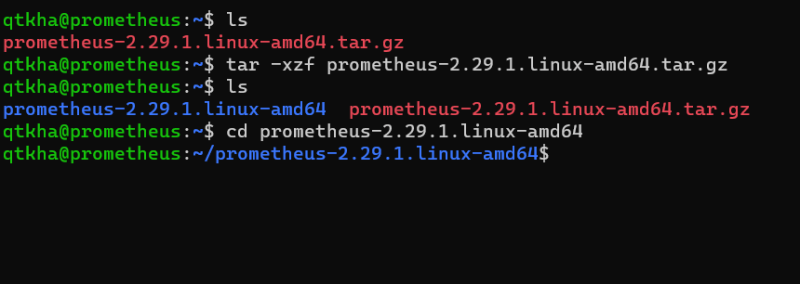
**Running Prometheus**

* Lets try to run Prometheus on a Linux machine
* Lets create a vm
* Download the linux tar file <https://prometheus.io/download/> 
* Now in the prometheus.yml file ensure you have the following content

# my global config

global:

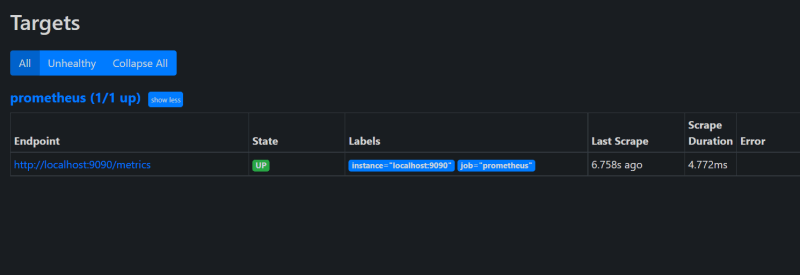
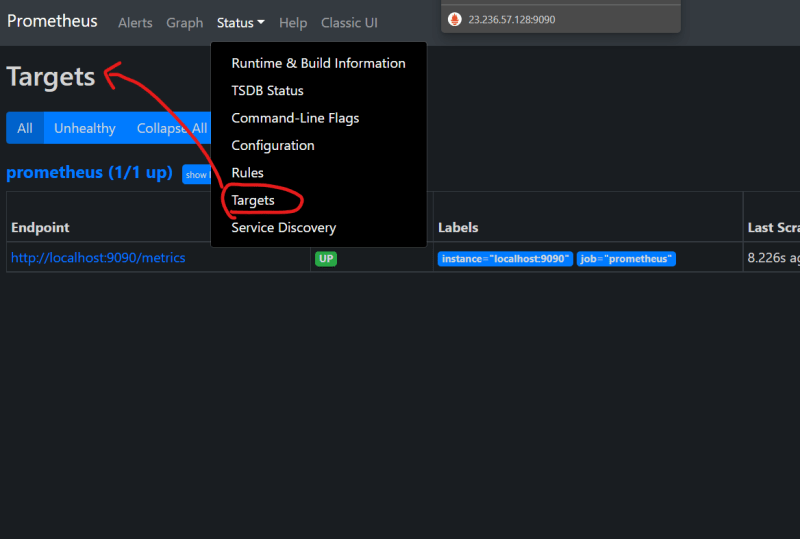
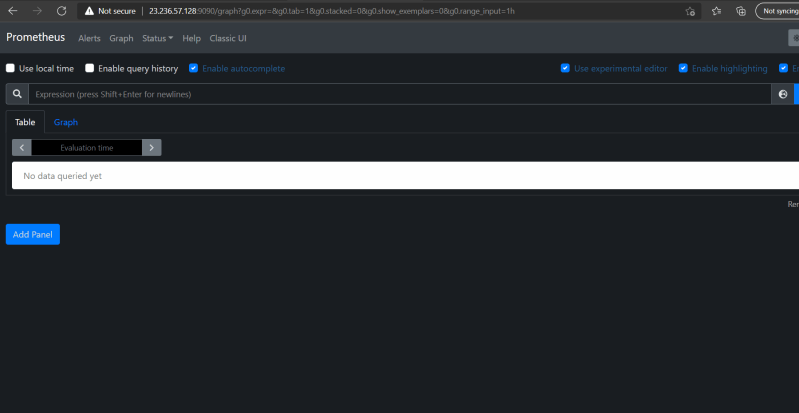
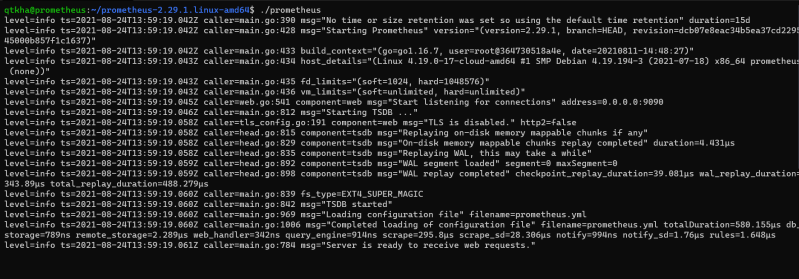
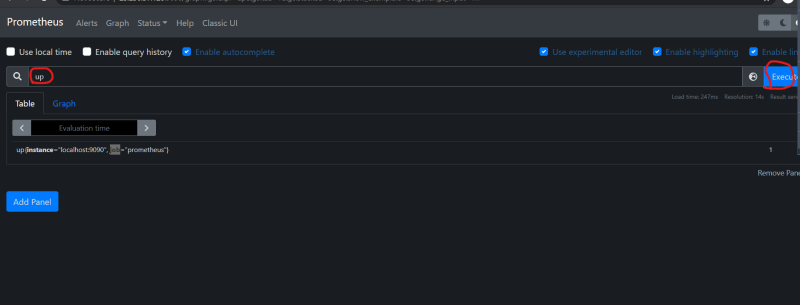
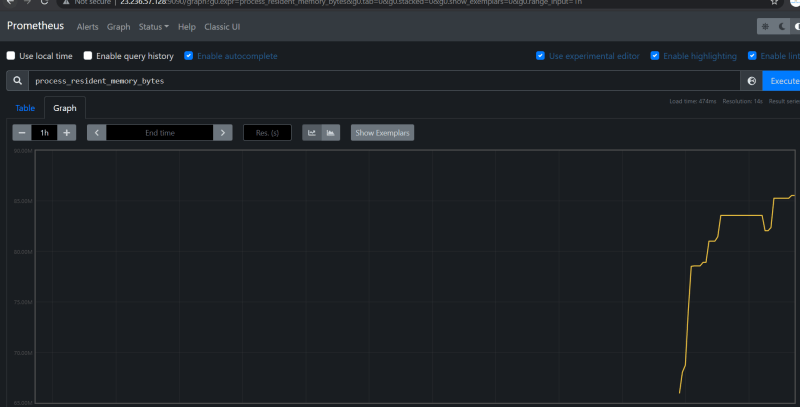
scrape\_interval: 10s

scrape\_configs:

- job\_name: "prometheus"

static\_configs:

- targets: ["localhost:9090"]

* Now lets try to run prometheus 
* Open the prometheus expression browser and execute up 
* There is a single result with value 1 and the name up{instance="localhost:9090", job="prometheus"}.
* up is a special metric added by Prometheus when it performsa scrape and 1 indicates that the scrape was succesful. Then instance is a label, indicating the target wthat was scrapted and the job label here comes from job\_name in the prometheus.yml file 
* Lets try to download a Node exporter [Refer Here](https://prometheus.io/download/)
  + untar the exporter and try to run the node exporter with ./node\_exporter
  + Now access the metrics from node exporter http://<ip&gt;:9100
* Now lets add some more info to the prometheus.yml to scrape node metrics exported by node exporter

global:

scrape\_interval: 10s

scrape\_configs:

- job\_name: "prometheus"

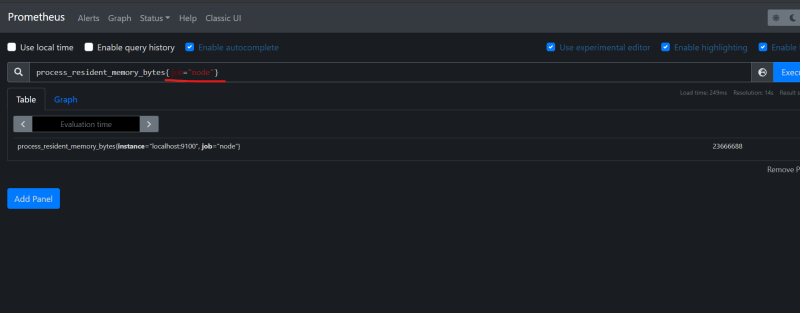
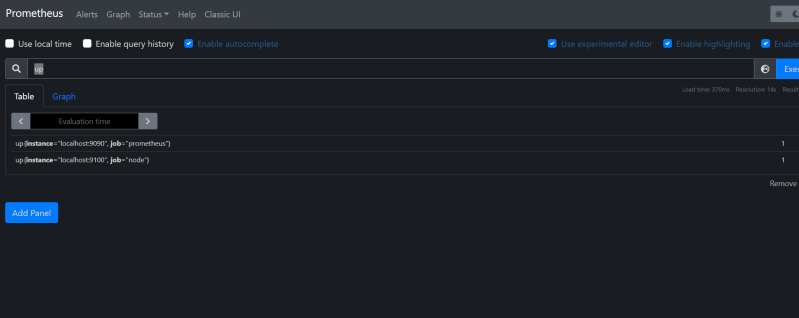
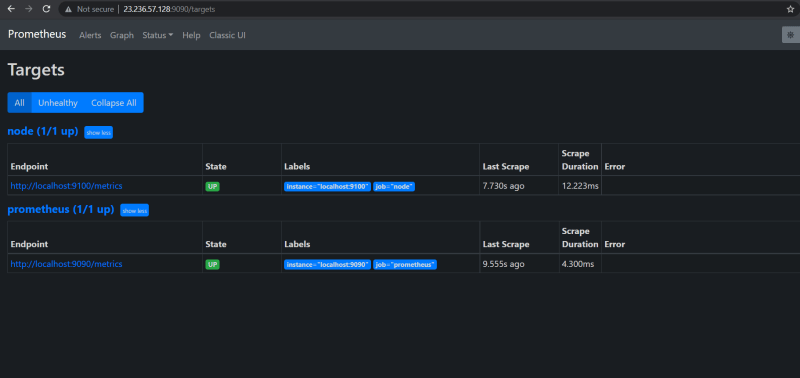
static\_configs:

- targets: ["localhost:9090"]

- job\_name: node

static\_configs:

- targets: ["localhost:9100"]

* and restart prometheus (./prometheus) 
* Now lets try to add some alert manager configuration

global:

scrape\_interval: 10s

alerting:

alertmanagers:

- static\_configs:

- localhost:9093

scrape\_configs:

- job\_name: "prometheus"

static\_configs:

- targets: ["localhost:9090"]

- job\_name: node

static\_configs:

- targets: ["localhost:9100"]

* Now create a new file called as rules.yaml

groups:

- name: example

rules:

- alert: InstanceDown

expr: up == 0

for: 1m