

Imran Tell AWS Cloudwatch

1. Monitoring AWS ec2 instance CPU Utilization.

For this exercise we need an ec2 instance where we will cause high cpu load and test.

Create a centos 6 ec2 instance.

Install stress tool, which can create a high CPU usage.

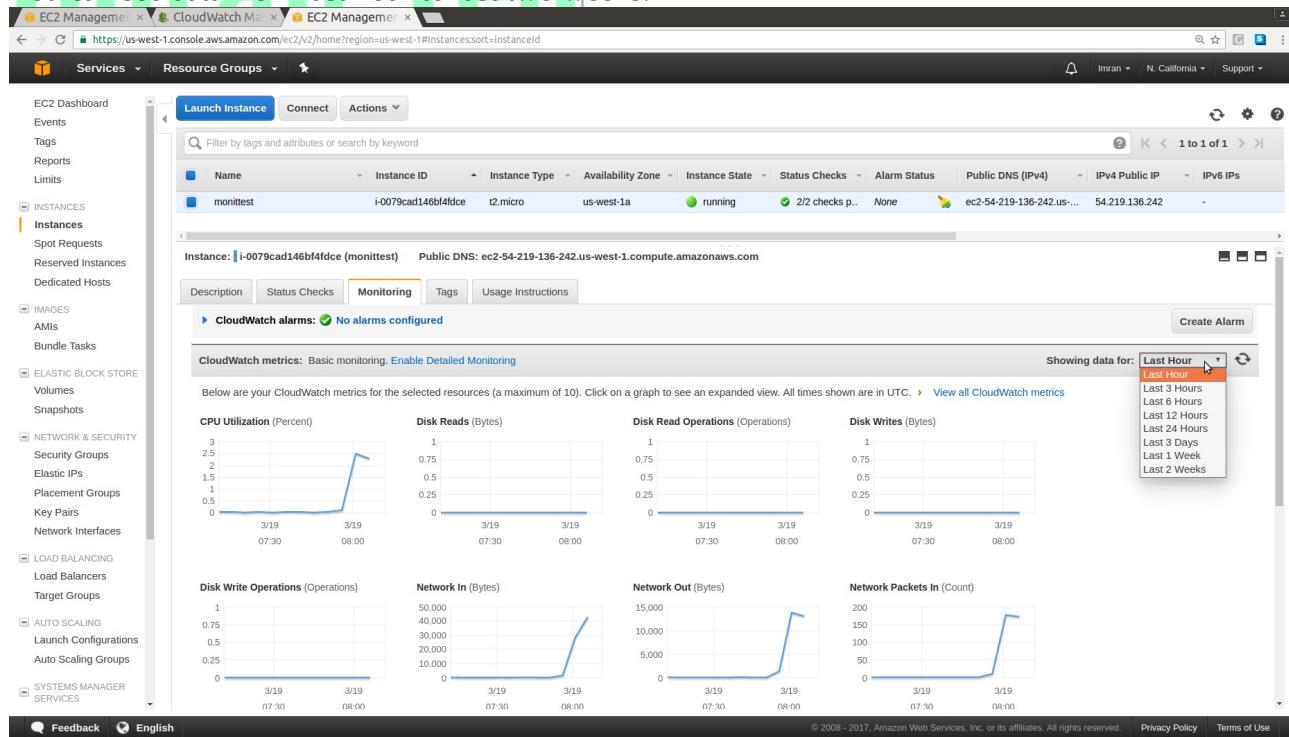
```
File Edit View Search Terminal Tabs Help
root@ip-172-31-4-16:~ imran@DevOps:~+
imran@DevOps:~/keys$ ssh -i TS-Ncalifornia.pem centos@54.219.136.242
Last login: Sun Mar 19 08:04:16 2017 from 183.82.216.42
[centos@ip-172-31-4-16 ~]$ sudo -i
[root@ip-172-31-4-16 ~]# yum install -y -q epel-release
[root@ip-172-31-4-16 ~]# yum install -y -q stress
[root@ip-172-31-4-16 ~]# stress
'stress' imposes certain types of compute stress on your system

Usage: stress [OPTION [ARG]] ...
 -?, --help      show this help statement
 --version     show version statement
 -v, --verbose   be verbose
 -q, --quiet     be quiet
 -n, --dry-run   show what would have been done
 -t, --timeout N timeout after N seconds
 --backoff N    wait factor of N microseconds before work starts
 -c, --cpu N     spawn N workers spinning on sqrt()
 -i, --io N      spawn N workers spinning on sync()
 -m, --vm N      spawn N workers spinning on malloc()/free()
 --vm-bytes B   malloc B bytes per vm worker (default is 256MB)
 --vm-stride B  touch a byte every B bytes (default is 4096)
 --vm-hang N    sleep N secs before free (default none, 0 is inf)
 --vm-keep       redirty memory instead of freeing and reallocating
 -d, --hdd N     spawn N workers spinning on write()/unlink()
 --hdd-bytes B  write B bytes per hdd worker (default is 1GB)

Example: stress --cpu 8 --io 4 --vm 2 --vm-bytes 128M --timeout 10s
```

I have created a ec2 instance with tag Name: monittest. We can check lots of metrics by selecting the instance monitoring tab. All those graphs for CPU,Disk & network usage is collected by Cloudwatch monitoring tool by default.

You can see data from last hour to last two weeks.



- Click on Cloudwatch service from AWS main dashboard.

Go to Metrics

Logs
Metrics NEW

Browse Metrics

Search Metrics X

Forums
Report an Issue

Alarm Summary

You do not have any alarms created in the US West (N. California) region. Alarms allow you to send notifications or execute Auto Scaling actions in response to any CloudWatch metric.

You can now use Amazon CloudWatch alarms to monitor the estimated charges on your AWS bill and receive email alerts whenever charges exceed a threshold you define. Visit the CloudWatch US East (N. Virginia) region to manage your billing alarms.

[Go to CloudWatch US East \(N. Virginia\) region](#)



Create Alarm

Service Health



Current Status	Details
Amazon CloudWatch Service	Service is operating normally View complete service health details

Imran Teli

- Click on Ec2

All metrics Graphed metrics Graph options

Q Search for any metric, dimension or resource id

1,024 Metrics

ApplicationELB	Auto Scaling	EBS	EC2
118 Metrics	16 Metrics	289 Metrics	456 Metrics
ELB	RDS	S3	
56 Metrics	85 Metrics	4 Metrics	

Imran Teli

All metrics Graphed metrics Graph options

All > EC2 Q Search for any metric, dimension or resource id

456 Metrics

By Auto Scaling Group	By Image (AMI) Id	Per-Instance Metrics
26 Metrics	14 Metrics	402 Metrics
Aggregated by Instance Type	Across All Instances	
7 Metrics	7 Metrics	

Imran Teli

Visualpath

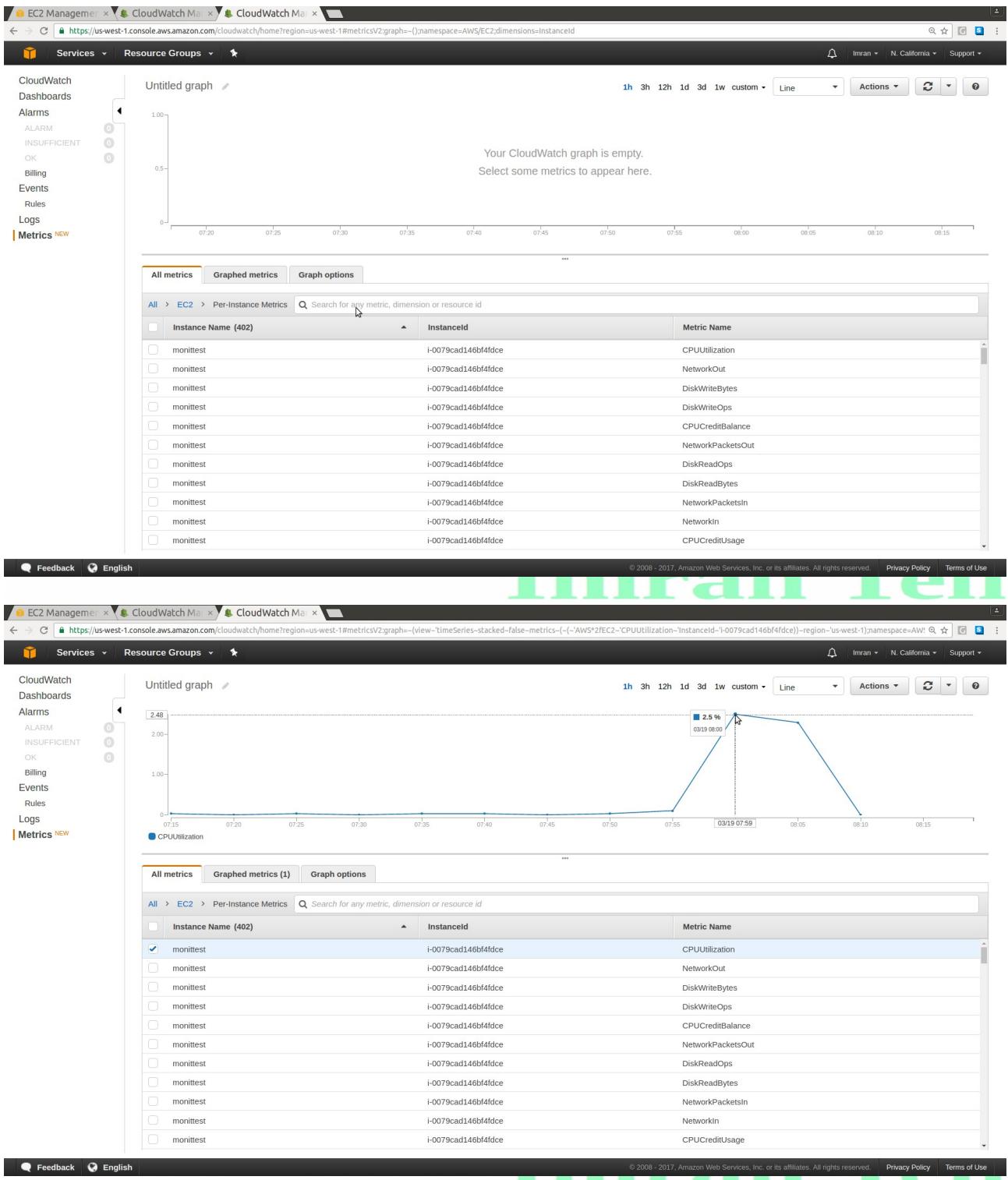
Imran Teli

Imran Teli

Visualpath

Imran Teli

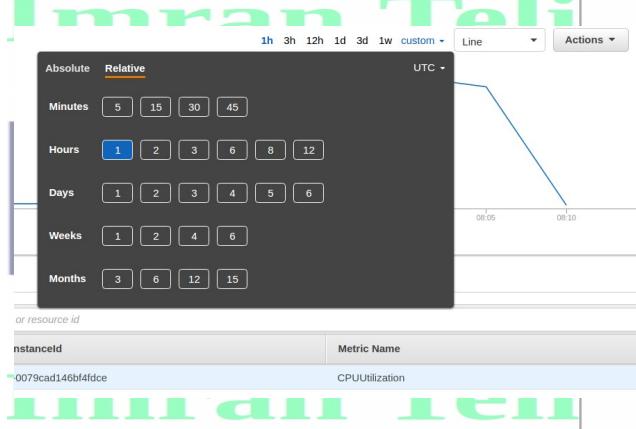
Imran Teli



As you can see above for instance monittest its showing cpu utilization graph for last 1 hour. We can customize the graph display.



- Click custom and select relative or absolute time to see graph details as per your wish.



Visualpath

Imran Teli

Visualpath

Setting up Alarm.

Click on Alarms ==> Create Alarm.

The screenshot shows the 'Create Alarm' dialog box. On the left, there's a sidebar with 'Services' (CloudWatch, Dashboards, Alarms), 'Resource Groups' (Create Alarm selected), and filters ('All alarms'). The main area shows the 'Create Alarm' dialog with two tabs: '1. Select Metric' (selected) and '2. Define Alarm'. The '1. Select Metric' tab includes a 'Search Metrics' input field and a 'Browse Metrics' dropdown. Below it is a section titled 'CloudWatch Metrics by Category' with links to ApplicationELB Metrics (118), Auto Scaling Metrics (16), EBS Metrics (289), EC2 Metrics (456), ELB Metrics (56), and RDS Metrics (85).

- Search with instance ID to find all the metrics related to our instance.

The screenshot shows the 'Create Alarm' dialog box again, but this time the search bar in the '1. Select Metric' tab is filled with the instance ID '0079cad146bf4fdce'. The results show a table of metrics for the specified instance:

InstanceId	InstanceName	Metric Name
i-0079cad146bf4fdce	monittest	CPUTCreditBalance
i-0079cad146bf4fdce	monittest	CPUTCreditUsage
i-0079cad146bf4fdce	monittest	CPUUtilization
i-0079cad146bf4fdce	monittest	DiskReadBytes
i-0079cad146bf4fdce	monittest	DiskReadOps
i-0079cad146bf4fdce	monittest	DiskWriteBytes
i-0079cad146bf4fdce	monittest	DiskWriteOps
i-0079cad146bf4fdce	monittest	NetworkIn

Imran Teli

- Put a check mark on CPUUtilization against monittest.

1. Select Metric
2. Define Alarm

The screenshot shows the AWS CloudWatch Metrics interface. At the top, there's a search bar with the ID '0079cad146bf4fdce'. Below it, several navigation tabs are visible: Per-Instance Metrics, By Auto Scaling Group, By Image (AMI) Id, Aggregated by Instance Type, and Across All Instances. The 'Per-Instance Metrics' tab is selected. The main area displays a table titled 'EC2 > Per-Instance Metrics'. The table has columns for InstanceId, InstanceName, and Metric Name. It lists three metrics for the instance 'i-0079cad146bf4fdce': CPUCreditBalance, CPUCreditUsage, and CPUUtilization. The 'CPUUtilization' row has a checked checkbox in the first column.

- Select next.

The screenshot shows the 'Create Alarm' wizard in the AWS CloudWatch Alarms section. The '2. Define Alarm' step is active. In the 'Alarm Threshold' section, the name is set to 'monitCPUHighLoad' and the description is 'Monitor High CPU Load for monitest instance.' The condition is defined as 'Whenever: CPUUtilization is >= 70 for 1 consecutive period(s)'. In the 'Actions' section, a notification is configured to send an email to 'imranteli0706@gmail.com' whenever the alarm goes off. The 'Alarm Preview' section shows a graph where a blue line spikes above a red threshold line at 70, indicating an alarm trigger. The preview also specifies a namespace of 'AWS/Ec2', instance ID 'i-0079cad146bf4fdce', instance name 'monitest', and metric name 'CPUUtilization'. The period is set to 5 minutes and the statistic to average.

- Click Create Alarm, once done.

You will get an email from AWS for verifying email address, once its verified you will start receiving email alert whenever the instance cpu load crosses beyond 70 %.

Imran Teli

Imran Teli

Visualpath

Imran Teli

Imran Teli

In

h
eli

- To test that we can use stress utility and cause high cpu on our instance.

```
[root@ip-172-31-4-16 ~]# uptime
08:43:43 up 2:47, 2 users, load average: 0.00, 0.00, 0.00
[root@ip-172-31-4-16 ~]# stress --cpu 90 -v --timeout 300
```

```
stress: dbug: [5638] --> hogcpu worker 10 [5719] forked
stress: dbug: [5638] using backoff sleep of 27000us
stress: dbug: [5638] setting timeout to 180s
stress: dbug: [5638] --> hogcpu worker 9 [5720] forked
stress: dbug: [5638] using backoff sleep of 24000us
```

- Login from another tab and run top command to see it real time from the system.

```
imran@DevOps:~/keys$ ssh -i TS-Ncalifornia.pem centos@54.219.136.242
Last login: Sun Mar 19 08:39:07 2017 from 183.82.216.42
[centos@ip-172-31-4-16 ~]$ sudo -i
[root@ip-172-31-4-16 ~]# top
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	19232	1488	1224	S	0.0	0.1	0:01.46	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.01	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	stopper/0
6	root	RT	0	0	0	0	S	0.0	0.0	0:00.01	watchdog/0
7	root	20	0	0	0	0	S	0.0	0.0	0:00.42	events/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	events/0
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	events_long/0
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	events_power_ef
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cgroup
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khelper
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	netns
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	async/mgr
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pm
16	root	20	0	0	0	0	S	0.0	0.0	0:00.00	xenwatch
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	xenbus
18	root	20	0	0	0	0	S	0.0	0.0	0:00.02	sync_supers
19	root	20	0	0	0	0	S	0.0	0.0	0:00.02	bdi-default
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kintegrityd/0
21	root	20	0	0	0	0	S	0.0	0.0	0:00.08	kblockd/0
22	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kacpid

Imran Teli

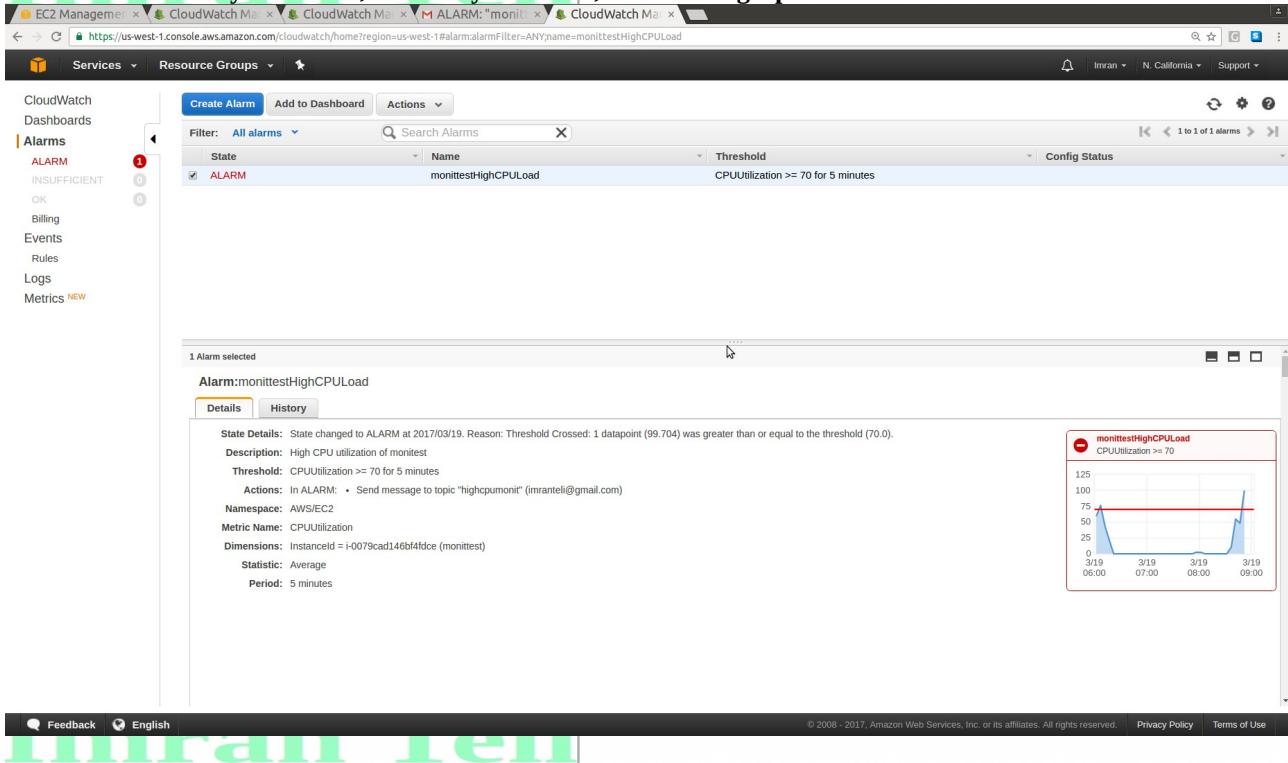
Imran Teli

- Observe the load average, its above 80 and all the processes are stress.

```
File Edit View Search Terminal Tabs Help
root@ip-172-31-4-16:~ root@ip-172-31-4-16:~
top - 08:50:47 up 2:54, 3 users, load average: 82.75, 37.89, 15.09
Tasks: 170 total, 91 running, 79 sleeping, 0 stopped, 0 zombie
Cpu(s): 99.7%us, 0.3%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1018504k total, 288148k used, 730356k free, 10732k buffers
Swap: 0k total, 0k used, 0k free, 161204k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
5639 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5640 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5641 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5642 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5643 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5644 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5645 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5646 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5647 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5648 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5649 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5650 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5651 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5652 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5653 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5654 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5655 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5656 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5657 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5658 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5659 root 20 0 6516 188 100 R 1.0 0.0 0:01.67 stress
5660 root 20 0 6516 188 100 R 1.0 0.0 0:01.70 stress
```

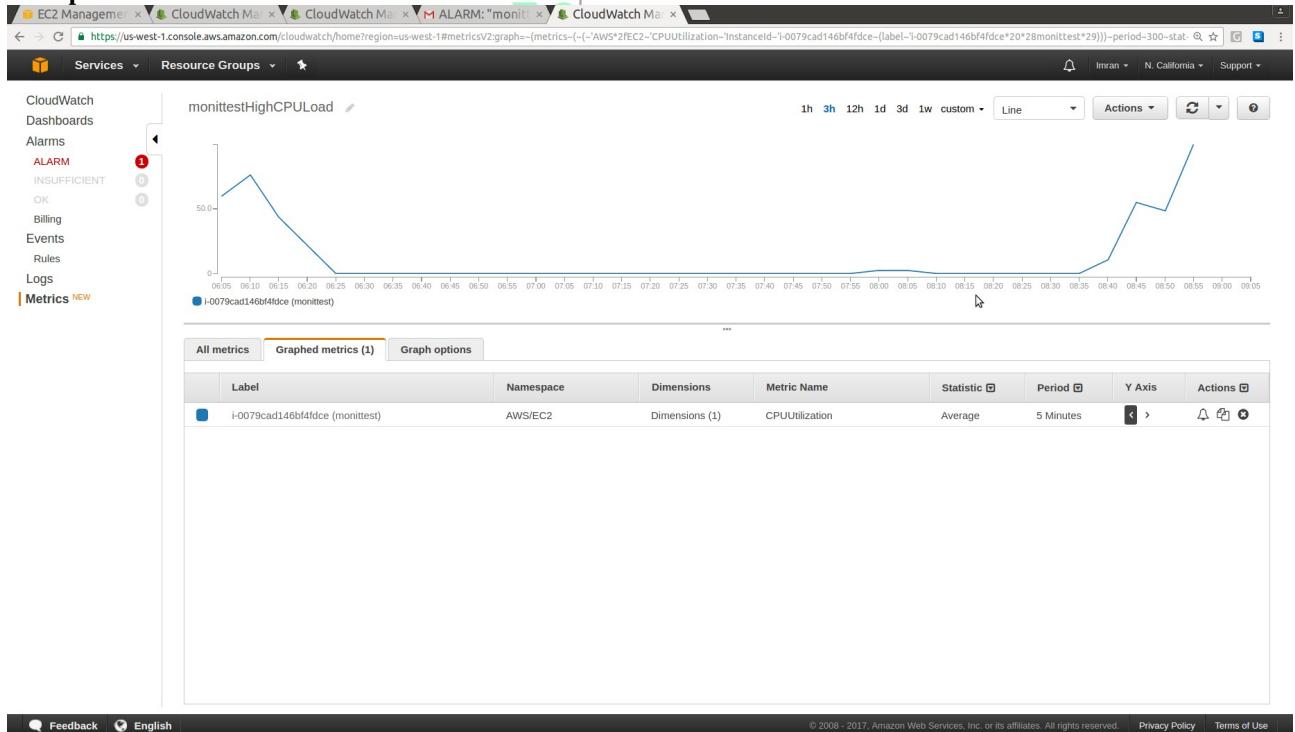
- After few minutes you will see, email in your inbox, Alarm and graph in red.



- Email from AWS



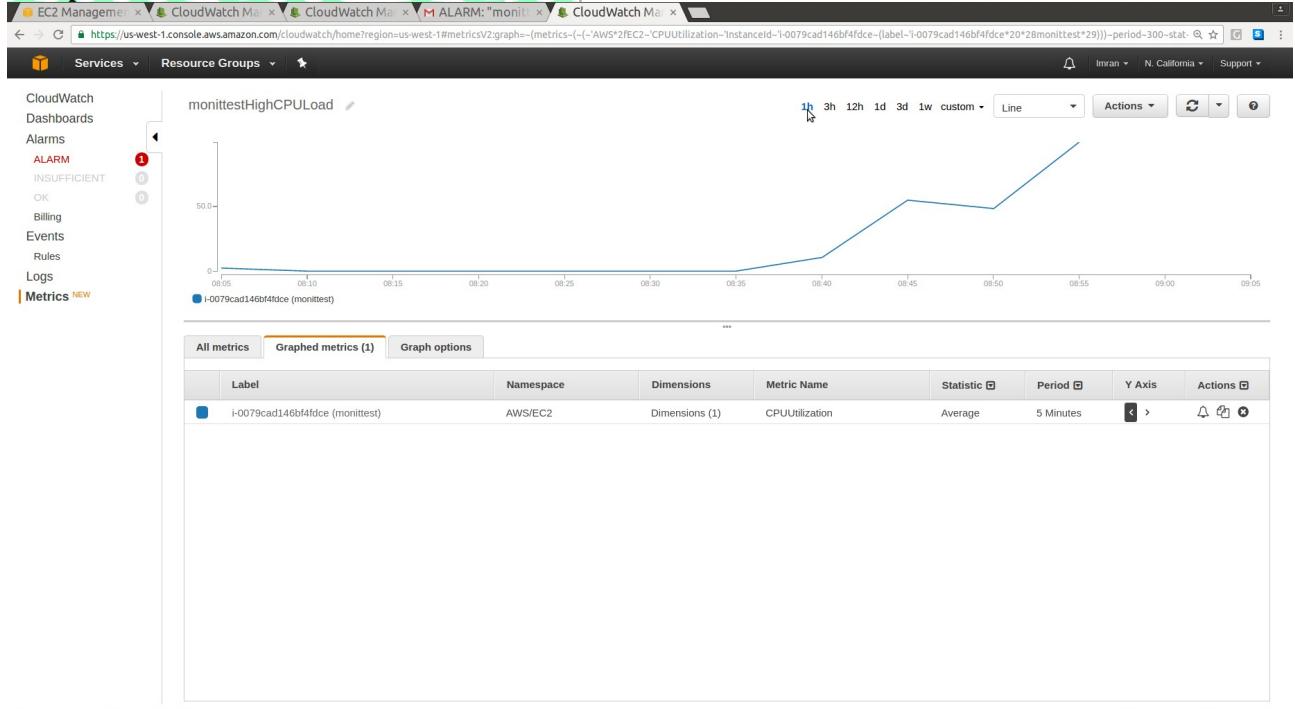
- Graph for our CPU Utilization last 3 hours.



Feedback English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Graph for our CPU Utilization last 1 hour.



Feedback English

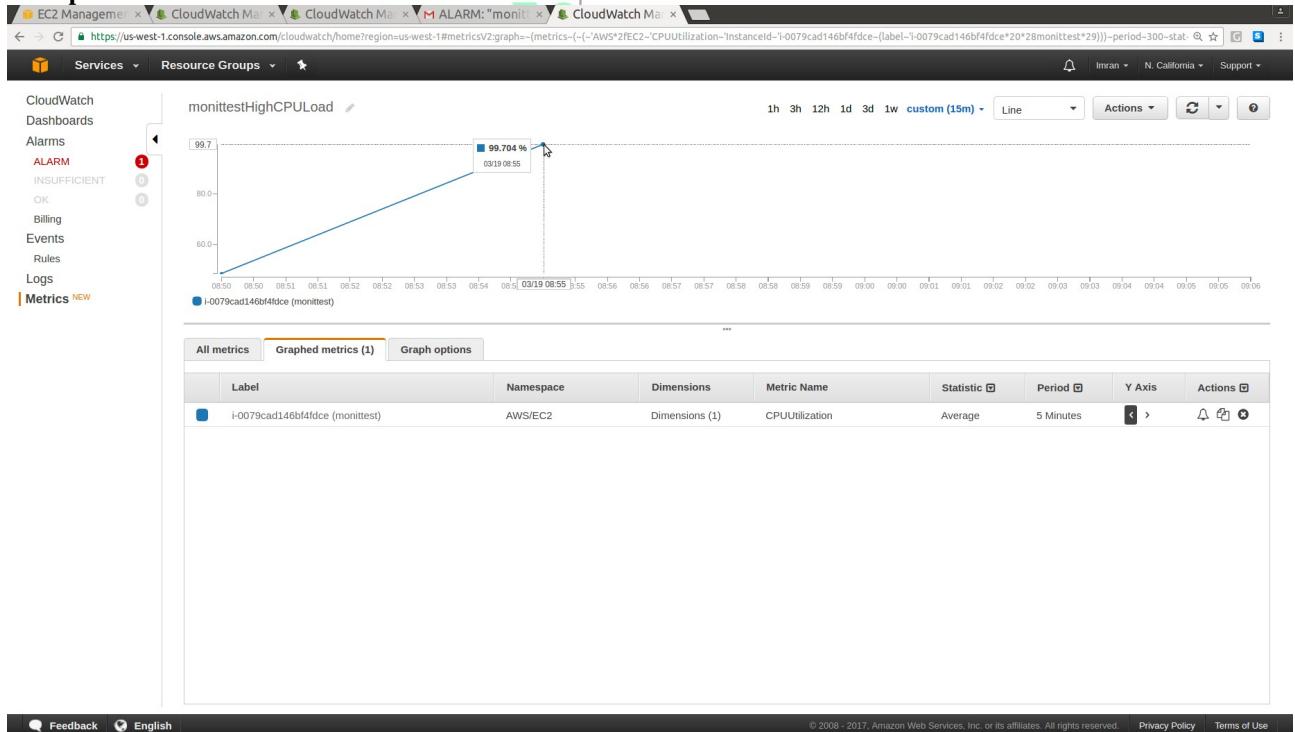
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Visualpath

Imran Teli

Imran Teli

- Graph for our CPU Utilization last 15 minutes.



Feedback English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Imran Teli

- You can also change/add the actions of alarms like EC2 actions to stop/reboot/terminate instance.

The screenshot shows the 'Modify Alarm' dialog for the alarm 'monittestHighCPULoad'. The '2. Define Alarm' tab is selected. The 'Whenever' section shows a threshold of 70% for 1 consecutive period. The 'Actions' section contains two parts: a 'Notification' section where an email address 'imran@gmail.com' is listed, and an 'EC2 Action' section where the action is set to 'Reboot this instance'. The background shows a list of alarms and a metrics graph for the same instance and metric.



2. Monitoring AWS Billing.

- Go to My Billing Dashboard ==> Preferences ==> Put a check mark on Receive billing alerts.

The screenshot shows the 'Preferences' page for AWS Billing. On the left, there's a sidebar with links like Dashboard, Bills, Cost Explorer, Budgets, Reports, Cost Allocation Tags, Payment Methods, Payment History, Consolidated Billing, and Preferences (which is currently selected). The main content area is titled 'Preferences' and contains three sections: 'Receive PDF Invoice By Email', 'Receive Billing Alerts', and 'Receive Billing Reports'. The 'Receive Billing Alerts' section has a checked checkbox and a descriptive text about monitoring AWS usage charges. Below it, there's a field to 'Save to S3 Bucket:' with a 'Verify' button and a 'Save preferences' button at the bottom.

The screenshot shows the 'Billing Alarms' page in the AWS CloudWatch Metrics service. The left sidebar has links for CloudWatch, Dashboards, Alarms (which is selected), ALARM, INSUFFICIENT, OK, Events, Rules, Logs, and Metrics. The main content area is titled 'Billing Alarms' and contains text about how CloudWatch can help monitor AWS bills via email alerts. It includes a 'Create Alarm' button and a link to learn more about billing alerts. The right side has an 'Additional Info' section with links to Getting Started Guide, Monitoring Scripts Guide, Overview and Features, Documentation, Forums, and Report an Issue. The footer includes links for Feedback, English, Privacy Policy, and Terms of Use.

Imran Teli

Visualpath

- Create Alarm.

The screenshot shows the 'Create Alarm' dialog for a 'Billing Alarm'. The left sidebar shows 'CloudWatch Metrics' selected under 'Alarms'. The main area is titled 'Billing Alarm' with the sub-section 'When my total AWS charges for the month exceed: \$ 250 USD send a notification to: imran@gmail.com'. A reminder message states: 'Reminder: for each address you add, you will receive an email from AWS with the subject "AWS Notification - Subscription Confirmation". Click the link provided in the message to confirm that AWS may deliver alerts to that address.' Below this are 'showing simple options' and 'show advanced' links. On the right, the 'Alarm Preview' section shows a graph titled 'EstimatedCharges > 250' with a red horizontal line at the 250 mark. The Y-axis ranges from 0 to 300, and the X-axis shows dates from 3/14 to 3/18. A single data point is plotted at approximately 270 on 3/18. The 'Additional Info' sidebar includes links to 'Getting Started Guide', 'Monitoring Scripts Guide', 'Overview and Features', 'Documentation', 'Forums', and 'Report an Issue'. At the bottom are 'Cancel', 'Previous', 'Next', and a blue 'Create Alarm' button.

- View Alarm.

The screenshot shows the CloudWatch Metrics service interface after creating the 'BillingAlarm'. The left sidebar shows 'CloudWatch Metrics' selected under 'Alarms'. A green success message box says 'Your alarm BillingAlarm has been saved.' The main area displays a table of alarms with one entry: 'State: OK', 'Name: BillingAlarm', and 'Threshold: EstimatedCharges > 250 for 6 hours'. Below the table, a detailed view for 'BillingAlarm' shows 'State Details: State changed to OK at 2017/03/19. Reason: Threshold Crossed: 1 datapoint (0.72) was not greater than the threshold (250.0.)'. It lists 'Description: In ALARM: • Send message to topic "NotifyMe" (imranteli@gmail.com)', 'Namespace: AWS/Billing', 'Metric Name: EstimatedCharges', 'Dimensions: Currency = USD', 'Statistic: Maximum', and 'Period: 6 hours'. To the right is a preview chart for 'BillingAlarm' showing 'EstimatedCharges > 250' with a red threshold line at 250 and a single data point at approximately 270. The bottom of the page includes 'Feedback', 'English', 'Privacy Policy', and 'Terms of Use' links.

Imran Teli

