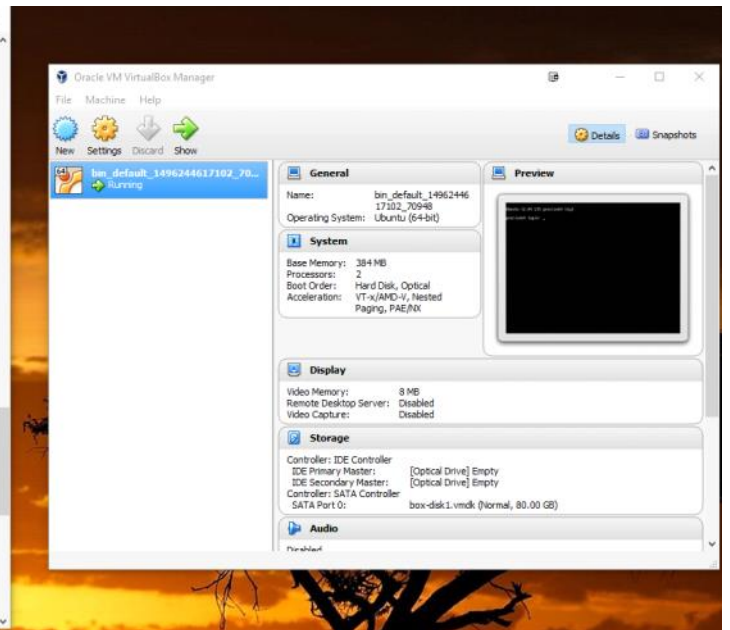


Friday, June 2, 2017 1:29 PM

```

Administrator Command Prompt
C:\Windows\system32> vagrant up
-- default: VirtualBox version: 5.1
--> default: Mounting shared folders...
-- default: /vagrant -> C:\hashicorp\vagrant\bin
C:\hashicorp\vagrant\bin> vagrant destroy
-- default: are you sure you want to destroy the 'default' VM? [Y/n] y
--> default: Forcing shutdown of VM...
--> default: Destroying VM and associated drives...
C:\hashicorp\vagrant\bin> vagrant init hashicorp/precise64
"vagrantfile" already exists in this directory. remove it before
running "vagrant init"
C:\hashicorp\vagrant\bin> vagrant init hashicorp/precise64
-- "vagrantfile" has been placed in this directory. You are now
ready to "vagrant up" your first virtual environment! please read
the comments in the vagrantfile as well as documentation on
"vagrantup.com" for more information on using vagrant.
C:\hashicorp\vagrant\bin> vagrant up
-- bringing machine 'default' up with "virtualbox" provider...
--> default: Importing base box "hashicorp/precise64"
--> default: Matching MAC address for NAT networking...
--> default: Checking if box "hashicorp/precise64" is up to date...
--> default: Setting the name of the VM: bin_default_127.0.0.1__18026
--> default: Clearing any previously set network interfaces...
--> default: Preparing network interfaces based on configuration...
--> default: Adapter 1: not
--> default: Forwarding ports...
--> default: 22 (guest) => 2222 (host) (adapter 1)
--> default: Mounting VM...
--> default: Waiting for machine to boot. This may take a few minutes...
-- default: SSH address: 127.0.0.1:2222
-- default: SSH username: vagrant
-- default: SSH auth method: private key
-- default:
-- default: Vagrant insecure key detected. Vagrant will automatically replace
-- default: this with a newly generated keypair for better security.
-- default:
-- default: Inserting generated public key within guest...
-- default: Removing insecure key from the guest if it's present...
-- default: Key imported successfully and reconnecting using new SSH key...
--> default: Machine booted and ready!
--> default: Checking for guest additions in VM...
-- default: The guest additions on this VM do not match the installed version
-- default: of VirtualBox! In most cases this is fine, but in rare cases it can
-- default: present things such as shared folders from working properly. If you see
-- default: shared folder errors, please make sure the guest additions within the
-- default: virtual machine match the version of VirtualBox you have installed on
-- default: your host and reload your VM.
-- default:
-- default: Guest additions version: 4.2.0
-- default: VirtualBox version: 5.1
--> default: Mounting shared folders...
-- default: /vagrant -> C:\hashicorp\vagrant\bin
C:\hashicorp\vagrant\bin>

```



### Host Tags

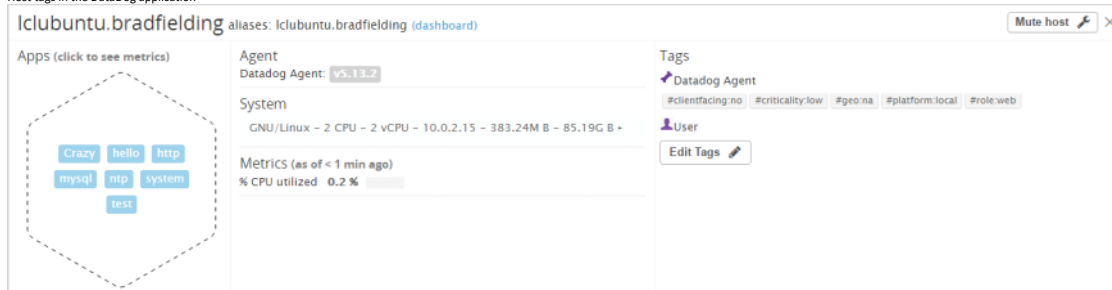
### Tags in host config file

```
# Force the hostname to whatever you want. (default: auto-detected)
hostname: lclubuntu.bradfielding

# Enable the trace agent.
# apm_enabled: false

# Set the host's tags (optional)
tags: role:web, geo:na, criticality:low, clientfacing:no, platform:local
```

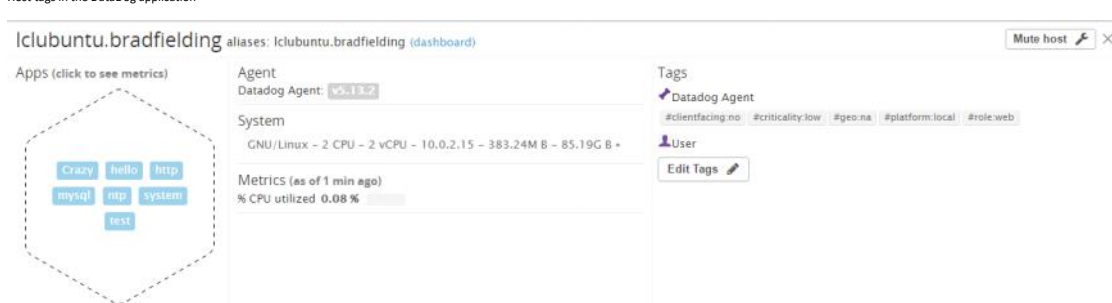
### Host tags in the DataDog application



Tags in host config file

```
22
23 # Force the hostname to whatever you want. (default: auto-detected)
24 hostname: gpwinserver.bradfielding
25
26 # Set the host's tags (default: no tags)
27 tags: role:ad, geo:asiapac, criticality:high, clientfacing:no, platform:gpc
```

## Host tags in the DataDog application



Agent Check showing MySQL success

```

Checks
=====

matt_random (5.13.2)
-----
- instance #0 [OK]
- Collected 1 metric, 0 events & 0 service checks

http (5.13.2)
-----
- instance #0 [OK]
- instance #1 [OK]
- instance #2 [OK]
- Collected 1 metric, 2 events & 0 service checks

network (5.13.2)
-----
- instance #0 [OK]
- Collected 15 metrics, 0 events & 0 service checks

crazy_meter (5.13.2)
-----
- instance #0 [OK]
- Collected 1 metric, 0 events & 0 service checks

mysql (5.13.2)
-----
- instance #0 [OK]
- Collected 168 metrics, 0 events & 1 service check
- Dependencies:
  - pymysql: 0.6.6.None

ntp (5.13.2)
-----
- Collected 0 metrics, 0 events & 0 service checks

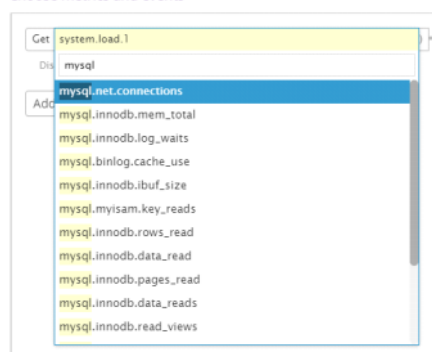
disk (5.13.2)
-----
- instance #0 [OK]
- Collected 32 metrics, 0 events & 0 service checks

hello (5.13.2)
-----
- instance #0 [OK]
- Collected 1 metric, 0 events & 0 service checks

```

DataDog App showing available MySQL metrics

Choose metrics and events

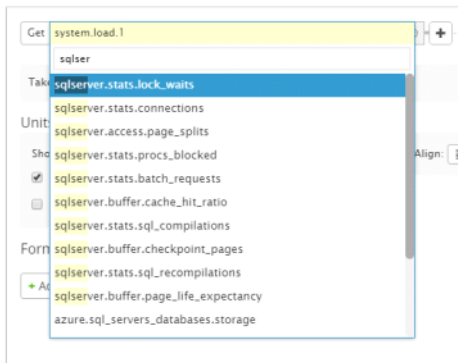


MS SQL integration

Agent Info Showing success

DataDog application showing available SQL metrics

## 2 Choose metrics and events



### Custom Agent Check

Check named - matt\_random

```
Checks
=====

matt_random (5.13.2)
-----
- instance #0 [OK]
- Collected 1 metric, 0 events & 0 service checks

http (5.13.2)
-----
- instance #0 [OK]
- instance #1 [OK]
- instance #2 [OK]
- Collected 1 metric, 2 events & 0 service checks

network (5.13.2)
-----
- instance #0 [OK]
- Collected 15 metrics, 0 events & 0 service checks

mysql (5.13.2)
-----
- instance #0 [OK]
- Collected 168 metrics, 0 events & 1 service check
- Dependencies:
  - pymysql: 0.6.6.None

ntp (5.13.2)
-----
- Collected 0 metrics, 0 events & 0 service checks

disk (5.13.2)
-----
- instance #0 [OK]
- Collected 32 metrics, 0 events & 0 service checks

hello (5.13.2)
-----
- instance #0 [OK]
- Collected 1 metric, 0 events & 0 service checks
```

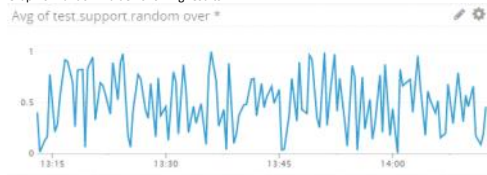
### Script Source

```
import random
from checks import AgentCheck
class MattCheck(AgentCheck):
    def check(self, instance):
        self.gauge('test.support.random', random.random())
```

### Script Config

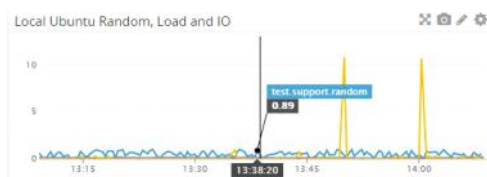
```
init_config:
min_collection_interval: 20
instances:
[]
```

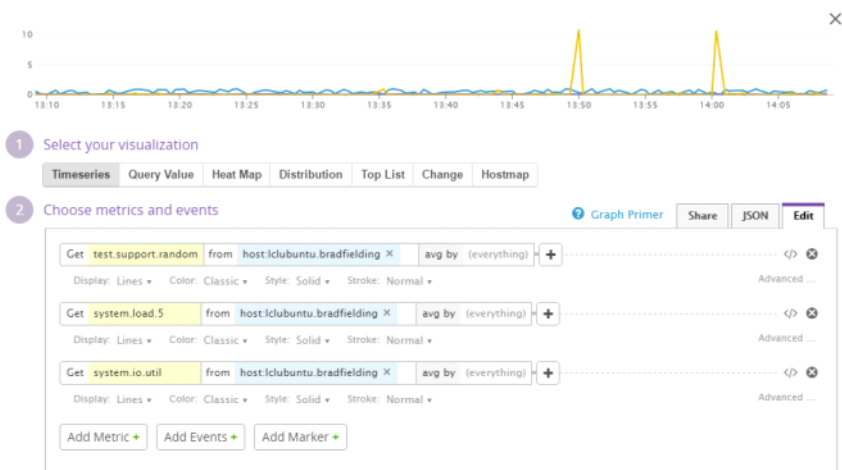
### Graph of Random Value - showing results



## Level 2 - Visualizing Data

Created basic Dashboard showing 3 metrics from the local ubuntu host





Box around graph at higher than .9



Emailed result of same annotation

**DATADOG**

Matthew A. Bradfield (@bradfield.matt@gmail.com) mentioned you in a comment:

Matthew A. Bradfield  
Local Ubuntu Test Random Metric

This very crazy peak coincides with Boston HQ employees returning from lunch. Please log this @matt@bradfielding.co

02 Jun, 18:56:13 UTC

Reply to @bradfield.matt@gmail.com

To manage your Datadog subscriptions, click [here](#).

### Level 3 - Alerting on your Data

Monitor to notify of random metric going over .9. Created as multi-alert to trigger by each host. Included conditional formatting so Alert and Alert Recovery have different messages

Get **test.support.random** from (everywhere) excluding (none) avg by **host** X +

Multi Alert Trigger a separate alert for each **host** X reporting your metric

## 2 Set alert conditions

**Threshold Alert** Change Alert Anomaly Alert An alert is triggered whenever a metric crosses a threshold. ?

Trigger when the metric is **above** the threshold **at least once** during the last **5 minutes** for any **host**

Alert threshold: **0.9** (0.9)

Warning threshold: **Warning threshold is**

**Require** a full window of data for evaluation.

Note: We highly recommend you select "Do Not Require" for sparse metrics, otherwise some evaluations will be skipped.

**Do not notify** if data is missing.

Note: the missing data window must be at least 2x the evaluation period above to work

**[Never]** automatically resolve this event from a triggered state.

For new hosts, wait **300** seconds before evaluating this monitor

Delay evaluation by seconds

## Say what's happening

Use message template variables ?

☒ Preview ☒ Edit ☒ Include triggering tags in notification title

Markdown supported

{{host.name}} Concerning Crazy level

Too much crazy in {{host.name}}

Is host client facing {{host.clientfacing}}

Platform {{host.platform}}

{{#is\_alert}}1) check to see is service interruption for host provider{{/is\_alert}}  
{{#is\_alert}}2) Identify any affected services running on the host.{{/is\_alert}}  
{{#is\_alert}}3) If host is client facing email all affected users and prepare for failover{{/is\_alert}}  
{{#is\_alert}}4) If host is not client facing, email owner and monitor for 2 hours per SLA{{/is\_alert}}

Service:

**[Never]** renotify if the monitor has not been resolved.

## Notify your team

**Matthew A. Bradfield** X

**Do not notify** alert recipients when this alert is modified

**Do not restrict** editing this monitor to its creator or administrators

Notification Received when alert was triggered



[Triggered on {host.azureubuntu.bradfielding}] Server azureubuntu.bradfielding Concerning Crazy level

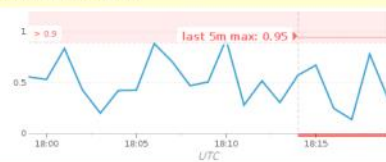
Too much crazy in azureubuntu.bradfielding

Is host client facing yes

Platform azure

1) check to see is service interruption for host provider  
2) Identify any affected services running on the host  
3) If host is client facing email all affected users and prepare for failover  
4) If host is not client facing, email owner and monitor for 2 hours per SLA

@bradfield.matt@gmail.com



Crazy.meter over host:azureubuntu.bradfielding was > 0.9 at least once during the last 5m.

The monitor was last triggered at Fri Jun 02 2017 18:20:00 UTC (23 secs ago)

[\[Monitor Status\]](#) - [\[Edit Monitor\]](#) - [\[View azureubuntu.bradfielding\]](#)

This alert was raised by account Datadog Recruiting Candidate

Comment in Datadog

Notification received when alert was recovered

[Recovered on (host:gpeubuntu.bradfielding)] Server gpeubuntu.bradfielding Concerning Crazy level  
Too much crazy in gpeubuntu.bradfielding  
Is host client facing yes

Platform

- 1) Confirm all services are available to the host
- 2) If host is client facing, file incident report and monitor for 24 hours

[@bradfield.matt@gmail.com](#)



Crazy.level over host:gpeubuntu.bradfielding was  $\leq 0.9$  at all times during the last 5m.

The monitor was last triggered at Fri Jun 02 2017 18:07:00 UTC (15 mins ago)

[\[Monitor Status\]](#) - [\[Edit Monitor\]](#) - [\[View gpeubuntu.bradfielding\]](#)

This alert was raised by account Datadog Recruiting Candidate

[Comment in Datadog](#)

Scheduled Downtime

## Schedule Downtime

### 1 Choose what to silence

Monitor:  ×

× ?

### 2 Schedule

From:  to:

Repeat:  this downtime

Repeat:  ? [Preview planned recurrences](#)

Repeat every:  days

Until:

Summary: **Daily**

### 3 Add a message

[↔ Preview](#) [✎ Edit](#) Markdown supported

Daily downtime for Crazy level monitor. Runs from 1900-0900 everyday without end date. @bradfield.matt@gmail.com

### 4 Notify your team

×

[Cancel](#)

[Save](#)

Email Notification

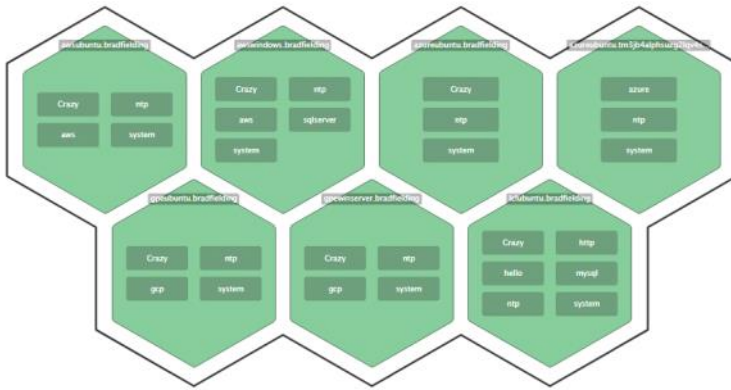
Matthew A. Bradfield ([@bradfield.matt@gmail.com](#)) mentioned you in a comment:

 **Matthew A. Bradfield**  
Matthew A. Bradfield scheduled downtime on [\[\(host.name\)\] Concerning Crazy level](#) from 11:00PM on June 2 to 1:00PM UTC on June 3.  
Daily downtime for Crazy level monitor. Runs from 1900-0900 everyday without end date.  
[@matt@bradfielding.co](#)  
02 Jun, 18:50:58 UTC

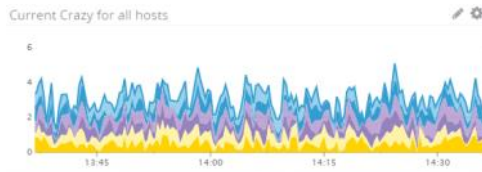
[Reply to @bradfield.matt@gmail.com](#)

## Additional Steps Complete

- 1) Deployed Ubuntu and Windows Server to Google Cloud Platform, AWS and Azure, deployed agents to all machines. Updated tags and hostnames appropriately for all



2) Created a 'Crazy\_Meter' agent check and configured it on all of the hosts successfully



3) Configured Integrations for

- Azure
- AWS
- GCP
- SQL Server
- MSFT Event Viewer
- Azure SQL DB

4) Created a number of different graphs on a 'Crazy Meter' dashboard.

[https://app.datadoghq.com/dash/297649/crazy-meter?live=true&page=0&is\\_auto=false&from\\_ts=1496425260196&to\\_ts=1496428860196&tile\\_size=m](https://app.datadoghq.com/dash/297649/crazy-meter?live=true&page=0&is_auto=false&from_ts=1496425260196&to_ts=1496428860196&tile_size=m)



Dashboard includes a number of different data types/representations including

- Current average crazy level across all reporting hosts

0.5195



1 Select your visualization

Timeseries **Query Value** Heat Map Distribution Top List Change Hostmap

2 Choose metrics and events [Graph Primer](#) [Share](#) [JSON](#) [Edit](#)

Get **Crazy.meter** from (everywhere) avg by (everything) +

Take the **Average** value from the displayed timeframe.

Units and formatting

Show: 4 decimals

☒ Autoscale EG 1000000 -> 1M

☐ Use Custom Units

Align:

Format according to these rules

|             |    |           |           |                  |  |
|-------------|----|-----------|-----------|------------------|--|
| If value is | <= | than 0.50 | show with | Green background |  |
| If value is | >  | than 0.50 | show with | Yellow text      |  |
| If value is | >  | than 0.75 | show with | Red background   |  |

3 Give your graph a title (or leave blank for suggested title)

Current Systemwide Crazy Level

[Cancel](#) [Preview](#) [Save](#)

b. Current Crazy level on all 'critical' hosts

0.53

1 Select your visualization

Timeseries **Query Value** Heat Map Distribution Top List Change Hostmap

2 Choose metrics and events [Graph Primer](#) [Share](#) [JSON](#) [Edit](#)

Get **Crazy.meter** from **criticality:high** x avg by (everything) +

Take the **Average** value from the displayed timeframe.

Units and formatting

Show: 2 decimals

☒ Autoscale EG 1000000 -> 1M

☐ Use Custom Units

Align:

Format according to these rules

|             |    |           |           |                  |  |
|-------------|----|-----------|-----------|------------------|--|
| If value is | <= | than 0.5  | show with | Green background |  |
| If value is | >= | than 0.51 | show with | Red background   |  |

[+ Add Rule](#)

3 Give your graph a title (or leave blank for suggested title)

Critical Resource Crazy level

[Cancel](#) [Preview](#) [Save](#)

c. Change in crazy value over last hour broken down by client facing vs. non-client facing hosts



no

+2%

yes

+4%

1

Select your visualization

Timeseries

Query Value

Heat Map

Distribution

Top List

Change

Hostmap

2

Choose metrics and events

Graph Primer

Share

JSON

Edit

Crazy.meter

Take the

Average

From hosts and tags

(Entire Infrastructure)

Break it down by

clientfacing

Compare to

an hour before

Show the

relative

change

Show

decreases

as better

Order by

change

ascending

Include present value

3

Give your graph a title (or leave blank for suggested title)

Client Facing Crazy vs last hour

Cancel

Preview

Save

d. Highest crazy value reported per region

0.73

emea

0.66

na

0.53

asiapac

1

Select your visualization

Timeseries

Query Value

Heat Map

Distribution

Top List

Change

Hostmap

2

Choose metrics and events

Graph Primer

Share

JSON

Edit

Crazy.meter

Take the

Max

From hosts and tags

(Entire Infrastructure)

Break it down by

geo

Sort by

Average

over time

Descending

limit to

10

items

Conditional Formatting (optional) You may use shorthand such as "1.5K" for "1500" in thresholds

If value is

<

than

0.53

show with

Green background

Else if

<

than

0.55

show with

Yellow background

Else if

<

than

0.6

show with

Red background

Add another condition

3

Give your graph a title (or leave blank for suggested title)

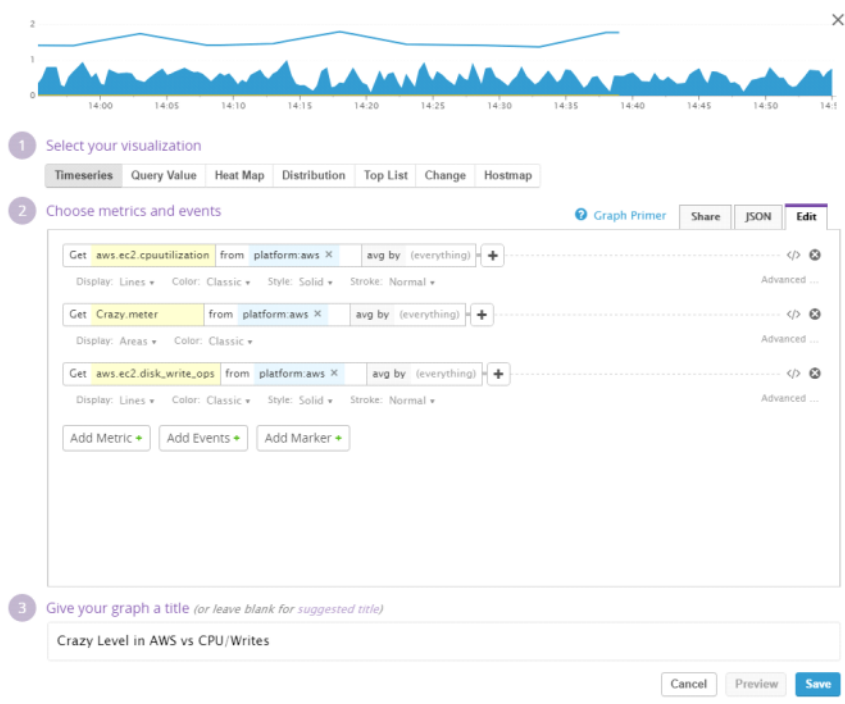
Highest Crazy Reported by Region

Cancel

Preview

Save

e. Crazy level in AWS hosts vs. CPU and number of writes for AWS hosts



f. Creating graphs and using appropriate tagging to automatically ensure new hosts are added to the appropriate visualizations

