Linux Academy

Live! Lab

# Using the CloudWatch Logs Agent, Log Groups, and SNS Notifications

# Contents



## Lab Connection Information

- Labs may take up to five minutes to build
- Access to an AWS Console is provided on the Live! Lab page, along with your login credentials
- Ensure you are using the N. Virginia region
- Labs will automatically end once the alloted amount of time finishes

### *Related Courses*

*AWS Certified DevOps Engineer - Professional*

### *Related Videos*

*CloudWatch Concepts and Terminology*

### *Need Help?*

*Linux Academy Community*

*... and you can always send in a support ticket on our website to talk to an instructor!*

For this lab, we are simulating using the CloudWatch Logs Agent in combination with SNS notification and CloudWatch alarms to alert an on call engineer. We will use nginx logs to trigger the alarm when certain metrics are met.

# CloudWatch Logs Agent

Log in to the AWS Console using the credentials provided on the Live! Lab page. An EC2 instance with the CloudWatch Log Agent has already been created for use in this lab. We want to SSH into this instance. Navigate to the **EC2 Dashboard** and select **Instances**. Choose your instance and copy the public IP address, then open the terminal on your workstation.

```
[user@workstation] ssh linuxacademy@<IPADDRESS>
```

The password for the *linuxacademy* user is *123456*.

Start nginx:

```
[linuxacademy@ip] sudo service nginx start
```

We now need to modify the `/etc/awslogs/awslogs.conf` file. Find the `/var/logs/messages` launch group, and copy in the information for the `access.logs` group:

```
[/var/log/nginx/access.log]
datetime_format = %Y-%m-%d %H:%M:%S
file = /var/log/nginx/access.log
buffer_duration = 5000
log_stream_name = APP_ID {instance_id}
initial_position = end_of_file
log_group_name = /var/log/nginx/access.log
```

Save and exit the file. Restart the awslogs service:

```
[linuxacademy@ip] sudo service awslogs restart
```

# SNS Notifications and CloudWatch Metrics

Return to the AWS Console. Copy the public IP address of the server and visit the address in your web browser to confirm the nginx sample page is present and nginx is working.

We now need to create the SNS topic to alert our engineer. From the AWS Console, navigate to the **Simple Notification Service Dashboard**. Click **Get Started**, **Create Topic**. We set our **Topic name** to *On_Call_Engineer*. Leave **Display name** blank, and press **Create topic**.

We want to alert our engineer via email. Select **Create subscription** and, under **Protocol**, select *Email*. For **Endpoint** enter your own email address. Log in to your email to confirm the subscription via the email sent.

Navigate to the **CloudWatch Dashboard**. Select **Logs** from the left menu to see if the logs group added earlier has initiated. If the */var/log/nginx/access.log* group has not yet been added, wait until it appears. You may need to refresh the page.

Select the group, and click **Create Metric Filter**. Under the **Filter Pattern** text box, click **Show examples** and click the 400 level HTTP response to populate the text box. Press **Test Pattern**. We do not have any matches for this pattern yet. Instead, change the `statusCode=4` value to `statusCode=2` in the **Filter Pattern** above. **Test Pattern** again. Once more, no results. Set the value to `3` and **Test Pattern** again. It should finally capture. Since we can't replicate a 500 error in this lab, we want to use a testable parameter, so select **Assign Metric**.

Set the **Metric name** to *OnCall* and leave the rest with default settings. **Create Filter**. We can now create the alarm.

Press **Create Alarm**. Give the alarm a **Name** of *OnCallEngineer* and leave the **Description** blank. Set it so **Whenever OnCall is** *>= 1* **for** *1* **consecutive period(s)**. To the right, set the **Period** to *1 Minute*. To sync it with the SNS topic we already created, press **+ Notification** under the **Actions** heading, and set **Send notification to** *On_Call_Engineer*. **Create Alarm**.

Return to the nginx page in your browser and refresh a few times to send some traffic to the page. Wait for the minute threshold to pass. Refresh the CloudWatch page until it records the triggered alarm. Check your email to confirm the presence of an AWS notification based upon your alarm.

This lab is now complete.