspective.

Go to Part 2

Back to Main

AWS and SA Fundamentals

Architecture 101

AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Horizontal scaling is achieved by adding additional machines into a pool of resources, each of which provide the same service. Horizontal scaling suffers none of the size limitations of vertical scaling and can scale to nearly infinite levels but requires application support to scale effectively.

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

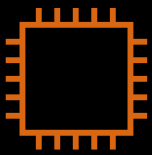
AWS Architecture 101

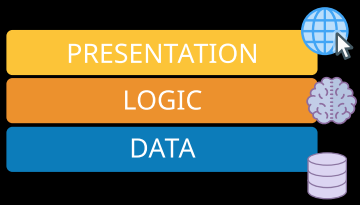
Product Fundamentals

Identity and Access Control
Section 2

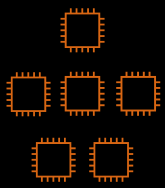
Compute
Section 3

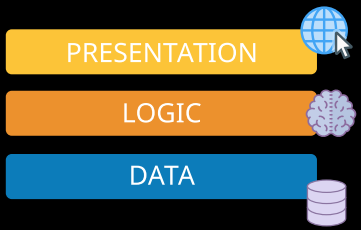
Networking
Section 4





Architecturally, applications consist of three tiers: the **presentation tier**, which interacts with the consumer of the application; the **logic tier**, which delivers the application functionality; and the **data tier**, which controls interaction with a database of some kind. If these tiers are implemented in the same code base and not separated, we refer to it as a **monolithic application**. A monolithic application is hard to scale and generally has to be done vertically.





Applications, if designed correctly, implement the tiers as **isolated components**. Architecturally, these can be provisioned on **separate** machines or pools of machines. As each tier has differing demands on CPU, memory, and disk I/O, it allows each tier's performance to be managed independently.

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Tightly Coupled

Loosely (De)Coupled

Back

Next

Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Architecture 101

AWS Architecture 101

Product Fundamentals

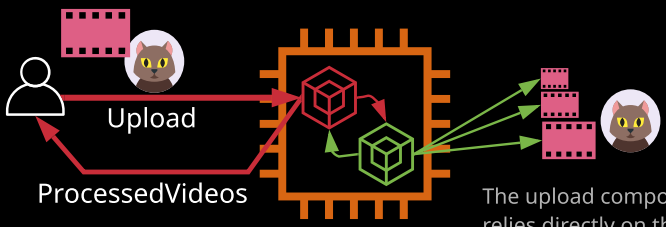
Identity and Access Control
Section 2

Compute
Section 3

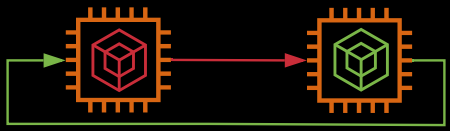
Networking
Section 4

AWS and SA Fundamentals

Architecture 101



The upload component relies directly on the conversion worker component.



In a **tightly coupled** system or architecture, components are not only **directly linked** to each other but also **dependent** on each other. All components share the workload, and the overall system speed is dependent on its slowest part.

A component failure typically means the entire system is impacted, and the system can generally only scale as a single entity.

In this example, media conversion delays could impact the ability to accept uploads.

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

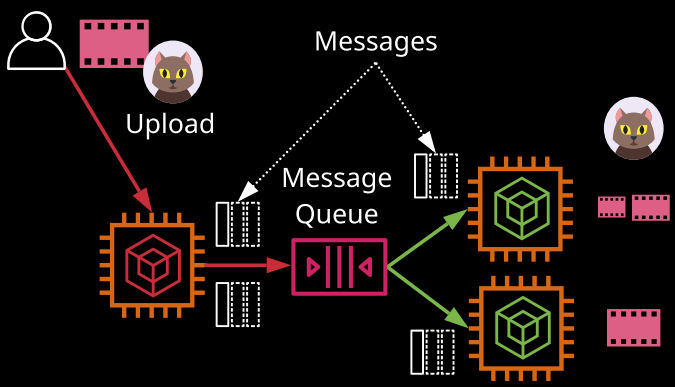
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



In a **loosely** or **decoupled** architecture, each component can **operate independently**. The components communicate using an intermediate entity, such as a **message queue**. This process is asynchronous, meaning messages can be added to and taken from the queue at different rates and/or times.

This allows for failure or scaling of one component without directly impacting others.

Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Architecture 101

AWS Architecture 101

Product Fundamentals

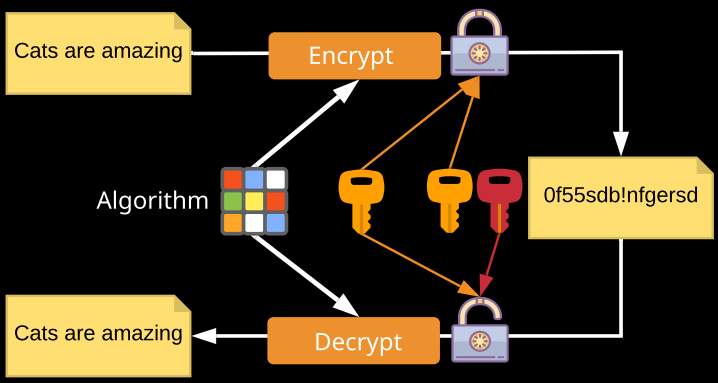
Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Encryption is the process of taking **plaintext** and converting it into **ciphertext**, and converting ciphertext into plaintext. Plaintext and ciphertext can be text, images, or any other data.

Encryption generally uses an **algorithm** and one or more **keys**. It is commonly used to encrypt data **at rest** or **in transit**.



The process can be **symmetrical** (where the **same** key is used for encryption and decryption) or **asymmetrical** (where **different** keys — called **public** and **private** keys — are used).

AWS and SA Fundamentals
Section 1

Architecture 101

AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

There are a number of terms you might hear in this course and the exam that you need to understand at a high level:

Cost efficient or **cost effective**: Implementing a solution within AWS using products or product features that provide the required service for as little initial and ongoing cost as possible. Using your funds effectively and knowing if product X is better or worse than product Y for a given solution.

Secure: In a systems architecture context, implementing a given solution that secures data and operations as much as possible from an internal or external attack.

Application session state: Data that represents what a customer is doing, what they have chosen, or what they have configured. Examples include items and quantities in a shopping cart, notes on an X-ray, and 3D position of a real-time heart scan. Session state can be stored on a server (**stateful** server) or externally to a server (**stateless** server).

Undifferentiated heavy lifting: A part of an application, system, or platform that is not specific to your business. Allowing a vendor (AWS) to handle this part frees your staff to work on adding direct value to your customers.

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

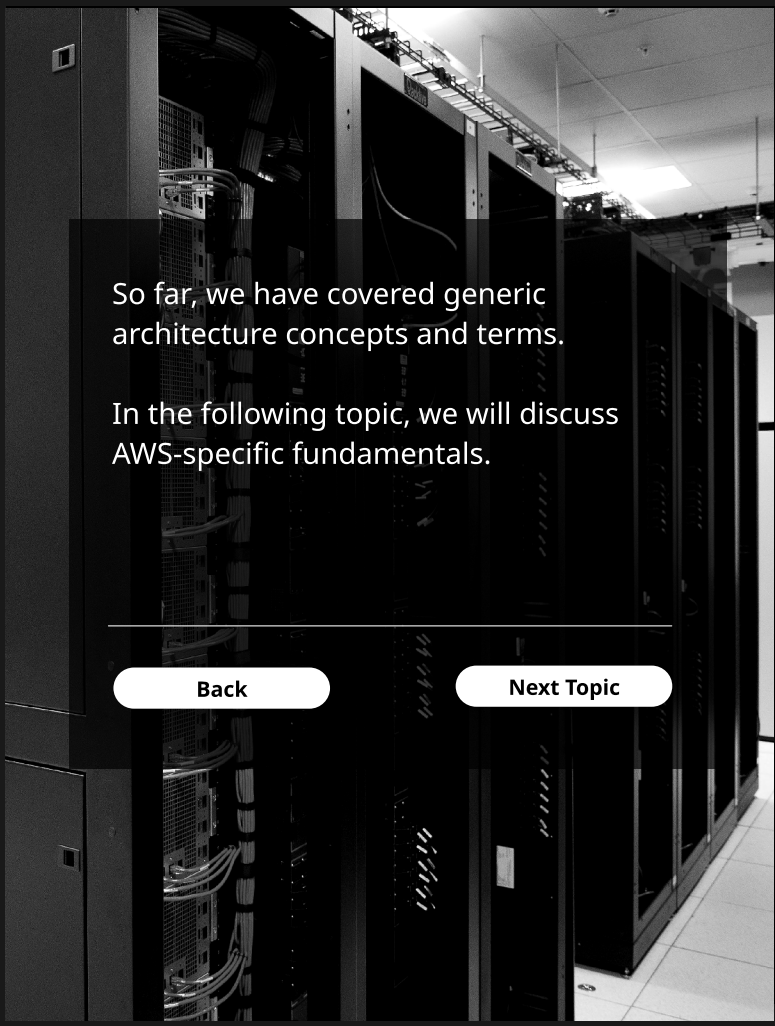
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



So far, we have covered generic architecture concepts and terms.

In the following topic, we will discuss AWS-specific fundamentals.

Back

Next Topic

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

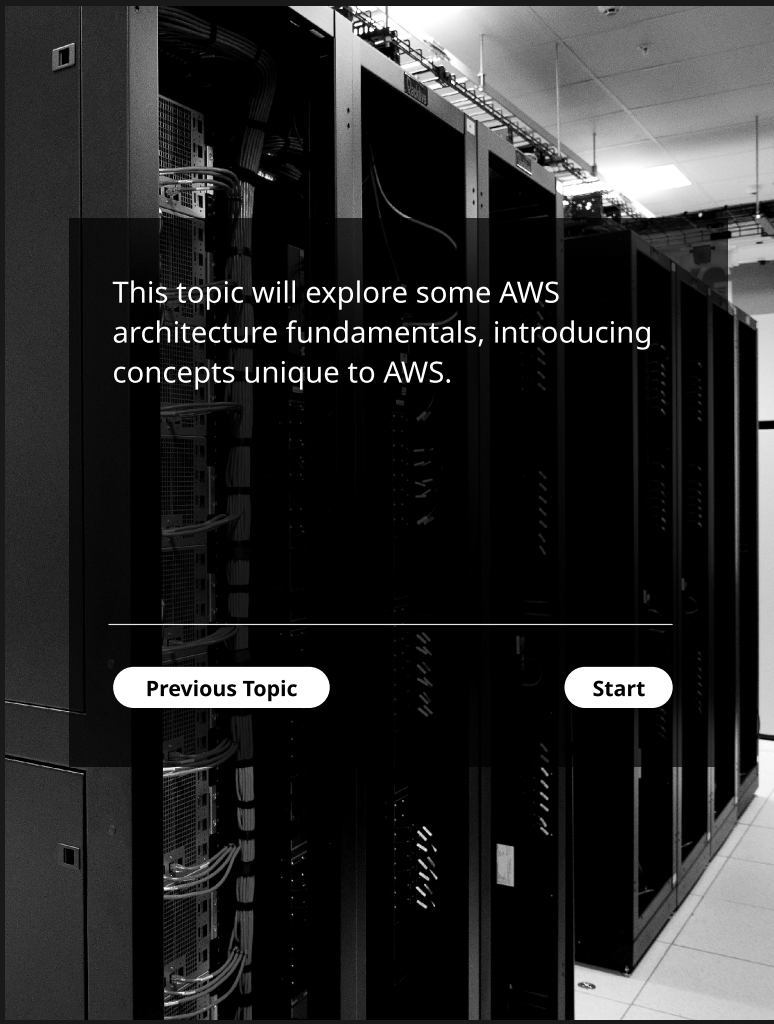
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



This topic will explore some AWS architecture fundamentals, introducing concepts unique to AWS.

Previous Topic

Start

Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Architecture 101

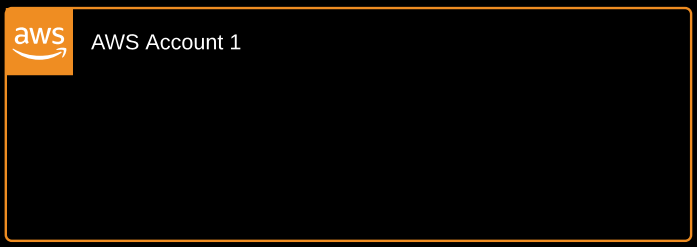
AWS Architecture 101

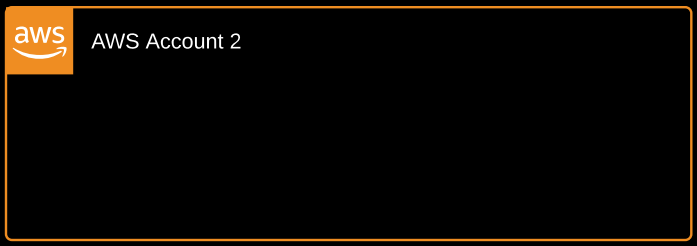
Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4





Authentication

Authorization

Billing

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

AWS accounts are **isolated**. They are created initially with a single **root user**. This user, via its username/password/APIKeys, is the **only identity** that can use (authenticate to) the account. If account credentials are leaked, the impact (blast radius) is limited to that account.

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

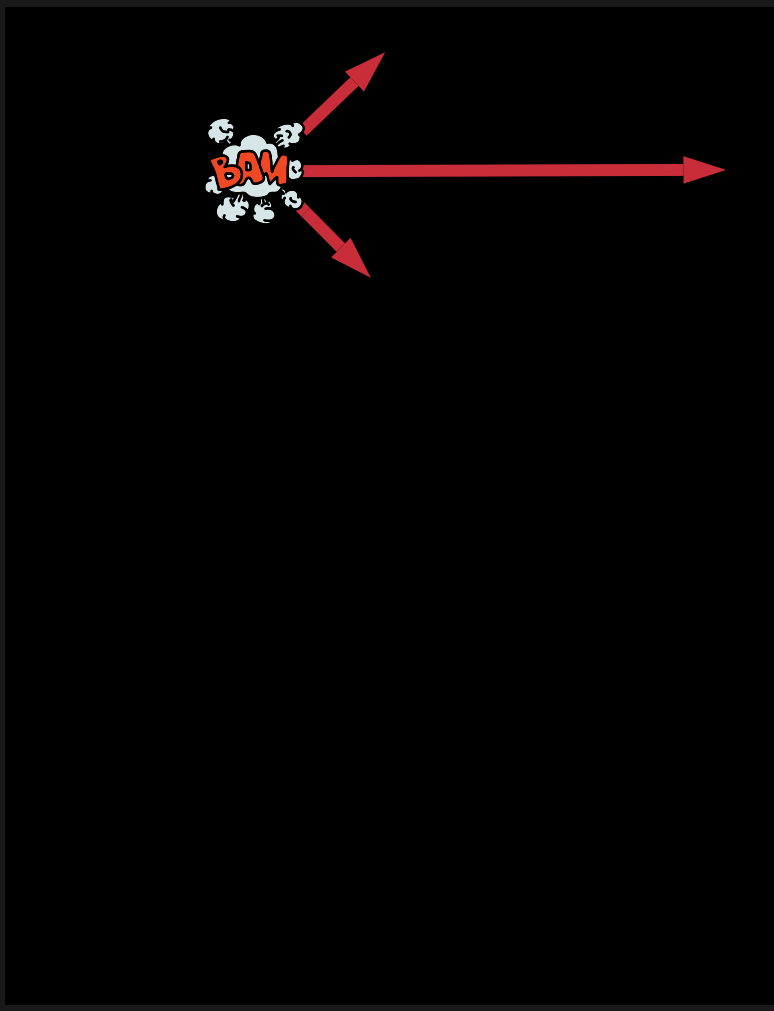
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



Go to Part 2

Back to Main

AWS and SA Fundamentals

AWS Architecture 101

AWS and SA Fundamentals
Section 1

Architecture 101

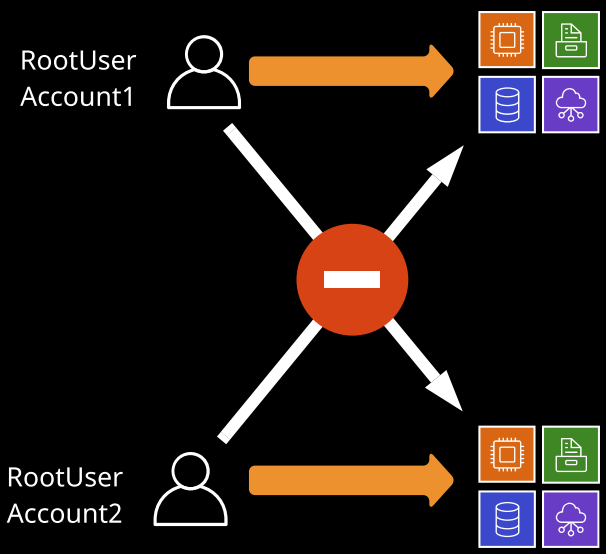
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



Authorization is controlled on a per-account basis. The root user starts with **full control** of the account and its resources. Additional identities can be created or external identities (AWS or otherwise) can be **granted access**. Unless defined otherwise, **no identity** except the account root user has access to resources.

Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Architecture 101

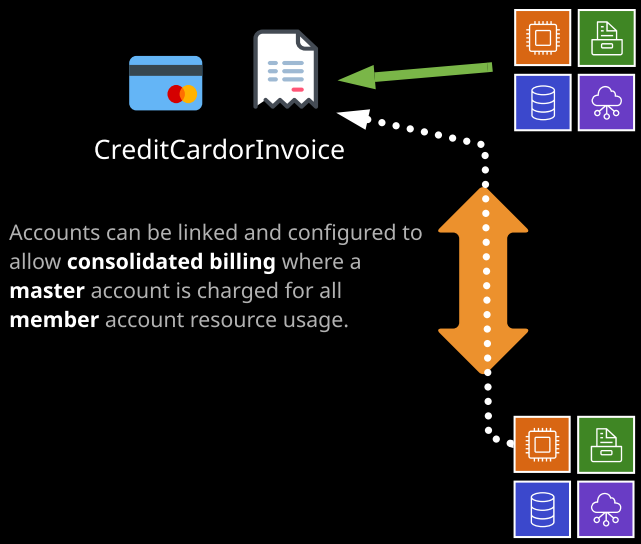
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



Accounts can be linked and configured to allow **consolidated billing** where a **master** account is charged for all **member** account resource usage.

Every AWS account has its own isolated billing information. This is initially in the form of an attached credit card, but established accounts can be converted to use traditional, term-based invoicing. By default, you are only billed for resources in your account. Billing or security exploits are limited to a single account.

Go to Part 2

Back to Main

AWS and SA Fundamentals

AWS Architecture 101

AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

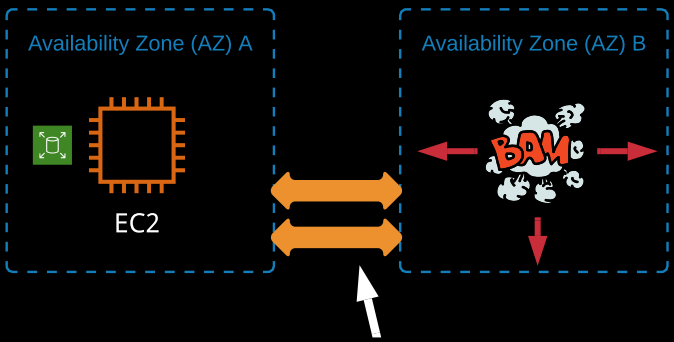
Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

ap-southeast-2 (Sydney)

Regions contain multiple Availability Zones (**AZs**), which are separated and isolated networks. A failure in one AZ generally **won't impact another**.



AZs in the same region are connected with **redundant**, **high-speed**, **low-latency** network connections.

Most AWS services run within AZs. Some series operate from one AZ, while others replicate between AZs. Some services allow you to choose the AZ to use, and some don't.

Edge locations are small pockets of AWS compute, storage, and networking close to major populations and are generally used for **edge computing** and **content delivery**.

Back

Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

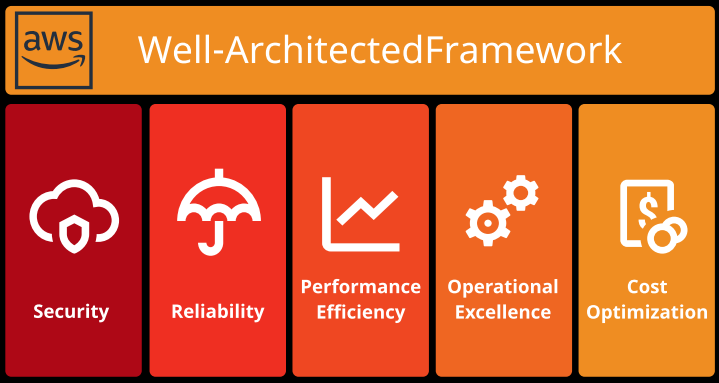
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



Back
Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

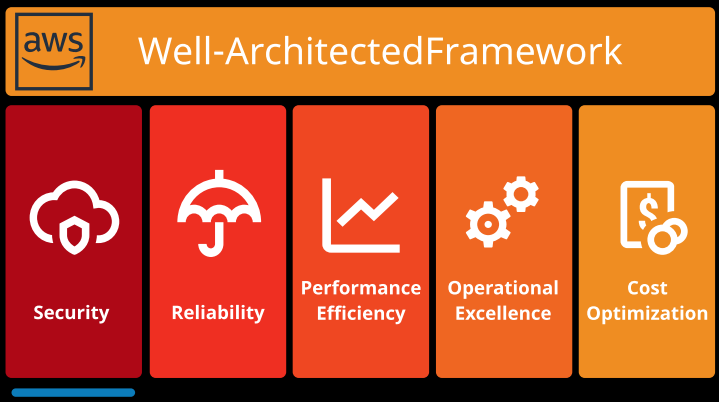
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



The **security pillar** includes the ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.

Design Principles

- Implement a strong identity foundation.
- Enable traceability.
- Apply security at all layers.
- Automate security best practices.
- Protect data in transit and at rest.
- Prepare for security events.

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

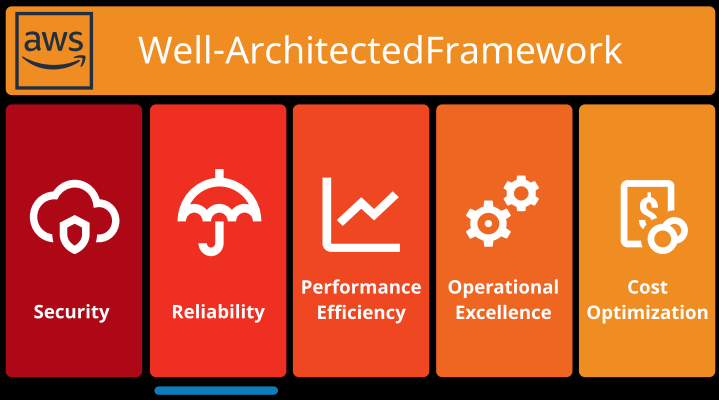
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



The **reliability pillar** includes the ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions, such as misconfigurations or transient network issues.

Design Principles

- Test recovery procedures.
- Automatically recover from failure.
- Scale horizontally to increase aggregate system availability.
- Stop guessing capacity.
- Manage change in automation.

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

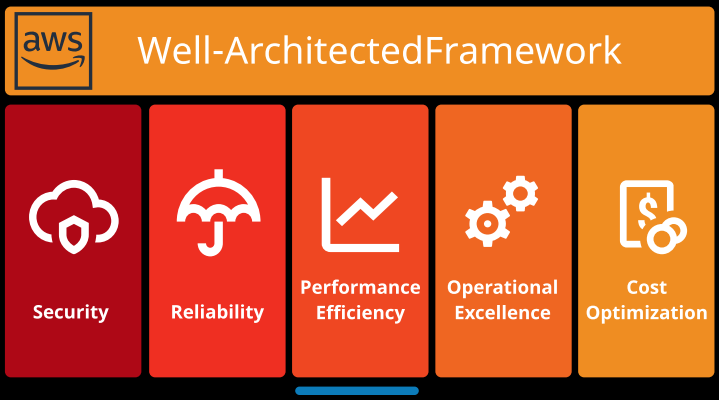
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



The **performance efficiency** pillar includes the ability to use computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve.

Design Principles

- Democratize advanced technologies.
- Go global in minutes.
- Use serverless architectures.
- Experiment more often.
- Mechanical sympathy.

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

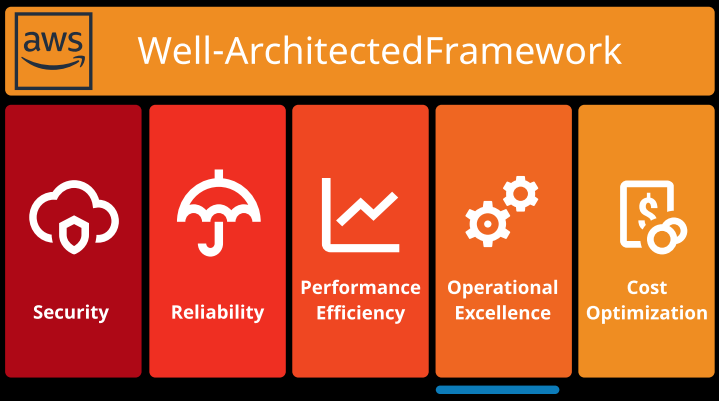
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



The **operational excellence** pillar includes the ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures.

Design Principles

- Perform operations as code.
- Annotate documentation.
- Make frequent, small, reversible changes.
- Refine operations procedures frequently.
- Anticipate failure.
- Learn from all operational failures.

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

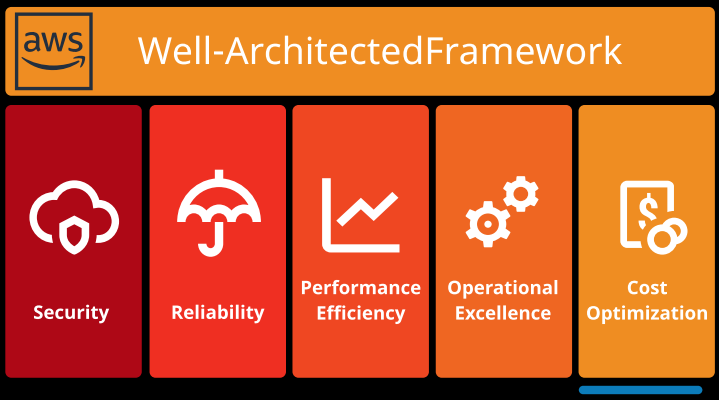
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



The **cost optimization** pillar includes the ability to avoid or eliminate unneeded cost or suboptimal resources.

Design Principles

- Adopt a consumption model.
- Measure overall efficiency.
- Stop spending money on data center operations.
- Analyze and attribute expenditure.
- Use managed services to reduce cost of ownership.

Go to Part 2

Back to Main

AWS and SA Fundamentals

AWS Architecture 101

AWS and SA Fundamentals
Section 1
Architecture 101
AWS Architecture 101
Product Fundamentals
Identity and Access Control
Section 2
Compute
Section 3
Networking
Section 4

Vertical

Horizontal

Elastic

Back

Next

Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Architecture 101

AWS Architecture 101

Product Fundamentals

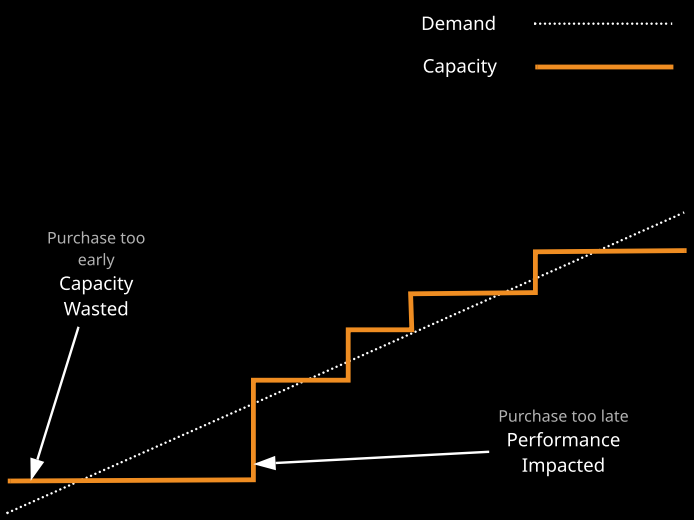
Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

AWS and SA Fundamentals

AWS Architecture 101



Traditional legacy systems use **vertical scaling**. An attempt is made to **forecast** demand and purchase servers **ideally before** the demand passes current capacity. Purchase too **early** and capacity is **wasted**. Purchase too **late** and performance is **impacted**.

Go to Part 2

Back to Main

AWS and SA Fundamentals

AWS Architecture 101

AWS and SA Fundamentals
Section 1

Architecture 101

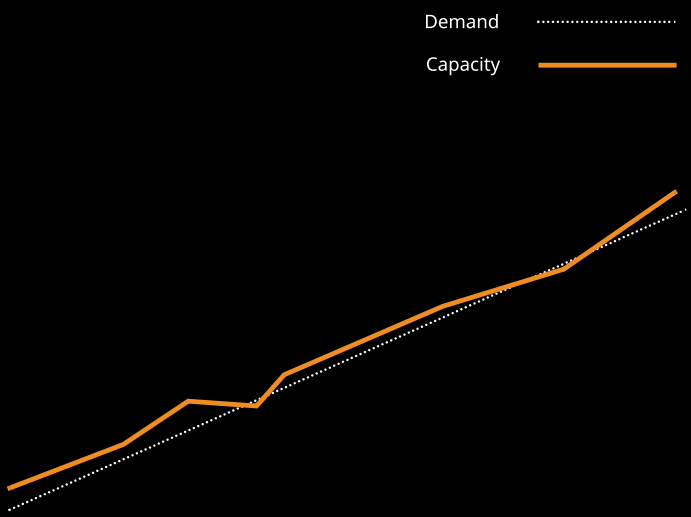
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



When horizontal scaling is used (**more**, **smaller** servers), capacity can be maintained **closer to demand**. There is less waste because servers are smaller and there's less risk of performance impact as each increase is less expensive, so it generally requires less approval.

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

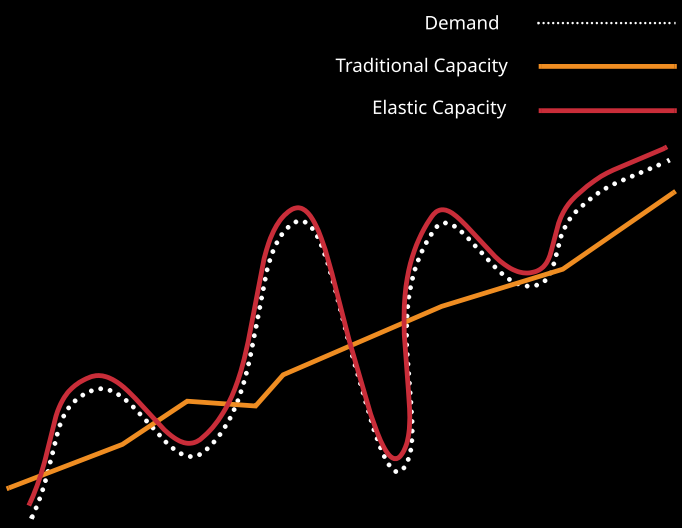
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



Elasticity, or **elastic scaling**, is where **automation** and **horizontal scaling** are used in conjunction to match capacity with demand. Demand is rarely so linear — it can increase or decrease, often in a rapid and sudden way. An efficient platform should scale **OUT** and **IN**, matching demands on that system.

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101

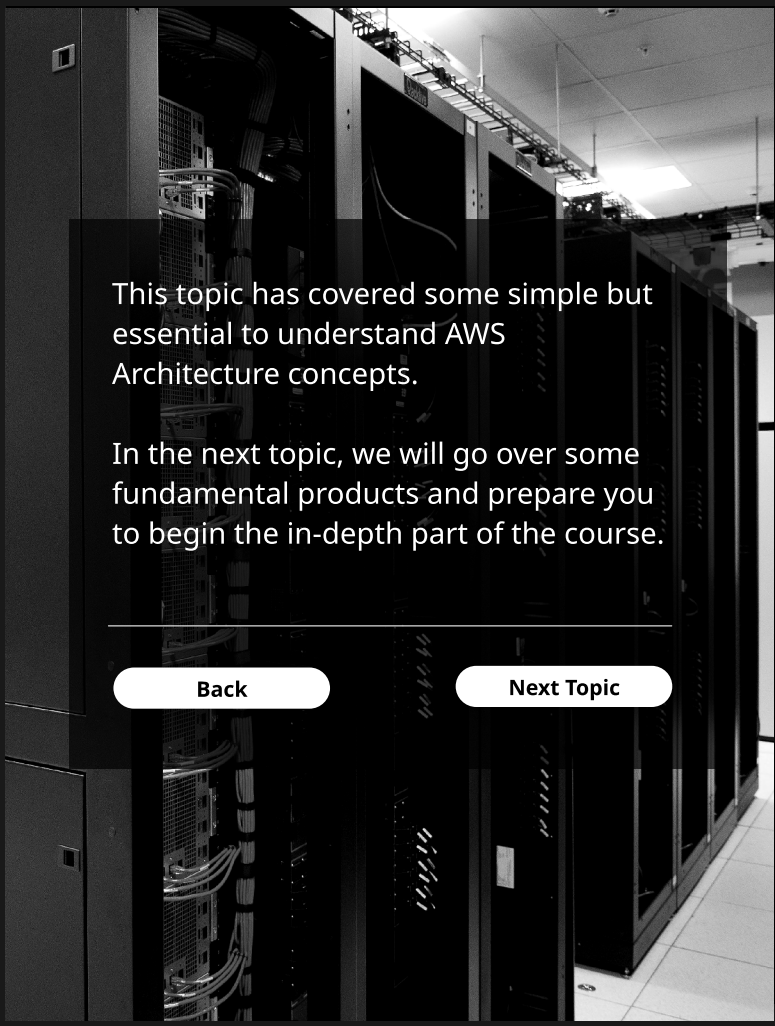
AWS Architecture 101

Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



This topic has covered some simple but essential to understand AWS Architecture concepts.

In the next topic, we will go over some fundamental products and prepare you to begin the in-depth part of the course.

Back

Next Topic

Go to Part 2

Back to Main

AWS and SA Fundamentals

Product Fundamentals

Course Navigation

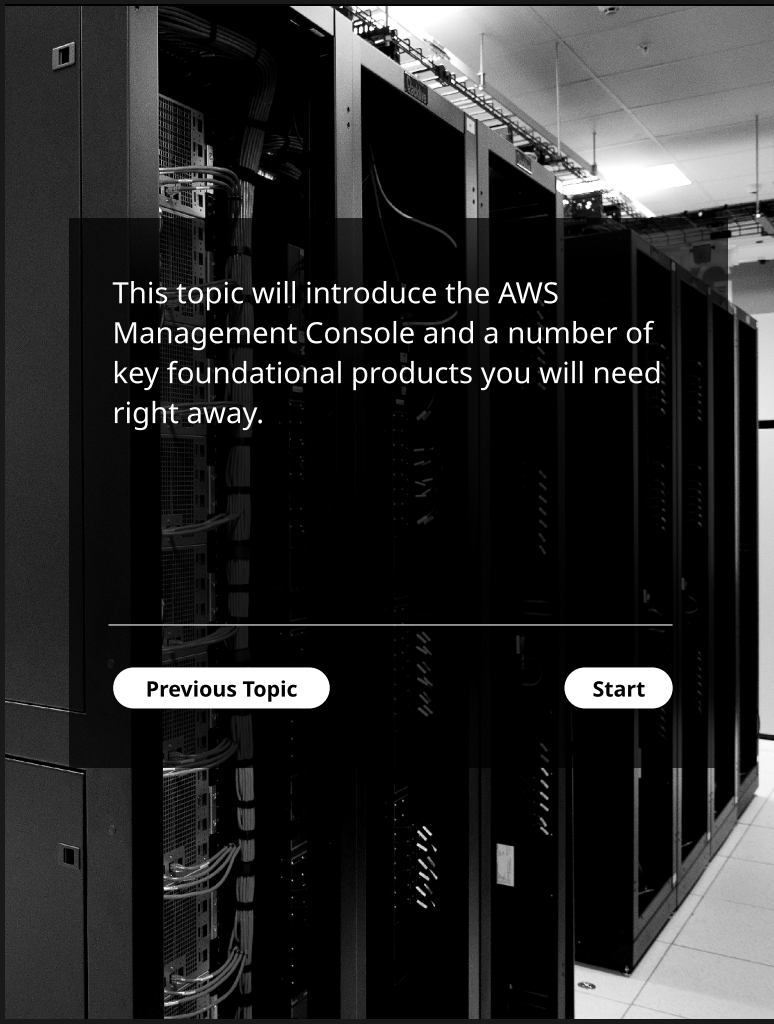
AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



This topic will introduce the AWS Management Console and a number of key foundational products you will need right away.

Previous Topic

Start

Go to Part 2

Back to Main

AWS and SA Fundamentals

Product Fundamentals

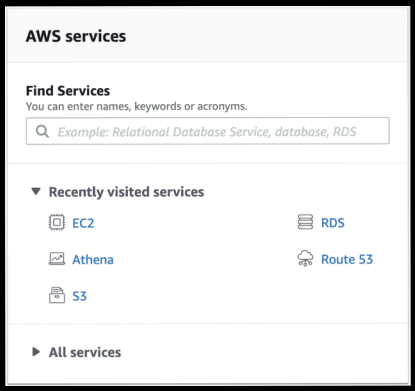
AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



AWS services allows quick navigation to any AWS service via the **Find Services box** or the **All Services dropdown**. Recently visited services are available for quick access.

Back

Next

Go to Part 2

Back to Main

AWS and SA Fundamentals

Product Fundamentals

AWS and SA Fundamentals
Section 1

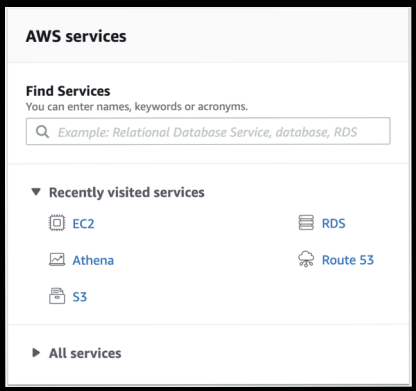
- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Clicking the **AWS** button will always return you to the **main AWS console** regardless of your current location.



AWS services allows quick navigation to any AWS service via the **Find Services box** or the **All Services dropdown**. Recently visited services are available for quick access.

Back

Next

Go to Part 2

Back to Main

AWS and SA Fundamentals

Product Fundamentals

AWS and SA Fundamentals
Section 1

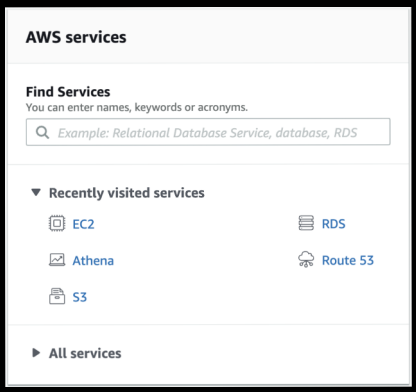
- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Clicking the **Services dropdown** will display a categorized breakdown of AWS services and allow access to frequent services. Using this, and opening a given service in a **new browser tab**, is often the preferred way to access a given service via the console.



AWS services allows quick navigation to any AWS service via the **Find Services box** or the **All Services dropdown**. Recently visited services are available for quick access.

Back

Next

Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

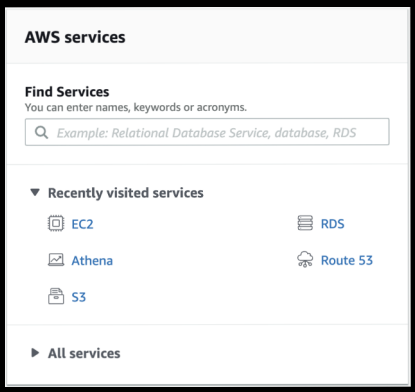
Compute
Section 3

Networking
Section 4

AWS and SA Fundamentals

Product Fundamentals

The **pin** allows you to **drag** frequently used services to the menu bar on a long-term, project, or task basis.



AWS services allows quick navigation to any AWS service via the **Find Services box** or the **All Services dropdown**. Recently visited services are available for quick access.

Back

Next

Go to Part 2

Back to Main

Linux Academy

AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

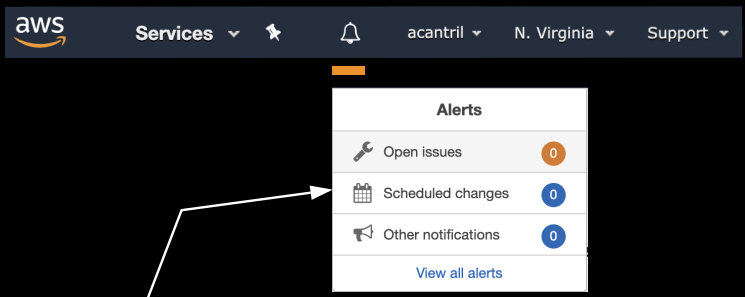
Identity and Access Control
Section 2

Compute
Section 3

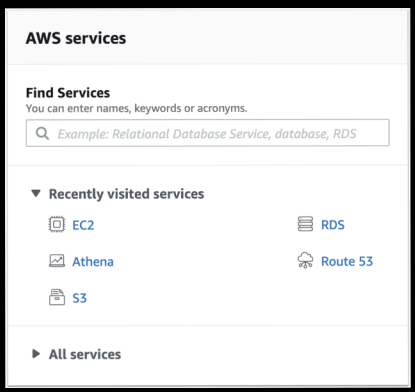
Networking
Section 4

AWS and SA Fundamentals

Product Fundamentals



Clicking **notifications** will display any relevant AWS events for your account. This can include **open issues** (support tickets), **scheduled changes** that will impact your account, and any other AWS notifications.



AWS services allows quick navigation to any AWS service via the **Find Services box** or the **All Services dropdown**. Recently visited services are available for quick access.

AWS and SA Fundamentals

Product Fundamentals

AWS and SA Fundamentals
Section 1

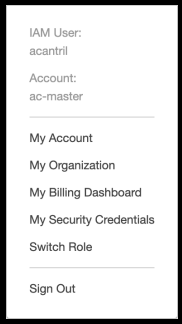
- Architecture 101
- AWS Architecture 101
- Product Fundamentals

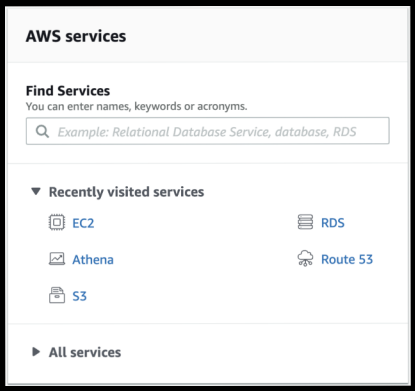
Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

The **accounts dropdown** will allow quick access to account-related areas of the AWS console. This includes the AWS account page, AWS organizations (if applicable), billing dashboard (if available), and your personal security credentials page (IAM user or root user).





AWS services allows quick navigation to any AWS service via the **Find Services box** or the **All Services dropdown**. Recently visited services are available for quick access.

Back

Next

Go to Part 2

Back to Main

AWS and SA Fundamentals

Product Fundamentals

AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

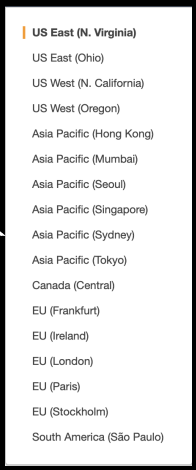
Identity and Access Control
Section 2

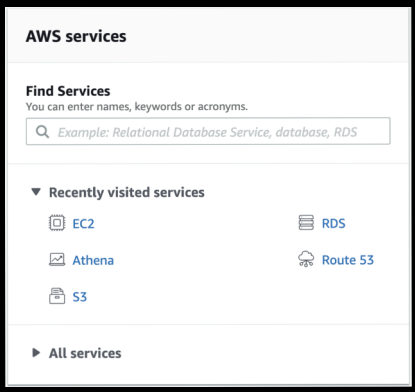
Compute
Section 3

Networking
Section 4

The **region dropdown** is where you can change the region you are interacting with. For any services that are "per region," this will show the **region name** for the AWS region (e.g., **US East (N. Virginia)** rather than **us-east-1**).

When you use services such as S3, IAM, or Route 53, this will show **Global**.





AWS services allows quick navigation to any AWS service via the **Find Services box** or the **All Services dropdown**. Recently visited services are available for quick access.

AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

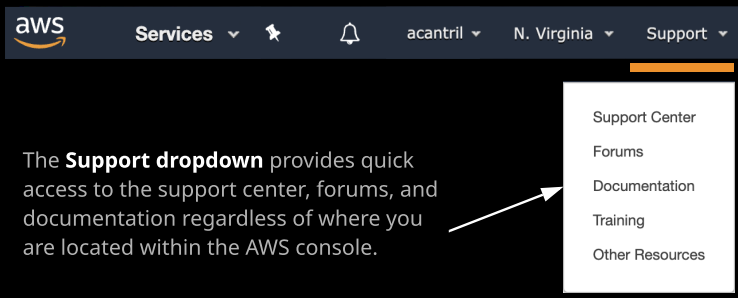
Identity and Access Control
Section 2

Compute
Section 3

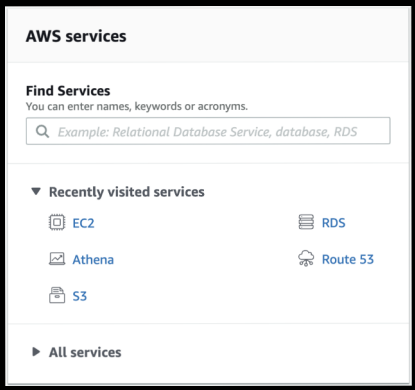
Networking
Section 4

AWS and SA Fundamentals

Product Fundamentals



The **Support dropdown** provides quick access to the support center, forums, and documentation regardless of where you are located within the AWS console.



AWS services allows quick navigation to any AWS service via the **Find Services box** or the **All Services dropdown**. Recently visited services are available for quick access.

Back

Next

Go to Part 2

Back to Main

AWS and SA Fundamentals

Product Fundamentals

AWS and SA Fundamentals
Section 1

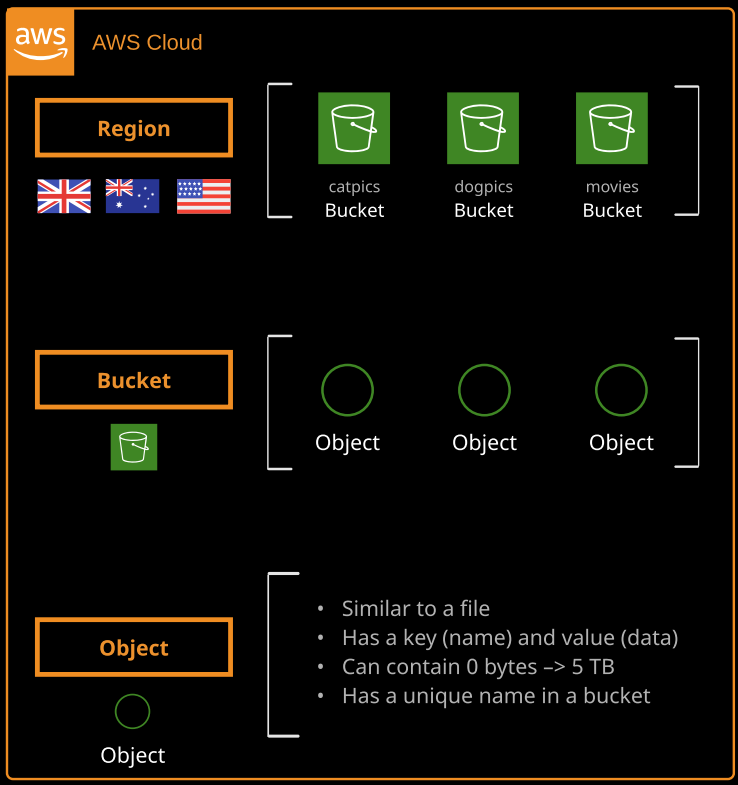
- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Simple Storage Service (**S3**) is a global **object** storage platform that can be used to store objects in the form of text files, photos, audio, movies, large binaries, or other object types.



Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101
AWS Architecture 101
Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Exam Facts and Figures: S3 Fundamentals

- Bucket names have to be **globally unique**
- **Minimum of three** and **maximum of 63** characters — no uppercase or underscores
- Must start with a **lowercase letter** or **number** and can't be formatted as an IP address (1.1.1.1)
- Default **100 buckets** per account, and **hard 1,000-bucket limit** via support request
- **Unlimited objects** in buckets
- **Unlimited total capacity** for a bucket
- An object's **key** is its **name**
- An object's **value** is its **data**
- An object's size is from **0 to 5 TB**

Go to Part 2
Back to Main

AWS and SA Fundamentals

Product Fundamentals

AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

CloudFormation is an Infrastructure as Code (**IaC**) product — you can **create**, **manage**, and **remove** infrastructure using **JSON** or **YAML**.

1

Template

A CFN template is JSON or YAML. It contains **logical resources** and configuration.

2

Stack

Stacks are created and modified based on templates, which can be changed and used to update a stack.

3

Physical Resources

Stacks take **logical resources** from a template and create, update, or delete the **physical resources** in AWS.

CloudFormation is effective if you **frequently deploy** the same infrastructure or you require **guaranteed consistent configuration**.

Course Navigation

AWS and SA Fundamentals
Section 1

Architecture 101
AWS Architecture 101
Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Exam Facts and Figures: CloudFormation Fundamentals

A CloudFormation (CFN/cfn) template is used to initially create a CFN stack. A stack **creates**, **updates**, and **deletes** physical AWS resources based on its **logical resources**, which are based on the contents of a **template**.

- A CFN template is written in **JSON** or **YAML**.
- A template can create up to **200 resources**.
- If a **stack** is **deleted**, then, by default, any **resources** it has created are also **deleted**.
- A stack can be **updated** by uploading a **new version** of a template.
- **New** logical resources cause **new** physical resources.
- **Removed** logical resources cause the stack to **delete** physical resources.
- **Changed** logical resources **update** with **some disruption** or **replace** physical resources.

Go to Part 2
Back to Main

Course Navigation

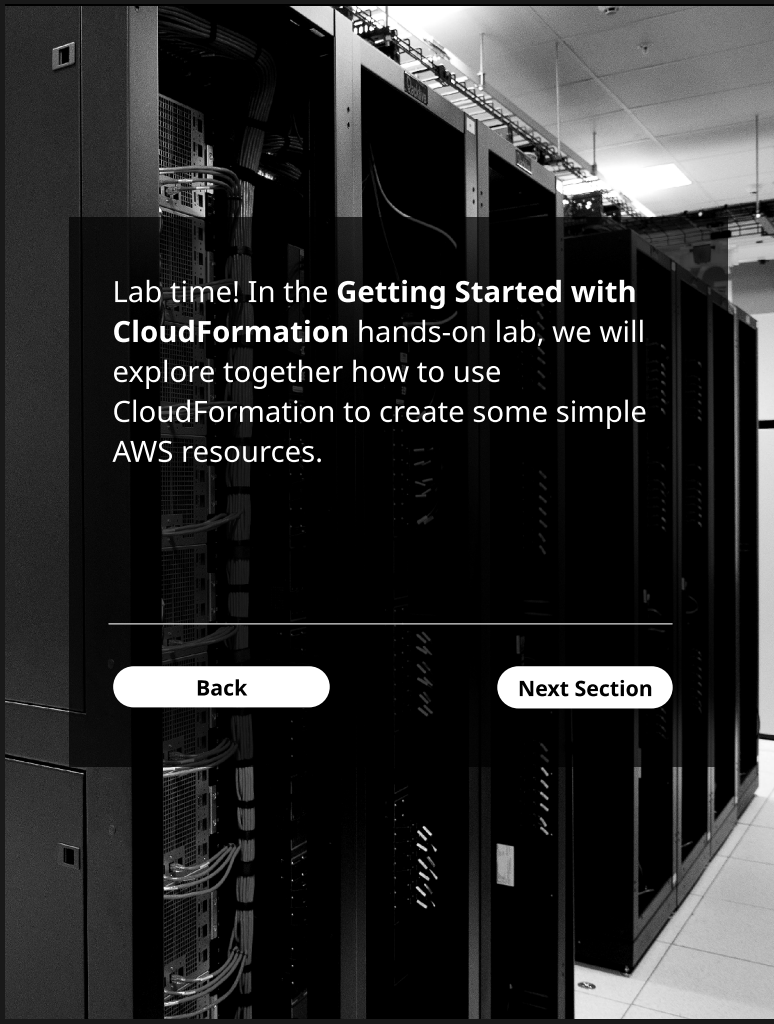
AWS and SA Fundamentals
Section 1

- Architecture 101
- AWS Architecture 101
- Product Fundamentals

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4



Lab time! In the **Getting Started with CloudFormation** hands-on lab, we will explore together how to use CloudFormation to create some simple AWS resources.

Back

Next Section

Go to Part 2

Back to Main

Identity and Access Control

Identity and Access Management

Course Navigation

AWS and SA Fundamentals
Section 1

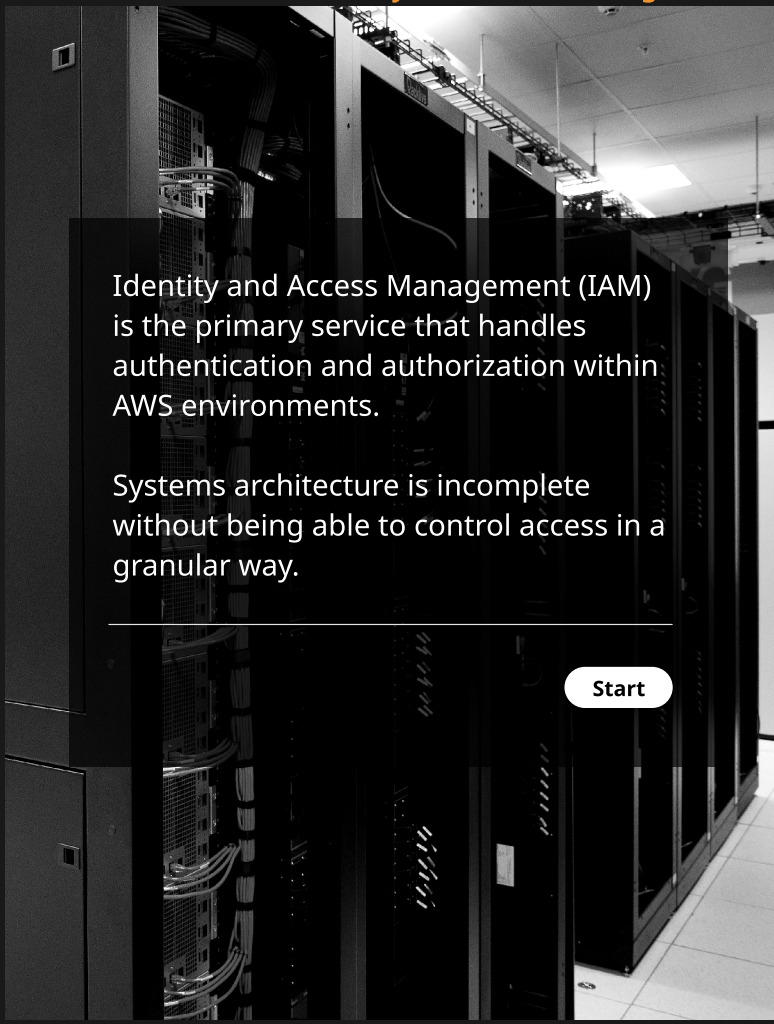
Identity and Access Control
Section 2

IAM

Multi-Account and Orgs

Compute
Section 3

Networking
Section 4



Identity and Access Management (IAM) is the primary service that handles authentication and authorization within AWS environments.

Systems architecture is incomplete without being able to control access in a granular way.

Start

Go to Part 2

Back to Main

Identity and Access Control

Identity and Access Management

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

IAM

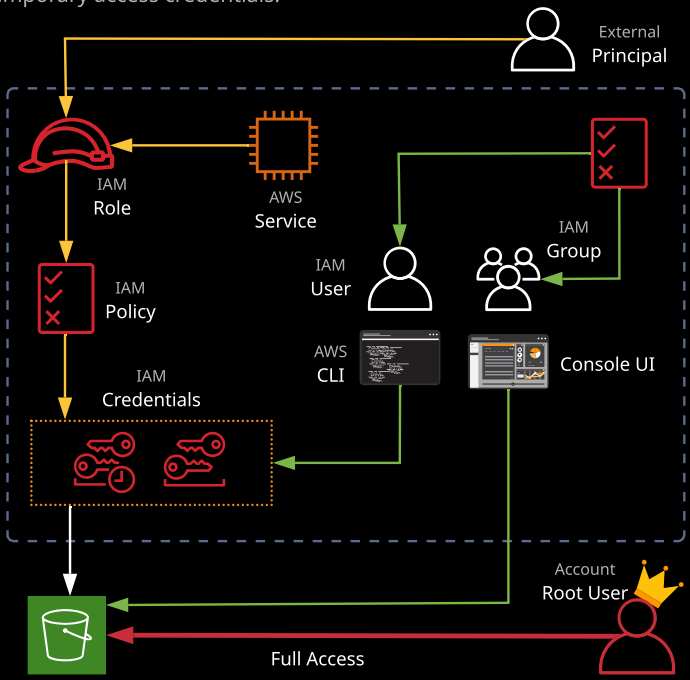
Multi-Account and Orgs

Compute
Section 3

Networking
Section 4

IAM controls access to AWS services via **policies** that can be attached to **users**, **groups**, and **roles**. Users are given long-term credentials to access AWS resources (username and password or access keys).

Roles allow for short-term access to resources when assumed, using temporary access credentials.



Back

Next

Go to Part 2

Back to Main

Linux Academy

Identity and Access Control

Identity and Access Management

Amazon Resource Name (ARN)

ARNs always **begin with:**

```
arn:partition:service:region:account-id:
```

`partition` = `aws` or `aws-cn` (for China)
`service` = the AWS service: `s3`, `ec2`, `rds`, `dynamodb`
`region` = region code: `us-east-1`, `ap-southeast-2`

And, depending on service, **finish with:**

```
resource
resourcetype/resource
resourcetype/resource/qualifier
resourcetype/resource:qualifier
resourcetype:resource
resourcetype:resource:qualifier
```

Example ARNs:

```
arn:aws:iam::123456789012:user/roffle
arn:aws:s3:::myamazingcatpics/truffs.jpeg
arn:aws:dynamodb:us-east-1:123456789012:table/ratemycats
```

In some cases, **wildcards are supported:**

```
arn:aws:ec2:us-east-1:123456789012:instance/*
```

Fields with `::` omit the value, and `*` is a wildcard match.

Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

IAM

Multi-Account and Orgs

Compute
Section 3

Networking
Section 4

Identity and Access Control

Identity and Access Management

An **IAM policy** (policy document) is known as an **identity policy** when attached to an identity or a **resource policy** when attached to a resource. They have no effect until they are attached to something.

A policy document is a list of **statements**:

```
{
    "Version": "2012-10-17",
    "Statement": [{...},{...},{...}]
}
```

Each statement matches a request to AWS. Requests are matched based on their `Action` (or actions), which are the API calls or operations being attempted and the `Resource` (or resources) the request is against. A given statement results in an `Allow` or `Deny` for the request.

```
{
    "Sid": "SpecificTable",
    "Effect": "Allow",
    "Action":
        [
            "dynamodb:BatchGet*",
            "dynamodb:DescribeStream",
            "dynamodb:DescribeTable",
            "dynamodb:Get*", "dynamodb:Query",
            "dynamodb:Scan", "dynamodb:BatchWrite*",
            "dynamodb:CreateTable",
            "dynamodb:Delete*", "dynamodb:Update*",
            "dynamodb:PutItem"
        ],
    "Resource": "arn:aws:dynamodb:*:*:table/CatPics"
}
```

Identity and Access Control

Identity and Access Management

Exam Tips: IAM Policies

- If a request isn't explicitly allowed, it's implicitly (**default**) denied.
- If a request is explicitly denied, it **overrides everything else**.
- If a request is explicitly allowed, it's allowed unless denied by an explicit deny.
- Remember: **DENY** -> **ALLOW** -> **DENY**
- Only attached policies have any impact.
- When evaluating policies, all applicable policies are merged:
  - All identity (user, group, role) and any resource policies
- Managed policies allow the same policy to impact many identities.
- Inline policies allow exceptions to be applied to identities.
- AWS-managed policies are low overhead but lack flexibility.
- Customer-managed policies are flexible but require administration.
- Inline and managed policies can apply to users, groups, and roles.

Identity and Access Control

Identity and Access Management

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

IAM

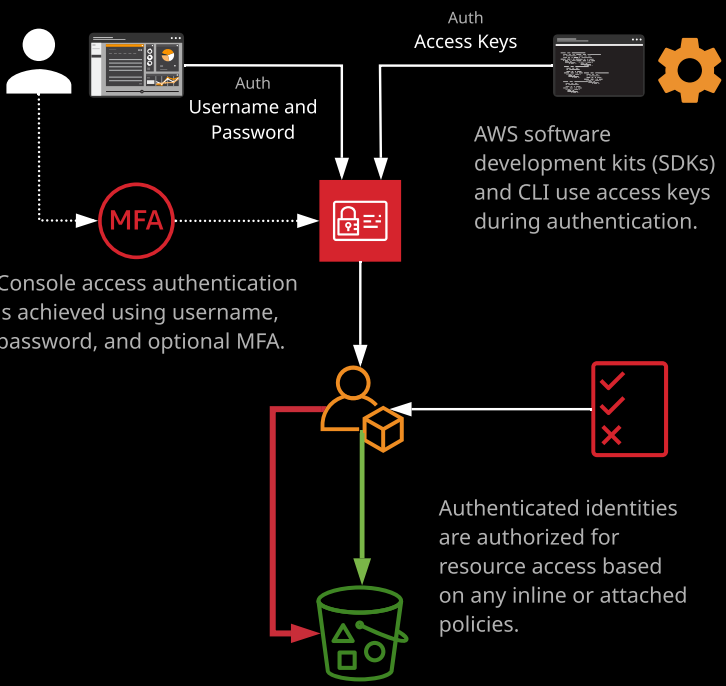
Multi-Account and Orgs

Compute
Section 3

Networking
Section 4

IAM users are a type of IAM identity suitable for **long-term** access for a **known entity** (human, service, application).

Principals authenticate to IAM users either with a **username** and **password** or using **access keys**.



Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

IAM

Multi-Account and Orgs

Compute
Section 3

Networking
Section 4

Exam Facts and Figures: IAM Users

- Hard limit of 5,000 IAM users per account — this is **important**, as it can impact architecture
- 10 group memberships per IAM user
- Default maximum of 10 managed policies per user
- No inline limit, but you cannot exceed 2048 characters for **all** inline policies on a IAM user
- 1 MFA per user
- 2 access keys per user

Go to Part 2

Back to Main

Identity and Access Control

Identity and Access Management

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

IAM

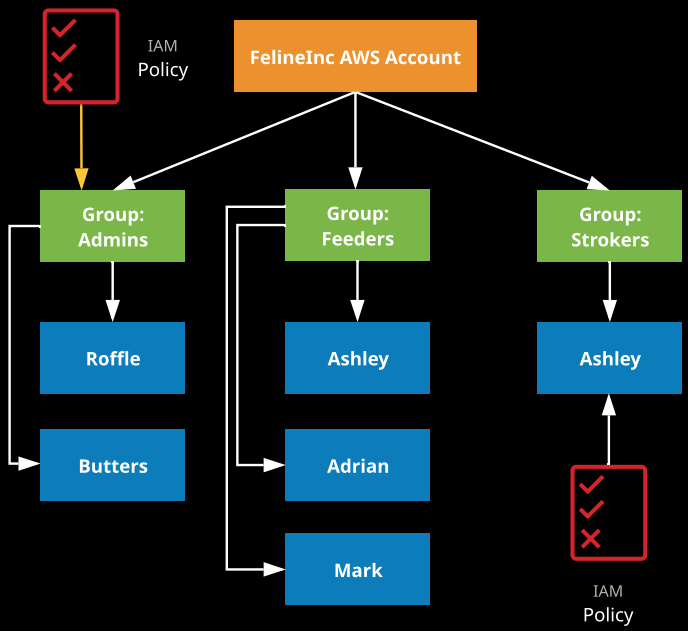
Multi-Account and Orgs

Compute
Section 3

Networking
Section 4

An **IAM group** is a **collection of IAM users**. Groups allow easier administration over sets of IAM users. Inline and managed policies can be applied to groups that **flow on to members** of that group.

Groups are **not** a true identity — they cannot be the principal in a policy, so they can't be used in resource policies.



Back

Next

Go to Part 2

Back to Main

Identity and Access Control

Identity and Access Management

Exam Facts and Figures: IAM Groups

- Groups are an admin feature to group IAM users.
- Groups can contain many IAM users, and users can be in many groups.
- IAM inline policies can be added to IAM groups — and these flow on to IAM users who are members.
- Managed IAM policies can be attached and flow on to IAM users who are members.
- Groups are not "**true**" identities, and they can't be referenced from resource policies.
- Groups have **no** credentials.

Go to Part 2

Back to Main

Identity and Access Control

Identity and Access Management

Access keys are a pair of values used by applications, SDKs, or the AWS command line to authenticate to AWS.

Access keys consist of two parts: the **access key ID** and **secret access key**. The access key ID is the public part of the key and is stored by AWS once generated.

Access Key ID	`AKIAIOSFODNN7EXAMPLE`

The secret access key is the sensitive and private part of the access key, available only once when the access key is initially generated. It's stored only by the owner and should never be revealed.

Secret Access Key	`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`

An IAM user is the only identity that uses access keys. They are allowed two sets. They can be created, deleted, enabled, and disabled.

They can't be used to log in to the console, and they don't expire. If anyone finds a set of access keys, they have access to the permissions of the IAM user to which they belong.

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

IAM

Multi-Account and Orgs

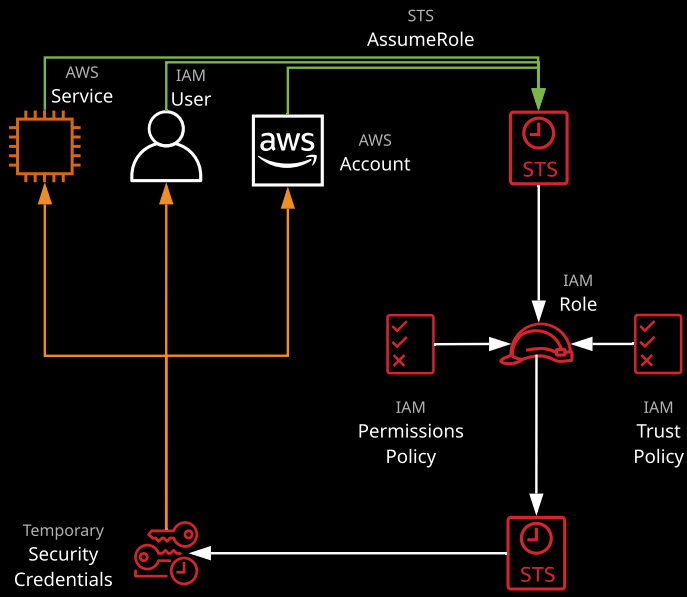
Compute
Section 3

Networking
Section 4

Identity and Access Control

Identity and Access Management

IAM roles are **assumed** by another identity allowed in the **trust policy** — IAM user, AWS service, another AWS account, web identity, or even an anonymous identity. When a role is assumed, the Security Token Service (**STS**) generates a **time-limited** set of access keys (temporary security credentials). These access keys have the permissions defined in the permissions policy. IAM roles have no long-term credentials (access keys or username and password).



Identity and Access Control

Identity and Access Management

Exam Facts and Figures: IAM Roles

- IAM roles have no long-term credentials.
- They are `sts:AssumeRole` by another identity:
  - IAM users
  - AWS services
  - External accounts
  - Web identities
- Temporary security credentials are generated by STS.
- **Trust** policy controls which identities can assume the role.
- **Permissions** policy defines the permissions provided.
- Temporary credentials expire.
- Example scenarios:
  - Company merger
  - AWS service access
  - "Break-glass"-style extra access
  - Cross-account access
  - Web identity federation

Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

IAM

Multi-Account and Orgs

Compute
Section 3

Networking
Section 4

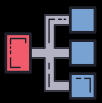
Identity and Access Control

Multi-Account and Orgs

AWS Organizations is a service for managing **multiple accounts** within a single business. Rather than managing many accounts, with many isolated sets of logins and individual bills, Organizations allows consolidation.

All accounts within an AWS Organization can **consolidate bills** into a single account — one bill covering all business usage. Organizations can share bulk discounts and even easily manage accounts and permissions and limit account usage using **service control policies**.

Consolidated Billing



All Features

Previous Topic

Next

Go to Part 2

Back to Main

Identity and Access Control

Multi-Account and Orgs

Course Navigation

AWS and SA Fundamentals
Section 1

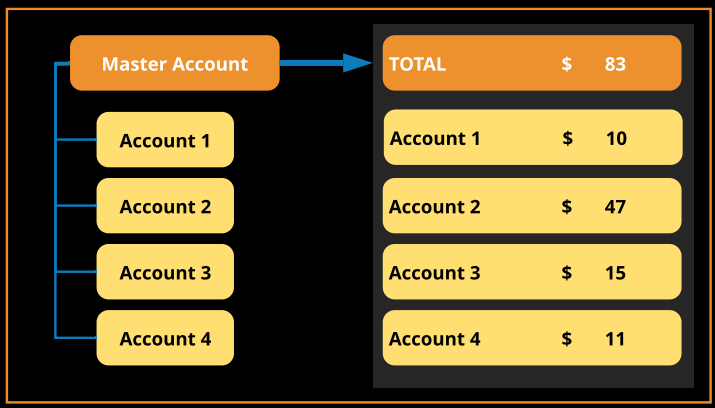
Identity and Access Control
Section 2

IAM

Multi-Account and Orgs

Compute
Section 3

Networking
Section 4



Go to Part 2

Back to Main

Identity and Access Control

Multi-Account and Orgs

Course Navigation

AWS and SA Fundamentals
Section 1

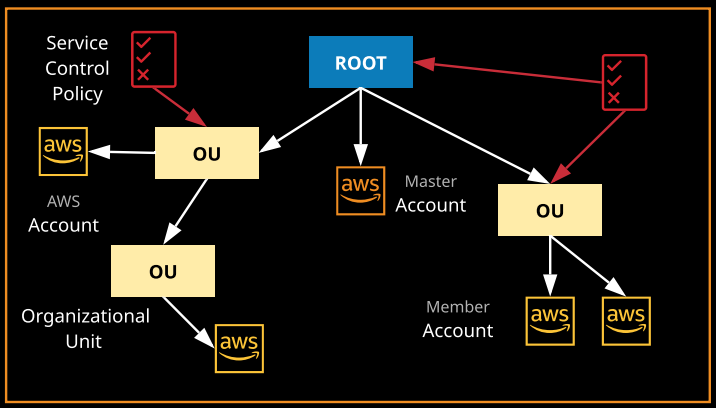
Identity and Access Control
Section 2

IAM

Multi-Account and Orgs

Compute
Section 3

Networking
Section 4



Go to Part 2

Back to Main

Linux Academy

Identity and Access Control

Multi-Account and Orgs

AWS and SA Fundamentals
Section 1

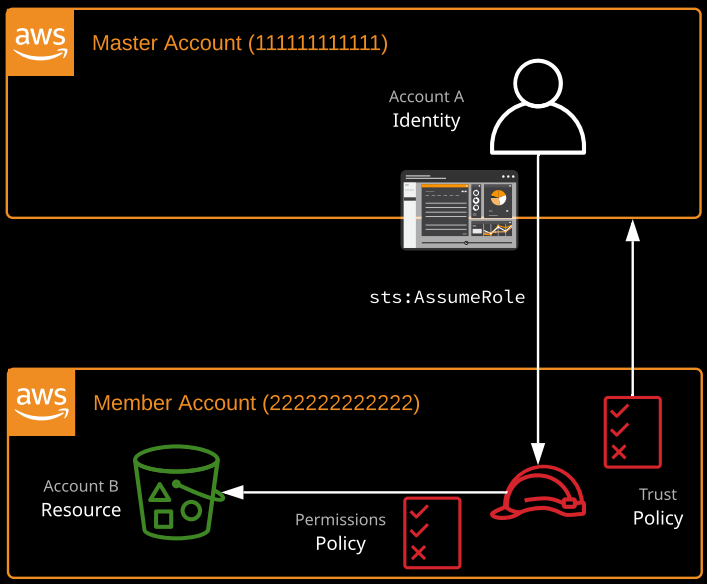
Identity and Access Control
Section 2
IAM
Multi-Account and Orgs

Compute
Section 3

Networking
Section 4

Role switching is a method of accessing one account from another using only one set of credentials. It is used both within **AWS Organizations** and between two **unconnected accounts**.

1. A role in Account B **trusts** Account A.
2. An identity in Account A can **assume** the role in Account B...
3. ...and, using that role, it can **operate inside** Account B.



Course Navigation

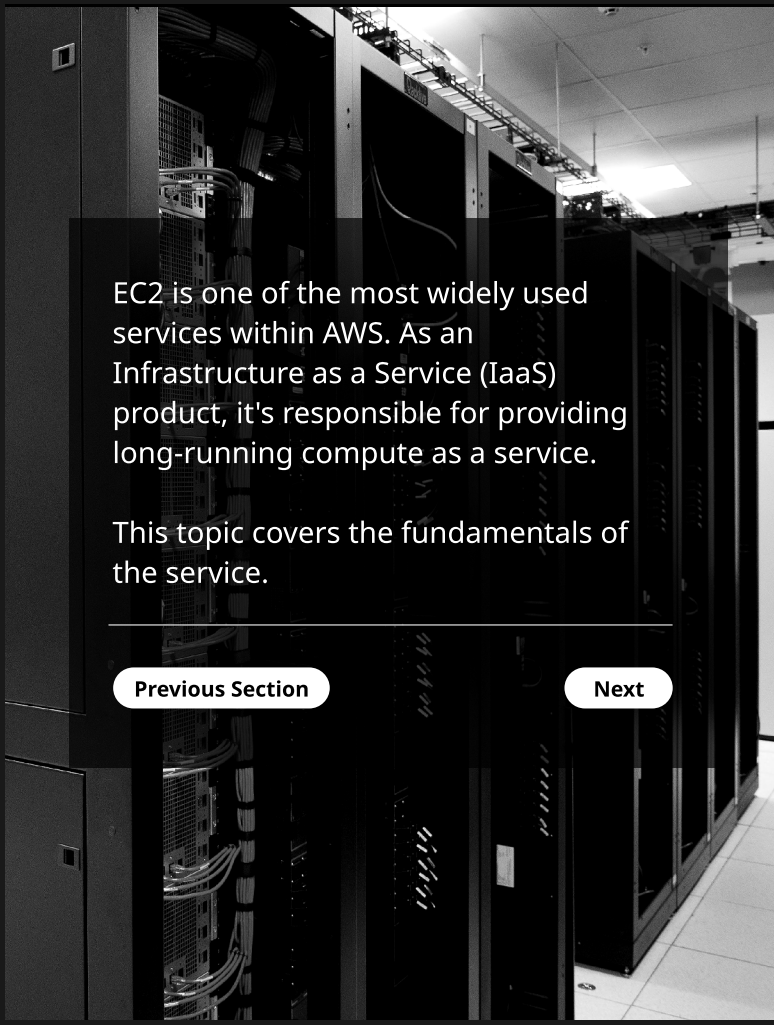
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4



EC2 is one of the most widely used services within AWS. As an Infrastructure as a Service (IaaS) product, it's responsible for providing long-running compute as a service.

This topic covers the fundamentals of the service.

Previous Section

Next

Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

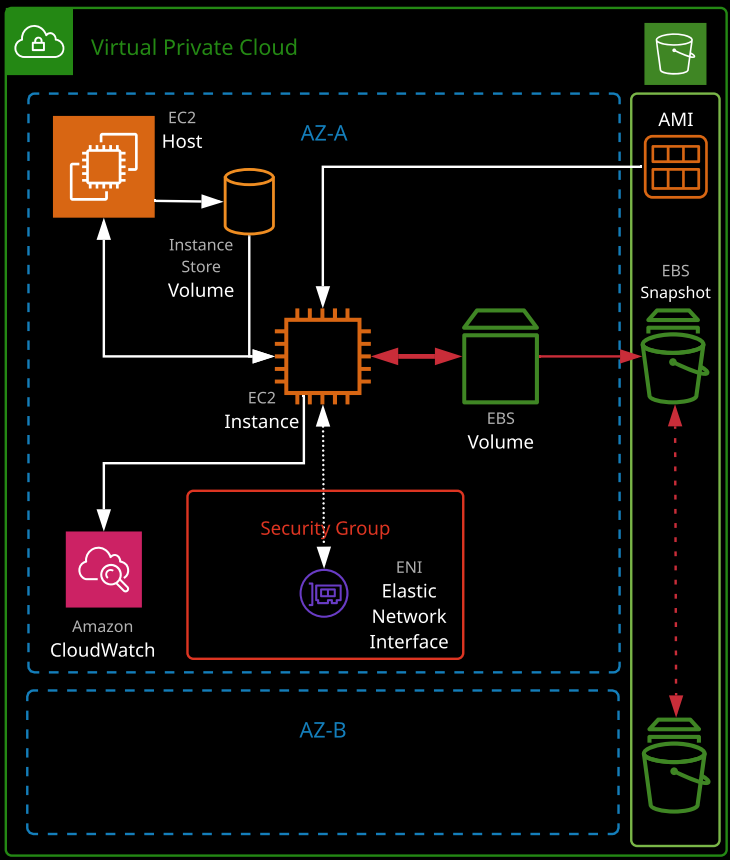
Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

Compute

EC2 Fundamentals



Back
Next

Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

EC2 instances are grouped into families, which are designed for a specific broad type workload. The type determines a certain set of features, and sizes decide the level of workload they can cope with.

The current EC2 families are **general purpose**, **compute optimized**, **memory optimized**, **storage optimized**, and **accelerated computing**.

Instance types include:

- **T2** and **T3**: Low-cost instance types that provide burst capability
- **M5**: For general workloads
- **C4**: Provides more capable CPU
- **X1** and **R4**: Optimize large amounts of fast memory
- **I3**: Delivers fast IO
- **P2**, **G3**, and **F1**: Deliver GPU and FPGAs

Instance sizes include nano, micro, small, medium, large, x.large, 2x.large, and larger.

Special Cases

- "**a**": Use AMD CPUs
- "**A**": Arm based
- "**n**": Higher speed networking
- "**d**": NVMe storage

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

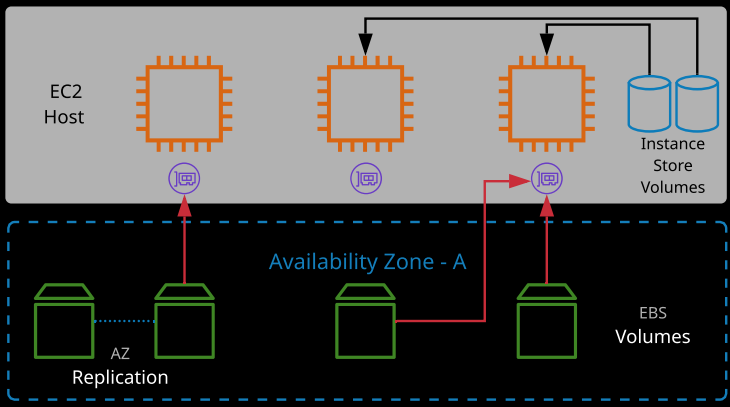
- EC2 Fundamentals
- EC2 Intermediate
- EC2 Advanced
- Serverless Compute
- Containers

Networking
Section 4

Compute

EC2 Fundamentals

Elastic Block Store (EBS) is a storage service that **creates and manages volumes** based on four underlying storage types. Volumes are **persistent**, can be **attached and removed** from EC2 instances, and are **replicated** within a single AZ.



Volume Types

- Mechanical **sc1** and **st1**; Solid State **gp2** and **io1**
- **sc1**: Lowest cost, infrequent access, can't be boot volume
- **st1**: Low cost, throughput intensive, can't be a boot volume
- **gp2**: Default, balance of IOPS/MiB/s - burst pool IOPS per GB
- **io1**: Highest performance, can adjust size and IOPS separately

To protect against **AZ failure**, EBS snapshots (to S3) can be used. Data is **replicated** across AZs in the region and (optionally) internationally.

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

Exam Facts and Figures: EBS

- EBS supports a maximum per-instance throughput of 1,750 MiB/s and 80,000 IOPS. If you need more ... instance store volumes.

General Purpose (gp2): (SSD)
Default for most workloads
- 3 IOPS/GiB (100 IOPS – 16,000 IOPS)
- Bursts up to 3,000 IOPS (credit based)
- 1 GiB – 16 TiB size, max throughput p/vol of 250 MiB/s

Provisioned IOPS SSD (io1): (SSD)
- Used for applications that require sustained IOPS performance
- Large database workloads
- Volume size of 4 GiB – 16 TiB up to 64,000 IOPS per volume
- Max throughput p/vol of 1,000 MiB/s

Throughput Optimized (st1): (HDD)
- Low storage cost
- Used for frequently accessed, throughput-intensive workloads (streaming, big data)
- Cannot be a boot volume
- Volume size of 500 GiB – 16 TiB
- Per-volume max throughput of 500 MiB/s and IOPS 500

Cold HDD (sc1): (HDD)
- Lowest cost
- Infrequently accessed data
- Cannot be a boot volume
- Volume size of 500 GiB – 16 TiB
- Per-volume max throughput of 250 MiB/s and 250 IOPS

Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

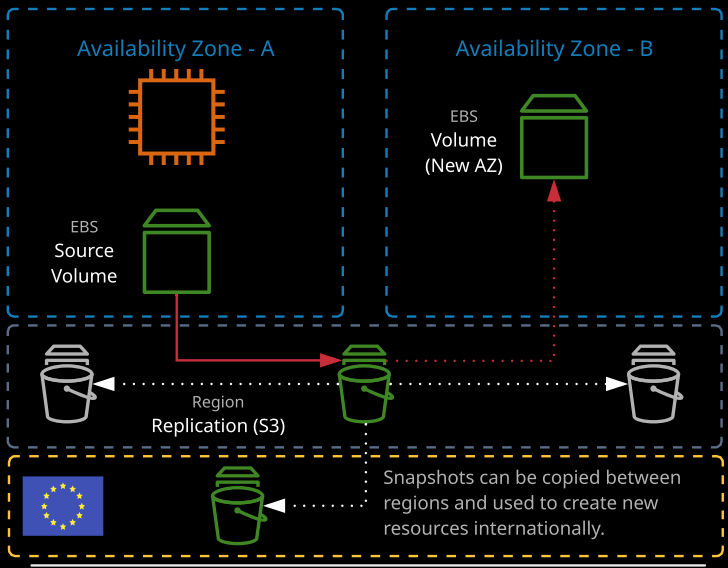
EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

EBS snapshots are a **point-in-time backup** of an EBS volume stored in **S3**. The initial snapshot is a **full copy** of the volume. Future snapshots only store the **data changed since the last snapshot**.

Snapshots can be used to create new volumes and are a great way to **move** or **copy** instances between **AZs**. When creating a snapshot of the root/boot volume of an instance or busy volume, it's recommended the instance is powered off, or disks are "flushed."

Snapshots can be copied between regions, shared, and automated using Data Lifecycle Manager (DLM).



Back
Next

Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

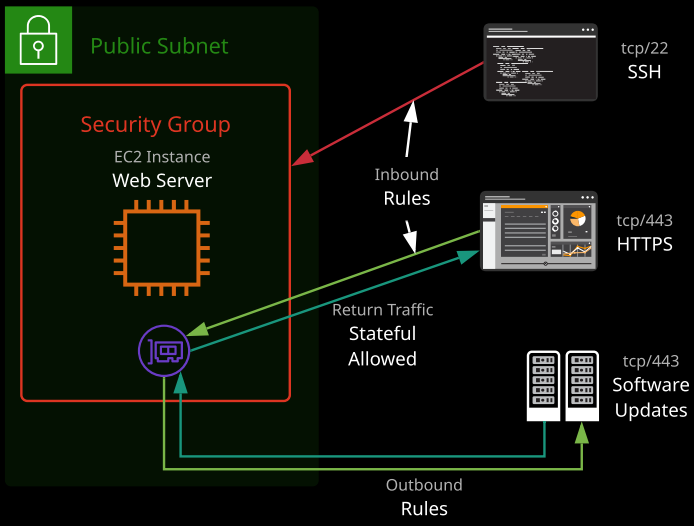
EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

Security groups are software **firewalls** that can be attached to **network interfaces** and (by association) products in AWS. Security groups each have inbound rules and outbound rules. A rule allows traffic **to** or **from** a source (IP, network, named AWS entity) and protocol.

Security groups have a hidden **implicit/default** deny rule but **cannot explicitly deny traffic.**

They are stateful — meaning for any traffic allowed in/out, the return traffic is automatically allowed. Security groups can reference AWS resources, other security groups, and even themselves.



Back

Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

Instance Metadata

Instance metadata is data relating to the instance that can be accessed from **within the instance itself** using a utility capable of accessing HTTP and using the URL:

http://169.254.169.254/latest/meta-data

AMIusedtocreatetheinstance:
`http://169.254.169.254/latest/meta-data/ami-id`
InstanceID:
`http://169.254.169.254/latest/meta-data/instance-id`
Instancetype:
`http://169.254.169.254/latest/meta-data/instance-type`

Instance metadata is a way that scripts and applications running on EC2 can get **visibility of data** they would normally need API calls for.

The metadata can provide the current **external IPv4 address** for the instance, which isn't configured on the instance itself but provided by the internet gateway in the VPC. It provides the **Availability Zone** the instance was launched in and the **security groups** applied to the instance. In the case of spot instances, it also provides the **approximate time** the instance will terminate.

For the exam: Remember the IP address to access metadata.

Course Navigation

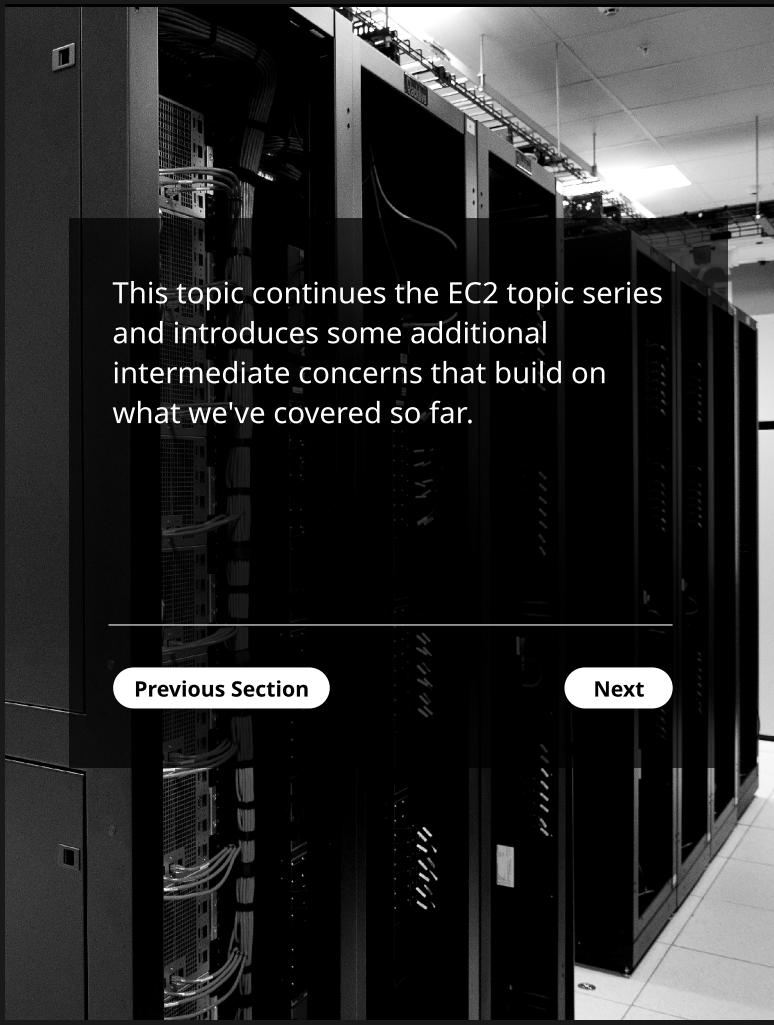
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4



This topic continues the EC2 topic series and introduces some additional intermediate concerns that build on what we've covered so far.

Previous Section

Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

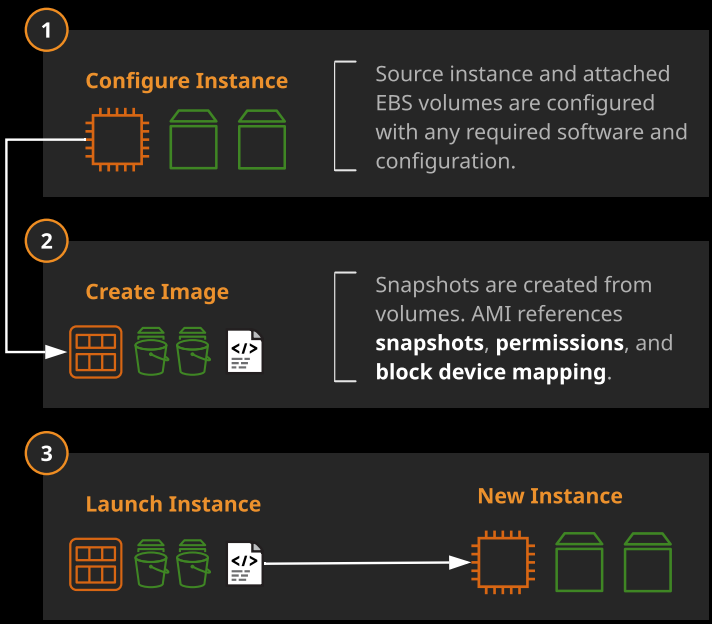
Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

AMIs (Amazon Machine Images) are used to build instances. They store **snapshots** of EBS volumes, **permissions**, and a **block device mapping**, which configures how the instance OS sees the attached volumes. AMIs can be **shared**, **free**, or **paid** and can be copied to other **AWS regions**.



With appropriate **launch permissions**, instances can be created from an AMI. EBS volumes are created using snapshots as the source, and an EC2 instance is created using the block device mapping to reference its new volumes.

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

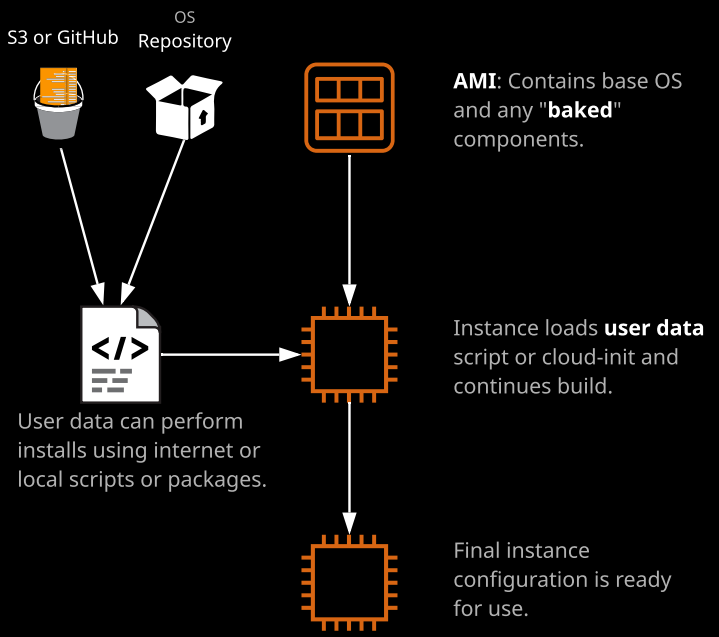
Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

Bootstrapping is a process where **instructions are executed on an instance during its launch process**. Bootstrapping is used to configure the instance, perform software installation, and add application configuration.

In EC2, user data can be used to run **shell scripts** (Bash or PowerShell) or run **cloud-init** directives.



Course Navigation

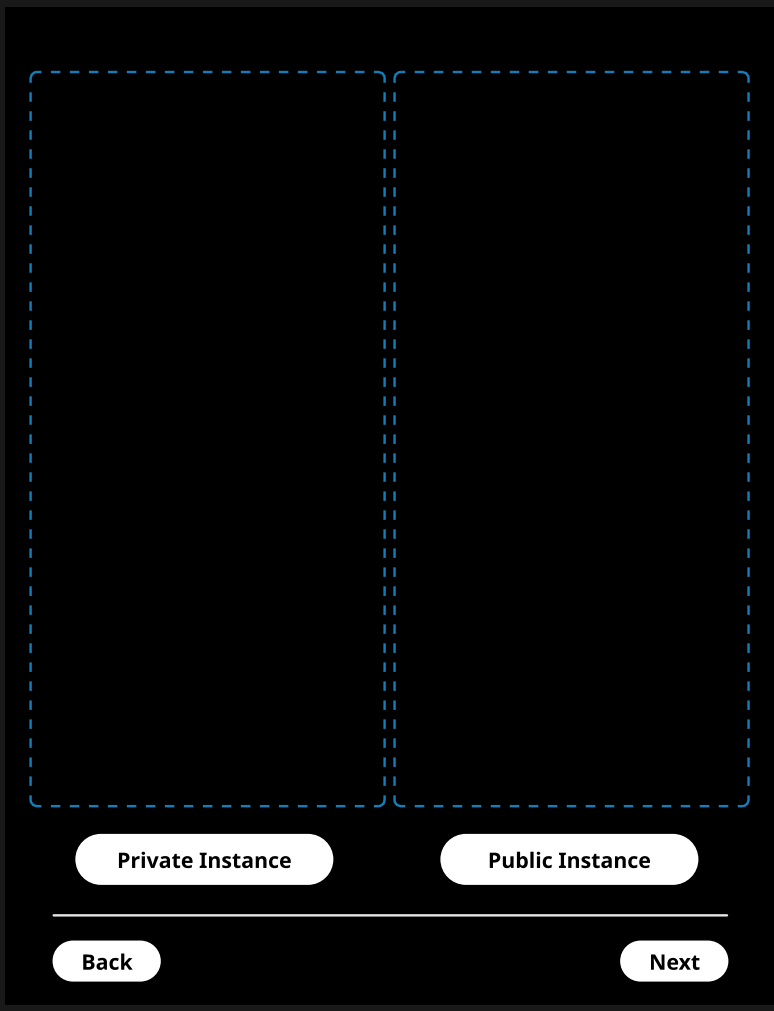
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4



Go to Part 2
Back to Main

Course Navigation

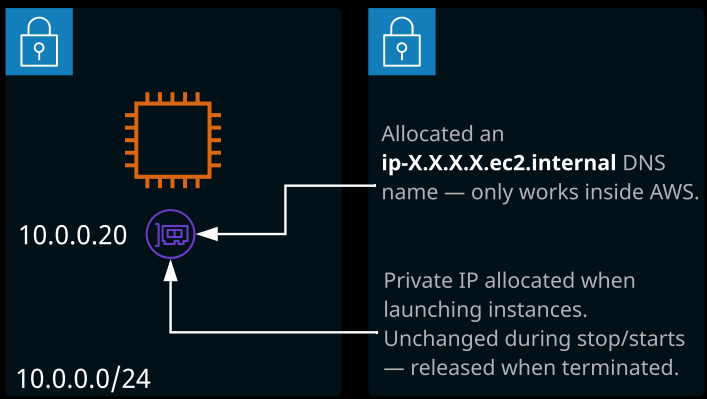
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4



Go to Part 2
Back to Main

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

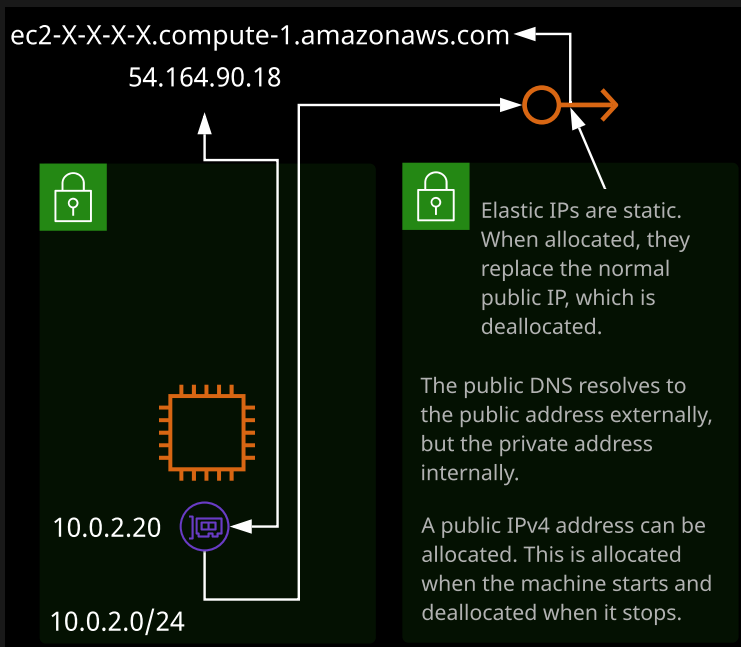
Compute
Section 3

- **EC2 Fundamentals**
- **EC2 Intermediate**
- EC2 Advanced
- Serverless Compute
- Containers

Networking
Section 4

Compute

EC2 Intermediate



Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

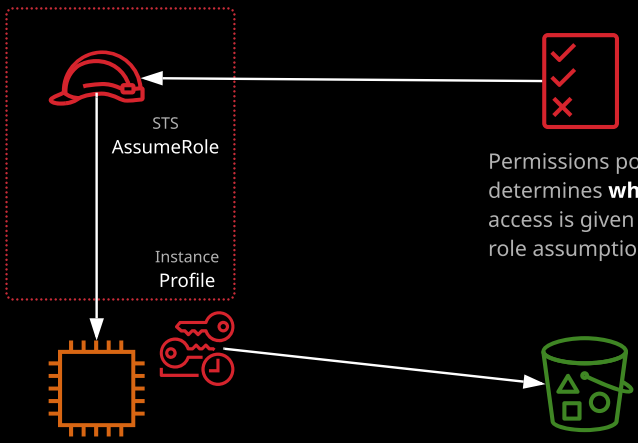
Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

EC2 **instance roles** are IAM roles that can be "assumed" by EC2 using an intermediary called an **instance profile**. An instance profile is either created automatically when using the console UI or manually when using the CLI. It's a container for the role that is associated with an EC2 instance.

The instance profile allows applications on the EC2 instance to access the credentials from the role using the **instance metadata**.



Course Navigation

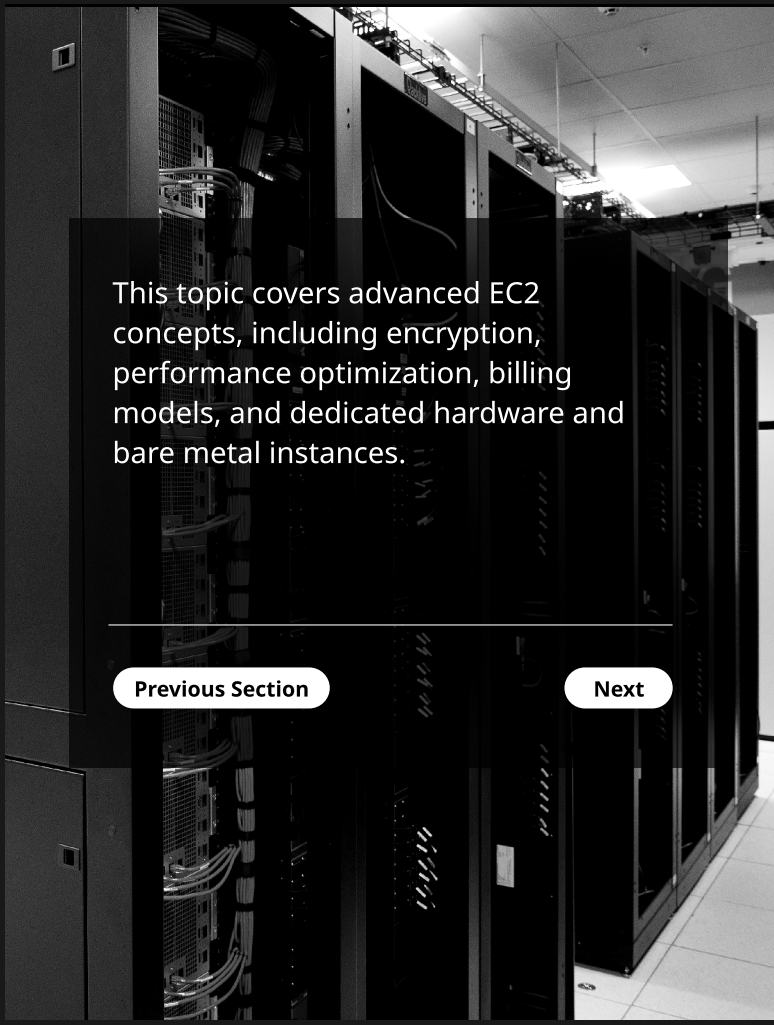
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4



This topic covers advanced EC2 concepts, including encryption, performance optimization, billing models, and dedicated hardware and bare metal instances.

Previous Section

Next

Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

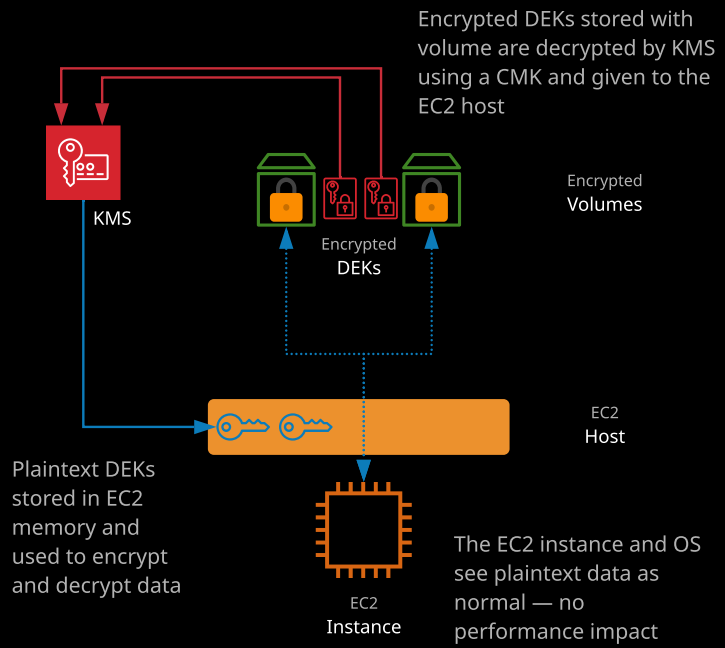
- **EC2 Fundamentals**
- **EC2 Intermediate**
- **EC2 Advanced**
- Serverless Compute
- Containers

Networking
Section 4

Compute

EC2 Advanced

Volume encryption uses EC2 host hardware to encrypt data **at rest** and **in transit** between EBS and EC2 instances. Encryption generates a data encryption key (**DEK**) from a customer master key (**CMK**) in each region. A **unique** DEK encrypts each volume. Snapshots of that volume are encrypted with the **same DEK**, as are any volumes created from that snapshot.



Back

Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

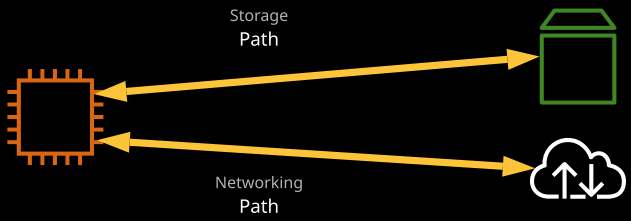
Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

Legacy non-EBS-optimized instances used a shared networking path for data and storage communications.

EBS-optimized mode, which was **historically optional** and is now the **default**, adds optimizations and dedicated communication paths for storage and traditional data networking. This allows consistent utilization of both — and is one required feature to support higher performance storage.



Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

Traditionally, virtual networking meant a virtual host (EC2 host) arranging access for *n* virtual machines to access one physical network card — this multitasking is done in software and is typically slow.

Enhanced networking uses **SR-IOV**, which allows a single physical network card to appear as multiple physical devices. Each instance can be given one of these (fake) physical devices. This results in **faster transfer rates**, **lower CPU usage**, and **lower consistent latency**. EC2 delivers this via the Elastic Network Adapter (ENA) or Intel 82599 Virtual Function (VF) interface.

Cluster PG | Partition PG | Spread PG

Back | Next

Go to Part 2

Back to Main

Course Navigation

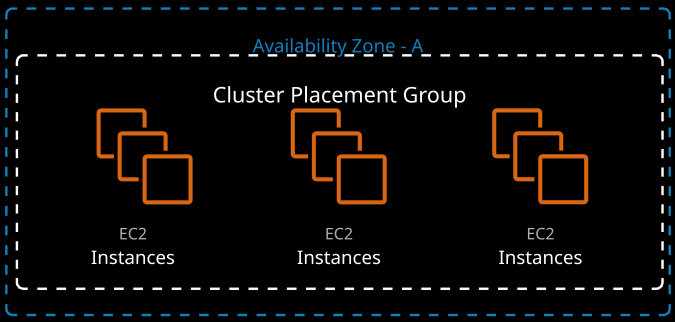
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4



Cluster placement groups place instances physically near each other in a single AZ. Every instance can talk to every other instance at the same time at full speed. Works with enhanced networking for peak performance.

Cluster PG

Go to Part 2
Back to Main

Course Navigation

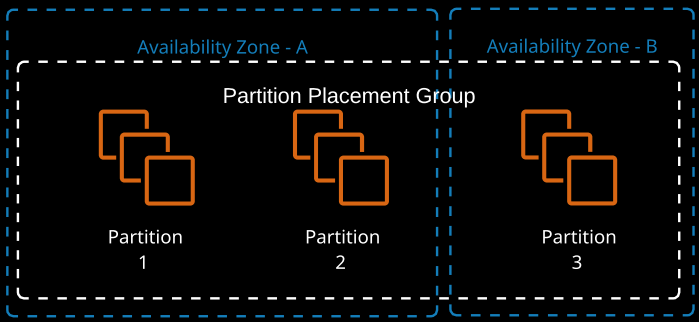
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4



Instances deployed into a partition placement group (PPG) are separated into partitions (max of seven per AZ), each occupying isolated racks in AZs/regions. PPG can span multiple AZs in a region. PPGs minimize failure to a partition and give you visibility on placement.

Partition PG

Go to Part 2
Back to Main

Course Navigation

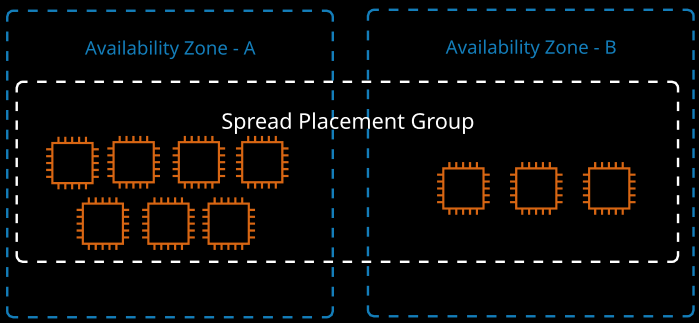
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4



Spread placement groups (SPGs) are designed for a max of seven instances per AZ that need to be separated. Each instance occupies a partition and has an isolated fault domain. Great for email servers, domain controllers, file servers, and application HA pairs.

Spread PG

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

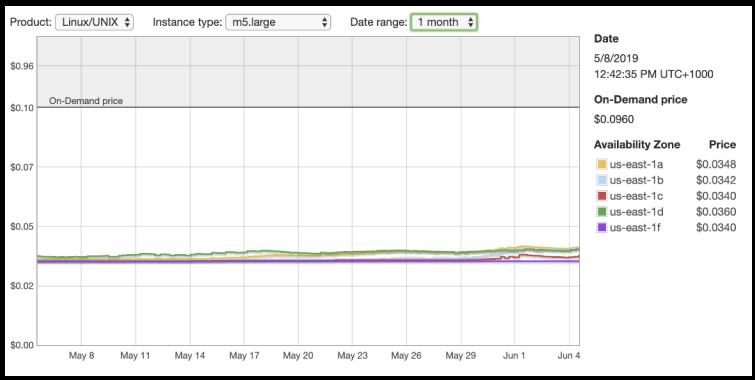
Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

Spot instances allow **consumption of spare AWS capacity** for a given instance type and size in a specific AZ. Instances are provided for as long as your bid price is above the spot price, and you only ever pay the spot price. If your bid is exceeded, instances are terminated with a two-minute warning.



Spot fleets are a container for "capacity needs." You can specify pools of instances of certain types/sizes aiming for a given "capacity." A minimum percentage of on-demand can be set to ensure the fleet is always active.

Spot instances are perfect for non-critical workloads, burst workloads, or consistent non-critical jobs that can tolerate interruptions without impacting functionality.

Spot is **not** suitable for long-running workloads that require stability and cannot tolerate interruptions.

Back
Next

Go to Part 2
Back to Main

Course Navigation

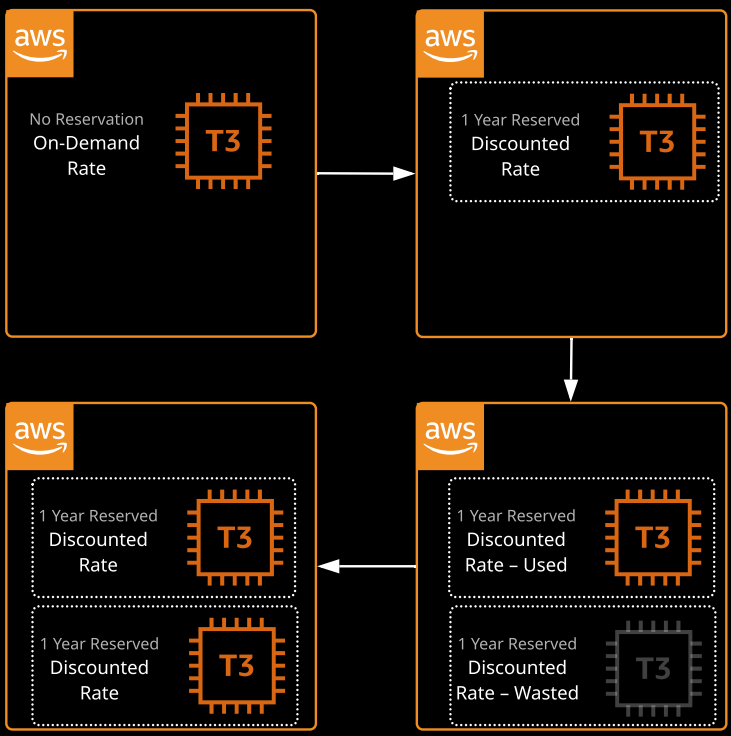
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4



Reserved instances lock in a reduced rate for **one** or **three** years. **Zonal** reserved instances include a **capacity** reservation. Your commitment incurs costs even if instances aren't launched. Reserved purchases are used for long-running, understood, and consistent workloads.

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

Exam Facts and Figures: Reserved and Spot

Key Facts

- Instance size/type have an AZ spot price.
- Bid more, instance provisioned for spot price. Less = termination.
- Spot fleets are containers, allowing capacity to be managed.
- Reservations are zonal (AZ) or regional.
- One or three years, no upfront, partial upfront, all upfront.
- You pay regardless of EC2 instance using a reservation.
- Regional is more flexible — but has no capacity reservation.

When to Use Reserved Purchases

- Base/consistent load
- Known and understood growth
- Critical systems/components

When to Use Spot Instances/Fleets

- Burst-y workloads
- Cost-critical, which can cope with interruption

When to Use On-Demand

- Default or unknown demand
- Anything in between reserved/spot
- Short-term workloads that cannot tolerate interruption

Go to Part 2
Back to Main

AWS and SA Fundamentals
Section 1

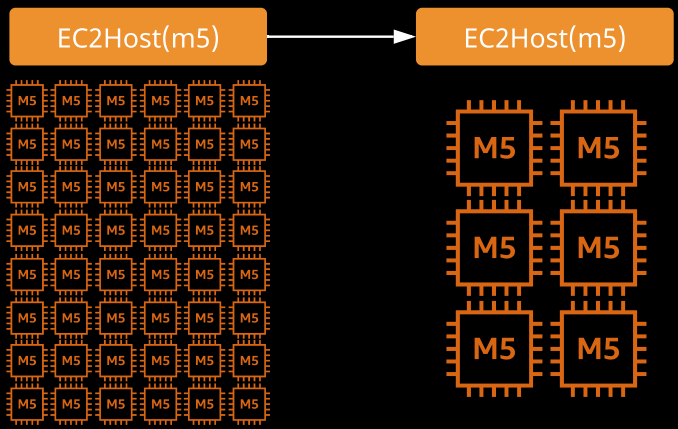
Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

Dedicated hosts are EC2 hosts for a given type and size that can be dedicated to you. The number of instances that can run on the host is fixed — depending on the type and size.



An on-demand or reserved fee is charged for the dedicated host — there are no charges for instances running on the host. Dedicated hosts are generally used when software is licensed per core/CPU and not compatible with running within a shared cloud environment.

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

This topic focuses on the "serverless" architecture — both in terms of general architecture and the use of AWS products and services.

Previous Section
Next

Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

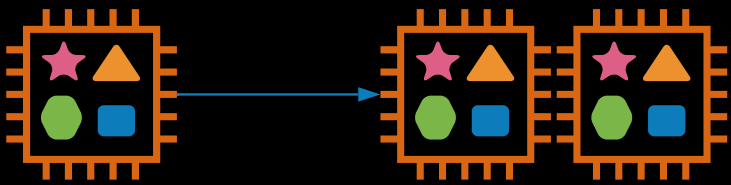
Identity and Access Control
Section 2

Compute
Section 3

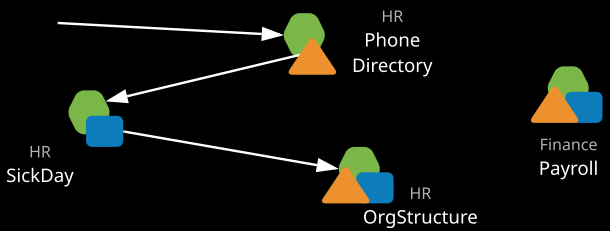
EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

A microservices architecture is the inverse of a monolithic architecture. Instead of having all system functions in one codebase, components are **separated into microservices** and operate independently. A microservice does one thing — and does it well. Operations, updates, and scaling can be done on a per-microservice basis.



Inflexible scaling: Either increasing the instance size or duplicating the instance



Microservices operate as independent applications. They allow direct communication between components and the end user. If one part of the system requires more capacity, that service can be scaled and updated as needed.

Back
Next

Go to Part 2
Back to Main

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

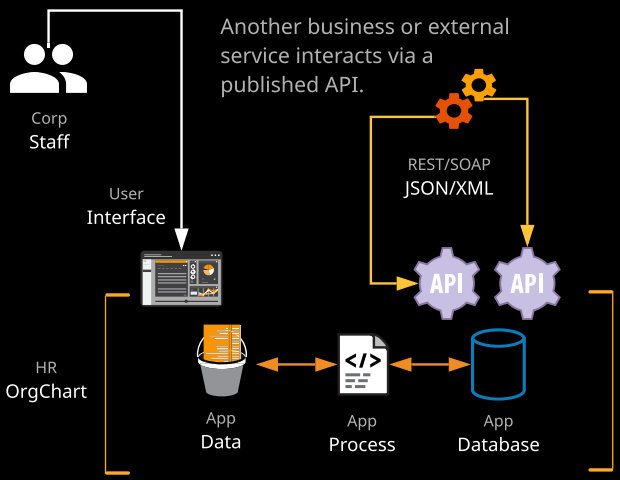
Compute
Section 3

- EC2 Fundamentals
- EC2 Intermediate
- EC2 Advanced
- Serverless Compute

Containers

Networking
Section 4

You access the AWS Console via a **user interface**, which is designed for a human being. An **API** (or **application programming interface**) is an interface accessed (consumed) by another **service** or **application**.



An API endpoint hosts one or more APIs and makes them available on a network (private or public internet). APIs remain static — they are abstracted from what the code inside the service is doing. API consumers don't care *how* things are done — only that the interface works. That's what allows lower-risk changes.

The AWS CLI tools use the AWS APIs.

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

When using an event-driven architecture, a system operates around "events" that represent **an action or a change of state** — e.g., a button being clicked, a file being uploaded, or a temperature dropping below a certain level. It's efficient because events are generated and pushed, rather than things being polled. Polling requires always-on compute and doesn't scale well.

Traditional

Event-Driven

Course Navigation

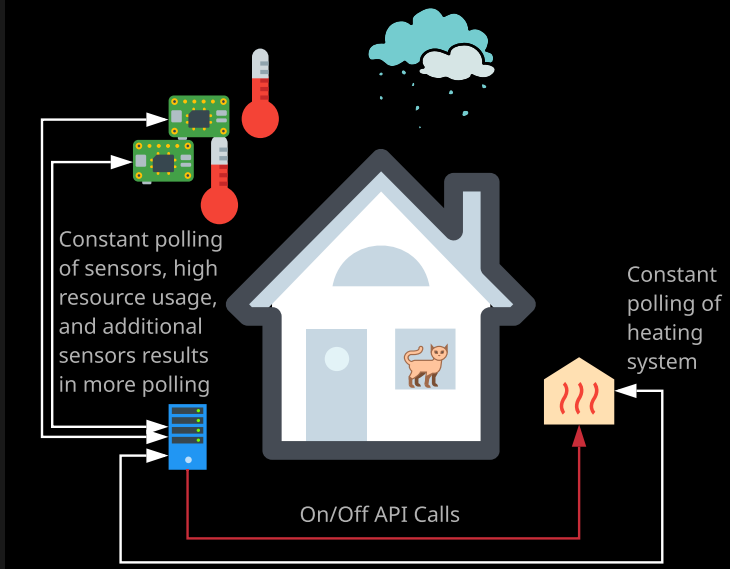
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4



Go to Part 2
Back to Main

Compute

Serverless Compute

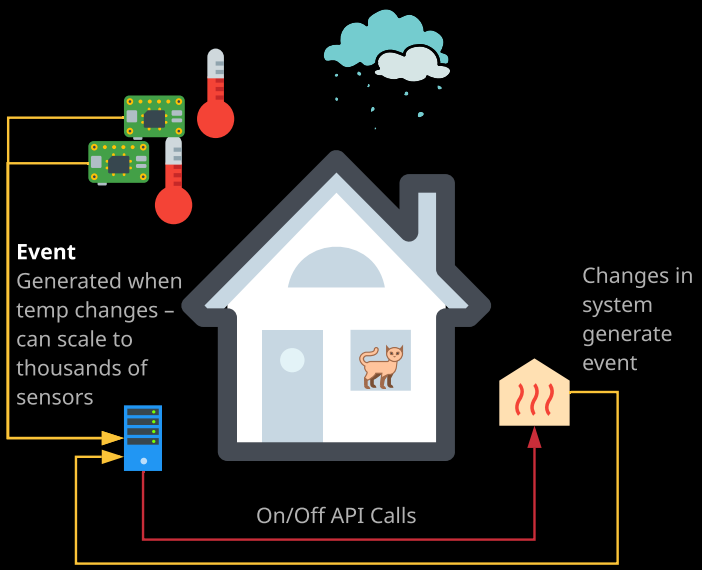
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

- EC2 Fundamentals
- EC2 Intermediate
- EC2 Advanced
- Serverless Compute
- Containers

Networking
Section 4



Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

- **EC2 Fundamentals**
- **EC2 Intermediate**
- **EC2 Advanced**
- **Serverless Compute**
- Containers

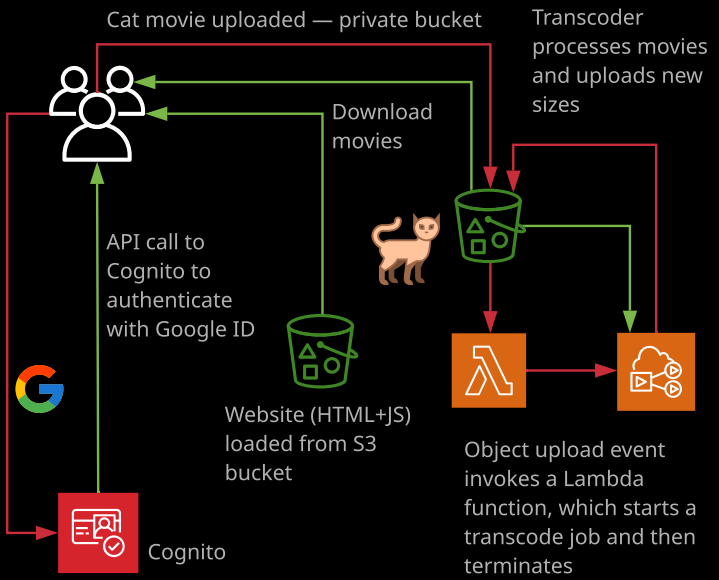
Networking
Section 4

Compute

Serverless Compute

Serverless architecture consists of two main principles, including **BaaS** (or **Backend as a Service**), which means using third-party services where possible rather than running your own. Examples include Auth0 or Cognito for authentication and Firebase or DynamoDB for data storage.

Serverless also means using an event-driven architecture where possible, using **FaaS** (or **Function as a Service**) products to provide application logic. These functions are only active (invoked) when they are needed (when an event is received).



AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

- EC2 Fundamentals
- EC2 Intermediate
- EC2 Advanced
- Serverless Compute

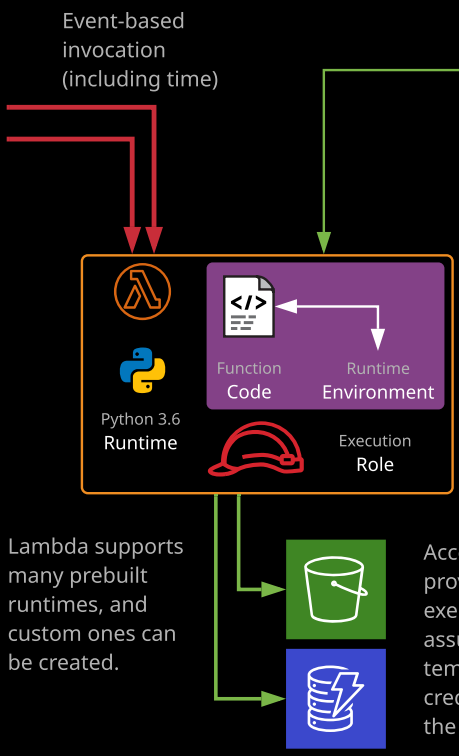
Containers

Networking
Section 4

Compute

Serverless Compute

Lambda is a **FaaS** product. Functions are code, which run in a runtime. Functions are invoked by **events**, perform actions for up to **15 minutes**, and terminate. Functions are also **stateless** — each run is clean.



Back
Next

Go to Part 2
Back to Main

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

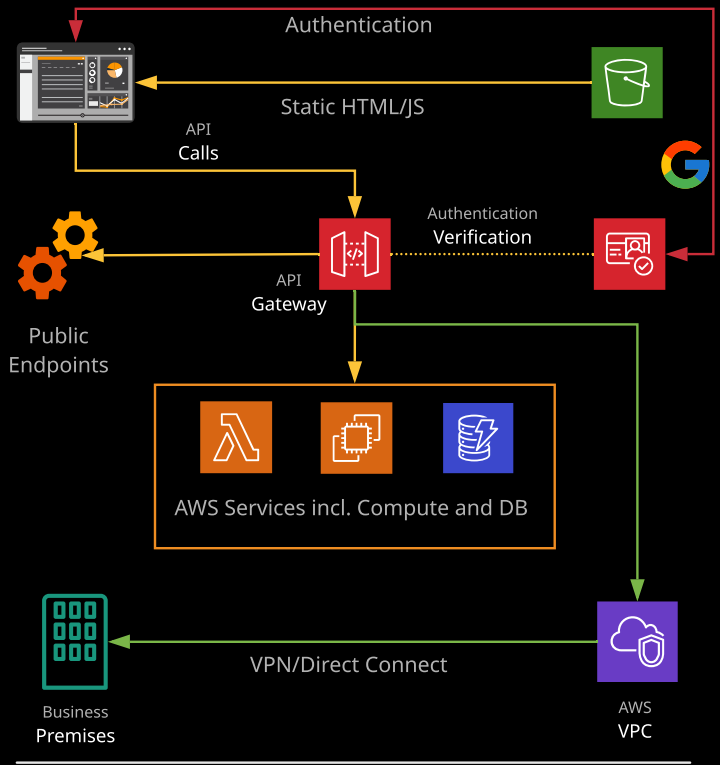
- EC2 Fundamentals
- EC2 Intermediate
- EC2 Advanced
- Serverless Compute
- Containers

Networking
Section 4

Compute

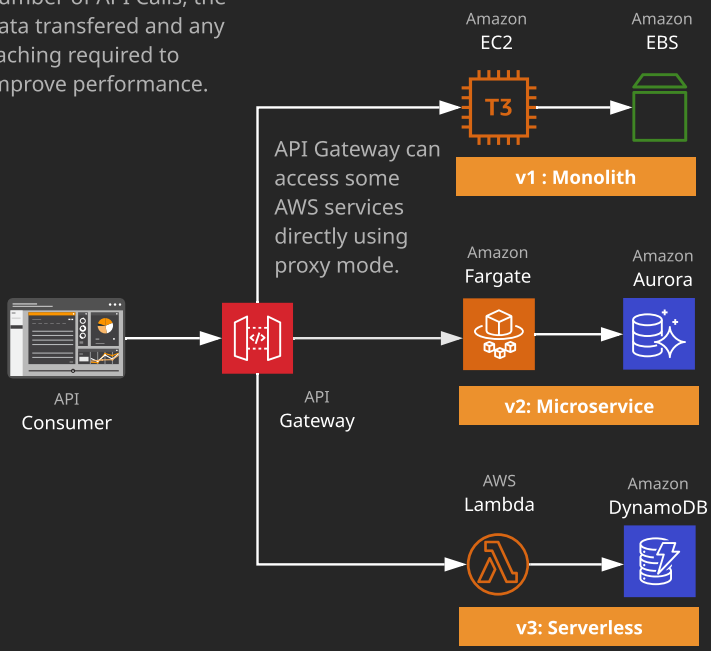
Serverless Compute

API Gateway is a managed API endpoint service. It can be used to create, publish, monitor, and secure APIs "as a service." API Gateway can use other AWS services for compute (FaaS/IaaS) as well as to store and recall data.



AWS and SA
Fundar
Sect
Identity a
Cor
Sect
Com
Sect
EC2 Funda
EC2 Interr
EC2 Advar
Serverless
Containers
Netw
Sect

Pricing is based on the number of API Calls, the data transfered and any caching required to improve performance.



APIs can be migrated to API Gateway in a monolithic form, and gradually moved to a microservices architecture and then once components have been fully broken up a serverless & FaaS based architecture.

X

Go to Part 2
Back to Main

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

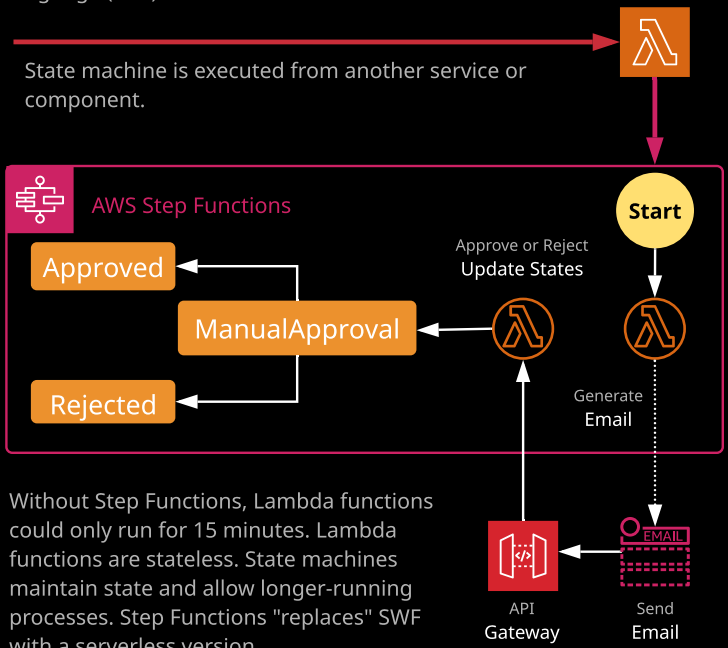
- **EC2 Fundamentals**
- **EC2 Intermediate**
- **EC2 Advanced**
- **Serverless Compute**
- Containers

Networking
Section 4

Compute

Serverless Compute

Step Functions is a **serverless visual workflow service** that provides **state machines**. A state machine can orchestrate other AWS services with simple logic, branching, and parallel execution, and it maintains a **state**. Workflow steps are known as **states**, and they can perform work via **tasks**. Step Functions allows for **long-running serverless workflows**. A state machine can be defined using Amazon States Language (**ASL**).



Without Step Functions, Lambda functions could only run for 15 minutes. Lambda functions are stateless. State machines maintain state and allow longer-running processes. Step Functions "replaces" SWF with a serverless version.

Course Navigation

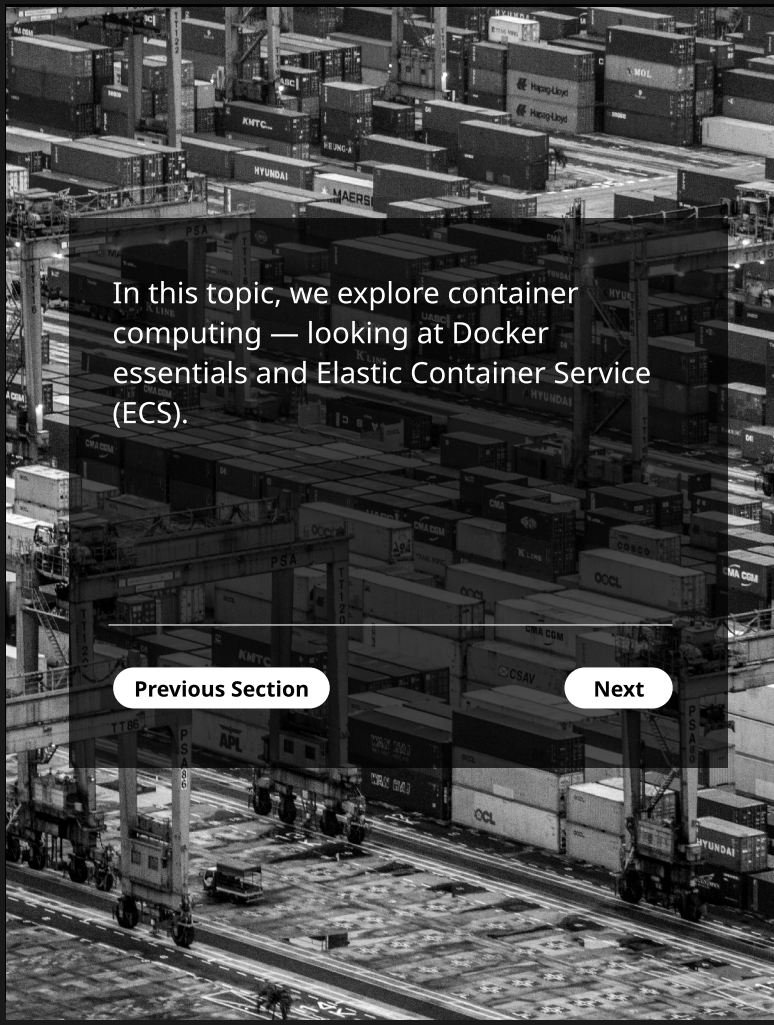
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

- EC2 Fundamentals
- EC2 Intermediate
- EC2 Advanced
- Serverless Compute
- Containers

Networking
Section 4



In this topic, we explore container computing — looking at Docker essentials and Elastic Container Service (ECS).

Previous Section

Next

Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

- EC2 Fundamentals
- EC2 Intermediate
- EC2 Advanced
- Serverless Compute
- Containers

Networking
Section 4

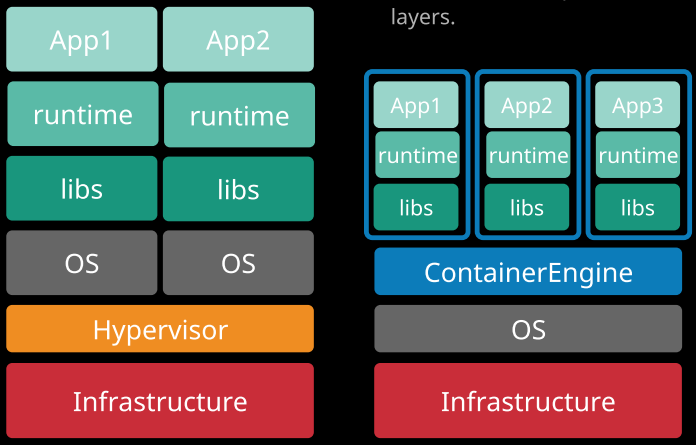
Compute

Containers

A container is a package that contains an application, libraries, and file system required to run it. Containers run on a container engine that generally runs within a single OS, such as Linux. Containers provide the isolation benefits of virtualization but are more lightweight, allowing faster starts and more dense packing within a host.

A popular container engine is **Docker**, which is the basis for Elastic Container Service (**ECS**).

An image is a collection of file system layers. Docker file systems are differential — each layer stores differences from previous layers.



Virtual Machines/Instances

Containers

Back

Next

Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

Compute

Containers

ECS is a managed container engine. It allows Docker containers to be deployed and managed within AWS environments. ECS can use infrastructure clusters based on EC2 or Fargate where AWS manages the backing infrastructure.

EC2 Mode

Fargate Mode

Back

Next Section

Go to Part 2

Back to Main

Course Navigation

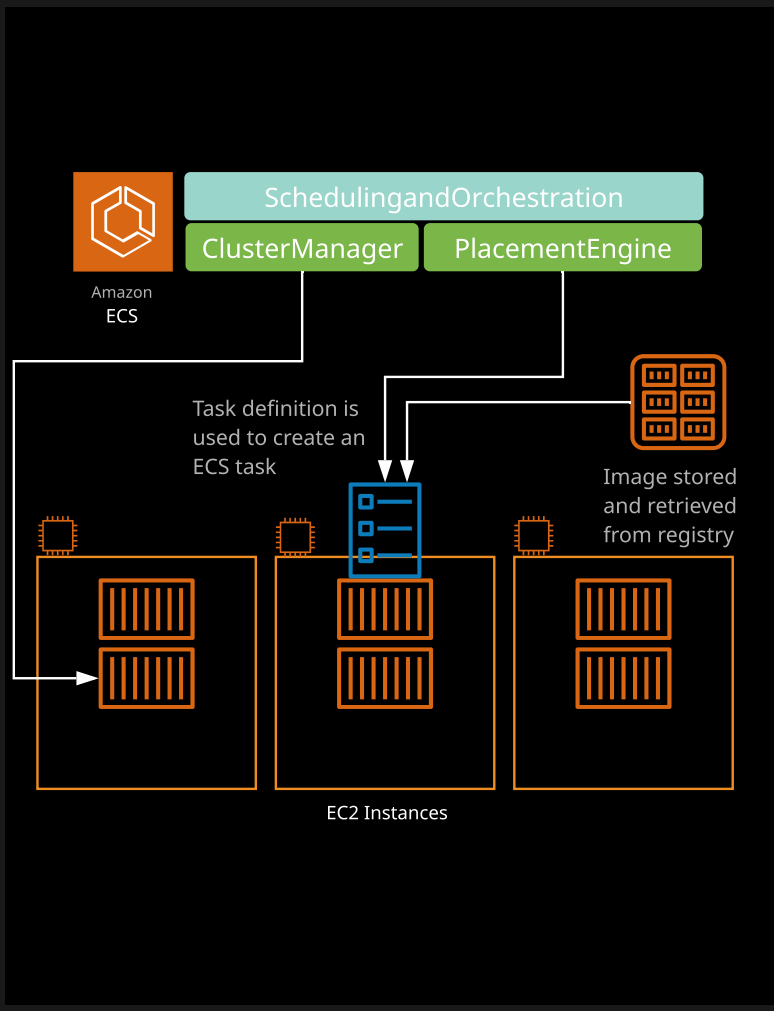
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

- EC2 Fundamentals
- EC2 Intermediate
- EC2 Advanced
- Serverless Compute
- Containers

Networking
Section 4



Go to Part 2
Back to Main

Course Navigation

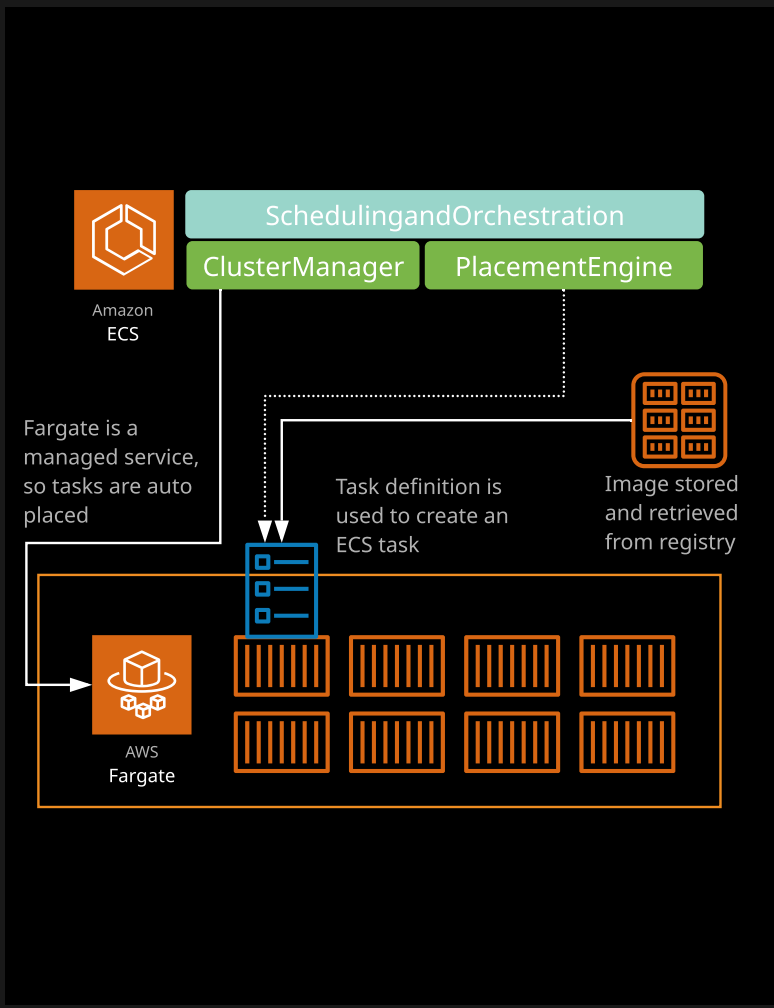
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

- EC2 Fundamentals
- EC2 Intermediate
- EC2 Advanced
- Serverless Compute
- Containers

Networking
Section 4



Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

EC2 Fundamentals
EC2 Intermediate
EC2 Advanced
Serverless Compute
Containers

Networking
Section 4

Exam Hints: Elastic Container Service (ECS)

Cluster
A logical collection of ECS resources — either ECS EC2 instances or a logical representation of managed Fargate infrastructure

Task Definition
Defines your application. Similar to a Dockerfile but for running containers in ECS. Can contain multiple containers.

Container Definition
Inside a task definition, a container definition defines the individual containers a task uses. It controls the CPU and memory each container has, in addition to port mappings for the container.

Task
A single running copy of any containers defined by a task definition. One working copy of an application (e.g., DB and web containers).

Service
Allows task definitions to be scaled by adding additional tasks. Defines minimum and maximum values.

Registry
Storage for container images (e.g., ECS Container Registry or Dockerhub). Used to download image to create containers.

Go to Part 2
Back to Main

Course Navigation

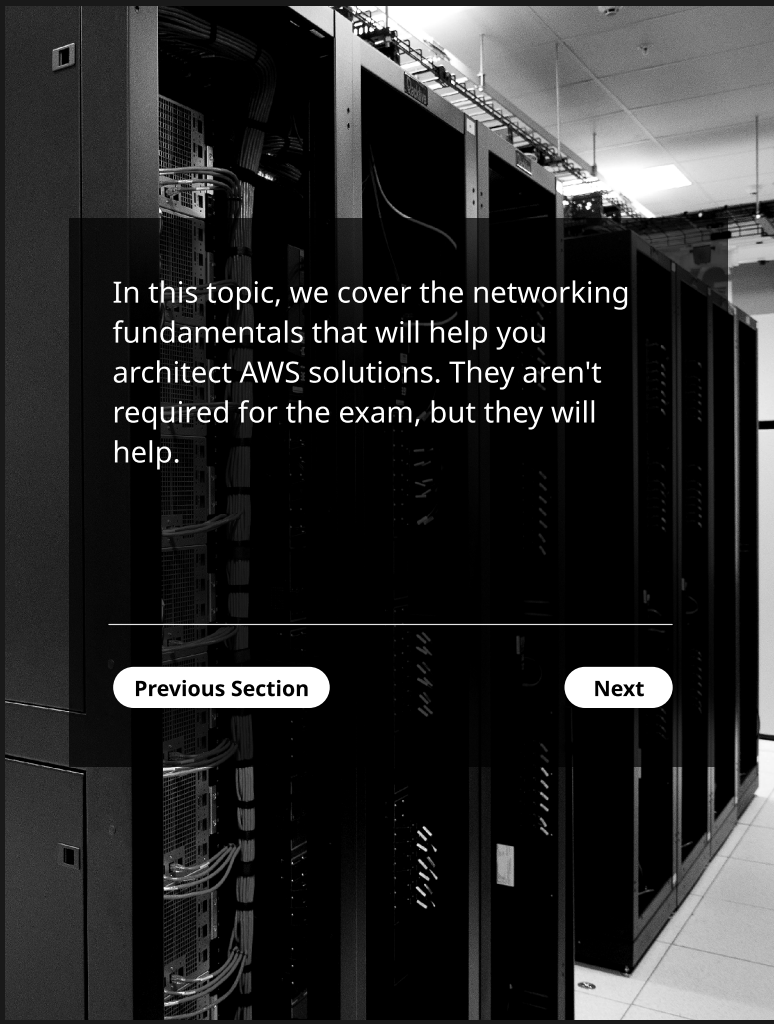
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53



In this topic, we cover the networking fundamentals that will help you architect AWS solutions. They aren't required for the exam, but they will help.

Previous Section

Next

Go to Part 2

Back to Main

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

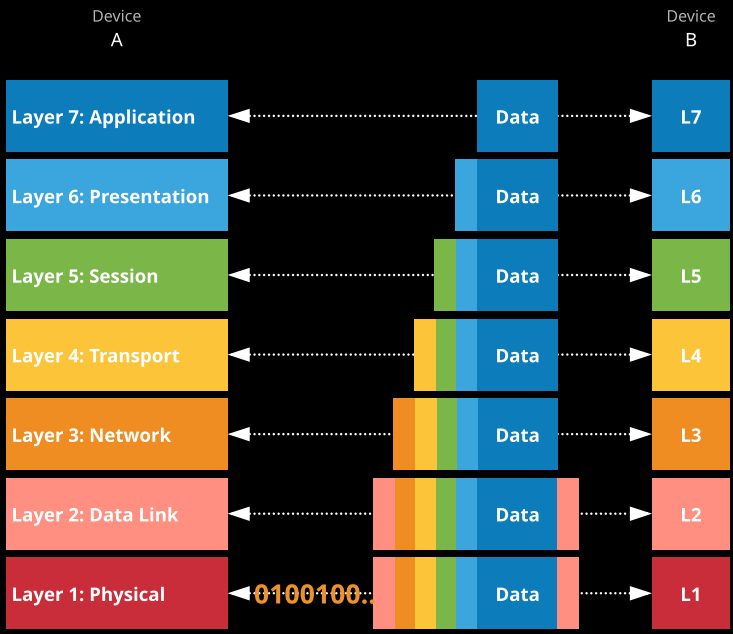
Networking
Section 4

- Network Fundamentals
- AWS Private Networking
- Advanced VPC
- AWS DNS Fundamentals
- Advanced Route 53

Networking

Network Fundamentals

The Open Systems Interconnection (OSI) Model is a standard used by networking manufacturers globally. It was created and published in 1984; it splits all network communications into seven layers. Each layer serves the layer that's above it plus the layer beneath it which adds additional capabilities. Data between two devices travels down the stack on the *device's A-side* (wrapped at each layer) and gets transmitted before moving up the stack at the *devices B-side* (where the wrapping gets stripped away at every stage). This data wrapping process is called *encapsulation.*



Back Next

Go to Part 2

Back to Main

Course Navigation

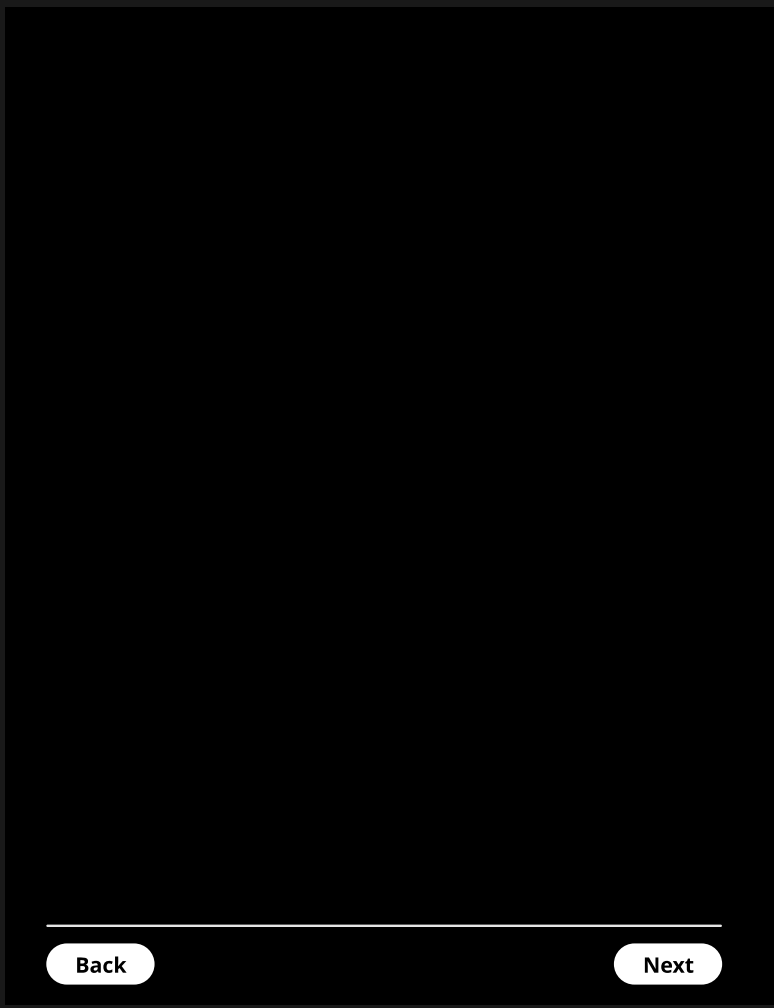
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53



AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

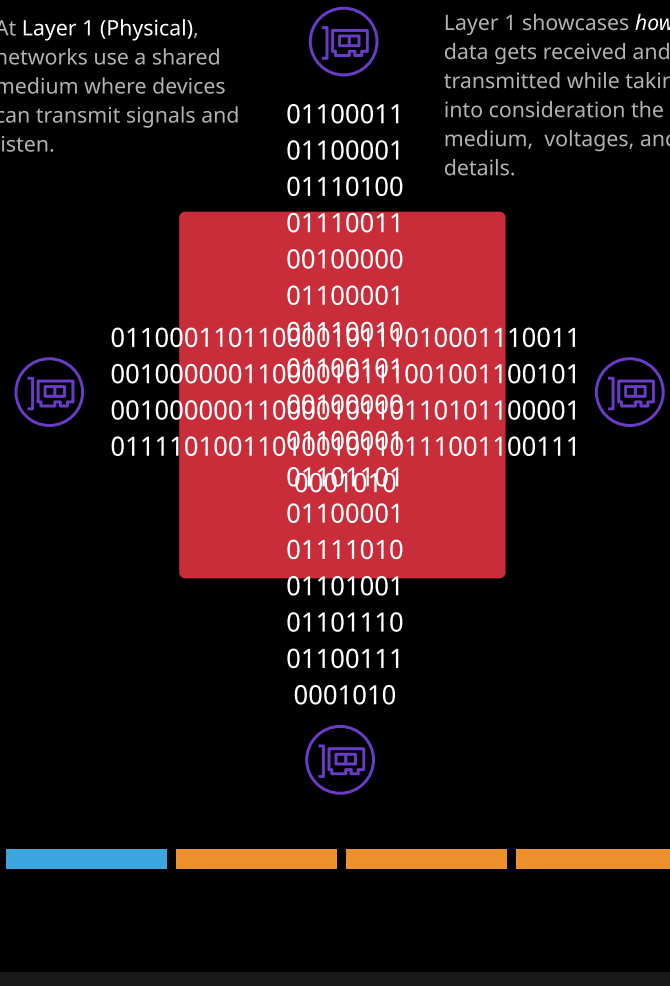
- Network Fundamentals
- AWS Private Networking
- Advanced VPC
- AWS DNS Fundamentals
- Advanced Route 53

Networking

Network Fundamentals

At Layer 1 (Physical), networks use a shared medium where devices can transmit signals and listen.

Layer 1 showcases *how* data gets received and transmitted while taking into consideration the medium, voltages, and RF details.



Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

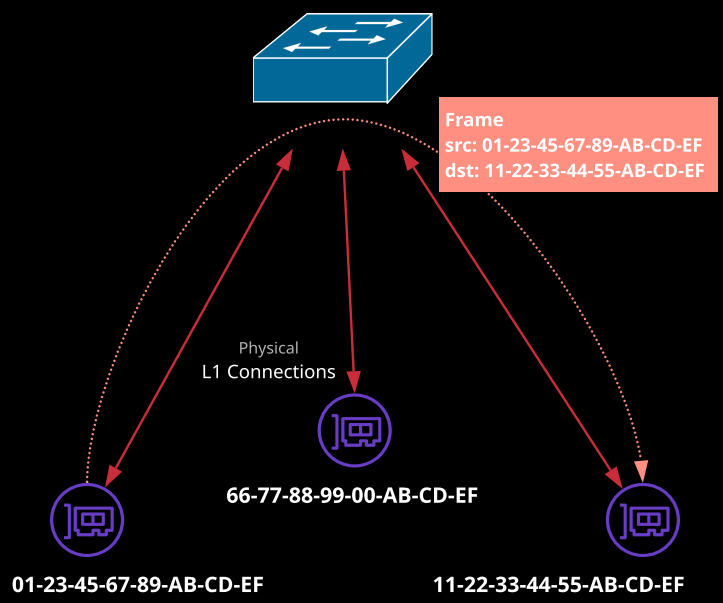
Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Layer 2 (Data Link) adds MAC addresses (01-23-45-67-89-AB-CD-EF) that can be used for **named** communication between two devices on a local network. Additionally, layer 2 adds controls over the media, avoiding cross-talk, this allows a back-off time and retransmission.



L2 communications use L1 to broadcast and listen. L2 runs on top of L2.

Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

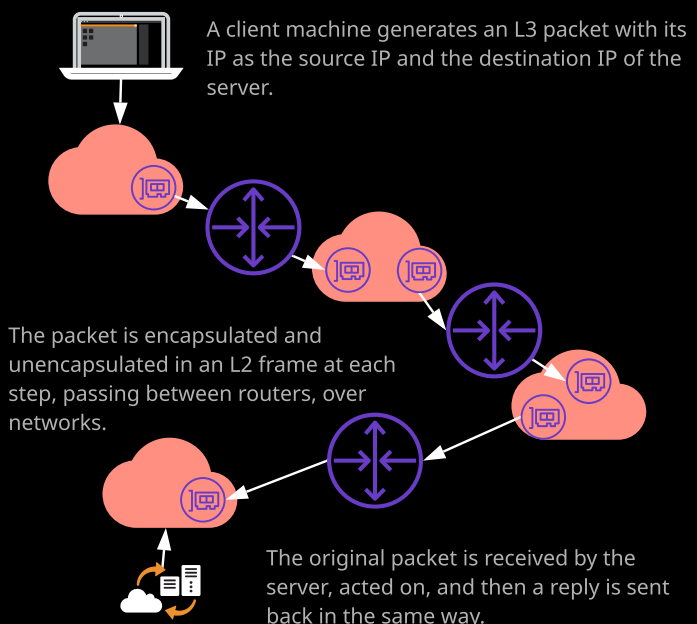
Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

The Network Layer (L3) allows for unique device-to-device communication over interconnected networks. L3 devices can pass packets over tens or even hundreds of L2 networks. The packets remain largely unchanged during this journey — traveling within different L2 frames as they pass over various networks.



Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

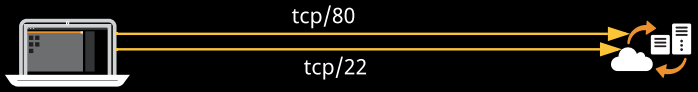
Identity and Access Control
Section 2

Compute
Section 3

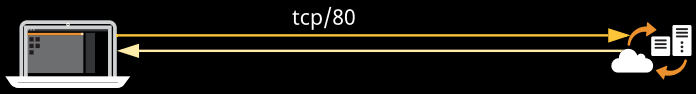
Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

L3 allows for an IP to communicate with another IP — but only a single stream, so one conversation between the two.



L4 (**Transport**) adds **TCP** and **UDP**. TCP is designed for reliable transport, and UDP is aimed at speed. TCP uses **segments** to ensure data is received in the **correct order** and adds **error checking** and "**ports**," allowing different streams of communications to the same host (e.g., tcp/22 and tcp/80).



L5 (**Session**) adds the concept of sessions, so that request and reply communication streams are viewed as a single "session" of communication between client and server.

L6 (**Presentation**) adds data conversion, encryption, compression, and standards that **L7** can use. L7 (**Application**) is where protocols (such as HTTP, SSH, and FTP) are added. For example, HTTP (L7) running over TLS (L6) is HTTPS.

Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Back

Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

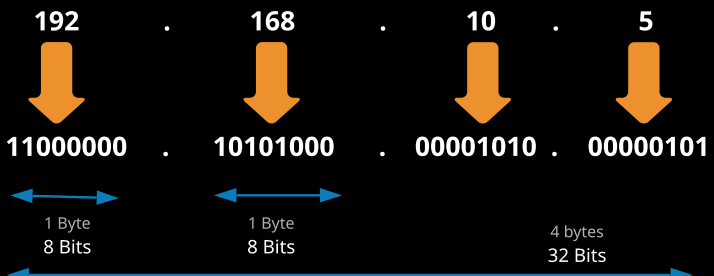
Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

IPv4 addresses are how two devices can communicate at layer 4 and above of the OSI seven-layer model. IP addresses (IPs) are actually 32-bit binary values but are represented in dotted-decimal notation to make them easier for humans to read and understand.

IPv4 Address (Dotted-Decimal Notation)



IPs are split into a **network** part and a **node** or **host** part. The netmask (e.g., 255.255.255.0) or prefix (e.g., /24) shows where this split occurs.

IP	192	168	10	5
Binary	11000000	10101000	00001010	00000101
Subnet Mask	255	255	255	0
Prefix /24	11111111	11111111	11111111	

Network Part = 1's **Node = 0's**

Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Within the IPv4 address space (0.0.0.0 to 255.255.255.255), there are certain addresses that are **reserved** or **special** in some way:

- **0.0.0.0** and **0.0.0.0/0**: Represents **all** IP addresses
- **255.255.255.255**: IP address used to broadcast to all IP addresses everywhere (this is generally filtered and not passed between networks)
- **127.0.0.1**: Localhost or loopback. Whatever the IP address of the device you are using, it can be referenced by itself as 127.0.0.1. So a web server on your laptop will always be ip:80 or 127.0.0.1:80
- **169.254.0.1** to **169.254.255.254**: A range of IP addresses that a device can auto configure with if it's using DHCP and fails to automatically get an IP from a DHCP server.

Historically IP addresses were split into classes: (including)

- Class **A** (/**8**): **1.0.0.0** to **126.255.255.255** — **126** networks, **16,777,214** nodes in each (+2 reserved)
- Class **B** (/**16**): **128.0.0.0** to **191.255.255.255** — **16,382** networks, **65,534** nodes in each (+2 reserved).
- Class **C** (/**24**): **192.0.0.0** to **223.255.255.255** — **2,097,150** networks, **254** nodes in each (+2 reserved)

Class A networks were initially allocated to large organizations, Class B to medium, and Class C to small businesses. As the supply of IPv4 addresses became low, the class system of IPs were related with CIDR (**more on this next**).

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

IP classes have a number of ranges within them used for private networking only:

- **10.0.0.0** to **10.255.255.255**: Private networking within the Class A range
- **172.16.0.0** to **172.31.255.255**: Private networking within the Class B range (16 Class B networks)
- **192.168.0.0** to **192.168.255.255**: Private networking within the Class C range (256 Class C networks)

These ranges are often used on private business networks, cloud networks, and home networks.

Go to Part 2
Back to Main

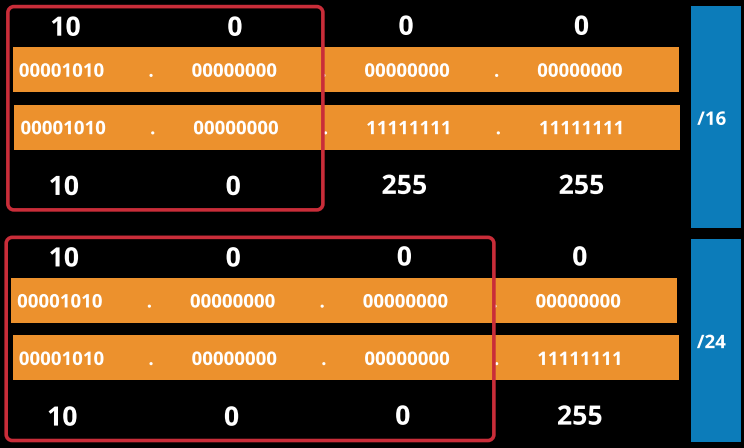
Networking

Network Fundamentals

CIDR (Classless Inter-Domain Routing) is used for IPv4 IP networking rather than the class system — it allows more effective allocation and subnetworking.

Either you are allocated a network range to use, or you decide on it. It will be represented as network/prefix (e.g., 10.0.0.0/16).

The network address is your starting point. The prefix is the number of bits the network uses, the remaining bits, and the node part is yours to use. The node (or host) part is yours from all 0's to all 1's.



Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

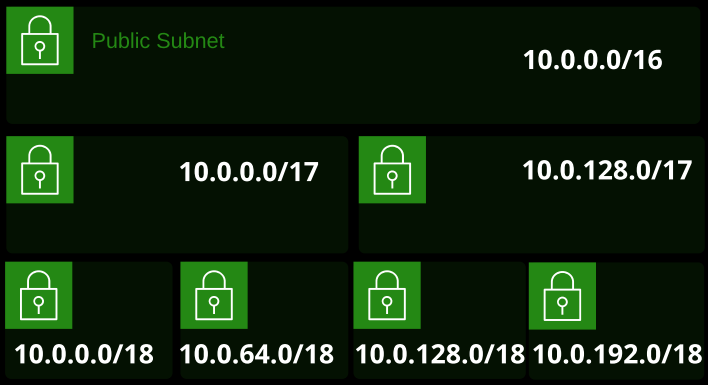
Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Subnetting is a process of breaking a network down into smaller subnetworks. You might be allocated a public range for your business or decide on a private range for a VPC. Subnetting allows you to break it into smaller allocations for use in smaller networks (e.g., VPC subnets).

If you pick 10.0.0.0/16 for your VPC, it's a single network from 10.0.0.0 to 10.0.255.255 and offers 65,536 addresses. That VPC could have a single subnet within it that's also 10.0.0.0/16.



With a certain size of VPC, increasing the prefix creates two smaller networks. Increasing again creates four even smaller networks. Increasing *again* creates eight even smaller — and so on.

You won't need to know this from memory — there are plenty of cheat sheets available to help you along the way.

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Local device-to-device communication takes place using L1 (Physical) and L2 (Data Link) using MAC addresses and physical 0's and 1's. This doesn't scale across LANs, so a method of network-to-network transit is needed. IP routing provides this. The method used depends on if the two devices are local, in a known remote network, or in an unknown network.

Local | Known | Unknown

Back | Next

Go to Part 2

Back to Main

Course Navigation

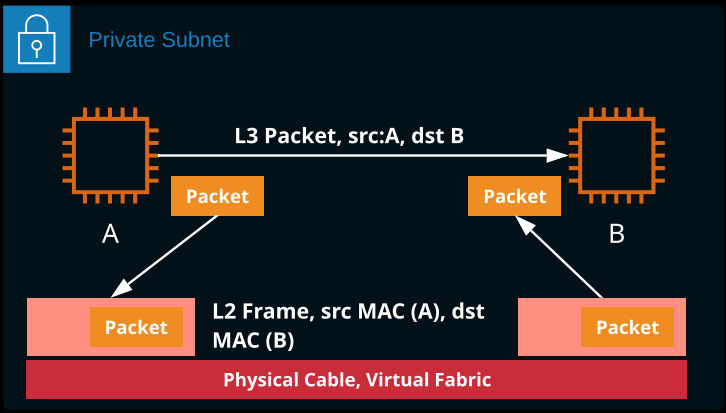
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53



IP-to-IP communication that occurs locally doesn't use a router. ARP is used to find the MAC address for the destination IP address. The IP packet is created at L3 and passed to L2, where it's encapsulated inside an ethernet (L2) frame. The frame is sent to the destination MAC address. Once received, the L2 frame is removed and the IP packet is passed to L3.

Go to Part 2

Back to Main

Course Navigation

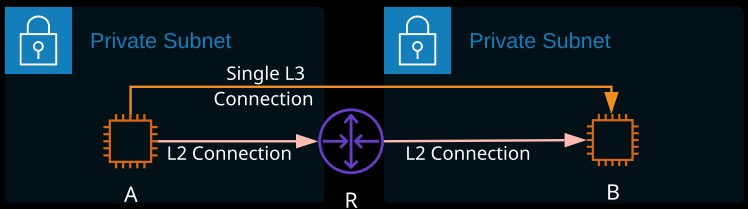
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53



If Instance A wants to communicate with Instance B, it can use its IP and subnet mask to determine if B is local. If it's not, then the following process occurs:

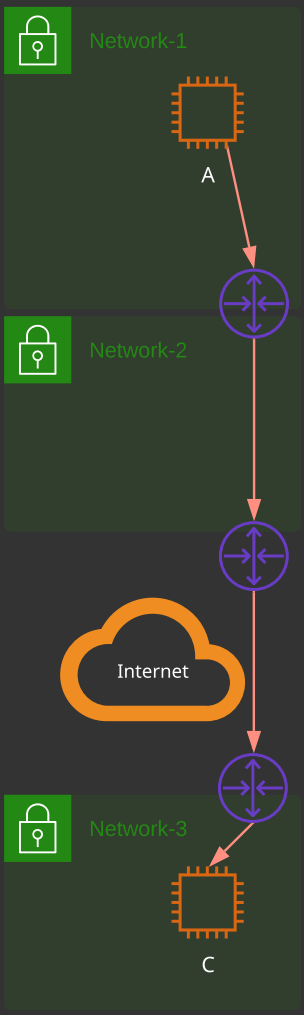
- "A" generates an L3 packet — the SRC is the IP-"A", the DST is IP-"B"
- "A" knows its default gateway (Router) IP, so it uses ARP to find the Router MAC
- "A" passes the L3 packet to L2, wraps it in an L2 frame, and sends this to the R-MAC address (not the MAC address of B)
- "R" receives this, strips away the Layer 2 frame, and checks the DST IP
- It knows the network of IP-"B" because it's connected to it
- "R" uses ARP to find the MAC of "B," generates a frame TO "B", puts the unaltered IP packet inside, and sends to MAC-"B"
- "B" receives the frame, strips it away, and passes the packet to L3

Go to Part 2

Back to Main

Routing

Unknown Network



Routing works equally well whether the network of the remote instance is known or not. In this case, Instance A is attempting to communicate with Instance C.

Instance A knows Instance C is not local, so it creates an IP packet with a dst of Instance C. It passes the packet down to L2 and asks for it to be addressed to the MAC address of Router A (its default router/route).

Router A

Router A strips the L2 frame and reviews the destination address of the L3 packet. It doesn't know Network 3, so it has no knowledge of how to get there. It does have a "default router," which is Router B. It creates an L2 frame with a dst MAC of Router B and wraps it around the unchanged packet.

Router B

The internet uses a routing protocol called Border Gateway Protocol (BGP). This protocol exchanges routes. Router C would advertise Network 3, and Router B would learn about Network 3 via Router C. Router B would advertise Network 3 via Router C. Router A would learn all Router B's routes and all routes it knows about.

Router C

Router C receives the L2 frame, strips it away, and reviews the L3 packet. It now knows it's in the same network, and it finds the MAC address of the DST IP address of C. A new L2 frame is created, with a dst MAC address of C, and it forwards it in.

At scale, this is how the internet works: Unchanged packets being passed around from router to router, each time using a new L2 connection.

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

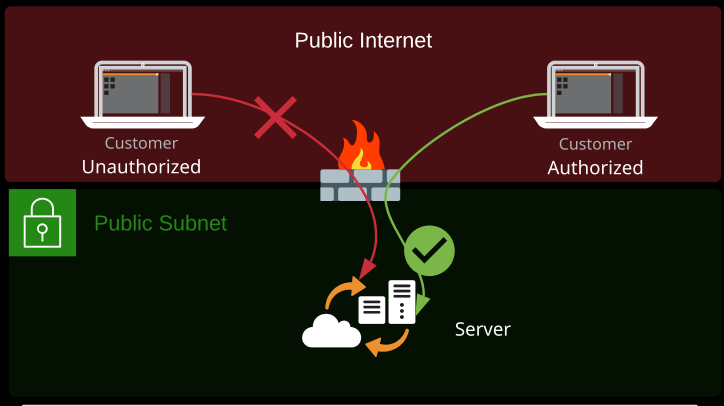
Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

A firewall is a device that historically sits at the border between different networks and monitors traffic flowing between them. A firewall is capable of reading packet data and either **allowing** or **denying** traffic based on that data.

Firewalls establish a **barrier between networks** of different security levels and historically have been the first line of defense against perimeter attacks.

What data a firewall can read and act on depends on the OSI layer the firewall operates at:

- Layer 3 (**Network**): Source/destination IP addresses or ranges
- Layer 4 (**Transport**): Protocol (TCP/UDP) and port numbers
- Layer 5 (**Session**): As layer 4, but understand response traffic
- Layer 7 (**Application**): Application specifics (e.g., HTML paths, images)



Back
Next

Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

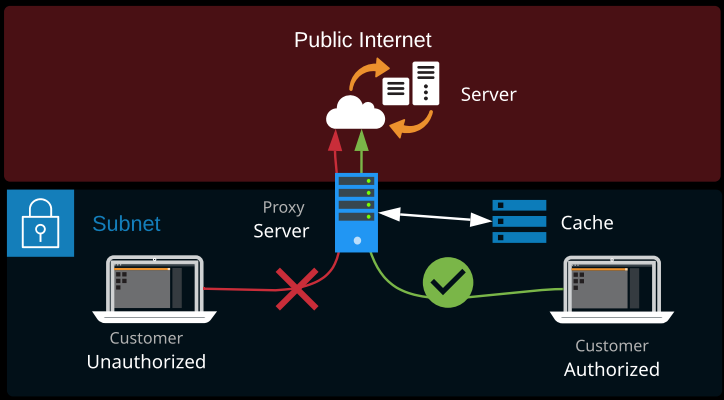
Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

A proxy server is a type of gateway that sits between a private and public network (e.g., the internet). A proxy server is something that generally needs application support and is configured in the operating system, a web browser, or another application.

The client makes a connection to the proxy server, and the proxy makes a connection to the destination server. Proxy servers can provide filtering (child safety, malware, removing adult content) or it can act as a web cache, speeding up web access for a large organization at a remote site.

Proxy servers can also choose to pass on traffic or not based on things network layer appliances can't, like username or elements of a corporate identity — department, age, security privilege, or the DNS name rather than IP (**remember this for the exam**).



Back
Next Topic

Go to Part 2
Back to Main

Course Navigation

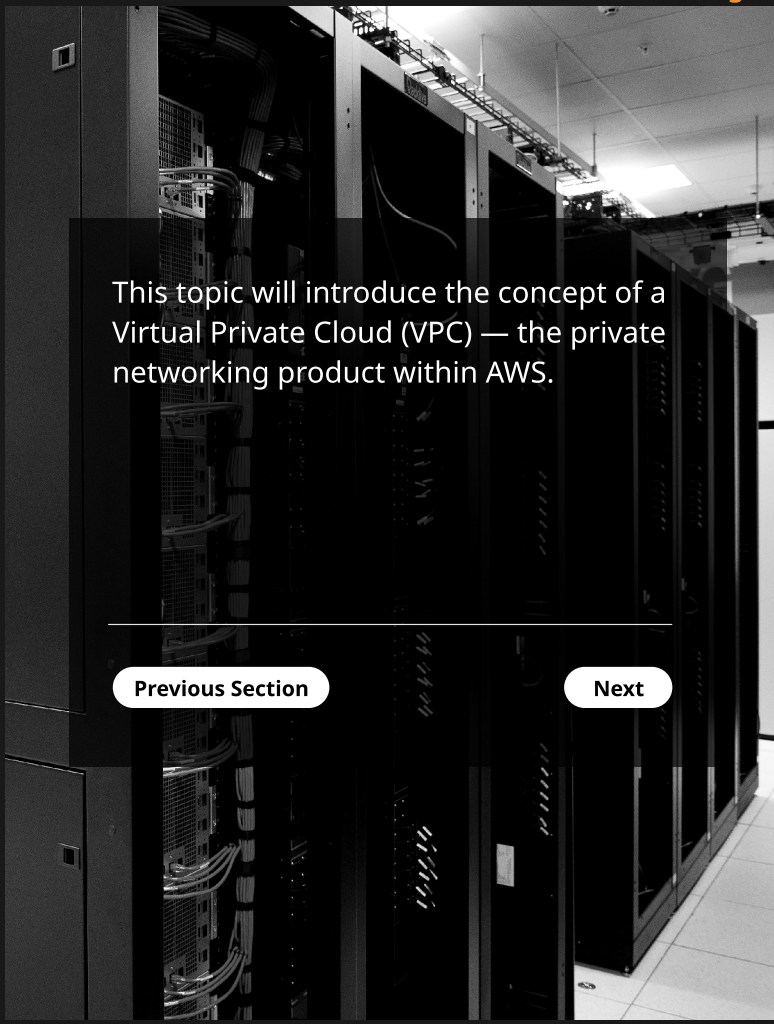
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53



This topic will introduce the concept of a Virtual Private Cloud (VPC) — the private networking product within AWS.

Previous Section

Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Virtual Private Cloud (VPC):

- A private network within AWS. It's your private data center inside the AWS platform.
- Can be configured to be public/private or a mixture
- Regional (can't span regions), highly available, and can be connected to your data center and corporate networks
- Isolated from other VPCs by default
- VPC and subnet: max /16 (65,536 IPs) and minimum /28 (16 IPs)
- VPC subnets can't span AZs (1:1 mapping)
- Certain IPs are reserved in subnets (*see architecture diagram)*

Region Default VPC:

- Required for some services, used as a default for most
- Pre-configured with all required networking/security
- Configured using a /16 CIDR block (172.31.0.0/16)
- A /20 public subnet in each AZ, allocating a public IP by default
- Attached internet gateway with a "main" route table sending all IPv4 traffic to the internet gateway using a 0.0.0.0/0 route
- A default DHCP option set attached.
- SG: Default — all from itself, all outbound
- NACL: Default — allow all inbound and outbound

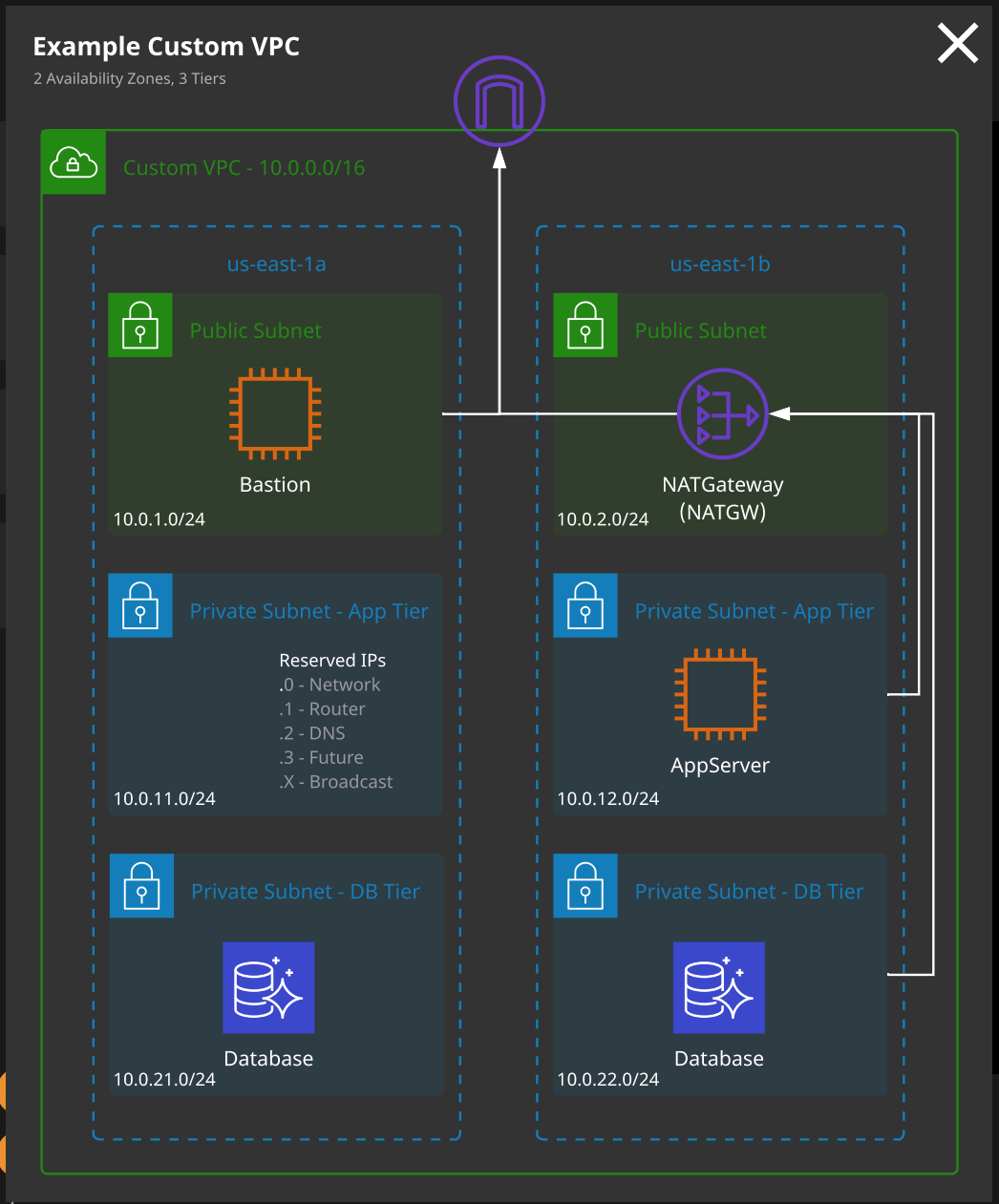
Custom VPC:

- Can be designed and configured in any valid way
- You need to allocate IP ranges, create subnets, and provision gateways and networking, as well as design and implement security.
- When you need multiple tiers or a more complex set of networking
- Best practice is to **not** use default for most production things

Back | VPC Architecture | Next

Go to Part 2

Back to Main


Example Custom VPC
2 Availability Zones, 3 Tiers
Custom VPC - 10.0.0.0/16
us-east-1a
Public Subnet
Bastion
10.0.1.0/24
Private Subnet - App Tier
Reserved IPs
.0 - Network
.1 - Router
.2 - DNS
.3 - Future
.X - Broadcast
10.0.11.0/24
Private Subnet - DB Tier
Database
10.0.21.0/24
us-east-1b
Public Subnet
NATGateway
(NATGW)
10.0.2.0/24
Private Subnet - App Tier
AppServer
10.0.12.0/24
Private Subnet - DB Tier
Database
10.0.22.0/24

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

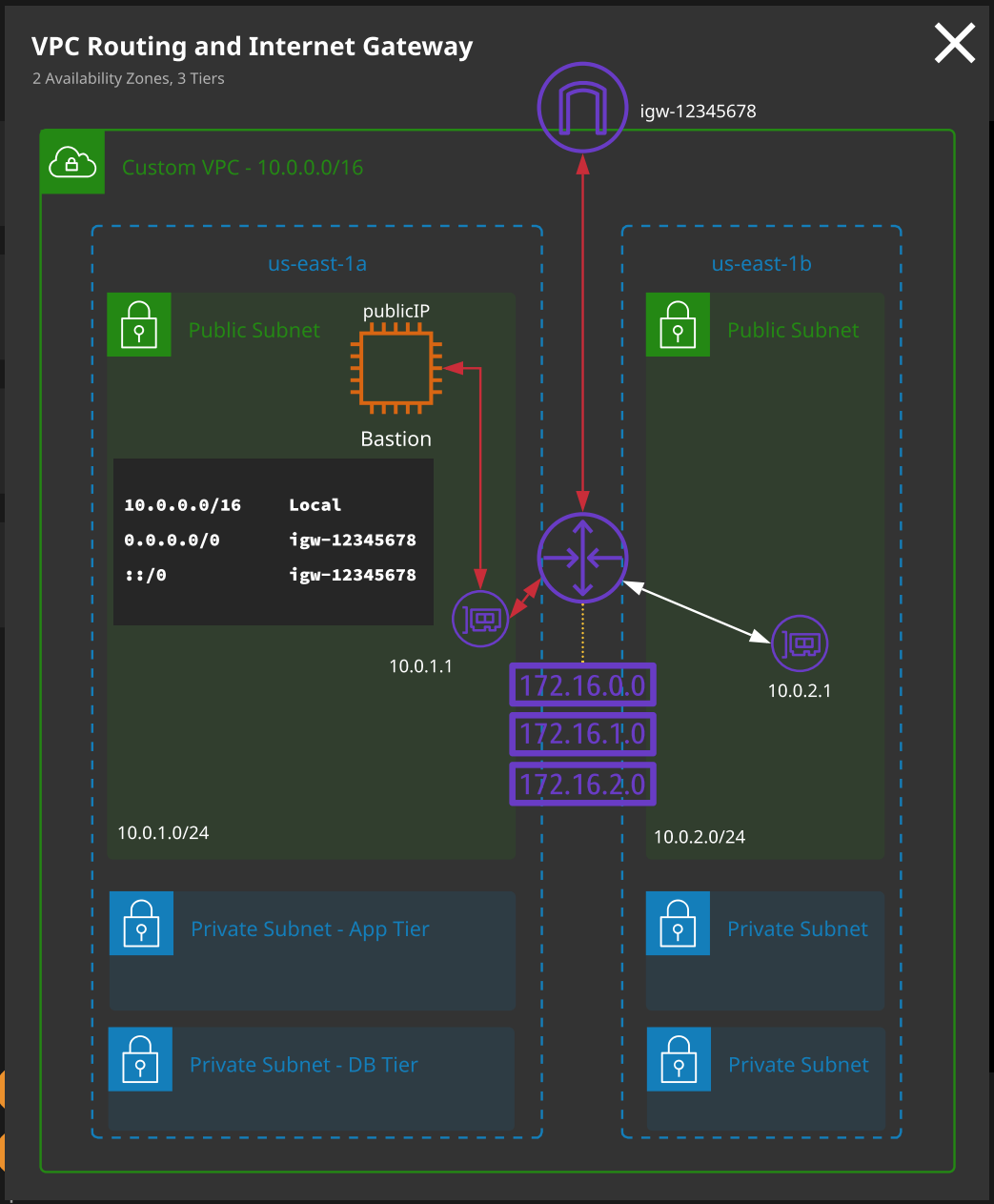
Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

VPC Routing:

- Every VPC has a virtual routing device called the VPC router.
- It has an interface in any VPC subnet known as the "subnet+1" address — for 10.0.1.0/24, this would be 10.0.1.1/32.
- The router is highly available, scalable, and controls data entering and leaving the VPC and its subnets.
- Each VPC has a "main" route table, which is allocated to all subnets in the VPC by default. A subnet must have one route table.
- Additional "custom" route tables can be created and associated with subnets — but only one route table (RT) per subnet.
- A route table controls what the VPC router does with traffic leaving a subnet.
- An internet gateway is created and attached to a VPC (1:1). It can route traffic for public IPs **to** and **from** the internet.

Routes:

- A RT is a collection of routes that are used when traffic **from** a subnet arrives at the VPC router.
- Every route table has a **local** route, which matches the CIDR of the VPC and lets traffic be routed between subnets.
- A route contains a **destination** and a **target**. Traffic is forwarded to the target if its destination matches the route destination.
- If multiple routes apply, the most specific is chosen. A /32 is chosen before a /24, before a /16.
- Default routes (0.0.0.0/0 v4 and ::/0 v6) can be added that match any traffic not already matched.
- Targets can be IPs or AWS networking gateways/objects
- A subnet is a public subnet if it is (1) configured to allocate public IPs, (2) if the VPC has an associated internet gateway, and (3) if that subnet has a default route **to** that internet gateway.


VPC Routing and Internet Gateway
2 Availability Zones, 3 Tiers
igw-12345678
Custom VPC - 10.0.0.0/16
us-east-1a
us-east-1b
Public Subnet
publicIP
Bastion
Public Subnet
10.0.0.0/16 Local
0.0.0.0/0 igw-12345678
::/0 igw-12345678
10.0.1.1
172.16.0.0
172.16.1.0
172.16.2.0
10.0.2.1
10.0.1.0/24
10.0.2.0/24
Private Subnet - App Tier
Private Subnet
Private Subnet - DB Tier
Private Subnet

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

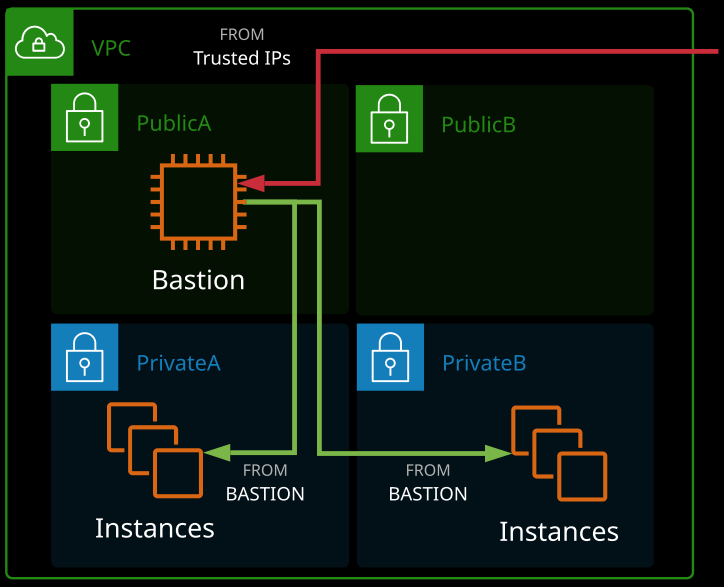
Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Bastion Hosts (or Jumpboxes):

- A host that sits at the perimeter of a VPC
- It functions as an entry point to the VPC for trusted admins.
- Allows for updates or configuration tweaks remotely while allowing the VPC to stay private and protected
- Generally connected to via SSH (Linux) or RDP (Windows)
- Bastion hosts must be kept updated, and security hardened and audited regularly
- Multifactor authentication, ID federation, and/or IP blocks.



Back

Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

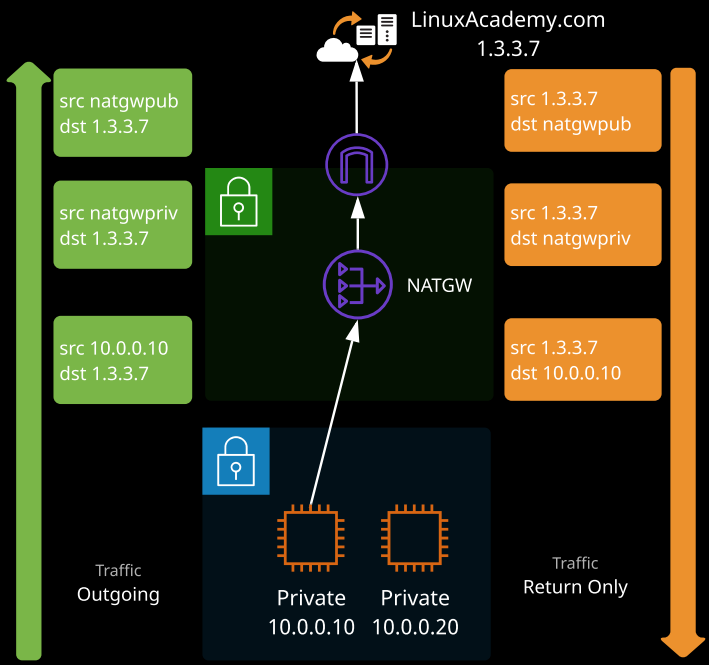
Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Network address translation (**NAT**) is a method of remapping source IPs or destination IPs of packets. It can be used in a number of ways.

- **Static NAT**: A private IP is mapped to a public IP (what IGWs do)
- **Dynamic NAT**: A range of private addresses are mapped onto one or more public (used by your home router and NAT gateways)



Back Next

Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

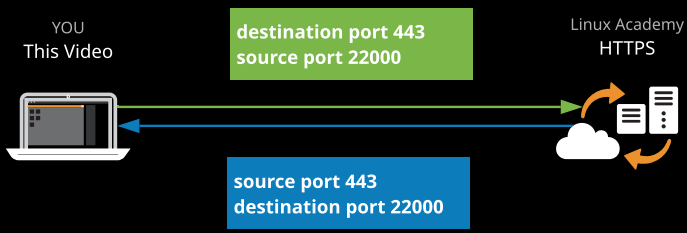
Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Network Access Control Lists (NACLs):

- NACLs operate at Layer 4 of the OSI model (TCP/UDP and below).
- A subnet has to be associated with a NACL — either the VPC default or a custom NACL.
- NACLs only impact traffic crossing the boundary of a subnet.
- NACLs are collections of rules that can explicitly **allow** or **deny** traffic based on its protocol, port range, and source/destination.
- Rules are processed in number order, lowest first. When a match is found, that action is taken and processing stops.
- The "*" rule is processed last and is an implicit **deny**.
- NACLs have two sets of rules: **inbound** and **outbound**.

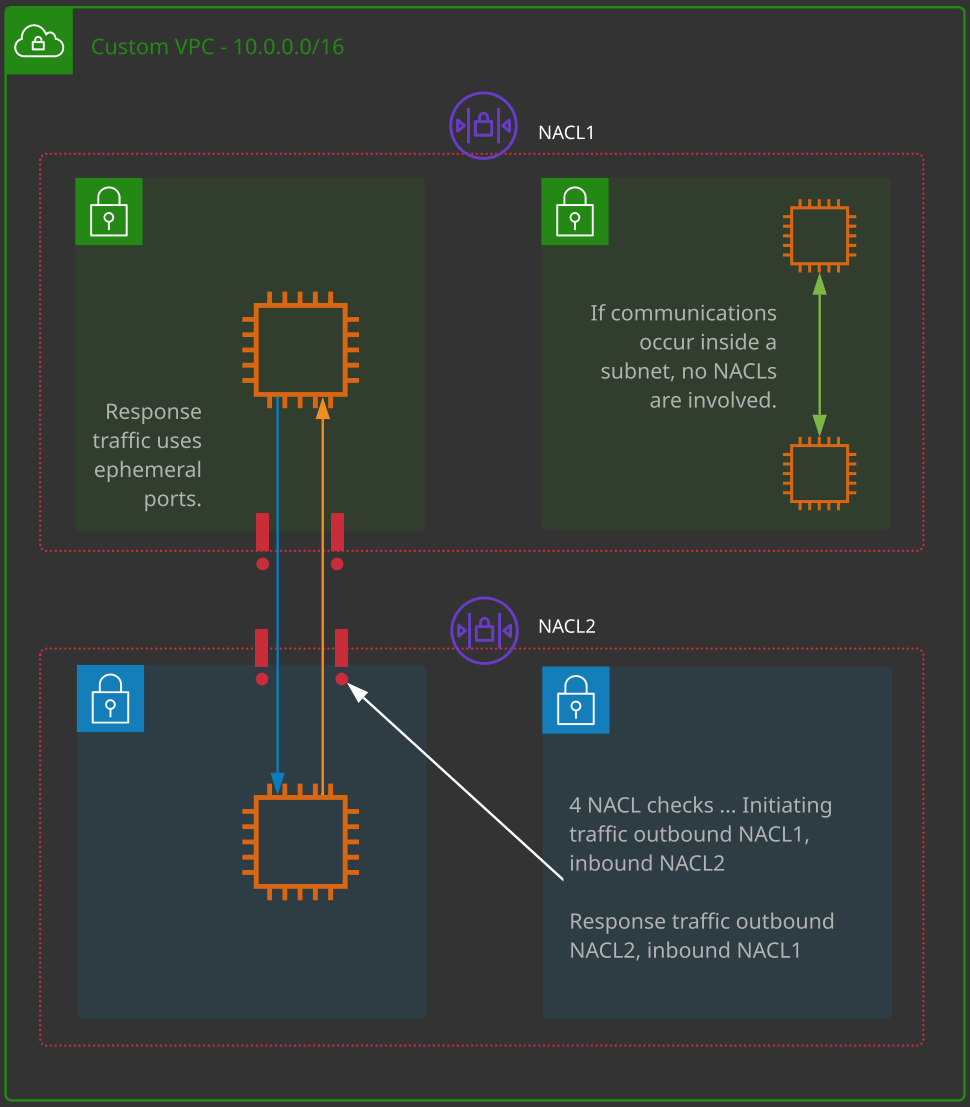
Ephemeral Ports:

- When a client initiates communications with a server, it is **to** a well-known port number (e.g., tcp/443) on that server.
- The response is from that well-known port to an ephemeral port on the client. The client decides the port.
- NACLs are stateless, they have to consider **both** initiating and response traffic — state is a session-layer concept.



Network Access Control List (NACL)

Exam Hints



Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

This topic will introduce advanced VPC concepts such as VPC peering, endpoints, IPv6 use within a VPC, egress-only gateways, and subnet/VPC sharing.

Previous Section

Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

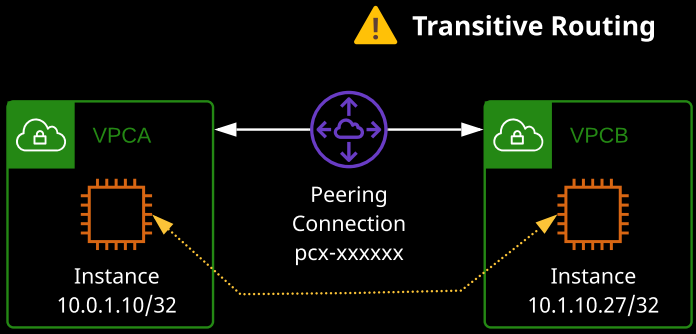
Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

VPC Peering:

- Allows direct communication between VPCs.
- Services can communicate using private IPs from VPC to VPC.
- VPC peers can span AWS accounts and even regions (with some limitations).
- Data is encrypted and transits via the AWS global backbone.
- VPC peers are used to link two VPCs at layer 3: company mergers, shared services, company and vendor, auditing.

Important Limits and Considerations:

- VPC CIDR blocks cannot overlap.
- VPC peers connect two VPCs — routing is not transitive.
- Routes are required at both sides (remote CIDR -> peer connection).
- NACLs and SGs can be used to control access.
- SGs can be referenced but **not** cross-region.
- IPv6 support is not available cross-region.
- DNS resolution to private IPs can be enabled, but it's a setting needed at both sides.



Back
Next

Go to Part 2
Back to Main

Course Navigation

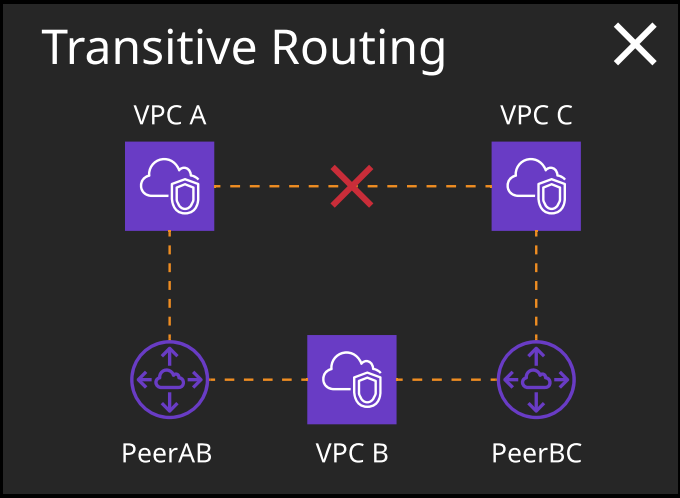
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53



Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

VPC endpoints are gateway objects created within a VPC. They can be used to connect to AWS public services without the need for the VPC to have an attached internet gateway and be public.

VPC Endpoint Types:

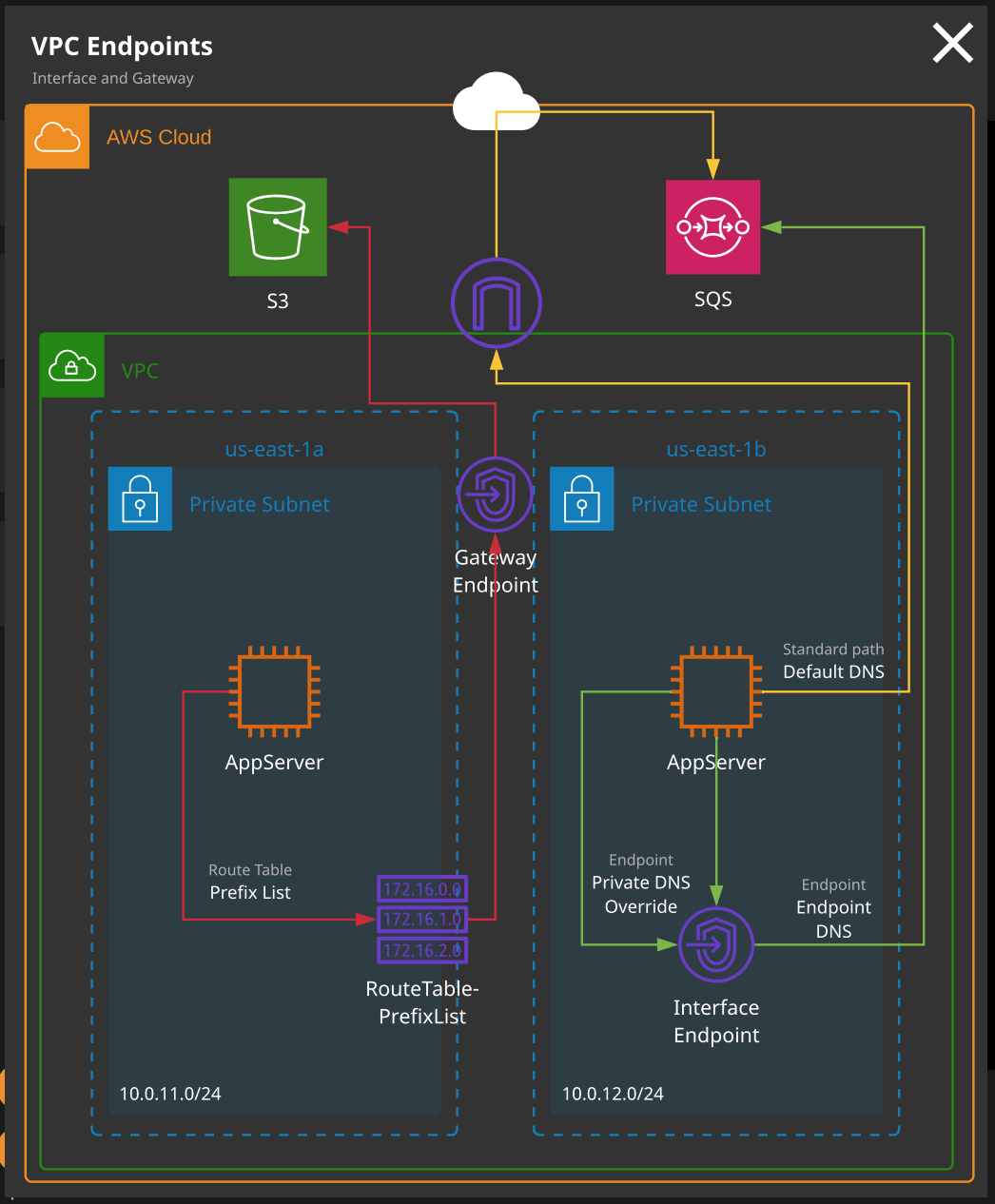
- Gateway endpoints: Can be used for DynamoDB and S3
- Interface endpoints: Can be used for everything else (e.g., SNS, SQS)

When to Use a VPC Endpoint:

- If the entire VPC is private with no IGW
- If a specific instance has no public IP/NATGW and needs to access public services
- To access resources restricted to specific VPCs or endpoints (private S3 bucket)

Limitations and Considerations:

- Gateway endpoints are used via route table entries — they are gateway devices. Prefix lists for a service are used in the destination field with the gateway as the target.
- Gateway endpoints can be restricted via policies.
- Gateway endpoints are HA across AZs in a region.
- Interface endpoints are interfaces in a specific subnet. For HA, you need to add multiple interfaces — one per AZ.
- Interface endpoints are controlled via SGs on that interface. NACLs also impact traffic.
- Interface endpoints add or replace the DNS for the service — no route table updates are required.
- Code changes to use the endpoint DNS, or enable private DNS to override the default service DNS.


VPC Endpoints
Interface and Gateway
AWS Cloud
S3
SQS
VPC
us-east-1a
Private Subnet
Gateway
Endpoint
us-east-1b
Private Subnet
Standard path
Default DNS
AppServer
AppServer
Route Table
Prefix List
172.16.0.0
172.16.1.0
172.16.2.0
RouteTable-
PrefixList
Endpoint
Private DNS
Override
Endpoint
Endpoint
DNS
Interface
Endpoint
10.0.11.0/24
10.0.12.0/24

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

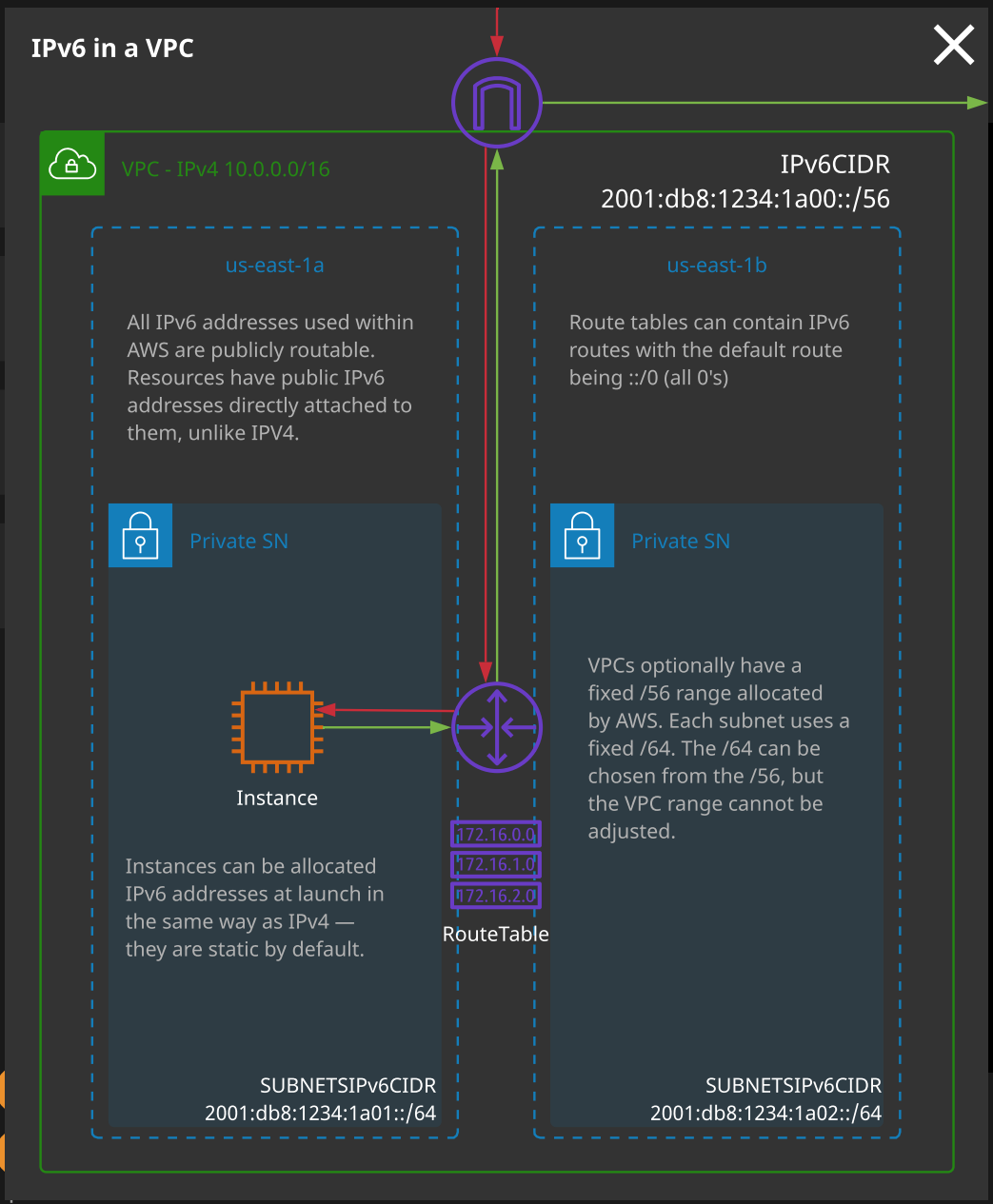
IP version 6 (IPv6) is supported within AWS but not across every product and not with every feature.

IPv6 VPC Setup:

- It is currently opt-in — it is disabled by default.
- To use it, the first step is to request an IPv6 allocation. Each VPC is allocated a /56 CIDR from the AWS pool — this cannot be adjusted.
- With the VPC IPv6 range allocated, subnets can be allocated a /64 CIDR from within the /56 range.
- Resources launched into a subnet with an IPv6 range can be allocated a IPv6 address via DHCP6.

Limitations and Considerations:

- DNS names are not allocated to IPv6 addresses.
- IPv6 addresses are all publicly routable — there is no concept of private vs. public with IPv6 (unlike IPv4 addresses).
- With IPv6, the OS is configured with this public address via DHCP6.
- Elastic IPs aren't relevant with IPv6.
- Not currently supported for VPNs, customer gateways, and VPC endpoints.


IPv6 in a VPC
VPC - IPv4 10.0.0.0/16
IPv6CIDR
2001:db8:1234:1a00::/56
us-east-1a
All IPv6 addresses used within AWS are publicly routable. Resources have public IPv6 addresses directly attached to them, unlike IPV4.
us-east-1b
Route tables can contain IPv6 routes with the default route being ::/0 (all 0's)
Private SN
Private SN
Instance
VPCs optionally have a fixed /56 range allocated by AWS. Each subnet uses a fixed /64. The /64 can be chosen from the /56, but the VPC range cannot be adjusted.
172.16.0.0
172.16.1.0
172.16.2.0
RouteTable
Instances can be allocated IPv6 addresses at launch in the same way as IPv4 — they are static by default.
SUBNETSIPv6CIDR
2001:db8:1234:1a01::/64
SUBNETSIPv6CIDR
2001:db8:1234:1a02::/64

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

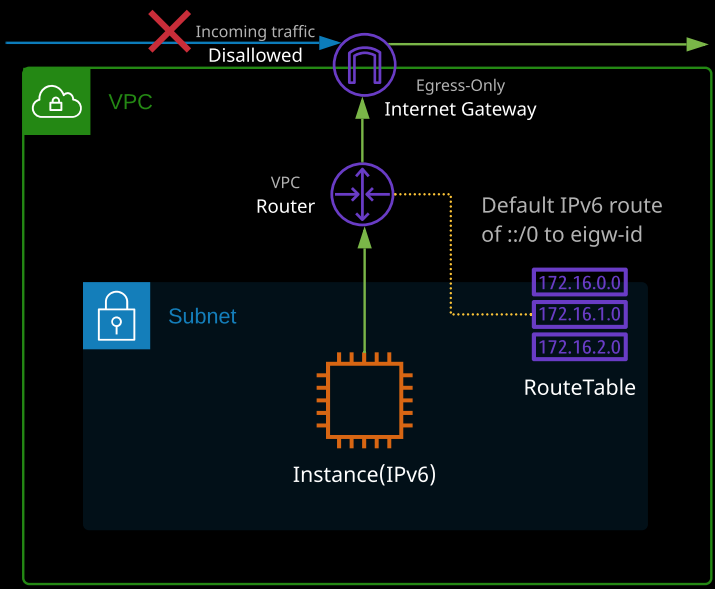
Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Egress-only internet gateways provide IPv6 instances with outgoing access to the public internet using IPv6 but prevent the instances from being accessed **from** the internet.

NAT isn't required with IPv6, and so NATGW's aren't compatible with IPv6. Egress-only gateways provide the outgoing-only access of a NATGW but do so without adjusting any IP addresses.

Architecturally, they are otherwise the same as an IGW.



Back
Next Topic

Go to Part 2
Back to Main

Course Navigation

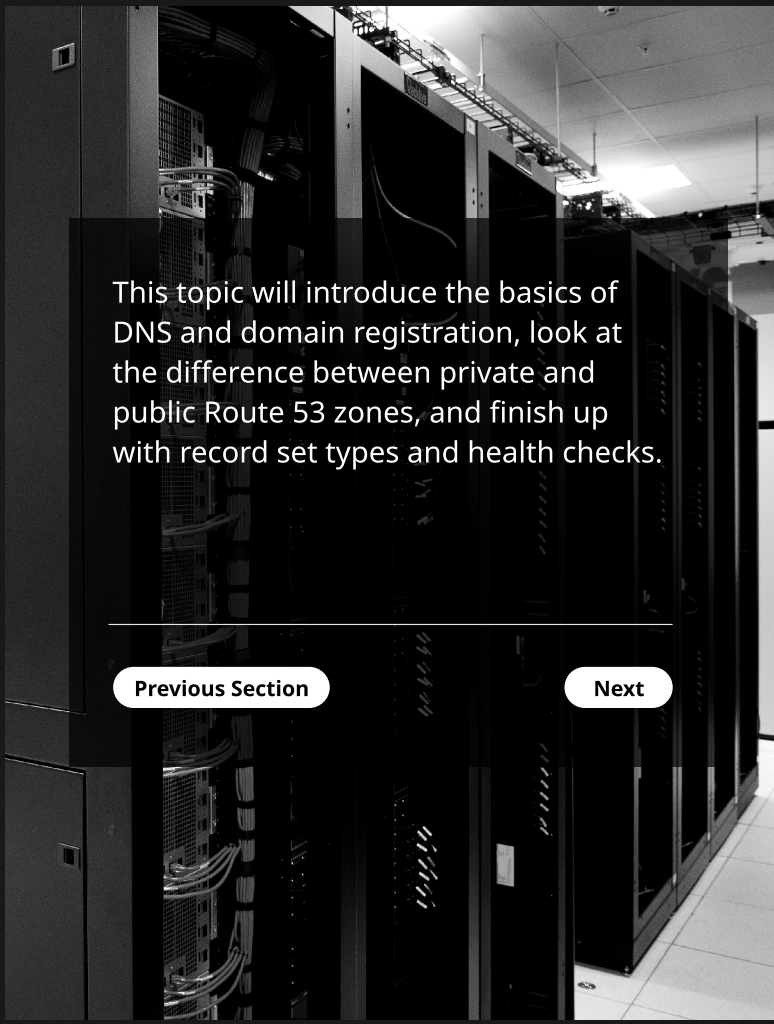
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53



This topic will introduce the basics of DNS and domain registration, look at the difference between private and public Route 53 zones, and finish up with record set types and health checks.

Previous Section

Next

Go to Part 2

Back to Main

Course Navigation

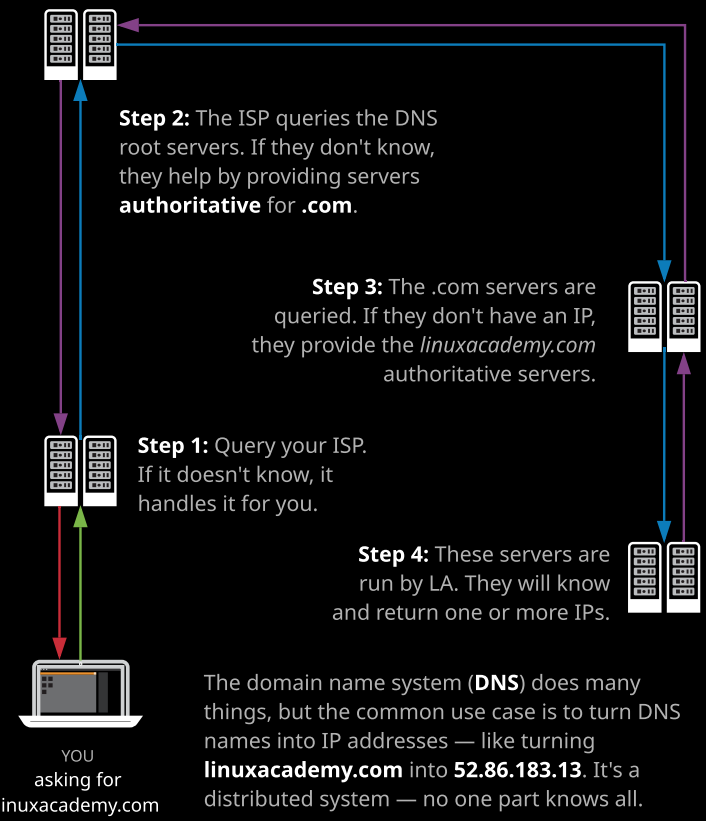
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53



DNS Terms

DNS Root Servers:
Trust starts somewhere. The DNS root servers are that trust — a group of servers that are authoritative to give answers about the root zone. TLDs are controlled by the root zone.

Top-Level Domain (TLD):
The top tier in the DNS hierarchy. Generally structured into geographic codes — such as .au, .us, .uk — and generic TLDs — such as .com, .org, and .edu. Large orgs or country orgs are delegated control of these by the root servers to be authoritative.

Subdomain:
Anything between a host and a TLD is a subdomain. *Linuxacademy.com* is a .com subdomain. .co.uk is a subdomain of .uk — an organization is delegated control of subdomains and is authoritative.

Zone and Zone File:
A zone or zone file is a mapping of IPs and hosts for a given subdomain. The zone file for linuxacademy.com would contain a record for www.

Records:
DNS has lots of record types — A, MX, AAAA, CNAME, TXT, NS (explained later) — and each does different things.

Name Server:
A name server is a server that runs a DNS service and can either store or cache information for the DNS platform. Whether a name server caches or acts as an authority depends on if it's referenced from a higher level. (See *authoritative* below.)

Authoritative:
The root servers are authoritative for the root zone — they are trusted by every operating system and networking stack globally. The root servers delegate ownership of a part of the hierarchy, such as .com, to an organization. That organization runs name servers that become authoritative — they can answer queries with authority. Because the root points at these servers, they are authoritative. These .com name servers can point at servers for sub domains (e.g., *linuxacademy.com*) that then become authoritative.

Hosts:
A record in a zone file: www, mail, catgifserver, doggowebserver.

FQDN: Fully qualified domain name — the host and domains: *www.linuxacademy.com*.

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Registering a domain within DNS consists of a few steps and components. With many services — such as Route 53, Hover.com, and even GoDaddy — these steps appear to happen together.

STEP 1: Check the domain is available:
This step is usually done during registration, but for a given domain, a check needs to occur against the database of the TLD or subdomain operator. Registering *bestcatpicsintheworldever.com* requires a check with Verisign who operate .com, and registering *amazingcats.co.uk* would need a check against .co.uk.

STEP 2: Purchase the domain via a registrar:
The domain operator allows companies to sell domains within the domain it manages. Buying a *something.com* domain via Route 53 means Route 53 taking payment and then contacting Verisign (.com operator) and adding a record into the .com zone that represents your domain.

STEP 3: Hosting the domain:
Registering a .com domain gives you the rights to specify name servers (NS) to be authoritative for that domain. You need to manage or pay for DNS hosting or name servers that are configured for your domain, and inform the .com operator to link those servers with your domain record. Route 53 allows you to register a domain **and** host it, or just **host** it, or just **register** it.

STEP 4: Records in the zone file:
On the Name Servers that are authoritative/host the domain, you need to add records into the zone file — www, mail, ftp, etc. This completes the chain, and these are accessible from the internet.

Back
Next

Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

A zone or hosted zone is a container for DNS records relating to a particular domain (e.g., *linuxacademy.com*). Route 53 supports public hosted zones, which influence the domain that is visible from the internet and VPCs. Private hosted zones are similar but accessible only from the VPCs they are associated with.

Public Zones:

- A public hosted zone is created when you register a domain with Route 53, when you transfer a domain into Route 53, or if you create one manually.
- A hosted zone (zone) has the same name as the domain it relates to — e.g., *linuxacademy.com* will have a hosted zone called *linuxacademy.com*.
- A public zone is accessible either from internet-based DNS clients (e.g., your laptop) or from within any AWS VPCs.
- A hosted zone will have "name servers" — these are the IP addresses you can give to a domain operator, so Route 53 becomes "authoritative" for a domain.

Private Zones:

- Private zones are created manually and associated with one or more VPCs — they are only accessible from those VPCs.
- Private zones need **`enableDnsHostnames`** and **`enableDnsSupport`** enabled on a VPC.
- Not all Route 53 features supported — limits on heath checks
- Split-view DNS is supported, using the same zone name for public and private zones — providing VPC resources with different records (e.g., testing, internal versions of websites).
  - With split view, private is preferred — if no matches, public is used.

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

DNS supports different types of records, each providing different functionality. At an associate level, the important ones are:

A Record (and AAAA): For a given host (www), an A record provides an IPv4 address (e.g., 10.0.0.1) and an AAAA provides an IPv6 address.

CNAME Record: Allows aliases to be created (not the same as an alias record). A machine such as *allthethings.linuxacademy.com* might have CNAMES for www, ftp, and images. Each of these CNAMEs points at an existing record in the domain. www -> *allthethings.linuxacademy.com*. CNAMEs cannot be used at the APEX of a domain (e.g., *linuxacademy.com*).

MX Record: MX records provide the mail servers for a given domain. Each MX record has a priority. Remote mail servers use this to locate the server to use when sending to *someuser@linuxacademy.com*.

NS Record: Used to set the authoritative servers for a subdomain. .com would have NS servers for *linuxacademy.com*.

TXT Record: Used for descriptive text in a domain — often used to verify domain ownership (Gmail/Office365).

Alias Records: An extension of CNAME — can be used like an A record, with the functionality of a CNAME and none of the limitations. Can refer to AWS logical services (load balancers, S3 buckets), and AWS doesn't charge for queries of alias records against AWS resources.

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Health checks can be created within Route 53 and are used to influence Route 53 routing decisions. There are three type of health checks:

- Health checks that monitor the health of an endpoint — e.g., IP address or hostname
- Health checks that monitor the health of another health check (these are referred to as calculated health checks)
- Health checks that monitor CloudWatch alarms — you might want to consider something unhealthy if your DynamoDB table is experiencing performance issues

Route 53 Health Checkers:

- Global health check system that checks an endpoint in an agreed way with an agreed frequency.
- >18% of checks report healthy = healthy, <18% healthy = unhealthy

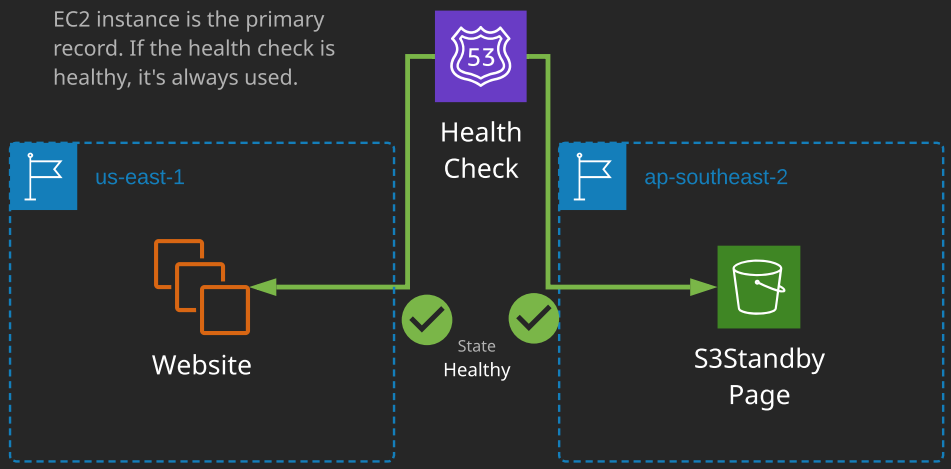
Types of Health Check:

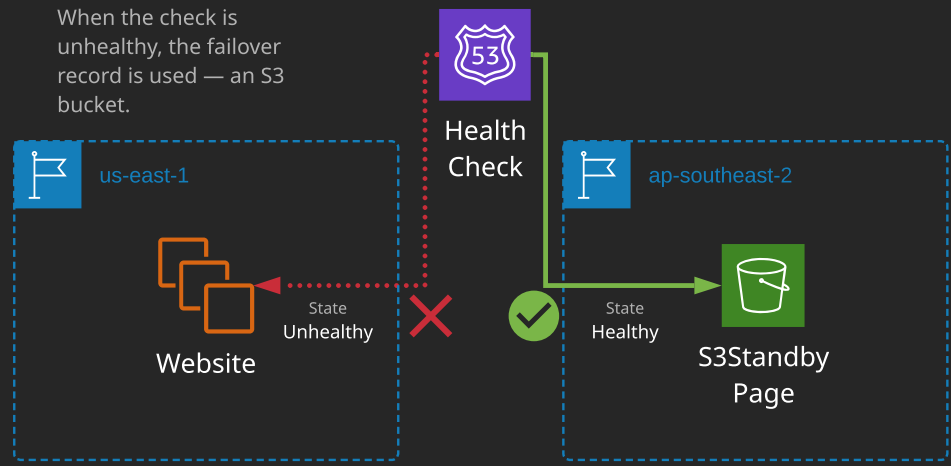
- HTTP and HTTPS: tcp/80 or tcp/443 connection check in less than four seconds. Reporting 2XX or 3XX code within two seconds.
- TCP health check: tcp connection within 10 seconds
- HTTP/S with string match: All the checks as with HTTS/HTTPS but the body is checked for a string match

Route 53 and Health Checks:

- Records can be linked to health checks. If the check is unhealthy, the record isn't used.
- Can be used to do failover and other routing architectures (more in the next topic)

Route 53 Health Checks and Failover





Course Navigation

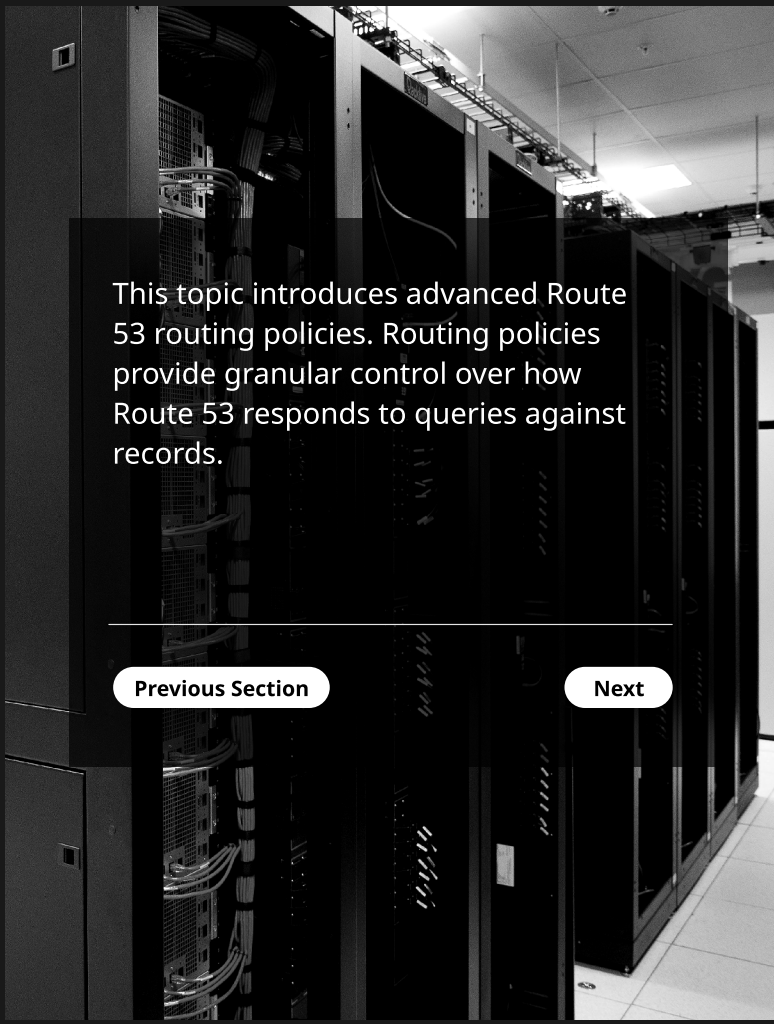
AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53



This topic introduces advanced Route 53 routing policies. Routing policies provide granular control over how Route 53 responds to queries against records.

Previous Section

Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

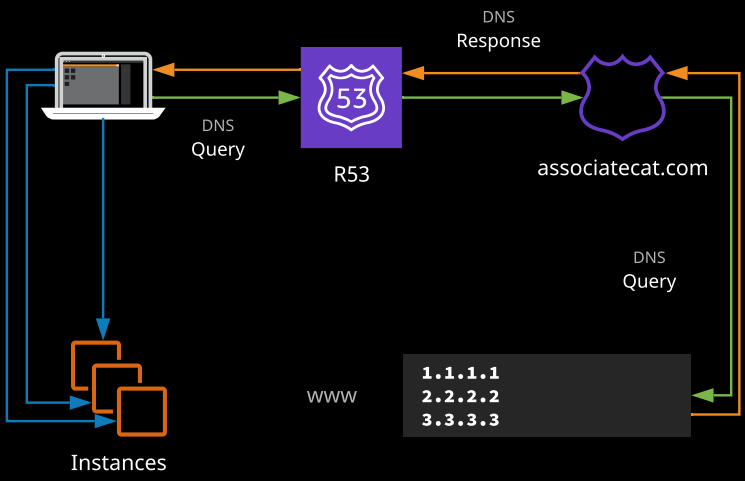
Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

A simple routing policy is a single record within a hosted zone that contains one or more values. When queried, a simple routing policy record returns all the values in a randomized order.



The DNS client (the laptop) receives a randomized list of IPs as a result. The client can select the appropriate one and, in this example, initiate an HTTP session with a resource.

Pros: Simple, the default, even spread of requests
Cons: No performance control, no granular health checks, for alias type — only a single AWS resource

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

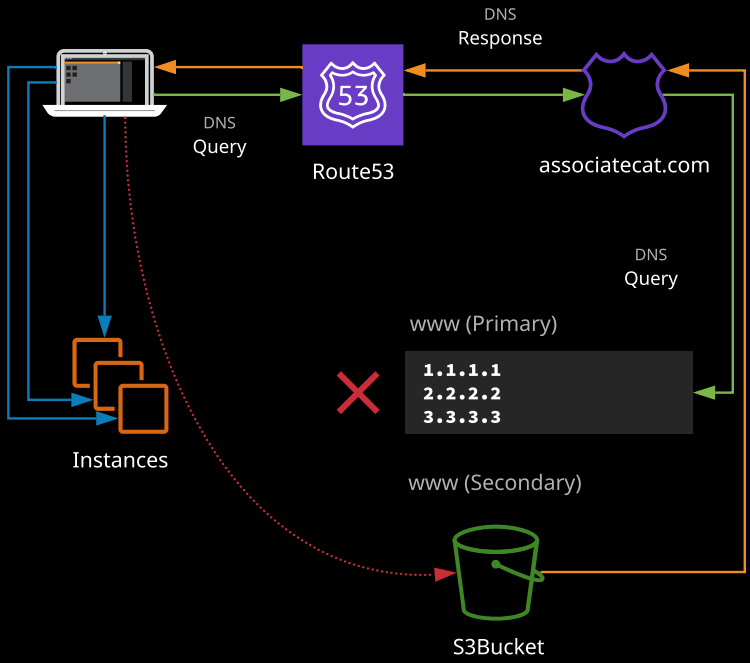
Networking
Section 4

- Network Fundamentals
- AWS Private Networking
- Advanced VPC
- AWS DNS Fundamentals
- Advanced Route 53

Networking

Advanced Route 53

Failover routing allows you to create two records with the same name. One is designated as the primary and another as secondary. Queries will resolve to the primary — unless it is unhealthy, in which case Route 53 will respond with the secondary.



Failover can be combined with other types to allow multiple primary and secondary records. Generally, failover is used to provide emergency resources during failures.

Back | Next

Go to Part 2

Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

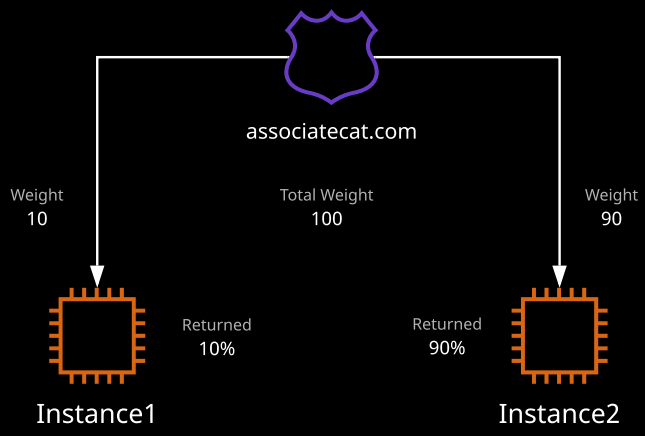
Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Weighted routing can be used to control the amount of traffic that reaches specific resources. It can be useful when testing new software or when resources are being added or removed from a configuration that doesn't use a load balancer.



Records are returned based on a ratio of their weight to the total weight, assuming records are healthy.

AWS and SA Fundamentals
Section 1

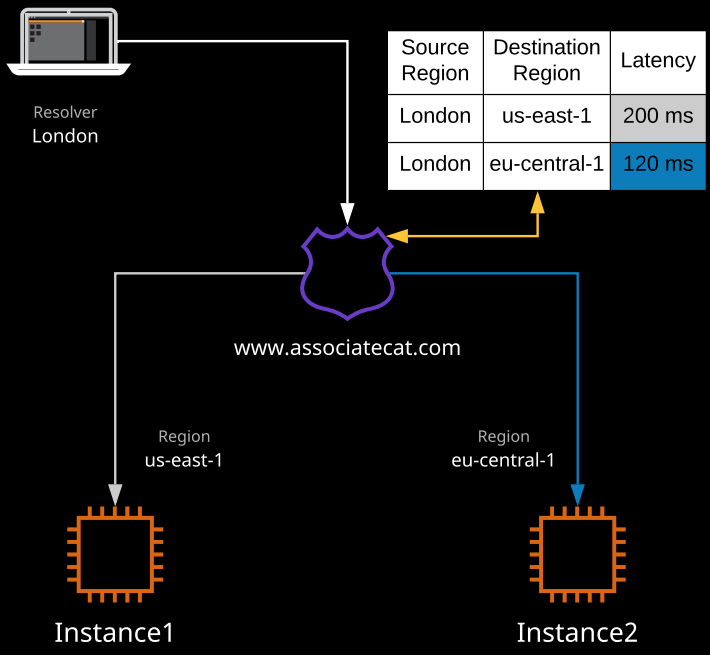
Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

With latency-based routing, Route 53 consults a latency database each time a request occurs to a given latency-based host in DNS from a resolver server. Record sets with the same name are considered part of the same latency-based set. Each is allocated to a region. The record set returned is the one with the lowest latency to the resolver server.



AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

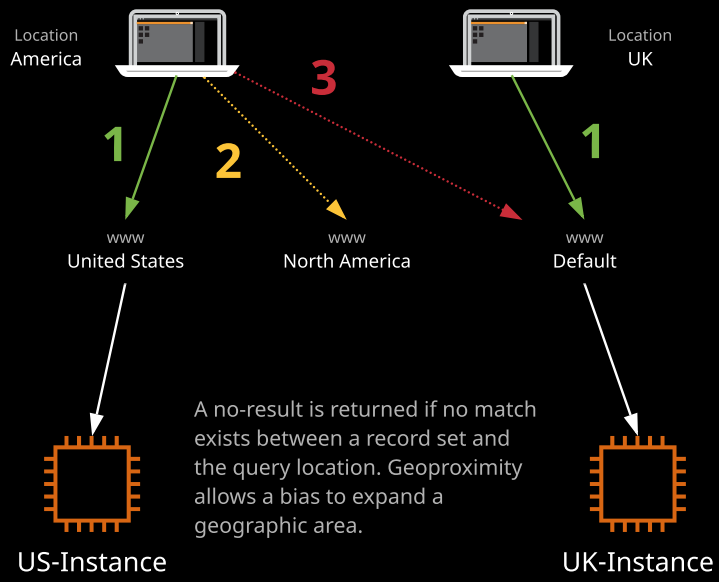
Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Networking

Advanced Route 53

Geolocation routing lets you choose the resources that serve your traffic based on the geographic region from which queries originate. A record set is configured for a continent or country. That record set is used for queries in that same region, with more specific matches taking priority.

A record set can be set as the default that gets returned if the IP matching process fails or if no record set is configured for the originating query region.



Back
Next Section

Go to Part 2
Back to Main

Course Navigation

AWS and SA Fundamentals
Section 1

Identity and Access Control
Section 2

Compute
Section 3

Networking
Section 4

Network Fundamentals
AWS Private Networking
Advanced VPC
AWS DNS Fundamentals
Advanced Route 53

Storage and Content Delivery

S3 Architecture

Course Navigation

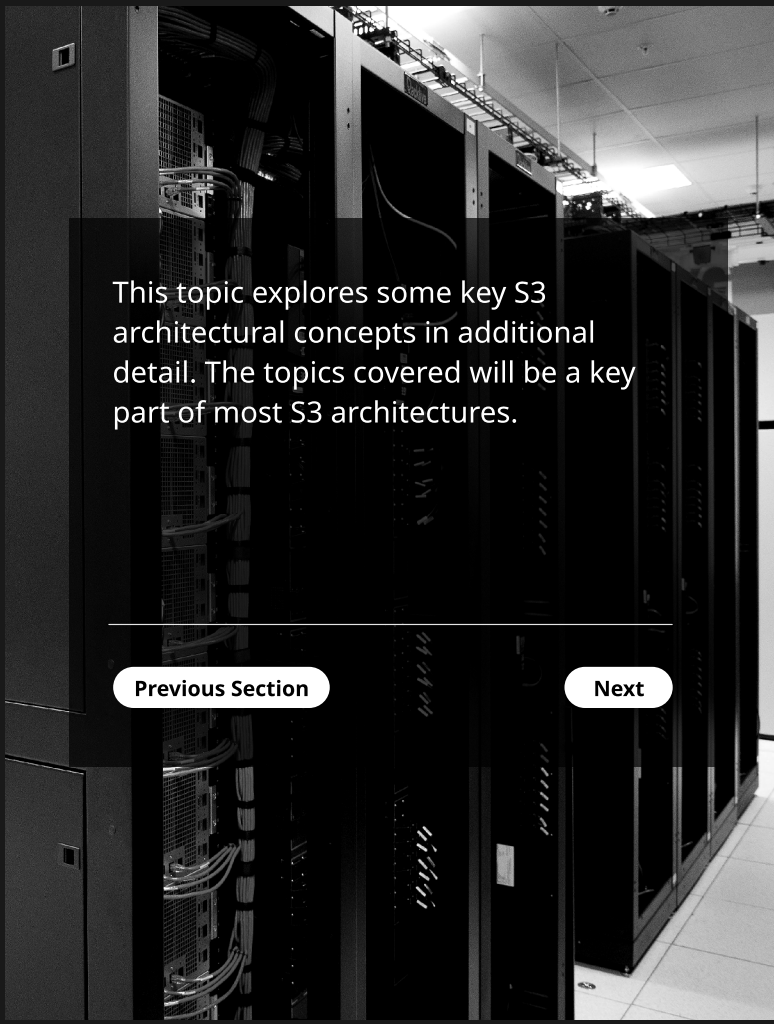
Storage and Content Delivery
Section 5

- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8



This topic explores some key S3 architectural concepts in additional detail. The topics covered will be a key part of most S3 architectures.

Previous Section

Next

Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

Databases
Section 6

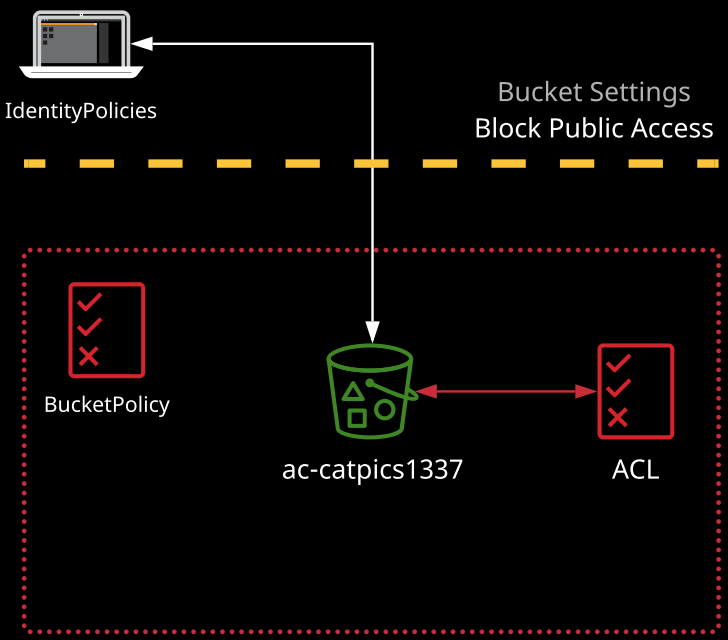
Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Storage and Content Delivery

S3 Architecture

Bucket authorization within S3 is controlled using **identity policies** on AWS identities, as well as **bucket policies** in the form of resource policies on the bucket and bucket or object **ACLs**.



Final authorization is a combination of all applicable policies. Priority order is (1) Explicit Deny, (2) Explicit Allow, (3) Implicit Deny.

Back
Next

Go to Part 1
Back to Main

Course Navigation

Storage and Content Delivery
Section 5

- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Block public access is a setting applied on top of any existing settings as a protection.

It can disallow **all** public access granted to a bucket and objects using ACLs or bucket policies.

It can also block new public access grants using bucket policies or ACLs.

IMPORTANT: Block public access overrules any other public grant.

Go to Part 1

Back to Main

Course Navigation

Storage and Content Delivery
Section 5

S3 Architecture
S3 Performance and HA
CloudFront
Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

```
{
  "Version" : "2012-10-17",
  "Statement" : [
   {
    "Sid" : "FirstStatement",
    "Effect" : "Allow",
    "Action" : ["iam:ChangePassword"],
    "Resource" : "*"
   },
   {
    "Sid" : "SecondStatement",
    "Effect" : "Allow",
    "Action" : "s3:ListAllMyBuckets",
    "Resource" : "*"
   },
   {
    "Sid" : "ThirdStatement",
    "Effect" : "Allow",
    "Action" : [ "s3:List*", "s3:Get*" ],
    "Resource" : [
     "arn:aws:s3:::ac-catpics1337",
     "arn:aws:s3:::ac-catpics1337/*"
    ]
   }
  ]
}
```

Identity policies attached to IAM users, roles, or groups can include S3 elements. This only works for identities in the same account as the bucket.

Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Storage and Content Delivery

S3 Architecture

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Sid" : "AddPerm",
      "Effect" : "Allow",
      "Principal" : "*",
      "Action" : ["s3:GetObject"],
      "Resource" : [
        "arn:aws:s3:::ac-catpics1337/*"
      ]
    }
  ]
}
```

Resource policies apply to a resource. They can be used to authorize access to a bucket or objects inside a bucket to large numbers of identities. Bucket policies can also apply to anonymous accesses (public access).

Go to Part 1

Back to Main

Storage and Content Delivery

S3 Architecture

Uploads to S3 are generally done using the S3 console, the CLI, or directly using the APIs. Uploads either use a single operation (known as a single PUT upload) or multipart upload.

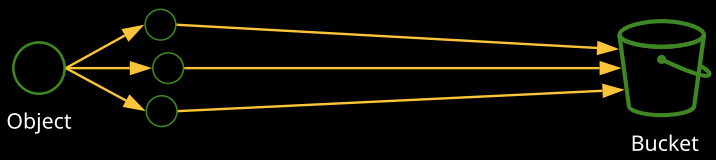
Single PUT Upload

Object is uploaded in a single stream of data

Limit of **5 GB**, can cause performance issues, and if the upload fails the **whole upload** fails

Multipart Upload

An object is broken up into parts (up to **10,000**), each part is **5 MB** to **5 GB**, and the last part can be less (the remaining data)



Multipart upload is **faster** (parallel uploads), and the individual parts can fail and be retried individually. AWS recommends multipart for anything over **100 MB**, but it's required for anything beyond **5 GB**.

Storage and Content Delivery
Section 5

- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Storage and Content Delivery

S3 Architecture

Amazon S3 buckets can be configured to host websites. Content can be uploaded to the bucket and when enabled, **static web hosting** will provide a unique endpoint URL that can be accessed by any web browser. S3 buckets can host many types of content, including:

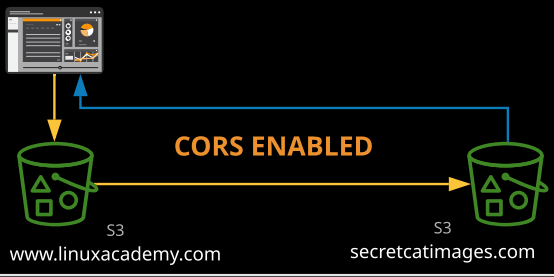
- HTML, CSS, JavaScript
- Media (audio, movies, images)

S3 can be used to host front-end code for serverless applications or an offload location for static content. CloudFront can also be added to improve the speed and efficiency of content delivery for global users or to add SSL for custom domains.

Route 53 and alias records can also be used to add human-friendly names to buckets.

Cross-Origin Resource Sharing (CORS)

CORS is a security measure allowing a web application running in one domain to reference resources in another.



Storage and Content Delivery
Section 5

- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Storage and Content Delivery

S3 Architecture

Data between a client and S3 is encrypted **in transit**. Encryption **at rest** can be configured on a **per-object** basis.

- **Client-Side Encryption**: The client/application is responsible for managing both the encryption/decryption process and its keys. This method is generally only used when strict security compliance is required — it has significant admin and processing overhead.
- **Server-Side Encryption with Customer-Managed Keys** (**SSE-C**): S3 handles the encryption and decryption process. The customer is still responsible for key management, and keys must be supplied with each PUT or GET request.
- **Server-Side Encryption with S3-Managed Keys** (**SSE-S3**): Objects are encrypted using **AES-256** by S3. The keys are generated by S3 (using KMS on your behalf). Keys are stored with objects in an encrypted form. If you have permissions on the object (e.g., S3 Read or S3 Admin), you can decrypt and access it.
- **Server-Side Encryption with AWS KMS-Managed Keys** (**SSE-KMS**): Objects are encrypted using individual keys generated by KMS. Encrypted keys are stored with the encrypted objects. Decryption of an object needs both S3 and KMS key permissions (role separation).

Bucket Default Encryption

Objects are encrypted in S3, not buckets. Each PUT operation needs to specify encryption (and type) or not. A bucket default captures any PUT operations where no encryption method/directive is specified. It doesn't enforce what type can and can't be used. Bucket policies can enforce.

Storage and Content Delivery

S3 Architecture

Storage and Content Delivery
Section 5

- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

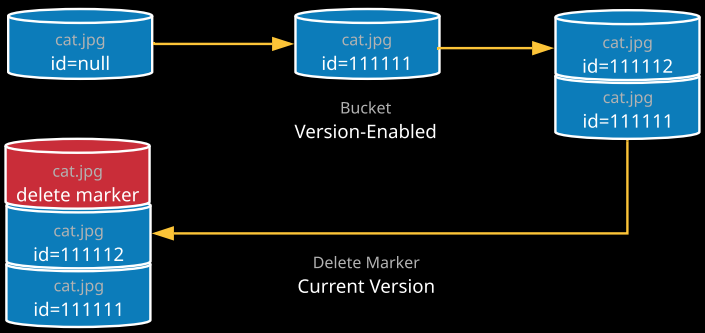
Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Versioning can be enabled on an S3 bucket. Once enabled, any operations that would otherwise modify objects generate new versions of that original object. Once a bucket is version-enabled, it can never be fully switched off — only **suspended**.

With versioning enabled, an AWS account is billed for all versions of all objects. Object deletions by default don't delete an object — instead, a delete marker is added to indicate the object is deleted (this can be undone). Older versions of an object can be accessed using the object name and a version ID. Specific versions can be deleted.



MFA Delete is a feature designed to prevent accidental deletion of objects. Once enabled, a one-time password is required to delete an object version or when changing the versioning state of a bucket.

Storage and Content Delivery
Section 5

- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Storage and Content Delivery

S3 Architecture

A **presigned URL** can be created by an identity in AWS, providing access to an object using the creator's access permissions. When the presigned URL is used, AWS verifies the **creator's access** to the object — not yours. The URL is encoded with authentication built in and has an expiry time.

Presigned URLs can be used to **download** or **upload** objects.

Any identity can create a presigned URL — even if that identity doesn't have access to the object.

Example presigned URL scenarios:

- Stock images website — media stored privately on S3, presigned URL generated when an image is purchased
- Client access to upload an image for process to an S3 bucket

When using presigned URLs, you may get an error. Some common situations include:

- The presigned URL has expired — seven-day maximum
- The permissions of the creator of the URL have changed
- The URL was created using a role (36-hour max) and the role's temporary credentials have expired (aim to never create presigned URLs using roles)

Storage and Content Delivery

S3 Performance and HA

Course Navigation

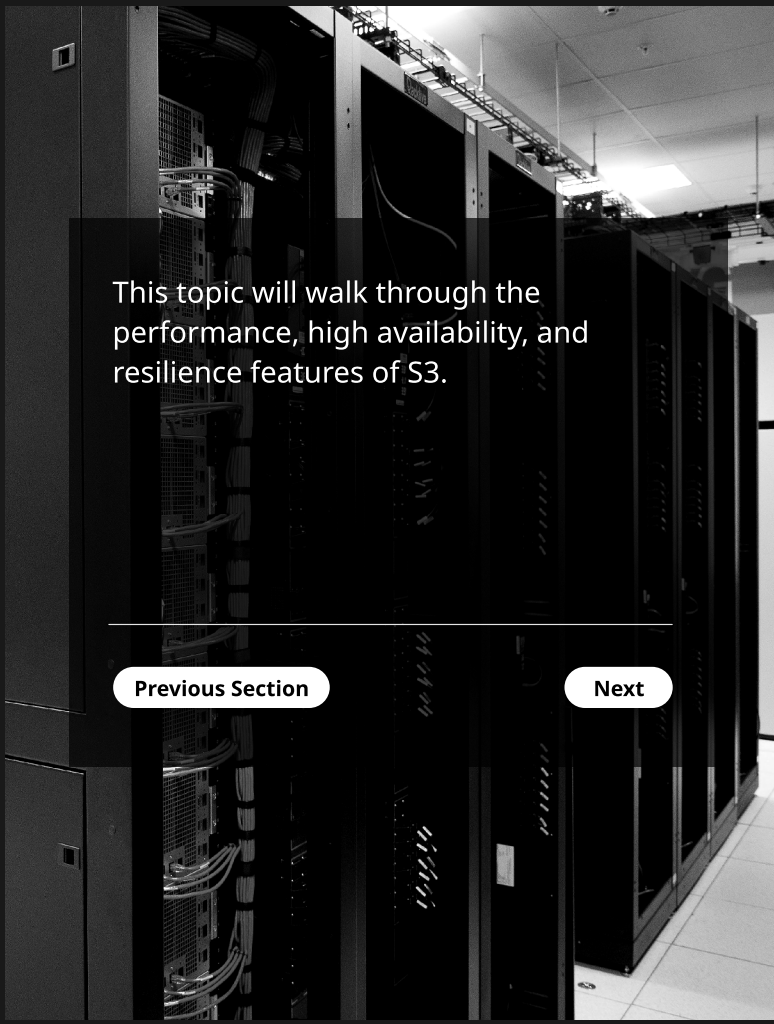
Storage and Content Delivery
Section 5

- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8



This topic will walk through the performance, high availability, and resilience features of S3.

Previous Section

Next

Go to Part 1

Back to Main

Storage and Content Delivery

S3 Performance and HA

All objects within an S3 bucket use a **storage class**, also known as a **storage tier**. Storage classes influence the cost, durability, availability, and "first byte latency" for objects in S3. The class used for an object can be changed manually or using lifecycle policies.

Standard

- Default, all-purpose storage or when usage is unknown
- 99.999999999% (11 nines) durability and four nines availability
- Replicated in + AZs — no minimum object size or retrieval fee

Standard Infrequent Access (Standard-IA)

- Objects where real-time access is required but infrequent
- 99.9% availability, 3+ AZ replication, cheaper than Standard
- 30-day and 128 KB minimum charges and object retrieval fee

One Zone-IA

- Non-critical and/or reproducible objects
- 99.5% availability, only 1 AZ, 30-day and 128 KB minimum charges
- Cheaper than Standard and Standard-IA

Glacier

- Long-term archival storage (warm or cold backups)
- Retrievals could take minutes or hours (faster = higher cost)
- 3+ AZ replication, 90-day and 40 KB minimum charge and retrieval

Glacier Deep Archive

- Long-term archival (cold backups) — 180-day and 40 KB minimums
- Longer retrievals but cheaper than Glacier — replacement for tape-style storage

Storage and Content Delivery

S3 Performance and HA

Storage and Content Delivery
Section 5

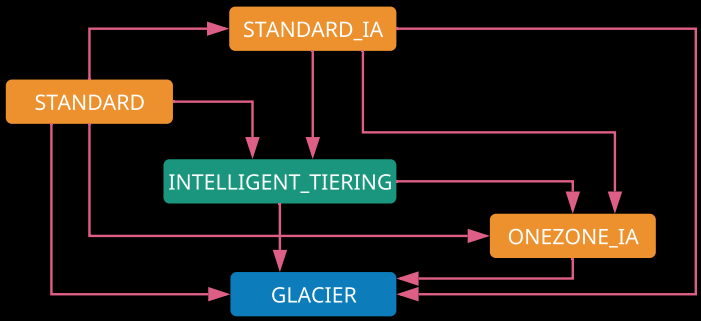
- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Storage classes can be controlled via **lifecycle rules**, which allow for the automated transition of objects between storage classes, or in certain cases allow for the expiration of objects that are no longer required. Rules are added at a bucket level and can be enabled or disabled based on business requirements.



Objects smaller than 128 KB cannot be transitioned into **INTELLIGENT_TIERING**. Objects must be in the original storage class for a minimum of 30 days before transitioning them to either of the IA storage tiers. Instead of transitioning between tiers, objects can be configured to expire after certain time periods. At the point of expiry, they are deleted from the bucket.

Objects can be archived into Glacier using lifecycle configurations. The objects remain inside S3, managed from S3, but Glacier is used for storage. Objects can be restored into S3 for temporary periods of time — after which, they are deleted. If objects are encrypted, they remain encrypted during their transition to Glacier or temporary restoration into S3.

Course Navigation

Storage and Content Delivery
Section 5

S3 Architecture
S3 Performance and HA
CloudFront
Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Intelligent-Tiering

Intelligent-Tiering is a special type of storage class designed for unknown or unpredictable access patterns. It moves objects automatically between two tiers — one designed for frequent access, the other for infrequent.

Objects that aren't accessed for 30 days are moved to the infrequent tier, which offers lower costs. If an object in this tier is accessed, it's moved to the frequent access tier at no cost. Intelligent-Tiering adds a monthly automation and monitoring fee — but does away with any retrieval costs.

Go to Part 1
Back to Main

Storage and Content Delivery

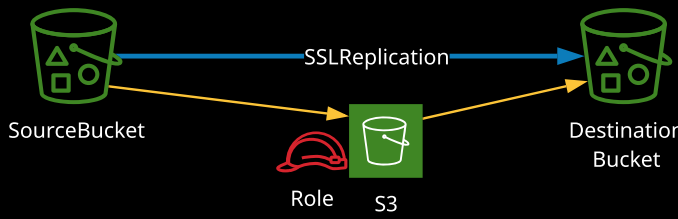
S3 Performance and HA

S3 cross-region replication (S3 CRR) is a feature that can be enabled on S3 buckets allowing one-way replication of data from a source bucket to a destination bucket in another region.

By default, replicated objects keep their:

- Storage class
- Object name (key)
- Owner
- Object permissions

Replication configuration is applied to the source bucket, and to do so requires versioning to be enabled on both buckets. Replication requires an IAM role with permissions to replicate objects. With the replication configuration, it is possible to override the storage class and object permissions as they are written to the destination.



Excluded from Replication

- System actions (lifecycle events)
- Any existing objects from before replication is enabled
- SSE-C encrypted objects — only SSE-S3 and (if enabled) KMS encrypted objects are supported

Storage and Content Delivery

CloudFront

Course Navigation

Storage and Content Delivery
Section 5

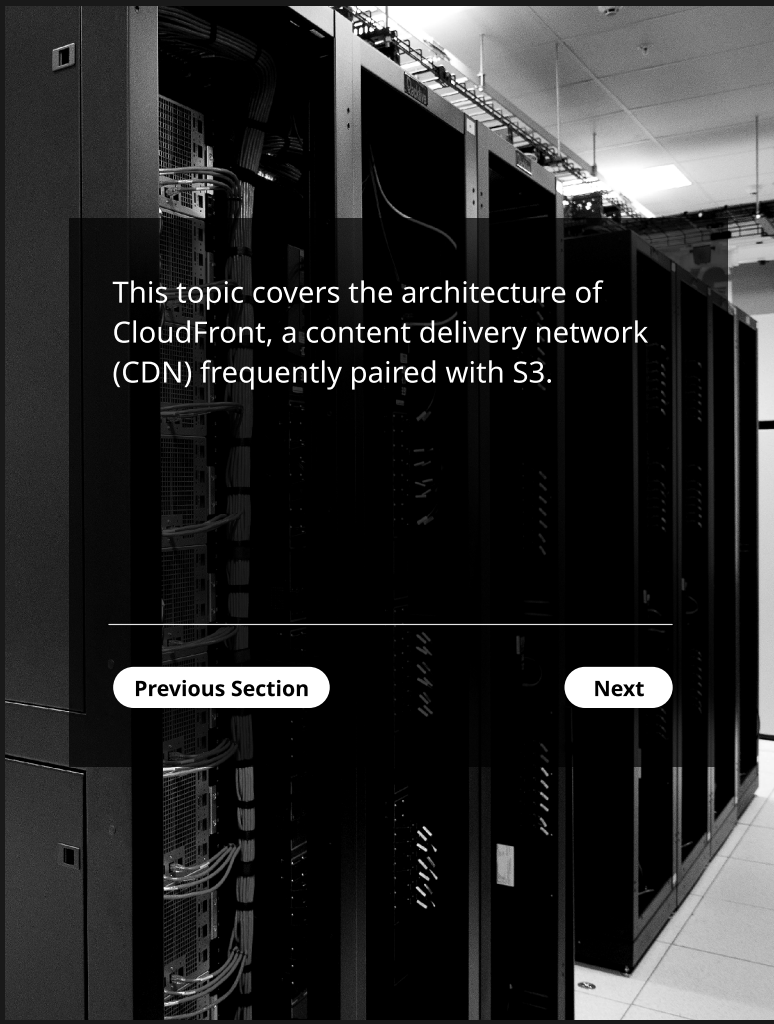
- S3 Architecture
- S3 Performance and HA
- CloudFront

Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8



This topic covers the architecture of CloudFront, a content delivery network (CDN) frequently paired with S3.

Previous Section

Next

Go to Part 1

Back to Main

Storage and Content Delivery

CloudFront

Storage and Content Delivery
Section 5

- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

CloudFront is a content delivery network (CDN). A CDN is a global cache that stores copies of your data on edge caches, which are positioned as close to your customers as possible. It has three main benefits: lower latency, higher transfer speeds, and reduced load on the content server.

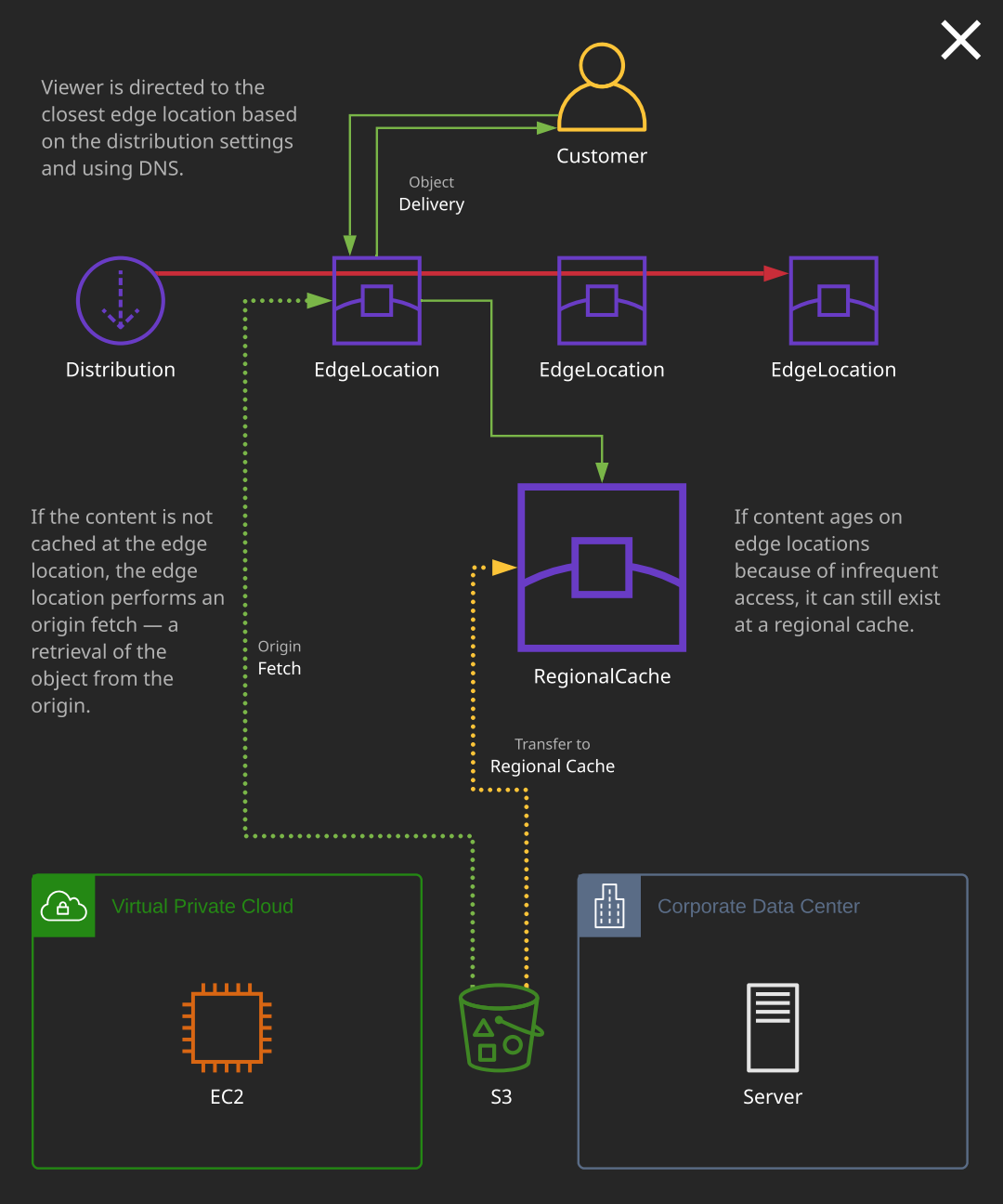
CloudFront Components

- **Origin:** The server or service that hosts your content. Can be an S3 bucket, web server, or Amazon MediaStore.
- **Distribution:** The "configuration" entity within CloudFront. It's where you configure all aspects of a specific "implementation" of CloudFront from.
- **Edge Location:** The local infrastructure that hosts caches of your data. Positioned in over 150 locations globally in over 30 countries.
- **Regional Edge Caches:** Larger versions of edge locations. Less of them but have more capacity and can serve larger areas.

Caching Process

- Create a distribution and point at one or more origins. A distribution has a DNS address that is used to access it.
- The DNS address directs clients at the closest available edge location.
- If the edge location has a cached copy of your data, it's delivered locally from that edge location.
- If it's not cached, the edge location attempts to download it from either a regional cache or from the origin (known as an origin fetch).
- As the edge location receives the data, it immediately begins forwarding it and caches it for the next visitor.

Content can expire, be discarded, and be recached. Or you can explicitly invalidate content to remove it from caches.


Viewer is directed to the closest edge location based on the distribution settings and using DNS.
Customer
Object
Delivery
Distribution
EdgeLocation
EdgeLocation
EdgeLocation
If the content is not cached at the edge location, the edge location performs an origin fetch — a retrieval of the object from the origin.
Origin
Fetch
RegionalCache
If content ages on edge locations because of infrequent access, it can still exist at a regional cache.
Transfer to
Regional Cache
Virtual Private Cloud
EC2
S3
Corporate Data Center
Server

Storage and Content Delivery

CloudFront

Storage and Content Delivery
Section 5

- S3 Architecture
- S3 Performance and HA
- CloudFront

Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

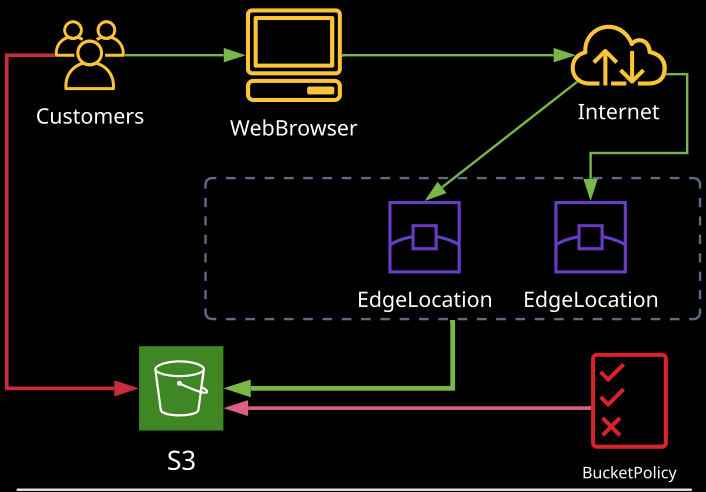
Application, Analytics, and Operations
Section 8

By default, CloudFront is fully **publicly accessible** — anyone with the DNS endpoint address can access content cached by the distribution.

A distribution can be configured to be **private** where each access requires a signed URL or cookie. This is done by setting the **trusted signers** on the distribution.

Private distributions can be bypassed by going straight to the origin (e.g., an S3 bucket).

An origin access identity (**OAI**) is a virtual identity that can be associated with a distribution. An S3 bucket can then be restricted to only allow this OAI to access it — all other identities can be denied.



Back
Next Topic

Go to Part 1
Back to Main

Storage and Content Delivery

CloudFront

Course Navigation

Storage and Content Delivery
Section 5

- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8


```
{
    "Version": "2008-10-17",
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront
Origin Access Identity XXXXXXXXXXXX"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::mybucket/*"
        }
    ]
}
```


Go to Part 1

Back to Main

Storage and Content Delivery

Network File Systems

Course Navigation

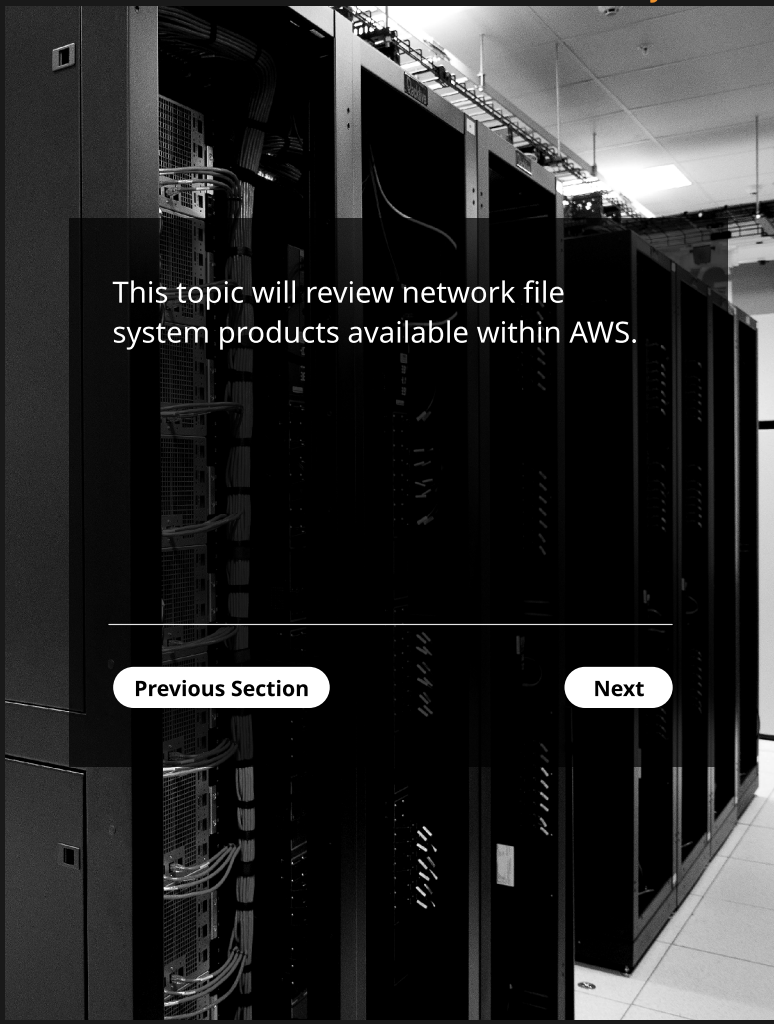
Storage and Content Delivery
Section 5

- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8



This topic will review network file system products available within AWS.

Previous Section

Next

Go to Part 1

Back to Main

Linux Academy

Storage and Content Delivery

Network File Systems

Course Navigation

Storage and Content Delivery
Section 5

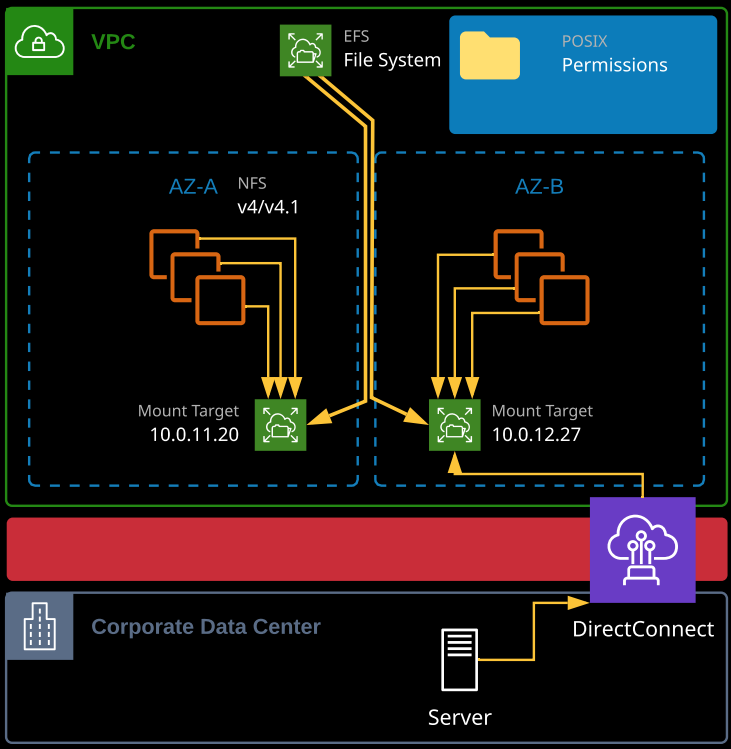
- S3 Architecture
- S3 Performance and HA
- CloudFront
- Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Amazon EFS is an implementation of the Network File System (NFSv4) delivered as a service. File systems can be created and mounted on multiple Linux instances **at the same time**.



Back

Next Section

Go to Part 1

Back to Main

Course Navigation

Storage and Content Delivery
Section 5

S3 Architecture
S3 Performance and HA
CloudFront
Network File Systems

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Exam Facts and Figures: Elastic File System

EFS is an implementation of the NFSv4 protocol within AWS. Use EFS when you need a "file system" that can be accessed from multiple instances (e.g., shared media, home folders, documentation, shared logs).

- Its base entity is a file system.
- The file system is accessed via "mount targets" that are placed in subnets inside a VPC and have an IP address.
- The file system is "mounted" on Linux instances. (**Important**: EFS is currently only supported in Linux.)
- File systems are accessible from a VPC or from on-premises locations via a VPN or Direct Connect.

EFS has two performance modes: **General Purpose** (the default and suitable for 99% of needs) and **Max I/O** (which is designed for when a large number of instances [as in, hundreds] need to access the file system).

EFS has two throughput modes: **Bursting Throughput** and **Provisioned Throughput**.

- Bursting = 100 MiB/s base burst. 100 MiB/s per 1 TiB size. Earning 50 MiB/s per TiB of storage.
- Throughput mode allows control over throughput independently of file system size.

Security groups are used to control access to NFS mount targets.

EFS supports two storage classes: **Standard** and **Infrequent Access (IA)**. Lifecycle management is used to move files between classes based on access patterns.

Go to Part 1

Back to Main

Course Navigation

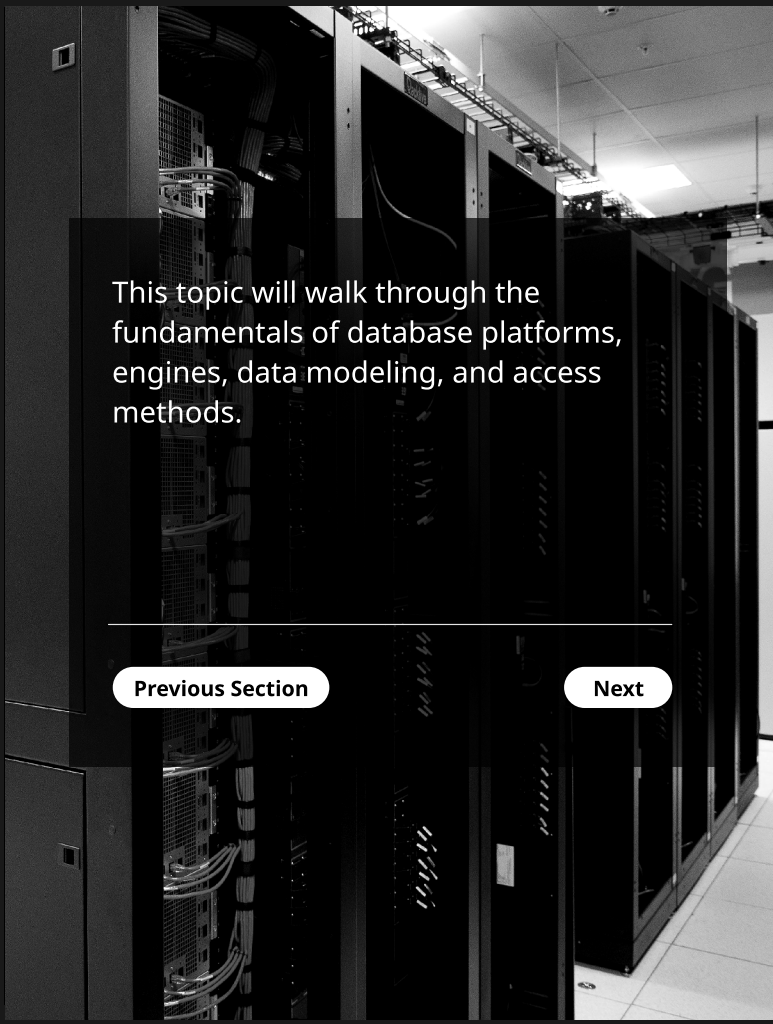
Storage and Content Delivery
Section 5

Databases
Section 6

DB Fundamentals
SQL: RDS
SQL: Aurora
NoSQL Databases
In-Memory Caching

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8



This topic will walk through the fundamentals of database platforms, engines, data modeling, and access methods.

Previous Section

Next

Go to Part 1

Back to Main

Databases

DB Fundamentals

Storage and Content Delivery
Section 5

Databases
Section 6

- DB Fundamentals
- SQL: RDS
- SQL: Aurora
- NoSQL Databases
- In-Memory Caching

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Relational database management systems (**RDBMS**) are used when the data to be managed has formal and fixed relationships. Data is stored on disk as "rows," and entire rows must be parsed even if individual attributes are all that's needed. Reading one attribute from 10,000 records requires 10,000 rows to be read from the disk.

Every table has a **schema** that defines a fixed layout for each row, which is defined when the table is created. Every row in the table needs to have all the attributes and the correct data types.

RDBMS conforms to the **ACID system**: Atomicity, Consistency, Isolation, and Durability. This impacts the ability to achieve high performance levels and limits scalability, but for more applications of an RDBMS, the trade-off is worth it. SQL (Structured Query Language) is used to interact with most SQL (relational) database products.

ID	Name	Color
0001	Roffle	B/W
0002	Penny	All
0003	Winkie	White
0004	Truffles	Mixed

ID	H_ID	Human Underling
0001	0001	Adrian
0001	0002	Nat
0002	0001	Adrian
0002	0002	Nat
0003	0001	Adrian
0003	0002	Nat
0004	0001	Adrian
0004	0002	Nat

Fixed relationships exist between tables based on keys. Queries join tables to use information from both.

Storage and Content Delivery
Section 5

Databases
Section 6

- DB Fundamentals
- SQL: RDS
- SQL: Aurora
- NoSQL Databases
- In-Memory Caching

Hybrid and Scaling
Section 7

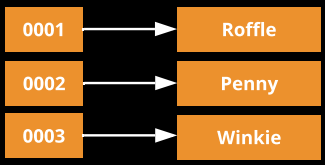
Application, Analytics, and Operations
Section 8

Databases

DB Fundamentals

Key Value

Data is stored as key and value pairs. Super fast queries and ability to scale. No relationships and weak schema.



Amazon DynamoDB

Document

Data is stored as structured key and value pairs called documents. Operations on documents are highly performant.

```
{
    name : "truffles",
    age : "6",
    status : "cat",
    underlings : ["Adrian, "Nat"]
}
```

MongoDB

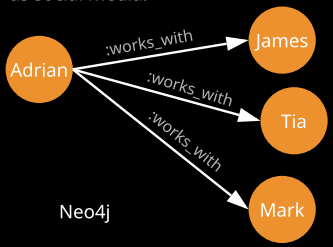
Column

Data is stored in columns rather than rows. Queries against attribute sets, such as all DOBs or all surnames, are fast. Great for data warehousing and analytics.

Amazon Redshift

Graph

Designed for dynamic relationships. Stores data as nodes and relationships between those nodes. Ideal for human-related data, such as social media.



Neo4j

Back
Next Topic

Go to Part 1
Back to Main

Course Navigation

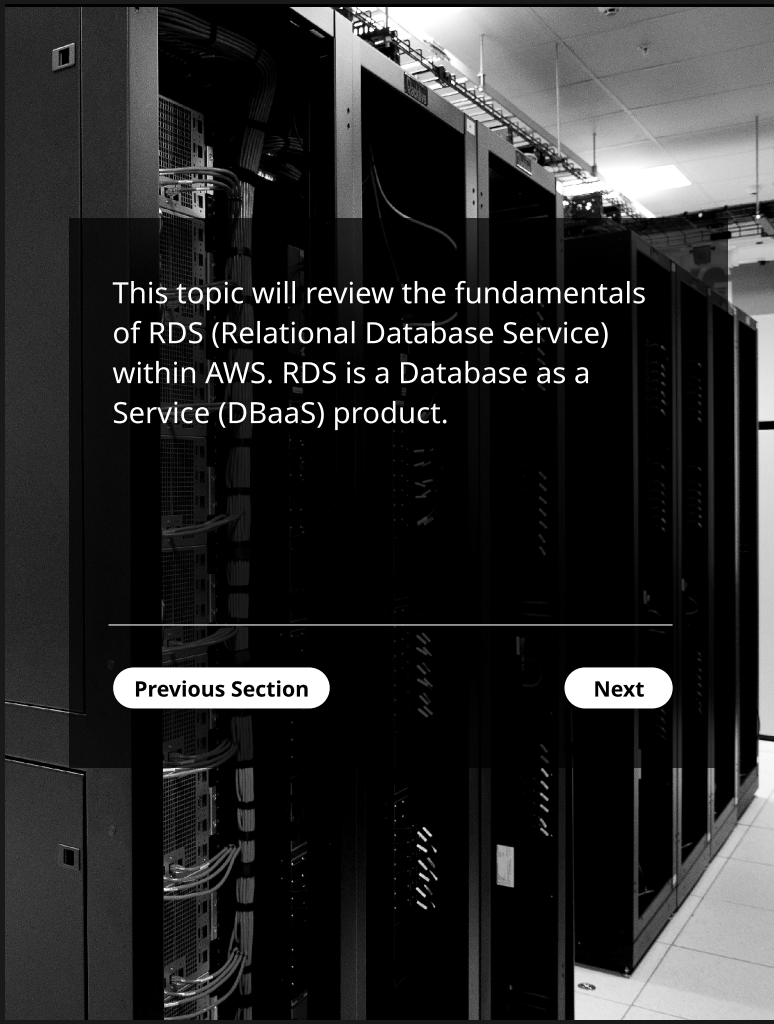
Storage and Content Delivery
Section 5

Databases
Section 6

DB Fundamentals
SQL: RDS
SQL: Aurora
NoSQL Databases
In-Memory Caching

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8



This topic will review the fundamentals of RDS (Relational Database Service) within AWS. RDS is a Database as a Service (DBaaS) product.

Previous Section

Next

Go to Part 1

Back to Main

Databases

SQL: RDS

Storage and Content Delivery
Section 5

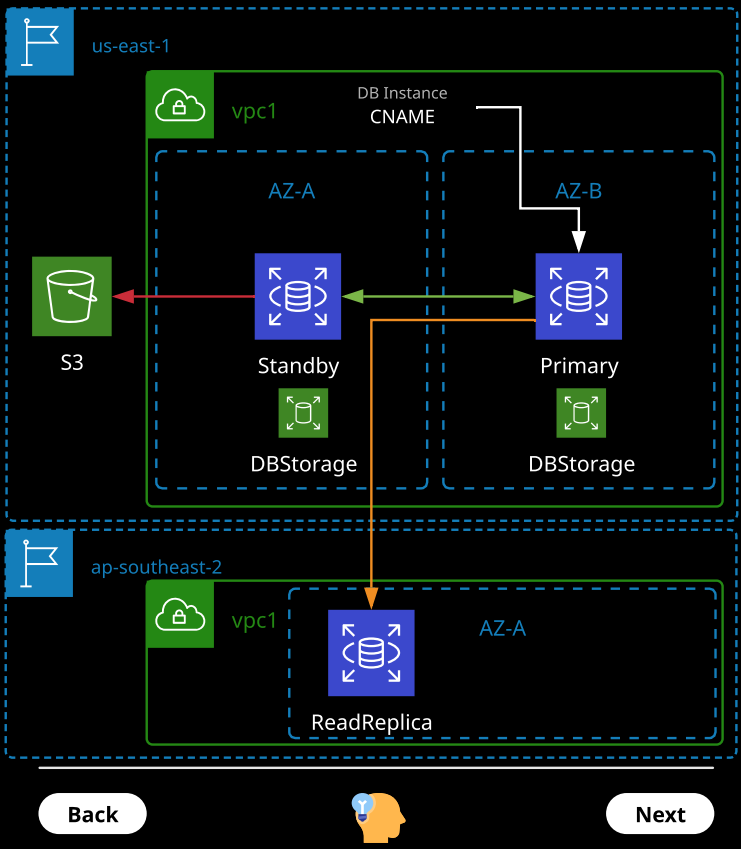
Databases
Section 6

- DB Fundamentals
- SQL: RDS
- SQL: Aurora
- NoSQL Databases
- In-Memory Caching

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

RDS is a Database as a Service (**DBaaS**) product. It can be used to provision a fully functional database without the admin overhead traditionally associated with DB platforms. It can perform at scale, be made **publicly accessible**, and can be configured for demanding availability and durability scenarios.



Back
Next
Go to Part 1
Back to Main

Exam Hints and Key Facts: RDS

RDS supports a number of database engines:

- MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server
- Aurora: An in-house developed engine with substantial feature and performance enhancements

RDS can be deployed in single AZ or Multi-AZ mode (for resilience) and supports the following instance types:

- General purpose (currently DB.M4 and DB.M5)
- Memory optimized (currently DB.R4 and DB.R5, and DB.X1e and DB.X1 for Oracle)
- Burstable (DB.T2 and DB.T3)

Two types of storage are supported:

- General Purpose SSD (gp2): 3 IOPS per GiB, burst to 3,000 IOPS (pool architecture like EBS)
- Provisioned IOPS SSD (io1): 1,000 to 80,000 IOPS (engine dependent) size, and IOPS can be configured independently

RDS instances are charged based on:

- Instance size
- *Provisioned* storage (not used)
- IOPS if using io1
- Data transferred **out**
- Any backups/snapshots beyond the 100% that is free with each DB instance

RDS supports encryption with the following limits/restrictions/conditions:

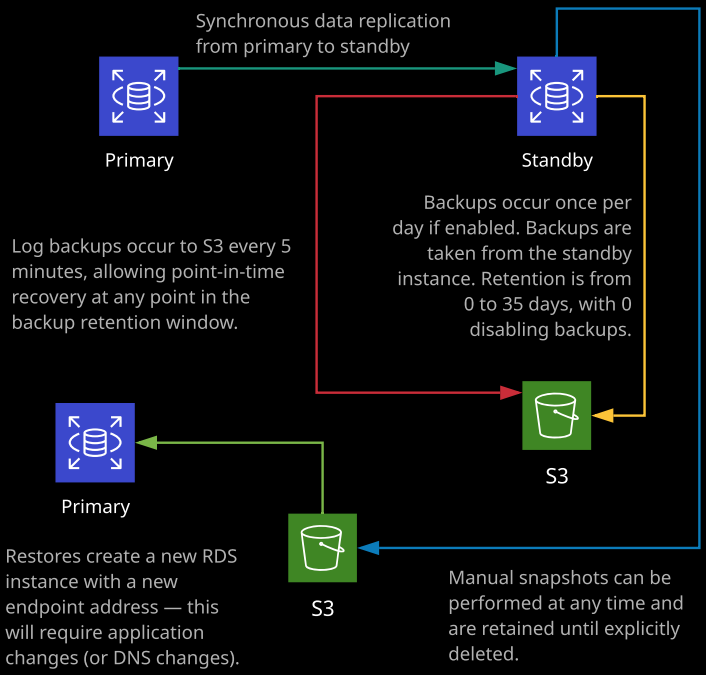
- Encryption can be configured when creating DB instances.
- Encryption can be added by taking a snapshot, making an encrypted snapshot, and creating a new encrypted instance from that encrypted snapshot.
- Encryption cannot be removed.
- Read Replicas need to be the same state as the primary instance (encrypted or not).
- Encrypted snapshots can be copied between regions — but a new destination region KMS CMK is used (because they are region specific).

Network access to an RDS instance is controlled by a **security group (SG)** associated with the RDS instance.

Databases

SQL: RDS

RDS is capable of a number of different types of backups. Automated backups to S3 occur daily and can be retained from 0 to 35 days (with 0 being disabled). Manual snapshots are taken manually and exist until deleted, and point-in-time log-based backups are also stored on S3.



Course Navigation

Storage and Content Delivery
Section 5

Databases
Section 6

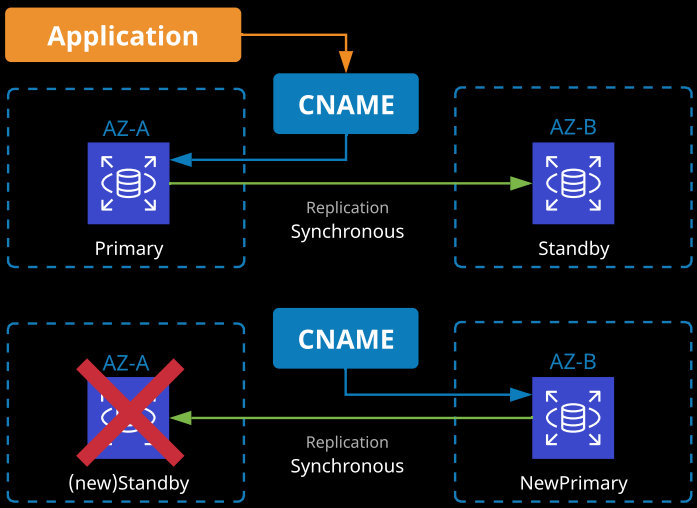
DB Fundamentals
SQL: RDS
SQL: Aurora
NoSQL Databases
In-Memory Caching

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

RDS Multi-AZ

- RDS can be provisioned in single or Multi-AZ mode.
- Multi-AZ provisions a primary instance and a standby instance in a different AZ of the same region.
- Only the primary can be accessed using the instance CNAME.
- There is no performance benefit, but it provides a better RTO than restoring a snapshot.



Replication of data is synchronous — it's copied in real time from the primary to the standby as it's written. The primary and master each have their own storage. Backups are taken using the standby, ensuring no performance impact. Maintenance is performed on the standby first, which is then promoted to minimize downtime.

Back
Next

Go to Part 1
Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

- DB Fundamentals
- SQL: RDS
- SQL: Aurora
- NoSQL Databases
- In-Memory Caching

Hybrid and Scaling
Section 7

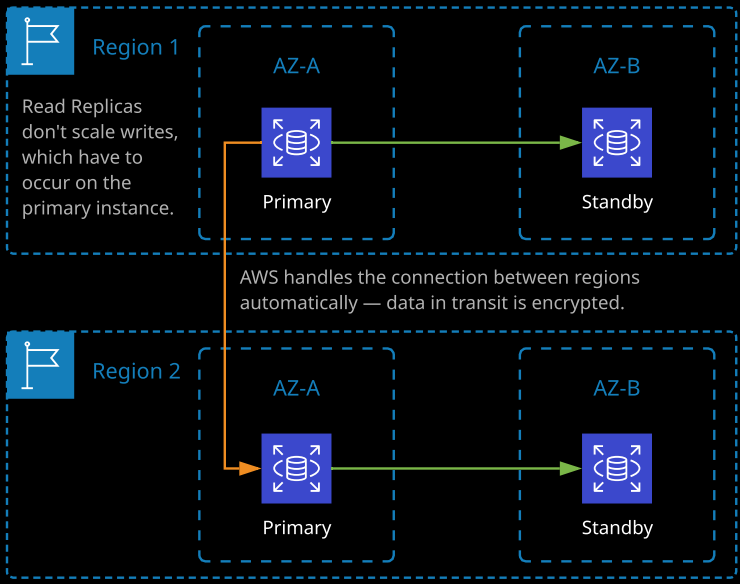
Application, Analytics, and Operations
Section 8

Databases

SQL: RDS

Read Replicas are read-only copies of an RDS instance that can be created in the same region or a different region from the primary instance.

Read Replicas can be addressed independently (each having their own DNS name) and used for read workloads, allowing you to scale reads. Five Read Replicas can be created from a RDS instance, allowing a 5x increase in reads. Read Replicas can be created from Read Replicas, and they can be promoted to primary instances and can be themselves Multi-AZ.



Reads from a Read Replica are eventually consistent — normally seconds, but the application needs to support it.

Course Navigation

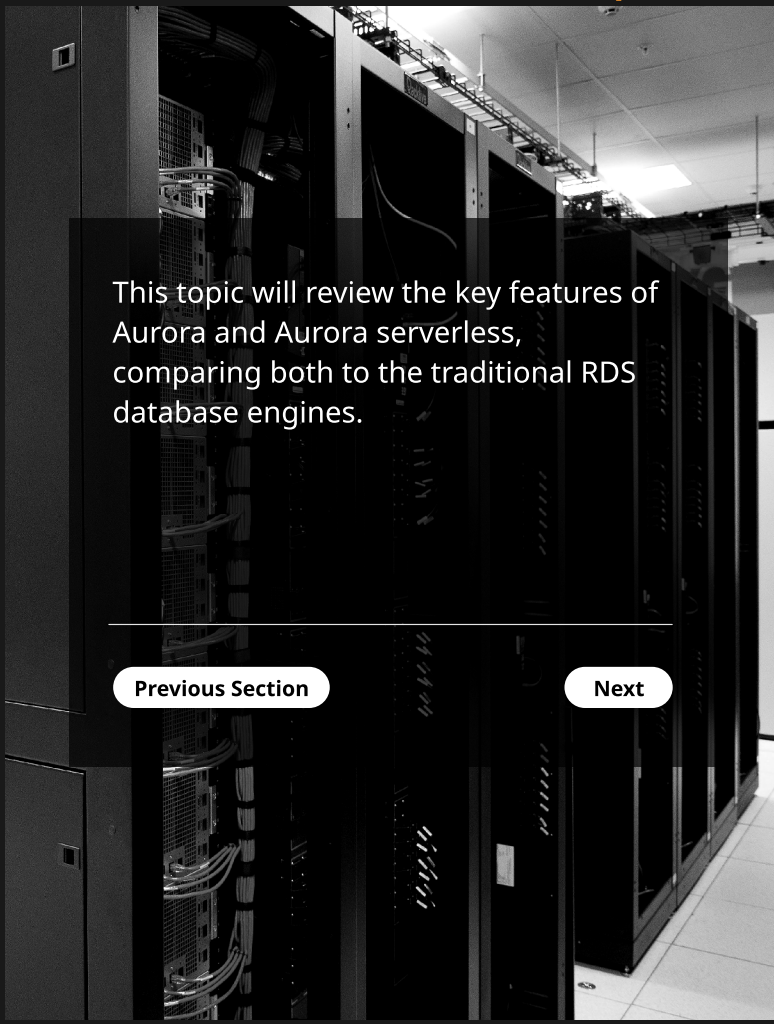
Storage and Content Delivery
Section 5

Databases
Section 6

DB Fundamentals
SQL: RDS
SQL: Aurora
NoSQL Databases
In-Memory Caching

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8



This topic will review the key features of Aurora and Aurora serverless, comparing both to the traditional RDS database engines.

Previous Section

Next

Go to Part 1

Back to Main

Course Navigation

Storage and Content Delivery
Section 5

Databases
Section 6

DB Fundamentals
SQL: RDS
SQL: Aurora
NoSQL Databases
In-Memory Caching

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Aurora Architecture

Aurora is a database engine developed by AWS that is compatible with MySQL, PostgreSQL and associated tools.

Aurora operates with a radically different architecture as opposed to the other RDS database engines:

- Aurora uses a base configuration of a "cluster"
- A cluster contains a single primary instance and zero or more replicas

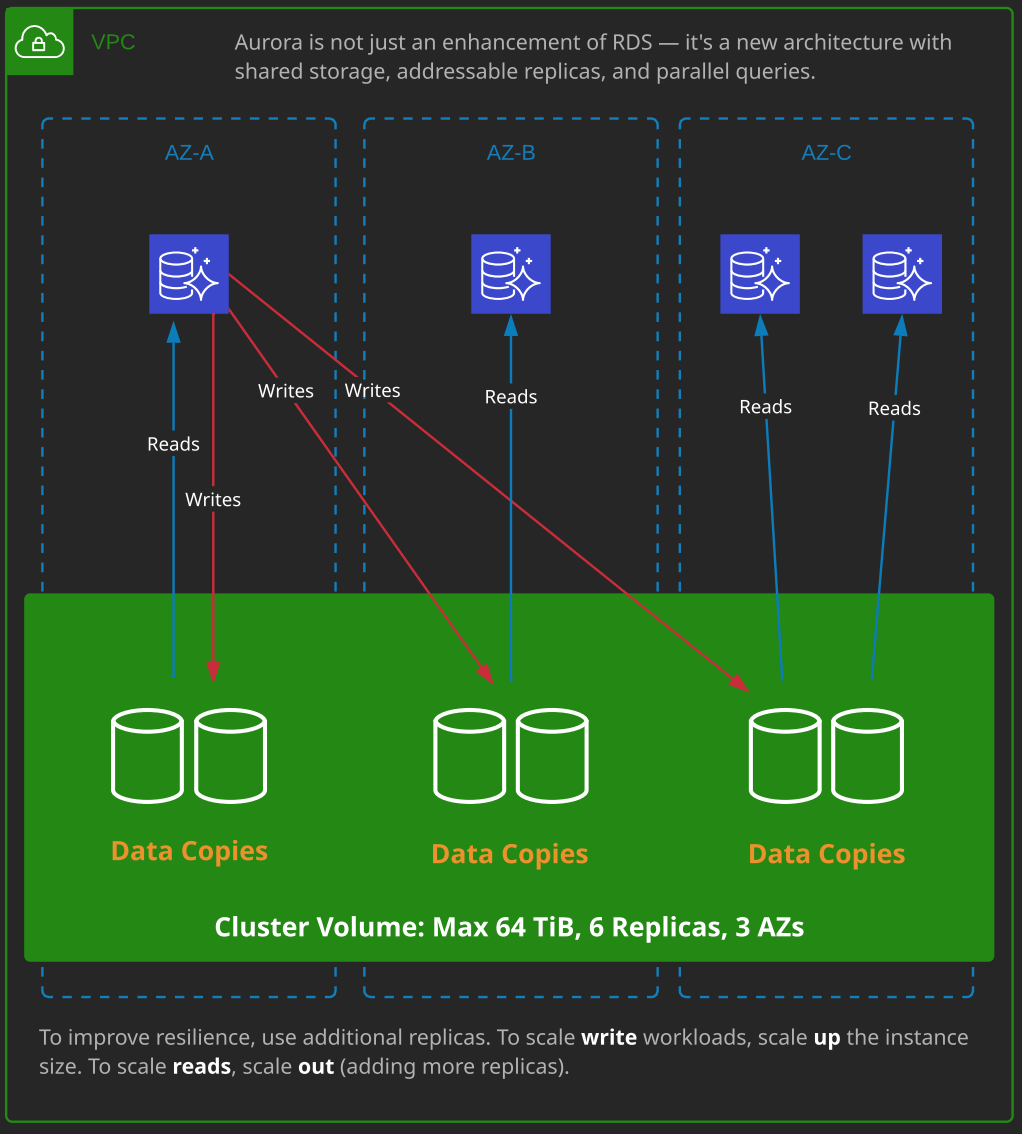
Cluster Storage

- All instances (primary and replicas) use the same shared storage — the cluster volumes.
  - Cluster volume is totally SSD based, which can scale to 64 TiB in size.
  - Replicates data six times, across three Availability Zones.
  - Aurora can tolerate two failures without writes being impacted and three failures without impacting reads.
  - Aurora storage is auto-healing.

Cluster Scaling and Availability

- Cluster volume scales automatically, only bills for **consumed data**, and is constantly backed up to S3.
- Aurora replicas improve availability, can be promoted to be a primary instance quickly, and allow for efficient read scaling.
- Reads and writes use the **cluster endpoint**.
- Reads can use the **reader endpoint**, which balances connections over all replica instances.

Aurora Cluster Architecture



To improve resilience, use additional replicas. To scale **write** workloads, scale **up** the instance size. To scale **reads**, scale **out** (adding more replicas).

Course Navigation

Storage and Content Delivery
Section 5

Databases
Section 6

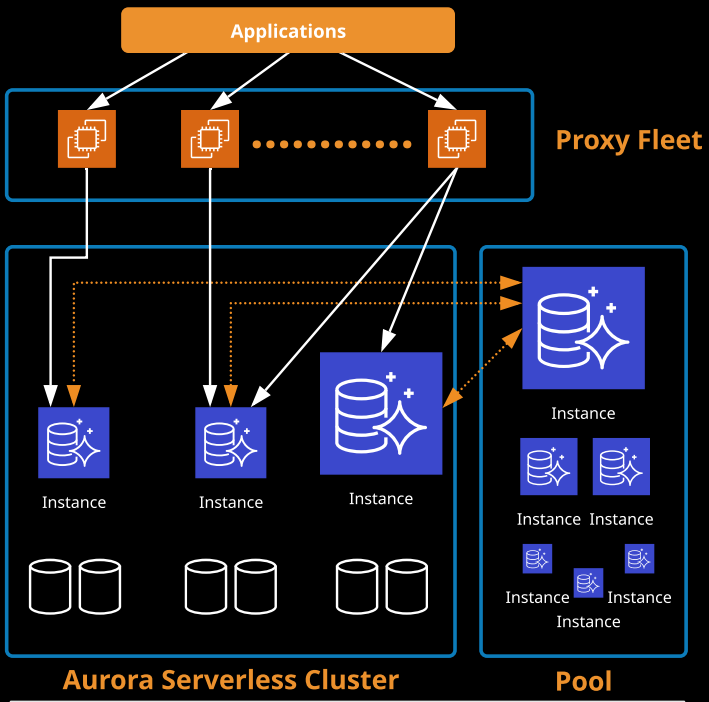
DB Fundamentals
SQL: RDS
SQL: Aurora
NoSQL Databases
In-Memory Caching

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Aurora Serverless

Aurora Serverless is based on the same database engine as Aurora, but instead of provisioning certain resource allocation, Aurora Serverless handles this as a service. You simply specify a minimum and maximum number of Aurora capacity units (**ACUs**) — Aurora Serverless can use the **Data API.**



Back
Next Topic

Go to Part 1
Back to Main

Course Navigation

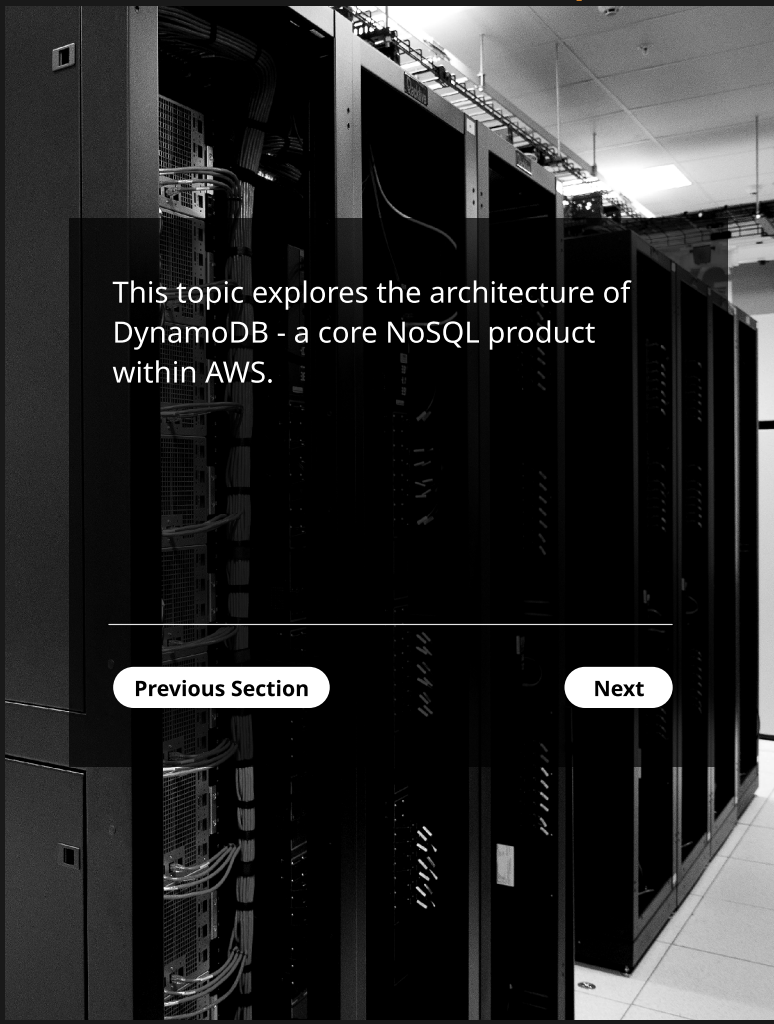
Storage and Content Delivery
Section 5

Databases
Section 6

DB Fundamentals
SQL: RDS
SQL: Aurora
NoSQL Databases
In-Memory Caching

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8



This topic explores the architecture of DynamoDB - a core NoSQL product within AWS.

Previous Section

Next

Go to Part 1

Back to Main

Databases

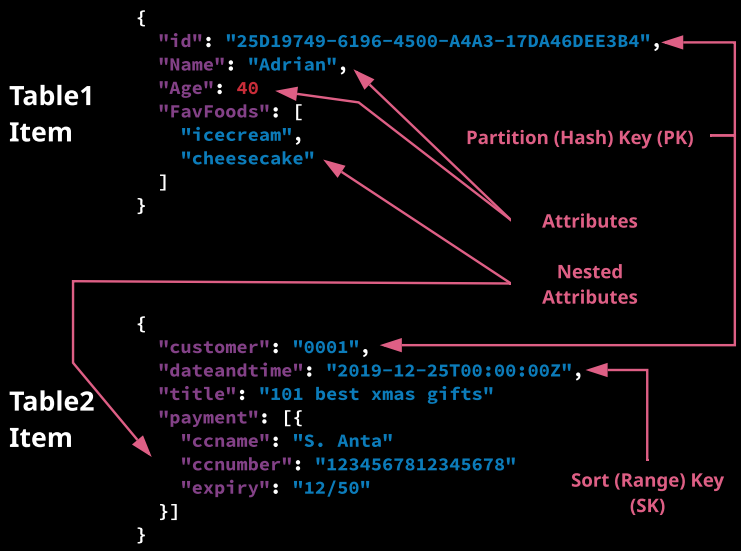
NoSQL Databases

DynamoDB is a NoSQL database service. It's a global service, partitioned regionally and allows the creation of tables.

A **TABLE** is a collection of items that share the same partition key (PK) or partition key and sort key (SK) together with other configuration and performance settings.

An **ITEM** is a collection of attributes (up to **400 KB** in size) inside a table that shares the **same key structure** as every other item in the table.

An **ATTRIBUTE** is a **key** and **value** — an attribute name and value.



Course Navigation

Storage and Content Delivery
Section 5

Databases
Section 6

DB Fundamentals
SQL: RDS
SQL: Aurora
NoSQL Databases
In-Memory Caching

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

DynamoDB has two read/write capacity modes: **provisioned throughput** (default) and **on-demand mode**.

When using on-demand mode, DynamoDB automatically scales to handle performance demands and bills a per-request charge.

Consistency

When using provisioned throughput mode, each table is configured with read capacity units (**RCU**) and write capacity units (**WCU**).

Every operation on ITEMS consumes at least 1 RCU or WCU — partial RCU/WCU cannot be consumed.

Read Capacity Units

One RCU is 4 KB of data read from a table per second in a strongly consistent way. Reading 2 KB of data consumes 1 RCU, reading 4.5 KB of data takes 2 RCU, reading 10* 400 bytes takes 10 RCU. If eventually consistent reads are okay, 1 RCU can allow for 2 x 4 KB of data reads per second. Atomic transactions require 2x the RCU.

Write Capacity Units

One WCU is 1 KB of data or less written to a table. An operation that writes 200 bytes consumes 1 WCU, an operation that writes 2 KB consumes 2 WCU. Five operations of 200 bytes consumes 5 WCU. Atomic transactions require 2x the WCU to complete.

Provisioned Throughput Example Calculations

Course Navigation

Storage and Content Delivery
Section 5

Databases
Section 6

DB Fundamentals
SQL: RDS
SQL: Aurora
NoSQL Databases
In-Memory Caching

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

DynamoDB Consistency

DynamoDB is highly resilient and replicates data across multiple AZs in a region. When you receive a HTTP 200 code, a write has been completed and is durable. This doesn't mean it's been written to all AZs — this generally occurs within a second.

An eventually consistent read will request data, preferring speed. It's possible the data received may not reflect a recent write. Eventual consistency is the default for read operations in DDB.

A strongly consistent read ensures DynamoDB returns the most up-to-date copy of data — it takes longer but is sometimes required for applications that require consistency.

Go to Part 1
Back to Main

Course Navigation

Storage and Content Delivery
Section 5

Databases
Section 6

DB Fundamentals
SQL: RDS
SQL: Aurora
NoSQL Databases
In-Memory Caching

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Provisioned Throughput Calculations

A system needs to store 60 patient records of 1.5 KB, each, every minute. What WCU should you allocate on the patient record table?

- 60 records per minute = ~1 per second (and the DDB RCU/WCU buffer can smooth this out if not)
- Each record is 1.5 KB. 1 WCU = 1 KB per second, so each record requires 2 WCU.
- A WCU setting of 2 is required on the table.

A weather application reads data from a DynamoDB table. Each item in the table is 7 KB in size. How many RCUs should be set on the table to allow for 10 reads per second?

- 1 item is 7 KB, which is 2 RCU (1 RCU is 4 KB).
- 10 reads per second for 7 KB items = 20 RCU
- But the question didn't specify if eventual or strong consistency is required. The default is eventual, which allows for 2 reads of 4 KB per second for 1 RCU.
- Assuming eventually consistent reads, the answer is 10 RCU.

Go to Part 1
Back to Main

Databases

NoSQL Databases

When enabled, streams provide an ordered list of changes that occur to items within a DynamoDB table. A stream is a rolling 24-hour window of changes. Streams are enabled **per table** and only contain data from the point of being enabled.

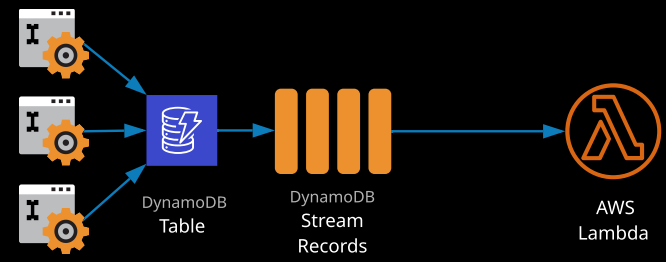
Every stream has an ARN that identifies it globally across all tables, accounts, and regions.

Streams can be configured with one of four view types:

- **KEYS_ONLY**: Whenever an item is added, updated, or deleted, the key(s) of that item are added to the stream.
- **NEW_IMAGE**: The entire item is added to the stream "post-change."
- **OLD_IMAGE**: The entire item is added to the stream "pre-change."
- **NEW_AND_OLD_IMAGES**: Both the new and old versions of the item are added to the stream.

Triggers

Streams can be integrated with AWS Lambda, invoking a function whenever items are changed in a DynamoDB table (a DB trigger).



Databases

NoSQL Databases

Indexes provide an alternative representation of data in a table, which is useful for applications with varying query demands. Indexes come in two forms: local secondary indexes (LSI) and global secondary indexes (GSI). Indexes are interacted with as though they are tables, but they are just an alternate representation of data in an existing table.

Local secondary indexes must be created at the same time as creating a table. They use the same partition key but an alternative sort key. They share the RCU and WCU values for the main table.

Global secondary indexes can be created at any point after the table is created. They can use different partition *and* sort keys. They have their own RCU and WCU values.

UserID	Game	HighScore	DateandTime
0001	Beat Saber	10	2019-07-21T00:01:00Z
0001	WoW	458	2019-07-21T13:37:00Z
0002	Beat Saber	100000	2019-07-21T13:38:00Z
0003	RapBattle	67	2019-07-23T13:38:00Z

Efficient queries can only be done on user ID and filtered or sorted using game.

UserID	DateandTime	Game	HighScore
0001	2019-07-21T00:01:00Z	Beat Saber	10
0001	2019-07-21T13:37:00Z	Wow	458
0002	2019-07-21T13:38:00Z	Beat Saber	100000
0003	2019-07-23T13:38:00Z	RapBattle	67

An LSI lets you use an alternative sort key to allow filtering on date and time instead.

Game	DateandTime	UserID	HighScore
Beat Saber	2019-07-21T00:01:00Z	0001	10
Beat Saber	2019-07-21T13:38:00Z	0002	100000
RapBattle	2019-07-23T13:38:00Z	0003	67
WoW	2019-07-21T13:37:00Z	0001	458

A GSI can use an alternative PK and SK — in this example, maybe for a high score table per game.

Course Navigation

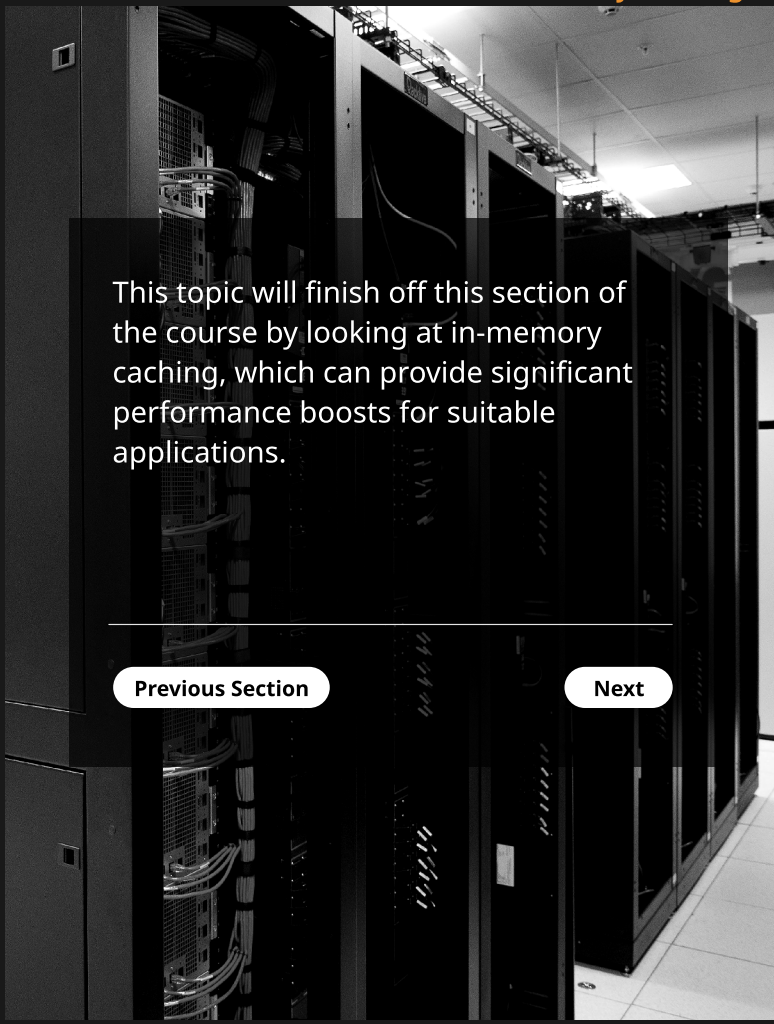
Storage and Content Delivery
Section 5

Databases
Section 6

- DB Fundamentals
- SQL: RDS
- SQL: Aurora
- NoSQL Databases
- In-Memory Caching

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8



This topic will finish off this section of the course by looking at in-memory caching, which can provide significant performance boosts for suitable applications.

Previous Section

Next

Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

- DB Fundamentals
- SQL: RDS
- SQL: Aurora
- NoSQL Databases
- In-Memory Caching

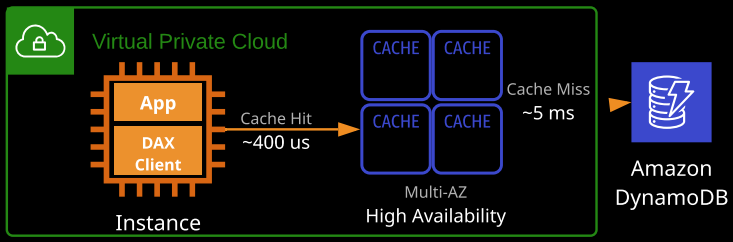
Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

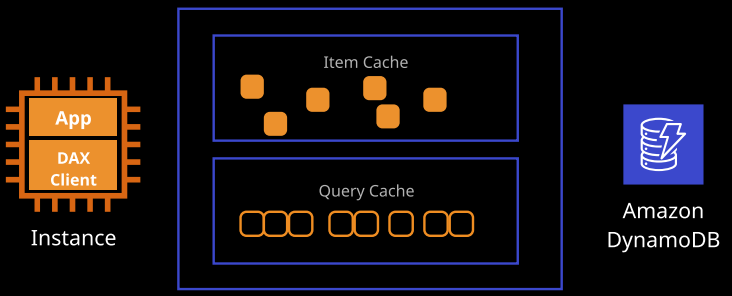
Databases

In-Memory Caching

DynamoDB Accelerator (DAX) is an in-memory cache designed specifically for DynamoDB. Results delivered from DAX are available in microseconds rather than in the single-digit milliseconds available from DynamoDB.



DAX maintains two distinct caches: the item cache and the query cache. The item cache is populated with results from `GetItem` and `BatchGetItem` and has a five-minute default TTL. The query cache stores results of `Query` and `Scan` operations and caches based on the parameters specified.



Back
Next

Go to Part 1
Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

- DB Fundamentals
- SQL: RDS
- SQL: Aurora
- NoSQL Databases
- In-Memory Caching

Hybrid and Scaling
Section 7

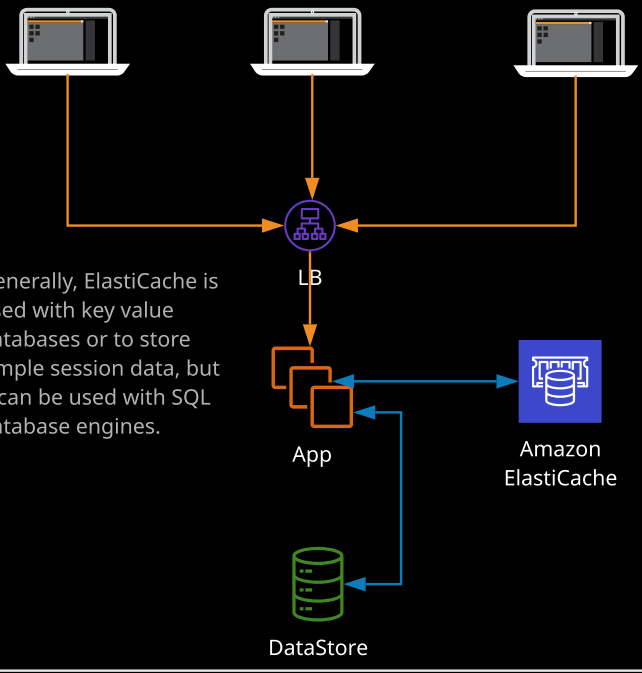
Application, Analytics, and Operations
Section 8

Databases

In-Memory Caching

ElastiCache is a managed in-memory data store supporting the Redis or Memcached engines. ElastiCache is used for two common use cases:

- Offloading database reads by caching responses, improving application speed and reducing costs
- Storing user session state, allowing for stateless compute instances (used for fault-tolerant architectures)



Generally, ElastiCache is used with key value databases or to store simple session data, but it can be used with SQL database engines.

Course Navigation

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

This topic will walk through the load balancing and Auto Scaling features of AWS, which support scalable and elastic architectures.

Previous Section

Next

Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

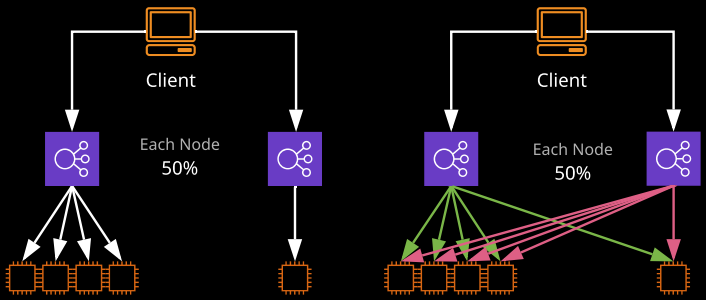
- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

Hybrid and Scaling

LB and Auto Scaling

- Load balancing is a method used to distribute incoming connections across a group of servers or services.
- Incoming connections are made to the load balancer, which distributes them to associated services.
- Elastic Load Balancing (ELB) is a service that provides a set of highly available and scalable load balancers in one of three versions: Classic (CLB), Application (ALB), and Network (NLB).
- ELBs can be paired with Auto Scaling groups to enhance high availability and fault tolerance — automating scaling/elasticity.
- An elastic load balancer has a DNS record, which allows access at the external side.



A node is placed in each AZ the load balancer is active in. Each node gets 1/N of the traffic, where N is the number of nodes. Historically, each node could only load balance to instances in the same AZ. This results in uneven traffic distribution. Cross-zone load balancing allows each node to distribute traffic to all instances.

An elastic load balancer can be public facing, meaning it accepts traffic from the public internet, or internal, which is only accessible from inside a VPC and is often used between application tiers.

An elastic load balancer accepts traffic via listeners using protocol and ports. It can strip HTTPS at this point, meaning it handles encryption/decryption, reducing CPU usage on instances.

Back
Next

Go to Part 1
Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

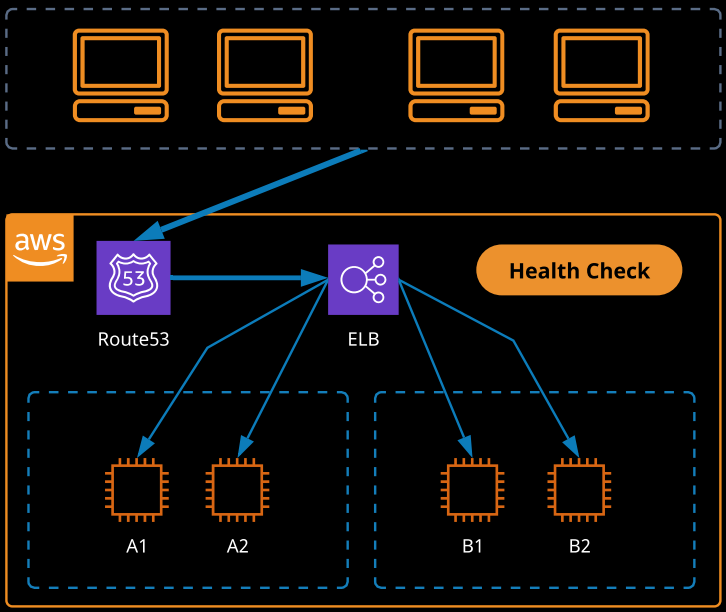
Application, Analytics, and Operations
Section 8

Hybrid and Scaling

LB and Auto Scaling

Classic Load Balancers are the oldest type of load balancer and generally should be avoided for new projects.

- Support Layer 3 & 4 (TCP and SSL) and some HTTP/S features
- It isn't a Layer 7 device, so no real HTTP/S
- One SSL certificate per CLB — can get expensive for complex projects
- Can *offload* SSL connections — HTTPS to the load balancer and HTTP to the instance (lower CPU and admin overhead on instances)
- Can be associated with Auto Scaling groups
- DNS A Record is used to connect to the CLB



Course Navigation

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

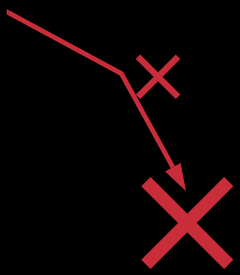
- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

LB Health Checks

Health checks can be configured to check the health of any attached services. If a problem is detected, incoming connections won't be routed to instances until it returns to health.

CLB health checks can be TCP, HTTP, HTTPS, and SSL based on ports 1-65535. With HTTP/S checks, a HTTP/S path can be tested.



Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

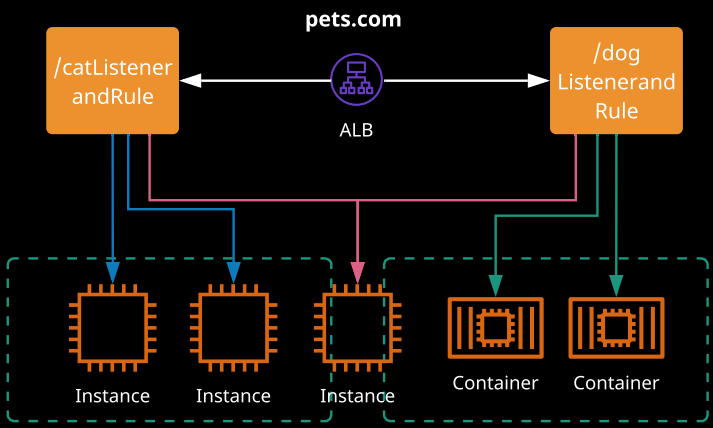
- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

Hybrid and Scaling

LB and Auto Scaling

- Application Load Balancers (ALBs) operate at Layer 7 of the OSI model. They understand HTTP and HTTPS and can load balance based on this protocol layer.
- ALBs are now recommended as the default LB for VPCs. They perform better than CLBs and are almost always cheaper.
- Content rules can direct certain traffic to specific target groups.
  - Host-based rules: Route traffic based on the host used
  - Path-based rules: Route traffic based on URL path
- ALBs support EC2, ECS, EKS, Lambda, HTTPS, HTTP/2 and WebSockets, and they can be integrated with AWS Web Application Firewall (WAF).
- Use an ALB if you need to use containers or microservices.
- Targets -> Target Groups -> Content Rules
- An ALB can host multiple SSL certificates using SNI.



Back

Next

Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

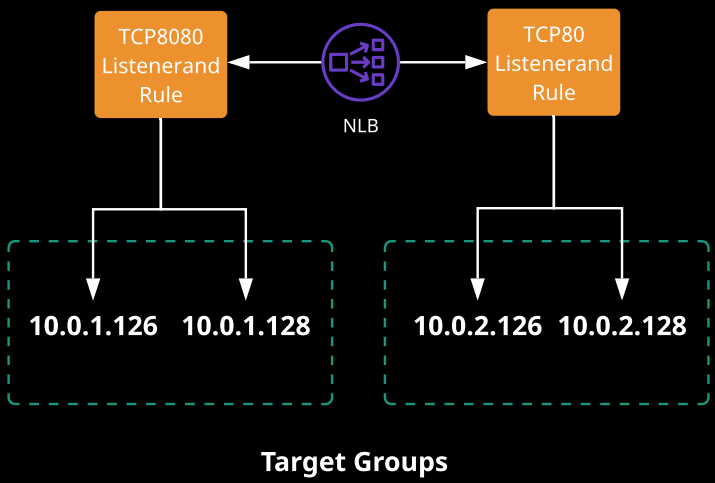
Application, Analytics, and Operations
Section 8

Hybrid and Scaling

LB and Auto Scaling

Network Load Balancers (NLBs) are the newest type of load balancer and operate at Layer 4 of the OSI network model. There are a few scenarios and benefits to using an NLB versus an ALB:

- Can support protocols other than HTTP/S because it forwards upper layers unchanged
- Less latency because no processing above Layer 4 is required
- IP addressable — static address
- Best load balancing performance within AWS
- Source IP address preservation — packets unchanged
- Targets can be addressed using IP address



Back Next

Go to Part 1
Back to Main

Hybrid and Scaling

LB and Auto Scaling

Launch templates and **launch configurations** allow you to configure various configuration attributes that can be used to launch EC2 instances. Typical configurations that can be set include:

- AMI to use for EC2 launch
- Instance type
- Storage
- Key pair
- IAM role
- User data
- Purchase options
- Network configuration
- Security group(s)

Launch templates address some of the weaknesses of the legacy launch configurations and add the following features:

- Versioning and inheritance
- Tagging
- More advanced purchasing options
- New instance features, including:
  - Elastic graphics
  - T2/T3 unlimited settings
  - Placement groups
  - Capacity reservations
  - Tenancy options

Launch templates should be used over launch configurations where possible. **Neither can be edited after creation** — a new version of the template or a new launch configuration should be created.

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

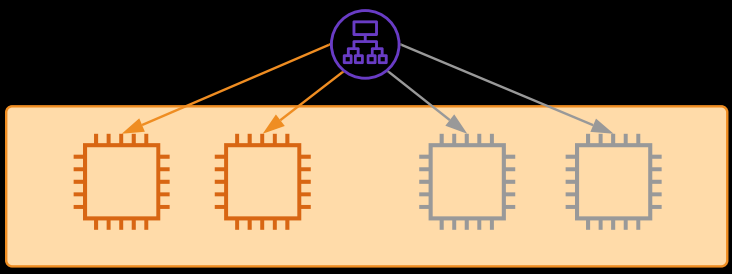
- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

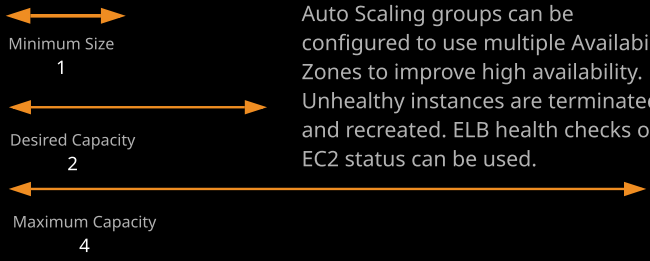
Application, Analytics, and Operations
Section 8

Hybrid and Scaling

LB and Auto Scaling

Auto Scaling groups use launch configurations or launch templates and allow automatic scale-out or scale-in based on configurable metrics. Auto Scaling groups are often paired with elastic load balancers.





Auto Scaling groups can be configured to use multiple Availability Zones to improve high availability. Unhealthy instances are terminated and recreated. ELB health checks or EC2 status can be used.

Metrics such as CPU utilization or network transfer can be used either to scale out or scale in using scaling policies. Scaling can be manual, scheduled, or dynamic. Cooldowns can be defined to ensure rapid in/out events don't occur.

Scaling policies can be simple, step scaling, or target tracking.

Hybrid and Scaling

VPN and Direct Connect

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

This topic walks through the hybrid connectivity options within AWS, covering VPC VPN and Direct Connect.

Previous Section

Next

Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

- **LB and Auto Scaling**
- **VPN and Direct Connect**
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

Hybrid and Scaling

VPN and Direct Connect

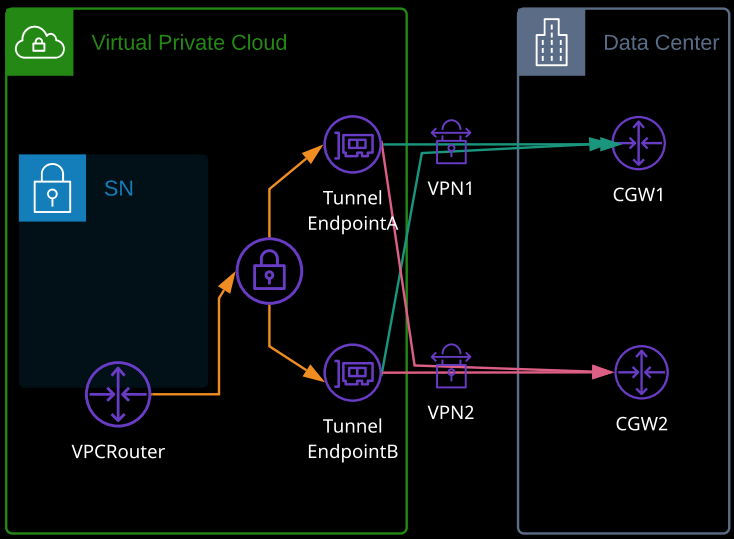
VPC Virtual Private Networks (**VPNs**) provide a software-based secure connection between a VPC and on-premises networks.

VPC VPN Components

- A Virtual Private Cloud (VPC)
- Virtual private gateway (VGW) attached to a VPC
- A customer gateway (CGW) — configuration for on-premises router
- VPN connection (using one or two IPsec tunnels)

Best Practice and HA

- Use dynamic VPNs (uses BGP) where possible
- Connect both tunnels to your CGW — VPC VPN is HA by design
- Where possible, use two VPN connections and two CGWs



Back

Next

Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

- **LB and Auto Scaling**
- **VPN and Direct Connect**
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

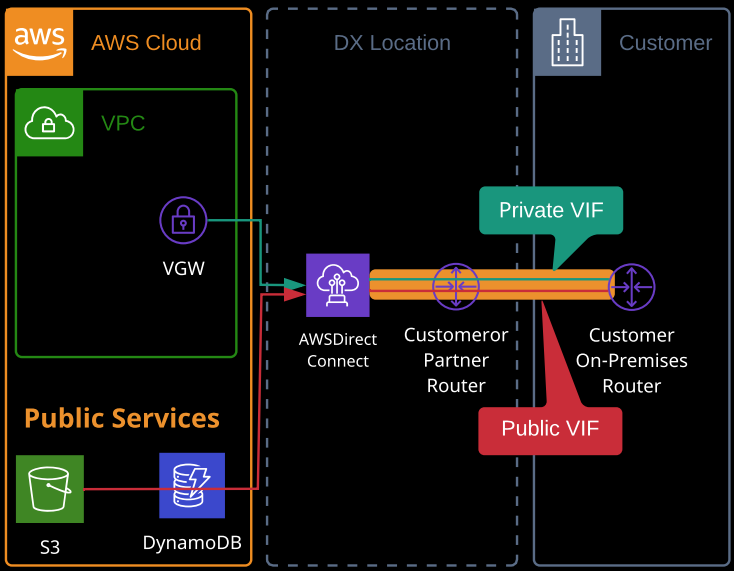
Hybrid and Scaling

VPN and Direct Connect

A Direct Connect (**DX**) is a physical connection between your network and AWS either directly via a cross-connect and customer router at a DX location or via a DX partner.

Dedicated Connections are direct via AWS and use single-mode fiber, running either **1 Gbps** using 1000Base-LX or **10 Gbps** using 10GBASE-LR.

Virtual interfaces (VIFs) run on top of a DX. Public VIFs can access AWS public services such as S3 only. Private VIFs are used to connect into VPCs. DX is not highly available or encrypted.



Back
Next

Go to Part 1
Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

Hybrid and Scaling

VPN and Direct Connect

Choosing between Direct Connect (DX) and VPC VPN is a critical part of any connectivity-based exam questions.

VPN

- Urgent need — can be deployed in minutes
- Cost constrained — cheap and economical
- Low end or consumer hardware — DX requires BGP
- Encryption required
- Flexibility to change locations
- Highly available options available
- Short-term connectivity (DX generally has physical minimums due to the physical transit connections required) — not applicable if you are in a DX location because then it's almost on demand

Direct Connect

- Higher throughput
- Consistent performance (throughput)
- Consistent low latency
- Large amounts of data — cheaper than VPN for higher volume
- No contention with existing internet connection

Both

- VPN as a cheaper HA option for DX
- VPN as an additional layer of HA (in addition to two DX)
- If some form of connectivity is needed immediately, provides it before the DX connection is live
- Can be used to add encryption over the top of a DX (public VIF VPN)

Back

Next Topic

Go to Part 1

Back to Main

Hybrid and Scaling

Snowball and Snowmobile

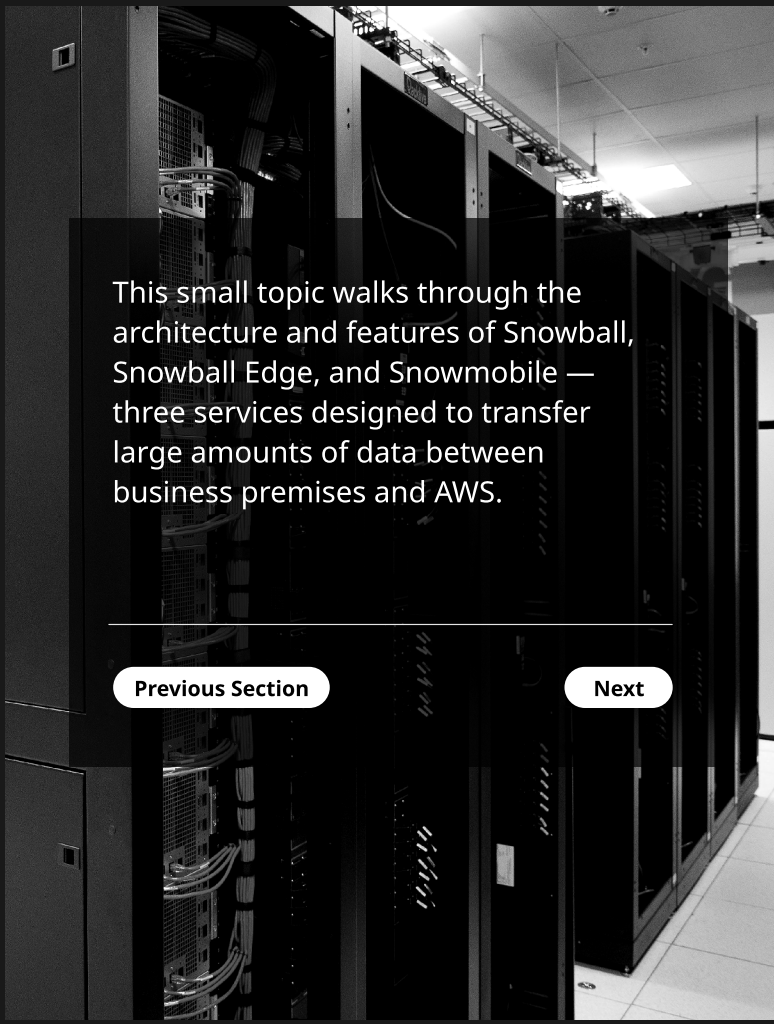
Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8



This small topic walks through the architecture and features of Snowball, Snowball Edge, and Snowmobile — three services designed to transfer large amounts of data between business premises and AWS.

Previous Section

Next

Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

Hybrid and Scaling

Snowball and Snowmobile

AWS provide three methods for moving large amounts of data quickly in and out of AWS:

- Snowball
- Snowball Edge
- Snowmobile

With any of the snow* devices, you don't need to worry about writing code or the speed or data allocation of your internet, VPN, or DX connection. With snow*, you log a job and receive an empty device or one full of the data requested. You can perform a data copy with your usual tooling and ship the device back.

Snowball

- Can be used for **in** or **out** jobs
- Log a job and an empty device or device with data is shipped
- Ideal for TB or PB data transfers — 50 TB or 80 TB capacity per Snowball
- 1 Gbps (RJ45 1Gbase-TX) or 10 Gbps (LR/SR) using a SFP
- Data encryption using KMS
- Generally used from 10 TB -> 10 PB (the economical range)
- Larger jobs or multiple locations can use multiple Snowballs
- End-to-end process time is low for the amount of data week(s)

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

Hybrid and Scaling

Snowball and Snowmobile

Snowball Edge

- Includes both storage and compute
- Larger capacity
- 10 Gbps (RJ45), 10/25 Gbps (SFP), 45/50/100 Gbps (QSFP+)
- Compute can be used for local instances or Lambda functionality
- Three versions:
  - Edge Storage Optimized: 80 TB, 24 vCPU, and 32 GiB RAM
  - Edge Compute Optimized: 100 TB + 7.68 TB NVMe, 52 vCPUs, and 208 GiB RAM
  - Edge Compute Optimized with GPU: As above with a GPU equivalent to P3 EC2 instance
- Compute can be used for local IoT, for data processing prior to ingestion into AWS, and much more
- Used in the same type of situations as Snowballs but when compute is required

Snowmobile

- Portable storage data center within a shipping container on a semi truck
- Available in certain areas via special order from AWS
- Used when single location 10 PB+ is required
- Each Snowmobile can transfer up to 100 PB
- Not economical for sub 10 PB and where multiple locations are required
- Situated on side and connected into your data center for the duration of the transfer

Back

Next Topic

Go to Part 1

Back to Main

Hybrid and Scaling

Data and DB Migration

Course Navigation

Storage and Content Delivery
Section 5

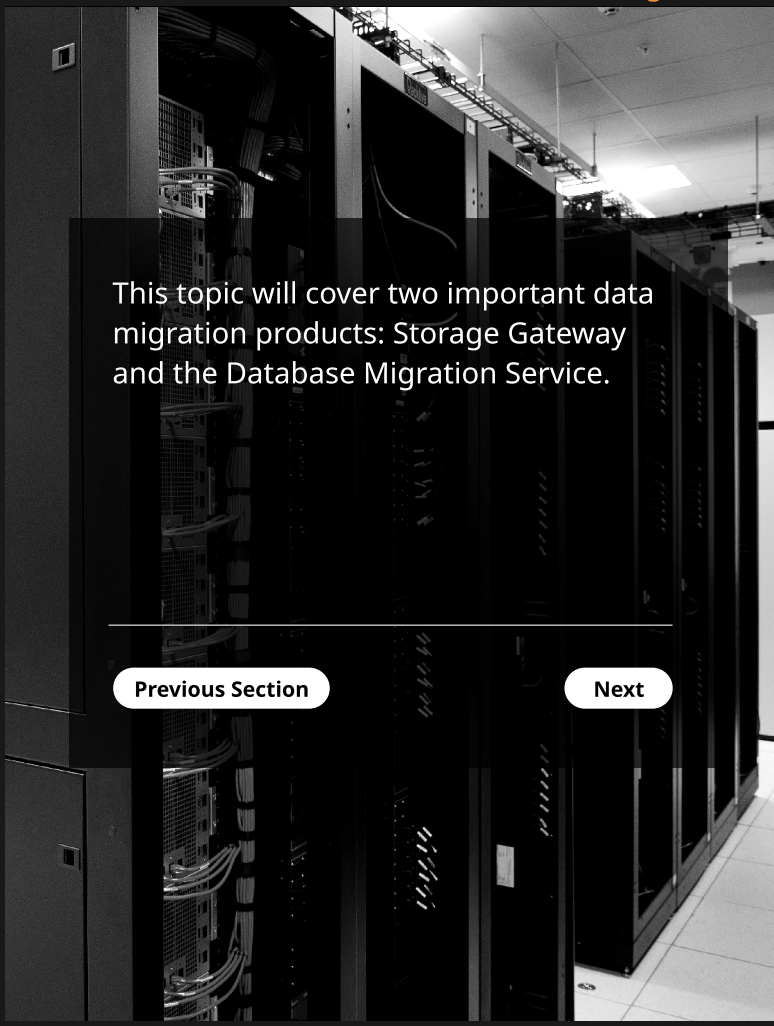
Databases
Section 6

Hybrid and Scaling
Section 7

- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration

ID Federation and SSO

Application, Analytics, and Operations
Section 8



This topic will cover two important data migration products: Storage Gateway and the Database Migration Service.

Previous Section

Next

Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

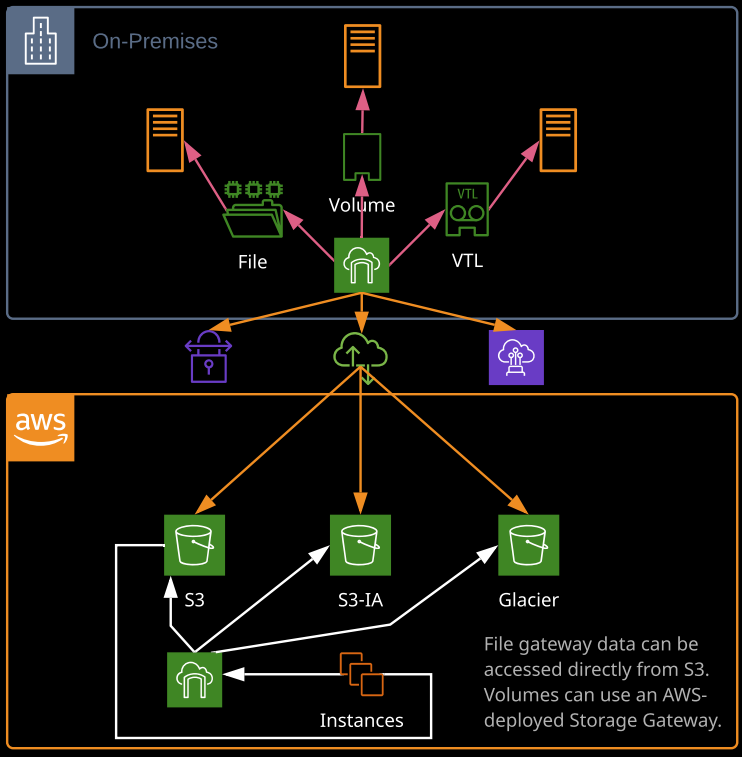
- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

Hybrid and Scaling

Data and DB Migration

Storage Gateway is a hybrid storage service that allows you to migrate data into AWS, extending your on-premises storage capacity using AWS. There are three main types of Storage Gateway: **file gateway**, **volume gateway**, and **tape gateway**.



Back
Next

Go to Part 1
Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

- **LB and Auto Scaling**
- **VPN and Direct Connect**
- **Snowball and Snowmobile**
- **Data and DB Migration**
- ID Federation and SSO

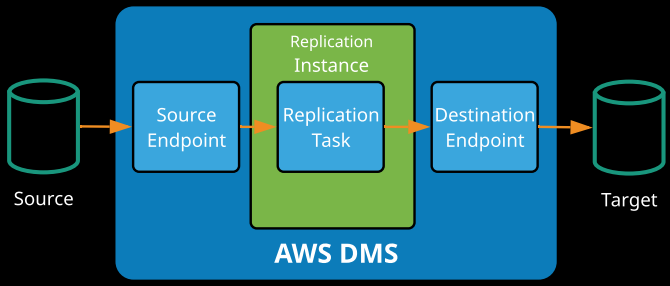
Application, Analytics, and Operations
Section 8

Hybrid and Scaling

Data and DB Migration

Database Migration Service (AWS DMS) is a service to migrate relational databases. It can migrate **to** and **from** any locations with network connectivity to AWS.

- DMS is compatible with a broad range of DB sources, including Oracle, MS SQL, MySQL, MariaDB, PostgreSQL, MongoDB, Aurora, and SAP.
- Data can be synced to most of the above engines, as well as Redshift, S3, and DynamoDB.
- You can also use the Schema Conversion Tool (AWS SCT) to transform between different database engines as part of a migration.



With DMS at a high level, you provision a replication instance, define source and destination endpoints that point at source and target databases, and create a replication task. DMS handles the rest, and you can continue using your database while the process runs. DMS is useful in a number of common scenarios:

- Scaling database resources **up** or **down** without downtime
- Migrating databases from on-premises to AWS, from AWS to on-premises, or **to/from** other cloud platforms.
- Moving data between different DB engines, including schema conversion
- Partial/subset data migration
- Migration with little to no admin overhead, as a service

Back

Next Topic

Go to Part 1

Back to Main

Course Navigation

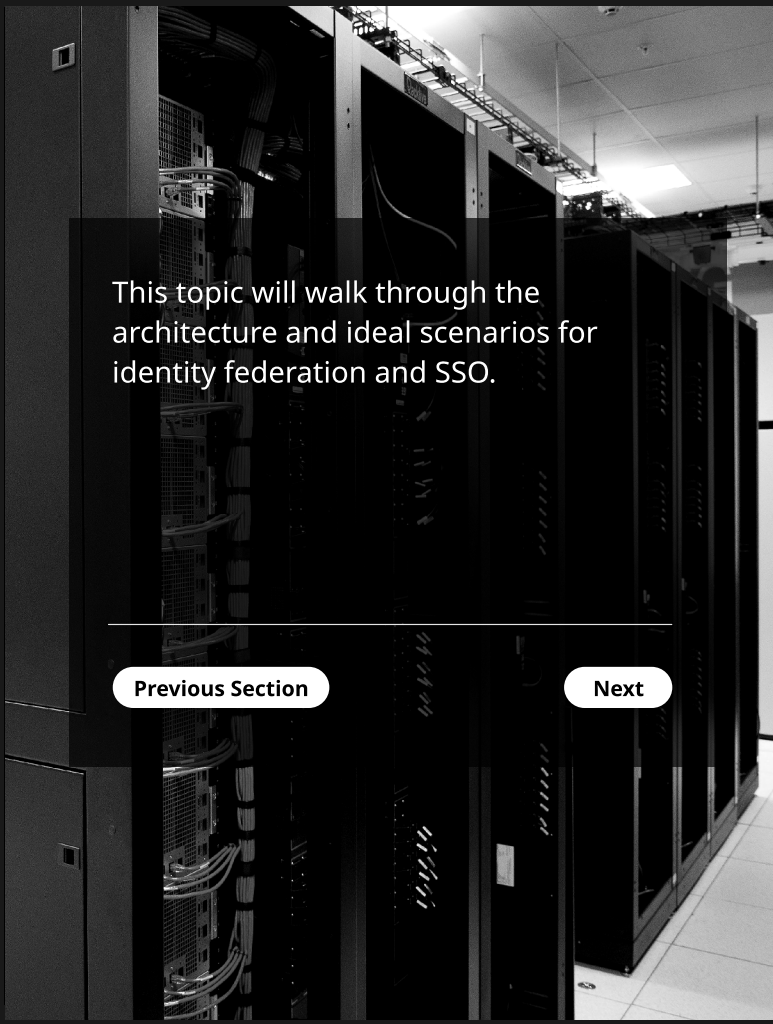
Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8



This topic will walk through the architecture and ideal scenarios for identity federation and SSO.

Previous Section

Next

Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

Hybrid and Scaling

ID Federation and SSO

Identity federation (IDF) is an architecture where identities of an external identity provider (IDP) are recognized. Single sign-on (SSO) is where the credentials of an external identity are used to allow access to a local system (e.g., AWS).

Types of IDF include:

- **Cross-account roles**: A remote account (IDP) is allowed to assume a role and access your account's resources.
- **SAML 2.0 IDF**: An on-premises or AWS-hosted directory service instance is configured to allow Active Directory users to log in to the AWS console.
- **Web Identity Federation**: IDPs such as Google, Amazon, and Facebook are allowed to assume roles and access resources in your account.

Cognito and the **Secure Token Service** (**STS**) are used for IDF. A federated identity is verified using an external IDP and by proving the identity (using a token or assertion of some kind) is allowed to swap that ID for temporary AWS credentials by assuming a role.

SAML 2.0 | Web Identity

Back | Next

Go to Part 1

Back to Main

Course Navigation

Storage and Content Delivery
Section 5

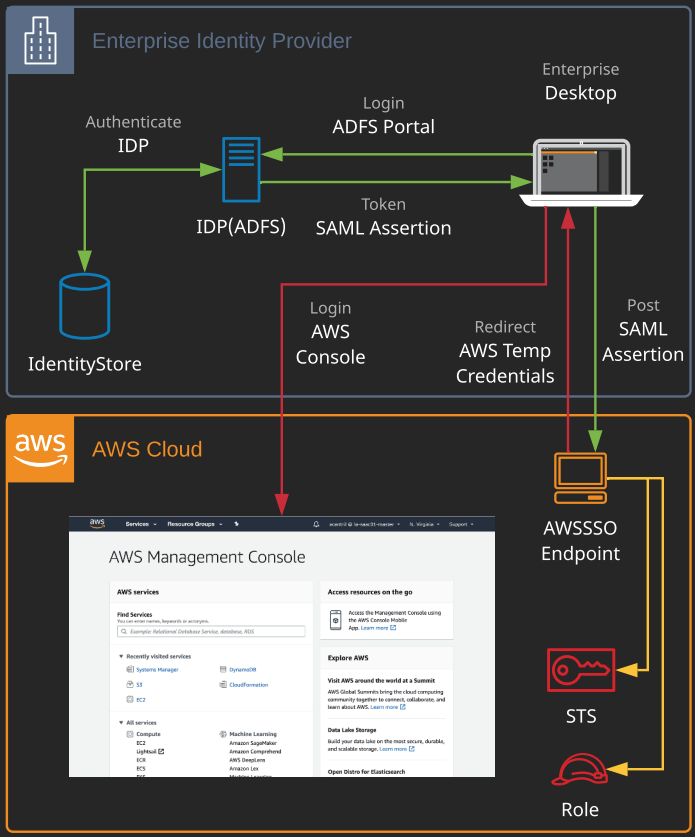
Databases
Section 6

Hybrid and Scaling
Section 7

- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

SAML 2.0 Federation



Go to Part 1

Back to Main

Linux Academy

Course Navigation

Storage and Content Delivery
Section 5

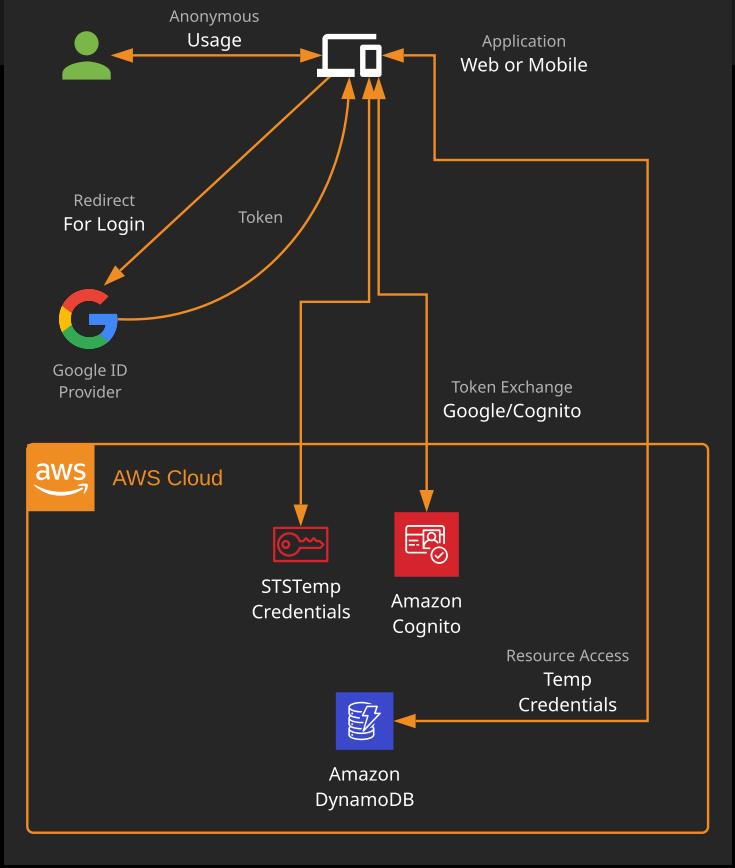
Databases
Section 6

Hybrid and Scaling
Section 7

- LB and Auto Scaling
- VPN and Direct Connect
- Snowball and Snowmobile
- Data and DB Migration
- ID Federation and SSO

Application, Analytics, and Operations
Section 8

Web Identity Federation



Go to Part 1

Back to Main

Linux Academy

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

LB and Auto Scaling
VPN and Direct Connect
Snowball and Snowmobile
Data and DB Migration
ID Federation and SSO

Application, Analytics, and Operations
Section 8

Hybrid and Scaling

ID Federation and SSO

What is just as important as **how** to use IDF is **when** to use IDF. The exam will test your understanding of situations where IDF should be used versus IAM identities.

Enterprise Access to AWS Resources

- Users/staff have an existing pool of identities.
- You need those identities to be used across all enterprise systems, including AWS.
- Access to AWS resources using SSO.
- Potentially tens or hundreds of thousands of users — more than IAM can handle.
- You might have an ID team within your business.

Mobile and Web Applications

- Mobile or web application requires access to AWS resources.
- You need a certain level of guest access — and extra once logged in.
- Customers have other identities — Google, Twitter, Facebook, etc. — and need to use those.
- You don't want credentials stored within the application.
- Could be millions or more users — beyond the capabilities of IAM.
- Customers might have multiple third-party logins, but they represent one real person.

Centralized Identity Management (AWS Accounts)

- Tens or hundreds of AWS accounts in an organization.
- Need central store of IDs — either IAM or an external provider.
- Role switching used from an ID account into member accounts.

Application, Analytics, and Operations

Course Navigation

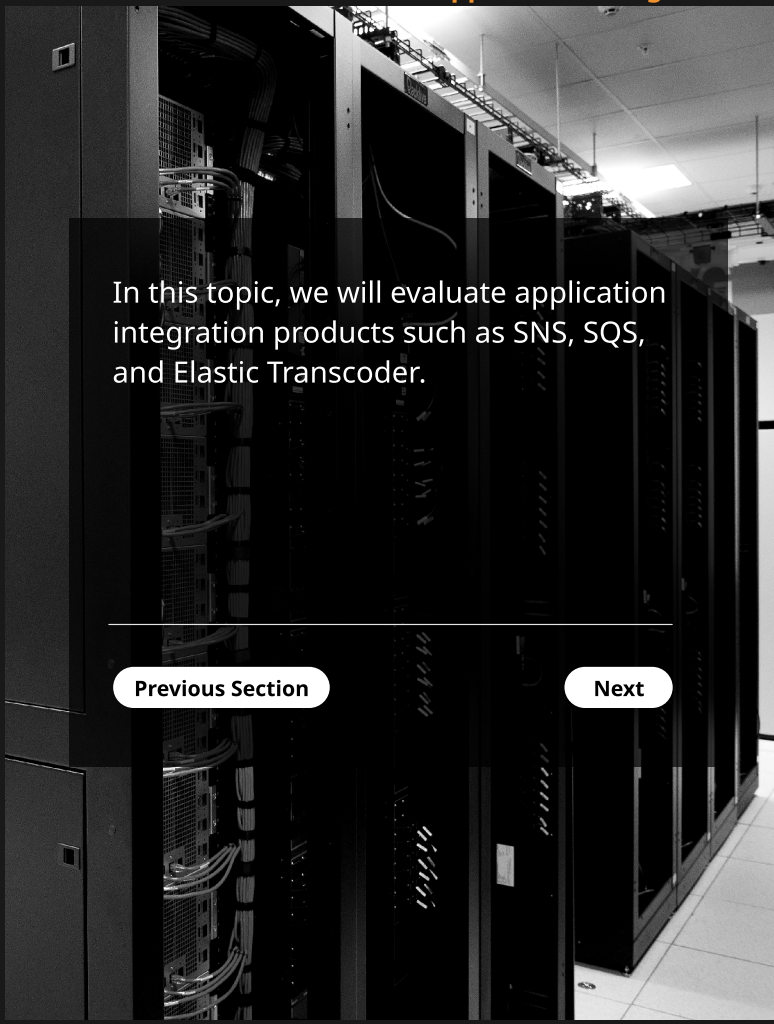
Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Application Integration
Analytics
Logging and Monitoring
Operations
Deployment



In this topic, we will evaluate application integration products such as SNS, SQS, and Elastic Transcoder.

Previous Section

Next

Go to Part 1

Back to Main

Application, Analytics, and Operations

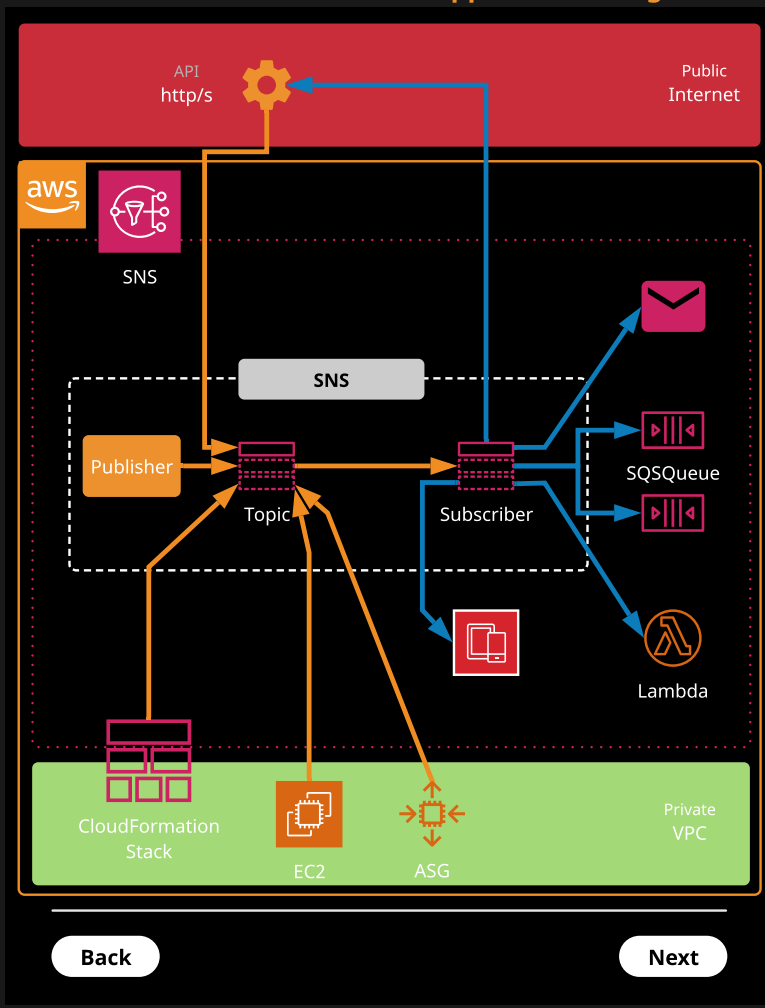
Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

- Application Integration
- Analytics
- Logging and Monitoring
- Operations
- Deployment



Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Application Integration

Analytics

Logging and Monitoring

Operations

Deployment

SNS Essentials

- SNS coordinates and manages the sending and delivery of messages. Messages sent to a topic are delivered to subscribers.
- SNS is integrated with many AWS services and can be used for certain event notifications (e.g., CloudFormation stack creation).
- Using SNS, CloudWatch can notify admins of important alerts.
- SNS can be used for mobile push notifications.

SNS Components

- **Topic**
  - An isolated configuration for SNS, including permissions
    - Messages (**<= 256 KB**) are sent to a topic
    - Subscribers to that topic receive messages
- **Subscriber**
  - Endpoints that receive messages for a topic
    - HTTP(S)
    - Email and Email-JSON
    - SQS (message can be added to one or more queues)
    - Mobile push notifications (iOS, Android, Amazon, MS)
    - Lambda functions (function invoked)
    - SMS (cellular message)
- **Publisher**
  - An entity that publishes/sends messages to queues
    - Application
    - AWS services, including S3 (S3 events), CloudWatch, CloudFormation, etc.

Go to Part 1

Back to Main

Application, Analytics, and Operations

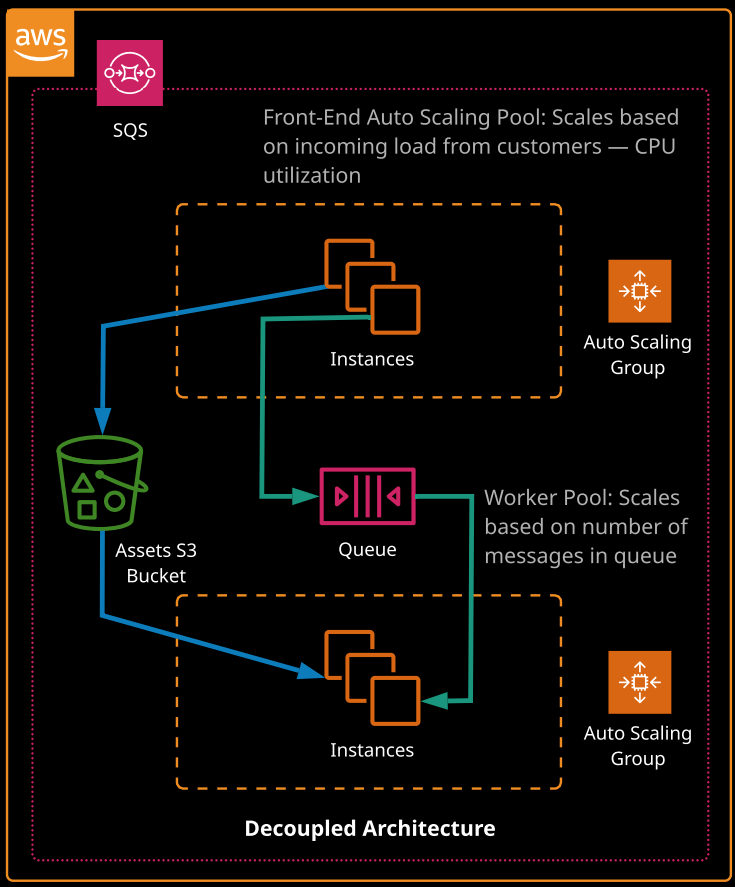
Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

- Application Integration
- Analytics
- Logging and Monitoring
- Operations
- Deployment



Back

Next

Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Application Integration

Analytics

Logging and Monitoring

Operations

Deployment

SQS Essentials

- Simple Queue Service (SQS) provides fully managed, highly available message queues for inter-process/server/service messaging.
- SQS is used mainly to create decoupled architectures.
- Messages are added to a queue and retrieved via *polling*.

Polling Types

- Short polling: Available messages are returned ASAP - a short poll might return 0 messages. Causes increased number of API calls.
- Long polling: Waits for messages for a given *WaitTimeSeconds*
- More Efficient: Less empty API calls/responses

There are two types of queues: **standard queues** and **FIFO queues**.

Each SQS message can contain up to 256 KB of data but can link data stored in S3 for any larger payloads.

When a message is polled, it is hidden in the queue. It can be deleted when processing is completed — otherwise, after a *VisibilityTimeout* period, it will return to the queue.

Queues can be configured with a *maxReceiveCount,* allowing messages that are failing to be moved to a dead-letter queue.

Lambda functions can be invoked based on messages on a queue offering better scaling and faster response than Auto Scaling groups for any messages that can be processed quickly.

Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

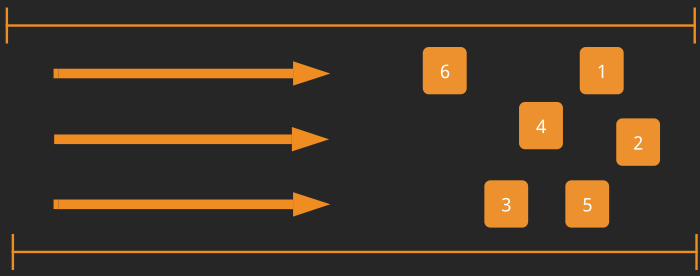
Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Application Integration
Analytics
Logging and Monitoring
Operations
Deployment

Standard vs. FIFO Queues

Standard queues are distributed and scalable to nearly unlimited message volume. The order is not guaranteed, best-effort only, and messages are guaranteed to be delivered at least once but sometimes more than once.



FIFO queues ensure **first-in, first-out** delivery. Messages are delivered once only — duplicates do not occur. The throughput is limited to ~3,000 messages per second with batching or ~300 without by default.

Application, Analytics, and Operations

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Application Integration

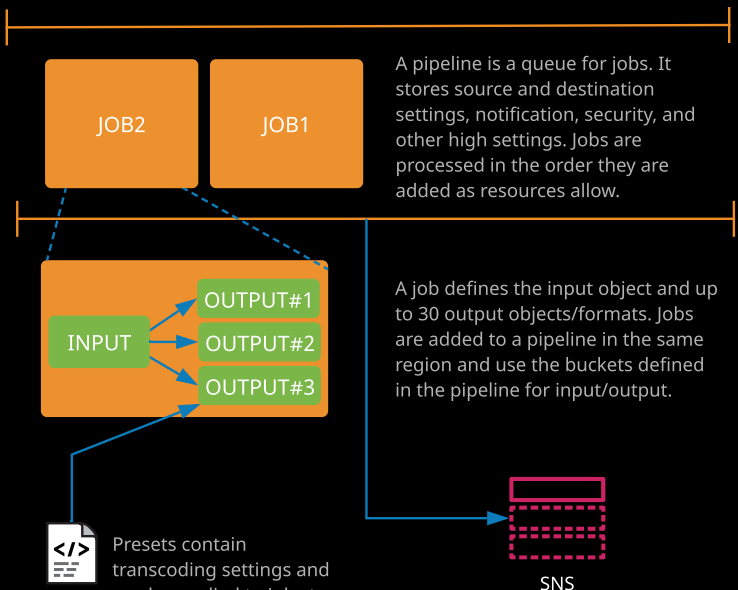
Analytics

Logging and Monitoring

Operations

Deployment

Elastic Transcoder is an AWS service that allows you to convert media files from an input format to one or more output formats. It's delivered as a service, and you are billed a per-minute charge while using the service.



Back

Next Topic

Go to Part 1

Back to Main

Application, Analytics, and Operations

Course Navigation

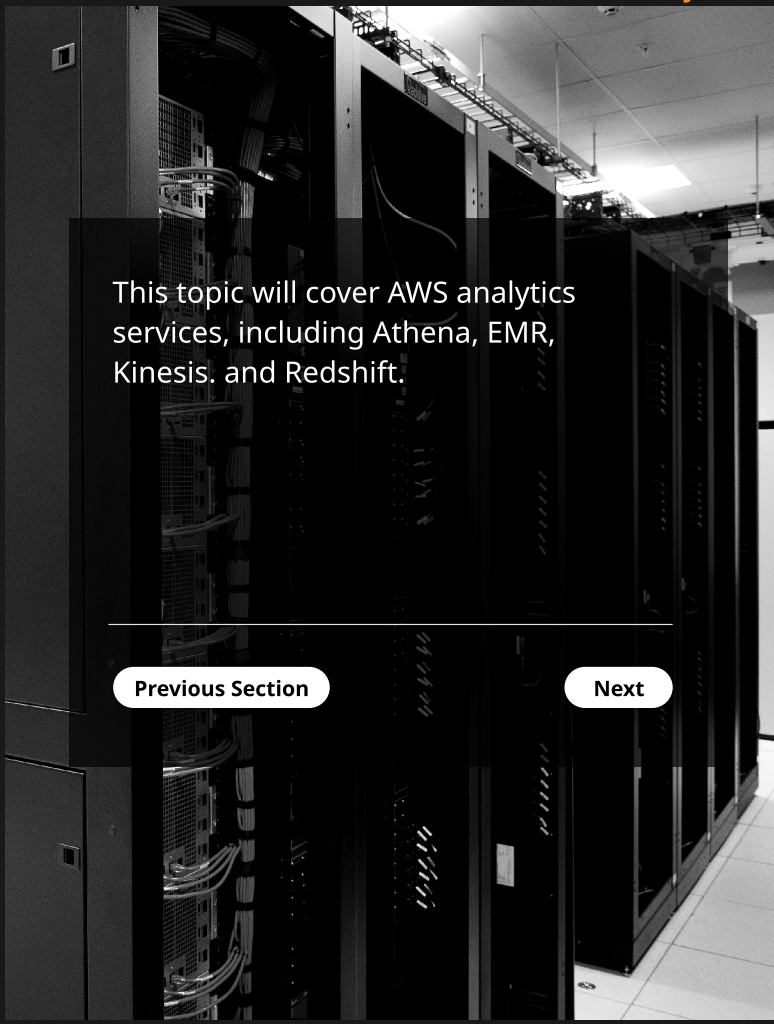
Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

- Application Integration
- Analytics
- Logging and Monitoring
- Operations
- Deployment



This topic will cover AWS analytics services, including Athena, EMR, Kinesis. and Redshift.

Previous Section

Next

Go to Part 1

Back to Main

Application, Analytics, and Operations

Storage and Content Delivery
Section 5

Databases
Section 6

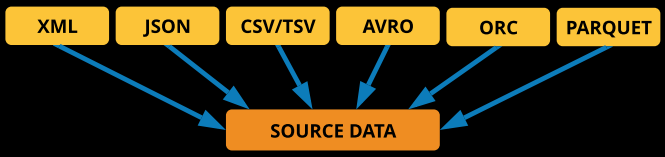
Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

- Application Integration
- Analytics
- Logging and Monitoring
- Operations
- Deployment

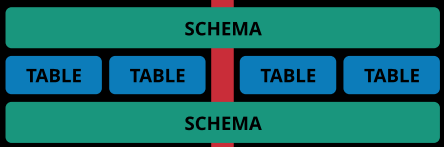
Analytics

Amazon Athena is an interactive query service that utilizes **schema-on-read**, allowing you to run ad-hoc SQL-like queries on data from a range of sources. Results are returned in seconds, and you are billed only for the compute time used and any existing storage costs.



Athena can query many forms of structured, semi-structured, and unstructured data in S3.

Athena can be used to query various AWS logs, including flow logs and ELB logs.



Tables are defined in a data catalog and are applied on read. Athena allows SQL queries against data stored on S3, through the schema-on-read tables.

AmazonAthena

No data is modified by Athena, and output can be sent to visualization tools.

Application, Analytics, and Operations

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

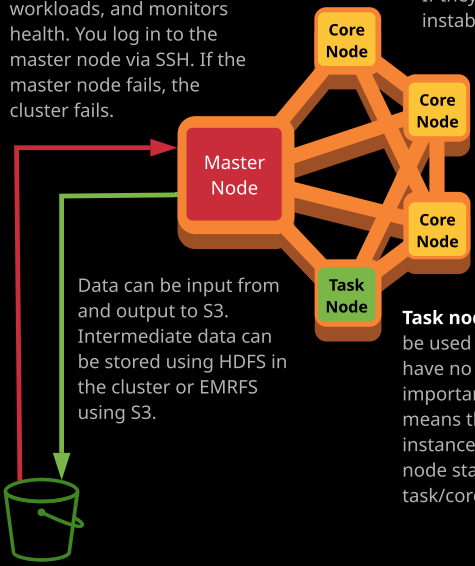
Application, Analytics, and Operations
Section 8

- Application Integration
- Analytics
- Logging and Monitoring
- Operations
- Deployment

Amazon **Elastic MapReduce** (**EMR**) is a tool for large-scale parallel processing of big data and other large data workloads. It's based on the Apache Hadoop framework and is delivered as a managed cluster using EC2 instances. EMR is used for huge-scale log analysis, indexing, machine learning, financial analysis, simulations, bioinformatics, and many other large-scale applications.

The **master node** manages the cluster. It manages HDFS naming, distributes workloads, and monitors health. You log in to the master node via SSH. If the master node fails, the cluster fails.

EMR clusters have zero or more **core nodes**, which are managed by the master node. They run tasks and manage data for HDFS. If they fail, it can cause cluster instability.



Data can be input from and output to S3. Intermediate data can be stored using HDFS in the cluster or EMRFS using S3.

Task nodes are optional. They can be used to execute tasks, but they have no involvement with important cluster functions, which means they can be used with spot instances. If task nodes fail, a core node starts the task on another task/core node.

Application, Analytics, and Operations

Storage and Content Delivery
Section 5

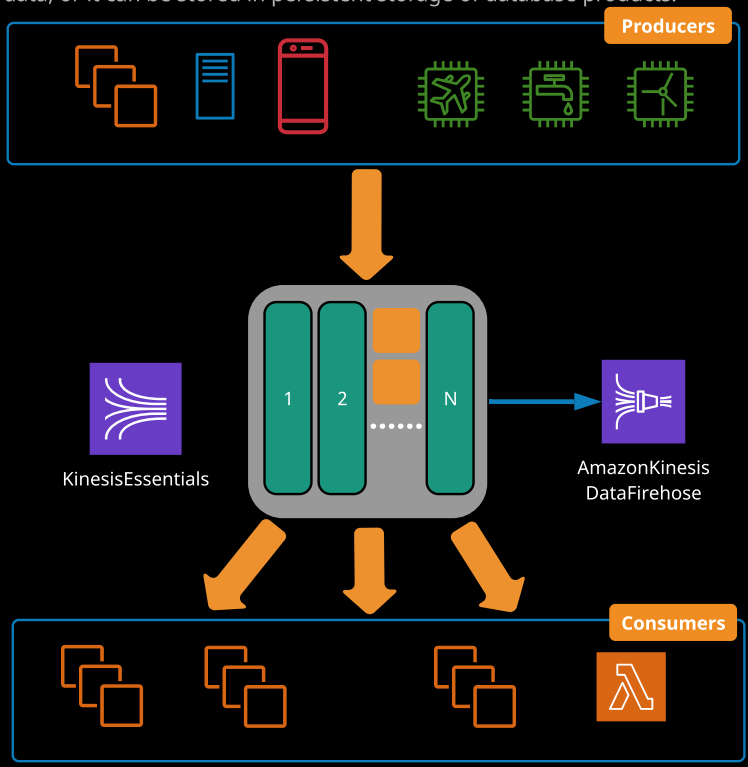
Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

- Application Integration
- Analytics
- Logging and Monitoring
- Operations
- Deployment

Kinesis is a scalable and resilient streaming service from AWS. It is designed to ingest large amounts of data from hundreds, thousands, or even millions of producers. Consumers can access a rolling window of that data, or it can be stored in persistent storage of database products.



Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Application Integration

Analytics

Logging and Monitoring

Operations

Deployment

Kinesis Essentials

Kinesis Stream

A Kinesis stream can be used to collect, process, and analyze a large amount of incoming data. A stream is a public service accessible from inside VPCs or from the public internet by an unlimited number of producers.

Kinesis streams include storage for all incoming data with a 24-hour default window, which can be increased to seven days for an additional charge. Data records are added by producers and read by consumers.

Kinesis Shard

Kinesis shards are added to streams to allow them to scale. A stream starts with at least one shard, which allows 1 MiB of ingestion and 2 MiB of consumption. Shards can be added or removed from streams.

Kinesis Data Record

The basic entity written to and read from Kinesis streams, a data record can be up to 1 MB in size.

You would use Kinesis rather than SQS when you need many producers and many consumers as well as a rolling window of data. SQS is a queue; Kinesis allows lots of independent consumers reading the same data window.

Application, Analytics, and Operations

Storage and Content Delivery
Section 5

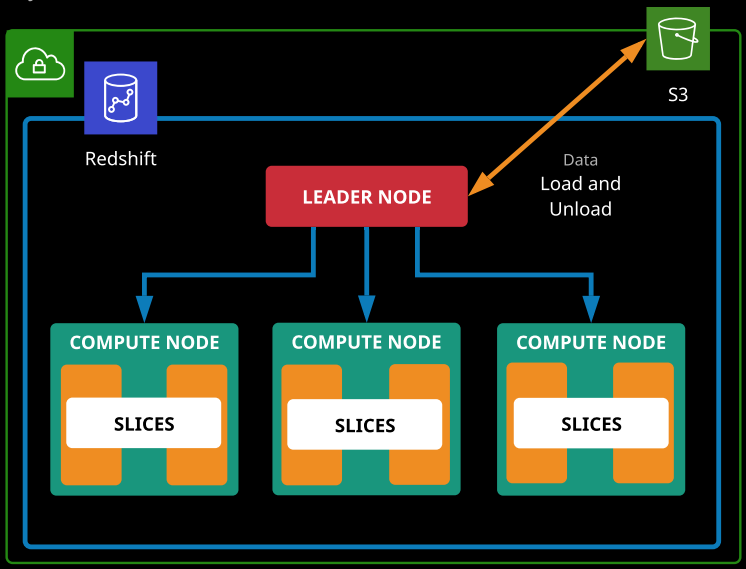
Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

- Application Integration
- Analytics
- Logging and Monitoring
- Operations
- Deployment

Redshift is a petabyte-scale data warehousing solution. It's a column-based database designed for analytical workloads. Generally, a relational store like RDS would be used for OLTP workloads (e.g., queries, inserts, updates, and deletes), and Redshift would be used for OLAP (e.g., retrieval and analytics). Multiple databases become source data to be injected into a data warehouse solution such as Redshift.



Data can be loaded **from** S3 and unloaded **to** S3. Additionally, backups can be performed to S3, and various AWS services such as Kinesis can inject data into Redshift.

Application, Analytics, and Operations

Course Navigation

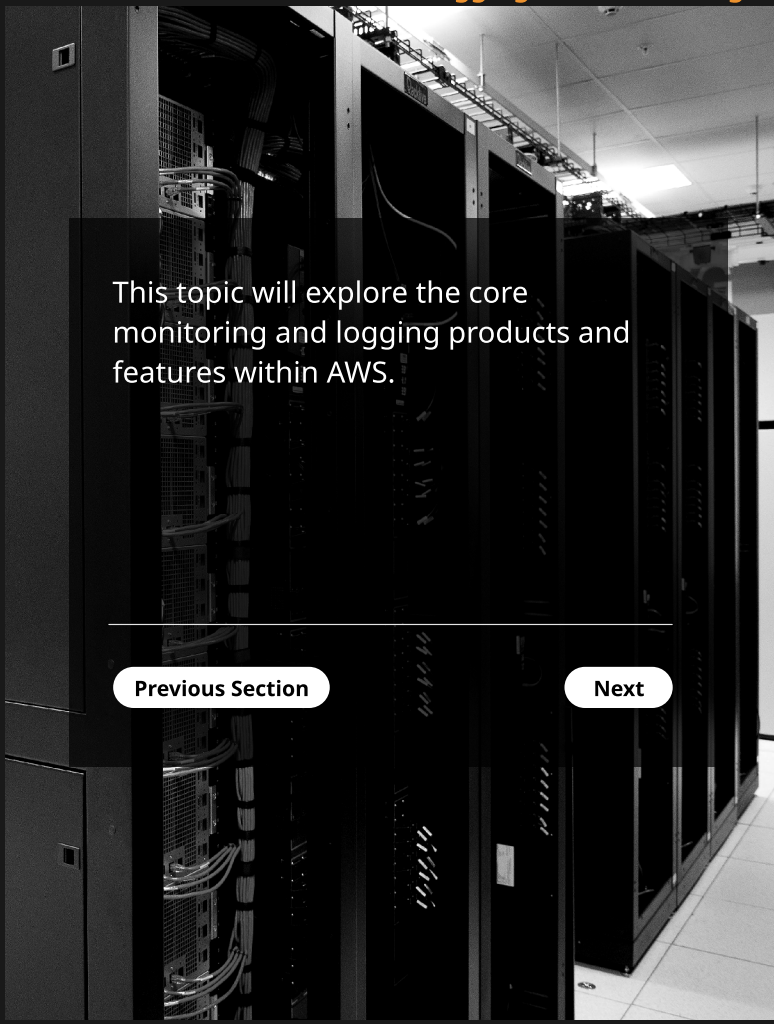
Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

- Application Integration
- Analytics
- Logging and Monitoring
- Operations
- Deployment



This topic will explore the core monitoring and logging products and features within AWS.

Previous Section

Next

Go to Part 1

Back to Main

Application, Analytics, and Operations

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

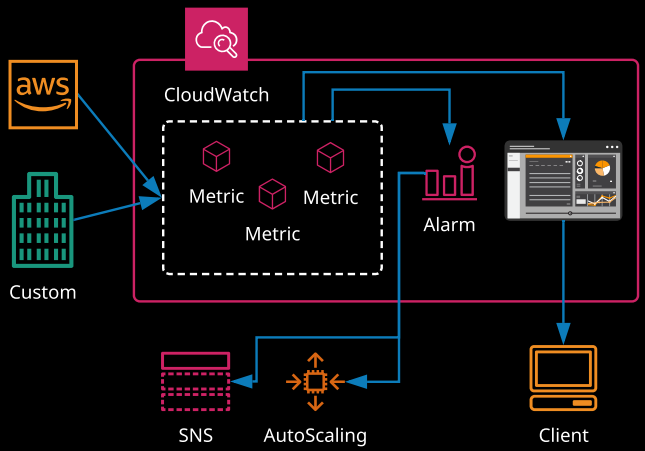
- Application Integration
- Analytics
- Logging and Monitoring
- Operations
- Deployment

CloudWatch is a service that provides near real-time monitoring of AWS products. In essence, it's a metrics repository. You can import custom metric data in real-time from some AWS services and on-premises platforms.

Data retention is based on granularity:

- One-hour metrics are retained for 455 days
- Five-minute metrics for 63 days
- One-minute metrics for 15 days

Metrics can be configured with alarms that can take actions, and data can be presented as a dashboard.



Course Navigation

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Application Integration
Analytics
Logging and Monitoring
Operations
Deployment

CloudWatch Metrics and Alarms

A CloudWatch metric is a set of data points over time. An example is CPU utilization of an EC2 instance.

Alarms can be created on metrics, taking an action if the alarm is triggered.

Alarms have three states:

- **INSUFFICIENT**: Not enough data to judge the state — alarms are often start in this state.
- **ALARM**: The alarm threshold has been breached (e.g., > 90% CPU).
- **OK**: The threshold has not been breached.

Alarms have a number of key components:

- **Metric**: The data points over time being measured
- **Threshold**: Exceeding this is bad (static or anomaly)
- **Period**: How long the threshold should be bad before an alarm is generated
- **Action**: What to do when an alarm triggers
  - SNS
  - Auto Scaling
  - EC2

Go to Part 1
Back to Main

Application, Analytics, and Operations

Storage and Content Delivery
Section 5

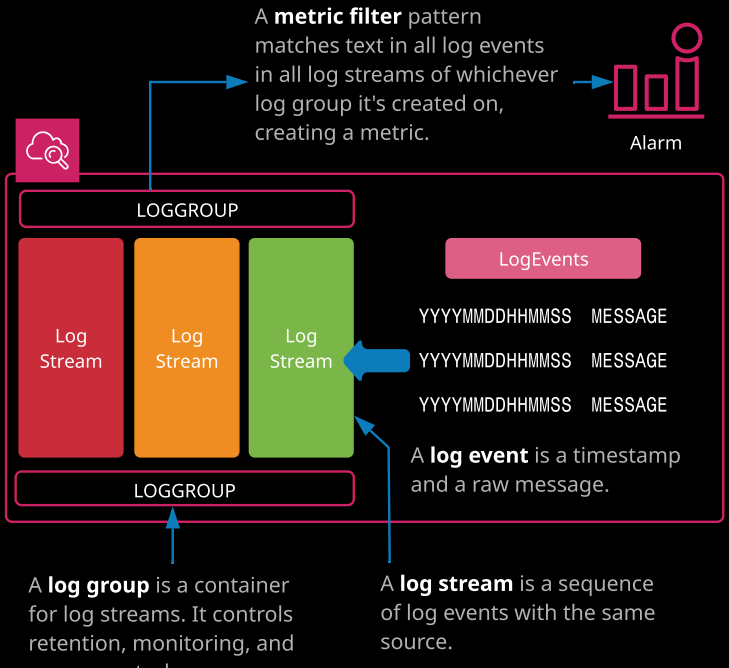
Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

- Application Integration
- Analytics
- Logging and Monitoring
- Operations
- Deployment

CloudWatch Logs provides functionality to store, monitor, and access logs from EC2, on-premises servers, Lambda, CloudTrail, Route 53, VPC Flow Logs, custom applications, and much more. Metric filters can be used to analyze logs and create metrics (e.g., failed SSH logins).



Application, Analytics, and Operations

Logging and Monitoring

Course Navigation

Storage and Content Delivery
Section 5

Databases
Section 6

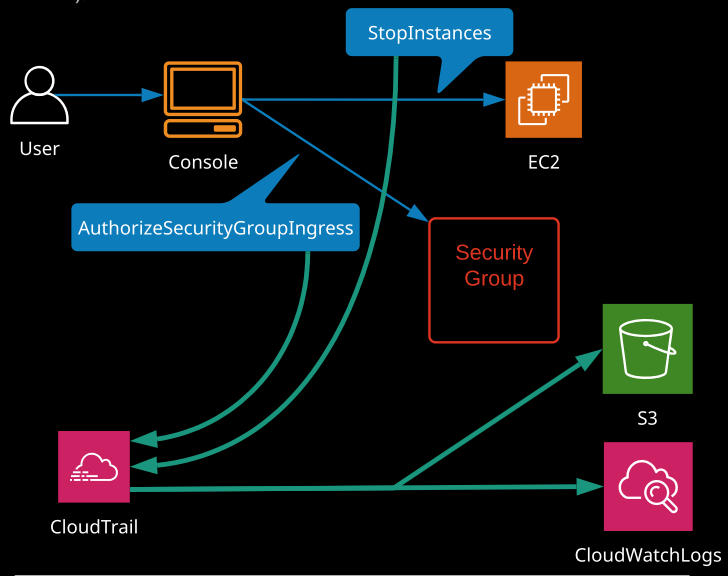
Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

- **Application Integration**
- **Analytics**
- **Logging and Monitoring**
- Operations
- Deployment

CloudTrail is a governance, compliance, risk management, and auditing service that records account activity within an AWS account. Any actions taken by users, roles, or AWS services are recorded to the service. Activity is recorded as a CloudTrail event, and by default you can view 90 days via event history. Trails can be created, giving more control over logging and allowing events to be stored in S3 and CloudWatch Logs.

Events can be **management events** that log control plane events (e.g., user login, configuring security, and adjusting security groups) or **data events** (e.g., object-level events in S3 or function-level events in Lambda).



Back
Next

Go to Part 1
Back to Main

Application, Analytics, and Operations

Storage and Content Delivery
Section 5

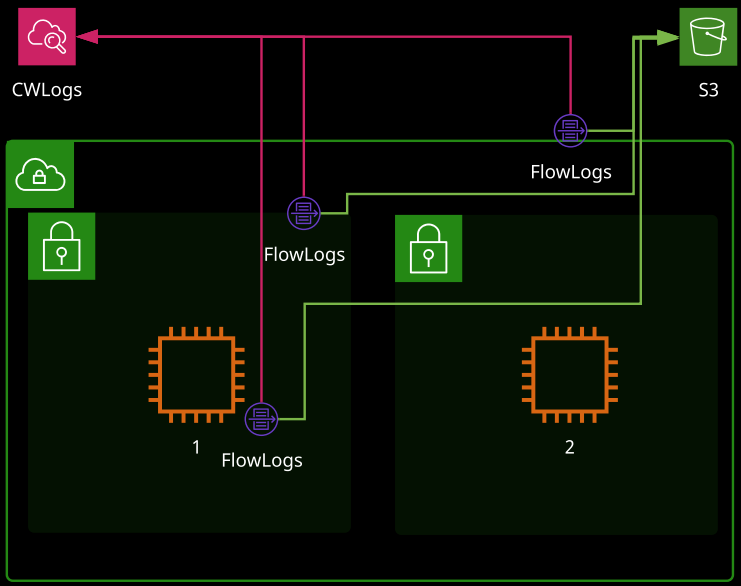
Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

- Application Integration
- Analytics
- Logging and Monitoring
- Operations
- Deployment

VPC Flow Logs allows you to capture metadata about the traffic flowing in and out of networking interfaces within a VPC. Flow logs can be placed on a specific network interface, a subnet, or an entire VPC and will capture metadata from the capture point and anything within it. Flow logs aren't real-time and don't capture the actual traffic — only metadata on the traffic.



Flow logs capture `account-id`, `interface-id`, `srcaddr`, `dstaddr`, `srcport`, `dstport`, `protocol`, `packets`, `bytes`, `start`, `end`, `action`, and `log-status`.

Flow logs don't capture some traffic, including Amazon DNS server, Windows license activation, 169.254.169.254, DHCP traffic, and VPC router.

Application, Analytics, and Operations

Course Navigation

Storage and Content Delivery
Section 5

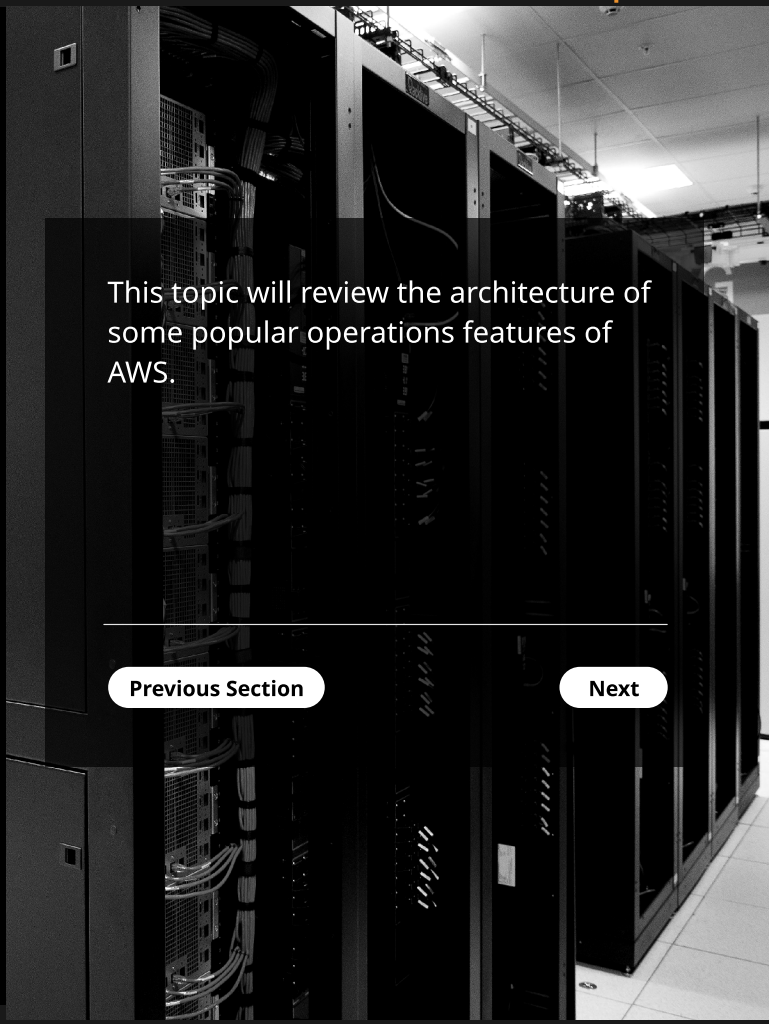
Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

- Application Integration
- Analytics
- Logging and Monitoring
- Operations

Deployment



This topic will review the architecture of some popular operations features of AWS.

Previous Section

Next

Go to Part 1

Back to Main

Application, Analytics, and Operations

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

- Application Integration
- Analytics
- Logging and Monitoring
- Operations

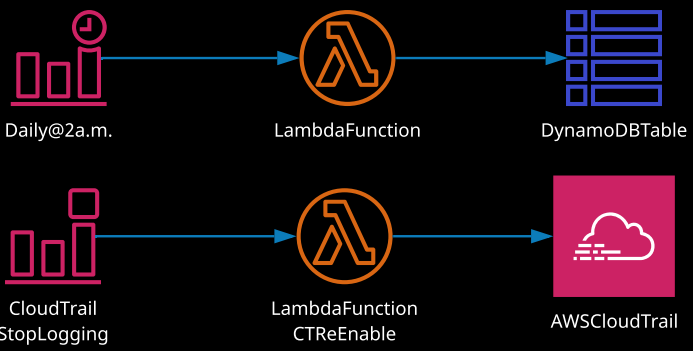
Deployment

CloudWatch Events has a near real-time visibility of changes that happen within an AWS account. Using rules, you can match against certain events within an account and deliver those events to a number of supported targets.

Within rules, many AWS services are natively supported as event sources and deliver the events directly. For others, CloudWatch allows event pattern matching against CloudTrail events. Additional rules support scheduled events as sources, allowing a cron-style function for periodically passing events to targets.

Some examples of event targets include:

- EC2 instances
- Lambda functions
- Step Functions state machines
- SNS topics
- SQS queues



Application, Analytics, and Operations

Operations

AWS **Key Management Service** (**KMS**) provides regional, secure key management and encryption and decryption services. KMS is **FIPS 140-2 level 2 validated**, and certain aspects support level 3 (*exam hint*). Everything in KMS is regional. KMS can use CloudHSM via Custom Key Stores (FIPS 140-2 Level 3)

KMS manages customer master keys (**CMK**), which are created in a region and never leave the region or KMS. They can encrypt or decrypt data **up to 4 KB**. CMKs have key policies and can be used to create other keys.

CMK

- KMS can **encrypt** data up to 4 KB with a CMK — you supply the data and specify the key to use.
- It can **decrypt** data up to 4 KB — you provide the ciphertext, and it returns the plaintext.
- You can also **re-encrypt** up to 4 KB — you supply the ciphertext, the new key to use, and you are returned new ciphertext (at no point do you see the plaintext).

EncryptedData EncryptionKey

DataEncryptionKey (DEK)

KMS can generate a **data encryption key** (**DEK**) using a CMK. You or a service can use a **DEK** to encrypt or decrypt data of any size. KMS supplies a plaintext version and an encrypted version.

The encrypted DEK and encrypted data can be stored together. KMS is used to decrypt the DEK, which can then decrypt the data.

EncryptedData EncryptionKey

EncryptedData

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Application Integration
Analytics
Logging and Monitoring
Operations

Deployment

Customer Master Keys (CMK)

There are three types of CMK:

Type	CanView	CanManage	DedicatedtoMyAccount
CustomerManaged	Yes	Yes	Yes
AWSManagedCMK	Yes	No	Yes
AWSOwnedCMK	No	No	No

AWS Managed CMK: Used by default if you pick encryption within most AWS services and formatted as *aws/service-name*. Only the service they belong to can use them directly.

Customer Managed CMK: Certain services allow you to pick a CMK you manage. Customer managed CMKs allow key rotation configuration, and they can be controlled via key policies and enabled/disabled.

AWS Owned CMK: Keys used by AWS on a shared basis across many accounts — you normally don't see these.

Application, Analytics, and Operations

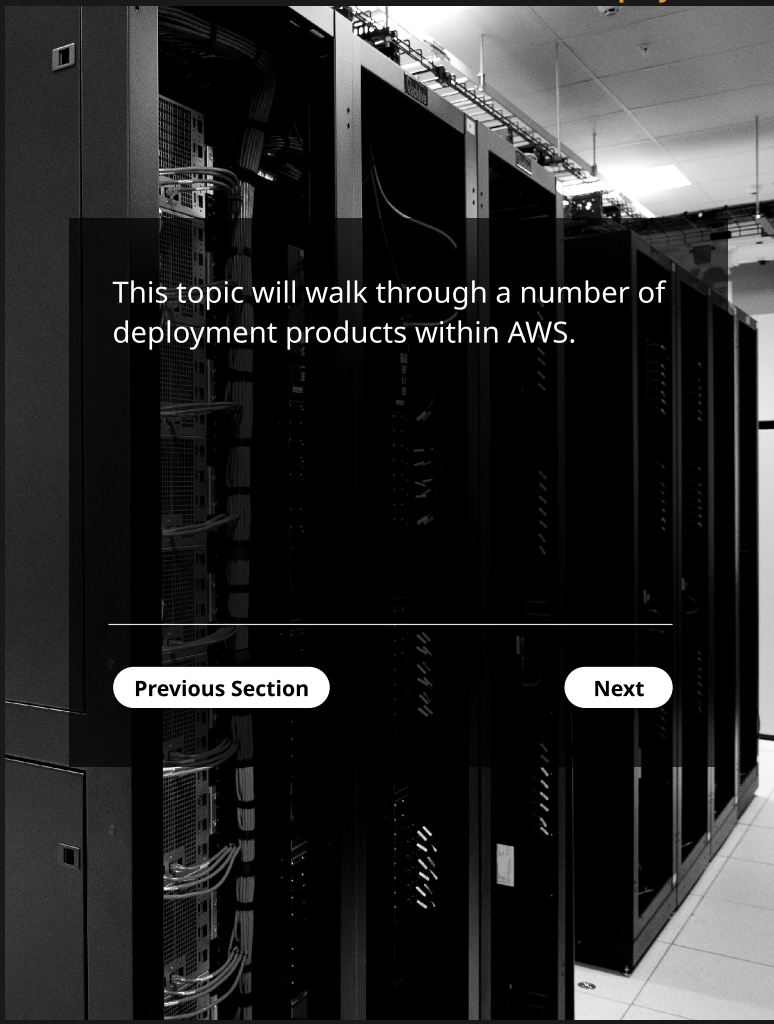
Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Application Integration
Analytics
Logging and Monitoring
Operations
Deployment



This topic will walk through a number of deployment products within AWS.

Previous Section

Next

Go to Part 1

Back to Main

Application, Analytics, and Operations

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Application Integration
Analytics
Logging and Monitoring
Operations
Deployment

CloudFormation is an Infrastructure as Code (**IaC**) product — you can **create**, **manage**, and **remove** infrastructure using **JSON** or **YAML**.

1

Template

A CFN template is JSON or YAML. It contains **logical resources** and configuration.

2

Stack

Stacks are created and modified based on templates, which can be changed and used to update a stack.

3

Physical Resources

Stacks take **logical resources** from a template and create, update, or delete the **physical resources** in AWS.

CloudFormation is effective if you **frequently deploy** the same infrastructure or you require **guaranteed consistent configuration**.

Application, Analytics, and Operations

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Application Integration
Analytics
Logging and Monitoring
Operations
Deployment

Elastic Beanstalk (**EB**) is a Platform as a Service product. It allows you to deploy code and, with very little effort or modifications, the service will provision the infrastructure on your behalf.

Elastic Beanstalk handles provisioning, monitoring, Auto Scaling, load balancing, and software updating for you — you just worry about the cost.

Elastic Beanstalk supports a number of languages and platforms:

- Java
- .NET
- Node.js
- PHP
- Ruby
- Python
- Go
- Docker
- Apache
- IIS
- Nginx
- Tomcat

EB Architecture Overview

Patterns and Anti-Patterns for Elastic Beanstalk

- **YES**: To provision an environment for an application with little admin overhead
- **YES**: If you use one of the supported languages and can add EB-specific config
- **NO**: If you want low-level infrastructure control
- **NO**: If you need Chef support

Deployment Options

All at Once: An updated application version is deployed to all instances. Quick and simple but not recommended for production deployments.

Rolling: Splits instances into batches and deploys one batch at a time.

Rolling with additional Batch: As above, but provisions a new batch, deploying and testing before removing the old batch (immutable).

Blue/Green: Maintain two environments, deploy, and swap CNAME.

Storage and Content Delivery
Section 5

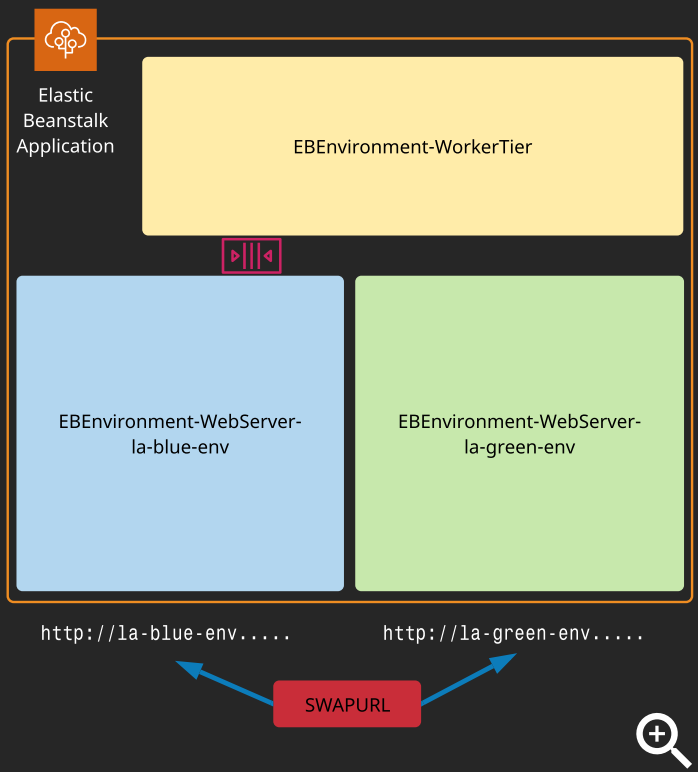
Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Application Integration
Analytics
Logging and Monitoring
Operations
Deployment

Elastic Beanstalk (High Level)



Go to Part 1
Back to Main

Application, Analytics, and Operations

Storage and Content Delivery
Section 5

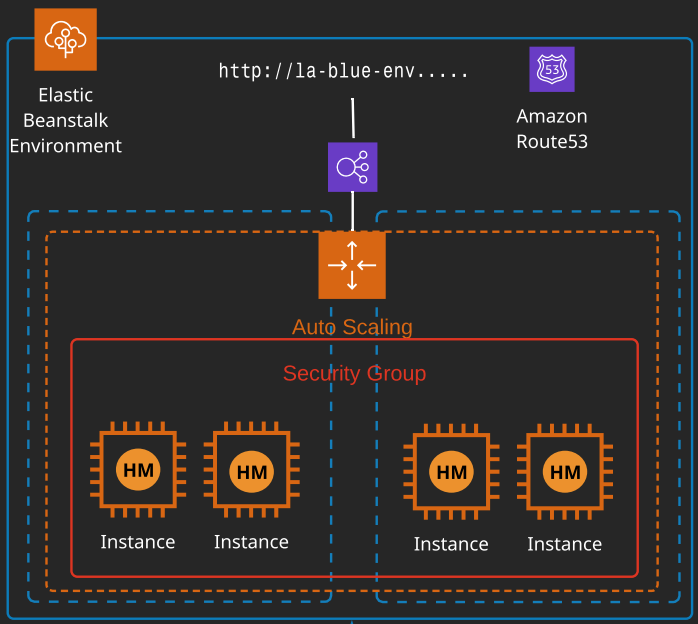
Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

- Application Integration
- Analytics
- Logging and Monitoring
- Operations
- Deployment

Elastic Beanstalk (Detailed)



Go to Part 1

Back to Main

Storage and Content Delivery
Section 5

Databases
Section 6

Hybrid and Scaling
Section 7

Application, Analytics, and Operations
Section 8

Application Integration
Analytics
Logging and Monitoring
Operations
Deployment

- **OpsWorks** is an implementation of the Chef configuration management and deployment platform.
- OpsWorks moves away from the low-level configurability of CloudFormation but not as far as Elastic Beanstalk.
- OpsWorks lets you create a stack of resources with layers and manage resources as a unit.

OpsWorks Components

- **Stacks**
  - A unit of managed infrastructure
  - Can use stacks per application or per platform
  - Could use stacks for development , staging, or production environments
- **Layers**
  - Comparable to application tiers within a stack
  - e.g., database layer, application layer, proxy layer
  - Recipes are generally associated with layers and configure what to install on instances in that layer
- **Instances**
  - Instances are EC2 instances associated with a layer
  - Configured as 24/7, load based, or time based
- **Apps**
  - Apps are deployed to layers from a source code repo or S3
  - Actual deployment happens using recipes on a layer.
  - Other recipes are run when deployments happen, potentially to reconfigure other instances
- **Recipes**
  - **Setup**: Executed on an instance when first provisioned
  - **Configure**: Executed on **all** instances when instances are added or removed
  - **Deploy** and **Undeploy**: Executed when apps are added or removed
  - **Shutdown**: Executed when an instance is shut down but before it's stopped

Exam Preparation

The AWS Certified Solutions Architect - Associate examination is intended for individuals who perform a solutions architect role and have one or more years of hands-on experience designing available, cost-effective, fault-tolerant, and scalable distributed systems on AWS.

Exam Overview:

- Multiple choice, multiple answer
- 130 minutes
- $150 USD

Domain	%ofExaminationResult
Domain1:DesignResilientArchitectures	34%
Domain2:DefinePerformantArchitectures	24%
Domain3:SpecifySecureApplicationsandArchitectures	26%
Domain4:DesignCost-OptimizedArchitectures	10%
Domain5:DefineOperationallyExcellentArchitectures	6%

Exam Preparation

The AWS Certified Solutions Architect - Associate examination is intended for individuals who perform a solutions architect role and have one or more years of hands-on experience designing available, cost-effective, fault-tolerant, and scalable distributed systems on AWS.

Exam Overview:

- Multiple choice, multiple answer
- 130 minutes
- $150 USD

Domain	%ofExaminationResult
Domain1:DesignResilientArchitectures	34%
Domain2:DefinePerformantArchitectures	24%
Domain3:SpecifySecureApplicationsandArchitectures	26%
Domain4:DesignCost-OptimizedArchitectures	10%
Domain5:DefineOperationallyExcellentArchitectures	6%

Domain 1: Design Resilient Architectures

- 1.1 Choose reliable/resilient storage.
- 1.2 Determine how to design decoupling mechanisms using AWS services.
- 1.3 Determine how to design a multi-tier architecture solution.
- 1.4 Determine how to design high availability and/or fault tolerant architectures.

Exam Preparation

The AWS Certified Solutions Architect - Associate examination is intended for individuals who perform a solutions architect role and have one or more years of hands-on experience designing available, cost-effective, fault-tolerant, and scalable distributed systems on AWS.

Exam Overview:

- Multiple choice, multiple answer
- 130 minutes
- $150 USD

Domain	%ofExaminationResult
Domain1:DesignResilientArchitectures	34%
Domain2:DefinePerformantArchitectures	24%
Domain3:SpecifySecureApplicationsandArchitectures	26%
Domain4:DesignCost-OptimizedArchitectures	10%
Domain5:DefineOperationallyExcellentArchitectures	6%

Domain 2: Define Performant Architectures

- 2.1 Choose performant storage and databases.
- 2.2 Apply caching to improve performance.
- 2.3 Design solutions for elasticity and scalability.

Exam Preparation

The AWS Certified Solutions Architect - Associate examination is intended for individuals who perform a solutions architect role and have one or more years of hands-on experience designing available, cost-effective, fault-tolerant, and scalable distributed systems on AWS.

Exam Overview:

- Multiple choice, multiple answer
- 130 minutes
- $150 USD

Domain	%ofExaminationResult
Domain1:DesignResilientArchitectures	34%
Domain2:DefinePerformantArchitectures	24%
Domain3:SpecifySecureApplicationsandArchitectures	26%
Domain4:DesignCost-OptimizedArchitectures	10%
Domain5:DefineOperationallyExcellentArchitectures	6%

Domain 3: Specify Secure Applications and Architectures

- 3.1 Determine how to secure application tiers.
- 3.2 Determine how to secure data.
- 3.3 Define the networking infrastructure for a single VPC application.

Exam Preparation

The AWS Certified Solutions Architect - Associate examination is intended for individuals who perform a solutions architect role and have one or more years of hands-on experience designing available, cost-effective, fault-tolerant, and scalable distributed systems on AWS.

Exam Overview:

- Multiple choice, multiple answer
- 130 minutes
- $150 USD

Domain	%ofExaminationResult
Domain1:DesignResilientArchitectures	34%
Domain2:DefinePerformantArchitectures	24%
Domain3:SpecifySecureApplicationsandArchitectures	26%
Domain4:DesignCost-OptimizedArchitectures	10%
Domain5:DefineOperationallyExcellentArchitectures	6%

Domain 4: Design Cost-Optimized Architectures

- 4.1 Determine how to design cost-optimized storage.
- 4.2 Determine how to design cost-optimized compute.

Exam Preparation

The AWS Certified Solutions Architect - Associate examination is intended for individuals who perform a solutions architect role and have one or more years of hands-on experience designing available, cost-effective, fault-tolerant, and scalable distributed systems on AWS.

Exam Overview:

- Multiple choice, multiple answer
- 130 minutes
- $150 USD

Domain	%ofExaminationResult
Domain1:DesignResilientArchitectures	34%
Domain2:DefinePerformantArchitectures	24%
Domain3:SpecifySecureApplicationsandArchitectures	26%
Domain4:DesignCost-OptimizedArchitectures	10%
Domain5:DefineOperationallyExcellentArchitectures	6%

Domain 5: Define Operationally Excellent Architectures

- 5.1 Choose design features in solutions that enable operational excellence.