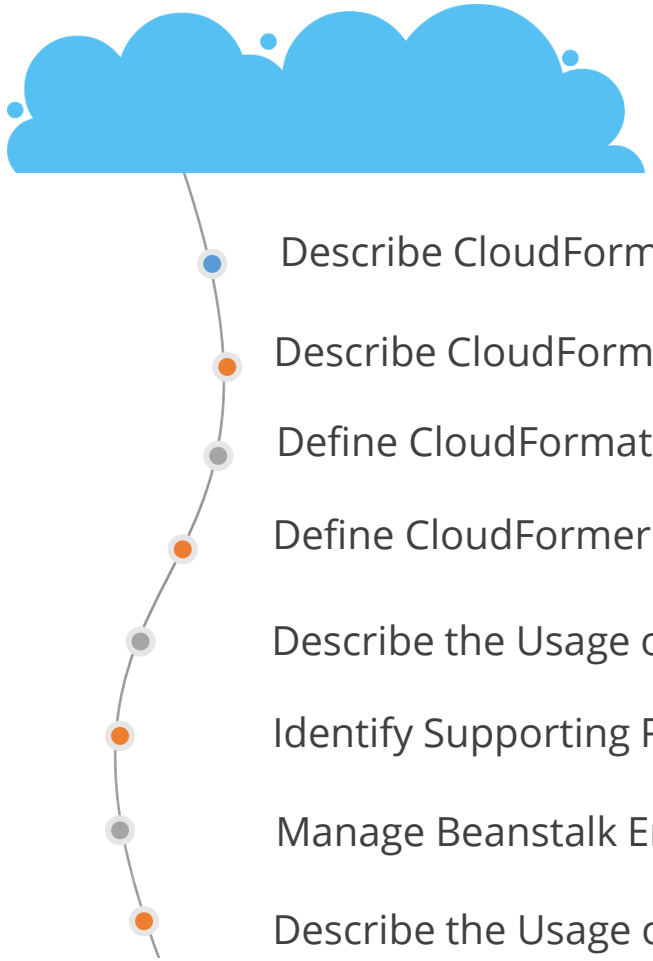
AWS Certified Developer Associate

Lesson 9: Management Tools and Lambda

simplilearn

What You'll Learn



- Describe CloudFormation
- Describe CloudFormation Template
- Define CloudFormation Stacks
- Define CloudFormer
- Describe the Usage of Elastic Beanstalk
- Identify Supporting Platforms of Beanstalk
- Manage Beanstalk Environments
- Describe the Usage of Lambda

Basic Concepts of CloudFormation

AWS CloudFormation

- Automated infrastructure provisioning tool
- Creates and manages a collection of AWS resources
- Enables infrastructure as a code
- Simplifies infrastructure management
- Helps to quickly replicate your infrastructure

CloudFormation Template

```
Resources:
  HelloBucket:
    Type: AWS::S3::Bucket
```

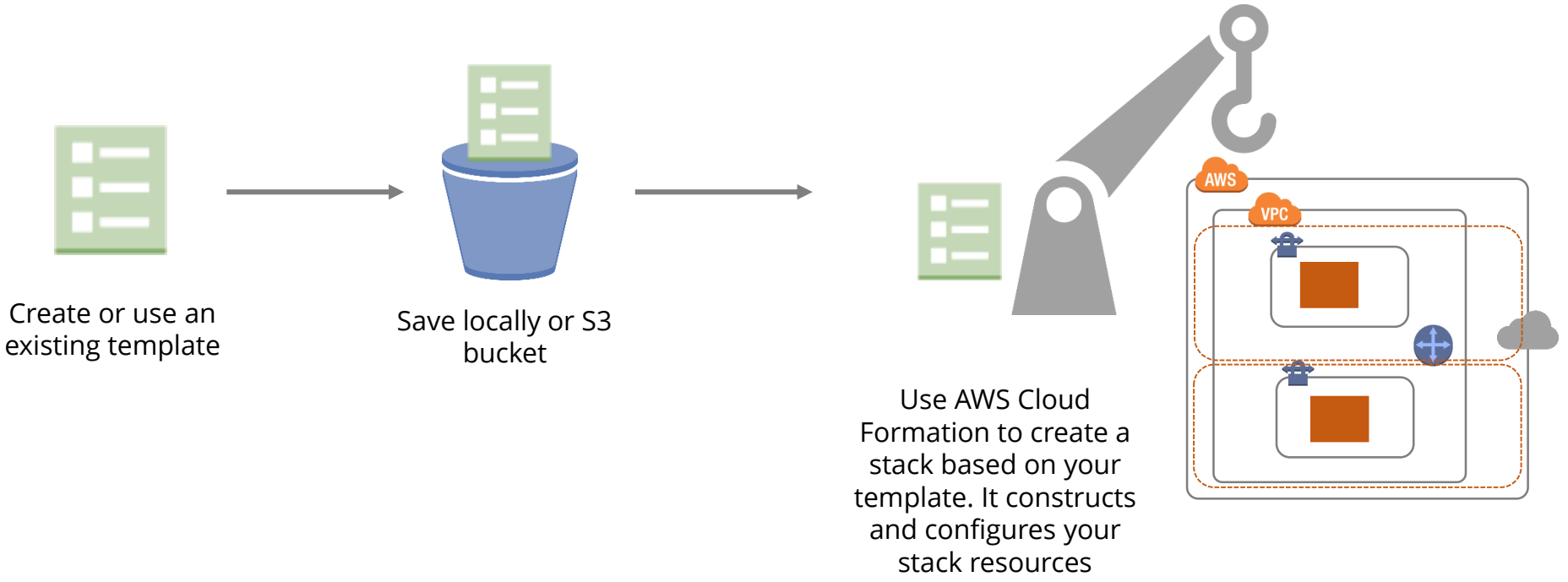
JSON Syntax

```
{
    "Resources" : {
        "HelloBucket" : {
            "Type" : "AWS::S3::Bucket"
        }
    }
}
```

YAML

```
{
    "Resources" : {
        "HelloBucket" : {
            "Type" : "AWS::S3::Bucket",
            "Properties" : {
               "AccessControl" : "PublicRead"
            }
        }
    }
}
```

CloudFormation Template (Contd.)



Create or use an existing template

Save locally or S3 bucket

Use AWS Cloud Formation to create a stack based on your template. It constructs and configures your stack resources

Template Anatomy

```
"Parameters" : {
  "InstanceTypeParameter" : {
    "Type" : "String",
    "Default" : "t2.micro",
    "AllowedValues" : ["t2.micro", "m1.small", "m1.large"],
    "Description" : "Enter t1.micro, m1.small, or m1.large. Default is t1.micro."
  }
}
```

Development

Production

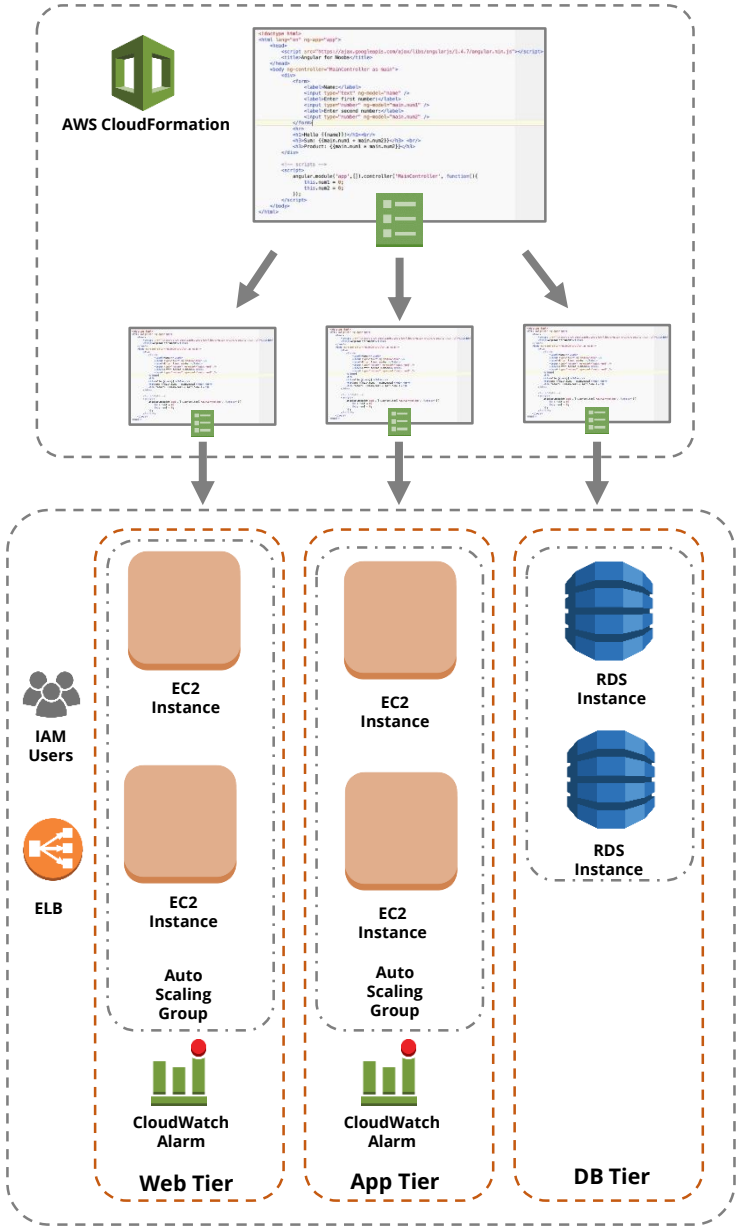
```
"Mappings" : {
  "Mapping01" : {
    "Key01" : {
      "Name" : "Value01"
    },
    "Key02" : {
      "Name" : "Value02"
    },
    "Key03" : {
      "Name" : "Value03"
    }
  }
}
```

Template Anatomy (Contd.)


```
"Outputs" : {
  "BackupLoadBalancerDNSName" : {
    "Description": "The DNSName of the backup load balancer",
    "Value" : { "Fn::GetAtt" : [ "BackupLoadBalancer", "DNSName" ]},
    "Condition" : "CreateProdResources"
  },
  "InstanceID" : {
    "Description": "The Instance ID",
    "Value" : { "Ref" : "EC2Instance" }
  }
}
```

Working with Stacks

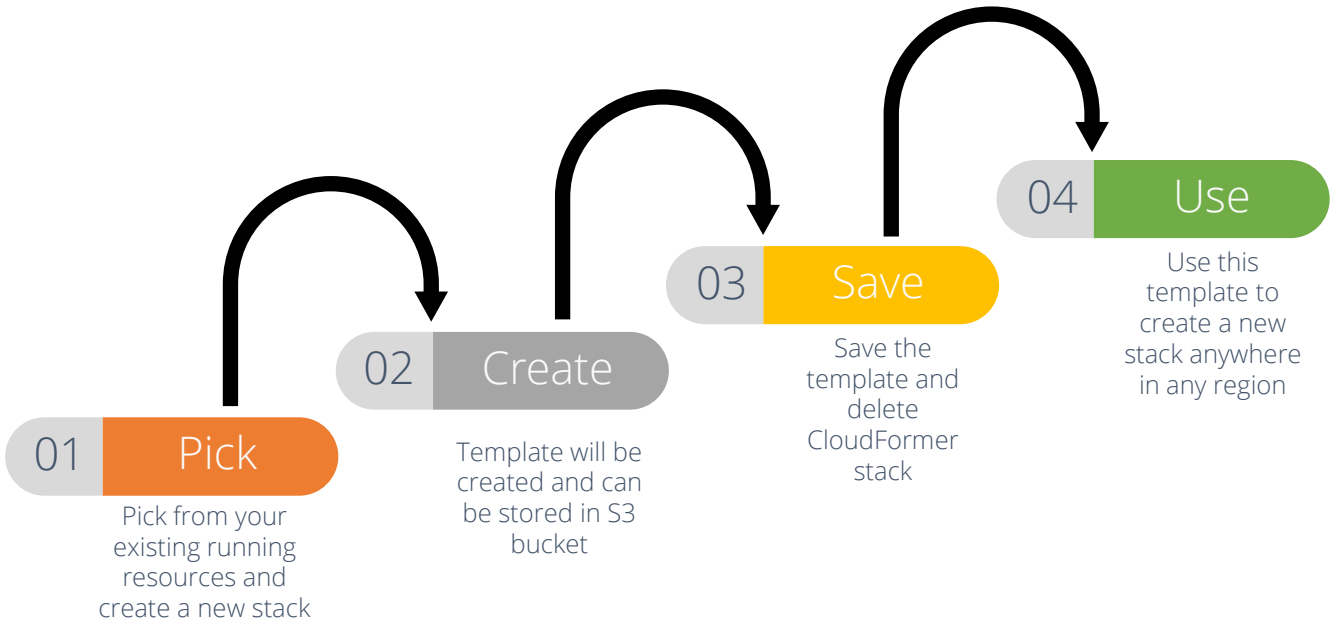
- A stack is a collection of AWS resources in a single unit
- It supports multiple resources
- It provides the capability to revert to the previous version
- It removes all resources during stack update failure
- Its policy manages eligibility to apply and impact resources



simplilearn

CloudFormer

CloudFormer is a tool that helps create a template from existing AWS architecture in your account.



CloudFormer Usage Scenario

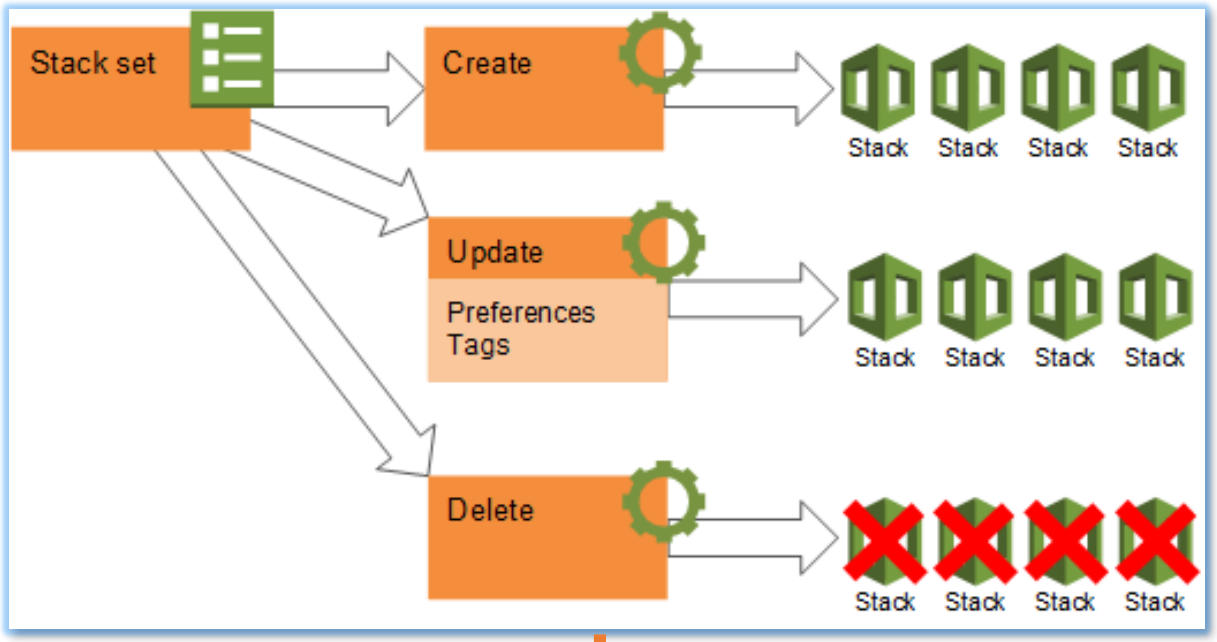
Use Case: You have created a three-tier project that includes EC2 web server, multiple EC2 app servers for high availability, and AWS RDS MySql database server. Along with this, you would have the following in your application architecture:

- VPC
- Security Groups to protect your servers
- Network ACLs as second layer of security
- Elastic IPs
- CloudWatch Alarms
- Load Balancers
- Route 53 to map server IP to domains

Requirements: Replicate the above architecture in a new region for disaster recovery

Solution: CloudFormer is the tool that will help you create a template of the above architecture and you can easily create a new stack using that template in a new region.

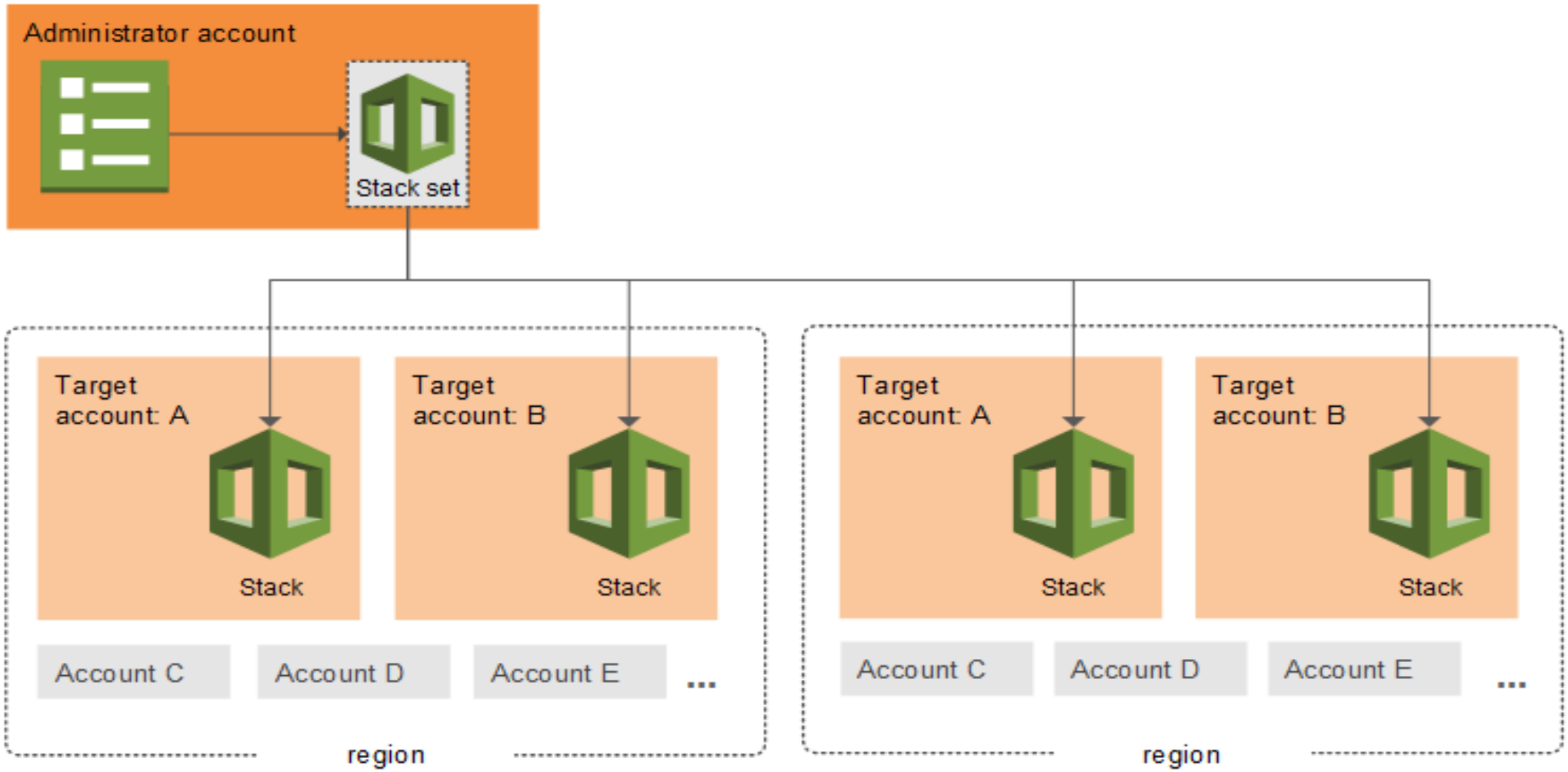
CloudFormation Stack Sets



AWS CloudFormation StackSets allows you to manage your stacks by enabling you to update or delete your existing stacks across multiple accounts and regions as a single entity

Cross-account stacks are managed by establishing trust between administrator account (in which stack set is created) and the target account (whose stack we need to update)

Stack Set Architecture



Knowledge Check

KNOWLEDGE CHECK

Which are the valid sections of the CloudFormation Template? (Choose 2)

a. Outputs

b. Inputs

c. Parameters

d. Events

KNOWLEDGE CHECK	Which are the valid sections of the CloudFormation Template? (Choose 2)

a. Outputs

b. Inputs

c. Parameters

d. Events

The correct answer is **Outputs & Parameters**

Explanation: Parameters is an optional section, but very helpful. It is used to customize the template by passing values to the template while it's being created. Output is an optional section. It gives the details of the value that are returned whenever the stack property is viewed.

KNOWLEDGE CHECK	You need to create similar architecture in three different accounts and further in two regions under each account. Which of the following will solve the purpose?

a. CloudFormation templates can be copied, and stacks can be launched independently.

b. CloudFormer will help create template from stacks. This can then be migrated to different accounts and regions.

c. CloudFormation StackSets is best for this scenario as it will allow you to manage everything as a single entity.

d. Designer tool will create templates in an easy way. This can then be taken to other places.

KNOWLEDGE CHECK	You need to create similar architecture in three different accounts and further in two regions under each account. Which of the following will solve the purpose?

a. CloudFormation templates can be copied, and stacks can be launched independently.

b. CloudFormer will help create template from stacks. This can then be migrated to different accounts and regions.

c. CloudFormation StackSets is best for this scenario as it will allow you to manage everything as a single entity.

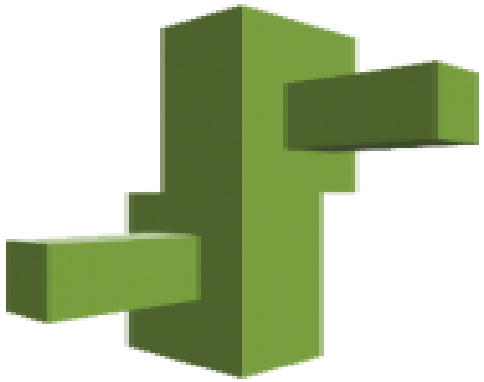
d. Designer tool will create templates in an easy way. This can then be taken to other places.

The correct answer is **c**

AWS CloudFormation StackSets allows you to manage your stacks by enabling you to update or delete your existing stacks across multiple accounts and regions as a single entity.

Basic Concepts of Elastic Beanstalk

AWS Elastic Beanstalk

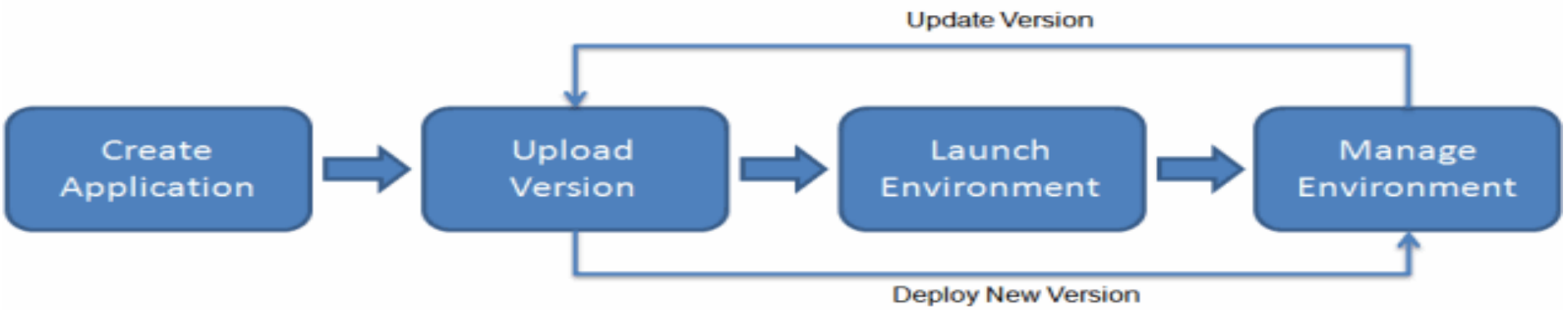


Service for deploying and scaling web applications and services

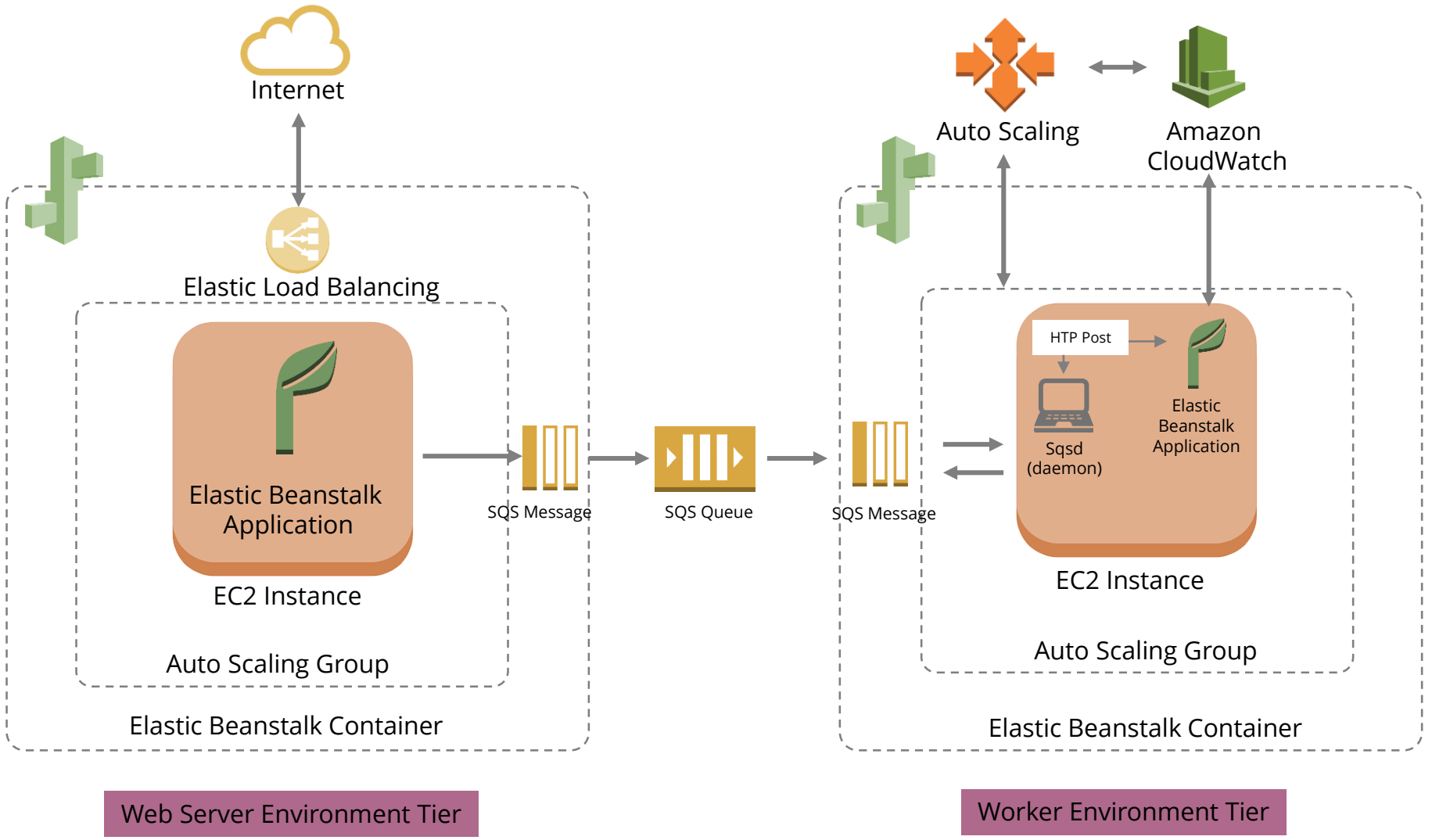
- Supports applications in various languages
- Analyzes uploaded code, creates required stack
- Uses application version in a deployable bundle
- Provides control over AWS resources to the users
- Users can control and monitor the environment

Elastic Beanstalk – Core Components

Application	Collection of various Beanstalk components
Collection of AWS resources	**Environment**



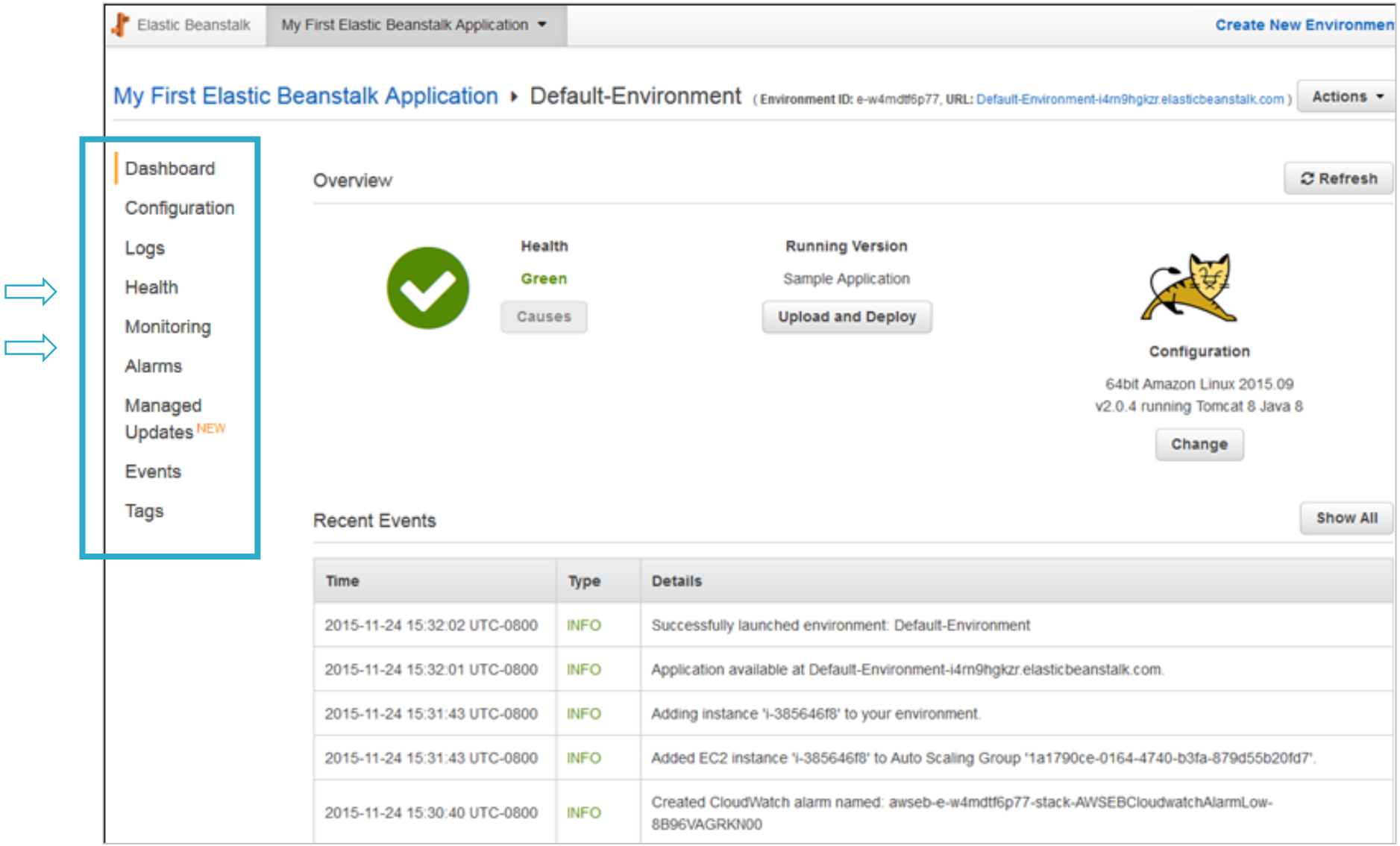
Elastic Beanstalk – Architecture



Supporting Platforms

Single Container Docker	Multicontainer Docker	Preconfigured Docker	Go
Java SE	Java with Tomcat	.NET on Windows Server with IIS	Node.js
PHP	Python	Ruby	

Managing your Environment



Knowledge Check

KNOWLEDGE CHECK

Which are the programing languages that Elastic Beanstalk supports? (Choose 2)

a. C ++

b. Java

c. Ruby

d. Visual Basic

KNOWLEDGE CHECK

Which are the programing languages that Elastic Beanstalk supports? (Choose 2)

a. C ++

b. Java

c. Ruby

d. Visual Basic

The correct answer is **Java & Ruby**

Explanation: Elastic Beanstalk supports programing languages: Java, PHP, Python, Ruby, and Go.

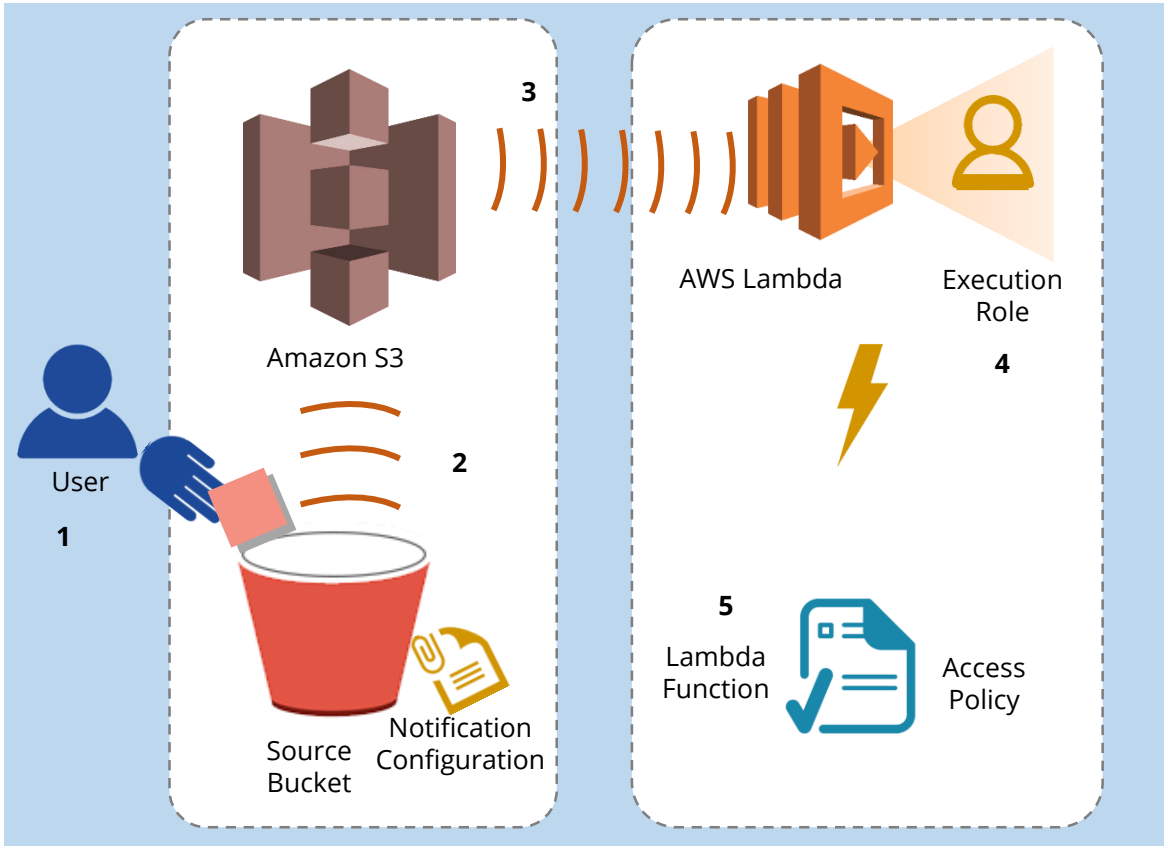
Basic Concepts of Lambda

Managing your Environment

- Runs code without any provisioning or configuration
- Runs code virtually for any application
- Currently supports Node.js, Java, and Python
- Ideal to work with any data triggers from DynamoDB, S3, or Kinesis

Functions of Lambda

1. Package your code in a deployable package
2. Deploy your code and create a Lambda function
3. Monitor your Lambda function



Knowledge Check

KNOWLEDGE CHECK

When you use AWS Lambda, which of the following resources do you control?

a. EC2 Instances

b. Elastic Load Balancer

c. SQS Queues

d. None

KNOWLEDGE CHECK

When you use AWS Lambda, which of the following resources do you control?

a. EC2 Instances

b. Elastic Load Balancer

c. SQS Queues

d. None

The correct answer is **None**

Explanation: You don't have control on any underlying resources, and can't change the operating system or language at runtime.

Practice Assignment: Cloud Formation

To create cloud formation stack and update the stack

Create and update a stack

You will need to create a stack in CloudFormation and then update the stack using CloudFormation from the AWS Management Console.

Prerequisites:

- AWS account.
- Create an initial stack from the sample template, save it locally on your system as a text file. Keep a note of the location where the file is saved.

Task:

To create and update a stack in AWS CloudFormation.

Quiz

QUIZ 1	In CloudFormation's output section, what attribute should you use to get an ELB's URL?

a. DNSURL

b. DNSName

c. DNSEndPoint

d. None of the above

QUIZ 1	In CloudFormation's output section, what attribute should you use to get an ELB's URL?

a. DNSURL

b. DNSName

c. DNSEndPoint

d. None of the above

The correct answer is **DNSName**

Explanation: You can use the "GetAtt" function to get the attribute value of "DNSName."

QUIZ 2	In the CloudFormation template, which is the required section?

a. Parameters

b. Output

c. Resources

d. Conditions

QUIZ 2	In the CloudFormation template, which is the required section?

a. Parameters

b. Output

c. Resources

d. Conditions

The correct answer is **Resources**

Explanation: Templates are made up of eight sections, among them only the Resources section is mandatory, while the other seven sections are optional.

QUIZ 3	What are the two types of environment tiers in Elastic Beanstalk?

a. Web server environment tier

b. Worker environment tier

c. Backend environment tier

d. Frontend environment tier

QUIZ 3	What are the two types of environment tiers in Elastic Beanstalk?

a. Web server environment tier

b. Worker environment tier

c. Backend environment tier

d. Frontend environment tier

The correct answer is **Web server environment tier & Worker environment tier**

Explanation: There are two types of environment tiers: Web server environment tier to handle HTTP requests, and worker environment tier to handle backend tasks.

QUIZ 4	Which programing language does Lambda currently support?

a. Ruby

b. Perl

c. Java

d. Go

QUIZ 4	Which programing language does Lambda currently support?

a. Ruby

b. Perl

c. Java

d. Go

The correct answer is **Java**

Explanation: Lambda currently supports Node.js, Java, and Python programing languages, so your code should be written in one of these languages.

Key Takeaways

- Cloud formation uses stacks that are templates to fully provision and configure the specified resources. Cloud formation templates are text files made of JSON or YAML standard stored in S3.
- Cloud formation lets you save your template with versioning enabled, it saves different versions of the same template, to track the changes done in each version.
- Elastic Beanstalk supports applications developed in Java, PHP, .NET, Node.js, Python, and Ruby and container types of languages.
- Elastic beanstalk lets you use the management console's Beanstalk environment page to monitor and control your various environments.
- AWS Lambda is a compute service where you can upload your code, and let the service run your code using AWS resources without you provisioning or configuring it.
- Lambda code is written in Node.js, Java, and Python programing languages. Amazon Kinesis is a fully managed AWS service used for real-time processing of large streams of data.
- Amazon Kinesis supports multiple applications to access the same live-stream data concurrently and independently.

This concludes “Management Tools, Lambda, and Kinesis”.

The next lesson is “Networking Overview/VPC”