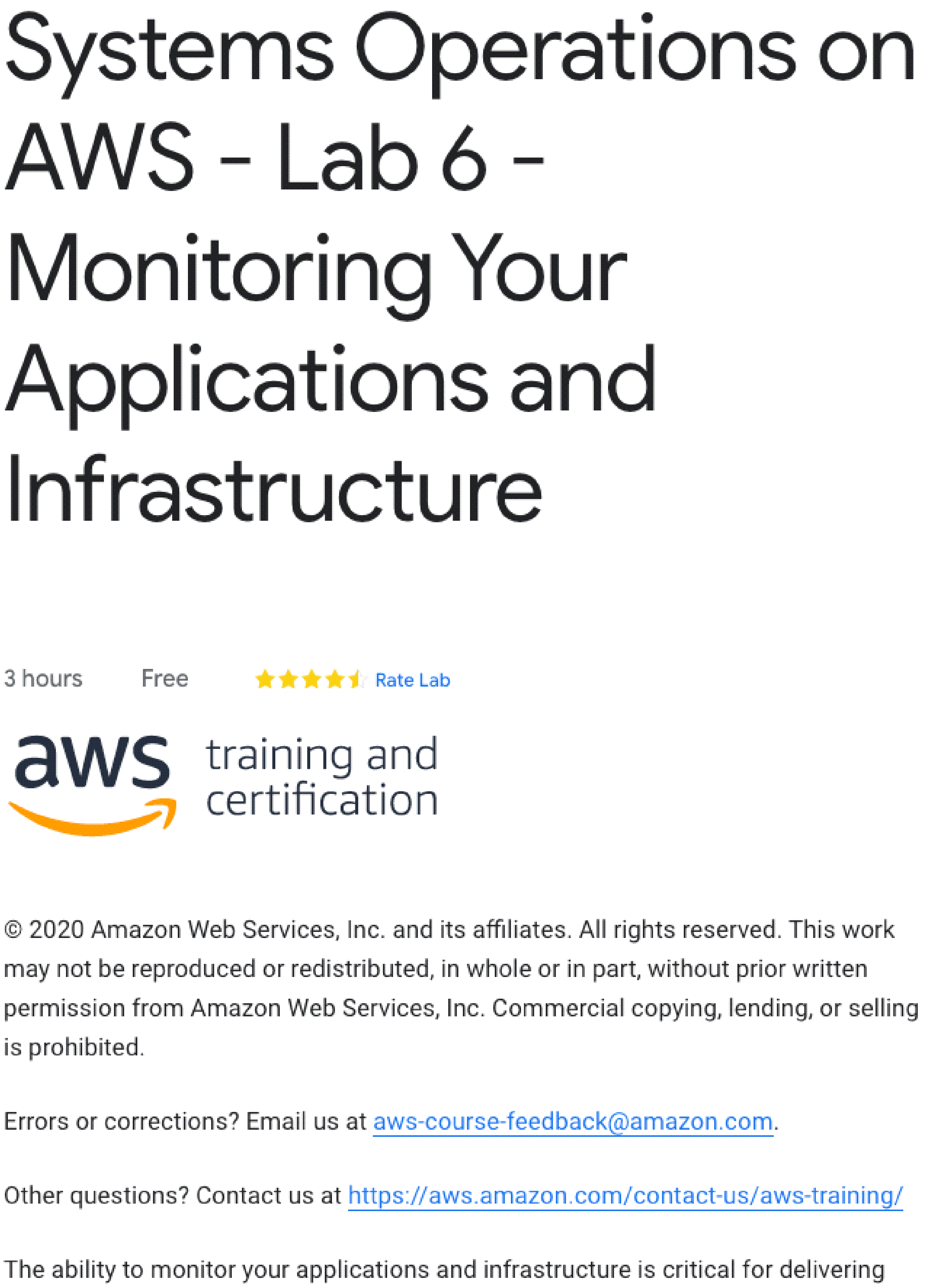
Lab 6 - Monitoring Your Applications and Infrastructure

Wednesday, April 22, 2020 8:31 AM



Systems Operations on AWS - Lab 6 - Monitoring Your Applications and Infrastructure

3 hours Free ★★★★½ Rate Lab

aws training and certification

Errors or corrections? Email us at aws-course-feedback@amazon.com.

Other questions? Contact us at https://aws.amazon.com/contact-us/aws-training/

The ability to monitor your applications and infrastructure is critical for delivering

The ability to monitor your applications and infrastructure is critical for delivering reliable, consistent IT services.

Monitoring requirements range from collecting statistics for long-term analysis through to quickly reacting to changes and outages. Monitoring can also support compliance reporting by continuously checking that infrastructure is meeting organizational standards.

This lab shows you how to use Amazon CloudWatch Metrics, Amazon CloudWatch Logs, Amazon CloudWatch Events and AWS Config to monitor your applications and infrastructure.

The lab will demonstrate how to:

- Use AWS Systems Manager Run Command to install the **CloudWatch Agent** on Amazon EC2 instances
- Monitor Application Logs using CloudWatch Agent and **CloudWatch Logs**
- Monitor system metrics using CloudWatch Agent and **CloudWatch Metrics**
- Create real-time notifications using **CloudWatch Events**
- Track infrastructure compliance using **AWS Config**

Duration

This lab will require approximately **40 minutes** to complete.

Accessing the AWS Management Console

Start Lab

1. At the top of your screen, launch your lab by clicking **Start Lab**

   This will start the process of provisioning your lab resources. An estimated amount of time to provision your lab resources will be displayed. You must wait for your resources to be provisioned before continuing.

   ℹ If you are prompted for a token, use the one distributed to you (or credits you have purchased).

2. Open your lab by clicking **Open Console**
   This will open an AWS Management Console sign-in page.

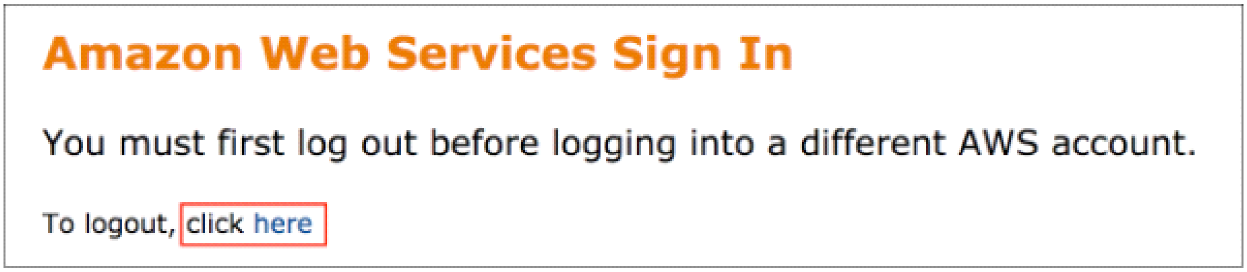
3. On the Sign-in page, configure:

   - **IAM user name:** `awsstudent`
   - **Password:** Paste the value of **Password** located to the left of these instructions.
   - Click **Sign In**

   ⚠ **Please do not change the Region unless instructed**.

Common login errors

Error: You must first log out



Amazon Web Services Sign In

You must first log out before logging into a different AWS account.

To logout, click here

If you see the message, **You must first log out before logging into a different AWS account:**

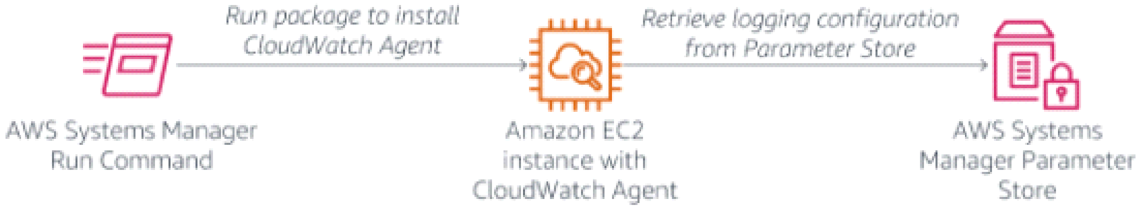
- Click **click here**
- Close your browser tab to return to your initial Qwiklabs window
- Click Open Console again

Task 1: Install the CloudWatch Agent

The **CloudWatch Agent** can be used to collect metrics from Amazon EC2 instances an on-premises servers including:

- **System-level metrics from Amazon EC2 instances**, such as: CPU allocation, free disk space and memory utilization. These metrics are collected from the machine itself and compliment the standard Amazon CloudWatch metrics collected by CloudWatch.
- **System-level metrics from on-premises servers**, enabling monitoring of hybrid environments and servers not managed by AWS.
- **System and Application Logs** from both Linux and Windows servers.
- **Custom metrics** from applications and services using the StatsD and collectd protocols.

In this task, you will use AWS Systems Manager to install the CloudWatch Agent on an Amazon EC2 instance. You will configure it to collect both application and system metrics.


Run package to install CloudWatch Agent
Retrieve logging configuration from Parameter Store
AWS Systems Manager Run Command
Amazon EC2 instance with CloudWatch Agent
AWS Systems Manager Parameter Store

4. In the **AWS Management Console**, on the **Services ▾** menu, click **Systems Manager**.

5. In the left navigation pane, click **Run Command**.

   💬 If there is no visible navigation pane, click the ≡ icon in the top-left corner to make it appear.

   You will use the Run Command to deploy a pre-written command that installs the CloudWatch Agent.

6. Click **Run a Command**

7. Select ◉ **AWS-ConfigureAWSPackage** (typically appears towards the top of the list).

8. In the **Command parameters** section, configure:

   - **Action:** *Install*
   - **Name:** `AmazonCloudWatchAgent`

9. In the **Targets** section:

   - Select ◉ **Choose instances manually**.
   - Select ☑ **Web Server**.

   This configuration will install the CloudWatch Agent on the *Web Server*.

10. At the bottom of the page, click **Run**

10. At the bottom of the page, click **Run**

11. Wait for the **Overall status** to change to *Success*. You can occasionally click ↻ refresh in the top right to update the status.

    You can view the output from the job to confirm that it ran successfully.

12. Under **Targets and outputs**, click instance name displayed under **Instance ID**

13. Expand ▸ **Step 1 - Output**.

    You should see the message: *Successfully installed arn:aws:ssm:::package/AmazonCloudWatchAgent*

    You will now configure CloudWatch Agent to collect the desired log information. The instance has a web server installed, so you will configure CloudWatch Agent to collect the web server logs and also general system metrics.

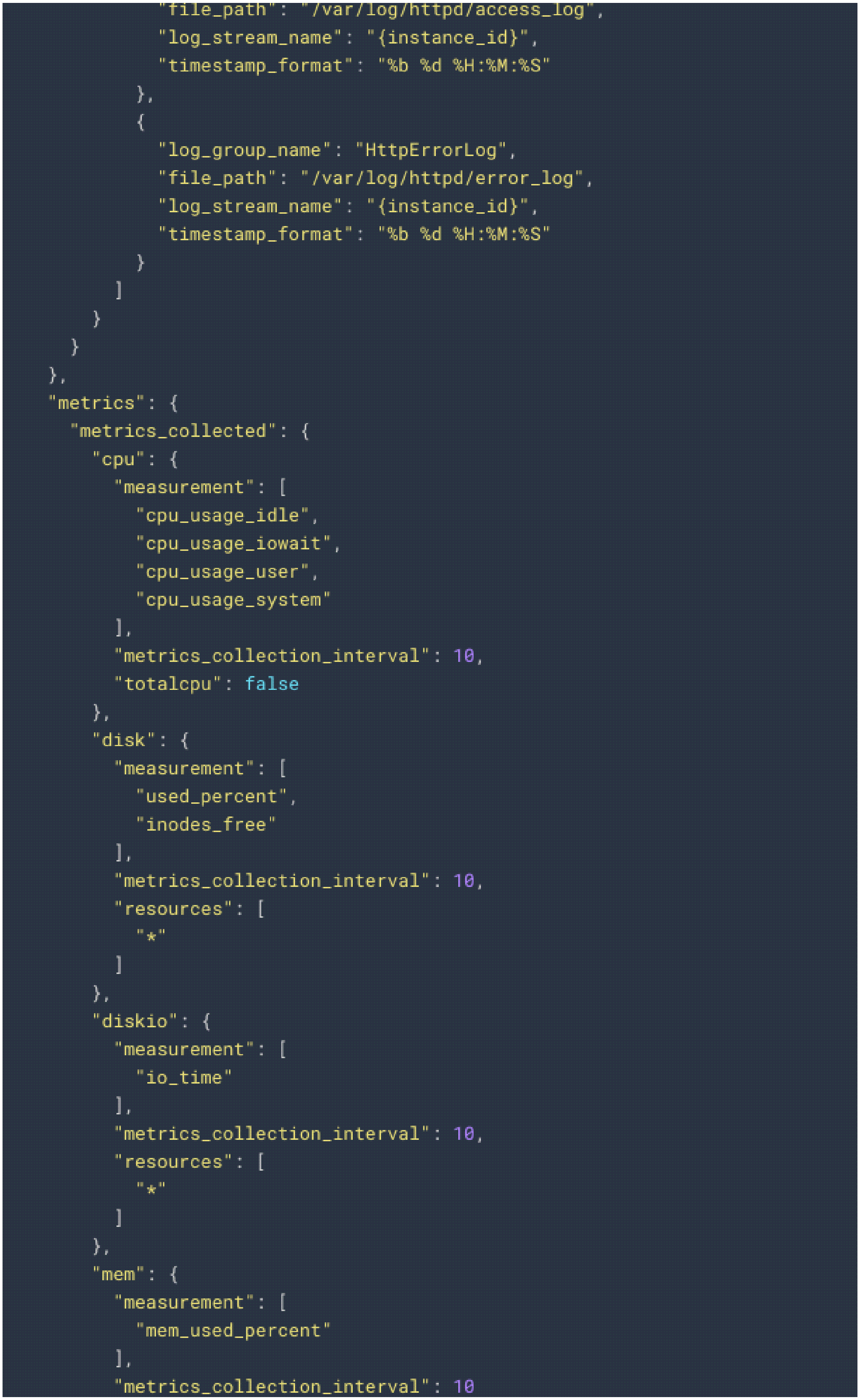
    You will store the configuration file in AWS Systems Manager Parameter Store, which can then be fetched by the CloudWatch Agent.

14. In the left navigation pane, click **Parameter Store**.

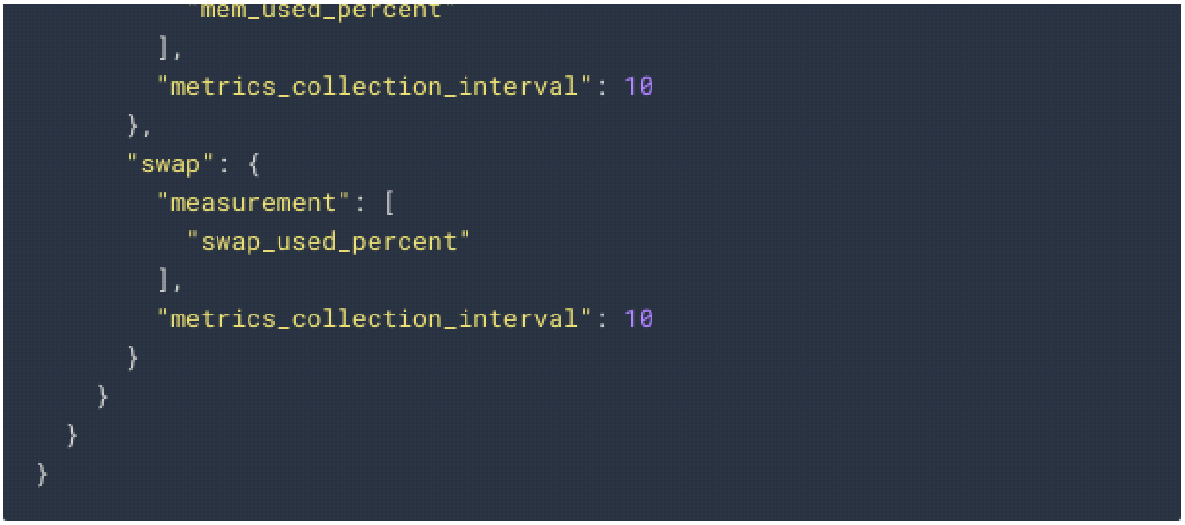
15. Click **Create parameter** then configure:

    - **Name:** `Monitor-Web-Server`
    - **Description:** `Collect web logs and system metrics`
    - **Value:** Paste the configuration shown below:

```
{
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "log_group_name": "HttpAccessLog",
            "file_path": "/var/log/httpd/access_log",
            "log_stream_name": "{instance_id}",
            "timestamp_format": "%b %d %H:%M:%S"
```



```
            "file_path": "/var/log/httpd/access_log",
            "log_stream_name": "{instance_id}",
            "timestamp_format": "%b %d %H:%M:%S"
          },
          {
            "log_group_name": "HttpErrorLog",
            "file_path": "/var/log/httpd/error_log",
            "log_stream_name": "{instance_id}",
            "timestamp_format": "%b %d %H:%M:%S"
          }
        ]
      }
    }
  },
  "metrics": {
    "metrics_collected": {
      "cpu": {
        "measurement": [
          "cpu_usage_idle",
          "cpu_usage_iowait",
          "cpu_usage_user",
          "cpu_usage_system"
        ],
        "metrics_collection_interval": 10,
        "totalcpu": false
      },
      "disk": {
        "measurement": [
          "used_percent",
          "inodes_free"
        ],
        "metrics_collection_interval": 10,
        "resources": [
          "*"
        ]
      },
      "diskio": {
        "measurement": [
          "io_time"
        ],
        "metrics_collection_interval": 10,
        "resources": [
          "*"
        ]
      },
      "mem": {
        "measurement": [
          "mem_used_percent"
        ],
        "metrics_collection_interval": 10
```



```
          "mem_used_percent"
        ],
        "metrics_collection_interval": 10
      },
      "swap": {
        "measurement": [
          "swap_used_percent"
        ],
        "metrics_collection_interval": 10
      }
    }
  }
}
```

Examine the above configuration. It defines the following items to be monitored:

- **Logs:** Two web server log files to be collected and sent to Amazon CloudWatch Logs
- **Metrics:** CPU, disk and memory metrics to send to Amazon CloudWatch Metrics

16. Click **Create parameter**

This parameter will be referenced when starting the CloudWatch Agent.

You will now use another *Run Command* to start the CloudWatch Agent on the Web Server.

17. In the left navigation pane, click **Run Command**.

18. Click **Run command**

19. Click 🔍 then:

- *Document name prefix*
- *Equal*
- `AmazonCloudWatch-ManageAgent`
- Press Enter

Before running the command, you can view the definition of the command.

20. Click **AmazonCloudWatch-ManageAgent** (click on the name itself).

20. Click **AmazonCloudWatch-ManageAgent** (click on the name itself).

    A new web browser tab will open, showing the definition of the command.

    Browse through the content of each tab to see how a Command Document is defined.

21. Click the **Content** tab and scroll to the bottom to see the actual script that will run on the target instance.

    The script references the AWS Systems Manager Parameter Store because it will retrieve the CloudWatch Agent configuration you defined earlier.

22. Close the current web browser tab, which should return you to the *Run a command* tab you were using earlier.

23. Select ⊙ **AmazonCloudWatch-ManageAgent** (click the circle).

24. In the **Command parameters** section, configure:

    - **Action:** *configure*
    - **Mode:** *ec2*
    - **Optional Configuration Source:** *ssm*
    - **Optional Configuration Location:** `Monitor-Web-Server`
    - **Optional Restart:** *yes*

    This configures the Agent to use the configuration you previously stored in the Parameter Store.

25. In the **Targets** section:

    - Select ⊙ **Choose instances manually**.
    - Select ☑ **Web Server**.

26. Click **Run**

27. Wait for the **Overall status** to change to *Success*. You can occasionally click ↻ refresh in the top right to update the status.

27. Wait for the **Overall status** to change to *Success*. You can occasionally click ↻ refresh in the top right to update the status.

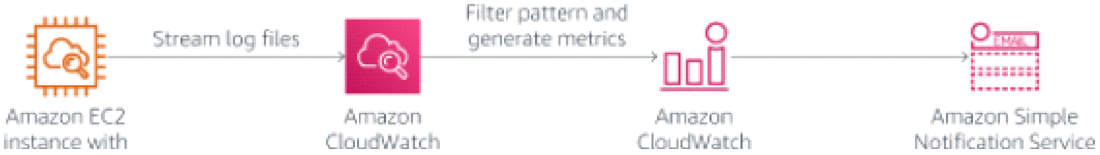
    CloudWatch Agent is now running on the instance, sending log and metric data to Amazon CloudWatch.

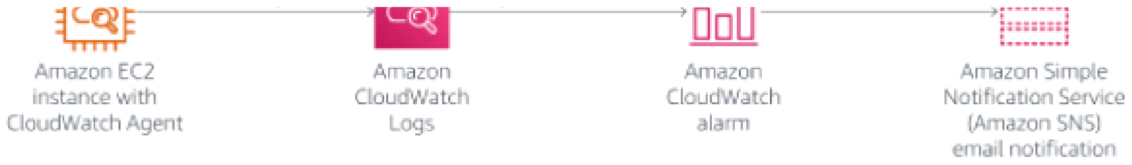
Task 2: Monitor Application Logs using CloudWatch Logs

You can use **Amazon CloudWatch Logs** to monitor applications and systems using **log data**. For example, CloudWatch Logs can track the number of errors that occur in your application logs and send you a notification whenever the rate of errors exceeds a threshold you specify.

CloudWatch Logs uses your existing log data for monitoring; so, no code changes are required. For example, you can monitor application logs for specific literal terms (such as "NullReferenceException") or count the number of occurrences of a literal term at a particular position in log data (such as "404" status codes in web server access log). When the term you are searching for is found, CloudWatch Logs reports the data to a CloudWatch metric that you specify. Log data is encrypted while in transit and while it is at rest.

In this task you will generate log data on the Web Server, then monitor the logs using CloudWatch Logs.




Amazon EC2 instance with CloudWatch Agent
Amazon CloudWatch Logs
Amazon CloudWatch alarm
Amazon Simple Notification Service (Amazon SNS) email notification

The Web Server generates two types of log data:

- Access Logs
- Error Logs

You will begin by accessing the web server.

28. Copy the **WebServerIP** value shown to the left of these instructions

29. Open a new web browser tab, paste the *WebServerIP* you copied, then press Enter.

    You should see a web server **Test Page**.

    You will now generate log data by attempting to access a page that does not exist.

30. Append `/start` to the browser URL and press Enter.

    You will receive an **error message** because the page is not found. **This is okay!** It will generate data in the access logs that are being sent to CloudWatch Logs.

31. Keep this tab open in your web browser, but return to the browser tab showing the AWS Management Console.

32. On the **Services ⌄** menu, click **CloudWatch**.

33. In the left navigation pane, click **Logs**.

    You should see two logs listed: **HttpAccessLog** and **HttpErrorLog**.

    ⚠ If these logs are not listed, try waiting a minute, then click 🔄 **Refresh**. If the logs still do not appear, request assistance from your instructor.

still do not appear, request assistance from your instructor.

34. Click **HttpAccessLog** (click on the actual name).

35. Click the value displayed under **Logs Streams**.

    Log data should be displayed, consisting of **GET** requests that were sent to the web server. You can view further information by ▸ expanding the lines. The log data includes information about the computer and browser that made the request.

    You should see a line with your **/start** request with a code of 404, which means that the page was not found.

    This demonstrates how log files can be automatically shipped from an Amazon EC2 instance, or an on-premises server, to CloudWatch Logs. The log data is accessible without having to log in to each individual server. Log data can also be collected from multiple servers, such as an Auto Scaling fleet of web servers.

Create a Metric Filter in CloudWatch Logs

You will now configure a Filter to identify *404 Errors* in the log file. This would normally be an indication that the web server is generating invalid links that users are clicking.

36. In the left navigation pane click **Logs**.

37. Select ⊙ **HttpAccessLog** (click the circle, not the link).

38. Click **Create Metric Filter**

    A **filter pattern** defines the fields in the log file and filters the data for specific values.

39. Paste this line into **Filter Pattern:**

    ```
    [ip, id, user, timestamp, request, status_code=404, size]
    ```

```
[ip, id, user, timestamp, request, status_code=404, size]
```

This tells CloudWatch Logs how to interpret the fields in the log data and defines a filter to only find lines with **status_code=404**, which indicates that a page was not found.

40. Click Test Pattern

41. In the **Results** section, click **Show test results**.

    You should see at least one result with a *$status_code* of **404**. This indicates that a page was requested that was not found.

42. Click **Assign Metric**

43. Configure these **Metric Details:**

    - **Metric Namespace:** `LogMetrics`
    - **Metric Name:** `404Errors`

44. Click **Create Filter**

    This metric filter can now be used in an Alarm.

Create an Alarm using the Filter

You will now configure an Alarm to send a notification when too many *404 Not Found* errors are received.

45. Click **Create Alarm** (it is next to ✏ ✖).

46. Configure these settings:

    - **Period:** *1 minute*
    - **Conditions:**
        - Whenever 404Errors is: ● **Greater/Equal**

    - Whenever 404Errors is: ◉ **Greater/Equal**
    - than: `5`

- Click Next

47. For **Notification**, configure:

    - **Select an SNS Topic:** ◉ **Create new topic**
    - **Email endpoints that will receive the notification:** Enter an email address that you can access from the classroom
    - Click Create topic
    - Click Next

48. For **Name and description**, configure:

    - **Alarm name:** `404 Errors`
    - **Alarm description:** `Alert when too many 404s detected on an instance`
    - Click Next

49. Click Create alarm

50. Go to your email, look for a confirmation message and select the **Confirm subscription** link.

51. Return to the **AWS Management Console**.

52. In the left navigation pane, click **CloudWatch** (at the very top).

    Your alarm should appear in orange, indicating that there is *INSUFFICIENT DATA* to trigger the alarm. This is because no data has been received in the past minute.

    You will now access the web server to generate log data.

53. Return to the web browser tab with the web server.

    💬 If the tab is no longer open, copy the *WebServerIP* shown to the left of the instructions you are current reading, and open a new web page with that IP address.

💬 If the tab is no longer open, copy the *WebServerIP* shown to the left of the instructions you are current reading, and open a new web page with that IP address.

54. Attempt to go to pages that do not exist by adding a page name after the IP address. Repeat this **at least 5 times**.

    For example: *http://54.11.22.33/start2*

    Each separate request will generate a separate log entry.

55. Wait 1-2 minutes for the Alarm to trigger. You can occasionally click ↻ Refresh to update the status.

    The graph shown on the CloudWatch page should turn red to indicate that it is in the *ALARM* state.

56. Check your email. You should have received an email titled *ALARM: "404 Errors"*.

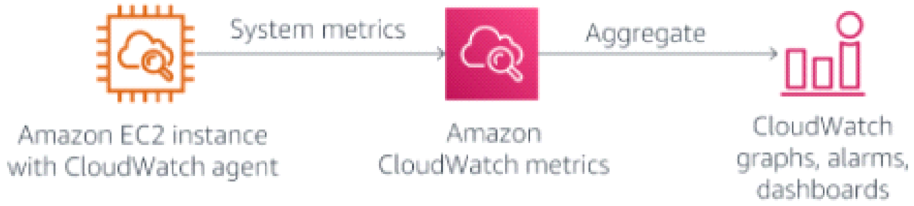
    This demonstrates how you can create an Alarm from application log data and receive alerts when unusual behavior is detected in the log file. The log file is easily accessible within Amazon CloudWatch Logs to perform further analysis to diagnose the activities that led to the Alarm being triggered.

Task 3: Monitor Instance Metrics using CloudWatch

Metrics are data about the performance of your systems. Amazon CloudWatch stores metrics for AWS services you use. You can also publish your own application metrics either via CloudWatch Agent or directly from your application. Amazon CloudWatch can present the metrics for search, graphs, dashboards, and alarms.

In the task, you will use explore metrics provided by Amazon CloudWatch.

In the task, you will use explore metrics provided by Amazon CloudWatch.


System metrics
Aggregate
Amazon EC2 instance with CloudWatch agent
Amazon CloudWatch metrics
CloudWatch graphs, alarms, dashboards

57. On the **Services ⌄** menu, click **EC2**.

58. In the left navigation pane, click **Instances**.

59. Select ☑ **Web Server**.

60. Click the **Monitoring** tag in the lower half of the page.

    Examine the metrics presented. You can also **click on a chart** to display more information.

    CloudWatch captures metrics about CPU, Disk and Network usage on the instance. These metrics view the instance 'from the outside' as a *virtual machine* but do not give insight into what is running 'inside' the instance, such as measuring free memory or free disk space. Fortunately, you can obtain information about what is happening *inside* the instance by using information captured by **CloudWatch Agent**, because CloudWatch Agent runs *inside* the instance to collect metrics.

61. Click **View all CloudWatch metrics** (displayed just above the charts).

    This will return you to the CloudWatch console.

    The lower half of the page will display the various metrics that have been collected

This will return you to the CloudWatch console.

The lower half of the page will display the various metrics that have been collected by CloudWatch. Some are automatically generated by AWS while others were collected by the CloudWatch Agent.

62. Click **CWAgent**, then **device, fstype, host, path**.

    You will see the disk space metrics being captures by CloudWatch Agent.

63. Click **CWAgent** (in the line that says *All > CWAgent > device, fstype, host, path*).
64. Click **host**.

    You will see metrics relating to system memory.

65. Click **All** (in the line that says *All > CWAgent > device, fstype, host, path*).

    Explore the other metrics that are being captured by CloudWatch. These are automatically-generated metrics coming from the AWS services that have been used in this AWS account.

    You can ☑ select metrics that you wish to appear on the graph.

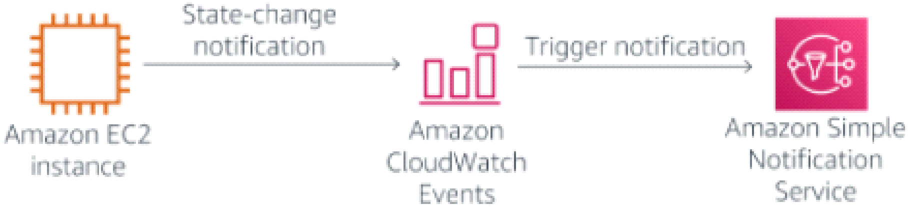
Task 4: Create Real-Time Notifications

Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in AWS resources. Simple rules can match events and route them to one or more target functions or streams. CloudWatch Events becomes aware of operational changes *as they occur*.

CloudWatch Events responds to these operational changes and takes corrective action as necessary, by sending messages to respond to the environment, activating

CloudWatch Events responds to these operational changes and takes corrective action as necessary, by sending messages to respond to the environment, activating functions, making changes, and capturing state information. You can also use CloudWatch Events to schedule automated actions that self-trigger at certain times using cron or rate expressions.

In this task, you will create a real-time notification that informs you when an instance is Stopped or Terminated.



66. In the left navigation pane, click **Rules**.

67. Click **Create rule**

68. In the **Event Source** section, configure:

- **Service Name:** *EC2*
- **Event Type:** *EC2 Instance State-change Notification*
- ⦿ Specific state(s)
- From the drop-down menu, select **stopped** and **terminated**

69. In the **Targets** section on the right, configure:

- ⊕ **Add target**
- Click **Lambda function** and change it to **SNS topic**
- **Topic:** *Default_CloudWatch_Alarms_Topic*

- Click **Lambda function** and change it to **SNS topic**
- **Topic:** *Default_CloudWatch_Alarms_Topic*

  This is the Amazon SNS topic that was created when you configured a CloudWatch Alarm earlier in the lab.

70. Click **Configure details** (at the bottom of the page).

71. In the **Rule definition**, configure:

    - **Name:** `Instance_Stopped_Terminated`
    - Click **Create rule**

Configure Real-Time Notification

In additional to receiving an email, you can configure Amazon Simple Notification Service (SNS) to send you a notification to your phone via SMS.

72. On the **Services ⌄** menu, click **Simple Notification Service**.

73. In the left navigation pane, click **Topics**.

74. Click the link in the **Name** column.

    You should see a single subscription associated with your email address. You will now add an SMS notification.

    💬 If you do not have a phone that can receive SMS messages, you can skip this step.

75. Click **Create subscription** then configure:

    - **Protocol:** *SMS*
    - **Endpoint:** Enter your cell phone number in International format (eg +14155557000 or +917513200000)
    - Click **Create subscription**

You are now ready to trigger a real-time alert!

You are now ready to trigger a real-time alert!

76. On the **Services ⌄** menu, click **EC2**.

77. In the left navigation pane, click **Instances**.

78. Select ☑ **Web Server**.

79. Click **Actions ⌄** then **Instance State > Stop**, then **Yes, Stop**

The *Web Server* instance will enter the *stopping* state. After a minute it will enter the *stopped* state.

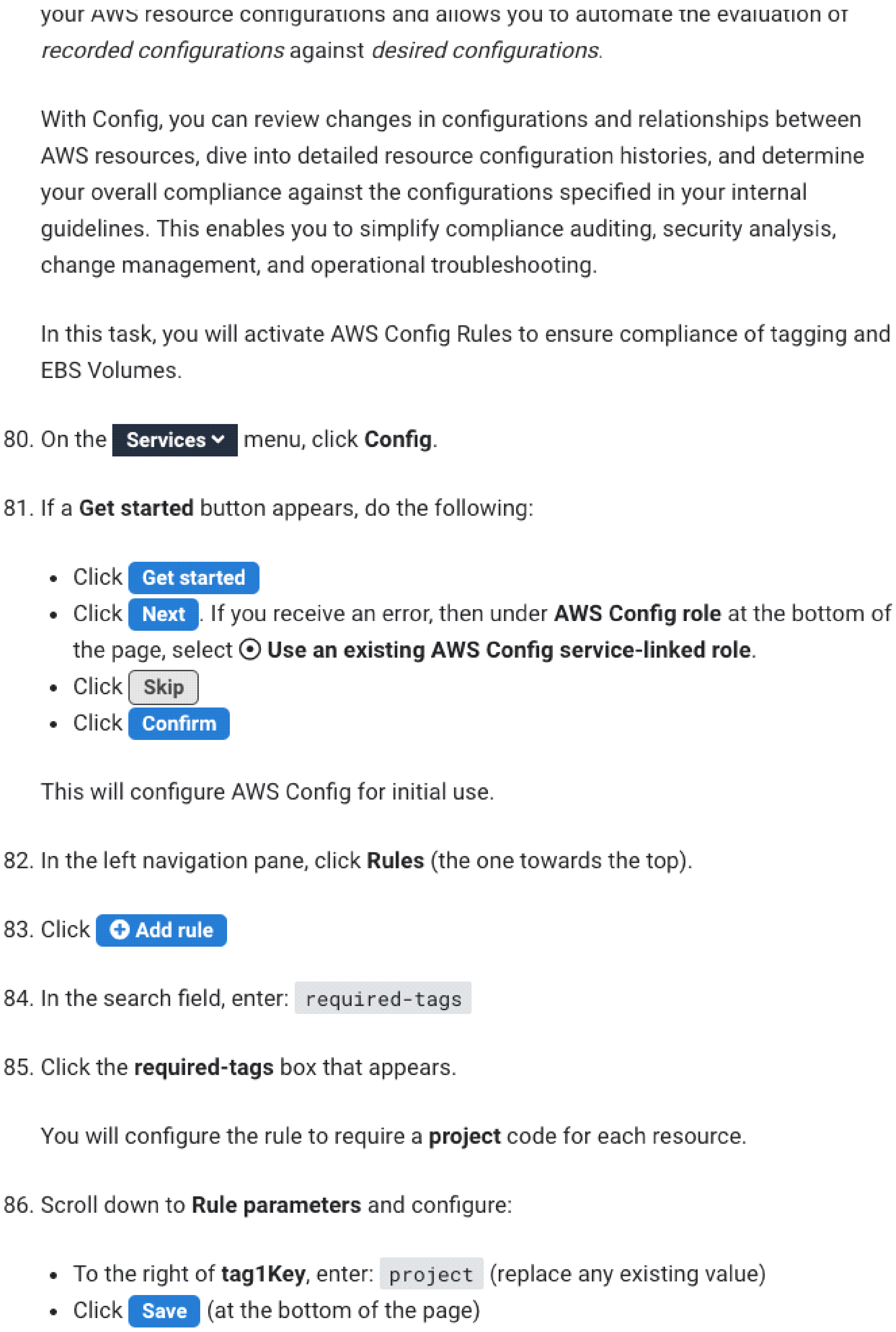
You should then receive an SMS message with details of the instance that was stopped.

The message is formatted in JSON. To receive an easier-to-read message, you could create an AWS Lambda function that is triggered by CloudWatch Events. The Lambda function could then format a more friendly message and send it via Amazon SNS.

This demonstrates how easy it is to receive real-time notifications when infrastructure changes.

Task 5: Monitor for Infrastructure Compliance

AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of *recorded configurations* against *desired configurations*.



your AWS resource configurations and allows you to automate the evaluation of *recorded configurations* against *desired configurations*.

With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. This enables you to simplify compliance auditing, security analysis, change management, and operational troubleshooting.

In this task, you will activate AWS Config Rules to ensure compliance of tagging and EBS Volumes.

80. On the **Services ⌄** menu, click **Config**.

81. If a **Get started** button appears, do the following:

    - Click **Get started**
    - Click **Next**. If you receive an error, then under **AWS Config role** at the bottom of the page, select ⦿ **Use an existing AWS Config service-linked role**.
    - Click **Skip**
    - Click **Confirm**

    This will configure AWS Config for initial use.

82. In the left navigation pane, click **Rules** (the one towards the top).

83. Click **⊕ Add rule**

84. In the search field, enter: `required-tags`

85. Click the **required-tags** box that appears.

    You will configure the rule to require a **project** code for each resource.

86. Scroll down to **Rule parameters** and configure:

    - To the right of **tag1Key**, enter: `project` (replace any existing value)
    - Click **Save** (at the bottom of the page)

- Click **Save** (at the bottom of the page)

This rule will now look for resources that do not have a *project* tag. This will take a few minutes to complete, so please continue with the next steps. There is no need to wait.

You will now add a rule that looks for Amazon EBS volumes that are not attached to Amazon EC2 instances.

87. Click ⊕ **Add rule**

88. In the search field, enter: `ec2-volume-inuse-check`

89. Click the **ec2-volume-inuse-check** box that appears.

90. Click **Save**

91. Wait until at least one of the rules has completed evaluation. Click **Refresh** ↻ in the top-right every 60 seconds to update the status.

    💬 If you receive a message that there are *No resources in scope*, please wait a few minutes longer. This message is an indication that AWS Config is still scanning available resources. The message will eventually disappear.

92. Click each of the rules to view the result of the audits.

    Amongst the results should be:

    - **required-tags:** A compliant EC2 Instance (because the Web Server has a *project* tag) and many non-compliant resources that do not have a *project* tag
    - **ec2-volume-inuse-check:** One compliant volume (attached to an instance) and one non-compliant volume (*not* attached to an instance)

    AWS Config has a large library of pre-defined compliance checks and you can create additional checks by writing your own AWS Config Rule using AWS Lambda.

Lab Complete

Congratulations! You have completed the lab.

The lab has demonstrated how to:

- Use AWS Systems Manager Run Command to install the **CloudWatch Agent** on Amazon EC2 instances
- Monitor Application Logs using CloudWatch Agent and **CloudWatch Logs**
- Monitor system metrics using CloudWatch Agent and **CloudWatch Metrics**
- Create real-time notifications using **CloudWatch Events**
- Track infrastructure compliance using **AWS Config**

End Lab

Follow these steps to close the console, end your lab, and evaluate the experience.

93. Return to the AWS Management Console.

94. On the navigation bar, click **awsstudent@<AccountNumber>**, and then click **Sign Out**.

95. Click **End Lab**

96. Click **OK**

97. (Optional):

97. (Optional):

- Select the applicable number of stars ☆
- Type a comment
- Click **Submit**

    - 1 star = Very dissatisfied
    - 2 stars = Dissatisfied
    - 3 stars = Neutral
    - 4 stars = Satisfied
    - 5 stars = Very satisfied

You may close the dialog if you don't want to provide feedback.