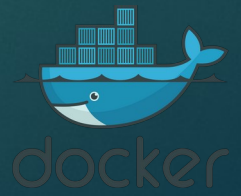
VIRTUALIZATION & CLOUD COMPUTING



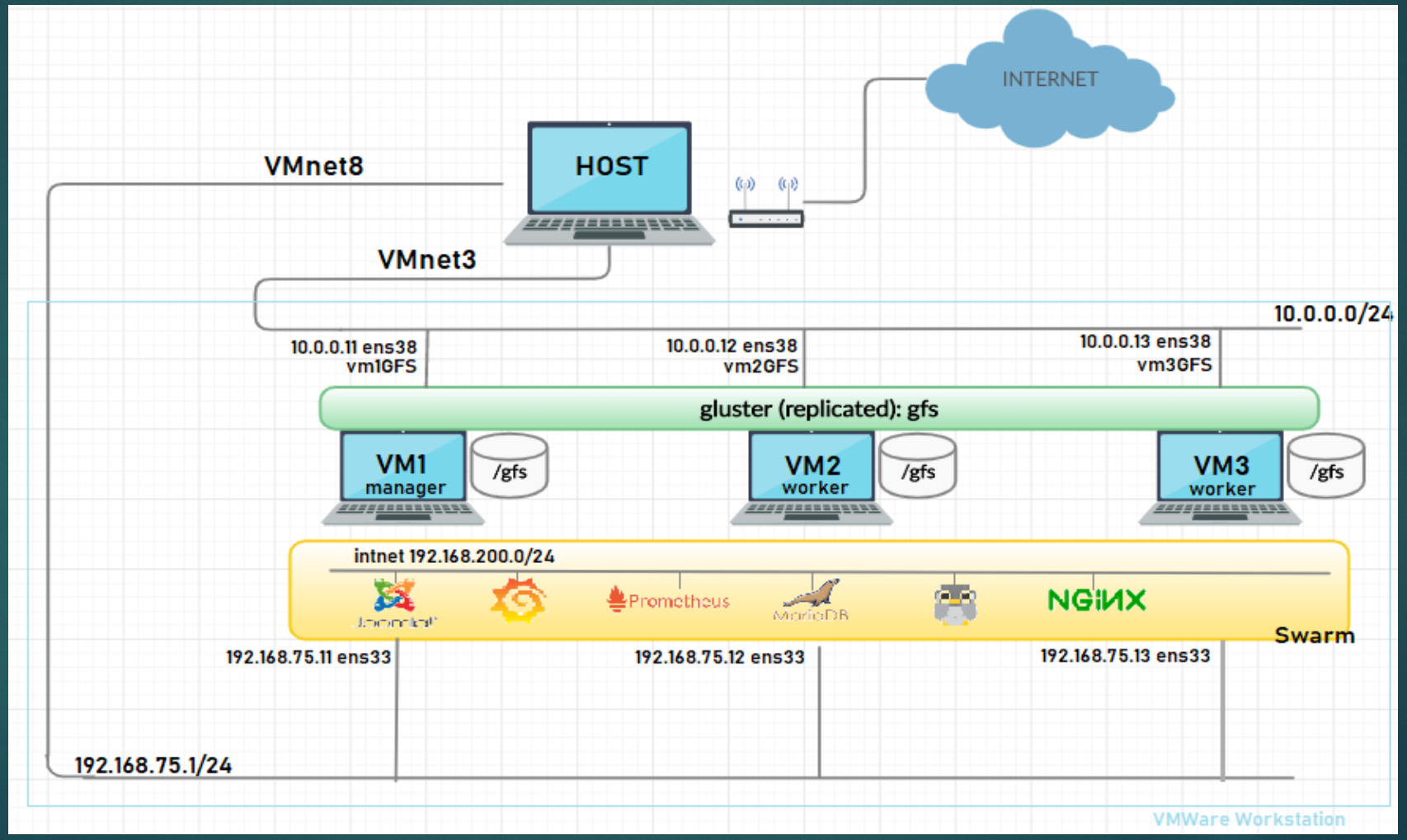
AUTHORS:

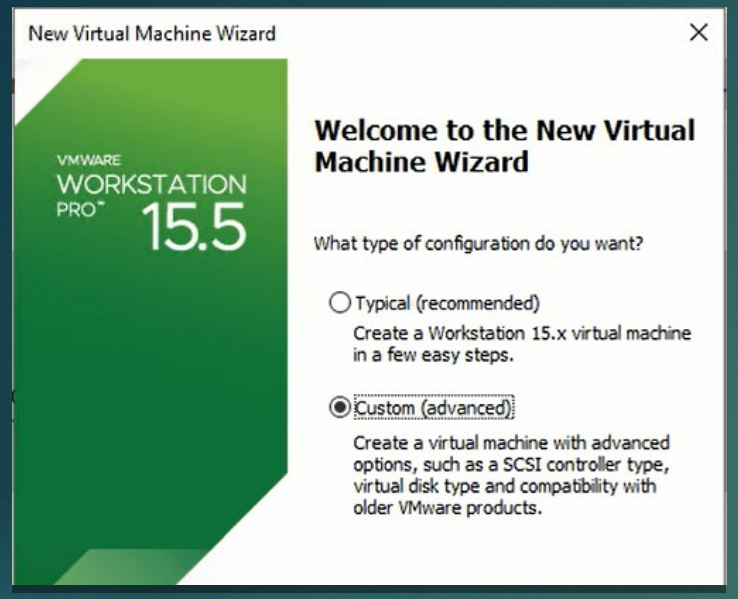
SENATORI DAVIDE

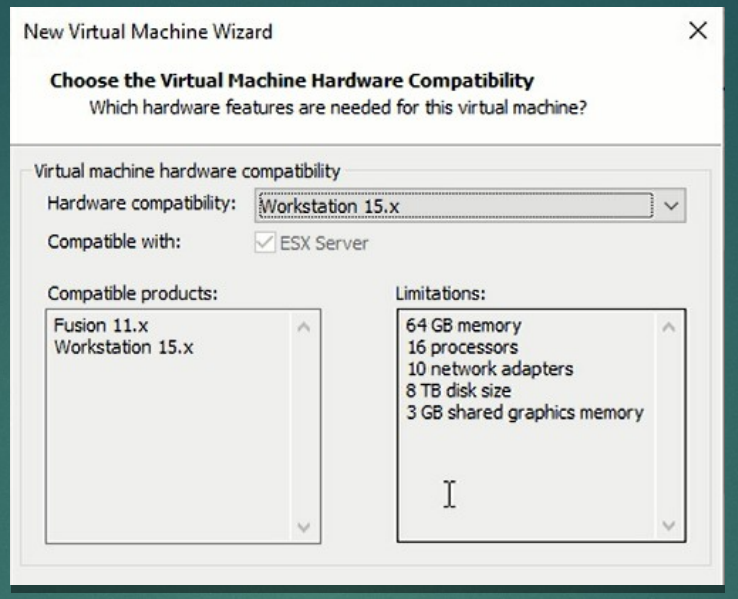
XIA CHENGHAO

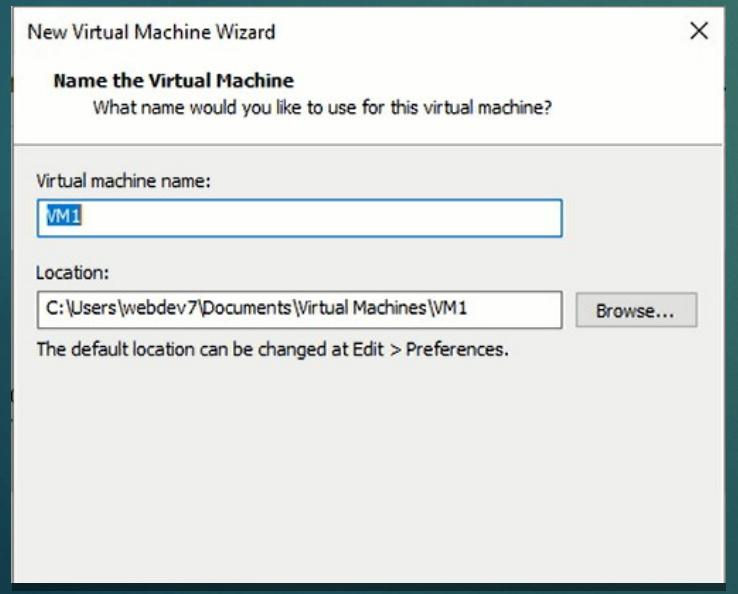
INTRODUZIONE

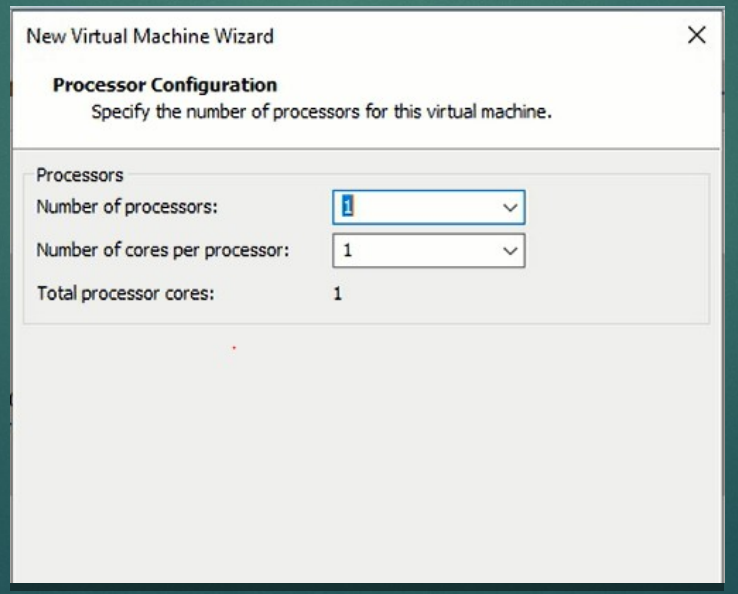
L’obiettivo del progetto è di implementare una applicazione multi-tier deployata in un Docker Swarm attraverso 3 Virtual Machines.



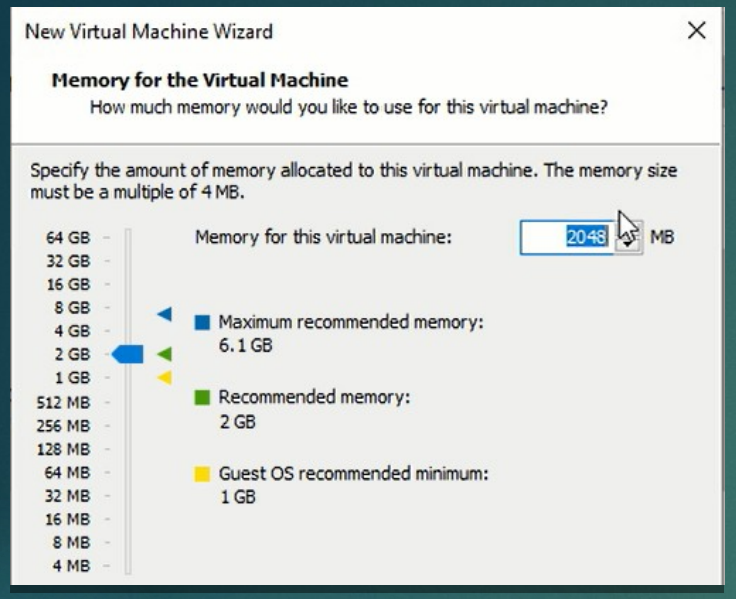


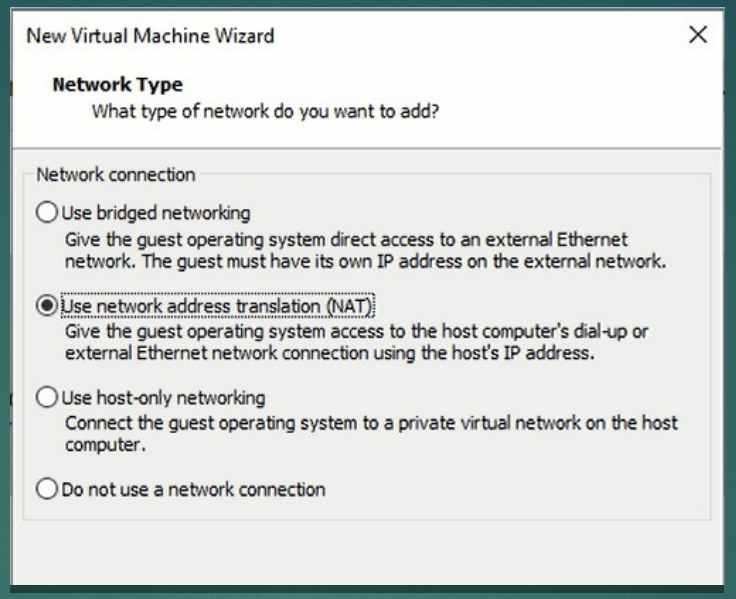


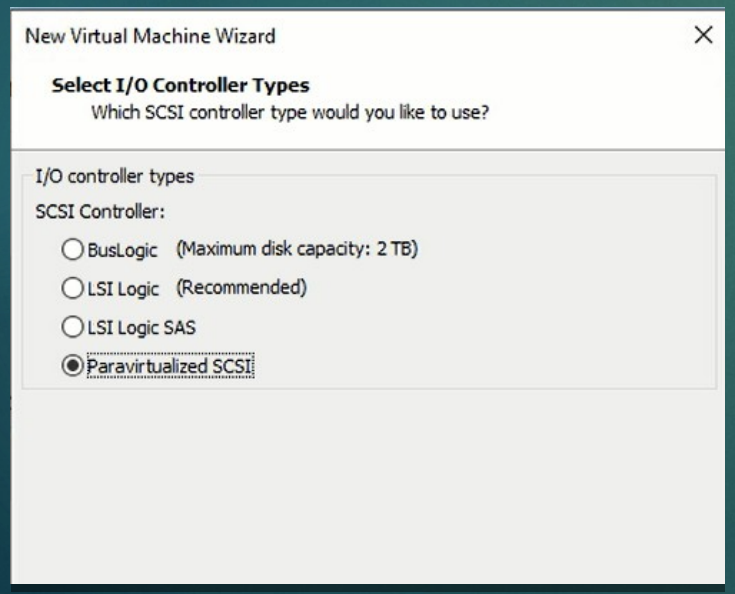


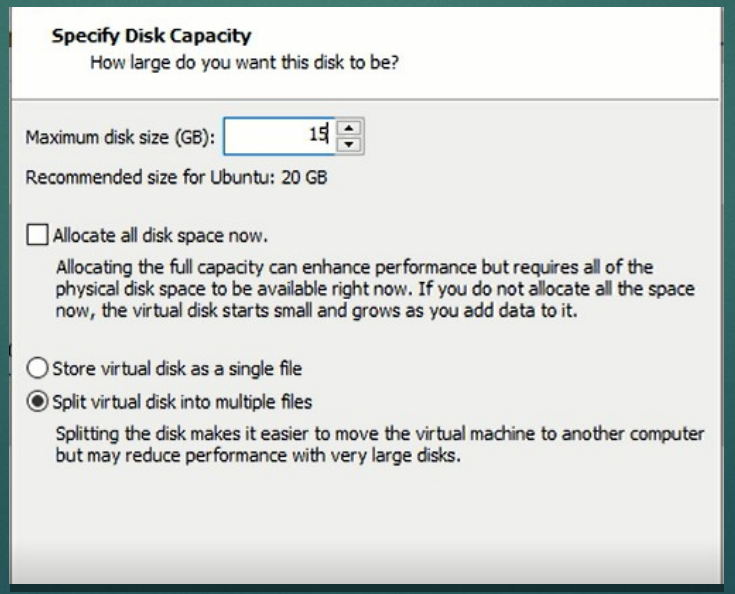


CONFIGURAZIONE VMS

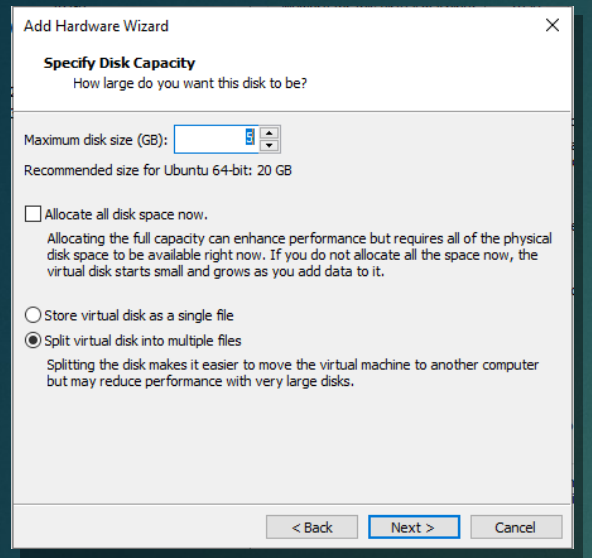


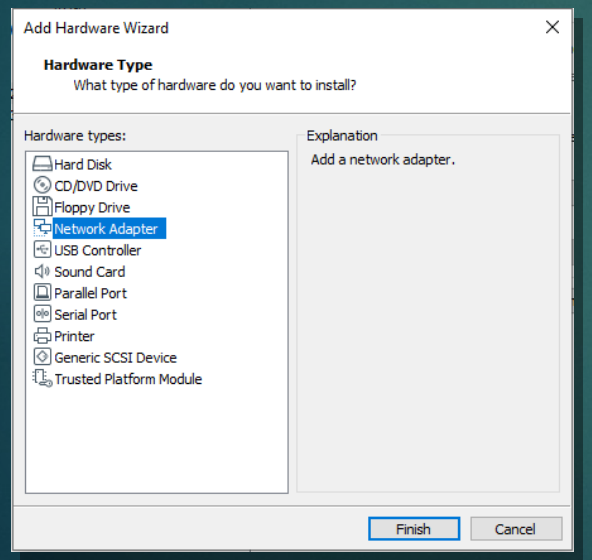


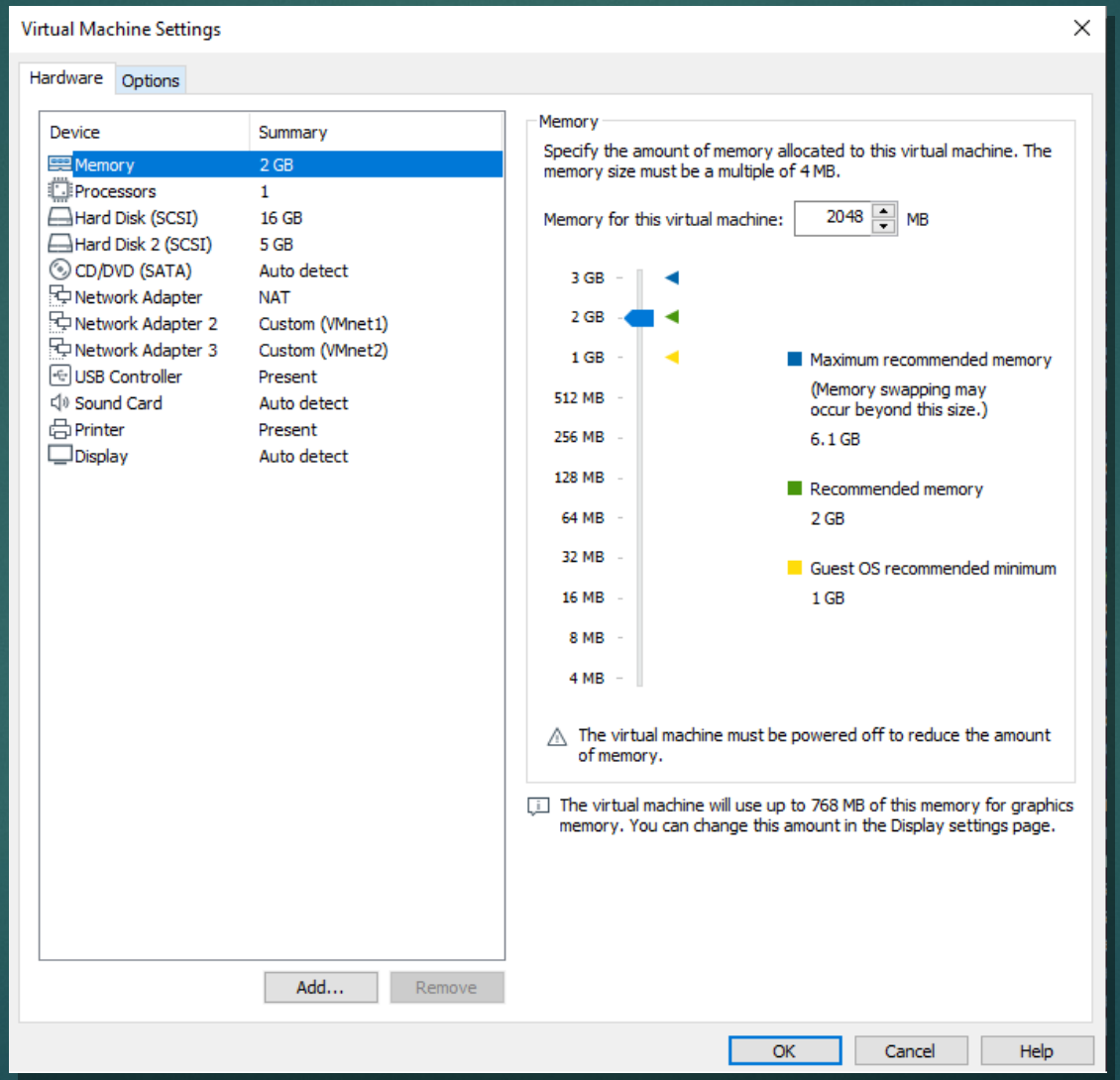




CONFIGURAZIONE VMS







CONFIGURAZIONE VMS

CONFIGURAZIONE HD & Docker

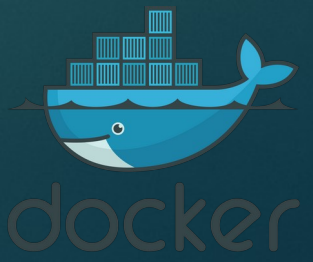
```
dav@ubuntu: ~
dav@ubuntu:~$ sudo fdisk -l
Disk /dev/fd0: 1.4 MiB, 1474560 bytes, 2880 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x90909090

Device     Boot      Start        End    Sectors  Size Id Type
/dev/fd0p1       2425393296 4850786591 2425393296  1.1T 90 unknown
/dev/fd0p2       2425393296 4850786591 2425393296  1.1T 90 unknown
/dev/fd0p3       2425393296 4850786591 2425393296  1.1T 90 unknown
/dev/fd0p4       2425393296 4850786591 2425393296  1.1T 90 unknown


Disk /dev/sda: 15 GiB, 16106127360 bytes, 31457280 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x960dd11f

Device     Boot    Start      End  Sectors  Size Id Type
/dev/sda1  *        2048 29456383 29454336   14G 83 Linux
/dev/sda2       29458430 31455231  1996802  975M  5 Extended
/dev/sda5       29458432 31455231  1996800  975M 82 Linux swap / Solaris
```

```
dav@ubuntu: ~
dav@ubuntu:~$ sudo mkfs.ext4 /dev/sdb
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 1310720 4k blocks and 327680 inodes
Filesystem UUID: 9befd926-2524-4f83-b861-ca5451058b0b
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
dav@ubuntu: ~
dav@ubuntu:~$ sudo apt-get install \
> apt-transport-https\
> ca-certificates \
> curl \
> gnupg-agent \
> software-properties-common
```

```
dav@ubuntu: ~
dav@ubuntu:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt
-key add -
OK
dav@ubuntu:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.
com/linux/ubuntu $(lsb_release -cs) stable"
dav@ubuntu:~$ sudo apt-get update
Get:1 https://download.docker.com/linux/ubuntu xenial InRelease [66.2 kB]
Hit:2 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Get:4 https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages [13.
1 kB]
Hit:5 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:6 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
Fetched 79.3 kB in 0s (145 kB/s)
Reading package lists... Done
dav@ubuntu:~$
```

```
File  Edit  View  Search  Terminal  Help
dav@vm1:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io

dav@ubuntu:~$ sudo usermod -aG docker dav
dav@ubuntu:~$ sudo systemctl restart docker
```



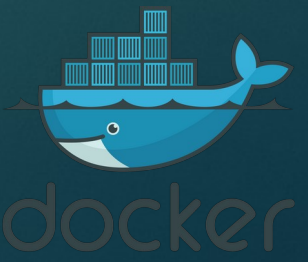
Modifica file hosts ed interfaces:

```
dav@vm1:~$ sudo nano  /etc/host

127.0.0.1       localhost
127.0.1.1       vm1

192.168.75.11 vm1
192.168.75.12 vm2
192.168.75.13 vm3

10.0.0.11 vm1GFS
10.0.0.12 vm2GFS
10.0.0.13 vm3GFS
```

```
dav@vm1:~$ sudo nano /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto ens39
iface ens39 inet static
address 10.0.0.11
netmask 255.255.255.0
network 10.0.0.0
gateway 10.0.0.2

auto ens33
iface ens33 inet static
address 192.168.75.11
netmask 255.255.255.0
network 192.168.75.0
gateway 192.168.75.2
```

CONFIGURAZIONE IP ADDRESS

A questo punto, viene eseguita la clonazione della VM1 in VM2 e VM3. Vengono inoltre modificati i valori dei files /etc/hosts, /etc/network/interfaces ed /etc/host nelle due nuove VM.



CREAZIONE DOCKER SWARM

E’ stato creato un nuovo swarm con il seguente cmd:

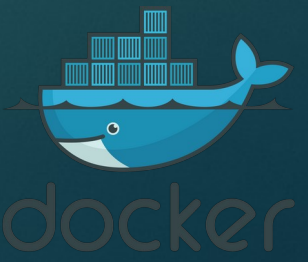
```
dav@vm1:~$ sudo docker swarm init --advertise-addr ens33
[sudo] password for dav:
Swarm initialized: current node (lv668l9aepejl2wb8f8dtlyv9) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-0hrrk1f9adtls4bc7egi2uiifx7h176vb18v10ios
umdqnpcvo-934cpzv4iitd7gzw3uyhh0uv7 192.168.75.11:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow
 the instructions.
```

In seguito è stato eseguito il seguente cmd nelle altre due VM:

```
dav@vm3:~$ docker swarm join --token SWMTKN-1-4b6qkg951n22jc1f4fjfg5xcjks9jvquj1
i5h8l8ja0tasglz0-7ju8db7nwotmirxonsipi6hpg 192.168.1.172:2377
```

Controllo:

```
dav@vm1:~$ docker node ls
ID                            HOSTNAME            STATUS              AVAILABILITY        MANAGER STATUS      ENGINE VERSION
jmqqamktw541ngrkriuk6y6d1     ubuntu              Ready               Active                                  19.03.8
fbulkznkmv8e7k2ll7ip66we2 *   vm1                 Ready               Active              Leader              19.03.8
gsiti4bg5rp3krz4u8c2q2gcw     vm3                 Ready               Active                                  19.03.8
```



CONFIGURAZIONE GLUSTERFS

Installazione e avvio di GlusterFs:

```
dav@vm3:~$ sudo apt -y install glusterfs-server glusterfs-client
```

Start del service e peer con gli altri nodi:

```
dav@vm3:~$ sudo gluster peer probe vm1GFS
[sudo] password for dav:
peer probe: success. Host vm1GFS port 24007 already in peer list
dav@vm3:~$ sudo gluster peer probe vm2GFS
peer probe: success. Host vm2GFS port 24007 already in peer list
dav@vm3:~$ sudo gluster peer status
Number of Peers: 2

Hostname: vm2GFS
Uuid: 6c444dc7-498b-413e-8091-12e3233c463e
State: Peer in Cluster (Connected)

Hostname: vm1GFS
Uuid: 5714bbd6-6a82-434b-96a1-4c6ec669ba5c
State: Peer in Cluster (Connected)
```

Setup delle cartelle di GlusterFS:

```
mkdir -p /gluster/bricks/1
mkdir -p /gluster/bricks/2
mkdir -p /gluster/bricks/3
```

```
echo '/dev/sdb /gluster/bricks/1 ext4 defaults 0 0' >> /etc/fstab
echo '/dev/sdb /gluster/bricks/2 ext4 defaults 0 0' >> /etc/fstab
echo '/dev/sde /gluster/bricks/3 ext4 defaults 0 0' >> /etc/fstab
```

```
dav@vm1:~$ sudo mount -a
```

CONFIGURAZIONE GLUSTERFS

Creazione e start del nuovo volume gfs:

```
dav@vm1:~$ sudo gluster volume create gfs replica 3 \
> vm1GFS:/gluster/bricks/1/brick \
> vm2GFS:/gluster/bricks/2/brick \
> vm3GFS:/gluster/bricks/3/brick
```

```
dav@vm1:~$ sudo gluster volume start
```

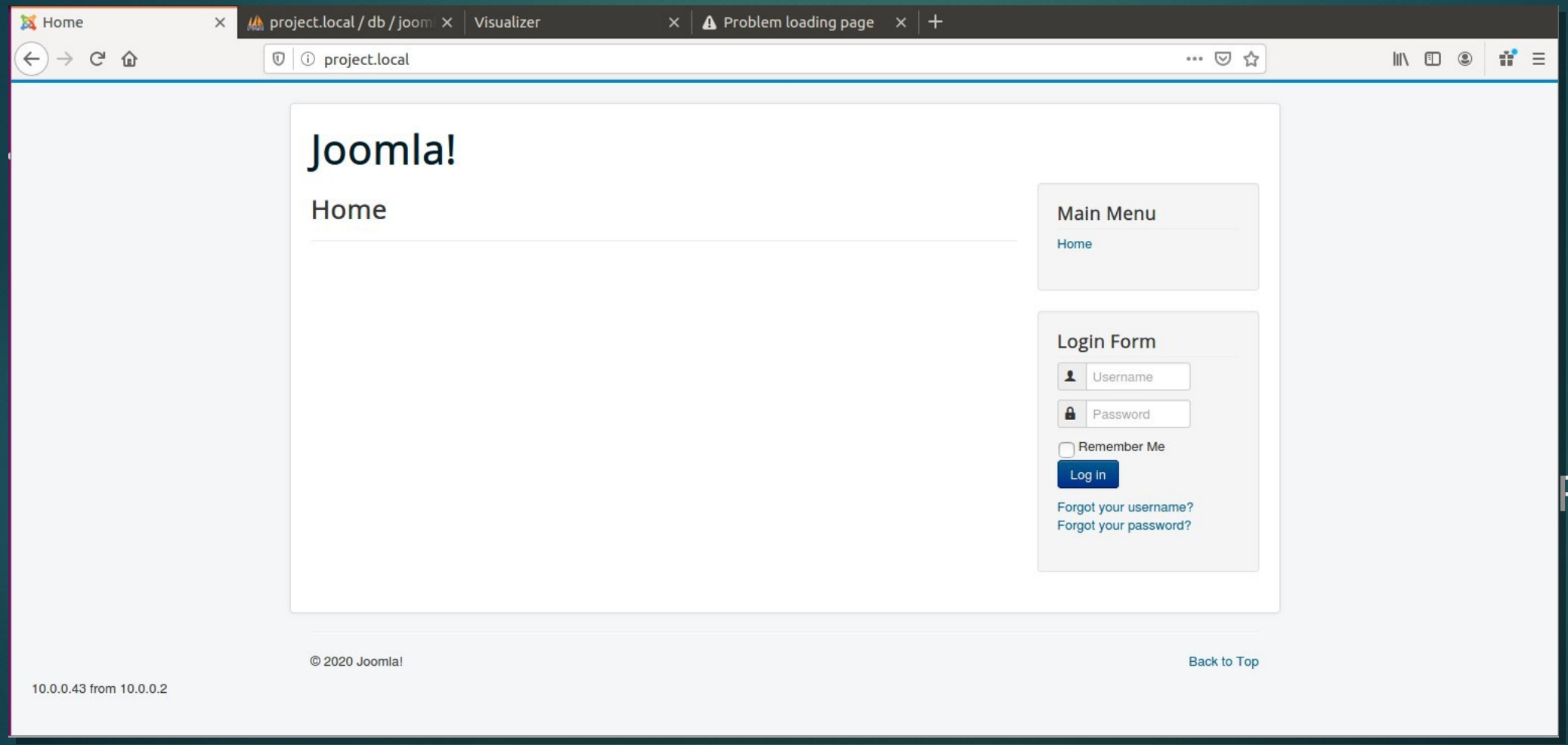
Controllo info e stato:

```
dav@vm1:~$ sudo gluster volume status gfs
[sudo] password for dav:
Status of volume: gfs
Gluster process                                 TCP Port  RDMA Port  Online  Pid
------------------------------------------------------------------------------
Brick vm1GFS:/gluster/bricks/1/brick            49152     0          Y       1743
Brick vm2GFS:/gluster/bricks/2/brick            49152     0          Y       1773
Brick vm3GFS:/gluster/bricks/3/brick            49152     0          Y       5256
NFS Server on localhost                         N/A       N/A        N       N/A
Self-heal Daemon on localhost                   N/A       N/A        Y       1416
NFS Server on vm2GFS                            N/A       N/A        N       N/A
Self-heal Daemon on vm2GFS                      N/A       N/A        Y       1656
NFS Server on vm3GFS                            N/A       N/A        N       N/A
Self-heal Daemon on vm3GFS                      N/A       N/A        Y       5278

Task Status of Volume gfs
------------------------------------------------------------------------------
There are no active volume tasks
```

Setup security:

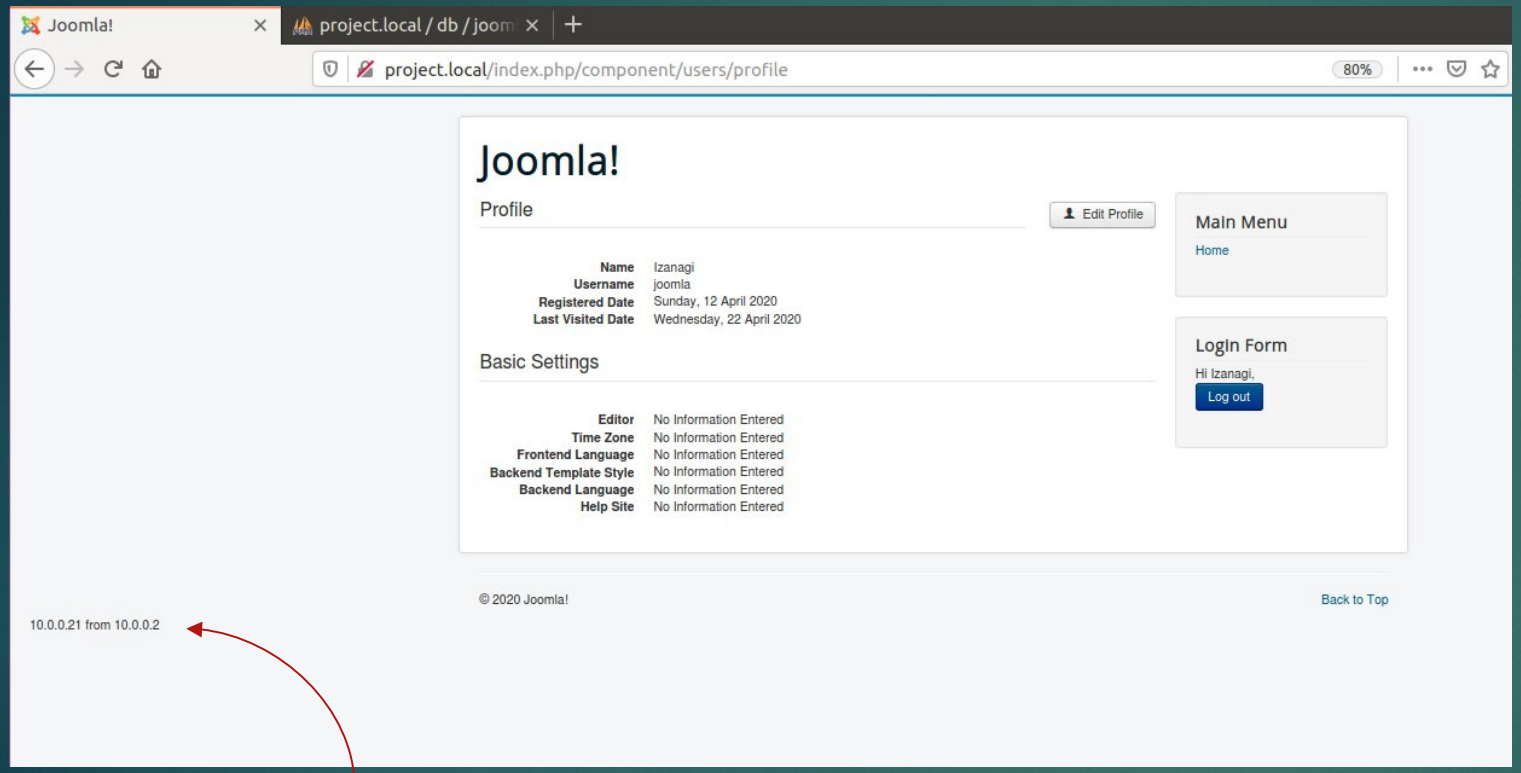
```
dav@vm1:~$ sudo gluster volume set gfs auth.allow 10.0.0.11,10.0.0.12,10.0.0.13
volume set: success
dav@vm1:~$ sudo gluster volume set gfs nfs.disable Off
volume set: success
```

JOOMLA



RISULTATO JOOMLA

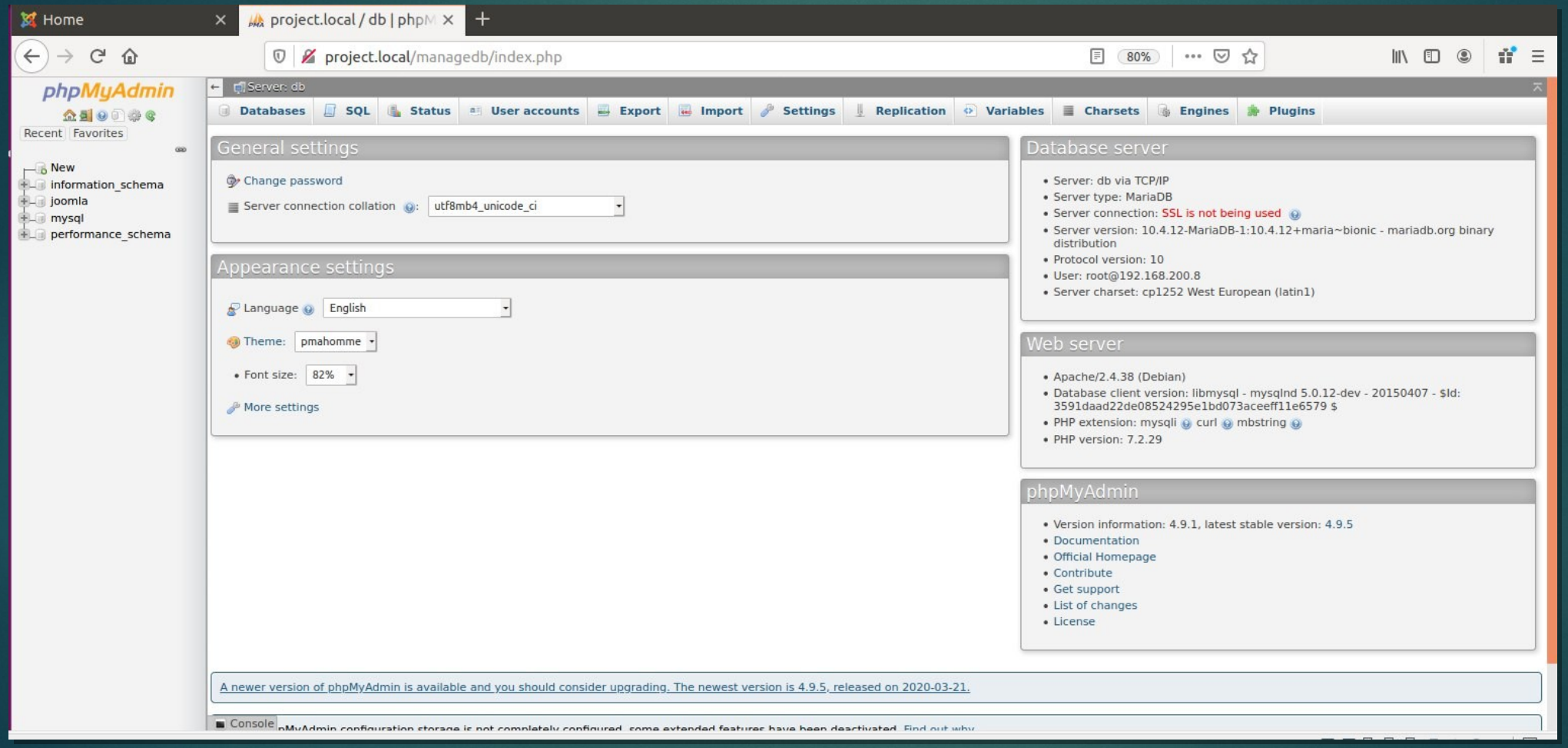
JOOMLA




```
version: "3"

services:
 joomla:
  image: myjoomla
  hostname: jo
  container_name: jo
  ports:
   - 8081:80
  depends_on:
   - db
  deploy:
   mode: replicated
   replicas: 3
  labels:
   - "com.docker.lb.hosts=project.local"#
   - "com.docker.lb.sticky_session_cookie=session"
   - "com.docker.lb.port=3001"
  volumes:
   - /gfs/joomla/:/var/www/html/
  networks:
   - intnet
#   - extnet
```

```
echo $_SERVER['SERVER_ADDR'];
echo " from ";
echo gethostbyaddr($_SERVER['REMOTE_ADDR']);
```

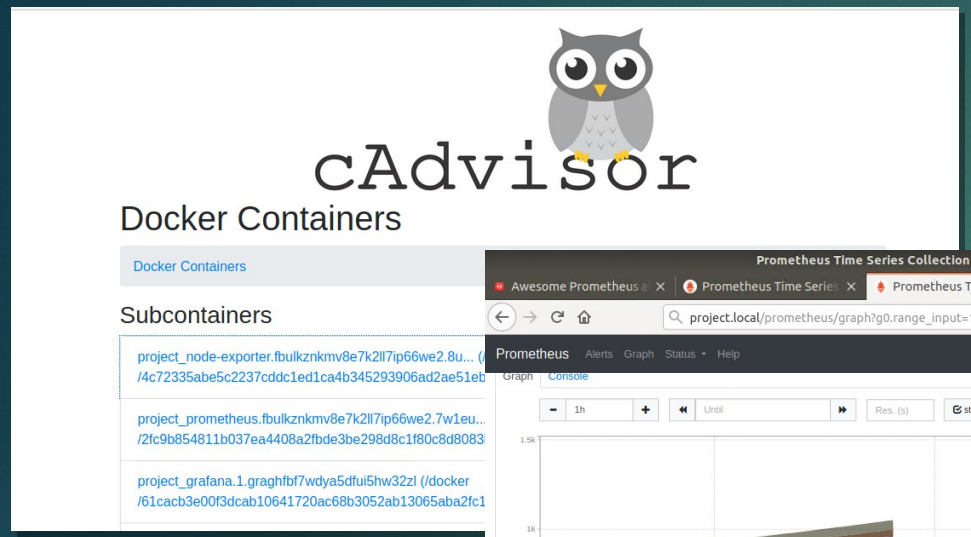
MARIADB

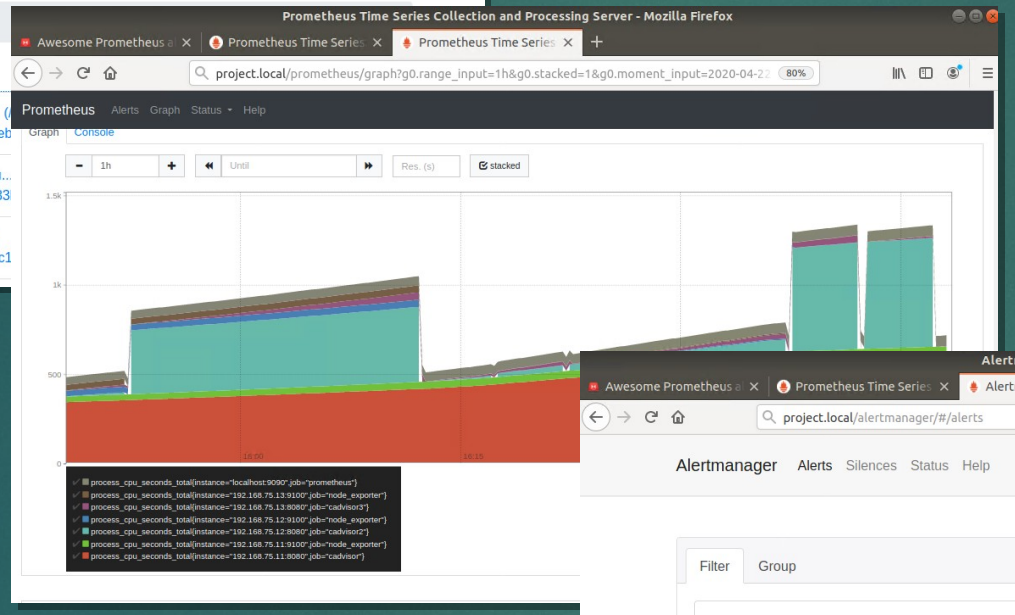


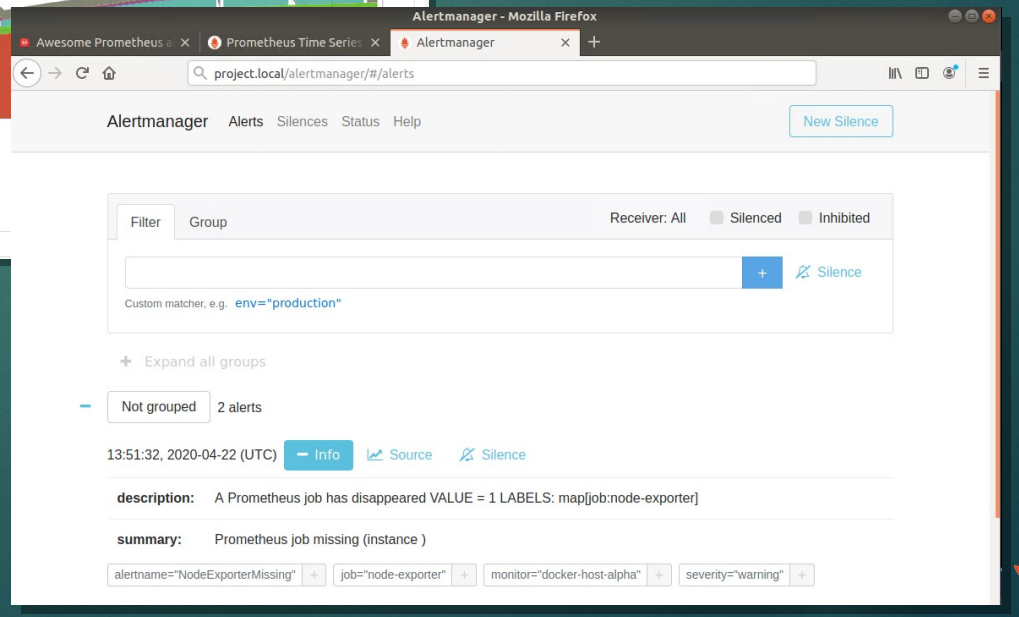
RISULTATO MARIADB

MariaDB MySQL

MONITORING

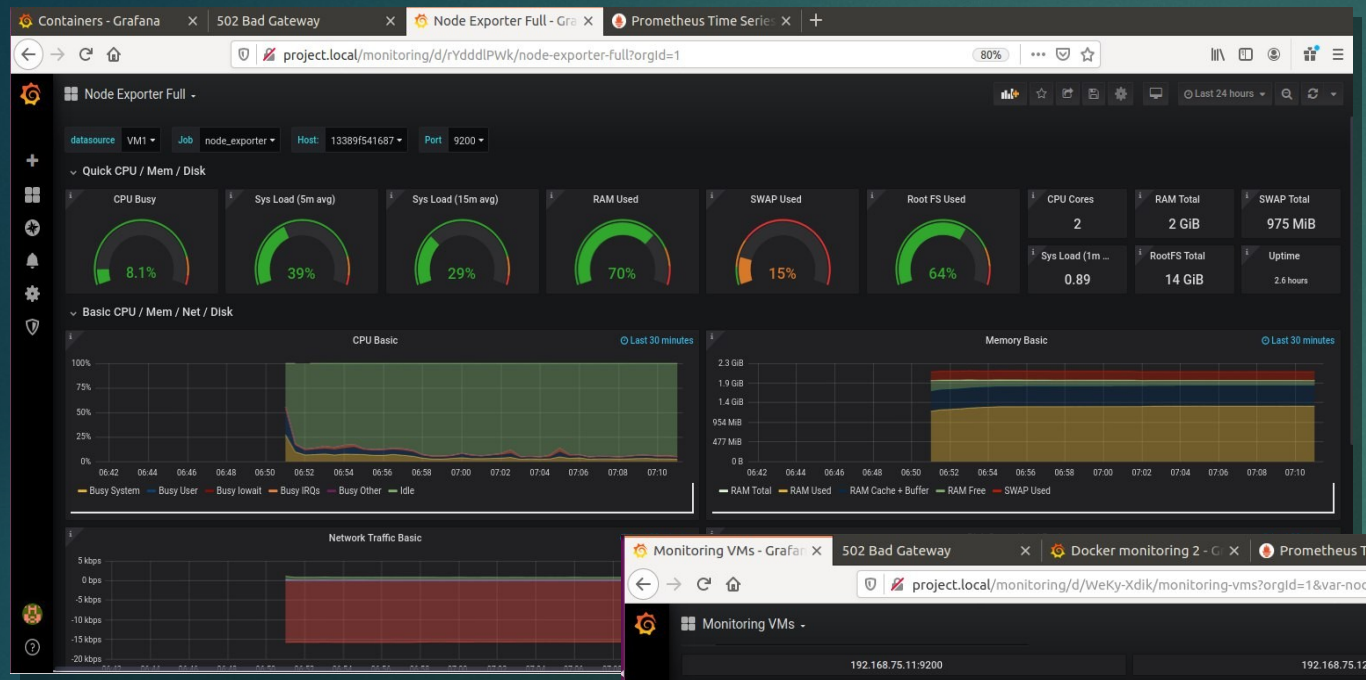

cAdvisor
Docker Containers
Docker Containers
Subcontainers
project_node-exporter.fbulkznkmv8e7k2ll7ip66we2.8u... (/
/4c72335abe5c2237cddc1ed1ca4b345293906ad2ae51eb
project_prometheus.fbulkznkmv8e7k2ll7ip66we2.7w1eu...
/2fc9b854811b037ea4408a2fbde3be298d8c1f80c8d8083
project_grafana.1.graghfbf7wdya5dfui5hw32zl (/docker
/61cacb3e00f3dcab10641720ac68b3052ab13065aba2fc1

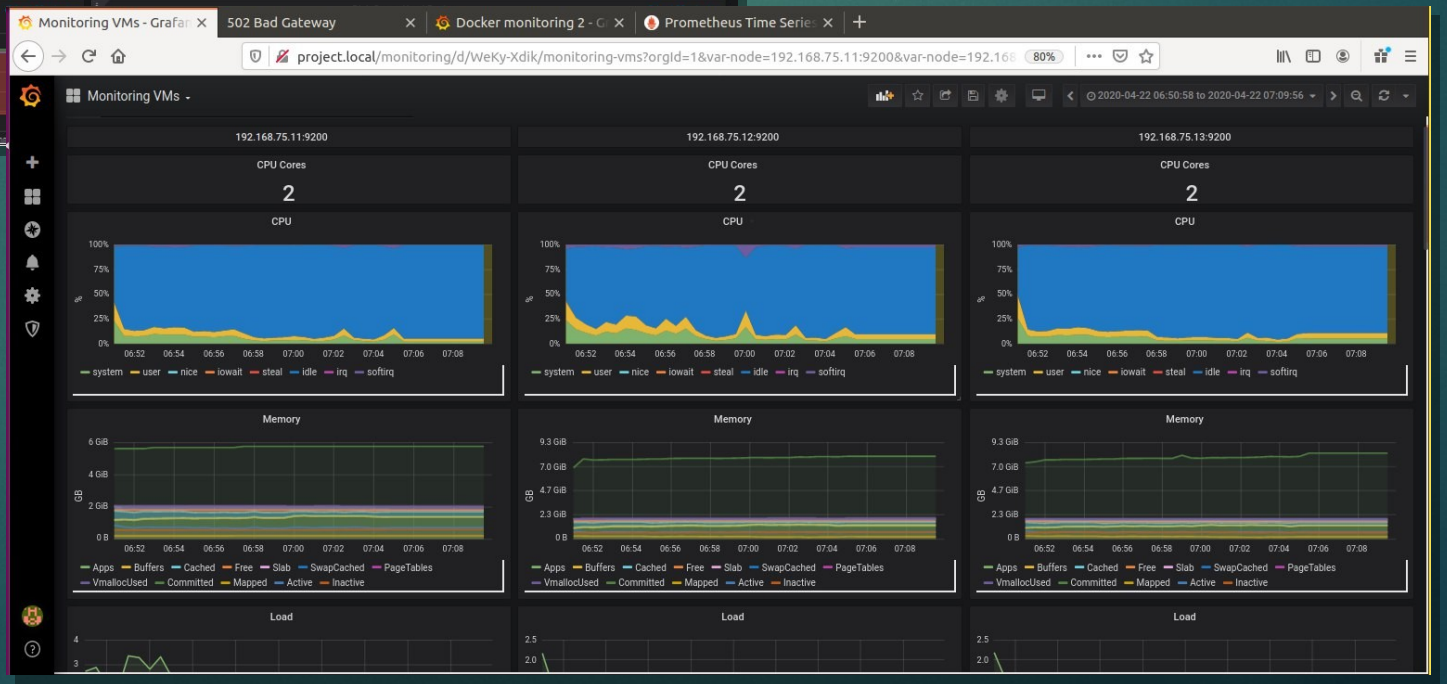

Prometheus Time Series Collection and Processing Server - Mozilla Firefox
Awesome Prometheus a
Prometheus Time Series
Prometheus Time Series
project.local/prometheus/graph?g0.range_input=1h&g0.stacked=1&g0.moment_input=2020-04-22
Prometheus
Alerts
Graph
Status
Help
Graph
Console
1h
Until
Res. (s)
stacked
1.5k
1k
500
0
process_cpu_seconds_total{instance="localhost:9090",job="prometheus"}
process_cpu_seconds_total{instance="192.168.75.13:9100",job="node_exporter"}
process_cpu_seconds_total{instance="192.168.75.13:8080",job="cadvisor3"}
process_cpu_seconds_total{instance="192.168.75.12:9100",job="node_exporter"}
process_cpu_seconds_total{instance="192.168.75.12:8080",job="cadvisor2"}
process_cpu_seconds_total{instance="192.168.75.11:9100",job="node_exporter"}
process_cpu_seconds_total{instance="192.168.75.11:8080",job="cadvisor"}


Alertmanager - Mozilla Firefox
Awesome Prometheus a
Prometheus Time Series
Alertmanager
project.local/alertmanager/#/alerts
Alertmanager
Alerts
Silences
Status
Help
New Silence
Filter
Group
Receiver: All
Silenced
Inhibited
Silence
Custom matcher, e.g. env="production"
Expand all groups
Not grouped
2 alerts
13:51:32, 2020-04-22 (UTC)
Info
Source
Silence
description:
A Prometheus job has disappeared VALUE = 1 LABELS: map[job:node-exporter]
summary:
Prometheus job missing (instance)
alertname="NodeExporterMissing"
job="node-exporter"
monitor="docker-host-alpha"
severity="warning"


Prometheus

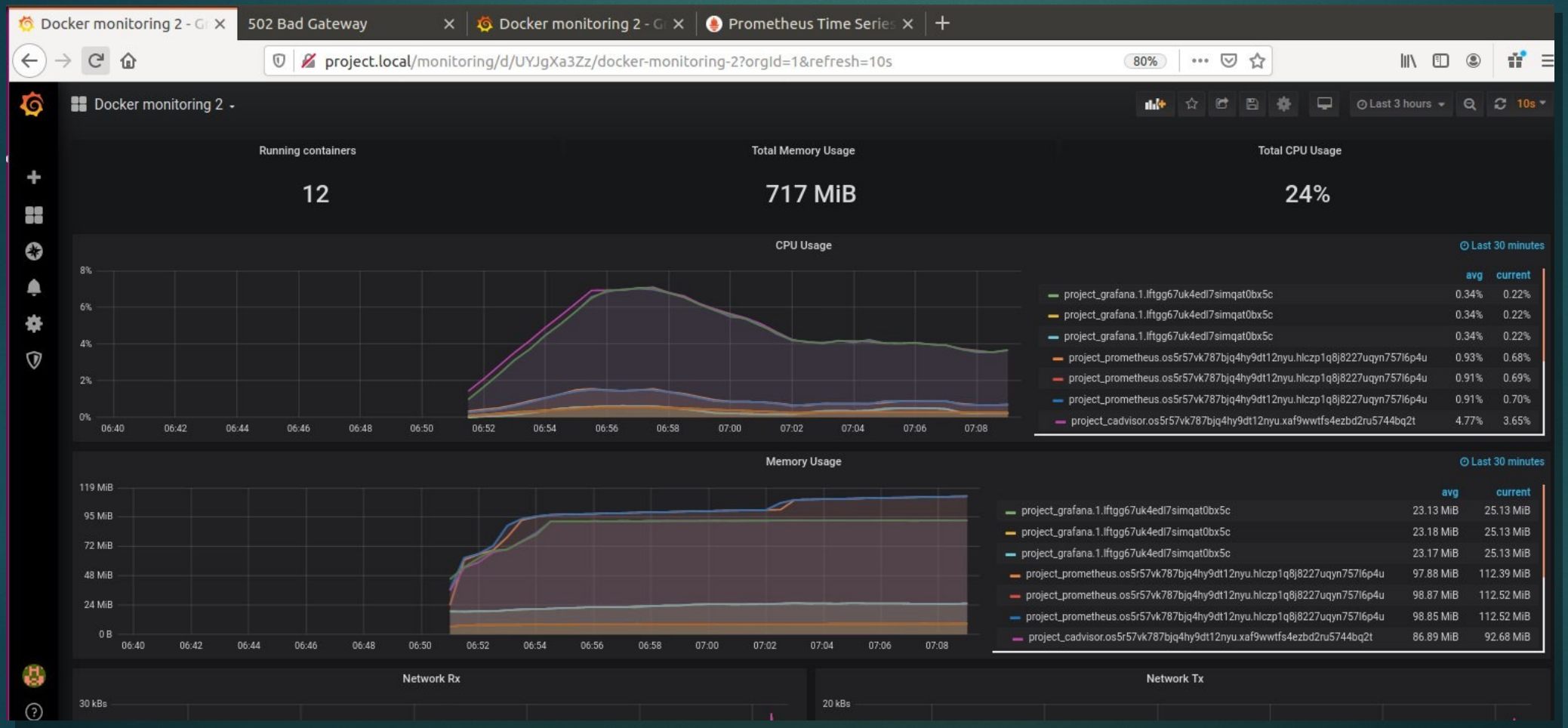
MONITORING





Monitoring VMS

MONITORING



Containers Monitoring

MONITORING

File /gfs/prometheus/alertmanager/config.yml :

```
route:
    group_by: [Alertname]
    receiver: email-me

receivers:
    - name: email-me
      email_configs:
      - to: davsen.ge@gmail.com
        from: senatori.projects@gmail.com
        smarthost: smtp.gmail.com:587
        auth_username: "senatori.projects@gmail.com"
        auth_identity: "senatori.projectsgmail.com"
        auth_password: "mypwsd"
```

File /gfs/prometheus/prometheus.yml

```
rule_files:
  - "alert.rules"

# alert
alerting:
  alertmanagers:
  - scheme: http
    static_configs:
    - targets:
      - "alertmanager:9093"
```

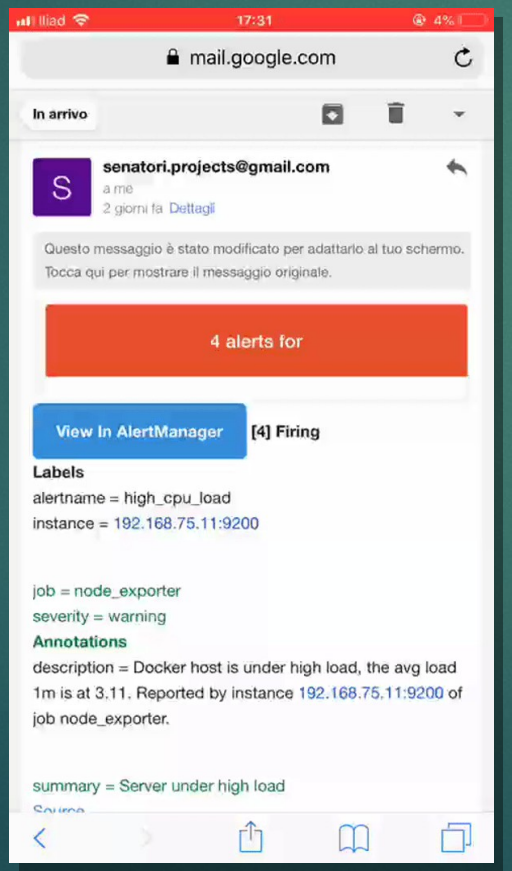
File docker-compose:

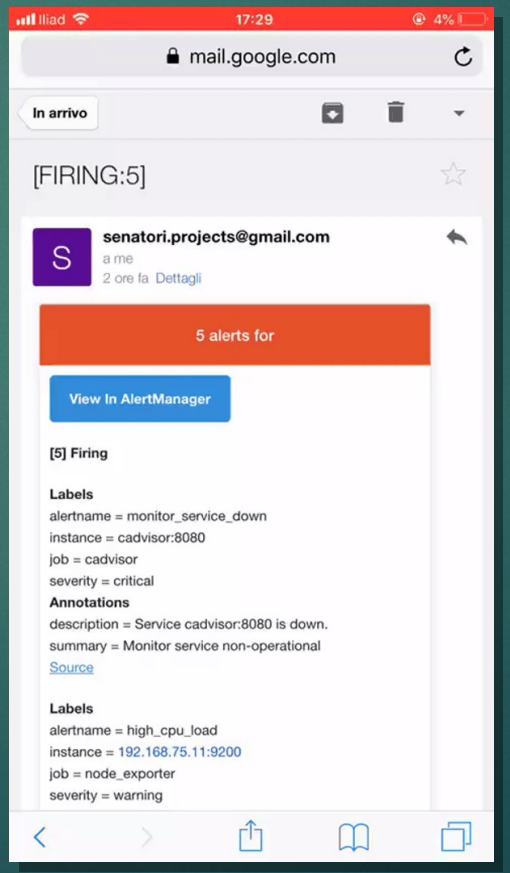
```
  alertmanager:
      image: prom/alertmanager
      ports:
        - 9093:9093
      volumes:
        - /gfs/prometheus/alertmanager/:/etc/alertmanager
      command:
        - '--config.file=/etc/alertmanager/config.yml'
        - '--storage.path=/alertmanager'
      deploy:
        placement:
          constraints:
            - node.role == manager
```

AlertManager

MONITORING

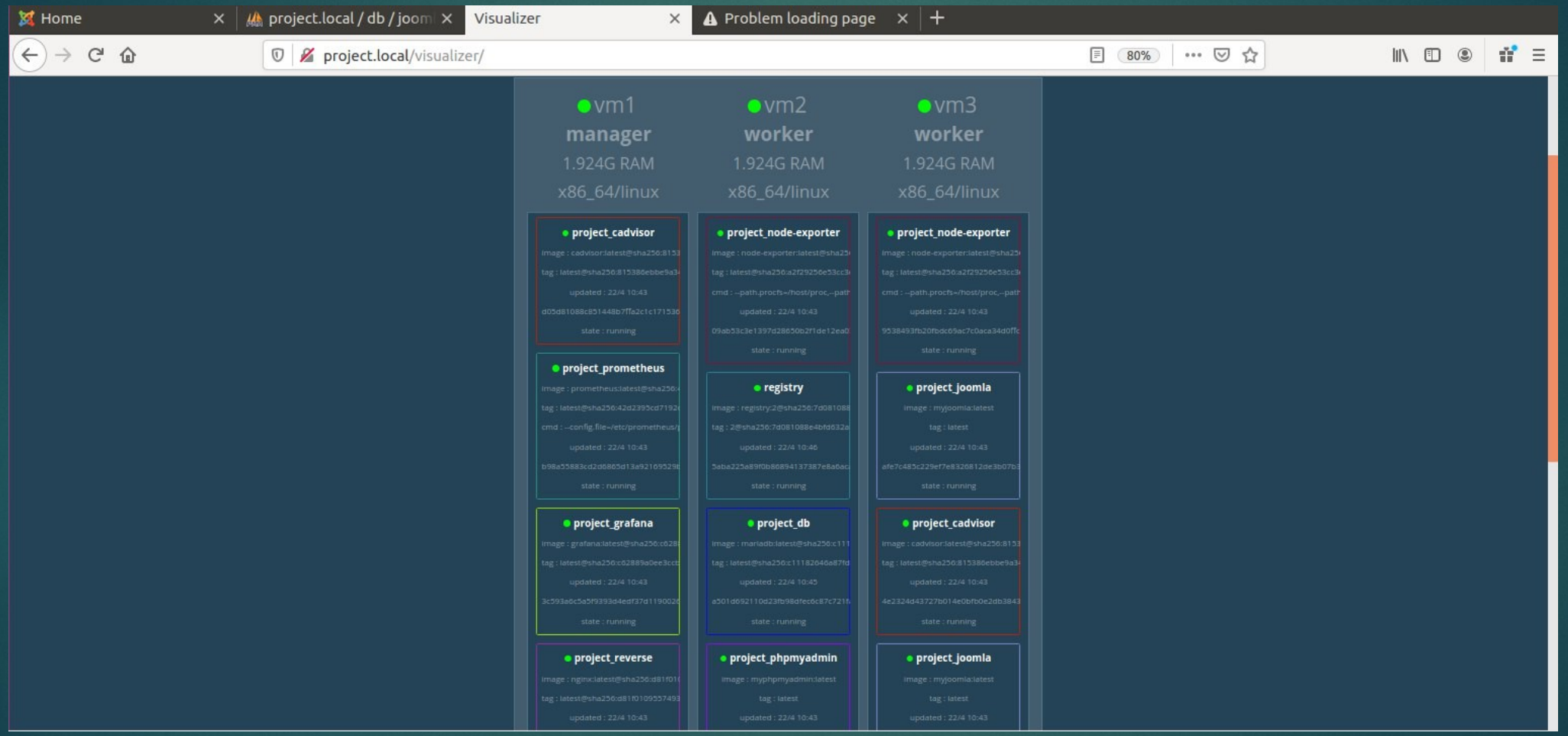
senatori.projects 20	**[FIRING:4]** - 4 alerts for View in AlertManager [4] Firing Labels alertname = high_cpu_load instan...
senatori.projects 11	**[FIRING:5]** - 5 alerts for View in AlertManager [5] Firing Labels alertname = monitor_service_do...
senatori.projects 2	**[FIRING:3]** - 3 alerts for View in AlertManager [3] Firing Labels alertname = monitor_service_do...
senatori.projects 3	**[FIRING:2]** - 2 alerts for View in AlertManager [2] Firing Labels alertname = high_memory_load s...





AlertManager

MONITORING



Visualizer

RISPOSTE E BONUS

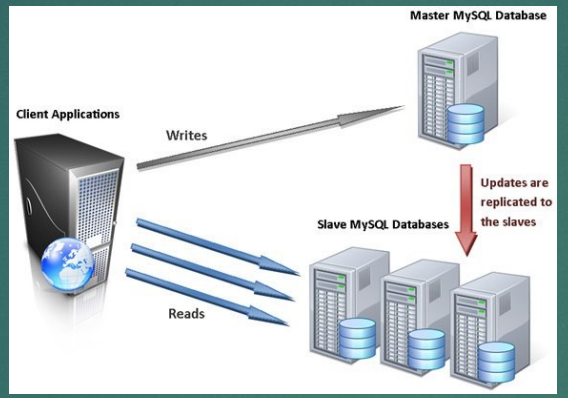
Split-brain

Soluzioni per attenuare:

1) Replica 3 volume

2) Arbitro

DB replicato

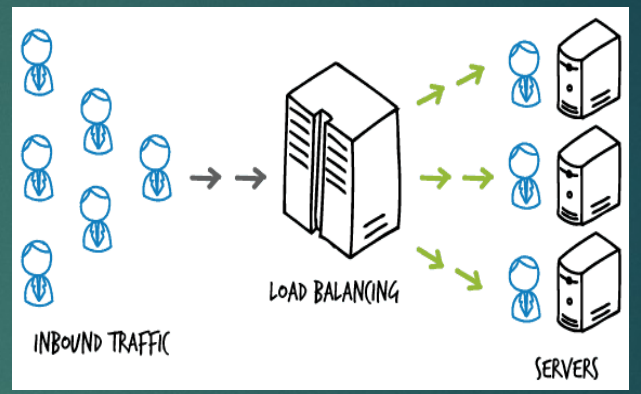


MySQL group replication (GR) è un server MySQL plugin che permette di creare repliche high-avaiabiles e fault-tolerant.

-Single primary: solo un server alla volta accetta update.

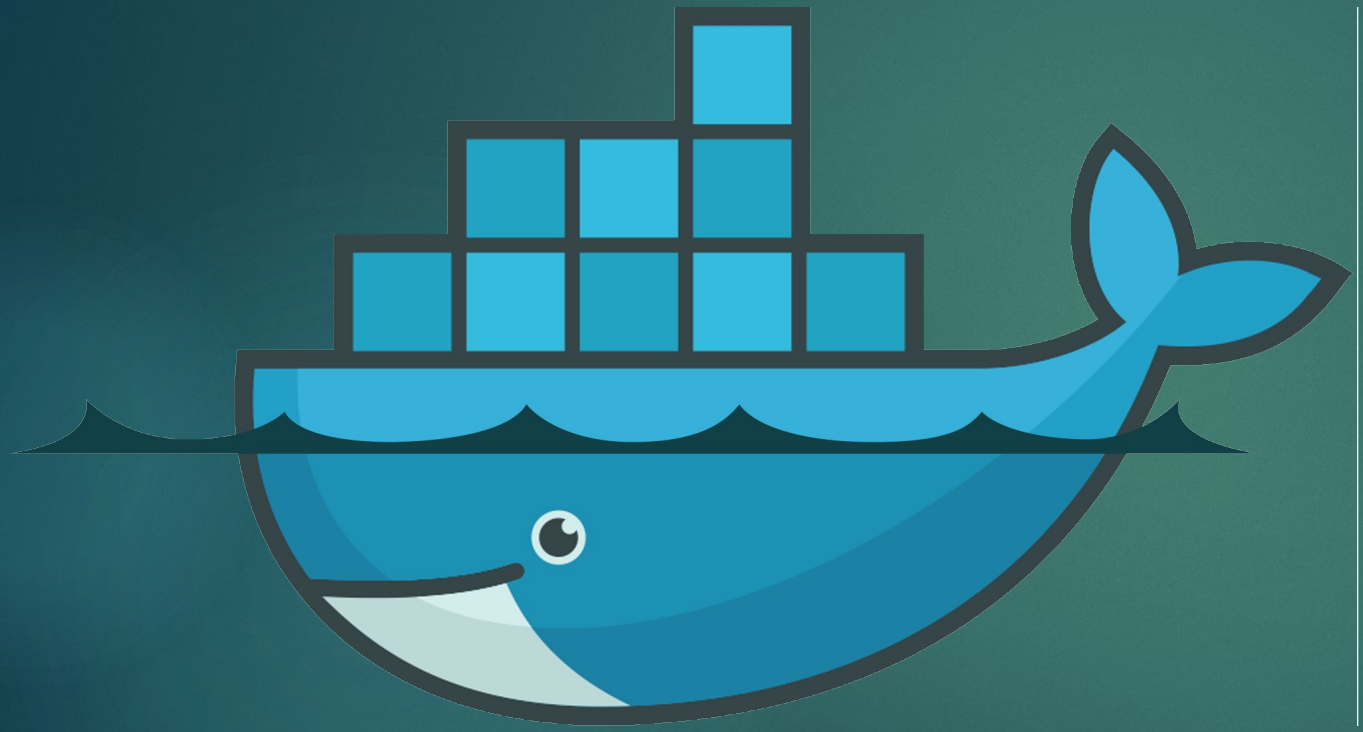
-Multi primary: tutti i servers accettano update.

Load Balancing



Implementato attraverso Nginx

Sviluppi futuri?
-Manager Fault tollerant



GRAZIE PER L'ATTENZIONE