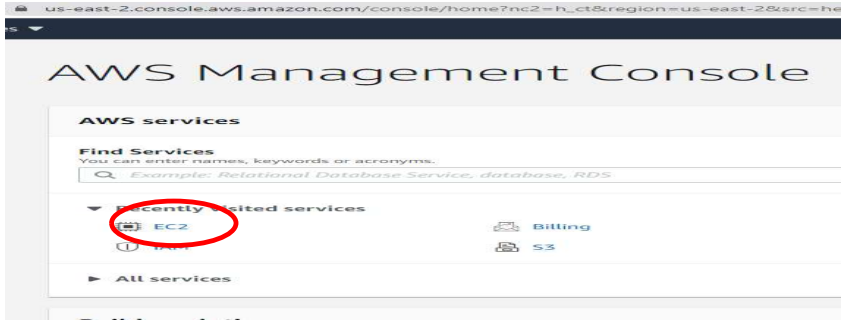
DevOps Monitoring Challenge

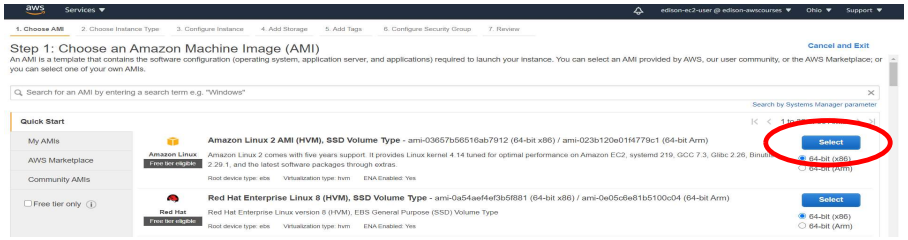
Edison Chukwuemeka

Setup EC2 Instances

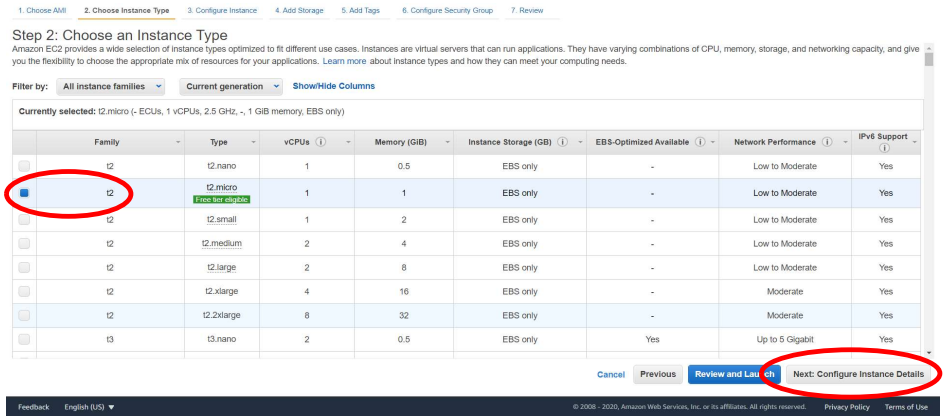
- Log into AWS console: aws.amazon.com
- Select the EC2 service as shown below.



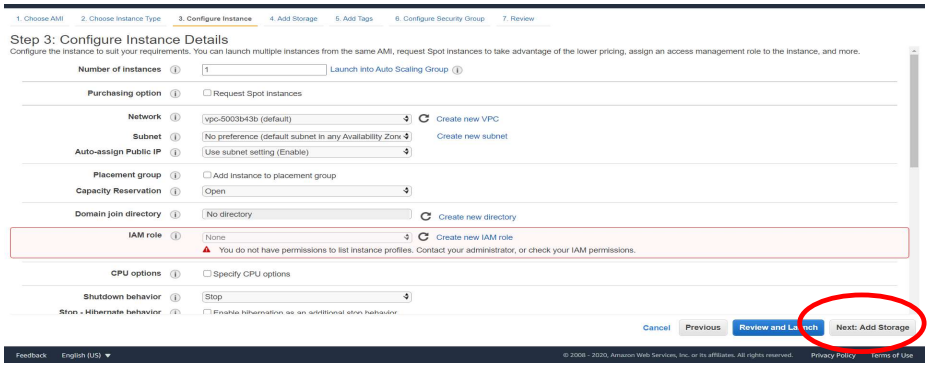
- Select the Amazon Linux 2 AMI SSD Volume machine.



- Select t2.micro instance and click next.

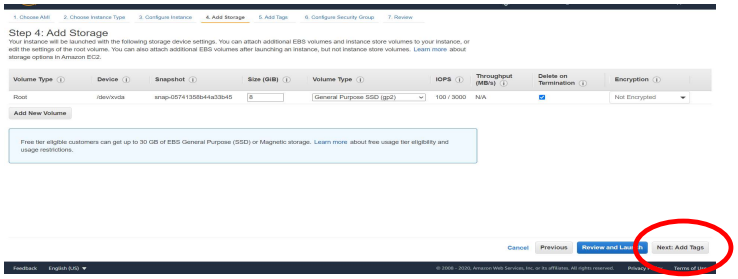


- Configure the EC2 Instance.

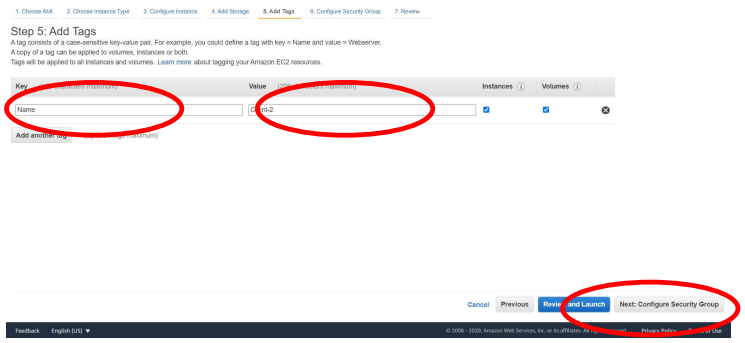


Setup EC2 Instances

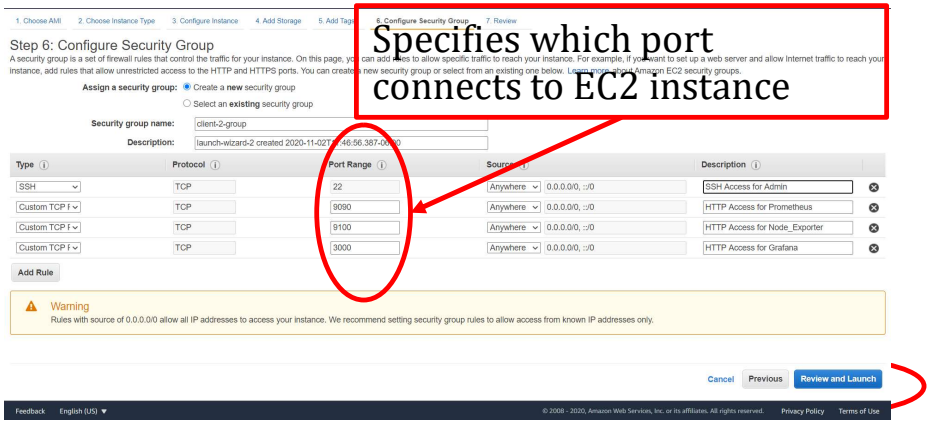
- Add Storage



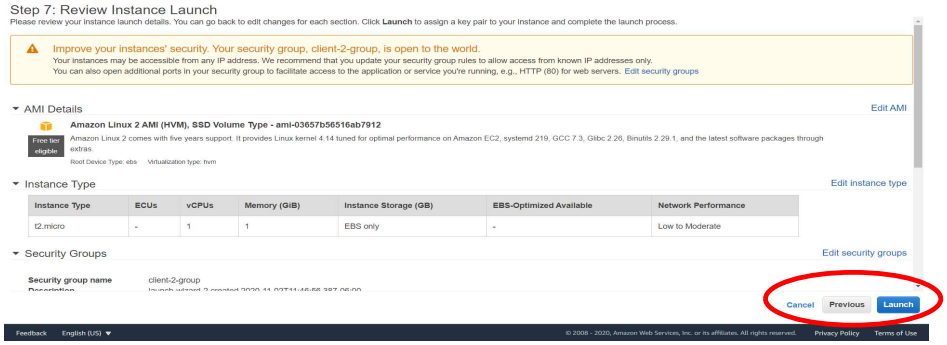
- Add Name tag to the EC2 Instance



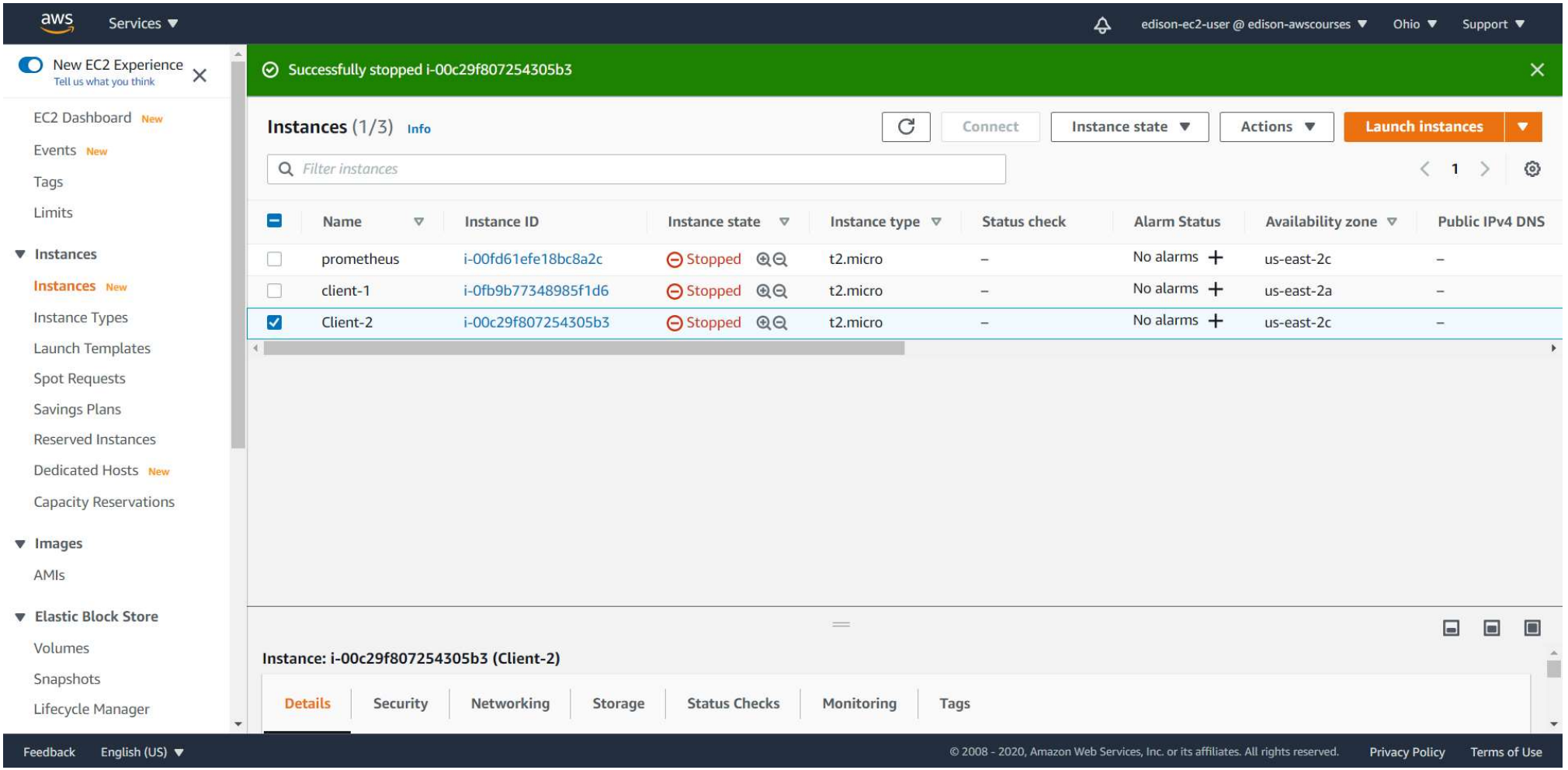
- Configure the Security group



- Review the details of the EC2 instance and launch the instance.



Setup EC2 Instances

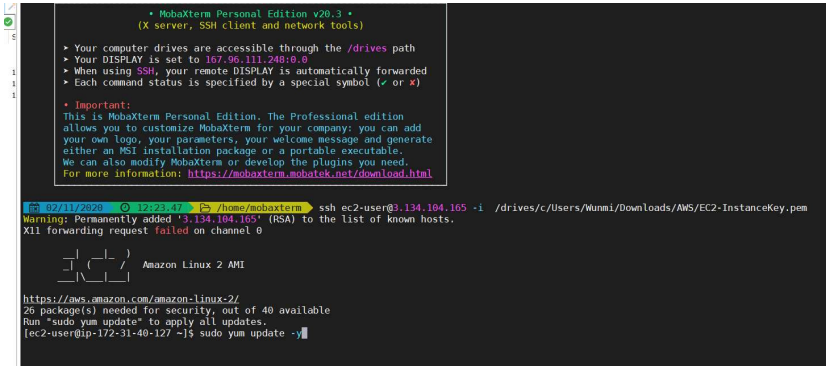


Configure Prometheus to monitor 2 EC2 instances

Configure Prometheus to monitor client-2:

- ssh to client-2
- Update the server:

$ sudo yum update -y



- Create a user - node_exporter:

$ sudo useradd --no-create-home --shell /bin/false node_exporter

- Create a user - prometheus:

$ sudo useradd --no-create-home --shell /bin/false Prometheus
$ sudo mkdir /etc/Prometheus
$ sudo mkdir /var/lib/Prometheus
$ sudo chown prometheus:prometheus /etc/Prometheus
$ sudo chown prometheus:prometheus /var/lib/Prometheus

- Download prometheus:

$ curl -LO
https://github.com/prometheus/prometheus/releases/download/v2.22.0/prometheus-2.22.0.linux-amd64.tar.gz

- Validate the downloaded file:

$ sha256sum prometheus-2.22.0.linux-amd64.tar.gz

- Unpack the downloaded file:

$ tar xvf prometheus-2.22.0.linux-amd64.tar.gz

- Copy the binary file to /usr/local/bin and set the user and group ownership to Prometheus user:

$ sudo cp prometheus-files/prometheus /usr/local/bin/
$ sudo cp prometheus-files/promtool usr/local/bin/

- Download node_exporter:

$ curl -LO
https://github.com/prometheus/node_exporter/releases/download/v1.0.1/node_exporter-1.0.1.linux-amd64.tar.gz

- Validate the downloaded file:

$ sha256sum node_exporter-1.0.1.linux-amd64.tar.gz

- Unpack the downloaded file:

$ tar xvf node_exporter-1.0.1.linux-amd64.tar.gz

- Copy the binary file to /usr/local/bin and set the user and group ownership to node_exporter user:

$ sudo cp node_exporter-1.0.1.linux-amd64/node_exporter /usr/local/bin/
$ sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter

Configure Prometheus to monitor 2 EC2 instances

- Create prometheus.service file:

$ sudo nano /etc/systemd/system/prometheus.service

```
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
   --config.file /etc/prometheus/prometheus.yml \
   --storage.tsdb.path /var/lib/prometheus/ \
   --web.console.templates=/etc/prometheus/consoles \
   --web.console.libraries=/etc/prometheus/console_libraries

[Install]
WantedBy=multi-user.target
```

- Create node_exporter.service file:

$ sudo nano /etc/systemd/system/node_exporter.service

```
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

[Service]
User=node_exporter
Group=node_exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=multi-user.target
```

Configure Prometheus to monitor 2 EC2 instances

- Configure Prometheus to scrape data from node_exporter on the EC2 instances.

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9100']
  - job_name: 'client-1'
    scrape_interval: 5s
    static_configs:
      - targets: ['172.31.15.4:9100']
  - job_name: 'client-2'
    scrape_interval: 5s
    static_configs:
      - targets: ['172.31.40.127:9100']
```

- Reload systemd:

$ sudo systemctl daemon-reload

- Start Prometheus service

$ sudo systemctl start Prometheus

- Check the status of Prometheus service.

$ sudo systemctl status Prometheus

- Enable Prometheus service.

$ sudo systemctl enable prometheus

- Start the node_exporter service:

$ sudo systemctl start node_exporter

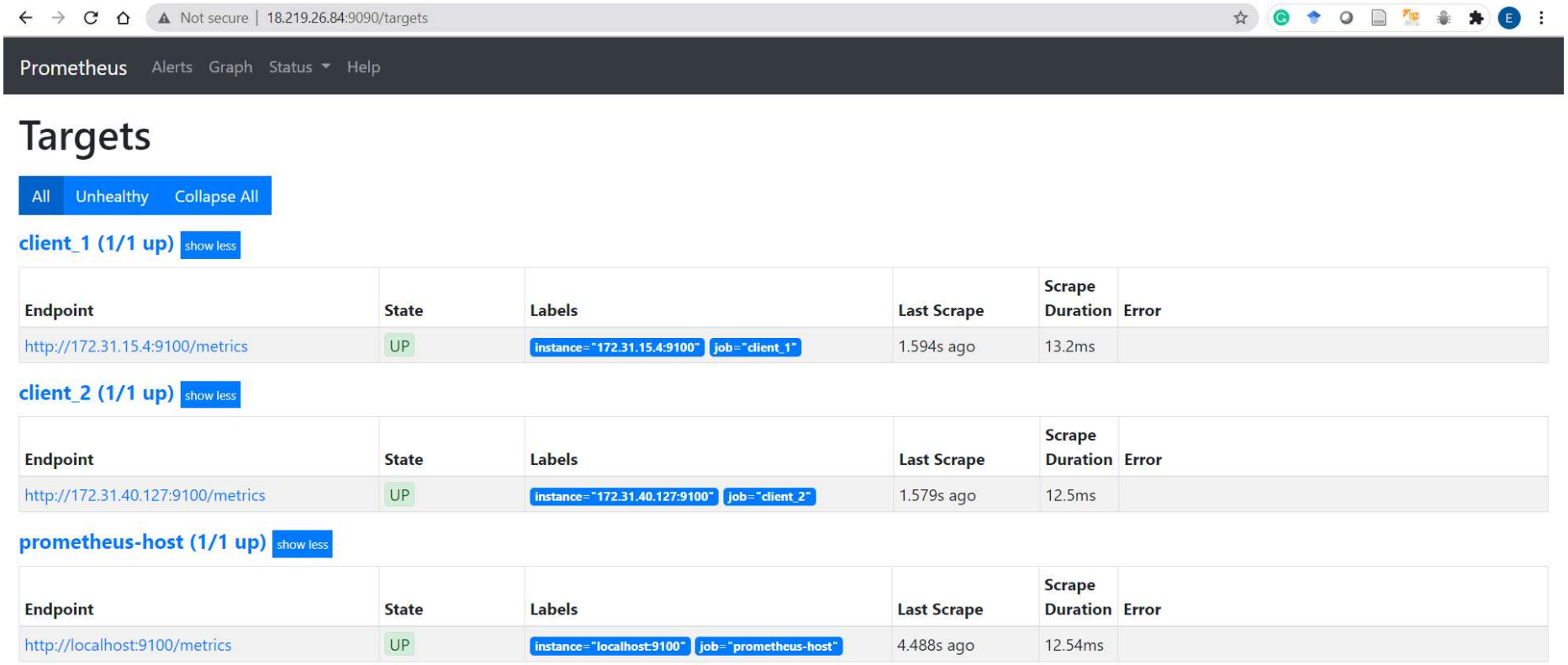
- Check the status of the service:

$ sudo systemctl status node_exporter

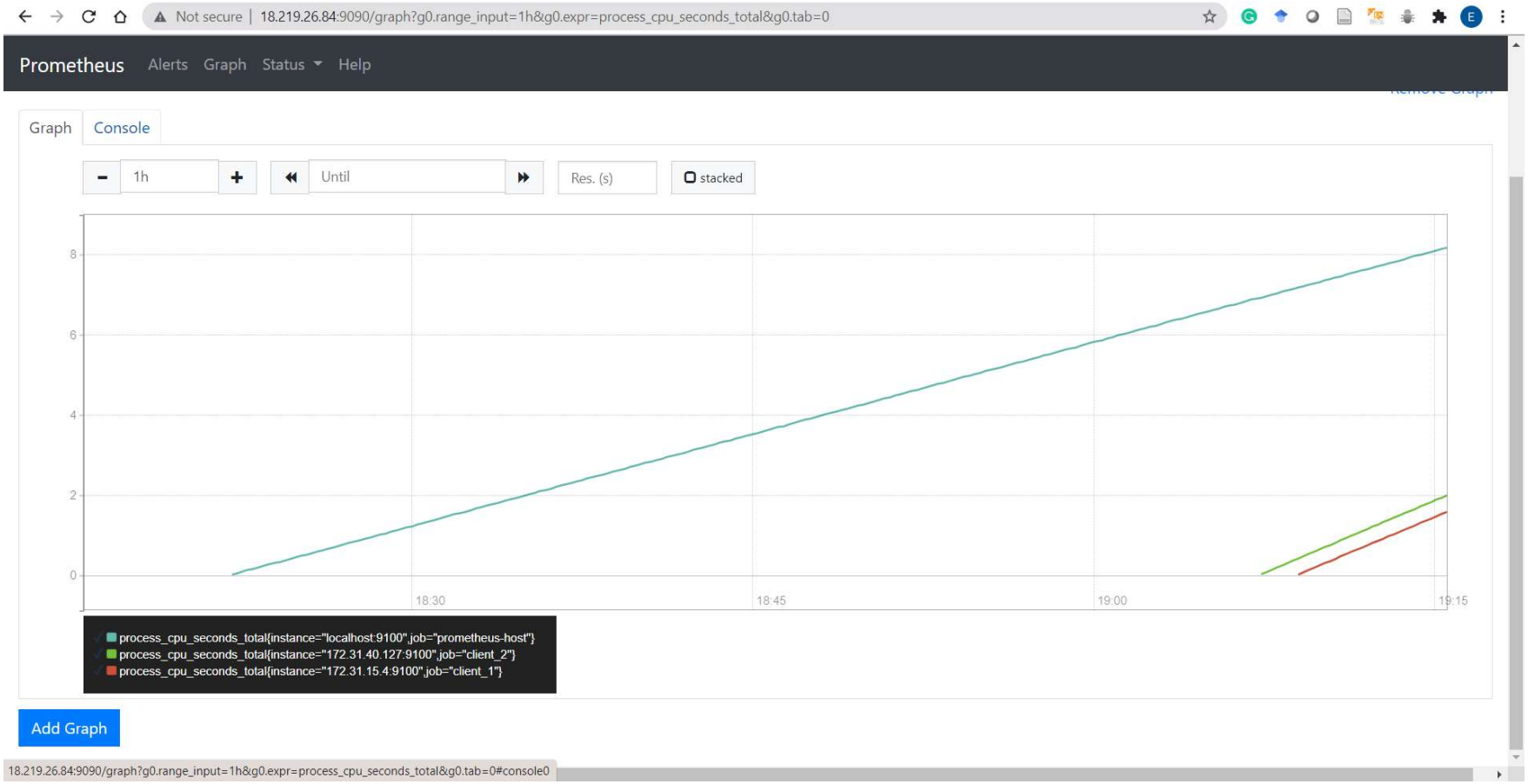
- Enable node_exporter:

$ sudo systemctl enable node_exporter

Monitor 2 EC2 Instances through Prometheus



Monitor 2 EC2 Instances through Prometheus



Connect Prometheus to Grafana

Install Grafana on the control machine – Prometheus-host

- SSH to the EC2 Instance – Prometheus-host:

$ ssh ec2-user@18.219.26.84 -i /drives/c/Users/Wunmi/Downloads/AWS/EC2-InstanceKey.pem

- Update installed packages

$ sudo yum update -y

- Add a new yum repo for Grafana called grafana.repo

$ sudo nano /etc/yum.repos.d/grafana.repo

[grafana]

name=grafana

baseurl=https://packages.grafana.com/oss/rpm

repo_gpgcheck=1

enabled=1

gpgcheck=1

gpgkey=https://packages.grafana.com/gpg.key

sslverify=1

sslcacert=/etc/pki/tls/certs/ca-bundle.crt

- Install Grafana:

$ sudo yum install Grafana –y

- Reload system:

$ sudo systemctl daemon-reload

- Start Grafana Server:

$ sudo systemctl start grafana-server

- Check the status of Grafana Server:

$ sudo systemctl status grafana-server

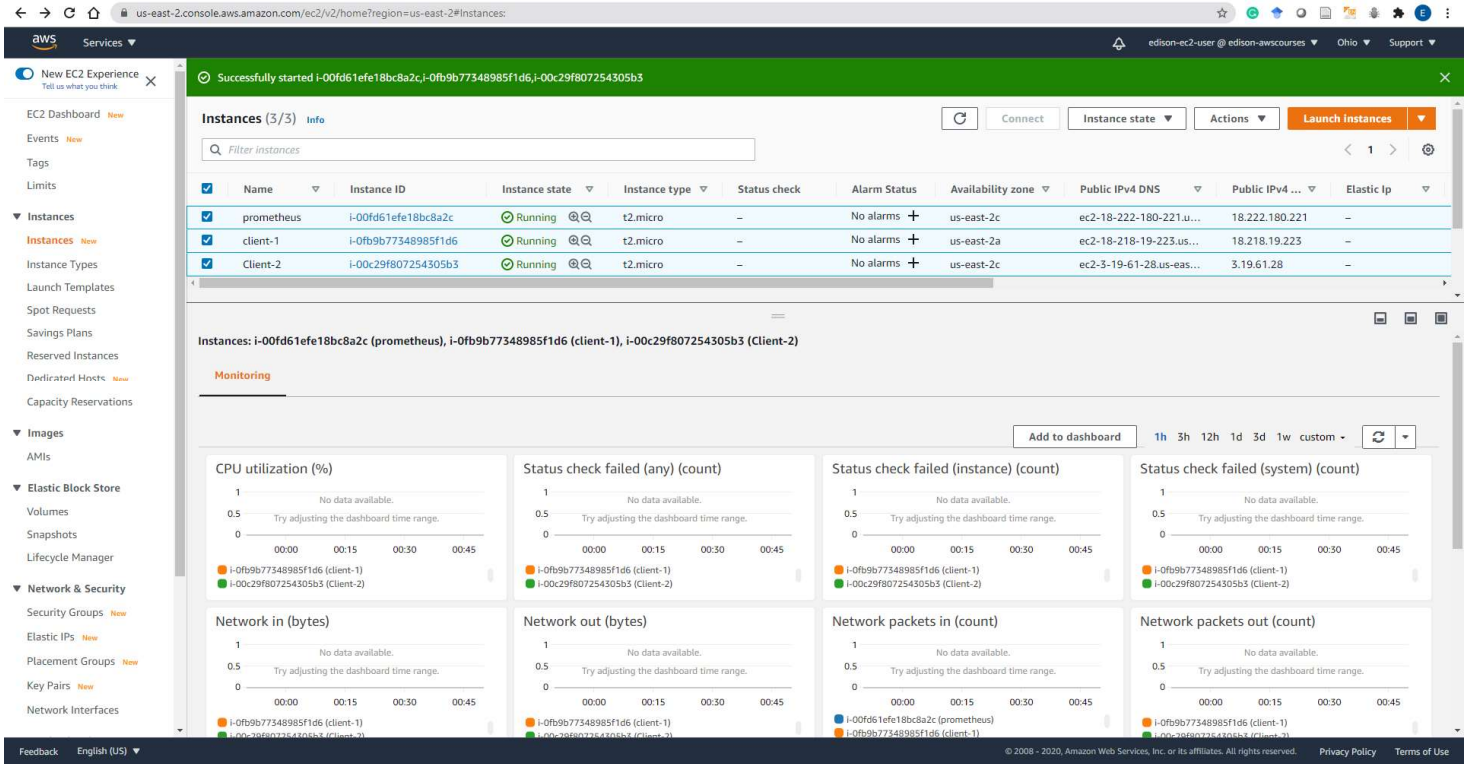
- Enable Grafana start upon booting the EC2 instance

$ sudo systemctl enable grafana-server.service

Connect Prometheus to Grafana

Connect to Grafana Server:

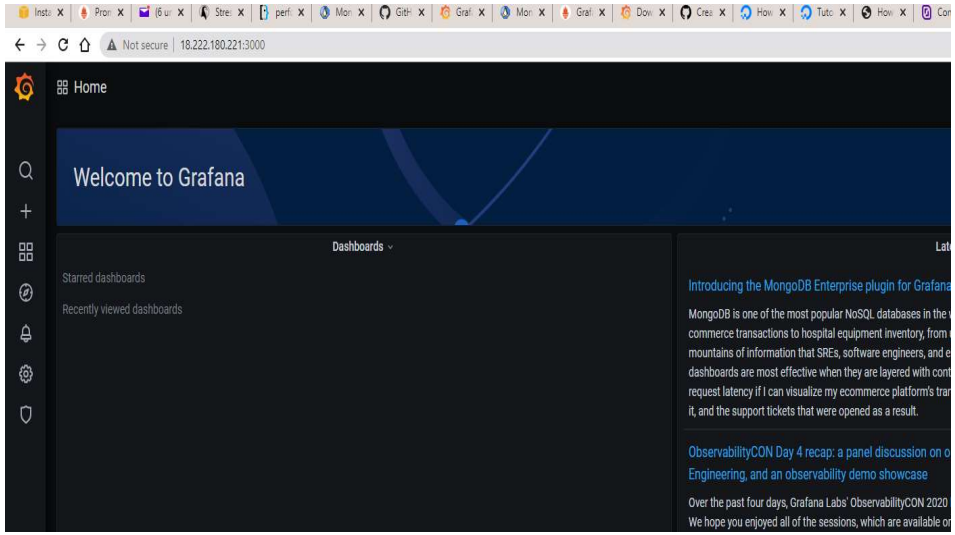
➢ Launch the EC2 instances:



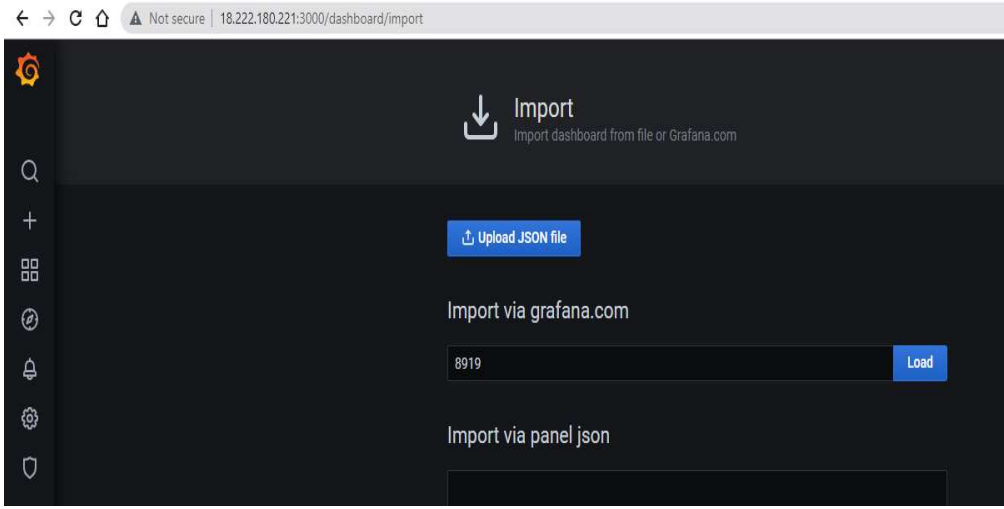
Connect Prometheus to Grafana

- Connect to Grafana using the pubic ip of the host machine with port 3000

$ 18.222.180.221:3000

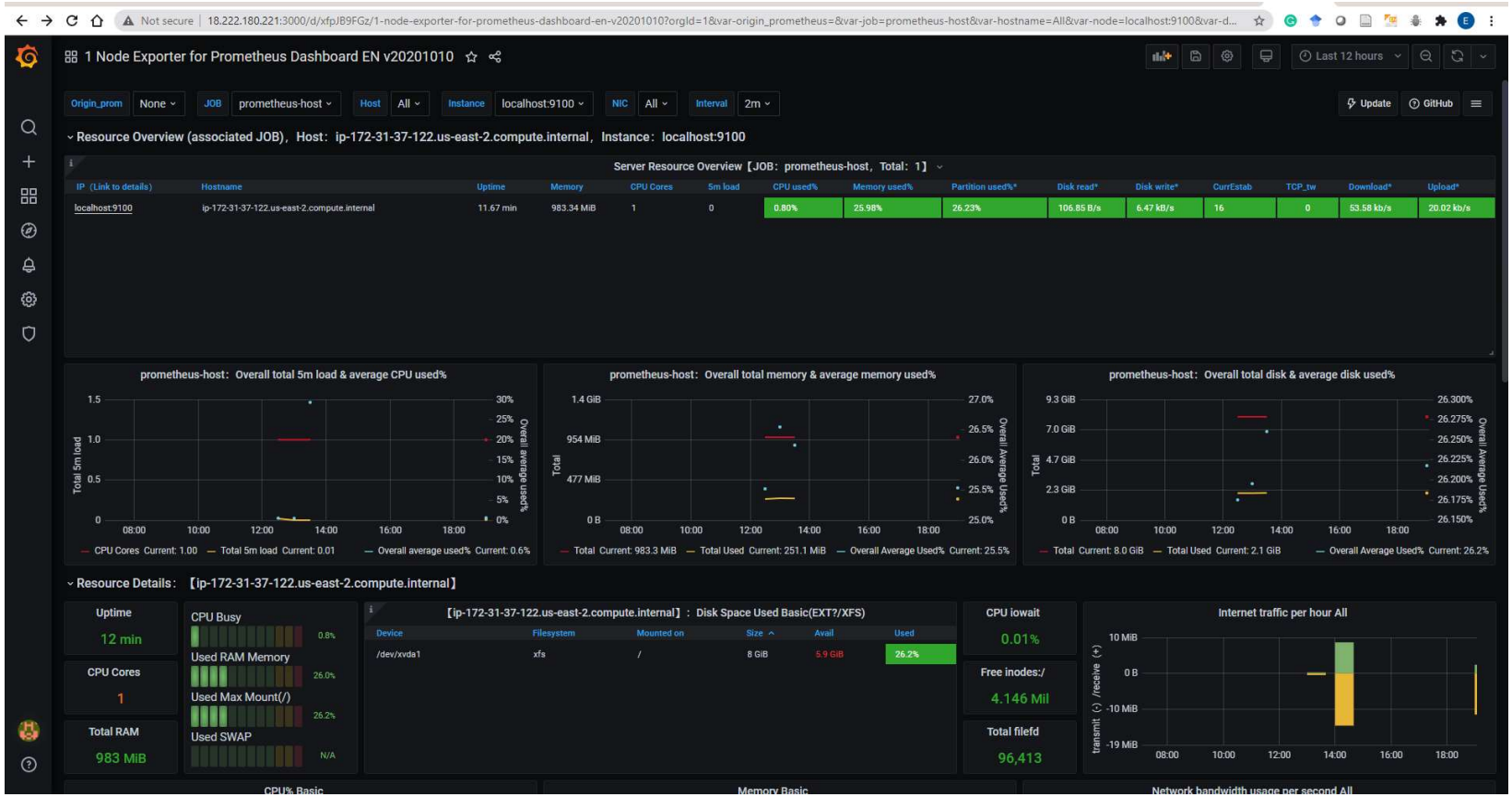


- Load a Dashboard



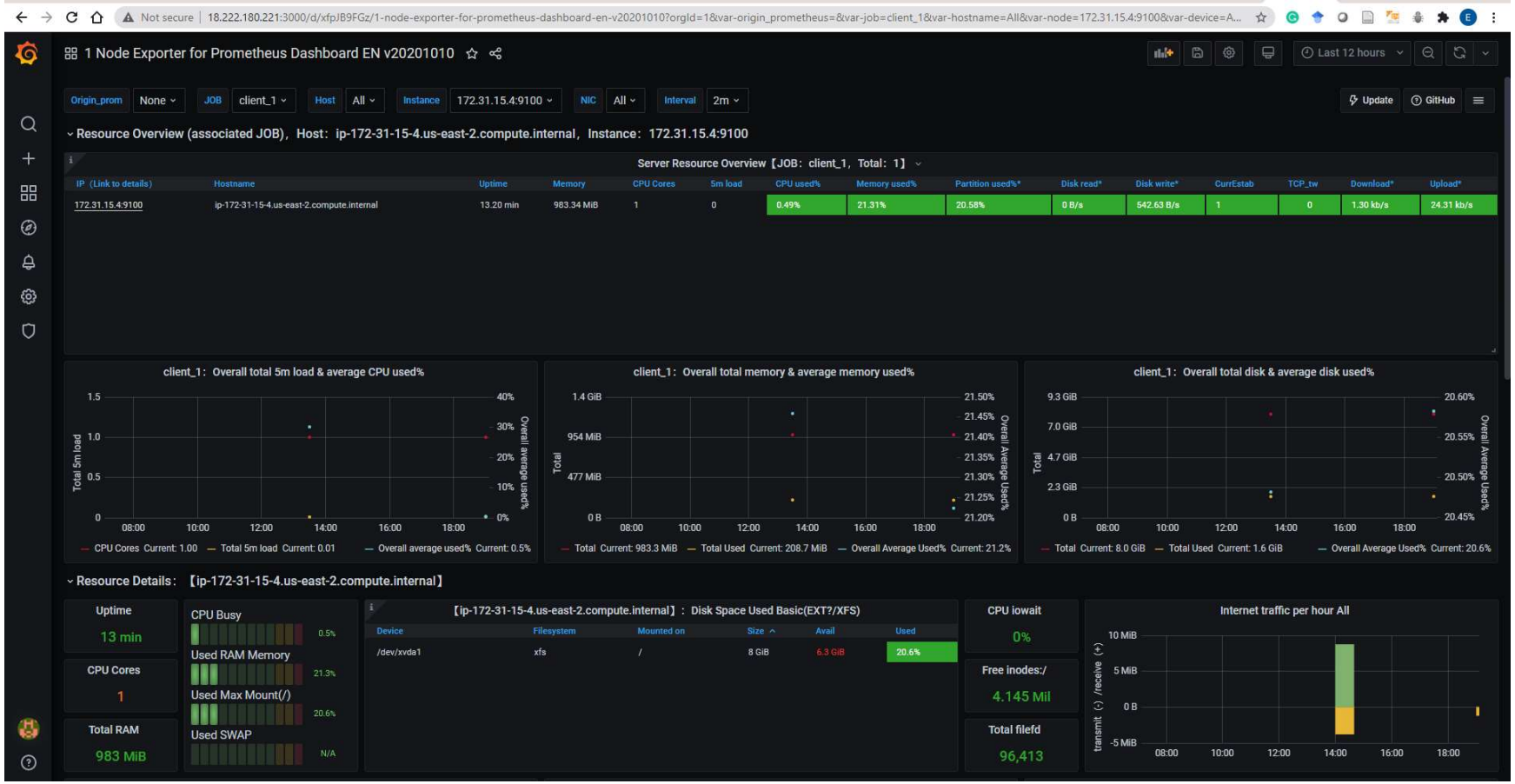
Collect and Display cpu, memory and disk usage

Prometheus-host server



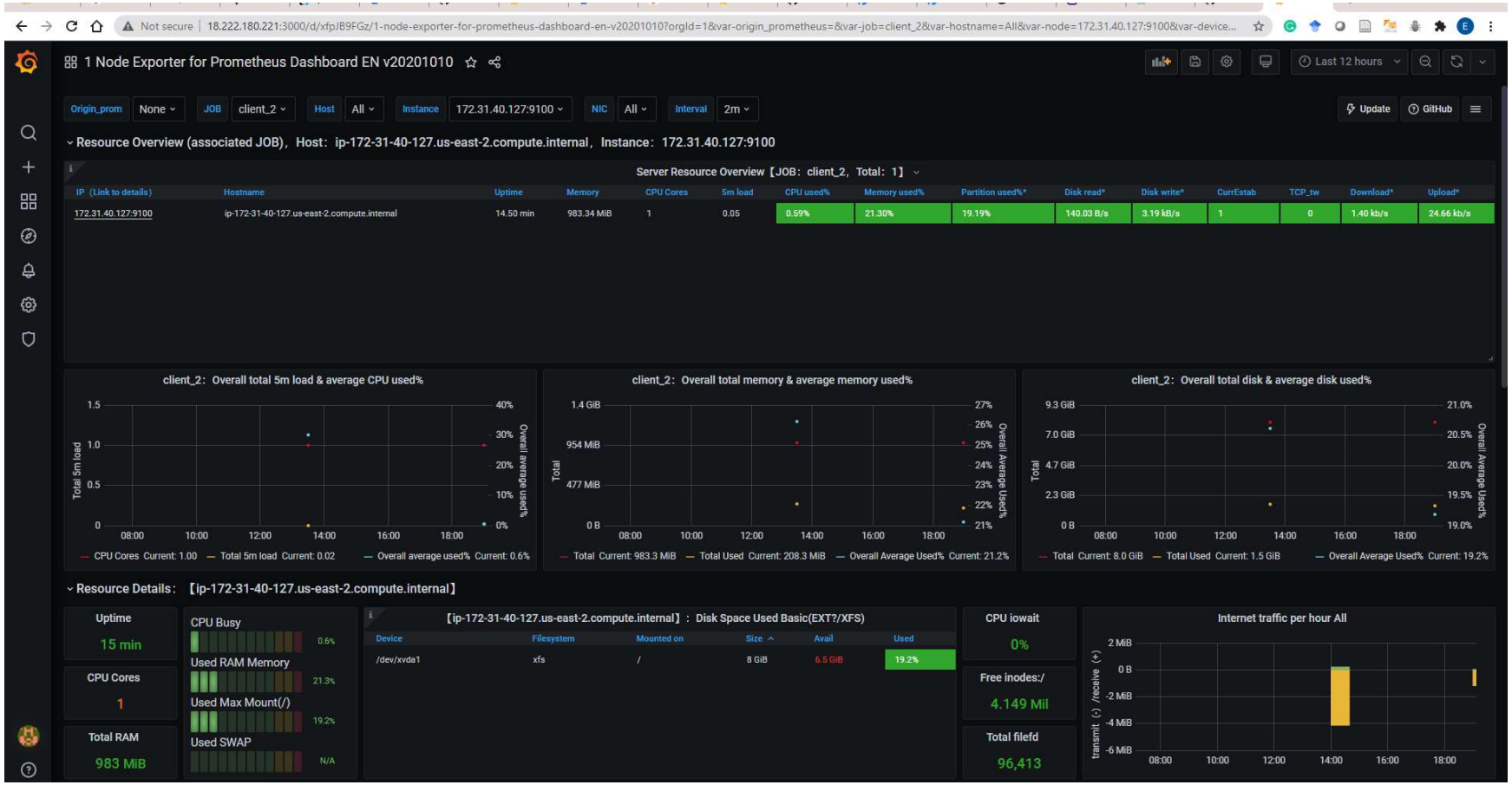
Collect and Display cpu, memory and disk usage

Client-1 server



Collect and Display cpu, memory and disk usage

Client-2 server



Simulate high cpu, memory, and disk usage on monitored instances

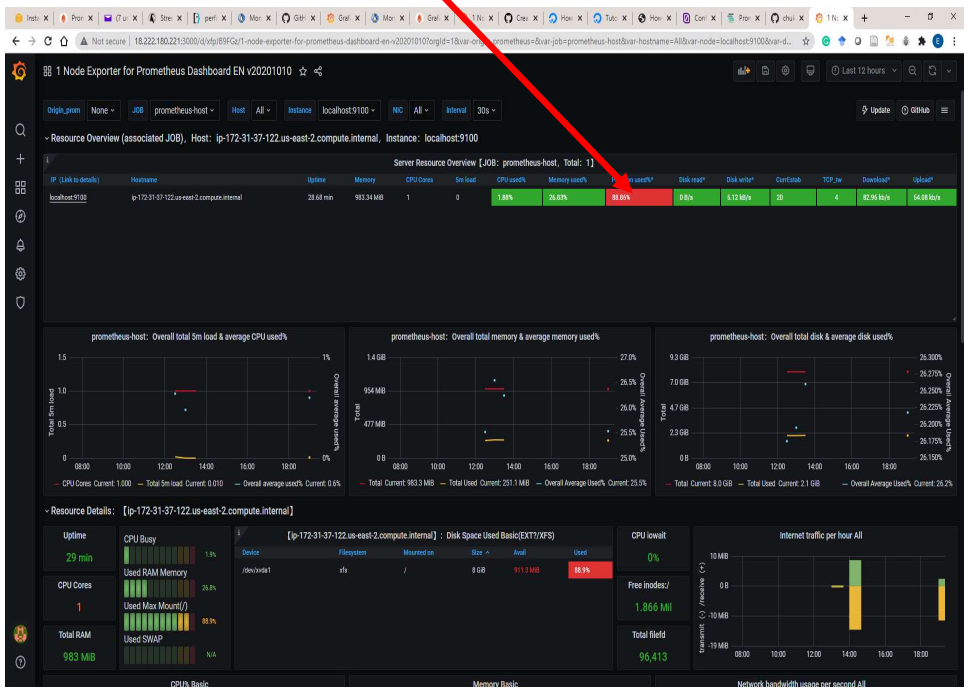
Prometheus-Host Server – Simulate Disk usage

```
[ec2-user@ip-172-31-37-122 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M     0  474M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  448K  492M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  2.1G  5.9G  27% /              ==============➔ Before running the test
tmpfs            99M     0   99M   0% /run/user/1000
[ec2-user@ip-172-31-37-122 ~]$ fallocate -l 5G test-disk
[ec2-user@ip-172-31-37-122 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M     0  474M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  448K  492M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  7.1G  912M  89% /       ================➔ After Running the test
tmpfs            99M     0   99M   0% /run/user/1000
[ec2-user@ip-172-31-37-122 ~]$ rm test-disk
[ec2-user@ip-172-31-37-122 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M     0  474M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  448K  492M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  2.1G  5.9G  27% /       ==============➔ After deleting the file.
tmpfs            99M     0   99M   0% /run/user/1000
[ec2-user@ip-172-31-37-122 ~]$
```

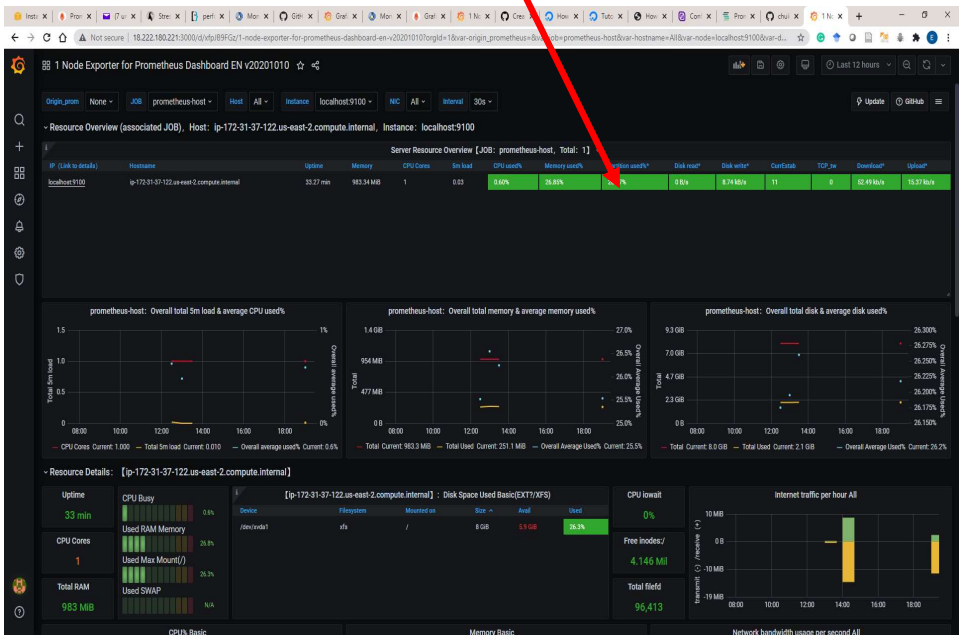
Simulate high cpu, memory, and disk usage on monitored instances

Prometheus-Host Server – Simulate Disk usage

After Allocating the memory.



After Deleting the file.

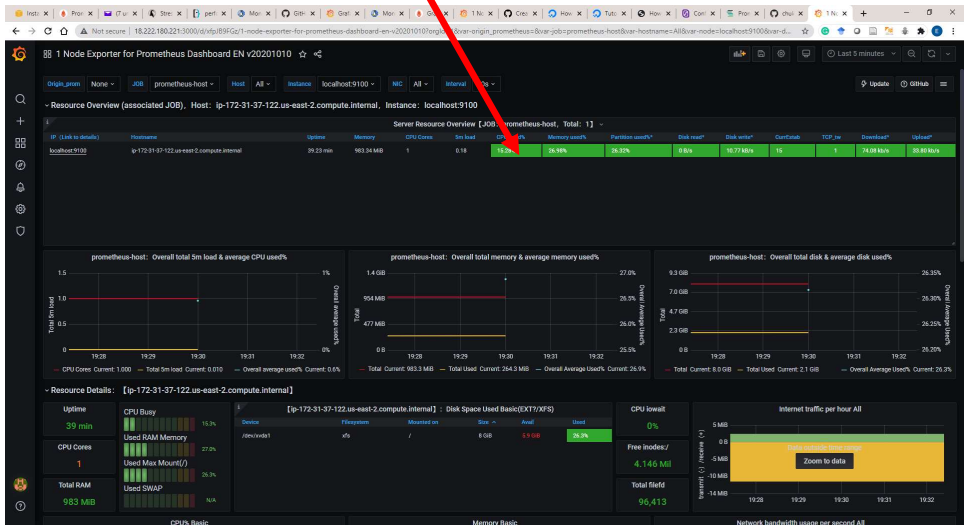


Simulate high cpu, memory, and disk usage on monitored instances

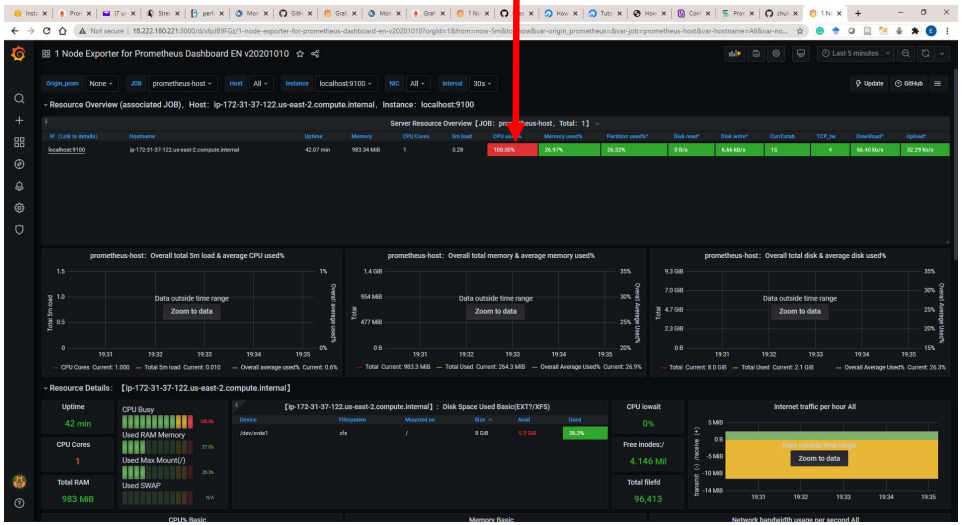
Prometheus-Host Server – Simulate cpu usage

[ec2-user@ip-172-31-37-122 ~]$ nproc | xargs seq | xargs -n1 -P4 md5sum /dev/zero

Before running the process.



Running the test.

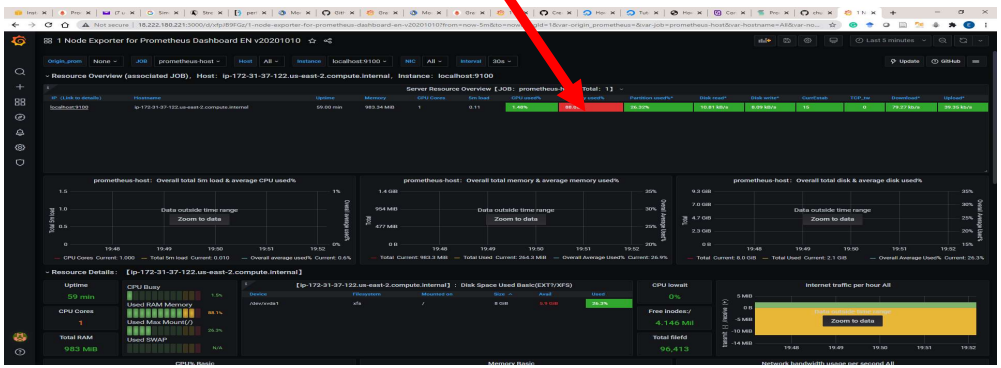


Simulate high cpu, memory, and disk usage on monitored instances

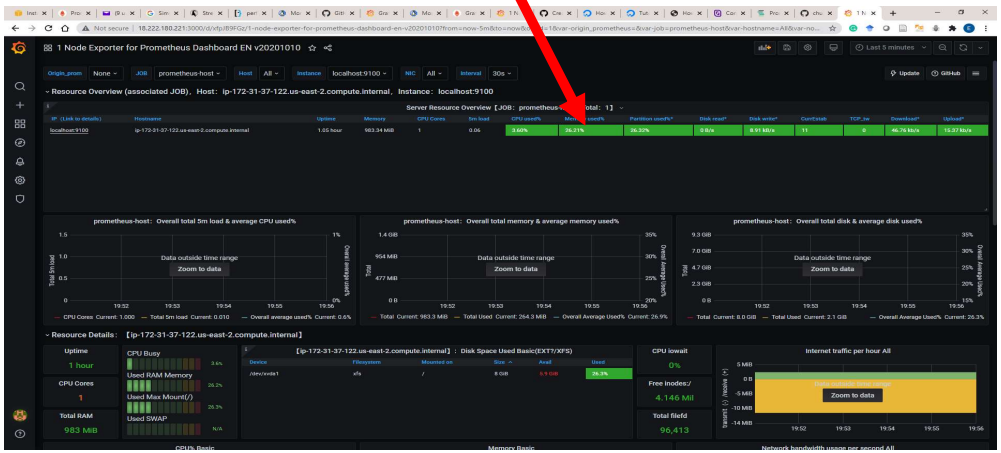
Prometheus-Host Server – Simulate Memory usage:
[ec2-user@ip-172-31-37-122 ~]$ python memoryLoadRunner.py 600

```
#!/usr/bin/env python
import sys
import time

if len(sys.argv) != 2:
    print "usage: fillmem <number-of-megabytes>"
    sys.exit()

count = int(sys.argv[1])
megabyte = (0,) * (1024 * 1024 / 8)
data = megabyte * count
while True:
    time.sleep(1)
```

Running the Test



After running the Test



Simulate high cpu, memory, and disk usage on monitored instances

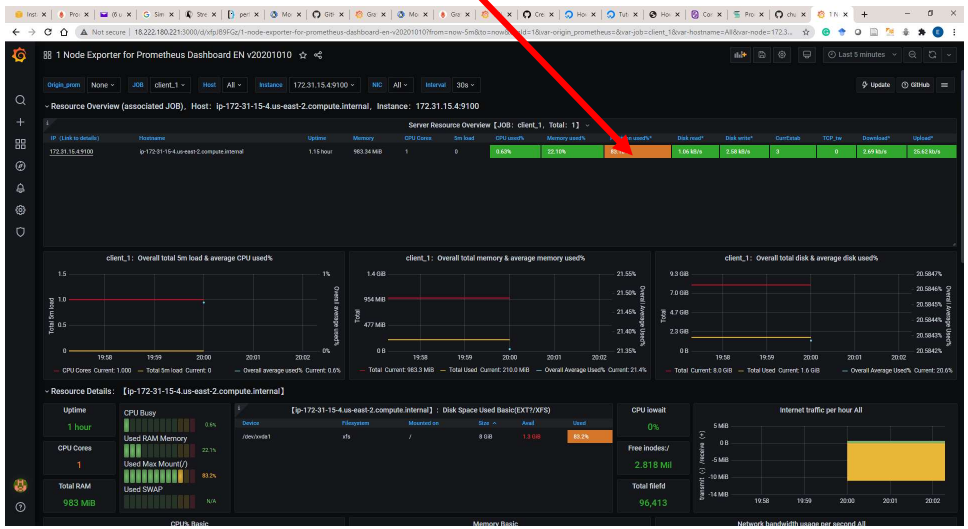
Client-1 Server – Simulate Disk usage

```
[ec2-user@ip-172-31-15-4 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M     0  474M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  436K  492M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.7G  6.4G  21% /    ==========➔ Before allocating disk space.
tmpfs            99M     0   99M   0% /run/user/1000
[ec2-user@ip-172-31-15-4 ~]$ fallocate -l 5G test-client-1_disk
[ec2-user@ip-172-31-15-4 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M     0  474M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  436K  492M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  6.7G  1.4G  84% /      ==========➔ After Allocating disk space
tmpfs            99M     0   99M   0% /run/user/1000
[ec2-user@ip-172-31-15-4 ~]$ rm test-client-1_disk
[ec2-user@ip-172-31-15-4 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M     0  474M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  436K  492M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.7G  6.4G  21% /    ==========➔ After deleting the allocated disk space
tmpfs            99M     0   99M   0% /run/user/1000
[ec2-user@ip-172-31-15-4 ~]$
```

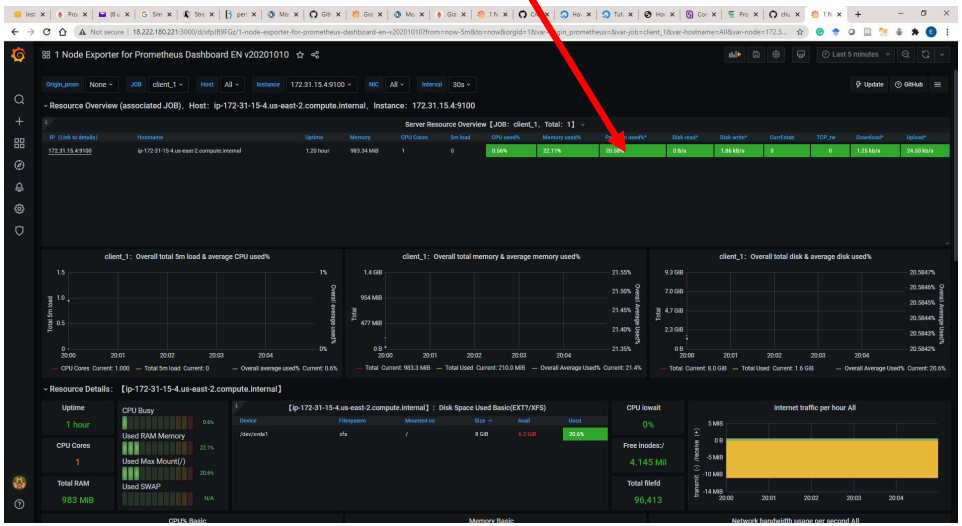
Simulate high cpu, memory, and disk usage on monitored instances

Client-1 Server – Simulate Disk usage

After Allocating the disk space.



After Deleting the allocated disk space.

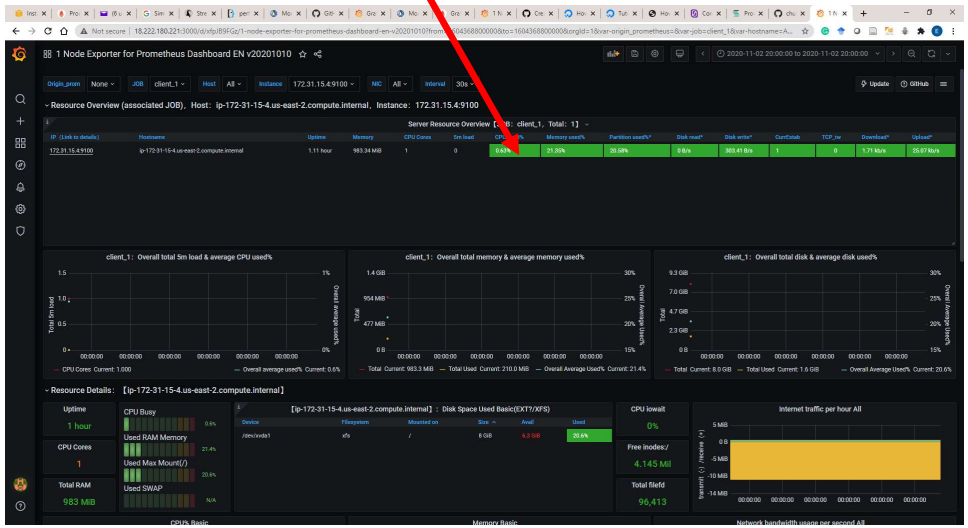


Simulate high cpu, memory, and disk usage on monitored instances

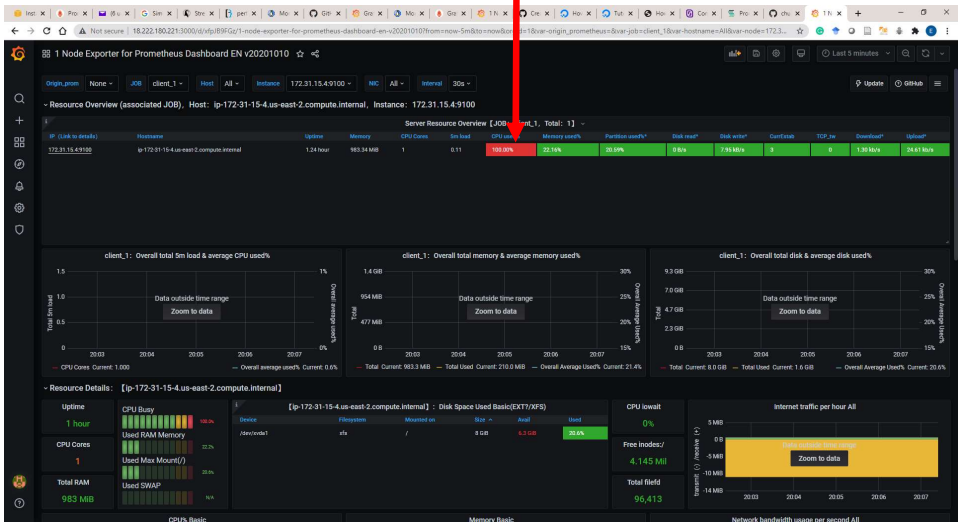
Prometheus-Host Server – Simulate cpu usage

[ec2-user@ip-172-31-15-4 ~]$ nproc | xargs seq | xargs -n1 -P4 md5sum /dev/zero

Before running the process.



Running the test.

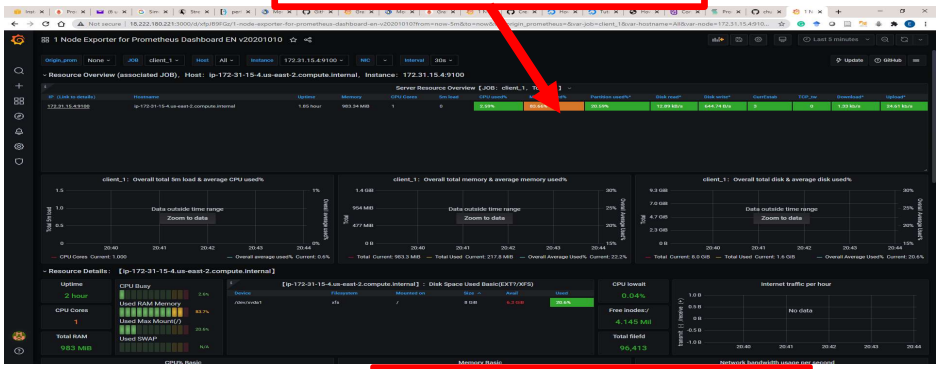


Simulate high cpu, memory, and disk usage on monitored instances

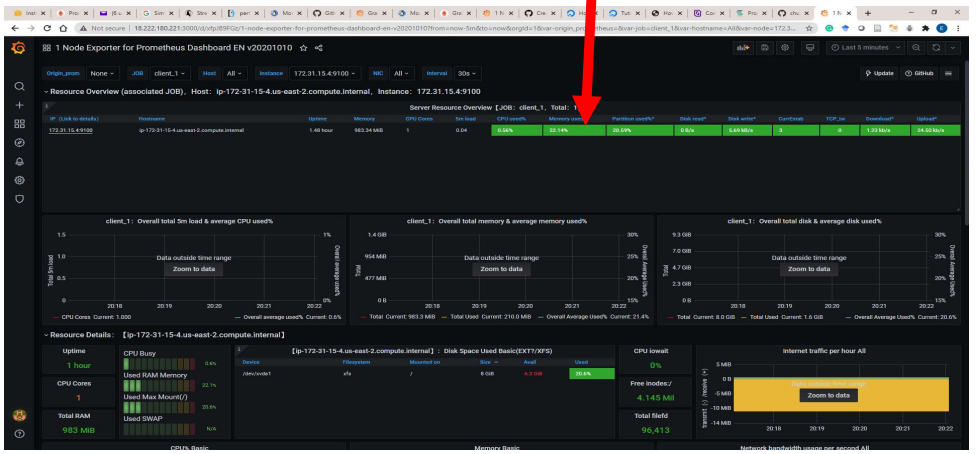
Prometheus-Host Server – Simulate Memory usage:
[ec2-user@ip-172-31-15-4 ~]$ python memoryLoadRunner.py 600

```
#!/usr/bin/env python
import sys
import time

if len(sys.argv) != 2:
    print "usage: fillmem <number-of-megabytes>"
    sys.exit()

count = int(sys.argv[1])
megabyte = (0,) * (1024 * 1024 / 8)
data = megabyte * count
while True:
    time.sleep(1)
```

Running the Test



After running the Test



Simulate high cpu, memory, and disk usage on monitored instances

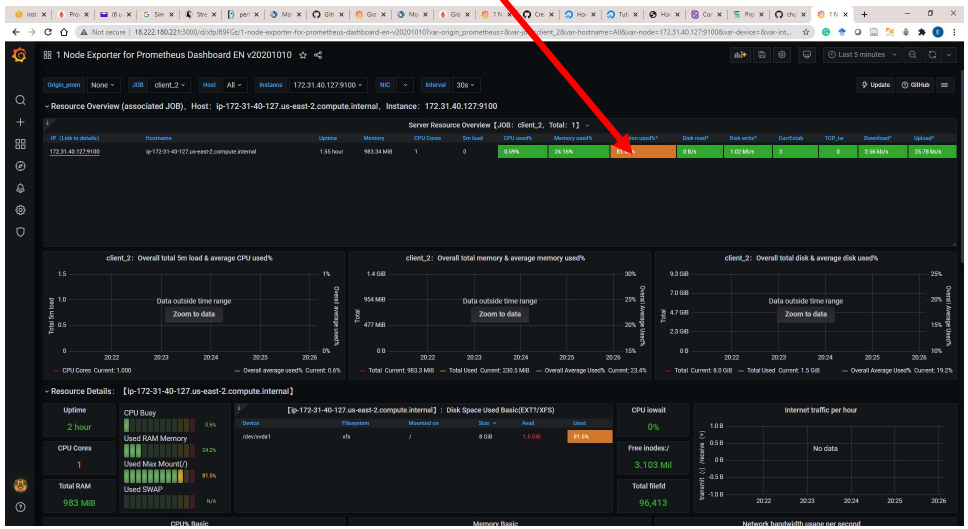
Client-2 Server: Simulate Disk usage

```
[ec2-user@ip-172-31-40-127 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M     0  474M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  436K  492M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.6G  6.5G  19% / ================➔ before allocating disk space
tmpfs            99M     0   99M   0% /run/user/1000
[ec2-user@ip-172-31-40-127 ~]$ fallocate -l 5G test-client-2
[ec2-user@ip-172-31-40-127 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M     0  474M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  436K  492M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  6.6G  1.5G  82% /     ============➔ After allocating disk space
tmpfs            99M     0   99M   0% /run/user/1000
[ec2-user@ip-172-31-40-127 ~]$ rm test-client-2
[ec2-user@ip-172-31-40-127 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M     0  474M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  436K  492M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.6G  6.5G  19% /     ==========➔ After deallocating the disk space
tmpfs            99M     0   99M   0% /run/user/1000
[ec2-user@ip-172-31-40-127 ~]$
```

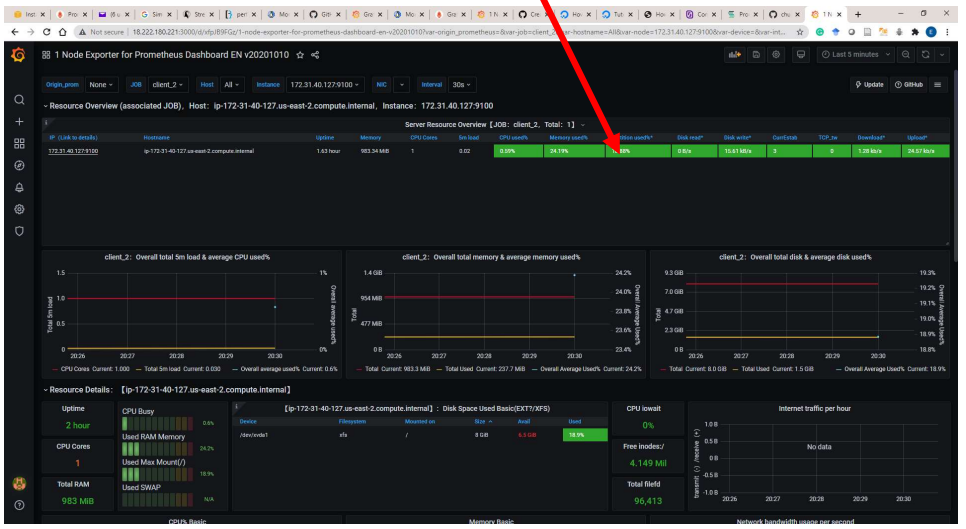
Simulate high cpu, memory, and disk usage on monitored instances

Prometheus-Host Server – Simulate Disk usage

After Allocating the 5G disk space.



After Deleting the file.

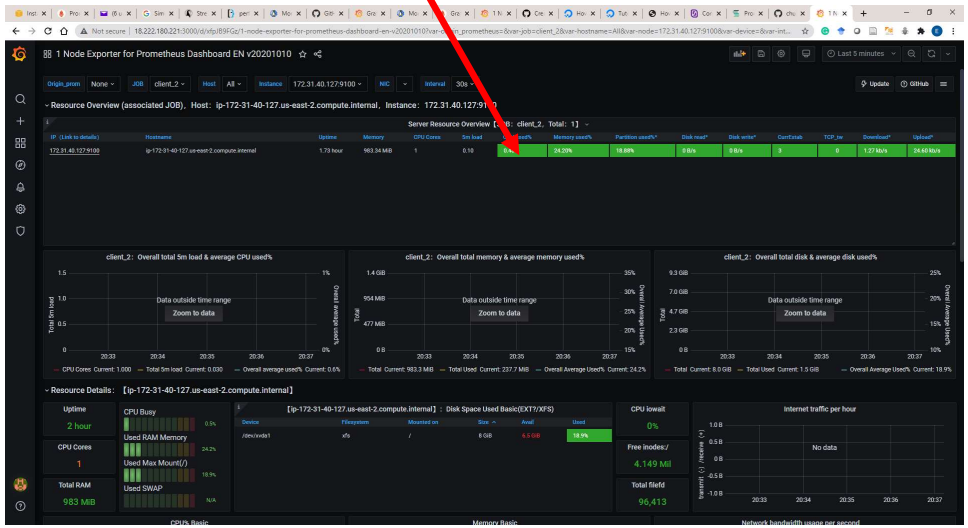


Simulate high cpu, memory, and disk usage on monitored instances

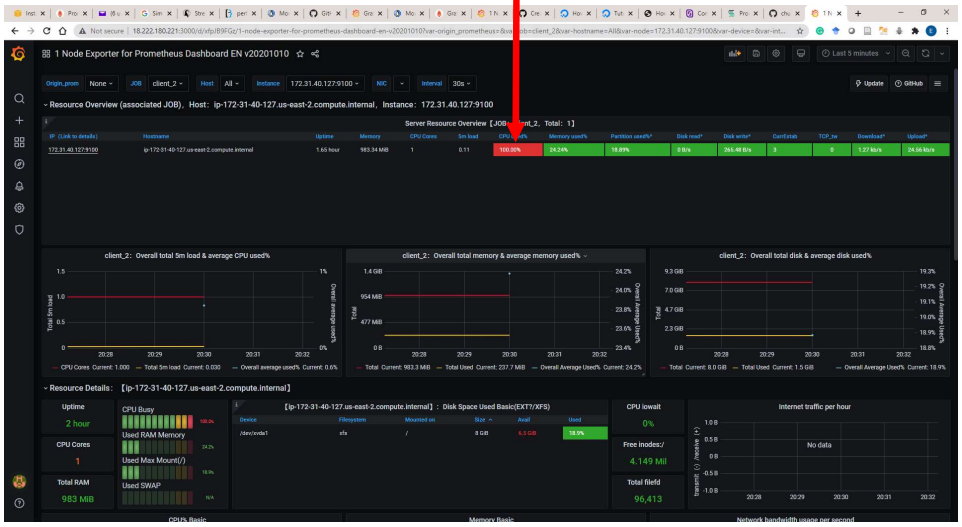
Prometheus-Host Server – Simulate cpu usage

[ec2-user@ip-172-31-40-127 ~]$ nproc | xargs seq | xargs -n1 -P4 md5sum /dev/zero

Before running the process.



Running the test.

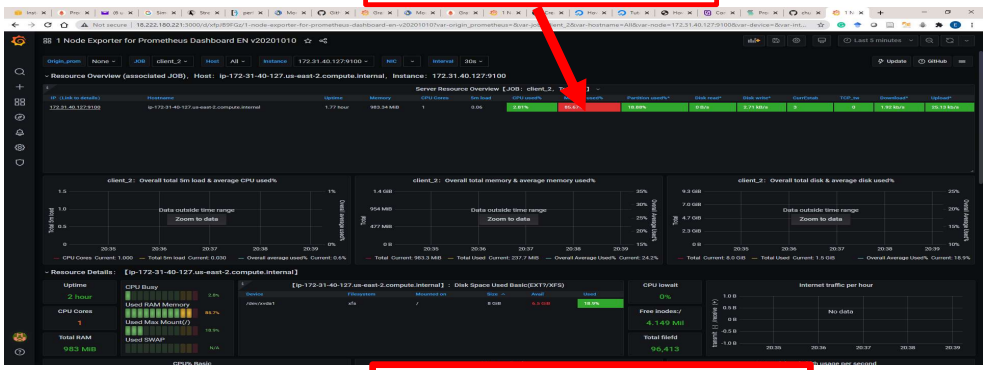


Simulate high cpu, memory, and disk usage on monitored instances

Prometheus-Host Server – Simulate Memory usage:
[ec2-user@ip-172-31-40-127 ~]$ python memoryLoadRunner.py 600

```
#!/usr/bin/env python
import sys
import time

if len(sys.argv) != 2:
    print "usage: fillmem <number-of-megabytes>"
    sys.exit()

count = int(sys.argv[1])
megabyte = (0,) * (1024 * 1024 / 8)
data = megabyte * count
while True:
    time.sleep(1)
```

Running the Test



After running the Test