# Assignment 3

Sunday, October 24, 2021 8:32 PM

The first step I took in this assignment to find the secrets is to grep for the output with

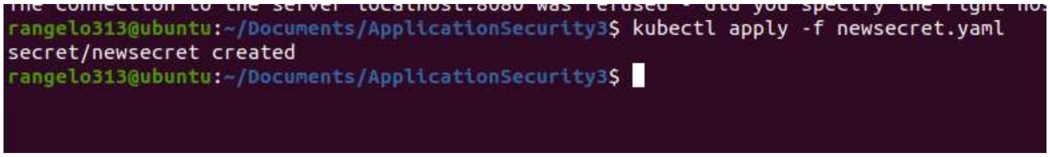
Grep -I "secret" in each subdirectory

By doing this I found the following results:

```
GiftcardSite/k8/django-deploy.yaml:37:            secretKeyRef:
GiftcardSite/k8/django-deploy.yaml:38:              name: admin-login-secrets
GiftcardSite/k8/django-deploy.yaml:43:            secretKeyRef:
GiftcardSite/k8/django-deploy.yaml:44:              name: admin-login-secrets
GiftcardSite/k8/django-admin-pass-secret.yaml:2:kind: Secret
GiftcardSite/k8/django-admin-pass-secret.yaml:4:   name: admin-login-secrets
GiftcardSite/GiftcardSite/settings.py:23:# SECURITY WARNING: keep the secret key used in production
secret!
GiftcardSite/GiftcardSite/settings.py:24:SECRET_KEY =
'kmgysa#fz+9(z1*=c0ydrjizk*7sthm2ga1z4=^61$cxcq8b$l'
    env:
        - name: MYSQL_ROOT_PASSWORD
          value: thisisatestthing.
```

These files stored sensitive data so to go about correcting this data, I created my own yaml file named newsecret that contained appropriate base64 encoding for some of the secrets and applied an environmental variable with export SECRET_KEY = <value>

After this, I went into settings.py and changed my secret_key hardcoded value to os.environment.get(SECRET_KEY) to use what I had in local storage, and this ensured the code could successfully run without exposing the secret key to the public



```
The connection to the server localhost:8080 was refused - did you specify the right hos
rangelo313@ubuntu:~/Documents/ApplicationSecurity3$ kubectl apply -f newsecret.yaml
secret/newsecret created
rangelo313@ubuntu:~/Documents/ApplicationSecurity3$ 
```

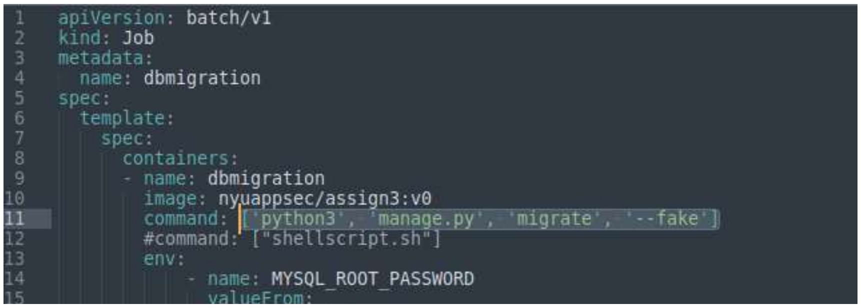
Newsecret.yaml is exported and is in the home directory to be viewed if necessary.

Part 2

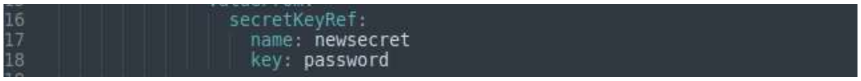
Looking at the instructions it looks like we are needed to do a migration with kubernetes jobs. To do this we must create a yaml file to integrate using django's migration functionality. In order to successfully do this, we must assign execute permissions and enter the following:
Kubectl apply –f integration.yaml
Python manage.py migrations will seem like django can do this job for us.
The setup.sql script seems to be performing the database migrations from the models file and seeds the database to populate it with data.

I had to enter –fake to avoid a LegacySite Products `already exists error`
`UPDATE: this has been fixed, --fake is not necessary after doing the following:`

modify `db/Dockerfile` to remove the lines that performs the migrations and database seeding similtaneously. This requires use to comment/remove lines from the Dockerfile.

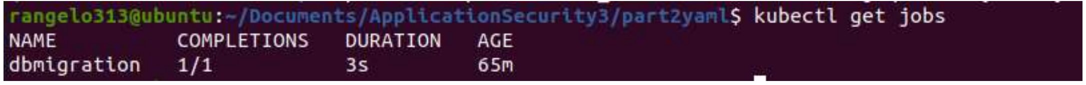


```
apiVersion: batch/v1
kind: Job
metadata:
  name: dbmigration
spec:
  template:
    spec:
      containers:
      - name: dbmigration
        image: nyuappsec/assign3:v0
        command: ['python3', 'manage.py', 'migrate', '--fake']
        #command: ["shellscript.sh"]
        env:
            - name: MYSQL_ROOT_PASSWORD
              valueFrom:
```

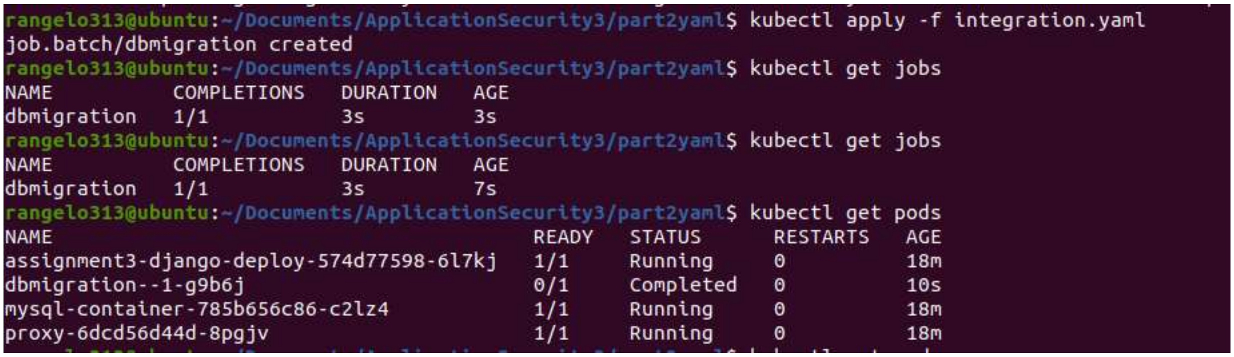


```
            secretKeyRef:
              name: newsecret
              key: password
```

Then view successful dbmigration job



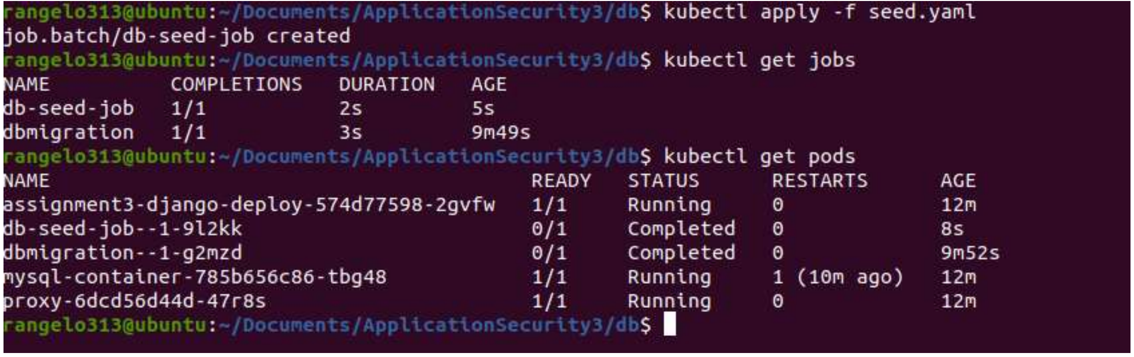
```
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part2yaml$ kubectl get jobs
NAME          COMPLETIONS   DURATION   AGE
dbmigration   1/1           3s         65m
```



```
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part2yaml$ kubectl apply -f integration.yaml
job.batch/dbmigration created
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part2yaml$ kubectl get jobs
NAME          COMPLETIONS   DURATION   AGE
dbmigration   1/1           3s         3s
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part2yaml$ kubectl get jobs
NAME          COMPLETIONS   DURATION   AGE
dbmigration   1/1           3s         7s
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part2yaml$ kubectl get pods
NAME                                       READY   STATUS      RESTARTS   AGE
assignment3-django-deploy-574d77598-6l7kj  1/1     Running     0          18m
dbmigration--1-g9b6j                       0/1     Completed   0          10s
mysql-container-785b656c86-c2lz4           1/1     Running     0          18m
proxy-6dcd56d44d-8pgjv                     1/1     Running     0          18m
```

As a result, an administrator of this webapp can execute migrations without having to rebuild anything.

Seeding a database is also a feature that Django.yaml file under the db directory.
has built in, through the manage.py loaddata commands in order to provide initial data to a datebase. As a result, I created a seed.yaml file to seed the database and applied it to my kubernetes instance with kubectl apply –f seed.yaml and exported the file for grading purposes. This created a db-seed-job.



```
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/db$ kubectl apply -f seed.yaml
job.batch/db-seed-job created
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/db$ kubectl get jobs
NAME          COMPLETIONS   DURATION   AGE
db-seed-job   1/1           2s         5s
dbmigration   1/1           3s         9m49s
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/db$ kubectl get pods
NAME                                       READY   STATUS      RESTARTS      AGE
assignment3-django-deploy-574d77598-2gvfw  1/1     Running     0             12m
db-seed-job--1-9l2kk                       0/1     Completed   0             8s
dbmigration--1-g2mzd                       0/1     Completed   0             9m52s
mysql-container-785b656c86-tbg48           1/1     Running     1 (10m ago)   12m
proxy-6dcd56d44d-47r8s                     1/1     Running     0             12m
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/db$ 
```

To verify these changes log into mysql and view with the following:
mysql> SHOW TABLES;

SOURCES: https://stackoverflow.com/questions/60061241/commands-passed-to-a-kubernetes-job-and-pod

**Part 3**
For this part I was viewing views.py for potential flaws. From the looks of it- Prometheus was recording a secret password into the counter during a POST request request method.
graphs[pword].inc
In other instances it seems to be mapping to other keys as well that is unsafe. Views.py modified is in the part3 directory.

To fix this I removed out several lines in views.py that I deemed to be vulnerable.

```
graphs['r_counter'] = Counter('python_request_r_posts', 'The total number'\
 + ' of register posts.')
graphs['l_counter'] = Counter('python_request_l_posts', 'The total number'\
 + ' of login posts.')
graphs['b_counter'] = Counter('python_request_b_posts', 'The total number'\
 + ' of card buy posts.')
graphs['g_counter'] = Counter('python_request_g_posts', 'The total number'\
 + ' of card gift posts.')
graphs['u_counter'] = Counter('python_request_u_posts', 'The total number'\
 + ' of card use posts.')
```

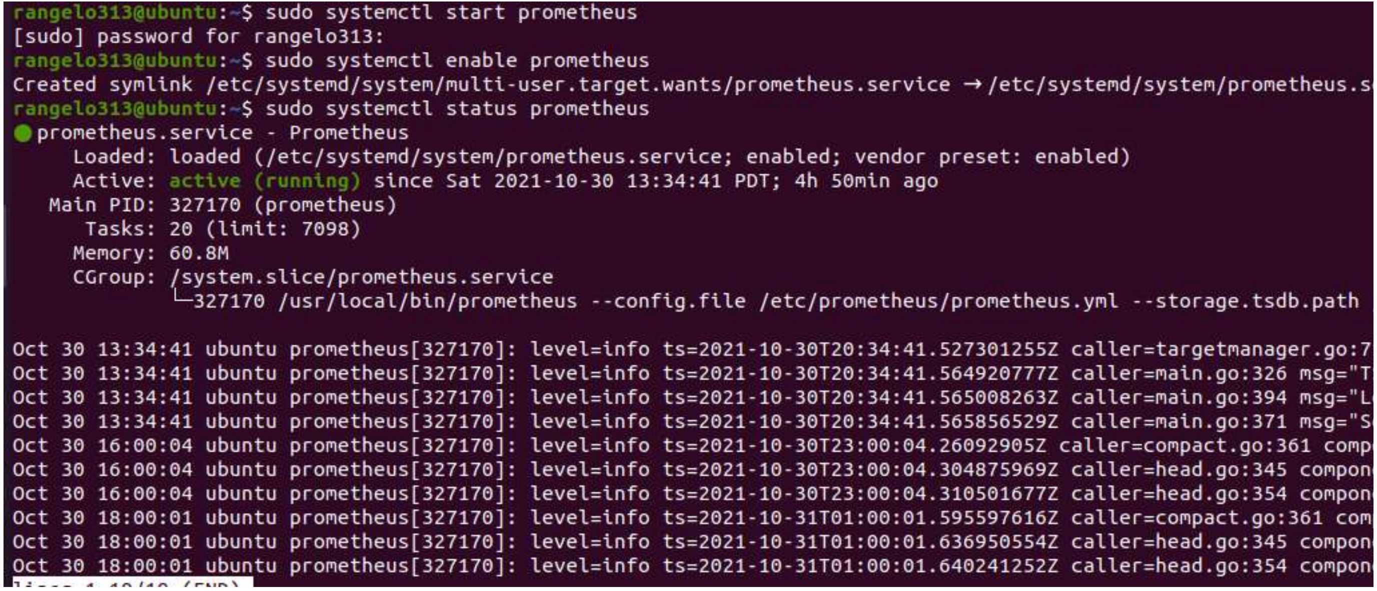
I then added a line of my own to account for 404 errors:

```
graphs['error_counter'].inc() #rc4544
```

I then installed helm then prometheus as a service and ran it as follows:

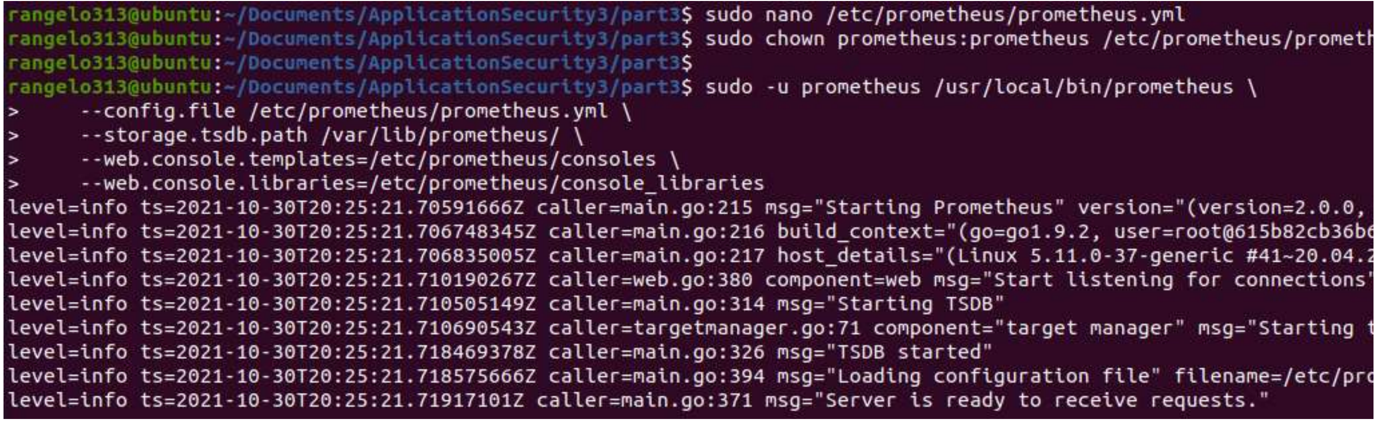
```
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3
chmod 700 get_helm.sh
./get_helm.sh


helm repo add prometheus-community https://prometheus-community.github.io/helm-charts

helm install prometheus prometheus-community/prometheus


kubectl expose service prometheus-server --type=NodePort --target-port=9090 --name=prometheus-server-np

minikube service prometheus-server-np

Kubectl get configmap
kubectl edit configmap prometheus-server
export KUBE_EDITOR="nano"

kubectl get configmap prometheus-server -o yaml
```

https://www.fosstechnix.com/install-prometheus-and-grafana-on-kubernetes-using-helm/#prerequisites
https://blog.marcnuri.com/prometheus-grafana-setup-minikube



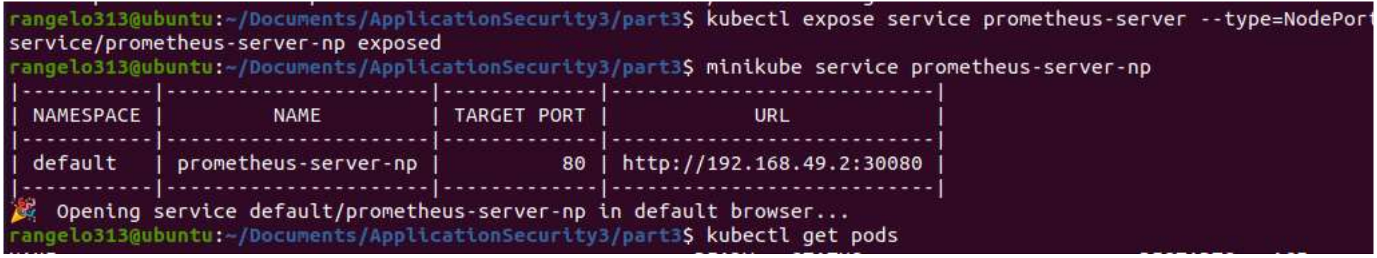
```
rangelo313@ubuntu:~$ sudo systemctl start prometheus
[sudo] password for rangelo313:
rangelo313@ubuntu:~$ sudo systemctl enable prometheus
Created symlink /etc/systemd/system/multi-user.target.wants/prometheus.service → /etc/systemd/system/prometheus.s
rangelo313@ubuntu:~$ sudo systemctl status prometheus
● prometheus.service - Prometheus
     Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2021-10-30 13:34:41 PDT; 4h 50min ago
   Main PID: 327170 (prometheus)
      Tasks: 20 (limit: 7098)
     Memory: 60.8M
     CGroup: /system.slice/prometheus.service
             └─327170 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path

Oct 30 13:34:41 ubuntu prometheus[327170]: level=info ts=2021-10-30T20:34:41.527301255Z caller=targetmanager.go:7
Oct 30 13:34:41 ubuntu prometheus[327170]: level=info ts=2021-10-30T20:34:41.564920777Z caller=main.go:326 msg="T
Oct 30 13:34:41 ubuntu prometheus[327170]: level=info ts=2021-10-30T20:34:41.565008263Z caller=main.go:394 msg="L
Oct 30 13:34:41 ubuntu prometheus[327170]: level=info ts=2021-10-30T20:34:41.565856529Z caller=main.go:371 msg="S
Oct 30 16:00:04 ubuntu prometheus[327170]: level=info ts=2021-10-30T23:00:04.26092905Z caller=compact.go:361 comp
Oct 30 16:00:04 ubuntu prometheus[327170]: level=info ts=2021-10-30T23:00:04.304875969Z caller=head.go:345 compon
Oct 30 16:00:04 ubuntu prometheus[327170]: level=info ts=2021-10-30T23:00:04.310501677Z caller=head.go:354 compon
Oct 30 18:00:01 ubuntu prometheus[327170]: level=info ts=2021-10-31T01:00:01.595597616Z caller=compact.go:361 com
Oct 30 18:00:01 ubuntu prometheus[327170]: level=info ts=2021-10-31T01:00:01.636950554Z caller=head.go:345 compon
Oct 30 18:00:01 ubuntu prometheus[327170]: level=info ts=2021-10-31T01:00:01.640241252Z caller=head.go:354 compon
```

From here, I edited the prometheus yml file and ran it as follows:



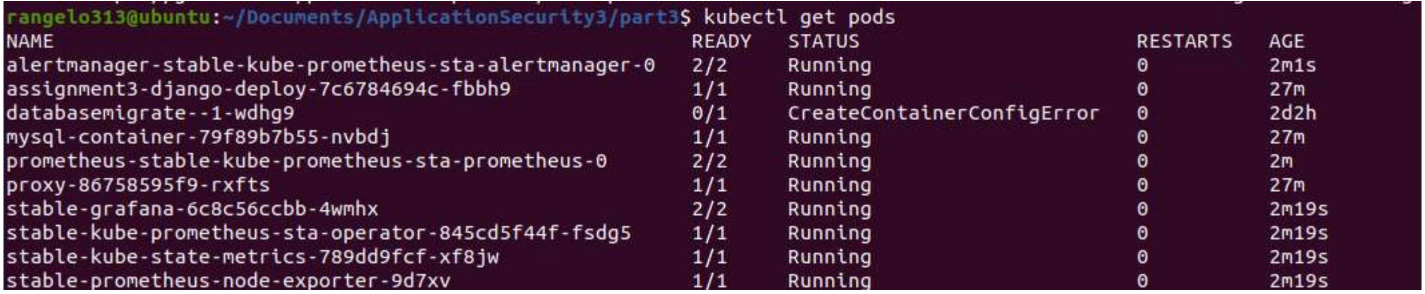
```
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part3$ sudo nano /etc/prometheus/prometheus.yml
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part3$ sudo chown prometheus:prometheus /etc/prometheus/prometh
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part3$
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part3$ sudo -u prometheus /usr/local/bin/prometheus \
>     --config.file /etc/prometheus/prometheus.yml \
>     --storage.tsdb.path /var/lib/prometheus/ \
>     --web.console.templates=/etc/prometheus/consoles \
>     --web.console.libraries=/etc/prometheus/console_libraries
level=info ts=2021-10-30T20:25:21.70591666Z caller=main.go:215 msg="Starting Prometheus" version="(version=2.0.0,
level=info ts=2021-10-30T20:25:21.706748345Z caller=main.go:216 build_context="(go=go1.9.2, user=root@615b82cb36b6
level=info ts=2021-10-30T20:25:21.706835005Z caller=main.go:217 host_details="(Linux 5.11.0-37-generic #41~20.04.2
level=info ts=2021-10-30T20:25:21.710190267Z caller=web.go:380 component=web msg="Start listening for connections"
level=info ts=2021-10-30T20:25:21.710505149Z caller=main.go:314 msg="Starting TSDB"
level=info ts=2021-10-30T20:25:21.710690543Z caller=targetmanager.go:71 component="target manager" msg="Starting t
level=info ts=2021-10-30T20:25:21.718469378Z caller=main.go:326 msg="TSDB started"
level=info ts=2021-10-30T20:25:21.718575666Z caller=main.go:394 msg="Loading configuration file" filename=/etc/pro
level=info ts=2021-10-30T20:25:21.71917101Z caller=main.go:371 msg="Server is ready to receive requests."
```

From here I ran prometheus using minikube



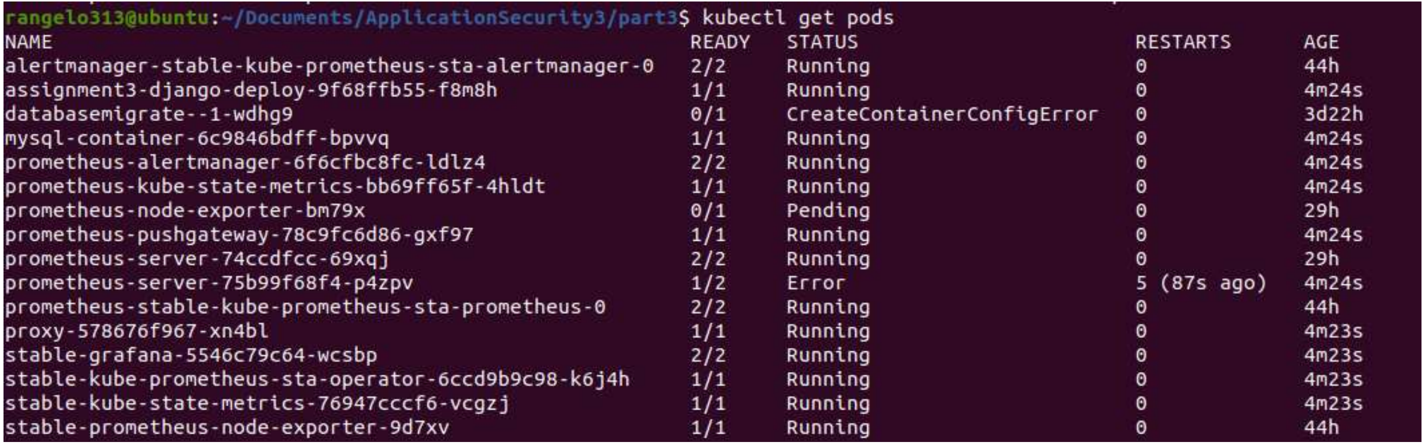
```
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part3$ kubectl expose service prometheus-server --type=NodePort
service/prometheus-server-np exposed
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part3$ minikube service prometheus-server-np
|-----------|----------------------|-------------|---------------------------|
| NAMESPACE |         NAME         | TARGET PORT |            URL            |
|-----------|----------------------|-------------|---------------------------|
| default   | prometheus-server-np |          80 | http://192.168.49.2:30080 |
|-----------|----------------------|-------------|---------------------------|
🎉  Opening service default/prometheus-server-np in default browser...
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part3$ kubectl get pods
```

I then ensured that the pods were all running and stable as indicated (note: I fixed the databasemigrate job and replaced it with dbmigrate)



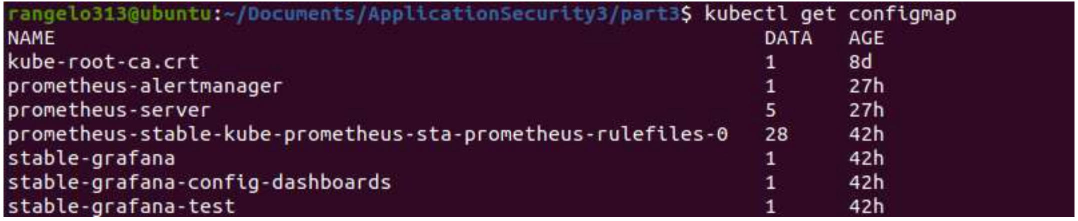
```
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part3$ kubectl get pods
NAME                                                READY   STATUS                       RESTARTS   AGE
alertmanager-stable-kube-prometheus-sta-alertmanager-0   2/2     Running                      0          2m1s
assignment3-django-deploy-7c6784694c-fbbh9          1/1     Running                      0          27m
databasemigrate--1-wdhg9                            0/1     CreateContainerConfigError   0          2d2h
mysql-container-79f89b7b55-nvbdj                    1/1     Running                      0          27m
prometheus-stable-kube-prometheus-sta-prometheus-0  2/2     Running                      0          2m
proxy-86758595f9-rxfts                              1/1     Running                      0          27m
stable-grafana-6c8c56ccbb-4wmhx                     2/2     Running                      0          2m19s
stable-kube-prometheus-sta-operator-845cd5f44f-fsdg5   1/1     Running                      0          2m19s
stable-kube-state-metrics-789dd9fcf-xf8jw           1/1     Running                      0          2m19s
stable-prometheus-node-exporter-9d7xv               1/1     Running                      0          2m19s
```

I noticed that it created another server pod so I deleted this with kubectl delete <podname>



```
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part3$ kubectl get pods
NAME                                                READY   STATUS                       RESTARTS       AGE
alertmanager-stable-kube-prometheus-sta-alertmanager-0   2/2     Running                      0              44h
assignment3-django-deploy-9f68ffb55-f8m8h           1/1     Running                      0              4m24s
databasemigrate--1-wdhg9                            0/1     CreateContainerConfigError   0              3d22h
mysql-container-6c9846bdff-bpvvq                    1/1     Running                      0              4m24s
prometheus-alertmanager-6f6cfbc8fc-ldlz4            2/2     Running                      0              4m24s
prometheus-kube-state-metrics-bb69ff65f-4hldt       1/1     Running                      0              4m24s
prometheus-node-exporter-bm79x                      0/1     Pending                      0              29h
prometheus-pushgateway-78c9fc6d86-gxf97             1/1     Running                      0              4m24s
prometheus-server-74ccdfcc-69xqj                    2/2     Running                      0              29h
prometheus-server-75b99f68f4-p4zpv                  1/2     Error                        5 (87s ago)    4m24s
prometheus-stable-kube-prometheus-sta-prometheus-0  2/2     Running                      0              44h
proxy-578676f967-xn4bl                              1/1     Running                      0              4m23s
stable-grafana-5546c79c64-wcsbp                     2/2     Running                      0              4m23s
stable-kube-prometheus-sta-operator-6ccd9b9c98-k6j4h   1/1     Running                      0              4m23s
stable-kube-state-metrics-76947cccf6-vcgzj          1/1     Running                      0              4m23s
stable-prometheus-node-exporter-9d7xv               1/1     Running                      0              44h
```

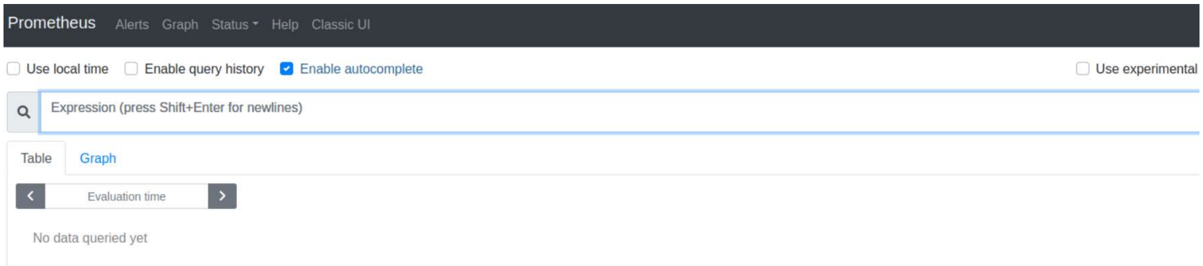
After this, I visited the proxy to ensure that Prometheus was properly configured



```
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part3$ kubectl get configmap
NAME                                                   DATA   AGE
kube-root-ca.crt                                       1      8d
prometheus-alertmanager                                1      27h
prometheus-server                                      5      27h
prometheus-stable-kube-prometheus-sta-prometheus-rulefiles-0   28     42h
stable-grafana                                         1      42h
stable-grafana-config-dashboards                       1      42h
stable-grafana-test                                    1      42h
```

As it was successfully, my last step is to configure this for the GiftCardSite to work with Prometheus. To do this I looked at configmaps

From here, we can insert the following in the : prometheusserverconfig map to map it to GiftCardSite according to the following document https://github.com/prometheus-operator/prometheus-operator/blob/master/Documentation/additional-scrape-config.md

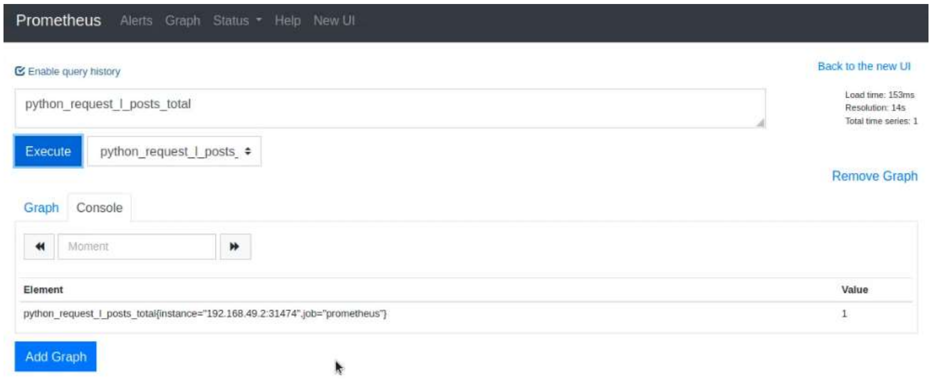


☐ Use local time ☐ Enable query history ☑ Enable autocomplete ☐ Use experimental

Expression (press Shift+Enter for newlines)

No data queried yet

I looked back at the UI just to be sure of my findings-



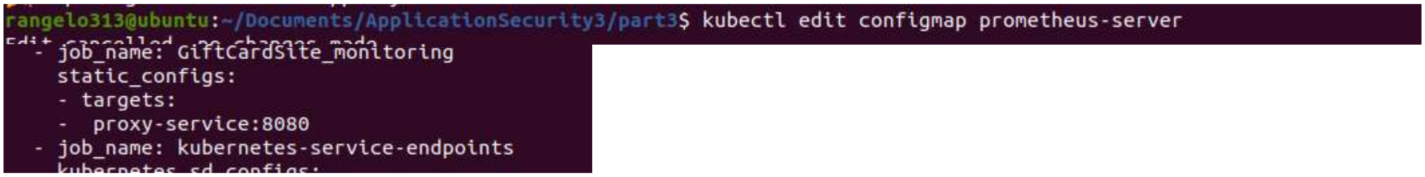
python_request_l_posts_total

Load time: 153ms
Resolution: 14s
Total time series: 1

| Element | Value |
|---|---|
| python_request_l_posts_total{instance="192.168.49.2:31474",job="prometheus"} | 1 |

I then edited the configmap prometheus-server after entering export KUBE_EDITOR="nano"

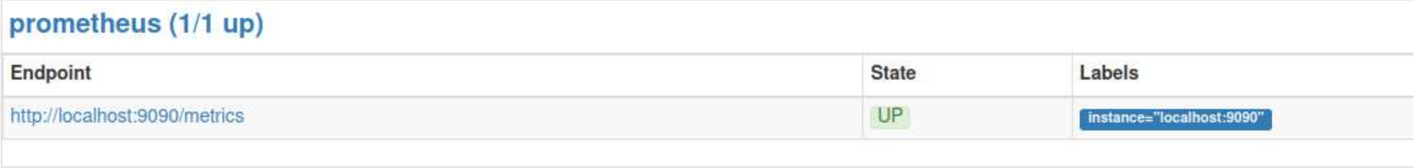
These changes can be viewed with myprometheus.yaml.



```
rangelo313@ubuntu:~/Documents/ApplicationSecurity3/part3$ kubectl edit configmap prometheus-server
  - job_name: GiftCardSite_monitoring
    static_configs:
    - targets:
    -  proxy-service:8080
  - job_name: kubernetes-service-endpoints
    kubernetes_sd_configs:
```

From here, we can port forward with the following if the 8080 port is occupied (optional):
kubectl port-forward deployment/prometheus-pushgateway 9092

From here if we click on metrics on the website, we will find the key created in view.py



## prometheus (1/1 up)

| Endpoint | State | Labels |
|---|---|---|
| http://localhost:9090/metrics | UP | instance="localhost:9090" |