

# MavaraTech - ESB - Install Infra

- [Install Kubernetes](#)
- [Postgres Using Swarm](#)
  - [Install Docker Swarm](#)
  - [Install Postgres](#)
- [Install NFS Server](#)
- [Install Rabbitmq and Volumes](#)
- [Install Integropia](#)
- [Install SSO](#)

# Install Kubernetes

مین‌کیم می‌ظن تار hostname دون ره یارب

```
hostnamectl set-hostname {HOSTNAME}
```

دوش داچای ریز سردا آتسا زاین دون ره یارب

```
mkdir -p /home/data
```

مین‌کیم ب‌صن ار rpm یاه چی‌کپ ریز تاروت‌سد زا هدافتسا اب دون ره یارب

دنراد رارق ~/ یروت‌کریاد رد ب‌صن یاه لیاف همه دوش یم صرف

```
tar -xf ./k8s_tools.tar.gz
cd ./k8s-offline
yum install -y --cacheonly --disablerepo=* ./rpm/*.rpm
```

مین‌کیم داچای ار زی‌تن‌ربوک یارب registry رن‌یت‌ناک، زی‌تن‌ربوک یاه دون زا یکی رد

registry کی و زی‌تن‌ربوک یاه دون زا یکی رد زی‌تن‌ربوک یاه چی‌می‌یارب registry کی تسارتهب)  
داچای دراد یی‌الاب storage رادقم هک سی‌باید یاه‌رورس زا یکی یور رب اه چی‌می‌ریاس یارب یلصا  
(دوش

```
cd ./images
docker load -i registry.docker
docker run -d -p 5000:5000 --restart=always --name registry -v rke-registry:/var/lib/registry
registry:latest

#test:
curl -X GET http://{IP}:{PORT}/v2/_catalog
```

registry تاصخشم مود PORT و IP یاجب و زی‌تن‌ربوک صوصخم registry تاصخشم لوا PORT و IP یاجب  
مین‌کیم صخشم ار یلصا

(دوش‌یم ارجا اه دون همه یور تاروت‌سد نی‌ا)

```
sudo systemctl enable docker
sudo systemctl start docker
```

```
vi /etc/docker/daemon.json
{
"insecure-registries":["{IP}:{PORT}","{IP}:{PORT}"],
"group": "docker"
}
systemctl restart docker
```

مین‌کیم ارجا اه دون همه یارب ار ریز تاروت‌سد

```
adduser mydocker
passwd mydocker
```

```
sudo groupadd docker
sudo usermod -aG docker mydocker
newgrp docker

export UUUU=mydocker
sudo chmod go-w /home/$UUUU
```

مین‌کیم ارجا ار ریز روت‌سد اه دون همه یازا هب یلصا دون زا

```
ssh-keygen
cat ~/.ssh/id_rsa.pub | ssh mydocker@{IP} "mkdir -p /home/mydocker/.ssh && cat >>
/home/mydocker/.ssh/authorized_keys && chmod 600 /home/mydocker/.ssh/authorized_keys"
```

مین‌کیم ارجا اه دون همه یارب ار ریز تاروت‌سد

```
export UUUU=mydocker
sudo chmod 700 /home/$UUUU/.ssh
sudo chmod 644 /home/$UUUU/.ssh/authorized_keys
sudo chown $UUUU:$UUUU /home/$UUUU/.ssh/authorized_keys
sudo chown $UUUU:$UUUU /home/$UUUU/.ssh
sudo usermod -d /home/$UUUU $UUUU
sudo systemctl restart sshd
```

مین‌کیم ارجا اه دون همه یارب ار ریز تاروت‌سد

```
nano /etc/ssh/sshd_config

PubkeyAuthentication yes
AllowAgentForwarding yes
```

```
AllowTcpForwarding yes
```

```
sudo cat /etc/ssh/sshd_config | grep -E
```

```
"PubkeyAuthentication| AllowAgentForwarding| AllowTcpForwarding"
```

```
sudo systemctl restart sshd
```

مینکیم ارجا اه دون همه یارب زاین دروم یاه تروپ ندرک زاب یارب اریز تاروتسد

```
sudo firewall-cmd --permanent --add-port=5000/tcp
sudo firewall-cmd --permanent --add-port=80/tcp
sudo firewall-cmd --permanent --add-port=443/tcp
sudo firewall-cmd --permanent --add-port=2376/tcp
sudo firewall-cmd --permanent --add-port=6443/tcp
sudo firewall-cmd --permanent --add-port=8472/udp
sudo firewall-cmd --permanent --add-port=9099/tcp
sudo firewall-cmd --permanent --add-port=10250/tcp
sudo firewall-cmd --permanent --add-port=10254/tcp
sudo firewall-cmd --permanent --add-port=30000-32767/tcp
sudo firewall-cmd --permanent --add-port=30000-32767/udp
sudo firewall-cmd --permanent --add-port=2379/tcp
sudo firewall-cmd --permanent --add-port=2380/tcp
sudo firewall-cmd --permanent --add-port=7946/tcp
sudo firewall-cmd --permanent --add-port=7946/udp
sudo firewall-cmd --permanent --add-port=179/tcp
sudo firewall-cmd --permanent --add-port=4789/udp
sudo firewall-cmd --permanent --add-port=5473/tcp
sudo firewall-cmd --permanent --direct --add-rule ipv4 filter FORWARD 99 -o cali+ -j ACCEPT
sudo firewall-cmd --permanent --direct --add-rule ipv4 filter FORWARD 99 -i cali+ -j ACCEPT
sudo firewall-cmd --direct --add-rule ipv4 filter FORWARD 99 -o cali+ -j ACCEPT
sudo firewall-cmd --direct --add-rule ipv4 filter FORWARD 99 -i cali+ -j ACCEPT
sudo firewall-cmd --set-log-denied=all
sudo firewall-cmd --reload
```

تسا لوا دون یارب اهنت تاروتسد یق بام

دوش شوپ لوصخم یرتسیجر یور زی تنربوک یاه جیمی ات دوش یم ارجا ریز روتسد

(دوش یراذگیاج زی تنربوک registry تروپ و یپ یآ)

```
unzip ~/k8s_images.zip -d ~/
```

```
./rancher/rancher-load-images.sh --image-list ./ALL-IMG.txt --images ~/k8s_images.tar.gz --
```

```
registry {IP}:{PORT} 2>&1 | tee ./load.log
```

دوچوم لیاف حیحص نژرو رادقم {VERSION} یاج هب) دوش یم ماجنا helm ب صن یارب ریز روتسد  
(دوش یراذگیاج

```
cd ./helm
tar -zxvf helm-{VERSION}.tar.gz
ll /usr/local/bin | grep helm
sudo cp linux-amd64/helm /usr/local/bin/helm
ll /usr/local/bin | grep helm
sudo chmod +x /usr/local/bin/helm
```

دوش یم ماجنا kubectl ب صن یارب ریز روتسد

```
cd ./kubectl
echo "$(kubectl.sha256) kubectl" | sha256sum --check
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

مینکیم هدامآ ب صن یارب ار rke لیاف ریز تاروتسد زا هدافتسا اب

```
cd ./rke
sudo chmod +x ./rke
./rke --version
```

میوش یم rke گیفناک طیحم دراو ریز روتسد اب

```
./rke config --name cluster.yml
```

میهدیم خساپ ریز تروصب دون ره یارب

```
[+] Cluster Level SSH Private Key Path [ ~/.ssh/id_rsa]: #مینزیم رتنی ا) ضرفشپ خساپ
[+] Number of Hosts [1]: #اه دون دادعت
[+] SSH Address of host (1) [none]: #لو ا دون یپ یآ
[+] SSH Port of host (1) [22]: #مینزیم رتنی ا) ضرفشپ خساپ
[+] SSH Private Key Path of host (10.10.10.12) [none]: #مینزیم رتنی ا) ضرفشپ خساپ
[-] You have entered empty SSH key path, trying fetch from SSH key parameter
[+] SSH Private Key of host (10.10.10.12) [none]: #مینزیم رتنی ا) ضرفشپ خساپ
[-] You have entered empty SSH key, defaulting to cluster level SSH key: ~/.ssh/id_rsa
[+] SSH User of host (10.10.10.12) [ubuntu]: mydocker
[+] Is host (10.10.10.12) a Control Plane host (y/n)? [y]: y #رتسم رطن دروم دون هکی تروصرد
y تس ا
```

```
[+] Is host (10.10.10.12) a Worker host (y/n)? [n]: y # یس ا رتسم رطن دروم دون مکی تروصرد
[+] Is host (10.10.10.12) an etcd host (y/n)? [n]: y # یس ا رکرو رطن دروم دون مکی تروصرد
[+] Override Hostname of host (10.10.10.12) [none]: node1 # دون مین تس اه
[+] Internal IP of host (10.10.10.12) [none]: # (مینزیم رتنی ا) صرفشیپ خس اپ
[+] Docker socket path on host (10.10.10.12) [/var/run/docker.sock]: # (رتنی ا) صرفشیپ خس اپ
(مینزیم)
```

می هدیم خس اپ ا ریز تال اوس باوج تیاهن رد

```
[+] Network Plugin Type (flannel, calico, weave, canal, aci) [canal]: canal # مکبش عون
[+] Authentication Strategy [x509]: # (مینزیم رتنی ا) صرفشیپ خس اپ
[+] Authorization Mode (rbac, none) [rbac]: # (مینزیم رتنی ا) صرفشیپ خس اپ
[+] Kubernetes Docker image [rancher/hyperkube: v1.26.8-rancher1]: # (مینزیم رتنی ا) صرفشیپ خس اپ
[+] Cluster domain [cluster.local]: # (مینزیم رتنی ا) صرفشیپ خس اپ
[+] Service Cluster IP Range [10.43.0.0/16]: # (مینزیم رتنی ا) صرفشیپ خس اپ
[+] Enable PodSecurityPolicy [n]: # (مینزیم رتنی ا) صرفشیپ خس اپ
[+] Cluster Network CIDR [10.42.0.0/16]: # (مینزیم رتنی ا) صرفشیپ خس اپ
[+] Cluster DNS Service IP [10.43.0.10]: # (مینزیم رتنی ا) صرفشیپ خس اپ
[+] Add addon manifest URLs or YAML files [no]: # (مینزیم رتنی ا) صرفشیپ خس اپ
```

می هدیم ری یغت ا ریز دراوم و مینکیم زاب رگشیاری و اب ا ر هدش داجی ا cluster.yml لیاف

```
kubelet:
  extra_binds:
    - "/home/data: /home/data"

-----

ssh_agent_auth: true

-----

addons: | -
  ---
  apiVersion: v1
  kind: Service
  metadata:
    name: custom-ingress-nginx-controller
    namespace: ingress-nginx
```

```
spec:
  ports:
    - name: http
      port: 80
      protocol: TCP
      targetPort: http
    - name: https
      port: 443
      protocol: TCP
      targetPort: https
  selector:
    app: ingress-nginx
    app.kubernetes.io/instance: ingress-nginx
  sessionAffinity: None
  type: LoadBalancer
```

مینکیم میظنت system\_images تمسق رد اه جیمی همه یارب ار registry تروپ و یپ یآ نیچمه  
لثم روطب (مینکیم یراذگیاج ار {IP}:{PORT} اه image زا لبق)

```
system_images:
  etcd: rancher/mirrored-coreos-etcd:v3.5.6 ----- هب دوش لیدبت -----> etcd:
  {IP}:{PORT}/rancher/mirrored-coreos-etcd:v3.5.6
```

ماچنا اه دون همه هب ssk-key اب لاصتا ندرک کچ یارب 5 طخ روتسد) مینکیم ارجا ار ریز روتسد  
(دوشیم)

```
eval "$(ssh-agent -k)" && eval $(ssh-agent) && ssh-add ~/.ssh/id_rsa
ssh-add -l

#test:
ssh -t mydocker@{IP} "docker ps"
```

دوش یم غورش بصن ریز روتسد زا هدافتسا اب

```
./rke up
```

رد kubectl هدافتسا یارب ار هدش بصن زیتنربوک گیفناک لیاف ریز تاروتسد زا هدافتسا اب  
مینکیم یپک هطوبرم سردآ

```
mkdir -p /root/.kube/
cp ./rke/kube_config_cluster.yml /root/.kube/config
```

```
kubectl get nodes
```

:مینی‌کیم ل‌صاح نان‌ی‌م‌طا زی‌ت‌ن‌ر‌ب‌وک ب‌ص‌ن ت‌ح‌ص زا ری‌ز ر‌وت‌س‌د زا ه‌د‌اف‌ت‌س‌ا اب

```
kubectl get nodes -o wide
```



# Postgres Using Swarm

# Install Docker Swarm

میں کی تم تس اوہ دون ہمہ یارب ار لاوریاف تامیظنت

```
firewall-cmd --add-port=2376/tcp --permanent
firewall-cmd --add-port=2377/tcp --permanent
firewall-cmd --add-port=7946/tcp --permanent
firewall-cmd --add-port=7946/udp --permanent
firewall-cmd --add-port=4789/udp --permanent
firewall-cmd --reload
```

میں کی تم تس بسانم HostName اوہ دون ہمہ یارب

```
hostnamectl set-hostname {NEW_NAME}
```

میں زیم ار ریز روتسد رتسم دون رد و میریگی م رظن رد رتسم ار دون کی

```
docker swarm init
```

قباطم و دنک دیلوت نکوت یاراد دنماک کی ات میںک یم ارجا یلعف Manager دون رد ار ریز روتسد  
میرور یم شپ دہدی م دنماک ہجیتن ہک یلحارم

```
docker swarm join-token manager
```

میںک یم میظنت ،اوہ نآ HOSTNAME ساسارب ار Swarm رتسالک رد اوہدون یاہ بسچرب ہلحرم نیار  
دون رہ بسچرب ساسارب ،حیحص دون رد اوہ سیورس رارقتسا یارب دنک یم کمک یراذگ بسچرب  
نیعیعت رتسالک وضع یاہدون یارب ہلحرم نیار رد ہک ییہ بسچرب عقاو رد ؛میںک یریگ میمصت  
رارق ہدافتسا دروم اوہدون یور (Postgres لاثم یارب) اوہ WORKLOAD رارقتسا و یدنب نامز رد ،دوش یم  
دریگ یم

```
docker node update --label-add region={LABEL} {HOSTNAME}
```

دوش یم ہدافتسا ریز روتسد زا اوہ label ہارمہ ہب اوہدون تسیل ہدہاشم تہج

```
docker node ls -q | xargs docker node inspect -f '{{ .ID }} [{{ .Description.Hostname }}]: {{  
.Spec.Labels }} - {{ .Status.Addr }}'
```

# Install Postgres

میں کی م داجی سی باتی د یاه دون ہمہ رد ار ریز یروت کریاد

```
mkdir -p /home/postgres/data
```

دع 2 زا شی ب اه رورس دادعت هک ی تروصرد): ریز تروص ب هدرک داجی docker-compose.yaml لی اف کی  
(دی هری غت ار لی اف تسا

```
version: "3.6"
services:
  zk1:
    image: {REGISTRY_IP}:{REGISTRY_PORT}/bitnami/zookeeper: 3.9.1
    deploy:
      placement:
        constraints:
          - "node.labels.region==db-1"
    ports:
      - target: 2181
        published: 2181
        mode: host
      - target: 2888
        published: 2888
        mode: host
      - target: 3888
        published: 3888
        mode: host
    networks:
      db_net: null
    environment:
      - ALLOW_ANONYMOUS_LOGIN=yes
      - ZOO_SERVER_ID=1
      - ZOO_SERVERS=0.0.0.0:2888:3888,{SECOND_SERVER_IP}:2888:3888
  zk2:
    image: {REGISTRY_IP}:{REGISTRY_PORT}/bitnami/zookeeper: 3.9.1
    deploy:
```

```
placement:
  constraints:
    - "node.labels.region==db-2"
```

ports:

```
- target: 2181
  published: 2181
  mode: host
- target: 2888
  published: 2888
  mode: host
- target: 3888
  published: 3888
  mode: host
```

networks:

```
db_net: null
```

environment:

```
- ALLOW_ANONYMOUS_LOGIN=yes
- ZOO_SERVER_ID=2
- ZOO_SERVERS={FIRST_SERVER_IP}: 2888: 3888, 0. 0. 0. 0: 2888: 3888
```

pgnode1:

```
image: {REGISTRY_IP}:{REGISTRY_PORT}/ghcr.io/zalando/spilo-15: 3. 0- p1
```

ports:

```
- target: 5432
  published: 5432
  mode: host
```

deploy:

```
replicas: 1
```

placement:

```
constraints:
  - "node.labels.region==db-1"
```

environment:

```
ZOOKEEPER_HOSTS: {FIRST_SERVER_IP}: 2181, {SECOND_SERVER_IP}: 2181
PGPASSWORD_STANDBY: {PASSWORD}
PGPASSWORD_ADMIN: {PASSWORD}
PGPASSWORD_SUPERUSER: {PASSWORD}
SCOPE: pgCluster
```

networks:

```
db_net: null
```

volumes:

```
- /home/postgres/data: /home/postgres/pgdata
```

pgnode2:

```
image: {REGISTRY_IP}:{REGISTRY_PORT}/ghcr.io/zalando/spilo-15:3.0-p1
```

ports:

```
- target: 5432
  published: 5432
  mode: host
```

deploy:

```
replicas: 1
placement:
  constraints:
    - "node.labels.region==db-2"
```

environment:

```
ZOOKEEPER_HOSTS: {FIRST_SERVER_IP}: 2181,{SECOND_SERVER_IP}: 2181
PGPASSWORD_STANDBY: {PASSWORD}
PGPASSWORD_ADMIN: {PASSWORD}
PGPASSWORD_SUPERUSER: {PASSWORD}
SCOPE: pgCluster
```

networks:

```
db_net: null
```

volumes:

```
- /home/postgres/data: /home/postgres/pgdata
```

networks:

db\_net:

```
name: mavara-pgsql-network
external: true
driver: overlay
```

و {REGISTRY\_IP} ، {REGISTRY\_PORT} ، {FIRST\_SERVER\_IP} ، {SECOND\_SERVER\_IP} ری‌داقم  
دینک می‌ظنت حیحص ری‌داقم ساسا رب ار {PASSWORD}

مینکیم ارجا ار ریز تاروتسد Manager Swarm دون یور

```
unzip postgres-cluster.zip
cd postgres-cluster
chmod +x install-postgres-cluster.sh
./install-postgres-cluster.sh {REGISTRY_IP}:{REGISTRY_PORT}
```

دوجوم docker-compose.yaml لی‌اف هک یروت‌کری‌اد رد ، ال‌اب روتسد ندوب زیم‌آ تی‌قفوم تروص رد

مین‌کی‌م ارجا ار ریز تاروت‌سد ت‌سا

```
docker network create -d overlay --attachable mavara-pgsql-network
docker stack deploy --compose-file docker-compose.yml the-pgsql-stack
```

دزاد‌رپ یم PostgreSQL رت‌سال‌ک تی‌عضو ی‌سر‌رب ه‌ب ریز ی‌اه دن‌ماک

```
docker service ls
docker stack services the-pgsql-stack
```

سی‌ورس مان رخ‌آ رد ار اه instance ه‌رامش) دش‌اب‌یم اه سی‌ورس گال ی‌سر‌رب ی‌ارب ریز تاروت‌سد  
(دی‌ه‌د ری‌ی‌غت

```
docker service logs the-pgsql-stack_zk1
docker service logs the-pgsql-stack_pgnode1
```

دوش یم ارجا Manager دون کی قی‌رط زا `docker stack rm` دن‌ماک، رت‌سال‌ک ندروآ نی‌ی‌اپ ته‌ج

```
docker stack rm the-pgsql-stack
```

# Install NFS Server

File ناو نعب اهرورس زا یکی تسا زاین ،Kubernetes یاهرورس نیب یکارتشا Storage هدفاتسا یارب ددرگ نییعت Server

نآ یور ار NFS Server و میریگی م رظن رد روطنم نیا یارب ار سیبایتی د یاهرورس زا یکی الومعم مینکی م ب ص ن

دشابیم دوجوم k8s-offline چی کپ رد NFS Server هب طوبرم RPM یاه چی کپ

```
tar -xf ./k8s_tools.tar.gz
cd ./k8s-offline
yum install -y --cacheonly --disablerepo=* ./rpm/*.rpm
```

مینکی م هدفاتسا ریز تاروتسد زا NFS Server ندرک لاعف یارب

```
systemctl enable rpcbind
systemctl enable nfs-server
systemctl enable nfs-lock
systemctl enable nfs-idmap
systemctl start rpcbind
systemctl start nfs-server
systemctl start nfs-lock
systemctl start nfs-idmap
```

میهدیم نآ هب ار ریز یاه permission و هدرک داچیا NFS تهج یارب یروتکریاد کی

```
mkdir /home/itg
chown -R nfsnobody:nfsnobody /home/itg
chmod -R 777 /home/itg
```

لخشم هندنک هدفاتسا یپیآ چنر یسرتسد اب |[etc/exports](#)| لیاف رد ار هدش صخش م یروتکریاد مینکی م

```
/home/itg          xxx. xxx. xxx. 0/24(rw, sync, no_subtree_check, no_root_squash)
```

مینکی م تراتسیر ار nfs server

```
systemctl restart nfs-server
```



```
firewall-cmd --permanent --zone=public --add-service=nfs  
firewall-cmd --permanent --zone=public --add-service=mountd  
firewall-cmd --permanent --zone=public --add-service=rpc-bind  
firewall-cmd --reload
```

# Install Rabbitmq and Volumes

دوش داچای ریز یروتکریاد زیتنربوک یاه دون مامت رد تسیابی مادتبا

```
mkdir -p /home/data/logs
```

و یرازگیاج حیحص تروصب ار {NFS\_IP} و {REGISTRY\_PORT} ، {REGISTRY\_IP} ریداقم ریز روتسد رد دینک ارجا

```
unzip install-rabbitmq-vol.zip
cd install-rabbitmq-vol
chmod +x install_volumes_rabbitmq.sh
./install_volumes_rabbitmq.sh {REGISTRY_IP} {REGISTRY_PORT} {NFS_IP}
```

مینکیم یسررب ار rabbitmq یارجا تیعضو ریز روتسد زا هدافتسا اب

```
kubectl -n infra get all
```

مینکیم یسررب ار اه Volume تیعضو ریز روتسد زا هدافتسا اب

```
kubectl -n itg get pv,pvc
```

# Install Integropia

و یرازگی اچ حیحص تروصب ار {NFS\_IP} و {REGISTRY\_PORT} ، {REGISTRY\_IP} ریداقم ریز روتسد رد دینک ارج

```
unzip install-itg.zip
cd install-itg
chmod +x install.sh
./install.sh
```

مینکیم کچ ار ESB یارجا تیعضو ریز روتسد زا هدافتسا اب

```
kubectl -n itg get all
```

# Install SSO

## اب SSO ب ص ن د ن ت س م Keycloak

ن ژرو Keycloak زا ه د ا ف ت س ا اب Single Sign-On (SSO) س ی و ر س کی ر ا ر ق ت س ا و د ا ج ا ل ح ا ر م د ن ت س م ن ی و ، Docker Registry ه ب ش و پ و د ل ی ب ، Docker Image ت خ ا س ل م ا ش ل ح ا ر م . د ه د ی م ح ی ص و ت ا ر 23 . ت س ا K u b e r n e t e s ی و ر ی و ل پ ی د

### ل ح ا ر م

## 1. Docker Image ت خ ا س

د و ش ی م ه د ا ف ت س ا ر ی ز `Dockerfile` ل ی ا ف ، Docker Image د ا ج ا ی ی ا ر ب

```
FROM quay.io/keycloak/keycloak:23.0.3 as builder

# Enable health and metrics support
ENV KC_HEALTH_ENABLED=true
ENV KC_METRICS_ENABLED=true

# Configure a database vendor
ENV KC_DB=postgres

WORKDIR /opt/keycloak

# for demonstration purposes only, please make sure to use proper certificates in production
instead

RUN keytool -genkeypair -storepass password -storetype PKCS12 -keyalg RSA -keysize 2048 -
  dname "CN=server" -alias server -ext "SAN: c=DNS: localhost, IP: 127.0.0.1" -keystore
  conf/server.keystore
```

```
ADD --chown=keycloak:keycloak ./com.mavartech.login-natcode-jar-with-dependencies.jar
/opt/keycloak/providers/com.mavartech.login-natcode-jar-with-dependencies.jar

RUN /opt/keycloak/bin/kc.sh build

FROM quay.io/keycloak/keycloak:23.0.3

COPY --from=builder /opt/keycloak/ /opt/keycloak/

# change these values to point to a running postgres instance
ENV KC_DB=postgres
#ENV KC_DB_URL=<DBURL>
#ENV KC_DB_USERNAME=<DBUSERNAME>
#ENV KC_DB_PASSWORD=<DBPASSWORD>
ENV KC_HOSTNAME=localhost
ENV KC_HOSTNAME_STRICT_HTTPS=false
ENV TZ=Asia/Tehran
ENTRYPOINT ["/opt/keycloak/bin/kc.sh", "start", "--debug", "--spi-theme-static-max-age=1", "--spi-theme-cache-themes=false", "--spi-theme-cache-templates=false"]
```

دش ه فاضا `/opt/keycloak/providers/` ی روت ک ریاد هب هدش لی اپ ماک اواج نی گالپ ، لی اف نی رد  
ت.س ا

**دلی ب روت سرد**

```
docker build -t keycloak-custom:23 .
```

ش فیر عت یا و ت ح م زا `23` هخ سن و `keycloak-custom` مان اب ار Docker Image روت سرد نی **حیضوت**  
دزاسیم `Dockerfile` رد

## Docker Registry هب Docker Image شوپ 2.

مینی کیم شوپ `192.168.30.26:5000` رورس یور Docker یرت سیجر هب ار هدش هتخاس Docker Image

**تاروت سرد**

```
docker tag keycloak-custom: 23 192.168.30.26: 5000/keycloak-custom: 23
```

شوپ یارب ات دنکیم داچیا هدشهتخاس Docker Image یارب دیدج گت کی روتسد نیا **حیضوت**  
دوش هدامآ یرتسیجر هب ندرک

```
docker push 192.168.30.26: 5000/keycloak-custom: 23
```

اسرا (|192.168.30.26: 5000|) هدشخصشم یرتسیجر هب ار Docker Image روتسد نیا **حیضوت**  
دنکیم

## Kubernetes یوریولپید 3.

دوشیم هدافتسا ریز |keycloak-deploy. yaml| لیاف زا Keycloak، رارقتسا یارب

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: keycloak-deployment
  namespace: nioc
  labels:
    app: keycloak
spec:
  replicas: 1
  selector:
    matchLabels:
      app: keycloak
  template:
    metadata:
      labels:
        app: keycloak
    spec:
      containers:
        - name: keycloak
          image: image
          volumeMounts:
            - mountPath: /opt/keycloak/themes
              name: files-vol
```

```
    subPath: keycloak/themes
env:
  - name: KC_DB_URL
    valueFrom:
      configMapKeyRef:
        name: integration-configs
        key: keycloak-db-url
  - name: KC_DB_USERNAME
    valueFrom:
      configMapKeyRef:
        name: integration-configs
        key: keycloak-db-username
  - name: KC_DB_PASSWORD
    valueFrom:
      configMapKeyRef:
        name: integration-configs
        key: keycloak-db-password
  - name: KC_HOSTNAME
    valueFrom:
      configMapKeyRef:
        name: integration-configs
        key: keycloak-db-hostname
  - name: KEYCLOAK_ADMIN
    valueFrom:
      configMapKeyRef:
        name: integration-configs
        key: keycloak-admin-user
  - name: KEYCLOAK_ADMIN_PASSWORD
    valueFrom:
      configMapKeyRef:
        name: integration-configs
        key: keycloak-admin-password
  - name: KC_HOSTNAME_STRICT_HTTPS
    value: "false"
  - name: KC_HOSTNAME_STRICT
    value: "false"
  - name: DEBUG_PORT
    value: " *:8787"
volumes:
  - name: files-vol
```

```
persistentVolumeClaim:
  claimName: pvc-files
```

---

```
apiVersion: v1
kind: Service
metadata:
  name: keycloak-service
  namespace: nioc
spec:
  selector:
    app: keycloak
  ports:
    - name: app
      protocol: TCP
      port: 8443
      targetPort: 8443
    - name: debug
      protocol: TCP
      port: 8787
      targetPort: 8787
```

ی:ولپی د روتسد

```
kubectl apply -f keycloak-deploy.yaml
```

هدرک لامغا Kubernetes رتسالک یور ار `keycloak-deploy.yaml` لیاف روتسد نی ا **حوضوت** دنکیم زاغآ ار Keycloak یولپید

## SSO هئارا یارب Ingress میظنت 4.

دوشیم هئارا نیمادباس یور SSO سیورس، ریز `ingress.yaml` لیاف زا هدافتسا اب

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
```



```

kubernetes.io/ingress.class: nginx
nginx.ingress.kubernetes.io/backend-protocol: HTTPS
nginx.ingress.kubernetes.io/keep-alive-requests: "10000"
nginx.ingress.kubernetes.io/large-client-header-buffers: 8 512k
nginx.ingress.kubernetes.io/max-worker-connections: "100000"
nginx.ingress.kubernetes.io/proxy-body-size: 8m
nginx.ingress.kubernetes.io/proxy-buffer-size: 256k
nginx.ingress.kubernetes.io/proxy-buffering: "on"
nginx.ingress.kubernetes.io/proxy-buffers: 4 512k
nginx.ingress.kubernetes.io/rewrite-target: /
nginx.ingress.kubernetes.io/ssl-redirect: "false"
nginx.ingress.kubernetes.io/upstream-keepalive-connections: "2000"
nginx.ingress.kubernetes.io/use-forwarded-headers: "true"
nginx.ingress.kubernetes.io/use-proxy-protocol: "true"
nginx.org/server-snippets: |
    listen 80 443 backlog=4096;
name: integration-ingress-ssl
namespace: nioc
spec:
  rules:
  - host: mysso.nioc.ir
    http:
      paths:
      - backend:
          service:
            name: keycloak-service
            port:
              number: 8443
          path: /
          pathType: Prefix

```

لایف روتس:

```
kubectl apply -f ingress.yaml
```

امیظنت و هدرک لامغا Kubernetes رتسالك یور ار `ingress.yaml` لایف روتس د نیا: **حیضوت**  
 دنکیم داچای ار SSO سیورس هب یسرتسد یارب Ingress

# هجیتن

گا. دوب دهاوخ سرتسد رد |myssso.nioc.ir| نیمادباس یور SSO سیورس، الاب لحارم یارجا زا سپ  
دینک هدافتسا یبایبیع یارب طبترم یاهگال زا، دمآشیپ هلحرم ره رد یلکش

## ی:سررب یارب دیفم تاروتسد

اهدآپ تیعضو یسررب

```
kubectl get pods -n nioc
```

دهدیم شیانم ار |nioc| namespace رد هدرقتسم یاهدآپ تیعضو روتسد نیا: **حیضوت**

دآپ یاهگال یسررب

```
kubectl logs [POD_NAME] -n nioc
```

شیانم تالکشم ای اطخ یسررب یارب ار صخشیم دآپ کی یاهگال روتسد نیا: **حیضوت**  
دهدیم