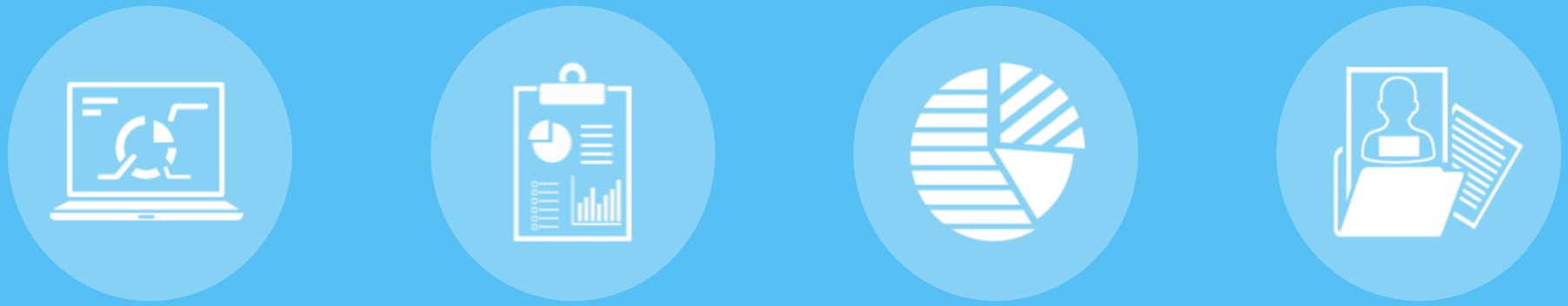
AWS Technical Essentials

Lesson 6—Deployment and Management



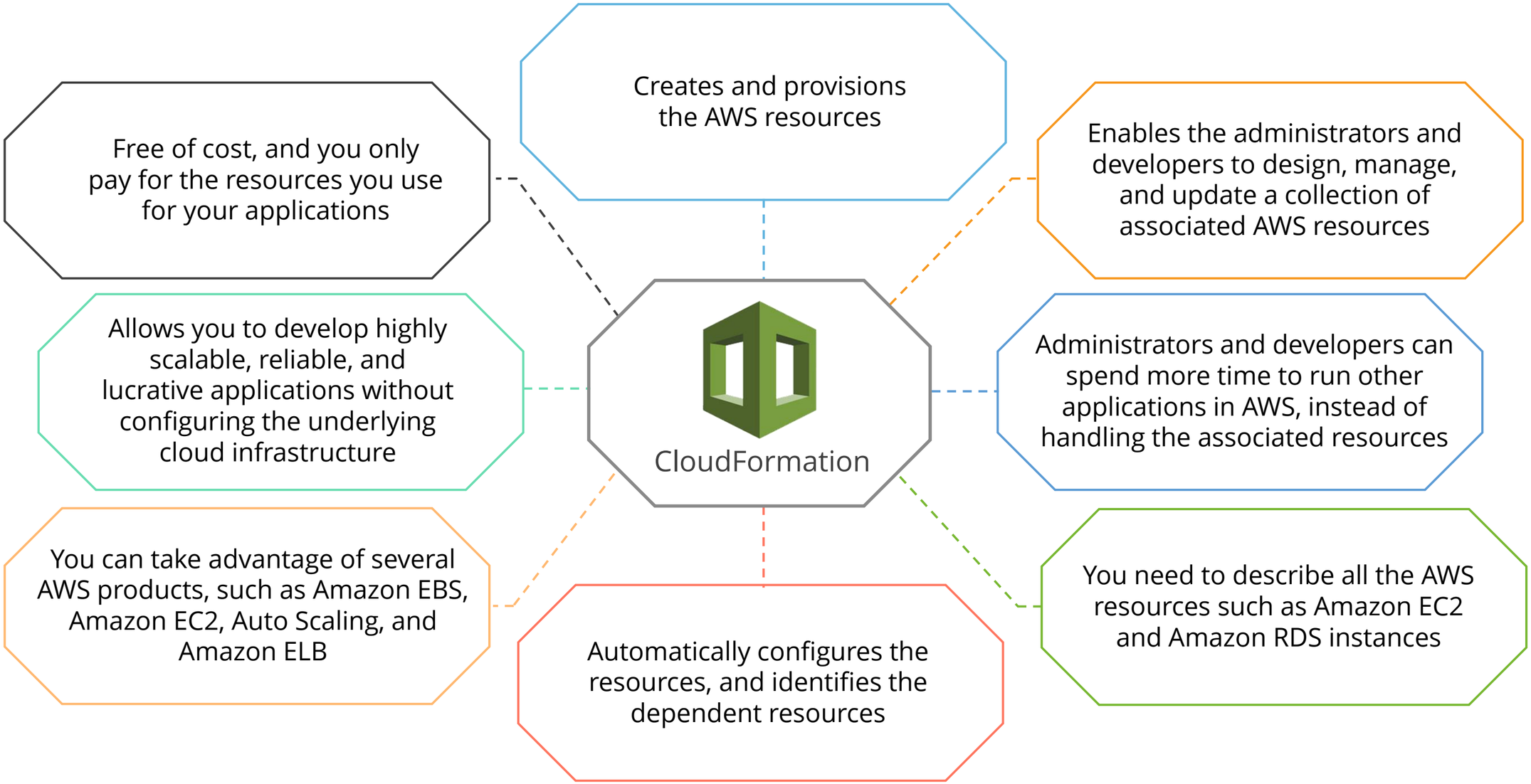
simplilearn

WHAT YOU'LL LEARN

- AWS CloudFormation
- Creating templates and stacks to configure the resources in Amazon CloudFormation
- Amazon CloudWatch, Metrics, and Alarms
- Amazon Identity and Access Management (IAM)

AWS CloudFormation

AWS CloudFormation—Introduction



Three Key Benefits of Amazon CloudFormation

You are ensured of easy control and tracking of the infrastructure

You get to work with a simplified infrastructure management

You can easily and quickly replicate your infrastructure across regions

CloudFormation Components—Templates

The key components of AWS CloudFormation are templates and stacks.

template1

```
1  {
2    "AWSTemplateFormatVersion": "2010-09-09",
3    "Description": "AWS CloudFormation Sample Template VPC_Single_Instance_In_Subnet.",
4    "Parameters": {
5      "InstanceType": {
6        "Description": "WebServer EC2 instance type",
7        "Type": "String",
8        "Default": "t2.micro",
9        "AllowedValues": [
10         "t1.micro",
11       ],
12       "ConstraintDescription": "must be a valid EC2 instance type."
13     },
14     "KeyName": {
15       "Description": "Name of an existing EC2 KeyPair to enable SSH access to the instance.",
16       "Type": "AWS::EC2::KeyPair::KeyName",
17       "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
18     },
19
20   "Mappings": {
21     "AWSInstanceType2Arch": {
22       "t1.micro": {
23         "Arch": "PV64"
24       },
25
26     },
27   "Resources": {
28     "VPC": {
29       "Type": "AWS::EC2::VPC",
30       "Properties": {
31         "EnableDnsSupport": "true",
32         "EnableDnsHostnames": "true",
33         "CidrBlock": "10.0.0.0/16"
34       },
35     },
36
37     "InternetGateway": {
38       "Type": "AWS::EC2::InternetGateway",
39       "Metadata": {
40         "AWS::CloudFormation::Designer": {
41           "id": "a166c4f5-7cc4-429b-b9d8-2c8c43facc63"
42         }
43       }
44     },
```

- Describes resources, their properties, and runtime parameters
- Conforms to the JavaScript Object Notation
- Depicts the AWS infrastructure
- File extension: .txt, .json, or .template
- A blueprint for creating and configuring your resources

Features of a Template

CloudFormation Components—Stacks

A Stack is an array of resources manageable as a single unit.

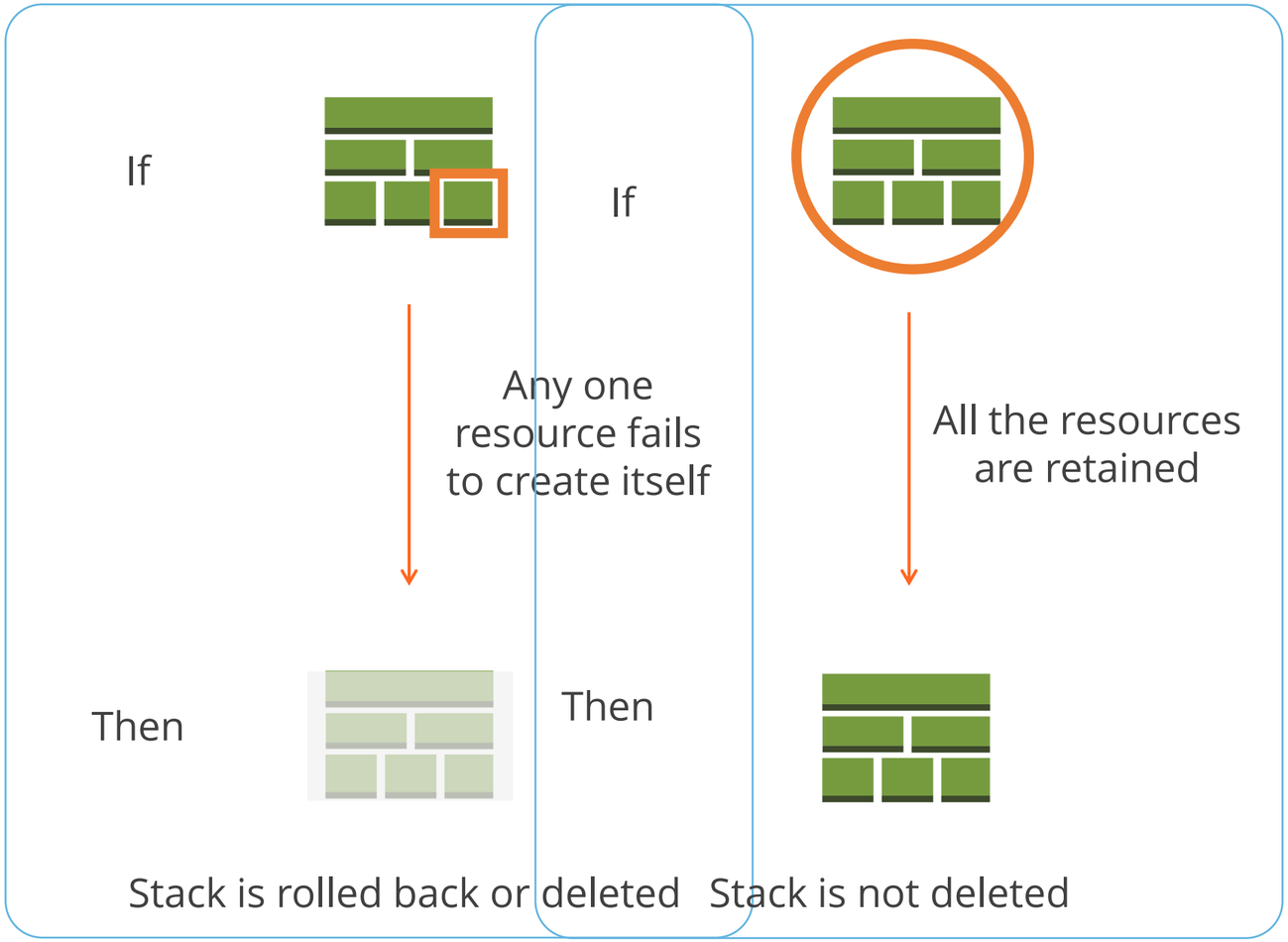
A stack can have all the resources needed to run a Web application, such as a Database or a Web Server.

Stack

AWS CloudFormation creates, manages, and updates a collection of resources by creating, managing, and updating stacks.

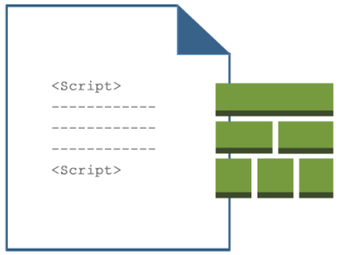
CloudFormation Components—Stacks

A Stack is handled as a single unit.



Template—Example

The AWS CloudFormation Console allows creating and updating templates along with the related collection of resources or stacks.



Stacks with Template

Describes

- Amazon EC2 instance
- Instance type
- Key pair name
- EBS volume
- Additional resources

Append

Template—Example

Creating Resources

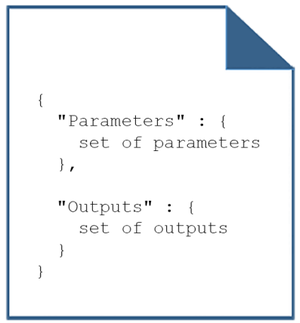
- Submit the template
- Form a stack
- Configure resources

Updating Stack

- Modify or update the resources
- Make necessary changes to the template
- Re-submit to AWS CloudFormation

Reusable Templates

You can create a template to reuse it.



Reusable Template

Specify input parameters

Determine parameter values at the time of creating the stack

Template Anatomy

template1

Open Brace

```
1  {
2    "Parameters": {
3      "InstanceType": {
4        "Description": "WebServer EC2 instance type",
5        "Type": "String",
6        "Default": "t2.micro",
7        "AllowedValues": [
```

```
65        "MinLength": "9",
66        "MaxLength": "18",
67        "Default": "0.0.0.0/0",
68        "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
69        "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
70      }
71    }
72  }
```

Closed Brace

Components | Template

Eight Sections of Template Structure

1
2
3
4
5
6
7
8

FormatVersion: It mentions the AWS template version.

Syntax:

```
"AWSTemplateFormatVersion" : "2010-09-09"
```

Example:

template1

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "AWS CloudFormation Sample Template VPC_Single_Instance_In_Subnet: Sample template showing how to create a VPC and add an EC2 instance with an Elastic IP address and a security group. **WARNING** This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.",
  "Parameters": {
    "InstanceType": {
      "Description": "WebServer EC2 instance type",
      "Type": "String",
      "Default": "t2.micro",
```

Eight Sections of Template Structure

1
2
3
4
5
6
7
8

Description: This follows the FormatVersion section, and contains a text string that describes the template.

Syntax:

```
"Description" : "Here are some details about the template."
```

Example:

template1

```
1 {
2   "AWSTemplateFormatVersion": "2010-09-09",
3   "Description": "AWS CloudFormation Sample Template VPC_Single_Instance_In_Subnet: Sample template showing how to create a VPC
and add an EC2 instance with an Elastic IP address and a security group. **WARNING** This template creates an Amazon EC2
instance. You will be billed for the AWS resources used if you create a stack from this template.",
4   "Parameters": {
5     "InstanceType": {
6       "Description": "WebServer EC2 instance type",
7       "Type": "String",
8       "Default": "t2.micro",
```

Eight Sections of Template Structure

1
2
3
4
5
6
7
8

Metadata Section: It provides extra information about the template.

Syntax:

```
"Metadata" : {
  "Instances" : {"Description" : "Information about the instances"},
  "Databases" : {"Description" : "Information about the databases"}
}
```

Example:

```
283        "VpcId": {
284          "Ref": "VPC"
285        }
286      },
287      "Metadata": {
288        "AWS::CloudFormation::Designer": {
289          "id": "3df467ad-673c-4c48-a41c-3ac1626961e3"
290        }
291      }
292    },
293    "InternetGateway": {
294      "Type": "AWS::EC2::InternetGateway",
295      "Metadata": {
296        "AWS::CloudFormation::Designer": {
297          "id": "a166c4f5-7cc4-429b-b9d8-2c8c43facc63"
298        }
299      }
300    },
```

Eight Sections of Template Structure

1
2
3
4
5
6
7
8

Parameters Section: It contains values to be passed at runtime to the template. These values are passed while creating or updating a stack.

Syntax:

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

Example:

```
1  {
2    "AWSTemplateFormatVersion": "2010-09-09",
3    "Description": "AWS CloudFormation Sample Template VPC_Single_Instance_In_Subnet: Sample template showing how to create a VPC and add an EC2 instance with an Elastic IP address and a security
     group. **WARNING** This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.",
4    "Parameters": {
5      "InstanceType": {
6        "Description": "WebServer EC2 instance type",
7        "Type": "String",
8        "Default": "t2.micro",
9        "AllowedValues": [
10         "t1.micro",
11         "t2.micro",
12       ],
13       "ConstraintDescription": "must be a valid EC2 instance type."
14     },
15     "KeyName": {
```

Eight Sections of Template Structure

1
2
3
4
5
6
7
8

Mappings Section: It includes mapping of keys. Their values are similar to a lookup table.

Syntax:

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

Example:

```
29    },
30    "Mappings": {
31      "AWSInstanceType2Arch": {
32        "t1.micro": {
33          "Arch": "PV64"
34        },
35        "t2.micro": {
36          "Arch": "HVM64"
37        },
38        "t2.small": {
39          "Arch": "HVM64"
40        },
41        "t2.medium": {
42          "Arch": "HVM64"
43        },
44      }
45    }
```

Eight Sections of Template Structure

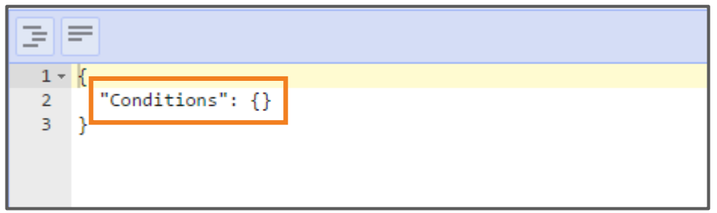
1
2
3
4
5
6
7
8

Conditions Section: It specifies the conditions for creating some specific resources only when they are fulfilled.

Syntax:

```
"Conditions" : {
  "Logical ID" : {Intrinsic function}
}
```

Example:



Eight Sections of Template Structure

1
2
3
4
5
6
7
8

Resources Section: It contains the stack resources along with their properties. Resources is the only mandatory section.

Syntax:

```
"Resources" : {
    "Logical ID" : {"Type" : "Resource type",
        "Properties" : {Set of properties}
    }
}
```

Example:

```
45    }
46    "Resources": {
47      "VPC": {
48        "Type": "AWS::EC2::VPC",
49        "Properties": {
50          "EnableDnsSupport": "true",
51          "EnableDnsHostnames": "true",
52          "CidrBlock": "10.0.0.0/16"
53        },
54        "Metadata": {
55          "AWS::CloudFormation::Designer": {
56            "id": "96a791f0-938b-4ebe-9f3c-b3fe2a588aee"
57          }
58        }
59      },
60    },
```

Eight Sections of Template Structure

1
2
3
4
5
6
7
8

Outputs Section: It contains the returned values while observing the stack's properties.

Syntax:

```
"Outputs" : {"Logical ID" : {
    "Description" : "Information about the value",
    "Value" : "Value to return"
  }
}
```

Example:

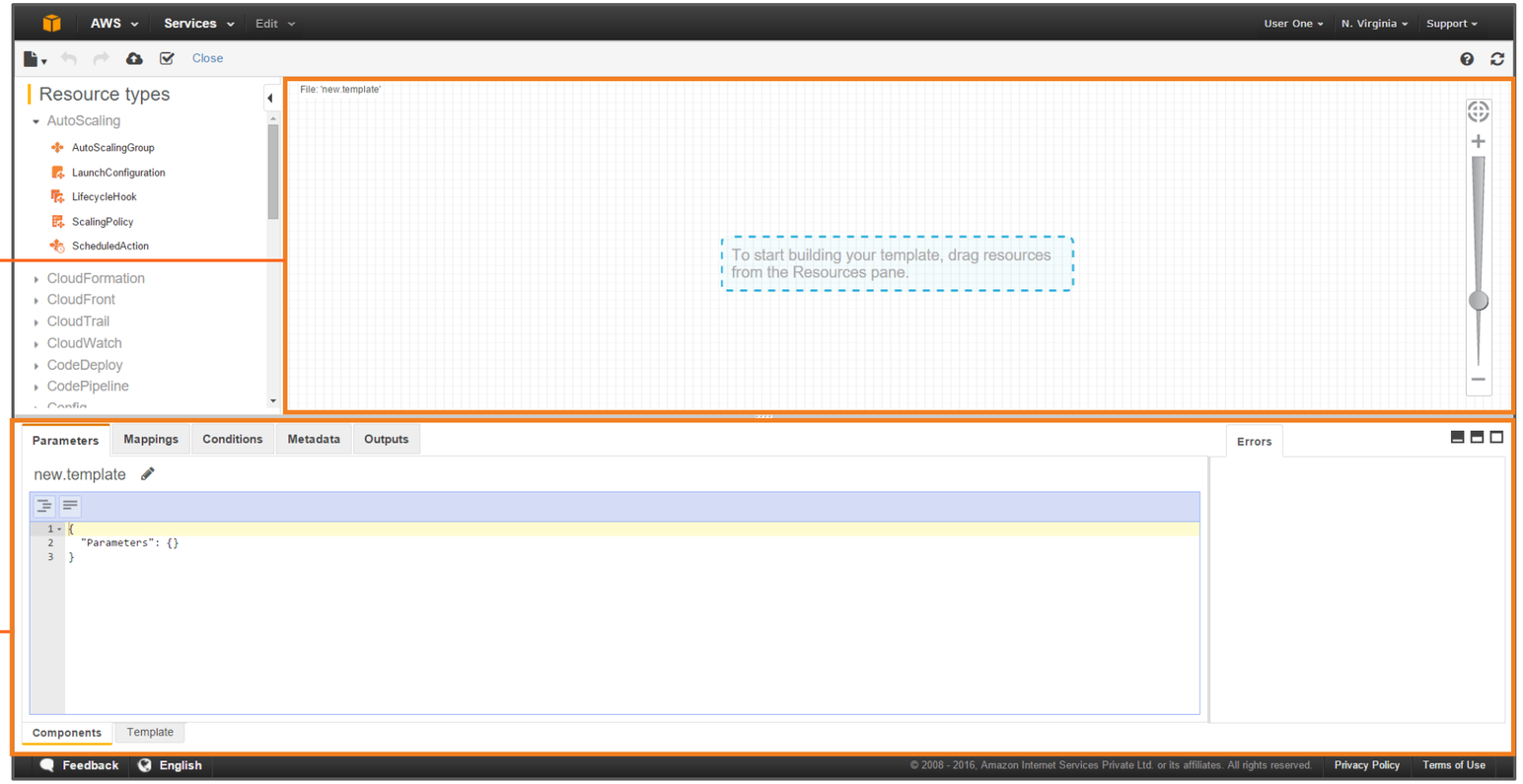
```
301   },
302   "Outputs": {
303     "URL": {
304       "Value": {
305         "Fn::Join": [
306           "",
307           [
308             "http://",
309             {
310               "Fn::GetAtt": [
311                 "WebServerInstance",
312                 "PublicIp"
313               ]
314             }
315           ]
316         ]
317       },
318       "Description": "Newly created application URL"
319     }
320   },
```

Creating a Template

To create a stack, create a template using either JSON Editor or AWS CloudFormation Designer tool. This enables you to create, view, and modify a template graphically.

AWS CloudFormation Designer tool

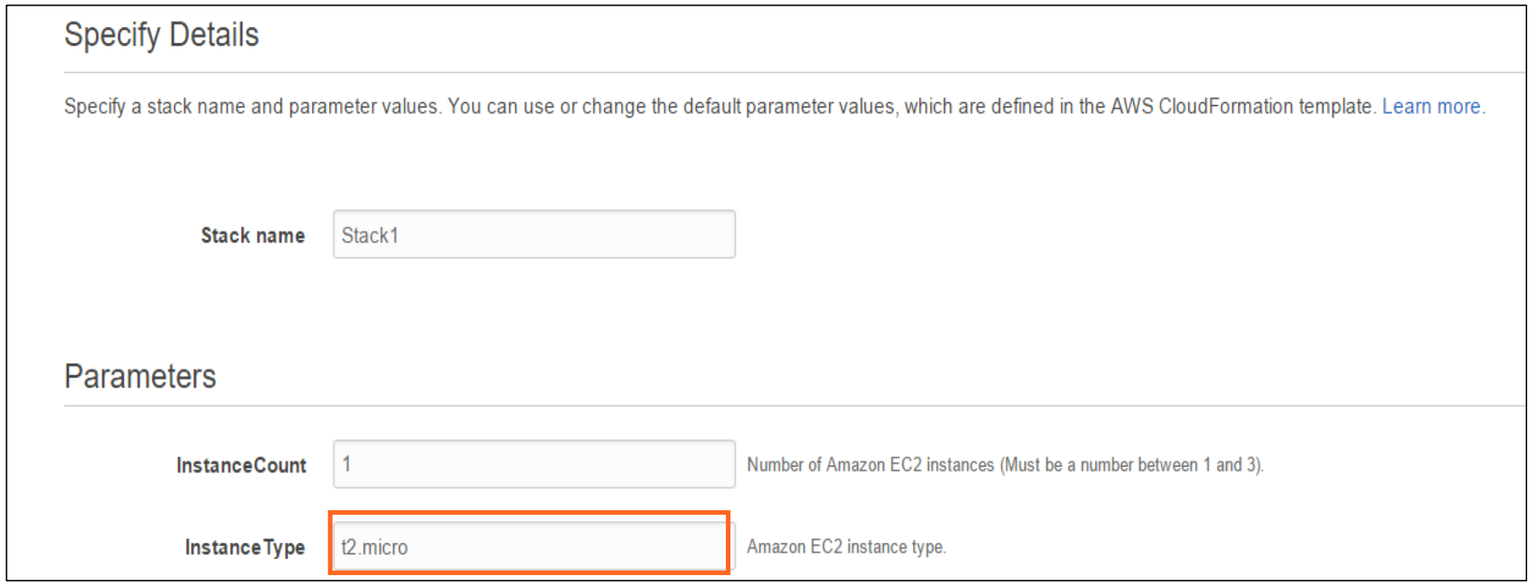
JSON Editor



Working of Amazon CloudFormation

When you create a stack, the **CloudFormation** service invokes the underlying services to configure the resources.

If you use a template describing an Amazon EC2 instance with a t2 dot micro type for creating a stack, the **CloudFormation** service invokes the Amazon EC2 create instance API, and states the type as t2 dot micro.



Steps to Create and Configure Resources

1. Create a template, or choose an existing template
2. Save the template with a suitable extension, Locally or in Amazon S3 bucket
3. Create a stack, and mention the template's location, which can be either your local computer or Amazon S3

Demo 01—Creating a Stack

(Refer to the E-Learning course: Screen Number – 6.3)

Knowledge Check

KNOWLEDGE CHECK 1	Which of the following services creates and provisions AWS resources?

a. AWS CloudWatch

b. AWS CloudFormation

c. AWS Identity and Access Management

d. AWS CloudFront

KNOWLEDGE CHECK

Which of the following services creates and provisions AWS resources?

a. AWS CloudWatch

b. AWS CloudFormation

c. AWS Identity and Access Management

d. AWS CloudFront

The correct answer is **b.**

Explanation: AWS CloudFormation service creates and provisions AWS resources.

KNOWLEDGE CHECK 2	What is handled as a single unit?

a. Templates

b. Units

c. Stack

d. Resources

KNOWLEDGE CHECK

What is handled as a single unit?

a. Templates

b. Units

c. Stack

d. Resources

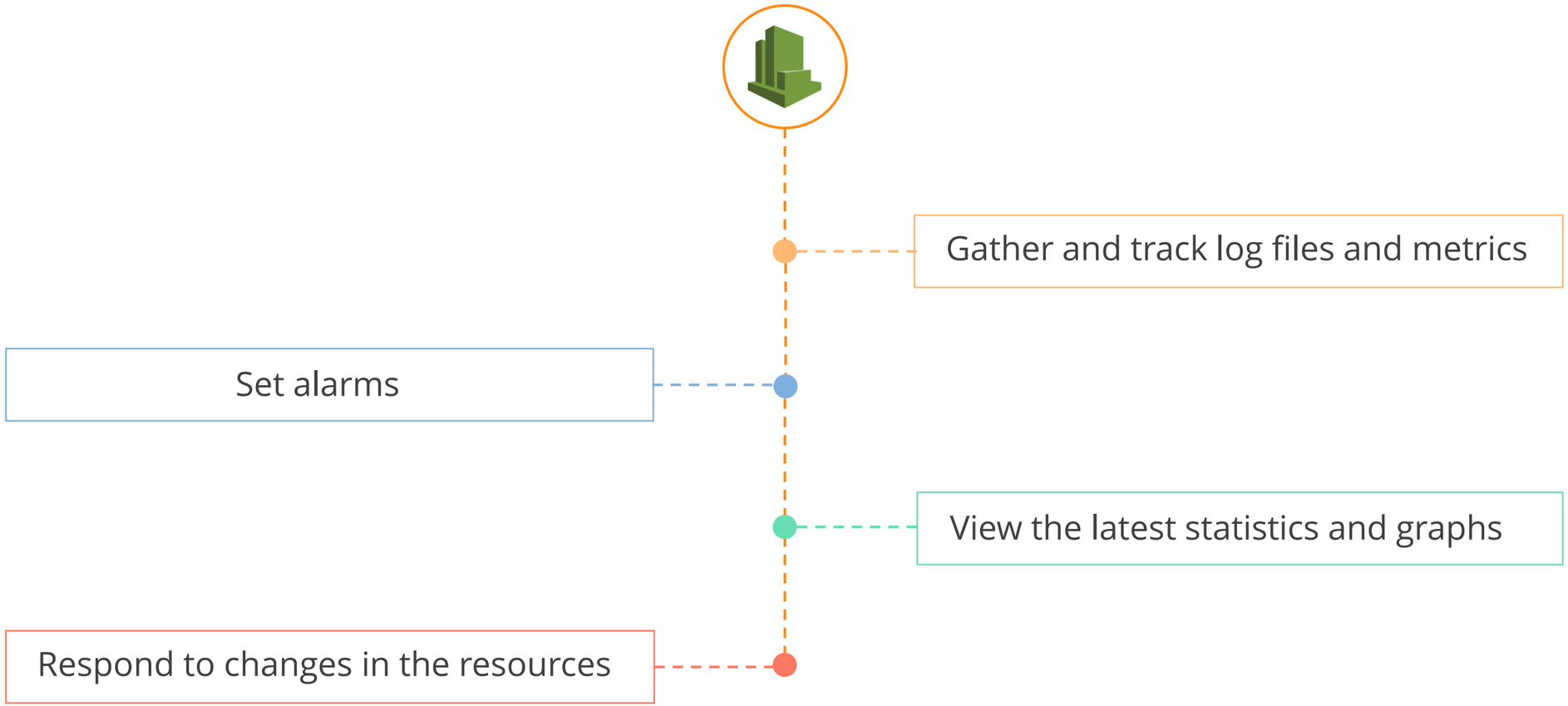
The correct answer is c.

Explanation: A stack is handled as a single unit.

Amazon CloudWatch Metrics

Introduction to Amazon CloudWatch

Amazon CloudWatch is a service that allows real-time monitoring of cloud resources such as, Amazon EC2, Amazon RDS instances, and other applications.



Goal of Amazon CloudWatch

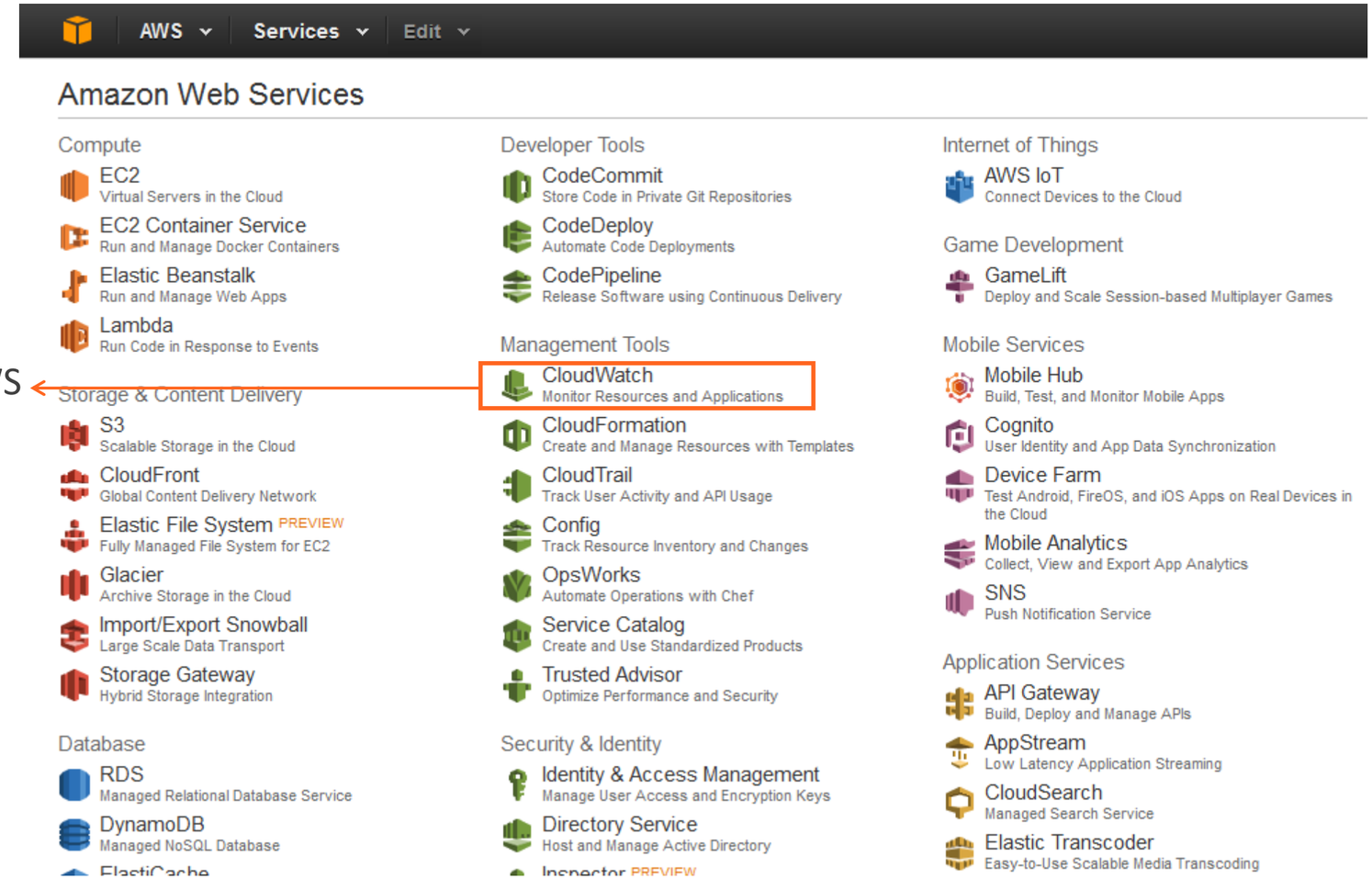
The goal of Amazon CloudWatch is to give system-wide insights into the usage of resources, performance of applications, and status of conducted tasks.

- View the trends
- Troubleshoot systems
- Set up an automated action

AWS CloudWatch eliminates the need to set up and manage your own monitoring systems.

Accessing Amazon CloudWatch

Click **CloudWatch**, under Management Tools in the AWS Management Console



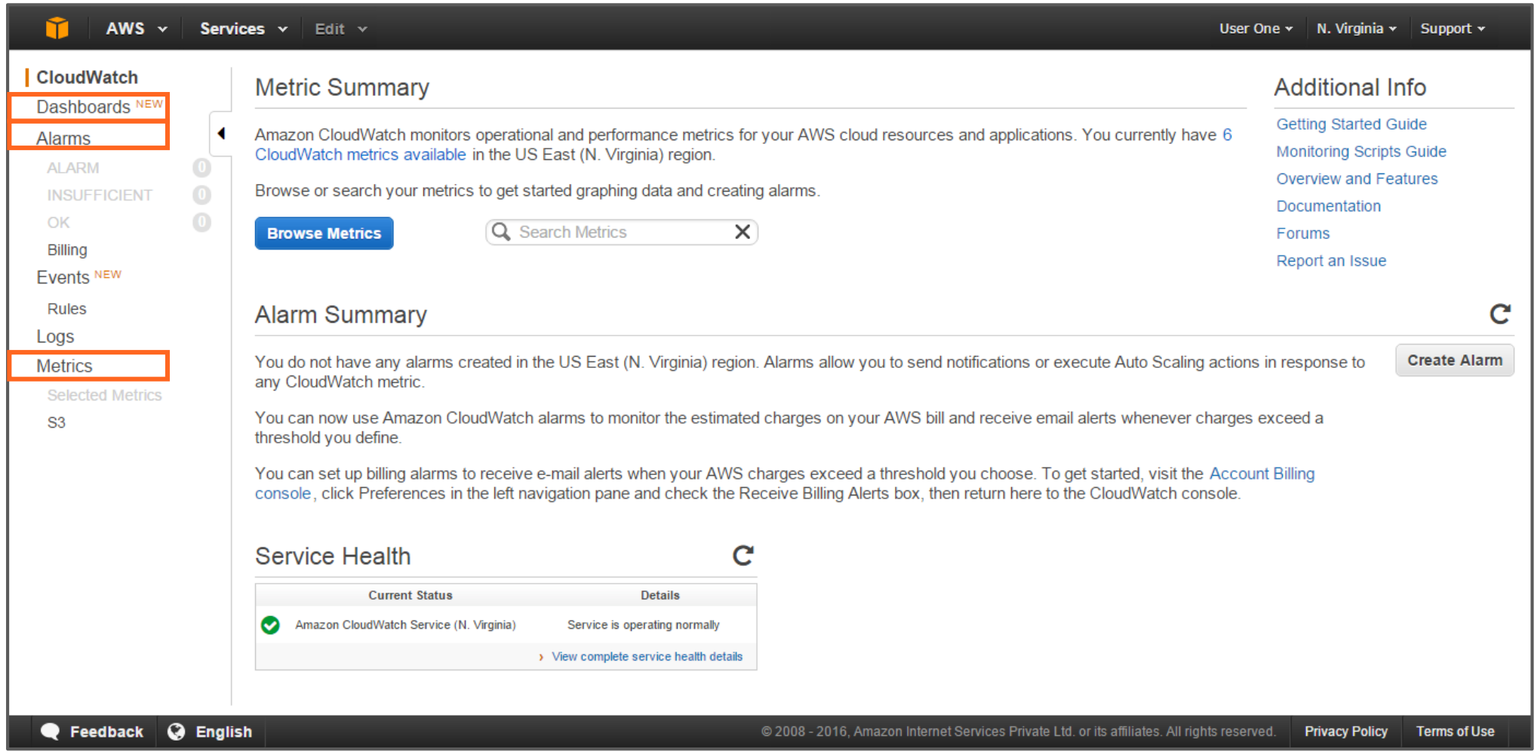
Using Amazon CloudWatch

Use Amazon CloudWatch under the AWS Free Tier.

The Free Tier includes:

- 3 dashboards with 50 metrics each.
- 10 alarms.
- 10 detailed monitoring metrics.
- 1 million API requests.

It includes all the basic metrics for EC2, EBS, ELB, and RDS instances.

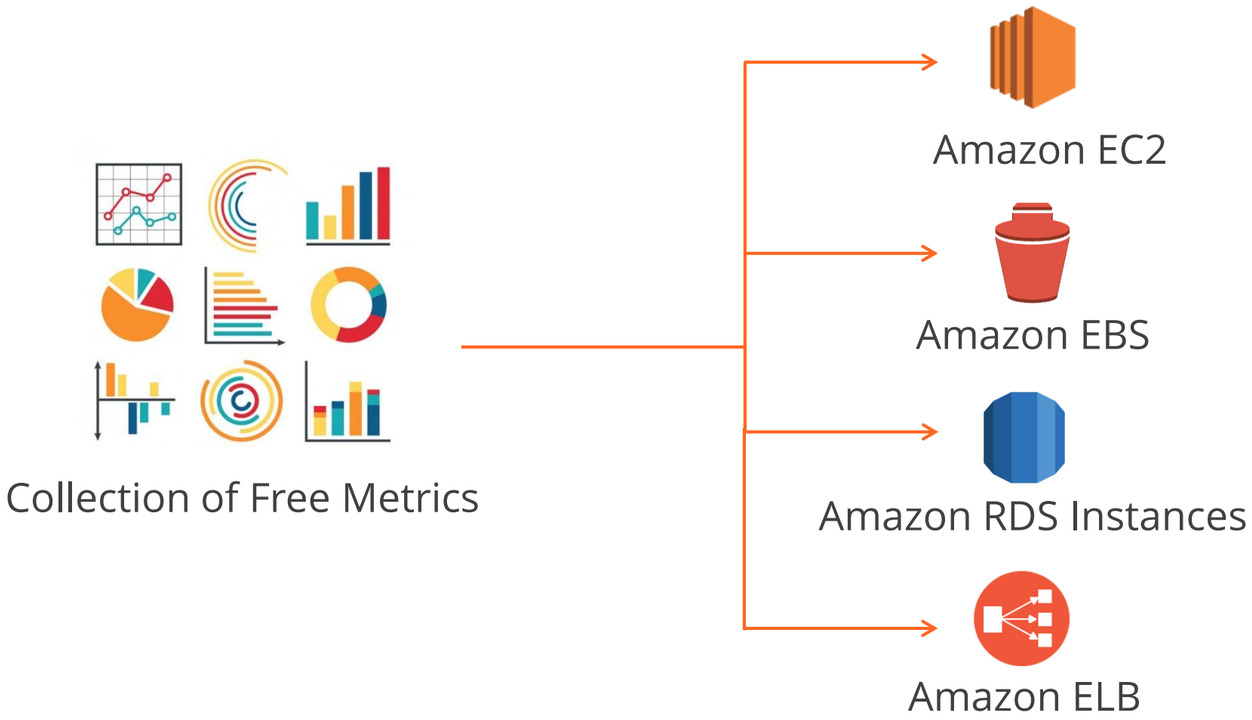


Benefits of Amazon CloudWatch

1. You can monitor the Amazon EC2 instance by viewing the basic metrics for disk and CPU usage, along with the data transfer rate at no extra cost.
2. You can keep an eye on other AWS resources at no extra cost or software.
3. You can monitor custom metrics generated by your own applications.
4. You can efficiently monitor and store the logs for troubleshooting your applications, and systems at any point in time.
5. You can receive notifications or take a quick, automated action by setting an alarm on any metrics.
6. You can view the statistics and graphs in the CloudWatch Dashboard. You can reuse few graphs of custom metrics to quickly monitor the operational status, and detect any issue.
7. You can easily detect and respond to resource changes due to CloudWatch Events.

Overview of Metrics

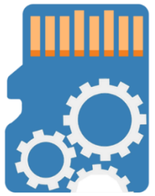
Metrics refer to indicators that signify the performance of employed AWS resources, and your applications in the cloud.



Metrics such as request count, latency, and CPU utilization exists for these resources.

Custom Metrics

Amazon CloudWatch monitors custom metrics generated by your applications which includes:



Memory Usage

Page Load Times

Error Rates

You can provide these metrics through a simple Application Program Interface, or API request.

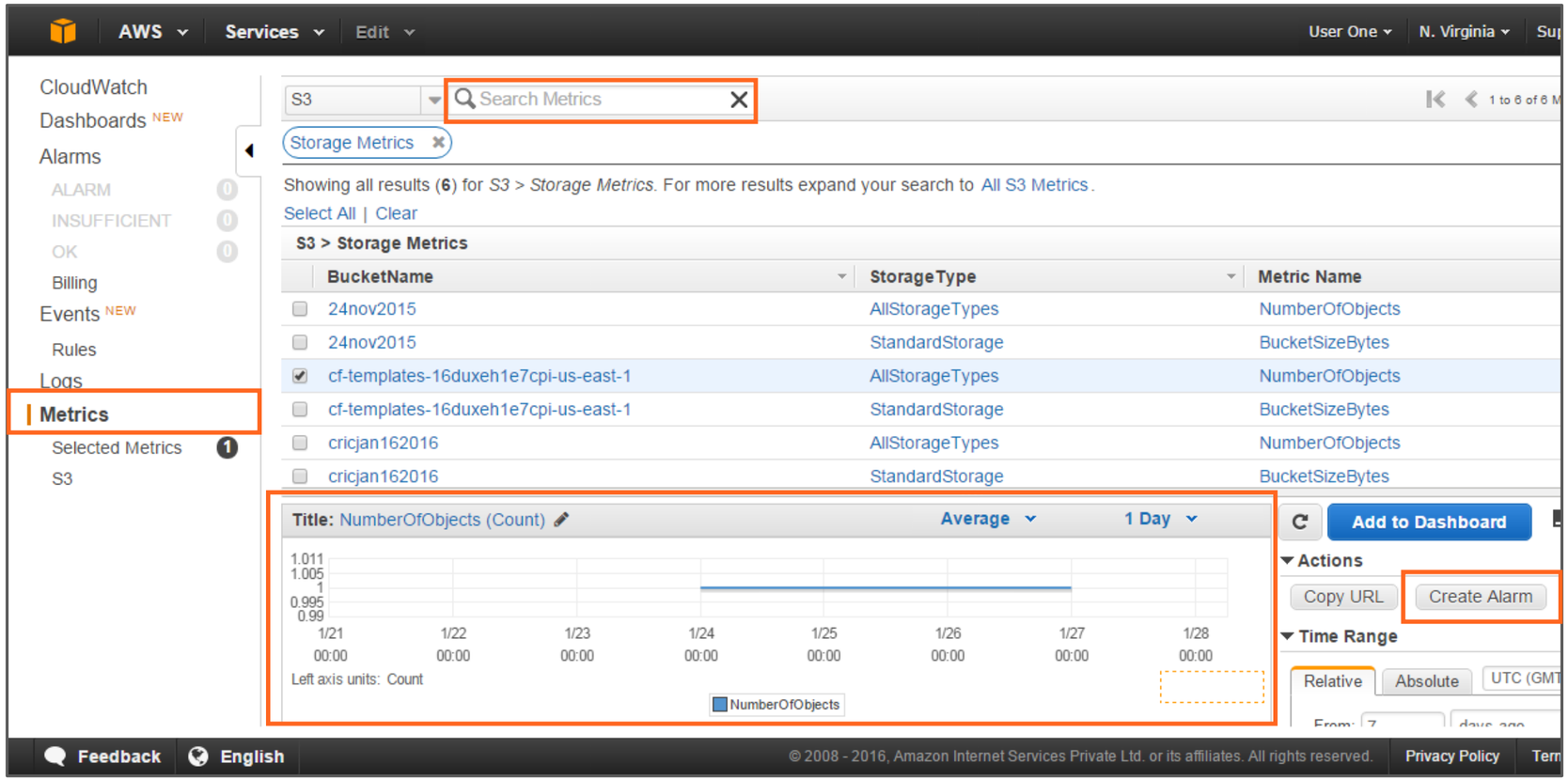
You can set your applications to send a specific page load time through an API. All CloudWatch functionalities, such as statistics and alarms, are accessible at up to one-minute frequency.

Working with Metrics

Amazon CloudWatch is capable of loading all metrics into your AWS account. It enables you to:

- Search them
- Create graphs
- Set alarms

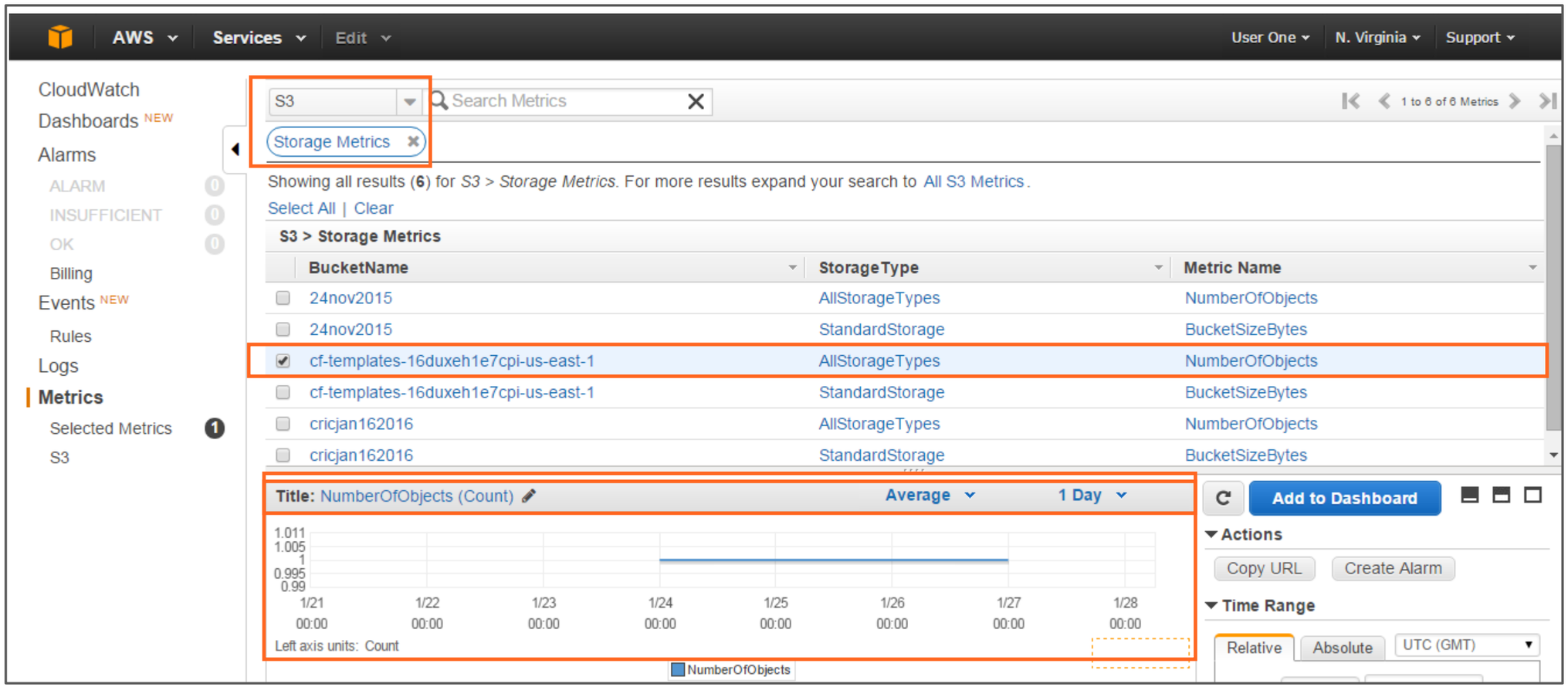
To view the metrics, click the **Metrics** menu on the left pane in the Amazon CloudWatch Console.



Working with Metrics

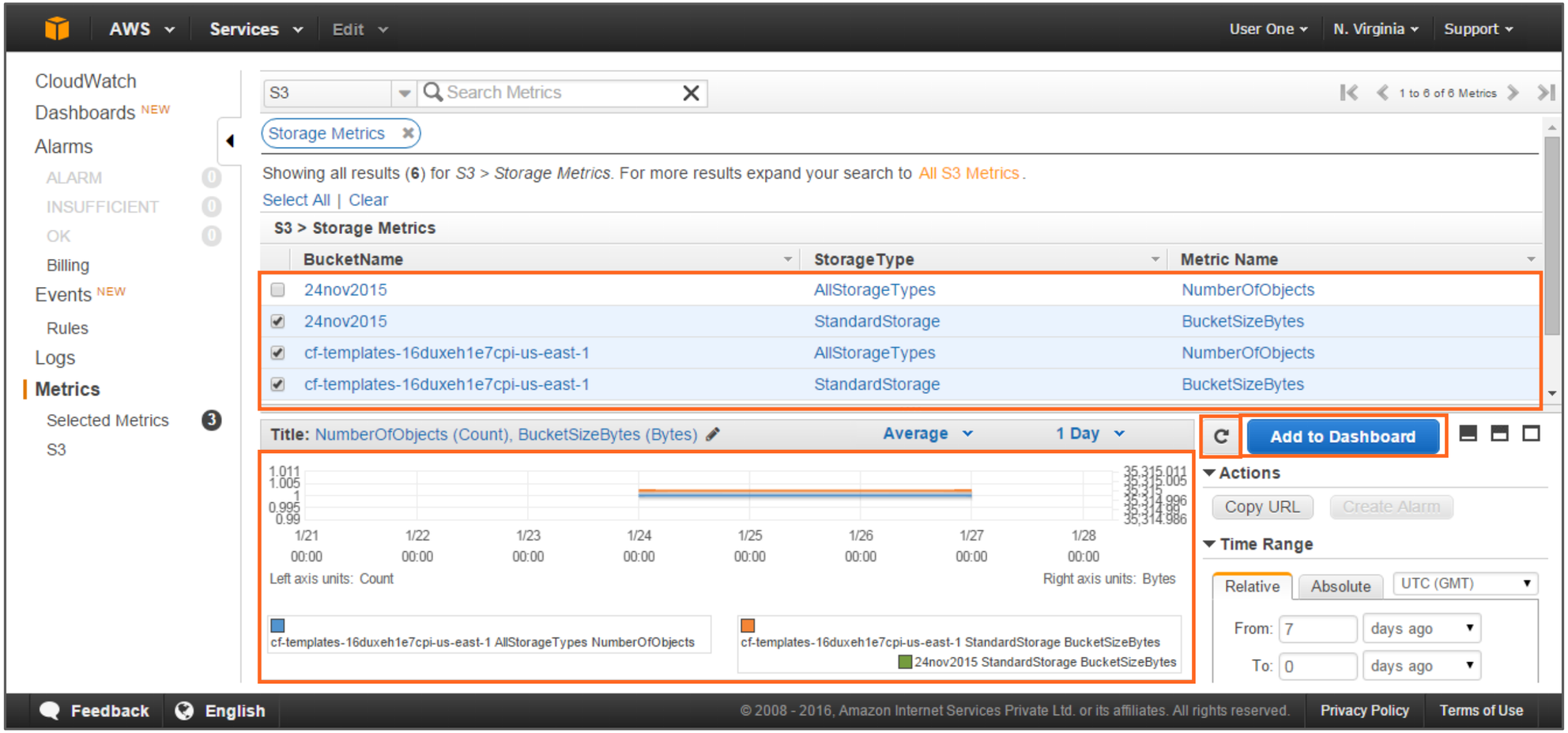
Select a metric of your choice to display an interactive graph in the bottom pane.

Also, you can change the Name, Statistic Value, and Period for the selected metrics.



Working with Metrics

You can view a graph showing correlated behavior or trend patterns for multiple metrics.



Selected multiple CloudWatch Metrics

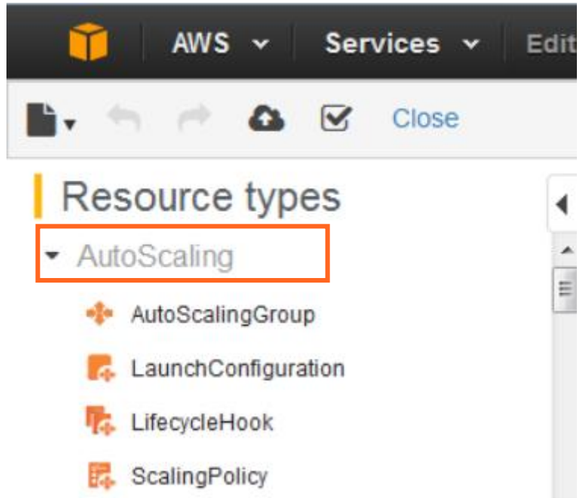
Demo 02—Obtaining the Statistics of Average CPU Utilization Metric

(Refer to the E-Learning course: Screen Number – 6.6)

Alarms

Overview of Alarms

Alarms are set in any metric to obtain notifications, or respond automatically if a particular metric goes beyond the mentioned threshold.

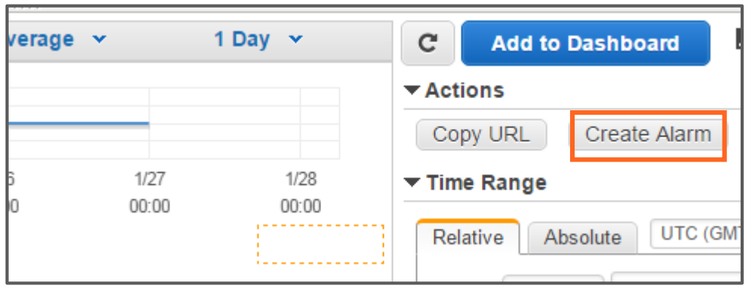


Example 1:

If an Amazon EC2 metric extends beyond the alarm threshold, you can dynamically remove the instances using the Auto Scaling service.

Example 2:

Set an alarm to shut down the underutilized or unused Amazon EC2 instances.



Working of Alarms

An alarm keeps a watch on a single metric for a specified period, and performs single or multiple actions.

1. It takes a specified action, when the gained metric value surpasses the specified threshold of the metric monitored.

2. A change in state makes the alarm invoke an action, but only for the lasting changes.

3. No action is taken until the changed state of a resource or application is maintained for the mentioned number of periods.

Working of Alarms

Taking an action involves passing a notification to the Auto Scaling policy or the Amazon Simple Notification Service, or SNS topic.

Amazon SNS notifications

No extra actions taken.

Auto Scaling policy

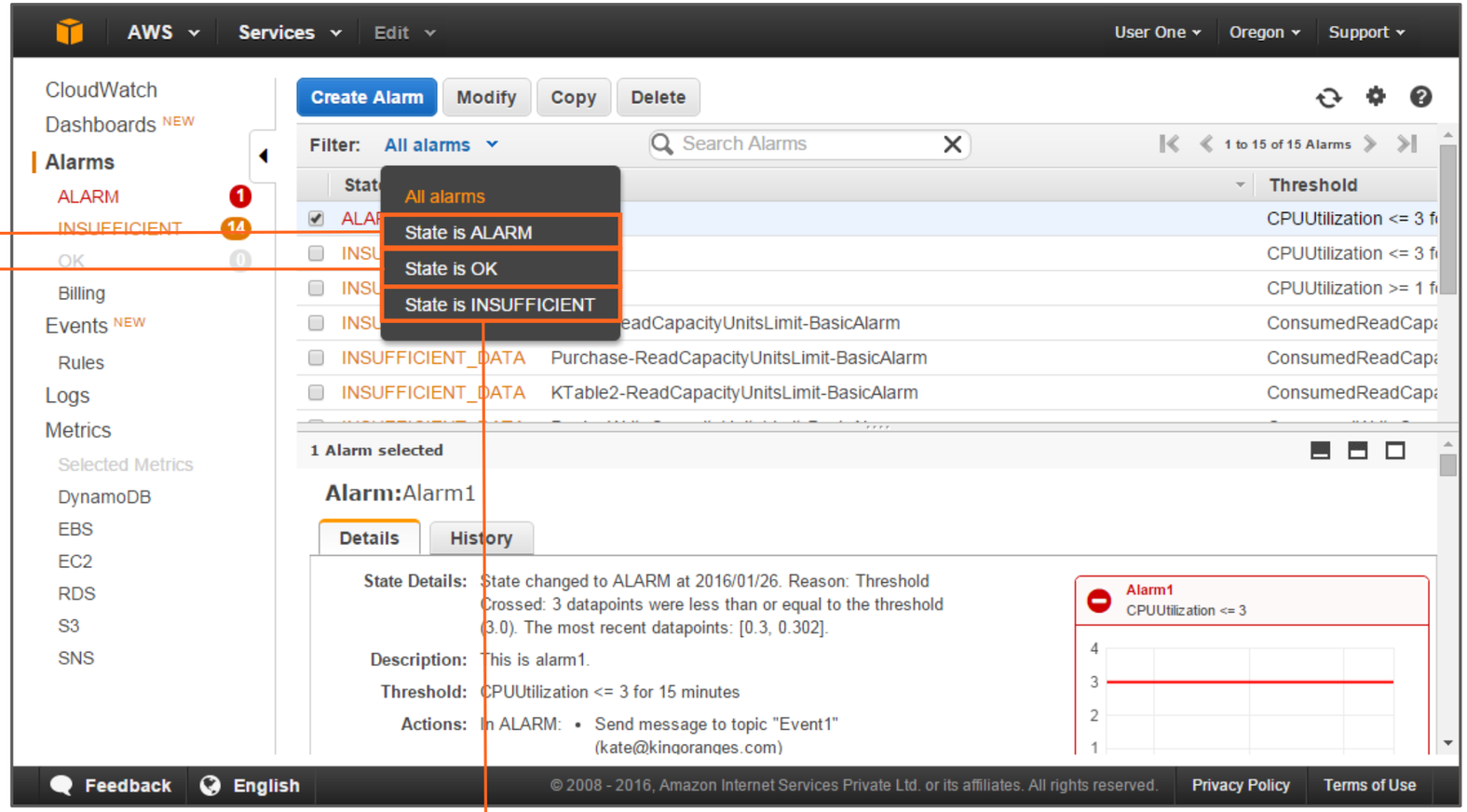
AWS CloudWatch invokes an action every time a new state is sustained.

It is your responsibility to ensure the actions are performed. Because, CloudWatch is not designed to monitor the actions, or its resulting errors.

Alarm States

Metric value has surpassed the specified threshold

The gained metric value is within the specified threshold

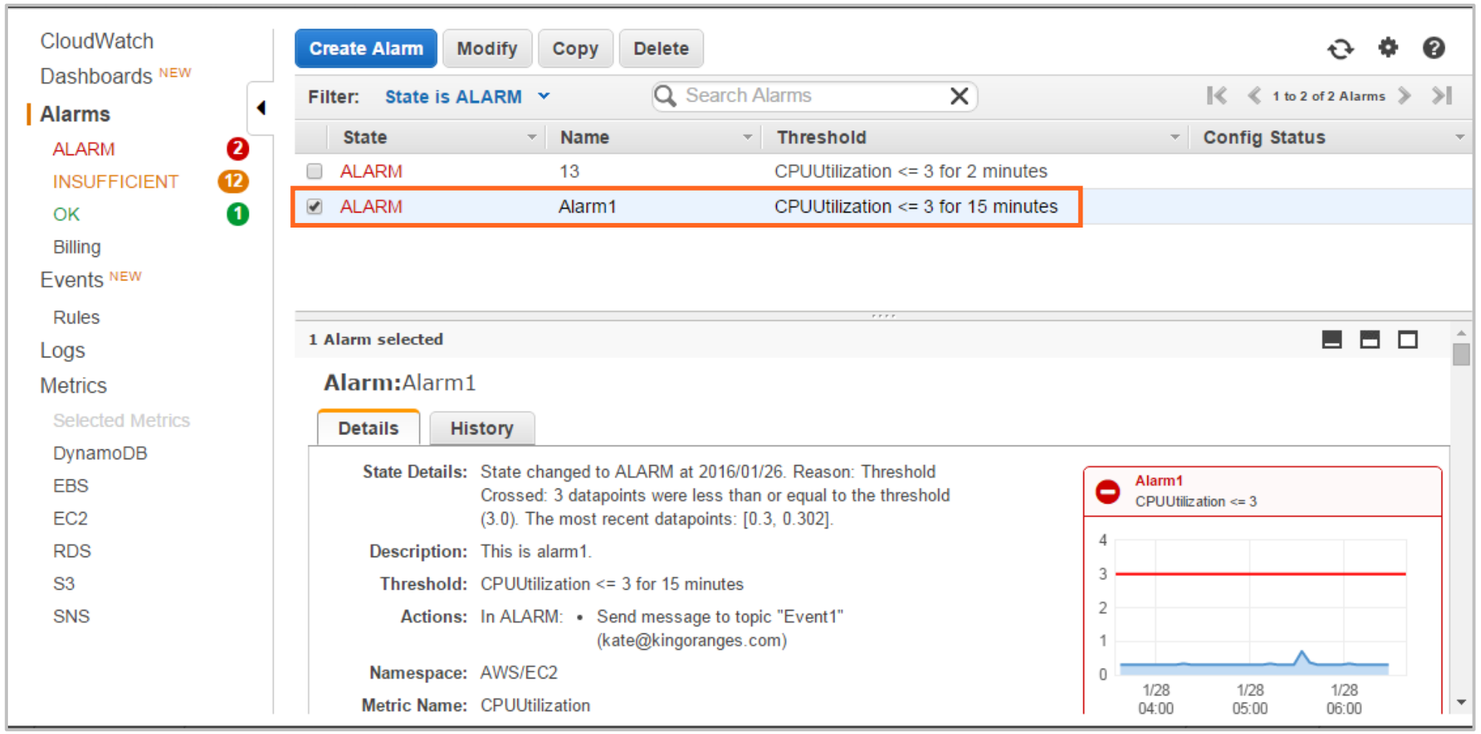


No enough data exists for the metric to figure out the alarm state

Example of ALARM State

Set an alarm for **CPUUtilization** of an EC2 instance. Here, if the threshold for **CPUUtilization** is less than or equal to 3 for 3 consecutive periods, the state of the alarm should be changed to **ALARM**. And the period is 5 minutes.

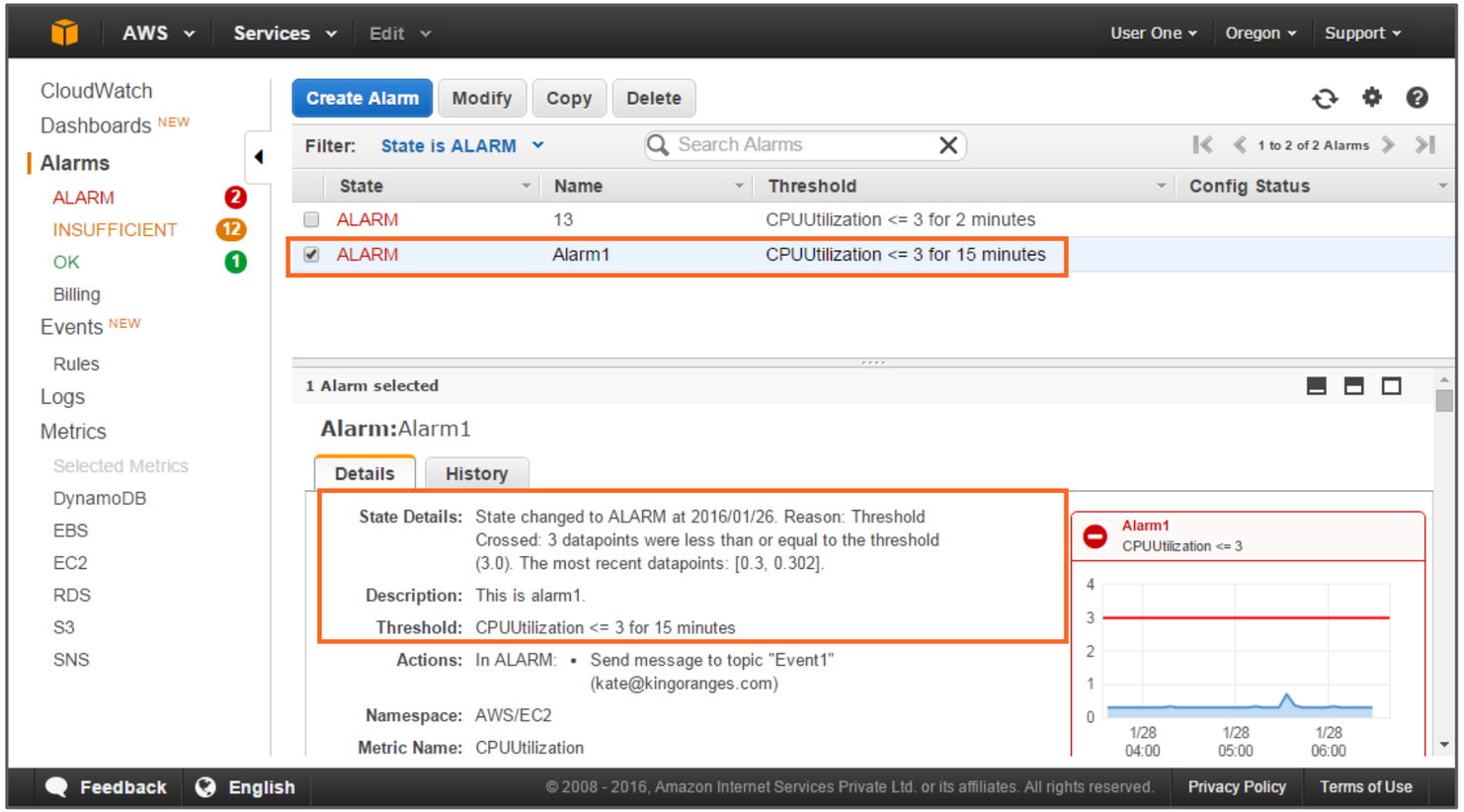
The **CloudWatch** service monitors the CPU utilization after every 5 minutes.



Example of ALARM State

Assuming the data point for the **CPUUtilization** value is monitored as 0.3 for 3 consecutive periods of 5 minutes.

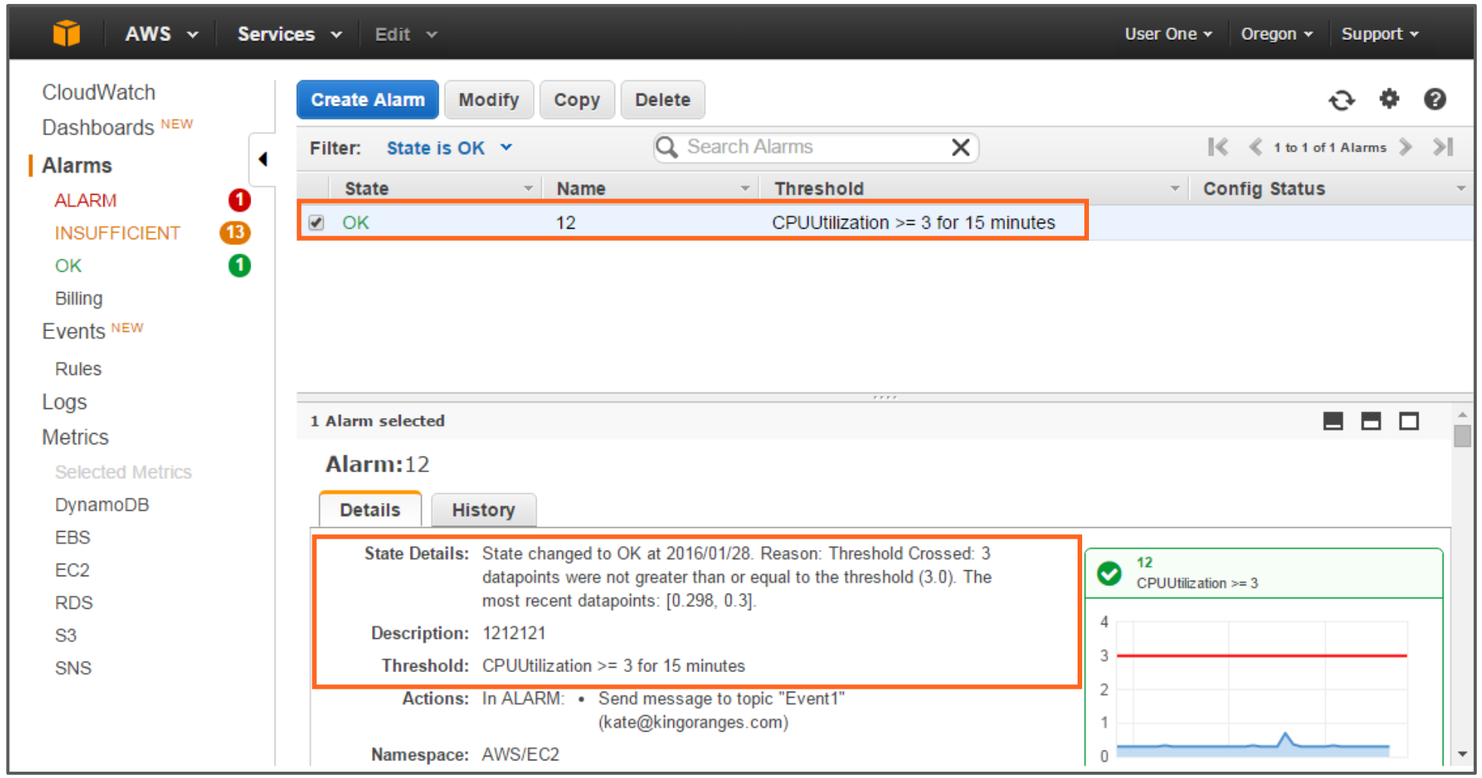
The value for the threshold, **CPUUtilization** <=3 becomes true. In this case, the alarm would change its state to **ALARM**.



Example of OK State

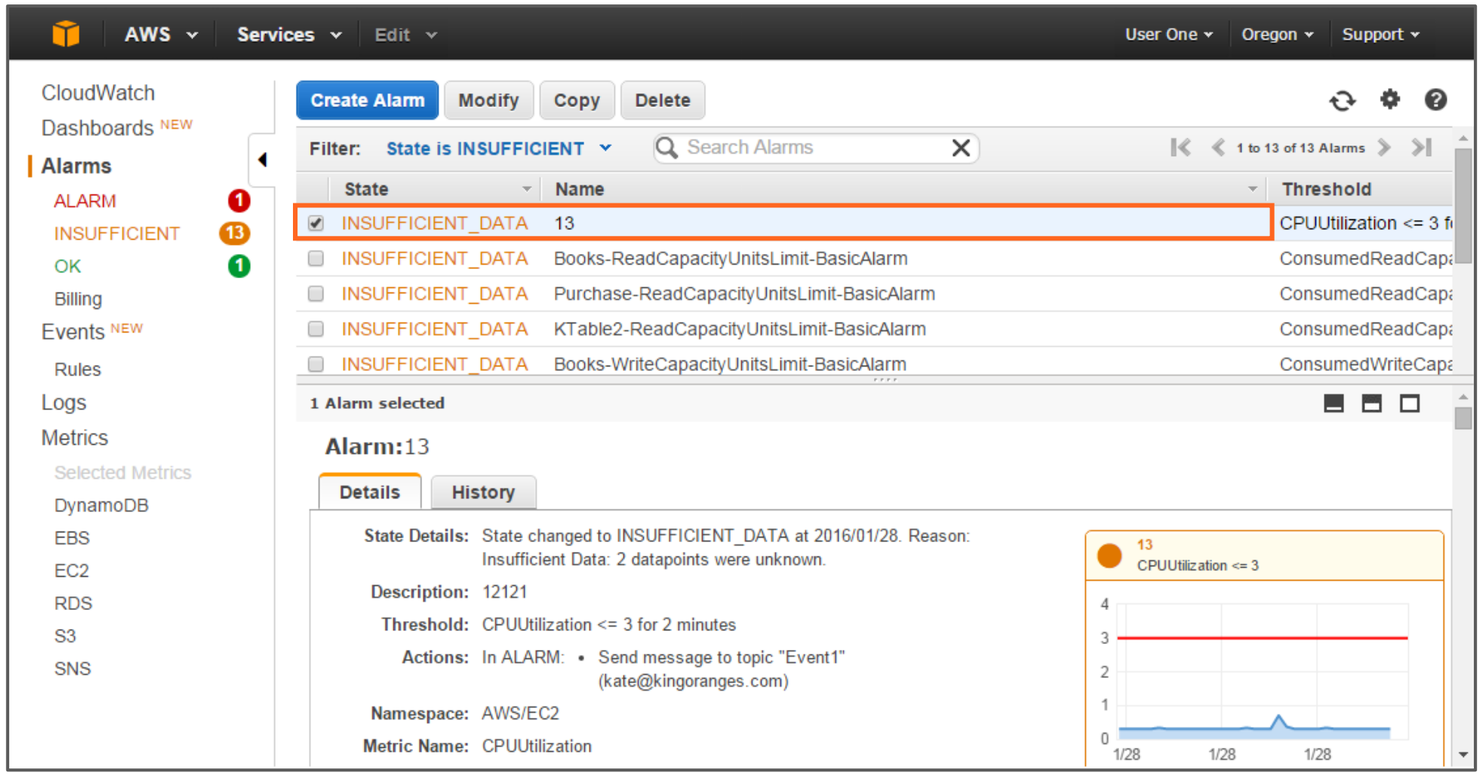
Assuming the data point for the **CPUUtilization** value is monitored as 0.3, and the required **CPUUtilization** condition is changed to greater than or equal to 3.

In this case, the value for the threshold, **CPUUtilization** >=3 becomes false. So, the alarm would change its state to **OK**.



Example of INSUFFICIENT_DATA State

An alarm is set to monitor the Amazon EBS volume, and the EBS may not pass the metric data for that volume. This means, no activity occurs for monitoring the volume. In such a situation, the alarm may change its state to **INSUFFICIENT_DATA**.



Demo 03—Creating an Alarm

(Refer to the E-Learning course: Screen Number – 6.8)

Demonstrate how to create an Alarm.

Knowledge Check

KNOWLEDGE CHECK 1	You can have up to _____ alarms per AWS account.

a. 3500

b. 4000

c. 5000

d. 5500

KNOWLEDGE CHECK	You can have up to _____ alarms per AWS account.

a. 3500

b. 4000

c. 5000

d. 5500

The correct answer is **c.**

Explanation: You can have up to 5000 alarms per AWS account.

KNOWLEDGE CHECK 2	Keeping an eye on other AWS resources at no extra cost or software is the benefit of _____.

a. AWS CloudFront

b. AWS CloudFormation

c. AWS Identity and Access Management

d. AWS CloudWatch

KNOWLEDGE CHECK	Keeping an eye on other AWS resources at no extra cost or software is the benefit of _____.

a. AWS CloudFront

b. AWS CloudFormation

c. AWS Identity and Access Management

d. AWS CloudWatch

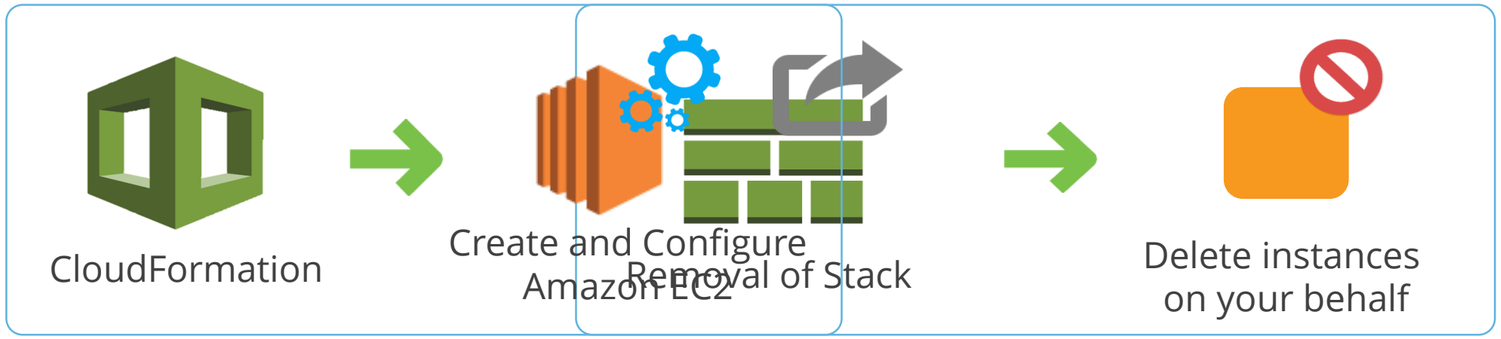
The correct answer is **d.**

Explanation: One of the significant benefits of using the Amazon CloudWatch service is that you can keep an eye on other AWS resources at no extra cost or software.

AWS Identity and Access Management (IAM)

Need for AWS Identity and Access Management (IAM)

AWS CloudFormation can only take those actions you are permitted to perform.



To manage such permissions, you use AWS Identity and Access Management, or IAM.

Nine Key Features of IAM

1. Allow other people to use your resources without providing password
2. Different permissions to different users, for different resources
3. Allow applications running on Amazon EC2 to securely access AWS resources
4. Enable two-factor authentication for individual users and your account
5. Users can temporarily access your account, using your account password
6. Obtain logs that have information about the users who are a part of the IAM identities
7. Supports storing, transmitting, and processing of credit card data
8. Integrate IAM with almost all other AWS services
9. Free to use and you pay only for other AWS services

Functionality of Identity and Access Management

IAM allows you to manage the following three entities:

1

IAM users and access

Manage users by giving individual credentials such as passwords and multi-factor authentication code, after creating them through IAM.

2

IAM roles and permissions

Manage user roles by controlling permissions for the operations that they can perform, after creating them.

3

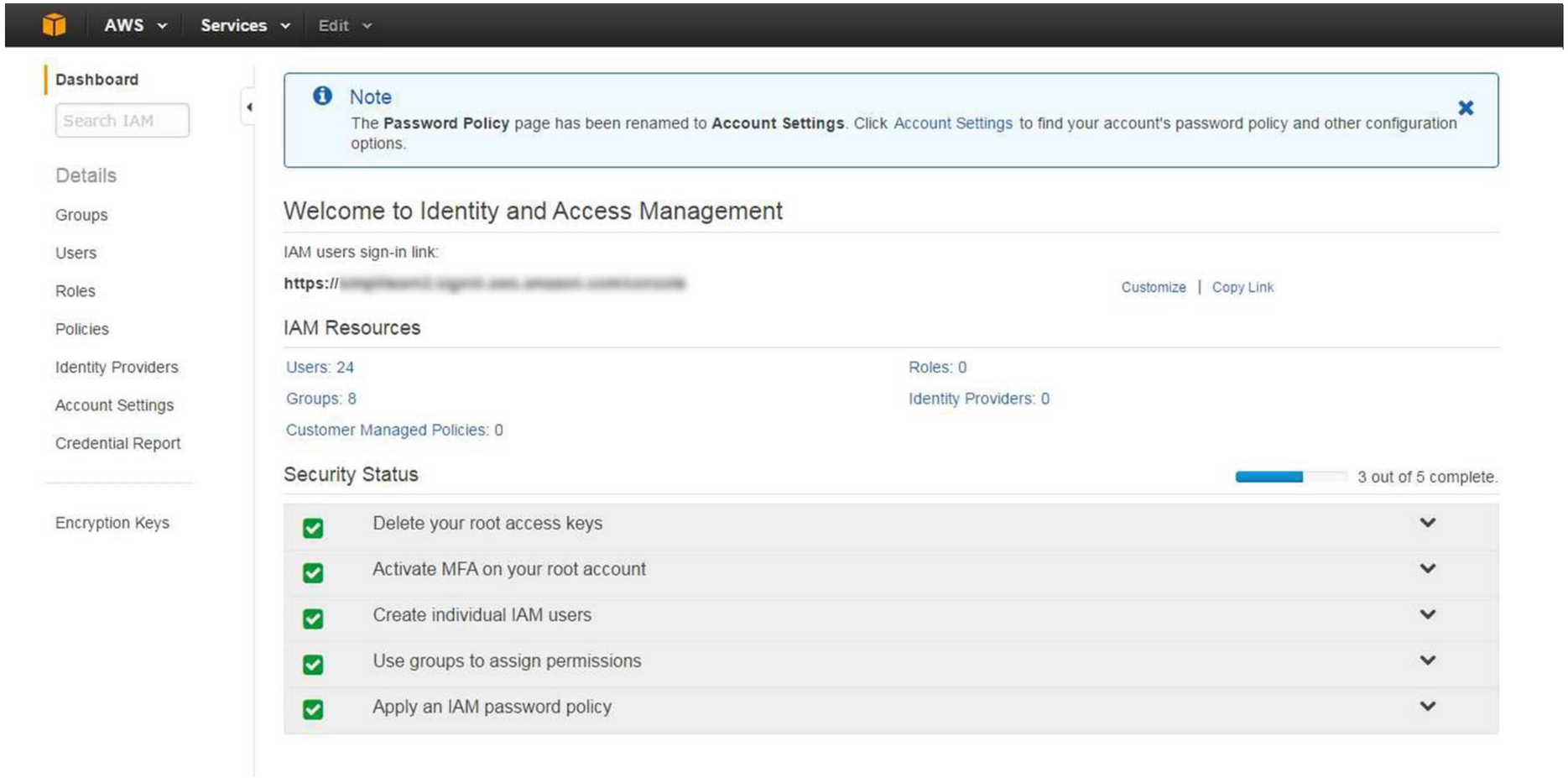
Federated users and permissions

Here use the feature of identity federation for enabling several identities without creating an IAM user for each identity.

Accessing AWS Identity and Access Management

To access IAM:

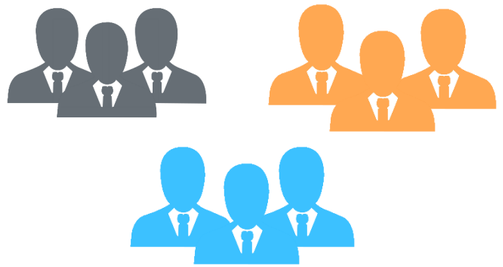
Click Identity and Access Management under the Security and Identity section of your AWS Management Console.



Identity and Access Management Users

Create single or several IAM users

Assign security credentials



One or more groups

Permissions for controlling

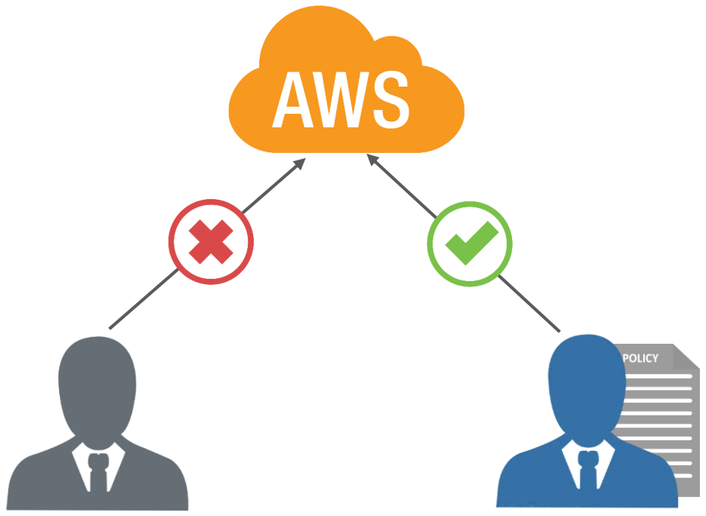
IAM users can be the end users who need to access cloud content, administrators who need to manage cloud resources, and systems that need programmatic cloud data access.

Need for an IAM User

Create an IAM user for the following reasons:

01	To prevent user from accessing your root account
02	To assign different authorization permissions or policies to some users accessing particular services and their resources
03	To use the AWS Command Line Interface
04	To use a role
05	To allow federated users or external identities to access resources securely

Policies and Users



Without policy, no access to AWS account

Access or usage permissions by creating a policy

A policy refers to a document containing actions to be performed, and the resources on which those actions can be performed.

If a policy does not mention accessing the books table through an Amazon DynamoDB action, the user cannot access the table through any Amazon DynamoDB action.

User Groups

A group is a set of IAM users with common permissions assigned through a group policy.

- You can create a group called Warehouse Administrators, and provide suitable permissions to that group. Any group user automatically has the permissions you assign to the group.
- To grant administrator privileges to a user, add the user to the Warehouse Administrators group.
- If a user changes jobs in your company, move the user to the new group, instead of changing the permissions.

Demo 04—Creating the First IAM User
(Refer to the E-Learning course: Screen Number – 6.11)

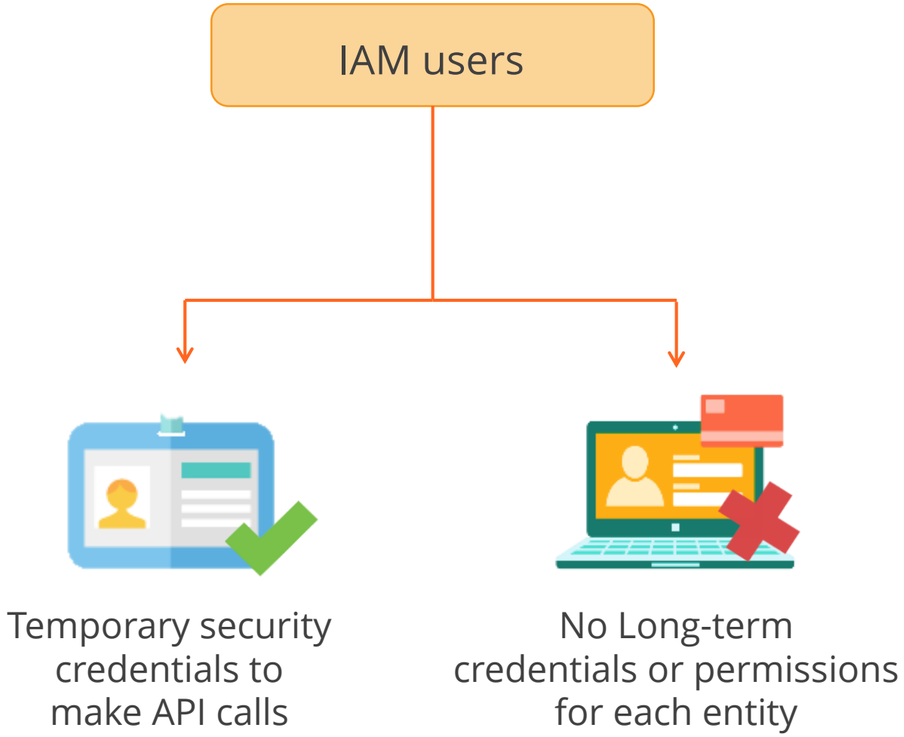
Demo 05—Creating an Administrator Group
(Refer to the E-Learning course: Screen Number – 6.12)

Demo 06—Adding the user to the Administrator Group (Refer to the E-Learning course: Screen Number – 6.13)

IAM Roles

Roles

IAM roles are similar to users, but are AWS identities associated with permission policies that determine what the identity can perform. A role has no credentials associated with it. When you assign a role to a user, access keys are generated automatically and sent to the user.



Scenarios for Using Roles

IAM roles enable you to delegate or hand over access to applications, services, or users that do not have access to AWS resources.

- You want to give users of an AWS account access to resources in the other account
- You want a mobile app to use cloud resources, without embedding credentials in the application where users can extract them
- You want to give access to users having identities outside AWS, such as in your business directory
- You want to give the external parties access to your AWS account for auditing your resources

Demo 07—Creating a Role

Quiz

QUIZ 1	Choose the option that describes CloudFormation.

a. Keeps a watch on AWS resources

b. Creates instances for AWS resources

c. Creates and designs AWS resources

d. Creates and provisions AWS resources

QUIZ 1	Choose the option that describes CloudFormation.

a. Keeps a watch on AWS resources

b. Creates instances for AWS resources

c. Creates and designs AWS resources

d. Creates and provisions AWS resources

The correct answer is **d.**

Explanation: AWS CloudFormation service creates and provisions AWS resources.

QUIZ 2	A stack is handled as a _____.

a. instance

b. device

c. storage device

d. single unit

QUIZ 2	A stack is handled as a _____.

a. instance

b. device

c. storage device

d. single unit

The correct answer is **d.**

Explanation: A stack is handled as a single unit. So, the AWS CloudFormation service rolls back the stack, or deletes any created resources if even one resource fails to successfully create itself.

QUIZ 3	What does Amazon CloudWatch include while it monitors the custom metrics that applications generate?

a. Memory usage, page load times, and error rates.

b. Percentage of resource utilization.

c. Performance of resources through graphs.

d. Only error rate after troubleshooting applications.

QUIZ 3	What does Amazon CloudWatch include while it monitors the custom metrics that applications generate?

a. Memory usage, page load times, and error rates.

b. Percentage of resource utilization.

c. Performance of resources through graphs.

d. Only error rate after troubleshooting applications.

The correct answer is **a.**

Explanation: Amazon CloudWatch monitors the custom metrics your applications generate, and this includes memory usage, page load times, and error rates.

QUIZ 4	What does IAM mean?

a. Service that allows controlling access

b. Service that allows access to CloudFormation

c. Service that allows access to CloudWatch

d. Service that allows access to EC2

QUIZ 4	What does IAM mean?

a. Service that allows controlling access

b. Service that allows access to CloudFormation

c. Service that allows access to CloudWatch

d. Service that allows access to EC2

The correct answer is **a.**

Explanation: IAM is a free Web service that allows controlling access to AWS resources as well as services securely for the users.

QUIZ 5	What is a policy in IAM?

a. Document containing information about the user

b. Document containing actions to be performed

c. Document containing rules to be followed

d. Document containing privacy information

QUIZ 5	What is a policy in IAM?

a. Document containing information about the user

b. Document containing actions to be performed

c. Document containing rules to be followed

d. Document containing privacy information

The correct answer is **b.**

Explanation: A policy refers to the document containing actions to be performed, and the resources on which those actions are performed.

Quiz 6	Any resource or action NOT mentioned in the policy is prohibited.

a. False

b. True

QUIZ 6

Any resource or action NOT mentioned in the policy is prohibited.

a. False

b. True

The correct answer is **b.**

Explanation: Any resource or action not mentioned in the policy is prohibited by default.

Quiz 7

What is a set of IAM users with common permissions assigned through a group policy?

a. Stack

b. Group

c. Units

d. Metrics

QUIZ 7	What is a set of IAM users with common permissions assigned through a group policy?

a. Stack

b. Group

c. Units

d. Metrics

The correct answer is **b.**

Explanation: A group is a set of IAM users with common permissions assigned through a group policy.

Key Takeaways

Key Takeaways

AWS CloudFormation creates, configures, manages, and updates AWS resources described in a JSON format template.

template1

```
1  {
2    "AWSTemplateFormatVersion": "2010-09-09",
3    "Description": "AWS CloudFormation Sample Template VPC_Single_Instance_In_Subnet.",
4    "Parameters": {
5      "InstanceType": {
6        "Description": "WebServer EC2 instance type",
7        "Type": "String",
8        "Default": "t2.micro",
9        "AllowedValues": [
10         "t1.micro",
11       ],
12       "ConstraintDescription": "must be a valid EC2 instance type."
13     },
14     "KeyName": {
15       "Description": "Name of an existing EC2 KeyPair to enable SSH access to the instance.",
16       "Type": "AWS::EC2::KeyPair::KeyName",
17       "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
18     },
19
20   "Mappings": {
21     "AWSInstanceType2Arch": {
22       "t1.micro": {
23         "Arch": "PV64"
24       },
25
26     },
27   "Resources": {
28     "VPC": {
29       "Type": "AWS::EC2::VPC",
30       "Properties": {
31         "EnableDnsSupport": "true",
32         "EnableDnsHostnames": "true",
33         "CidrBlock": "10.0.0.0/16"
34       },
35     },
36
37     "InternetGateway": {
38       "Type": "AWS::EC2::InternetGateway",
39       "Metadata": {
40         "AWS::CloudFormation::Designer": {
41           "id": "a166c4f5-7cc4-429b-b9d8-2c8c43facc63"
42         }
43       }
44     },
```

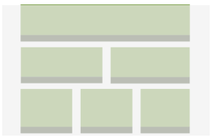
Key Takeaways

A Stack is a set of resources, managed as a single unit.

If

Any one resource fails to create itself

Then



Stack is rolled back or deleted

If

All the resources are retained

Then

Stack is not deleted

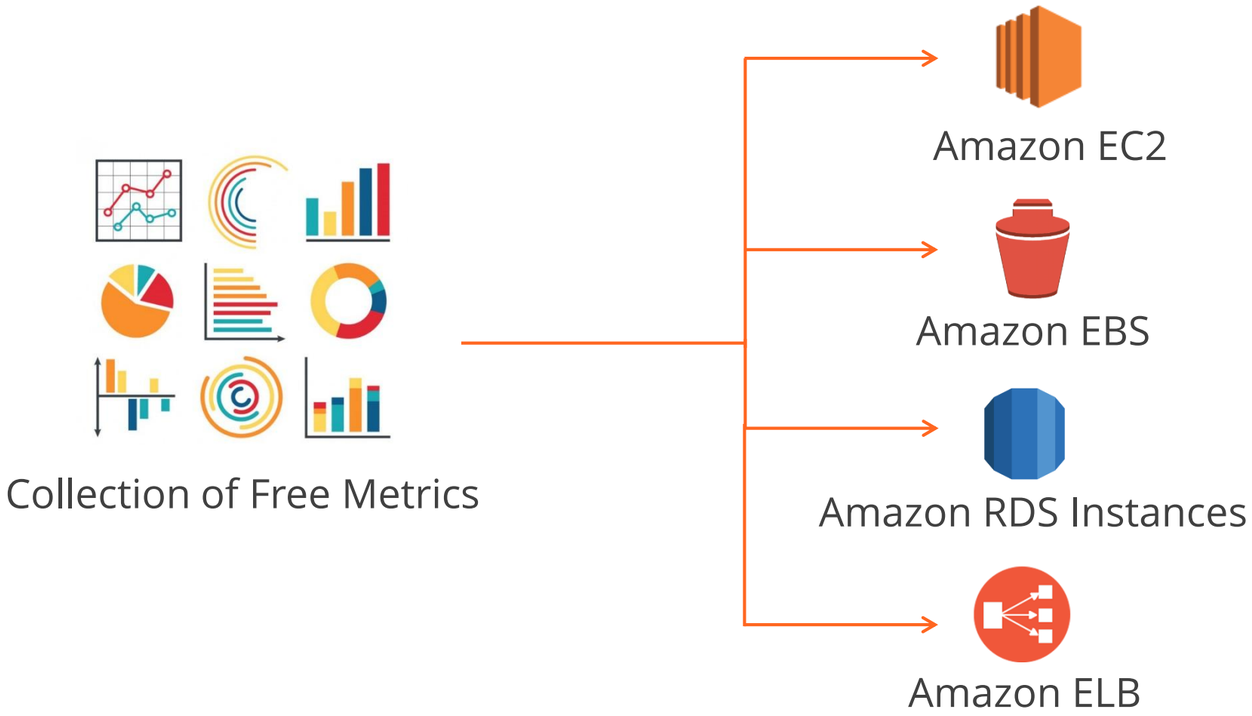
Key Takeaways

Amazon CloudWatch allows real-time monitoring of your cloud resources and applications, by tracking the log files and metrics.

- Gather and track log files and metrics
- Set alarms
- View the latest statistics and graphs
- Respond to changes in the resources

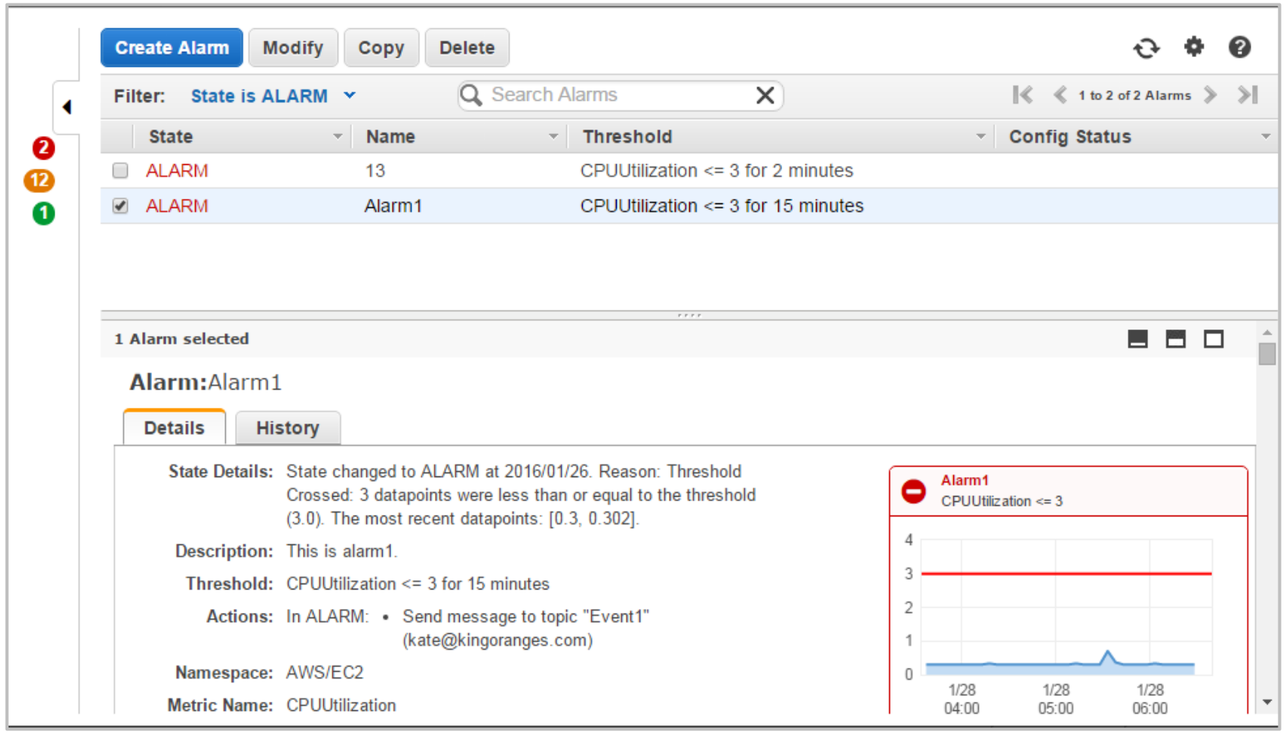
Key Takeaways

Metrics are indicators that signify the performance of your resources and applications in the cloud.



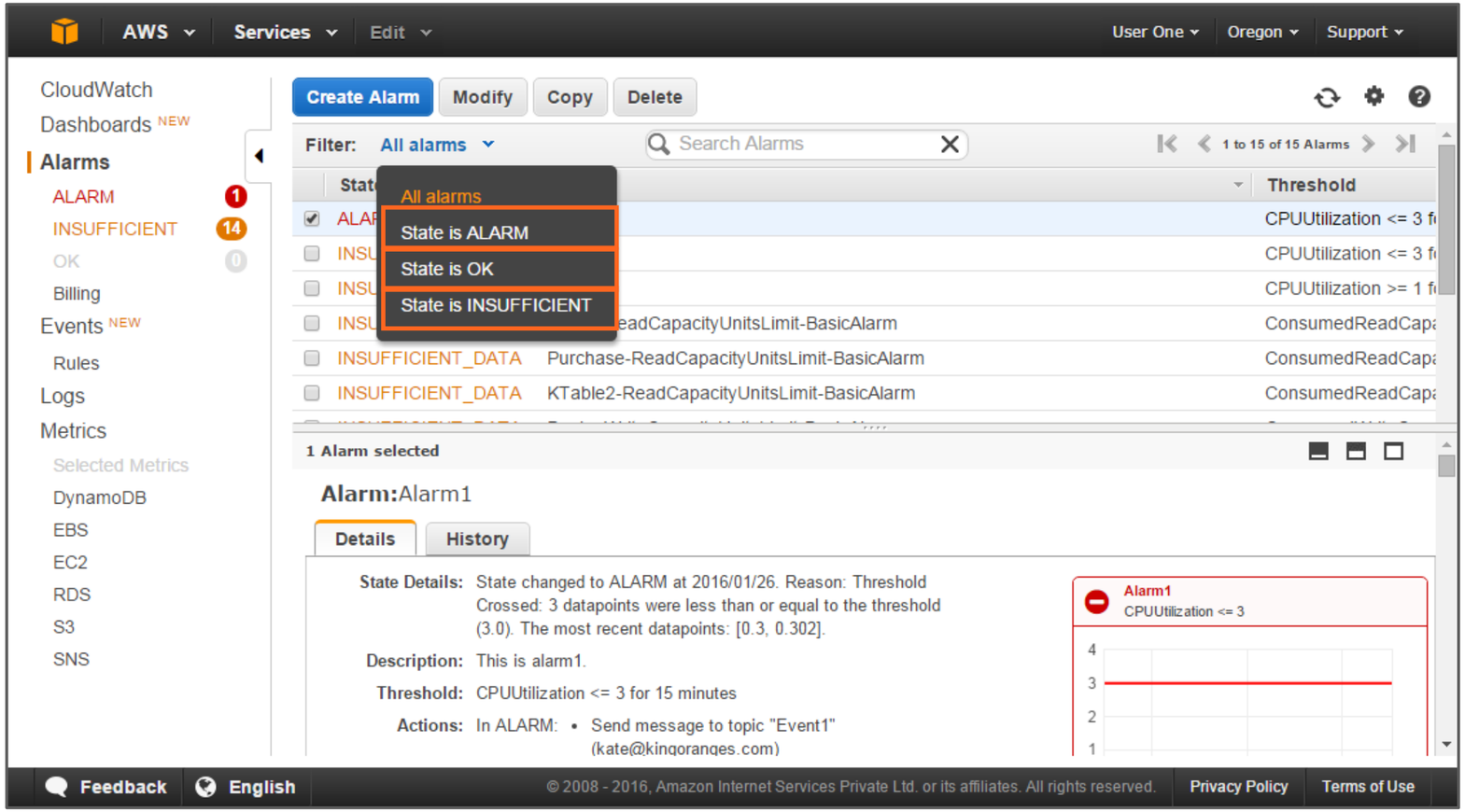
Key Takeaways

You can set an alarm on a metric for obtaining notifications and responding automatically if the metric value surpasses the stated threshold.



Key Takeaways

The state of an alarm can be OK, ALARM, or INSUFFICIENT_DATA.



Key Takeaways

IAM allows controlling securing access to AWS resources as well as services, for users, roles, and federated users.

1

IAM users and access

Manage users by giving individual credentials such as passwords and multi-factor authentication code, after creating them through IAM.

2

IAM roles and permissions

Manage user roles by controlling permissions for the operations that they can perform, after creating them

3

Federated users and permissions

Here use the feature of identity federation for enabling several identities without creating an IAM user for each identity.

Key Takeaways

- AWS CloudFormation creates, configures, manages, and updates AWS resources described in a JSON format template.
- A Stack is a set of resources, managed as a single unit.
- Amazon CloudWatch allows real-time monitoring of your cloud resources and applications, by tracking the log files and metrics.
- Metrics are indicators that signify the performance of your resources and applications in the cloud.

Key Takeaways (Contd.)

- You can set an alarm on a metric for obtaining notifications and responding automatically if the metric value surpasses the stated threshold.
- The state of an alarm can be OK, ALARM, or INSUFFICIENT_DATA.
- IAM allows controlling securing access to AWS resources as well as services, for users, roles, and federated users.

This Concludes 'Deployment and Management.'

THANK YOU