من کلیه گام های بخش ها رو به صورت یکجا صحبت میکنم :
اول اینکه یک هلم چارت داریم که این وظیفه بالا اوردن دیپلویمنت ها و اینگرس و سرویس مربوط به kaas-api هست از طرفی دیگر دو تا فایل دیگر داریم که وظیفه انها این است که postgres-master و postgres-slave رو بالا بیاورد.
این فایل ها رو میتوانید kaas-api/templates مشاهده کنید.
حال میریم به سراغ api های kaas-api این سرویس شامل ۵ api است که در پایتون و توسط fastapi مدیریت میشوند.

```python
@app.post("/applications")
def add_new_application(app_data: ...):...


@app.get("/deployments/{namespace}/{app_name}")
def get_deployment_status(namespace: ..., app_name: ... = ''):...


@app.post("/postgres")
def create_postgres_service(app_data: ...):...


@app.post("/ping")
def ping():...


@app.get("/health/{app_name}")
def health(app_name: ...):...
```

این موارد به این صورت کار میکنند :

Application

این api همون کار بخش اول گام دوم رو انجام میدهد که با استفاده از کتابخانه `kubernetes.client` یک سرویس و دیپلویمنت و اینگرس برای این اپ بالا میاورد از طرف دیگر یک کرون جاب هم برای هر اپلیکیشنی بالا میاریم به خاطر گام ۵ وظیفه این کران جان این است که اگر mentoring در درخواست true بود میاد و یک کران جاب میارد بالا تا به مسیر /healthz در آن سرویس ریکوست بزنه و ریسپانس اگر ۲۰۰ بود در دیتابیس آن را اپدیت کنه
Api هایی مربوط به دیپلویمنت نیز همان کار بخش دوم گام دوم را انجام میدهد.
در مورد بعدی api create postgres service رو داریم که میاد بیس یک پوستگرس رو دریافت میکنه و یک پوستگرس رو بالا میاره.
مورد بعدی چک کردن وضعیت health یک سرویس است که مانیتور شده بوده.
جزئیات چون زیاده اینها رو توی ارائه بهش اشاره میکنیم.
و به اسکرین شات گذاشتن از ریزالت ها بسنده میکنیم :

```
cafebazaar@Mohammad-2 cc_project % helm upgrade --install kaas-api ./kaas-api

Release "kaas-api" has been upgraded. Happy Helming!
NAME: kaas-api
LAST DEPLOYED: Wed Jul  3 20:13:59 2024
NAMESPACE: default
STATUS: deployed
REVISION: 3
TEST SUITE: None
NOTES:
1. Get the application URL by running these commands:
  http://kaas-api.localhost/
```

بالا اوردن هلم چارت

```
cafebazaar@Mohammad-2 cc_project % kubectl get pods
NAME                          READY   STATUS             RESTARTS         AGE
kass-api-7ff559979b-xr84h     0/1     CrashLoopBackOff   10 (3m21s ago)   31m
my-kaas-api-postgresql-0      1/1     Running            0                31m
postgres-master-0             1/1     Running            0                31m
postgres-slave-0              1/1     Running            0                31m
test-connection               0/1     Completed          0                31m
```

وضعیت پاد ها

حالا یک سرویس خودمون اضافه میکنیم که این سرویس /healthz رو ۲۰۰ برمیگردونه
سپس وضعیت پاد ها : که کران جاب ران شده رو مشاهده میکنید.

```
cafebazaar@Mohammad-2 ~ % kubectl get cronjobs
NAME                  SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
health-health-check   */5 * * * *   False     0        2m14s           45m
cafebazaar@Mohammad-2 ~ % kubectl get pods
NAME                                  READY   STATUS      RESTARTS   AGE
health-5bcc5fc95-7xdlp                1/1     Running     0          45m
health-health-check-28666910-xtk29    0/1     Completed   0          12m
health-health-check-28666915-dtrsh    0/1     Completed   0          7m29s
health-health-check-28666920-645ct    0/1     Completed   0          2m29s
kaas-api-postgresql-0                 1/1     Running     0          76m
kass-api-5f88f6d49d-dkbsg             1/1     Running     0          48m
postgres-master-0                     1/1     Running     0          76m
postgres-slave-0                      1/1     Running     0          76m
test-connection                       0/1     Completed   0          76m
```

سپس درخواست برای گرفتن استیت سلامتی یک پاد :

```
cafebazaar@Mohammad-2 cc_project % curl -X POST "http://kass-api.localhost:8000/health/health"
{"app_name":"health","failure_count":1,"success_count":14,"last_failure":"2024-07-03T16:41:53","last_success":"2024-07-03T16:50:12","created_at":"2024-07-03T13:27:50"}%
```

که میبینم به درستی باز میگرداند

تست:

```
C:\Users\Lenovo\Desktop\cc_project>helm package kaas-api
Successfully packaged chart and saved it to: C:\Users\Lenovo\Desktop\cc_project\kaas-api-0.1.0.tgz

C:\Users\Lenovo\Desktop\cc_project>helm install kaas-api ./kaas-api-0.1.0.tgz
NAME: kaas-api
LAST DEPLOYED: Thu Jul  4 01:06:30 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
1. Get the application URL by running these commands:
  http://kaas-api.localhost/
```

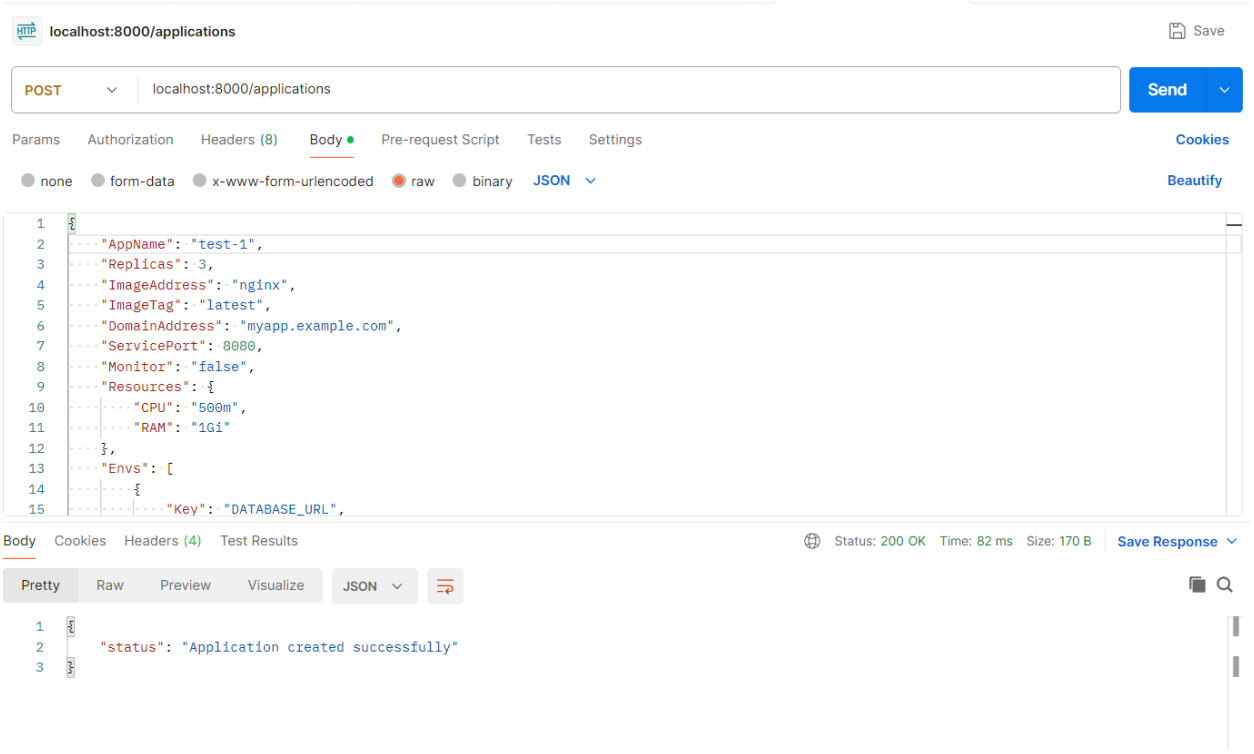
```
C:\Users\Lenovo\Desktop\cc_project>kubectl get pods
NAME                                                READY   STATUS      RESTARTS       AGE
kaas-api-59ff545b69-9bzvn                           1/1     Running     0              36s
kaas-api-postgresql-0                               1/1     Running     0              36s
nginx-ingress-ingress-nginx-controller-7b6bf7f4cd-5khn7   1/1     Running     10 (16m ago)   6d5h
postgres-master-0                                   1/1     Running     0              36s
postgres-slave-0                                    1/1     Running     0              36s
test-connection                                     0/1     Completed   0              36s
```

نمایش صحت عملکرد probe ها

```
C:\Users\Lenovo\Desktop\cc_project>kubectl logs kaas-api-59ff545b69-9bzvn
INFO:main:load incluster config passed
INFO:     Started server process [1]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO:main:startup: done
INFO:     10.244.0.1:33932 - "GET /startup HTTP/1.1" 200 OK
INFO:main:readiness: deployment status created in readiness
INFO:main:readiness: done
INFO:     10.244.0.1:33948 - "GET /ready HTTP/1.1" 200 OK
INFO:main:liveness: done
INFO:     10.244.0.1:56116 - "GET /healthz HTTP/1.1" 200 OK
INFO:main:readiness: deployment status created in readiness
INFO:main:readiness: done
INFO:     10.244.0.1:56104 - "GET /ready HTTP/1.1" 200 OK
INFO:main:liveness: done
INFO:     10.244.0.1:60662 - "GET /healthz HTTP/1.1" 200 OK
INFO:main:readiness: deployment status created in readiness
INFO:main:readiness: done
INFO:     10.244.0.1:60646 - "GET /ready HTTP/1.1" 200 OK
INFO:main:liveness: done
INFO:     10.244.0.1:40926 - "GET /healthz HTTP/1.1" 200 OK
INFO:main:readiness: deployment status created in readiness
INFO:main:readiness: done
INFO:     10.244.0.1:40918 - "GET /ready HTTP/1.1" 200 OK
```

نمایش صحت عملکرد api ساخت app

```
C:\Users\Lenovo>kubectl port-forward kaas-api-59ff545b69-9bzvn 8000
Forwarding from 127.0.0.1:8000 -> 8000
Forwarding from [::1]:8000 -> 8000
```




```
C:\Users\Lenovo\Desktop\cc_project>kubectl get pods
NAME                                                  READY   STATUS      RESTARTS       AGE
kaas-api-59ff545b69-9bzvn                             1/1     Running     0              17m
kaas-api-postgresql-0                                 1/1     Running     0              17m
nginx-ingress-ingress-nginx-controller-7b6bf7f4cd-5khn7   1/1     Running     10 (33m ago)   6d6h
postgres-master-0                                     1/1     Running     0              17m
postgres-slave-0                                      1/1     Running     0              17m
test-1-687f76bbc4-bj6wp                               1/1     Running     0              88s
test-1-687f76bbc4-c6rdf                               1/1     Running     0              88s
test-1-687f76bbc4-zww97                               1/1     Running     0              88s
test-connection                                       0/1     Completed   0              17m
```

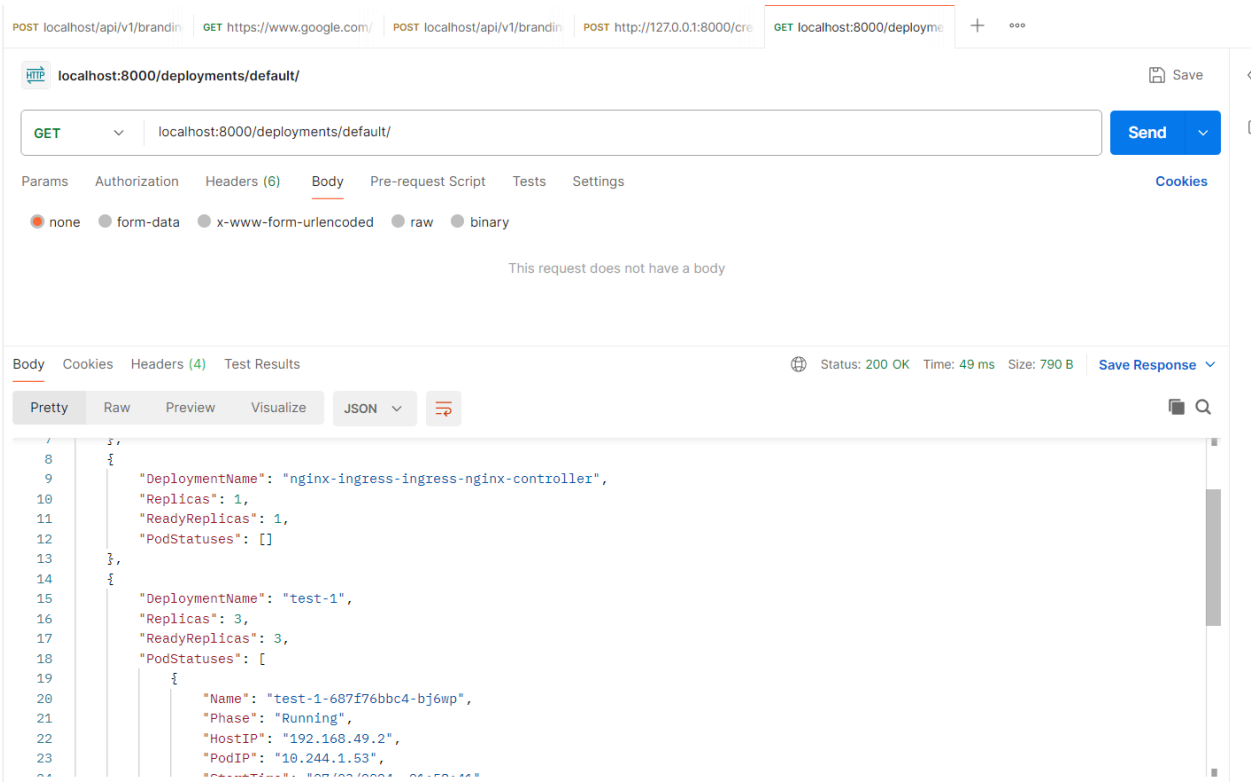
```
C:\Users\Lenovo\Desktop\cc_project>kubectl get deployments
NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE
kaas-api                              1/1     1            1           18m
nginx-ingress-ingress-nginx-controller   1/1     1            1           8d
test-1                                3/3     3            3           2m35s
```

```
C:\Users\Lenovo\Desktop\cc_project>kubectl get services
NAME                                          TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
kaas-api                                      ClusterIP      10.99.187.45     <none>        80/TCP                       19m
kaas-api-postgresql                           ClusterIP      10.97.235.195    <none>        5432/TCP                     19m
kaas-api-postgresql-hl                        ClusterIP      None             <none>        5432/TCP                     19m
kubernetes                                    ClusterIP      10.96.0.1        <none>        443/TCP                      59d
nginx-ingress-ingress-nginx-controller        LoadBalancer   10.99.164.84     127.0.0.1     80:32094/TCP,443:31347/TCP   8d
nginx-ingress-ingress-nginx-controller-admission   ClusterIP      10.106.99.235    <none>        443/TCP                      8d
postgres-master                               ClusterIP      10.103.41.50     <none>        5432/TCP                     19m
postgres-slave                                ClusterIP      10.102.162.222   <none>        5432/TCP                     19m
test-1                                        ClusterIP      10.111.85.27     <none>        8080/TCP                     3m20s
```

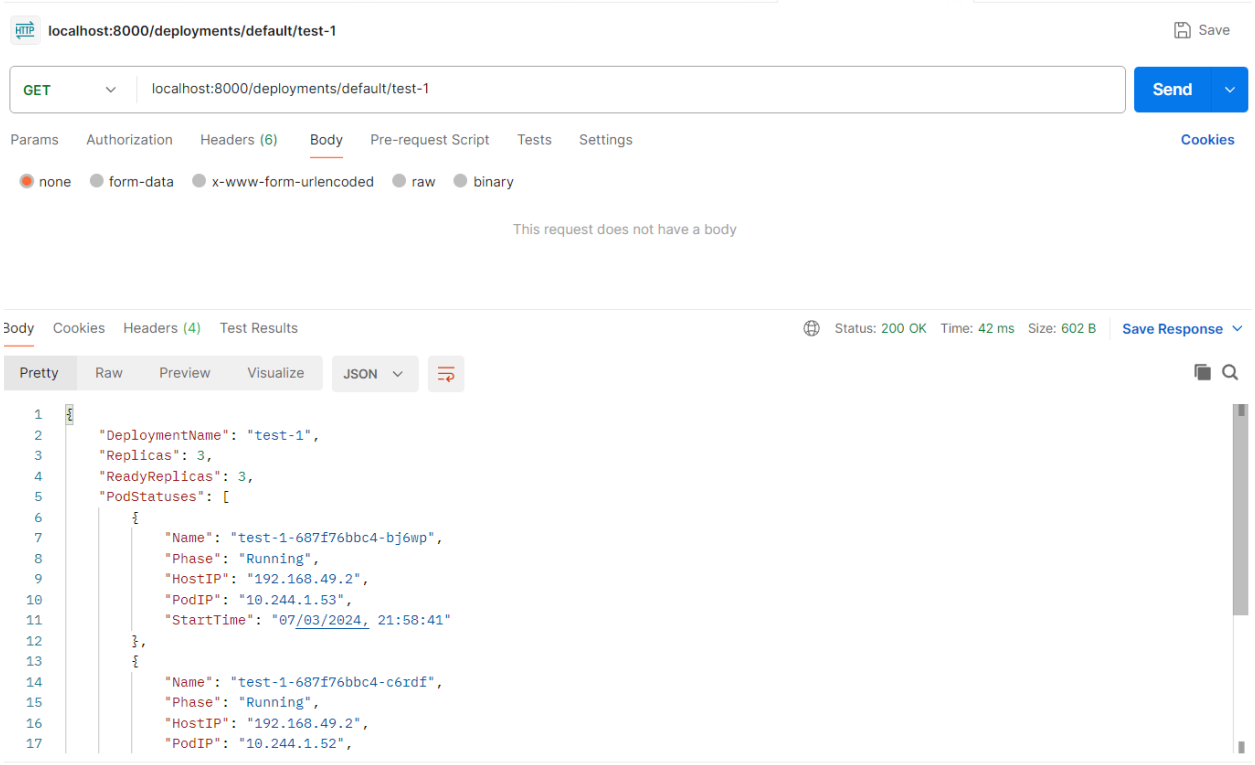
```
C:\Users\Lenovo\Desktop\cc_project>kubectl get ingress
NAME       CLASS    HOSTS                 ADDRESS     PORTS   AGE
kaas-api   nginx    kaas-api.localhost    127.0.0.1   80      20m
test-1     <none>   myapp.example.com                 80      4m2s
```

```
C:\Users\Lenovo\Desktop\cc_project>kubectl get secrets
NAME                                  TYPE                 DATA   AGE
kaas-api-postgresql                   Opaque               1      20m
my-postgres-secret                    Opaque               2      2d1h
nginx-ingress-ingress-nginx-admission Opaque               3      8d
sh.helm.release.v1.kaas-api.v1        helm.sh/release.v1   1      20m
sh.helm.release.v1.nginx-ingress.v1   helm.sh/release.v1   1      8d
test-1-secret                         Opaque               1      4m37s
```

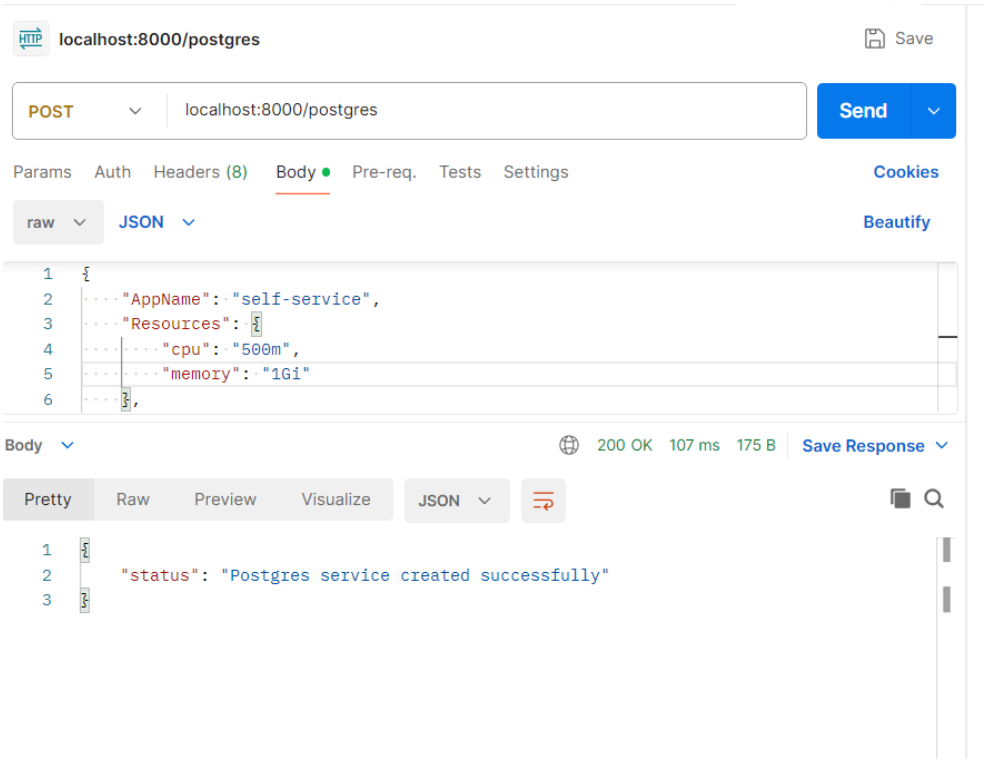
گرفتن وضعیت همه deploymentها



گرفتن وضعیت deployment یک اپ



ساخت سلف سرویس



```
C:\Users\Lenovo\Desktop\cc_project>kubectl get pods
NAME                                                  READY   STATUS      RESTARTS      AGE
kaas-api-59ff545b69-spptp                             1/1     Running     0             4m10s
kaas-api-postgresql-0                                 1/1     Running     0             4m10s
nginx-ingress-ingress-nginx-controller-7b6bf7f4cd-5khn7   1/1     Running     10 (74m ago)   6d6h
postgres-master-0                                     1/1     Running     0             4m10s
postgres-slave-0                                      1/1     Running     0             4m10s
self-service-0                                        1/1     Running     0             2m44s
test-1-687f76bbc4-bj6wp                               1/1     Running     0             42m
test-1-687f76bbc4-c6rdf                               1/1     Running     0             42m
test-1-687f76bbc4-zww97                               1/1     Running     0             42m
test-connection                                       0/1     Completed   0             4m11s
```

```
C:\Users\Lenovo\Desktop\cc_project>kubectl get secrets
NAME                                 TYPE                 DATA   AGE
kaas-api-postgresql                  Opaque               1      4m43s
my-postgres-secret                   Opaque               2      2d2h
nginx-ingress-ingress-nginx-admission   Opaque               3      8d
self-service-secret                  Opaque               2      3m16s
sh.helm.release.v1.kaas-api.v1       helm.sh/release.v1   1      4m43s
sh.helm.release.v1.nginx-ingress.v1  helm.sh/release.v1   1      8d
test-1-secret                        Opaque               1      42m
```

برای فعال سازی hpa در values.yaml آن را true میکنیم سپس هلم را upgrade میکنیم. (من برای تست threshold آن را 5 درصد گذاشته ام.)

```
C:\Users\Lenovo\Desktop\cc_project>kubectl get hpa
NAME       REFERENCE              TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
kaas-api   Deployment/kaas-api    3%/5%     1         10        1          43s
```

حالا یک اپ تستی میسازم و سپس روی آن لود می آورم تا ساخت رپلیکای خودکار را ببینیم:

In powershell :

```
for ($i = 1; $i -le 2; $i++) {
   kubectl run load-generator-$i --image=busybox -- /bin/sh -c "while true; do wget -q -O-
http://kaas-api; done"
}
```

میبینیم که با افزایش لود رپلیکای جدید اتوماتیک ساخته میشود

```
C:\Users\Lenovo\Desktop\cc_project>kubectl get pods
NAME                                                     READY   STATUS      RESTARTS       AGE
debug                                                    1/1     Running     0              6m38s
kaas-api-59ff545b69-2j8v7                                1/1     Running     0              43s
kaas-api-59ff545b69-5kkr4                                1/1     Running     0              43s
kaas-api-59ff545b69-8j8g5                                1/1     Running     0              43s
kaas-api-59ff545b69-rqg76                                1/1     Running     0              43s
kaas-api-59ff545b69-spptp                                1/1     Running     0              23m
kaas-api-postgresql-0                                    1/1     Running     0              23m
load-generator-1                                         1/1     Running     0              80s
load-generator-2                                         1/1     Running     0              80s
nginx-ingress-ingress-nginx-controller-7b6bf7f4cd-5khn7   1/1     Running     10 (94m ago)   6d7h
postgres-master-0                                        1/1     Running     0              23m
postgres-slave-0                                         1/1     Running     0              23m
self-service-0                                           1/1     Running     0              22m
test-1-687f76bbc4-bj6wp                                  1/1     Running     0              61m
test-1-687f76bbc4-c6rdf                                  1/1     Running     0              61m
test-1-687f76bbc4-zww97                                  1/1     Running     0              61m
test-2-79d74d464f-tbl6f                                  1/1     Running     0              11m
test-connection                                          0/1     Completed   0              23m
```

نصب prometheus , grafana

```
C:\Users\Lenovo\Desktop\cc_project>helm install prometheus prometheus-community/prometheus
NAME: prometheus
LAST DEPLOYED: Thu Jul  4 02:36:49 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The Prometheus server can be accessed via port 80 on the following DNS name from within your cluster:
prometheus-server.default.svc.cluster.local


Get the Prometheus server URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=prometheus,app.kubernetes.io/instance=prometheus" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace default port-forward $POD_NAME 9090


The Prometheus alertmanager can be accessed via port 9093 on the following DNS name from within your cluster:
prometheus-alertmanager.default.svc.cluster.local


Get the Alertmanager URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=alertmanager,app.kubernetes.io/instance=prometheus" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace default port-forward $POD_NAME 9093
#################################################################################
######   WARNING: Pod Security Policy has been disabled by default since    #####
######            it deprecated after k8s 1.25+. use                        #####
######            (index .Values "prometheus-node-exporter" "rbac"          #####
###### .          "pspEnabled") with (index .Values                         #####
######            "prometheus-node-exporter" "rbac" "pspAnnotations")       #####
######            in case you still need it.                                #####
#################################################################################


The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:
prometheus-prometheus-pushgateway.default.svc.cluster.local


Get the PushGateway URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace default -l "app=prometheus-pushgateway,component=pushgateway" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace default port-forward $POD_NAME 9091

For more information on running Prometheus, visit:
https://prometheus.io/
```

```
C:\Users\Lenovo\Desktop\cc_project>helm repo add grafana https://grafana.github.io/helm-charts
"grafana" already exists with the same configuration, skipping

C:\Users\Lenovo\Desktop\cc_project>helm install grafana grafana/grafana
NAME: grafana
LAST DEPLOYED: Thu Jul  4 02:35:48 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get your 'admin' user password by running:

   kubectl get secret --namespace default grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo


2. The Grafana server can be accessed via port 80 on the following DNS name from within your cluster:

   grafana.default.svc.cluster.local

   Get the Grafana URL to visit by running these commands in the same shell:
     export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=grafana,app.kubernetes.io/instance=grafana" -o jsonpath="{.items[0].metadata.name}")
     kubectl --namespace default port-forward $POD_NAME 3000

3. Login with the password from step 1 and the username: admin
#################################################################################
######   WARNING: Persistence is disabled!!! You will lose your data when   #####
######            the Grafana pod is terminated.                            #####
#################################################################################
```

```
C:\Users\Lenovo\Desktop\cc_project>kubectl get pod
NAME                                                READY   STATUS      RESTARTS        AGE
grafana-646db7ccff-l5qg6                            1/1     Running     0               65s
kaas-api-59ff545b69-rqg76                           1/1     Running     0               7m17s
kaas-api-59ff545b69-spptp                           1/1     Running     0               30m
kaas-api-postgresql-0                               1/1     Running     0               30m
nginx-ingress-ingress-nginx-controller-7b6bf7f4cd-5khn7   1/1     Running     10 (100m ago)   6d7h
postgres-master-0                                   1/1     Running     0               30m
postgres-slave-0                                    1/1     Running     0               30m
prometheus-alertmanager-0                           1/1     Running     0               4s
prometheus-kube-state-metrics-d7875bd57-gd68b       0/1     Running     0               4s
prometheus-prometheus-node-exporter-bb6s2           1/1     Running     0               4s
prometheus-prometheus-pushgateway-56985ddc76-w76tk  0/1     Running     0               4s
prometheus-server-67d6d48456-vcptj                  1/2     Running     0               4s
self-service-0                                      1/1     Running     0               28m
test-1-687f76bbc4-bj6wp                             1/1     Running     0               68m
test-1-687f76bbc4-c6rdf                             1/1     Running     0               68m
test-1-687f76bbc4-zww97                             1/1     Running     0               68m
test-2-79d74d464f-tbl6f                             1/1     Running     0               18m
test-connection                                     0/1     Completed   0               30m
```

هر دو را port forward میکنیم (9090 و 3000)

Prometheus:

kubernetes-service-endpoints (5/5 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://10.244.1.81:8080/metrics	UP	app_kubernetes_io_component="metrics" app_kubernetes_io_instance="prometheus" app_kubernetes_io_managed_by="Helm" app_kubernetes_io_name="kube-state-metrics" app_kubernetes_io_part_of="kube-state-metrics" app_kubernetes_io_version="2.12.0" helm_sh_chart="kube-state-metrics-5.20.0" instance="10.244.1.81:8080" job="kubernetes-service-endpoints" namespace="default" node="minikube" service="prometheus-kube-state-metrics"	41.22s ago	4.854ms	
http://10.244.1.58:8000/metrics	UP	app_kubernetes_io_instance="kaas-api" app_kubernetes_io_managed_by="Helm" app_kubernetes_io_name="kaas-api" helm_sh_chart="kaas-api-0.1.0" instance="10.244.1.58:8000" job="kubernetes-service-endpoints" namespace="default" node="minikube" service="kaas-api"	37.62s ago	6.085ms	
http://10.244.1.70:8000/metrics	UP	app_kubernetes_io_instance="kaas-api" app_kubernetes_io_managed_by="Helm" app_kubernetes_io_name="kaas-api" helm_sh_chart="kaas-api-0.1.0" instance="10.244.1.70:8000" job="kubernetes-service-endpoints" namespace="default" node="minikube" service="kaas-api"	30.175s ago	5.481ms	
http://10.244.1.37:9153/metrics	UP	instance="10.244.1.37:9153" job="kubernetes-service-endpoints" k8s_app="kube-dns" kubernetes_io_cluster_service="true" kubernetes_io_name="CoreDNS" namespace="kube-system" node="minikube" service="kube-dns"	42.572s ago	2.915ms	
http://192.168.49.2:9100/metrics	UP	app_kubernetes_io_component="metrics" app_kubernetes_io_instance="prometheus" app_kubernetes_io_managed_by="Helm" app_kubernetes_io_name="prometheus-node-exporter" app_kubernetes_io_part_of="prometheus-node-exporter" app_kubernetes_io_version="1.8.1"	47.672s ago	32.407ms	

Grafana

برای پیدا کردن پسورد :

kubectl get secret --namespace default grafana -o jsonpath="{.data.admin-password}"

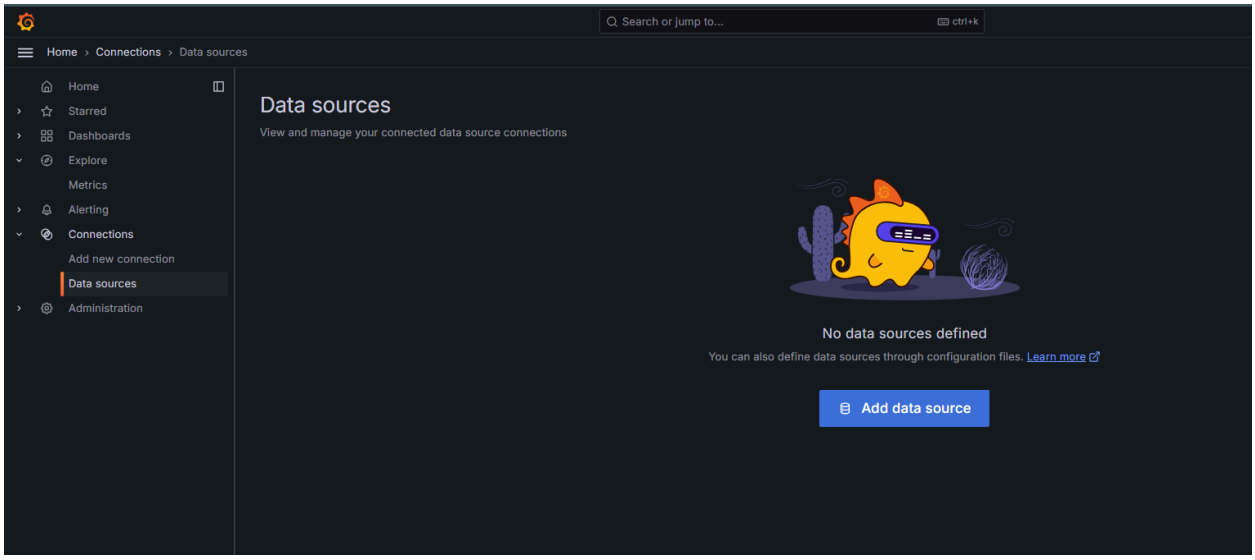
که خروجی base64 است

```
C:\Users\Lenovo\Desktop\cc_project>kubectl get secret --namespace default grafana -o jsonpath="{.data.admin-password}"
YnBSdExkeWgzYU82SUNRZzRiQ2J6TVIxR2l3aXRJRjJXTEJnaTBkSA==
```

آن را از این سایت دیکود میکنیم

https://www.base64decode.org/

حالا به گرفانا میرویم و در اینجا یک دیتاسورس از نوع prometheus میسازیم



و در قسمت url آن این را قرار میدهیم

http://prometheus-server.default.svc.cluster.local:80

سپس یک داشبورد با فایل dashboard.json میسازیم.