



**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

Application Logging

Cloud Infrastructure Engineering

**Nanyang Technological University
& Skills Union - 2022/2023**

Course Content

- Learners will understand:
 - Importance of application logging
 - Familiarizing with AWS CloudWatch
- Learners will be able to:
 - Set up Application Logging with AWS CloudWatch
 - Analyzing Log Data with AWS CloudWatch
 - Integrating AWS CloudWatch with Other AWS Services

Q1: Why do we need application logging in place?



Q2: What are some common monitoring/ logging tools in the market?



Q3: What are some metrics we would like to monitor?



Activity

Instructor

- Ask to use AWS use single region for all learner for easier monitoring

What is Application Logging?

- Application logging is the process of **recording events and data about an application's behavior and performance**. The main purpose of application logging is to provide information about the application's state and behavior, which can be used for troubleshooting, performance analysis, and security monitoring.
- Application logs can include a wide range of information, such as **error messages, performance metrics, and security events**. This information can be used to **diagnose problems, identify trends and patterns, and monitor the application's overall health and performance**.

What is Application Logging?

- Application logging can be done at different levels of detail, **ranging from high-level events to low-level debugging information**. The level of detail that is logged depends on the requirements of the specific application and the use case for the log data.
- Properly implemented, application logging can **provide valuable insight into an application's behavior, helping organizations to improve reliability, performance, and security**.

Goals of Application Logging

- Provide better/ full visibility of all the events in your application, especially ones that might cause or are **causing errors and warnings**.
- Provide a good starting point for software engineers, production support and operations teams to start debugging and allow for permanent fixes in the long run.

Examples of Application Logging tools



Syslog



AWS CloudWatch



elasticsearch



logstash



kibana



fluentd

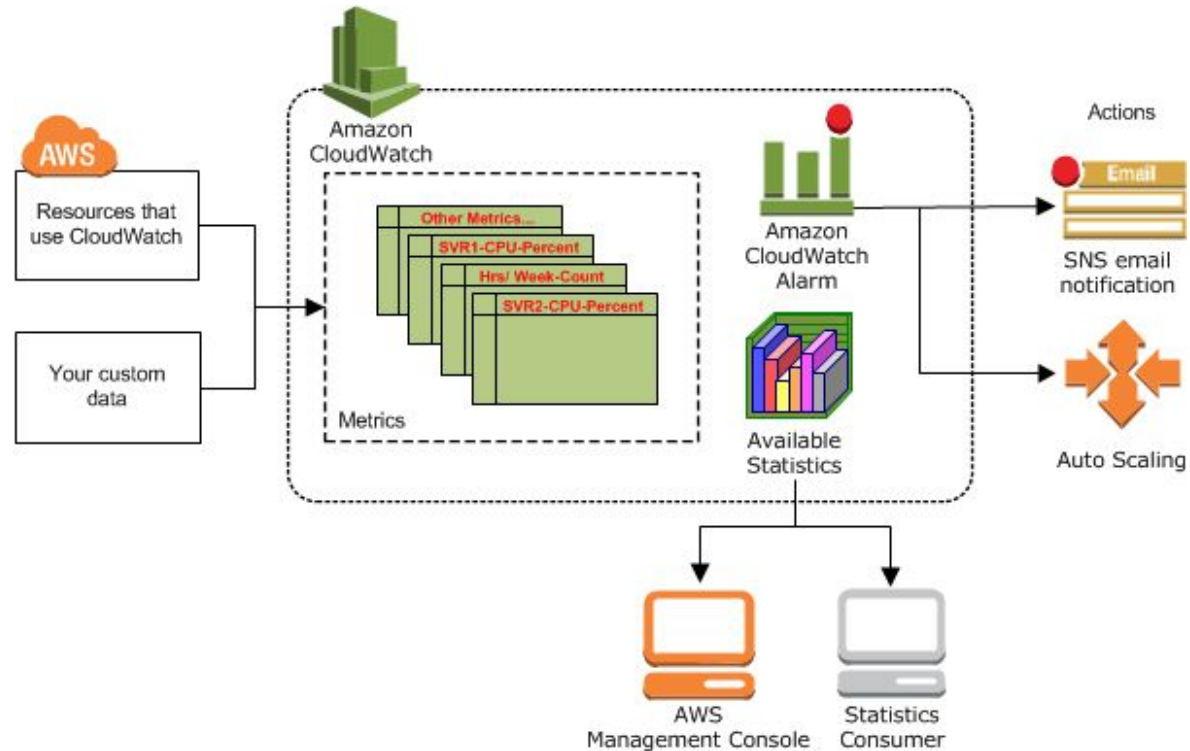
AWS CloudWatch



What is CloudWatch?

- AWS CloudWatch is a cloud-based **monitoring service** provided by AWS that allows organizations to **monitor and log performance metrics, events, and logs from a variety of AWS resources**. CloudWatch provides a **centralized platform** for monitoring and analyzing log data, making it easier to **identify trends, resolve issues, and improve performance**.
- With AWS CloudWatch, organizations can **collect and store log data** from a wide range of sources, including AWS services, on-premises servers, and custom applications. The service provides **real-time monitoring and alerting capabilities**, allowing organizations to receive notifications about critical issues in real-time.

What is CloudWatch?



What is CloudWatch?



Features of CloudWatch

AWS CloudWatch provides a range of features and tools for working with log data, including:

- **Log Insights:** A search and analysis tool that allows users to search, filter, and visualize log data.
- **Metric Filters:** A feature that allows users to extract and transform log data into custom metrics, making it easier to monitor specific aspects of an application.
- **Dashboards:** A visual representation of log data, including charts, graphs, and tables, that can be used to monitor key performance metrics and events.
- **Alarms:** A feature that allows users to set thresholds on log data and receive notifications when those thresholds are exceeded.

Features of CloudWatch

AWS CloudWatch is **highly scalable and available**, allowing organizations to monitor and analyze log data from a wide range of sources, regardless of volume or location.

By using AWS CloudWatch, organizations can **improve their visibility into the performance and behavior of their applications**, helping them to resolve issues more quickly, monitor performance more effectively, and improve security.

Setting Up AWS CloudWatch



Pre-requisites

Create an AWS Account.

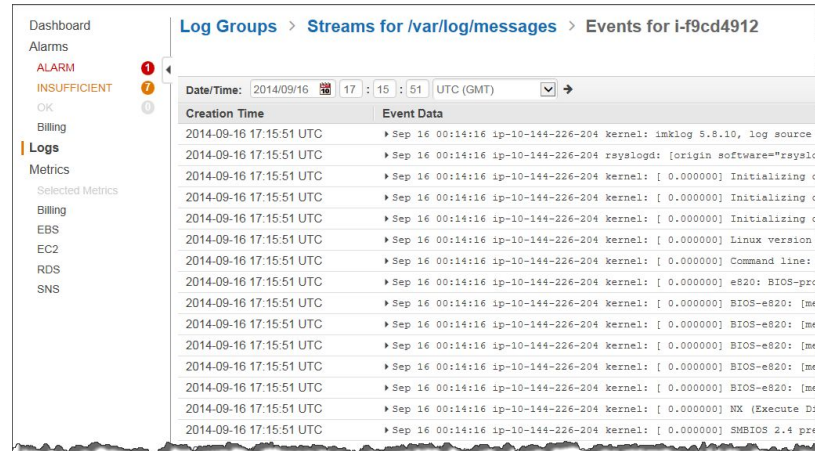
If you don't already have an AWS account, you will need to create one. This can be done by visiting the AWS website and following the on-screen instructions.

Getting Started

Configure a Log Group: Log Groups are collections of log streams that represent a group of log events.

Getting Started

Send Log Data to CloudWatch: To send log data to CloudWatch, you need to configure the source of the logs to send data to the appropriate Log Group. This can be done using a variety of methods, including using the CloudWatch Logs agent, using the CloudWatch API, or by integrating with other AWS services such as Amazon EC2.



The screenshot shows the AWS CloudWatch console interface. On the left is a navigation menu with links to Dashboard, Alarms, Billing, Logs, Metrics, EBS, EC2, RDS, and SNS. The 'Logs' section is selected. The main area displays the breadcrumb path: Log Groups > Streams for /var/log/messages > Events for i-f9cd4912. Below this, there is a filter bar for 'Date/Time' set to 2014/09/16 17:15:51 UTC (GMT). A table of log events follows, with columns for 'Creation Time' and 'Event Data'. The events are kernel logs from an instance, showing messages like 'imklog 5.8.10, log source', 'rsyslogd: [origin software="rsyslogd"', 'Initializing', 'Linux version', 'Command line:', 'e820: BIOS-pro', 'BIOS-e820: [ma', 'BIOS-e820: [mem', 'BIOS-e820: [ma', 'BIOS-e820: [ma', 'BIOS-e820: [ma', 'NX (Execute Dis', and 'SMBIOS 2.4 pre'.

Creation Time	Event Data
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: imklog 5.8.10, log source
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 rsyslogd: [origin software="rsyslogd"
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] Initializing
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] Initializing
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] Initializing
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] Initializing
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] Linux version
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] Command line:
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] e820: BIOS-pro
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] BIOS-e820: [ma
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] BIOS-e820: [mem
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] BIOS-e820: [ma
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] BIOS-e820: [ma
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] BIOS-e820: [ma
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] NX (Execute Dis
2014-09-16 17:15:51 UTC	» Sep 16 00:14:16 ip-10-144-226-204 kernel: [0.000000] SMBIOS 2.4 pre

Getting Started

Configure Metric Filters: Metric filters are used to extract and transform log data into custom metrics that can be used for monitoring.

aws

Services

Resource Groups

Step 1: Define Pattern

Step 2: Assign Metric

Define Logs Metric Filter

Editing Filter "message-Web-management-request-allowed-2" for Log Group "Sonicwall_Log_Group"

You can use metric filters to monitor events in a log group as they are sent to CloudWatch Logs. You can monitor and count specific terms or extract values from log events and associate the results with a metric. [Learn more about pattern syntax.](#)

Filter Pattern

(\$ message="Web management request allowed")

Show examples

Select Log Data to Test

SW_Log_Stream

Test Pattern

Clear

```
{ "serial": "C0EAE4CED0F8", "timestamp": 1560278582, "msgId": 608, "categoryname": "Security"
{ "serial": "C0EAE4CED0F8", "timestamp": 1560278609, "msgId": 608, "categoryname": "Security"
{ "serial": "C0EAE4CED0F8", "timestamp": 1560278613, "msgId": 38, "categoryname": "Network", "
{ "serial": "C0EAE4CED0F8", "timestamp": 1560278623, "msgId": 36, "categoryname": "Network", "
{ "serial": "C0EAE4CED0F8", "timestamp": 1560278675, "msgId": 38, "categoryname": "Network", "
{ "serial": "C0EAE4CED0F8", "timestamp": 1560278684, "msgId": 36, "categoryname": "Network", "

```

Results

Found 4 matches out of 50 event(s) in the sample log.

Show test results

Cancel

Assign Metric

Getting Started

Create Alarms: Alarms are used to set thresholds on log data and receive notifications when those thresholds are exceeded.

Create Alarm

1. Select Metric

2. Define Alarm

Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name:

test

Description:

my first alarm

Whenever:

CassServerCount

is:

<

6

for:

1

consecutive period(s)

Actions

Define what actions are taken when your alarm changes state.

Notification

Delete

Whenever this alarm:

State is ALARM

Send notification to:

New list

Email list:

+ Notification


+ AutoScaling Action

+ EC2 Action

Alarm Preview

This alarm will trigger when the blue line goes below the red line for a duration of 5 minutes

CassServerCount < 6



Namespace:

C5

Metric Name:

CassServerCount

Period:

5 Minutes

Statistic:

Average

Cancel

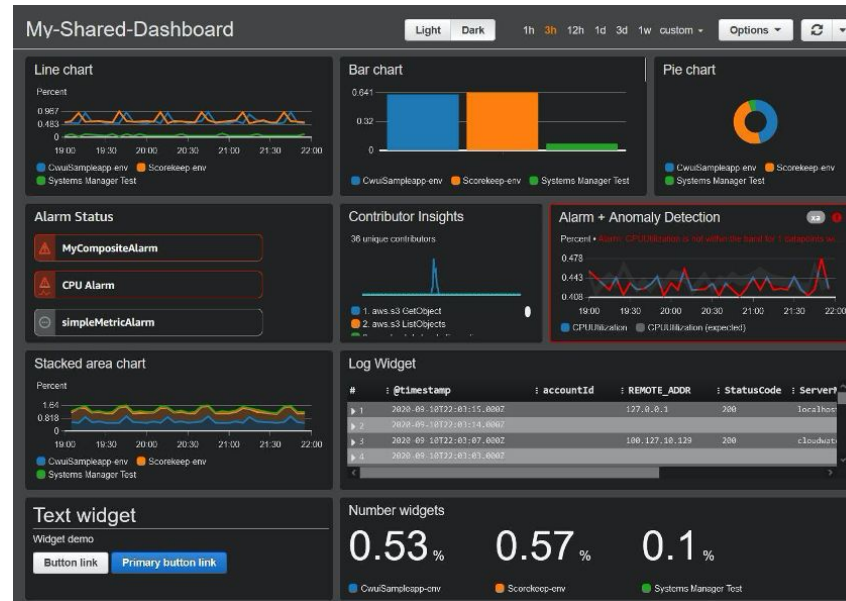
Back

Next

Create Alarm

Getting Started

Create Dashboards: Dashboards are used to visualize log data, including charts, graphs, and tables.



Types of CloudWatch Queries

Simple Text Search: A simple text search allows you to search log messages for a specific string or phrase. For example, you can search for all log messages that contain the word "error".

Field Search: A field search allows you to search log messages based on specific fields, such as timestamp, severity, and source. For example, you can search for all log messages that were generated by a specific application.

Regular Expression Search: A regular expression search allows you to search log messages using a pattern, such as a regular expression. For example, you can search for all log messages that contain an IP address.

Types of CloudWatch Queries

Metric Filter: A metric filter allows you to search log data based on a specific metric, such as error rate or response time. For example, you can search for all log messages that contain a response time that is greater than a specified threshold.

Structured Query: A structured query allows you to search log data using a specific structure, such as JSON. For example, you can search for all log messages that contain specific values for a specific field, such as "severity":"error".

Examples of CloudWatch Queries

Search for all log messages containing the word "error":

```
fields @timestamp, @message
```

```
| sort @timestamp desc
```

```
| filter @message like /error/
```

Examples of CloudWatch Queries

Search for all log messages generated by a specific application:

```
fields @timestamp, @message
```

```
| sort @timestamp desc
```

```
| filter @message like /MyApplication/
```

Examples of CloudWatch Queries

Search for all log messages containing a specific IP address:

```
| sort @timestamp desc
```

```
| filter @message like /[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+/
```

Examples of CloudWatch Queries

Search for all log messages with a response time greater than 1 second:

```
fields @timestamp, @message
```

```
| sort @timestamp desc
```

```
| filter @message like /response time: [1-9]+.[0-9]+s/
```

Examples of CloudWatch Queries

Search for all log messages containing specific values for a specific field, such as "severity":"error":

```
fields @timestamp, @message
```

```
| sort @timestamp desc
```

```
| parse @message "severity\"\":\"*\""" as severity
```

```
| filter severity like /error/
```

AWS CloudWatch Best Practices



Centralized Logging

Centralizing log data makes it **easier to manage, monitor, and search logs**, and helps ensure that logs are not lost if individual instances fail.

AWS CloudWatch provides a centralized log repository, which makes it easier to manage and monitor logs from multiple sources.

Log Contextual Information

Logging contextual information such as **request data, user information, and error details** makes it easier to understand and diagnose issues.

This information can be included in log messages using standard log fields, such as timestamp, severity, and source.

Use Structured Log Data

Using structured log data, such as **JSON**, makes it **easier to search, filter, and analyze log data**.

AWS CloudWatch provides native support for structured log data, which makes it easier to search and filter log messages.

Enable Log Retention

Enable log retention to ensure that **logs are kept for a specified period of time**.

Log retention can be configured using AWS CloudWatch, which provides the ability to set the retention period for logs, as well as the ability to archive logs to **Amazon S3** for longer-term storage.

Use Alarms & Metrics

Use alarms and metrics to **monitor important log data** and **receive notifications** when thresholds are exceeded.

Alarms and metrics can be **used to monitor things like error rates, response times, and resource utilization**, which can help **identify issues** before they become major problems.

Rotate Logs Regularly

Regularly rotating logs helps ensure that **logs are not lost**, and makes it easier to **manage disk space and storage costs**.

AWS CloudWatch provides the ability to rotate logs automatically, which makes it easier to manage disk space and storage costs.

Monitor & Analyze Log Data

Monitoring and analyzing log data is key to ensuring that your application is running smoothly and that **issues are identified and addressed quickly**.

AWS CloudWatch provides a variety of tools and features, such as **dashboards, alarms, and metrics**, that make it **easier to monitor and analyze log data**.

Activity

Learner:

- Clean up AWS.
- Remove/delete/terminate all service/ resources that you created.

Instructor

- Clean up AWS.
- Remove/delete/terminate all service/ resources that you created.
- Check the AWS account after learner clean up.

End

