

به نام خدا

پروژه نهایی

رایانش ابری

گروه سوم

علیرضا زارع زین آبادی ۹۹۳۱۰۲۲

امیرفاضل کوزه گر کالجی ۹۹۳۱۰۹۹

ترم پاییز ۱۴۰۲

گام اول

پیاده سازی سرویس ها با پایتون، جنگو، سلری و پستگره انجام شده است.

سرویس api:

1. مدیریت درخواست ها:

```
1 @api_view(["POST", "GET"])
2 def submit_server_or_check_server(request):
3     if request.method == "GET":
4         server_id = request.query_params.get('id')
5         try:
6             server = Server.objects.get(id=server_id)
7             serializer = ServerSerializer(server)
8             return Response(serializer.data)
9         except Server.DoesNotExist:
10            return Response(status=404)
11     else:
12         try:
13             address = request.data['address']
14             server = Server(address=address)
15             server.save()
16             return Response(status=201)
17         except:
18            return Response(status=400)
19
20
21
22 @api_view(["GET"])
23 def check_all_servers(request):
24     servers = Server.objects.all()
25     serializer = ServerSerializer(servers, many=True)
26     return Response(serializer.data)
```

2. مدل مورد استفاده:

```
1 class Server(models.Model):
2     id = models.UUIDField(primary_key=True, default=uuid.uuid4, editable=False)
3     address = models.CharField(max_length=255, verbose_name="Server Address")
4     success_count = models.PositiveIntegerField(default=0, verbose_name="Success Count")
5     failure_count = models.PositiveIntegerField(default=0, verbose_name="Failure Count")
6     last_failure = models.DateTimeField(null=True, blank=True, verbose_name="Last Failure Time")
7     created_at = models.DateTimeField(auto_now_add=True, verbose_name="Creation Time")
8
9     class Meta:
10         verbose_name_plural = "Servers"
11         ordering = ['-created_at']
12
13     def __str__(self):
14         return f"Server {self.address} - Success: {self.success_count}, Failures: {self.failure_count}"
15
16
```

```

1  PSQL_READ = {
2      'ENGINE': 'django.db.backends.postgresql_psycopg2',
3      'NAME': os.environ.get("POSTGRES_NAME", ""),
4      'USER': os.environ.get("POSTGRES_USER", ""),
5      'PASSWORD': os.environ.get("POSTGRES_PASSWORD", ""),
6      'HOST': os.environ.get("POSTGRES_HOST_READ"),
7      'PORT': int(os.environ.get("POSTGRES_PORT", "1")),
8  }
9  PSQL_WRITE = {
10     'ENGINE': 'django.db.backends.postgresql_psycopg2',
11     'NAME': os.environ.get("POSTGRES_NAME", ""),
12     'USER': os.environ.get("POSTGRES_USER", ""),
13     'PASSWORD': os.environ.get("POSTGRES_PASSWORD", ""),
14     'HOST': os.environ.get("POSTGRES_HOST_WRITE"),
15     'PORT': int(os.environ.get("POSTGRES_PORT", "1")),
16 }
17 SQLITE = {
18     "ENGINE": "django.db.backends.sqlite3",
19     "NAME": BASE_DIR / "db.sqlite3",
20 }
21
22 DATABASES = {
23     "read": SQLITE if os.environ.get("DEBUG", "1") == "1" else PSQL_READ,
24     "default": SQLITE if os.environ.get("DEBUG", "1") == "1" else PSQL_WRITE,
25 }
26
27
28 DATABASE_ROUTERS = ['healthServiceApi.models.PrimaryReplicaRouter']
29 # Password validation

```

```

1  class PrimaryReplicaRouter:
2      def db_for_read(self, model, **hints):
3          return 'read'
4
5      def db_for_write(self, model, **hints):
6
7          return 'default'
8
9      def allow_relation(self, obj1, obj2, **hints):
10
11         return True
12
13     def allow_migrate(self, db, app_label, model_name=None, **hints):
14
15         return True

```

سرویس زمانبند:

```
1 CELERY_BROKER_URL = os.environ.get("CELERY_BROKER_URL", "redis://localhost:6379");
2 CELERY_RESULT_BACKEND = os.environ.get("CELERY_RESULT_BACKEND", "redis://localhost:6379");
3 CELERY_BEAT_SCHEDULE = {
4     'check-server-health-every-hour': {
5         'task': 'healthServiceApi.tasks.check_server_health',
6         'schedule': int(os.environ.get("TESTCYCLE", "10")),
7     },
8 }
9 }
```

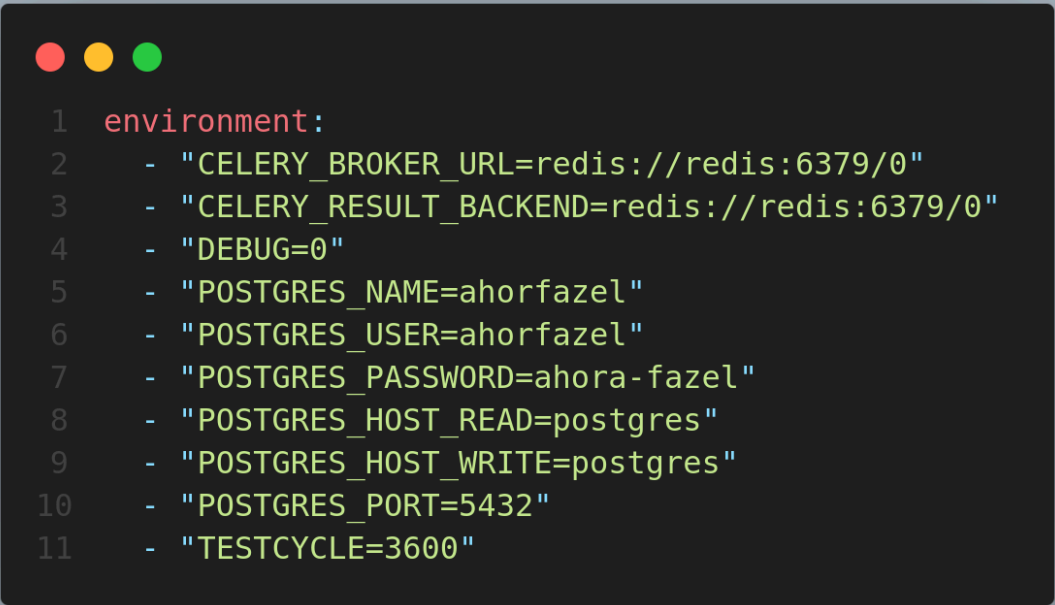
```
1 @shared_task
2 def check_server_health():
3     for server in Server.objects.all():
4         _address = server.address if "http://" in server.address or "https://" in server.address else f"http://{server.address}"
5         try:
6             response = requests.get(_address)
7             if response.status_code == 200:
8                 print(f"Server {_address} Is Alive")
9                 server.success_count += 1
10                server.save()
11            else:
12                print(f"Server {_address} Is Dead [Status = {response.status_code}]")
13                server.failure_count += 1
14                server.last_failure = now()
15                server.save()
16        except requests.RequestException as e:
17            print(f"Server {_address} Is Dead [Error = {str(e)}]")
18            server.failure_count += 1
19            server.last_failure = now()
20            server.save()
21
```

در اینجا به یک مسیج بروکر نیاز داشتیم که برای راحتی از ردیس استفاده شد.

با توجه به اینکه از پایتون برای پیاده سازی استفاده شد میتوان آنرا در یک لایه داکرایز کرد و نیاز به بیلد ندارد. همچنین این ایمیج برای کانتینر های ورکر سلری و بیت سلری نیز استفاده می شود پس کامند را در داکر کامپوز و مانیفست کوبرنیز مینویسیم.

```
1 FROM python:3.10.2-alpine
2 ENV PYTHONDONTWRITEBYTECODE 1
3 ENV PYTHONUNBUFFERED 1
4 WORKDIR /code
5 COPY ./ .
6 RUN apk add --no-cache --virtual .build-deps \
7     ca-certificates gcc postgresql-dev linux-headers musl-dev \
8     libffi-dev jpeg-dev zlib-dev
9
10
11 RUN pip install -r requirements.txt
12
13 EXPOSE 80
14
```

در داکر کامپوز نیز این متغیر های محلی برای کانفیگ پذیر بودن برنامه اسفاده شد
پورت برای جنگو چون باید در کامند آن بیاید میتواند کانفیگ پذیر باشد.



```
1 environment:
2   - "CELERY_BROKER_URL=redis://redis:6379/0"
3   - "CELERY_RESULT_BACKEND=redis://redis:6379/0"
4   - "DEBUG=0"
5   - "POSTGRES_NAME=ahorfazel"
6   - "POSTGRES_USER=ahorfazel"
7   - "POSTGRES_PASSWORD=ahora-fazel"
8   - "POSTGRES_HOST_READ=postgres"
9   - "POSTGRES_HOST_WRITE=postgres"
10  - "POSTGRES_PORT=5432"
11  - "TESTCYCLE=3600"
```

صحت اجرای داکر کامپوز:

```

File Edit View Terminal Tabs Help
~@hora@Ahora-PC: ~/docker-k8s-project/deployment
~@hora@Ahora-PC: ~/docker-k8s-project/deployment ~main~
~@hora@Ahora-PC: ~$ docker compose up -d
~$ Running S/D
Network deployment default Created 32.3s
✓ Container deployment-redis-1 Started 24.9s
✓ Container deployment-postgres-1 Started 28.1s
✓ Container deployment-django-1 Started 23.8s
✓ Container deployment-celerybeat-1 Started 23.8s
✓ Container deployment-celeryworker-2 Started 22.8s
~@hora@Ahora-PC: ~/docker-k8s-project/deployment ~main~
~@hora@Ahora-PC: ~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED        STATUS        PORTS                               NAMES
03b7f0a3c38c4      health:latest      "sh -c 'sleep 40 66..." 44 seconds ago Up 21 seconds 8080/tcp                          deployment-celerybeat-1
551d7d4d499a0      health:latest      "sh -c 'sleep 40 66..." 44 seconds ago Up 21 seconds 8080/tcp                          deployment-celeryworker-1
03b7f0a3c38c4      health:latest      "sh -c 'sleep 20 44..." 44 seconds ago Up 21 seconds 8080/tcp, 0.0.0.0:8080->80/tcp, :::8080->80/tcp deployment-django-1
44109f442403e      postgres:latest    "docker-entrypoint.s..." 53 seconds ago Up 20 seconds 0.0.0.0:5432->5432/tcp, :::5432->5432/tcp deployment-postgres-1
03bda48e7679       redis:latest        "docker-entrypoint.s..." 53 seconds ago Up 20 seconds 0.0.0.0:6379->6379/tcp, :::6379->6379/tcp deployment-redis-1
~@hora@Ahora-PC: ~/docker-k8s-project/deployment ~main~
~@hora@Ahora-PC: ~$ docker volume ls
DRIVER            VOLUME NAME
local             deployment-celerybeat-data
local             deployment-postgres-data
local             deployment-redis-data
~@hora@Ahora-PC: ~/docker-k8s-project/deployment ~main~
~@hora@Ahora-PC: ~$ docker network ls
NETWORK ID        NAME                DRIVER            SCOPE
b7c7170dc84       bridge             bridge           local
f8dc8737ca61     deployment-default bridge           local
11f9f957697       host               host             local
4d4fd4323174      none              null             local
~@hora@Ahora-PC: ~/docker-k8s-project/deployment ~main~
~@hora@Ahora-PC: ~$ docker logs -l
Observations to perform:
Apply all migrations: admin, auth, contenttypes, healthServiceApi, sessions
Running migrations:
No migrations to apply.
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
January 31, 2024 - 14:33:17
Django version 5.0.1, using settings 'healthService.settings'
Starting development server at http://0.0.0.0:8080/
Quit the server with CONTROL-C.

~@hora@Ahora-PC: ~/docker-k8s-project/deployment ~main~
~@hora@Ahora-PC: ~$ docker logs 55
/usr/local/lib/python3.10/site-packages/celery/platforms.py:829: SecurityWarning: You're running the worker with supervisor privileges: this is absolutely not recommended!

Please specify a different user using the --uid option.

User information: uid=0 euid=0 gid=0 egid=0

warnings.warn(SecurityWarning(ROOT_DISCOURAGED.format(
.....: celery@0551d7d6499a0 v5.3.6 (emerald-rush)

```

```

Terminal - ahora@Ahora-PC ~/docker-k8s-project/deployment

File Edit View Terminal Tabs Help
User information: uid=0 euid=0 gid=0 egid=0

warnings.warn(SecurityWarning(ROOT_DISCOURAGED.format(
----- celery@051d7d4d99a0 v5.3.6 (emerald-rush)
-----
----- Linux 5.15.6-92-generic x86_64 with 2024-01-31 14:33:20
-----
----- [config]
-----      > app: healthService-Bu7f0a3fff370
-----      > transport: redis://redis:6379/0
-----      > results: redis://redis:6379/0
-----      > concurrency: 8 (prefork)
-----      > task events: OFF (enable -E to monitor tasks in this worker)
-----
----- [queues]
-----      > celery exchange=celery(direct) key=celery

[tasks]
healthServiceApi.tasks.check_server_health

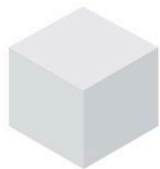
[2024-01-31 14:33:21,129: WARNING/MainProcess] /usr/local/lib/python3.10/site-packages/celery/worker/consumer/consumer.py:507: CPendingDeprecationWarning: The broker_connection_retry configuration setting will no longer determine
whether broker connection retries are made during startup in Celery 6.0 and above.
If you wish to retain the existing behavior for retrying connections on startup,
you should set broker_connection_retry_on_startup to True.
warnings.warn(
[2024-01-31 14:33:21,144: INFO/MainProcess] Connected to redis://redis:6379/0
[2024-01-31 14:33:21,145: WARNING/MainProcess] /usr/local/lib/python3.10/site-packages/celery/worker/consumer/consumer.py:507: CPendingDeprecationWarning: The broker_connection_retry configuration setting will no longer determine
whether broker connection retries are made during startup in Celery 6.0 and above.
If you wish to retain the existing behavior for retrying connections on startup,
you should set broker_connection_retry_on_startup to True.
warnings.warn(
[2024-01-31 14:33:21,148: INFO/MainProcess] mingle: searching for neighbors
[2024-01-31 14:33:22,162: INFO/MainProcess] mingle: all alone
[2024-01-31 14:33:22,210: INFO/MainProcess] celery@051d7d4d99a0 ready.
ahora@Ahora-PC ~/$ docker-k8s-project/deployment <main>
└─$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  STATUS
celery-beat-0  5.3.6 (emerald-rush)                celery-beat               Up
celery-worker-0  5.3.6 (emerald-rush)                celery                    Up
LocalTime -> 2024-01-31 14:33:20
configuration:
├─ broker -> redis://redis:6379/0
├─ loader -> celery.loaders.app.AppLoader
├─ scheduler -> celery.beat.PersistentScheduler
├─ db -> celery-beat-schedule
├─ logfile -> [stderr]@INFO
├─ maxinterval -> 5.80 minutes (30s)
[2024-01-31 14:33:20,806: INFO/MainProcess] beat: Starting...
ahora@Ahora-PC ~/$ docker-k8s-project/deployment <main>

```

در ادامه با دستور `docker build -t` از داکر فایل خود یک خروجی ایمج تهیه می کنیم و با دستور `docker push` ایمج تولید شده را وارد داکر هاب می کنیم.

```
mirFazelK@WSN-476 MINGW64 /e/university/sem7/cloud computing/CC/Projects/docker-k8s-project/healthService (main)
$ docker build -t health2:latest .
+] Building 3.3s (2/3)
=> [internal] load .dockerignore
=> => transferring context: 57B
=> [internal] load build definition from dockerfile
=> => transferring dockerfile: 351B
=> [internal] load metadata for docker.io/library/python:3.10.2-alpine
```

```
AmirFazelK@WSN-476 MINGW64 /e/university/sem7/cloud computing/CC/Proj
$ docker push amirfazel/health
Using default tag: latest
The push refers to repository [docker.io/amirfazel/health]
bbf7fa71bf00: Layer already exists
dc917bd5fc99: Layer already exists
6a277c40f434: Layer already exists
4542caa63ab7: Layer already exists
d381d1fc27a6: Layer already exists
4e4052a4ca72: Layer already exists
9a922de0e682: Layer already exists
f57e81d89e60: Layer already exists
8d3ac3489996: Layer already exists
```



amirfazel/health:latest

DIGEST: sha256:695524582a8759fbd8d5635f5753369c9ab7e235bfec8e9b871216c98a85896a

OS/ARCH	COMPRESSED SIZE ⓘ	LAST PUSHED	TYPE
linux/amd64	205.36 MB	a minute ago by amirfazel	Image

گام سوم

در ادامه با نوشتن فایل های توصیف کوبرنتیز و بالا آوردن پاد ها، موارد زیر را که شامل سرویس ها و پاد ها می باشد را داریم:

k get all:

```
Terminal - ahora@Ahora-PC:~/docker-k8s-project/deployment/helm
ahora@Ahora-PC:~/docker-k8s-project/deployment/helm$ k get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/celerybeat-deployment-dccfc448c-2fk94    1/1      Running   0           5m5s
pod/celeryworker-deployment-7fcsf8b66f-qk4t2  1/1      Running   0           5m5s
pod/health-release-alertmanager-0           1/1      Running   0           5m5s
pod/health-release-app-6dc65b9b4-tvr2t       1/1      Running   2 (2m39s ago)  5m5s
pod/health-release-grafana-b4cd8b5cb-4saa0   0/1      Running   1 (9m5s ago)   5m5s
pod/health-release-kube-state-metrics-658d7b4ff9-6wblt  1/1      Running   0           5m5s
pod/health-release-prometheus-node-exporter-kvfw  1/1      Running   0           5m5s
pod/health-release-prometheus-pushgateway-fcbf696df-49526  1/1      Running   0           5m5s
pod/health-release-prometheus-server-645f9fd75-nzp4c  2/2      Running   0           5m5s
pod/postgres-0                               1/1      Running   0           5m5s
pod/redis-deployment-65d8b57849-2nph6        1/1      Running   0           5m5s

NAME                                TYPE                                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/celerybeat-service           ClusterIP           10.104.87.95    <none>            80/TCP            5m13s
service/celeryworker-service         ClusterIP           10.104.87.95    <none>            80/TCP            5m13s
service/health-release-alertmanager ClusterIP           10.111.60.95    <none>            9093/TCP          5m13s
service/health-release-alertmanager-headless ClusterIP           10.111.60.95    <none>            9093/TCP          5m13s
service/health-release-grafana       LoadBalancer       10.105.57.148    <pending>         80-32751/TCP      5m13s
service/health-release-kube-state-metrics ClusterIP           10.104.144.154    <none>            8080/TCP          5m13s
service/health-release-prometheus-node-exporter ClusterIP           10.100.25.122     <none>            9100/TCP          5m13s
service/health-release-prometheus-pushgateway ClusterIP           10.96.240.211     <none>            9091/TCP          5m13s
service/health-release-prometheus-server ClusterIP           10.96.71.156     <none>            80/TCP            5m13s
service/health-release-service        ClusterIP           10.97.10.239     <none>            80/TCP            5m13s
service/kubernetes                    ClusterIP           10.96.0.1         <none>            443/TCP           8h
service/postgres-primary              ClusterIP           10.103.76.125    <none>            5432/TCP          5m13s
service/postgres-replica              ClusterIP           10.107.62.130    <none>            5432/TCP          5m13s
service/redis                         ClusterIP           10.98.145.126    <none>            6379/TCP          5m13s

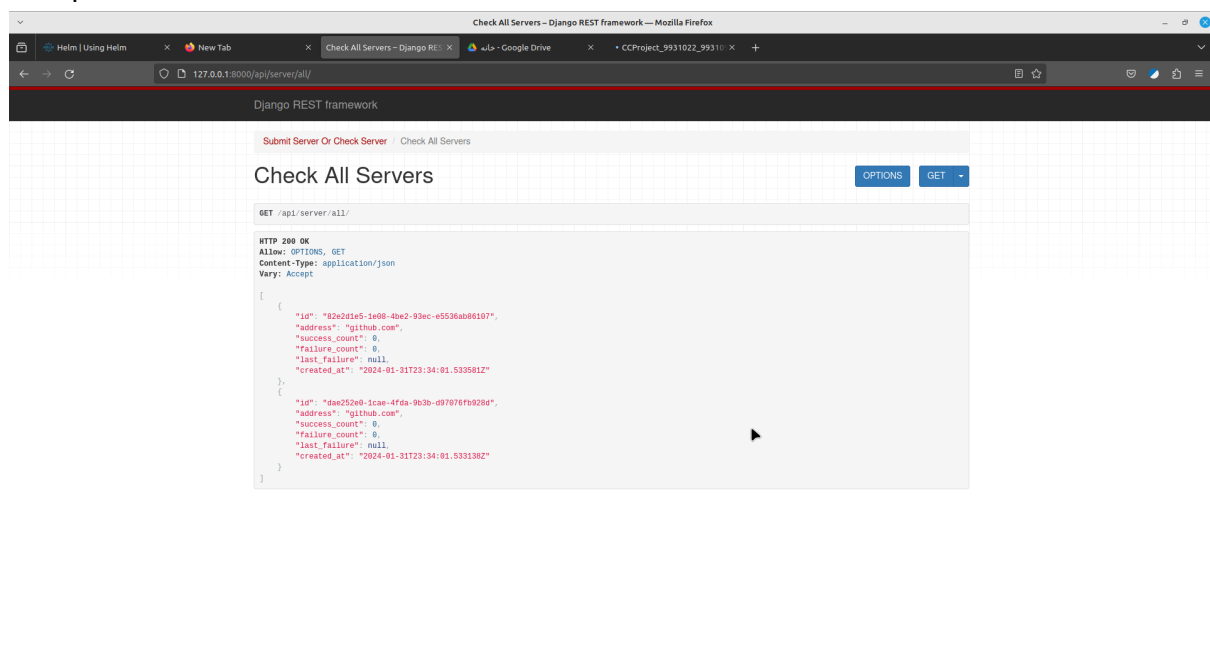
NAME                                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/health-release-prometheus-node-exporter  1          1          1          1             1             kubernetes.io/os=linux  5m5s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/celerybeat-deployment  1/1     1             1           5m5s
deployment.apps/celeryworker-deployment  1/1     1             1           5m5s
deployment.apps/health-release-app       0/1     1             0           5m5s
deployment.apps/health-release-grafana   0/1     1             0           5m5s
deployment.apps/health-release-kube-state-metrics  1/1     1             1           5m5s
deployment.apps/health-release-prometheus-pushgateway  1/1     1             1           5m5s
deployment.apps/health-release-prometheus-server  1/1     1             1           5m5s
deployment.apps/redis-deployment         1/1     1             1           5m5s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/celerybeat-deployment-dccfc448c  1          1          1           5m5s
replicaset.apps/celeryworker-deployment-7fcsf8b66f  1          1          1           5m5s
replicaset.apps/health-release-app-6dc65b9b4  1          1          1           5m5s
replicaset.apps/health-release-grafana-b4cd8b5cb  1          1          0           5m5s
replicaset.apps/health-release-kube-state-metrics-658d7b4ff9  1          1          1           5m5s
replicaset.apps/health-release-prometheus-pushgateway-fcbf696df  1          1          1           5m5s
replicaset.apps/health-release-prometheus-server-645f9fd75  1          1          1           5m5s
replicaset.apps/redis-deployment-65d8b57849  1          1          1           5m5s

NAME                                READY   AGE
statefulset.apps/health-release-alertmanager  1/1     5m5s
statefulset.apps/postgres                    1/1     5m5s
```

after port forward:



سوال سوم) در بخش stateful ، دو پاد تولید شد. یکی مربوط به خواندن یا slave و دیگری مربوط به نوشتن روی دیتابیس یا primary.

گام چهارم

در اینجا با وارد کردن دستور helm install کانتینر های کوبری خود را بالا می آوریم.

helm install

```
Terminal - ahora@Ahora-PC:~/docker-k8s-project/deployment/helm
File Edit View Terminal Tabs Help
~$ helm install healthr health
NAME: healthr
LAST DEPLOYED: Thu Feb 1 04:06:44 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
~$
```

k get pods after helm install:

```
Terminal - ahora@Ahora-PC:~/docker-k8s-project/deployment/helm
File Edit View Terminal Tabs Help
~$ helm install healthr health
NAME: healthr
LAST DEPLOYED: Thu Feb 1 04:06:44 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
~$ k get pods
NAME                                READY   STATUS             RESTARTS   AGE
calerybeat-deployment-64568df4d4-t7tg7 0/1     ContainerCreating  0          26s
coryworker-deployment-7c37f7b474-dpbmq 0/1     ContainerCreating  0          26s
healthr-alertmanager-0                 0/1     ContainerCreating  0          26s
healthr-app-759757444b-6jnrw           0/1     ContainerCreating  0          26s
healthr-grafana-96558544b-9exjf         0/1     ContainerCreating  0          22s
healthr-kube-state-metrics-75c469875-d9mzx 0/1     ContainerCreating  0          22s
healthr-prometheus-node-exporter-kt6zc  0/1     ContainerCreating  0          26s
healthr-prometheus-pushgateway-7f65d65d9-9rlz9 0/1     ContainerCreating  0          26s
healthr-prometheus-server-59c467954f-fvc4h 0/2     ContainerCreating  0          22s
my-postgres-postgresql-primary-0        1/1     Running            0          16m
my-postgres-postgresql-read-0           1/1     Running            0          16m
redis-deployment-65d8857849-w2z5z       0/1     ContainerCreating  0          26s
~$
```

helm install final:

```

File Edit View Terminal Tabs Help
ahora@ahora-PC: ~/docker-k8s-project/deployment/helm $ helm upgrade --install health health-0.1.0.tgz
[+] help package health
Successfully packaged chart and saved it to: /home/ahora/docker-k8s-project/deployment/helm/health-0.1.0.tgz
ahora@ahora-PC: ~/docker-k8s-project/deployment/helm $ helm upgrade --install health health-0.1.0.tgz
[+] help install health health
NAME: health
LAST DEPLOYED: Thu Feb 1 04:55:29 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
ahora@ahora-PC: ~/docker-k8s-project/deployment/helm $ helm upgrade --install health health-0.1.0.tgz
[+] k get pods
NAME                                READY    STATUS    RESTARTS   AGE
celerybeat-deployment-64568df4d4-hr4nc 0/1      Pending   0           11s
celeryworker-deployment-7c5f7f8474-n7gm9 0/1      Pending   0           11s
healthr-alertmanager-0                 0/1      Pending   0           7s
healthr-app-7597574448-lfuzn           0/1      Pending   0           8s
healthr-grafana-965585448-xfwdp         0/1      Pending   0           5s
healthr-kube-state-metrics-79c4d98b75-f4nd4 0/1      Pending   0           11s
healthr-postgresql-primary-0           0/1      Pending   0           7s
healthr-postgresql-read-0              0/1      Pending   0           7s
healthr-prometheus-node-exporter-vzjq9   0/1      Pending   0           11s
healthr-prometheus-pushgateway-7f65d4d5d9-nq7zg 0/1      Pending   0           11s
healthr-prometheus-server-9fc467954f-bgbnc 0/2      Pending   0           5s
redis-deployment-65d8b57849-x3lton      0/1      ContainerCreating 0           11s
ahora@ahora-PC: ~/docker-k8s-project/deployment/helm $ helm upgrade --install health health-0.1.0.tgz
[+]

```

```

Terminal - ahora@Ahora-PC:~/docker-k8s-project/deployment/helm
File Edit View Terminal Tabs Help

ahora@Ahora-PC: ~/docker-k8s-project/deployment/helm «main»
└─ k get pods

NAME                                READY   STATUS    RESTARTS   AGE
celerybeat-deployment-6450df44d-hrnc 1/1     Running   0           5612s
ciceworker-deployment-7c57f78474-w7gn9 1/1     Running   0           5612s
health-alertmanager-0                 1/1     Running   0           560s
health-app-75975f4448-1fuzn           0/1     Error     3 (56s ago) 560s
health-grafana-965385448-xfwdp        0/1     Running   1 (49s ago) 560s
health-kube-state-metrics-75cc469875-f4nd4 1/1     Running   0           5612s
health-postgresql-primary-0           1/1     Running   1 (87s ago) 560s
health-postgresql-read-0              0/1     Running   2 (45s ago) 560s
health-prometheus-node-exporter-wzjg0 1/1     Running   0           5612s
health-prometheus-pushgateway-7f65d5d89-nq7zg 1/1     Running   0           5612s
health-prometheus-server-59c467854f-bgbcn 2/2     Running   0           560s
redis-deployment-65d8d57849-k9lrm     1/1     Running   0           5612s

ahora@Ahora-PC: ~/docker-k8s-project/deployment/helm «main»
└─ helm uninstall health

release "health" uninstalled

ahora@Ahora-PC: ~/docker-k8s-project/deployment/helm «main»
└─ k get pods

NAME                                READY   STATUS    RESTARTS   AGE
celerybeat-deployment-6450df44d-hrnc 0/1     Terminating   0           5656s
ciceworker-deployment-7c57f78474-w7gn9 1/1     Terminating   0           5656s
health-alertmanager-0                 0/1     Terminating   0           5652s
health-app-75975f4448-1fuzn           0/1     Terminating   3           5652s
health-grafana-965385448-xfwdp        0/1     Terminating   1 (93s ago) 5650s
health-kube-state-metrics-75cc469875-f4nd4 0/1     Terminating   0           5656s
health-postgresql-primary-0           0/1     Terminating   1 (2m11s ago) 5652s
health-postgresql-read-0              0/1     Terminating   2 (89s ago) 5652s
health-prometheus-node-exporter-wzjg0 0/1     Terminating   0           5656s
health-prometheus-pushgateway-7f65d5d89-nq7zg 0/1     Terminating   0           5656s
health-prometheus-server-59c467854f-bgbcn 0/2     Terminating   0           5650s
redis-deployment-65d8d57849-k9lrm     0/1     Terminating   0           5656s

ahora@Ahora-PC: ~/docker-k8s-project/deployment/helm «main»

```

بخش امتیازی

probes

1)

liveness:

چک می کند تا ببیند که یک کانتینر به درستی اجرا می شود یا نه

readiness:

چک میکند تا ببیند آیا یک کانتینر آماده دریافت پهنای باند هست یا نه

startup:

چک می کند تا ببیند یک برنامه درون یک کانتینر به درستی اجرا می شود یا نه

(2)

برای دو پروب **liveness** و **startup** به دلیل داشتن یک اندپوینت **ping** و دانستن این موضوع که برای ایجاد پایگاه داده عملیات **migrate** انجام می شود پس می توان گفت تا بالا آمدن کانتینر برنامه هم بالا می آید هر دوی این پروب ها را با اندپوینت **ping** اطمینان می دهیم. پروب **readiness** هم مرتبط با دریافت پهنای باند می شد. در این برنامه هرگاه که قابلیت خواندن از دیتابیس را داشته باشیم می توانیم **readiness** را نیز اطمینان دهیم.

prometheus:

متریک ها:

- یک شمارنده برای تعداد ریکوئست ها
- یک شمارنده برای ریکوئست های ناموفق
- یک هیستوگرام برای زمان پاسخگویی ریکوئست ها
- یک **gauge** برای نگهداری از تعداد سرور های دیتا بیس

