Desafio Security Devops Kubernetes

Por: Carlos Henrique de C. Costa

1. Visão Geral

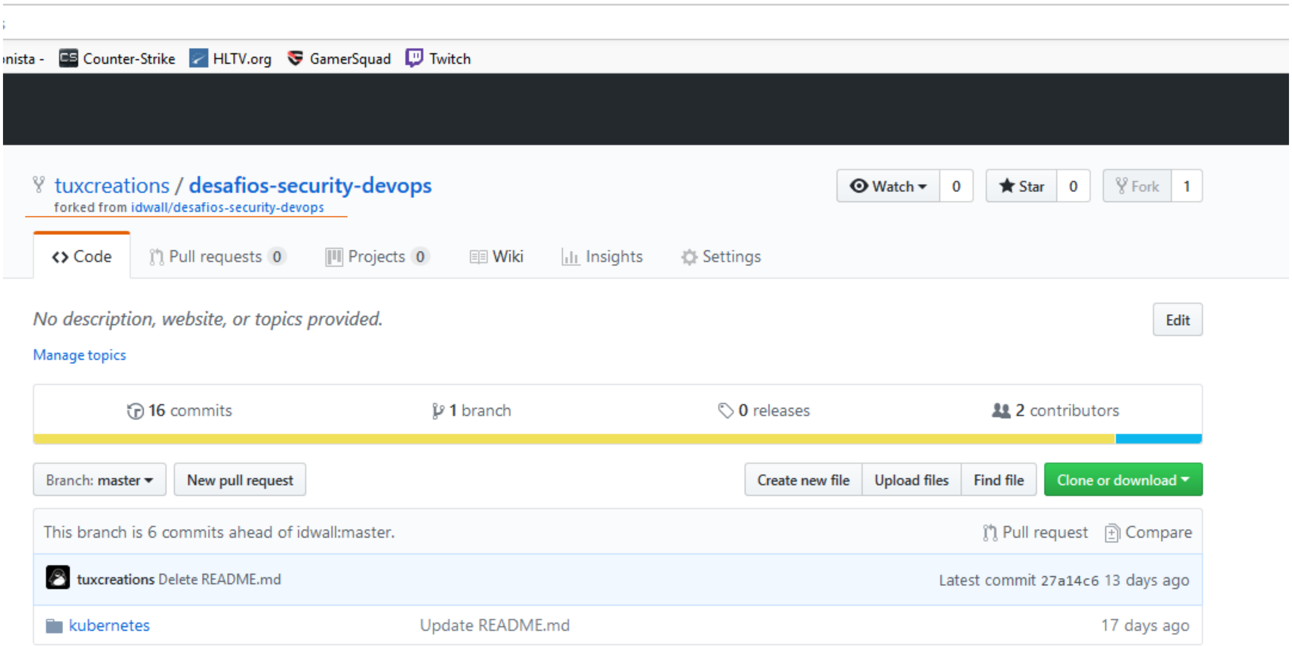
1.1 Objetivo

Este documento tem por objetivo responder ao desafio proposto pela IDWall em repositório GitHub. O desafio consiste em criar um cluster kubernetes contendo a aplicação demo e um banco de dados.

2. Resolução

2.1. Os seguinte passos foram realizados para tratativa do desafio proposto:

a) Realizado *fork* do repositório original para meu repositório pessoal no GitHub:



tuxcreations / desafios-security-devops
forked from idwall/desafios-security-devops

Watch 0 | Star 0 | Fork 1

<> Code | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

No description, website, or topics provided.

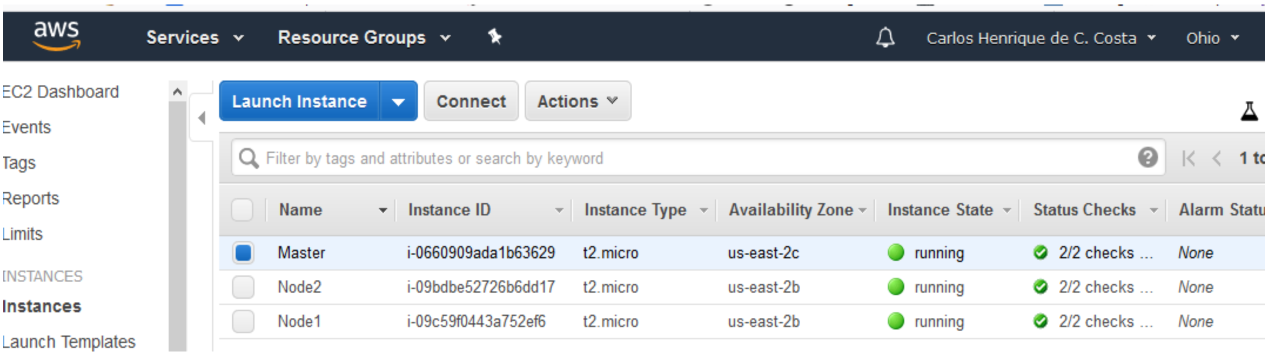
16 commits | 1 branch | 0 releases | 2 contributors

This branch is 6 commits ahead of idwall:master.

tuxcreations Delete README.md — Latest commit 27a14c6 13 days ago

kubernetes	Update README.md	17 days ago

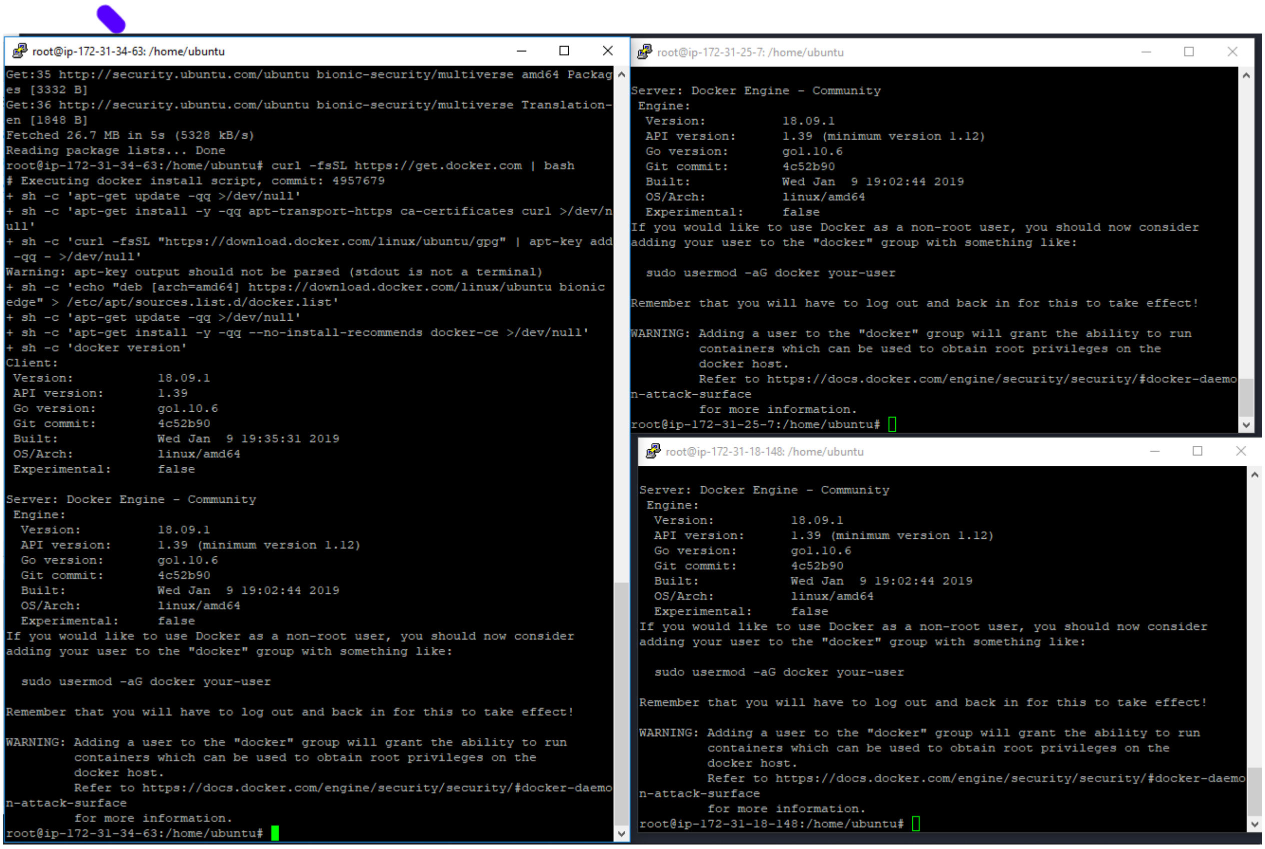
b) Criado ambiente AWS com três instancias Linux Ubuntu 18.04 LTS para testes:



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Statu
■	Master	i-0660909ada1b63629	t2.micro	us-east-2c	● running	✓ 2/2 checks ...	None
□	Node2	i-09bdbe52726b6dd17	t2.micro	us-east-2b	● running	✓ 2/2 checks ...	None
□	Node1	i-09c59f0443a752ef6	t2.micro	us-east-2b	● running	✓ 2/2 checks ...	None

2- Instalação do Docker e do Kubernetes:

a) Instalação Docker: *# curl -fsSL https://get.docker.com | bash*

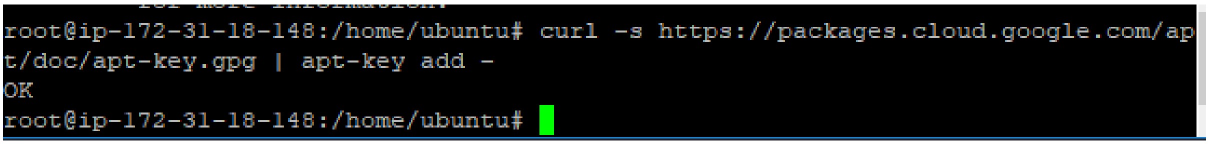

root@ip-172-31-34-63: /home/ubuntu
root@ip-172-31-25-7: /home/ubuntu
root@ip-172-31-18-148: /home/ubuntu

```
Get:35 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packag
es [3332 B]
Get:36 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-
en [1848 B]
Fetched 26.7 MB in 5s (5328 kB/s)
Reading package lists... Done
root@ip-172-31-34-63:/home/ubuntu# curl -fsSL https://get.docker.com | bash
# Executing docker install script, commit: 4957679
+ sh -c 'apt-get update -qq >/dev/null'
+ sh -c 'apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/n
ull'
+ sh -c 'curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | apt-key add
 -qq - >/dev/null'
Warning: apt-key output should not be parsed (stdout is not a terminal)
+ sh -c 'echo "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic
edge" > /etc/apt/sources.list.d/docker.list'
+ sh -c 'apt-get update -qq >/dev/null'
+ sh -c 'apt-get install -y -qq --no-install-recommends docker-ce >/dev/null'
+ sh -c 'docker version'
Client:
 Version:           18.09.1
 API version:       1.39
 Go version:        go1.10.6
 Git commit:        4c52b90
 Built:             Wed Jan  9 19:35:31 2019
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          18.09.1
  API version:      1.39 (minimum version 1.12)
  Go version:       go1.10.6
  Git commit:       4c52b90
  Built:            Wed Jan  9 19:02:44 2019
  OS/Arch:          linux/amd64
  Experimental:     false
If you would like to use Docker as a non-root user, you should now consider
adding your user to the "docker" group with something like:

  sudo usermod -aG docker your-user

Remember that you will have to log out and back in for this to take effect!

WARNING: Adding a user to the "docker" group will grant the ability to run
         containers which can be used to obtain root privileges on the
         docker host.
         Refer to https://docs.docker.com/engine/security/security/#docker-daemo
n-attack-surface
         for more information.
root@ip-172-31-34-63:/home/ubuntu#
```

```

Server: Docker Engine - Community
 Engine:
  Version:          18.09.1
  API version:      1.39 (minimum version 1.12)
  Go version:       go1.10.6
  Git commit:       4c52b90
  Built:            Wed Jan  9 19:02:44 2019
  OS/Arch:          linux/amd64
  Experimental:     false
If you would like to use Docker as a non-root user, you should now consider
adding your user to the "docker" group with something like:

  sudo usermod -aG docker your-user

Remember that you will have to log out and back in for this to take effect!

WARNING: Adding a user to the "docker" group will grant the ability to run
         containers which can be used to obtain root privileges on the
         docker host.
         Refer to https://docs.docker.com/engine/security/security/#docker-daemo
n-attack-surface
         for more information.
root@ip-172-31-25-7:/home/ubuntu#
```

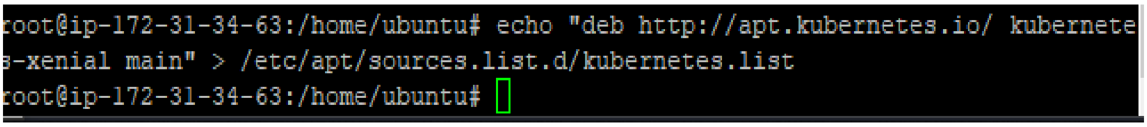
```

Server: Docker Engine - Community
 Engine:
  Version:          18.09.1
  API version:      1.39 (minimum version 1.12)
  Go version:       go1.10.6
  Git commit:       4c52b90
  Built:            Wed Jan  9 19:02:44 2019
  OS/Arch:          linux/amd64
  Experimental:     false
If you would like to use Docker as a non-root user, you should now consider
adding your user to the "docker" group with something like:

  sudo usermod -aG docker your-user

Remember that you will have to log out and back in for this to take effect!

WARNING: Adding a user to the "docker" group will grant the ability to run
         containers which can be used to obtain root privileges on the
         docker host.
         Refer to https://docs.docker.com/engine/security/security/#docker-daemo
n-attack-surface
         for more information.
root@ip-172-31-18-148:/home/ubuntu#
```

b) Instalação do repositório Kubernetes: *# curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -*



```
root@ip-172-31-18-148:/home/ubuntu# curl -s https://packages.cloud.google.com/ap
t/doc/apt-key.gpg | apt-key add -
OK
root@ip-172-31-18-148:/home/ubuntu#
```

c) Atualização do Kubernetes via repositório oficial: # **echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" > /etc/apt/sources.list.d/kubernetes.list**



```
root@ip-172-31-34-63:/home/ubuntu# echo "deb http://apt.kubernetes.io/ kubernete
s-xenial main" > /etc/apt/sources.list.d/kubernetes.list
root@ip-172-31-34-63:/home/ubuntu#
```

d) Instalação dos comandos kubectl, kubeadm e kubelet:

# apt-get update

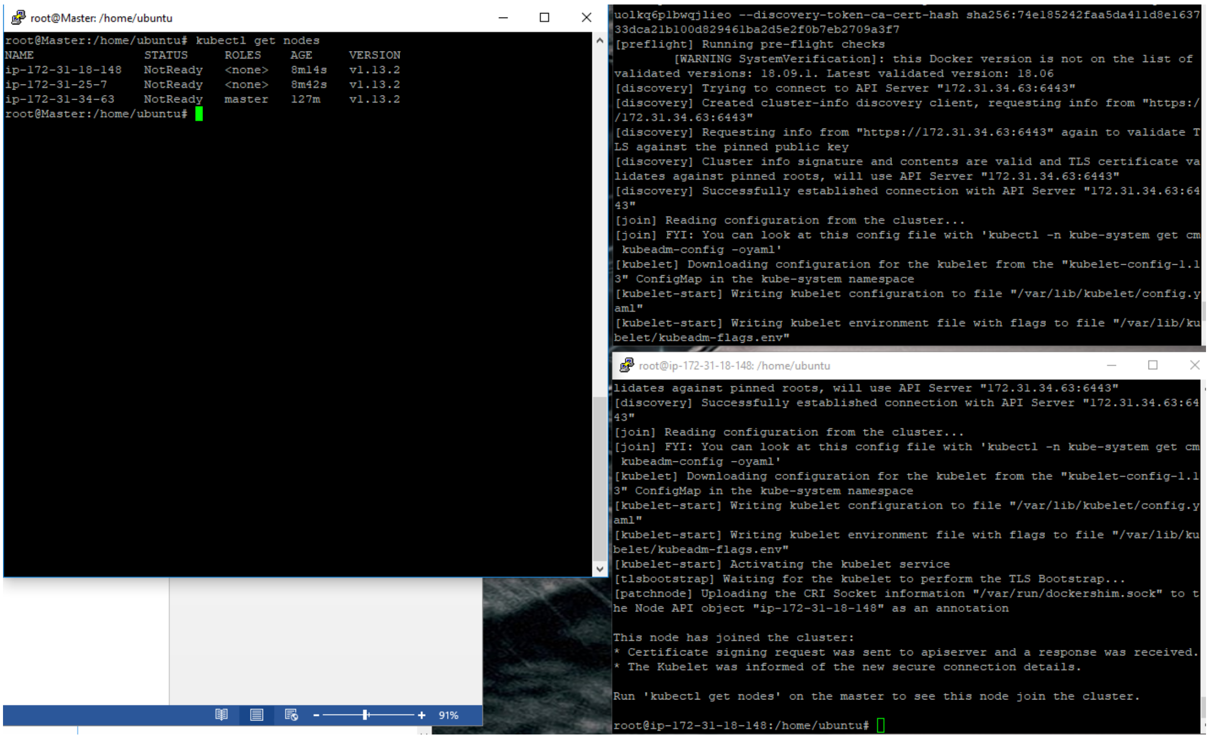
# apt-get install -y kubelet kubeadm kubectl

3- Criação do Cluster Kubernetes

```
# kubeadm init --apiserver-advertise-address $(hostname -i)

# mkdir -p $HOME/.kube

# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

# sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

4- Ingresso dos nodes ao *cluster*

```
kubeadm join 172.31.34.63:6443 --token q21xe6.uolkq6p1bwqjlieo --discovery-token-ca-cert-hash
sha256:74e185242faa5da411d8e163733dca21b100d829461ba2d5e2f0b7eb2709a3f7
```

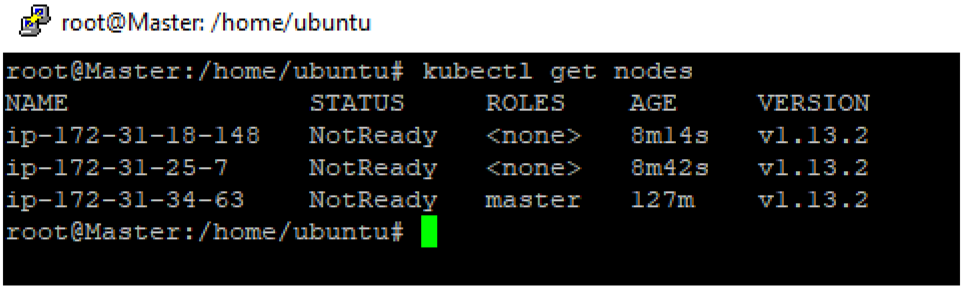

root@Master: /home/ubuntu
root@ip-172-31-18-148: /home/ubuntu
91%

```
root@Master:/home/ubuntu# kubectl get nodes
NAME               STATUS     ROLES    AGE     VERSION
ip-172-31-18-148   NotReady   <none>   8m14s   v1.13.2
ip-172-31-25-7     NotReady   <none>   8m42s   v1.13.2
ip-172-31-34-63    NotReady   master   127m    v1.13.2
root@Master:/home/ubuntu#
```

```
uolkq6p1bwqjlieo --discovery-token-ca-cert-hash sha256:74e185242faa5da411d8e1637
33dca21b100d829461ba2d5e2f0b7eb2709a3f7
[preflight] Running pre-flight checks
        [WARNING SystemVerification]: this Docker version is not on the list of
validated versions: 18.09.1. Latest validated version: 18.06
[discovery] Trying to connect to API Server "172.31.34.63:6443"
[discovery] Created cluster-info discovery client, requesting info from "https:/
/172.31.34.63:6443"
[discovery] Requesting info from "https://172.31.34.63:6443" again to validate T
LS against the pinned public key
[discovery] Cluster info signature and contents are valid and TLS certificate va
lidates against pinned roots, will use API Server "172.31.34.63:6443"
[discovery] Successfully established connection with API Server "172.31.34.63:64
43"
[join] Reading configuration from the cluster...
[join] FYI: You can look at this config file with 'kubectl -n kube-system get cm
 kubeadm-config -oyaml'
[kubelet] Downloading configuration for the kubelet from the "kubelet-config-1.1
3" ConfigMap in the kube-system namespace
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.y
aml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/ku
belet/kubeadm-flags.env"
```

```
lidates against pinned roots, will use API Server "172.31.34.63:6443"
[discovery] Successfully established connection with API Server "172.31.34.63:64
43"
[join] Reading configuration from the cluster...
[join] FYI: You can look at this config file with 'kubectl -n kube-system get cm
 kubeadm-config -oyaml'
[kubelet] Downloading configuration for the kubelet from the "kubelet-config-1.1
3" ConfigMap in the kube-system namespace
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.y
aml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/ku
belet/kubeadm-flags.env"
[kubelet-start] Activating the kubelet service
[tlsbootstrap] Waiting for the kubelet to perform the TLS Bootstrap...
[patchnode] Uploading the CRI Socket information "/var/run/dockershim.sock" to t
he Node API object "ip-172-31-18-148" as an annotation

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the master to see this node join the cluster.

root@ip-172-31-18-148:/home/ubuntu#
```

Nodes não inicializados por não ter um pod networking:

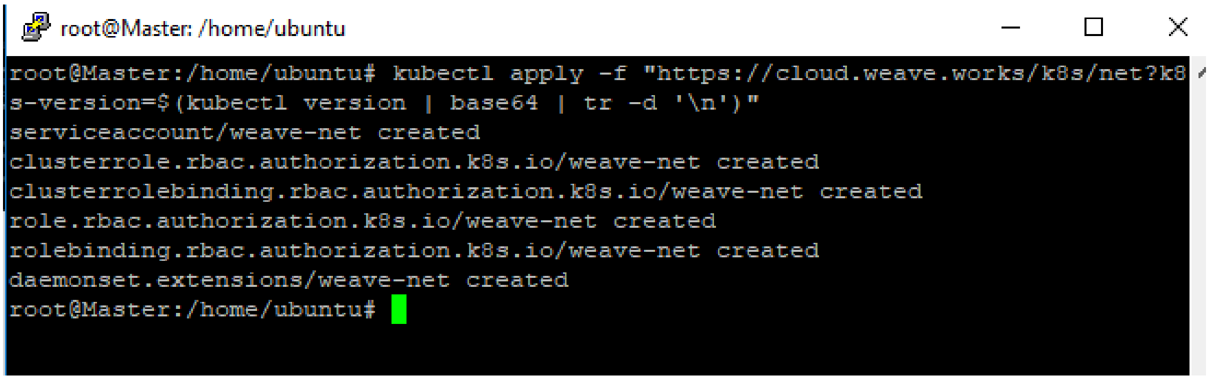


root@Master: /home/ubuntu

```
root@Master:/home/ubuntu# kubectl get nodes
NAME                 STATUS     ROLES    AGE     VERSION
ip-172-31-18-148     NotReady   <none>   8m14s   v1.13.2
ip-172-31-25-7       NotReady   <none>   8m42s   v1.13.2
ip-172-31-34-63      NotReady   master   127m    v1.13.2
root@Master:/home/ubuntu#
```

5- Instalação do pod Networking WEAVE

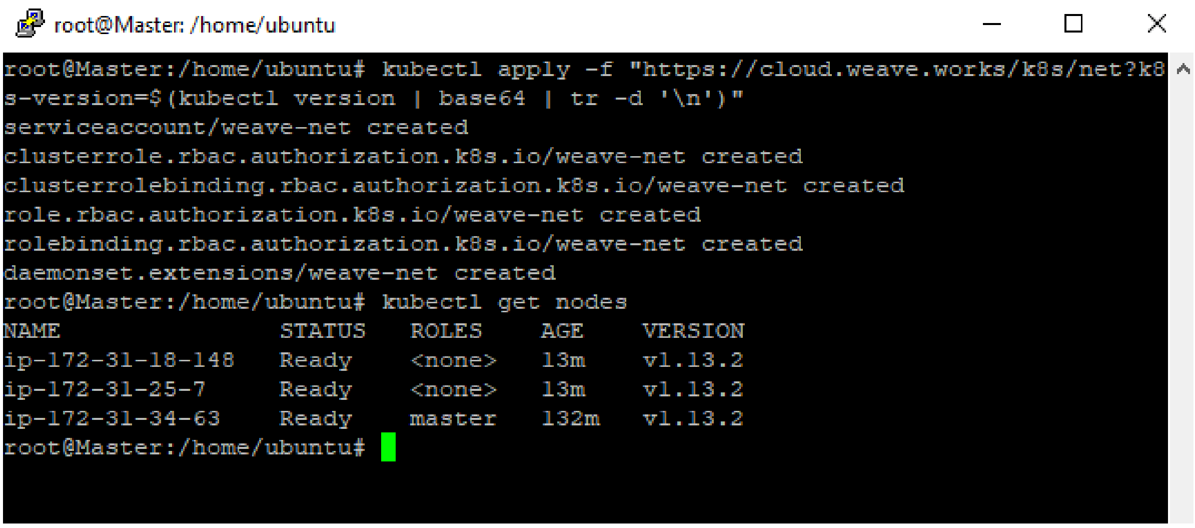
# kubectl apply -f https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')



root@Master: /home/ubuntu

```
root@Master:/home/ubuntu# kubectl apply -f "https://cloud.weave.works/k8s/net?k8
s-version=$(kubectl version | base64 | tr -d '\n')"
serviceaccount/weave-net created
clusterrole.rbac.authorization.k8s.io/weave-net created
clusterrolebinding.rbac.authorization.k8s.io/weave-net created
role.rbac.authorization.k8s.io/weave-net created
rolebinding.rbac.authorization.k8s.io/weave-net created
daemonset.extensions/weave-net created
root@Master:/home/ubuntu#
```

Nodes iniciados:



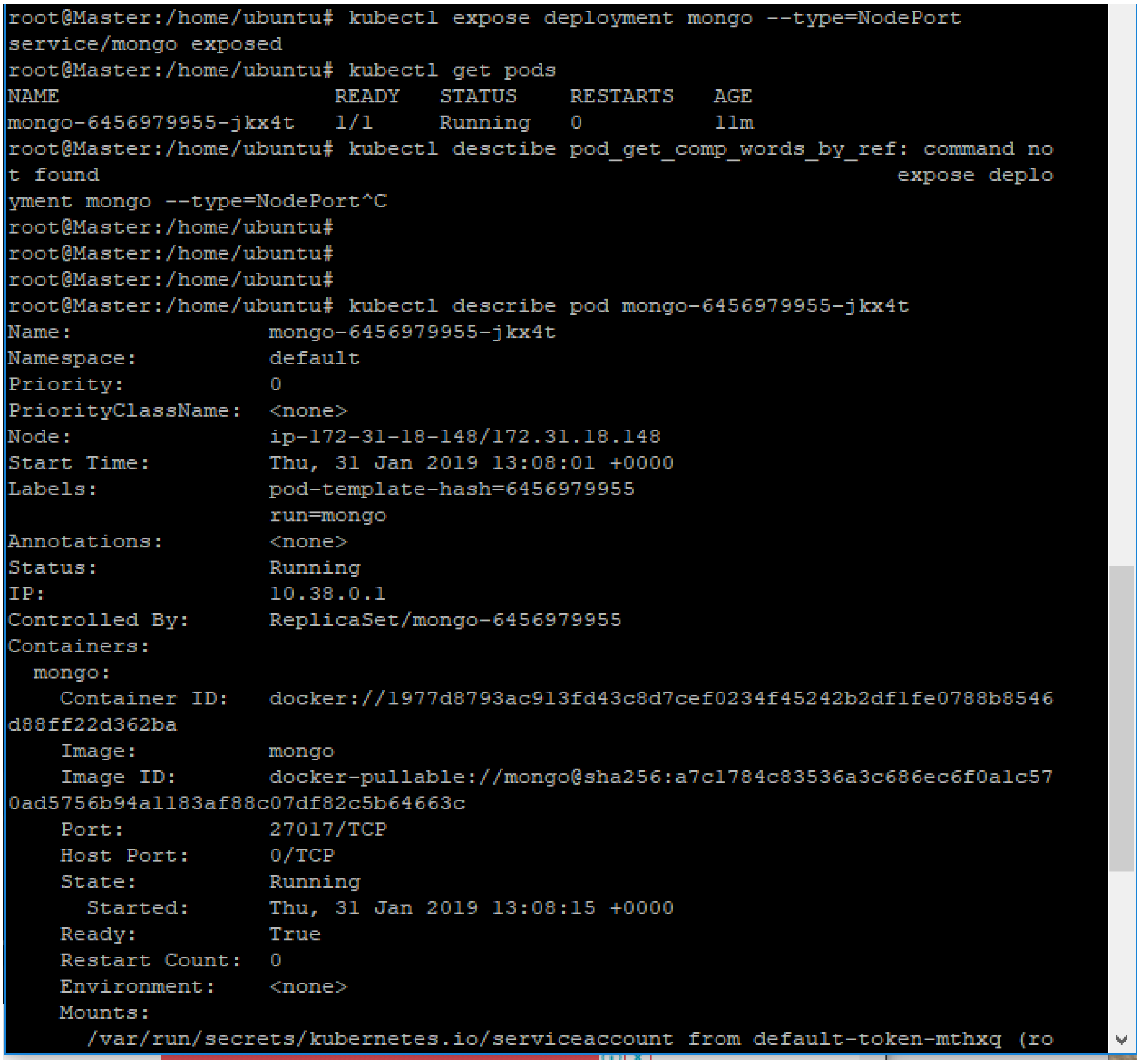
root@Master: /home/ubuntu

```
root@Master:/home/ubuntu# kubectl apply -f "https://cloud.weave.works/k8s/net?k8
s-version=$(kubectl version | base64 | tr -d '\n')"
serviceaccount/weave-net created
clusterrole.rbac.authorization.k8s.io/weave-net created
clusterrolebinding.rbac.authorization.k8s.io/weave-net created
role.rbac.authorization.k8s.io/weave-net created
rolebinding.rbac.authorization.k8s.io/weave-net created
daemonset.extensions/weave-net created
root@Master:/home/ubuntu# kubectl get nodes
NAME               STATUS   ROLES    AGE    VERSION
ip-172-31-18-148   Ready    <none>   13m    v1.13.2
ip-172-31-25-7     Ready    <none>   13m    v1.13.2
ip-172-31-34-63    Ready    master   132m   v1.13.2
root@Master:/home/ubuntu#
```

6- Instalação da imagem do banco de dados mongo

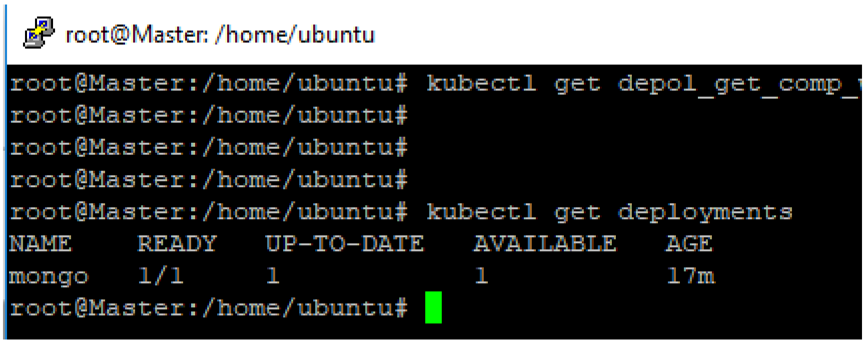
kubectl run mongo --image=mongo --port 27017

a) Expose do mongo para acesso da aplicação:
   # kubectl expose deployment mongo --type=NodePort



```
root@Master:/home/ubuntu# kubectl expose deployment mongo --type=NodePort
service/mongo exposed
root@Master:/home/ubuntu# kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
mongo-6456979955-jkx4t   1/1     Running   0          11m
root@Master:/home/ubuntu# kubectl desctibe pod_get_comp_words_by_ref: command no
t found
                                                                    expose deplo
yment mongo --type=NodePort^C
root@Master:/home/ubuntu#
root@Master:/home/ubuntu#
root@Master:/home/ubuntu#
root@Master:/home/ubuntu# kubectl describe pod mongo-6456979955-jkx4t
Name:               mongo-6456979955-jkx4t
Namespace:          default
Priority:           0
PriorityClassName:  <none>
Node:               ip-172-31-18-148/172.31.18.148
Start Time:         Thu, 31 Jan 2019 13:08:01 +0000
Labels:             pod-template-hash=6456979955
                    run=mongo
Annotations:        <none>
Status:             Running
IP:                 10.38.0.1
Controlled By:      ReplicaSet/mongo-6456979955
Containers:
  mongo:
    Container ID:   docker://1977d8793ac913fd43c8d7cef0234f45242b2df1fe0788b8546
d88ff22d362ba
    Image:          mongo
    Image ID:       docker-pullable://mongo@sha256:a7c1784c83536a3c686ec6f0a1c57
0ad5756b94a1183af88c07df82c5b64663c
    Port:           27017/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Thu, 31 Jan 2019 13:08:15 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-mthxq (ro
```

Deployment mong rodando:



root@Master: /home/ubuntu

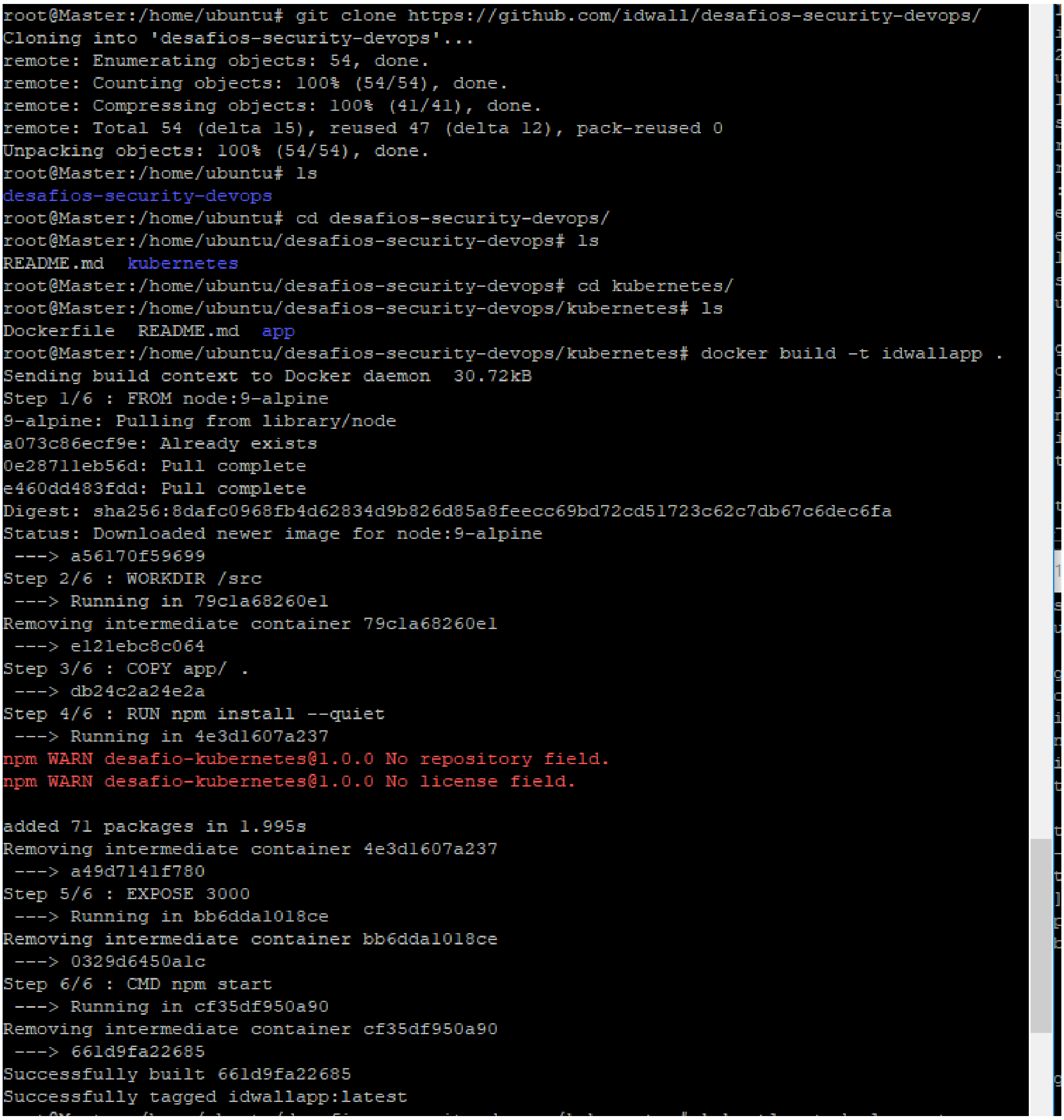
```
root@Master:/home/ubuntu# kubectl get depol_get_comp_
root@Master:/home/ubuntu#
root@Master:/home/ubuntu#
root@Master:/home/ubuntu#
root@Master:/home/ubuntu# kubectl get deployments
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
mongo   1/1     1            1           17m
root@Master:/home/ubuntu# █
```

7- Criando imagem da aplicação:

a)
```
git clone https://github.com/idwall/desafios-security-devops/
Cloning into 'desafios-security-devops'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 54 (delta 15), reused 47 (delta 12), pack-reused 0
Unpacking objects: 100% (54/54), done.
```

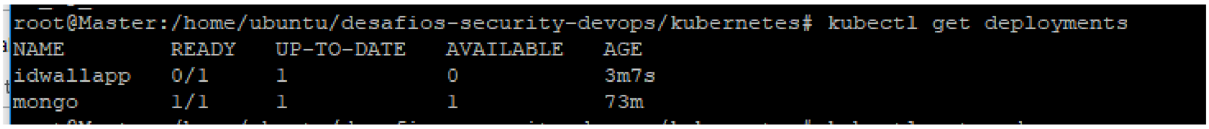
b)
```
# docker build -t idwallapp .
Sending build context to Docker daemon  30.72kB
Step 1/6 : FROM node:9-alpine
9-alpine: Pulling from library/node
a073c86ecf9e: Already exists
0e28711eb56d: Pull complete
e460dd483fdd: Pull complete
Digest:
sha256:8dafc0968fb4d62834d9b826d85a8feecc69bd72cd51723c62c7db67c6dec6fa
Status: Downloaded newer image for node:9-alpine
 ---> a56170f59699
Step 2/6 : WORKDIR /src
 ---> Running in 79c1a68260e1
Removing intermediate container 79c1a68260e1
 ---> e121ebc8c064
Step 3/6 : COPY app/ .
 ---> db24c2a24e2a
Step 4/6 : RUN npm install --quiet
 ---> Running in 4e3d1607a237
npm WARN desafio-kubernetes@1.0.0 No repository field.
npm WARN desafio-kubernetes@1.0.0 No license field.

added 71 packages in 1.995s
Removing intermediate container 4e3d1607a237
 ---> a49d7141f780
Step 5/6 : EXPOSE 3000
 ---> Running in bb6dda1018ce
Removing intermediate container bb6dda1018ce
 ---> 0329d6450a1c
Step 6/6 : CMD npm start
 ---> Running in cf35df950a90
Removing intermediate container cf35df950a90
 ---> 661d9fa22685
Successfully built 661d9fa22685
```

Successfully tagged idwallapp:latest



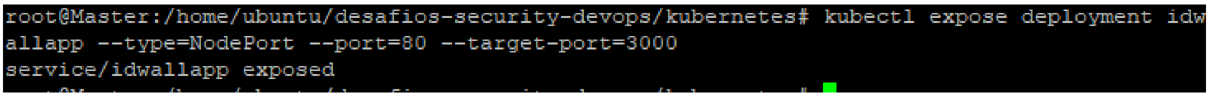
```
root@Master:/home/ubuntu# git clone https://github.com/idwall/desafios-security-devops/
Cloning into 'desafios-security-devops'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 54 (delta 15), reused 47 (delta 12), pack-reused 0
Unpacking objects: 100% (54/54), done.
root@Master:/home/ubuntu# ls
desafios-security-devops
root@Master:/home/ubuntu# cd desafios-security-devops/
root@Master:/home/ubuntu/desafios-security-devops# ls
README.md  kubernetes
root@Master:/home/ubuntu/desafios-security-devops# cd kubernetes/
root@Master:/home/ubuntu/desafios-security-devops/kubernetes# ls
Dockerfile  README.md  app
root@Master:/home/ubuntu/desafios-security-devops/kubernetes# docker build -t idwallapp .
Sending build context to Docker daemon  30.72kB
Step 1/6 : FROM node:9-alpine
9-alpine: Pulling from library/node
a073c86ecf9e: Already exists
0e28711eb56d: Pull complete
e460dd483fdd: Pull complete
Digest: sha256:8dafc0968fb4d62834d9b826d85a8feecc69bd72cd51723c62c7db67c6dec6fa
Status: Downloaded newer image for node:9-alpine
 ---> a56170f59699
Step 2/6 : WORKDIR /src
 ---> Running in 79c1a68260e1
Removing intermediate container 79c1a68260e1
 ---> e121ebc8c064
Step 3/6 : COPY app/ .
 ---> db24c2a24e2a
Step 4/6 : RUN npm install --quiet
 ---> Running in 4e3d1607a237
npm WARN desafio-kubernetes@1.0.0 No repository field.
npm WARN desafio-kubernetes@1.0.0 No license field.

added 71 packages in 1.995s
Removing intermediate container 4e3d1607a237
 ---> a49d7141f780
Step 5/6 : EXPOSE 3000
 ---> Running in bb6dda1018ce
Removing intermediate container bb6dda1018ce
 ---> 0329d6450a1c
Step 6/6 : CMD npm start
 ---> Running in cf35df950a90
Removing intermediate container cf35df950a90
 ---> 661d9fa22685
Successfully built 661d9fa22685
Successfully tagged idwallapp:latest
```

8- Criação do deployments da aplicação:



```
root@Master:/home/ubuntu/desafios-security-devops/kubernetes# kubectl get deployments
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
idwallapp   0/1     1            0           3m7s
mongo       1/1     1            1           73m
```

9- Exposição da aplicação:



```
root@Master:/home/ubuntu/desafios-security-devops/kubernetes# kubectl expose deployment idw
allapp --type=NodePort --port=80 --target-port=3000
service/idwallapp exposed
```

3. Dificuldades de resolução

Por dificuldades técnicas, não foi possível atender aos requisitos abaixo:

- Criar os manifestos de recursos kubernetes para rodar a aplicação (*services, ingresses, configmap* e qualquer outro que você considere necessário)
- Criar um *script* para a execução do *deploy* da aplicação e banco em uma única execução.
- Melhorias no Dockerfile da aplicação Web
- Utilização de *health check* na aplicação
- Utilizar algum gerenciador de Cache, como Redis, por exemplo
- Utilizar algum agregador de logs, como o Loggly, por exemplo
- Relatório de segurança da aplicação.