

JAVA SPRING BOOT DEPLOY ON KUBERNETES WITH DB

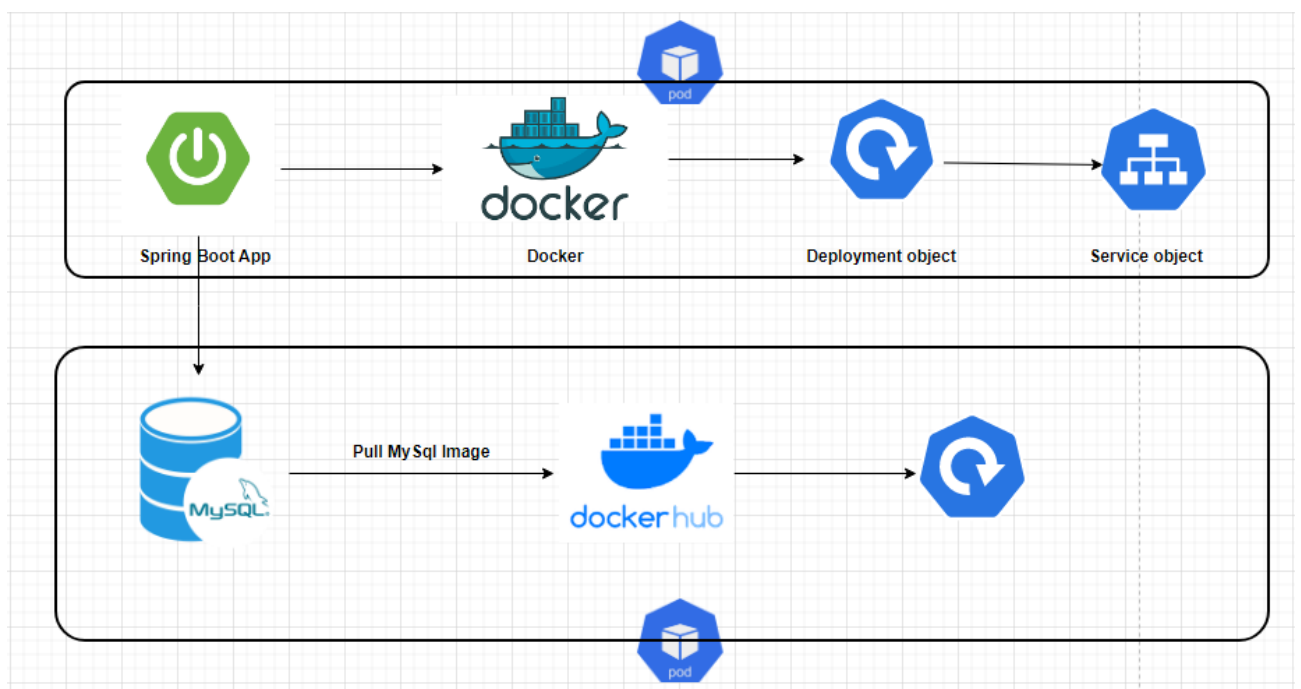
Requirements :

- Git - local version control system.
- GitHub - As Distributed version control system.
- Maven - As a Build Tool.
- docker -Containerization
- Kubernetes - As Container Management Tool.

Resources required :

- Free Tier AWS account.
- GitHub account (for source code and documentation).
- MobaXterm – enhanced terminal for windows with X11 Server tabbed SSH clients, network tool and much more.
- Git – local version control system.

Flow Diagram :



Setup Springboot Server

1. Launch an instance from an Amazon Linux 2 AMI with t2.medium

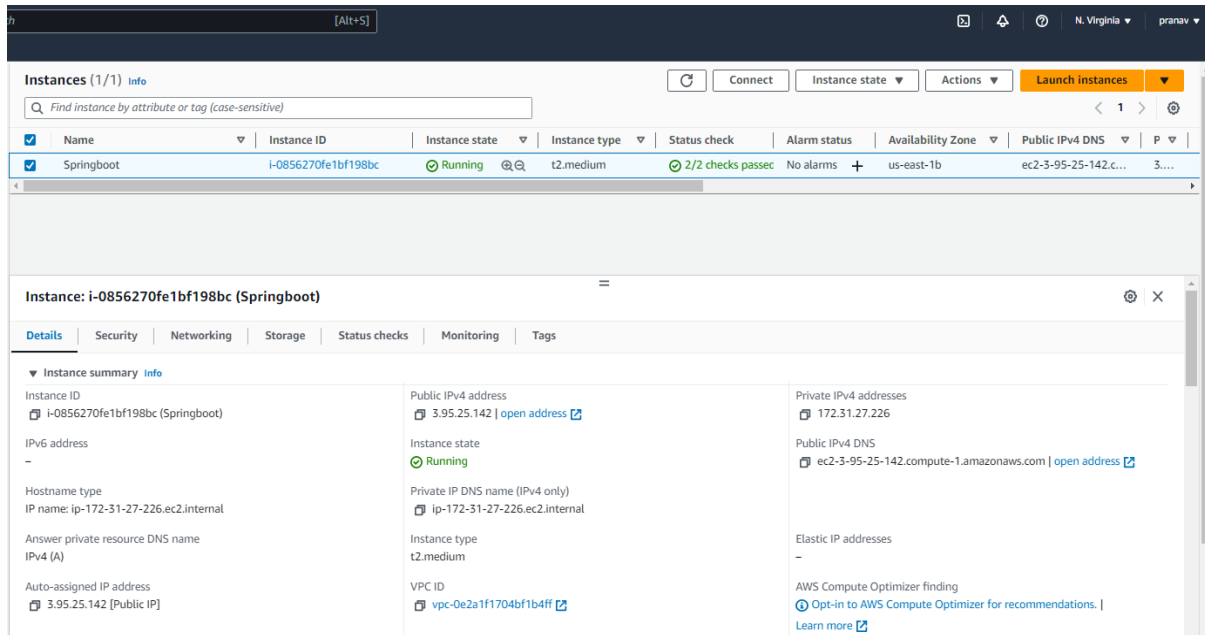


Fig.EC2 Instance for Springboot Server

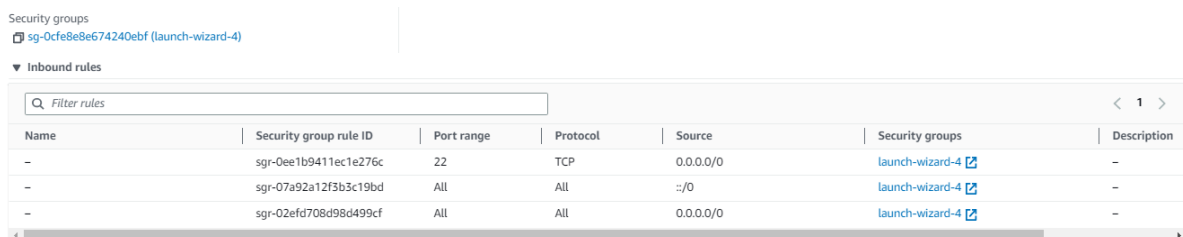


Fig.Security Group

2. Connect to your instance.
3. Update the packages and package caches you have installed on your instance.

```
Authenticating with public key "Imported-Openssh-Key"

• MobaXterm Personal Edition v22.0 •
  (SSH client, X server and network tools)

> SSH session to ec2-user@3.95.25.142
  • Direct SSH : ✓
  • SSH compression : ✓
  • SSH-browser : ✓
  • X11-forwarding : ✗ (disabled or not supported by server)

> For more info, ctrl+click on help or visit our website.

_ _ | _ _ | _ _ |
_| ( _ _ | _ _ | /
_| \ _ _ | _ _ |

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-27-226 ~]$ sudo su
[root@ip-172-31-27-226 ec2-user]# yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
```

4. Install the latest Docker Engine packages

```
[root@ip-172-31-27-226 ec2-user]# amazon-linux-extras install docker
Installing docker
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-docker
amzn2extra-kernel-5.10
(1/7): amzn2-core/2/x86_64/group.gz
(2/7): amzn2-core/2/x86_64/updateinfo
(3/7): amzn2extra-kernel-5.10/2/x86_64/updateinfo
(4/7): amzn2extra-docker/2/x86_64/primary_db
(5/7): amzn2extra-docker/2/x86_64/updateinfo
(6/7): amzn2extra-kernel-5.10/2/x86_64/primary_db
(7/7): amzn2-core/2/x86_64/primary_db
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:20.10.23-1.amzn2.0.1 will be installed
--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.23-1.amzn2.0.1.x86_64
--> Processing Dependency: libcgroup >= 0.40.rc1-5.15 for package: docker-20.10.23-1.amzn2.0.1.x86_64
--> Processing Dependency: containerd >= 1.3.2 for package: docker-20.10.23-1.amzn2.0.1.x86_64
--> Processing Dependency: pigz for package: docker-20.10.23-1.amzn2.0.1.x86_64
--> Running transaction check
--> Package containerd.x86_64 0:1.6.19-1.amzn2.0.1 will be installed
--> Package libcgroup.x86_64 0:0.41-21.amzn2 will be installed
--> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
--> Package runc.x86_64 0:1.1.4-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
docker x86_64 20.10.23-1.amzn2.0.1 amzn2extra-docker 41 M
Installing for dependencies:
containerd x86_64 1.6.19-1.amzn2.0.1 amzn2extra-docker 27 M
libcgroup x86_64 0.41-21.amzn2 amzn2-core 66 k
pigz x86_64 2.3.4-1.amzn2.0.1 amzn2-core 81 k
runc x86_64 1.1.4-1.amzn2.0.1 amzn2extra-docker 2.9 M
=====

Transaction Summary
Install Package: 1
Install Dependency: 4
Total Size: 77.6 MB
```

Docker Version:

```
[root@ip-172-31-27-226 ec2-user]# docker -v
Docker version 20.10.23, build 7155243
```

5. Start the Docker service

```
[root@ip-172-31-27-226 ec2-user]# service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: https://docs.docker.com
[root@ip-172-31-27-226 ec2-user]# service docker start
Redirecting to /bin/systemctl start docker.service
```

6. Install Conntrack and git:

```
[root@ip-172-31-27-226 ec2-user]# yum install conntrack -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package conntrack-tools.x86_64 0:1.4.4-5.amzn2.2 will be installed
--> Processing Dependency: libnetfilter_cttimeout.so.1(LIBNETFILTER_CTTIMEOUT 1.1)(64bit) for package: conntrack-tools-1.4.4-5.amzn2.2.x86_64
--> Processing Dependency: libnetfilter_cthelper.so.1(LIBNETFILTER_CTHELPER 1.0)(64bit) for package: conntrack-tools-1.4.4-5.amzn2.2.x86_64
--> Processing Dependency: libnetfilter_queue.so.1()(64bit) for package: conntrack-tools-1.4.4-5.amzn2.2.x86_64
--> Processing Dependency: libnetfilter_cttimeout.so.1()(64bit) for package: conntrack-tools-1.4.4-5.amzn2.2.x86_64
--> Processing Dependency: libnetfilter_cthelper.so.0()(64bit) for package: conntrack-tools-1.4.4-5.amzn2.2.x86_64
--> Running transaction check
--> Package libnetfilter_cthelper.x86_64 0:1.0.0-10.amzn2.1 will be installed
--> Package libnetfilter_cttimeout.x86_64 0:1.0.0-6.amzn2.1 will be installed
--> Package libnetfilter_queue.x86_64 0:1.0.2-2.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                                Arch                                Version                                Repository                                Size
=====
Installing:
conntrack-tools                        x86_64                              1.4.4-5.amzn2.2                      amzn2-core                                186 k
Installing for dependencies:
libnetfilter_cthelper                 x86_64                              1.0.0-10.amzn2.1                    amzn2-core                                18 k
libnetfilter_cttimeout                 x86_64                              1.0.0-6.amzn2.1                     amzn2-core                                18 k
libnetfilter_queue                     x86_64                              1.0.2-2.amzn2.0.2                   amzn2-core                                24 k
Transaction Summary
=====
Install 1 Package (+3 Dependent packages)

Total download size: 245 k
Installed size: 655 k
Downloading packages:
```

```
[root@ip-172-31-27-226 ec2-user]# yum install git -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.39.2-1.amzn2.0.1 will be installed
--> Processing Dependency: perl-Git = 2.39.2-1.amzn2.0.1 for package: git-2.39.2-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core-doc = 2.39.2-1.amzn2.0.1 for package: git-2.39.2-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core = 2.39.2-1.amzn2.0.1 for package: git-2.39.2-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.39.2-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Git::I18N) for package: git-2.39.2-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Git) for package: git-2.39.2-1.amzn2.0.1.x86_64
--> Running transaction check
--> Package git-core.x86_64 0:2.39.2-1.amzn2.0.1 will be installed
--> Package git-core-doc.noarch 0:2.39.2-1.amzn2.0.1 will be installed
--> Package perl-Git.noarch 0:2.39.2-1.amzn2.0.1 will be installed
--> Processing Dependency: perl(Error) for package: perl-Git-2.39.2-1.amzn2.0.1.noarch
--> Package perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2 will be installed
--> Running transaction check
--> Package perl-Error.noarch 1:0.17020-2.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                                Arch                                Version                                Repository                                Size
=====
Installing:
git                                    x86_64                              2.39.2-1.amzn2.0.1                  amzn2-core                                65 k
Installing for dependencies:
git-core                             x86_64                              2.39.2-1.amzn2.0.1                  amzn2-core                                8.8 M
git-core-doc                         noarch                              2.39.2-1.amzn2.0.1                  amzn2-core                                3.0 M
perl-Error                           noarch                              1:0.17020-2.amzn2                   amzn2-core                                32 k
perl-Git                             noarch                              2.39.2-1.amzn2.0.1                  amzn2-core                                41 k
perl-TermReadKey                     x86_64                              2.30-20.amzn2.0.2                   amzn2-core                                31 k
Transaction Summary
=====
Install 1 Package (+5 Dependent packages)
```

Git version:

```
[root@ip-172-31-27-226 ec2-user]# git version
git version 2.39.2
```

7. Install k8

```
[root@ip-172-31-27-226 ec2-user]# curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 80.0M 100 80.0M 0 0 113M 0 --:--:-- --:--:-- --:--:-- 112M
[root@ip-172-31-27-226 ec2-user]# sudo install minikube-linux-amd64 /usr/local/bin/minikube
```

8. Start Minikube

```
[root@ip-172-31-27-226 ec2-user]# /usr/local/bin/minikube start --force --driver=docker
* minikube v1.30.1 on Amazon 2 (xen/amd64)
! minikube skips various validations when --force is supplied; this may lead to unexpected behavior
* Using the docker driver based on user configuration
* The "docker" driver should not be used with root privileges. If you wish to continue as root, use --force.
* If you are running minikube within a VM, consider using --driver=none:
* https://minikube.sigs.k8s.io/docs/reference/drivers/none/
* Using Docker driver with root privileges
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.26.3 preload ...
  > preloaded-images-k8s-v18-v1...: 397.02 MiB / 397.02 MiB 100.00% 80.27 M
  > gcr.io/k8s-minikube/kicbase...: 373.53 MiB / 373.53 MiB 100.00% 51.32 M
* Creating docker container (CPUs=2, Memory=2200MB) ...
* Preparing Kubernetes v1.26.3 on Docker 23.0.2 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Verifying Kubernetes components...
* Enabled addons: storage-provisioner, default-storageclass
* kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Minikube version:

```
[root@ip-172-31-27-226 ec2-user]# /usr/local/bin/minikube version
minikube version: v1.30.1
commit: 08896fd1dc362c097c925146c4a0d0dac715ace0
```

9. Install kubectl

```
[root@ip-172-31-27-226 ec2-user]# curl -LO "https://dl.k8s.io/release/${curl -L -s https://dl.k8s.io/release/stable.txt}/bin/linux/amd64/kubectl"
% Total    % Received % Xferd  Average Speed   Time    Time     Current
           Dload  Upload   Total   Spent    Left     Speed
100 138    100 138    0    0  3537      0  --:--:-- --:--:-- --:--:-- 3631
100 46.9M 100 46.9M    0    0 100M      0  --:--:-- --:--:-- --:--:-- 100M
[root@ip-172-31-27-226 ec2-user]# sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
[root@ip-172-31-27-226 ec2-user]# ls -lRta
total 130088
-rw-r--r-- 1 ec2-user ec2-user 231 Jul 15 2020 .bashrc
-rw-r--r-- 1 ec2-user ec2-user 193 Jul 15 2020 .bash_profile
-rw-r--r-- 1 ec2-user ec2-user 18 Jul 15 2020 .bash_logout
drwxr-xr-x 3 root root 22 May 29 08:01 ..
drwx----- 2 ec2-user ec2-user 29 May 29 08:01 .ssh
drwx----- 3 ec2-user root 25 May 29 08:03 .cache
-rw-r--r-- 1 root root 83937631 May 29 08:05 minikube-linux-amd64
drwx----- 4 ec2-user ec2-user 131 May 29 08:09 .
-rw-r--r-- 1 root root 49258496 May 29 08:09 kubectl
```

Kubectl version:

```
[root@ip-172-31-27-226 ec2-user]# /usr/local/bin/kubectl version
WARNING: This version information is deprecated and will be replaced with the output from kubectl version --short. Use --output=yaml|json to get the full version
.
Client Version: version.Info{Major:"1", Minor:"27", GitVersion:"v1.27.2", GitCommit:"7f6f68fdabc4df88cfea2dcf9a19b2b830f1e647", GitTreeState:"clean", BuildDate:"2023-05-17T14:20:07Z", GoVersion:"go1.20.4", Compiler:"gc", Platform:"linux/amd64"}
Kustomize Version: v5.0.1
Server Version: version.Info{Major:"1", Minor:"26", GitVersion:"v1.26.3", GitCommit:"9e644106593f3f4aa98f8a84b23db5fa378900bd", GitTreeState:"clean", BuildDate:"2023-03-15T13:33:12Z", GoVersion:"go1.19.7", Compiler:"gc", Platform:"linux/amd64"}
```

9.1.Clone the repo

```
023-05-15 15:33:12Z ; Governor: go1.19.7 ; Computer: gc ; Platform: linux/amd64 ]
[root@ip-172-31-27-226 ec2-user]# cd /opt/
[root@ip-172-31-27-226 opt]# git clone https://github.com/spranav07/Springboot-k8s.git
Cloning into 'Springboot-k8s'...
remote: Enumerating objects: 78, done.
remote: Counting objects: 100% (78/78), done.
remote: Compressing objects: 100% (50/50), done.
remote: Total 78 (delta 2), reused 78 (delta 2), pack-reused 0
Receiving objects: 100% (78/78), 33.04 MiB | 41.87 MiB/s, done.
Resolving deltas: 100% (2/2), done.
```

```
[root@ip-172-31-27-226 opt]# ll
total 0
drwxr-xr-x 4 root root  33 May 18 22:57 aws
drwx--x--x 4 root root  28 May 29 08:04 containerd
drwxr-xr-x 2 root root   6 Aug 16  2018 rh
drwxr-xr-x 6 root root 216 May 29 08:11 Springboot-k8s
```

```
[root@ip-172-31-27-226 Springboot-k8s]# ls -lrta
total 32
drwxr-xr-x 6 root root   67 May 29 08:11 ..
drwxr-xr-x 4 root root   30 May 29 08:11 src
drwxr-xr-x 2 root root  178 May 29 08:11 .settings
-rw-r--r-- 1 root root   86 May 29 08:11 README.md
-rw-r--r-- 1 root root 1098 May 29 08:11 .project
-rw-r--r-- 1 root root 2062 May 29 08:11 pom.xml
-rw-r--r-- 1 root root 1710 May 29 08:11 Install.txt
-rw-r--r-- 1 root root  144 May 29 08:11 Dockerfile
-rw-r--r-- 1 root root 2209 May 29 08:11 db-deployment.yaml
-rw-r--r-- 1 root root 1236 May 29 08:11 .classpath
-rw-r--r-- 1 root root 1289 May 29 08:11 app-deployment.yaml
drwxr-xr-x 6 root root   216 May 29 08:11 .
drwxr-xr-x 6 root root   154 May 29 08:11 target
drwxr-xr-x 8 root root   163 May 29 08:11 .git
```

10. Make the DB UP

```
[root@ip-172-31-27-226 Springboot-k8s]# /usr/local/bin/kubectl get pods
No resources found in default namespace.
```

Creating Database:

```
[root@ip-172-31-27-226 Springboot-k8s]# /usr/local/bin/kubectl create -f db-deployment.yaml
persistentvolumeclaim/mysql-pv-claim created
deployment.apps/mysql created
service/mysql created
```

Display Pods:

```
[root@ip-172-31-27-226 Springboot-k8s]# /usr/local/bin/kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
mysql-6c748b7c67-ll62c             0/1     ContainerCreating   0           7s
```

Entering Mysql Database:

```
[root@ip-172-31-27-226 Springboot-k8s]# /usr/local/bin/kubectl exec -it mysql-6c748b7c67-1l62c /bin/bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
```

```
bash-4.2# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.42 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| pranav |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> exit
Bye
bash-4.2# exit
exit
```

Fig.Show Databases

11.Install Maven

```
[root@ip-172-31-27-226 Springboot-k8s]# yum install maven -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.7 kB 00:00:00
Resolving Dependencies
--> Running transaction check
--> Package maven.noarch 0:3.0.5-17.amzn2 will be installed
--> Processing Dependency: java >= 1.5 for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: sisu-inject-plexus for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: sisu-inject-bean for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: plexus-utils for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: plexus-sec-dispatcher for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: plexus-interpolation for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: plexus-containers-component-annotations for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: plexus-cipher for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: objectweb-asm for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.sonatype.sisu:sisu-inject-plexus) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.sonatype.plexus:plexus-sec-dispatcher) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.sonatype.plexus:plexus-cipher) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.sonatype.aether:aether-util) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.sonatype.aether:aether-spi) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.sonatype.aether:aether-impl) for package: maven-3.0.5-17.amzn2.noarch
```

Maven Version:

```
[root@ip-172-31-27-226 Springboot-k8s]# mvn -v
Apache Maven 3.0.5 (Red Hat 3.0.5-17)
Maven home: /usr/share/maven
Java version: 17.0.7, vendor: Amazon.com Inc.
Java home: /usr/lib/jvm/java-17-amazon-corretto.x86_64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.10.179-166.674.amzn2.x86_64", arch: "amd64", family: "unix"
```


12. Create the docker image

```
[root@ip-172-31-27-226 Springboot-k8s]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
gcr.io/k8s-minikube/kicbase	v0.0.39	67a4b1138d2d	7 weeks ago	1.05GB

Fig.Display images

```
[root@ip-172-31-27-226 Springboot-k8s]# docker build -t pranav27/springboot-crud-k8s:1.0 .
Sending build context to Docker daemon 73.36MB
Step 1/4 : FROM openjdk:8
8: Pulling from library/openjdk
001c52e26ad5: Pull complete
d9d4b9b6e964: Pull complete
2068746827ec: Pull complete
9daef329d350: Pull complete
d85151f15b66: Pull complete
52a8c426d30b: Pull complete
8754a66e0050: Pull complete
Digest: sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937f5452cb8
Status: Downloaded newer image for openjdk:8
--> b273004037cc
Step 2/4 : EXPOSE 8080
--> Running in 96f893d99178
Removing intermediate container 96f893d99178
--> b69872068d3b
Step 3/4 : ADD target/springboot-crud-k8s.jar springboot-crud-k8s.jar
--> 00f585aa1d3e
Step 4/4 : ENTRYPOINT ["java","-jar","/springboot-crud-k8s.jar"]
--> Running in 9094504507c4
Removing intermediate container 9094504507c4
--> 4a1322ceeac4
Successfully built 4a1322ceeac4
Successfully tagged pranav27/springboot-crud-k8s:1.0
```

Fig.creating Image

```
[root@ip-172-31-27-226 Springboot-k8s]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
pranav27/springboot-crud-k8s	1.0	4a1322ceeac4	18 minutes ago	565MB
gcr.io/k8s-minikube/kicbase	v0.0.39	67a4b1138d2d	7 weeks ago	1.05GB
openjdk	8	b273004037cc	10 months ago	526MB

13. docker login [Docker Hub account created]

```
[root@ip-172-31-27-226 Springboot-k8s]# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: pranav27
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```


14.Push image to Docker Hub

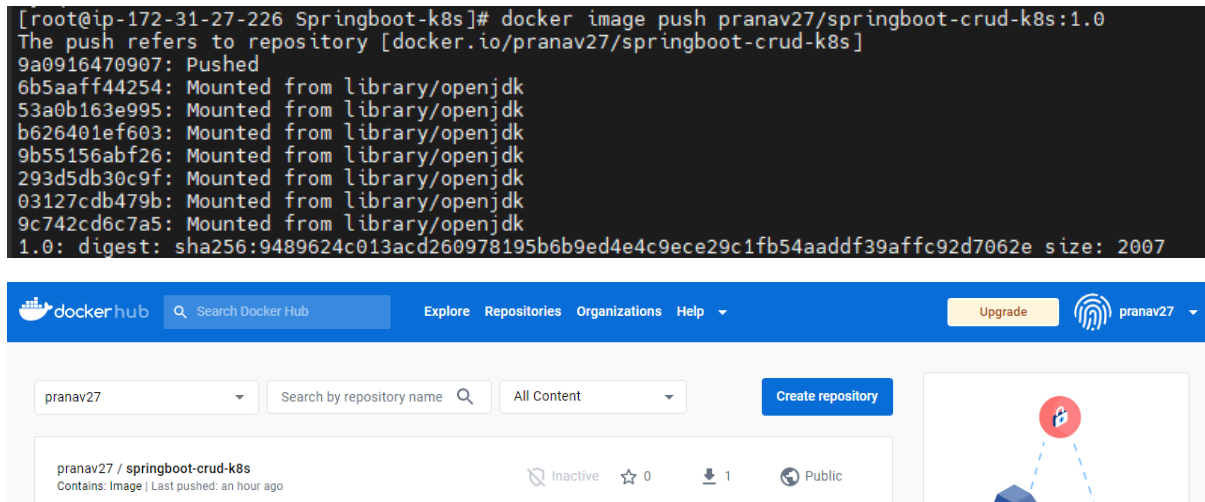


Fig.Docker Hub

15.

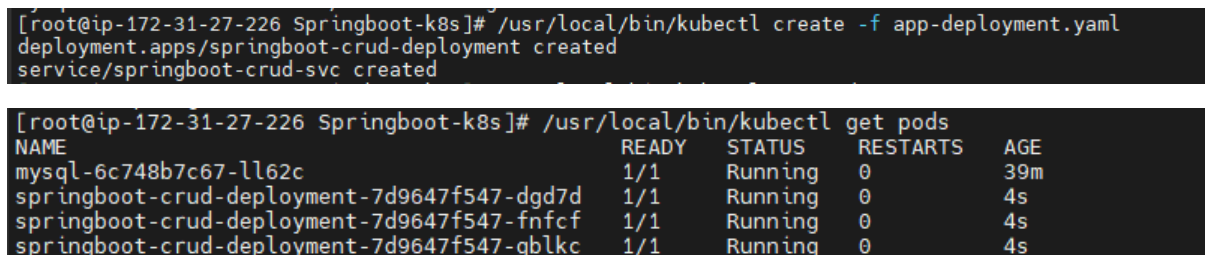


Fig.Display pods

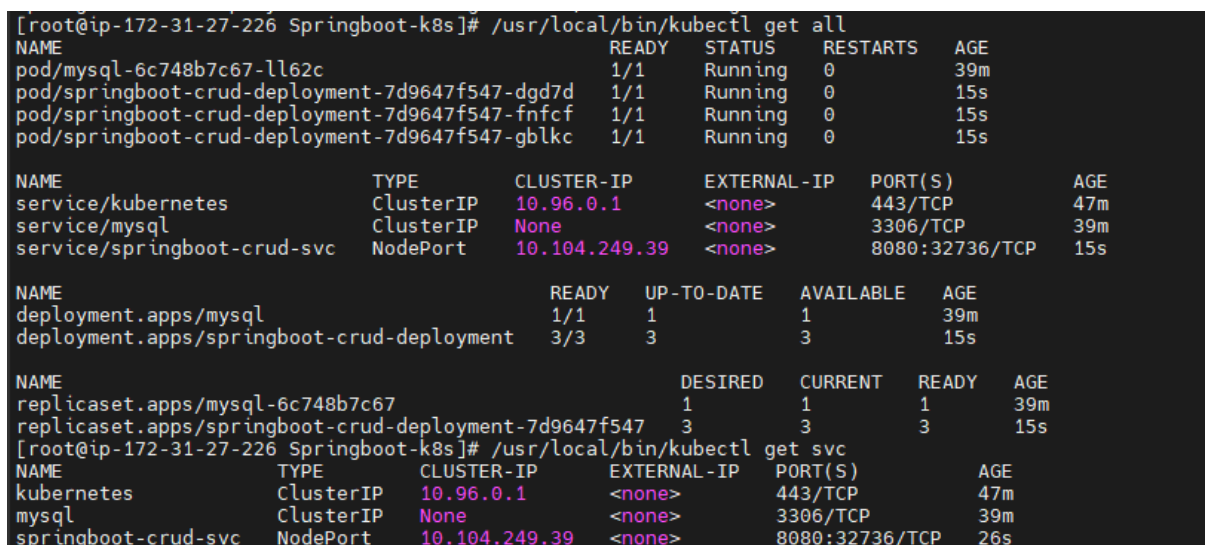


Fig.Display pods,deployment,services

16. To display minikube ip

```
[root@ip-172-31-27-226 Springboot-k8s]# /usr/local/bin/minikube ip  
192.168.49.2
```

17. Put port Forward

```
[root@ip-172-31-27-226 Springboot-k8s]# /usr/local/bin/kubectl port-forward --address 0.0.0.0 svc/springboot-crud-svc 8080:8080 &  
[1] 131669  
[root@ip-172-31-27-226 Springboot-k8s]# Forwarding from 0.0.0.0:8080 -> 8080  
Handling connection for 8080  
Handling connection for 8080  
Handling connection for 8080
```

18. JSON data to be hitted with post

The screenshot displays a REST client interface with the following details:

- Overview:** POST http://3.95.25.142:8080/orders
- Method:** POST
- URL:** http://3.95.25.142:8080/orders
- Body:** JSON format with the following content:

```
1 {  
2   "name": "chairs",  
3   "qty": 25,  
4   "price": 5000  
5 }
```
- Response:** 200 OK, 227 ms, 212 B. The response body is shown in a pretty-printed JSON format:

```
1 {  
2   "id": 3,  
3   "name": "chairs",  
4   "qty": 25,  
5   "price": 5000.0  
6 }
```

18.1 .Check Database

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| pranav    |
| sys       |
+-----+
5 rows in set (0.00 sec)
```

Fig.Display databases

```
mysql> show tables;
+-----+
| Tables_in_pranav |
+-----+
| hibernate_sequence |
| orders_tbl |
+-----+
2 rows in set (0.00 sec)
```

Fig.Display Tables

```
mysql> select * from orders_tbl;
+----+-----+-----+-----+
| id | name  | price | qty |
+----+-----+-----+-----+
| 1  | shoes | 6999  | 5   |
| 2  | keys  | 2000  | 100 |
| 3  | chairs | 5000  | 25  |
| 4  | doors | 10000 | 4   |
+----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Fig.Display items from orders_tbl

19. To get the required item using GET

The screenshot shows a REST client interface with the following components:

- Overview Tab:** Displays the request method (GET) and URL (http://3.95.25.142:8080/orders/3). It includes a 'Send' button and a 'Save' button.
- Params Tab:** Shows the query parameters. A table is visible with the following data:

Key	Value	Description
Key	Value	Description

Body Tab: Shows the response body in JSON format. The response is a JSON object with the following structure:

```
{  "id": 3,  "name": "chairs",  "qty": 25,  "price": 5000.0}
```

The response status is 200 OK, with a response time of 495 ms and a response size of 212 B. The response is saved as an example.

20.For Dashboard

IN ONE TERMINAL

```
[root@ip-172-31-27-226 Springboot-k8s]# /usr/local/bin/kubectl proxy --address='0.0.0.0' --accept-hosts='^*$'  
Starting to serve on [::]:8001
```

IN OTHER TERMINAL

```
Last login: Mon May 29 08:03:14 2023 from 122.172.87.21  
  
  _|  _|_ )  
 _| ( _|_ /  Amazon Linux 2 AMI  
 _|\__|__|  
  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-172-31-27-226 ~]$ sudo su -  
Last login: Mon May 29 08:03:22 UTC 2023 on pts/0  
[root@ip-172-31-27-226 ~]# /usr/local/bin/minikube dashboard  
* Enabling dashboard ...  
  - Using image docker.io/kubernetesui/dashboard:v2.7.0  
  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8  
* Some dashboard features require the metrics-server addon. To enable all features please run:  
  
    minikube addons enable metrics-server  
  
* Verifying dashboard health ...  
* Launching proxy ...  
* Verifying proxy health ...  
http://127.0.0.1:33035/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/
```

21.Hit Url in browser

<http://3.95.25.142:8001/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/workloads?namespace=default>

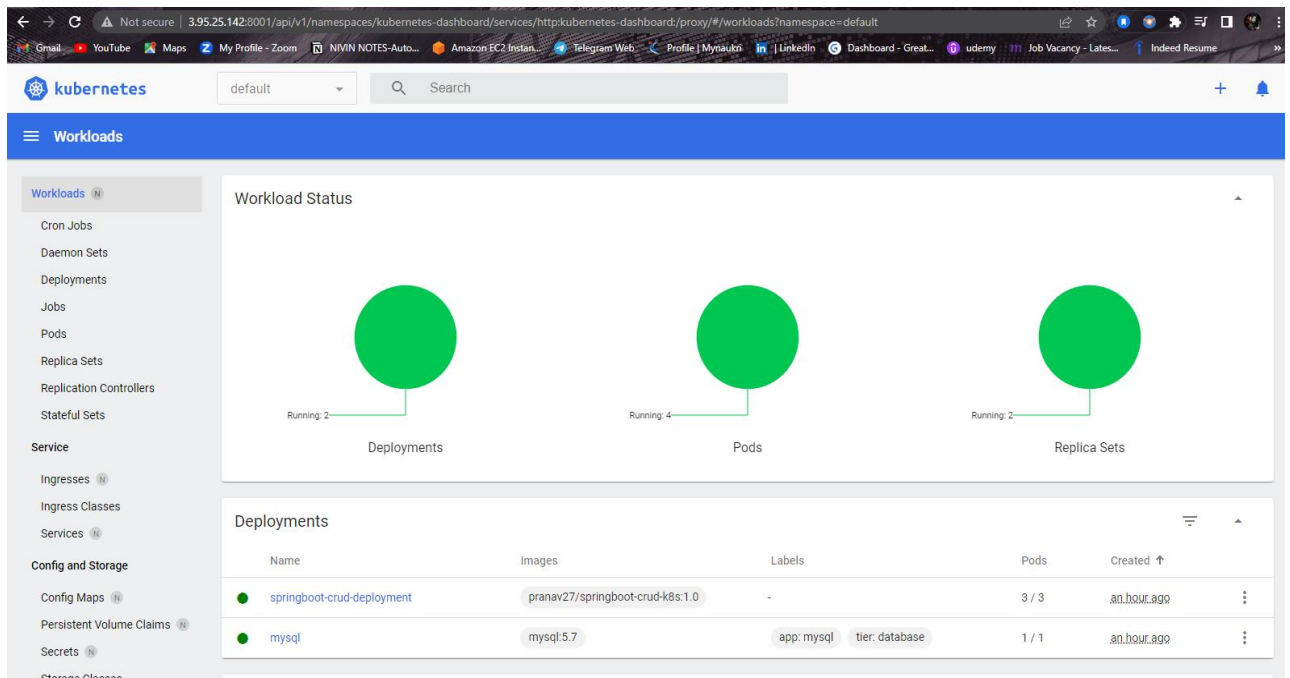


Fig.Kubernetes Dashboard

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
springboot-crud-deployment-7d9647f547-dgd7d	pranav27/springboot-crud-k8s:1.0	app: springboot-k8s-mysql, pod-template-hash: 7d9647f547	minikube	Running	0	-	-	an hour ago
springboot-crud-deployment-7d9647f547-fnfcf	pranav27/springboot-crud-k8s:1.0	app: springboot-k8s-mysql, pod-template-hash: 7d9647f547	minikube	Running	0	-	-	an hour ago
springboot-crud-deployment-7d9647f547-gblkc	pranav27/springboot-crud-k8s:1.0	app: springboot-k8s-mysql, pod-template-hash: 7d9647f547	minikube	Running	0	-	-	an hour ago
mysql-6c748b7c67-ll62c	mysql:5.7	app: mysql, pod-template-hash: 6c748b7c67, tier: database	minikube	Running	0	-	-	an hour ago

Fig.Pods in Dashboard

Name	Images	Labels	Pods	Created ↑
springboot-crud-deployment-7d9647f547	pranav27/springboot-crud-k8s:1.0	app: springboot-k8s-mysql, pod-template-hash: 7d9647f547	3 / 3	an hour ago
mysql-6c748b7c67	mysql:5.7	app: mysql, pod-template-hash: 6c748b7c67, tier: database	1 / 1	an hour ago

Fig.Replica sets in Dashboard

22.Cleaning Setup:

```
[root@ip-172-31-27-226 Springboot-k8s]# /usr/local/bin/kubectl delete deployment.apps/springboot-crud-deployment
deployment.apps "springboot-crud-deployment" deleted
[root@ip-172-31-27-226 Springboot-k8s]# /usr/local/bin/kubectl delete service/springboot-crud-svc
service "springboot-crud-svc" deleted
```

```
[root@ip-172-31-27-226 Springboot-k8s]# /usr/local/bin/kubectl get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/mysql-6c748b7c67-1l62c         1/1      Running   0           38m

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
service/kubernetes                  ClusterIP     10.96.0.1     <none>       443/TCP    46m
service/mysql                        ClusterIP     None          <none>       3306/TCP   38m

NAME                                READY    UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mysql               1/1      1             1           38m

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/mysql-6c748b7c67    1           1           1        38m
[root@ip-172-31-27-226 Springboot-k8s]# /usr/local/bin/kubectl get pods
NAME                                READY    STATUS    RESTARTS   AGE
mysql-6c748b7c67-1l62c             1/1      Running   0           38m
```

23.Terminating instance:

[Alt+S]									
N. Virginia prnav									
Instances (1) Info									
Find instance by attribute or tag (case-sensitive)									
1									
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	P
<input type="checkbox"/>	Springboot	i-0856270fe1bf198bc	Terminated	t2.medium	-	No alarms	us-east-1b	-	3...