Security Monitoring

**SPL-TF-200-SISBPL-10-EN - Version 1.0.7**

© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Overview**

As a security engineer at AnyCompany, you are responsible for monitoring the company network and Amazon Elastic Compute Cloud (Amazon EC2) instances for abnormal activity.

In this lab, you configure an Amazon Linux 2 instance to send log files to Amazon CloudWatch. You then create Amazon CloudWatch alarms and notifications to alert you to a specified number of login failures on your EC2 instances. Finally, you create a CloudWatch alarm and notification to monitor outgoing traffic through a NAT gateway.

OBJECTIVES

By the end of this lab, you will be able to:

* Configure an Amazon Linux 2 instance to send log files to Amazon CloudWatch
* Create Amazon CloudWatch alarms and notifications to monitor for failed login attempts
* Create Amazon CloudWatch alarms to monitor network traffic through a NAT gateway

TECHNICAL KNOWLEDGE PREREQUISITES

To successfully complete this lab, you should be familiar with basic navigation of the AWS Management Console and be comfortable running commands in a Linux command line interface (CLI).

DURATION

This lab requires approximately *45* minutes to complete.

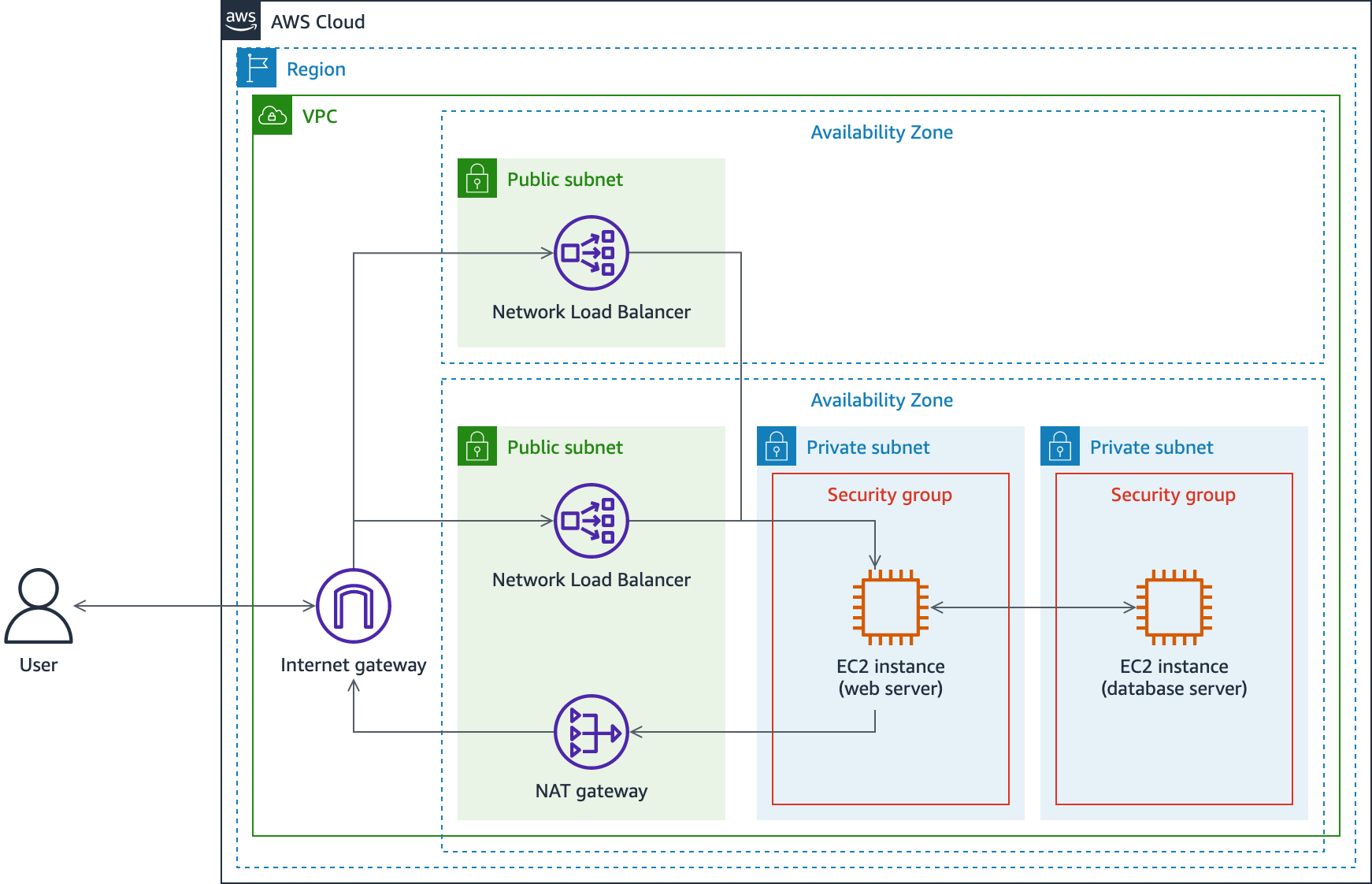
ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* A command that you must run
* A sample output that you can use to verify the output of a command or edited file
* A hint, tip, or important guidance
* Where to find more information
* A moment to pause to consider how you might apply a concept in your own environment or to initiate a conversation about the topic at hand
* A time when you might need to refresh a web browser page or list to show new information

ENVIRONMENT OVERVIEW

The following diagram shows the basic architecture of the lab environment:



The following list details the major resources in the diagram:

* A **VPC** with one **public subnet** and two **private subnets** in one Availability Zone, and one **public subnet** in a second Availability Zone.
* A **Network Load Balancer** with two nodes, one in each public subnet.
* An **EC2 instance** acting as a web server in the first private subnet.
* An **EC2 instance** acting as a database server in the second subnet.
* Two **security groups**, one for each instance based on its purpose.

The network traffic flows from an external user, through an internet gateway to one of the two Network Load Balancer nodes, to the web server. If the URL of the WordPress blog site running on the web server is requested, traffic flows to the database server as well.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

**Caution:** You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**WARNING:** **Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



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

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

**Task 1: Send local logs from an EC2 instance to Cloudwatch**

In this task, you install the Amazon CloudWatch agent on an Amazon Linux 2 EC2 instance. The instance serves as the database server for your customer facing blog site and contains sensitive information, so you would like to easily monitor any abnormalities.

TASK 1.1: INSTALL THE CLOUDWATCH AGENT

1. At the top of the **AWS Management Console** page, in the unified search bar, search for and choose

EC2

.

If you do not have the AWS Management Console tab open, follow the steps in the [Start Lab](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-200-SISBPL-1%3A1.0.7-51ad30a8/en-US#StartLab) section to log into the AWS Management Console.

1. In the navigation pane at the left of the page, choose **Instances**.
2. Select **Database Server** and then, at the upper-right of the page, choose **Connect**.
3. On the **Connect to instance** page, choose the **Session Manager** tab, and then choose **Connect**.

 A new web browser tab opens with a console connection to the instance. A set of commands are run automatically when you connect to the instance that change to the user’s home directory and display the path of the working directory, similar to this:

cd HOME; pwd

sh-4.2$ cd HOME; pwd

/home/ec2-user

sh-4.2$

First, install the Amazon CloudWatch agent package.

1. Run the following command to install the **CloudWatch agent** package:

 On a Windows-based computer, you might need to use **Ctrl + Shift + V** or open the context menu (right-click) to paste text into a Session Manager console window.

sudo yum install -y amazon-cloudwatch-agent

1. Run the following command to start the CloudWatch agent configuration wizard:

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

 The wizards opens to the following menu:

================================================================

= 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. Enter

1

 for **linux**.

1. For **Are you using EC2 or On-Premises hosts?**, enter

1

 for **EC2**.

1. For **Which user are you planning to run the agent?**, enter

2

 for **root**.

1. For **Do you want to turn on StatsD daemon?**, enter

2

 for **no**.

1. For **Do you want to monitor metrics from CollectD?**, enter

2

 for **no**.

1. For **Do you want to monitor any host metrics?**, enter

2

 for **no**.

1. For **Do you have any existing CloudWatch Log Agent configuration file to import for migration?**, enter

2

 for **no**.

1. For **Do you want to monitor any log files?**, enter

1

 for **yes**.

1. For **Log file path:**, enter

/var/log/secure

.

1. For **Log group name:**, enter

database\_server\_security\_logs

.

1. For **Log group class:** press **Enter** to keep the default value of **1.**
2. For **Log stream name:**, press **Enter** to keep the default value of **{instance\_id}**.
3. For **Log Group Retention in days**, press **Enter** to keep the default value of **1.**
4. For **Do you want to specify any additional log files to monitor?**, enter

2

 for **no**.

1. For **Do you want the CloudWatch agent to also retrieve X-ray traces?**, enter

2

 for **no**.

 If you need to modify the CloudWatch agent configuration in the future, you can find it at */opt/aws/amazon-cloudwatch-agent/bin/config.json*.

1. For **Do you want to store the config in the SSM parameter store?**, enter

2

 for **no**.

 You should receive a **Program exits now** message and return to the shell prompt.

1. Run the following command to start the CloudWatch agent service:

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json

 Notice the file path to the *config.json* file at the end of the command, which is the CloudWatch agent configuration file that you created with the CloudWatch agent wizard.

 The output should display information about the configuration file that is loaded by the agent…similar to this:

\*\*\*\*\*\* 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

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

2022/02/14 16:48:34 Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file\_config.json.tmp ...

Valid Json input schema.

I! Detecting run\_as\_user...

No csm configuration found.

No metric 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

1. Run the following command to verify the status of the CloudWatch agent service:

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

 The output should show that the CloudWatch agent is running, similar to this:

{

"status": "running",

"starttime": "2022-02-13T04:10:07+0000",

"configstatus": "configured",

"cwoc\_status": "stopped",

"cwoc\_starttime": "",

"cwoc\_configstatus": "not configured",

"version": "1.247347.4"

}

TASK 1.2: GENERATE AUTHENTICATION FAILURES IN THE SECURITY LOGS

1. Run the following command to attempt to log in as **dbdev**:

su dbdev

1. When prompted for a password, be creative and try to guess what the password might be! Remember, you are trying to generate authentication failures, so failed logins are what you want.
2. Repeat the previous two steps three more times to generate a total of 4 failed login attempts.

TASK 1.3: VERIFY THE LOG FILES ARE SENT TO CLOUDWATCH

1. Return to your web browser tab with the **EC2 management console**.
2. At the top of the page, in the unified search bar, search for and choose

CloudWatch

.

1. In the navigation pane at the left of the page, under **Logs**, choose **Log groups**.
2. On the **Log groups** page, select the link for the **database\_server\_security\_logs** log group.

 If the *database\_server\_security\_logs* is not listed, wait 1 minute, and then choose the  refresh button at the upper-right of the page.

1. On the **database\_server\_security\_logs** page, choose the **Log streams** tab.

In the **Log streams** section, notice there is one log stream that is named with the EC2 instance ID of the database server instance.

1. Choose the link for the log stream name to view the security logs from the database server instance.

 Congratulations! You have successfully installed the CloudWatch agent on an EC2 instance and configured it to send security logs to CloudWatch.

**Task 2: Create a CloudWatch alarm and notification**

Now that you are sending the security logs from the database server to CloudWatch, you can create CloudWatch metrics and alarms to monitor the logs for any abnormal behavior.

In this task, you create a CloudWatch metric filter to locate authentication failure event messages in the logs from the database server. You then create a CloudWatch alarm and notification to email you any time there are too many login failures within a 5 minute window.

TASK 2.1: CREATE A METRIC FILTER

1. In the navigation breadcrumbs at the top of the page, select the **database\_server\_security\_logs** link.
2. On the **database\_server\_security\_logs** page, choose the **Metric filters** tab.
3. At the upper-right of the **Metric filters** section, choose **Create metric filter**.
4. On the **Define pattern** page:

* In the **Create filter pattern** section, for **Filter pattern**, enter

"authentication failure"

.

* In the **Test pattern** section, for **Select log data to test**, select the EC2 instance ID for the database server.
  + The EC2 instance ID should match the value of **DatabaseServerId** listed to the left of these instructions.

1. Choose **Test pattern**.

 Under **Results**, you should find a number of *authentication failure* log message that matches the number of times you failed to log in as *dbdev* on the Database Server instance. Only the last 50 log messages are used as a sample to test your filter pattern with. If you don’t see any results during your pattern test, you can try generating authentication failures again.

1. At the bottom of the page, choose **Next**.
2. On the **Assign metric** page:

* In the **Create filter name** section, for **Filter name**, enter

database server authentication failures

.

* In the **Metric details** section, for **Metric namespace**, enter

authentication failures

 and verify the **Create new** option is toggled **on** .

* For **Metric name**, enter

database server authentication failures

.

* For **Metric value**, enter

1

.

* For **Default value**, enter

0

.

1. At the bottom of the page, choose **Next**.
2. On the **Review and create** page, choose **Create metric filter**.

 You should notice a  Metric filter “database server authentication failures” has been created. message at the top of the page.

TASK 2.2: CREATE A CLOUDWATCH ALARM FROM A METRIC FILTER

1. On the **database\_server\_security\_logs** page, on the **Metric filters** tab, you should see a card for **database server authentication failures** filter.
2. Select the checkbox  at the upper-right corner of the **database server authentication failures** card.
3. Choose **Create alarm**.

A new web browser tab opens to the *Specify metrics and conditions* page.

1. On the **Specify metrics and conditions** page, in the **Metric** section, verify the following defaults:

* For **Metric name**, enter

database server authentication failures

.

* For **Statistic**, select **Sum**.
* For **Period**, select **5 minutes**.

1. In the **Conditions** section:

* For **Threshold type**, select **Static**.
* For **Whenever database server authentication failures is…**, select **Greater**.
* For **than…**, enter

2

.

1. Expand the **Additional configuration** section, and then:

* For **Missing data treatment**, select **Treat missing data as good (not breaching threshold)**.

1. At the bottom of the page, choose **Next**.
2. On the **Configure actions** page, in the **Notification** section:

* For **Alarm state trigger**, select **In alarm**.
* For **Send a notification to the following SNS topic**, select  **Create new topic**.
* For **Create a new topic…**, enter

CloudWatch\_alarm\_notifications

.

* For **Email endpoints that will receive the notification…**, enter your email address.

1. Choose **Create topic**.
2. At the bottom of the page, choose **Next**.
3. On the **Add name and description** page:

* For **Alarm name**, enter

database server authentication failures alarm

.

* For **Alarm description**, enter

Alarms and notifies for more than 2 authentication failures over a span of 5 minutes

.

1. Choose **Next**.
2. At the bottom of the **Preview and create** page, choose **Create alarm**.

 You should notice two banners at the top of the page:

**Successfully created alarm database server authentication failures alarm.**

**Some subscriptions are pending confirmation**

You must confirm your email address before Amazon SNS can send notification messages to you.

1. Check your email for a message from **AWS Notifications** with a subject line of **AWS Notification - Subscription Confirmation**.

The message should look similar to this:

You have chosen to subscribe to the topic: **arn:aws:sns:us-west-2:111122223333:CloudWatch\_alarm\_notifications**

To confirm this subscription, click or visit the link below (If this was in error no action is necessary): Confirm subscription

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to sns-opt-out

1. Select the **Confirm subscription** link.

 A new web browser tab should open to a **Subscription confirmed!** page.

1. Close your web browser tab with the **Subscription confirmed!** page.
2. In the **Some subscriptions are pending confirmation** banner at top of the page, choose **View SNS Subscriptions**.
3. On the **Subscriptions** page, verify that the status for the **CloudWatch\_alarm\_notifications** topic is  Confirmed.
4. Close your web browser tab with the **Subscriptions** page.

TASK 2.3: VERIFY THE ALARM IS ACTIVATED

1. In the **CloudWatch** navigation pane at the left of the page, under **Alarms**, choose **All alarms**.
2. Select the **database server authentication failures alarm** link to view the alarm details.
3. Scroll down the page to the **Details** tab.

 Is the alarm **State** currently **In alarm** or **OK**? What do you think has caused the current state of the alarm?

When you configured the alarm, you set it to alert for two or more authentication failures in a five minute time span. If that situation occurs, the alarm state changes to **In alarm** until there is another time span of five or more minutes where there are no authentication failures, at which point the alarm state changes back to **OK**.

1. If the alarm state is  OK, return to the **Database Server** console session and attempt to log in as **dbdev** again.

Refer to Task 1.2 for the commands to attempt to log in as dbdev, if necessary.

1. Return to your web browser tab with the **database server authentication failures alarm** and refresh the page.

 It can take approximately 2 minutes for the alarm state to change to *In alarm*.

The alarm status should now be  In alarm and you should receive an email notification regarding the state change.

 Congratulations! You have successfully created a CloudWatch alarm and notification to monitor authentication failures on an EC2 instance.

**Task 3: Monitor a NAT gateway**

In addition to monitoring authentication failures, you want to monitor for abnormal outgoing network traffic.

In this task, you create a CloudWatch alarm to monitor metrics related to outgoing traffic through a NAT gateway.

TASK 3.1: CREATE A CLOUDWATCH ALARM TO MONITOR OUTGOING NETWORK TRAFFIC THROUGH THE NAT GATEWAY

1. In the **CloudWatch** navigation pane at the left of the page, under **Alarms**, choose **All alarms**.
2. At the upper-right of the **Alarms** page, choose **Create alarm**.
3. On the **Specify metric and conditions** page, choose **Select metric**.
4. In the **Metrics** section, under **AWS namespaces**, choose **NATGateway**.
5. Choose **Nat Gateway Metrics**.
6. In the search bar at the top of the **Metrics** section, search for

BytesOutToDestination

.

The list of metrics should now be filtered to only show the **BytesOutToDestination** metric name.

1. Select the metric with a **NatGatewayId** that matches the **NatGatewayId** value listed to the left of these instructions.
2. At the bottom of the page, choose **Select metric**.
3. On the **Specify metrics and conditions** page, in the **Metric** section:

* For **Metric name**, enter

BytesOutToDestination

.

* For **NatGatewayId**, verify it matches the **NatGatewayId** value listed to the left of these instructions.
* For **Statistic**, select **Sum**.
* For **Period**, select **15 minutes**.

1. In the **Conditions** section:

* For **Threshold type**, select **Static**.
* For **Whenever BytesOutToDestination is…**, select **Greater**.
* For **than…**, enter

2000000

.

1. At the bottom of the page, choose **Next**.
2. On the **Configure actions** page, in the **Notification** section:

* For **Alarm state trigger**, select **In alarm**.
* For **Send a notification to the following SNS topic**, select  **Select an existing SNS topic**.
* For **Send a notification to…**, select **CloudWatch\_alarm\_notifications**.

1. At the bottom of the page, choose **Next**.
2. On the **Add name and description** page:

* For **Alarm name**, enter

NAT gateway outgoing traffic alarm

.

* For **Alarm description**, enter

Alarms and notifies for more than 2 MB of outgoing network traffic over 15 minutes

.

1. Choose **Next**.
2. At the bottom of the **Preview and create** page, choose **Create alarm**.

 You should notice a  **Successfully created alarm NAT gateway outgoing traffic alarm.** message at the top of the page.

TASK 3.2: GENERATE ENOUGH OUTGOING TRAFFIC TO ACTIVATE THE ALARM

Next, you upload data to an S3 bucket to generate enough outgoing traffic to trigger the alarm. A sample file named *SampleData.zip* was added to the Database Server instance during the lab environment build process. The file has a size of approximately 2.3 MB.

1. Return to your web browser tab with the console connection to the **Database Server** instance.
2. Run the following command to set an environment variable to hold the name of the S3 bucket:

* Replace the **TEST\_BUCKET** placeholder value with the **TestBucket** value listed to the left of these instructions.

TestBucket=TEST\_BUCKET

1. Run the following command to upload the **SampleData.zip** file to an S3 bucket in the lab environment:

aws s3 cp ~/SampleData.zip s3://$TestBucket/

 The output should display an *upload* message, which indicates the file uploaded to S3 successfully.

1. Return to your web browser tab with the **CloudWatch console**.
2. If you are not on the **Alarms** page already, in the navigation pane at the left of the page, choose **All alarms**.
3. Wait approximately 1-2 minutes and then refresh the page. You should notice that **NAT gateway outgoing traffic alarm** now has a status of  In alarm. You should also receive a notification email from **AWS Notifications** with the details of the alarm status change.

 Congratulations! You have successfully created a CloudWatch alarm to monitor outgoing network traffic through a NAT gateway!

**Conclusion**

 Congratulations! You now have successfully:

* Configured an Amazon Linux 2 instance to send log files to Amazon CloudWatch
* Created Amazon CloudWatch alarms and notifications to monitor for failed login attempts
* Created Amazon CloudWatch alarms to monitor network traffic through a NAT gateway

**End lab**

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

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional resources**

* [Installing the CloudWatch agent using the command line](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/installing-cloudwatch-agent-commandline.html)
* [Create the CloudWatch agent configuration file with the wizard](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/create-cloudwatch-agent-configuration-file-wizard.html)
* [CloudWatch Logs agent reference](https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AgentReference.html)
* [Getting started with CloudWatch Logs](https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/CWL_GettingStarted.html)
* [Creating metrics from log events using filters](https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/MonitoringLogData.html)

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).