

Docker volumes

docker-volumes-nodejs-app

sudo git clone <https://github.com/rohitkumar1996/docker-volumes-nodejs-app>

cd docker-volumes-nodejs-app

ls
sudo nano Dockerfile

From
worker
copy
RUN
CMD

sudo docker build -t nodeedataapp:v1

① sudo docker run --name nodeapp -t -d -p
90:80 nodeedataapp:v1

② sudo docker exec -it nodeapp bash

ls
cd feedback

ls // dummy

Browser : 172.18.181.46 : 90

exit

① sudo docker rm -f nodeapp → accidentally removed

② sudo docker run --name

③ sudo docker exec -it nodeapp bash → up running
cd feedback
ls
exit

— data is lost no abs

Types of volume:

1. Bind mount
2. Anonymous Data
3. Docker Managed

where data is	After restart / deletion
On host system	same data
Docker volume	Fresh empty volume
Docker volume	Same volume

3.) Docker manager volumes

Production → resides on /var/lib/docker

① sudo docker rm -f

② sudo docker volume create dockervolume

sudo docker volume ls → list all the volumes

③ sudo docker --name nodeapp -it -d -p 90:80 -v dockervolume:/app feedback nodeapp:v1

④ sudo docker exec -it nodeapp bash

cd feedback

ls

Browser → put data

File types: Hello-world, hello.txt, dummy, etc.

ls

// dummy 'hi.txt' 'hello.txt'

exit

su root

switch to root

cd /var/lib/docker/volumes

→ Docker volume

ls

cd dockervolume

ls // data

cd -data/

ls // dummy 'hi.txt' 'hello.txt'

exit

→ stopping sudo docker ps

Docker command

ls /var/lib/docker

ls /var/lib/docker

Jenkins

+ New Item
cafestatic - bounded deploy

Save ✓

Configure

① Get
url → cafestatic-website.git

Branch → +/main

Build Steps

Execute Shell

cmd parts from cafestatic-deploy-shell-script.txt
 end parts from cafestatic-deploy-shell-script.txt
 // resync → similar to copy
 // sed → modify, edit, comment the content of a file
 // tee to awk @argp

Save ✓

+ New Item

cafestatic-docker-deploy
frustyle

Save ✓

Configure

① Get
url → cafestatic-website.git

Branch → +/main

Build Steps

Add plugins for docker

↳ Got to manage Jenkins → plugins → Available plugins
(if installed, installed plugin)

[Install] all docker named plugins

Now in Build Steps

↳ Docker Build and Publish

Repository Name

sheleyangl/cafestatic

image & storage name

Tag

v1

Registry Credentials

sheleyangl

IP: 3A.181.81.231

build ✓

Save
→ image successfully gets tagged but it will get permission denied error.

In Terminal

sudo chmod -R 777 /var/run/docker.sock
password: ✓

sudo usermod -aG docker jenkins

sudo systemctl restart jenkins

docker ls

wolf existing
without sudo
run docker ls

sudo usermod -aG docker \$USER

newgrp docker
displays nothing -> error / tests all images.
docker image ls

In Jenkins → cafestatic-docker-deploy

In build step [+ build step]

↳ Execute Docker command

Docker command

Create container

Image name

sheleyangl/cafestatic:v1

Container name

cafecapp -1

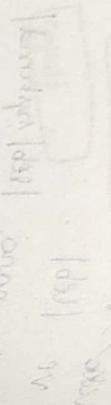
In Advanced [] → Publish all ports → port bindings
9180 different port

+ Build step Again
Execute docker command
 Docker command Start container
 container ID cafeapp
 [Save] build ✓
 172.18.181.46:91 → cafe static website
Create pipeline
 New view
 Type Build pipeline view
 [Create]
 → Edit view
 pipeline flow
 Select initial job cafe static baremetal
 [Save]
 In pipeline itself select initial job then configure post-build actions → Build on other projects
cafe static docker deploy
 • Trigger only if build is stable
 [Save]
Note: while running pipeline build on the don't run individually go to (cafe static - ci) & run directly there
 change v2 containerd cafe app

To Execute the Docker command
 cd python-fasttrack-program/
 @ docker run
 -v /lib
 sudo nano /lib/systemd/
 # Exec
 pasted
 G # Exec =
 Jenkins → manage jenkins → system → go to end docker builder
Docker url:
 unix:///var/run/docker.sock
 save
 In Docker deploy
 Build step
 execute docker command
 http://localhost:4243/api/v1.35/images/pull?repository=dockerfile:latest
 (①)
 run this cmd in terminal
 docker login -u shelbyangel
 sudo -u jenkins docker login -u shelbyangel
 password: (② access token from docker hub)
 G login succeeded.
 Manage jenkins → crypt credentials → system
 update
 log in password p caste → change password →
 user name → ed → disruption →
 [Save]

Docker compose

- * What all services are there & part of our app &
- * how storage is created.



- * MySQL → automatically source of data → mounted
- * MySQL data: /var/lib/mysql → automatically saves as mddb
- * If it needs to be backup of the folder for database.

host → machine → volumes → app → instance

host → volumes → instance

volumes:

- * MySQL-data → docker managed volume.

network → launches both MySQL database & webserver.

```

cd cafe-Dynamic-website
// 
sudo docker-compose up --build
  
```

```

Browser: 17.18.181.46:5000
          → coffee website runs
          → coffee stop →
  
```

Manually → both can speak to each other

cafe-Dynamic-website | docker-compose.yml

MySQL:

→ MySQL → pulling fresh image.

image: mysql: latest ← pulling fresh image.

variables: MySQL → port → 3306 → MySQL port

environment: MySQL → bridge port coming from MySQL → MySQL port → MySQL port

MySQL-Root-Password → MySQL root password → MySQL root password

MySQL-User → MySQL user → MySQL user

environment → using environmental variables → databases

user → MySQL user → MySQL user

password → MySQL password → MySQL password

port: 3306 → mapping the port

volumes: → folder → bind mount

host → volumes → host → bind mount

→ create db.sql what all data to be populated in side a folder

```

#!/bin/bash
sudo docker ps
sudo docker exec -it mysql bash
# ls
cd docker-interportant-control
# create-db.sql
  
```

mysql -u root -p

Moscow @ 123

Continuous Integration

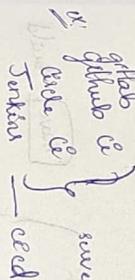
Show databases;
exit
cd /var/lib/mysql
" cd num.pop.dbs // todo explain
u exit

← environment variable created earlier
After building entire software only after that build is done (or off)
Developer was made to wait to find & fix the bug.
→ developer feedback to the developer.

- No continuous feedback to the developer.
- No productivity & slow process
- * infrequent build

What continuous Integration:

- Promote Tencourage the developer
- After every build (but the developer is given feedback)
- After commit, build, test & get feedback +
- * frequently committing the code to version control.



Jenkins → open source → used pipeline

* plug-ins for everything to integrate each tool after every stage. → executable / connect to other resources / sources

- Stand alone Mode. — same server
- Master slave Architecture — slave — ubuntu | slave — windows | slave — installed from and
- * Distributed architecture — Master — workers — executables
- Hybrid Mode

Waiting to work

MySQL Server → six workers about
MySQL Slave → six workers about
MySQL Hybrid → six workers about

po master-slave mode

package from

Dockerfile

Template

use the template by using Grunt scripts — In Jenkins

pipeline

Have later
aptly any

env → variable name

for pipeline stages

sh → to execute a shell command

who → to print

given syntax → get given script

script must be present showing

Tinkerfile → copy code, variables, functions with

pipeline-script

pipeline

[ok]

pipeline

paste

here

build

generate pipeline Script

Tomcat web

contexts

Tomcat 9.x Remote

contexts

Tomcat 9.0.40

contexts

pipeline Syntax:

(display:

sample step

WAR | EAR files

[* * /!*.war]

context path

mywarapp

Container

Tomcat 9.x Remote

contexts

Tomcat 9.0.40

contexts

Scanned with CamScanner

1) Simple Java - maven - app

① + New Project → Java - maven - compile
 Configure
 ② Get → URL
 Build Steps → Invoke top-level Maven targets
 Maven version : My maven
 Goals : clean validate compile

2) + New Project → Java - maven - codereview

① Get → URL
 Build Steps → Invoke top-level Maven targets
 Maven version : mymaven
 Goals : -P metrics pmd:pmd checkstyle:checkstyle findbugs:findbugs

Post build Actions :

Record complete maveneng & static analysis results.

Tools : PMD + Add tool
 CheckStyle
 FindBugs

plugins should be installed → Warnings
[https://...](#)

3) + New Project → Java - test cases

① Get → URL
 Build Steps → Invoke top-level Maven targets

Maven version : mymaven

Goals: test

Post build Actions: → publish JUnit test result report

Test reports XMLs: target/surefire-reports/*.xml

Test output retention: None

Java compile → code-review → test-cases → Java-maven-deploy

A) + New project → Java-package | Java-maven-deploy
 ① Git → URL
 Build steps → Invoke top-level Maven targets
 Maven version : mymaven
 Goals: package
 Build Step → Execute shell → command
`cd target
java -jar *.jar`

To build a pipeline:

+ New View → Name : Java-maven-pipeline
 ② Build pipeline View
 [create]

In Pipeline Flow : Select initial Job
 Java-maven-compile

[save]

Common step → Triggers
 Go to Codereview → Post-build Actions : → Build other projects

projects to build:

② Trigger only if build is Stable

X Go to codereview → Post-build other projects → Java-testcase
 Go to test cases → build other projects → Java-maven-deploy
 Go to Java

Go to codereview → Triggers → Java-maven-compile
 Go to Java-testcases → Triggers → Java-maven-codereview
 Go to Java-maven-deploy → Triggers → Java-kiteases

Go to pipeline → Run

2) JSP - Java Server Pages

Java-fsp-examples-maven-warapp-cd-d

- + New item → Java-fsp-example
① Get → URL
`https://java-fsp-maven-warapp-cd-d.git`

Build steps → Invoke top-level Maven targets

Maven version : my.maven

Goals : clean validate compile

Save build

- + New item → Java-fsp-codeanalysis

① Get → URL
Build steps → Invoke top-level Maven targets

Maven version : my.maven

Goals : -Pmetrics print:prod checkstyle:checkstyle:findbugs:pep

Post build action:

↳ Read complete warnings & static analysis results

Tools : PMD

checkstyle
findbugs

Build of workspace

- + New item → Java-fsp-deploy

① Get → URL
Build steps → Invoke top-level Maven targets

Maven version : my.maven

Goals : clean compile package

Post-build action → Deploy war file to a container

WAR/EAR files
* * / *.war

context path
My Maven App

Container → Tomcat 9.0.20 Remote

(localhost)

Tomcat 9.0.20 Remote
credentials null / 1234 (Tomcat9 - 181-60-3a.exe)
Tomcat URL `http://142.18.181.60:8081` gave blank

① Settings → credentials → global → [Add credentials]

username : mew
password : mew@123

id : tomcat manager script-cmd-60
description : ---

[create]

② Settings → plugins → Deploy to container < initial fail

to check Manage Rankens → Plugins

③ available plugin → ✓

After ① + ② deploy

To test if it's working

Working correctly for important tasks

3) Node.js - Monitoring - App

- + New item → nodejs-monitoring-pawaral

① Get → URL

☒ Reflect where this project can be run

label Expansion
ubuntuuser - 45

Build steps → Execute shell

Build steps → Execute shell
command
sudo apt-get install nodejs npm -y

sudo make test

sudo make test-report

sudo make run → X

Save build

In terminal:

```
sudo useradd -aG docker jenkins  
sudo systemctl restart jenkins  
ssh root@172.18.1.45  
cd jenkins  
ls //agent.jar  
sudo curl _____  
sudo java _____
```

Save build

② Get → URL

- + New item → nodejs-docker-deploy

☒ Reflect where this project can be run

label Expansion
ubuntuuser - 45

Build steps → Execute shell

command:

```
sudo docker run -f nodemonitoring  
sudo docker build . -f Dockerfile -t nodemonitoring  
sudo docker run --name nodemonitoring -it -d -p 3001:3000  
                                nodemonitoringapp:v2
```

Save build

⇒ Once connected then Build.

- * If agent is there after curl command no need to give curl directly run java → connected

Settings → Node → + New Node
Name: node-monitoring-decl
Launch method: permanent Agent
(create)

→ Remote root directory
File path: /var/www/jenkins

→ launch method
Launch agent by connecting it to the controller

#/master
Save

Save

Build step: nodejs - 45

Save

Build step: nodejs - 45

Save

To build a pipeline

modern + New View → Name: nadis-monitoring-cicd

① Build pipeline view

Create

In pipeline plan: Select initial Job
② nadis-monitoring-parametral

Save

Go to nadis-monitoring-parametral → Job
Part build Actions: Trigger parameterized build on other project

Build triggers: Projects to build

nadis-docker-deploy

Save

Trigger when build is not stable

Save

Trigger build without parameters

Save

'build'

Save

A) Python-Monitordapp-monitaring

+ New item → python-monitor-parametral

① Get → URL

* /master

UbuntuServer-35

Build steps → Execute shell

command sudo apt-get install python3-pytest

sudo make test

sudo make test-reaport

sudo make test-reaport

Build steps → Execute shell

command sudo apt-get install python3-pytest

sudo make test

sudo make test-reaport

label Expression

UbuntuServer-35

Get → URL

Build steps → Execute shell

command ls -l

sudo docker run -f pythonmonitoring1

sudo docker build -f build/Dockerfile -t pythonmonitordapp:1.0

sudo docker run --name pythonmonitordapp -it -d -p 5003:5000

pythonmonitordapp:v1

Save build

Save build

Save build

Save build

Save build

Save build

Imperative way

ssh min@172.18.181.

kubectl get nodes

// to know the details

{ 11 min.kubernetes.com → control-plane ← master.
min. " "
min. " "

→ // detailed info of a pod

kubectl get node -o wide

pods
// installed in default test place

kubectl get pods

kubectl get pods --all-namespaces

x kubectl delete deploy cafeweb

x kubectl get pods

container or pod

image name latest image is pulled.

→ kubectl run nginx --image nginx:latest

kubectl get pods

multiplexing
load bal.

→ service → outside world.

kubectl expose pod nginx --port 80 --target-port 80 --name nginx-service

kubectl get svc

NodePort → redundant

80: 32222 | TCP

To Cont. port remap to ip address of all workers

To access the container

① browser 172.18.18.60:32222 → port no.

worker of node Port

the cluster IP → private only accessed by us not through internet @ outside world
but only to pods. ex: mysql

→ CLUSTER-IP

To know

kubectl get pods -o wide

// deleted info of pods
Scheduler manager

The node has been

Nodes without any pods

"podname"

to describe a pod.

kubectl describe pods nginx

"kubectl get svc

1st pod will terminate one by one

with no down time.

Roll out

maximum upreach

spec

kubectl describe pods nginx

"kubectl -x

1 2 2 } roll out

1 1 2 } standard

1 1 1 } max scaled

cont name

kubectl get svc

re-create, little down-time

rollout strategy

minimum unavailable

kubectl set image deployment cafinweb cafinweb:static

"deployment"

Kubelet set image deployment cafinweb cafinweb:static

= see docker123/cafinwebstate:v2

Deployment → Abstraction

new Kubelet create deployment cafinweb --image

name: cafinwebstate

each version do set for v2.

kubectl get deployment -o yaml | grep "name: cafinwebstate"

"name: cafinwebstate" → check contains new

version

kubectl get pods

name: cafinwebstate

each version will be terminating

keep rolling this ↑

kubectl get all

name: cafinwebstate

each version will be terminating

keep rolling this ↑

kubectl get svc

name: cafinwebstate

each version will be terminating

keep rolling this ↑

kubectl expose deployment cafinweb --port 80

--type LoadBalancer

we can undo the deployment & go to previous version

→ we can undo the deployment & go to previous version

"reverse CHANGE-CASE"

→ kubectl set image deployment cafinweb cafinwebstate =

old ver

get ver

select

kubectl get svc

name: cafinwebstate

each version will be terminating

keep rolling this ↑

kubectl get svc

name: cafinwebstate

each version will be terminating

keep rolling this ↑

kubectl get svc

name: cafinwebstate

each version will be terminating

keep rolling this ↑

kubectl get svc

name: cafinwebstate

each version will be terminating

keep rolling this ↑

kubectl get svc

name: cafinwebstate

each version will be terminating

keep rolling this ↑

kubectl get svc

name: cafinwebstate

each version will be terminating

keep rolling this ↑

kubectl get svc

name: cafinwebstate

each version will be terminating

keep rolling this ↑

kubectl get svc

name: cafinwebstate

each version will be terminating

keep rolling this ↑

kubectl get svc

name: cafinwebstate

each version will be terminating

keep rolling this ↑

① Browser : 172.18.1.60:30604

RB... 1912013

status: latest
network interface
wavenet install kubernetes ← only on master

In Jenkins

+ New item → Kubernetes-cafetastic

→ URL [cafestatic-website-g](#)

① get

→ URL [cafestatic-website-g](#)

Build step: → Deploy to Kubernetes

Kube config
Kubernetes

config files
cafestatic-deployment.yaml, cafestatic-service

Go to next pipeline

↳



IP: 192.168.1.10
Port: 8080

values.yaml
values.yaml
values.yaml

Secret where passwords, API, tokens. — Secret object

Kudo get alone urc cafe-Dynamic-WebSite *initial work*

88h nniu@114.18.181.46 → nnnnnn

Kubecell get naked

When we were up

cd cafe-dynamic-website

phonetic

Ed Kubandu - Manifesto

Sudo nano message-passing system

Kubectl apply -f mysql-pv.yaml

Kubell get. No. 10

Initial separable deployment.

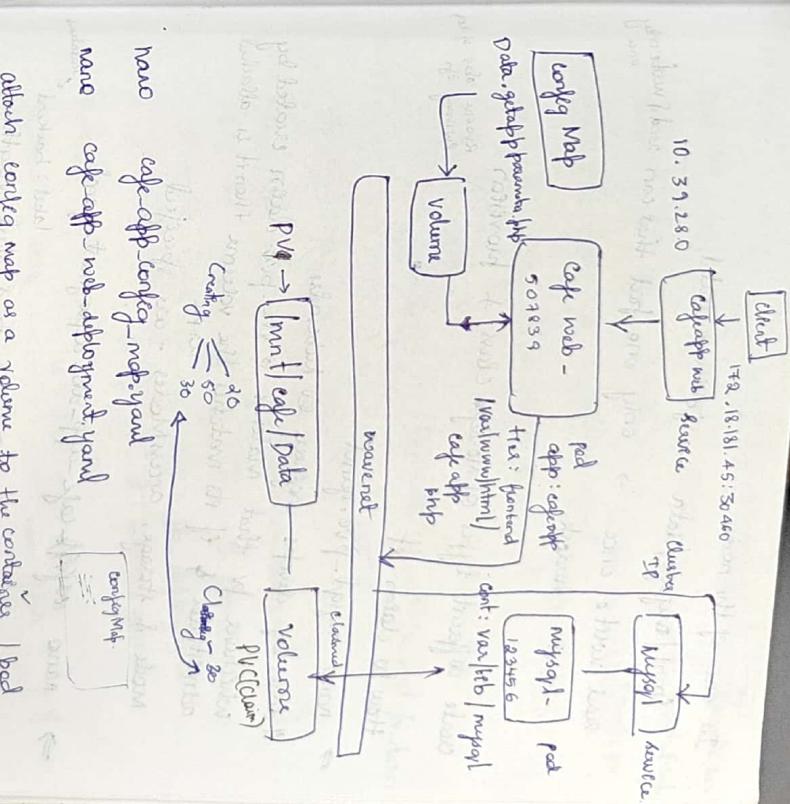
*
efficiently use your department & its resources

* As configuration changes we had to change the containers.
so ~~copying~~ new is defined all the configurations and attach
it to the container running inside a pod & further specified
& mounted inside the container.

present volumes of present volume claim

d develops well class

Mount | cap | data → var | lib | mysql



Scanned with CamScanner

In one of the nodes → folder is created

[mnt] mysql/data → only one fact that can read & write

read write once

with vols, backst

create different types of storage classes & provisioner

kubectl get pvc

second default mysql-pv-claim

How to claim of

→ nano mysql-pvc.yaml

↓ done by default attaches to the pvc-claim created by whichever pv that matches the volume than it is attached
admission & its matching in pvc

mode & storage, accessModes - are specified.

→ nano mysql-cafe-app-db-deployment.yaml

label: backed

strategy → recreate → terminate & start everything!

mysql-root-password → environment variable

mysqlUserPw: ~~otherwise all the way to mysql default~~

key: password

→ data: password: new!

→ nano mysql-pass.yaml

mysql Database. attach { configMap
volume: mysql-client-storage ->
an volume } kube
pvc

mysql -pv-client

mysql client: mySQL
long config param dont

keep this part → has mysql database field below

(but → ~~protect~~ police Reclaim Policy → reclaim → remains until
Storage bound
bound

→ kubectl apply -f cafe-app-configmap.yaml

kubectl describe configmap app

create pvc

kubectl get pvc

" no pvc "

kubectl apply -f mysql-pvc.yaml

second default mysql-pv-claim

kubectl get pvc

" mysql-pv "

kubectl get pods

name of pod

kubectl log .

name of pod

kubectl describe pods

name of pod

kubectl get service

name of pod

kubectl get nodes

name of node

kubectl get nodes

name of node

kubectl get pods

name of pod

kubectl exec -it cafe-web-... -- /bin/bash

name of pod

name of pod

cat getAppParameters.php

curl

kubectl get configmap

name of configmap

kubectl get secrets

name of secret

name of secret

secretName: "mySQL"

long config param dont

keep this part → has mysql database field below

(but → ~~protect~~ police Reclaim Policy → reclaim → remains until
Storage bound
bound

but = "mySQL";

restart the deployment

Kubernetes rollout what deploy/kubernetes

Kubernetes get pods

Kubernetes exec -it pod name kubectl exec -it -n kube-system /bin/bash

cat getAppParameters.php

\$appurl = "myurl"

exit

core-dns ~ myself

curl http://localhost

Kubernetes rollout restart deploy/kubernetes

curl http://localhost:3000/api/pods | jq .status.phase

③ Cloud9, ECR, ECS, EC2

④ Cloud9 open

Terminal → Prod

Copy & paste the curl cmd from dashboard page.

cd 1-no-containers/

npm start

New terminal → npm install

npm start

curl http://localhost:3000

// ready to receive report

curl http://localhost:3000/api/users

// returns json

cd 2-no-containers

ls

cat Dockerfile

sudo docker build -t mob-repo:latest

cat Dockerfile

sudo docker build -t mob-repo:latest

EOF → create

name → mob-supportkit → create

⑤ mob-repo → View push command

copy the token & auth

path in terminal.

[git] [push]

Application into Microservices.

relays don't work with this

relays

ECS

ECS → Image URI

[path]

container port: 3000

CPU: 0.5

Memory hard limit: 1 GB
Memory soft limit: 1 GB

- ② Elastic Container Service
- Cluster → Create Cluster
- Cluster name: [mob-cluster]
- CloudWatch Metrics: Amazon CloudWatch Metrics
- Infra: Fargate and self-managed instances
- ③ Create a new Auto Scaling group
- ④ Create a new Auto Scaling group
- provisioning mode: On demand
- Container runtime: Amazon Linux 2018.03
- Container labels: Name [mob-containter]

- ECS → Display Create Service
- Service name: [mob-task-service]
- Compute config: Launch type: [ecs]
- Launch type: ECS
- VPC - Lab - VPC
- Networking: VPC → Lab VPC
- Security group: sg-ECSSG
- Load Balancer: Create a new load balancer
- Target group: Create target group name: [mob-target]
- Target group: Target group name: [mob-target]
- ECS → Active Previously
- Go to ECS → Task definition load balancer → [mob-db]
- DNS name: Copy open in Browser
- Port: 3000
- Status: Ready to receive requests
- In URL → ...lapptunes

Scanned with CamScanner

ECR → npx → mob-solo → URL copy

terminal : cd ..

cd 5-containers

// ports

ls
sudo docker build -t mob-wireless-lab

sudo docker build -t mob-wireless-lab

task def family [mob-wireless-task]

Container config

Infrastructure config [Amazon EC2 instance]

Memory [1GB]

⑥ ECR → create [mob-wireless]

repository name [mob-wireless] → create [repo]

⑦ mob-wireless → view push commands.

1. wcr → wcr 2. wcr → wcr 3. wcr → wcr 4. wcr → wcr 5. wcr → wcr

containers-1

[mob-wireless-container]

task def

CRD - custom resource definition

marker note

status → target , service discovery

Nothing else.

Annotations

SSH session 172.18.181. —

of about 1 hour. In general, we can see that

Kubect get nodes

↳ Kubect get pods

↳ Prometheus

↳ Kubect get pods

↳ Kubect get pods

↳ Kubect apply -f nodejs -prometheus -deployment.yaml

172.18.18. — : 30080

172.18.18. — /metrics endpoint

put http://172.18.18.1:30080/metrics

interval: 1s

target

graph

use of source

nano nodejs - prometheus - /Metrics_Service_Monitor.yaml

Kubect apply -f nodejs - prometheus - service-monitor.yaml

Kubect get pods

Kubect get service monitor

Kubect get svc

Kubect get pods → no wide → // list of all nano result

① target

② graph

nano grafana - deploy - svc.yaml

Kubect apply -f grafana - deploy - svc.yaml

Kubect get pods

③ 172.18.181. — : 30030

admin

marker note

database → +Add → Node source → open → configure

Name → prometheus.

URL → http://prometheus:9090

nano prometheus - svc.yaml

Kubect apply -f prometheus - svc.yaml

Kubect get svc

Kubect edit svc prometheus

Kubect get svc