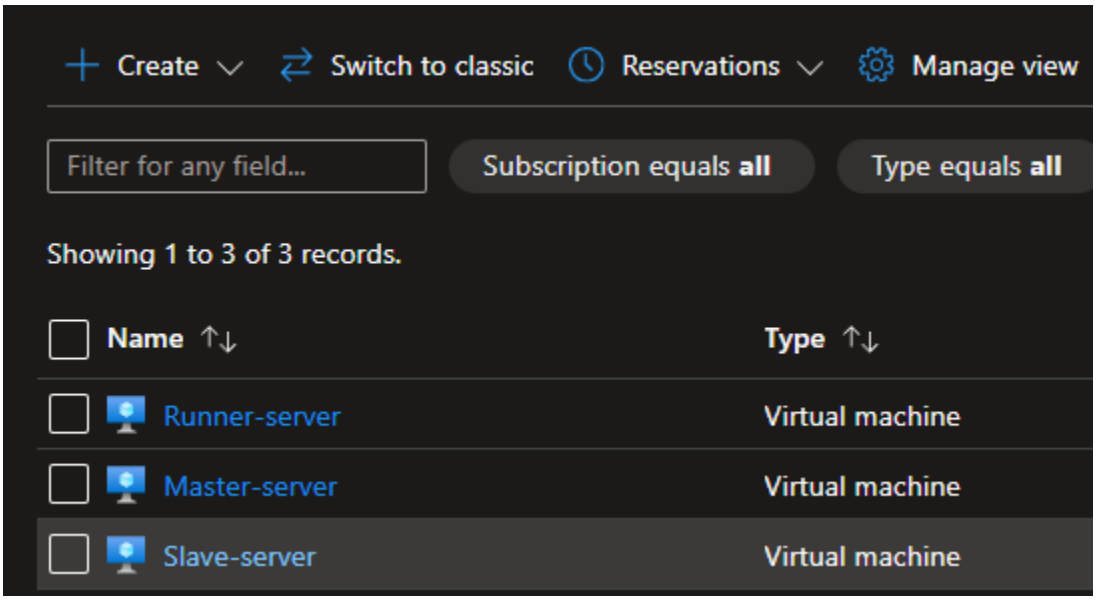
Boardgame- test

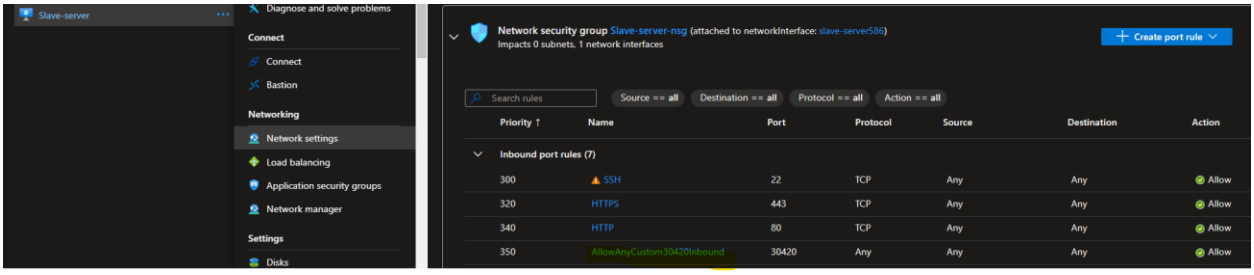
3 server we have to create.

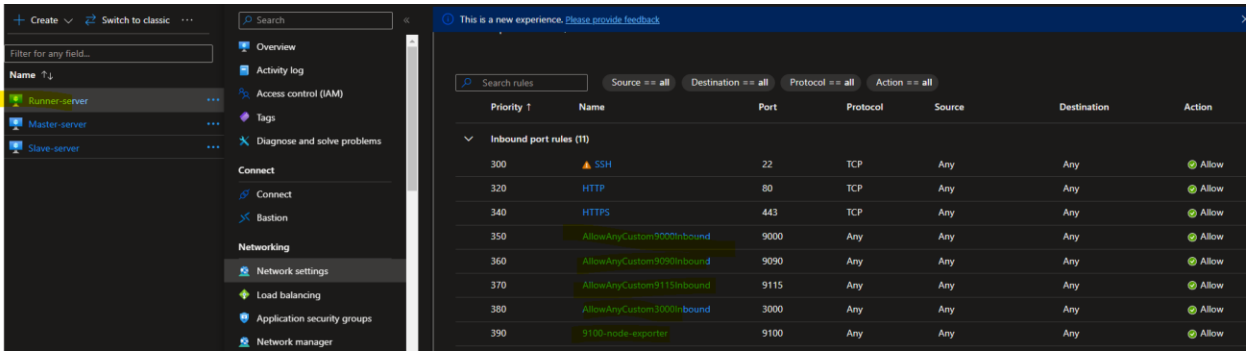
Runner server is integrating with the GitHub as a runner server. All the other tool we have to install / configure here.



We have to open the some Ports for incoming and outgoing traffic.

Kubernetes are running on the slave server.





Master node we have to run the script and got the access token.

```
  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 10.0.0.4:6443 --token dubc5q.tykr26j72dukq651 \
        --discovery-token-ca-cert-hash sha256:4c4f59a467151381e85de4b357f44d9e92287a6d26d59532c24fb136eb2c4b7e
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
```

Copy the token (yellow part) and past to the worker node. With root directory.

Sudo su

```
azadmin@VM02:~$ sudo su
root@VM02:/home/azadmin# kubeadm join 10.0.0.4:6443 --token dubc5q.tykr26j72dukq651 \
>         --discovery-token-ca-cert-hash sha256:4c4f59a467151381e85de4b357f44d9e92287a6d26d59532c24fb136eb2c4b7e
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
```

Once the node has joined.

kubectl get nodes (type the command and check the answer).

Note – control-plane (master server/node), <none> is worker node.

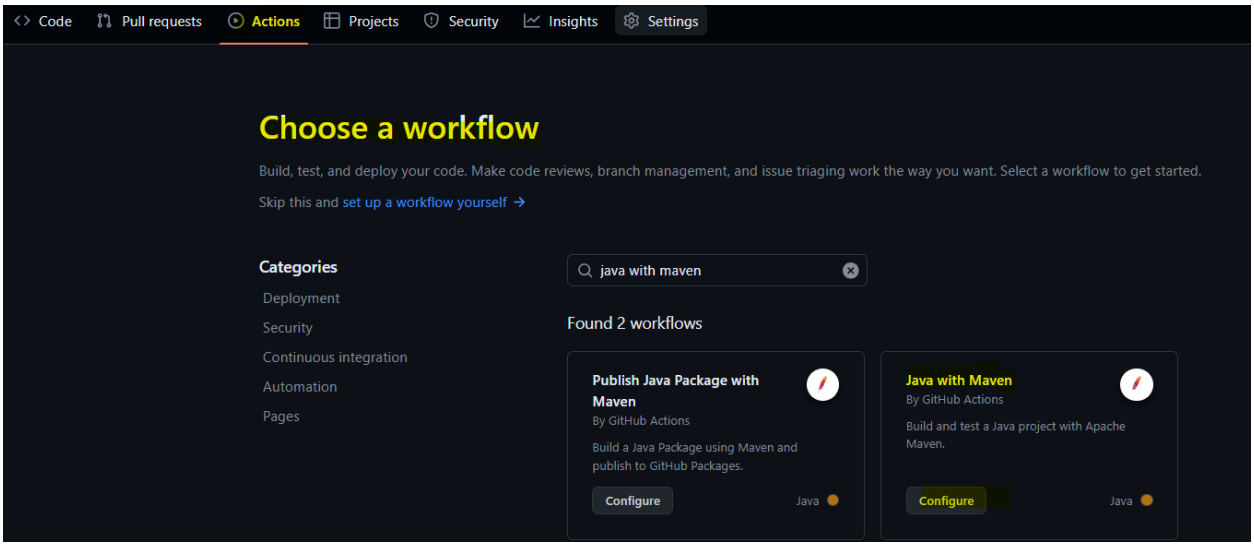
```
azadmin@VM01:~$ kubectl get nodes
NAME    STATUS   ROLES           AGE   VERSION
vm01    Ready    control-plane   48m   v1.28.1
vm02    Ready    <none>          44m   v1.28.1
azadmin@VM01:~$
```

Installed maven (on runner server)

```
azadmin@VM03-R:~/actions-runner$ mvn

Command 'mvn' not found, but can be installed with:

sudo apt install maven

azadmin@VM03-R:~/actions-runner$ sudo apt install maven
```

Sudo apt install maven -y

Configure the below snap as per the requirement.



Note expoter – used for the system level monitoring

Blackbox monitoring – used for the website monitoring

Monitoring tools

https://prometheus.io/download/

wget (url as copy as address)

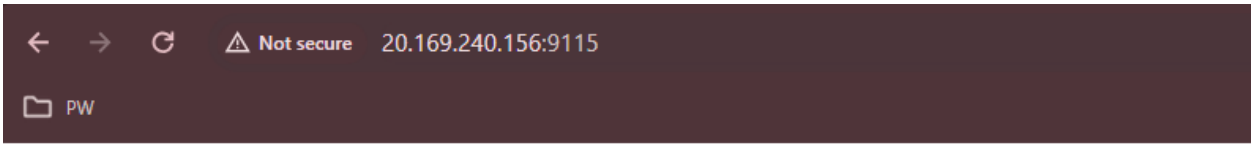
tar -xvf

mv (rename the file)

```
azadmin@Runner-server:~/monitoring$ ls
prometheus-2.51.1.linux-amd64.tar.gz
azadmin@Runner-server:~/monitoring$ tar -xvf prometheus-2.51.1.linux-amd64.tar.gz
prometheus-2.51.1.linux-amd64/
prometheus-2.51.1.linux-amd64/promtool
prometheus-2.51.1.linux-amd64/NOTICE
prometheus-2.51.1.linux-amd64/consoles/
prometheus-2.51.1.linux-amd64/consoles/prometheus.html
prometheus-2.51.1.linux-amd64/consoles/index.html.example
prometheus-2.51.1.linux-amd64/consoles/node-overview.html
prometheus-2.51.1.linux-amd64/consoles/node-cpu.html
prometheus-2.51.1.linux-amd64/consoles/node.html
prometheus-2.51.1.linux-amd64/consoles/prometheus-overview.html
prometheus-2.51.1.linux-amd64/consoles/node-disk.html
prometheus-2.51.1.linux-amd64/prometheus.yml
prometheus-2.51.1.linux-amd64/LICENSE
prometheus-2.51.1.linux-amd64/console_libraries/
prometheus-2.51.1.linux-amd64/console_libraries/prom.lib
prometheus-2.51.1.linux-amd64/console_libraries/menu.lib
prometheus-2.51.1.linux-amd64/prometheus
azadmin@Runner-server:~/monitoring$ ls
prometheus-2.51.1.linux-amd64  prometheus-2.51.1.linux-amd64.tar.gz
azadmin@Runner-server:~/monitoring$ rm prometheus-2.51.1.linux-amd64.tar.gz
azadmin@Runner-server:~/monitoring$ ls
prometheus-2.51.1.linux-amd64
azadmin@Runner-server:~/monitoring$ mv prometheus-2.51.1.linux-amd64/ prometheus
azadmin@Runner-server:~/monitoring$ ls
prometheus
azadmin@Runner-server:~/monitoring$ ls
prometheus
azadmin@Runner-server:~/monitoring$ cd prometheus/
azadmin@Runner-server:~/monitoring/prometheus$ ls
LICENSE  NOTICE  console_libraries  consoles  prometheus  prometheus.yml  promtool
azadmin@Runner-server:~/monitoring/prometheus$ ./prometheus &
```

Same process for black box download.

```
azadmin@Runner-server:~/monitoring$ ls
blackbox_exporter-0.25.0.linux-amd64.tar.gz  prometheus
azadmin@Runner-server:~/monitoring$ tar -xvf blackbox_exporter-0.25.0.linux-amd64.tar.gz
blackbox_exporter-0.25.0.linux-amd64/
blackbox_exporter-0.25.0.linux-amd64/NOTICE
blackbox_exporter-0.25.0.linux-amd64/blackbox_exporter
blackbox_exporter-0.25.0.linux-amd64/LICENSE
blackbox_exporter-0.25.0.linux-amd64/blackbox.yml
azadmin@Runner-server:~/monitoring$ ls
blackbox_exporter-0.25.0.linux-amd64  blackbox_exporter-0.25.0.linux-amd64.tar.gz  prometheus
azadmin@Runner-server:~/monitoring$ rm blackbox_exporter-0.25.0.linux-amd64.tar.gz
azadmin@Runner-server:~/monitoring$ ls
blackbox_exporter-0.25.0.linux-amd64  prometheus
azadmin@Runner-server:~/monitoring$ mv blackbox_exporter-0.25.0.linux-amd64/ blackbox-tool
azadmin@Runner-server:~/monitoring$ ls
blackbox-tool  prometheus
azadmin@Runner-server:~/monitoring$ cd blackbox-tool/
azadmin@Runner-server:~/monitoring/blackbox-tool$ ls
LICENSE  NOTICE  blackbox.yml  blackbox_exporter
azadmin@Runner-server:~/monitoring/blackbox-tool$ ./blackbox_exporter &
```



Blackbox Exporter

Probe prometheus.io for http_2xx

Debug probe prometheus.io for http_2xx

Metrics

Configuration

Recent Probes

Module	Target	Result	Debug

We have to configure

```
azadmin@Runner-server:~/monitoring$ cd prometheus/
azadmin@Runner-server:~/monitoring/prometheus$ ls
LICENSE  console_libraries  data         prometheus.yml
NOTICE   consoles           prometheus   promtool
azadmin@Runner-server:~/monitoring/prometheus$ vi prometheus.yml
azadmin@Runner-server:~/monitoring/prometheus$
```

https://github.com/prometheus/blackbox_exporter/blob/master/README.md

```
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]
  - job_name: 'blackbox'
    metrics_path: /probe
    params:
      module: [http_2xx]  # Look for a HTTP 200 response.
    static_configs:
      - targets:
        - http://prometheus.io    # Target to probe with http.
        - http://172.172.146.253:30420   # Target to probe with https.
        # Target to probe with http on port 8080.
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label:   address
        replacement: 20.169.240.156:9115
```

server IP with port numbers

server IP

We have to restart the Prometheus

First, we have to kill it.

```
LICENSE   console_libraries  data         prometheus.yml
NOTICE    consoles           prometheus   promtool
azadmin@Runner-server:~/monitoring/prometheus$ vi prometheus.yml
azadmin@Runner-server:~/monitoring/prometheus$ pgrep prometheus
28730
azadmin@Runner-server:~/monitoring/prometheus$ kill 28730
azadmin@Runner-server:~/monitoring/prometheus$ ts=2024-04-09T17:18:23.819Z calle
r=main go:964 level=warn msg="Received SIGTERM, exiting gracefully..."
```

Blackbox running

Blackbox Exporter

Probe prometheus.io for http_2xx

Debug probe prometheus.io for http_2xx

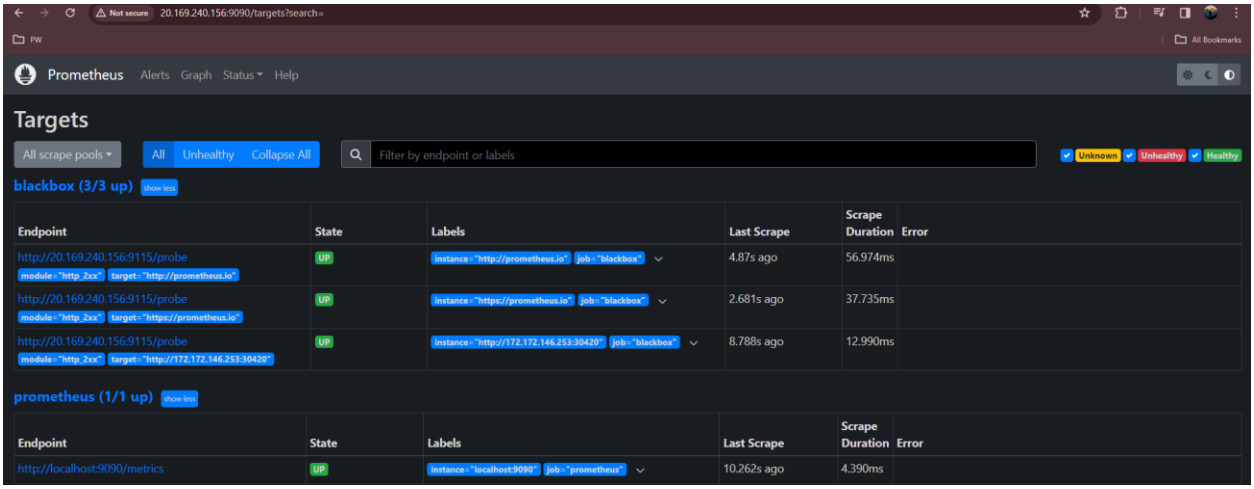
Metrics

Configuration

Recent Probes

Module	Target	Result	Debug
http_2xx	https://prometheus.io	Success	Logs
http_2xx	http://prometheus.io	Success	Logs
http_2xx	http://172.172.146.253:30420	Success	Logs
http_2xx	https://prometheus.io	Success	Logs
http_2xx	http://prometheus.io	Success	Logs
http_2xx	http://172.172.146.253:30420	Success	Logs
http_2xx	https://prometheus.io	Success	Logs
http_2xx	http://prometheus.io	Success	Logs
http_2xx	http://172.172.146.253:30420	Success	Logs
http_2xx	https://prometheus.io	Success	Logs
http_2xx	http://prometheus.io	Success	Logs
http_2xx	http://172.172.146.253:30420	Success	Logs
http_2xx	https://prometheus.io	Success	Logs
http_2xx	http://prometheus.io	Success	Logs
http_2xx	http://172.172.146.253:30420	Success	Logs
http_2xx	https://prometheus.io	Success	Logs

Prometheus running too

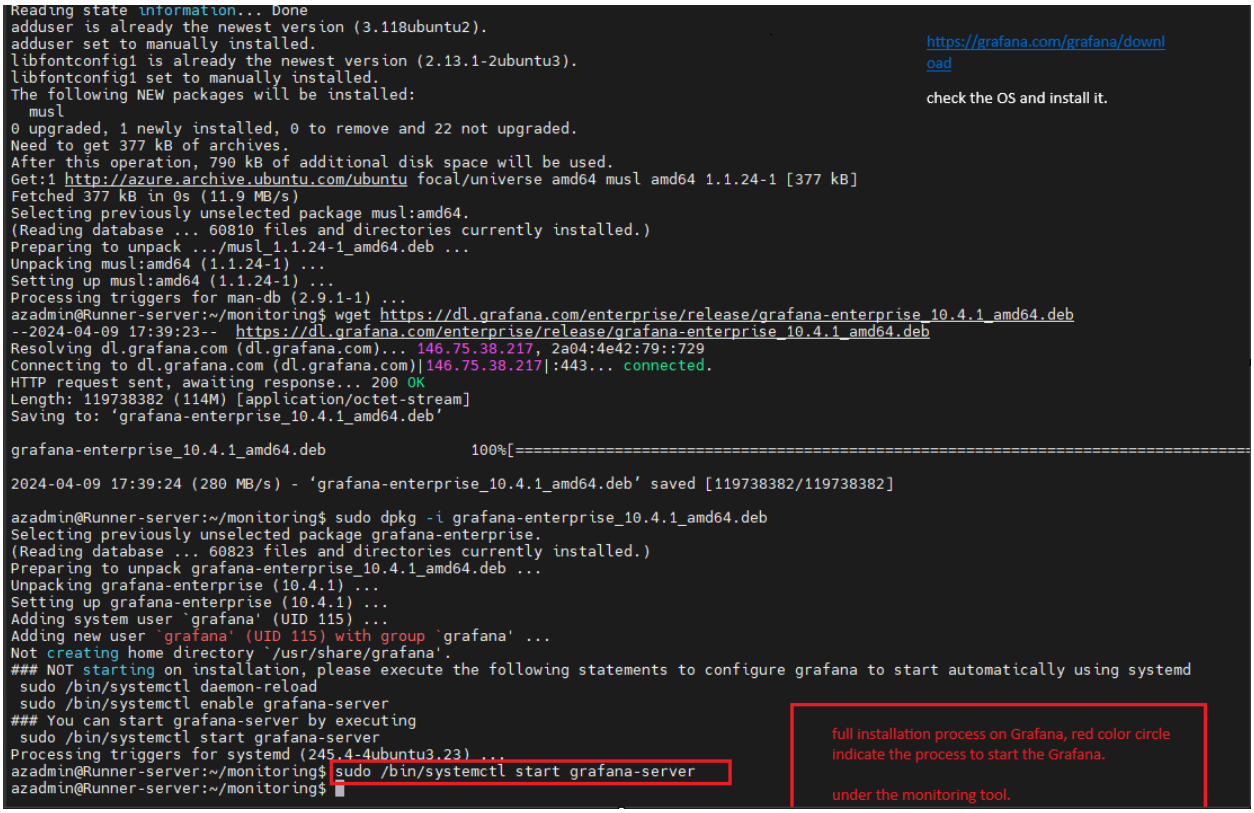


Grafana

Note – get this result proper visualize format we can use Grafana.

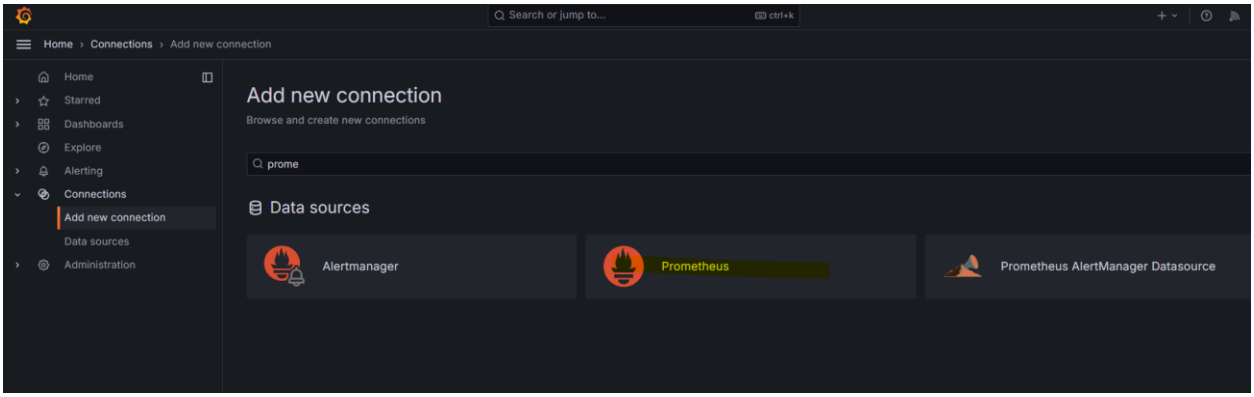
https://grafana.com/grafana/download

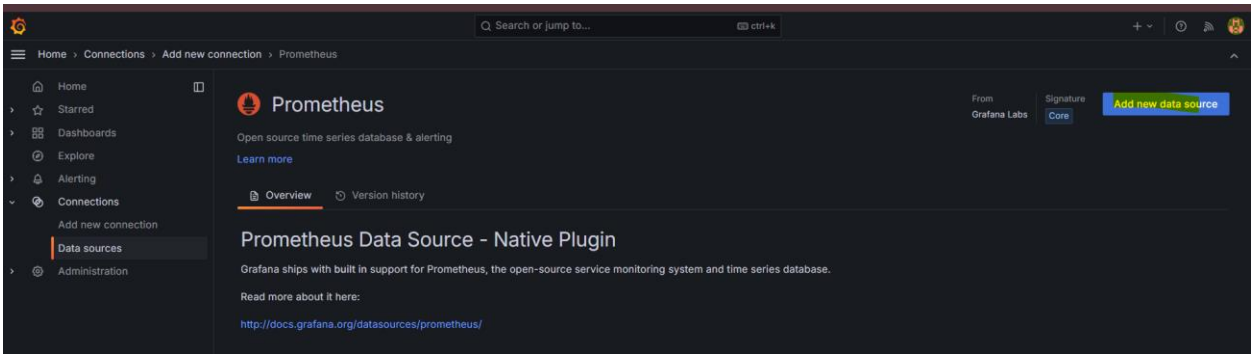
copy the command to the runner server

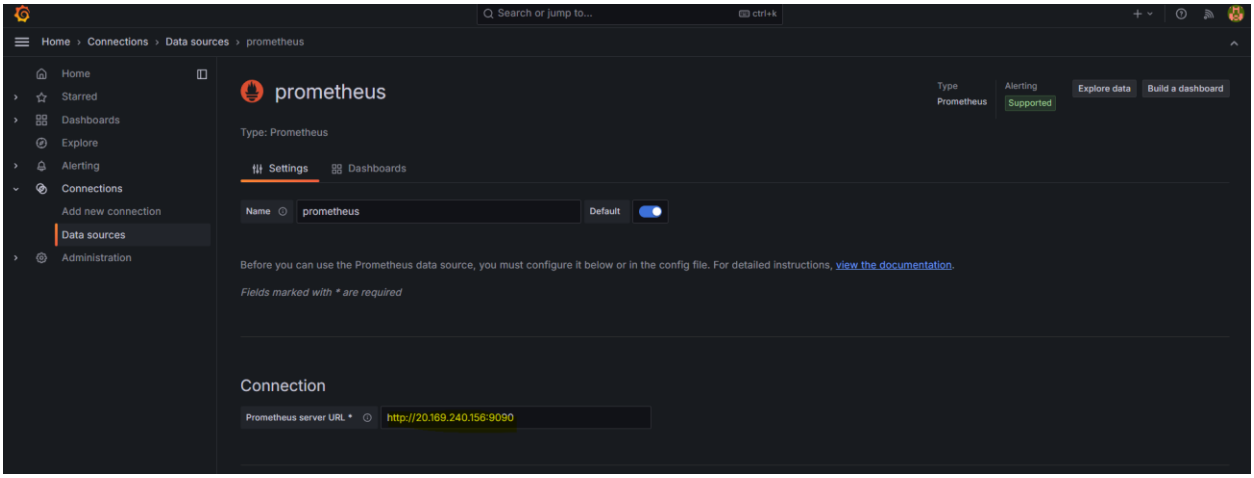


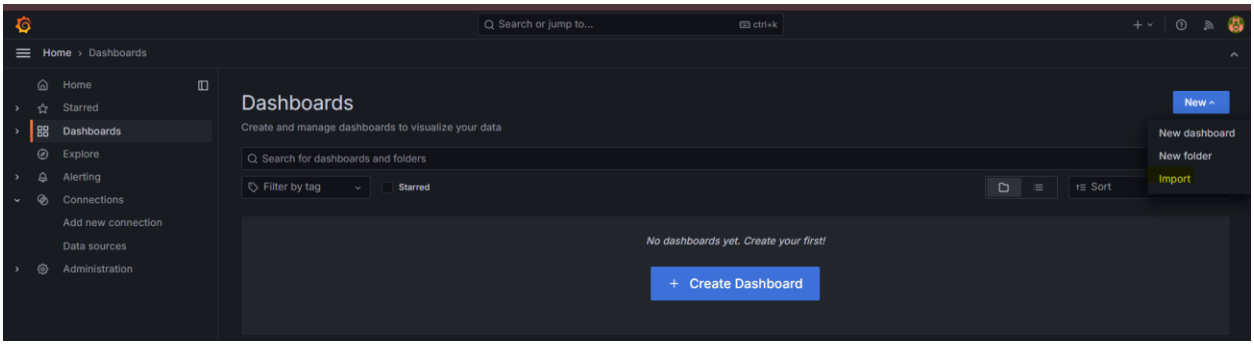
Check the runner IP with port 3000

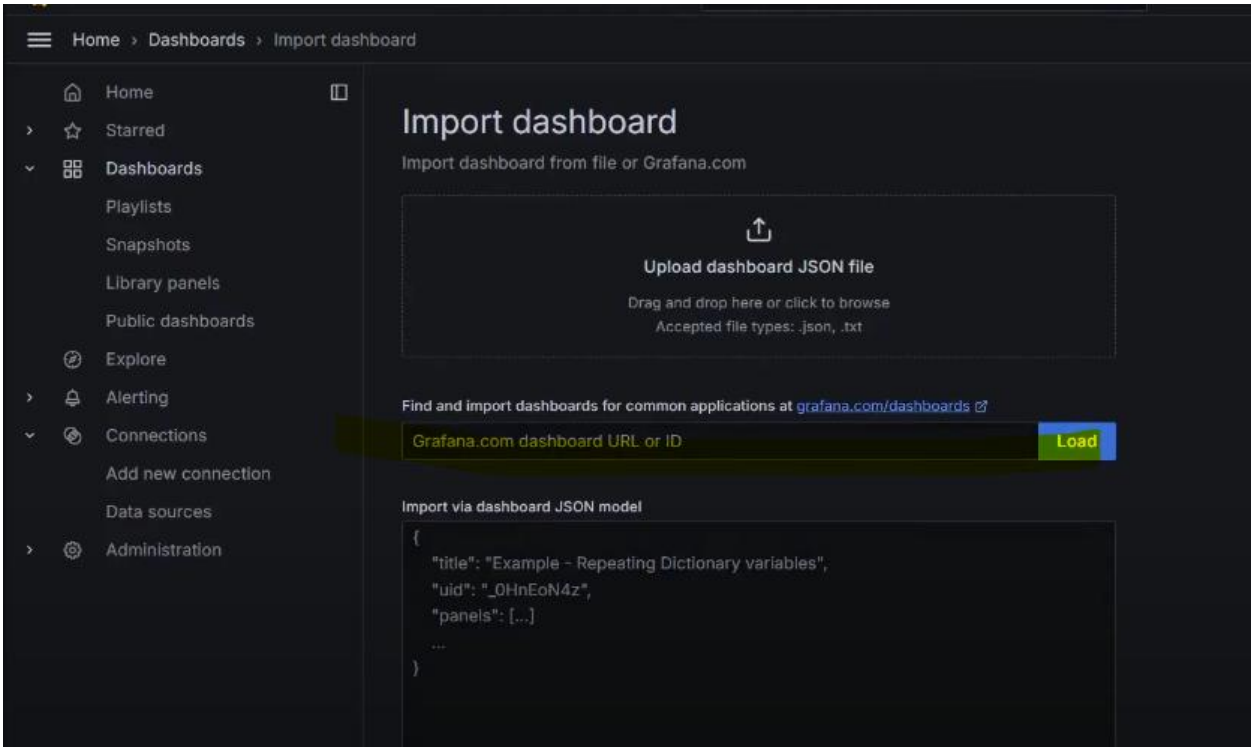
We have to follow the below steps

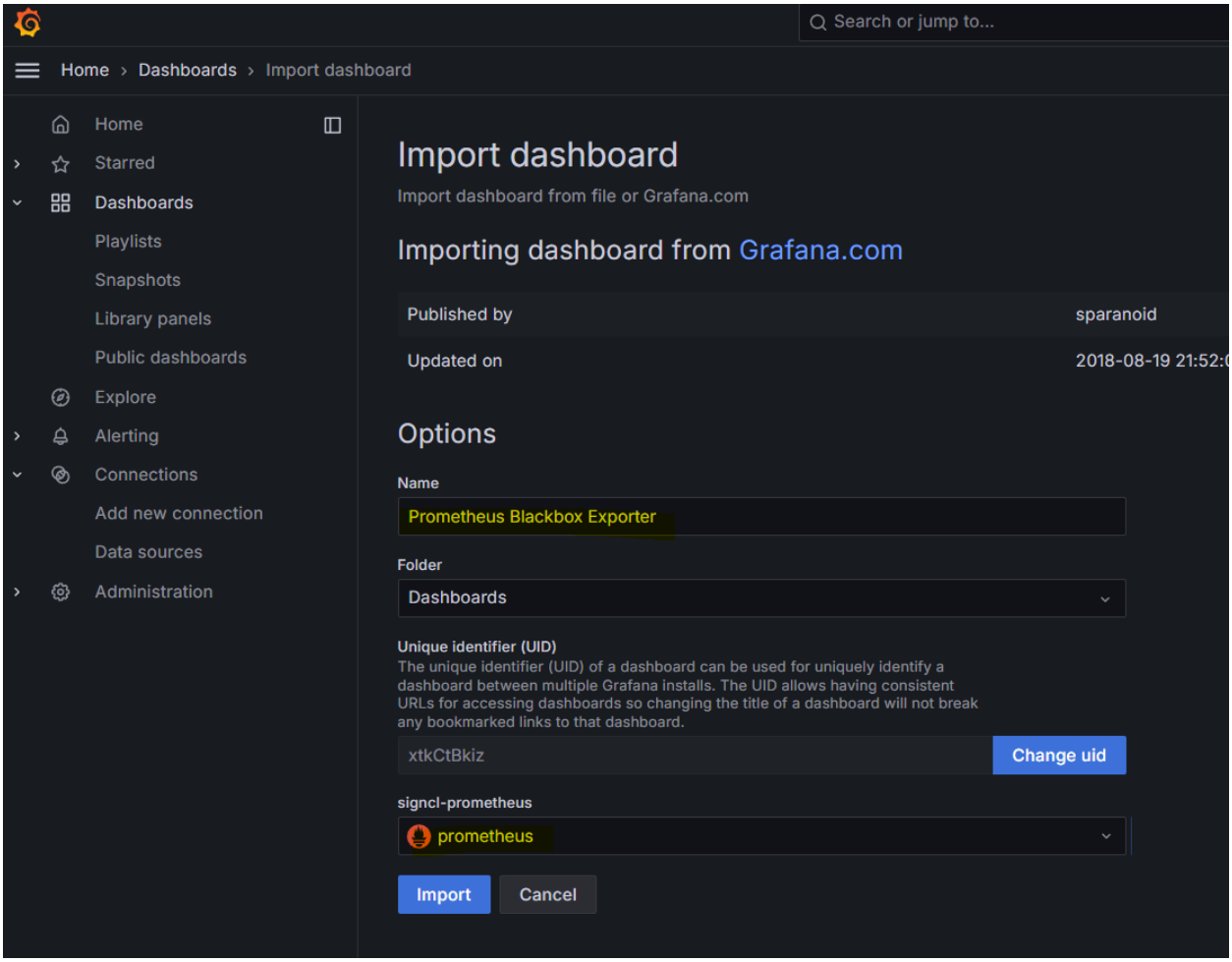


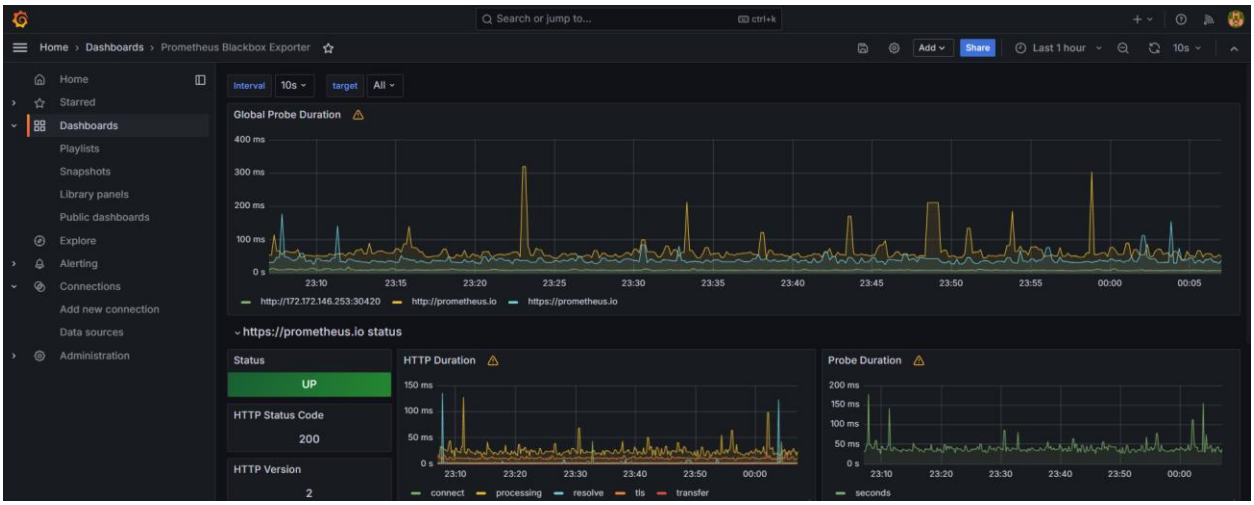





Search or jump to...
ctrl+k
Home › Dashboards
Home
Starred
Dashboards
Explore
Alerting
Connections
Add new connection
Data sources
Administration
Dashboards
Create and manage dashboards to visualize your data
New
New dashboard
New folder
Import
Search for dashboards and folders
Filter by tag
Starred
Sort
No dashboards yet. Create your first!
Create Dashboard


Home › Dashboards › Import dashboard
Home
Starred
Dashboards
Playlists
Snapshots
Library panels
Public dashboards
Explore
Alerting
Connections
Add new connection
Data sources
Administration
Import dashboard
Import dashboard from file or Grafana.com
Upload dashboard JSON file
Drag and drop here or click to browse
Accepted file types: .json, .txt
Find and import dashboards for common applications at grafana.com/dashboards
Grafana.com dashboard URL or ID
Load
Import via dashboard JSON model
{
"title": "Example - Repeating Dictionary variables",
"uid": "_0HnEoN4z",
"panels": [...]
...
}





Node exporter

```
azadmin@Runner-server:~/monitoring$ ls
blackbox-tool  grafana-enterprise_10.4.1_amd64.deb  node_exporter-1.7.0.linux-amd64.tar.gz  prometheus
azadmin@Runner-server:~/monitoring$ tar -xvf node_exporter-1.7.0.linux-amd64.tar.gz
node_exporter-1.7.0.linux-amd64/
node_exporter-1.7.0.linux-amd64/LICENSE
node_exporter-1.7.0.linux-amd64/node_exporter
node_exporter-1.7.0.linux-amd64/NOTICE
azadmin@Runner-server:~/monitoring$ ls
blackbox-tool  grafana-enterprise_10.4.1_amd64.deb  node_exporter-1.7.0.linux-amd64  node_exporter-1.7.0.linux-amd64.tar.gz  prometheus
azadmin@Runner-server:~/monitoring$ rm node_exporter-1.7.0.linux-amd64.tar.gz
azadmin@Runner-server:~/monitoring$ ls
blackbox-tool  grafana-enterprise_10.4.1_amd64.deb  node_exporter-1.7.0.linux-amd64  prometheus
azadmin@Runner-server:~/monitoring$ mv node_exporter-1.7.0.linux-amd64/ node_exporter
azadmin@Runner-server:~/monitoring$ ls
blackbox-tool  grafana-enterprise_10.4.1_amd64.deb  node_exporter  prometheus
azadmin@Runner-server:~/monitoring$ cd node_exporter/
azadmin@Runner-server:~/monitoring/node_exporter$ ls
LICENSE  NOTICE  node_exporter
azadmin@Runner-server:~/monitoring/node_exporter$ ./node_exporter &
```

Runner server IP with port number 9100

Prometheus Node Exporter

Version: (version=1.7.0, branch=HEAD, revision=7333465abf9efba81876303bb57e6fadb946041b)

- Metrics

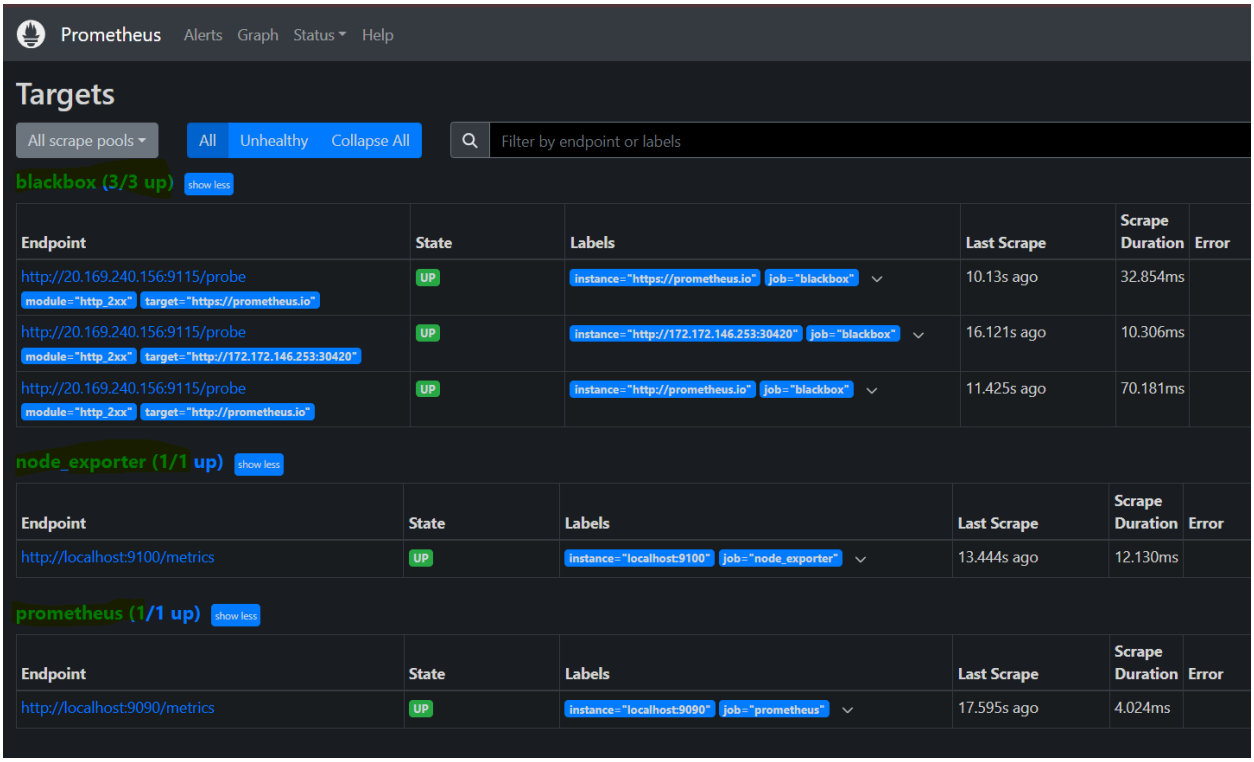
After write the yml code in the Prometheus.yml

write the command inside the prometheus.yml file

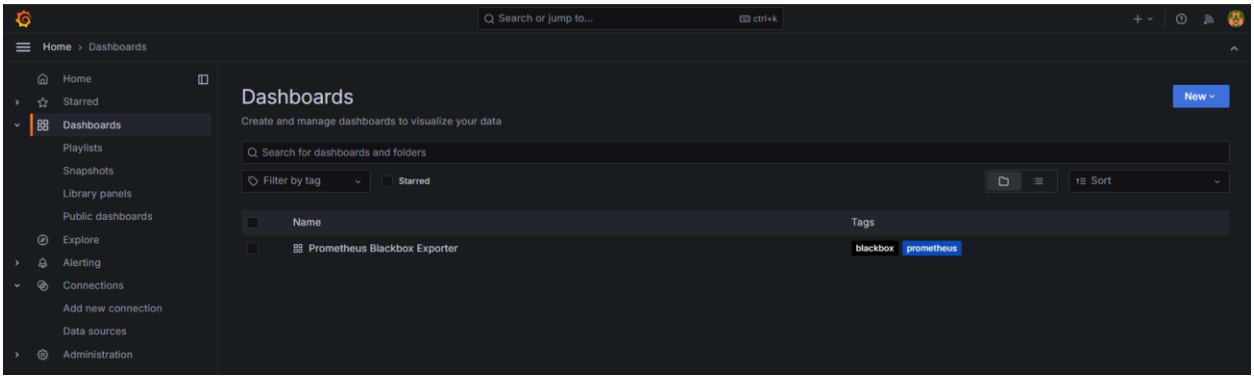
- job_name: ‘node_exporter # your tool name

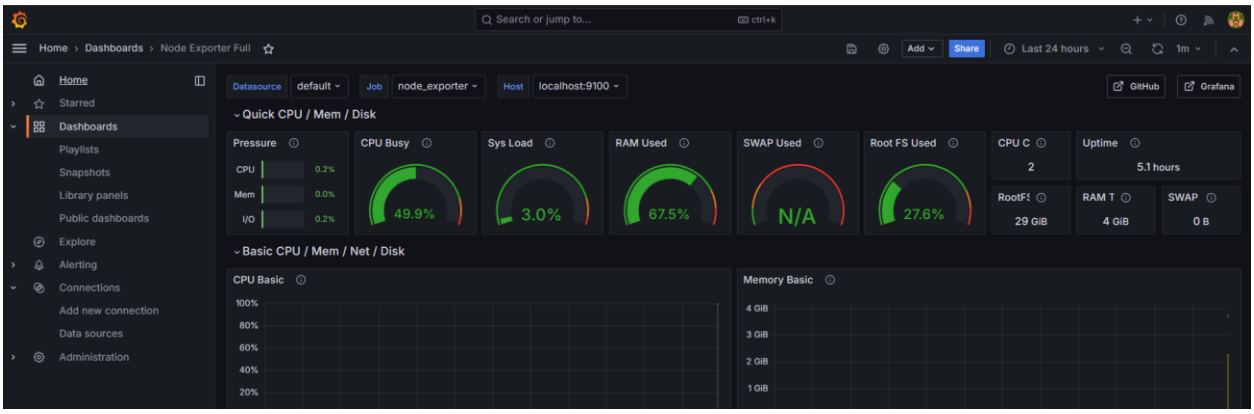
  static_configs:

   - targets: ['localhost:9100']



Goto the Grafana





What are we downloading the under the particular server.

Master server

```
azadmin@Master-server:~$ ls
1.sh  2.sh  bind.yml  cred.yml  role.yml  svc.yml
azadmin@Master-server:~$
```

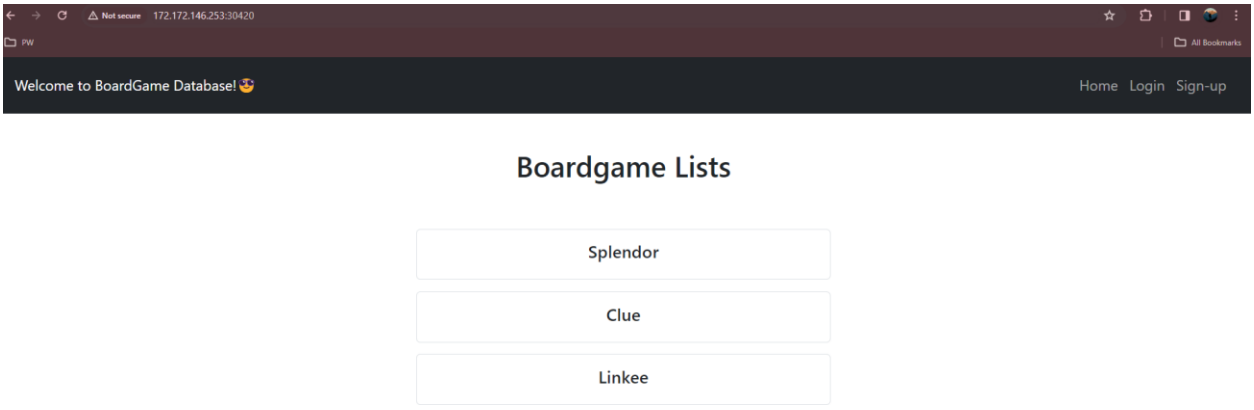
Slave server

```
Last login: Tue Apr  9 13:56:47 2024 from 103.27.51.228
azaadmin@Slave-server:~$ ls
1.sh
azaadmin@Slave-server:~$
```

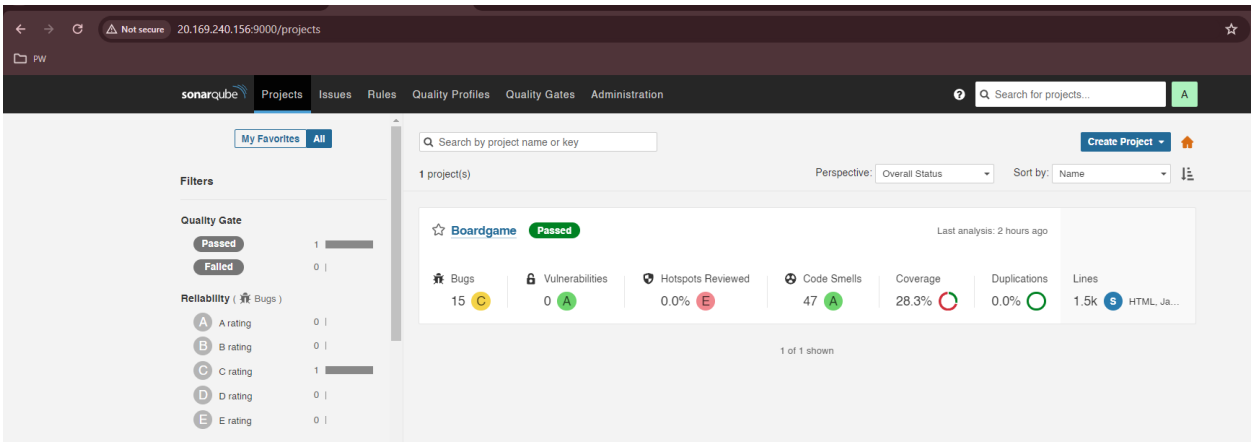
Runner

```
azadmin@Runner-server:~$ ls
actions-runner  monitoring  trivy.sh
azadmin@Runner-server:~$ cd monitoring/
azadmin@Runner-server:~/monitoring$ ls
blackbox-tool  grafana-enterprise_10.4.1_amd64.deb  node_exporter  prometheus
azadmin@Runner-server:~/monitoring$
```

Projected output



SonarQube check



Error, we faced when setup the pipeline.

```
1.sh  2.sh
azadmin@Master-server:~$ ./2.sh
I0409 13:44:47.035989   17061 version.go:256] remote version is much newer: v1.29.3; falling back to: stable-1.28
[init] Using Kubernetes version: v1.28.8
[preflight] Running pre-flight checks
error execution phase preflight: [preflight] Some fatal errors occurred:
        [ERROR Port-6443]: Port 6443 is in use
        [ERROR Port-10259]: Port 10259 is in use
        [ERROR Port-10257]: Port 10257 is in use
        [ERROR FileAvailable--etc-kubernetes-manifests-kube-apiserver.yaml]: /etc/kubernetes/manifests/kube-apiserver.yaml already exists
        [ERROR FileAvailable--etc-kubernetes-manifests-kube-controller-manager.yaml]: /etc/kubernetes/manifests/kube-controller-manager.yaml already exists
        [ERROR FileAvailable--etc-kubernetes-manifests-kube-scheduler.yaml]: /etc/kubernetes/manifests/kube-scheduler.yaml already exists
        [ERROR FileAvailable--etc-kubernetes-manifests-etcd.yaml]: /etc/kubernetes/manifests/etcd.yaml already exists
        [ERROR Port-10250]: Port 10250 is in use
        [ERROR Port-2379]: Port 2379 is in use
        [ERROR Port-2380]: Port 2380 is in use
        [ERROR DirAvailable--var-lib-etcd]: /var/lib/etcd is not empty
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher
cp: overwrite '/home/azadmin/.kube/config'?
```