EMAS-VNPT

TÀI LIỆU HƯỚNG DẪN CÀI ĐẶT

1. THIẾT LẬP BAN ĐẦU

- Các node:

  + Node 1 – gpu1: 192.168.5.61

  + Node 2 – gpu2: 192.168.5.62

  + Node 3 – GUI: 192.168.5.63

  + Node 4 – cpu1: 192.168.5.64

  + Node 5 – cpu2: 192.168.5.65

- Kích hoạt account root và cập nhật múi giờ:

```
sudo passwd
sudo timedatectl set-timezone Asia/Ho_Chi_Minh
```

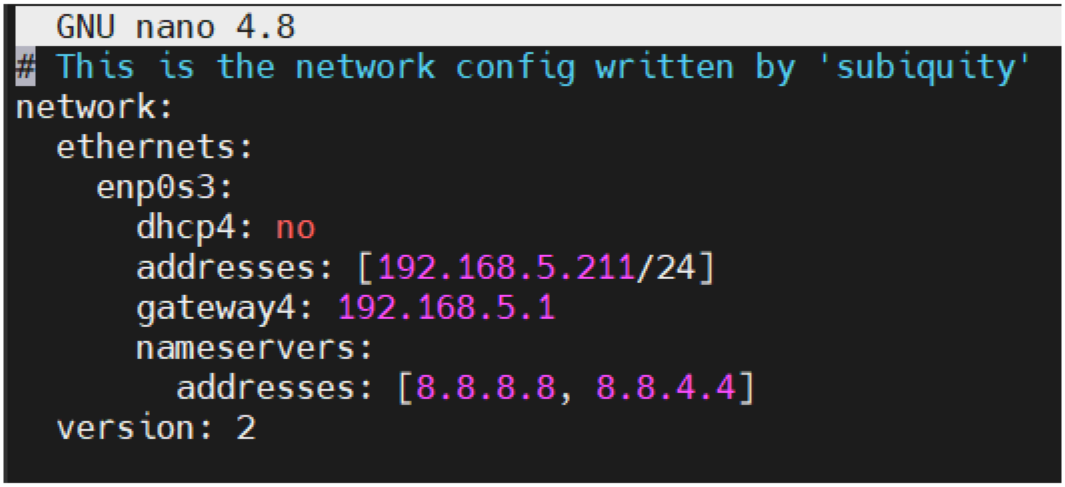
- Yêu cầu đặc quyền root:

```
sudo su
```

1.1. Cài đặt IP tĩnh

- Bước 1: Mở và chỉnh sửa cấu hình IP tương tự trên các node

```
nano /etc/netplan/00-installer-config.yaml
```



```
  GNU nano 4.8
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.5.211/24]
      gateway4: 192.168.5.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
  version: 2
```

- Bước 2: Apply cấu hình IP mới

```
netplan apply
```

1.2. Cài đặt JDK 11

- Bước 1: Update và upgrade server trước khi cài đặt JDK

```
apt update
```

- Bước 2: Cài đặt openJDK 11

```
apt install openjdk-11-jdk -y
```

1.3. Cài đặt Development Tools

- Bước 1: Cài đặt Development

```
apt install build-essential -y
```

2. HƯỚNG DẪN CÀI ĐẶT DỊCH VỤ 3RD PARTY

2.1. Cài đặt Nginx

- Bước 1: Cài đặt web server Nginx

```
apt install nginx -y
```

- Bước 2: Enable mỗi khi khởi động và start service

```
systemctl start nginx && systemctl enable nginx
```

2.2. Cài đặt Keepalived

- Bước 1: Cài đặt service Keepalived

```
apt-get -y install keepalived
```

- Bước 2: Cấu hình cho phép gắn địa chỉ IP ảo lên card mạng và IP Forward

```
echo "net.ipv4.ip_nonlocal_bind = 1" >> /etc/sysctl.conf
echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf
sysctl -p
```

- Bước 3: Cấu hình Keepalived

+ **Node 1**

```
nano /etc/keepalived/keepalived.conf
```

```
# Define the script used to check if haproxy is still working
vrrp_script chk_nginx {
   script "/usr/bin/killall -0 nginx"
   interval 2
   weight 2
}

# Configuration for Virtual Interface
```

```
vrrp_instance LB_VIP {
    interface enp0s3
    state MASTER        # set to BACKUP on the peer machine
    priority 101        # set to  99 on the peer machine
    virtual_router_id 51

    authentication {
        auth_type AH
        auth_pass abCD@1234     # Password for accessing vrrpd. Same on all devices
    }
    unicast_src_ip 192.168.5.61 # Private IP address of master
    unicast_peer {
        192.168.5.62            # Private IP address of the backup haproxy
    }
    advert_int 1
    # The virtual ip address shared between the two loadbalancers
    virtual_ipaddress {
        192.168.5.60 dev enp0s3
    }

    # Use the Defined Script to Check whether to initiate a fail over
    track_script {
        chk_nginx
    }
}
```

+ **Node 2**

```
nano /etc/keepalived/keepalived.conf
```

```
# Define the script used to check if haproxy is still working
vrrp_script chk_nginx {
   script "/usr/bin/killall -0 nginx"
   interval 2
   weight 2
}

# Configuration for Virtual Interface
vrrp_instance LB_VIP {
   interface enp0s3
   state BACKUP        # set to MASTER on the peer machine
   priority 99        # set to  101 on the peer machine
   virtual_router_id 51

   authentication {
      auth_type AH
      auth_pass abCD@1234     # Password for accessing vrrpd. Same on all devices
   }
   unicast_src_ip 192.168.5.62 # Private IP address of backup
   unicast_peer {
      192.168.5.61            # Private IP address of the master haproxy
   }
   advert_int 1
   # The virtual ip address shared between the two loadbalancers
```

```
    virtual_ipaddress {
        192.168.5.60 dev enp0s3
    }

    # Use the Defined Script to Check whether to initiate a fail over
    track_script {
        chk_nginx
    }
}
```

- Bước 3: Khởi động dịch vụ Keepalived lần lượt trên node 1 và 2

```
systemctl start keepalived && systemctl enable keepalived
```

- Bước 4: Check status service Keepalived

```
systemctl status keepalived.service
```

2.3. Cài đặt Percona XtraDB Cluster

- Bước 1: Cài đặt từ kho lưu trữ (Node 1, 2)

```
apt install -y wget gnupg2 lsb-release curl
wget https://repo.percona.com/apt/percona-release_latest.generic_all.deb
dpkg -i percona-release_latest.generic_all.deb
apt update
percona-release setup pxc80
apt install -y percona-xtradb-cluster
```

+ Sau đó nhập password cho account root của Database

- Bước 2: Cấu hình các node để nhân rộng thành cụm

+ Sau khi cài đặt Percona XtraDB Cluster trên mỗi node, ta phải cấu hình cụm. Trong dự án này, tôi sẽ trình bày cách cấu hình một cụm 2 node và 1 DB Management (**Galera Arbitrator)** trên node 3

Node	Host	IP
Node 1	cpu1	192.168.5.61
Node 2	cpu2	192.168.5.62
Node 3	GUI	192.168.5.63

+ **Node 1**

```
service mysql stop
nano /etc/mysql/my.cnf
```

```
[mysqld]
binlog_expire_logs_seconds=604800      # 7 ngày hết hạn logs
wsrep_provider=/usr/lib/galera4/libgalera_smm.so
wsrep_cluster_name=VNPT
wsrep_cluster_address=gcomm://
wsrep_node_name=VNPT-Cluster
wsrep_node_address=192.168.5.61
wsrep_sst_method=xtrabackup-v2
pxc_strict_mode=ENFORCING  # import file vào DB thì để tham số DISABLED
binlog_format=ROW
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
pxc-encrypt-cluster-traffic=OFF       # Tắt mã hóa SSL
```

+ **Node 2**

```
service mysql stop
nano /etc/mysql/my.cnf
```

```
[mysqld]

binlog_expire_logs_seconds=604800       # 7 ngày hết hạn logs

wsrep_provider=/usr/lib/galera4/libgalera_smm.so

wsrep_cluster_name=VNPT

wsrep_cluster_address=gcomm://192.168.5.61,192.168.5.62

wsrep_node_name=VNPT-Cluster

wsrep_node_address=192.168.5.62

wsrep_sst_method=xtrabackup-v2

pxc_strict_mode=ENFORCING  # import file vào DB thì để tham số DISABLED

binlog_format=ROW

default_storage_engine=InnoDB

innodb_autoinc_lock_mode=2

pxc-encrypt-cluster-traffic=OFF         # Tắt mã hóa SSL
```

- Bước 3: Khởi động node đầu tiên (Node 1)

```
systemctl start mysql@bootstrap.service
```

+ Tạo user để đồng bộ dữ liệu giữa các node

```
mysql -u root -p
```

```
show status like 'wsrep%';

CREATE USER 'emas'@'%' IDENTIFIED BY 'abCD@1234';

GRANT ALL PRIVILEGES ON *.* To 'emas'@'%';

GRANT GRANT OPTION ON *.* To 'emas'@'%';

FLUSH PRIVILEGES;

SELECT * from information_schema.user_privileges where grantee like "'emas'%";
```

+ Sửa lại file config

```
nano /etc/mysql/my.cnf
```

```
wsrep_cluster_address=gcomm://192.168.5.61,192.168.5.62
```

- Bước 4: Thêm node vào cụm (Node 2)

```
systemctl start mysql
mysql -u root -p
```

```
show status like "%wsrep%";
select user,host from mysql.user;
```

- Bước 5: Thiết lập DB Management - **Galera Arbitrator** (Node 3)

+ Cài đặt Galera Arbitrator từ kho lưu trữ

```
apt install -y wget gnupg2 lsb-release curl
wget https://repo.percona.com/apt/percona-release_latest.generic_all.deb
dpkg -i percona-release_latest.generic_all.deb
apt update
percona-release setup pxc80
apt install percona-xtradb-cluster-garbd
```

+ Cấu hình Galera Arbitrator

```
nano /etc/default/garb
```

```
# Copyright (C) 2012 Codership Oy
# This config file is to be sourced by garb service script.

# A comma-separated list of node addresses (address[:port]) in the cluster
GALERA_NODES="192.168.5.61:4567, 192.168.5.62:4567, 192.168.5.63:4567"

# Galera cluster name, should be the same as on the rest of the nodes.
GALERA_GROUP="VNPT"
```

```
# Optional Galera internal options string (e.g. SSL settings)
# see http://galeracluster.com/documentation-webpages/galeraparameters.html
# GALERA_OPTIONS=""

# Log file for garbd. Optional, by default logs to syslog
# Deprecated for CentOS7, use journalctl to query the log for garbd
LOG_FILE="/var/log/garb.log"
```

+ Tạo đường dẫn log, phân quyền và khởi động Galera Arbitrator

```
touch /var/log/garb.log
chmod a+rw /var/log/garb.log
systemctl start garbd
service garbd status
```

2.4. Cài đặt EFK Stack (Elasticsearch – Fluentd – Kibana)

2.4.1. Elasticsearch (trên node 1, 2, 3 tương tự)

- Bước 1: Lấy key và repo version Elasticsearch 7.x.x

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg
apt-get install apt-transport-https
echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list
```

- Bước 2: Cài đặt và cấu hình Elasticsearch

+ Cài đặt Elasticsearch

```
apt-get update && apt-get install elasticsearch
```

+ Cấu hình Elasticsearch

```
nano /etc/elasticsearch/elasticsearch.yml
```

```
cluster.name: VNPT
node.name: node-1
network.host: 192.168.5.61
http.port: 9200
discovery.seed_hosts: ["192.168.5.61", "192.168.5.62", "192.168.5.63"]
cluster.initial_master_nodes: ["node-1", "node-2", "node-3"]
```

```
nano /etc/elasticsearch/jvm.options
```

```
-Xms2g
-Xmx2g
```

+ Khởi động Elasticsearch

```
systemctl daemon-reload && systemctl enable elasticsearch.service
systemctl start elasticsearch.service
```

2.4.2. Kibana (trên node 1, 2, 3 tương tự)

- Bước 1: Cài đặt Kibana

```
apt-get install kibana
```

- Bước 2: Cấu hình Kibana

```
nano /etc/kibana/kibana.yml
```

```
server.port: 5601
server.host: "192.168.xxx.xxx"
server.publicBaseUrl: "http://192.168.5.61:5601"
elasticsearch.hosts: ["http://192.168.5.61:9200", "http://192.168.5.62:9200", "http://192.168.5.63:9200"]
```

- Bước 3: Khởi động Kibana

```
systemctl enable kibana.service &&  systemctl start kibana.service
```

2.4.3. Fluentd (trên node 1, 2, 3 tương tự)

- Bước 1: Cài đặt từ repo Fluentd

```
curl -fsSL https://toolbelt.treasuredata.com/sh/install-ubuntu-bionic-td-agent3.sh | sh
```

- Bước 2: Cài đặt Plugin Fluentd to Elasticsearch

```
td-agent-gem install fluent-plugin-elasticsearch
```

- Bước 3: Cấu hình Fluentd (đọc log nginx)

+ Tạo file cấu hình mới

```
mv /etc/td-agent/td-agent.conf /etc/td-agent/td-agent.conf.old
nano /etc/td-agent/td-agent.conf
```

```
<match **.*>
  @type copy
  <store>
    @type elasticsearch
    host 192.168.5.61
    port 9200
    index_name ${tag}
    type_name ${tag}
    enable_ilm true
    include_timestamp true
    flush_interval 5s
  </store>
  <store>
    @type stdout
  </store>
```

```
</match>


<source>
  @type forward
  port 24224
  bind 0.0.0.0
</source>


<source>
  @type tail
  path  /var/log/nginx/access.log
#  pos_file  /var/log/docker-nginx.pos
  tag GPU1.access-nginx
  <parse>
    @type nginx
    localtime true
    time_type string
    unmatched_lines
  </parse>
</source>
```

- Bước 4: Phân quyền và khởi động Fluentd

  chmod og+r -R /var/log/nginx/

  systemctl enable td-agent && systemctl restart td-agent

2.5. Cài đặt Redis

2.5.1. Redis-server

- Bước 1: Cài đặt từ kho lưu trữ

```
curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o /usr/share/keyrings/redis-archive-keyring.gpg

echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg] https://packages.redis.io/deb $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/redis.list

apt-get update && apt-get install redis -y

systemctl enable redis-server
```

- Bước 2: Cấu hình Redis-server **master** (Node 1)

```
nano /etc/redis/redis.conf
```

```
bind 0.0.0.0

masterauth 123      # Password tùy chọn

requirepass 123     # Nhập lại password
```

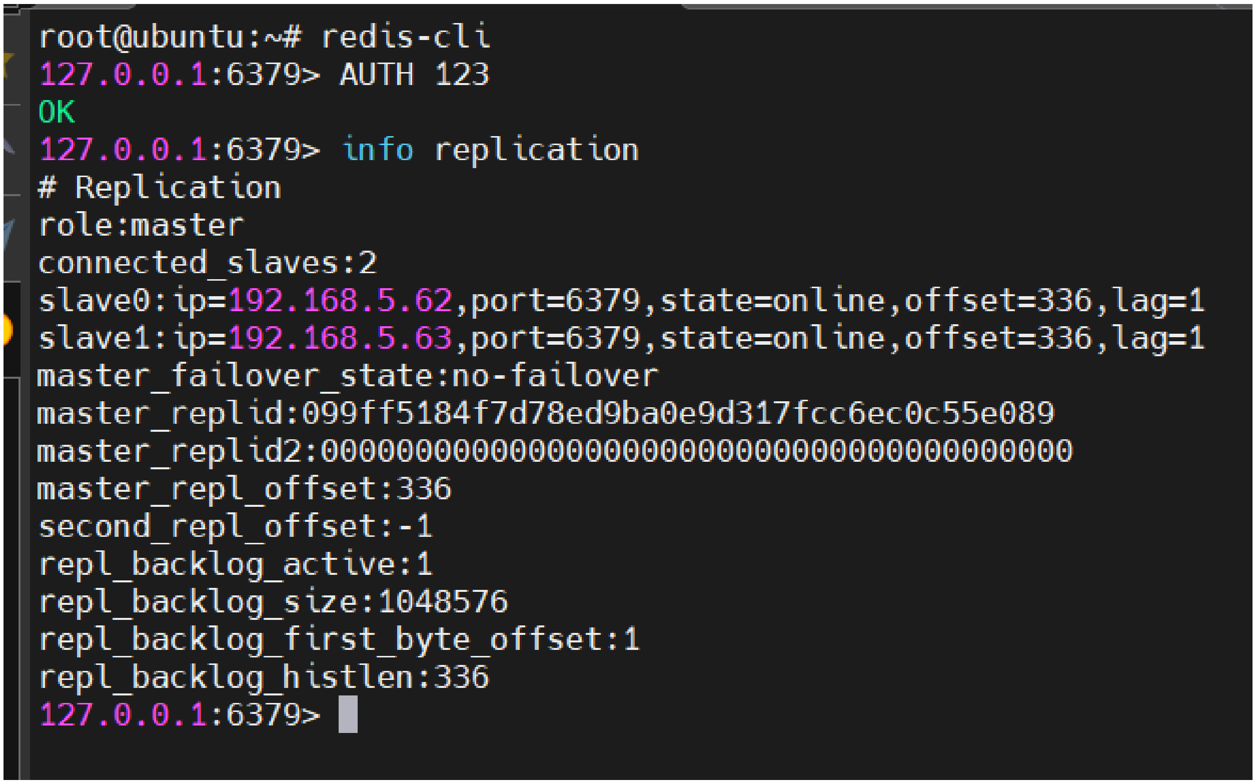
```
service redis-server restart
```

- Bước 3: Cấu hình Redis-server **slave** (Node 2, 3)

```
nano /etc/redis/redis.conf
```

```
bind 0.0.0.0

replicaof 192.168.5.61 6379         #IP node master

masterauth 123

requirepass 123
```

```
service redis-server restart
```



```
root@ubuntu:~# redis-cli
127.0.0.1:6379> AUTH 123
OK
127.0.0.1:6379> info replication
# Replication
role:master
connected_slaves:2
slave0:ip=192.168.5.62,port=6379,state=online,offset=336,lag=1
slave1:ip=192.168.5.63,port=6379,state=online,offset=336,lag=1
master_failover_state:no-failover
master_replid:099ff5184f7d78ed9ba0e9d317fcc6ec0c55e089
master_replid2:0000000000000000000000000000000000000000
master_repl_offset:336
second_repl_offset:-1
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:1
repl_backlog_histlen:336
127.0.0.1:6379>
```

2.5.2. Redis-sentinel

- Bước 1: Cấu hình Redis-sentinel

nano /etc/redis/sentinel.conf

```
daemonize yes

port 26379

bind 0.0.0.0

supervised systemd

pidfile "/run/redis/redis-sentinel.pid"

logfile "/var/log/redis/sentinel.log"

sentinel monitor redis-cluster 192.168.5.61 6379 2

sentinel auth-pass redis-cluster 123

sentinel down-after-milliseconds redis-cluster 3000

sentinel failover-timeout redis-cluster 10000

sentinel parallel-syncs redis-cluster 1
```

- Bước 2: Phân quyền và tạo file systemd cho service

chown redis:redis /etc/redis/sentinel.conf

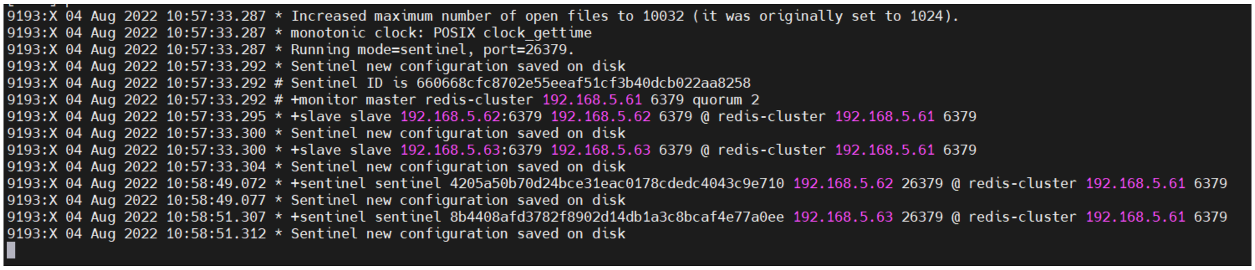
nano /etc/systemd/system/redis-sentinel.service

```
[Unit]
Description=Redis Sentinel
After=network.target

[Service]
User=redis
Group=redis
Type=notify
ExecStart=/usr/bin/redis-server /etc/redis/sentinel.conf --sentinel
ExecStop=/usr/bin/redis-cli shutdown
Restart=always

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

- Bước 3: Load lại daemon và khởi động service

systemctl daemon-reload && systemctl enable redis-sentinel

service redis-sentinel start



```
9193:X 04 Aug 2022 10:57:33.287 * Increased maximum number of open files to 10032 (it was originally set to 1024).
9193:X 04 Aug 2022 10:57:33.287 * monotonic clock: POSIX clock_gettime
9193:X 04 Aug 2022 10:57:33.287 * Running mode=sentinel, port=26379.
9193:X 04 Aug 2022 10:57:33.292 * Sentinel new configuration saved on disk
9193:X 04 Aug 2022 10:57:33.292 # Sentinel ID is 660668cfc8702e55eeaf51cf3b40dcb022aa8258
9193:X 04 Aug 2022 10:57:33.292 # +monitor master redis-cluster 192.168.5.61 6379 quorum 2
9193:X 04 Aug 2022 10:57:33.295 * +slave slave 192.168.5.62:6379 192.168.5.62 6379 @ redis-cluster 192.168.5.61 6379
9193:X 04 Aug 2022 10:57:33.300 * Sentinel new configuration saved on disk
9193:X 04 Aug 2022 10:57:33.300 * +slave slave 192.168.5.63:6379 192.168.5.63 6379 @ redis-cluster 192.168.5.61 6379
9193:X 04 Aug 2022 10:57:33.304 * Sentinel new configuration saved on disk
9193:X 04 Aug 2022 10:58:49.072 * +sentinel sentinel 4205a50b70d24bce31eac0178cdedc4043c9e710 192.168.5.62 26379 @ redis-cluster 192.168.5.61 6379
9193:X 04 Aug 2022 10:58:49.077 * Sentinel new configuration saved on disk
9193:X 04 Aug 2022 10:58:51.307 * +sentinel sentinel 8b4408afd3782f8902d14db1a3c8bcaf4e77a0ee 192.168.5.63 26379 @ redis-cluster 192.168.5.61 6379
9193:X 04 Aug 2022 10:58:51.312 * Sentinel new configuration saved on disk
```

2.6. Cài đặt RabbitMQ

- Bước 1: Add IP của toàn bộ node trong cụm vào mỗi node để đảm bảo các node có thể nhìn thấy nhau

```
nano /etc/hosts
```

```
192.168.5.61        cluster
192.168.5.62        node2
192.168.5.63        node3
```

- Bước 2: Cài đặt Erlang. RabbitMQ yêu cầu Erlang để hoạt động

```
apt update && sudo apt install curl software-properties-common apt-transport-https lsb-release -y

curl -fsSL https://packages.erlang-solutions.com/ubuntu/erlang_solutions.asc | gpg --dearmor -o /etc/apt/trusted.gpg.d/erlang.gpg

echo "deb https://packages.erlang-solutions.com/ubuntu $(lsb_release -cs) contrib" | sudo tee /etc/apt/sources.list.d/erlang.list

apt update && sudo apt install erlang -y
```

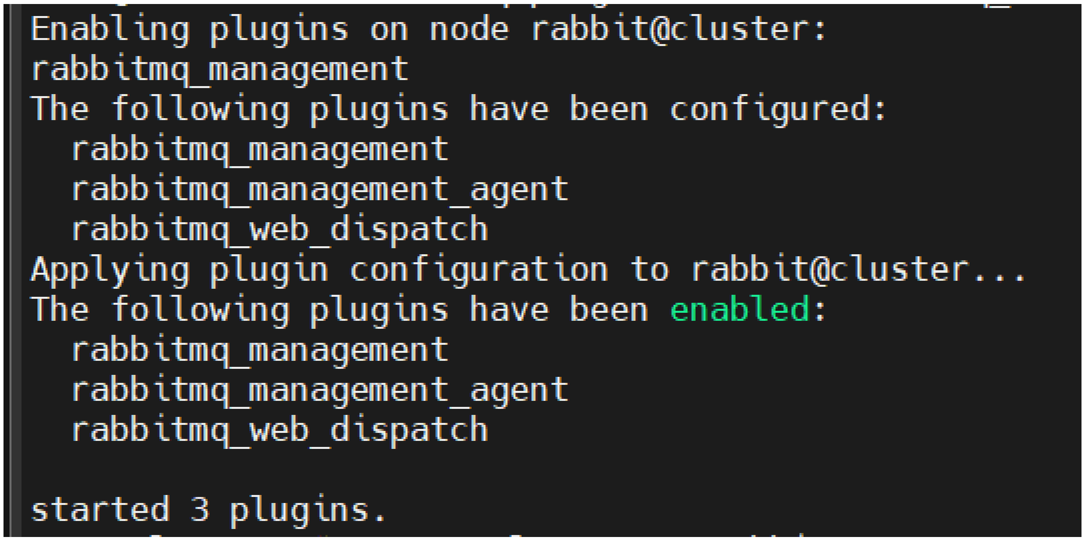
- Bước 3: Cài đặt Cài đặt RabbitMQ

```
curl -s https://packagecloud.io/install/repositories/rabbitmq/rabbitmq-server/script.deb.sh | sudo bash

apt update && apt install rabbitmq-server -y

systemctl status rabbitmq-server.service

systemctl enable rabbitmq-server.service
```

- Bước 4: Bật Plugin giao diện quản lý của RabbitMQ

```
rabbitmq-plugins enable rabbitmq_management

systemctl restart rabbitmq-server       # Restart lại dịch vụ xem có bị lỗi không
```

sudo ss -tunelp | grep 15672 # Dịch vụ Web sẽ được lắng nghe trên port TCP 15672, nếu tường lửa đang hoạt động, hãy mở cả hai port 5672 và 15672



```
Enabling plugins on node rabbit@cluster:
rabbitmq_management
The following plugins have been configured:
  rabbitmq_management
  rabbitmq_management_agent
  rabbitmq_web_dispatch
Applying plugin configuration to rabbit@cluster...
The following plugins have been enabled:
  rabbitmq_management
  rabbitmq_management_agent
  rabbitmq_web_dispatch

started 3 plugins.
```

- Bước 5: Copy cookie từ cluster vào các nodes còn lại

  scp /var/lib/rabbitmq/.erlang.cookie root@node2:/var/lib/rabbitmq/

  scp /var/lib/rabbitmq/.erlang.cookie root@node3:/var/lib/rabbitmq/

- Bước 6: Cấu hình RabbitMQ tham gia vào cụm (Node 2, 3)

  systemctl restart rabbitmq-server

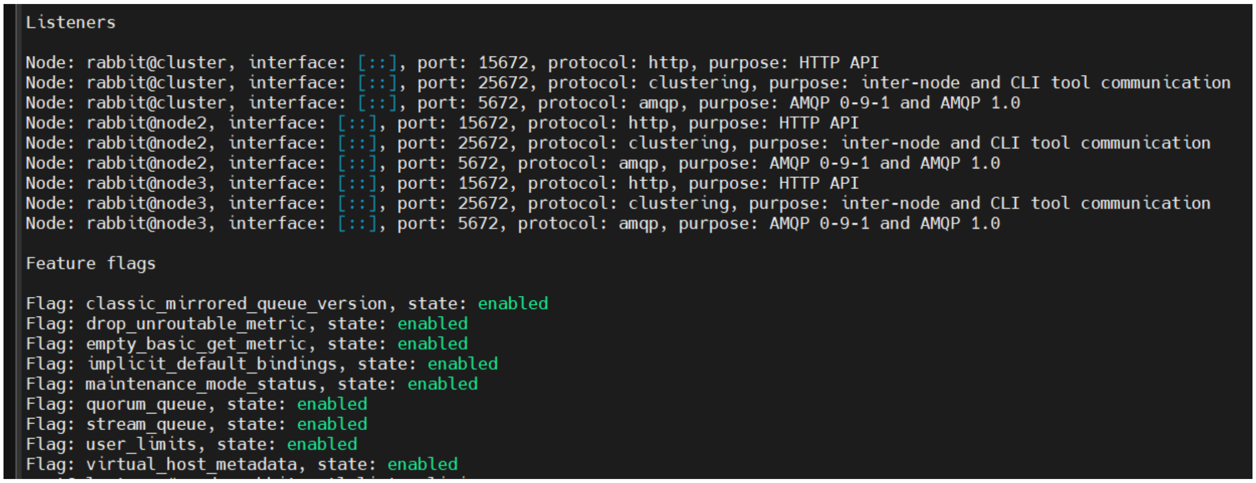
  sudo rabbitmqctl stop_app

  sudo rabbitmqctl reset

  sudo rabbitmqctl join_cluster rabbit@cluster

  sudo rabbitmqctl start_app

  sudo rabbitmqctl cluster_status



```
Listeners

Node: rabbit@cluster, interface: [::], port: 15672, protocol: http, purpose: HTTP API
Node: rabbit@cluster, interface: [::], port: 25672, protocol: clustering, purpose: inter-node and CLI tool communication
Node: rabbit@cluster, interface: [::], port: 5672, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0
Node: rabbit@node2, interface: [::], port: 15672, protocol: http, purpose: HTTP API
Node: rabbit@node2, interface: [::], port: 25672, protocol: clustering, purpose: inter-node and CLI tool communication
Node: rabbit@node2, interface: [::], port: 5672, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0
Node: rabbit@node3, interface: [::], port: 15672, protocol: http, purpose: HTTP API
Node: rabbit@node3, interface: [::], port: 25672, protocol: clustering, purpose: inter-node and CLI tool communication
Node: rabbit@node3, interface: [::], port: 5672, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0

Feature flags

Flag: classic_mirrored_queue_version, state: enabled
Flag: drop_unroutable_metric, state: enabled
Flag: empty_basic_get_metric, state: enabled
Flag: implicit_default_bindings, state: enabled
Flag: maintenance_mode_status, state: enabled
Flag: quorum_queue, state: enabled
Flag: stream_queue, state: enabled
Flag: user_limits, state: enabled
Flag: virtual_host_metadata, state: enabled
```

- Bước 7: Xóa vhost và user default, tạo vhost /emas và user quản trị viên truy cập từ xa quản lý cụm RabbitMQ

  rabbitmqctl delete_vhost /

  rabbitmqctl delete_user guest

  rabbitmqctl add_vhost /emas

  rabbitmqctl set_policy --vhost /emas ha-all "." '{"ha-mode":"all"}'*

  rabbitmqctl list_policies --vhost /emas

  rabbitmqctl add_user emas 123

  rabbitmqctl set_user_tags emas administrator

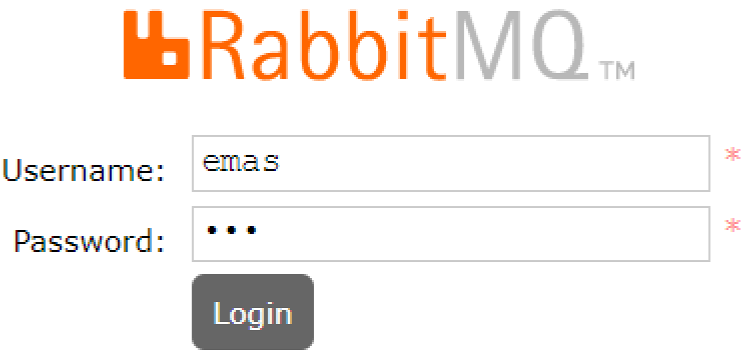
  rabbitmqctl set_permissions -p /emas emas "." ".*" ".*"*

- Bước 8: Truy cập web management để quản trị tất cả các nodes trong cụm RabbitMQ

  192.168.5.61:15672

  #User: emas

  #Password: 123



Username: emas *

Password: ••• *

2.7. Cài đặt Minio

- Bước 1: Thêm tên server vào file hosts và tải minio từ **https://dl.min.io/**

```
wget https://dl.min.io/server/minio/release/linux-amd64/archive/minio.RELEASE.2022-04-26T01-20-24Z
mv minio.RELEASE.2022-04-26T01-20-24Z minio
```

- Bước 2: Set quyền execute và đưa file vào thư mục bin. Tạo user và gán quyền cho minio

```
chmod +x minio
cp minio /usr/local/bin
useradd -r minio-user -s /sbin/nologin
chown minio-user:minio-user /usr/local/bin/minio
```

- Bước 3: Tạo và gán quyền cho các đường dẫn cấu hình

```
mkdir /usr/local/share/minio
chown minio-user:minio-user /usr/local/share/minio
mkdir /etc/minio
chown minio-user:minio-user /etc/minio
```

- Bước 4: LVM

```
pvcreate /dev/sdb
vgcreate minio /dev/sdb
```

```
lvcreate -n data_minio -l 100%FREE minio

mkfs.ext4 /dev/minio/data_minio

mkdir /data && mount /dev/minio/data_minio /data/

blkid /dev/minio/data_minio

/dev/minio/data_minio: UUID="..." TYPE="ext4"

nano /etc/fstab

UUID=... /data ext4 defaults 0 0

mount -a

mount | grep data

/dev/mapper/minio-data_minio on /data type ext4 (rw,relatime)
```

- Bước 5: Tạo folder và gán quyền chứa dữ liệu minio trong dự án emas là 2 vùng

```
mkdir -p /data/minio1

mkdir -p /data/minio2

chown -R minio-user:minio-user /data/
```

- Bước 6: Tạo file /etc/default/minio

```
nano /etc/default/minio

MINIO_ACCESS_KEY="emas"

MINIO_VOLUMES="http://192.168.5.61/data/minio1
http://192.168.5.61/data/minio2 http://192.168.5.62/data/minio1
http://192.168.5.62/data/minio2 http://192.168.5.63/data/minio1
http://192.168.5.63/data/minio2 http://192.168.5.64/data/minio1
http://192.168.5.64/data/minio2 http://192.168.5.65/data/minio1
http://192.168.5.65/data/minio2"

MINIO_OPTS="-C /etc/minio --address :9000 --console-address :9001"

MINIO_SECRET_KEY="abCD@1234"
```

- Bước 7: Tạo file /etc/systemd/system/minio.service

  nano /etc/systemd/system/minio.service

```
Description=MinIO
Documentation=https://docs.min.io
Wants=network-online.target
After=network-online.target
AssertFileIsExecutable=/usr/local/bin/minio

[Service]
WorkingDirectory=/usr/local/

User=minio-user
Group=minio-user
ProtectProc=invisible

EnvironmentFile=/etc/default/minio
ExecStartPre=/bin/bash -c "if [ -z \"${MINIO_VOLUMES}\" ]; then echo \"Variable MINIO_VOLUMES not set in /etc/default/minio\"; exit 1; fi"
ExecStart=/usr/local/bin/minio server $MINIO_OPTS $MINIO_VOLUMES

# Let systemd restart this service always
Restart=always

# Specifies the maximum file descriptor number that can be opened by this process
LimitNOFILE=1048576
```

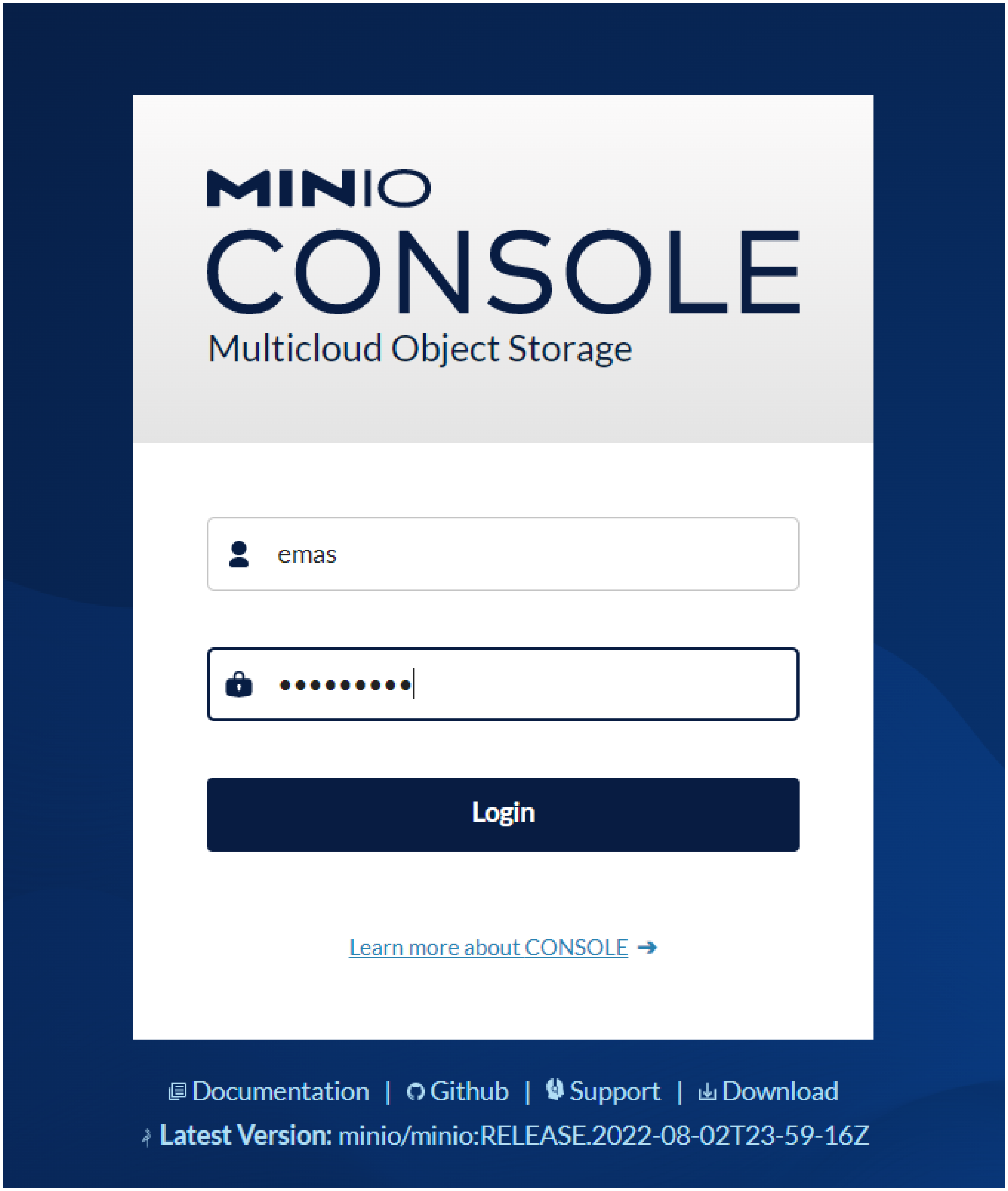
```
# Specifies the maximum number of threads this process can create
TasksMax=infinity

# Disable timeout logic and wait until process is stopped
TimeoutStopSec=infinity
SendSIGKILL=no

[Install]
WantedBy=multi-user.target

# Built for ${project.name}-${project.version} (${project.name})
```

- Bước 8: Reload lại deamon và khởi động service minio

  ```
  systemctl daemon-reload
  systemctl enable minio
  systemctl start minio
  systemctl status minio
  ```

- Bước 9: Truy cập trang quản trị Minio trên web browser

  ```
  192.168.5.61:9200
  #User: emas
  #Password: abCD@1234
  ```


MINIO
CONSOLE
Multicloud Object Storage
Login
Learn more about CONSOLE ➔
Documentation | Github | Support | Download

emas

•••••••••

Latest Version: minio/minio:RELEASE.2022-08-02T23-59-16Z