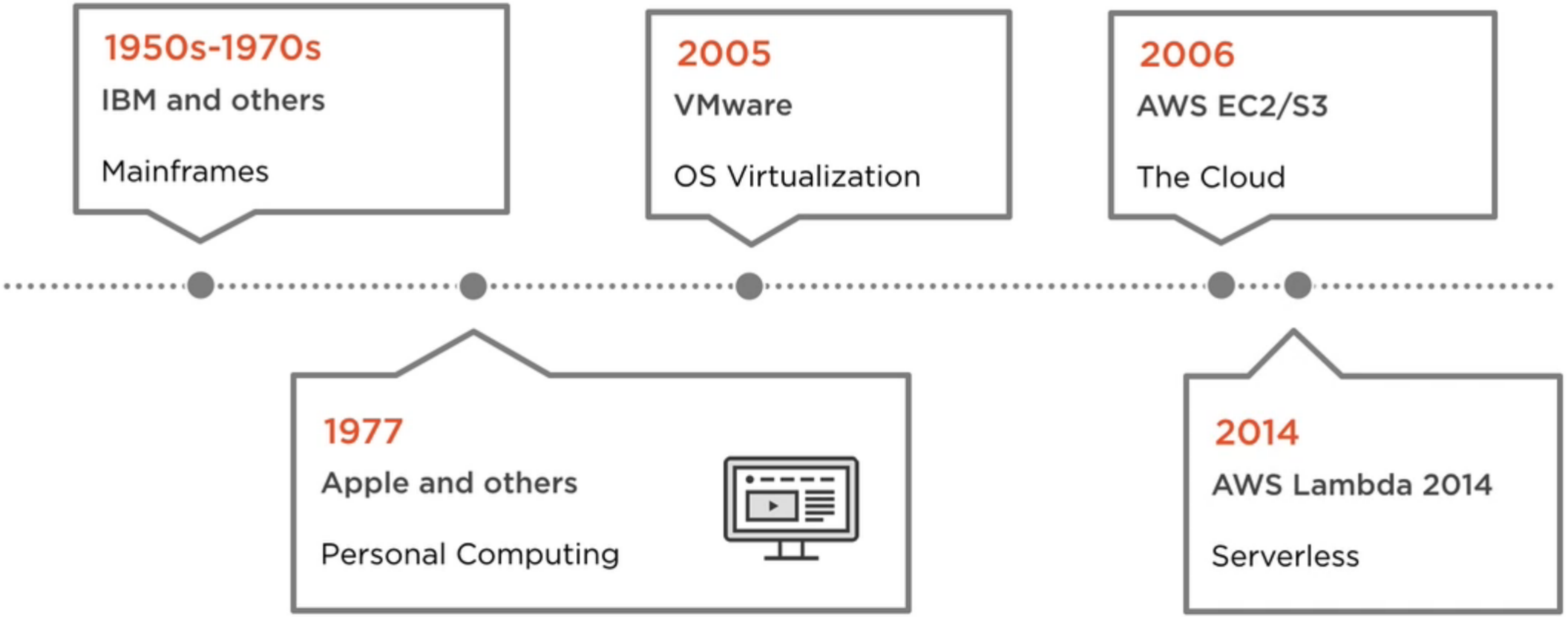

aws

AGENDA:

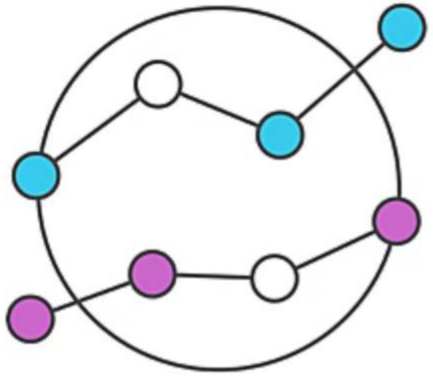
- Cloud Concepts
- Cloud Services
- Deployment Models
- Content Delivery and DNS Services
- Use Amazon Route 53 for DNS resolution and domain registration
- Use the Amazon CloudFront content delivery network (CDN)
- DevOps on AWS
  - Creating a Code Pipeline
  - Build a continuous integration and continuous delivery pipeline
  - Use AWS CodeCommit
  - AWS CodeBuild
  - AWS CodeDeploy
  - AWS CodePipeline and AWS CodeStar

What exactly is cloud computing?

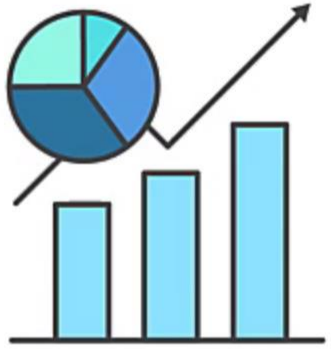
Cloud computing enables you to stop thinking of your infrastructure as hardware, and instead think of (and use) it as software.


1950s-1970s
IBM and others
Mainframes
1977
Apple and others
Personal Computing
2005
VMware
OS Virtualization
2006
AWS EC2/S3
The Cloud
2014
AWS Lambda 2014
Serverless

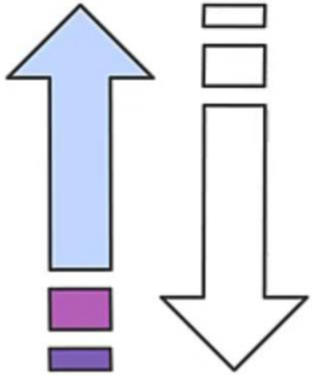
Benefits of cloud computing



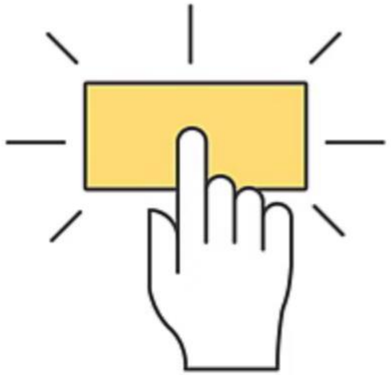
Trade capital expense for variable expense



Benefit from massive economies of scale



Stop guessing about capacity

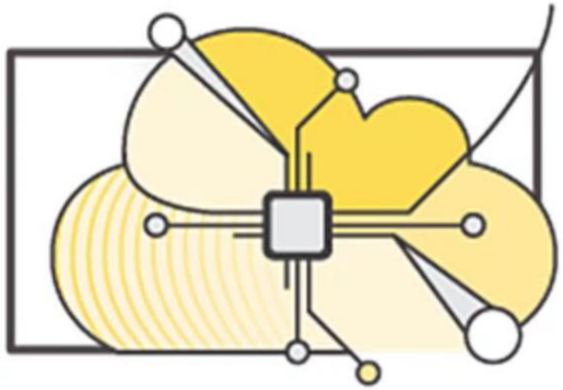


Increase speed and agility

Focus on what matters

Go global in minutes

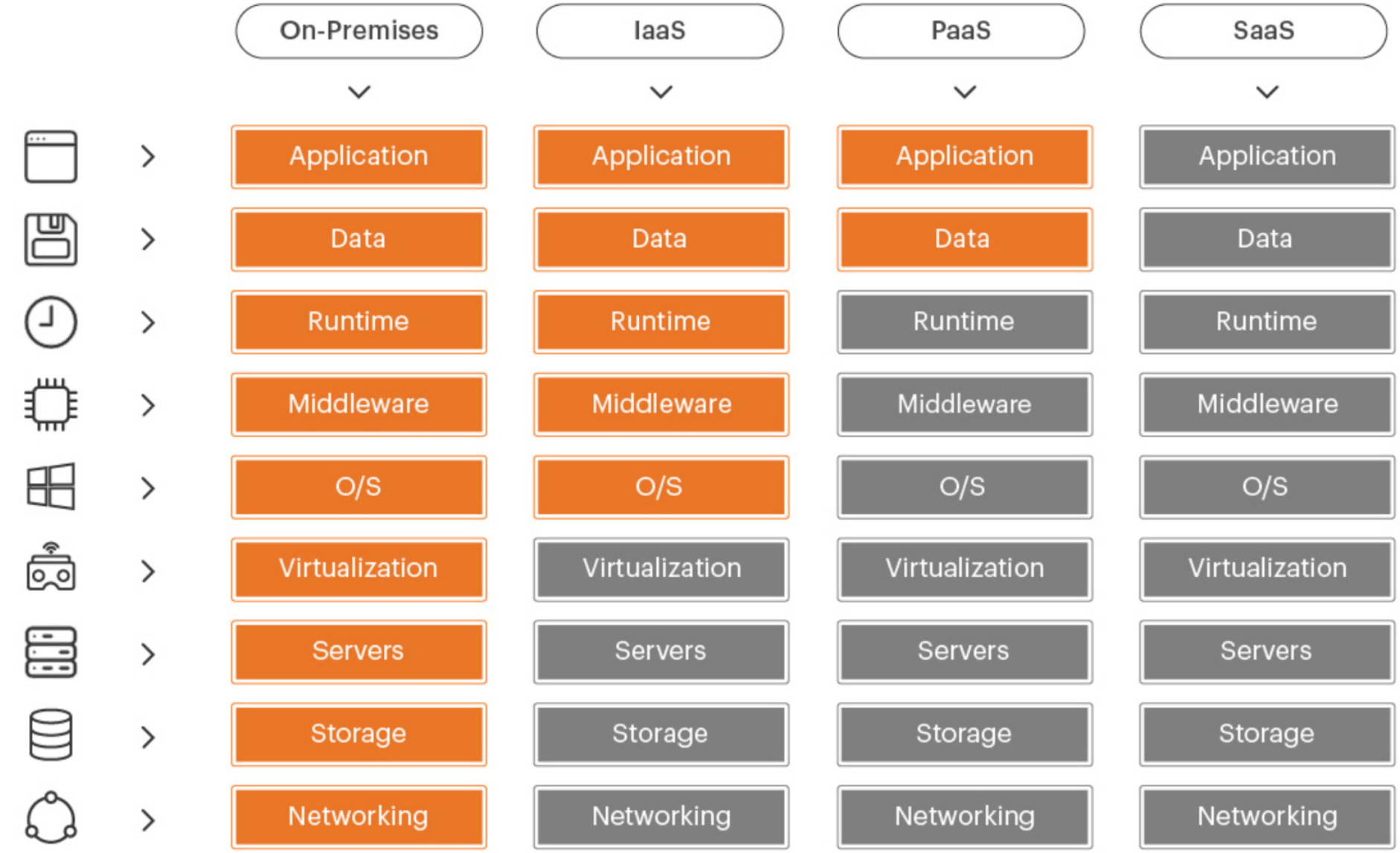
Cloud computing deployment models



Cloud

Hybrid

On-premises
(private cloud)



https://aws.amazon.com/types-of-cloud-computing/

What is AWS?

- AWS is a secure cloud platform that offers a broad set of global cloud-based products.
- AWS provides you with on-demand access to compute, storage, network, database, and other IT resources and management tools.
- AWS offers flexibility.
- You pay only for the individual services you need, for as long as you use them.
- AWS services work together like building blocks.

AWS foundational services

Applications

- Virtual desktops
- Collaboration and sharing

Platform Services

Databases	Analytics	Application services	Deployment and management	Mobile Services
Relational	Cluster computing	Queuing	Containers	Identity
NoSQL	Real-time	Orchestration	DevOps tools	Sync
Caching	Data warehouse	App Streaming	Resource templates	Mobile Analytics
	Data workflows	Transcoding	Usage tracking	Notifications
		Email	Monitoring and logs	
		Search		

Foundation Services


Compute (virtual, automatic scaling, and load balancing)


Networking


Storage (object, block, and archive)

Infrastructure

- Regions
- Availability Zones
- 
Edge locations

AWS categories of services


Analytics


Application Integration


AR and VR


Blockchain


Business Applications


Compute


Cost Management


Customer Engagement


Database


Developer Tools


End User Computing


Game Tech


Internet of Things


Machine Learning


Management and Governance


Media Services


Migration and Transfer


Mobile


Networking and Content Delivery


Robotics


Satellite


Security, Identity, and Compliance


Storage

Core AWS services

Amazon Virtual Private Cloud (Amazon VPC)

Amazon Elastic Compute Cloud (Amazon EC2)

Storage

AWS Identity and Access Management (IAM)

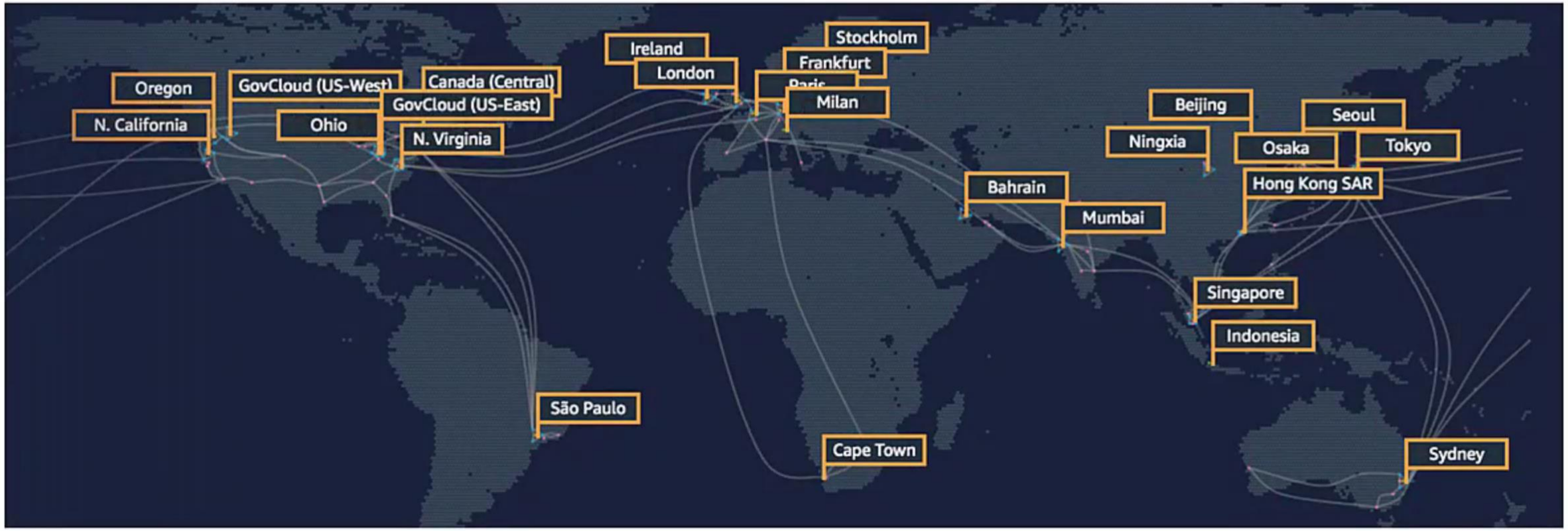
Amazon Relational Database Service

Amazon DynamoDB

Database

AWS Global Infrastructure

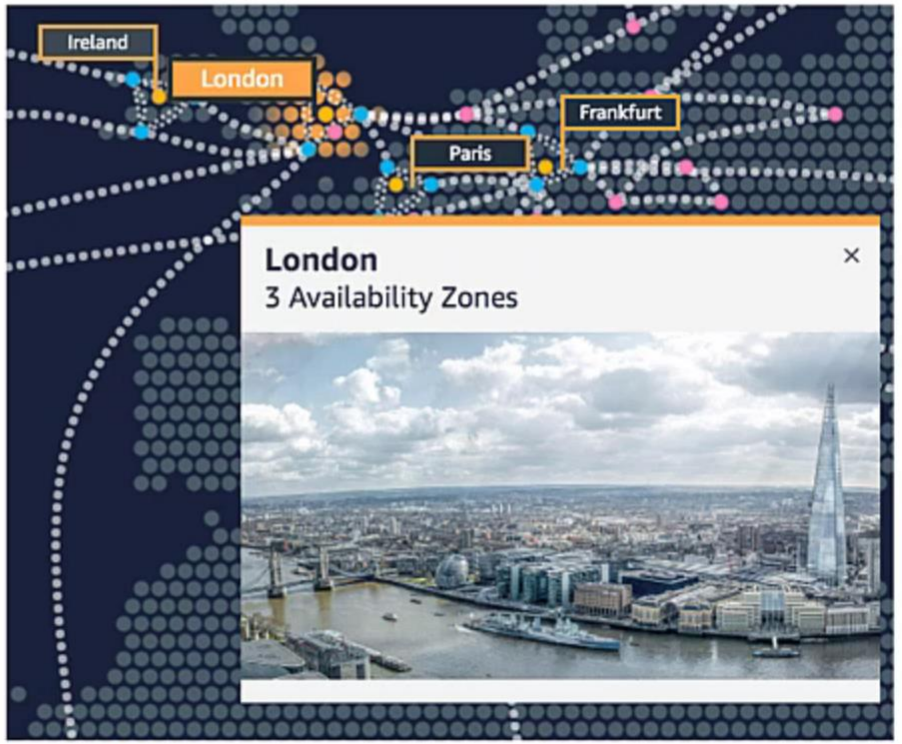
- The AWS Global Infrastructure is designed and built to deliver a flexible, reliable, scalable, and secure cloud computing environment with high-quality global network performance.
- This map shows the current AWS Regions and more that are coming soon.



AWS Regions

aws training and certification

- An AWS Region is a geographical area.
  - Data replication across Regions is controlled by you.
  - Communication between Regions uses AWS backbone network infrastructure.
- Each Region provides full redundancy and connectivity to the network.
- A Region typically consists of two or more Availability Zones.



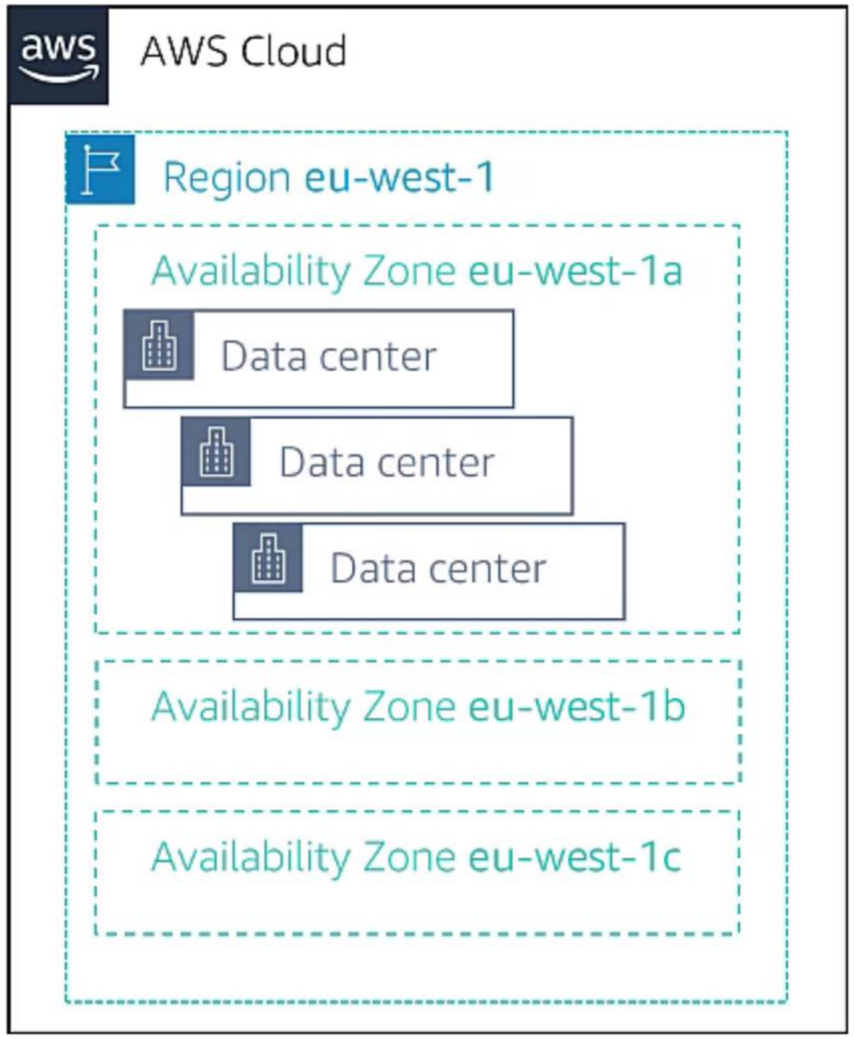
Example: London Region

Selecting a region

Determine the right region for your services, applications, and data based on these factors

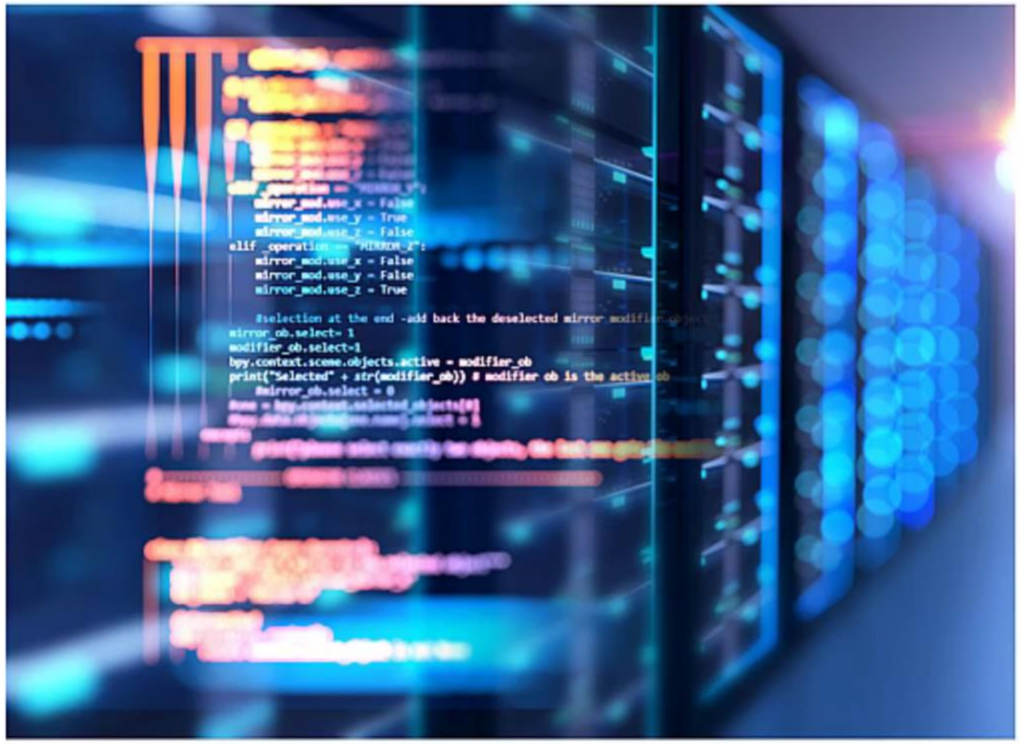
Availability Zones

- Each Region has multiple Availability Zones.
- Each Availability Zone is a fully isolated partition of the AWS infrastructure.
  - There are currently 69 Availability Zones worldwide
  - Availability Zones consist of discrete data centers
  - They are designed for fault isolation
  - They are interconnected with other Availability Zones by using high-speed private networking
  - You choose your Availability Zones.
  - AWS recommends replicating data and resources across Availability Zones for resiliency.

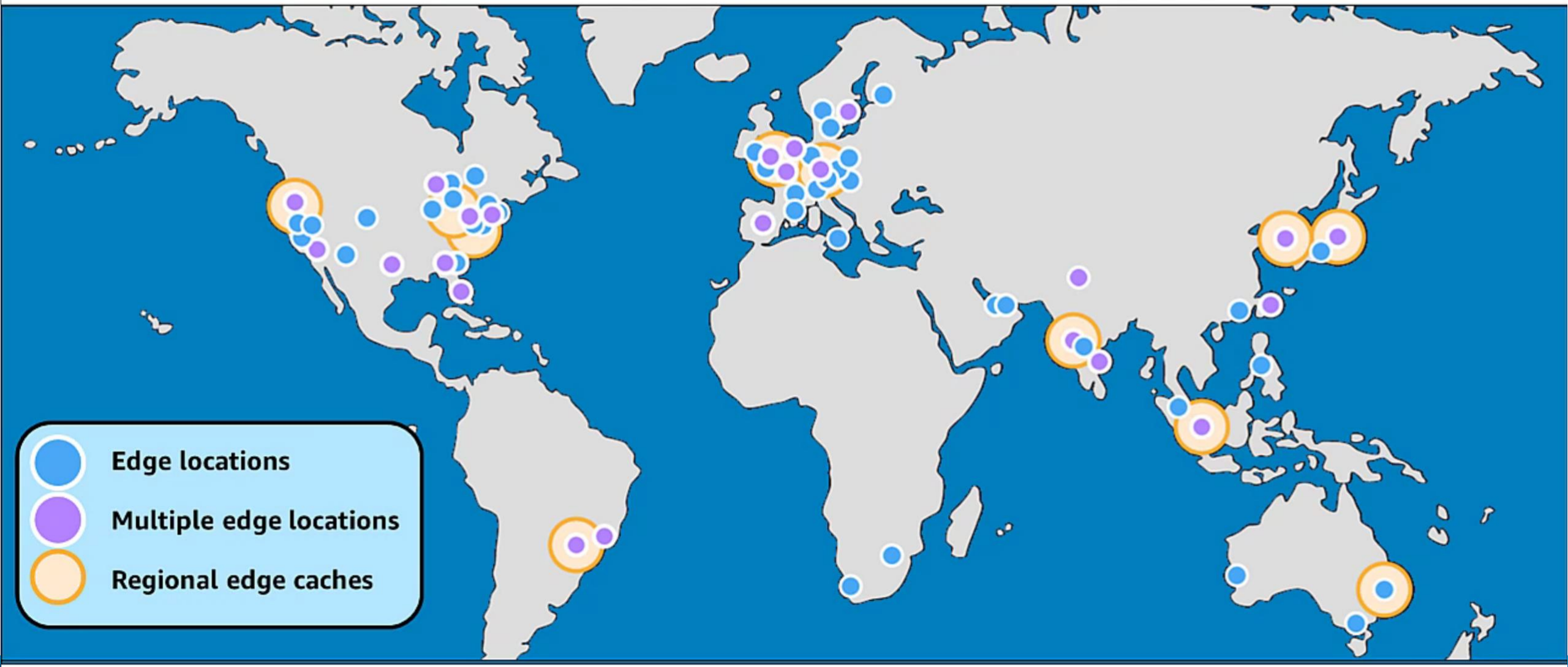


AWS data centers

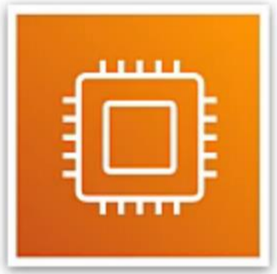
- AWS data centers are designed for security.
- Data centers are where the data resides and data processing occurs.
- Each data center has redundant power, networking, and connectivity, and is housed in a separate facility.
- A data center typically has 50,000 to 80,000 physical servers.



Edge locations: Reaching distant customers



Compute service category



AWS Compute services

Amazon EC2

Amazon EC2 Auto Scaling

Amazon Elastic Container Service (Amazon ECS)

Amazon EC2 Container Registry

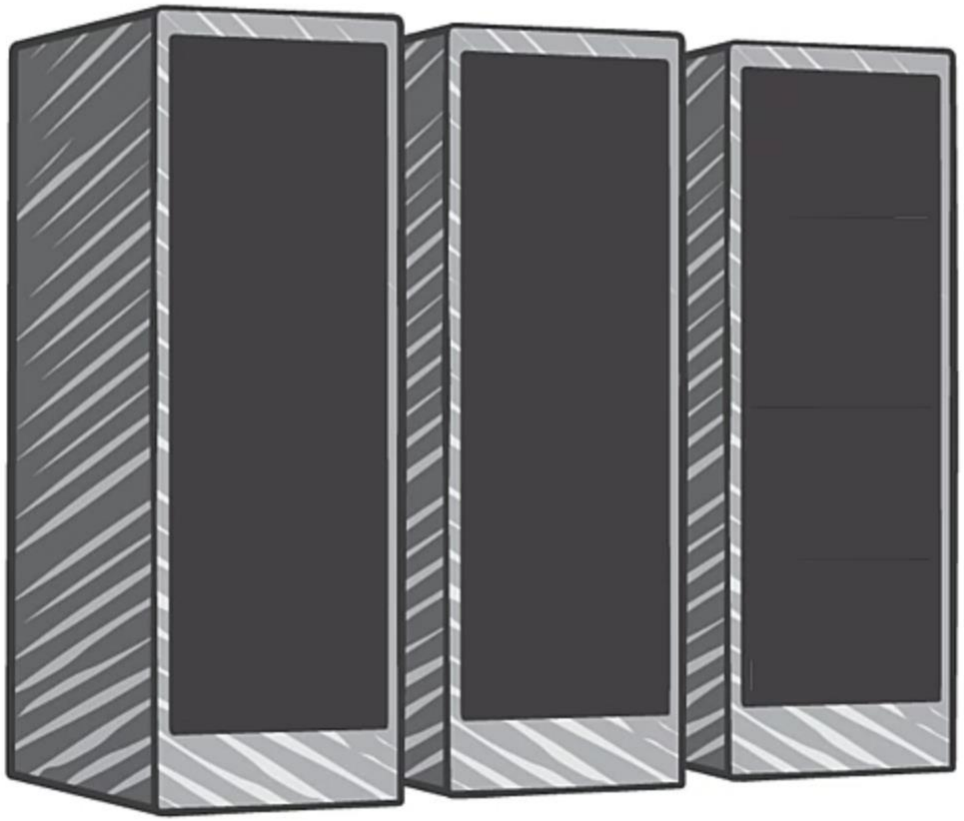
AWS Elastic Beanstalk

AWS Lambda

Amazon Elastic Kubernetes Service (Amazon EKS)

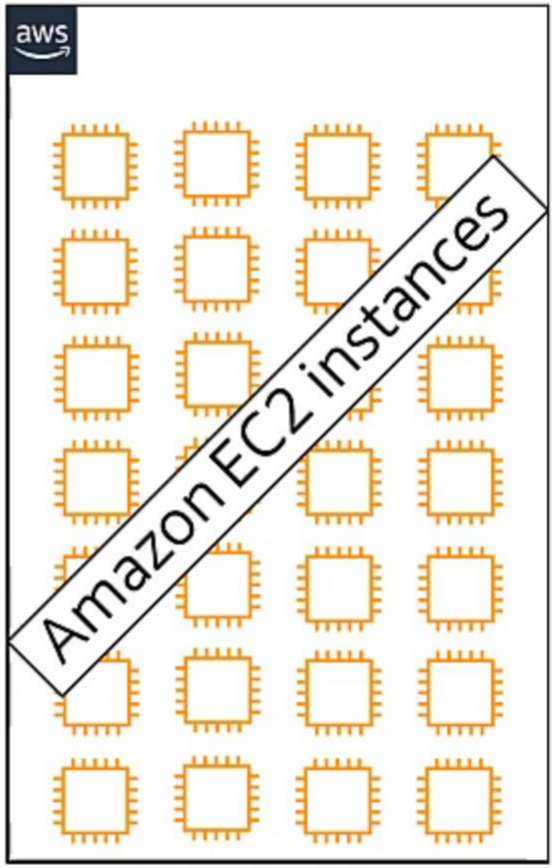
AWS Fargate

What is Amazon EC2?



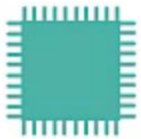
On-premises servers

- ✓ Application server
- ✓ Web server
- ✓ Database server
- ✓ Game server
- ✓ Mail server
- ✓ Media server
- ✓ Catalog server
- ✓ File server
- ✓ Computing server
- ✓ Proxy server



What's your use case?

Instance type details



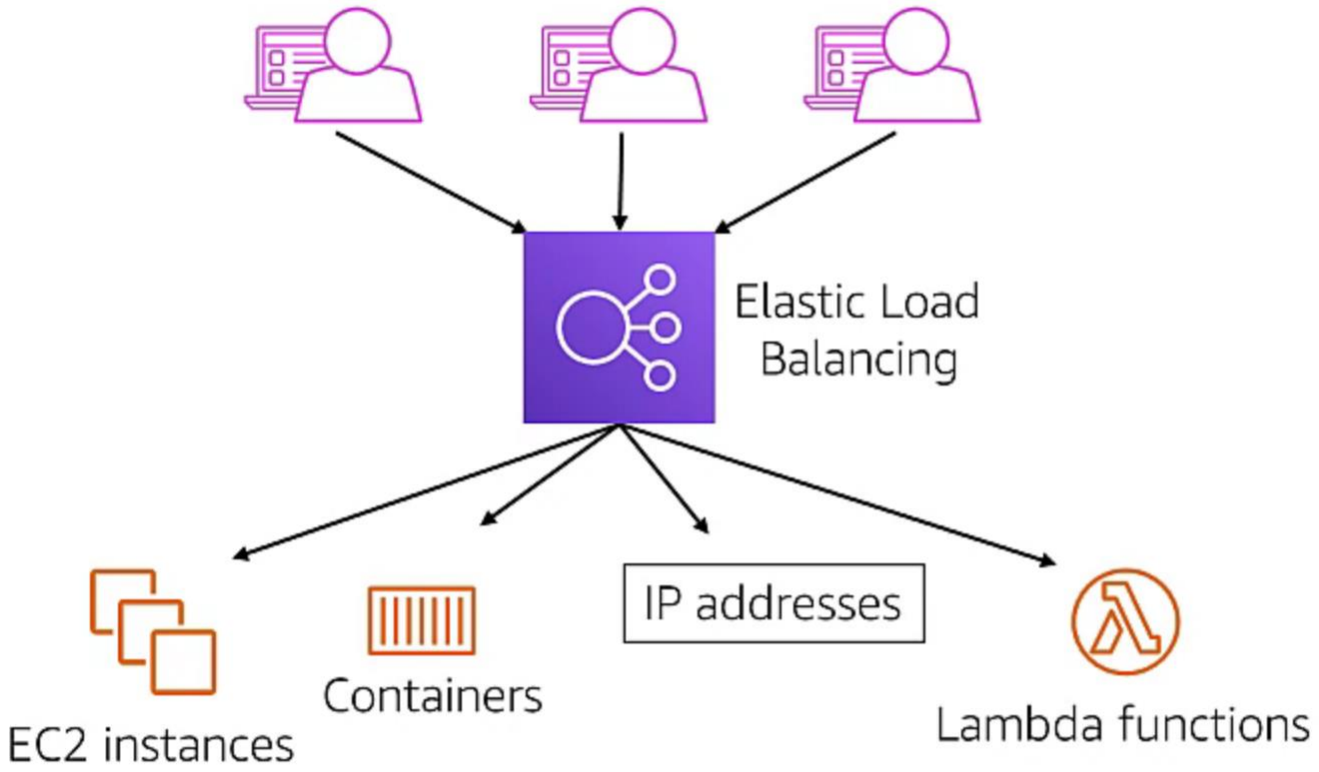
	General Purpose	Compute Optimized	Memory Optimized	Accelerated Computing	Storage Optimized
Instance Types	a1, m4, m5, t2, t3	c4, c5	r4, r5, x1, z1	f1, g3, g4, p2, p3	d2, h1, i3
Use Case	Broad	High performance	In-memory databases	Machine learning	Distributed file systems

Categorizing compute services

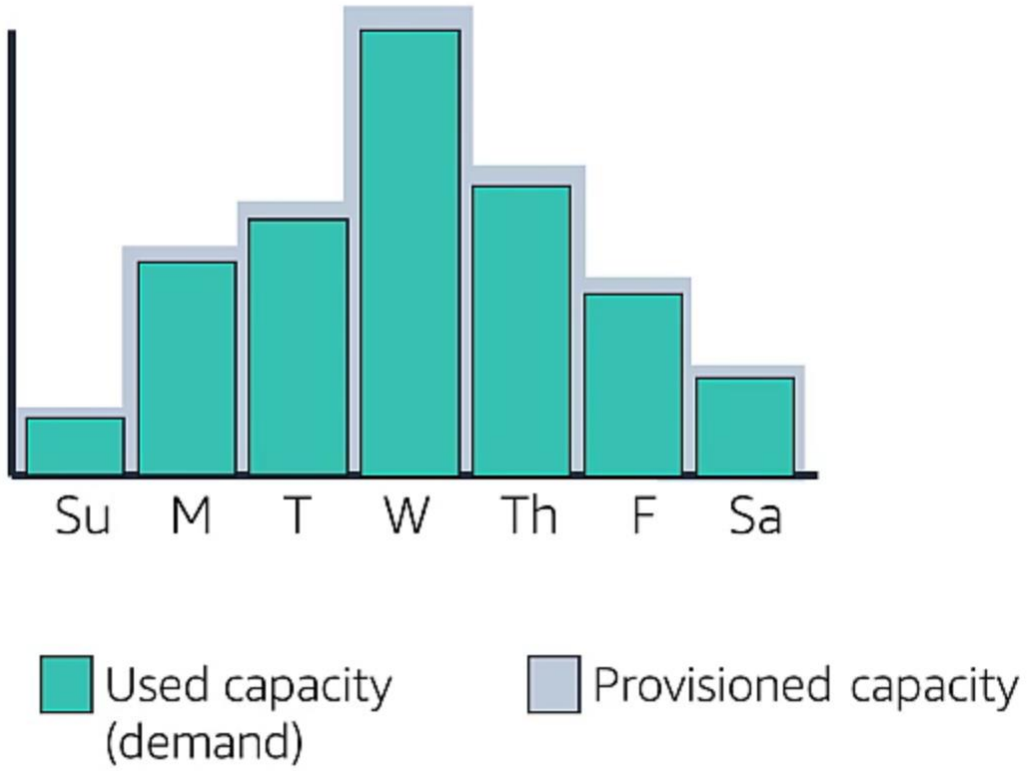
Services	Key Concepts	Characteristics	Ease of Use
• Amazon EC2	• Infrastructure as a service (IaaS) • Instance-based • Virtual machines	• Provision virtual machines that you can manage as you choose	A familiar concept to many IT professionals.
• AWS Lambda	• Serverless computing • Function-based • Low-cost	• Write and deploy code that executes on a schedule or that can be triggered by events • Use when possible (architect for the cloud)	A relatively new concept for many IT staff members, but easy to use after you learn how.
• Amazon ECS • Amazon EKS • AWS Fargate • Amazon ECR	• Container-based computing • Instance-based	• Spin up and execute jobs more quickly	AWS Fargate reduces administrative overhead, but you can use options that give you more control.
• AWS Elastic Beanstalk	• Platform as a service (PaaS) • For web applications	• Focus on your code (building your application) • Can easily tie into other services—databases, Domain Name System (DNS), etc.	Fast and easy to get started.

Elastic Load Balancing

- Distributes incoming application or network traffic across multiple targets in a single Availability Zone or across multiple Availability Zones.
- Scales your load balancer as traffic to your application changes over time.

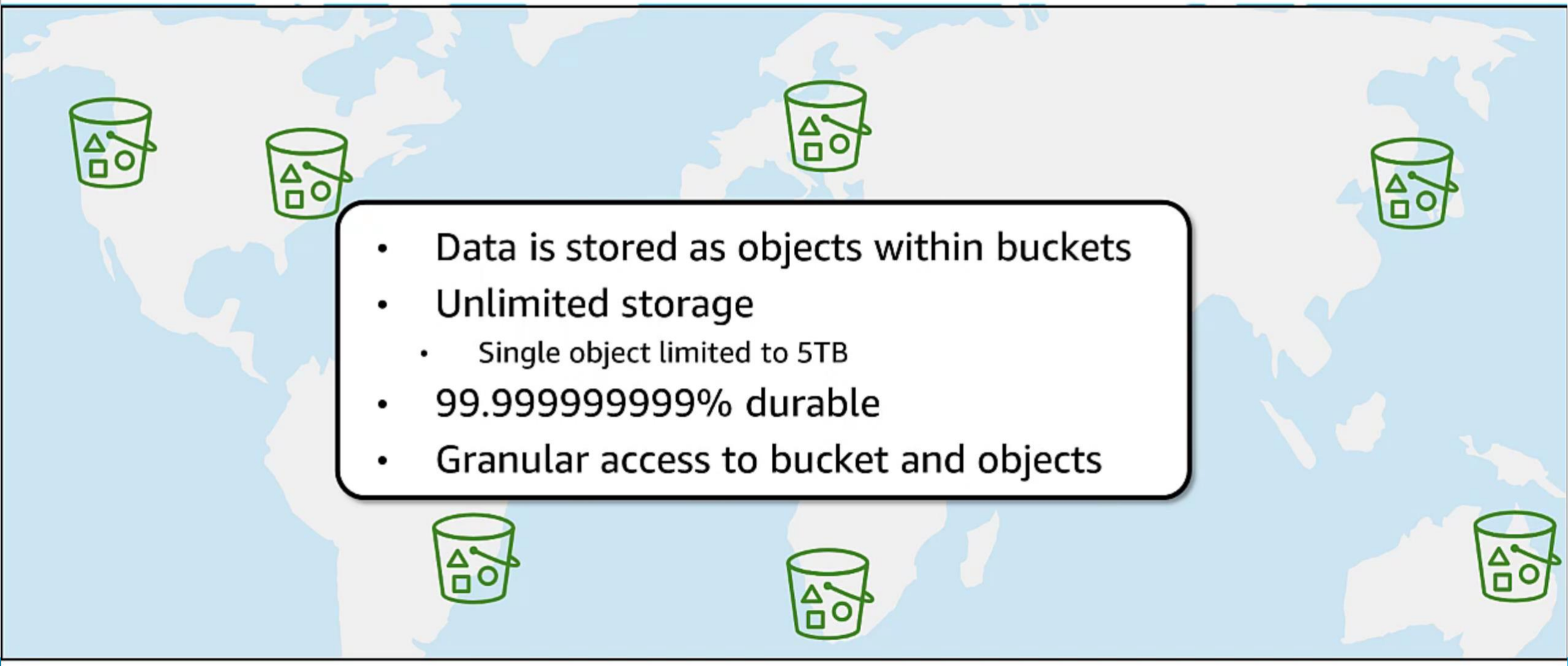


Amazon EC2 Auto Scaling



- Helps you maintain application availability
- Enables you to automatically add or remove EC2 instances according to conditions that you define
- Detects impaired EC2 instances and unhealthy applications, and replaces the instances without your intervention
- Provides several scaling options – Manual, scheduled, dynamic or on-demand, and predictive

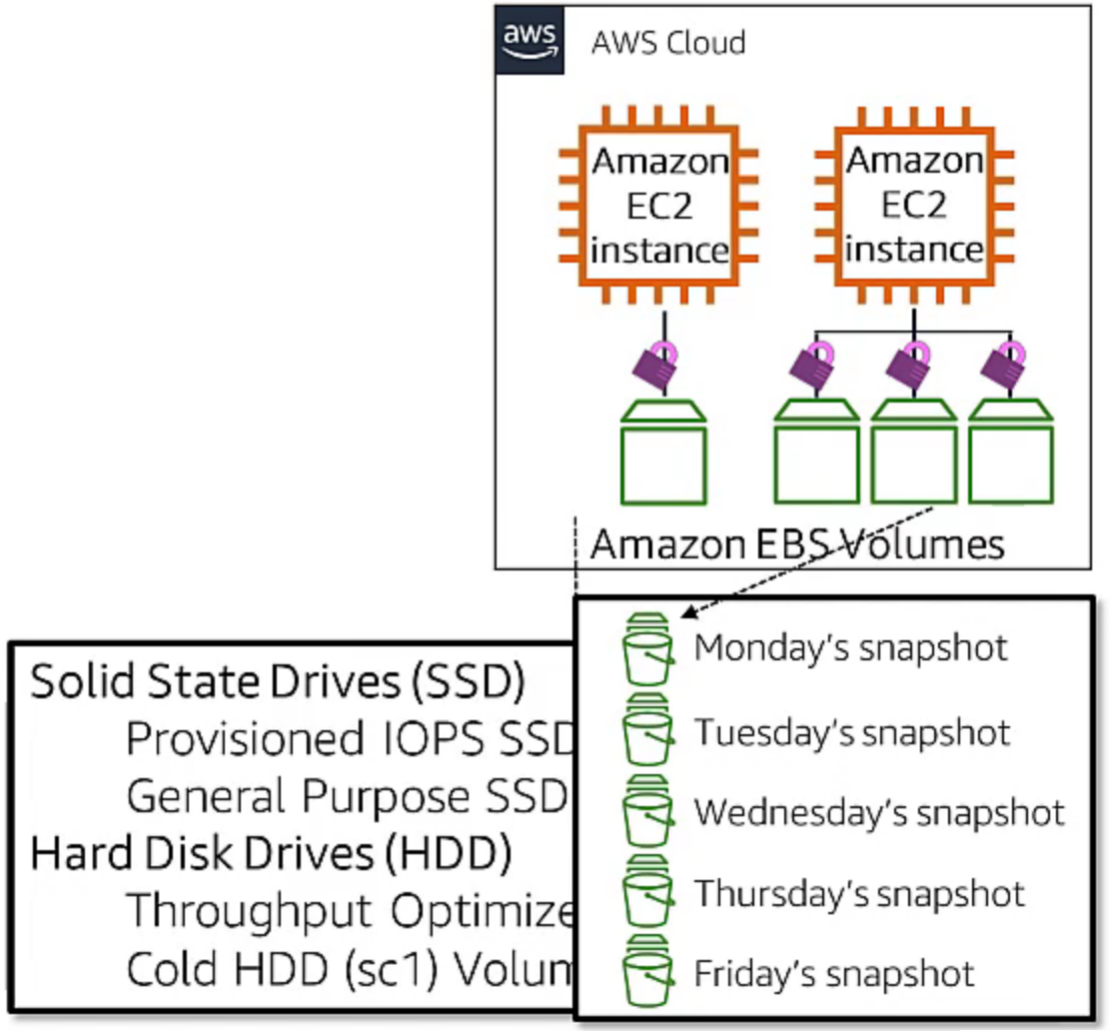
What is Amazon S3?



- Data is stored as objects within buckets
- Unlimited storage
  - Single object limited to 5TB
- 99.999999999% durable
- Granular access to bucket and objects

Amazon EBS

- Persistent block storage for instances
- Protected through replication
- Different drive types
- Scale up or down in minutes
- Pay for only what you provision
- Snapshot functionality
- Encryption available



Shared file systems

What if I have multiple instances that need to use the same storage?

Amazon EBS only attaches to one instance

Amazon S3 is an option but is not ideal

Amazon EFS and FSx are perfect for this task

What is Amazon S3 Glacier?

- Low-cost data archiving and long-term backup
- 3- to 5-hour or within 12 hours*
- Can configure lifecycle archiving of Amazon S3 content to Amazon Glacier

DIY vs. AWS database services

Databases on Amazon EC2

- Operating system access
- Need features of specific application

AWS Database Services

- Easy to set up, manage, maintain
- Push-button high availability
- Focus on performance
- Managed infrastructure

What is Amazon Relational Database Service?

A database service that makes it easy to set up, operate, and scale a relational database in the cloud

Amazon RDS Engines

- Easily scalable
- Automatic software patching
- Automated backups
- Database snapshots
- Multi-AZ deployments
- Automatic host replacement
- Encryption at rest and in transit

What is Amazon Aurora?

- Enterprise-class relational database
- MySQL- or PostgreSQL-compatible
- Up to 5X faster than standard MySQL databases
- Up to 3X faster than standard PostgreSQL databases
- Continuous backup to Amazon S3
- Up to 15 low-latency read replicas

What is Amazon DynamoDB?

aws training and certification

Fast and flexible NoSQL database service for any scale

- Fully managed
- Low-latency queries
- Fine-grained access control
- Regional and global options

Other purpose-built database services

Amazon Redshift
Fast, scalable data warehouse

Amazon DocumentDB
MongoDB-compatible database

Amazon Neptune
Graph database

Network service category

AWS network services

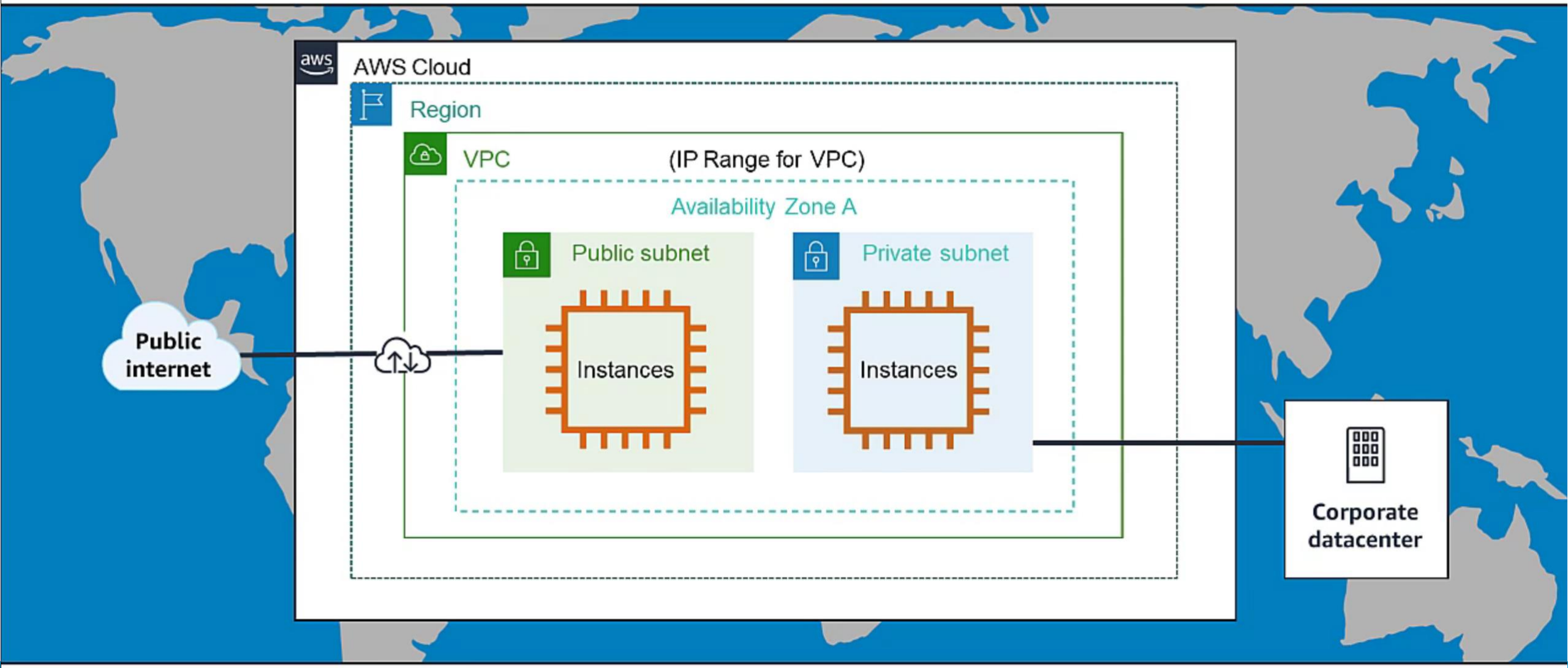
Amazon VPC

Amazon Route 53

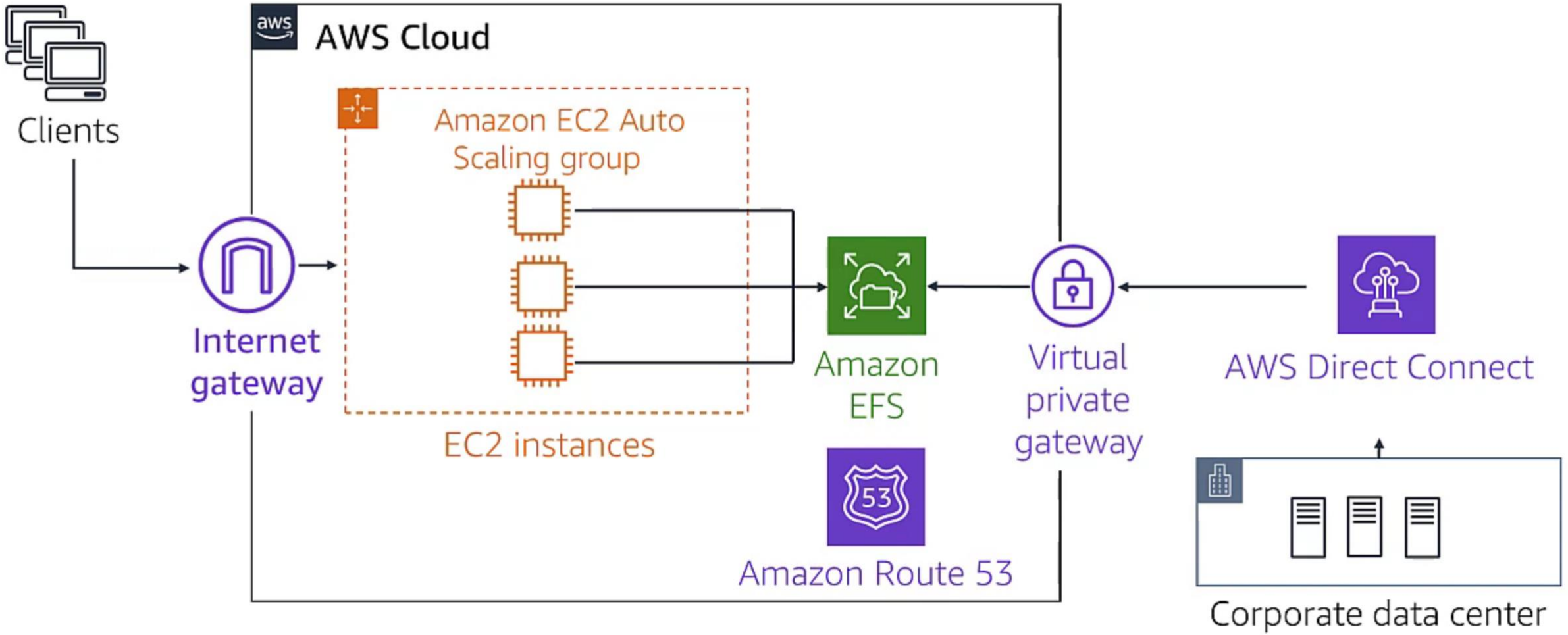
AWS Direct Connect

AWS VPN

Amazon VPC


aws
AWS Cloud
Region
VPC
(IP Range for VPC)
Availability Zone A
Public subnet
Private subnet
Instances
Instances
Public internet
Corporate datacenter

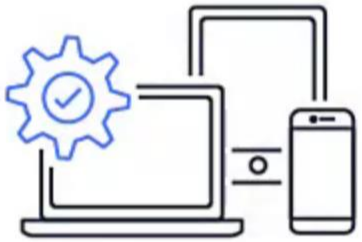
Putting it all together



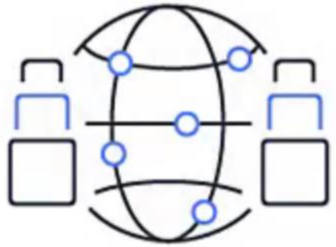
Security is our top priority

Designed for security

Constantly monitored



Highly automated



Highly available

Highly accredited

Security, identity, and compliance service category

AWS security, identity, and compliance services

AWS Identity and Access Management (IAM)

AWS Organizations

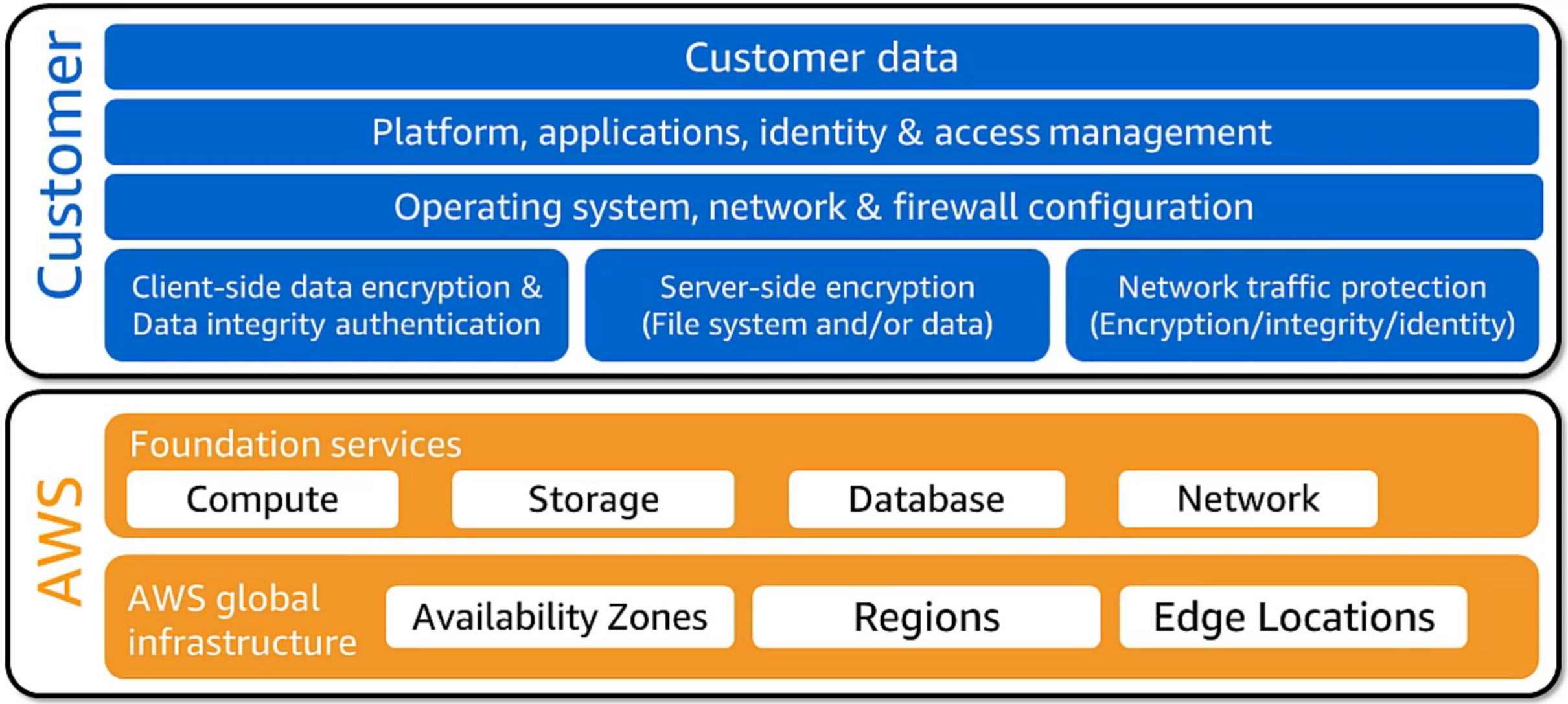
Amazon Cognito

AWS Artifact

AWS Key Management Service

AWS Shield

AWS shared responsibility model



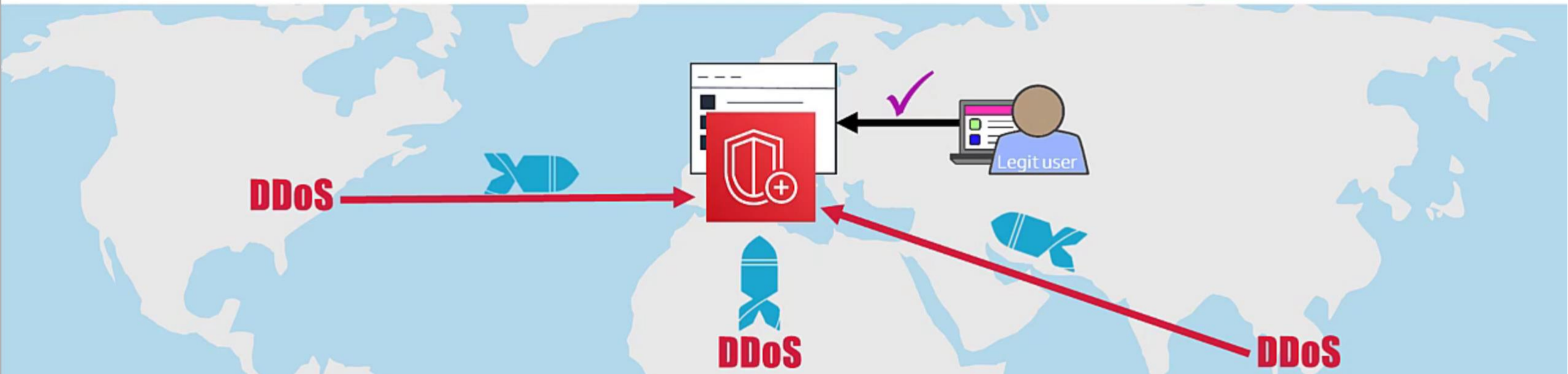
AWS IAM

AWS IAM

Securely **control access** to your AWS resources

- Assign granular permissions to users, groups, or roles
- Share temporary access to your AWS account
- Federate users in your corporate network or with an internet identity provider

What is AWS Shield?



- A managed DDoS protection service
- Always-on detection and mitigations
- Seamless integration and deployment
- Cost-efficient and customizable protection

Assurance programs

How do you pay for AWS?

Pay as you go

Save when you reserve

Pay less by using more

Amazon EC2: Four purchase types

On-Demand Instances

- Charged per hour/second*
- Short-term
- Unpredictable workloads

Reserved Instances

- Discount for 1 to 3 year commitments
- Applications with steady state usage

Dedicated Hosts

- Physical server dedicated to you
- Applications with specific compliance requirements

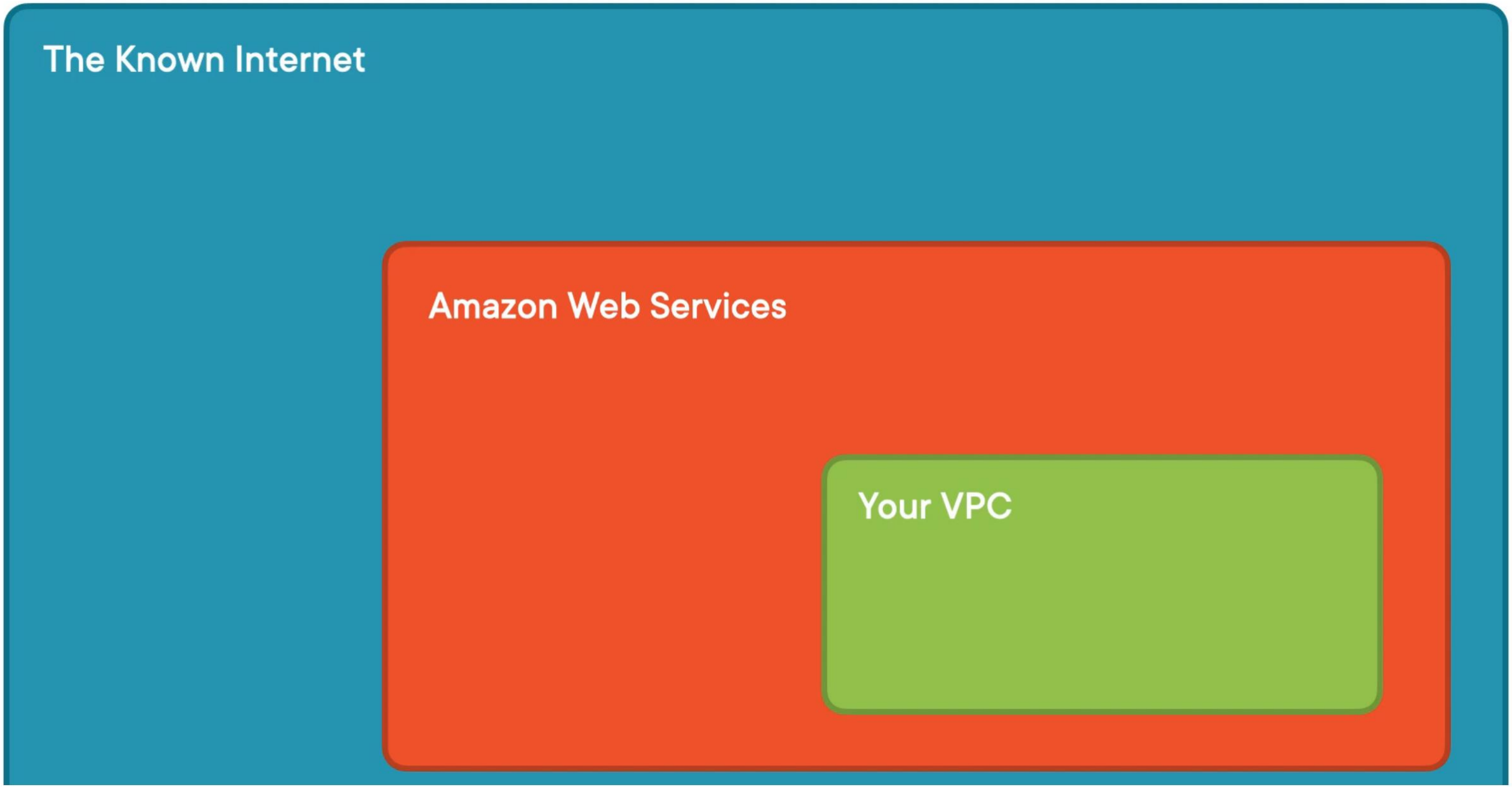
Spot Instances

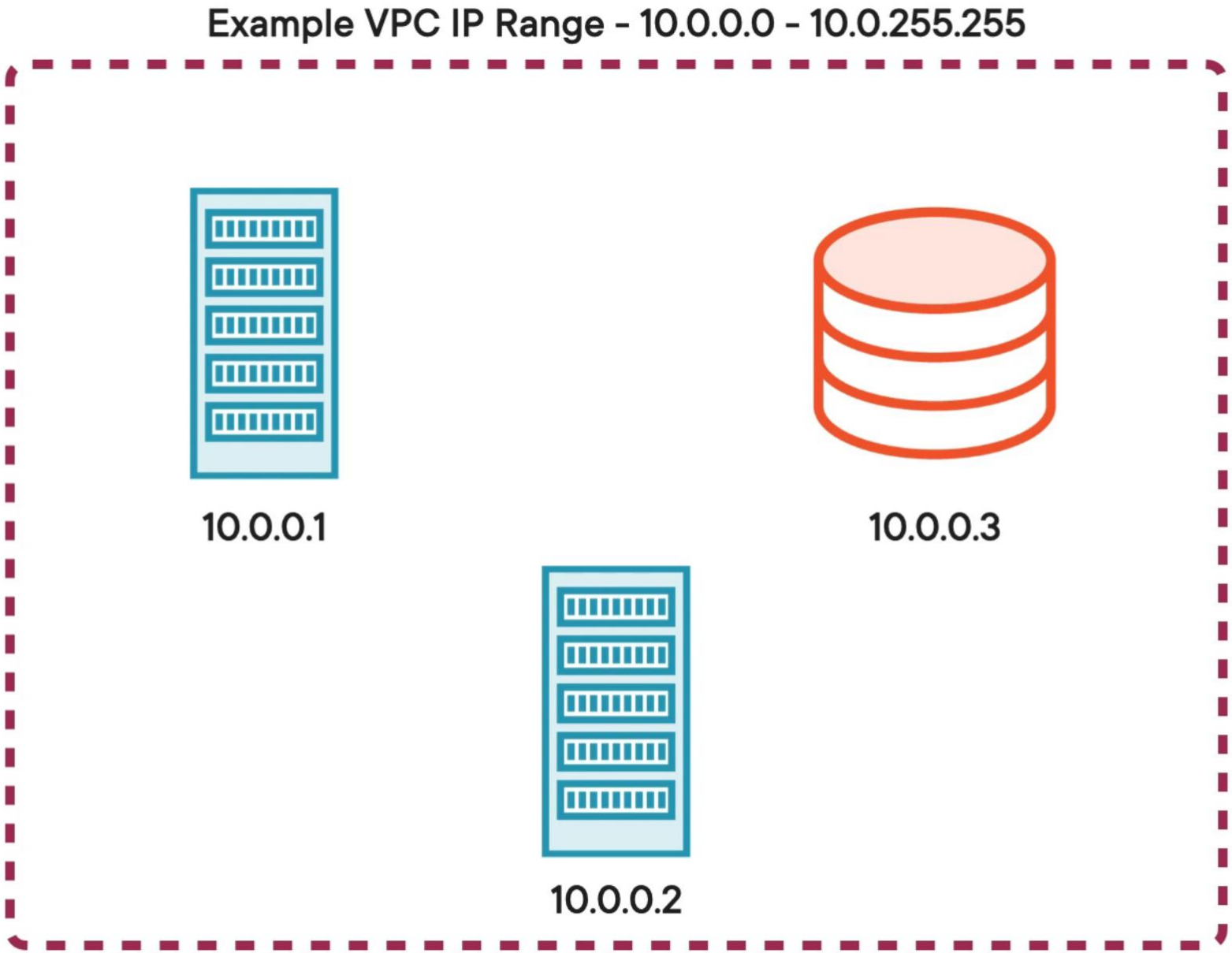
- Spare AWS capacity for up to 90% discount
- Applications with flexible start and end times
- Urgent computing needs for large amounts of capacity

AWS Free Tier

Enables you to gain free hands-on experience with the AWS platform, products, and services.

Virtual Private Cloud Overview


The Known Internet
Amazon Web Services
Your VPC

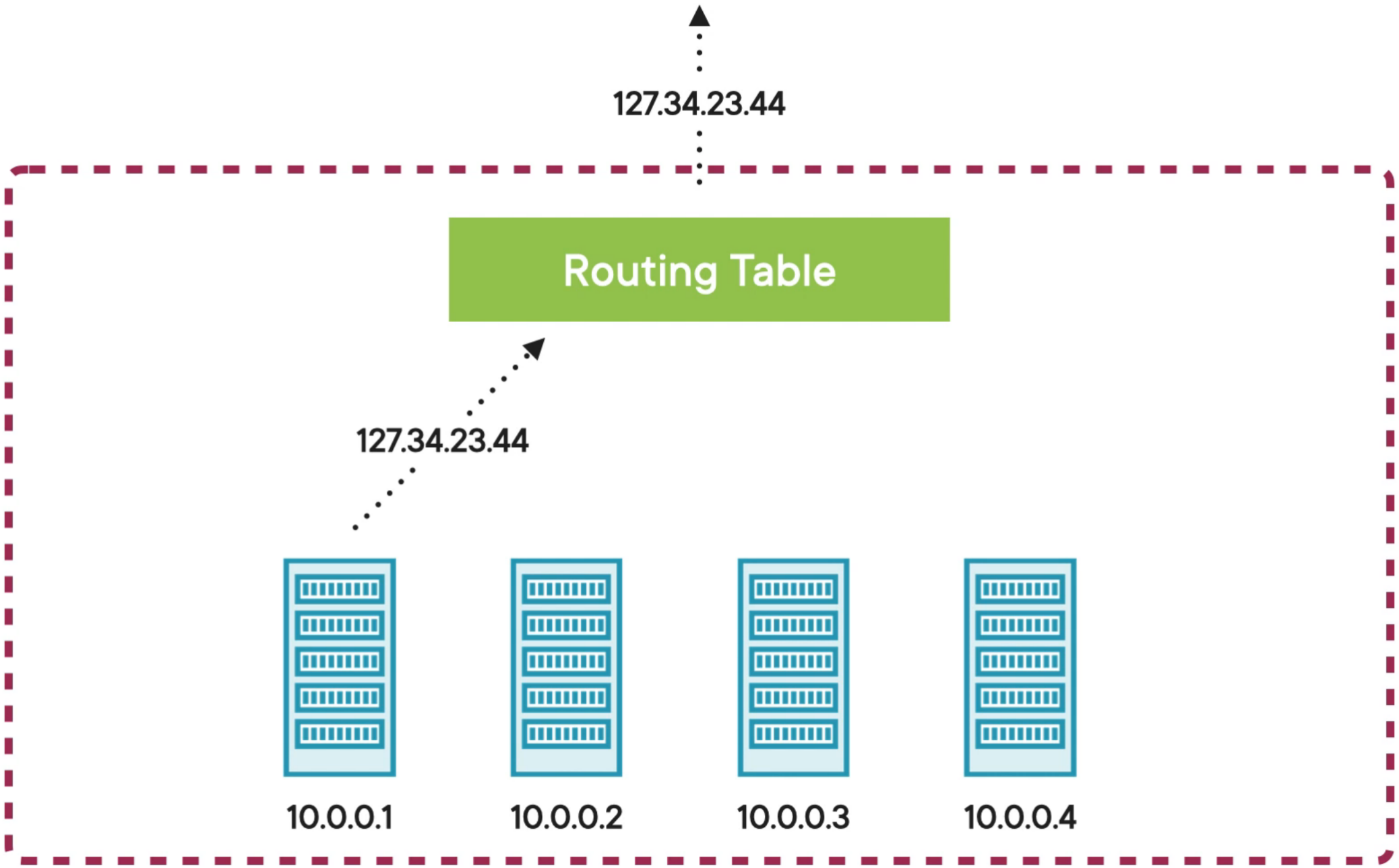

Example VPC IP Range - 10.0.0.0 - 10.0.255.255
10.0.0.1
10.0.0.3
10.0.0.2

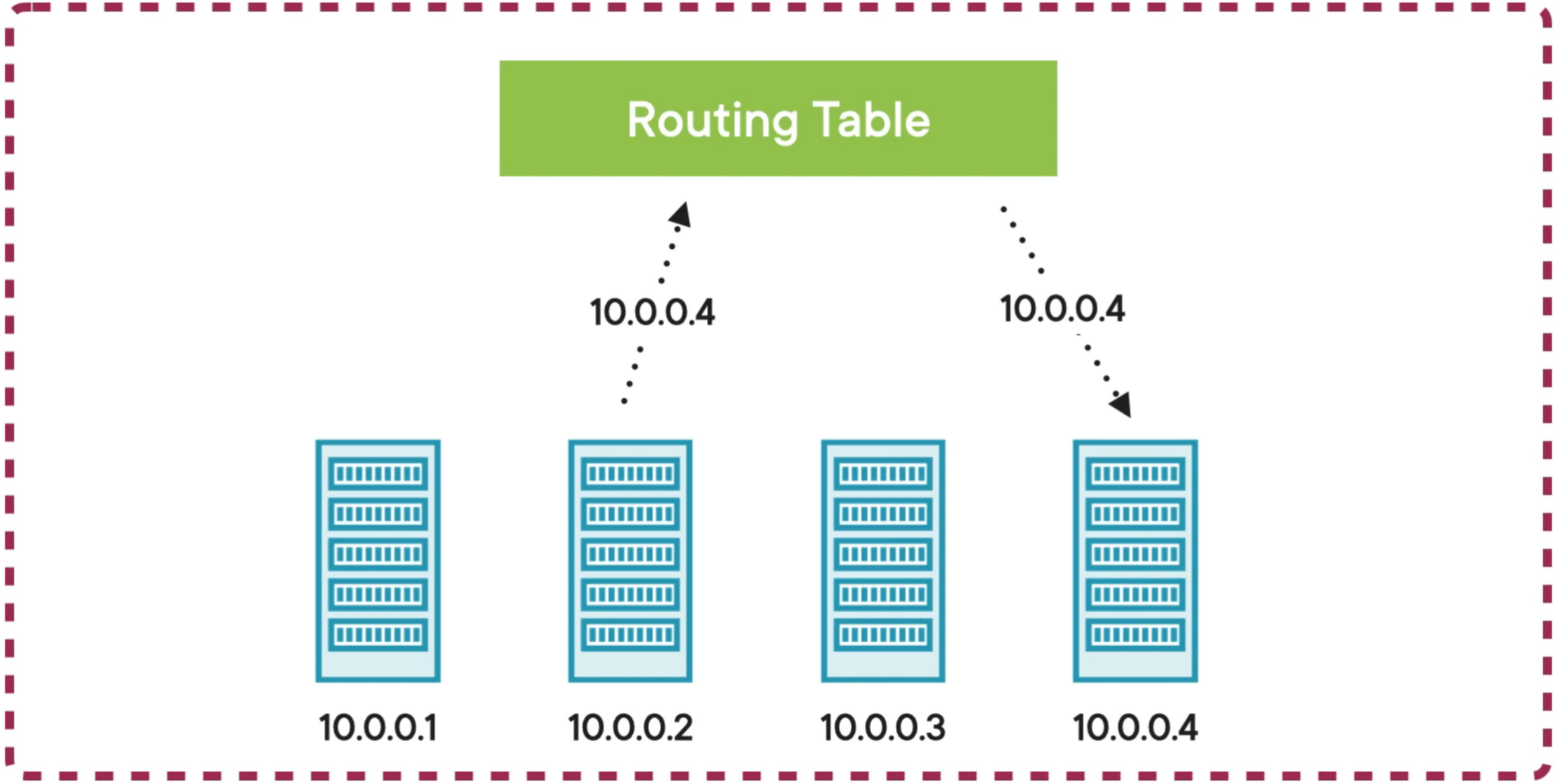
VPC is free!

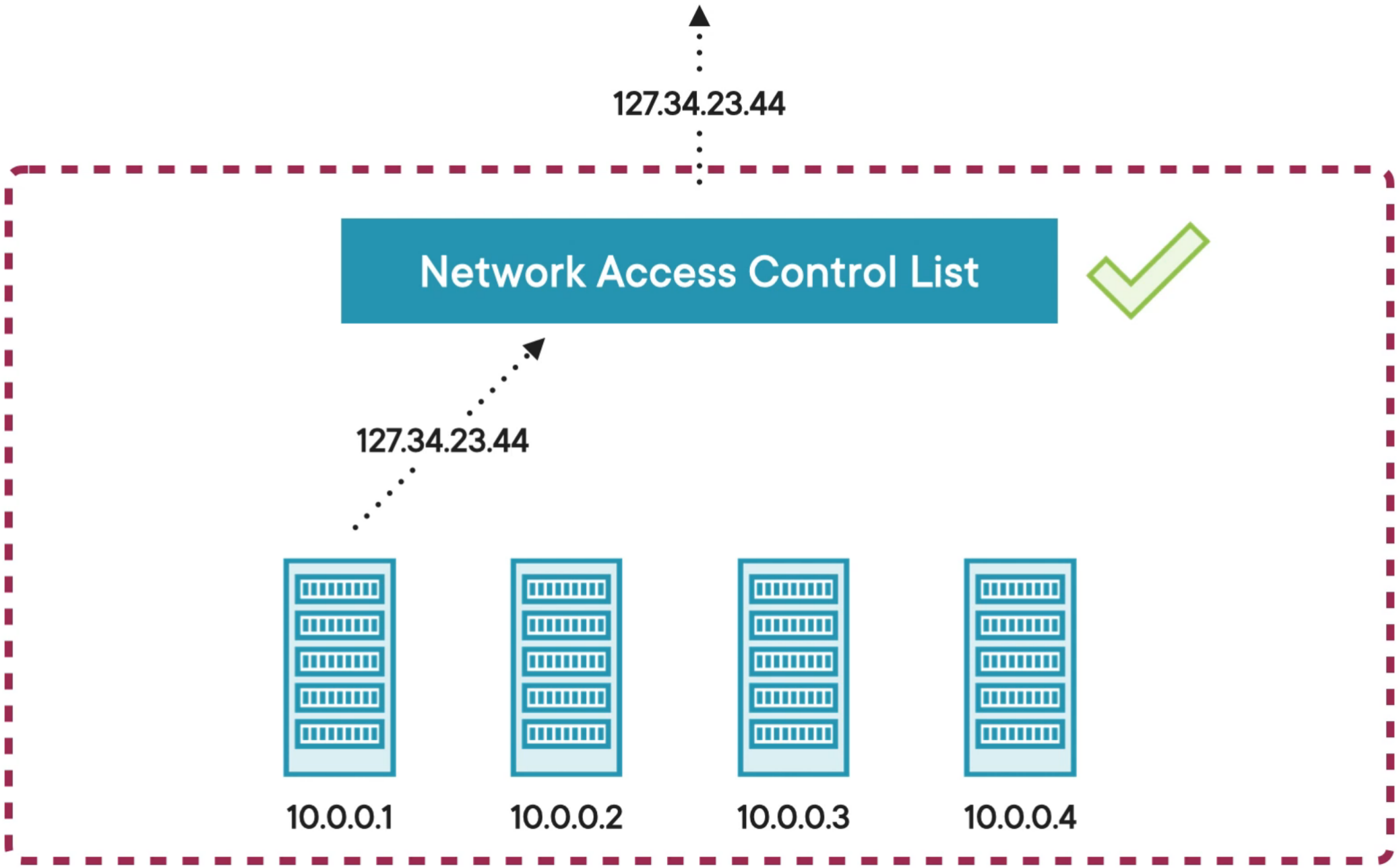
Security Group

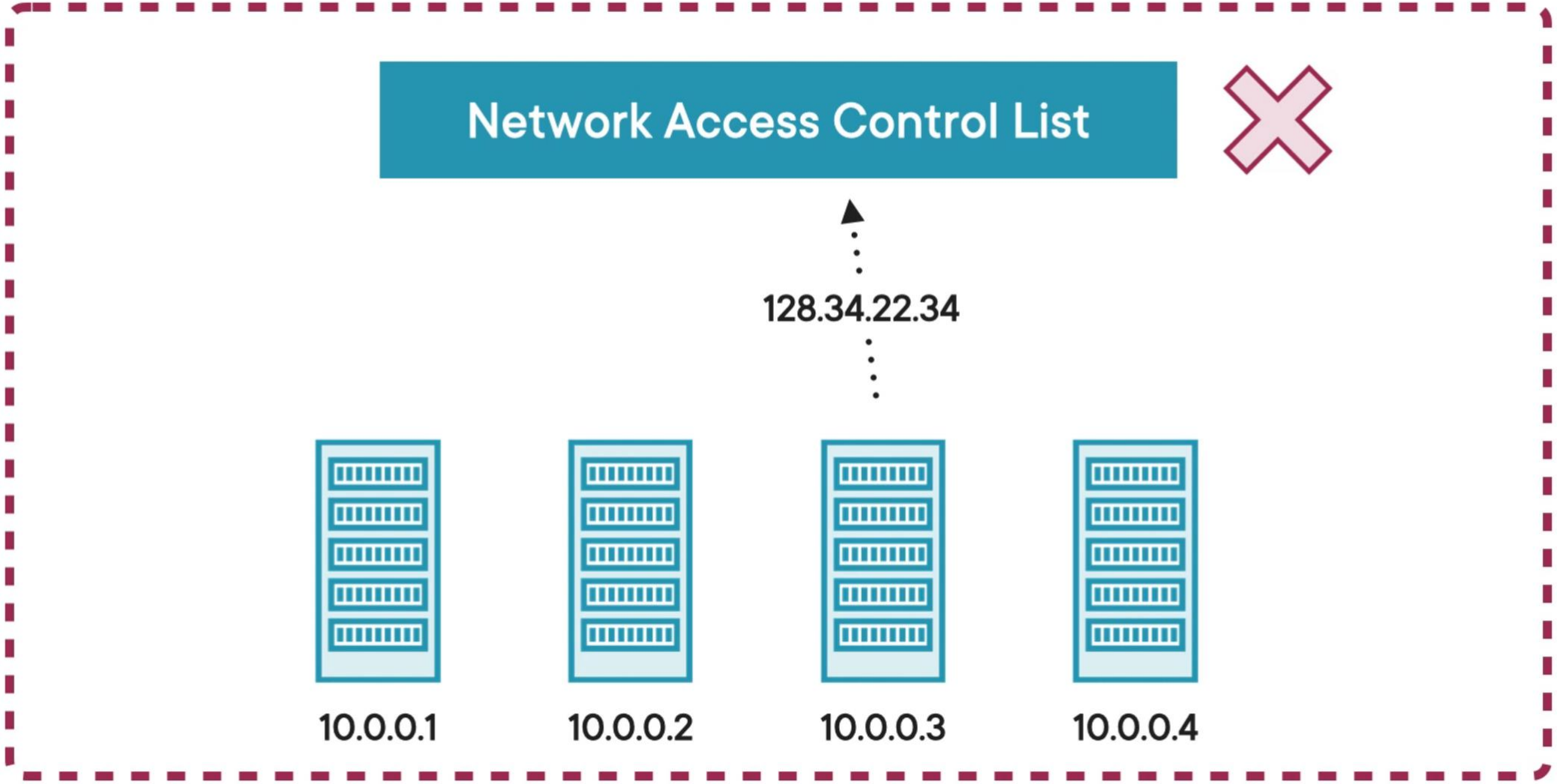
Defines allowed incoming/outgoing IP addresses and ports.
Kind of like a mini-firewall.

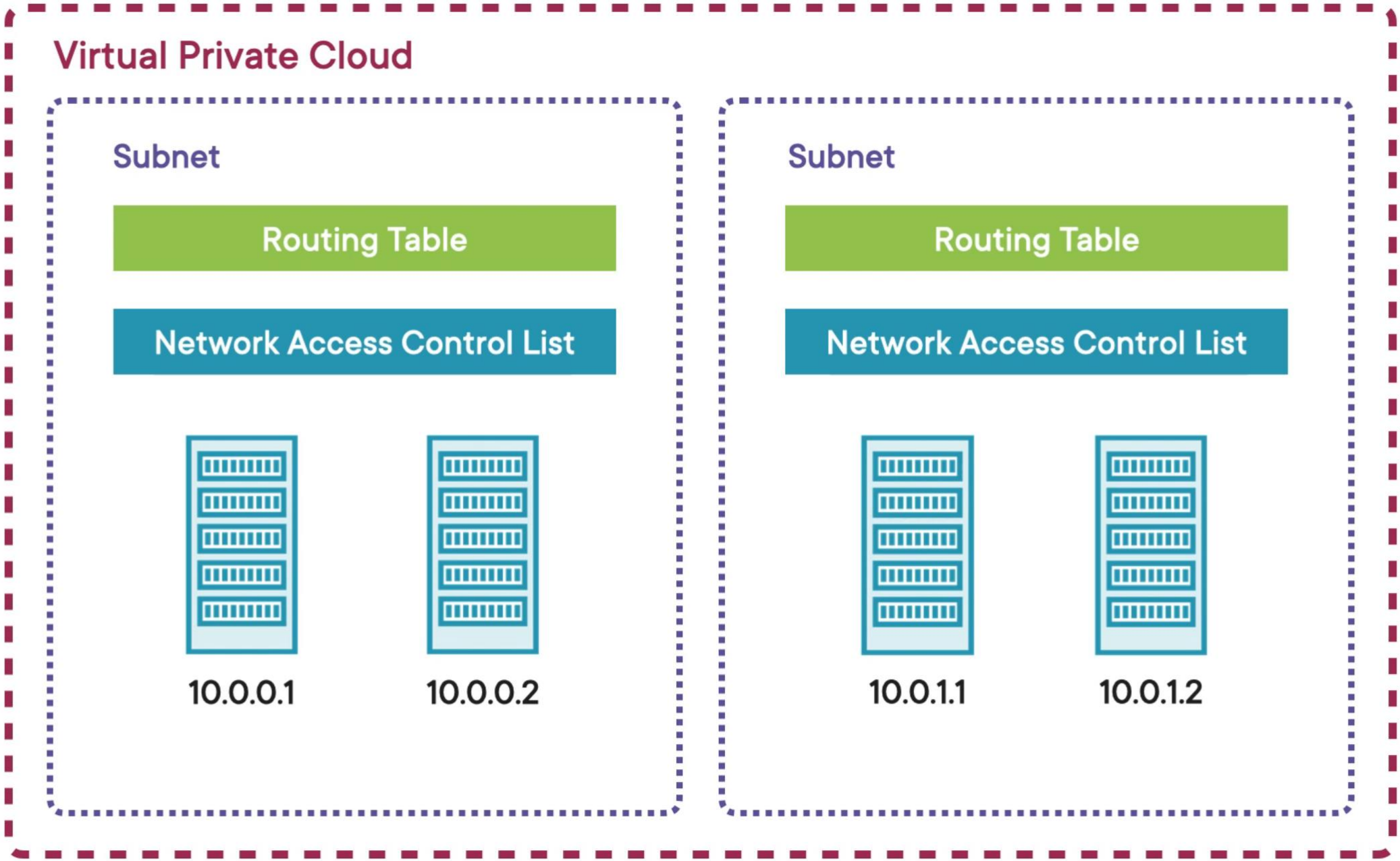
Security Groups


127.34.23.44
Routing Table
127.34.23.44
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4

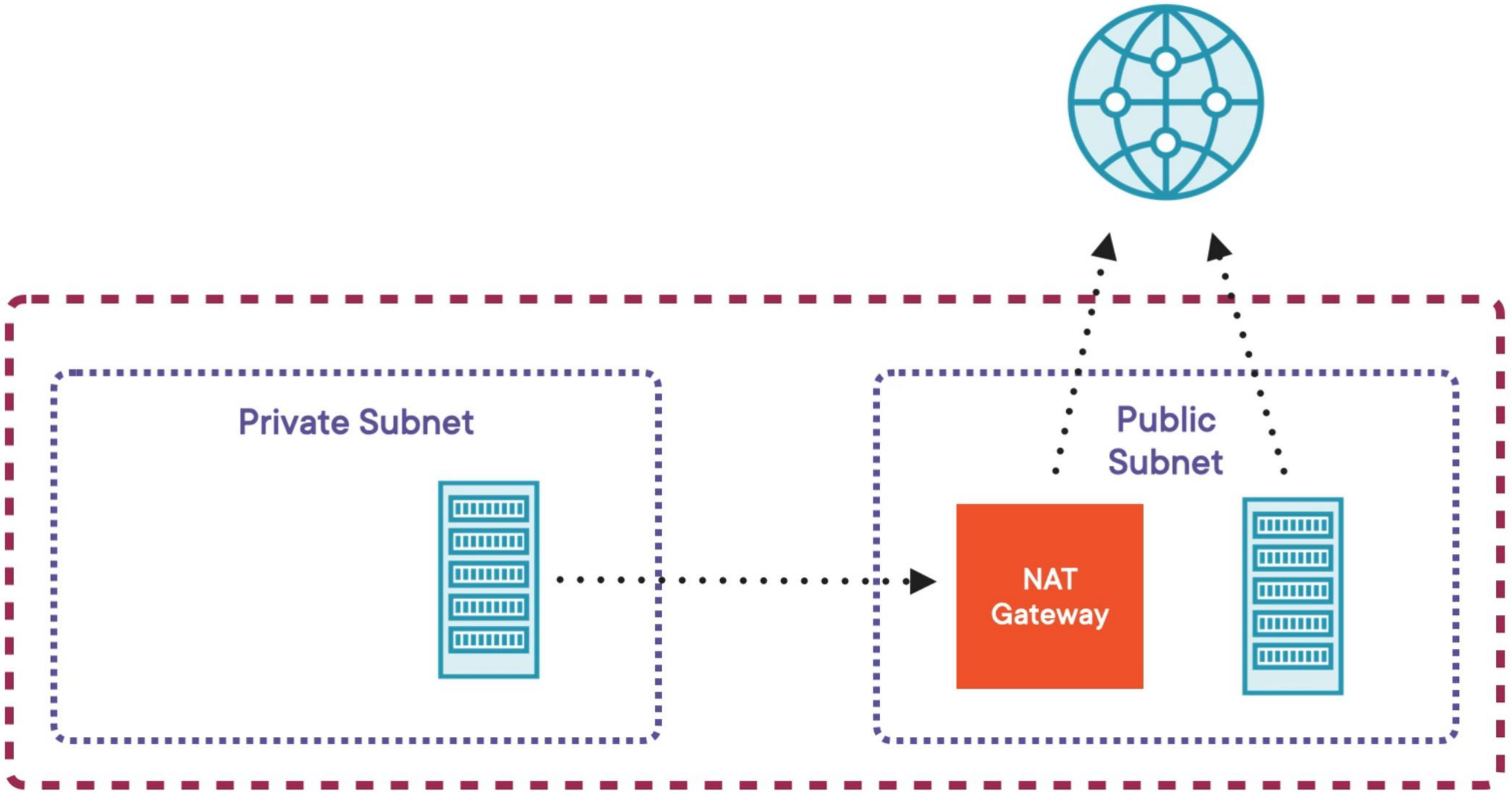

Routing Table
10.0.0.4
10.0.0.4
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4


127.34.23.44
Network Access Control List
127.34.23.44
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4

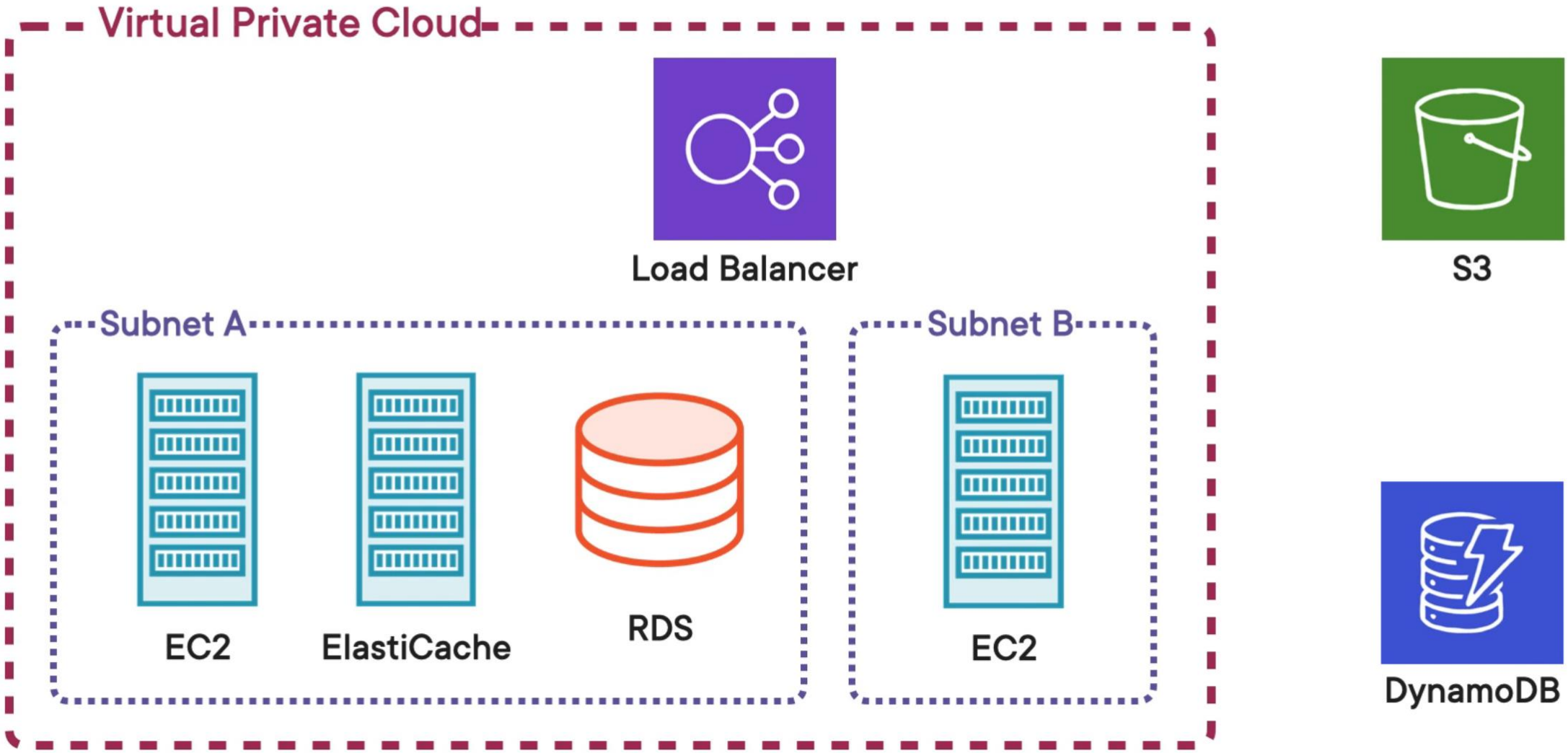

Network Access Control List
128.34.22.34
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4


Virtual Private Cloud
Subnet
Routing Table
Network Access Control List
10.0.0.1
10.0.0.2
Subnet
Routing Table
Network Access Control List
10.0.1.1
10.0.1.2

Public and Private Subnet Configuration



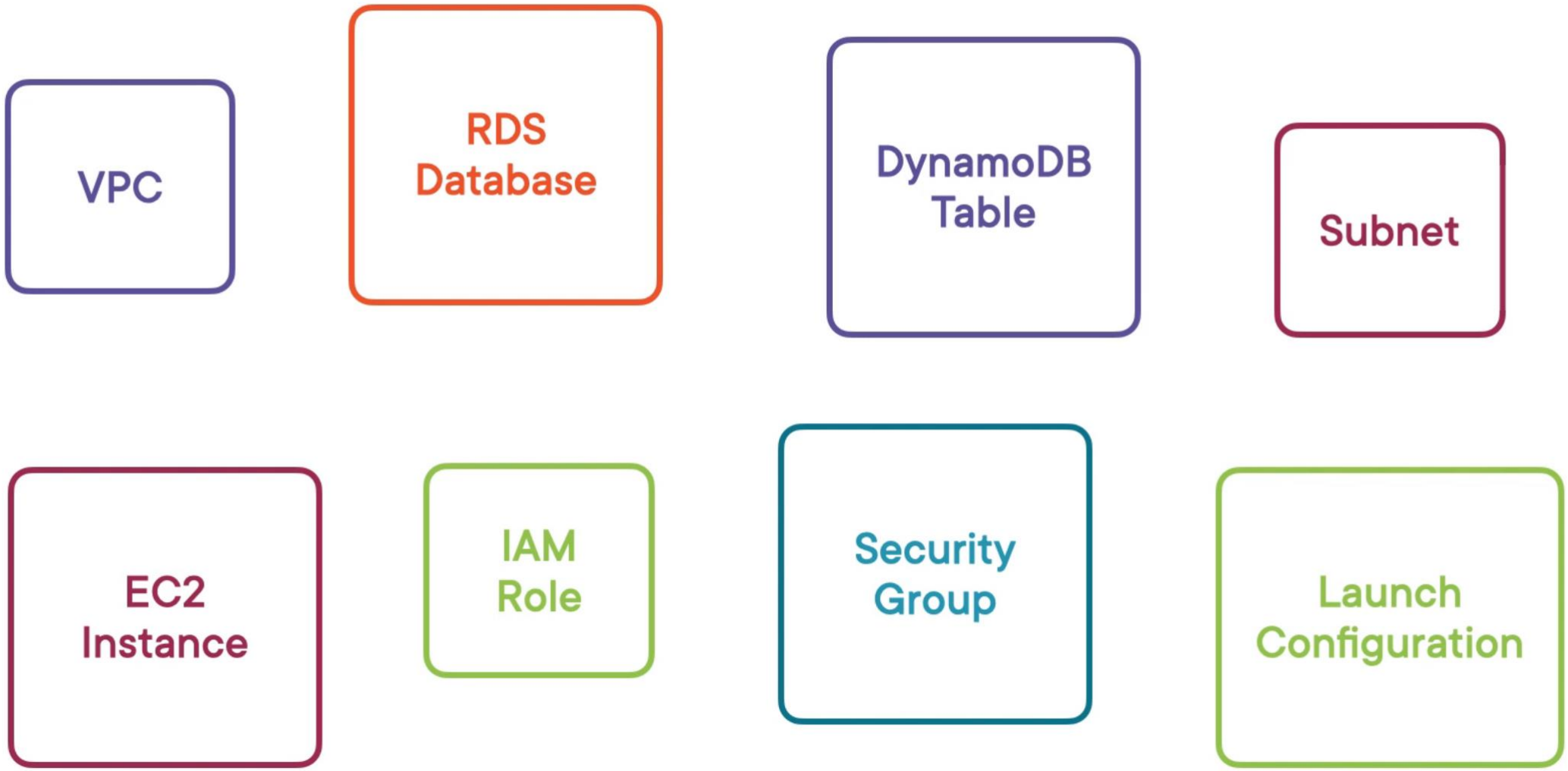
Pizza Luvrs Architecture Diagram

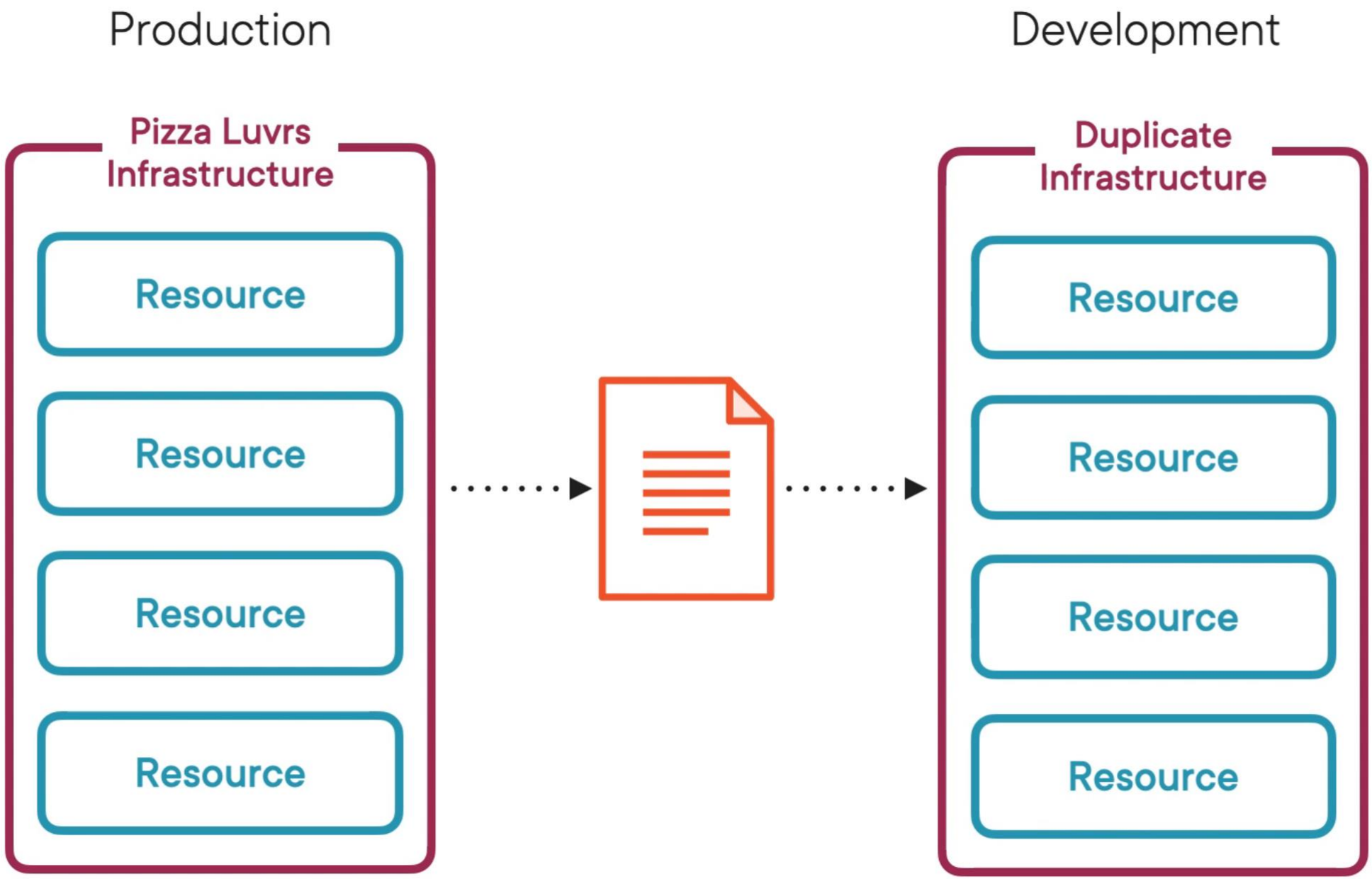


CloudFormation

Service to provision resources using templates.

Resources Created for Pizza Luvrs




Production
Development
Pizza Luvrs Infrastructure
Resource
Resource
Resource
Resource
Duplicate Infrastructure
Resource
Resource
Resource
Resource

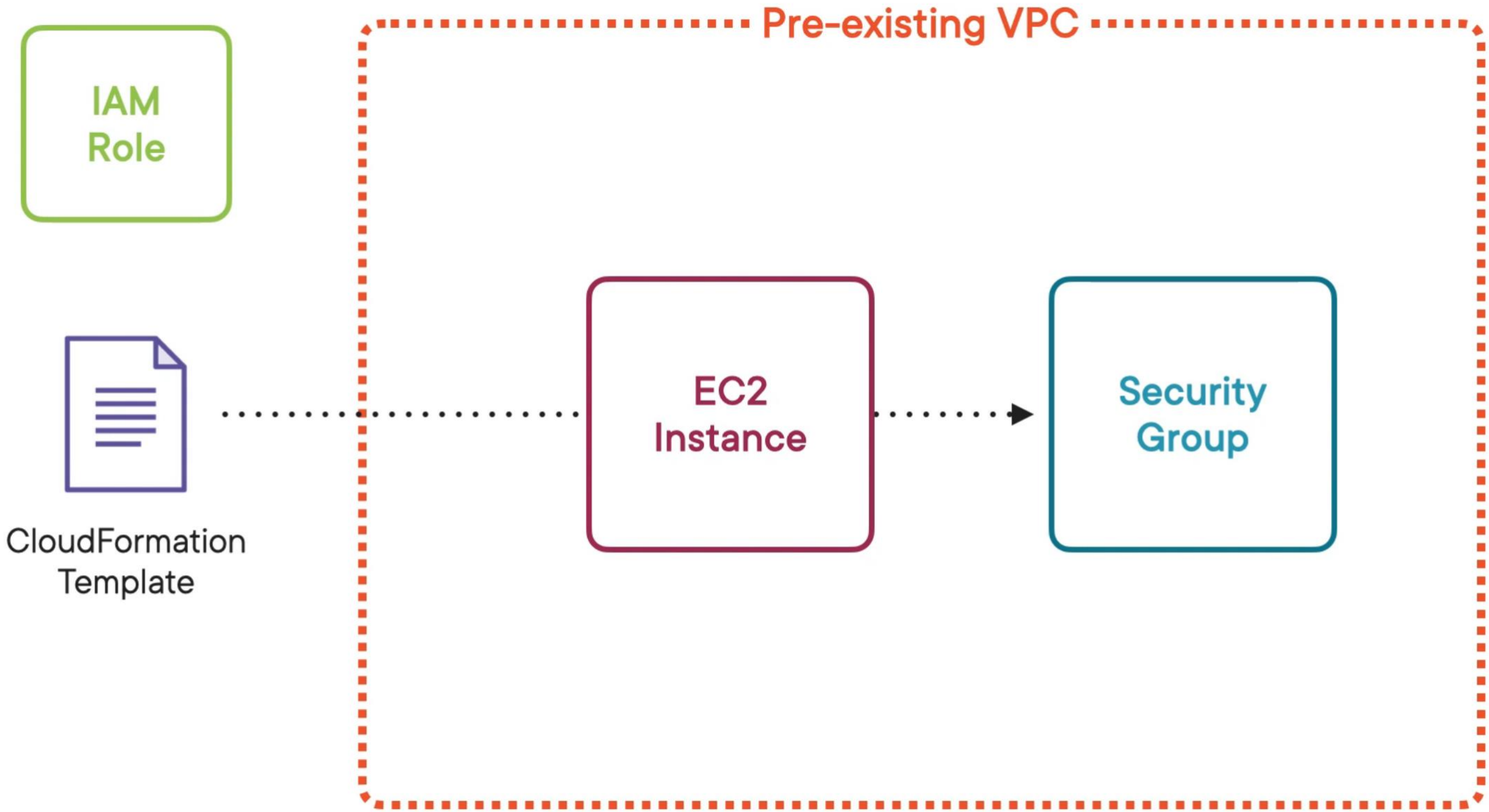
CloudFormation Template

JSON Document

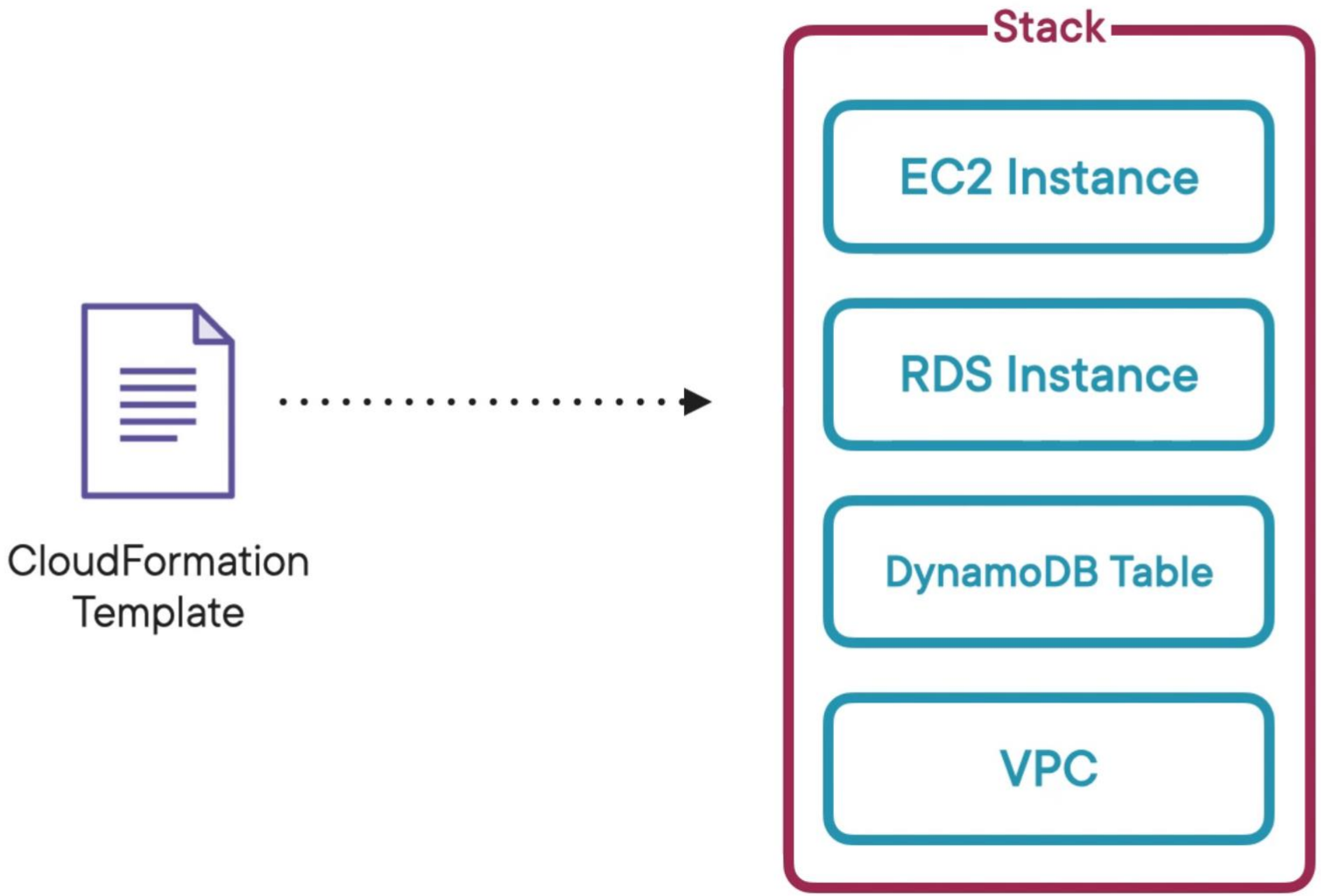
Contains configuratiion for resources

Can be used in version control

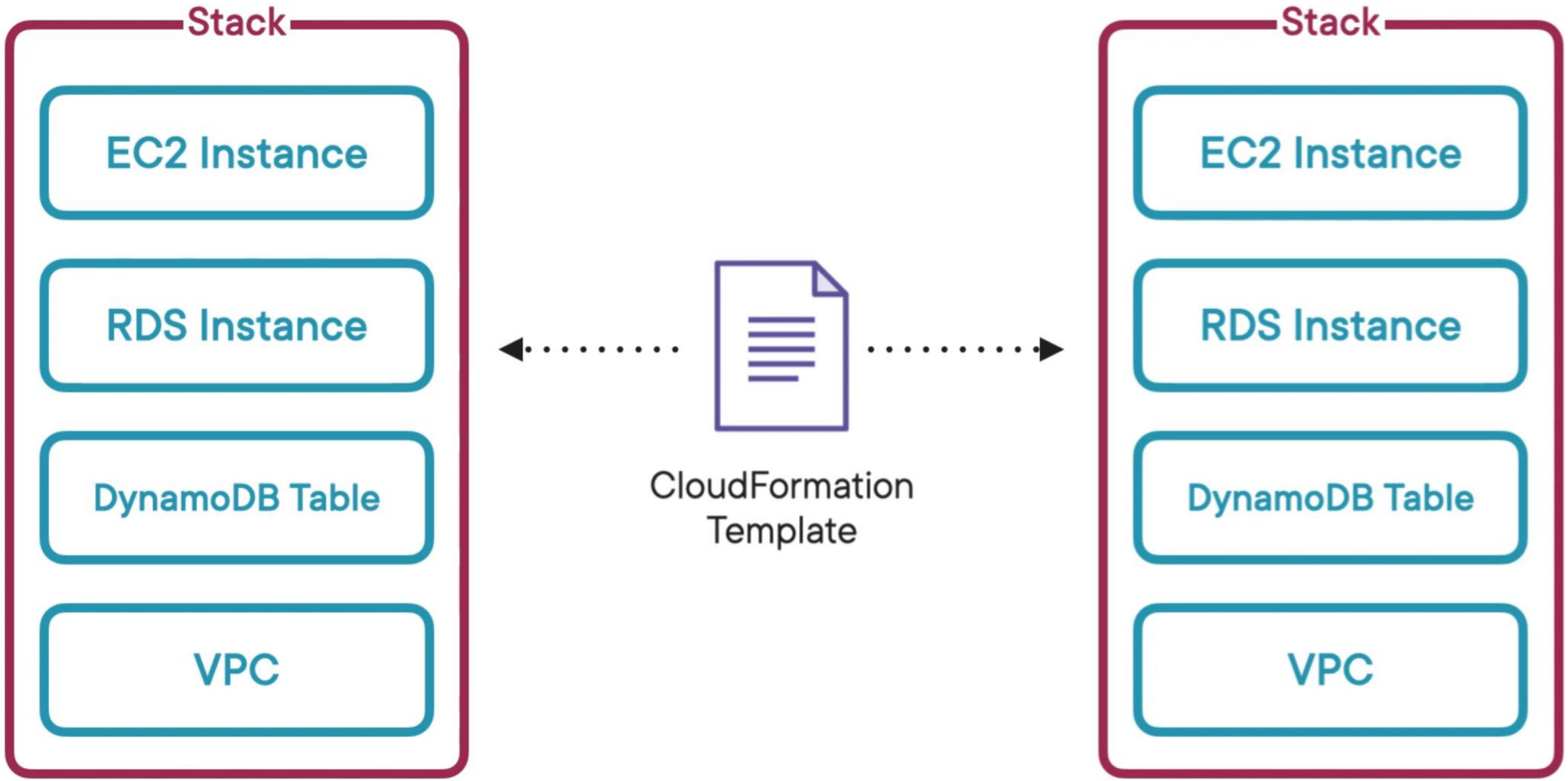
No Limit to Amount of resources


Pre-existing VPC
IAM Role
CloudFormation Template
EC2 Instance
Security Group

CloudFormation Stack



CloudFormation Stack



Other CloudFormation Operations

Updating
a Stack

Deleting
a Stack

CloudFormation

Only provisions resources

Elastic Beanstalk

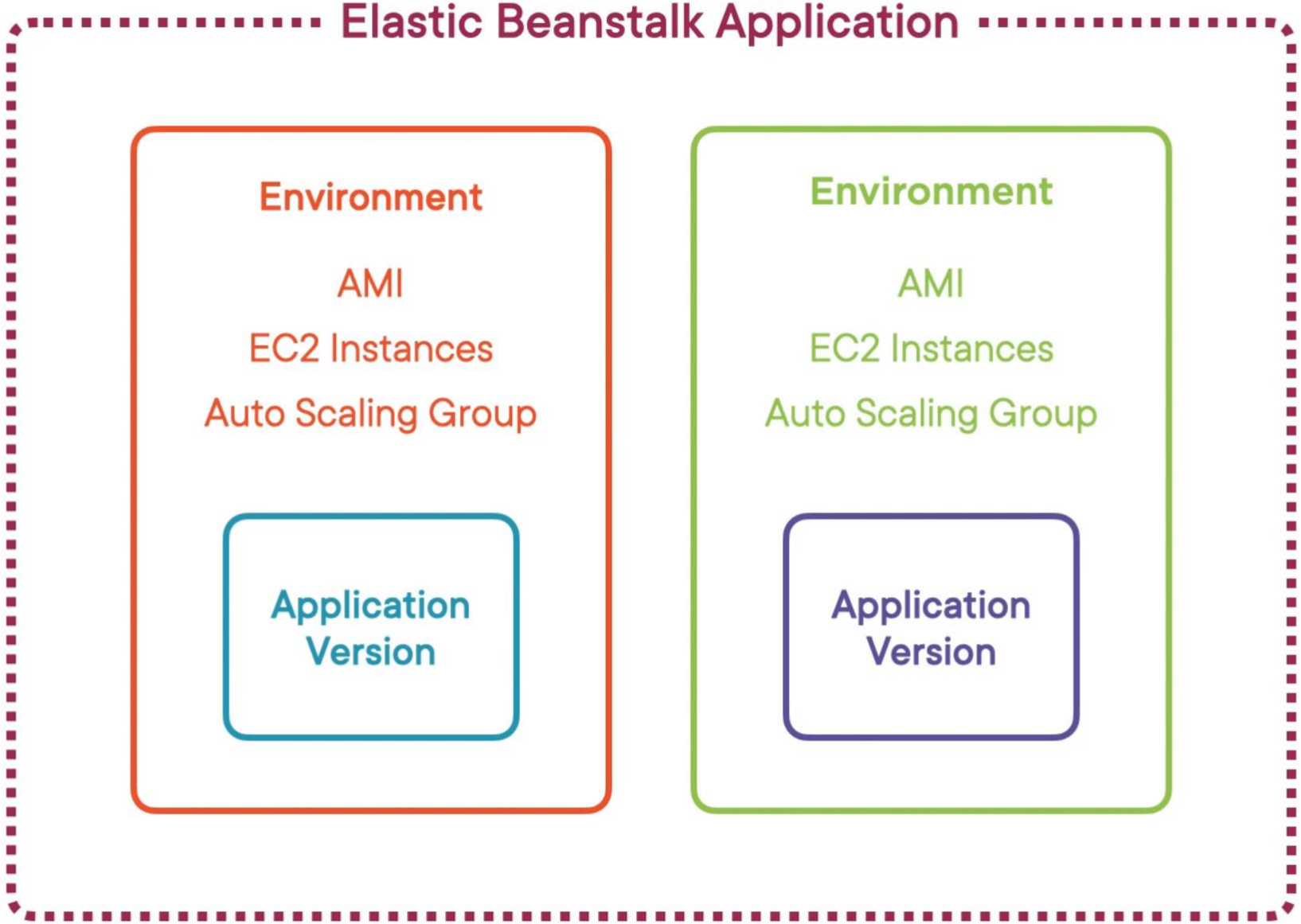
Provisions resources and runs your application

Elastic Beanstalk Application

Represents a logical application

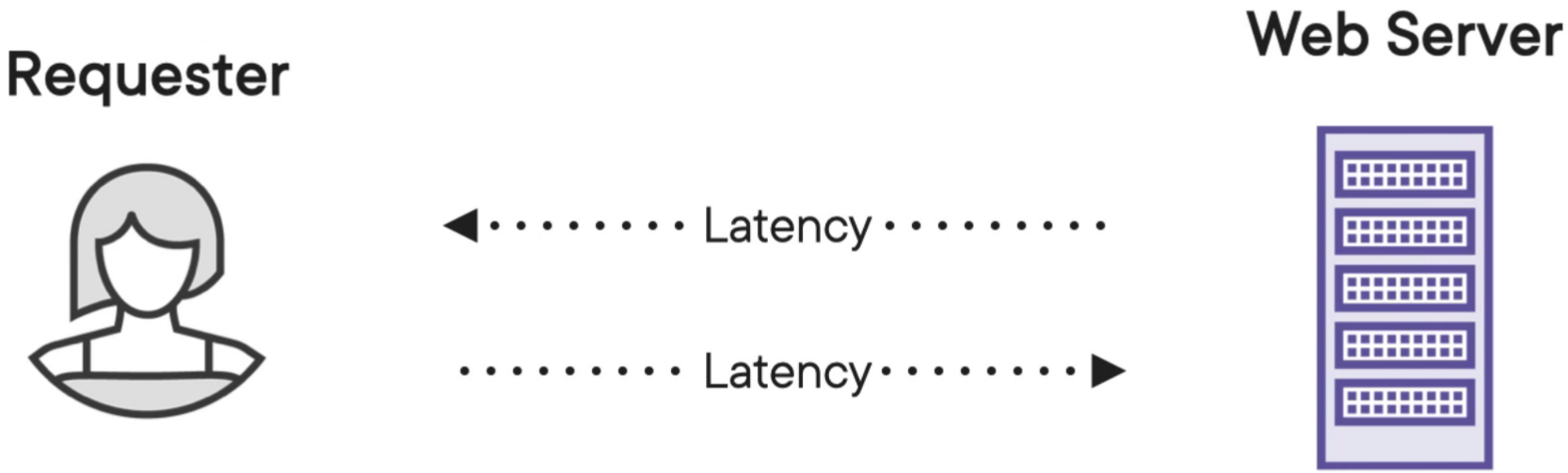
Runs a single platform (eg. Node.js, Java)

Has one or more application versions


Elastic Beanstalk Application
Environment
AMI
EC2 Instances
Auto Scaling Group
Application Version
Environment
AMI
EC2 Instances
Auto Scaling Group
Application Version

CloudFront

Global content delivery network designed to reduce latency and reduce application load.

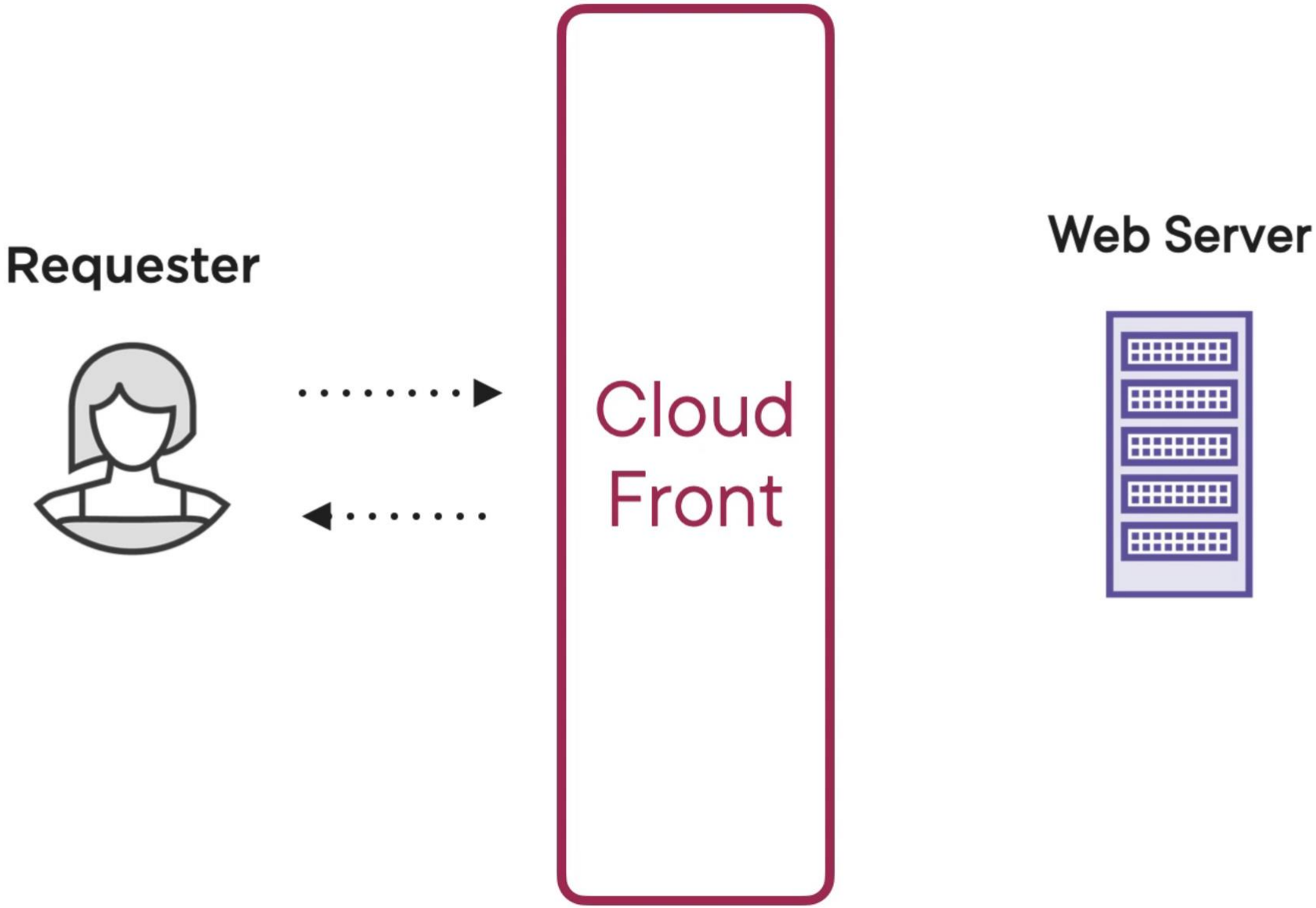

Requester
Web Server
Latency
Latency

Steps to Improve Latency

Improve application performance

Use larger EC2 instances

Reduce distance between user and app

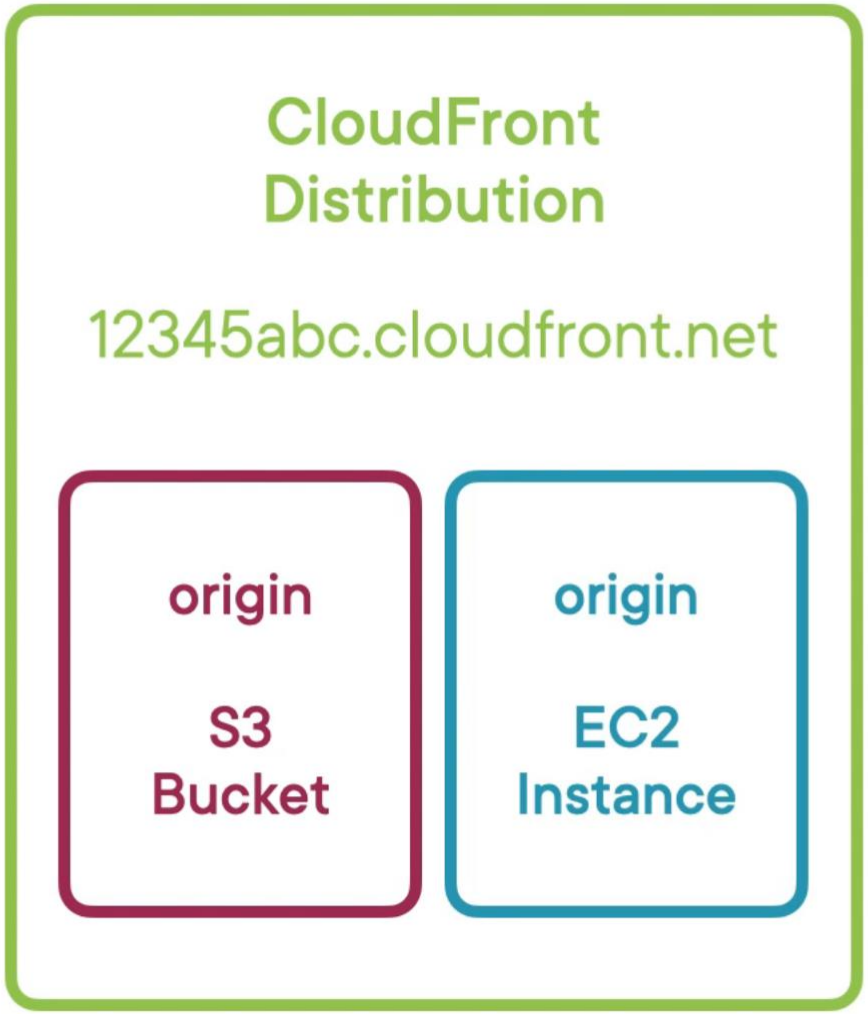

Requester
Cloud Front
Web Server

CloudFront

Integrates with S3, EC2, and load balancers

Edges “Objects” and serves them directly

Proxies dynamic content to origin source


CloudFront
Distribution
12345abc.cloudfront.net
origin
S3
Bucket
origin
EC2
Instance

CloudFront Distribution Behaviors

Determines cache behavior based on path

Set time-to-live for specific content

Fine-tuned control over caching behavior

Networking & Content Delivery

Amazon CloudFront

Securely deliver content with low latency and high transfer speeds

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency and high transfer speeds.

Get started with CloudFront

Enable accelerated, reliable and secure content delivery for Amazon S3 buckets, Application Load Balancers, Amazon API Gateway APIs, and more in 5 minutes or less.

Create a CloudFront distribution

AWS Free Tier

- 50 GB of data transfer out
- 2,000,000 HTTP or HTTPS requests
- 2,000,000 CloudFront Function invocations
- Each month for one year

Benefits and features

Reduce latency

The CloudFront network has 225+ points of presence (PoPs) that are connected by fully redundant, parallel 100 GbE fiber delivering ultra-low latency performance and high availability to your end users.

Improve security

Use CloudFront for perimeter protection, traffic encryption, and access controls. AWS Shield Standard defends traffic transmitted through CloudFront from DDoS attacks at no additional charge. For application

Create distribution

Origin

Origin domain
Choose an AWS origin, or enter your origin's domain name.

awseb-AWSEB-1MLY0BBUYSGCE-542738315.us-east-2.elb.amazonaws.com

Origin path - *optional* Info
Enter a URL path to append to the origin domain name for origin requests.

Enter the origin path

Name
Enter a name for this origin.

awseb-AWSEB-1MLY0BBUYSGCE-542738315.us-east-2.elb.amazonaws.com

Add custom header - *optional*
CloudFront includes this header in all requests that it sends to your origin.

Add header

Enable Origin Shield Info
Origin Shield is an additional caching layer that can help reduce the load on your origin and help protect its availability.

- (•) No
- ( ) Yes

Default cache behavior

Path pattern Info

Default (*)

Compress objects automatically Info

- ○ No
- ◉ Yes

Viewer

Viewer protocol policy

- ○ HTTP and HTTPS
- ◉ Redirect HTTP to HTTPS
- ○ HTTPS only

Allowed HTTP methods

- ○ GET, HEAD
- ○ GET, HEAD, OPTIONS
- ◉ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Cache HTTP methods

GET and HEAD methods are cached by default.

- ☐ OPTIONS

Restrict viewer access

If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content.

- ◉ No

Cache key and origin requests

We recommend using a cache policy and origin request policy to control the cache key and origin requests.

- (•) Cache policy and origin request policy (recommended)
- ( ) Legacy cache settings

Cache policy

Choose an existing cache policy or create a new one.

CachingOptimized

Create policy View policy

Origin request policy - *optional*

Choose an existing origin request policy or create a new one.

Select origin policy

Create policy

▶ Additional settings

Function associations - *optional* Info

Choose an edge function to associate with this cache behavior, and the CloudFront event that invokes the function.

	Function type	Function ARN / Name	Include body
Viewer request	No association		
Viewer response	No association		

Create cache policy

Details

Name
Enter a name for the cache policy.
pizza-luvrs-ca

Description - *optional*
Enter a description for the cache policy.

TTL settings Info

Minimum TTL
Minimum time to live in seconds.
1

Maximum TTL
Maximum time to live in seconds.
31536000

Default TTL
Default time to live in seconds.
86400

Cache key settings Info

1 | 31536000 | 86400

Cache key settings Info

Headers
Choose which headers to include in the cache key.
None

Query strings
Choose which query strings to include in the cache key.
All

Cookies
Choose which cookies to include in the cache key.
Include specified cookies

Allow
Add the name of the cookie to the cache key.
pzz4lyfe Remove

Add item

Compression support Info

Cache policy created.

pizza-luvrs-cache

Edit Delete

Details

Description	ID	Last modified
-	c1f4a884-dbbf-450c-ae17-36fa33af197a	September 12, 2021 at 8:31:22 PM UTC

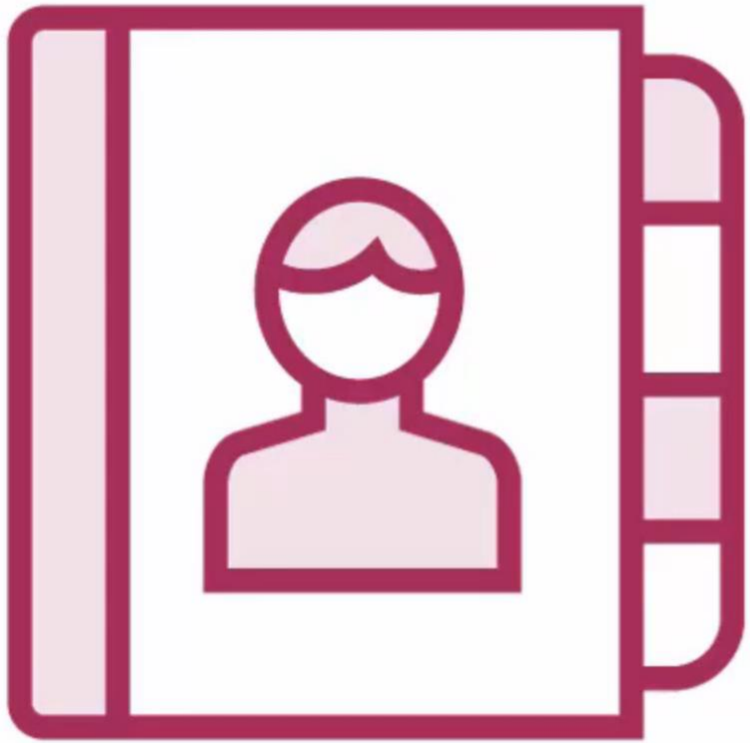
TTL settings Info

Minimum TTL (seconds)	Maximum TTL (seconds)	Default TTL (seconds)
1	31536000	86400

Cache key settings Info

Headers - None	Cookies - Include specified cookies	Query strings - All
	pzz4lyfe	

The Phone Book Analogy



Name: John Smith

Phone: 555-0100

Address: 123 Fictitious Street

Spouse: Jill Smith

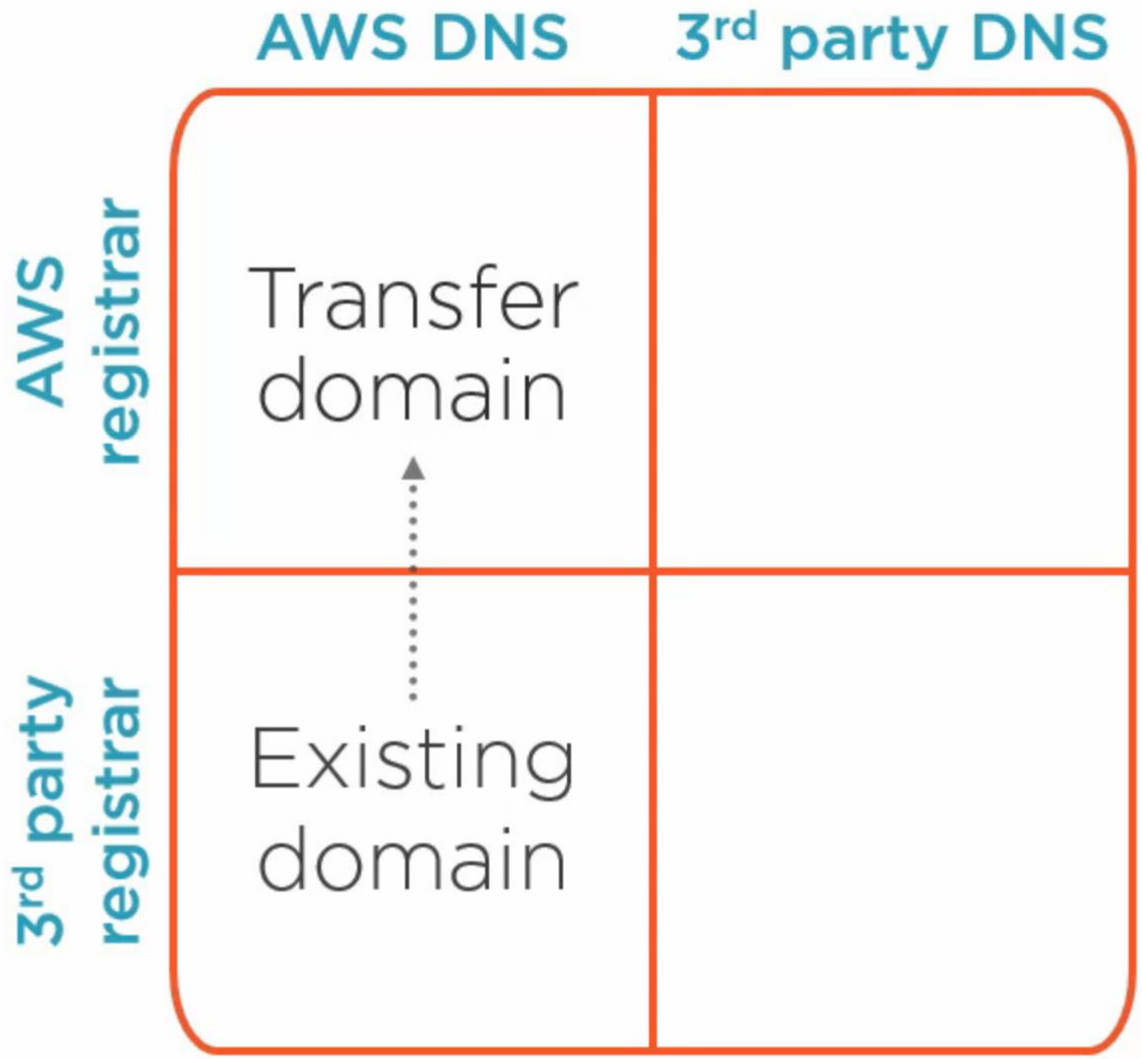
DNS Is a Database

Domain name	Resource type	Resource data
example.com	A (IP address)	93.184.216.34

Included in a resource record

Domain name

The list of labels on the path from the node to the root of the tree (RFC 1034)


AWS DNS
3rd party DNS
AWS registrar
3rd party registrar
Transfer domain
Existing domain

Domain Name Transfer Checklist

Remove transfer lock

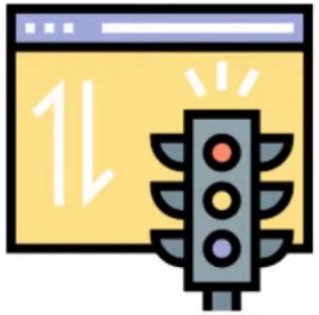
Obtain authorization code

Disable WHOIS privacy protection

Domain transfers can take up to 5 days to complete

Amazon Route 53 performs three main functions in any combination

1. If a website needs a name, Route 53 registers the name for the website (domain name)



2. Route 53 helps to connect the browser with the website or web application when the user enter the domain name

Route 53 checks health of resources by sending automated requests over the internet to a resource

Highly reliable – Ensures a consistent ability to route applications

Scalable - Automatically handles large queries without the user's interaction

Easy to use – Easy to sign up, configure DNS settings and provides fast response to queries

Cost effective - Pay only for the service used

Secure – The user secures the access rights by integrating Route 53 with AWS(IAM)

Simple Routing

- Simple Routing allows to configure DNS with no special Route 53 Routing
- IT routes traffic to a single resource, Ex- web server to a website
- With simple routing, multiple records with same name cannot be created but, multiple values can be specified in the same record

Failover Routing

- Failover Routing routes traffic to a resource when the resource is healthy or to a different resource when the previous resource is unhealthy
- The records can route traffic to anything from an Amazon S3 bucket as a website to a complex tree of records

Geolocation Routing

- Geolocation Routing routes the resources that based on the geographic location of the users
- It localizes the content and presents part or the entire website in the language of the user
- Geographic locations are specified by continent, by country, or by state in the United States

Traffic Flow
It routes end users to the endpoint that should provide the best user experience

Domain Registration
selects domain names and registers them with the AWS console

Health Checks
It monitors health and performance of applications

Weighted round-robin load balancing
It spreads traffic between several services via a round-robin algorithm

Amazon Route 53 can be accessed in the following ways:

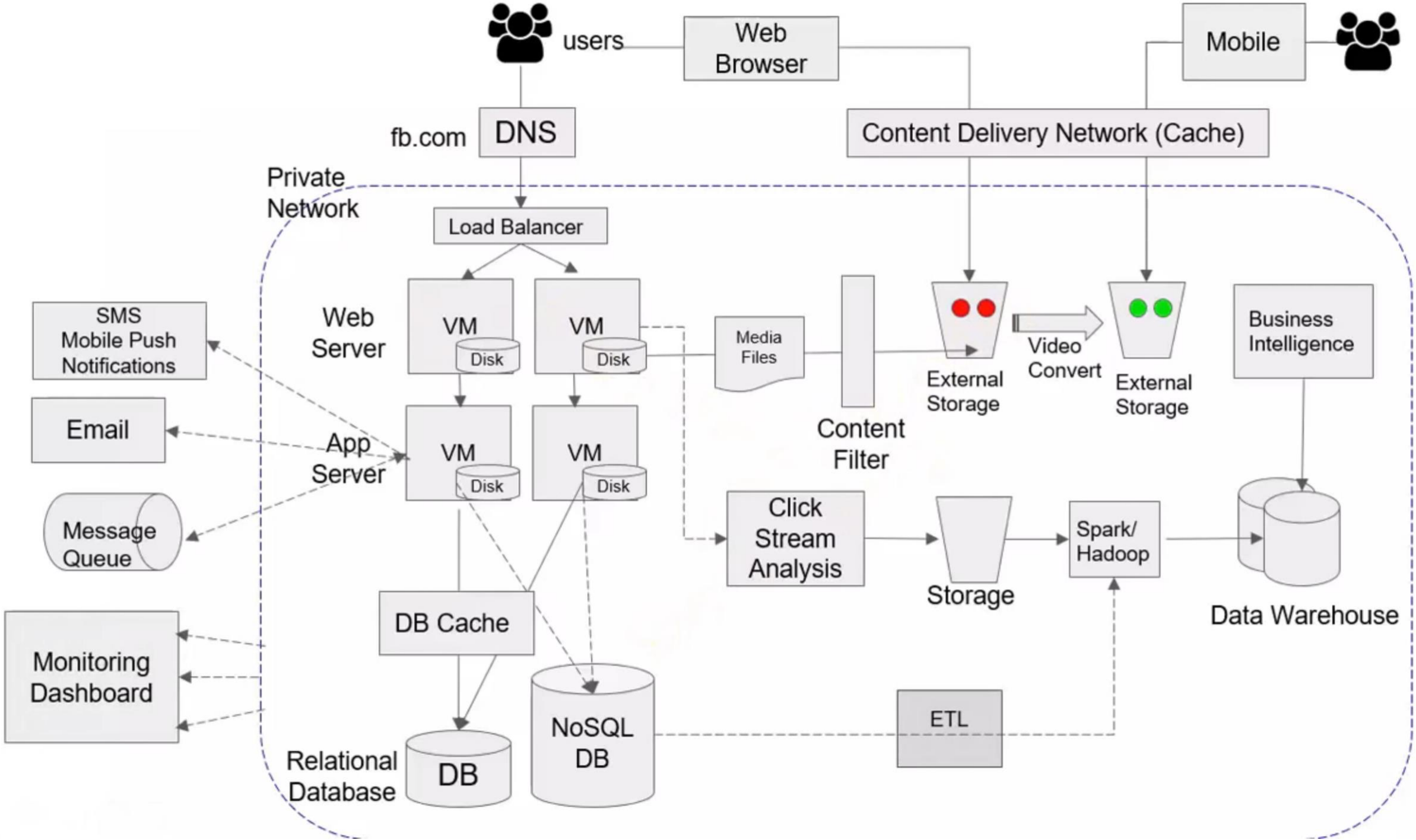
AWS Management Console

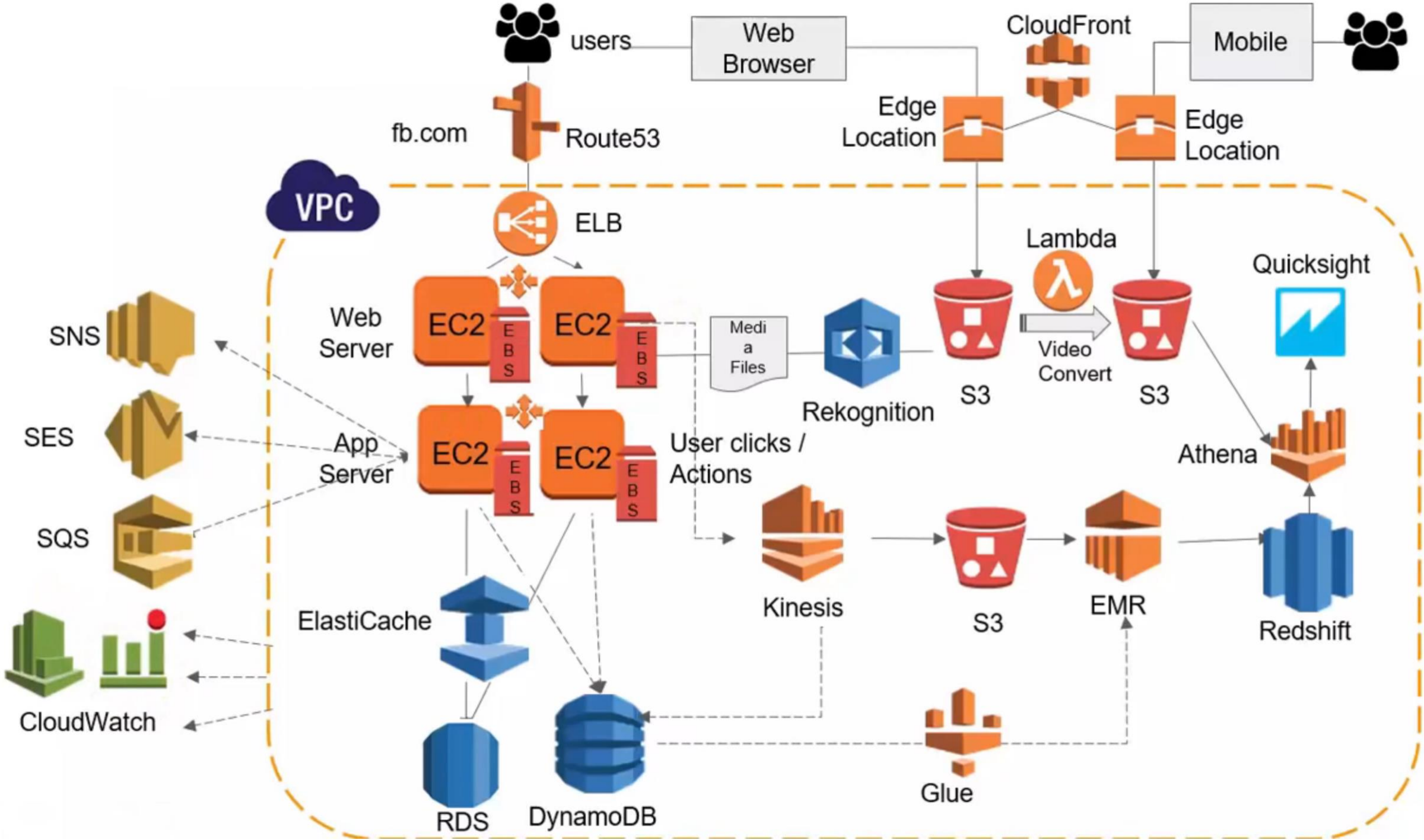
AWS SDKs

Route 53 API

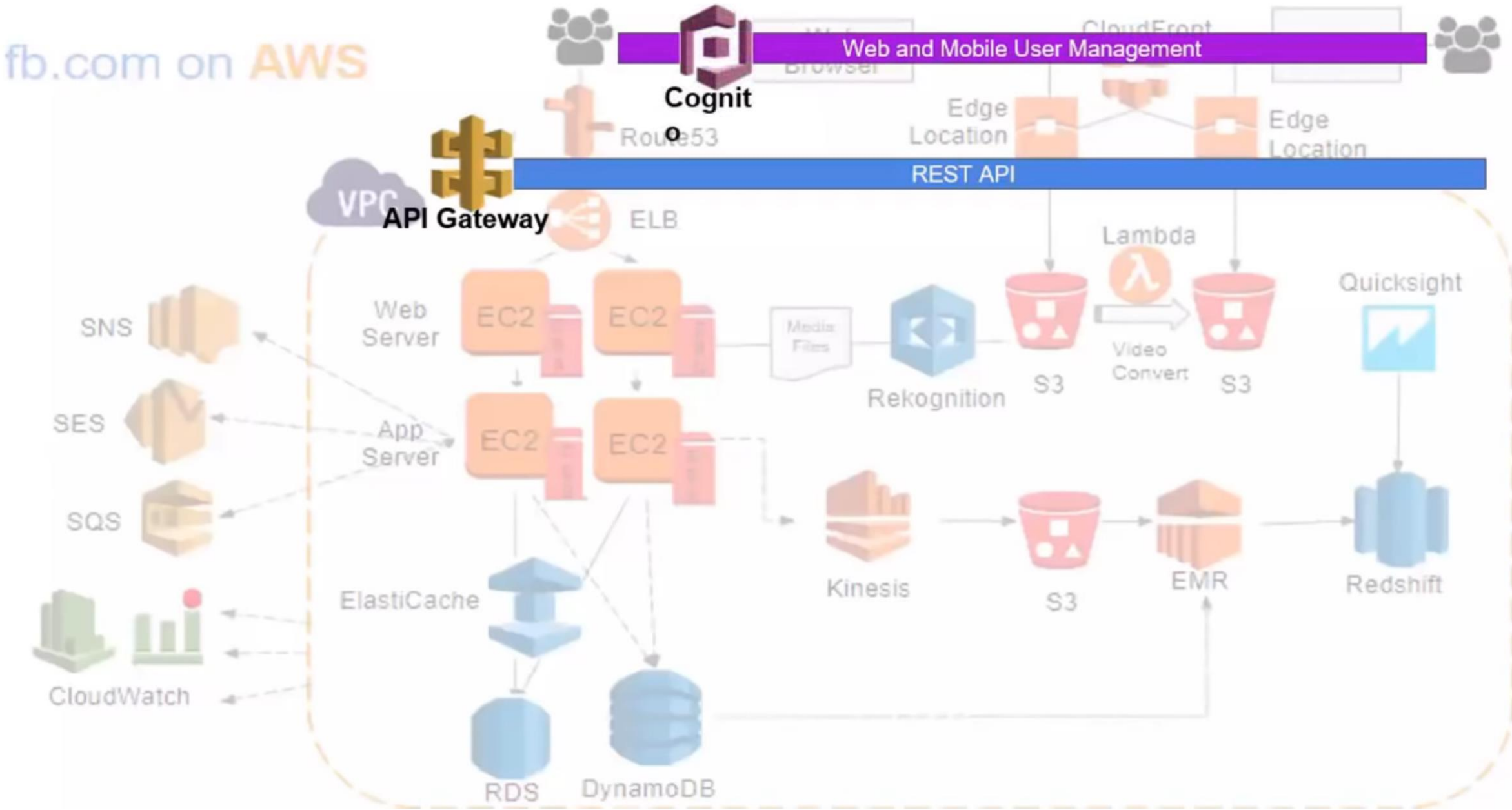
AWS Command Line Interface

AWS Tools for Windows PowerShell

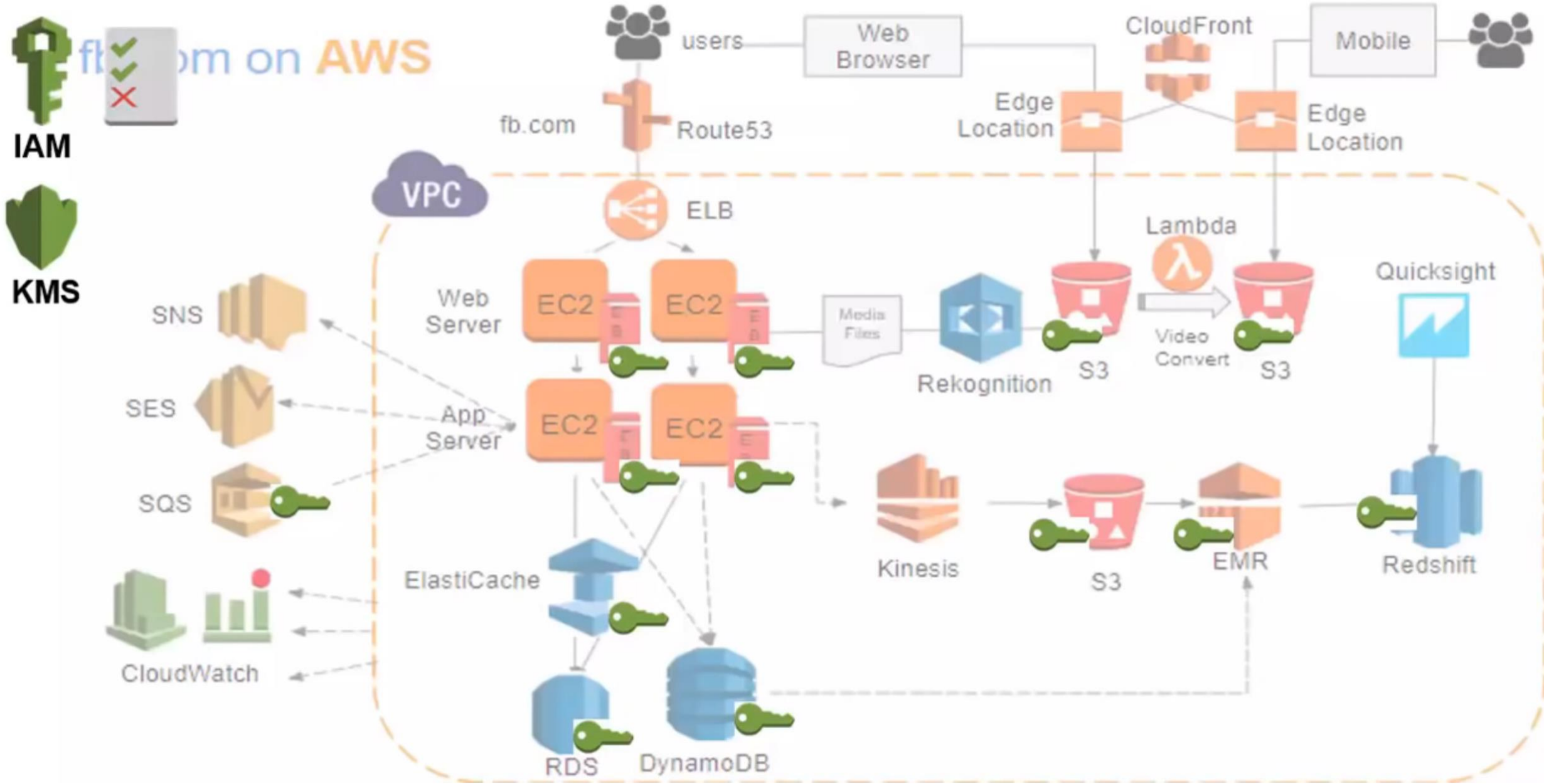

users
Web Browser
Mobile
fb.com
DNS
Content Delivery Network (Cache)
Private Network
Load Balancer
Web Server
VM
Disk
VM
Disk
Media Files
Content Filter
External Storage
Video Convert
External Storage
Business Intelligence
SMS Mobile Push Notifications
Email
App Server
VM
Disk
VM
Disk
Message Queue
Click Stream Analysis
Storage
Spark/ Hadoop
Data Warehouse
DB Cache
Monitoring Dashboard
NoSQL DB
ETL
Relational Database
DB


users
Web Browser
CloudFront
Mobile
fb.com
Route53
Edge Location
Edge Location
VPC
ELB
Lambda
Quicksight
Web Server
EC2
EC2
EBS
EBS
Medi a Files
Rekognition
S3
Video Convert
S3
SNS
SES
App Server
EC2
EC2
EBS
EBS
User clicks / Actions
Athena
SQS
ElastiCache
Kinesis
S3
EMR
Redshift
CloudWatch
RDS
DynamoDB
Glue

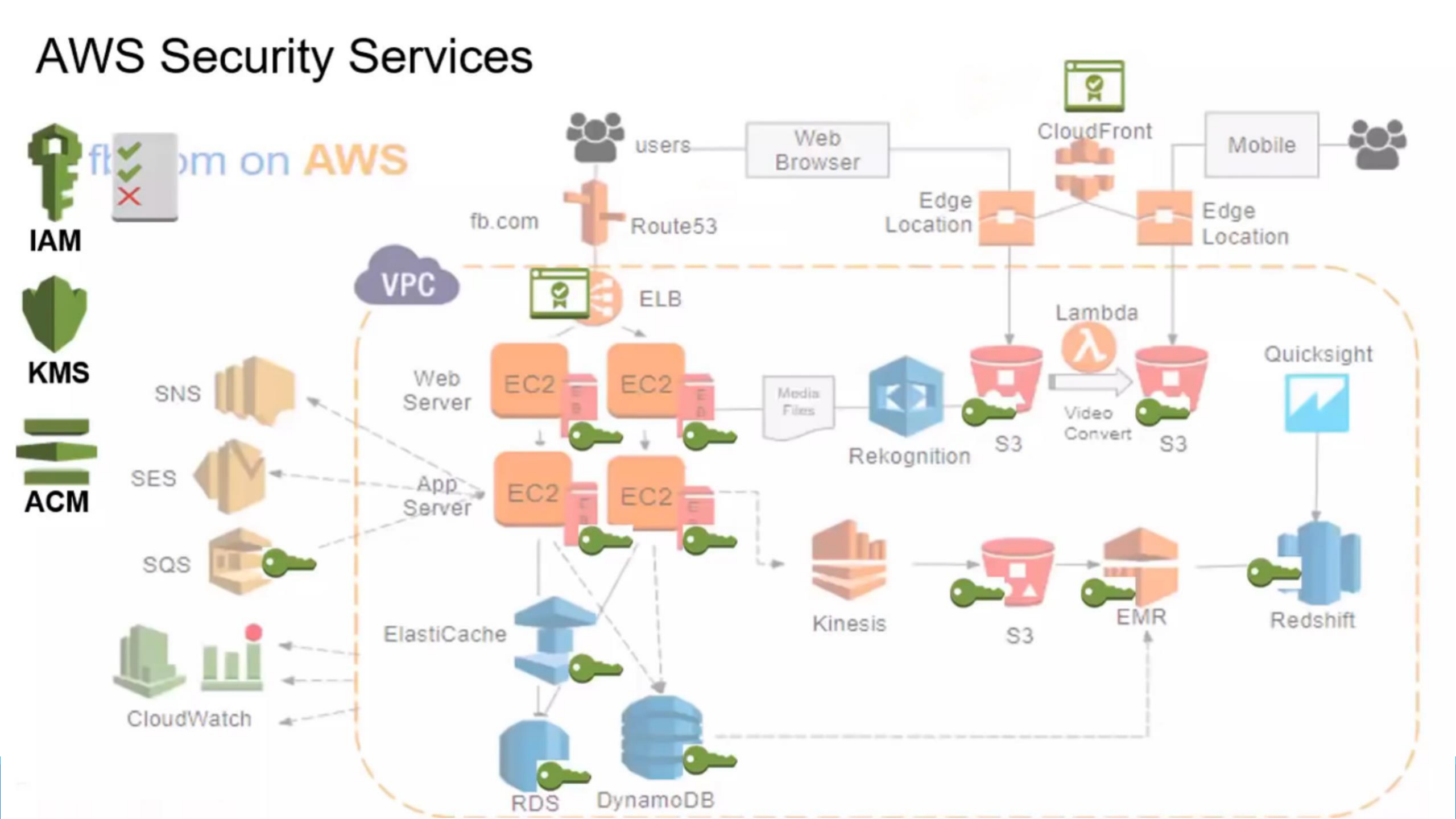
AWS Application Services



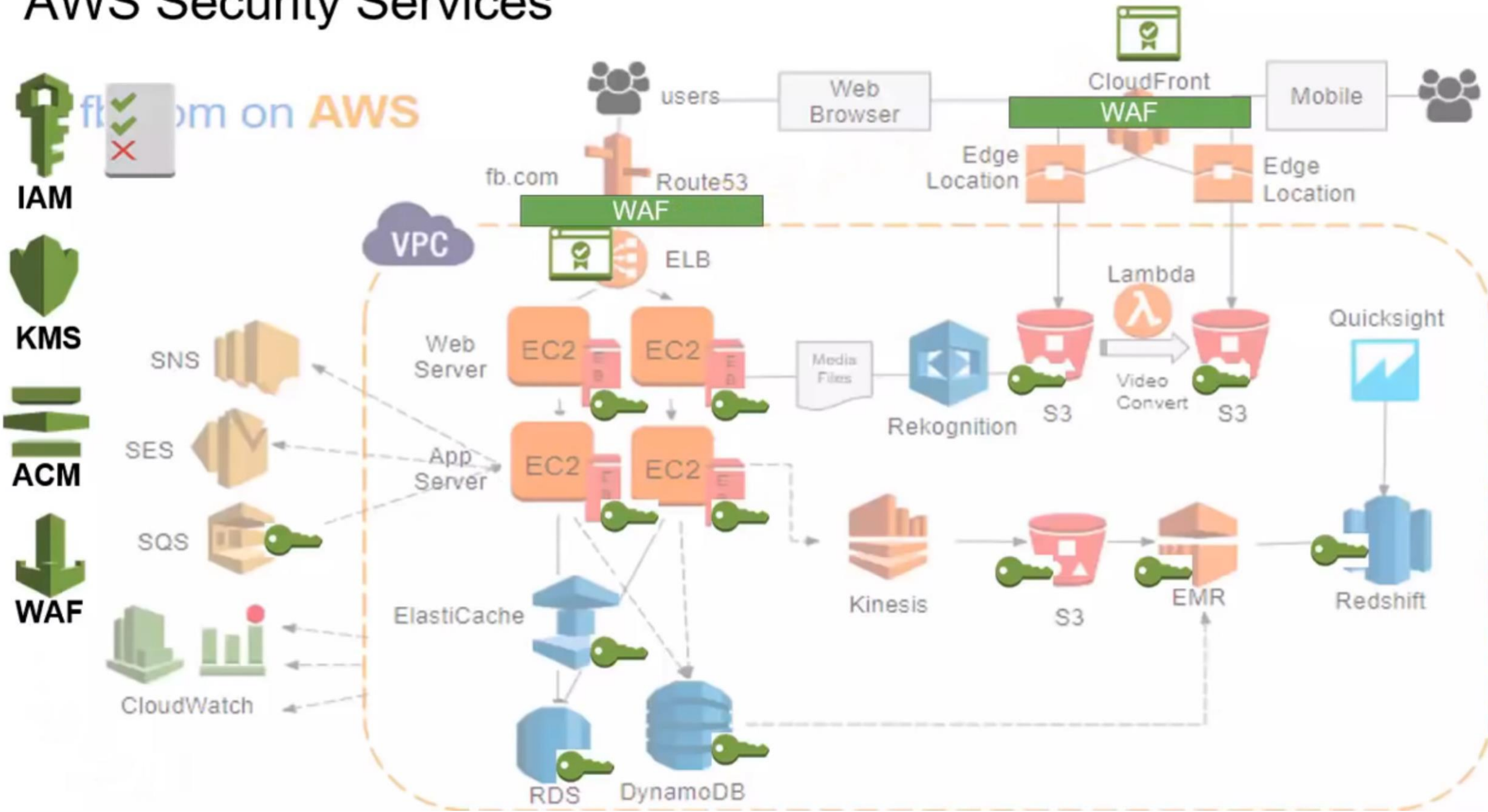
AWS Security Services



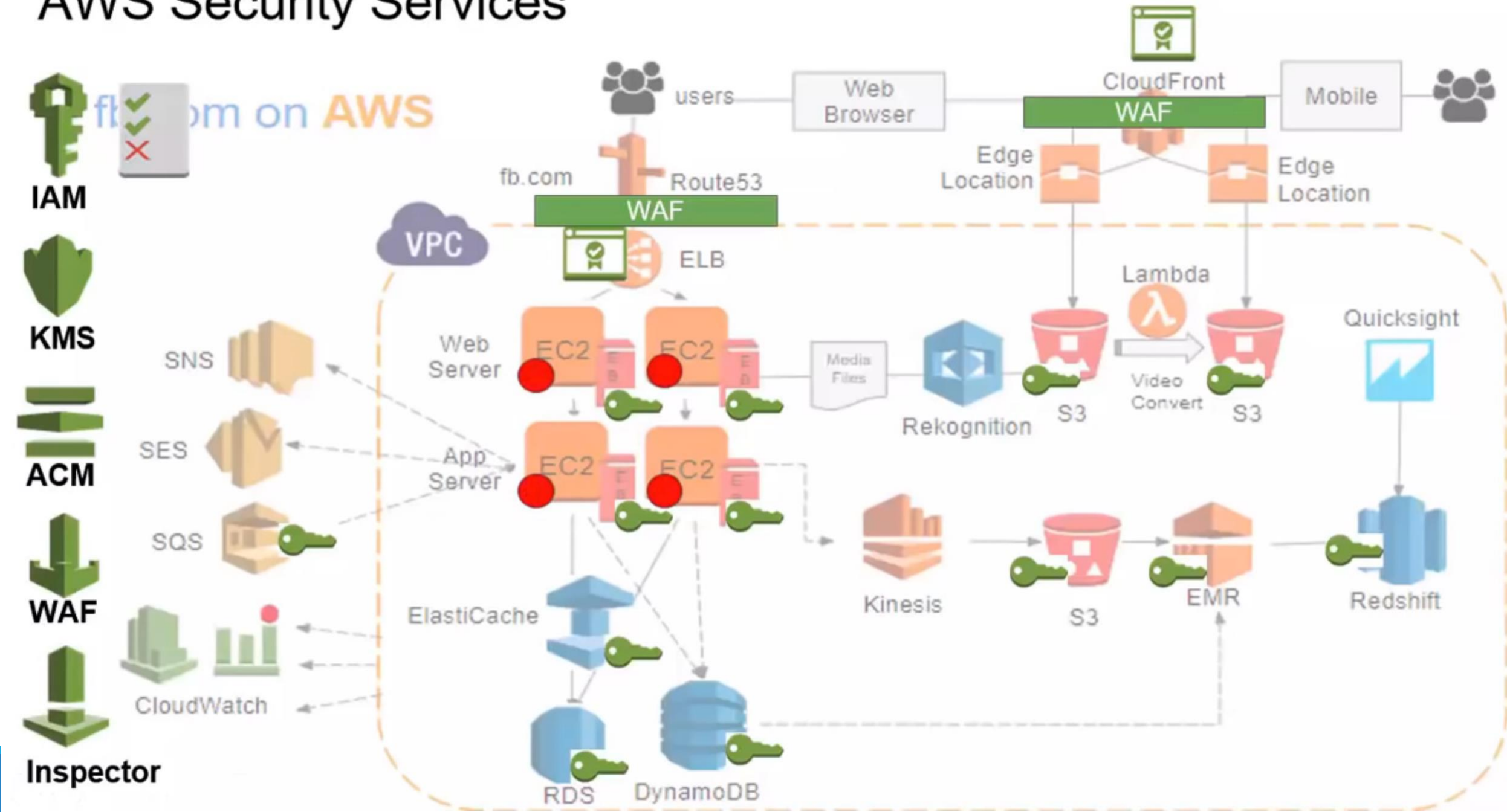
AWS Security Services



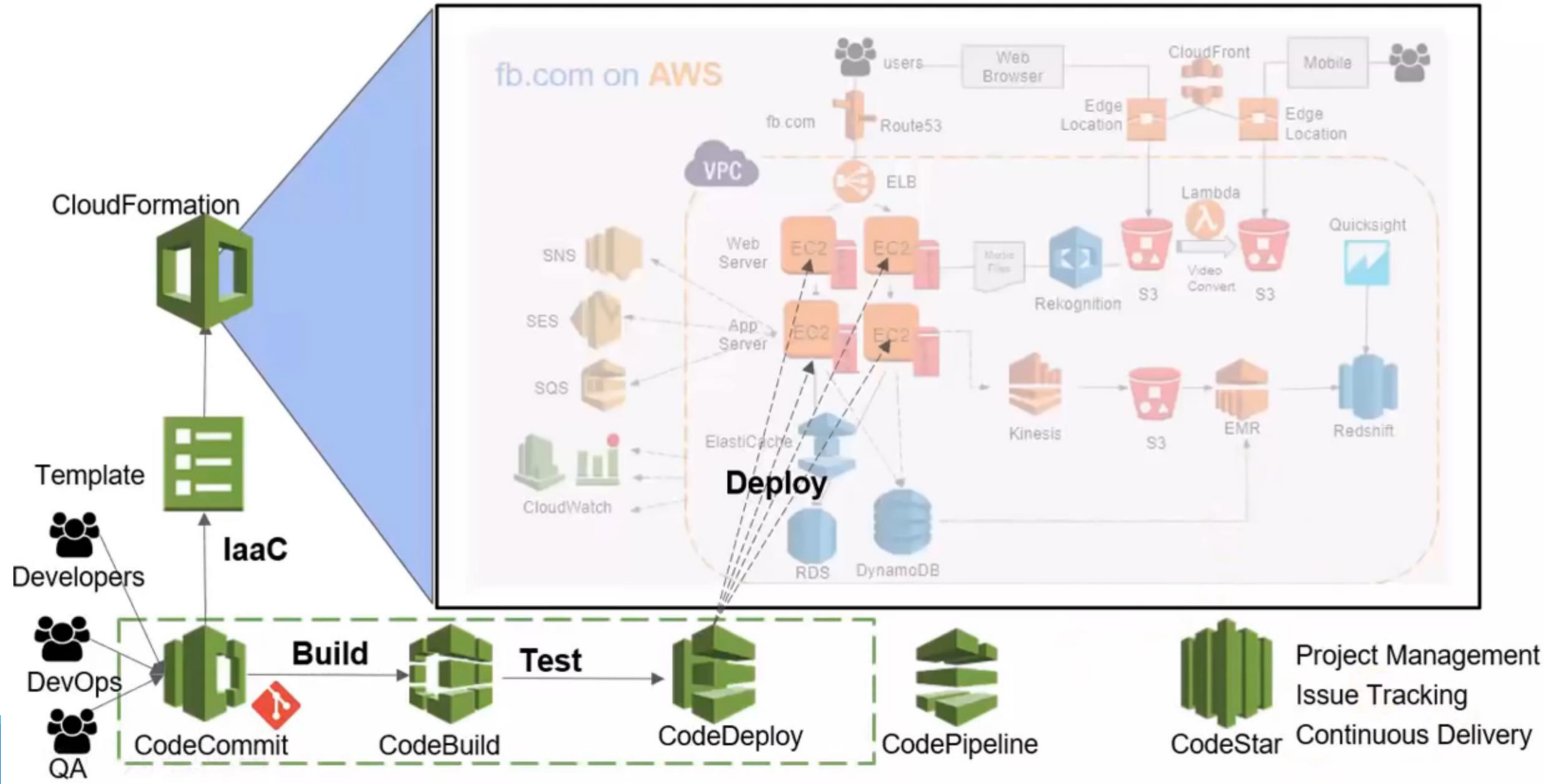
AWS Security Services



AWS Security Services



AWS Development and DevOps Services



Serverless vs. Traditional Architecture

1950s-1970s

IBM and others

Mainframes

1977

Apple and others

Personal Computing

2005

VMware

OS Virtualization

2006

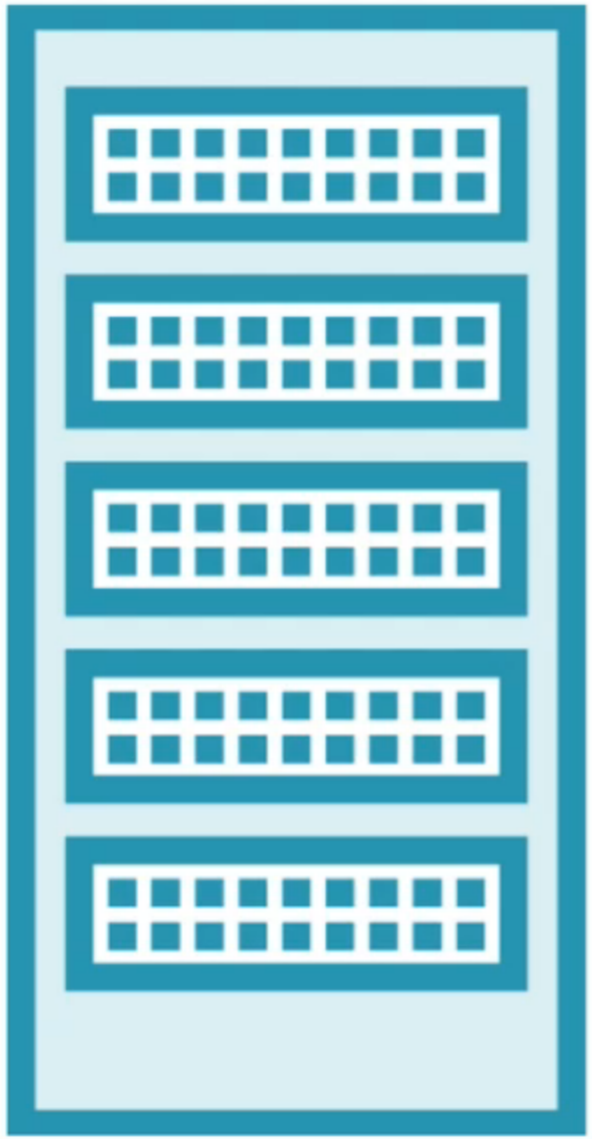
AWS EC2/S3

The Cloud

2014

AWS Lambda 2014

Serverless



Mainframes

- Large space requirement
- Installation
- Maintenance
- Cost
- Inflexibility

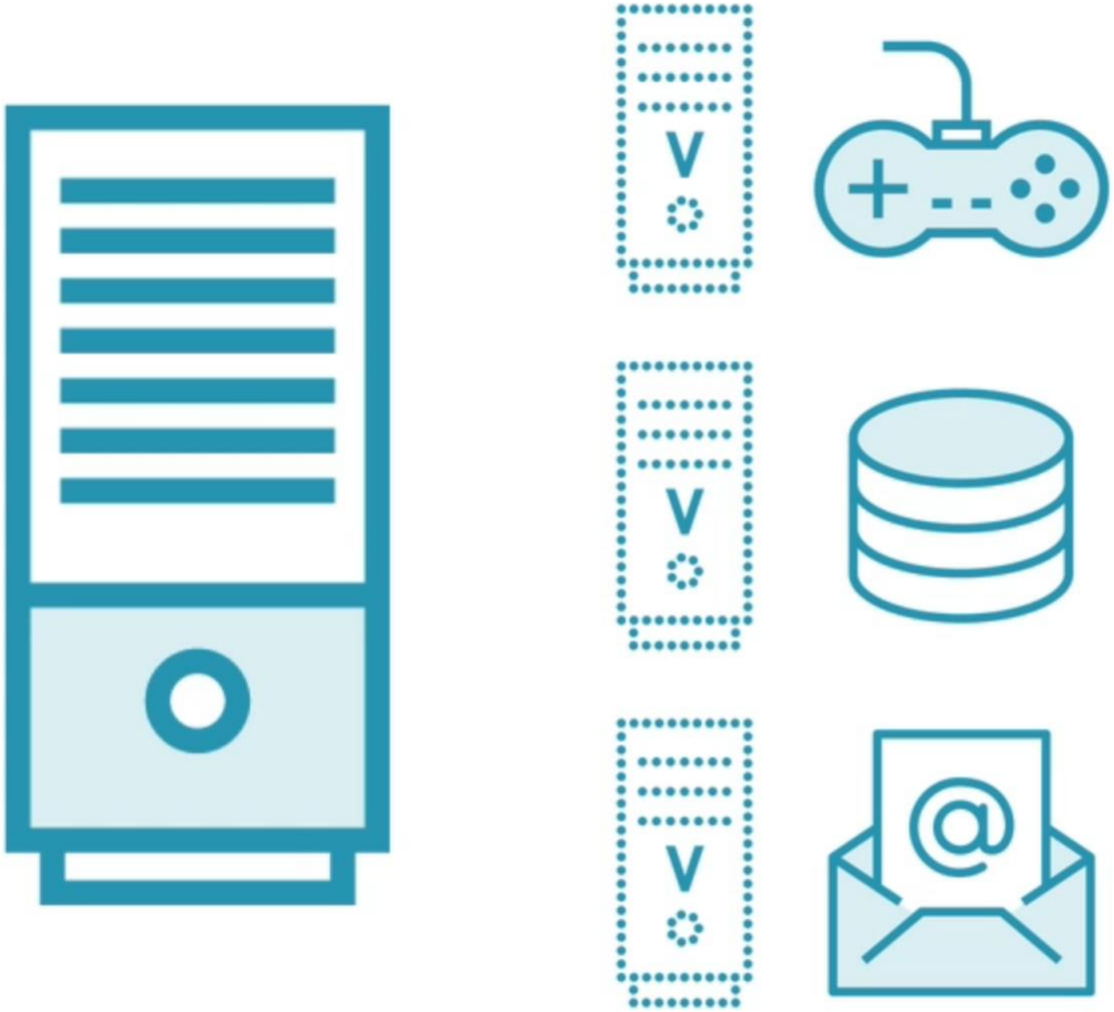
Personal Computing / Minicomputers

Lower barriers to entry

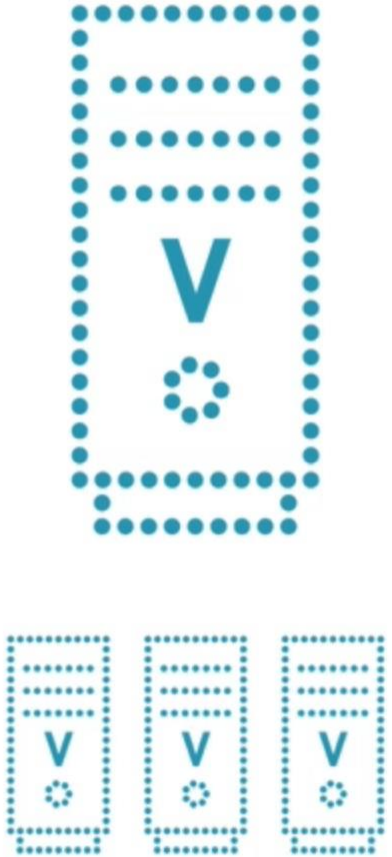
Reduced cost

Increased distribution

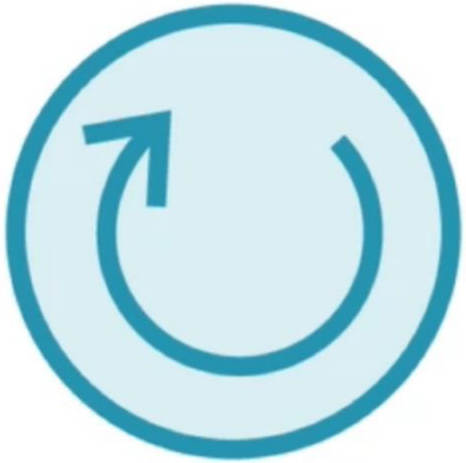
Virtualization and Hypervisors



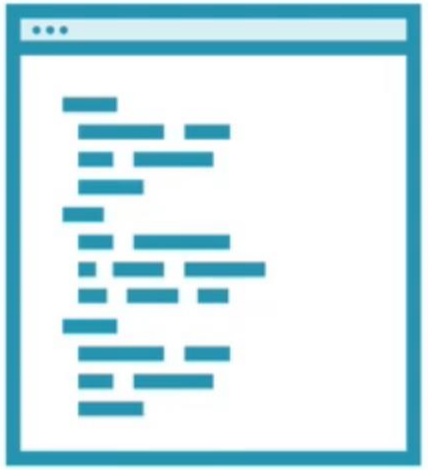
The Cloud - Amazon EC2



Serverless Functions



Event driven

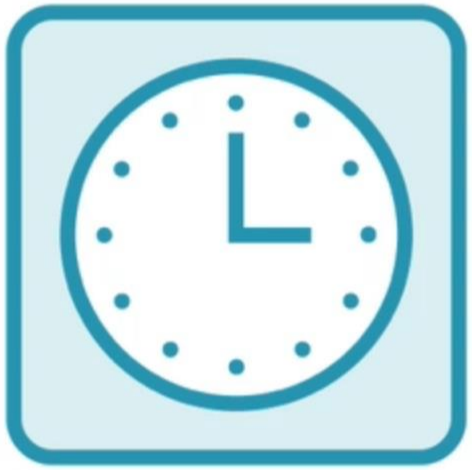


Code focused

Managed machines

Event Examples

File uploads



Scheduled times

API requests

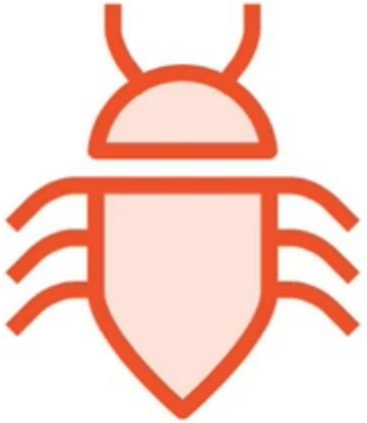
Serverless Benefits and Challenges

Benefits

- Cost and utilization
- Managed machines
- Service integrations
- Scaling

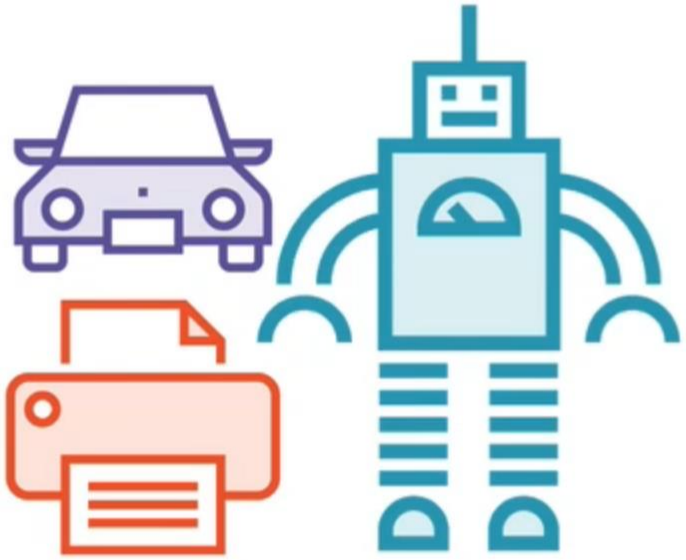
Challenges

- Debugging
- Control
- Cutting edge quirks



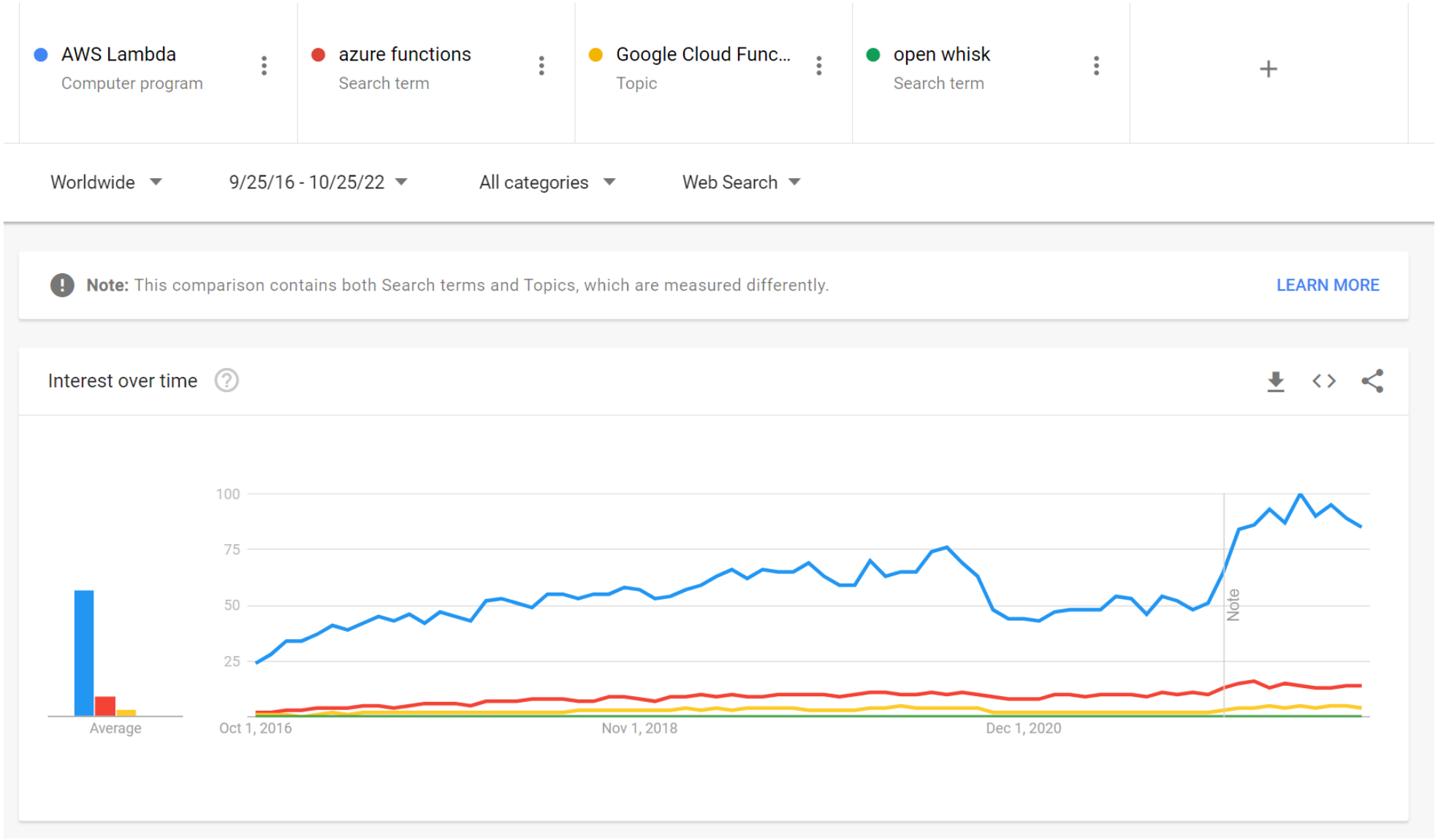
Why Learn Lambda?

Managed infrastructure

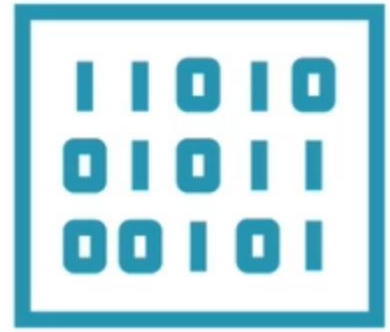


Internet of Things

Growing relevance


AWS Lambda
Computer program
azure functions
Search term
Google Cloud Func...
Topic
open whisk
Search term
Worldwide
9/25/16 - 10/25/22
All categories
Web Search
Note: This comparison contains both Search terms and Topics, which are measured differently.
LEARN MORE
Interest over time
100
75
50
25
Note
Average
Oct 1, 2016
Nov 1, 2018
Dec 1, 2020

How Is Lambda Used?



Stream data processing

Easy & scaleable APIs

Photo processing

Web applications

Prominent Serverless Function Providers

Market leaders:
AWS, Microsoft Azure

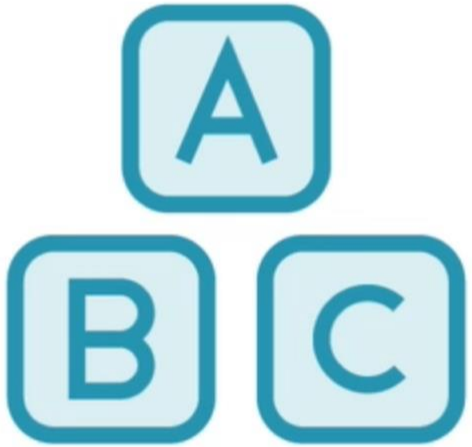
Other players:
iron.io, Cloudflare, OpenFaaS

Market Leader Comparison

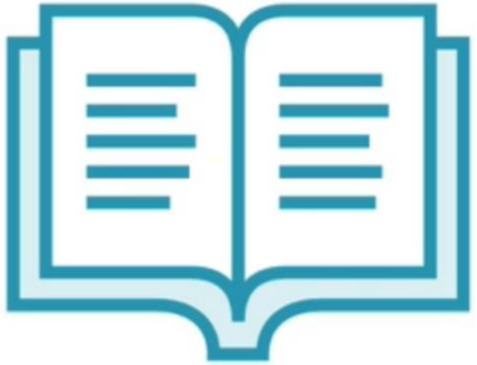
AWS Lambda	Azure Functions
Node, Python, Java, C#, PowerShell, Ruby, Go, user-provided runtimes	Node, Python, Java, C#, PowerShell, F# PHP, batch, bash, other executables
Built-in versioning	No built-in versioning
HTTP endpoints via API Gateway	HTTP endpoints via API Management
15 minute running time limit	10 minute limit (option for unlimited)
1000 concurrent functions (soft limit)	10 concurrent instances

Niche Providers

Iron.io



Cloudflare Workers



OpenFaaS

Globomantics Pet Care - Our Client

Needs:

Website uptime monitoring

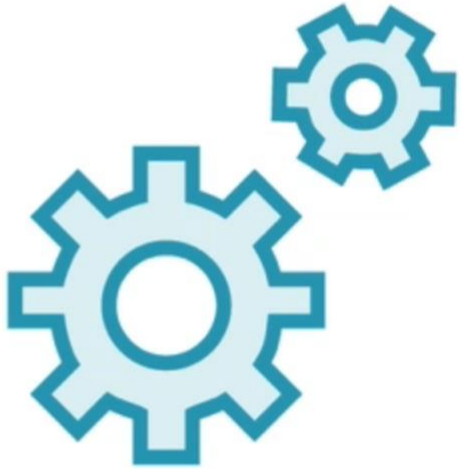
Social media automation

Custom business reminders

New customer service

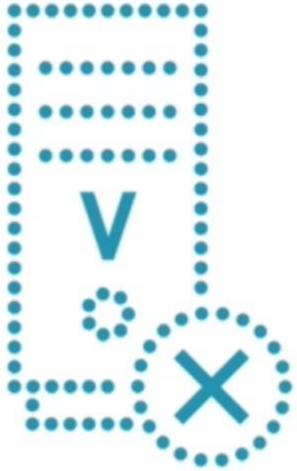
Our Four Lambda Projects

Simple
scheduled events



3rd party APIs

Business logic
and AWS SES



A full-fledged
serverless service

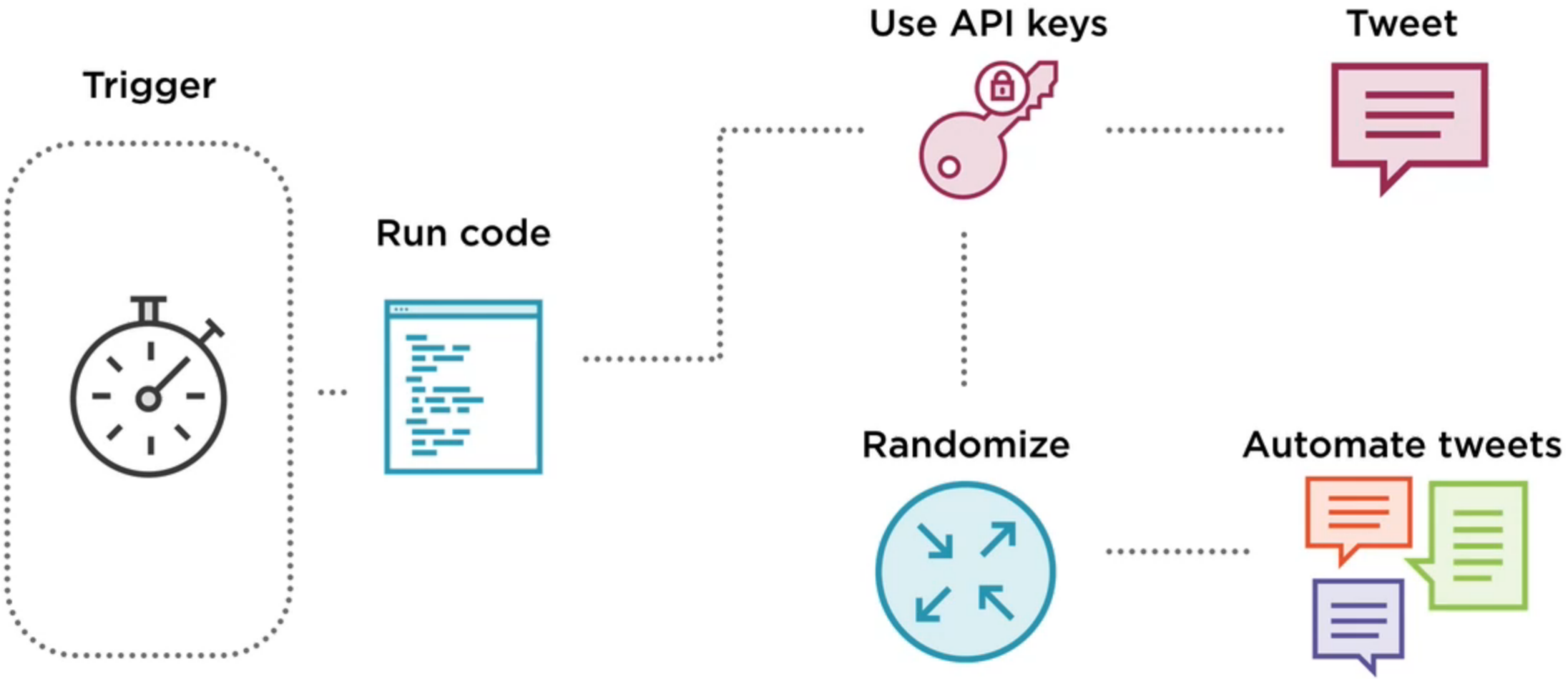
Lambda Canary

Set run interval

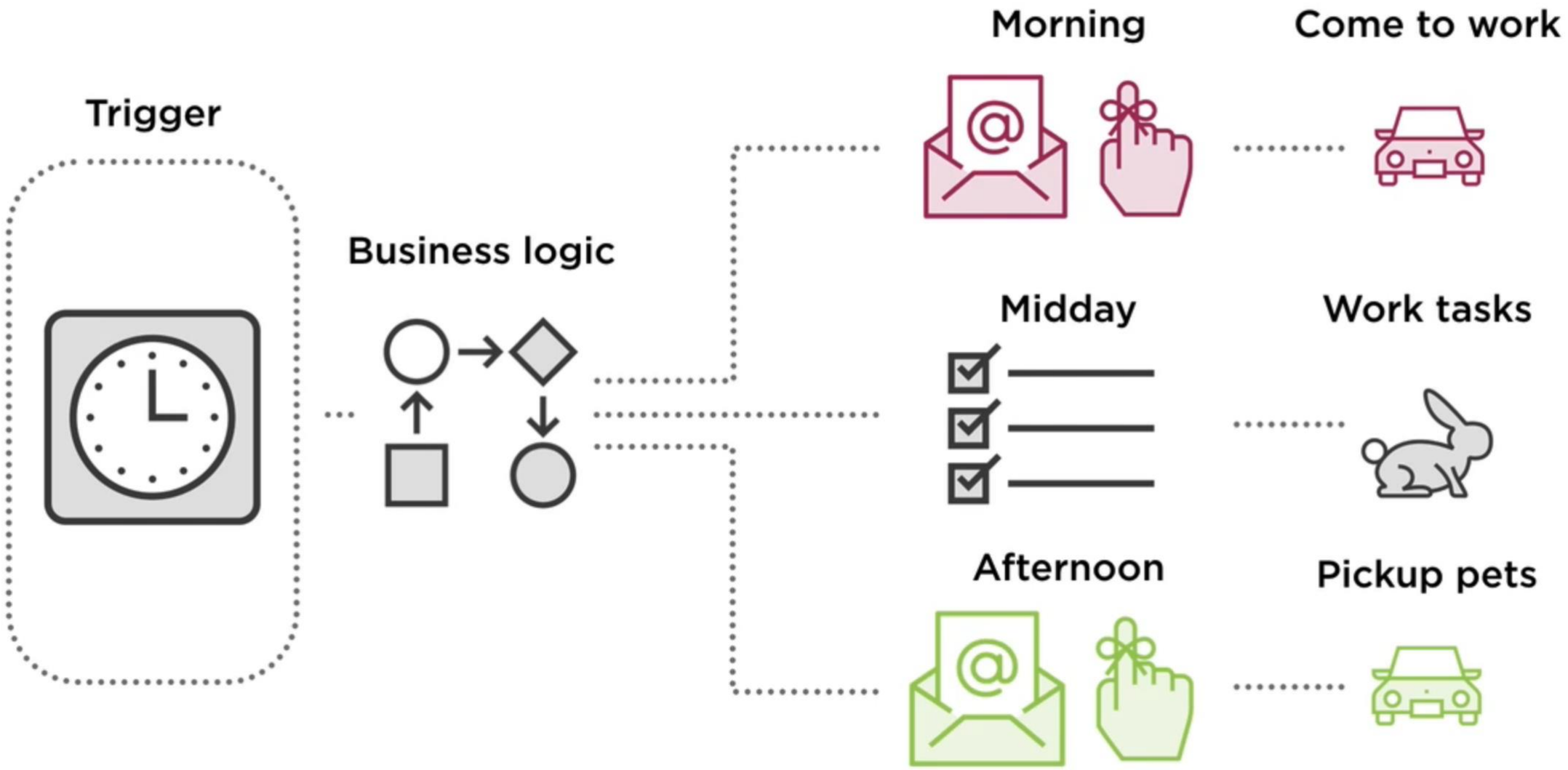
Function reviews website

Website status recorded

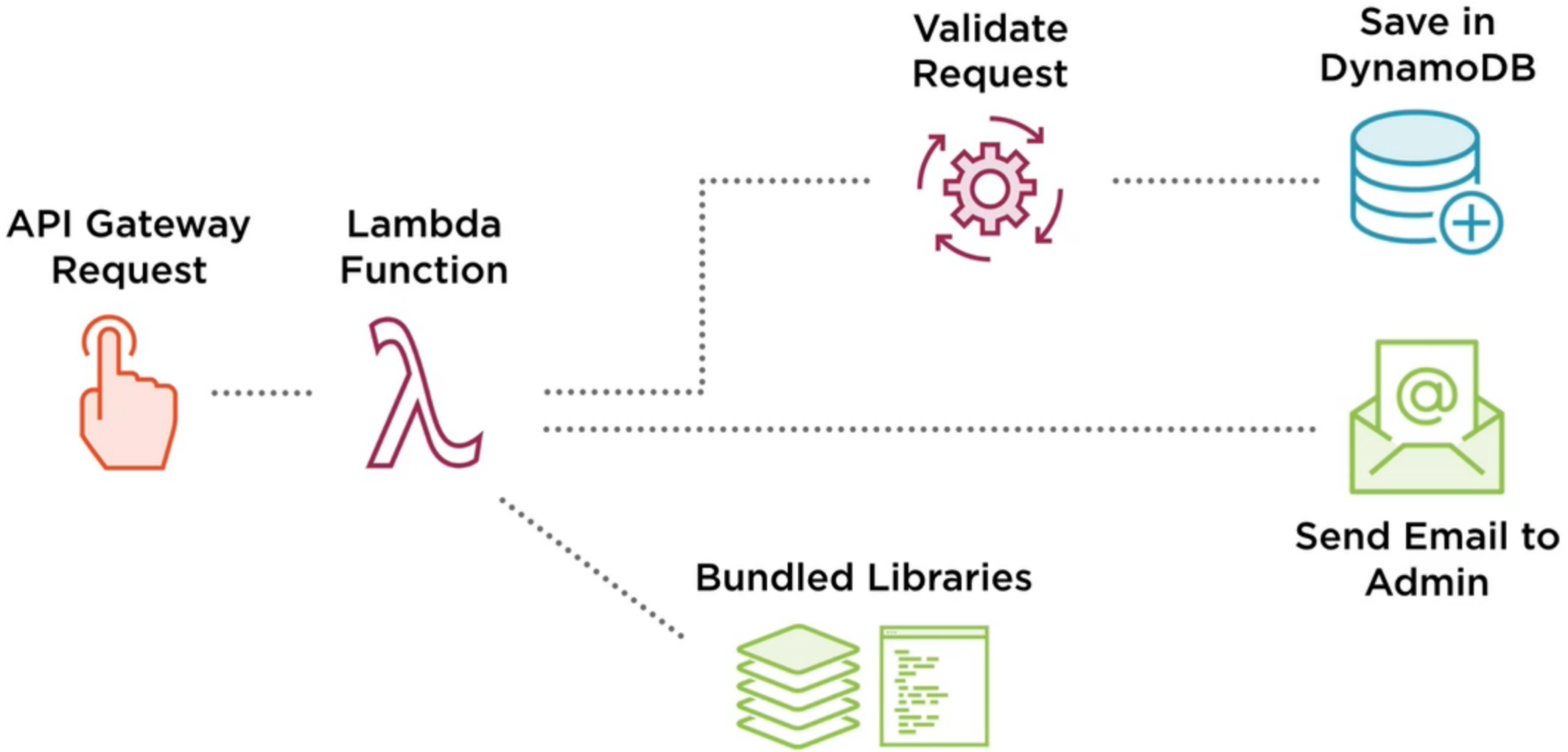
Twitter Bot


Trigger
Run code
Use API keys
Tweet
Randomize
Automate tweets

Workflow Automation

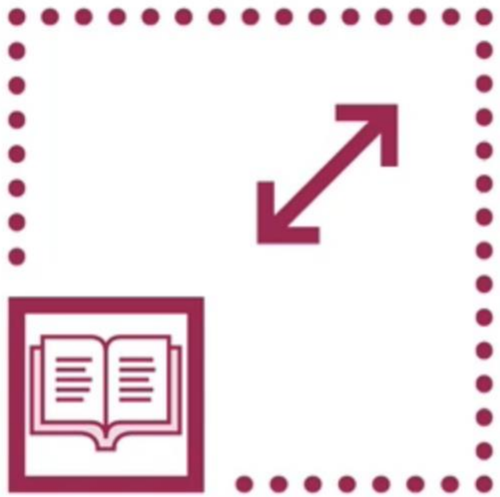


New Customers Service

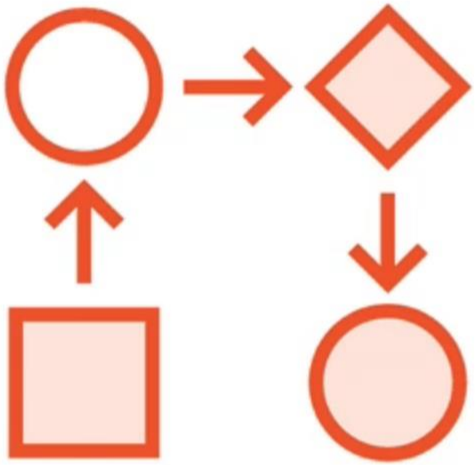


Starting with Lambda Functions

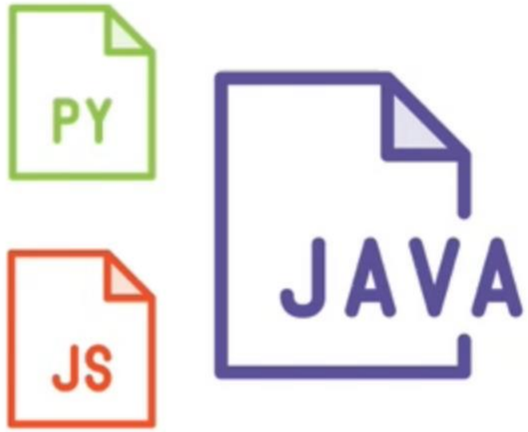
Size Limitations - Workarounds



Smaller libraries



Microservice architectures



Other runtime options

Resource Limitations

Ephemeral storage
< 512 MB

Maximum execution duration
< 900 seconds

Concurrent Lambda functions
< 1000

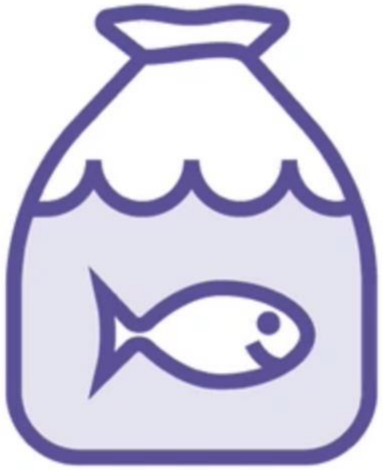
Resource Limitations - Workarounds

Load and store files in S3

Chain functions together

Ask AWS about increasing limits

Memory, CPU and VPC



Memory
128 MB - 3 GB

CPU scales with
memory

Virtual Private
Cloud

AWS Lambda Considerations

Event driven code

Code size limitations

Lambda as a component

Performance limitations

Long running workloads

Gathering Dependencies

Libraries

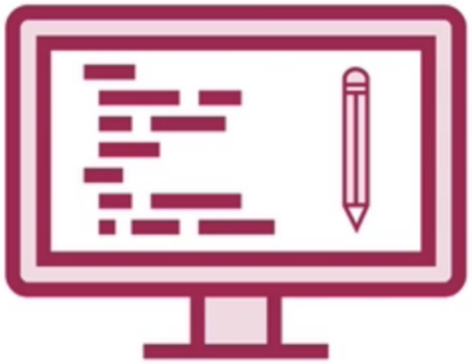
Other files

AWS services

API keys

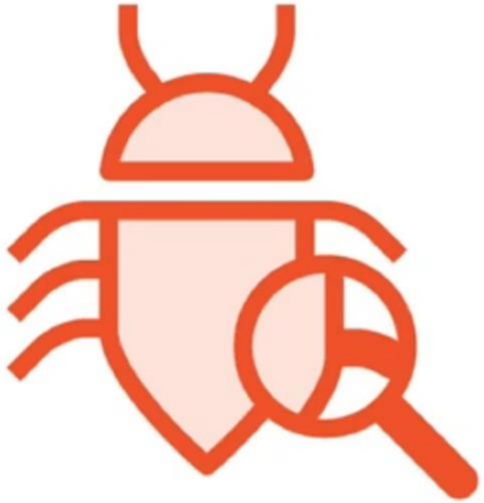
Writing Code

Locally

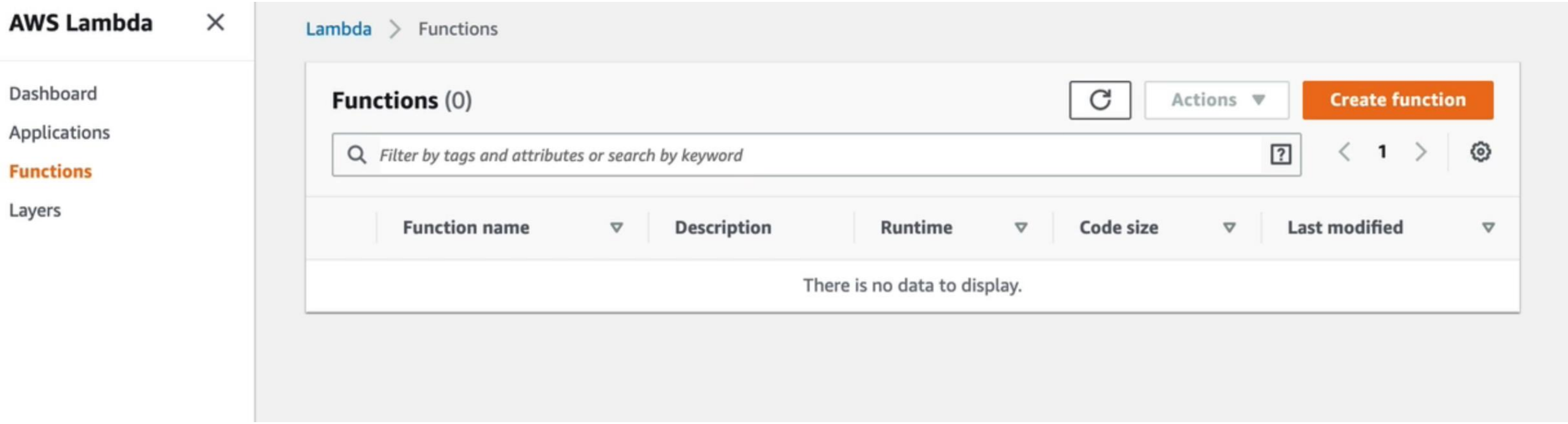


In AWS editor

Debugging



- Use logging and CloudWatch
- Environment differences
- Search for your issues
- Consider 3rd party tools


AWS Lambda
Dashboard
Applications
Functions
Layers
Lambda
Functions
Functions (0)
Actions
Create function
Filter by tags and attributes or search by keyword
1
Function name
Description
Runtime
Code size
Last modified
There is no data to display.

Create function Info

Choose one of the following options to create your function.

Author from scratch

Start with a simple Hello World example.

Use a blueprint

Build a Lambda application from sample code and configuration presets for common use cases.

Browse serverless app repository

Deploy a sample Lambda application from the AWS Serverless Application Repository.

Blueprints Info

Export

Filter by tags and attributes or search by keyword

1 2 3 4 5 6

kinesis-firehose-syslog-to-json

An Amazon Kinesis Firehose stream processor that

batch-get-job-python27

Returns the current status of an AWS Batch Job.

cloudfront-modify-response-header

Author from scratch

Start with a simple Hello World example.

Use a blueprint

Build a Lambda application from sample code and configuration presets for common use cases.

Browse serverless app repository

Deploy a sample Lambda application from the AWS Serverless Application Repository.

Blueprints Info

Export

Add filter

1

Keyword : canary

lambda-canary

Performs a periodic check of the given site, erroring out on test failure.

python3.7 · cron · testing

Cancel

Configure

Lambda > Functions > Create function > Configure blueprint lambda-canary

Basic information Info

Function name

lambda-canary

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the **IAM console**.

- (•) Create a new role with basic Lambda permissions
- ( ) Use an existing role
- ( ) Create a new role from AWS policy templates

ⓘ Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Lambda will create an execution role named lambda-canary-role-xu5e2jre, with permission to upload logs to Amazon CloudWatch Logs.

CloudWatch Events trigger

Remove

Rule
Pick an existing rule, or create a new one.

Create a new rule

Select or create a new rule

Rule name*
Enter a name to uniquely identify your rule.

canary

Rule description
Provide an optional description for your rule.

Five minute rule

Rule type
Trigger your target based on an event pattern, or based on an automated schedule.

- ○ Event pattern
- ◉ Schedule expression

Schedule expression*
Self-trigger your target on an automated schedule using Cron or rate expressions. Cron expressions are in UTC.

Lambda function code

Code is preconfigured by the chosen blueprint. You can configure it after you create the function. **Learn more** about deploying Lambda functions.

Runtime

Python 3.7

```
1  import os
2  from datetime import datetime
3  from urllib.request import Request, urlopen
4
5  SITE = os.environ['site']  # URL of the site to check, stored in the site environmen
6  EXPECTED = os.environ['expected']  # String expected to be on the page, stored in th
7
8
9  def validate(res):
10     '''Return False to trigger the canary
11
12     Currently this simply checks whether the EXPECTED string is present.
13     However, you could modify this to perform any number of arbitrary
14     checks on the contents of SITE.
15     '''
16     return EXPECTED in res
17
18
19 def lambda_handler(event, context):
20     print('Checking {} at {}...'.format(SITE, event['time']))
21     try:
```

```
26          print('Check failed!')
27          raise
```

Environment variables

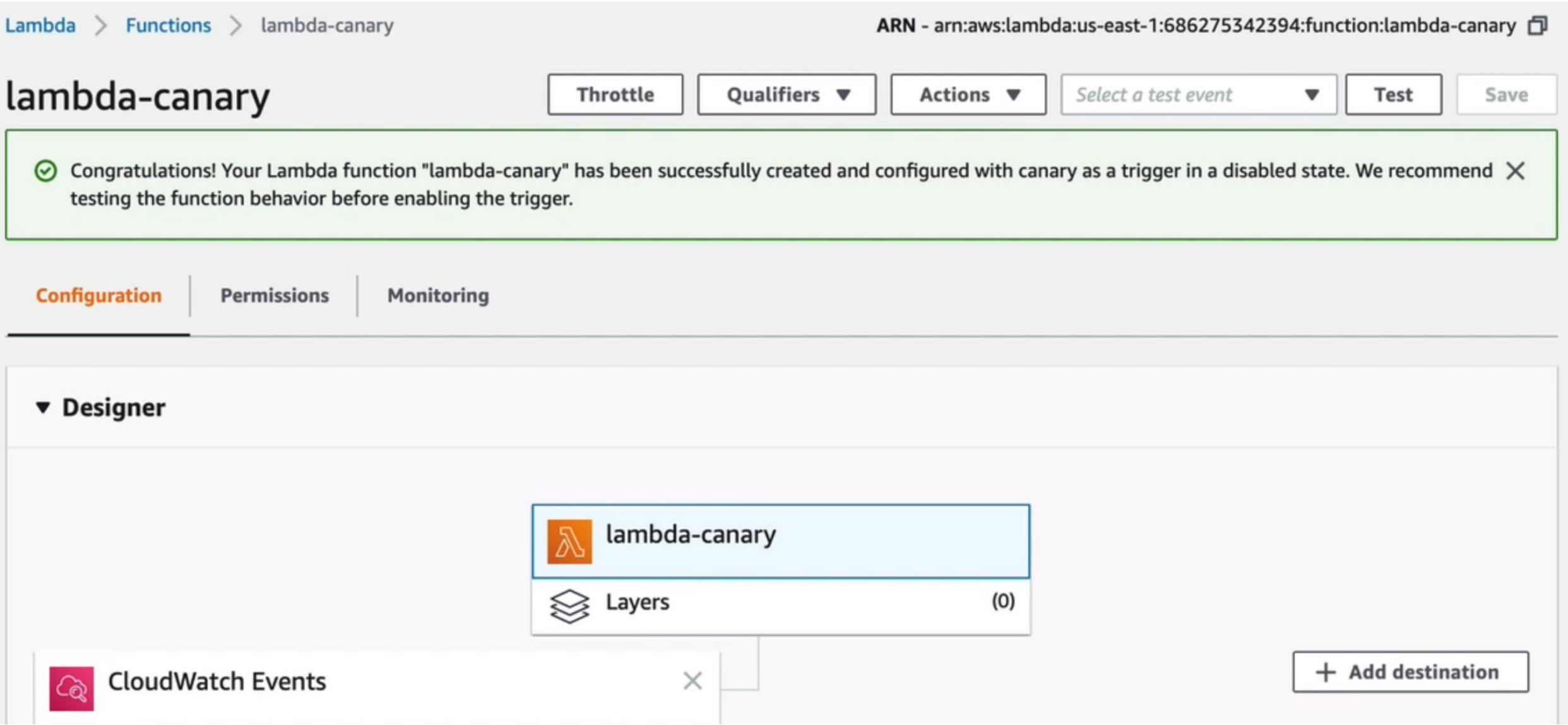
You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. Learn more

Key	Value	
site	https://www.amazon.com/	Remove
expected	Online Shopping	Remove

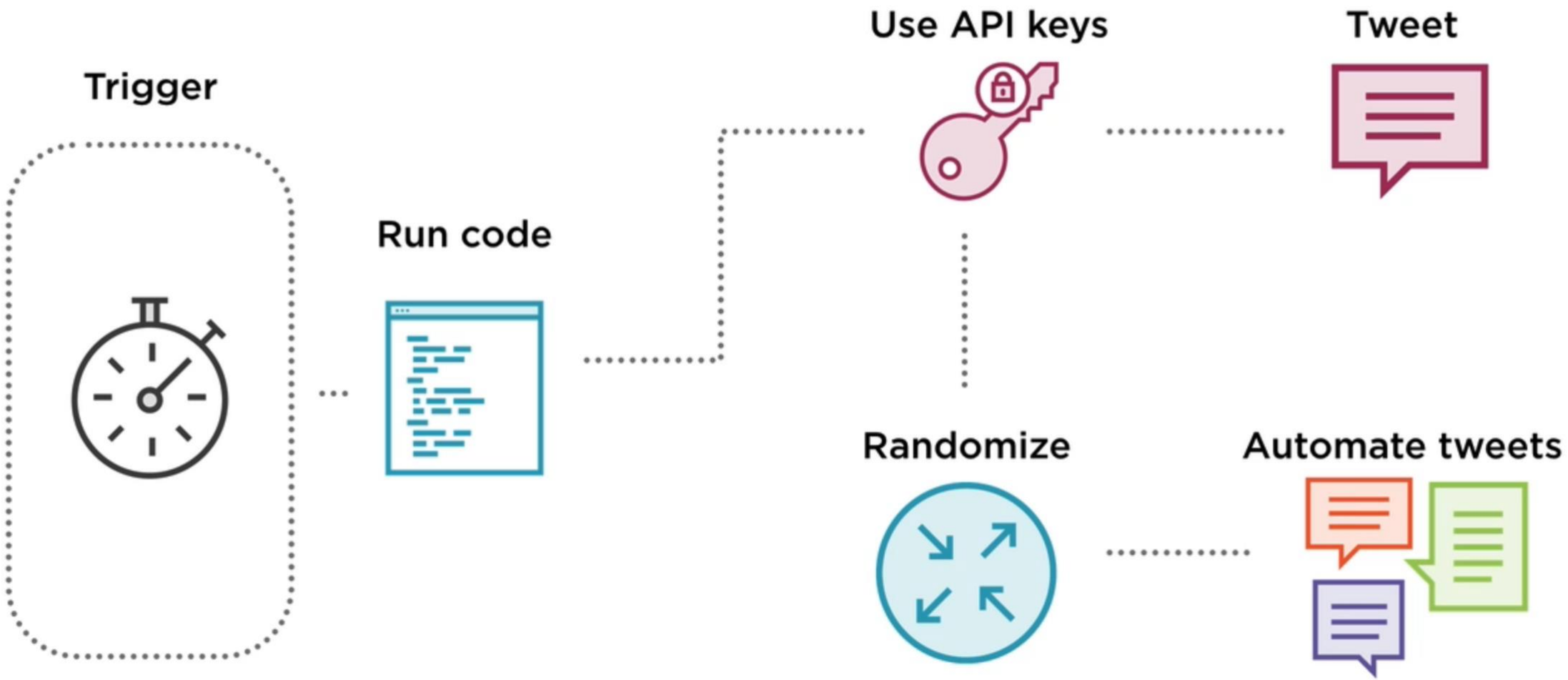
Add environment variable

▶ Encryption configuration

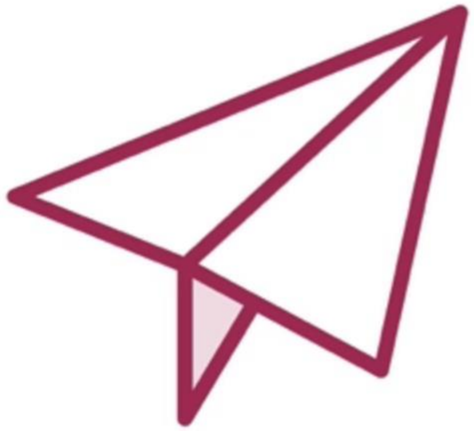
Cancel Create function


Lambda > Functions > lambda-canary
ARN - arn:aws:lambda:us-east-1:686275342394:function:lambda-canary
lambda-canary
Throttle
Qualifiers
Actions
Select a test event
Test
Save
Congratulations! Your Lambda function "lambda-canary" has been successfully created and configured with canary as a trigger in a disabled state. We recommend testing the function behavior before enabling the trigger.
Configuration
Permissions
Monitoring
Designer
lambda-canary
Layers
(0)
CloudWatch Events
Add destination

Determine Functionality - Twitter Bot



Credentials and Libraries



Sparrow Twitter bot shell

Twython

Twitter API keys

API Key Management Options

AWS Systems Manager Parameter Store

AWS Key Management Service (KMS)

DevOps on AWS

AWS Continuous Delivery Tools

CodeDeploy

AWS CodeDeploy is a service that automates code deployments to any instance, including Amazon EC2 instances and instances running on-premises. AWS CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during deployment, and handles the complexity of updating your applications.

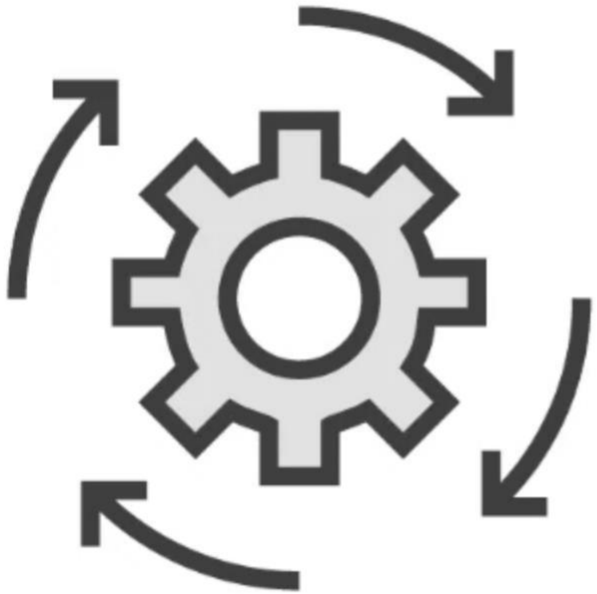
CodeBuild

AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers.

CodePipeline

AWS CodePipeline is a continuous delivery service that enables you to model, visualize, and automate the steps required to release your software.

DevOps and Continuous Delivery



- Automated deployments
- Repeatable and safer
- Smaller change sizes
- Deploy more frequently
- Faster feedback loops
- Learn and react quicker
- Deliver more value

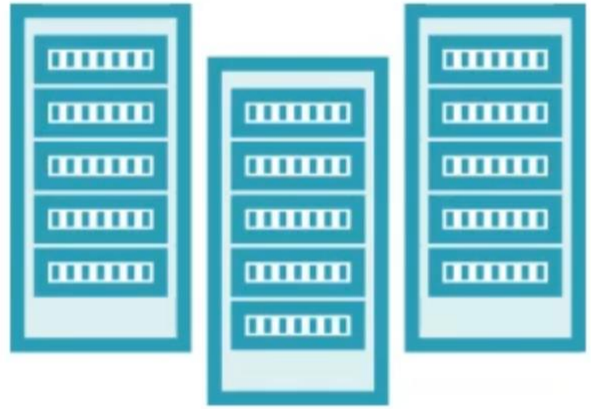
Benefits of CodeDeploy

- Automate your deployments
- Fully managed serverless service
- Free within AWS
- $0.02 per on-premises instance update
- Centralize control
- Versatile

CodeDeploy Components

Application

The core component



Deployment Group

A target for your deployment

Deployment

Roll out an app revision

A CodeDeploy Application

A unique name

An identifier that is unique inside its AWS account

Compute platform

EC2/On-premises, AWS Lambda, Amazon ECS

Deployment Groups

EC2/On-premises

Autoscaling groups
EC2 instance tags
On-premises hosts

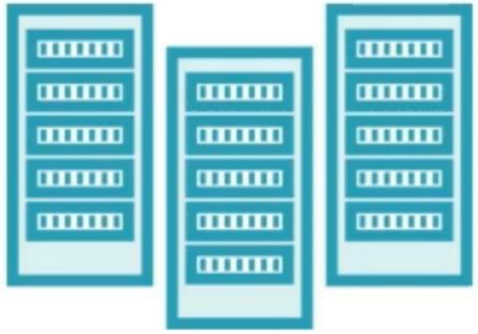
Lambda

Lambda function name

ECS

An ECS cluster +
An ECS service +
A load balancer +
Target Group 1 +
Target Group 2

A CodeDeploy Deployment



Deployment group

Targets for the deployment

Deployment type

"In Place" or "Blue-Green"

Deployment configuration

Such as "One At A Time"

Revision

An application version

Revisions

EC2/On-premises

GitHub commit ID
or
Zip file in S3

Lambda

AppSpec file
Function name
Function alias
Version info

ECS

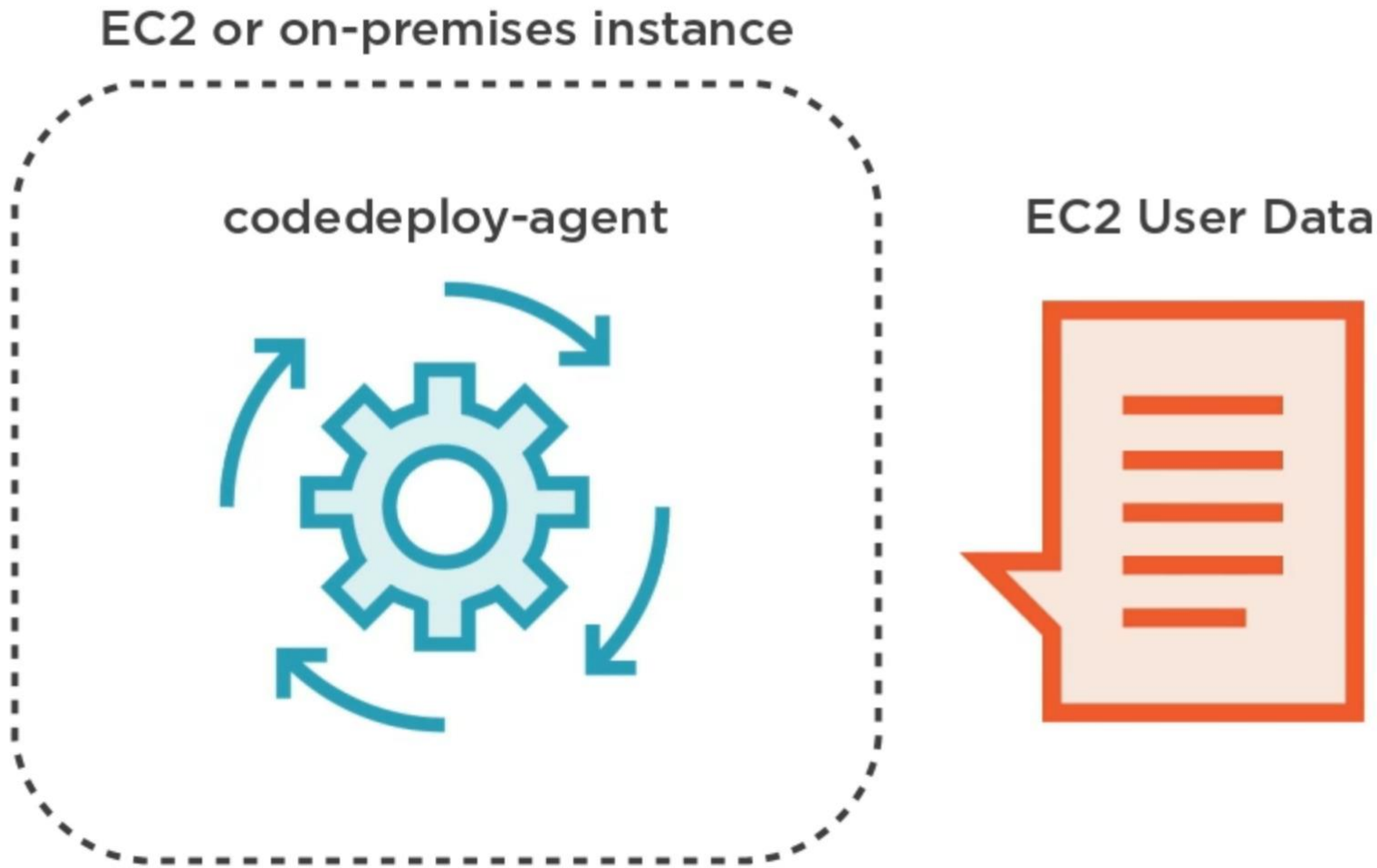
AppSpec file
Task Definition version
Container name
Container port

AppSpec File: appspec.yml

The application specification file (AppSpec file) is **a YAML -formatted or JSON-formatted file used by CodeDeploy to manage a deployment**. The AppSpec file for an EC2/On-Premises deployment must be named appspec. yml or appspec.

```yaml
version: 0.0
os: linux
files:
  - source: /src
    destination: /var/www/html
hooks:
  BeforeInstall:
    - location: scripts/before_install.sh
      timeout: 300
      runas: root
```

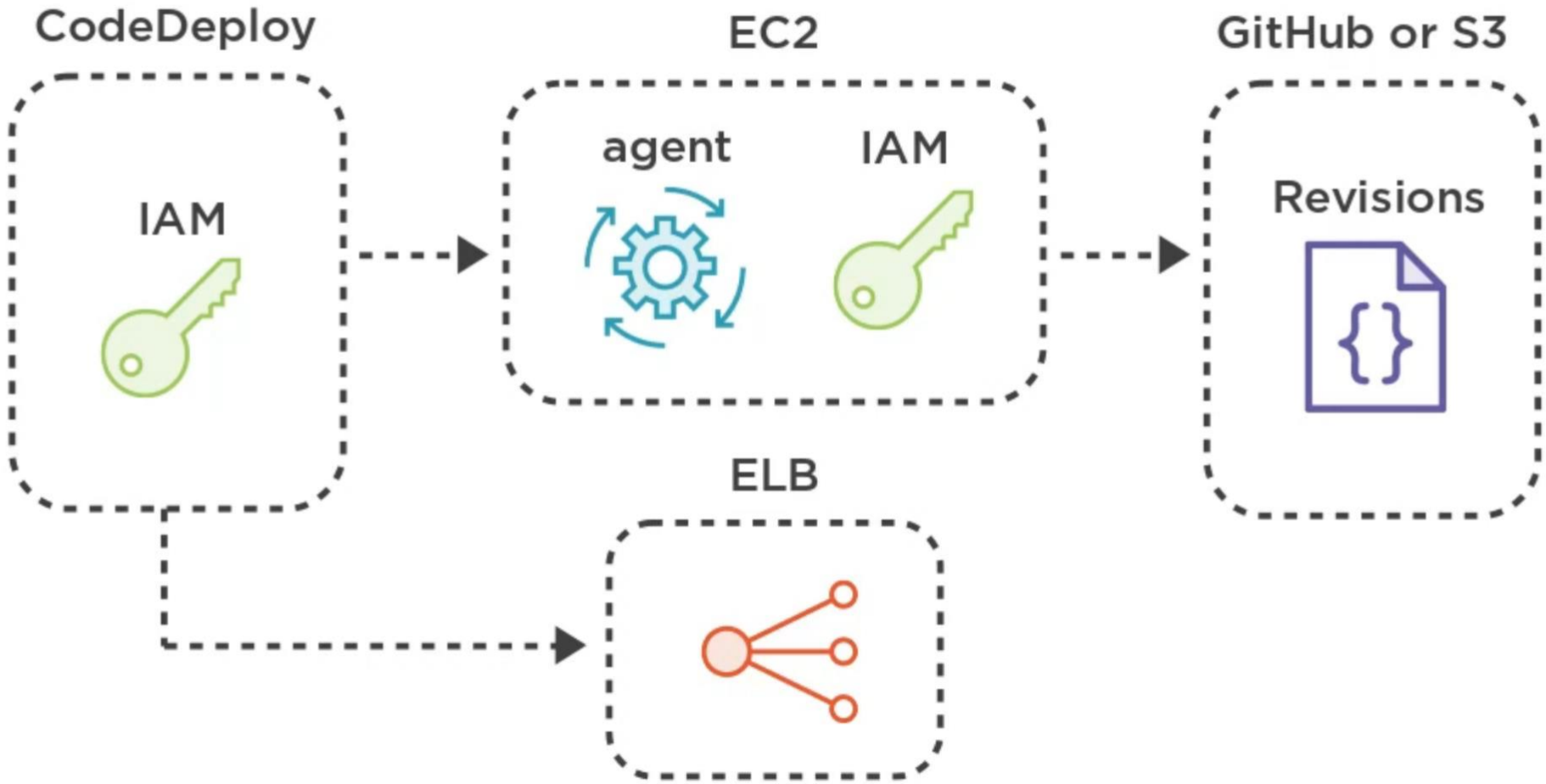
Installing the CodeDeploy Agent



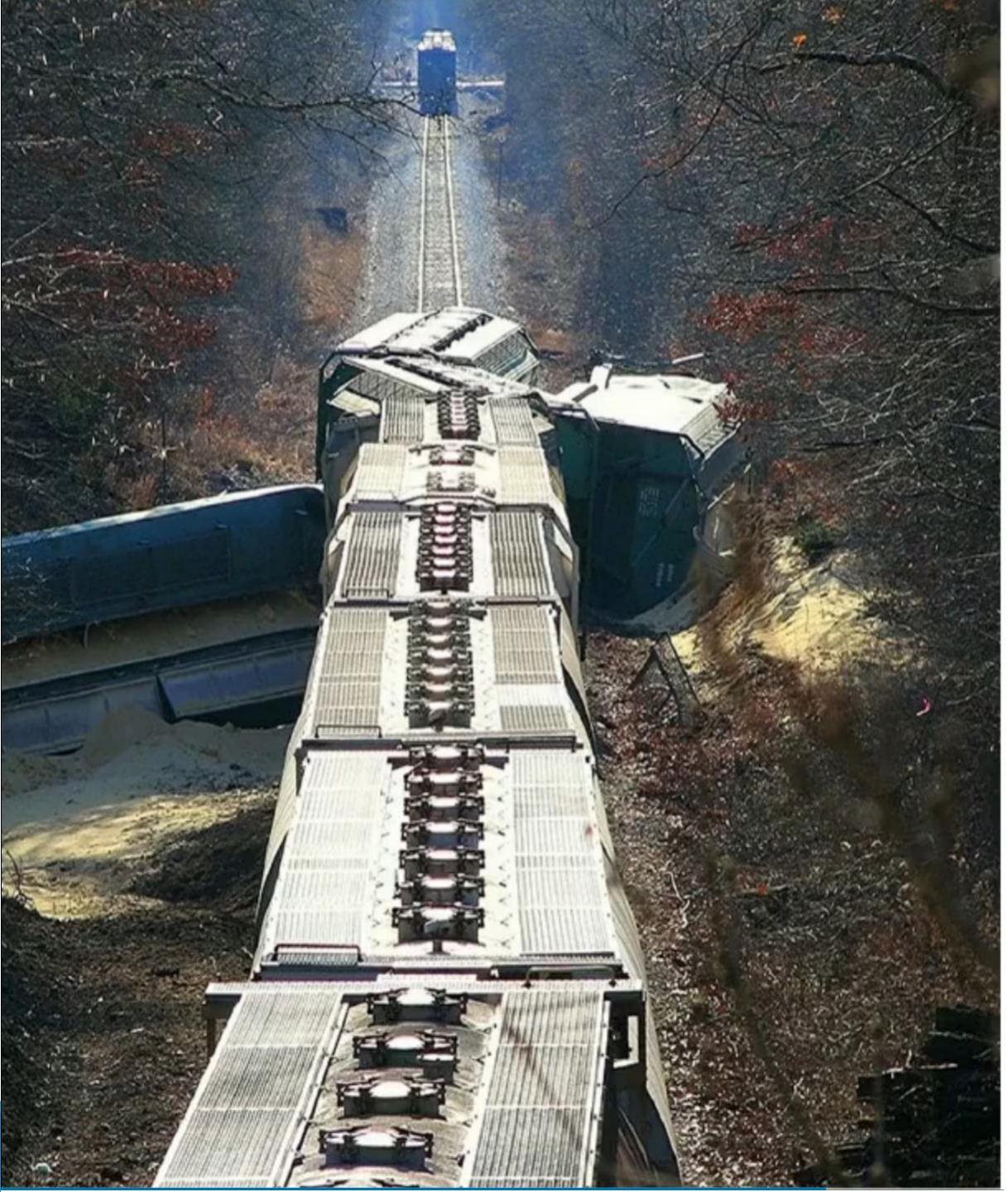
Install CodeDeploy - Red Hat

```
#!/bin/bash -xe
yum install -y ruby

cd /opt
curl -O https://aws-codedeploy-us-east-1.s3.amazonaws.com/
latest/install

chmod +x ./install
./install auto
```

Connecting All the Pieces



Continuous Integration with CodeBuild



Is Your CI System a Train Wreck?

- Software licensing
- CI software upgrades
- Underlying host management
- Agent configurations
- Hand crafted build projects
- Queued builds

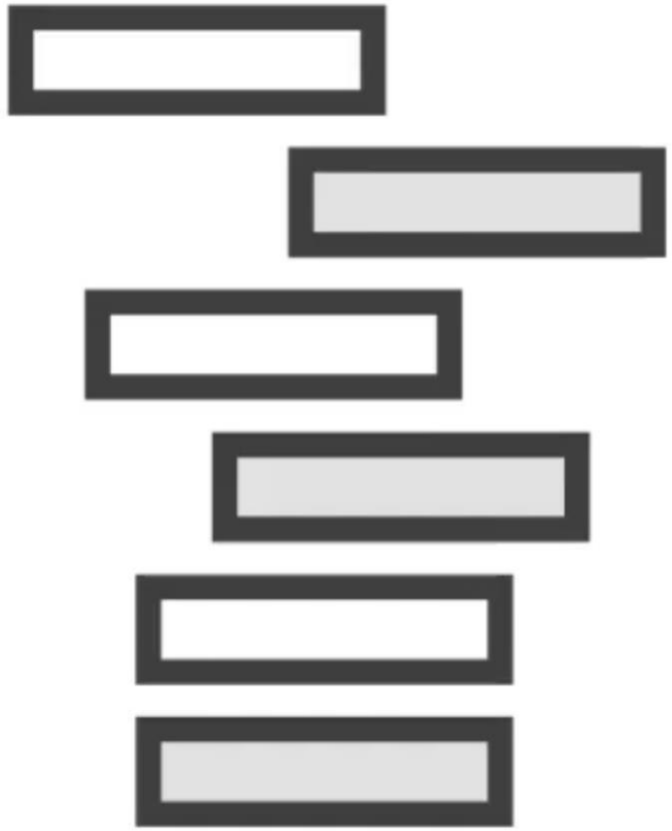
AWS Continuous Delivery Tools

CodeDeploy

CodeBuild

CodePipeline

Continuous Integration



DevOps and agile processes

Helps prevent merge conflicts

Triggers off code commits

Static code analysis

Compiles code

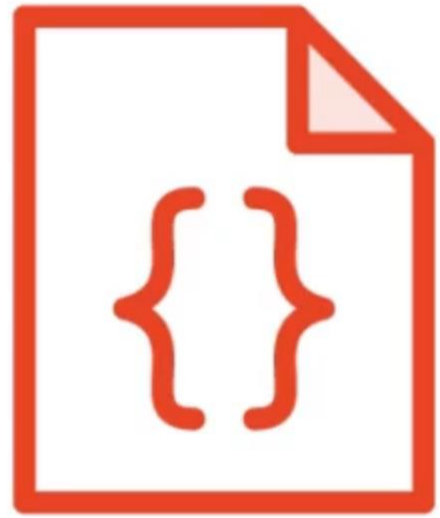
Runs tests

Packages and publishes artifacts

Catch and fix problems quicker

Continuous Integration Systems

Controller/agent



Open Source and free

Managed services

Alternatives to CodeBuild

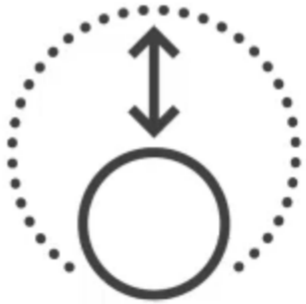
Jenkins

GitLab

Teamcity

Travis

Bamboo



CircleCI

Benefits of CodeBuild

- Fully managed build service
- Elastic scaling
- Pay only for build time used
- CLI and API driven
- Define builds as code
- Tightly knit AWS integrations
- Extensible

How Is CodeBuild Used in a CI/CD pipeline?

Runs builds	Compiles code	Publishes artifacts
Static code analysis	Runs tests	Sends notifications

CodeBuild Components

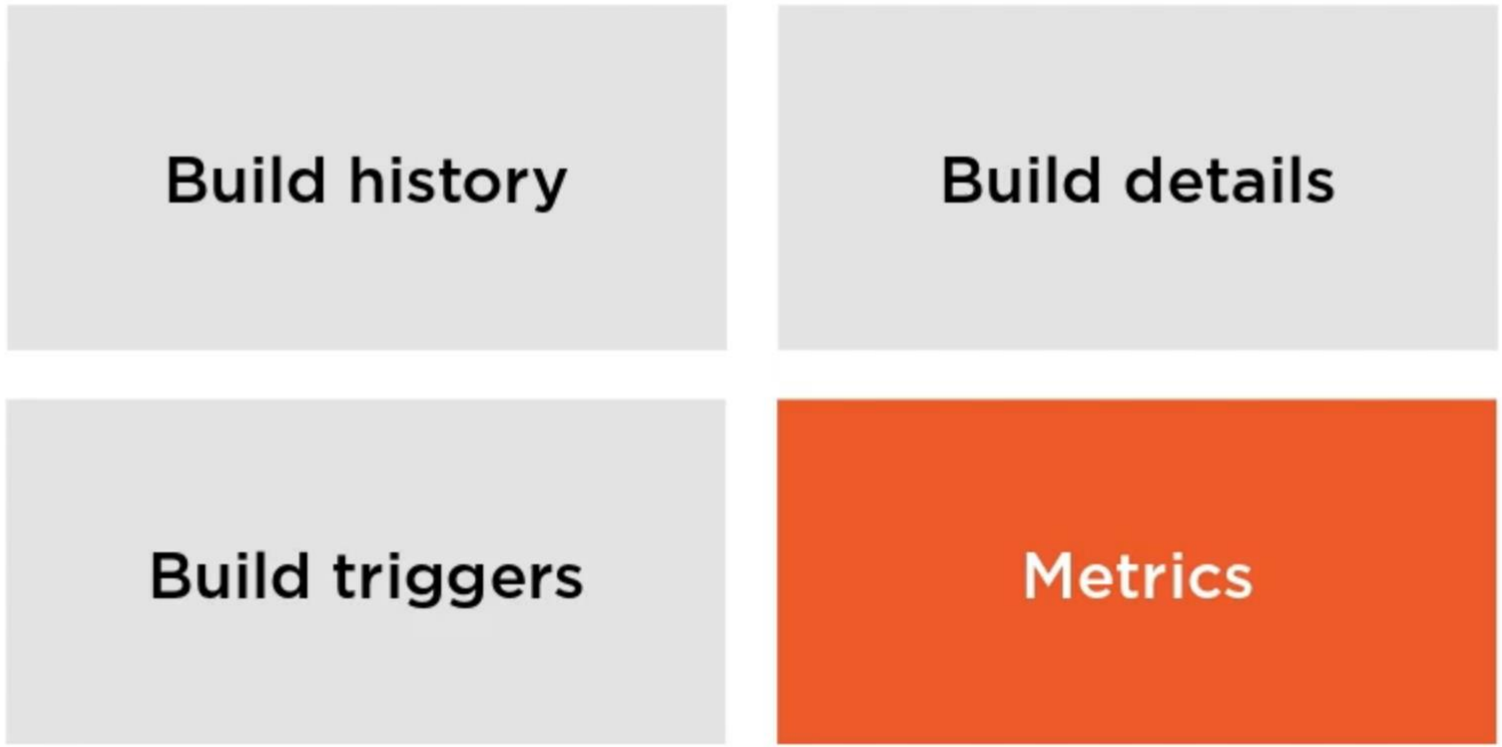
Build project

The definition of your project

Build or Build run

An execution of your Build project

Build Project



CodeBuild Project Build Details

Project configuration

Source

Environment

Build Spec

Artifacts

Logs

Builds or Build Runs

Build status

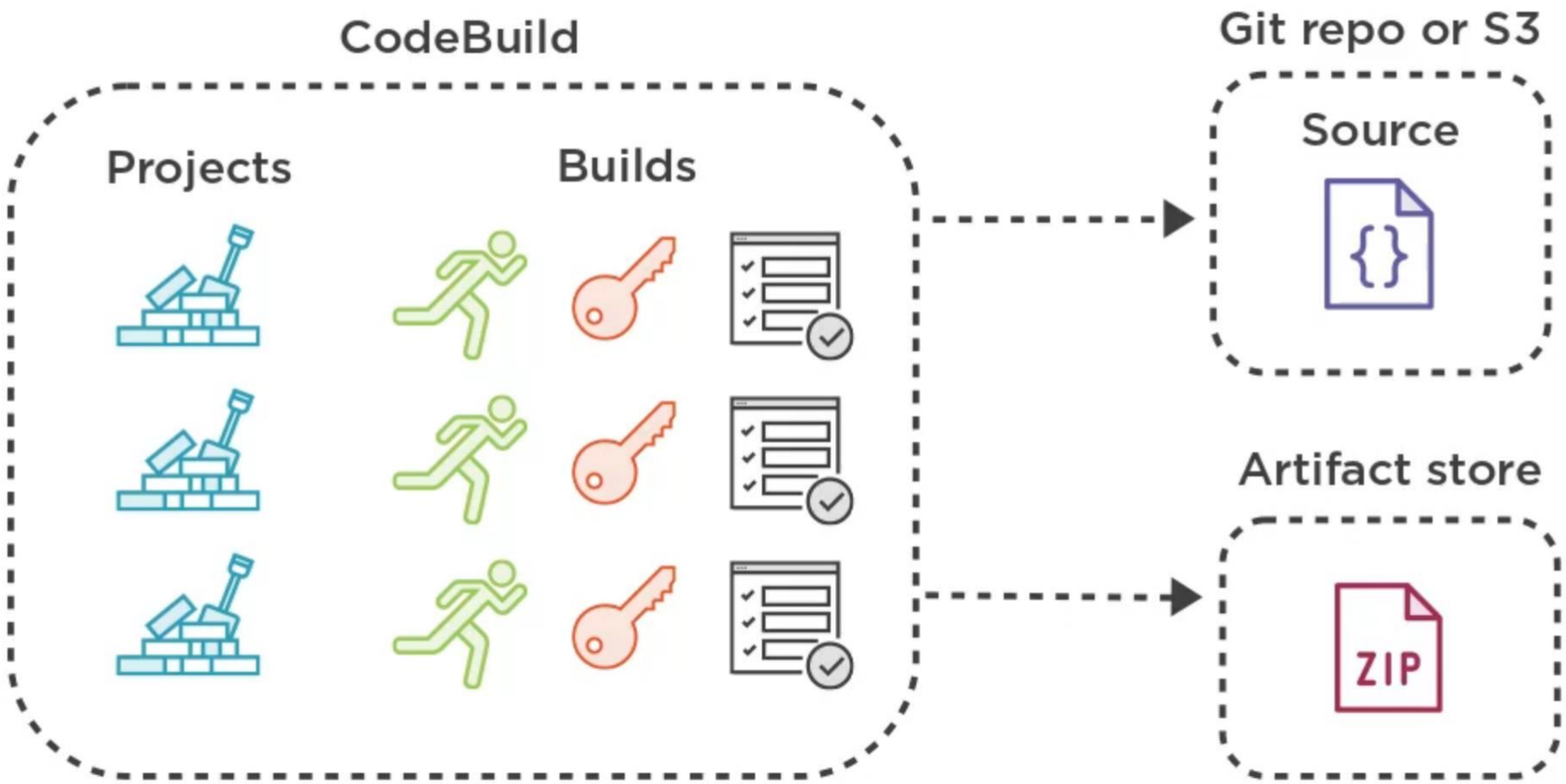
Build logs

Phase details

Environment variables

Build details

Connecting All the Pieces



Build Spec File: buildspec.yml

```
version: 0.2
env:
  variables:
    key: “value”
  parameter-store:
    key: “value”
phases:
  build:
    commands:
      - compile commands
      - test commands
artifacts:
  files:
    - target/messageUtil-1.0.jar
```

Testing Tools

Python

Java

Ruby

.NET

Golang

PHP

Developer Tools

CodeBuild

- ▶ **Source** • CodeCommit
- ▼ **Build** • CodeBuild
  - Getting started
  - Build projects
  - **Build history**
  - Account metrics
- ▶ **Deploy** • CodeDeploy
- ▶ **Pipeline** • CodePipeline

Feedback

Return to the old experience

☐	d-project:09323767-c2d1-480c-8bd0-41c3e5fbb9a9	Succeeded	python-build-project	-east-1-333652785952/codepipeline-demo/SourceArti/En5bLp2.zip	codepipeline/codepipeline-demo	20 seconds	53 minut ago
☐	JavaBuildProject:81274b54-b848-4bf8-af1d-1b41c8d9ecd0	Failed	JavaBuildProject	-	wes-novack	21 seconds	56 minut ago
☐	JavaBuildProject:80c8bafa-d949-4641-bf54-19a61a844831	Failed	JavaBuildProject	-	wes-novack	35 seconds	1 hour ag
☐	JavaBuildProject-TqFTNbP5njMa:a63b5608-3a25-4173-bb50-8849e47be1c5	Succeeded	JavaBuildProject-TqFTNbP5njMa	-	wes-novack	18 seconds	1 hour ag
☐	JavaBuildProject-TqFTNbP5njMa:a5cc502d-3530-4499-941c-35e1b4591ff	Succeeded	JavaBuildProject-TqFTNbP5njMa	-	wes-novack	25 seconds	2 hours a

Automating Software Releases with AWS CodePipeline

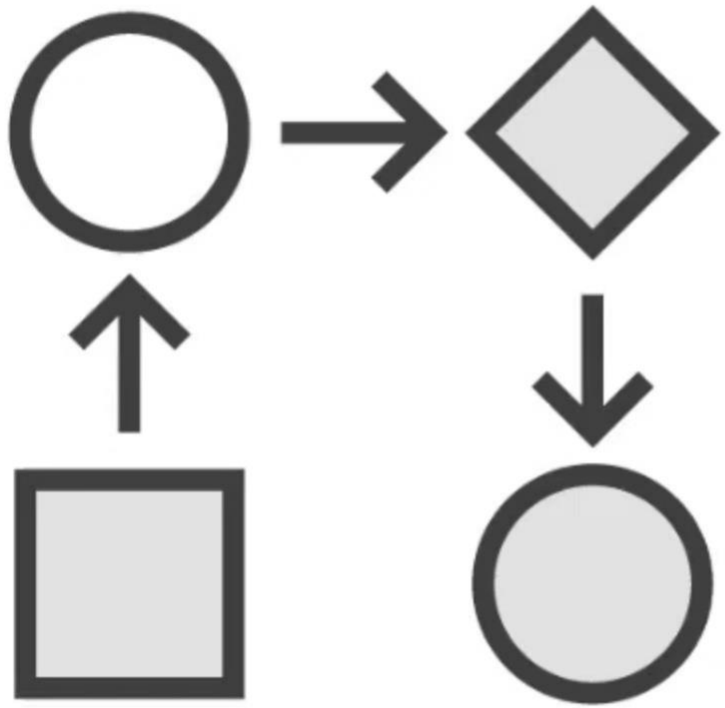
AWS Continuous Delivery Tools

CodeDeploy

CodeBuild

CodePipeline

Continuous Delivery Pipelines



A Continuous Delivery tool

Define your entire release process

Ties together disparate systems

Flows code out to production

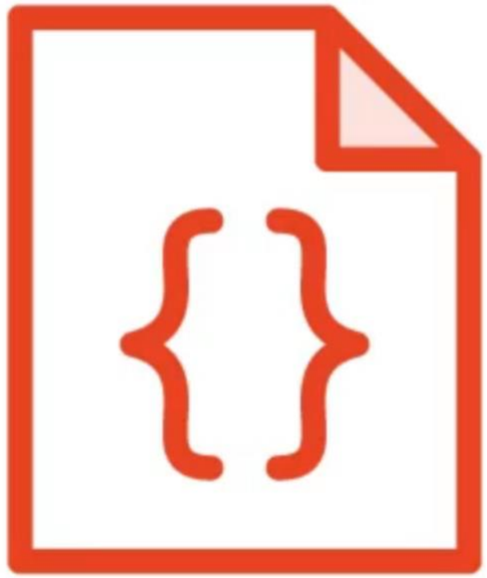
Prerequisites for CodePipeline

AWS account

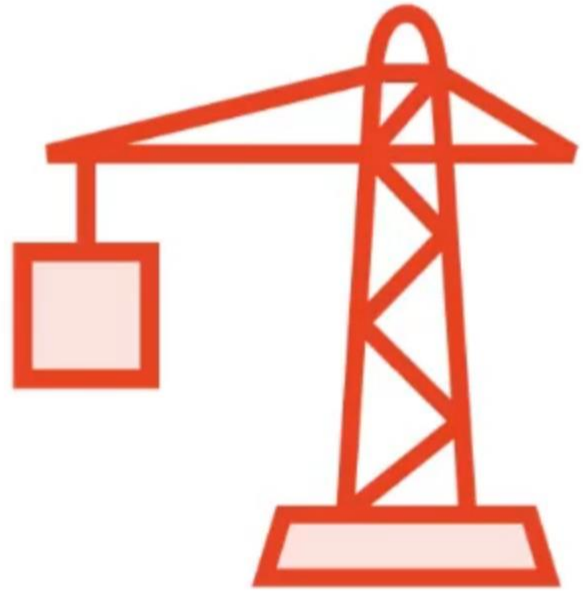
IAM credentials

Every pipeline requires a minimum of 2 stages

Prerequisites for CodePipeline



Source code



Build process

or

Deployment

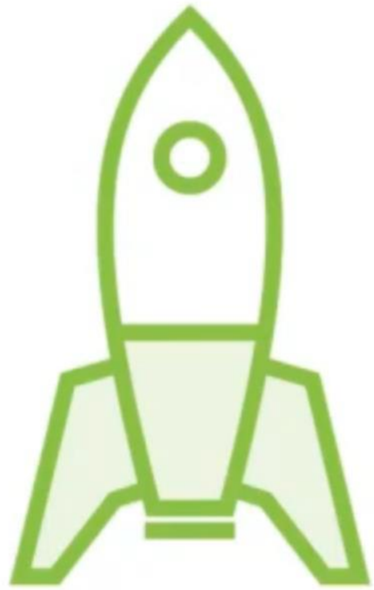
Benefits of CodePipeline

- Fully managed pipeline service
- Pay only for active pipelines
- Tightly knit AWS integrations
- Third party developer tool actions
- Extensible via Lambda and custom actions

CodePipeline Components

Pipelines

The definitions of software release processes



Executions

The invocations of a defined pipeline

Pipeline Components

Stages

A collection of action groups that contain actions

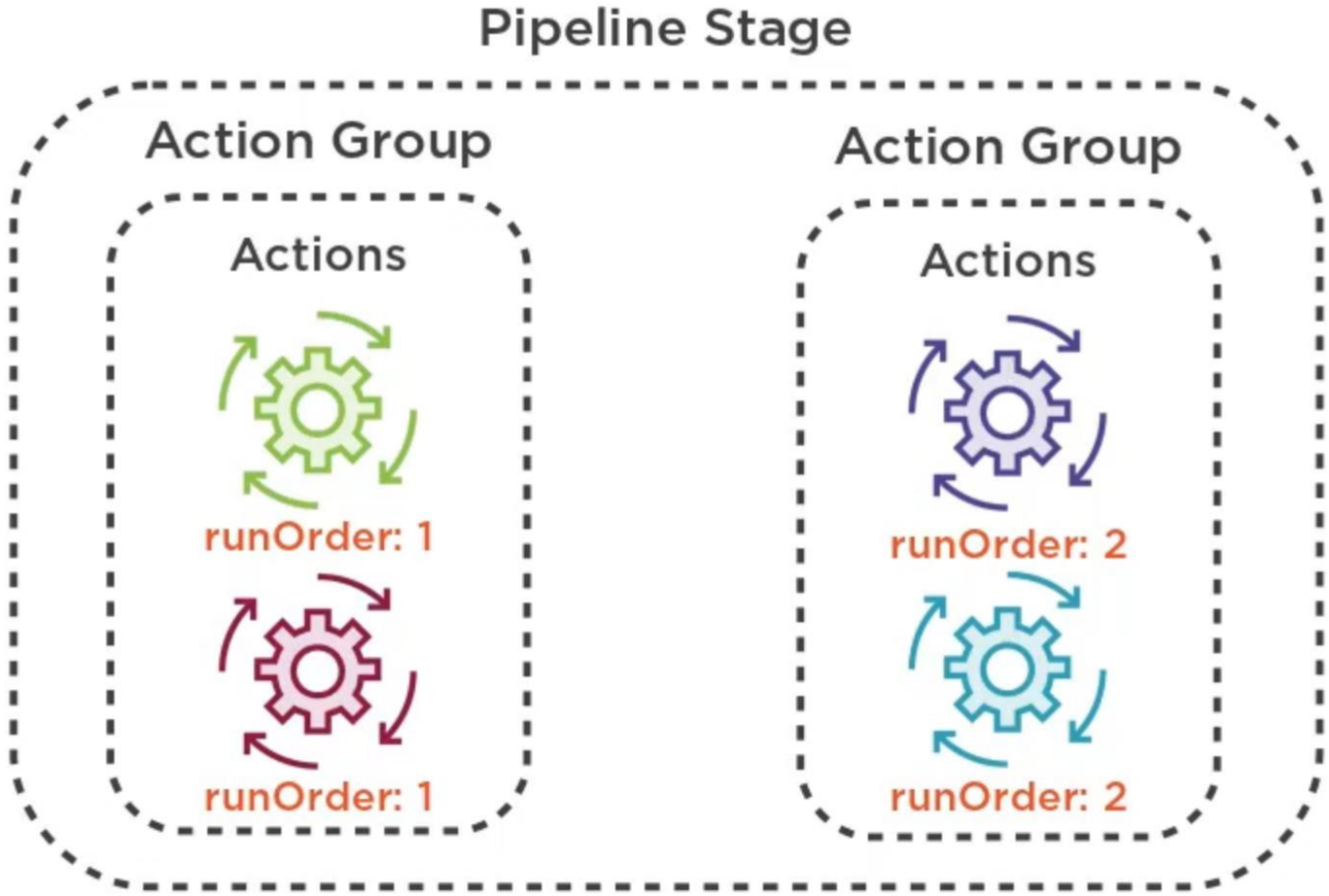
Transitions

The connections between stages

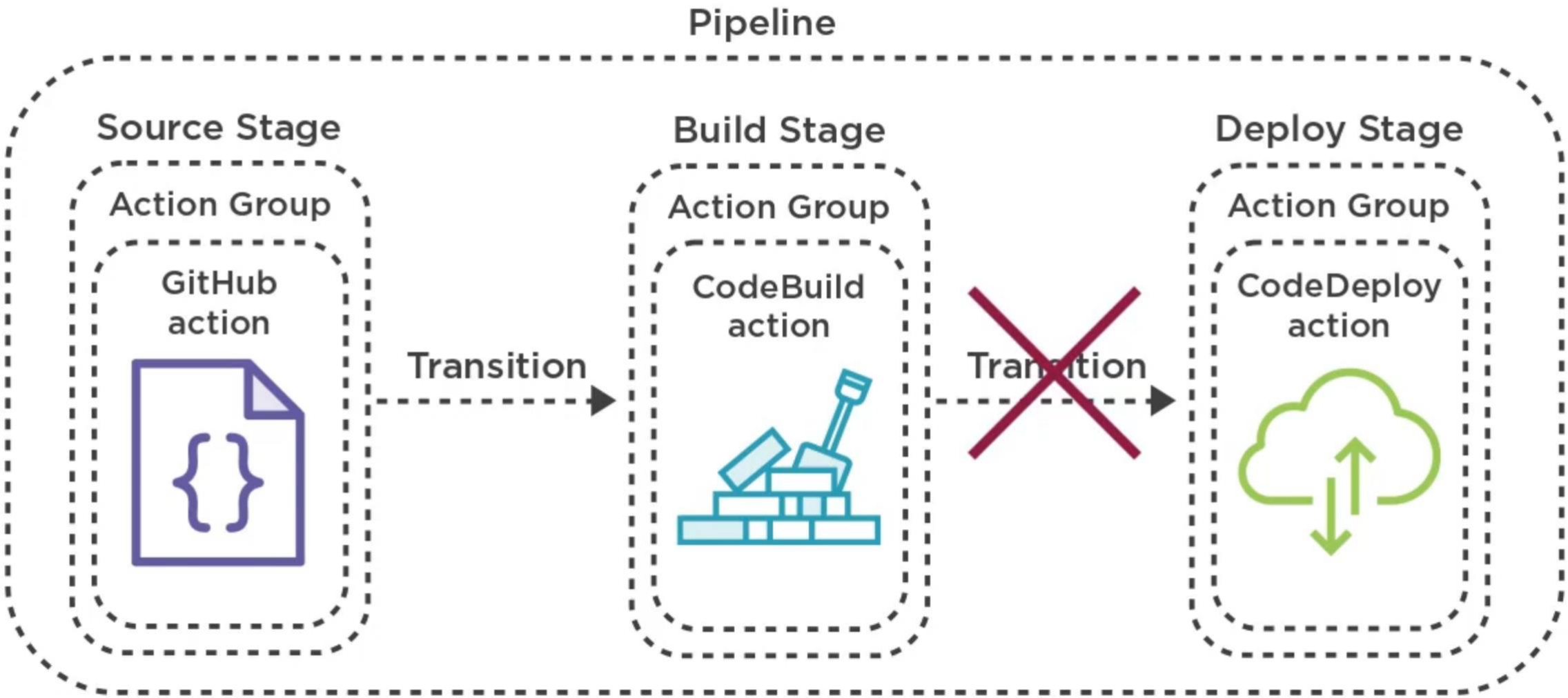
Artifacts

Bundles of code produced by a stage and ingested by another stage

A CodePipeline Stage



Visualizing CodePipeline



Action Types

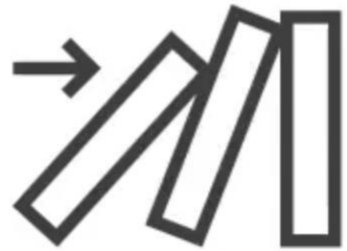
Source Action

Build Action

Test Action

Deploy Action

Approval Action



Invoke Action

Pipeline Execution Monitoring

CodeDeploy Triggers

Publish to an SNS topic when deployment events occur for a deployment group

CloudWatch Events

A configured rule can route CodePipeline event info to one or more targets

Developer Tools

CodePipeline

- ▶ **Source** • CodeCommit
- ▶ **Build** • CodeBuild
- ▶ **Deploy** • CodeDeploy
- ▼ **Pipeline** • CodePipeline
  - Getting started
  - Pipelines

Go to resource

Feedback

Return to the old experience

Developer Tools

AWS CodePipeline

visualize and automate the different stages of your software release process

AWS CodePipeline is a continuous integration and continuous delivery service for fast and reliable application and infrastructure updates. CodePipeline builds, tests, and deploys your code every time there is a code change, based on the release process models you define..

Create AWS CodePipeline pipeline

Get started with AWS CodePipeline by creating your first continuous delivery and continuous integration pipeline.

Create pipeline

Pricing (US)

Each active pipeline**	$1/month*

*All pipelines are free for the first 30 days.

How it works

Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review

Choose pipeline settings

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.

No more than 100 characters

Service role

- (•) New service role
  Create a service role in your account
- ( ) Existing service role
  Choose an existing service role from your account

Role name

Type your service role name

[x] Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

▸ Advanced settings

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Add source stage Info

Source

Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (Version 2)

New GitHub version 2 (app-based) action

To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. Learn more

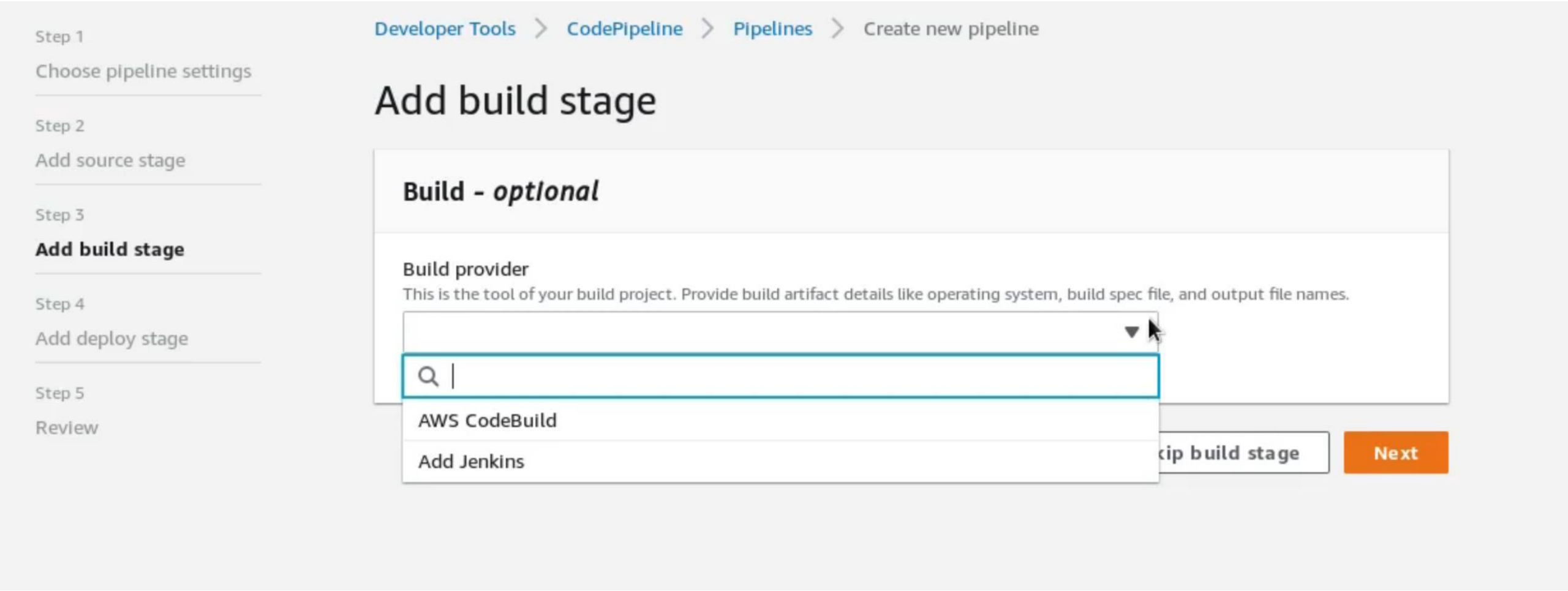
Connection

Choose an existing connection that you have already configured, or create a new one and then return to this task.

arn:aws:codestar-connections:us-east-1:444930770892:connection/dfc4c0b2 or Connect to GitHub

Ready to connect

Your GitHub connection is ready for use.


Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Developer Tools
CodePipeline
Pipelines
Create new pipeline
Add build stage
Build - optional
Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.
AWS CodeBuild
Add Jenkins
ip build stage
Next

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Add deploy stage

Deploy - *optional*

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS CodeDeploy

Region

US East - (N. Virginia)

Application name
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.

python-app

Deployment group
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.

Cancel | Previous | Skip deploy stage | Next

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Review

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name

demo

Artifact location

codepipeline-us-east-1-333652785952

Service role name

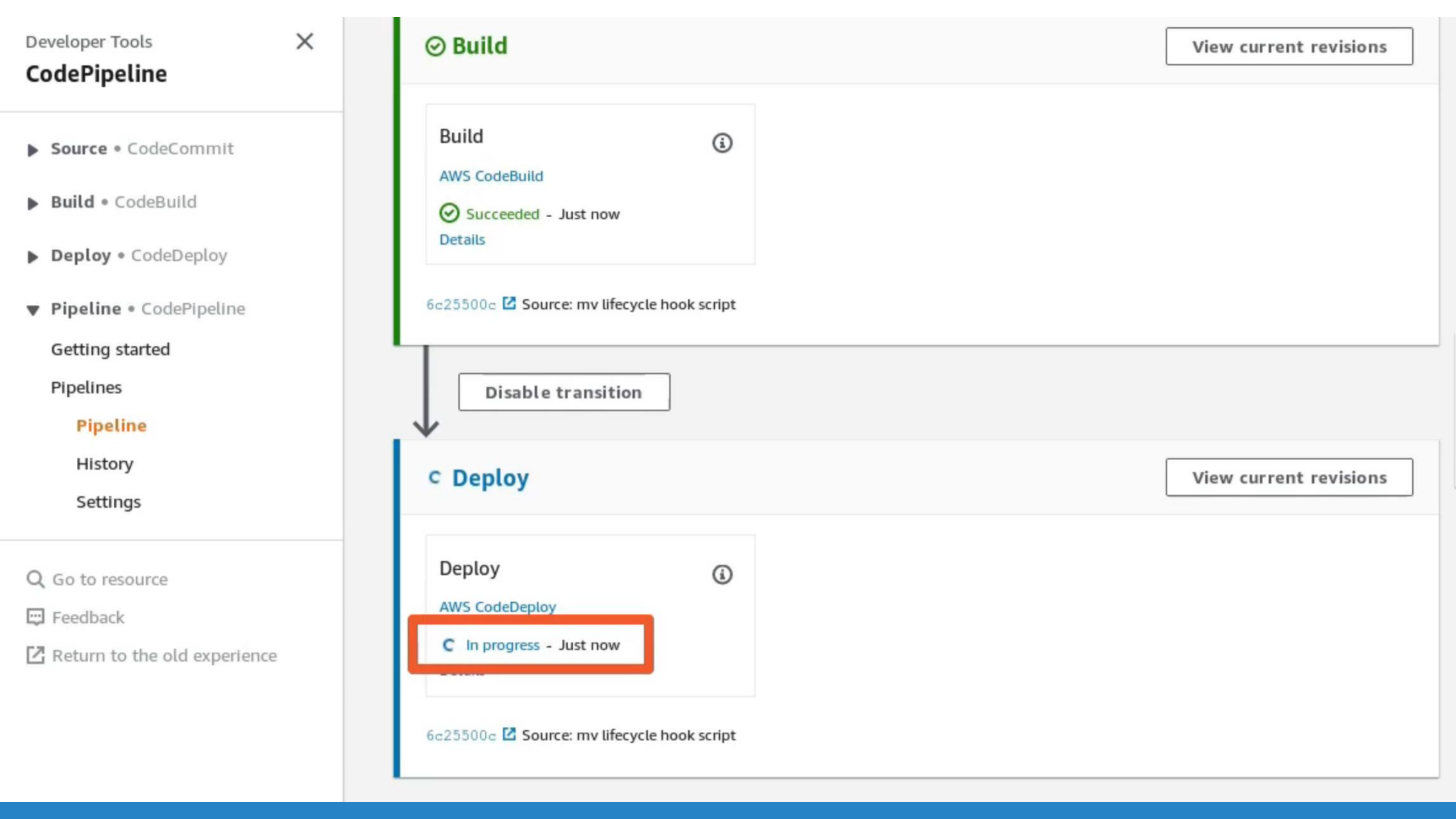
AWSCodePipelineServiceRole-us-east-1-demo

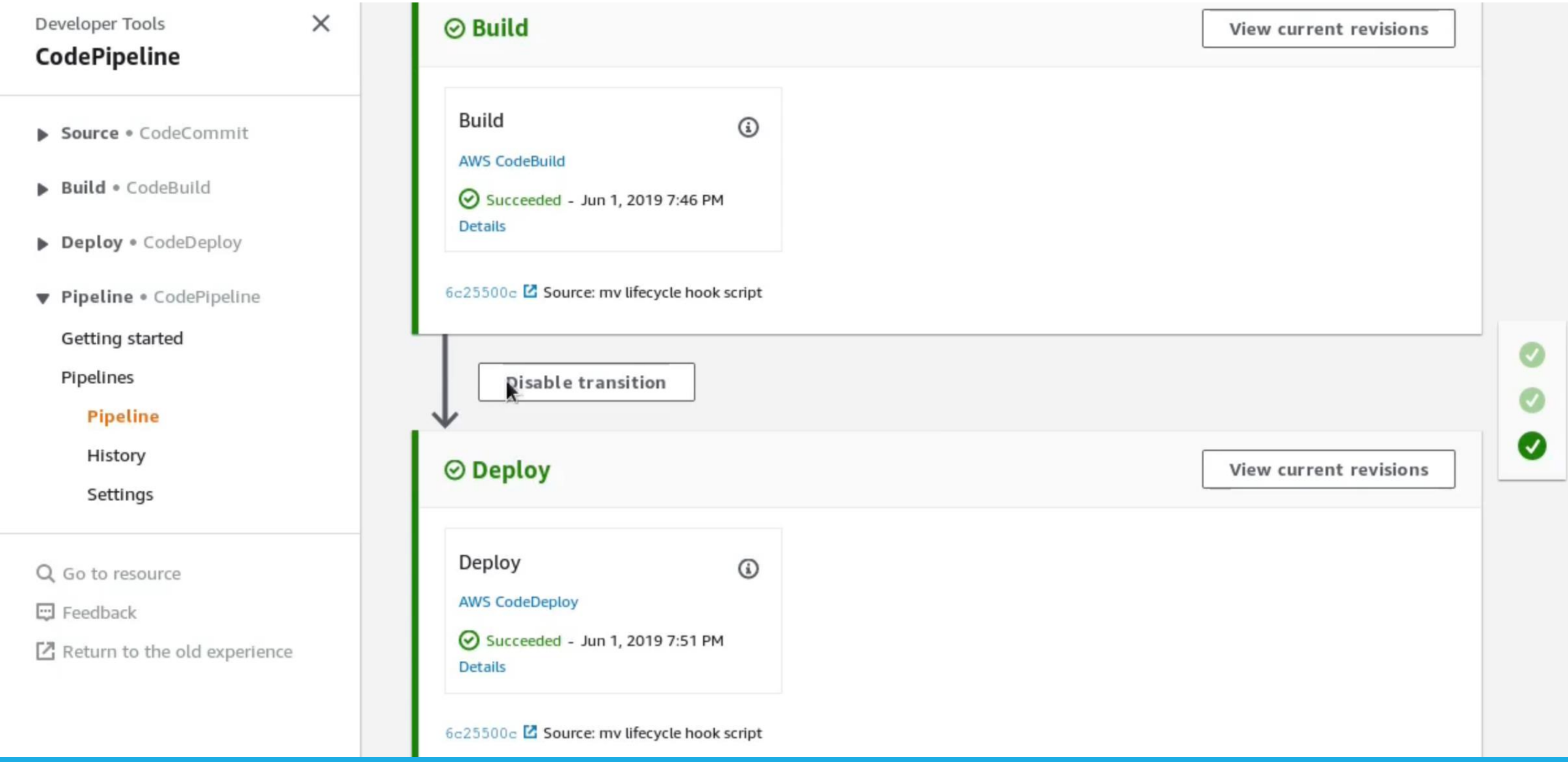
Step 2: Add source stage

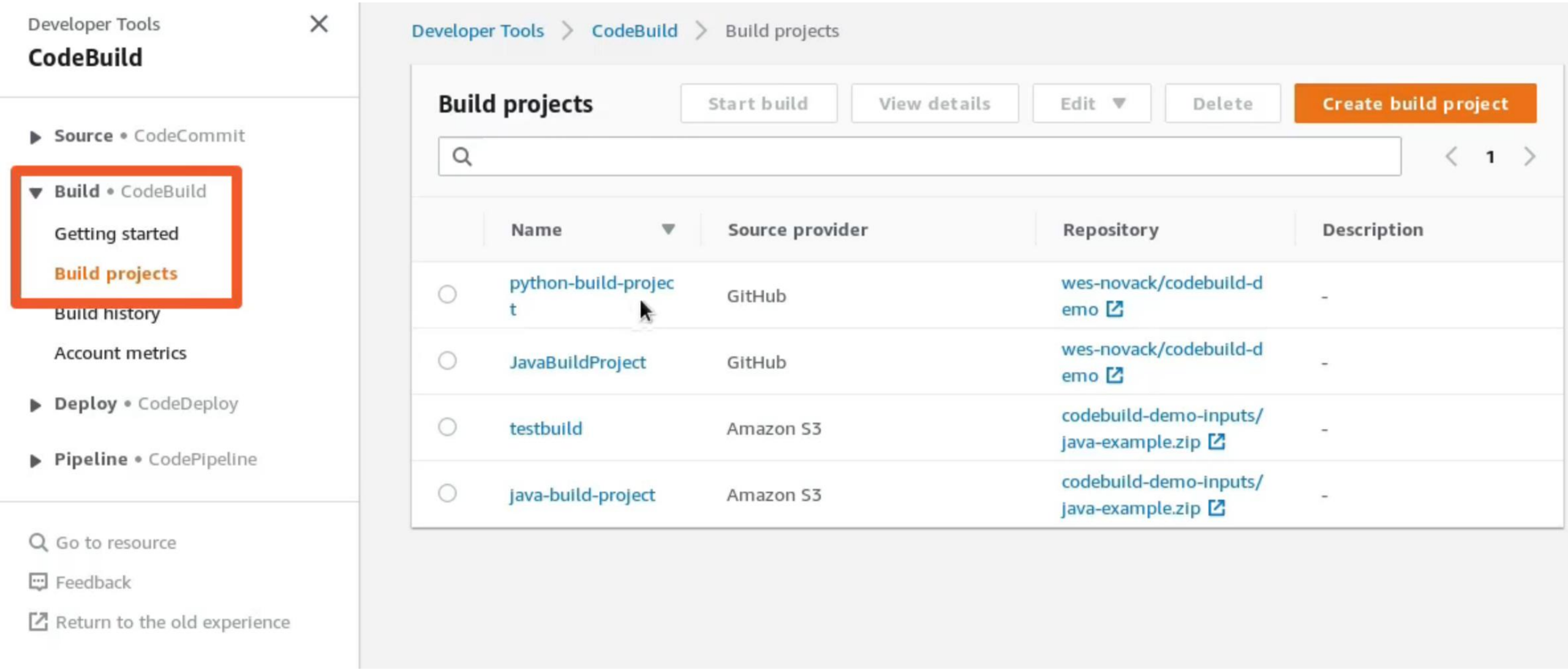
Source action provider

Source action provider

ThirdParty GitHub


Developer Tools
CodePipeline
Source • CodeCommit
Build • CodeBuild
Deploy • CodeDeploy
Pipeline • CodePipeline
Getting started
Pipelines
Pipeline
History
Settings
Go to resource
Feedback
Return to the old experience
Build
View current revisions
Build
AWS CodeBuild
Succeeded - Just now
Details
6c25500c
Source: mv lifecycle hook script
Disable transition
Deploy
View current revisions
Deploy
AWS CodeDeploy
In progress - Just now
6c25500c
Source: mv lifecycle hook script


Developer Tools
CodePipeline
Source • CodeCommit
Build • CodeBuild
Deploy • CodeDeploy
Pipeline • CodePipeline
Getting started
Pipelines
Pipeline
History
Settings
Go to resource
Feedback
Return to the old experience
Build
View current revisions
Build
AWS CodeBuild
Succeeded - Jun 1, 2019 7:46 PM
Details
6c25500c Source: mv lifecycle hook script
Disable transition
Deploy
View current revisions
Deploy
AWS CodeDeploy
Succeeded - Jun 1, 2019 7:51 PM
Details
6c25500c Source: mv lifecycle hook script


Developer Tools
CodeBuild
Source • CodeCommit
Build • CodeBuild
Getting started
Build projects
Build history
Account metrics
Deploy • CodeDeploy
Pipeline • CodePipeline
Go to resource
Feedback
Return to the old experience
Developer Tools > CodeBuild > Build projects
Build projects
Start build
View details
Edit
Delete
Create build project
1
Name
Source provider
Repository
Description
python-build-project
GitHub
wes-novack/codebuild-demo
-
JavaBuildProject
GitHub
wes-novack/codebuild-demo
-
testbuild
Amazon S3
codebuild-demo-inputs/java-example.zip
-
java-build-project
Amazon S3
codebuild-demo-inputs/java-example.zip
-

References

https://docs.docker.com/get-started/overview/

https://docs.docker.com/engine/install/ubuntu/

https://docs.docker.com/desktop/install/windows-install/

https://docs.docker.com/engine/reference/commandline/cli/

https://docs.docker.com/samples/

https://docs.docker.com/reference/