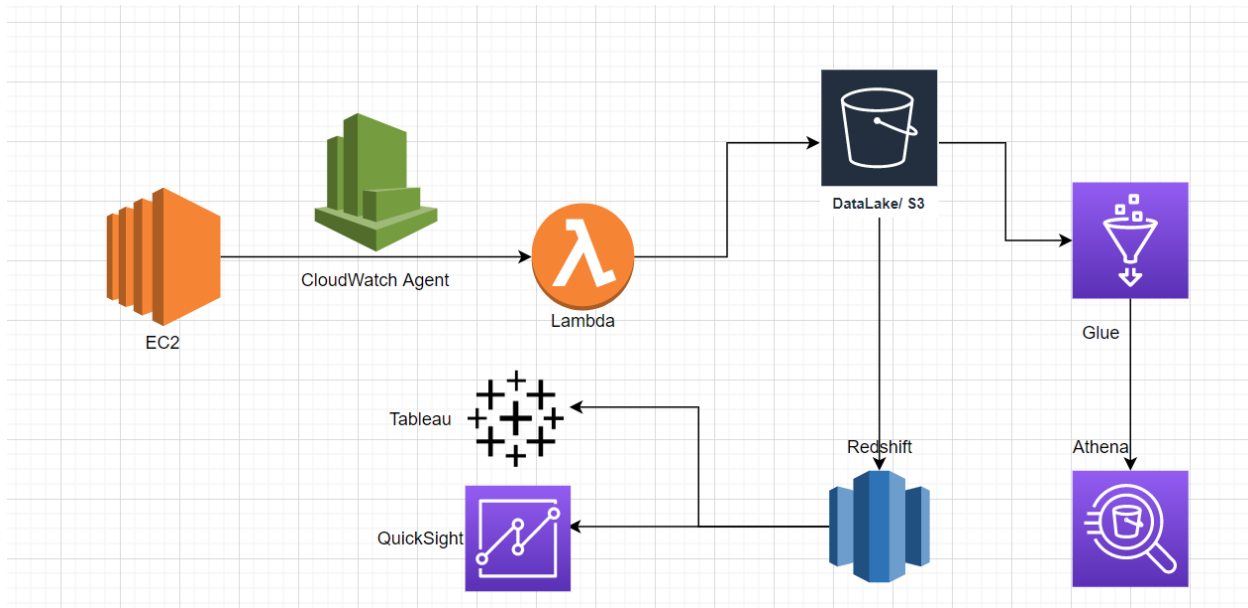


How to capture logs; store them in a data lake, and data warehouse; analyze; and publish reports, dashboards.

A complete end to end solution and a step-by-step implementation process

Enterprise Log Management Architecture



Steps to implement the solution:

1. Create the appropriate IAM role
2. Launch an EC2 instance (the server)
3. Install the httpd
4. Start httpd
5. Access two types of logs – access logs & error logs
6. Get the amazon-cloudwatch-agent.rpm
7. Run the agent wizard to install the cloudwatch agent on each server
8. Once complete verify the CloudWatch log groups
9. Create a Lambda function to copy the data into data lake/ S3
10. Use EventBridge to schedule to copy data into data lake/ S3
11. Using Athena & Glue create the DB and tables to query/ analyze the log data
12. Copy data into Redshift from S3
13. Create reports/ dashboards using AWS QuickSight or Tableau out of Redshift

Create an IAM Role

mm-cloudwatchagent-role with [CloudWatchAgentServerPolicy](#); [CloudWatchAgentAdminPolicy](#)

Install and start the httpd

\$ sudo yum install httpd

[ec2-user@ip-10-0-45-76 html]\$ ls -ltr

total 4

-rwxrwxrwx 1 root root 31 May 17 19:47 index.html

[ec2-user@ip-10-0-45-76 html]\$ pwd

/var/www/html

[ec2-user@ip-10-0-45-76 html]\$

[ec2-user@ip-10-0-45-76 html]\$ **sudo systemctl start httpd**

Accessing the log files from /var/log/httpd/

[ec2-user@ip-10-0-45-76 log]\$ sudo cat /var/log/httpd/access_log

73.192.163.126 - - [17/May/2022:19:52:14 +0000] "GET / HTTP/1.1" 200 31 "-" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36"

73.192.163.126 - - [17/May/2022:19:52:15 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "http://ec2-54-75-110-66.eu-west-1.compute.amazonaws.com/" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36"

[ec2-user@ip-10-0-45-76 log]\$ sudo cat /var/log/httpd/error_log

[Tue May 17 19:51:48.684227 2022] [suexec:notice] [pid 3500] AH01232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)

[Tue May 17 19:51:48.699295 2022] [lbmethod_heartbeat:notice] [pid 3500] AH02282: No slotmem from mod_heartbeat

[Tue May 17 19:51:48.699335 2022] [http2:warn] [pid 3500] AH10034: The mpm module (prefork.c) is not supported by mod_http2. The mpm determines how things are processed in your server. HTTP/2 has more demands in this regard and the currently selected mpm will just not do. This is an advisory warning. Your server will continue to work, but the HTTP/2 protocol will be inactive.

[Tue May 17 19:51:48.702387 2022] [mpm_prefork:notice] [pid 3500] AH00163: Apache/2.4.53 () configured -- resuming normal operations

[Tue May 17 19:51:48.702412 2022] [core:notice] [pid 3500] AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'

#!/bin/bash

#Install the agent

wget https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm

```
[ec2-user@ip-10-0-45-76 ~]$ wget https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm
```

```
--2022-05-17 21:32:22-- https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm
```

```
Resolving s3.amazonaws.com (s3.amazonaws.com)... 52.217.229.128
```

```
Connecting to s3.amazonaws.com (s3.amazonaws.com)|52.217.229.128|:443... Connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 46945036 (45M) [application/octet-stream]
```

```
Saving to: 'amazon-cloudwatch-agent.rpm'
```

```
100%[=====
=====>] 46,945,036 19.3MB/s in 2.3s
```

```
2022-05-17 21:32:24 (19.3 MB/s) - 'amazon-cloudwatch-agent.rpm' saved
[46945036/46945036]
```

```
[ec2-user@ip-10-0-45-76 ~]$
```

Install the package. If you downloaded an RPM package on a Linux server, change to the directory containing the package and enter the following:

```
sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

```
[ec2-user@ip-10-0-45-76 ~]$ sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

```
create group cwagent, result: 0
```

```
create user cwagent, result: 0
```

```
create group aoc, result: 0
```

```
create user aoc, result: 0
```

```
[ec2-user@ip-10-0-45-76 ~]$
```

Run the wizard

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
```

```
[ec2-user@ip-10-0-45-76 bin]$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
```

```
=====
```

```
= Welcome to the Amazon CloudWatch Agent Configuration Manager =
```

```
=                               =
```

```
= CloudWatch Agent allows you to collect metrics and logs from =
```

```
= your host and send them to CloudWatch. Additional CloudWatch =
```

```
= charges may apply.                               =
```

```
=====
```

```
On which OS are you planning to use the agent?
```

```
1. linux
```

2. windows

3. darwin

default choice: [1]:

1

Trying to fetch the default region based on ec2 metadata...

Are you using EC2 or On-Premises hosts?

1. EC2

2. On-Premises

default choice: [1]:

1

Which user are you planning to run the agent?

1. root

2. cwagent

3. others

default choice: [1]:

1

Do you want to turn on StatsD daemon?

1. yes

2. no

default choice: [1]:

1

Which port do you want StatsD daemon to listen to?

default choice: [8125]

What is the collect interval for StatsD daemon?

1. 10s

2. 30s

3. 60s

default choice: [1]:

3

What is the aggregation interval for metrics collected by StatsD daemon?

1. Do not aggregate

2. 10s

3. 30s

4. 60s

default choice: [4]:

4

Do you want to monitor metrics from CollectD? WARNING: CollectD must be installed or the Agent will fail to start

1. yes

2. no

default choice: [1]:

1

Do you want to monitor any host metrics? e.g. CPU, memory, etc.

1. yes

2. no

default choice: [1]:

1

Do you want to monitor cpu metrics per core?

1. yes

2. no

default choice: [1]:

1

Do you want to add ec2 dimensions (ImageId, InstanceId, InstanceType, AutoScalingGroupName) into all of your metrics if the info is available?

1. yes

2. no

default choice: [1]:

1

Do you want to aggregate ec2 dimensions (InstanceId)?

1. yes

2. no

default choice: [1]:

Are you satisfied with the above config? Note: it can be manually customized after the wizard completes to add additional items.

1. yes

2. no

default choice: [1]:

1

Do you have any existing CloudWatch Log Agent
(<http://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AgentReference.html>)
configuration file to import for migration?

1. yes

2. no

default choice: [2]:

2

Do you want to monitor any log files?

1. yes

2. no

default choice: [1]:

1

Log file path:

/var/log/httpd/access_log

Log group name:

default choice: [access_log]

Log stream name:

default choice: [{instance_id}]

Log Group Retention in days

1. -1

2. 1

3. 3

4. 5

5. 7

6. 14

7. 30

8. 60

9. 90

10. 120

11. 150

12. 180

13. 365

14. 400

15. 545

16. 731

17. 1827

18. 3653

default choice: [1]:

Do you want to specify any additional log files to monitor?

1. yes

2. no

default choice: [1]:

/var/log/httpd/error_log

The value /var/log/httpd/error_log is not valid to this question.

Please retry to answer:

Do you want to specify any additional log files to monitor?

1. yes

2. no

default choice: [1]:

1

Log file path:

/var/log/httpd/error_log

Log group name:

default choice: [error_log]

Log stream name:

default choice: [{instance_id}]

Log Group Retention in days

1. -1

2. 1

3. 3

4. 5

5. 7

6. 14

7. 30

8. 60

9. 90

10. 120

11. 150

12. 180

13. 365

14. 400

15. 545

16. 731

17. 1827

18. 3653

default choice: [1]:

Do you want to specify any additional log files to monitor?

1. yes

2. no

default choice: [1]:

2

Saved config file to /opt/aws/amazon-cloudwatch-agent/bin/config.json successfully.

Current config as follows:

```
{  
  
  "agent": {  
  
    "metrics_collection_interval": 60,  
  
    "run_as_user": "root"  
  
  },  
  
  "logs": {  
  
    "logs_collected": {
```

```
"files": {  
  "collect_list": [  
    {  
      "file_path": "/var/log/httpd/access_log",  
      "log_group_name": "access_log",  
      "log_stream_name": "{instance_id}",  
      "retention_in_days": -1  
    },  
    {  
      "file_path": "/var/log/httpd/error_log",  
      "log_group_name": "error_log",  
      "log_stream_name": "{instance_id}",  
      "retention_in_days": -1  
    }  
  ]  
}  
  
},  
  
"metrics": {  
  "aggregation_dimensions": [  

```

```
[
    "InstanceId"
]
],
"append_dimensions": {
    "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
    "ImageId": "${aws:ImageId}",
    "InstanceId": "${aws:InstanceId}",
    "InstanceType": "${aws:InstanceType}"
},
"metrics_collected": {
    "collectd": {
        "metrics_aggregation_interval": 60
    },
    "disk": {
        "measurement": [
            "used_percent"
        ],
        "metrics_collection_interval": 60,
        "resources": [
```

```

        "x":
    ]
},
"mem": {
    "measurement": [
        "mem_used_percent"
    ],
    "metrics_collection_interval": 60
},
"statsd": {
    "metrics_aggregation_interval": 60,
    "metrics_collection_interval": 60,
    "service_address": ":8125"
}
}
}
}

```

Please check the above content of the config.

The config file is also located at `/opt/aws/amazon-cloudwatch-agent/bin/config.json`.

Edit it manually if needed.

Do you want to store the config in the SSM parameter store?

1. yes

2. no

default choice: [1]:

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c ssm:configuration-parameter-store-name -s
```

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c ssm:AmazonCloudWatch-linux -s
```

OR

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:configuration-file-path -s
```

Create the types.db file

```
[ec2-user@ip-10-0-45-76 share]$ sudo mkdir -p /usr/share/collectd
```

```
[ec2-user@ip-10-0-45-76 share]$ sudo touch /usr/share/collectd/types.db
```

```
[ec2-user@ip-10-0-45-76 bin]$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json -s
```

```
***** processing amazon-cloudwatch-agent *****
```

```
/opt/aws/amazon-cloudwatch-agent/bin/config-downloader --output-dir /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d --download-source file:/opt/aws/amazon-cloudwatch-agent/bin/config.json --mode ec2 --config /opt/aws/amazon-cloudwatch-agent/etc/common-config.toml --multi-config default
```


2022/05/17 22:50:11 D! [EC2] Found active network interface

Successfully fetched the config and saved in /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp

Start configuration validation...

```
/opt/aws/amazon-cloudwatch-agent/bin/config-translator --input /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json --input-dir /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d --output /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml --mode ec2 --config /opt/aws/amazon-cloudwatch-agent/etc/common-config.toml --multi-config default
```

2022/05/17 22:50:11 Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp ...

2022/05/17 22:50:11 !! Valid Json input schema.

!! Detecting run_as_user...

2022/05/17 22:50:11 D! [EC2] Found active network interface

No csm configuration found.

Configuration validation first phase succeeded

```
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent -schematest -config /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml
```

Configuration validation second phase succeeded

Configuration validation succeeded

amazon-cloudwatch-agent has already been stopped

Created symlink from /etc/systemd/system/multi-user.target.wants/amazon-cloudwatch-agent.service to /etc/systemd/system/amazon-cloudwatch-agent.service.

Redirecting to /bin/systemctl restart amazon-cloudwatch-agent.service

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a status
```

CloudWatch Log Groups, Agent, LoGroupFilter:

The top screenshot shows the AWS CloudWatch console 'Log groups' page. The left sidebar shows the 'Log groups' link is selected. The main content area shows a list of log groups. The 'access_log' and 'error_log' log groups are highlighted in yellow. The table below shows the log groups and their retention periods.

Log group	Retention	Metric filters	Contributor Insights
/aws/lambda/awstest123func	3 months	-	-
/ecs/first-run-task-definition	Never expire	-	-
access_log	Never expire	-	-
error_log	Never expire	-	-

The bottom screenshot shows the AWS CloudWatch console 'Metrics' page. The left sidebar shows the 'Metrics' link is selected. The main content area shows a list of metrics. The 'ImageId, InstanceId, InstanceType, device, fstype, path' metric is highlighted in yellow. The table below shows the metrics and their dimensions.

Metric	Dimensions
ImageId, InstanceId, InstanceType, device, fstype, path	6
ImageId, InstanceId, InstanceType	1
InstanceId	2

Troubleshooting in case of any issues:

<https://aws.amazon.com/premiumsupport/knowledge-center/cloudwatch-push-logs-with-unified-agent/>

Lambda function & EventBridge code to copy the data into data lake/ s3:

The screenshot displays the AWS Lambda console interface. The top section shows the `index.js` code for a Lambda function. The code uses the `aws-sdk` to create a `CloudWatchLogs` client and then calls `createExportTask` to export data to S3. The function returns a 200 status code on success and a 501 status code on error.

```
1 const AWS = require('aws-sdk')
2 const cloudconfig = {
3   apiVersion: '2014-03-28',
4   region: 'eu-west-1', // replace with your region
5 }
6 const cloudwatchlogs = new AWS.CloudWatchLogs(cloudconfig)
7 exports.handler = async (event, context) => {
8   const params = {
9     destination: 'mm-cloudwatchlog-05292022', // replace with your bucket name
10    from: new Date().getTime() - 8640000,
11    logGroupName: 'access_log',
12    to: new Date().getTime(),
13    destinationPrefix: 'mm'
14  };
15  await cloudwatchlogs.createExportTask(params).promise().then((data) => {
16    console.log(data)
17    return ({
18      statusCode: 200,
19      body: data,
20    });
21  }).catch((err) => {
22    console.error(err)
23    return ({
24      statusCode: 501,
25      body: err,
26    });
27  });
28 }
```

The bottom section shows the **Configuration** tab of the Lambda function. Under the **Triggers** section, there is one trigger named **EventBridge (CloudWatch Events): mm-cwl-s3-export**. The trigger details are as follows:

- Description:** mm-cwl-s3-export
- Event bus:** default
- Schedule expression:** rate(1 hour)