**Configuring a Metric Filter and Alarm for Testing and Troubleshooting**

### Introduction

When first setting up CloudTrail, CloudWatch and SNS to work in conjunction it can be finicky. Add to that delays on the console, and it can even get frustrating. This step will confirm that what you have configured thus far is indeed working. It does this by creating a **Metric Filter** and **Alarm** that are:

* simple to setup
* easy to test
* quicker to confirm (by keying on an event that occurs without you having to start and stop instances, create or delete buckets, etc.)

Although this approach is not mandatory, it can be helpful by confirming your configuration thus far is indeed working (before adding more complex filters and alarms for various AWS services). This Lab Step can also help as part of troubleshooting efforts when needed.

### Instructions

1. If you are unaware of the public IP address on the local host you are signed into the AWS Console from, in a separate browser tab from the Lab, navigate to [www.whatismyip.com](http://www.whatismyip.com/) to obtain it. Copy your IP address.

2. Navigate to [CloudWatch Log groups](https://us-west-2.console.aws.amazon.com/cloudwatch/home?region=us-west-2#logsV2:log-groups).

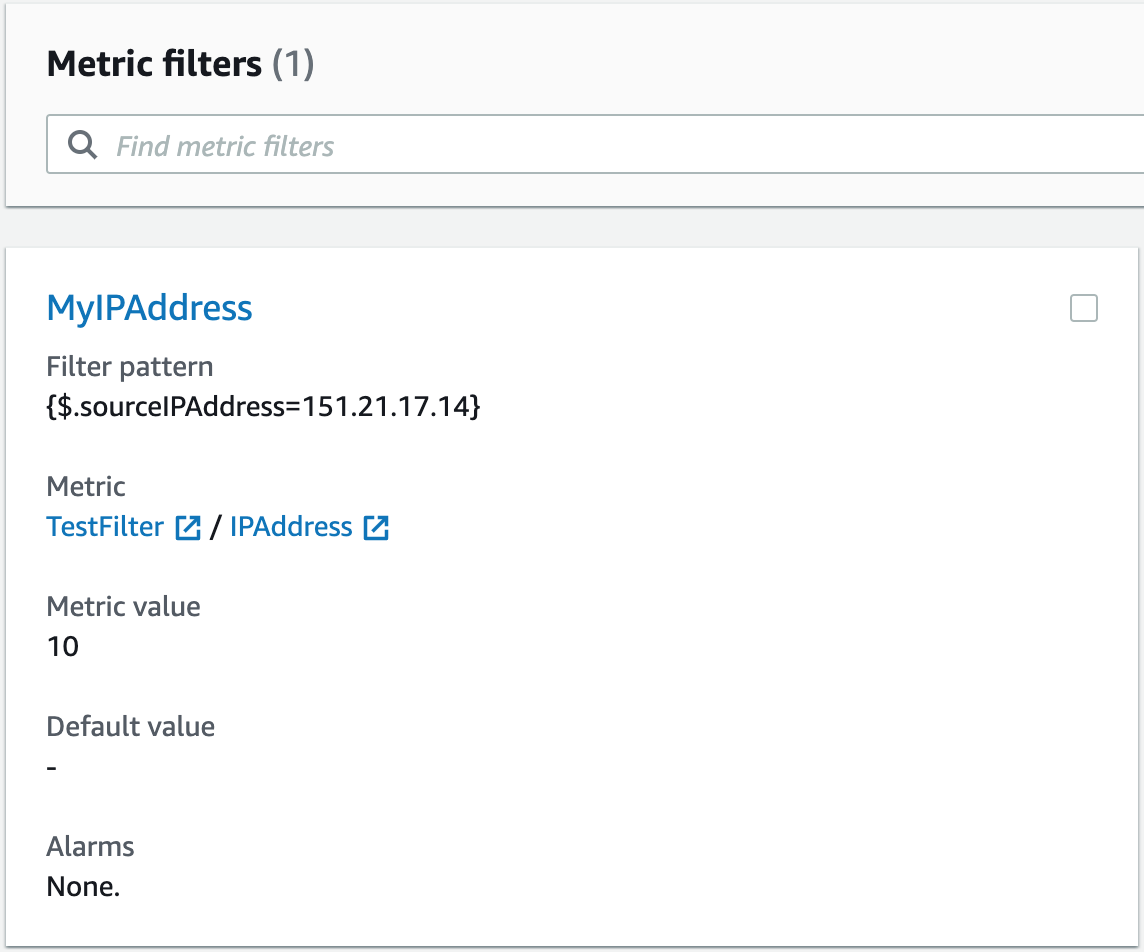
3. Click **Actions** > **Create Metric Filter**.

* For the **Filter Pattern**, enter:  {$.sourceIPAddress="YourIPaddress"}
  + Remember to paste your dot notation IP address into the filter pattern shown above. Example **Filter Pattern**: {$.sourceIPAddress = "104.220.50.205"}
* Click **Test Pattern**. The **Filter Pattern** will be used in a query against the default log data. Because your IP address is involved in most every event, the **Results** should be the majority of all entries in the sample log file. For example, about 45 matches of 50 events found. Regardless of the results, it will confirm the syntax for your filter is correct and the search was performed. Tip: Be sure there is not white space after the "}". The pattern is rather strict. If you receive errors about your pattern, try typing it in manually rather than copy/paste which can produce undesired results.

4. Click the**Next** button, then fill out the fields:

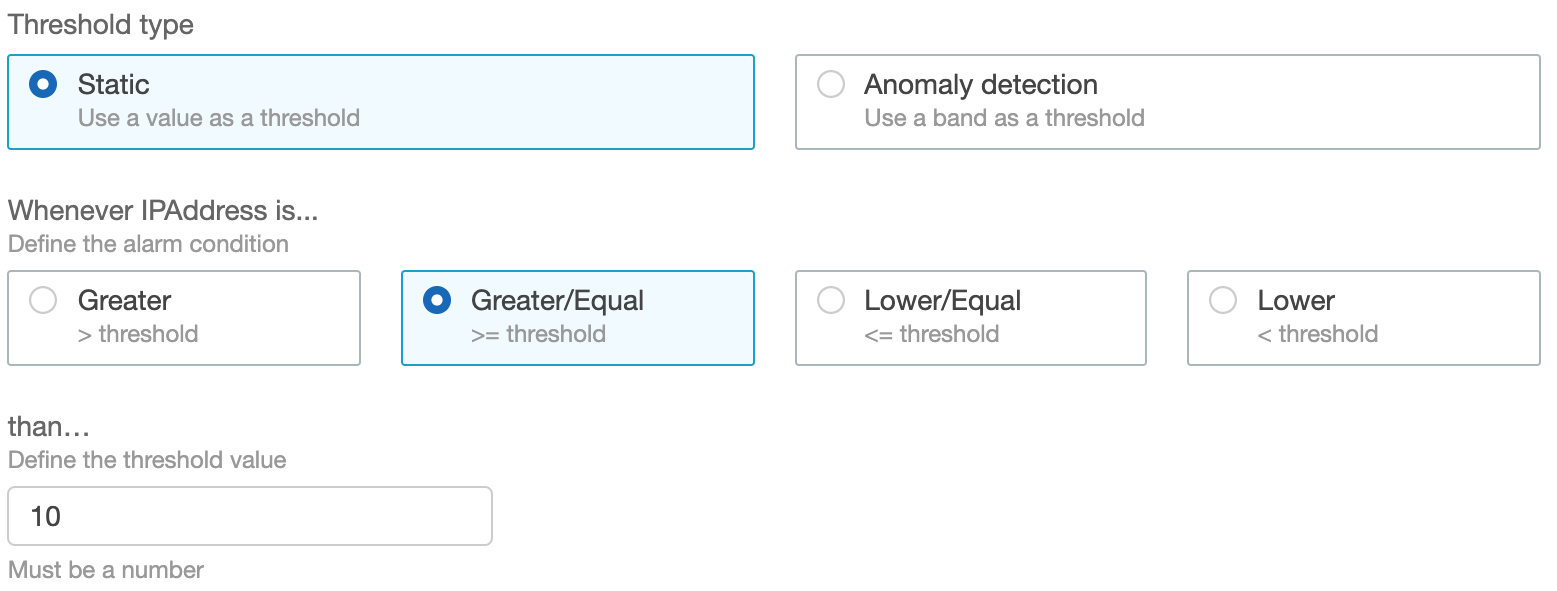
* For the **Filter Name** field enter MyIPAddress
* In the **Metric Details** section, enter TestFilter for the **Metric Namespace**
* For the **Metric Name** enter IPAddress
* For the **Metric Value** field enter 10 as the value.

5. Finally, click **Next** and then **Create Filter**. You will see the new metric filter:



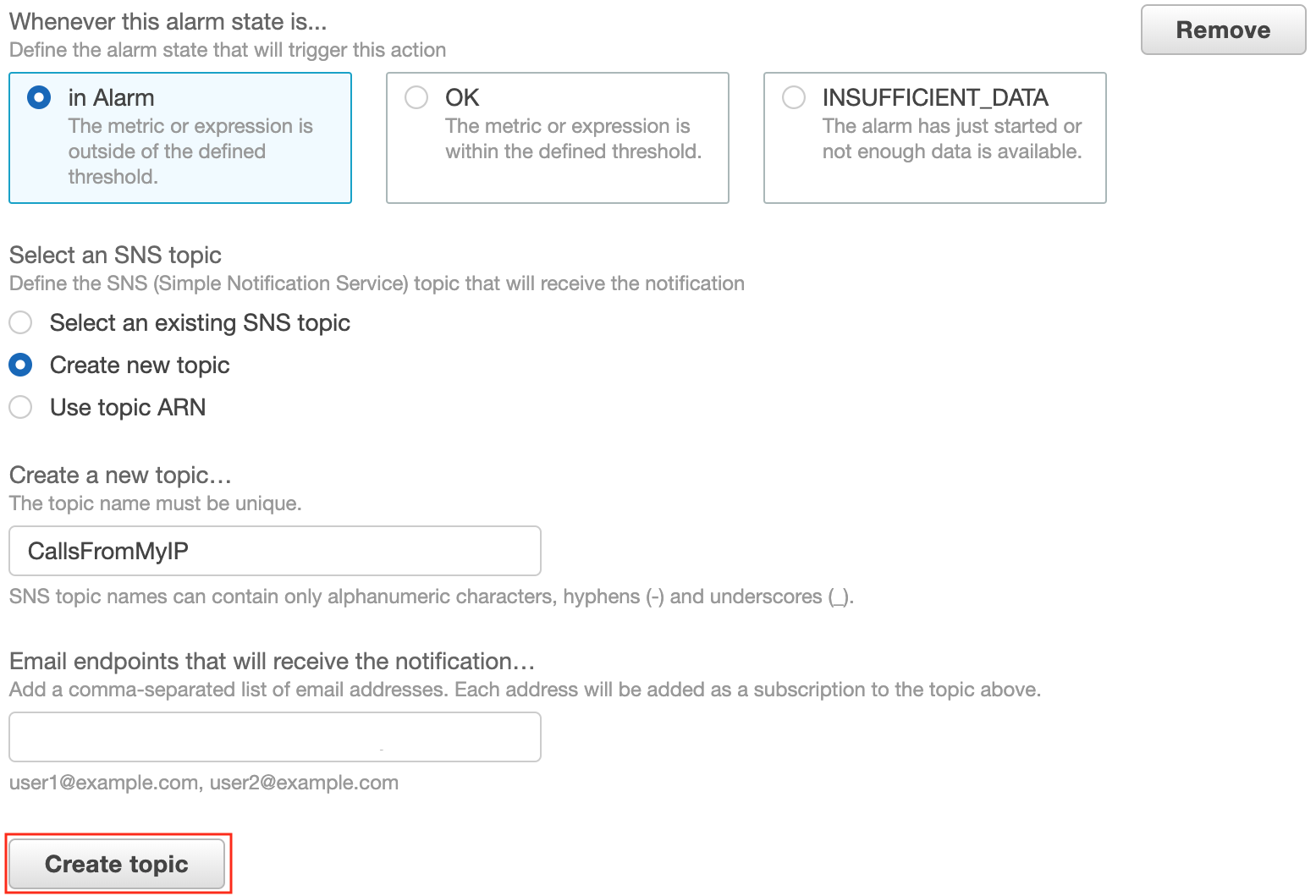
6. Select the metric filter and click on **Create alarm**.

7. Set the **Conditions** section as shown:



* **Threshold type**: **Static**
* **Whenever IPAddress is...**: **Greater/Equal**
* **than…**: 10

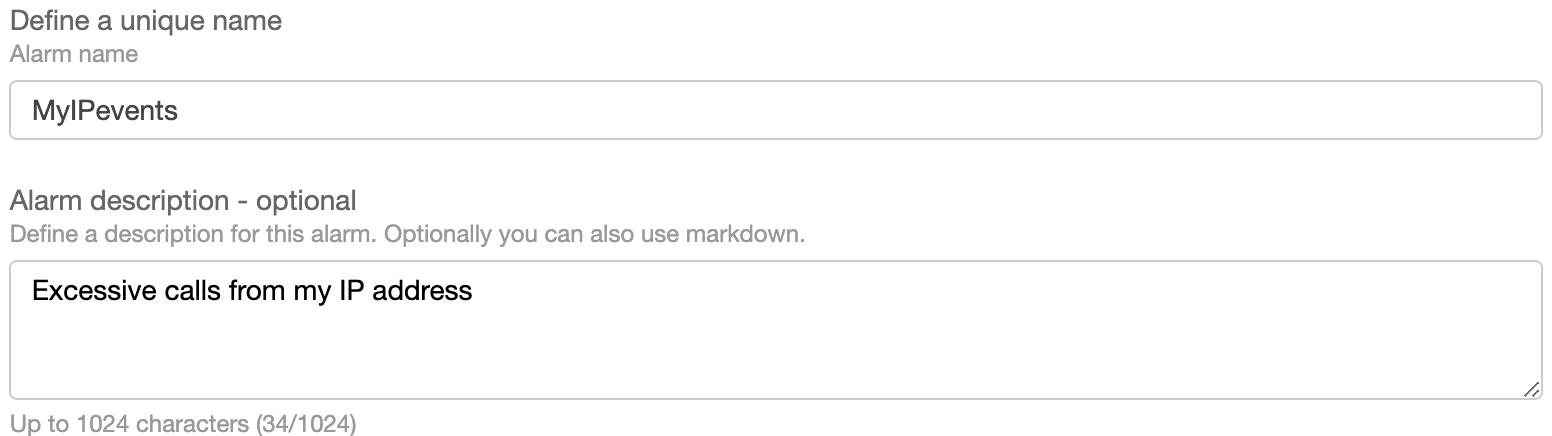
8. Click Next and set the section as shown:



* **Whenever this alarm state is...**: **In Alarm**
* **Select an SNS topic**: **Create new topic**
* **Create a new topic…**: CallsFromMyIP
* **Email endpoints that will receive the notification…**: your personal email

Click **Create topic** and then **Next**.

9. Set the section as shown:

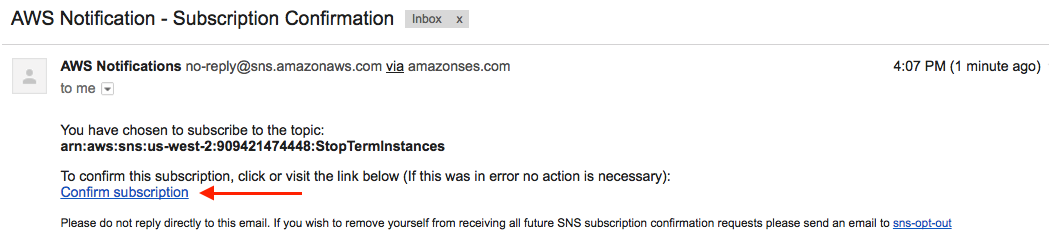


* **Define a unique name**: MyIPevents
* **Alarm description**: Excessive calls from my IP address

Click **Next**.

10. Click **Create Alarm** when ready.

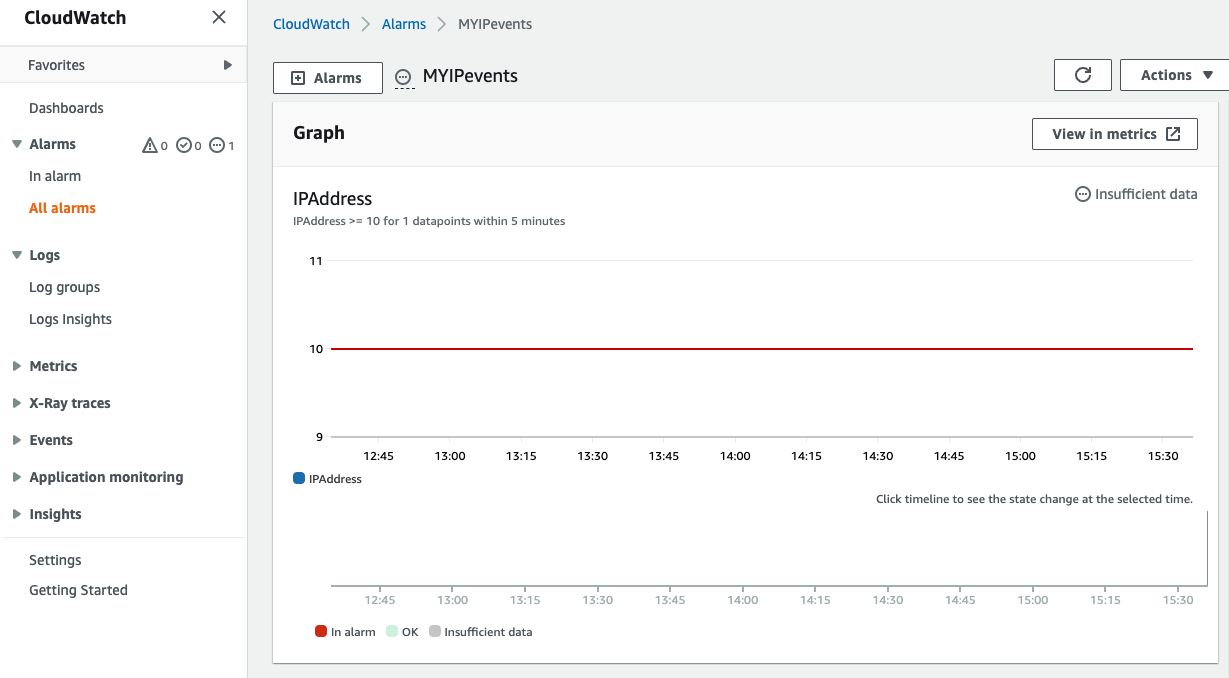
11. Check for an email from **AWS Notifications**. Open up the email and click the **Confirm subscription** link:



Notice in the email confirmation dialog that your email address changes to a green check mark and the **View Alarm** button is actionable now. Also, you should get a subscription confirmation in your email client. (For example, a confirmation message from Amazon Simple Notification Service (SNS) in a new browser tab if using a browser based email like Gmail.) If for some reason there is a delay and you are pressed for time, you can select **I will do it later**and continue with this Lab Step**.**(However, email notiications from raised alarms will not deliver to your inbox.)

Reminder: The importance and focus of this Lab is on mechanics around configuration and testing, not the actual service and alarm that is configured. Further, the focus of this Lab Step is to verify CloudWatch configuration is working prior to testing more compelling AWS services (such as EC2 or S3).

12. From the **CloudWatch > Alarms** page, select the **Alarm**. It should look similar to:



Your alarm will be in one of three states:

* **Alarm** - Alarm was triggered.
* **INSUFFICIENT** - Not enough data exists to either set and Alarm or set the status to OK. (You'll learn more about this in the next Lab Step.)
* **OK** - Alarm not triggered, status is normal.

You may even see it transition from one state to another. For example, from **Insufficient** to **Alarm**.

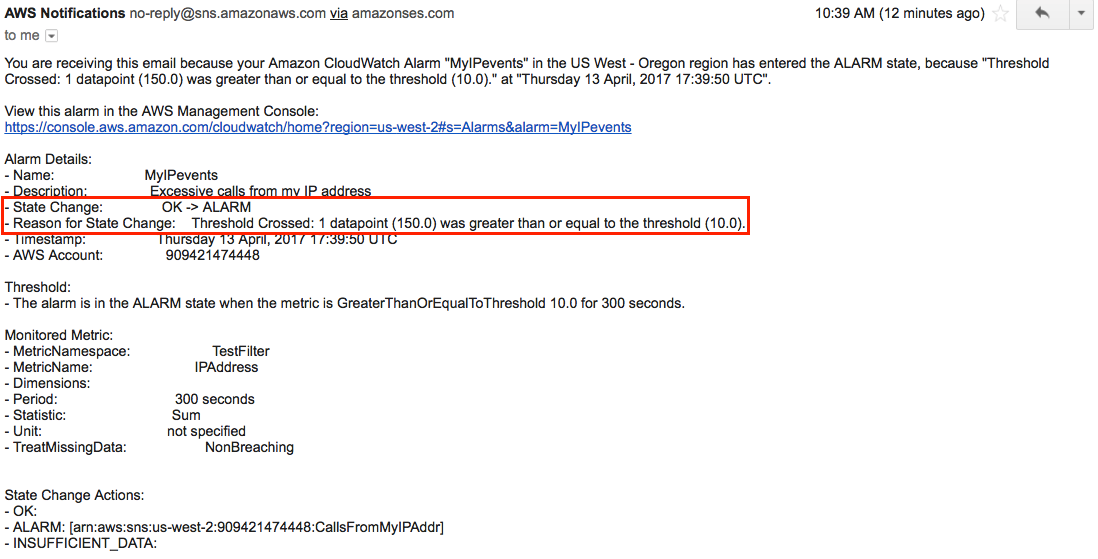
13. Navigate about reviewing what you just configured in **CloudWatch Metrics** and **Logs**. Tip: If an Alarm is not triggered with normal usage in the console, you can usually speed up the process by:

* Navigate back to **Alarms**. Select the alarm, then **Actions > Edit.**Change the following:
  + >= **1**
  + **Period: 1 Minute**

An example of an Alarm in the **ALARM** state:



14. When the alarm transitions to an **Alarm** state, check your email. If your subscription confirmed earlier, you should receive an email similar to the following:



### Summary

In this Lab Step you setup an environment such that CloudWatch metric filters and alarms with email notifications from Simple Notification Service (SNS) is tested. The example configuration was catered for ease of testing and/or troubleshooting purposes, not a compelling or useful scenario. However, when the configuration tested in this Lab Step works, it is likely a more complex configuration using other AWS services will work as well.

After you have completed this lab, see [Filter and Pattern Syntax](http://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/FilterAndPatternSyntax.html) in the AWS documentation for more information and examples.

VALIDATION CHECKS

**1Checks**

Check again

**Created CloudWatch Alarm**

Created CloudWatch Alarm

Amazon CloudWatch