

Choose a Topic

Introduction

Auto Scaling Deployment  
Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With  
OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

Lambda

## Amazon API Gateway

Amazon API Gateway is an AWS service that enables you to create, publish, maintain, monitor, and secure your own REST and WebSocket APIs at any scale. You can create robust, secure, and scalable APIs that access AWS or other web services, as well as data stored in the AWS Cloud. You can create APIs for use in your own client applications (apps). Or you can make your APIs available to third-party app developers.

How API Gateway Works

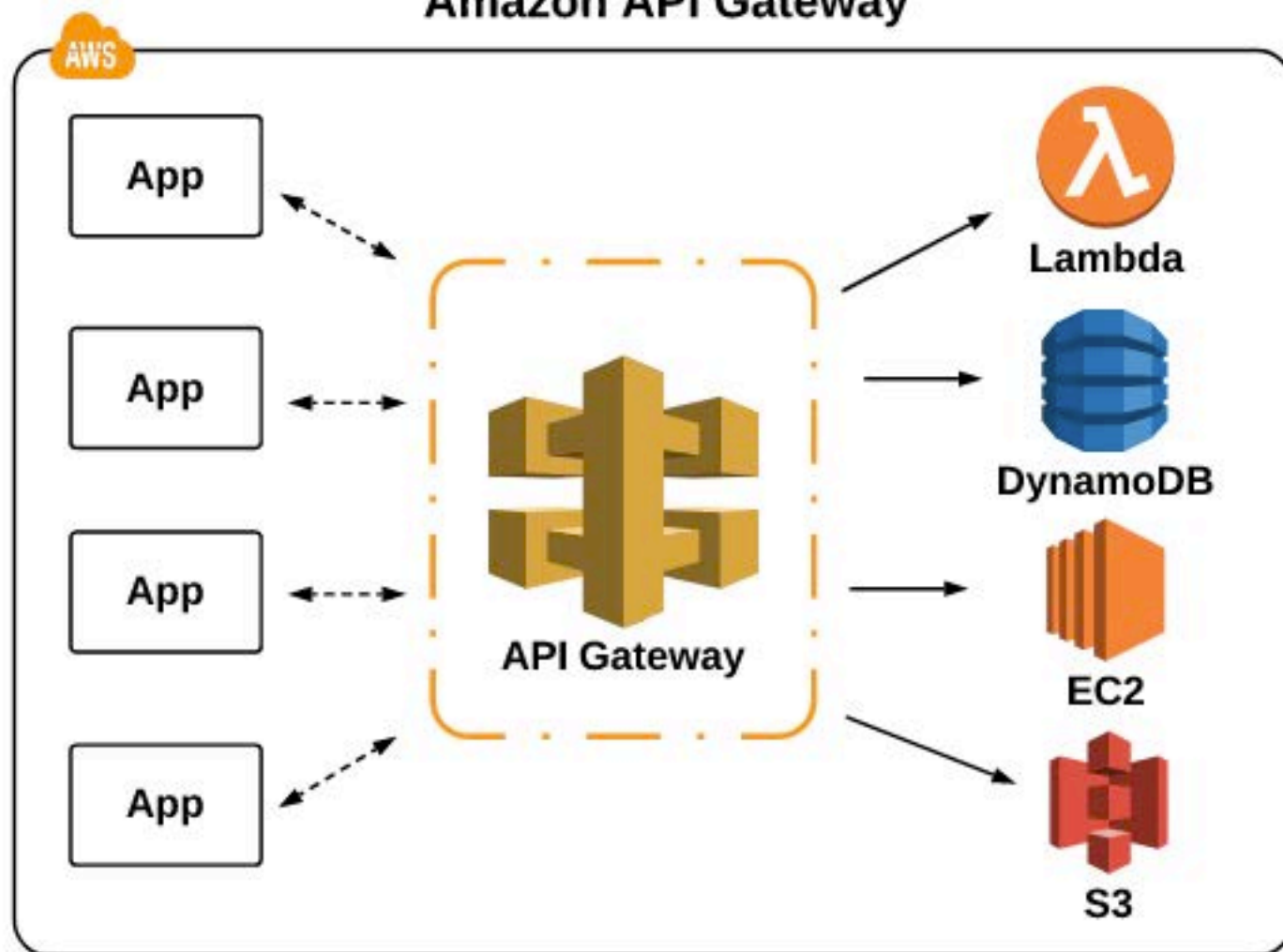
API Gateway Private Endpoints

API Gateway with Lambda

Deploying an API

REST and Web Socket APIs

## Amazon API Gateway





Choose a Topic

Introduction

Auto Scaling Deployment  
Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With  
OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

Lambda

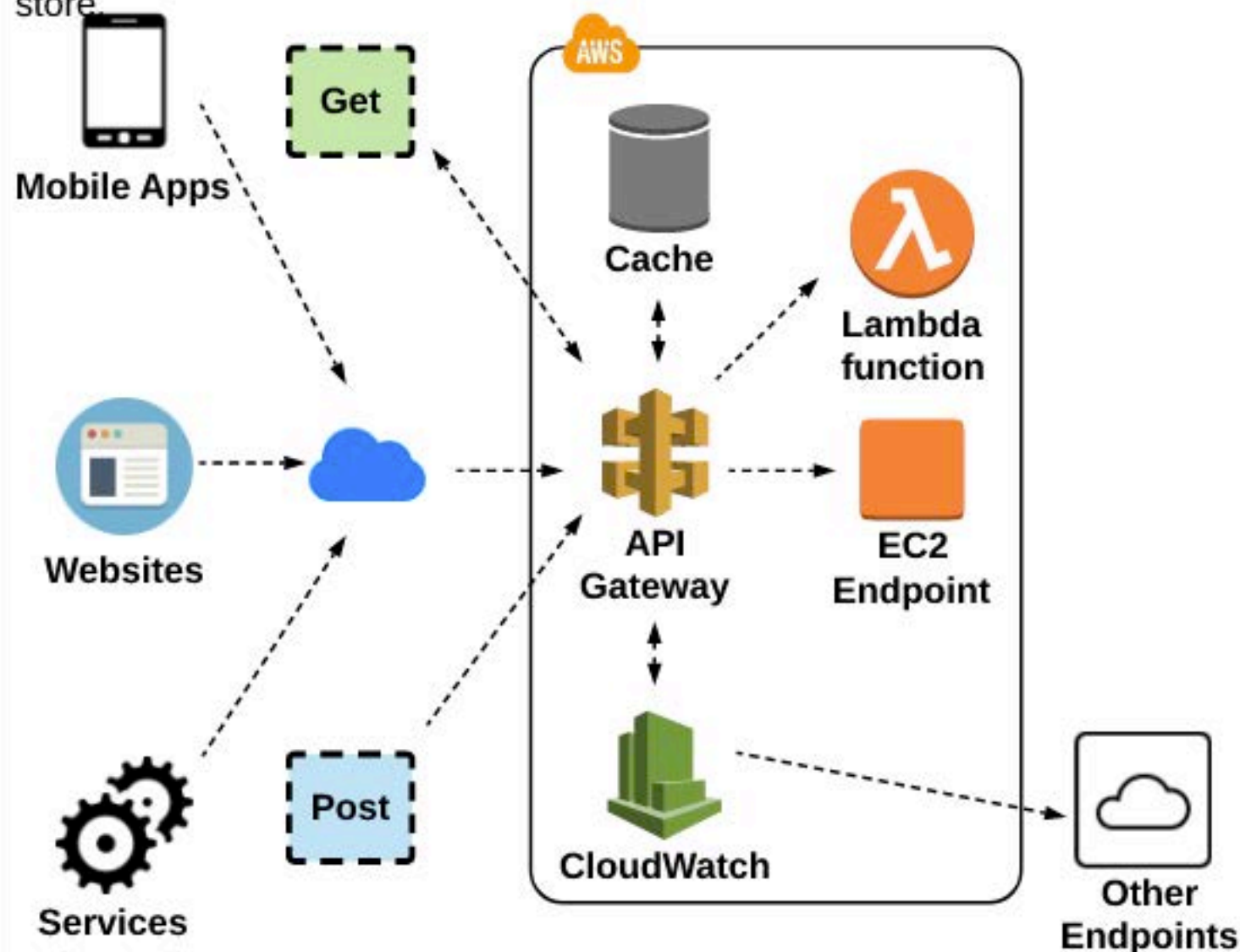
## Amazon API Gateway

How does API Gateway work? It's like a service contract. An app (or client application) gains programmatic access to AWS services, or a website on the internet, through one or more APIs, which are hosted in API Gateway. The app is at the API's frontend. The integrated AWS services and websites are located at the API's backend.

Back

## Amazon API Gateway

Once the App or client gains access to the website or service through API Gateway, they can use methods to view and alter the data. Specifically, HTTP GET and POST methods can be used to access the backend data store.





Choose a Topic

Introduction

Auto Scaling Deployment  
Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With  
OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

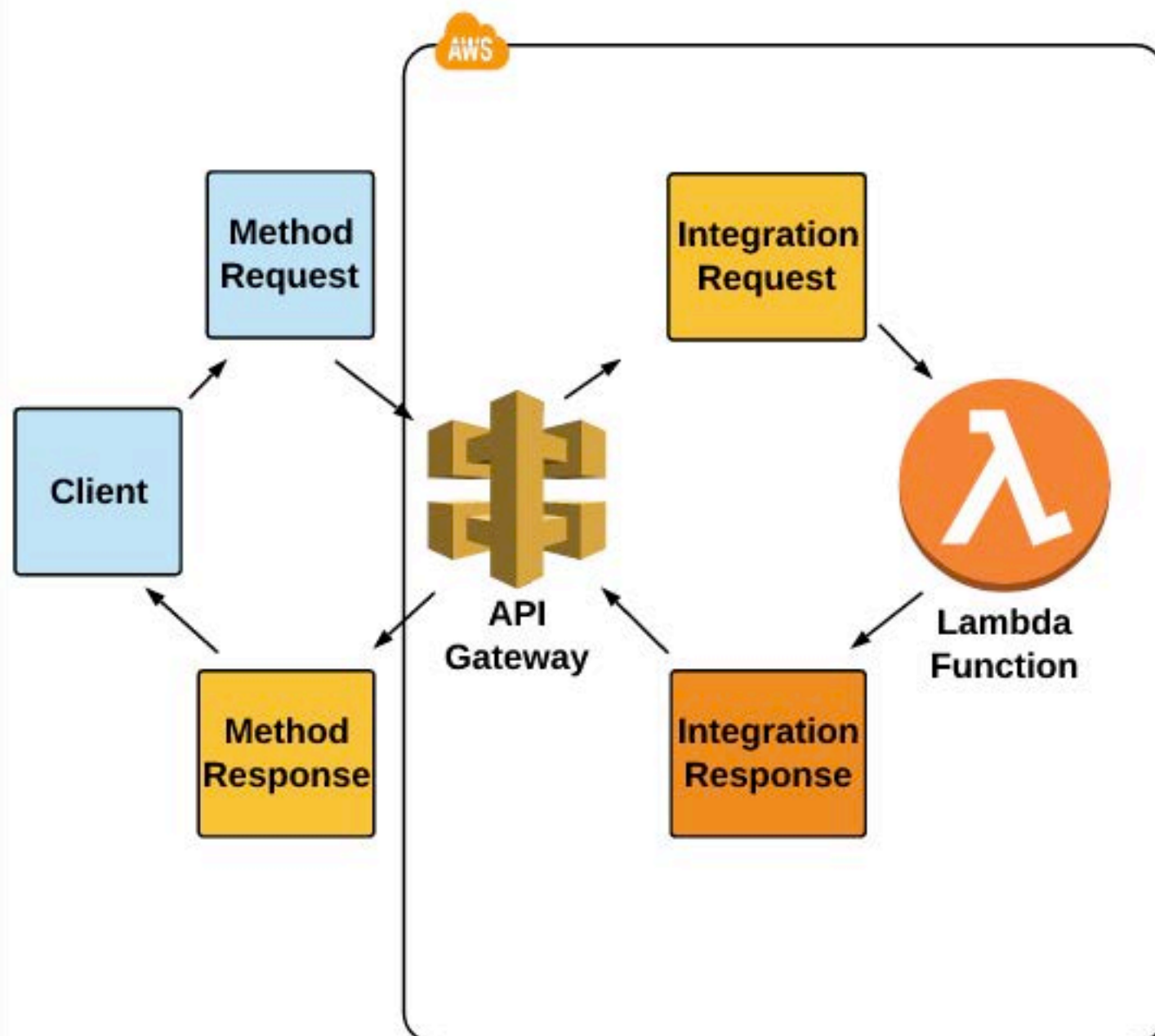
Lambda

## Amazon API Gateway

You can use API Gateway to trigger your Lambda Functions. When you call your API Gateway endpoint, with Get, or Put methods, your Lambda function will be invoked. API Gateway handles the request coming from the client and passes it to the Lambda function. The Lambda function returns a response to API Gateway which forwards it to the client.

[Back](#)

## API Gateway with Lambda





Choose a Topic

Introduction

Auto Scaling Deployment  
Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With  
OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

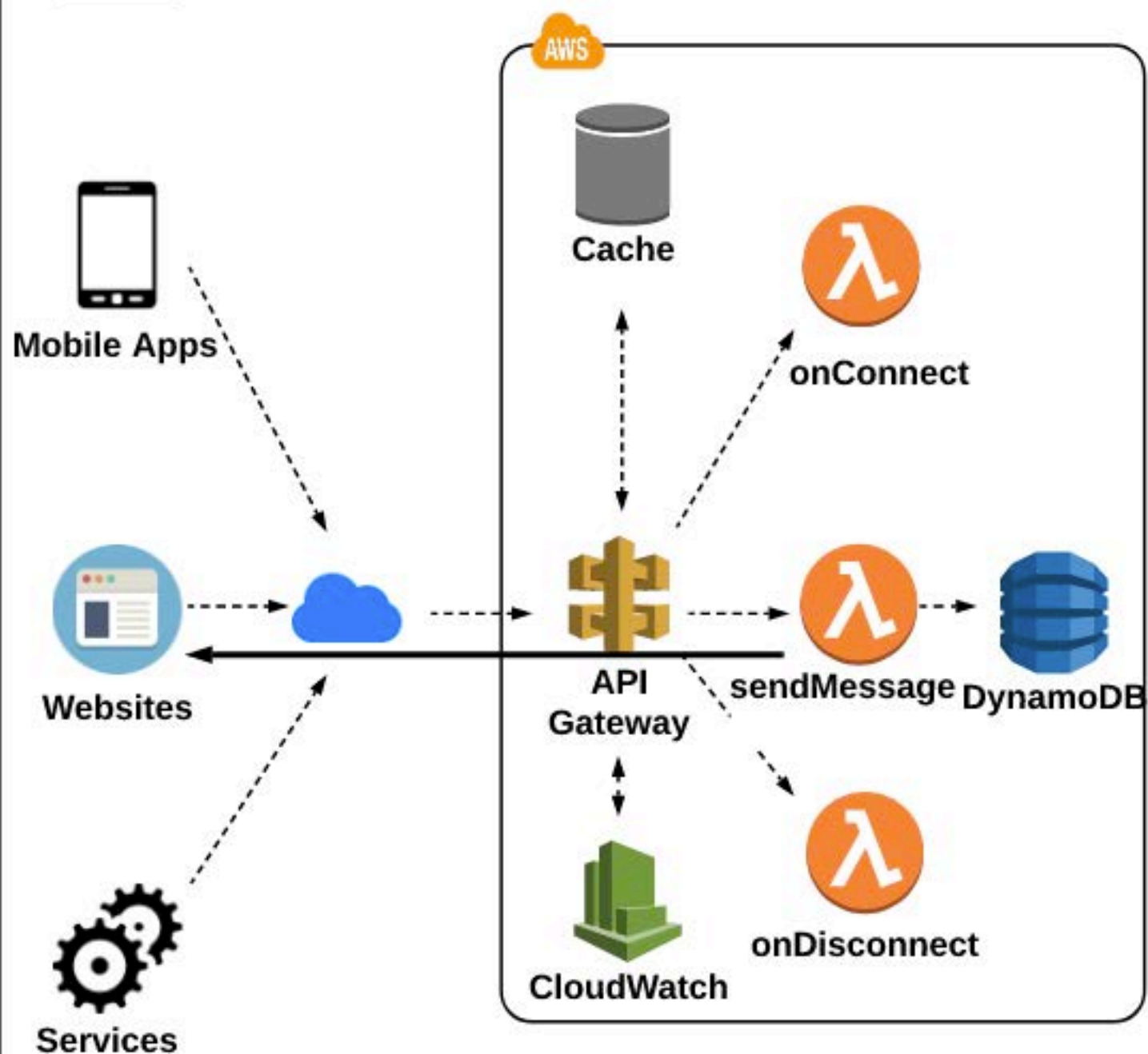
Lambda

## Amazon API Gateway

**REST API vs WEB Socket API** - A REST API receives messages from the client and forwards those to the backend resource. It also forwards responses from the backend to the client. But a Web Socket API supports two-way communication between client apps and your backend. The backend can send callback messages to connected clients. The backend services can **independently** send messages to its clients.

Back

## Web Socket API





## Choose a Topic

Introduction

Auto Scaling Deployment Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

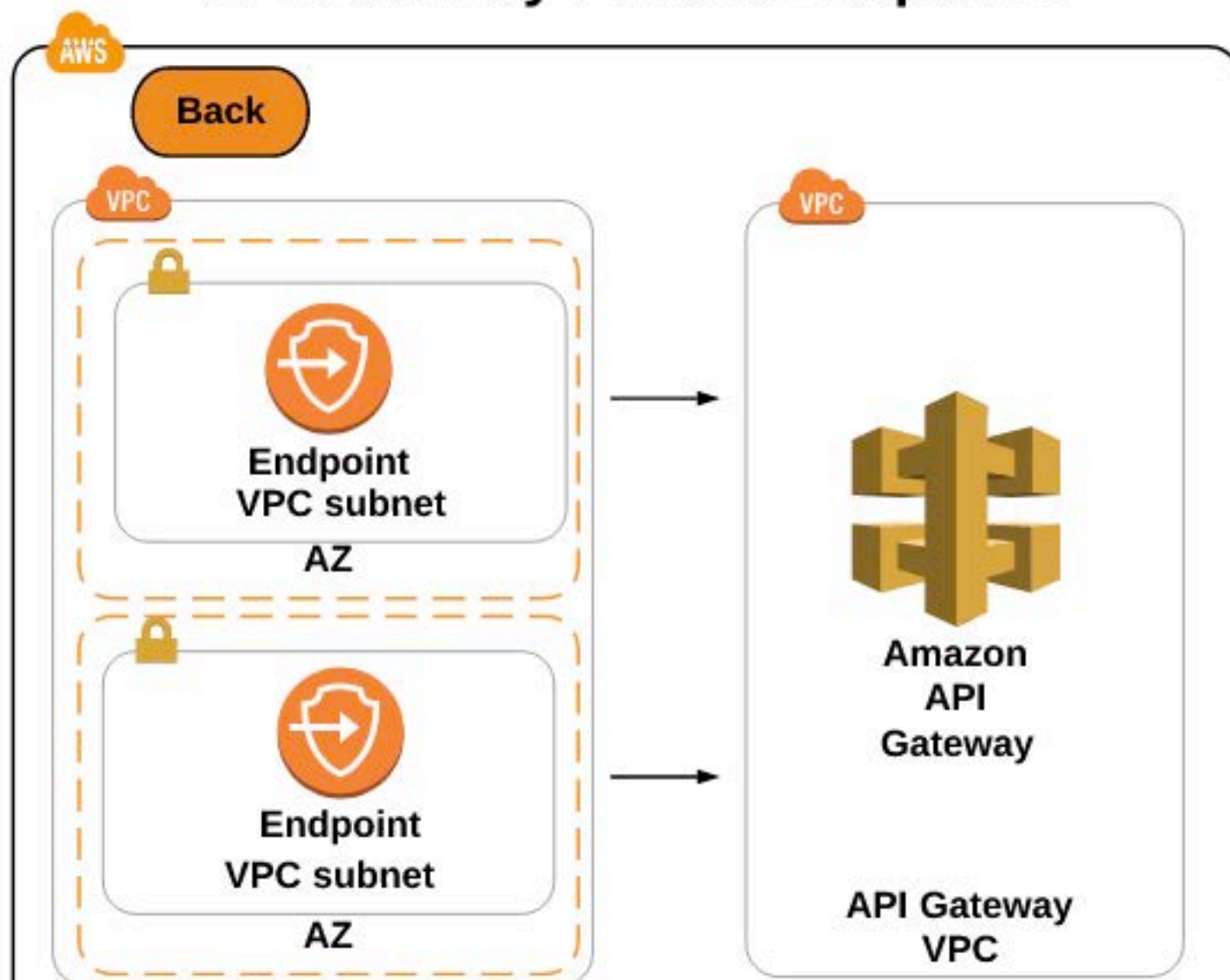
Deployment Pipelines

API Gateway

Lambda

## API Gateway

### API Gateway Private Endpoints



#### Private Endpoint Requirements

- A VPC with at least one subnet and DNS resolution enabled.
- A VPC Endpoint configured as such:
  - Service name = "com.amazonaws.{region}.execute-api"
  - Enable Private DNS Name = enabled
  - A security group set to allow TCP Port 443 inbound from either an IP range in your VPC or another security group in your VPC
- An API in API Gateway with:
  - Endpoint Type = "Private"
  - A resource policy giving access to your API from the VPC endpoint

## Choose a Topic

Introduction

Auto Scaling Deployment Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

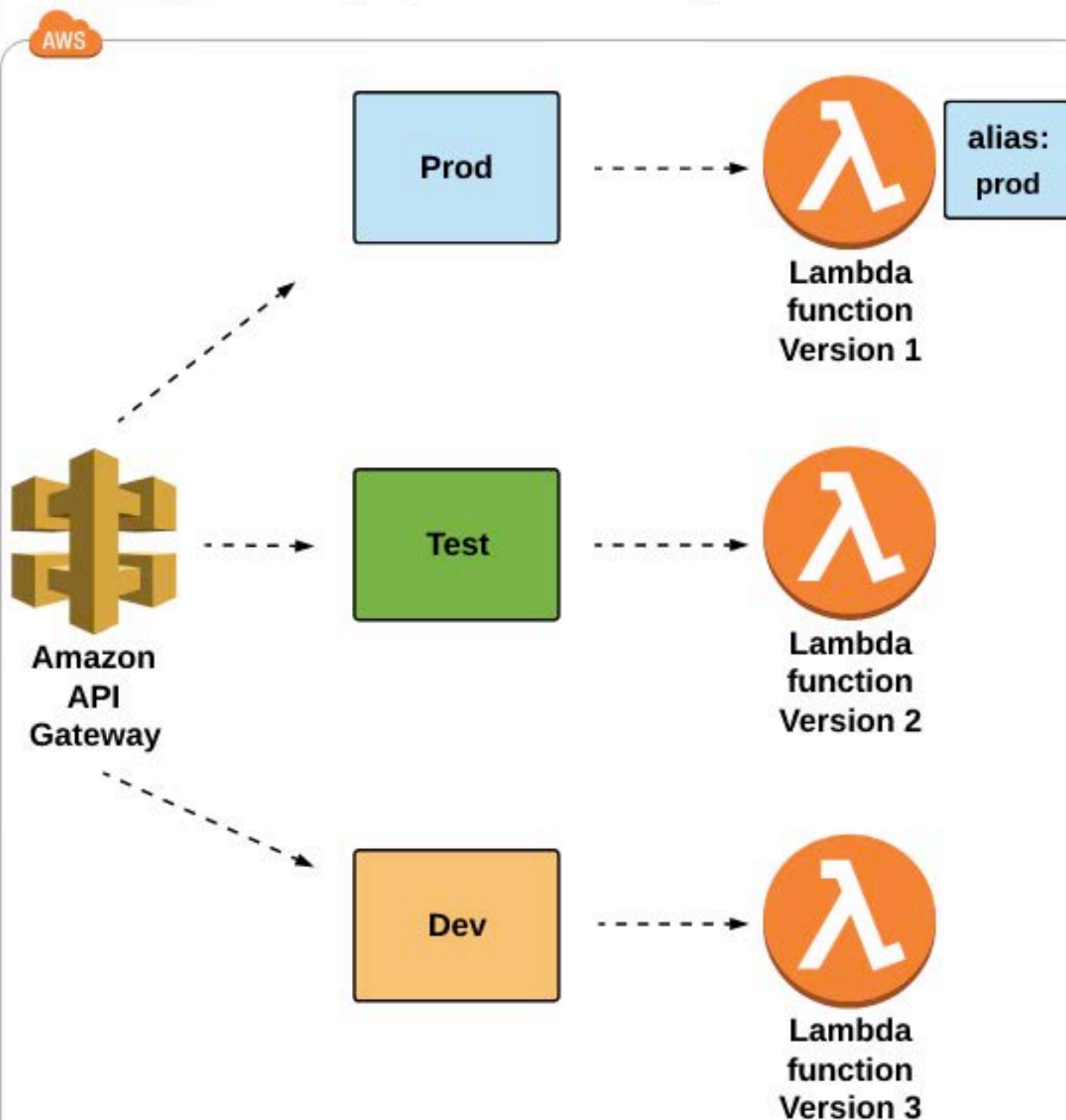
API Gateway

Lambda

## Amazon API Gateway

Back

### API Gateway with Lambda -Deployment/Versioning





Choose a Topic

Introduction

Auto Scaling Deployment  
Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With  
OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

Lambda

## Serverless Computing With AWS Lambda

AWS Lambda is a compute service that lets you run code without provisioning or managing servers. AWS Lambda executes your code only when needed and scales automatically, from a few requests per day to thousands per second. You pay only for the compute time you consume - there is no charge when your code is not running. With AWS Lambda, you can run code for virtually any type of application or backend service - all with zero administration. AWS Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code monitoring and logging.

How Lambda Works

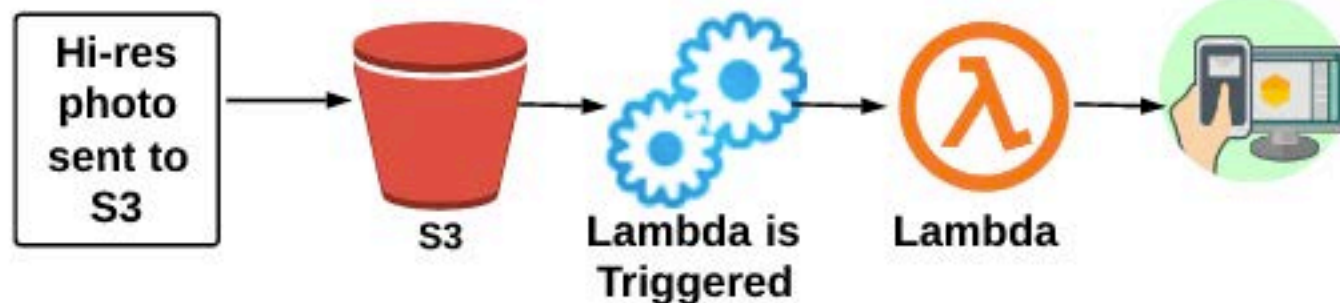
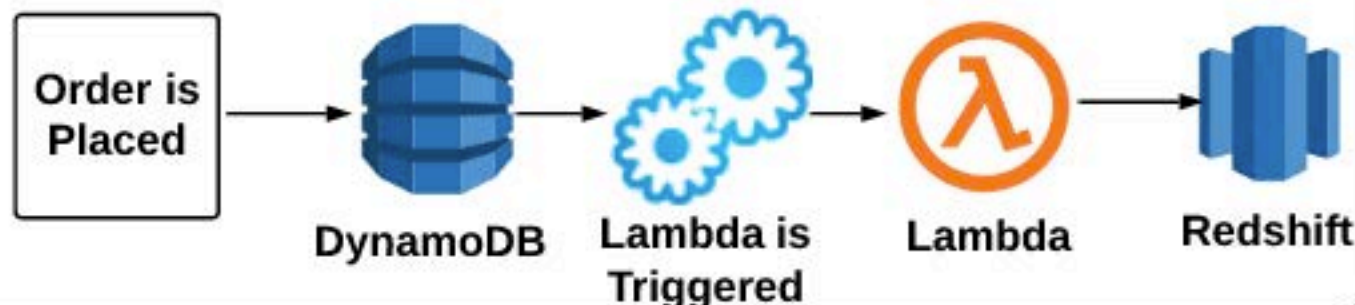
Lambda an Inside Look

Lambda Walkthrough

Lambda Versioning

AWS

## Lambda Converts HiRes photos to Mobile Size

ETL Using  
Lambda



Choose a Topic

Introduction

Auto Scaling Deployment  
Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With  
OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

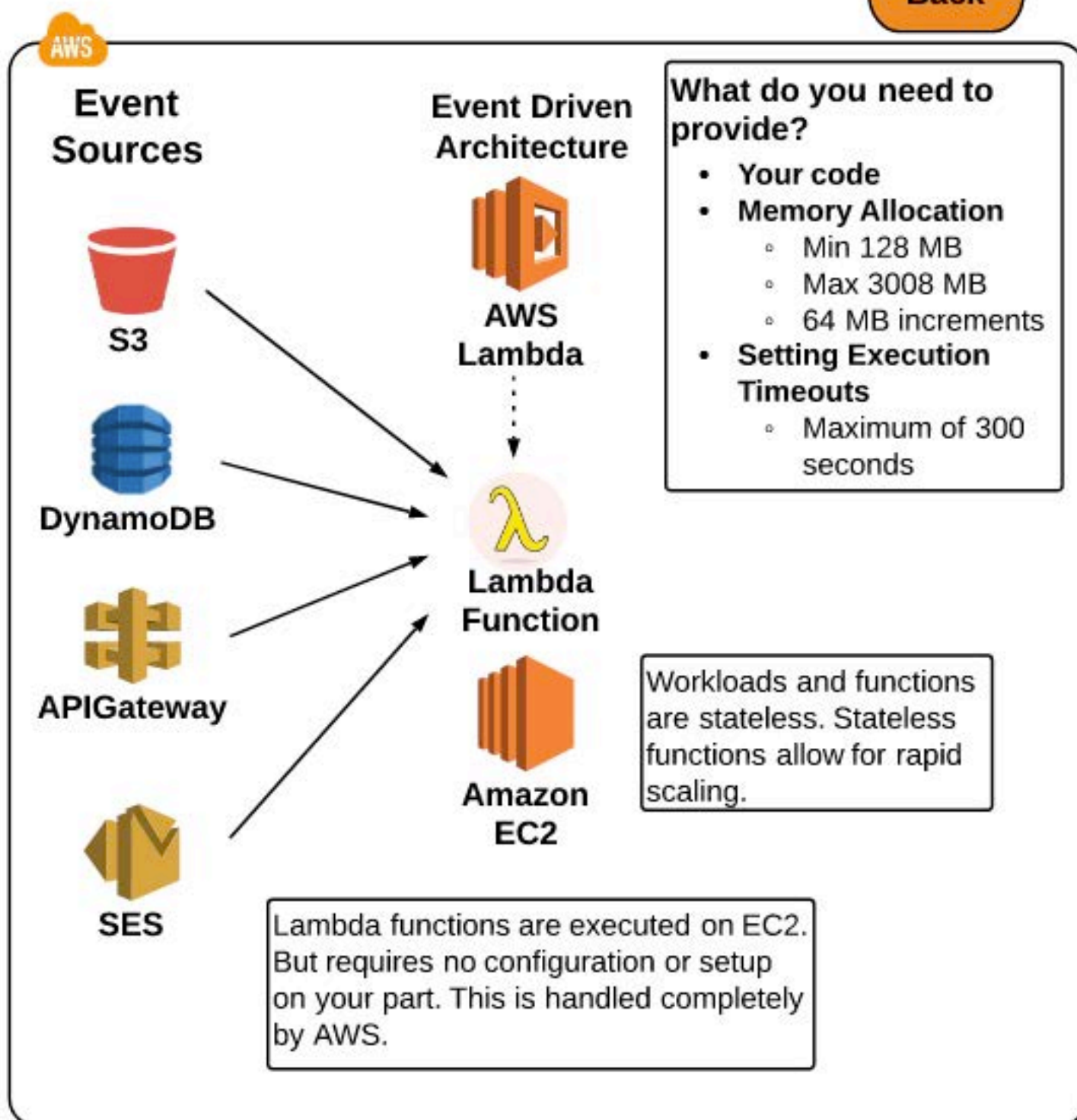
API Gateway

Lambda

## Serverless Computing With AWS Lambda

**How does Lambda work?** You upload your code to AWS Lambda. You DO NOT have to provision any servers. AWS Lambda handles provisioning. Lambda utilizes an event-driven architecture. An event "triggers" the execution of your code on AWS Lambda. Examples of events that trigger Lambda functions include: files placed in an S3 Bucket, DynamoDB table updates, SNS notifications, SQS messages, and several more.

Back





## Choose a Topic

Introduction

Auto Scaling Deployment Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

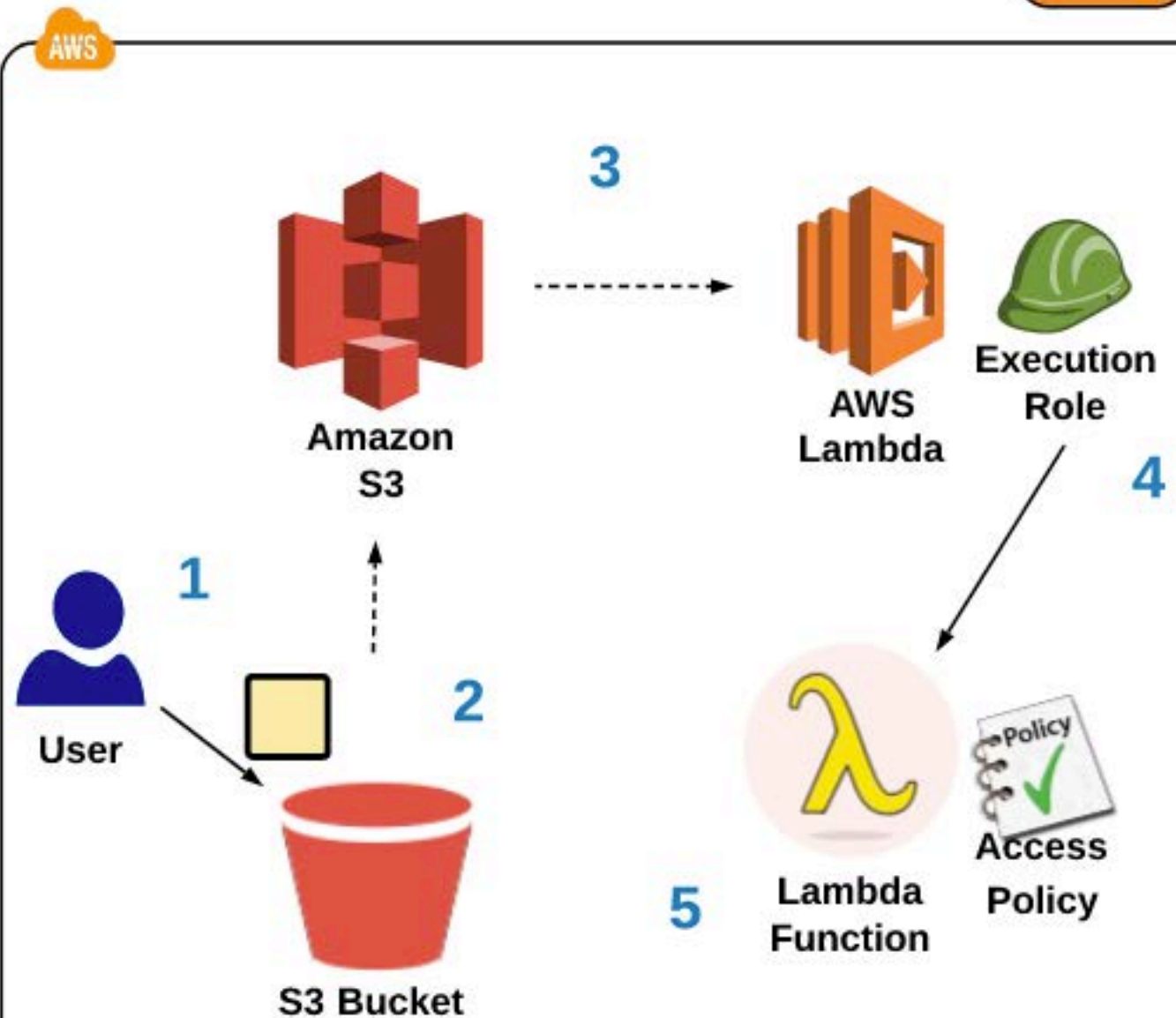
API Gateway

Lambda

## Serverless Computing With AWS Lambda

### Lambda an Inside Look

[Back](#)



1. User uploads an object to an S3 bucket (object-created event).
2. Amazon S3 detects the object-created event.
3. Amazon S3 invokes a Lambda function that is specified in the bucket notification configuration.
4. AWS Lambda executes the Lambda function by assuming the execution role that you specified at the time you created the Lambda function.
5. The Lambda function executes.



## Choose a Topic

Introduction

Auto Scaling Deployment  
Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With  
OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

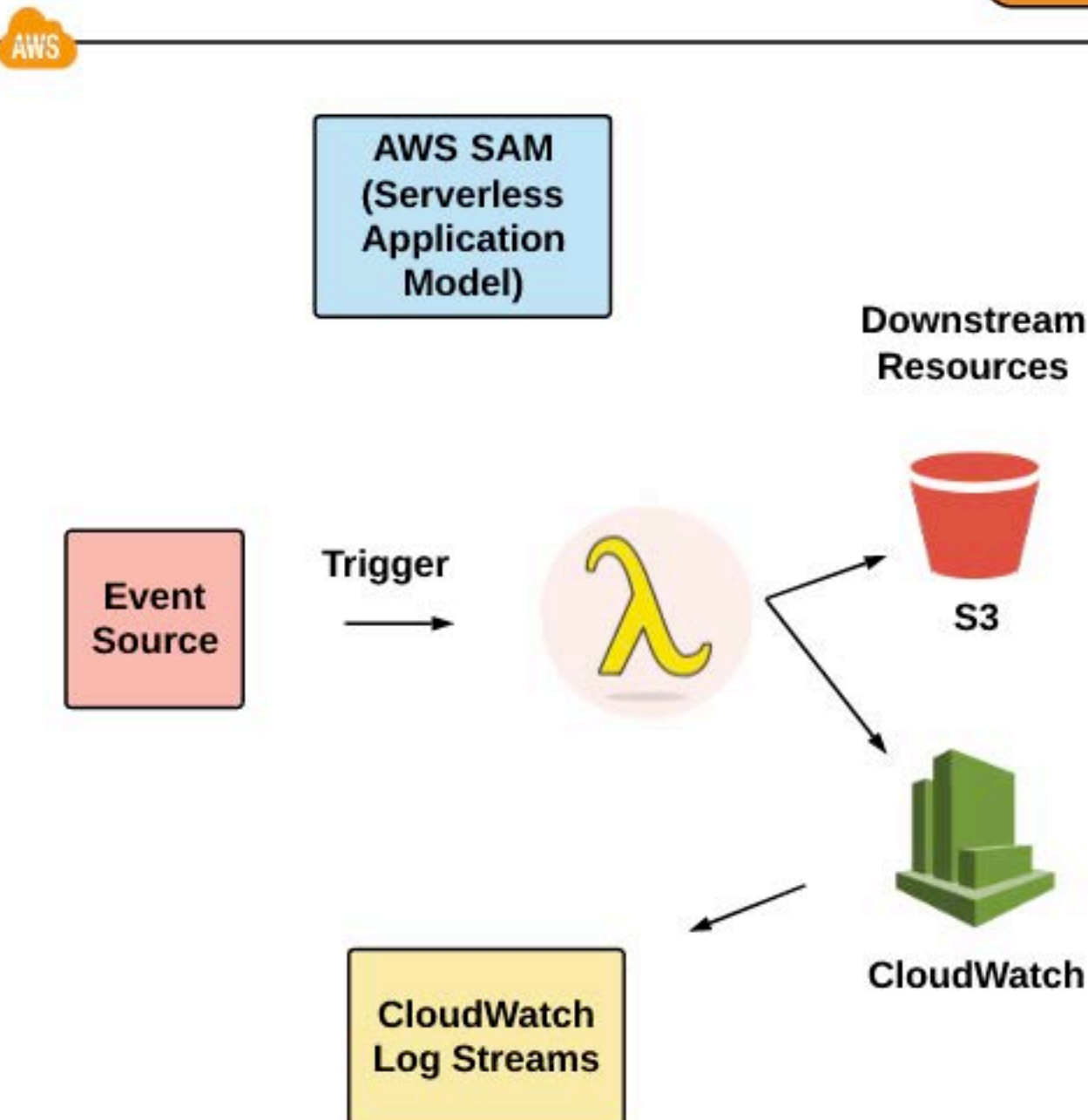
API Gateway

Lambda

## Serverless Computing With AWS Lambda

### Lambda Core Components

[Back](#)



### Lambda Requirements

- The Lambda Function (Code and Deployment Package)
- Memory Size Specification
- Execution Timeout
- IAM Role (The Execution Role)
- Event Source Mapping



Choose a Topic

Introduction

Auto Scaling Deployment  
Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With  
OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

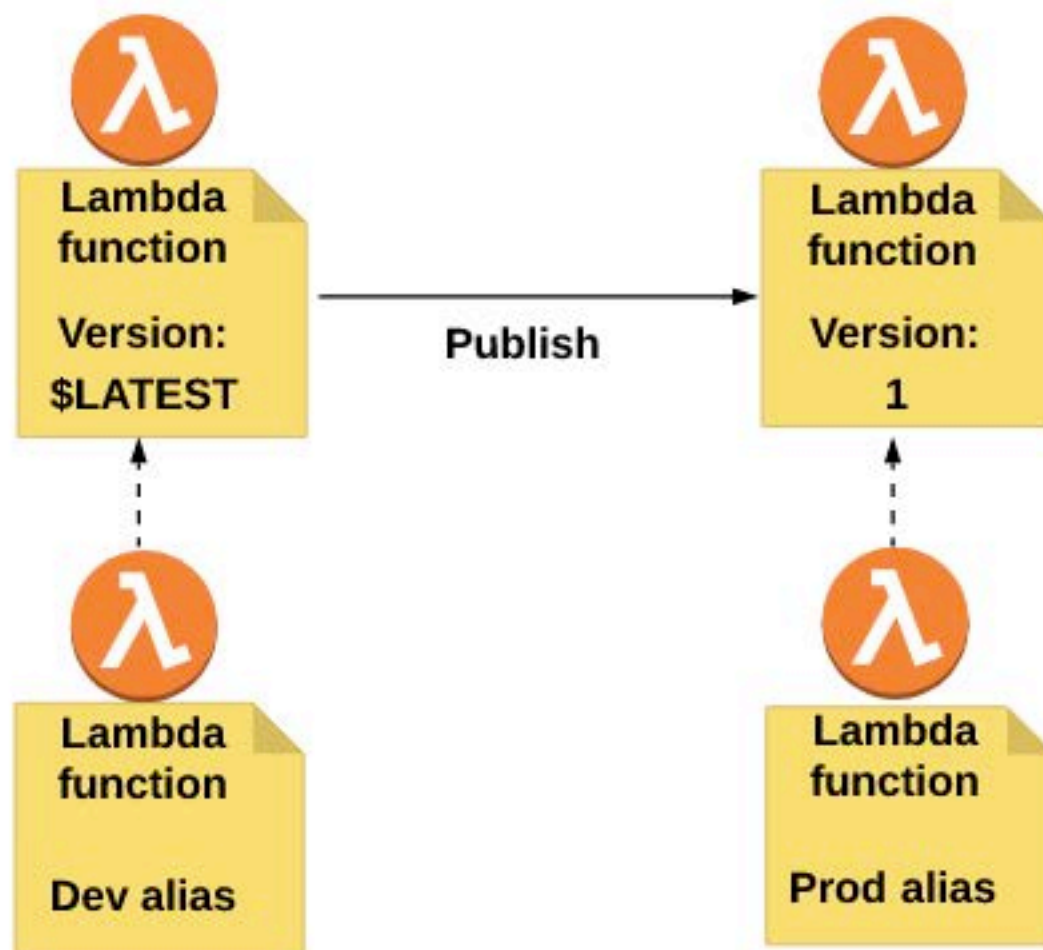
Lambda

## Serverless Computing With AWS Lambda

Back

## AWS Lambda Versioning

AWS

Lambda Versioning and Aliases

- You can publish one or more versions of your Lambda function.
- Each Lambda Function has a unique ARN.
- After you publish a version it can not be changed.
- The latest version of the function is tagged as \$LATEST
- Lambda supports creating aliases for each Lambda function version.
- An alias is a pointer to a specific Lambda function version.
- Each alias has a unique ARN.
- Aliases can be modified. You can update an alias to point to a different version.



Choose a Topic

Introduction

Auto Scaling Deployment  
Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With  
OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

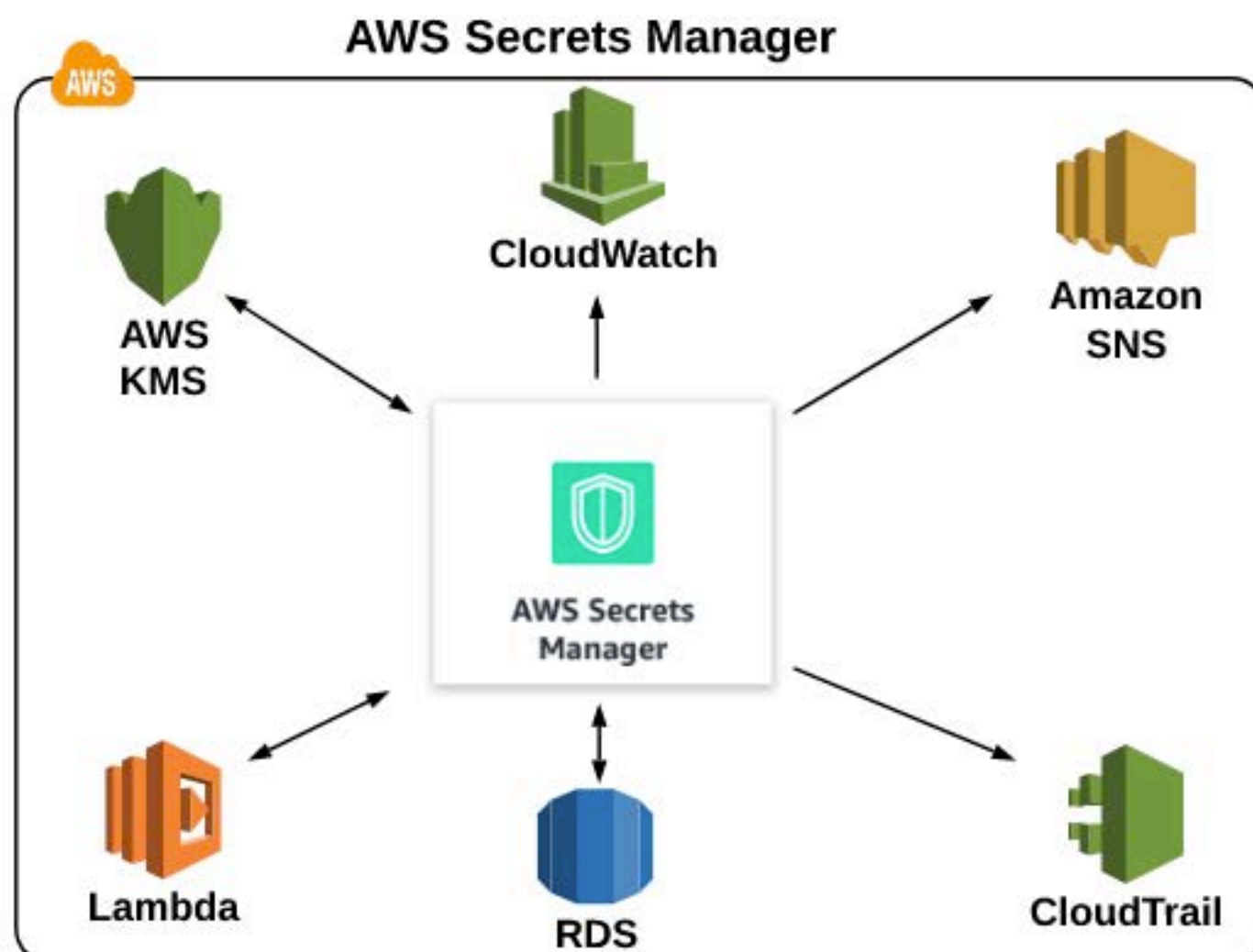
Lambda

Secrets Manager

## AWS Secrets Manager

AWS Secrets Manager is an AWS service that makes it easier for you to manage secrets. Secrets can be database credentials, passwords, third-party API keys, and even arbitrary text. You can store and control access to these secrets centrally by using the Secrets Manager console, the Secrets Manager command line interface (CLI), or the Secrets Manager API and SDKs.

## How Secrets Manager Works





Choose a Topic

Introduction

Auto Scaling Deployment  
Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With  
OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

Lambda

Secrets Manager

## AWS Secrets Manager

## Secrets Manager and How It Works?

Back

- AWS Secrets Manager is an AWS service that makes it easier for you to manage secrets. Secrets can be database credentials, passwords, third-party API keys, and even arbitrary text. You can store and control access to these secrets centrally by using the Secrets Manager console, the Secrets Manager command line interface (CLI), or the Secrets Manager API and SDKs.
- In the past, when you created a custom application that retrieves information from a database, you typically had to embed the credentials (the secret) for accessing the database directly in the application. When it came time to rotate the credentials, you had to do much more than just create new credentials. You had to invest time to update the application to use the new credentials. Then you had to distribute the updated application. If you had multiple applications that shared credentials and you missed updating one of them, the application would break.
- Secrets Manager enables you to replace hardcoded credentials in your code (including passwords), with an API call to Secrets Manager to retrieve the secret programmatically. This helps ensure that the secret can't be compromised by someone examining your code, because the secret simply isn't there. Also, you can configure Secrets Manager to automatically rotate the secret for you according to a schedule that you specify. This enables you to replace long-term secrets with short-term ones, which helps to significantly reduce the risk of compromise.

