aws re/start

Amazon CloudWatch

What you will learn

At the core of the lesson

You will learn how to:

- Describe an Amazon Web Services (AWS) monitoring service, Amazon CloudWatch
- Describe the three components of CloudWatch

Key terms:

- Amazon CloudWatch
- Basic Monitoring for Amazon Elastic Compute Cloud (Amazon EC2) instances
- Detailed Monitoring for Amazon EC2 instances
- CloudWatch metric
- CloudWatch alarm
- CloudWatch event

aws re/start

In this module, you will learn how to:

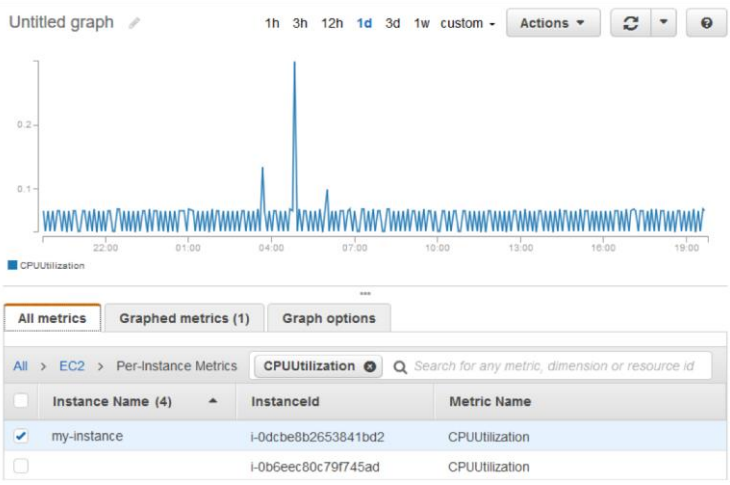
- Describe an AWS monitoring service, Amazon CloudWatch
- Describe the three components of CloudWatch

The goal of this module is to help you understand the monitoring resources that are available to power your solution. You will also review the different service features that are available so you can begin to understand how different choices affect things like solution availability.

Use AWS efficiently and gain insight

Using AWS

- To use AWS in an efficient way, you need insight into your AWS resources:
  - How do you know when you should launch more Amazon Elastic Compute Cloud (Amazon EC2) instances?
  - Is your application's performance or availability being affected because of insufficient capacity?
  - How much of your infrastructure is actually being used?



aws re/start

How do you capture this information? Without any kind of instrumentation, you are at a disadvantage. To use resources efficiently, you need insight into your resources.

You should understand:

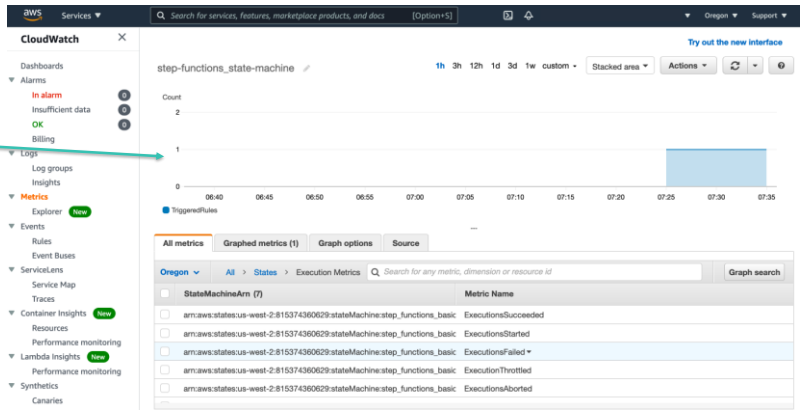
- How to know when you should launch more Amazon Elastic Compute Cloud (Amazon EC2) instances
- Whether your application's performance or availability being affected because of insufficient capacity
- How much of your infrastructure is actually being used

Monitoring resource performance

Amazon CloudWatch

When you monitor workload performance, you should ask yourself two critical questions:

1) How can you ensure that your workload has enough resources to meet fluctuating performance requirements?
2) How can you automate resource provisioning to occur on demand?



aws re/start

You can capture this information with Amazon CloudWatch.

When you run your applications on EC2 instances, it is critical to monitor the performance of your workload by using Amazon CloudWatch. When you monitor workload performance, you should ask yourself two critical questions:

1) How can you ensure that your workload has enough Amazon EC2 resources to meet fluctuating performance requirements?
2) How can you automate Amazon EC2 resource provisioning to occur on demand?

CloudWatch helps with performance monitoring. However, by itself, it will not add or remove EC2 instances. Amazon EC2 Auto Scaling can help with this situation.

With Amazon EC2 Auto Scaling, you can maintain the health and availability of your fleet. You can also dynamically scale your EC2 instances to meet demands during spikes and lulls.

What is Amazon CloudWatch?

Amazon CloudWatch

Monitors the state and utilization of most resources that you can manage under AWS

- Key concepts:
  - Standard metrics
  - Custom metrics
  - Alarms
  - Notifications

CloudWatch agent collects system-level metrics:

- EC2 instances
- On-premises servers

Amazon CloudWatch Terms


Metric

Alarm

Events

Events

5

aws re/start

The primary function of Amazon CloudWatch to monitor the performance and health of your AWS resources and applications. You can also use CloudWatch to collect and monitor log files from EC2 instances, AWS CloudTrail, Amazon Route 53, and other sources.

Amazon CloudWatch is a distributed statistics-gathering system. It collects and tracks your metrics from your applications. You can also create and use your own custom metrics and receive notifications when an alarm goes off.

CloudWatch has two different monitoring options:

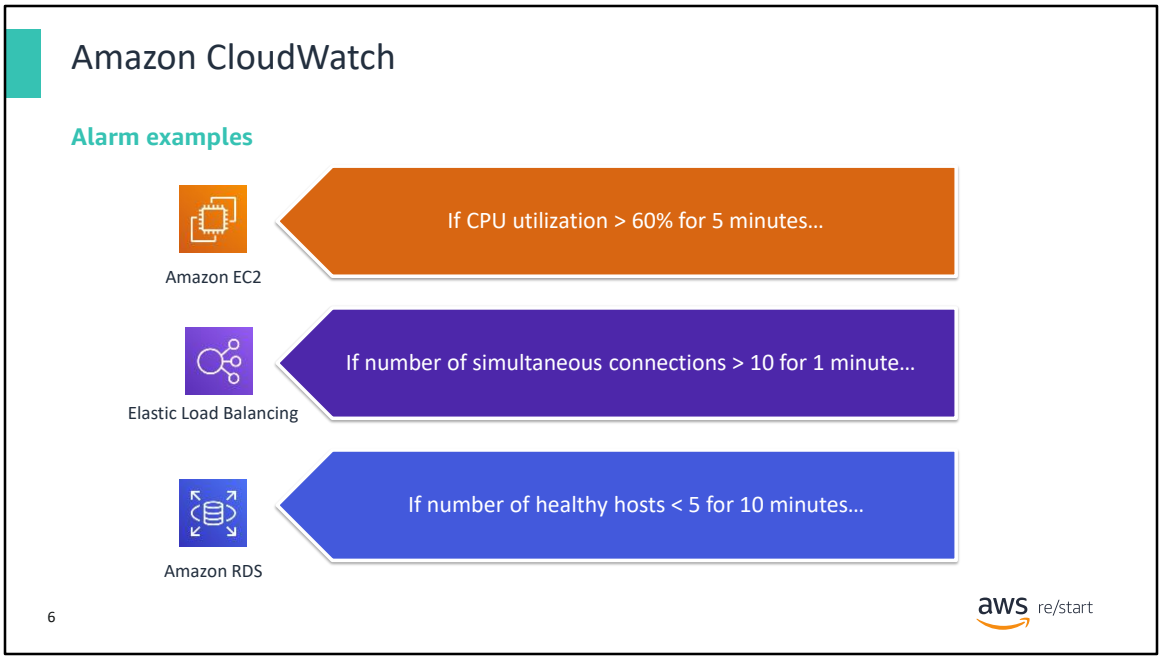
- **Basic Monitoring for Amazon EC2 instances**: Seven pre-selected metrics at a 5-minute frequency and three status check metrics at a 1-minute frequency, for no additional charge.
- **Detailed Monitoring for Amazon EC2 instances**: All metrics that are available to Basic Monitoring at a 1-minute frequency, for an additional charge. Instances with detailed monitoring enabled provide data aggregation by Amazon EC2, Amazon Machine Image (AMI) ID, and instance type.

CloudWatch retains metrics for 15 months, free of charge. CloudWatch metrics

support the following three retention schedules:

- 1-minute data points are available for 15 days.
- 5-minute data points are available for 63 days.
- 1-hour data points are available for 455 days.

To learn more, refer to the Amazon CloudWatch webpage.



These examples show some CloudWatch alarms. Take a moment to review each one.

Amazon CloudWatch actions

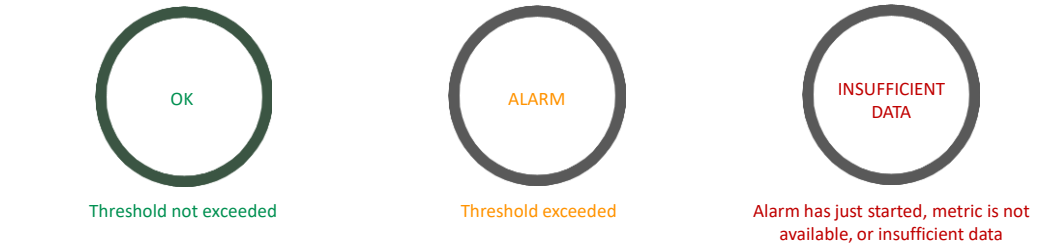
aws re/start

You can choose a number of actions to take based on the CloudWatch alarms.

Amazon CloudWatch alarms

- Test a selected metric against a specific threshold (greater than or equal to, less than or equal to)
- The **ALARM** state is not necessarily an emergency condition

Alarms have three possible states:



Threshold not exceeded

Threshold exceeded

Alarm has just started, metric is not available, or insufficient data

aws re/start

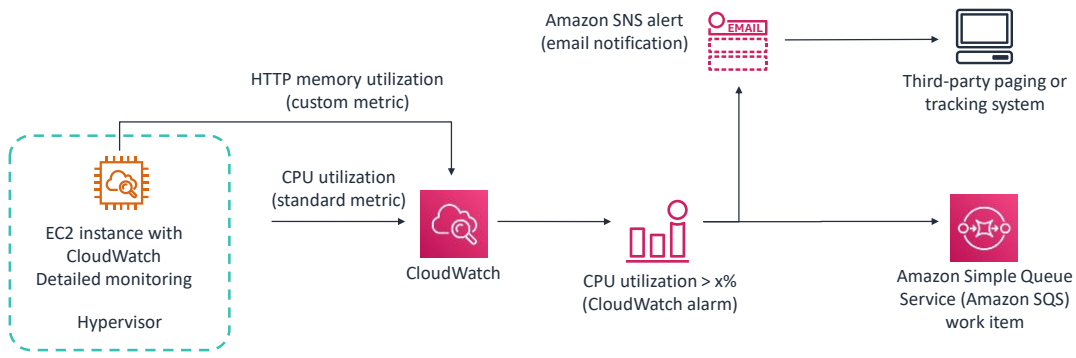
You can create a CloudWatch alarm that watches a single CloudWatch metric or the result of a math expression that is based on multiple CloudWatch metrics. The alarm performs one or more actions based on the value of the metric or expression relative to a *threshold* over several time periods.

An alarm has three possible states:

- *OK* – The metric is within the defined threshold.
- *ALARM* – The metric is outside the defined threshold.
- *INSUFFICIENT_DATA* – The alarm has just started, the metric is not available, or not enough data is available for the metric to determine the alarm state.

NOTE: ALARM is only a name that is given to the state, and does not necessarily signal an emergency condition that requires immediate attention. It means that the monitored metric is equal to, greater than, or less than a specified threshold value. For example, you could define an alarm that tells you when your CPUCreditBalance for a given T2 instance is running low. You might then process this notification programmatically to suspend a CPU-intensive job on the instance until your T2 credit balance is once again full.

CloudWatch monitoring example



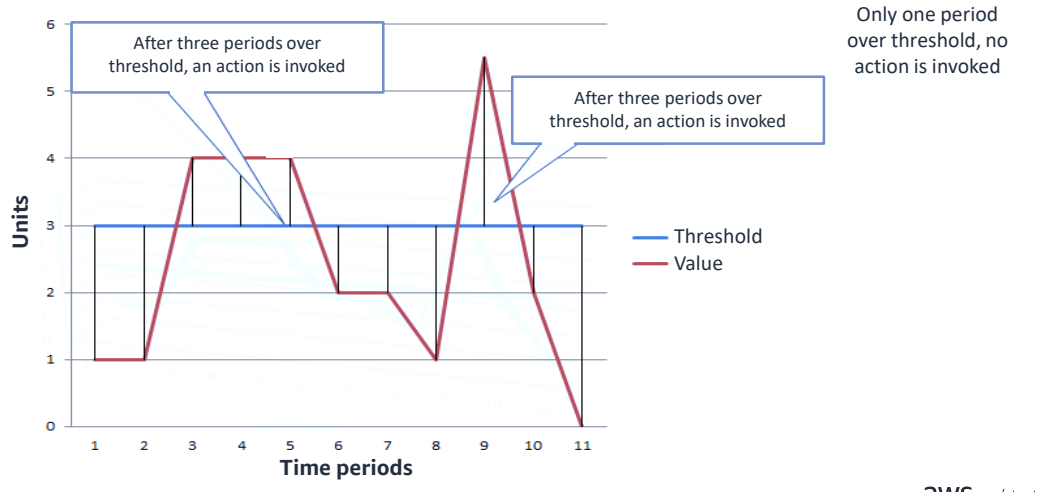
aws re/start

The diagram depicts an example of CloudWatch monitoring in action. On the left, an EC2 instance has a CloudWatch agent installed on it, and detailed monitoring has been enabled.

Two of the metrics that the agent sent are shown. The first metric is the *CPU utilization metric*. CPU utilization is a standard metric that is available in CloudWatch, and it is collected easily. However, the other metric—memory utilization on an EC2 instance—is not visible at the hypervisor layer. For this metric, a custom metric is defined to monitor the memory utilization of the *httpd* service.

Next, a CloudWatch alarm has been configured to trigger whenever the CPU utilization metric exceeds *x* percent. When the alarm is triggered, a message is sent by using Amazon SNS, and an email notification is generated. A third-party paging or tracking system receives the alert, which would likely trigger other actions, such as paging on-shift employees in the IT department. Meanwhile, the same CloudWatch alarm also sent a message on an Amazon Simple Queue Service (Amazon SQS) topic, which generated a work item.

CloudWatch alarms example



aws re/start

In the diagram, the CloudWatch alarm threshold is set to *3* and the minimum breach is *3 periods*. That is, the alarm invokes its action only when the threshold is breached for three consecutive periods. In the diagram, this situation happens with the third through fifth time periods, and the alarm's state is set to *ALARM*. At time period six, the value dips below the threshold, and the state reverts to *OK*. Later, during the ninth time period, the threshold is breached again, but not for the necessary three consecutive periods. Consequently, the alarm's state remains *OK*.

For more information about creating Amazon CloudWatch Alarms, refer to Using Amazon CloudWatch Alarms.

Metric components

Metric	Name and value
Namespace	Groups related metrics together
Dimensions	Name-value pairs that further categorize metrics
	Example: InstanceId is a dimension of CPU utilization
	Metric name + dimension = a new, unique metric
Period	Specified time (in seconds) over which metric was collected

aws re/start

Metrics are the fundamental concept in CloudWatch. A metric represents a time-ordered set of data points that are published to CloudWatch. Think of a metric as a variable to monitor, and the data points represent the values of that variable over time. For example, the CPU usage of a particular EC2 instance is one metric that Amazon EC2 provides. The data points themselves can come from any application or business activity that you collect data from.

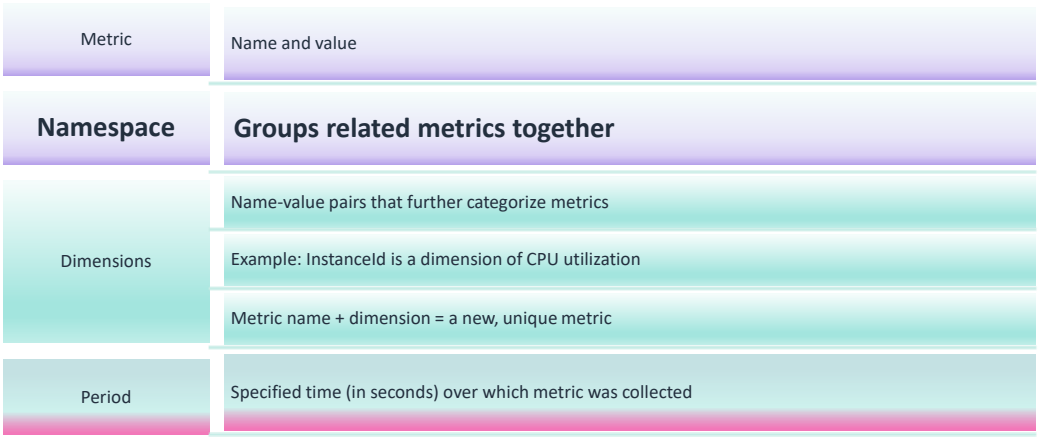
Metrics are uniquely defined by a name, a namespace, and zero or more dimensions. Each data point has a timestamp, and (optionally) a unit of measure. When you request statistics, the returned data stream is identified by namespace, metric name, dimension, and (optionally) the unit. Metrics exist only in the Region where they are created.

- A *namespace* is a container for CloudWatch metrics. Metrics in different namespaces are isolated from each other, so that metrics from different applications are not mistakenly aggregated into the same statistics. The AWS namespaces use the naming convention *AWS/<service>*. For example, Amazon EC2 uses the *AWS/EC2* namespace.
- A *dimension* is a name-value pair that uniquely identifies a metric. You can assign up to 10 dimensions to a metric. Each metric has specific characteristics that describe it, and you can think of dimensions as categories for those characteristics. Dimensions help you design a structure for your statistics plan. You can use dimensions to filter the results that CloudWatch returns. For example, when you

search for metrics, you can get statistics for a particular EC2 instance by specifying the *InstanceId* dimension.

- A *period* is the length of time that is associated with a specific CloudWatch statistic. Periods are defined in numbers of seconds. You can adjust how the data is aggregated by varying the length of the period. A period can be as short as 1 second or as long as 1 day (86,400 seconds).

Metric components - Namespace



aws re/start

A *namespace* is a container for CloudWatch metrics. Metrics in different namespaces are isolated from each other, so that metrics from different applications are not mistakenly aggregated into the same statistics. The AWS namespaces use the naming convention *AWS/<service>*. For example, Amazon EC2 uses the *AWS/EC2* namespace.

Metric components - Dimensions

Metric	Name and value
Namespace	Groups related metrics together
Dimensions	**Name-value pairs that further categorize metrics**
	Example: InstanceId is a dimension of CPU utilization
	Metric name + dimension = a new, unique metric
Period	Specified time (in seconds) over which metric was collected

A *dimension* is a name-value pair that uniquely identifies a metric. You can assign up to 10 dimensions to a metric. Each metric has specific characteristics that describe it, and you can think of dimensions as categories for those characteristics. Dimensions help you design a structure for your statistics plan. You can use dimensions to filter the results that CloudWatch returns. For example, when you search for metrics, you can get statistics for a particular EC2 instance by specifying the *InstanceId* dimension.

Metric components - Period

Metric	Name and value
Namespace	Groups related metrics together
Dimensions	Name-value pairs that further categorize metrics
	Example: InstanceId is a dimension of CPU utilization
	Metric name + dimension = a new, unique metric
Period	**Specified time (in seconds) over which metric was collected**

A *period* is the length of time that is associated with a specific CloudWatch statistic. Periods are defined in numbers of seconds. You can adjust how the data is aggregated by varying the length of the period. A period can be as short as 1 second or as long as 1 day (86,400 seconds).

Metric components

Namespace:

Groups related metrics together

```
{
    "Metrics": [
       {
            "Namespace": "AWS/S3",
            "Dimensions": [
                {
                    "Name“: "StorageType",
                    "Value": "GlacierStorage"
                },
                {
                    "Name": "BucketName",
                    "Value": “DOC-EXAMPLE-BUCKET"
                }
            ],
            "MetricName": "BucketSizeBytes"
        }
    ]
}
```


aws re/start

Here is an example. Suppose you have one (and only one) S3 bucket defined in your account. You run the following AWS Command Line Interface (AWS CLI) command:

```
aws cloudwatch list-metrics --namespace AWS/S3
```

The returned metric data will be similar to the example, where *DOC-EXAMPLE-BUCKET* is the name of the bucket.

The namespace of the bucket is returned, and it indicates that the metric is relevant to the Amazon S3 service. Two *dimensions* are also included in the response, and they are returned as name-value pairs.

Standard and custom metrics

Standard metrics:

- Grouped by service name
- Display graphically so that selected metrics can be compared
- Only appear if you have used the service in the past 15 months
- Reachable programmatically through the AWS Command Line Interface (AWS CLI) or application programming interface (API)

Alarm

Event (time-based)

Custom metrics:

- Grouped by user-defined namespaces
- Publish to CloudWatch by using the AWS CLI, an API, or a CloudWatch agent

Event (event-based)

Rule

16

aws re/start

As mentioned previously, CloudWatch has standard metrics and custom metrics.

Standard CloudWatch metrics are *grouped by service*. For example, if you open the AWS Management Console and then the CloudWatch service screen, you can choose a link to view all Amazon EC2 metrics. The metrics *display graphically* so that they can be compared.

To view available metrics by using the AWS CLI, use the `list-metrics` command to list them. For example, running the command `aws cloudwatch list-metrics --namespace AWS/S3` lists all the available standard Amazon S3 metrics.

Metrics cannot be deleted, but they automatically expire after 15 months if no new data is published to them. Data points that are older than 15 months expire on a rolling basis. As new data points come in, data that is older than 15 months is dropped.

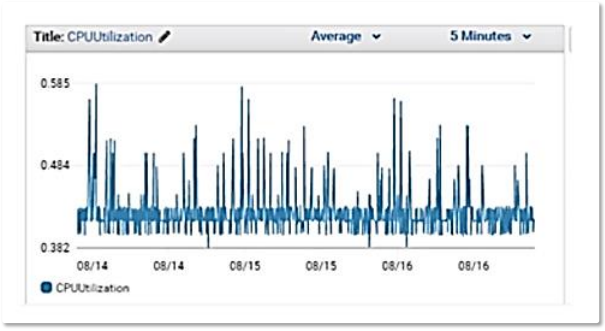
AWS services send metrics to CloudWatch. You can also publish your own metrics to CloudWatch by using the AWS CLI, an application programming interface (API), or a CloudWatch agent. Custom metrics are grouped by the namespace that you define

when you create them.

Monitoring and security

Use **CloudWatch** to monitor for suspicious activity, such as:

- Unusual, prolonged spikes in service usage, such as CPU, disk activity, or Amazon Relational Database (Amazon RDS) usage
- Set alerts on billing metrics (you must enable this feature in account settings)



aws re/start

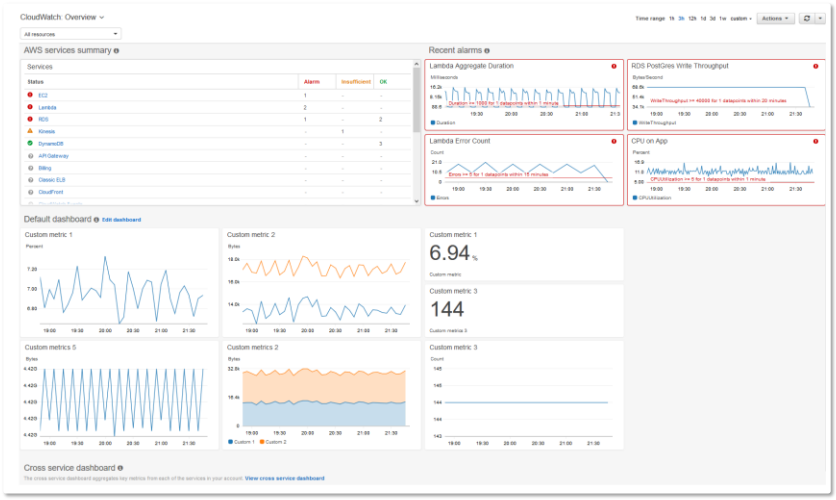
One common use of Amazon CloudWatch is to monitor account resources for suspicious activity.

For example, generating alerts based on billing data is a good way to detect a potential security violation of your AWS account. Some customers do not know that their credentials or AWS Identity and Access Management (IAM) access keys have been compromised until they receive a bill for thousands of dollars of unexpected charges. To detect this situation proactively, you could enable billing alerts in your account preferences, and then set CloudWatch alarms to alert you if estimated billing charges for the month have exceeded a specified threshold.

CloudWatch automatic dashboards

Amazon CloudWatch dashboards:

- Surface data about your running AWS services
- Can be used through existing monitoring tools



aws re/start

Amazon CloudWatch dashboards are customizable homepages in the CloudWatch console that you can use to monitor your resources in a single view. You can create customized views of the metrics and alarms for your AWS resources.

You can get aggregated views of the health and performance of all AWS resources through CloudWatch *automatic dashboards*. This feature enables you to monitor and explore account-based and resource-based views of metrics and alarms. You can drill down to figure out the root cause of performance issues.

Automatic dashboards are prebuilt with recommended best practices for AWS services. They remain resource aware, and they dynamically update to reflect the latest state of important performance metrics.

In the console screen capture, the upper-left area shows a list of AWS services that are being used in this account. It also shows the state of any alarms that are configured for those services. The upper-right area shows any recent alarms that were triggered. The number of alarms that display here will depend on whether alarms were both configured and triggered.

The lower part of the dashboard optionally enables you to add metrics about your own custom services or applications to the overview page. You can also bring forward additional key metrics from AWS services that you most want to monitor. By default, the bottom part of the automatic dashboard shows *Cross service dashboard metrics*.

However, it is a link to **Create a new Cloudwatch-Default dashboard**. If you name the new dashboard *CloudWatch-Default*, it displays on the main **CloudWatch:Overview** dashboard page.

Activate detailed instance monitoring

- By default, EC2 instances only **report data every 5 minutes** (AWS Free Tier).
  - For more information about the AWS Free Tier, refer to AWS Free Tier webpage.
- Enable detailed monitoring on an instance to **increase reporting frequency to once every minute**.
  - Extra charges apply

aws re/start

By default, EC2 instances are enabled for basic CloudWatch monitoring, with data available in *5-minute increments* as part of the AWS Free Tier. However, you can also enable detailed monitoring at an additional cost. After detailed monitoring is enabled, the monitoring data becomes available in *1-minute increments*.

For more information about the AWS Free Tier, refer to the AWS Free Tier webpage.

For more information about detailed monitoring, refer to Enable or turn off detailed monitoring for your instances.

Key takeaways

- Amazon CloudWatch monitors the performance and health of your resources and applications.
- It enables you to –
  - Track resource and application performance
  - Collect and monitor log files
  - Get notifications when an alarm goes off
- CloudWatch consists of three primary components –
  - Metrics
  - Alarms
  - Events

aws re/start

In summary, Amazon CloudWatch tracks and monitors the performance and health of your resources and applications.

It enables you to:
- Track resource and application performance
- Collect and monitor log files
- Get notified when an alarm goes off

CloudWatch consists of three primary components: metrics, alarms, and events.