AWS® Certified Cloud Practitioner

STUDY GUIDE

FOUNDATIONAL (CLF-C01) EXAM

Includes interactive online learning environment and study tools:

Two custom practice exams

100 electronic flashcards

Searchable key term glossary

BEN PIPER
DAVID CLINTON

SYBEX®
A Wiley Brand

Compliance If any of your resources must adhere to specific compliance requirements such as the Health Insurance Portability and Accountability Act (HIPAA) or the Payment Card Industry Data Security Standard (PCI DSS), you can tag those resources accordingly.

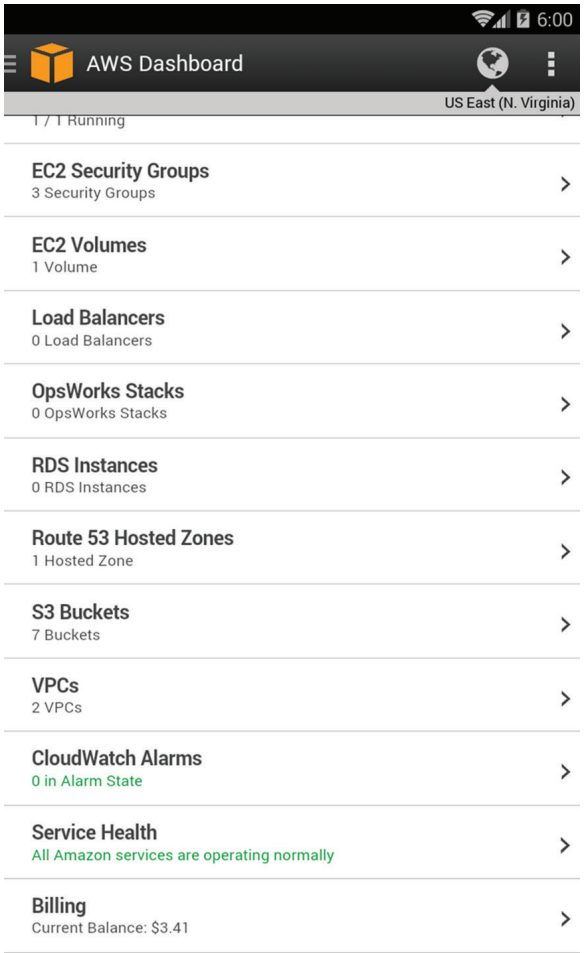
The AWS Console Mobile Application

The *AWS Console Mobile Application* is a smartphone application that lets you manage your AWS account and resources on the go. The application features a dashboard showing key information about your AWS account and resources, including the following:

- Service Health—View any current health issues with AWS services across different regions.
- CloudWatch Alarms—View alarm graphs and current alarm status.
- Billing—View your current billing balance and a graph of usage charges.

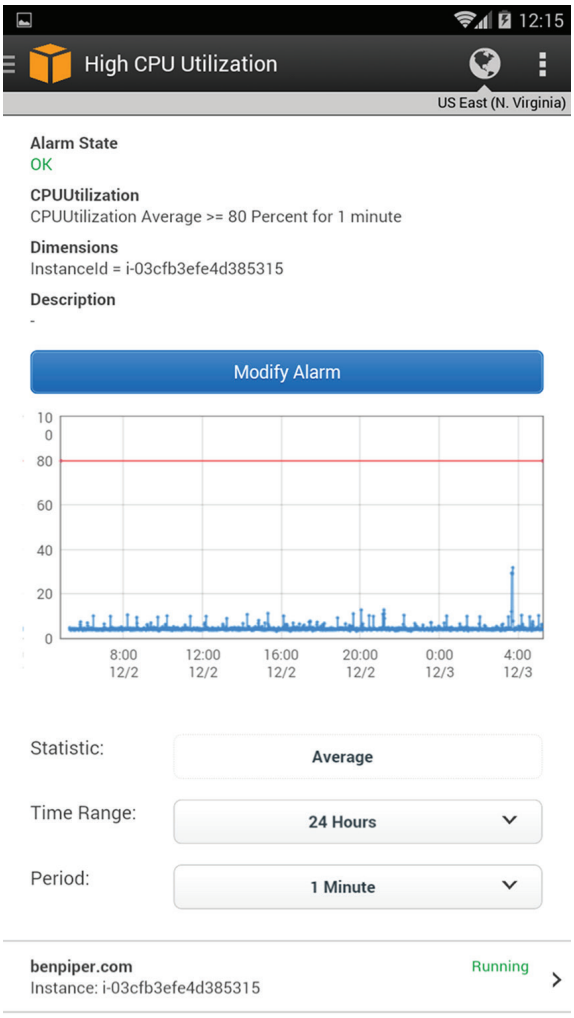
Take a look at Figure 6.11 for an example of what kind of information the dashboard can show you.

FIGURE 6.11 The AWS Console Mobile Application dashboard



You can use the application to make limited changes to some AWS resources, including CloudWatch Alarms, EC2 security groups, EC2 instances, and CloudFormation stacks. For example, you can view or modify a CloudWatch alarm, as shown in Figure 6.12.

FIGURE 6.12 Viewing a CloudWatch alarm from the AWS Console Mobile Application



Or you can stop or reboot an EC2 instance, as shown in Figure 6.13.

CloudWatch

Amazon CloudWatch is a key service that helps you plan, monitor, and fine-tune your AWS infrastructure and applications. It lets you collect, search, and visualize data from your applications and AWS resources in the form of logs, metrics, and events. Common CloudWatch use cases include the following:

Infrastructure monitoring and troubleshooting Visualize performance metrics to discover trends over time and spot outliers that might indicate a problem. Correlate metrics and logs across your application and infrastructure stacks to understand the root cause of failures and performance issues.

Resource optimization Save money and help with resource planning by identifying over-used or underused resources. Ensure performance and availability by using AWS Auto Scaling to automatically provision new EC2 instances to meet demand.

Application monitoring Create CloudWatch alarms to alert you and take corrective action when a resource's utilization, performance, or health falls outside of a threshold that you define.

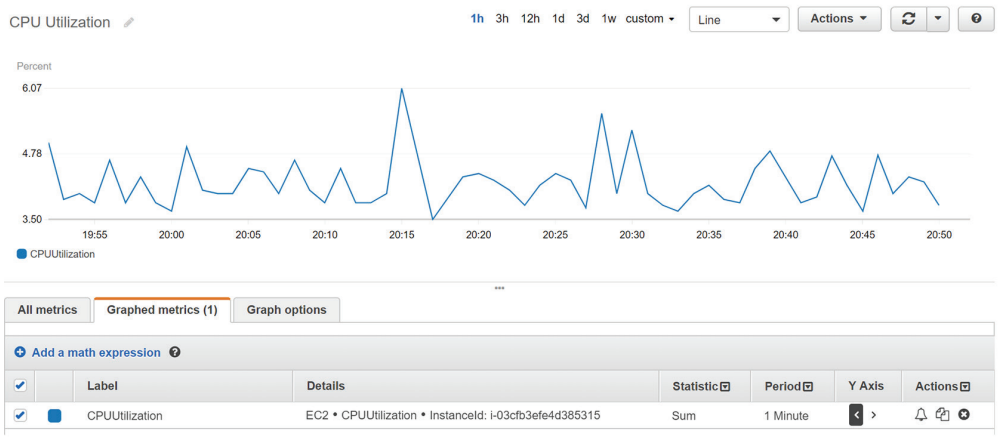
Log analytics Search, visualize, and correlate logs from multiple sources to help with troubleshooting and identify areas for improvement.

CloudWatch Metrics

CloudWatch Metrics is a feature that collects numeric performance metrics from both AWS and non-AWS resources such as on-premises servers. A *metric* is a variable that contains a time-ordered set of data points. Each data point contains a timestamp, a value, and optionally a unit of measure. For example, a data point for the CPU Utilization metric for an EC2 instance may contain a timestamp of December 25, 2018 13:37, a value of 75, and Percent as the unit of measure.

All AWS resources automatically send their metrics to CloudWatch. These metrics include things such as EC2 instance CPU utilization, S3 bucket sizes, and DynamoDB consumed read and write capacity units. CloudWatch stores metrics for up to 15 months. You can graph metrics to view trends and how they change over time, as illustrated in Figure 6.16.

FIGURE 6.16 Using CloudWatch to graph the CPU Utilization metric for an EC2 Instance



CloudWatch Alarms

A CloudWatch alarm watches over the value of a single metric. If the metric crosses a threshold that you specify (and stays there), the alarm will take an action. For example, you might configure an alarm to take an action when the average CPU utilization for an instance exceeds 80% for five minutes. The action can be one of the following:

Notification using Simple Notification Service The *Simple Notification Service* (SNS) allows applications, users, and devices to send and receive notifications from AWS. SNS uses a publisher-subscriber model, wherein a publisher such as an AWS service generates a notification and a subscriber such as an end user receives it. The communication channel that SNS uses to map publishers and subscribers is called a *topic*. SNS can send notifications to subscribers via a variety of protocols including the following:

- HTTP(S)
- Simple Queue Service (SQS)
- Lambda
- Mobile push notification
- Email
- Email-JSON
- Short Message Service (SMS) text messages

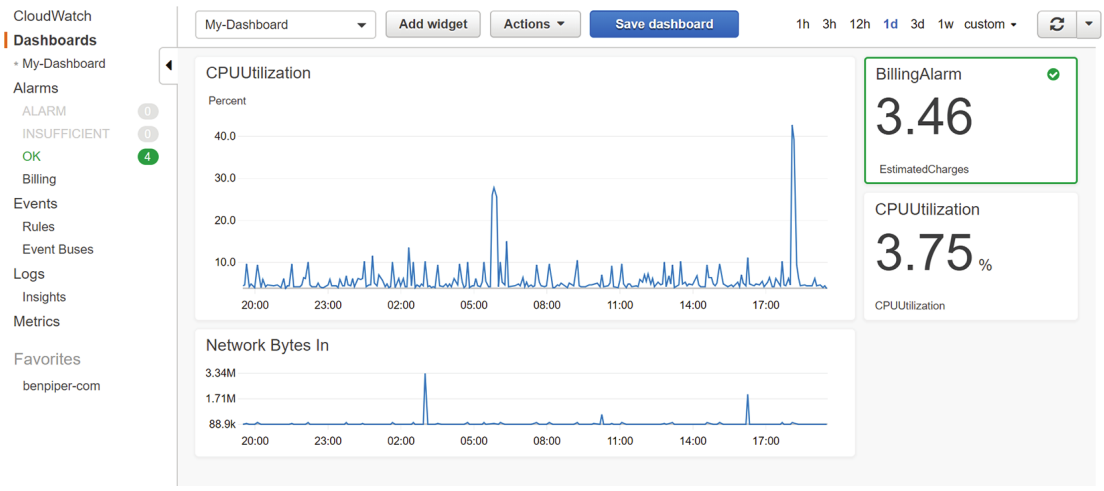
Auto Scaling action By specifying an EC2 Auto Scaling action, the EC2 Auto Scaling service can add or remove EC2 instances in response to changing demand. For example, if a metric indicates that instances are overburdened, you can have EC2 Auto Scaling respond by adding more instances.

EC2 action If you're monitoring a specific instance that's having a problem, you can use an EC2 action to stop, terminate, or recover the instance. Recovering an instance migrates the instance to a new EC2 host, something you may need to do if there's a physical hardware problem on the hardware hosting the instance.

CloudWatch Dashboards

CloudWatch dashboards are your one-stop shop for keeping an eye on all of your important metrics. You can create multiple dashboards and add to them metric graphs, the latest values for a metric, and CloudWatch alarms. You can save your dashboards for future use and share them with others. Dashboards can also visualize metrics from multiple AWS Regions, so you can keep an eye on the global health of your infrastructure. Check out Figure 6.17 for a sample CloudWatch dashboard.

FIGURE 6.17 A CloudWatch dashboard



CloudWatch Logs

CloudWatch Logs collects and stores log files from AWS and non-AWS sources and makes it easy to view, search, and extract custom metrics from them.

Log Events, Streams, and Groups

You configure your applications and AWS services to send log events to CloudWatch Logs. A log event is analogous to a line in a log file and always contains a timestamp and an event message. Many AWS services produce their own logs called *vended logs* that you can stream to CloudWatch Logs. Such logs include Route 53 DNS query logs, VPC flow logs, and CloudTrail logs. CloudWatch Logs can also receive custom logs from your applications, such as web server access logs.

CloudWatch Logs organizes log events by log streams by storing log events from the same source in a single log stream. For example, web server access logs from a specific EC2 instance would be stored in one log stream, while Route 53 DNS query logs would be stored in a separate log stream.

CloudWatch further organizes log streams into log groups. To organize related log streams, you can place them into the same log group. For instance, if you have several log streams that are collecting web server log events from multiple web servers, you can group all of those log streams into a single log group.

CloudWatch Logs stores log events indefinitely by default, but you can configure a log group's retention settings to delete events automatically. Retention settings range from 1 day to 10 years. You can also archive your logs by exporting them to an S3 bucket.

Metric Filters

A metric filter extracts data from log events in a log group and stores that data in a custom CloudWatch metric. For example, suppose a log event from a database server contains the time in milliseconds it takes to run a query. You may extract that value and store it as a CloudWatch metric so you can graph it and create an alarm to send a notification when it exceeds a certain threshold.

You can also use metric filters to track the number of times a particular string occurs. This is useful for counting the number of times a particular event occurs in a log, such as an error code. For example, you might want to track how many times a 403 Forbidden error appears in a web server log. You can configure a metric filter to count the number of times the error occurs in a given timeframe—five minutes, for example—and record that value in a CloudWatch custom metric.

Metric filters let you extract or derive quantitative data from log events, so metric values will always be numeric. You can't store a non-numeric string such as an IP address in a metric.

CloudWatch Events

The CloudWatch Events feature lets you continuously monitor for specific events that represent a change in your AWS resources—particularly write-only API operations—and take an action when they occur. For example, an EC2 instance going from the running state to the stopped state would be an event. An IAM user logging into the AWS Management Console would also be an event. CloudWatch Events can then automatically and immediately take actions in response to those events.

You start by creating a rule to define the events to monitor, as well as the actions you want to take in response to those events. You define the action to take by selecting a target, which is an AWS resource. Some targets you can choose from include the following:

- Lambda functions
- EC2 instances
- SQS queues
- SNS topics
- ECS tasks

CloudWatch responds to events as they occur, in real time. Unlike CloudWatch alarms, which take action when a metric crosses and remains crossing a numeric threshold, CloudWatch events trigger immediately. For example, you can create a CloudWatch event to send an SNS notification whenever an EC2 instance terminates. Or you could trigger a Lambda function to process an image file as soon as it hits an S3 bucket.

Cost Explorer updates report data at least once every 24 hours.

Summary

Starting out, you'll spend most of your time interacting with AWS using the AWS Management Console. It's always changing, but even when it does, AWS takes great care to let you know what changed. Sometimes AWS will even let you preview new console features before they go live, giving you the chance to adjust to the change before it's rolled out permanently.

As you find yourself working with AWS more and getting more familiar with the services, you'll begin to use the AWS Command Line Interface for many common tasks. The AWS Command Line Interface is a must for scripting AWS tasks and collecting information from your AWS resources in bulk.

CloudWatch collects metrics from AWS services. You can create alarms to take some action, such as a notification, when a metric crosses a threshold. CloudWatch receives and stores logs from AWS and non-AWS services and even extracts metrics from those logs using metric filters.

CloudTrail records events that occur against your AWS account. By default, the CloudTrail event history log captures the last 90 days of management events in each region. If you want to log more than this or customize the events that it logs, you must create a trail to cause CloudTrail to store events in an S3 bucket. You can also configure a trail to stream logs to CloudWatch Logs for storage, viewing, and searching.

Exam Essentials

Understand when to use the AWS Management Console versus the AWS CLI. The Management Console is required if you want to use the point-and-click interface and want to view visual elements such as CloudWatch graphs or Cost Explorer graphs. You can log into the Management Console using an email address and password for the root account. If you're logging in as an IAM user, you'll need the account alias or number, IAM username, and password. If MFA is set up, you'll be prompted for an MFA one-time passcode. The AWS CLI is what you'll use to manage your AWS resources manually from the command line or using scripts. It's good for repetitive or bulk tasks that would take a long time using the Web. To use the CLI, you need an access key ID and secret key.

Know how to use resource tags and resource groups. Resource tags are keys associated with your AWS resources. A key can optionally contain a value. You can use tags to label your resources according to whatever you like, be it owner, business unit, or

environment. You can group resources into a resource group according to resource tags or CloudFormation stacks.

Be able to identify use cases for CloudWatch. CloudWatch can collect logs and metrics from AWS and non-AWS services. Many AWS services such as EC2 automatically send metric data to CloudWatch. You can create alarms to trigger when a metric falls above or below a threshold. In response to an alarm, you can send a notification using SNS, or you can take an action using an Auto Scaling action or EC2 action. You can also graph metrics to view trends visually. CloudWatch Logs lets you aggregate and search log files. Some services, such as VPC and Route 53, can be configured to stream vended logs to CloudWatch logs. You can extract metrics from these logs using metric filters. CloudWatch events let you take actions in response to specific events that occur with your AWS resources, such as launching an EC2 instance or creating an S3 bucket. Unlike alarms that are triggered by metrics crossing a threshold, CloudWatch Events acts in response to specific API operations.

Know the options for developing applications that integrate with AWS. AWS offers SDKs for a variety of programming languages and platforms. You can use the SDKs to quickly develop desktop, server, web-based, or mobile apps that use AWS services. Although many AWS services offer the HTTPS-based AWS Query API that you can interface with directly, the SDKs handle the heavy lifting of request authentication, serialization, and connection management, freeing you up to write your application without having to learn the nitty-gritty API details of every AWS service you want to use.

Understand what CloudTrail does and how it differs from and integrates with CloudWatch. CloudTrail logs management and data operations on your account. By default, it logs 90 days of management events per region. If you want to log more than this or customize which events it logs, you can create a trail to log those events and store them in an S3 bucket. You can optionally stream CloudTrail logs to CloudWatch for storage, searching, and analysis.

Geolocation A Geolocation policy lets you route users based on their specific continent, country, or state.

Multivalue Answer A Multivalue Answer policy allows you to evenly distribute traffic across multiple resources. Unlike Weighted policies that return a single record, a Multivalue Answer policy returns all records, sorted in a random order.

Health Checks

All routing policies with the exception of Simple can use health checks to determine whether they should route users to a given resource. A health check can check one of three things: an endpoint, a CloudWatch alarm, or another health check. All health checks occur every 10 seconds or 30 seconds.

Endpoint Endpoint health checks work by connecting to the endpoint you want to monitor via HTTP, HTTPS, or TCP. Route 53 has health checkers in several AWS Regions, and you can choose which health checkers a health check uses. This lets you ensure that an endpoint is reachable from various locations around the world.

CloudWatch alarm A Route 53 health check can monitor the status of a CloudWatch alarm. This is useful if you want to consider a resource unhealthy if it's experiencing high latency or is servicing a high number of connections.

Calculated This type of health check monitors the status of other health checks. For example, if you want to consider the status of both an Endpoint health check and a CloudWatch alarm health check, you can create a Calculated health check to take both into account.

Traffic Flow and Traffic Policies

If you require complex routing scenarios for a public hosted zone, creating multiple resource records with a variety of different routing policies can become an administrative nightmare. As an alternative to manually engineering routing policies, you can use the Route 53 Traffic Flow visual editor to create a diagram to represent the desired routing.

The diagram you create represents a traffic policy that you can save and associate with a domain name by creating a policy record. Route 53 doesn't create the individual resource records but instead hides the routing behind the single policy record. The cost is currently $50 USD per month per policy record.

You can use the same routing policies that are available with normal resource records: Simple, Weighted, Latency, Failover, Geolocation, and Multivalue Answer. But in addition, Traffic Flow offers another routing policy that's not otherwise available: Geoproximity. The Geoproximity routing policy lets you direct users to a resource based on how close they are to a geographic location. This differs from the Geolocation routing policy that routes based on the user's specific continent, country, or state.