JMeter 3.2 + InfluxDB + Grafana + Slack

Using

Docker Containers

# **Prerequisites:**

* Ubuntu Server 16.x (VM on Virtual Box)
* Docker installed on Ubuntu
* JMeter 3.2 installed on Ubuntu and on a Mac or Windows system (to edit JMeter test plan)
* Access to my GitHub repository (for sample JMeter test plan) - https://github.com/smarigowda/learnDocker

# **GitHub repository**

**git clone https://github.com/smarigowda/learnDocker.git**

# **InfluxDB docker container**

Create a directory named influxDBData

Change directory into influxDBData and run the following command

**cd influxData**

**docker run -d --name=influxdb -p 8083:8083 -p 8086:8086 -v $PWD:/var/lib/influxdb influxdb**

# **influxdb client**

Run another container for influxdb client

**docker run --rm --link=influxdb -it influxdb influx -host influxdb**

You should see a command prompt (of influxdb client)

Run the following command to create a database inside the container

**> create database jmeterInfluxDBContainer**

Ref:

<https://docs.influxdata.com/influxdb/v0.9/tools/shell/>

<https://docs.influxdata.com/influxdb/v0.9/introduction/getting_started/>

# **Oracle java8 jdk on Ubuntu**

JMeter 3.2 requires java 8. Find below the steps to install Java 8 jdk on Ubuntu.

sudo add-apt-repository ppa:webupd8team/java

sudo apt-get update

sudo apt-get install oracle-java8-installer

**santosh@ubuntu:~/SAN/learnDocker$** java -version

java version "1.8.0\_131"

Java(TM) SE Runtime Environment (build 1.8.0\_131-b11)

Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)

**santosh@ubuntu:~/SAN/learnDocker$**

# **Grafana docker container**

Run the following command to start a docker container running Grafana

**docker run -d -i -p 3000:3000 grafana/grafana**

Browse the following url, it should display the login screen of Grafana. Replace 10.189.133.50 with your virtual machine IP

<http://10.189.133.50:3000/>

Configure Grafana to use Influx DB

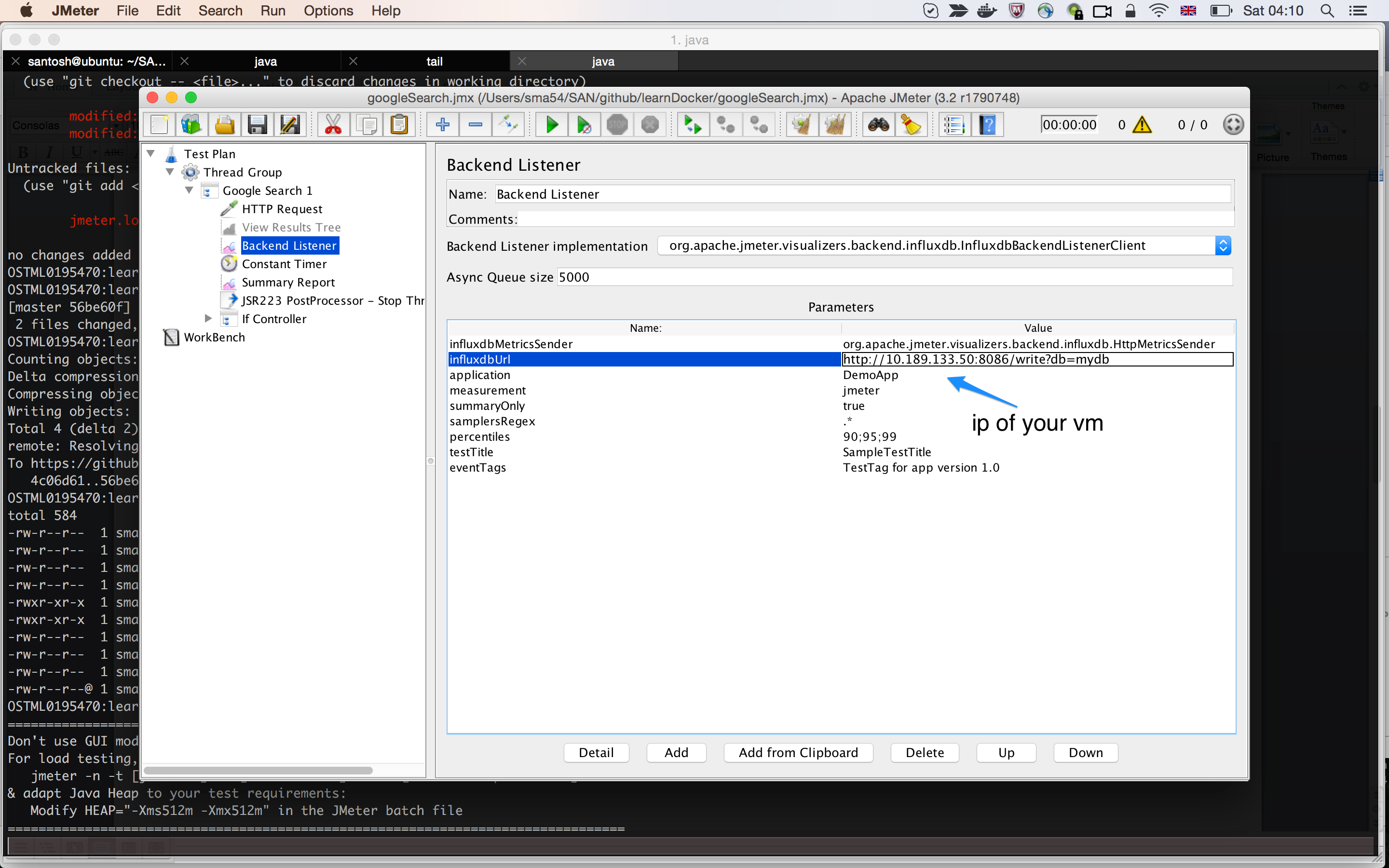
Login to Grafana using default user admin/admin

* click on Add Datasource
* Select Type as InfluxDB
* Set the url to <http://10.189.133.50:8086> (Replace 10.189.133.50 with your vm IP)
* Provide some Name (ex: jmeterInfluxDB)
* Influx DB Details: Fill in Database as jmeterInfluxDBContainer
* Click on Add
* Click on Save & Test
* You should see a message ‘Success Data source is Working’

# **Create JMeter Test Plan (Mac/ Windows)**

Refer the JMeter test plan googleSearch.jmx in my github repository

A screenshot of JMeter test plan is given below. Click on Backend Listener and **(1)** update the IP address of influxDB with your Virtual Machine IP **(2)** update the db name in the url to match the influxDB name

****

# **Run JMeter test**

Run the JMeter test plan in GUI mode on Windows/ Mac

**jmeter-t googleSearch.jmx**

**Or**

Run the JMeter test plan in non-GUI mode on Ubuntu server.

**jmeter -n -t googleSearch.jmx &**

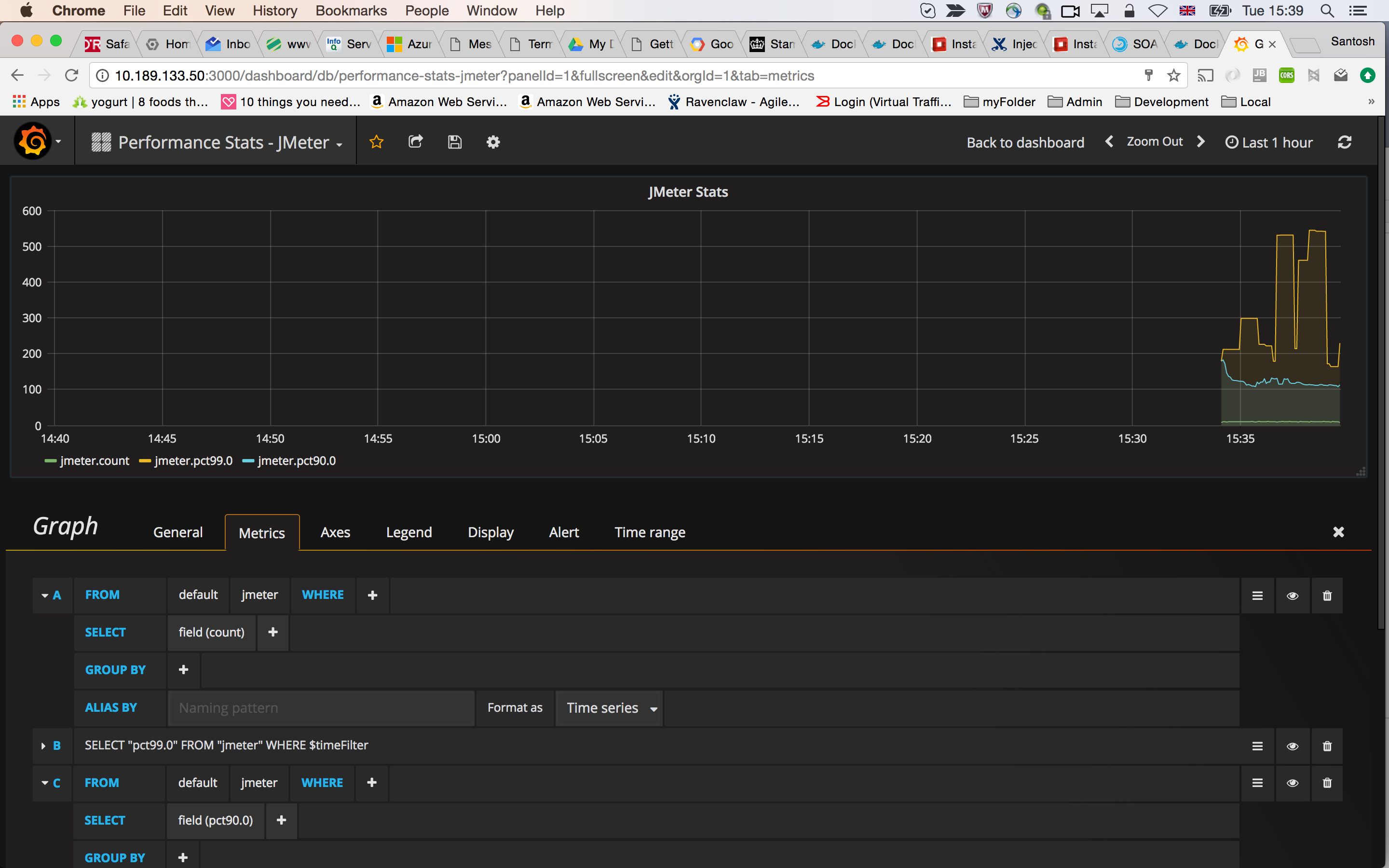
-n specifies JMeter to run in non-GUI mode.

# **View stats in Grafana**

While the test is running, you can see the jmeter results stats in Grafana by creating graphs. Create a graph to plot 99th, 95th Percentiles, Count of transactions and check if they match with the JMeter report.

# **Grafana Dashboard for JMeter Statistics**

A sample Grafana dashboard for JMeter statistics is given below. It shows the request count, 90th Percentile response time and 99th Percentile response time.



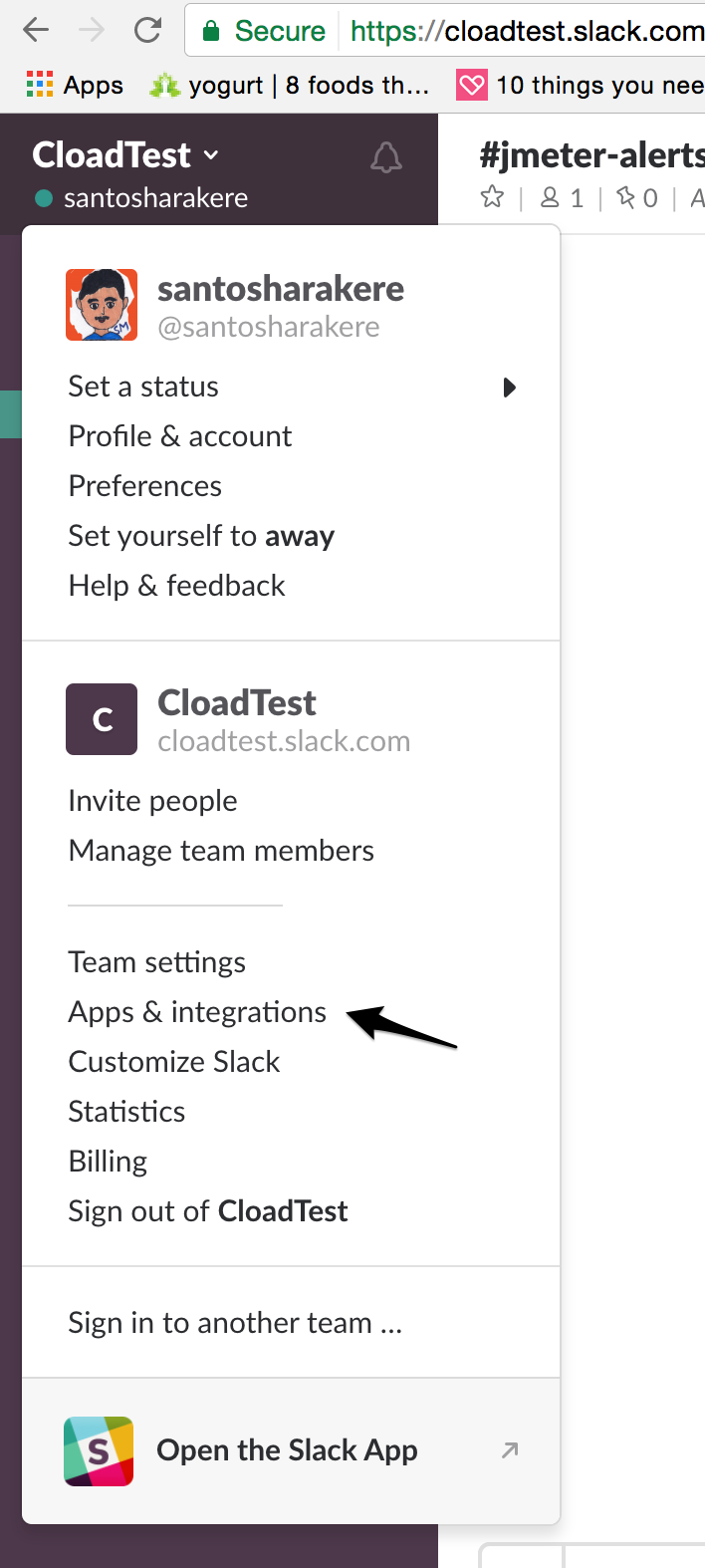
# **Slack Team Account**

Create a new slack team account and a slack channel.

Go to <https://slack.com/pricing> and create a new team by following online instructions and create a channel #jmeter-alerts (ex: I created a channel **cloadtest**, its url is https://cloadtest.slack.com)

Create an incoming webhook

Click on your team on the main page, you will see a pop up (shown below)

****

Click on ‘Apps & Integrations’, this will open a new web page. Click on the ‘Build’ menu on this page, this opens ‘Slack API’ page. Click on ‘incoming webhooks’ and then Click on ‘incoming-webhook-integration’.

Now choose a channel (ex: #jmeter-alerts). Click on ‘Add Incoming Webhook Integration’. This creates a new webhook, note down the webhook url.

Ex: https://hooks.slack.com/services/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

# **Notification Channel**

Create a notification channel in Grafana.

Steps:

Choose Notification Channels under Alerting

Click the button ‘New Channel’

Choose Type as Slack

Give some name (Slack Channel for JMeter Alerts)

Check the checkbox ‘Send on all alerts’

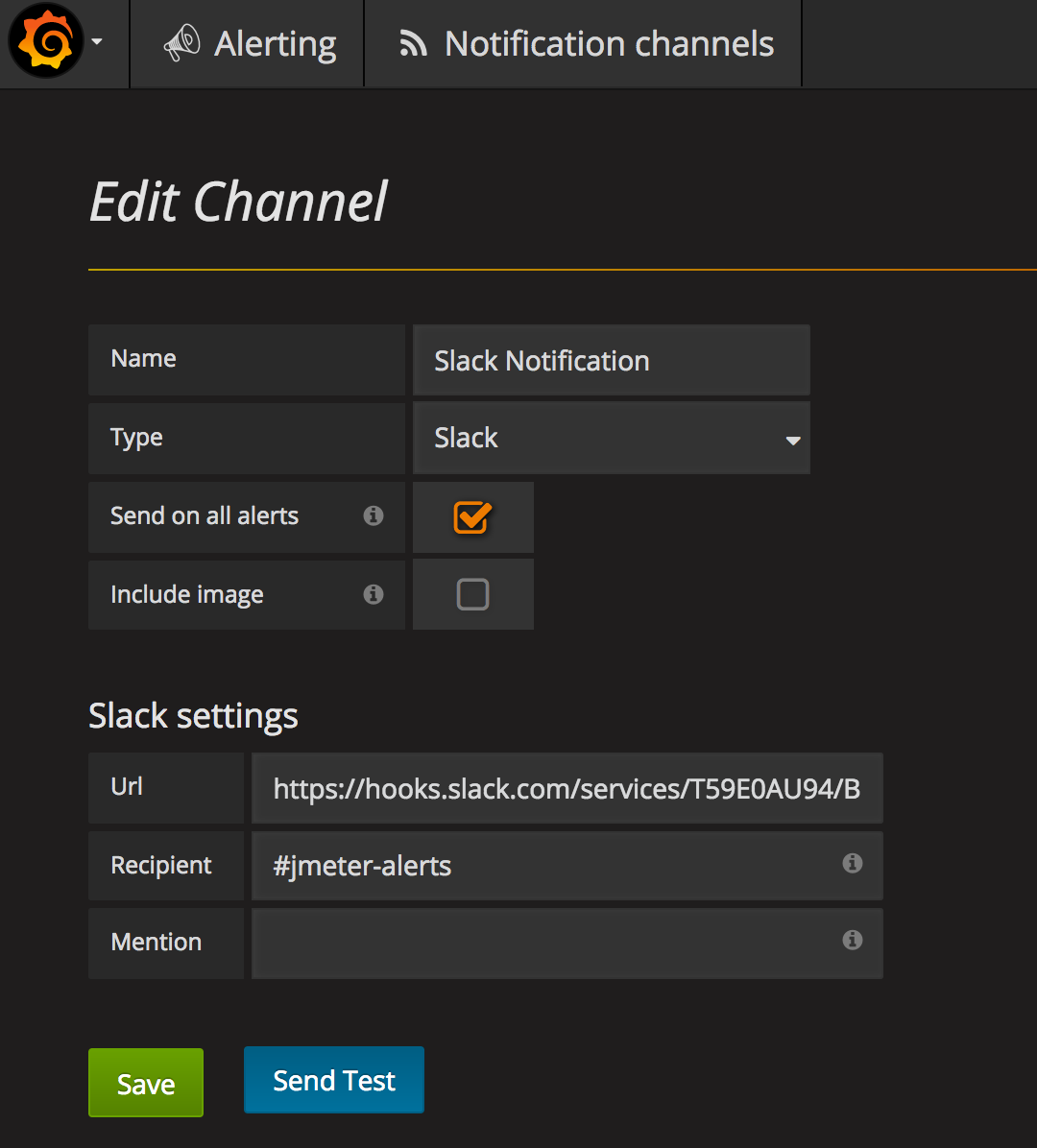
Fill in the URL (ex: https://hooks.slack.com/services/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)

Fill in the Recipient (#jmeter-alerts)

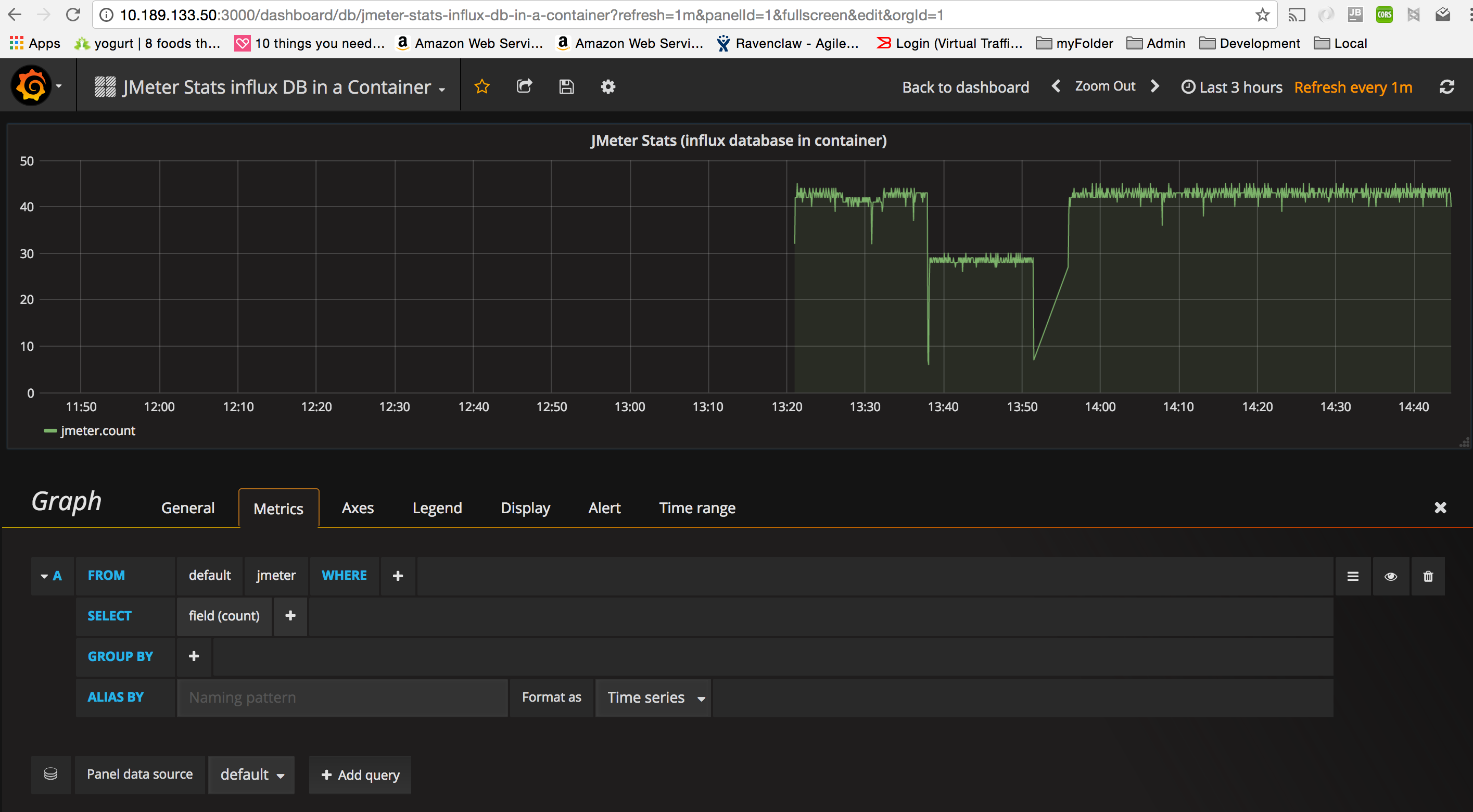
Save and Send Test.

You should see a test alert in Slack channel

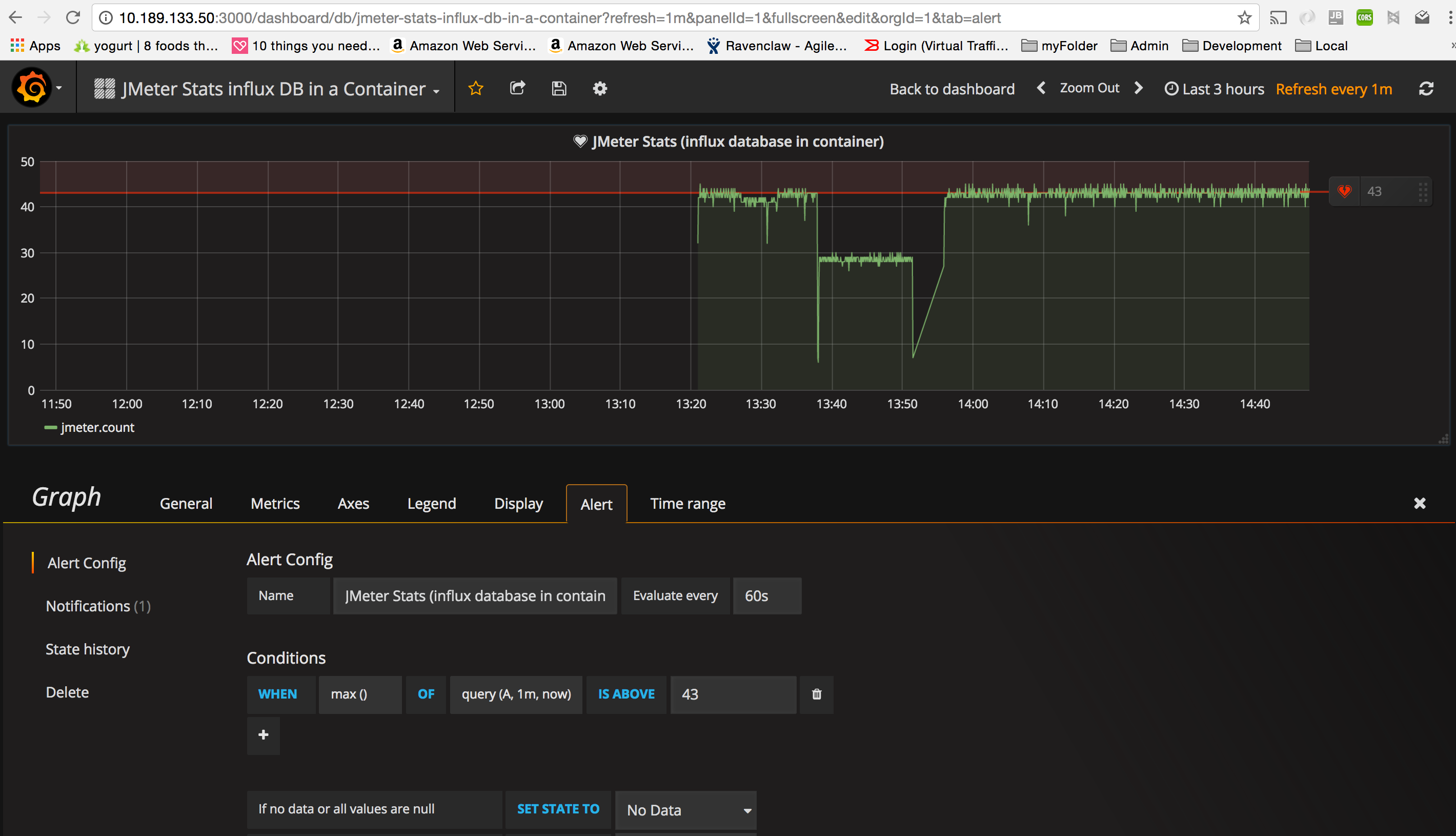
Sample Screenshot:



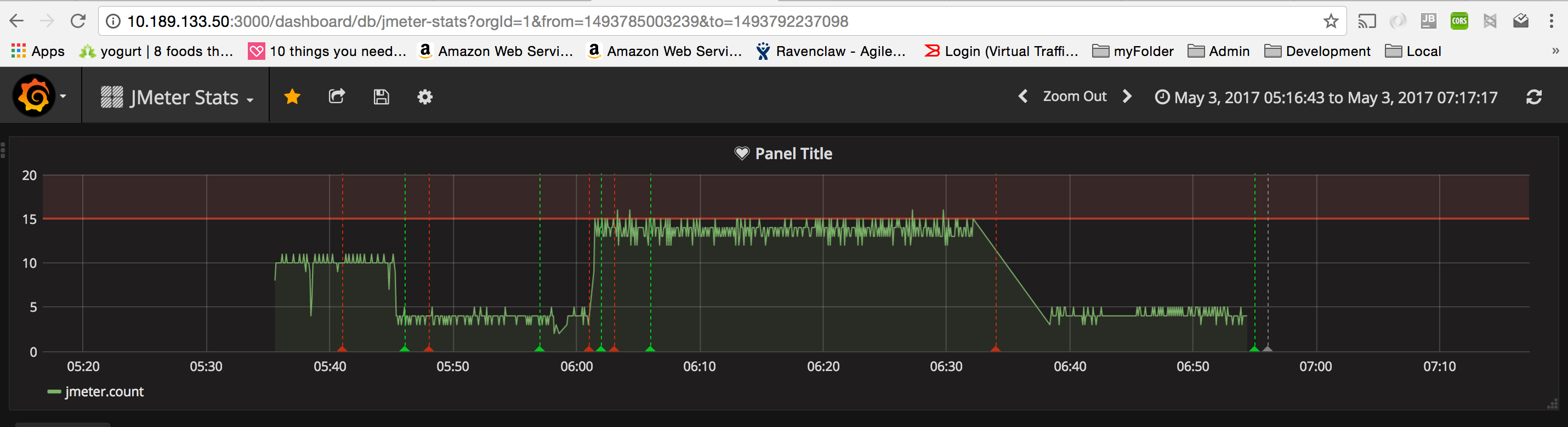
Create a dashboard in Grafana to view JMeter statistics. Screenshot of a sample dashboard is given below.



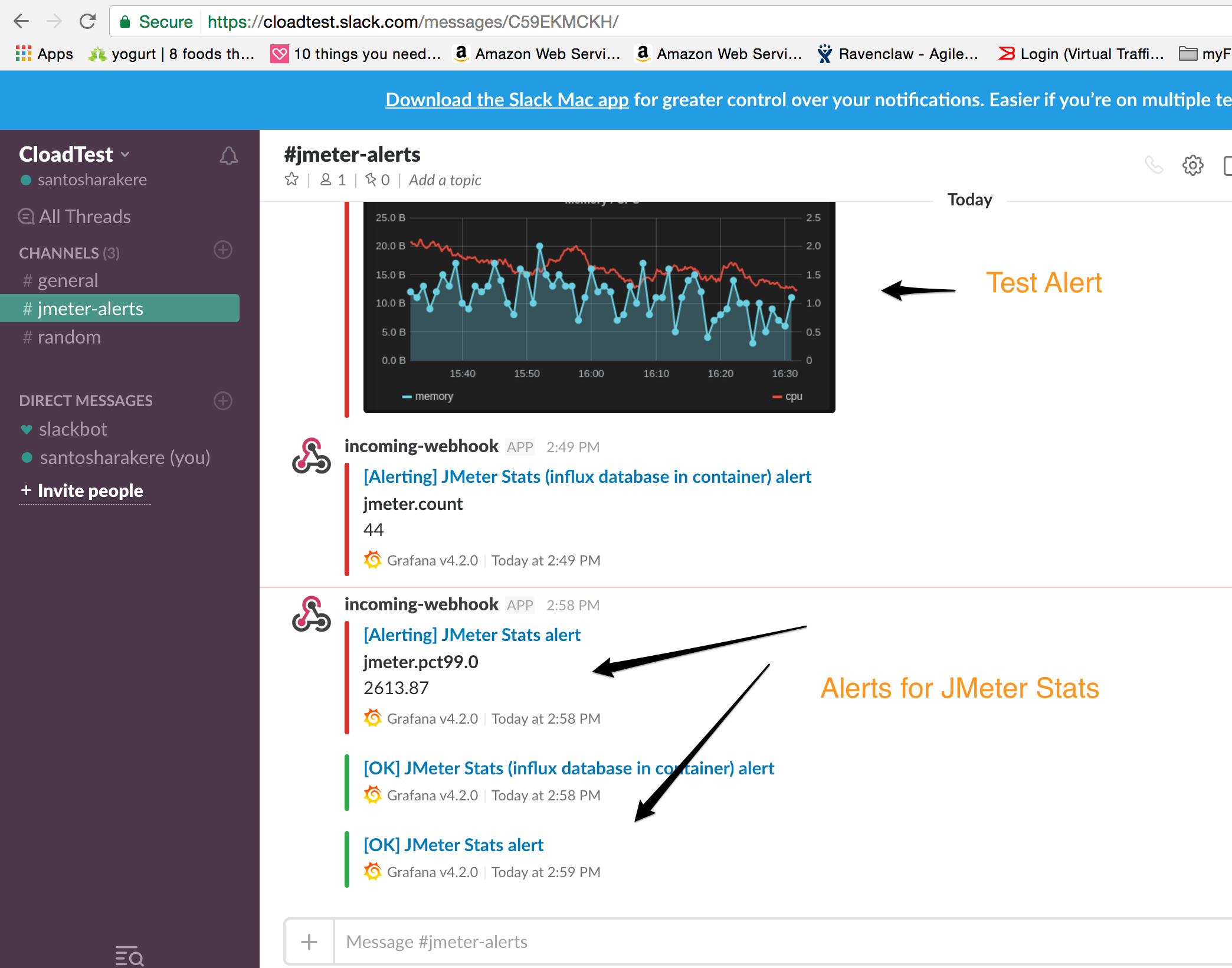
Configure Alert for the dashboard.



Grafana sends notifications on state changes, so OK -> Alerting, Alerting -> OK, OK -> NoData, etc



Notifications are sent to slack channel:



**Notes**

**# JMeter 3.2:**

santosh@ubuntu:~/SAN/learnDocker$ **which jmeter**

/home/santosh/SAN/Software/apache-jmeter-3.2/bin/jmeter

santosh@ubuntu:~/SAN/learnDocker$

**# Starting influx db container:**

santosh@ubuntu:~/SAN/influxData$ **docker ps**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

12d066413667 influxdb "/entrypoint.sh in..." 38 seconds ago Up 37 seconds 0.0.0.0:8083->8083/tcp, 0.0.0.0:8086->8086/tcp romantic\_austin

santosh@ubuntu:~/SAN/influxData$