

Hera部署文档

1. 部署说明

Hera operator的作用是在k8s集群中的指定namespace下一键拉起一个hera平台。该文档适用于有一定k8s基础(PV、PVC、Service、Pod、Deployment、DaemonSet等)的研发/运维同学。

Hera是一套企业级的可观测性平台，部署时复杂度非常高，部署前请认真阅读以下部署文档及相关的[operator介绍视频](#)。

2. 部署步骤

hera-all/hera-operator/hera-operator-server/src/main/resources/operator/

2.1 创建独立命名空间及账号

- 执行命令，生效auth yaml（默认会生成空间：hera-namespace，账号：admin-mone）

```
kubectl apply -f hera_operator_auth.yaml
```

2.2 创建 hera CRD

- 执行命令，生效crd yaml

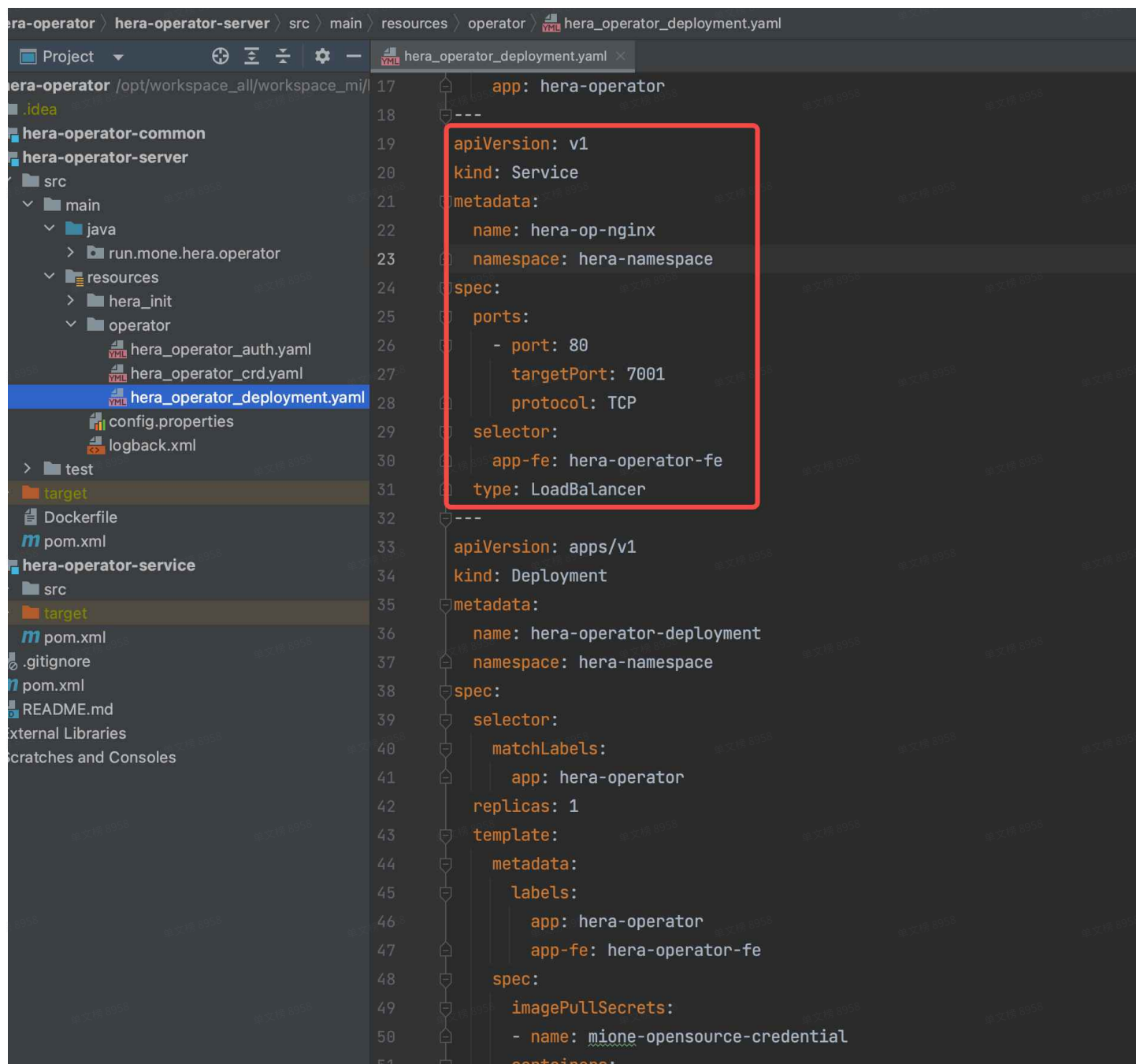
```
kubectl apply -f hera_operator_crd.yaml
```

2.3 部署operator

- 执行命令，部署operator

```
kubectl apply -f hera_operator_deployment.yaml
```

- 确保部署的operator工程端口7001，能够对外访问。hera部署需要在operator提供的对外页面上进行操作。默认例子中使用LoadBalancer方式对外暴露可访问的ip、port。如需使用其它方式请自行修改yaml。



```
hera-operator > hera-operator-server > src > main > resources > operator > hera_operator_deployment.yaml
Project hera_operator_deployment.yaml x
hera-operator /opt/workspace_all/workspace_mi/ 17 app: hera-operator
18 ---
19 apiVersion: v1
20 kind: Service
21 metadata:
22   name: hera-op-nginx
23   namespace: hera-namespace
24 spec:
25   ports:
26     - port: 80
27       targetPort: 7001
28       protocol: TCP
29   selector:
30     app-fe: hera-operator-fe
31   type: LoadBalancer
32 ---
33 apiVersion: apps/v1
34 kind: Deployment
35 metadata:
36   name: hera-operator-deployment
37   namespace: hera-namespace
38 spec:
39   selector:
40     matchLabels:
41       app: hera-operator
42   replicas: 1
43   template:
44     metadata:
45       labels:
46         app: hera-operator
47         app-fe: hera-operator-fe
48     spec:
49       imagePullSecrets:
50       - name: mione-opensource-credential
51       containers:
```

2.4 Operator页面操作

2.4.1 访问operator页面

如果是使用2.3步中LoadBalancer方式，请先找到"hera-op-nginx" service的对外ip。执行命令：

```
kubectl get service -n=hera-namespace
```

找到hera-op-nginx 对应的EXTERNAL-IP

默认访问地址：http://EXTERNAL-IP:80/，可见如下界面：

HERA Operator

①

第二步

③

第一步

K8S CR 元数据

* name: hera-bootstrap

* Namespace: hera-namespace

下一步

2.4.2 operator元数据填写

- name: hera-bootstrap
- Namespace: hera-namespace

k8s自定义资源名称，保持默认值不变

hera部署的独立空间，建议保持hera-namespace不变，如需改变请注意yaml全局变更

2.4.3 k8s访问方式确认

该步骤是生成hera平台中需要对外开放页面的访问ip:port。当前只支持k8s的LoadBalancer、NodePort方式。默认会先尝试LB模式，如若不支持，则选择NodePort（如果NodePort的ip未开放对外访问，则需另起代理，建议集群开启LB）。

HERA Operator

①

第二步

③

第一步

K8S访问方式确认

点我确认

K8S访问方式确认

访问方式：LoadBalancer

连接信息

hera.tpc.url:	1: 3:80	描述:	对外访问地址
hera.grafana.url:	1: 6:80	描述:	对外访问地址
hera.homepage.url:	1: :80	描述:	对外访问地址
hera.prometheus.url:	1: 86:80	描述:	对外访问地址
tpc.login.fe.url:	1: 18:80	描述:	对外访问地址
hera.alertmanager.url:	1: 15:80	描述:	对外访问地址

请记住hera.homepage.url，hera集群搭建完后，默认访问地址就是：http://\${hera.homepage.url}

2.4.4 集群配置

*k8s-serviceType*请勿修改

资源/部署初始化

k8s-serviceType

基于yaml创建资源：☐

连接信息

hera.tpc.url:	1: 3:80	描述:	对外访问地址
hera.grafana.url:	1: 56:80	描述:	对外访问地址
hera.homepage.url:	1: 1:80	描述:	对外访问地址
hera.prometheus.url:	1: 3:80	描述:	对外访问地址
tpc.login.fe.url:	1: 3:80	描述:	对外访问地址
hera.alertmanager.url:	1: 5:80	描述:	对外访问地址

Hera-mysql

目的是选择一个hera可用的mysql数据库，

☐ 如果需要k8s自动搭建一个数据库

则开启"基于yaml创建资源"按钮，默认的yaml会创建一个pv进行mysql的数据存储，如果沿用默认的yaml，一定要注意：1. 提前在宿主机node上创建目录/opt/hera_pv/hera_mysql（可更换目录，

同步修改此处yaml)；2. 找到创建目录的node名（可以执行 kubectl get node进行确认），替换此处的cn-bxxx52；3. 连接信息确保与yaml中信息一致，默认无需修改

hera-mysql

基于yaml创建资源：☒

```
1 apiVersion: v1
2 kind: PersistentVolume
3 metadata:
4   name: hera-mysql-pv
5 spec:
6   capacity:
7     storage: 10Gi
8   volumeMode: Filesystem
9   accessModes:
10    - ReadWriteOnce
11   persistentVolumeReclaimPolicy: Delete
12   storageClassName: hera-mysql-local-storage
13   local:
14     path: /opt/hera_pv/hera_mysql
15   nodeAffinity:
16     required:
17       nodeSelectorTerms:
18         - matchExpressions:
19           - key: kubernetes.io/hostname
20             operator: In
21             values:
22               - cn-b-52
23 ---
24 apiVersion: storage.k8s.io/v1
25 kind: StorageClass
```

备注：

local pv，确保磁盘目录已提前创建好

连接信息

* hera.datasource.url: 描述:

* hera.datasource.username: 描述:

* hera.datasource.password: 描述:

☐ 如果已有数据库，无需k8s创建

则：1. 关闭"基于yaml创建资源"按钮；2. 填写正确的已有数据库url、用户名、密码；3. 默认operator执行时会自动去改数据库进行创建hera数据库及表，如果填写的账号无建库、建表权限，则需提前手动去目标库中建好hera数据库和表，建表语句在operator源码hera-all/hera-operator/hera-operator-server/src/main/resources/hera_init/mysql/sql 目录下

hera-mysql

基于yaml创建资源：☐

连接信息

* hera.datasource.url: 描述:

* hera.datasource.username: 描述:

* hera.datasource.password: 描述:

Hera-es

目的是选择一个Hera可用的ES集群，并在ES中创建Hera所需要的索引模板。

☐ 如果需要k8s自动搭建一个ES

则需要开启“基于yaml创建资源”按钮。使用默认yaml创建的ES没有账号密码，如果需要设置账号密码，则需要：

1、修改左侧yaml中的xpack.security.enabled为true

- 2、修改右侧“连接信息”中的hera.es.username与hera.es.password的值，一般地，我们都会用elastic的账号，密码需要在ES服务启动后进行设置
- 3、在ES启动后，登入ES所在pod中，进入/usr/share/elasticsearch/bin目录执行elasticsearch-setup-passwords interactive命令，设置ES默认账号的密码，注意，这里设置的密码，需要与页面hera.es.password的值保持一致

hera-es

基于yaml创建资源：☒

```
1 apiVersion: v1
2 kind: ConfigMap
3 metadata:
4   labels:
5     product: k8s-elastic
6   name: elastic-config
7   namespace: hera-namespace
8 data:
9   elasticsearch.yaml: |
10    discovery.type: single-node
11    discovery.seed_hosts: ["127.0.0.1"]
12    network.host: 0.0.0.0
13    http.port: 9200
14    xpack.security.enabled: false
15 ---
16 apiVersion: apps/v1
17 kind: Deployment
18 metadata:
19   name: elasticsearch
20   namespace: hera-namespace
21 spec:
22   selector:
23     matchLabels:
24       run: elasticsearch
25   template:
```

备注: local es

连接信息

* hera.es.url: 描述: es连接地址, host:port

hera.es.username: 描述: es账号, 可不填

hera.es.password: 描述: es密码, 可不填

☐ 如果已有ES，无需k8s创建

- 则：1、关闭"基于yaml创建资源"按钮；
- 2、填写正确的已有ES集群的url、账号、密码
- 3、默认operator执行时会自动创建索引模版。如果填写的账号无创建索引模版的权限，则需要提前手动创建Hera所需要的索引模版。Hera的索引模版在operator源码run.mone.hera.operator.common.ESIndexConst中，以json的格式存储。

hera-es

基于yaml创建资源：☐

连接信息

* hera.es.url: 描述: es连接地址, host:port

hera.es.username: 描述: es账号, 可不填

hera.es.password: 描述: es密码, 可不填

hera-rocketMQ

目的是选择一个hera可用的RocketMQ

☐ 如果需要k8s自动搭建一个RocketMQ

1、需要开启“基于yaml创建资源”按钮。

2、使用默认yaml创建的RocketMQ没有accessKey\secretKey，如果需要设置accessKey\secretKey，则需要修改右侧“连接信息”中的hera.rocketmq.ak与hera.rocketmq.sk的值

3、如果需要更换RocketMQ broker的service，需要同时替换yaml中的service，以及hera-operator代码中的run.mone.hera.operator.service.RocketMQService类的成员变量"brokerAddr"的值

hera-rocketMQ

基于yaml创建资源: ☒

```
1 kind: Deployment
2 apiVersion: apps/v1
3
4
5 metadata:
6
7   name: rocketmq-name-service
8
9   namespace: hera-namespace
10
11 spec:
12
13   replicas: 1
14
15   selector:
16
17     matchLabels:
18
19       app: rocketmq-name-service
20
21       name_service_cr: rocketmq-name-service
22
23   template:
24
25     metadata:
```

备注: local rocketMQ

连接信息

* hera.rocketmq.nameserver: 描述: rocketMQ连接地址, host:port

hera.rocketmq.ak: 描述: rocketMQ accessKey, 可不填

hera.rocketmq.sk: 描述: rocketMQ secretKey, 可不填

☐ 如果已有RocketMQ，无需k8s搭建

则：1、关闭"基于yaml创建资源"按钮；

2、填写正确的已有RocketMQ集群的url、accessKey、secretKey

3、默认operator执行时会自动创建Hera所需要的topic。如果填写的url、ak、sk没有权限创建topic，或者已有RocketMQ集群不允许通过API创建topic，则需要提前手动创建好topic。Hera需要的topic在operator源码run.mone.hera.operator.service.RocketMQService类的成员变量"topics"中存储

hera-rocketMQ

基于yaml创建资源: ☐

连接信息

* hera.rocketmq.nameserver: 描述: rocketMQ连接地址, host:port

hera.rocketmq.ak: 描述: rocketMQ accessKey, 可不填

hera.rocketmq.sk: 描述: rocketMQ secretKey, 可不填

Hera-redis

目的是选择一个hera可用的Redis

☐ 如果需要k8s自动搭建一个Redis

则需要开启“基于yaml创建资源”按钮。使用默认yaml创建的redis没有密码，如果需要设置密码，则需要修改右侧hera.redis.password的值，与redis设置的密码保持一致

hera-redis

基于yaml创建资源：☒

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: redis-service
5   namespace: hera-namespace
6 labels:
7   name: redis
8 spec:
9   ports:
10    - port: 6379
11      protocol: TCP
12 selector:
13   name: redis
14 clusterIP: None
15 ---
16 apiVersion: apps/v1
17 kind: Deployment
18 metadata:
19   name: redis-deployment
20   namespace: hera-namespace
21 spec:
22   replicas: 1
23   selector:
24     matchLabels:
25       name: redis
```

备注: local pv, 确保磁盘目录已提前创建好

连接信息

* hera.redis.url:

描述: redis连接地址, host:port

hera.redis.password:

描述: redis密码, 可不填

☐ 如果已有Redis，无需k8s搭建

- 则：1、关闭"基于yaml创建资源"按钮；
2、填写正确的已有Redis集群的URL、密码

hera-redis

基于yaml创建资源：☐

连接信息

* hera.redis.url:

描述: redis连接地址, host:port

hera.redis.password:

描述: redis密码, 可不填

Hera-Nacos

hera集群内部的配置、注册中心，该集群建议走yaml创建方式，如果业务需要自行提供Nacos，请优先提供1.x版本的Nacos。

☐ Nacos集群

☐ 如果需要k8s自动搭建一个Nacos

则需要开启“基于yaml创建资源”按钮，注意yaml中的镜像地址、资源大小配置及右侧连接信息与yaml中保持一致

hera-nacos

基于yaml创建资源：☒

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: nacos
5   namespace: hera-namespace
6 labels:
7   app: nacos
8 spec:
9   replicas: 1
10  selector:
11    matchLabels:
12      app: nacos
13  template:
14    metadata:
15      labels:
16        app: nacos
17    spec:
18      containers:
19        - name: nacos-container
20          image: prod-mione-opensource-registry.cn-beijing.cr.aliyuncs.com/prod-mione-opensource/prod-mione-opensource-mone-nacos-mysql-v0.0.1
21      resources:
22        limits:
23          cpu: '1'
24          memory: 5Gi
```

备注：

load nacos

连接信息

* hera.nacos.address:

nacos:80

描述:

nacos地址 host:ip

hera.nacos.username:

nacos

描述:

nacos账号, 可不填

hera.nacos.password:

nacos

描述:

nacos密码, 可不填

连接信息

hera_app_config_#_DEFAULT_GROUP

```
1 spring.datasource.url=jdbc:mysql://mone-db-all:3306/hera?
   characterEncoding=utf8&useSSL=false
2 spring.datasource.username=root
3 spring.datasource.password=Mone_123456
4
```

☐ 如果已有Nacos，无需k8s创建

则需关闭“基于yaml创建资源”按钮，填写正确的nacos连接信息

hera-nacos

基于yaml创建资源：☐

连接信息

* hera.nacos.address:

nacos:80

描述:

nacos地址 host:ip

hera.nacos.username:

nacos

描述:

nacos账号, 可不填

hera.nacos.password:

nacos

描述:

nacos密码, 可不填

☐ Nacos配置

operator会默认将这里所列的配置初始化为nacos配置，如果提供的不是基于yaml所创建的nacos，请确认连接信息有权限调用config创建接口，否则需要提前去目标nacos中手动创建好。

连接信息

hera_app_config_#_DEFAULT_GROUP

hera_log_manager_open_#_DEFAULT_GROUP

hera_trace_config_#_DEFAULT_GROUP

log_stream_dataId_open_#_DEFAULT_GROUP

mi_tpc_#_DEFAULT_GROUP

mi_tpc_login_#_DEFAULT_GROUP

mimonitor_open_config_#_DEFAULT_GROUP

prometheus_agent_open_config_#_DEFAULT_GROUP

1 #server

2 dubbo.group=opensource-outer

3 dubbo.protocol.id=dubbo

4 dubbo.protocol.name=dubbo

5 dubbo.protocol.port=-1

6 dubbo.registry.address=nacos://nacos:80

7 nacos.username=nacos

8 nacos.password=nacos

9

10 youpin.log.group=opensource-outer

11 log.path=/home/work/log

12

13 mione.k8s.container.port=5195

14 mione.k8s.node.port=9101

15

保存配置

集群生效

上一步

Hera-Prometheus

目的是选择一个hera可用的prometheus

如果沿用默认的yaml，一定要注意：

1. 提前在宿主机node上创建目录/home/work/prometheus_hera_namespace_pv;
2. 找到创建目录的node名（可以执行 kubectl get node进行确认），替换此处的cn- xxx;

```
59 labels:
60   k8s-app: prometheus
61 spec:
62   capacity:
63     storage: 10Gi
64   accessModes:
65     - ReadWriteOnce
66   persistentVolumeReclaimPolicy: Retain
67   storageClassName: local-storage
68   local:
69     path: /home/work/prometheus_hera_namespace_pv
70   nodeAffinity:
71     required:
72       nodeSelectorTerms:
73         - matchExpressions:
74           - key: kubernetes.io/hostname
75             operator: In
76             values:
77               - cn-beijing.10.53.128.152
78 ---
79 apiVersion: v1
80 kind: PersistentVolumeClaim
81 metadata:
82   name: prometheus-hera-namespace
83   namespace: hera-namespace
84 labels:
```

Hera-Alertmanager

目的是选择一个hera可用的alertmanager

如果沿用默认的yaml，一定要注意：

1. 提前在宿主机node上创建目录/home/work/alertmanager_hera_namespace_pv;
2. 找到创建目录的node名（可以执行 `kubectl get node` 进行确认），替换此处的cn-;

hera-alertmanager

基于yaml创建资源:



```

30 name: alertmanager-hera-namespace
31 labels:
32   k8s-app: alertmanager-hera-namespace
33 spec:
34   capacity:
35     storage: 2Gi
36   accessModes:
37     - ReadWriteOnce
38   persistentVolumeReclaimPolicy: Retain
39   storageClassName: local-storage
40   local:
41     path: /home/work/alertmanager_hera_namespace_pv
42   nodeAffinity:
43     required:
44       nodeSelectorTerms:
45         - matchExpressions:
46             - key: kubernetes.io/hostname
47               operator: In
48               values:
49                 - cn-12
50 ---
51 kind: PersistentVolumeClaim
52 apiVersion: v1
53 metadata:
54   name: alertmanager-hera-namespace

```

Hera-Grafana

目的是选择一个hera可用的grafana

如果沿用默认的yaml，一定要注意：

1. 提前在宿主机node上创建目录/home/work/grafana_hera_namespace_pv;
2. 找到创建目录的node名（可以执行 kubectl get node进行确认），替换此处的cn- beijingxxx;
3. 在Hera-mysql中配置的host、user、port、password等内容需要在hera-grafana的相应db配置中进行覆盖

基于yaml创建资源: ☒

```
1639   storage: 4Gi
1640   accessModes:
1641     - ReadWriteOnce
1642   persistentVolumeReclaimPolicy: Retain
1643   storageClassName: local-storage
1644   local:
1645     path: /home/work/grafana_hera_namespace_pv
1646   nodeAffinity:
1647     required:
1648       nodeSelectorTerms:
1649         - matchExpressions:
1650           - key: kubernetes.io/hostname
1651             operator: In
1652             values:
1653               - cn-beijing-1a1
1654   ---
1655   apiVersion: v1
1656   kind: PersistentVolumeClaim
1657   metadata:
1658     name: grafana-hera-namespace
1659     namespace: hera-namespace
1660   labels:
1661     k8s-app: grafana-hera-namespace
1662   spec:
1663     accessModes:
1664       - ReadWriteOnce
```

基于yaml创建资源: ☒

```
81 # Sets the maximum time in minutes before timing out read of an incoming request and
82 # closing idle connections.
83 read_timeout = 0
84
85 ##### Database #####
86 [database]
87 # You can configure the database connection by specifying type, host, name, user and
88 # password
89 # as separate properties or as on string using the url property.
90 # Either "mysql", "postgres" or "sqlite3", it's your choice
91 type = mysql
92 host = mone-db-all:3306
93 name = mione_hera_grafana
94 user = root
95 # If the password contains # or ; you have to wrap it with triple quotes. Ex """"#password;""""
96 password = Mone_123456
97 # Use either URL or the previous fields to configure the database
98 # Example: mysql://user:secret@host:port/database
99 url = mysql://root:Mone_123456@mone-db-all:3306/mione_hera_grafana
100
101 # Max idle conn setting default is 2
102 max_idle_conn = 2
103
104 # Max conn setting default is 0 (mean not set)
```

连接信息

Hera-Node-exporter

目的是选择一个hera可用的node-exporter

如果沿用默认的yaml，一定要注意：

1. 提前在宿主机找到一个可用Port，并填入下图中的hostPort处，默认为9101，修改完成后，同步更新右侧连接信息中的mione.k8s.node.port内容。

hera-node-exporter

基于yaml创建资源: ☒

```
36 - '--no-collector.wmi'
37 - '--no-collector.hwmon'
38 - '--collector.filesystem.ignored-mount-
points="/(dev|proc|sys|var|lib/docker/.+|var/lib/kubelet/pods/.+)|($|/)'
39 - '--collector.netclass.ignored-devices="(veth.*[a-f0-9]{15})$"'
40 - '--collector.netdev.device-exclude="(veth.*[a-f0-9]{15})$"'
41 - '--no-collector.ipvs'
42 - '--web.listen-address=:9101'
43 ports:
44 - name: http
45   hostPort: 9101
46   containerPort: 9101
47   protocol: TCP
48 resources:
49   limits:
50     cpu: 250m
51     memory: 400Mi
52   requests:
53     cpu: 102m
54     memory: 180Mi
55 volumeMounts:
56 - name: sys
57   readOnly: true
58   mountPath: /host/sys
59   mountPropagation: HostToContainer
60 - name: root
```

连接信息

* mione.k8s.node.port: 9100 描述: node-exporter的port

备注: local pv, 确保磁盘目录已提前创建好

Hera-Cadvisor

目的是选择一个hera可用的cadvisor

如果沿用默认的yaml, 一定要注意:

1. 提前在宿主机找到一个可用Port, 并填入下图中的hostPort处, 默认为5195, 修改后同步更新右侧连接信息中的mione.k8s.container.port内容。

hera-cadvisor

基于yaml创建资源: ☒

```
37 path: /dev/disk
38 type: "
39 containers:
40 - name: cadvisor
41   image: prod-mione-opensource-registry.cn-beijing.cr.aliyuncs.com/prod-mione-
opensource/prod-mione-opensource:cadvisor-v0.42.0
42   args:
43     - '--port=5195'
44     - '--housekeeping_interval=10s'
45     - '--max_housekeeping_interval=15s'
46     - '--event_storage_event_limit=default=0'
47     - '--event_storage_age_limit=default=0'
48     - '--enable_metrics=cpuLoad,advtcp,tcp,network,app,cpu,disk,diskIO,memory,process'
49     - '--store_container_labels=false'
50     - '--docker_only'
51     - '--
whitelisted_container_labels=io.kubernetes.container.name,io.kubernetes.pod.name,io.kubernetes.
pod.namespace,annotation.io.kubernetes.container.restartCount'
52     - '--env_metadata_whitelist=ENV_ID,PROJECT_ID,APPLICATION,serverEnv'
53     - '--enable_load_reader=true'
54 ports:
55 - name: http
56   hostPort: 5195
57   containerPort: 5195
58   protocol: TCP
```

连接信息

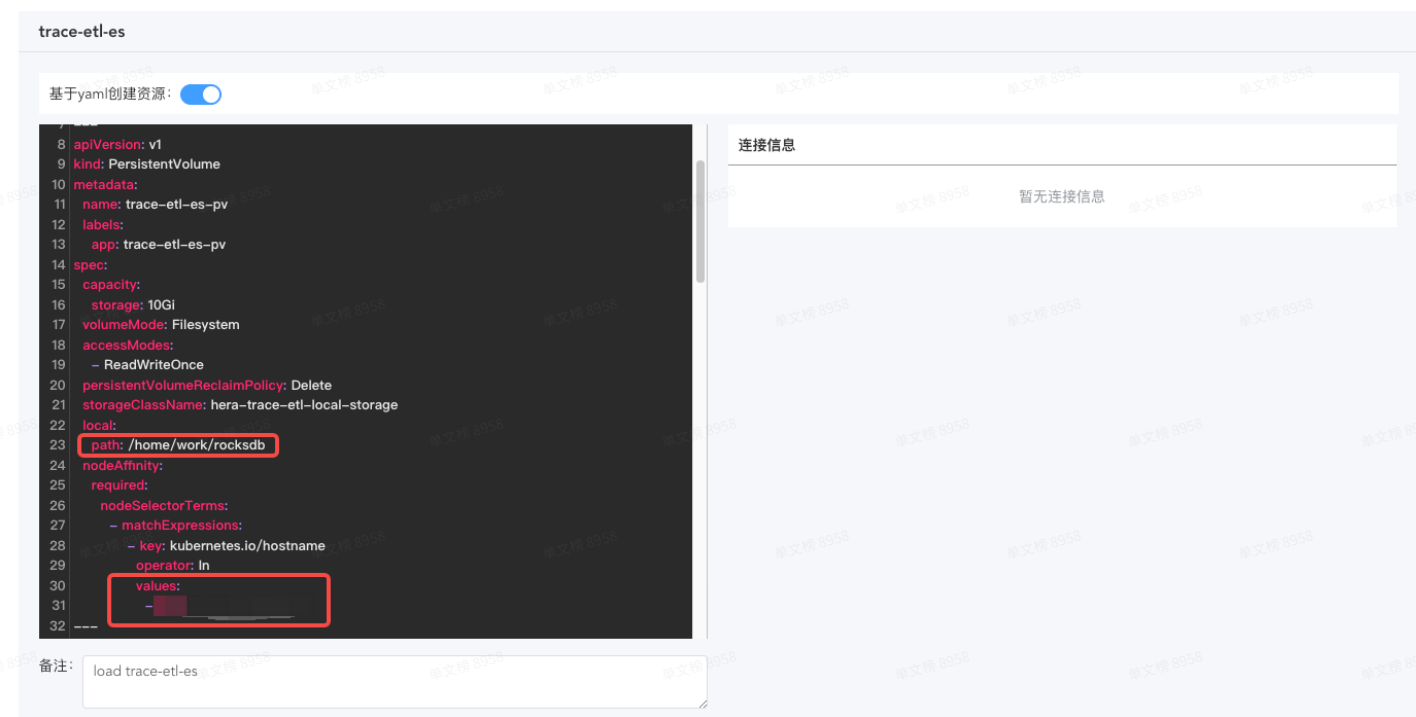
* mione.k8s.container.port: 5194 描述: cadvisor的port

Hera-trace-etl-es

需要注意的是:

- 1、该服务是StatefulSet类型服务
- 2、提前在宿主机node上创建目录/home/work/rocksdb (可更换目录, 同步修改此处yaml)

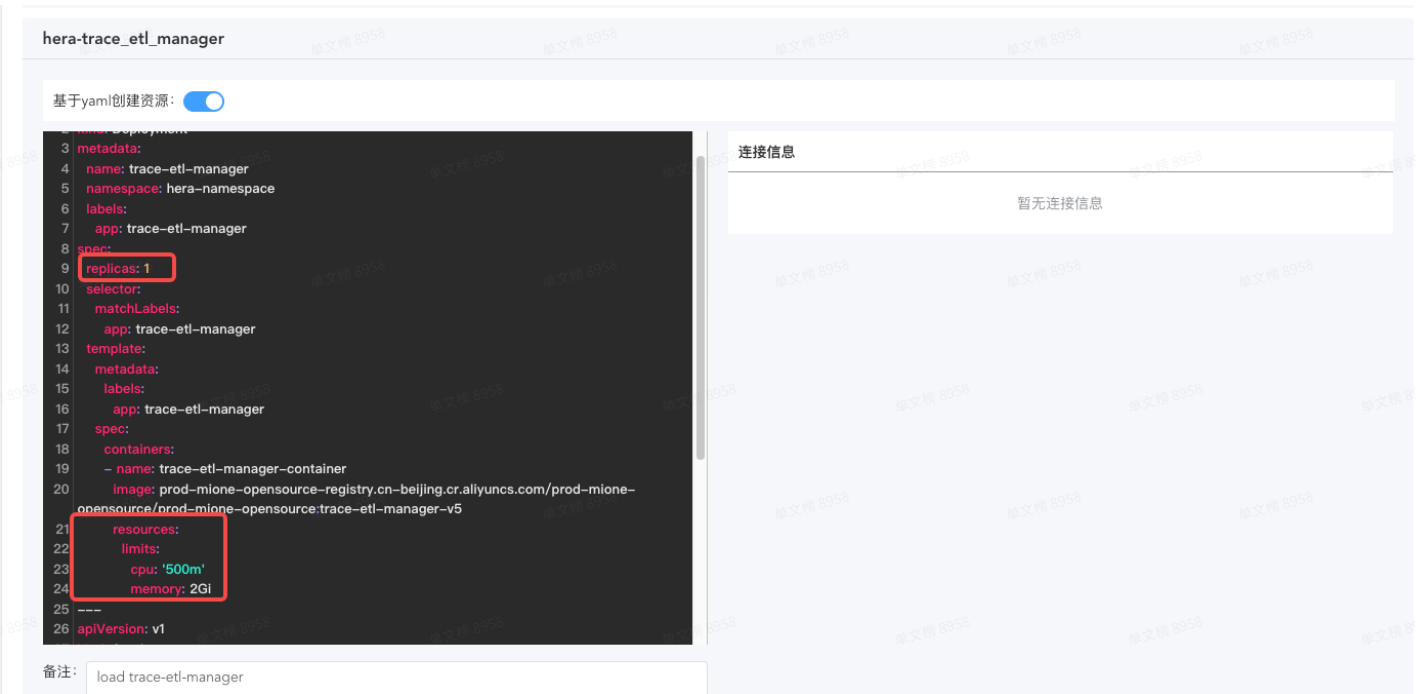
- 3、找到创建目录的node名（可以执行 kubectl get node进行确认），替换此处的 nodeSelectorTerms下的values的值
- 4、需要根据trace流量来修改pod副本数(replicas)与pod资源限制(resources)
- 5、服务的pod副本数尽量与RocketMQ的queue size保持一致



hera-trace_etl_manager

需要注意的是：

- 1、需要根据访问量来修改pod副本数(replicas)与pod资源限制(resources)

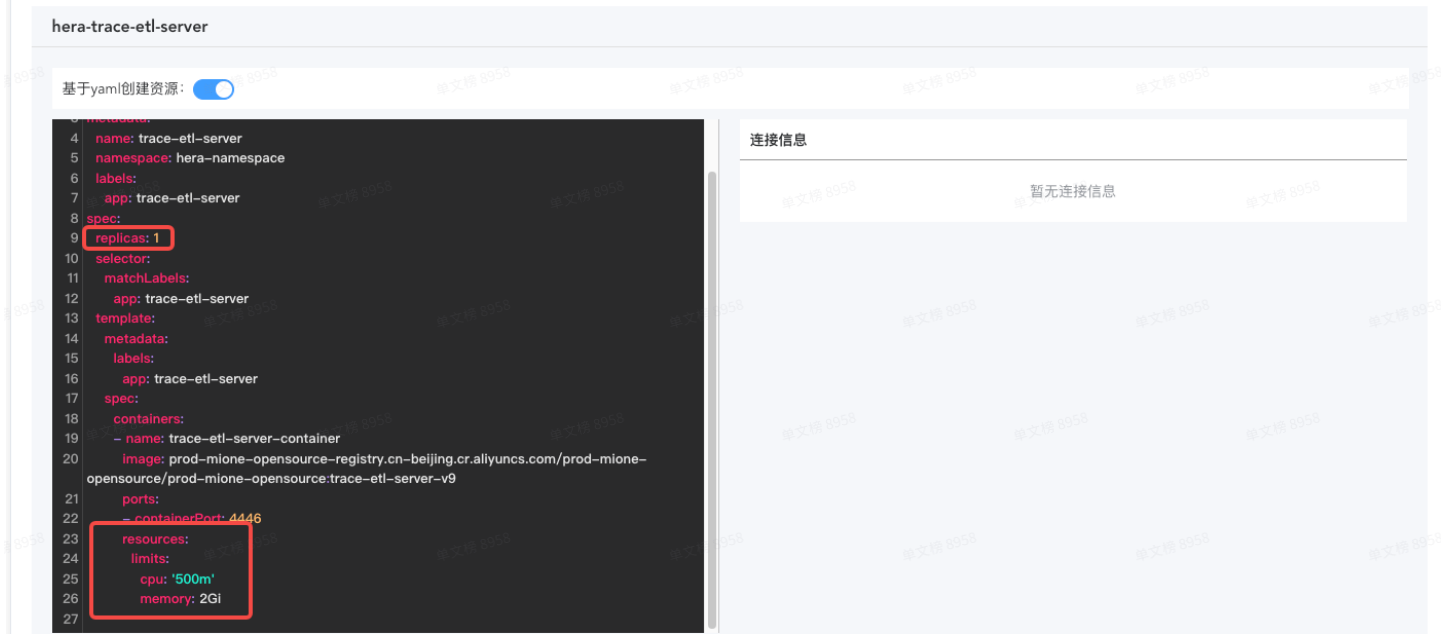


hera-trace-etl-server

需要注意的是：

1、需要根据访问量来修改pod副本数(replicas)与pod资源限制(resources)

2、服务的pod副本数需要与RocketMQ的queue size保持一致



hera-mimonitor

mimonitor是hera监控首页应用中心、指标监控、报警配置的后端服务，建议直接使用operator提供的基于yaml创建资源的方式部署。当然，也可以自行部署mimonitor服务（关闭给予yaml资源创建的开关），自己部署服务，需要调整前端对接的相应参数：如ip地址、端口号等。

注意点：在部署之前，先初始化数据库及对应的中间件资源。

1、mysql

在nacos配置mysql的数据库，在对应的数据库名下按照[hera-all/mimonitor/sql](#) 文件初始化数据库表。

2、rocketMq

按照nacos上的配置，在对应的rocket服务器上创建mq相应的topic和tag。

3、es

按照nacos上的配置，在对应的es服务器上创建es相应的索引。

4、使用operator自动创建资源，可以根据自己的实际流量情况调整副本数，replicas。实例中是一个副本；同样，可是根据需要在operator的yaml文件中调整k8s相关的资源，如 cpu、memory。

hera-mimonitor

基于yaml创建资源: ☒

连接信息

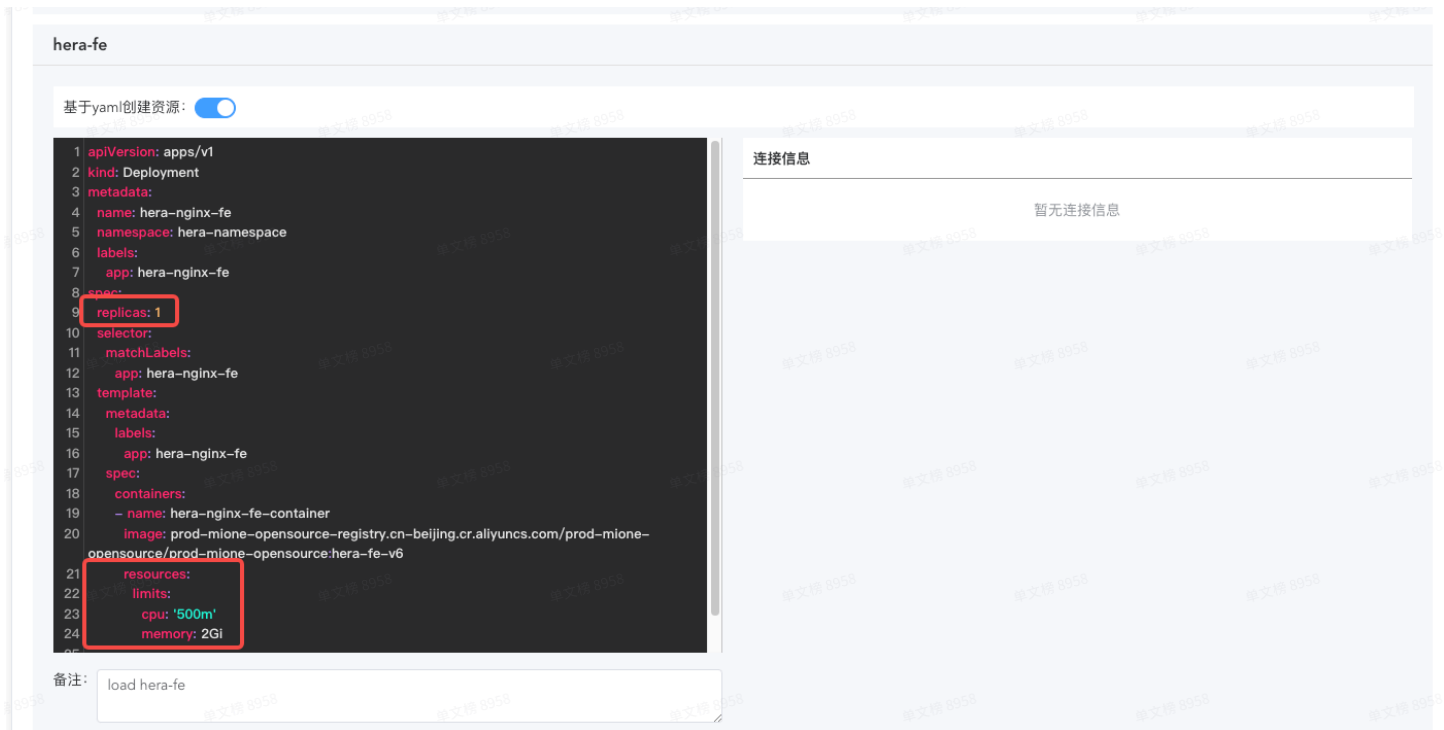
```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: mimonitor
5   namespace: hera-namespace
6   labels:
7     app: mimonitor
8 spec:
9   replicas: 1
10  selector:
11    matchLabels:
12      app: mimonitor
13  template:
14    metadata:
15      labels:
16        app: mimonitor
17    spec:
18      containers:
19        - name: mimonitor-container
20          image: prod-mione-opensource-registry.cn-beijing.cr.aliyuncs.com/prod-mione-opensource/prod-mione-opensource:mimonitor-v16
21          resources:
22            limits:
23              cpu: '500m'
24              memory: 2Gi
```

hera-fe

hera-fe是负责构建hera前端页面的yaml

需要注意的是:

- 1、需要根据访问量来修改pod副本数(replicas)与pod资源限制(resources)



hera-tpc-login

hera-tpc-login是负责构建tpclogin登陆服务后端的yaml

需要注意的是：

- 1、需要根据访问量来修改pod副本数(replicas)与pod资源限制(resources)
- 2、服务启动的配置像信息在上方的nacos配置中

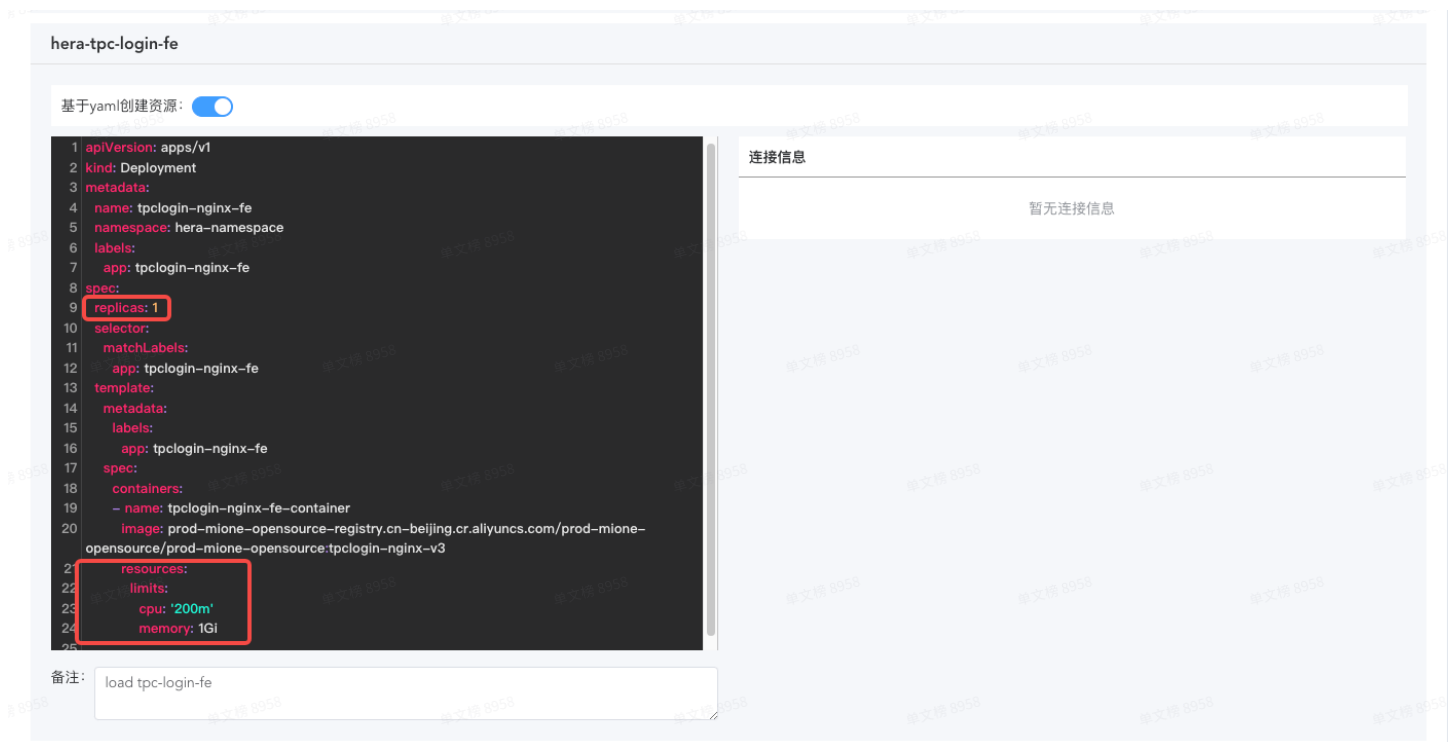
```
6 labels:
7   app: mi-tpclogin
8 spec:
9   replicas: 1
10  selector:
11    matchLabels:
12      app: mi-tpclogin
13  template:
14    metadata:
15      labels:
16        app: mi-tpclogin
17    spec:
18      containers:
19        - name: mi-tpclogin-container
20          image: prod-mione-opensource-registry.cn-beijing.cr.aliyuncs.com/prod-
mione-opensource/prod-mione-opensource:mi-tpclogin-v9
21      resources:
22        limits:
23          cpu: '500m'
24          memory: 2Gi
25 ---
26 apiVersion: v1
27 kind: Service
28 metadata:
29   name: mi-tpclogin
30   namespace: hera-namespace
```

hera-tpc-login-fe

hera-tpc-login-fe是负责构建tpclogin登录前端页面的yaml

需要注意的是：

- 1、需要根据访问量来修改pod副本数(replicas)与pod资源限制(resources)



hera-tpc

hera-tpc是负责构建tpc服务后端的yaml

需要注意的是：

- 1、 需要根据访问量来修改pod副本数(replicas)与pod资源限制(resources)
- 2、 服务相关配置在上方的nacos配置中配置

```
6 labels:
7   app: mi-tpc
8 spec:
9   replicas: 1
10  selector:
11    matchLabels:
12      app: mi-tpc
13  template:
14    metadata:
15      labels:
16        app: mi-tpc
17    spec:
18      containers:
19        - name: mi-tpc-container
20          image: prod-mione-opensource-registry.cn-beijing.cr.aliyuncs.com/prod-
mione-opensource/prod-mione-opensource:mi-tpc-v18
21      resources:
22        limits:
23          cpu: '500m'
24          memory: 2Gi
25 ---
26 apiVersion: v1
27 kind: Service
28 metadata:
29   name: mi-tpc
30   namespace: hera-namespace
```

hera-tpc-fe

hera-tpc-fe是负责构建tpc前端页面的yaml

需要注意的是：

- 1、需要根据访问量来修改pod副本数(replicas)与pod资源限制(resources)

hera-tpc-fe

基于yaml创建资源：☒

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: tpc-nginx-fe
5   namespace: hera-namespace
6   labels:
7     app: tpc-nginx-fe
8 spec:
9   replicas: 1
10  selector:
11    matchLabels:
12      app: tpc-nginx-fe
13  template:
14    metadata:
15      labels:
16        app: tpc-nginx-fe
17    spec:
18      containers:
19        - name: tpc-nginx-fe-container
20          image: prod-mione-opensource-registry.cn-beijing.cr.aliyuncs.com/prod-mione-opensource/prod-mione-opensource-nginx-v7
21      resources:
22        limits:
23          cpu: '200m'
24          memory: 1Gi
```

连接信息

暂无连接信息

备注: load tpc-fe

Hera-app

hera-app是负责负责hera系统中应用app相关逻辑操作，可以通过这个服务向外提供对应用的各种服务信息

需要注意的是：

- 1、需要根据访问量来修改pod副本数(replicas)与pod资源限制(resources)

hera-app

基于yaml创建资源：☒

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: hera-app
5   namespace: hera-namespace
6   labels:
7     app: hera-app
8 spec:
9   replicas: 1
10  selector:
11    matchLabels:
12      app: hera-app
13  template:
14    metadata:
15      labels:
16        app: hera-app
17    spec:
18      containers:
19        - name: hera-app-container
20          image: prod-mione-opensource-registry.cn-beijing.cr.aliyuncs.com/prod-mione-opensource/prod-mione-opensource-hera-app-v9
21      resources:
22        limits:
23          cpu: '300m'
24          memory: 1Gi
```

连接信息

暂无连接信息

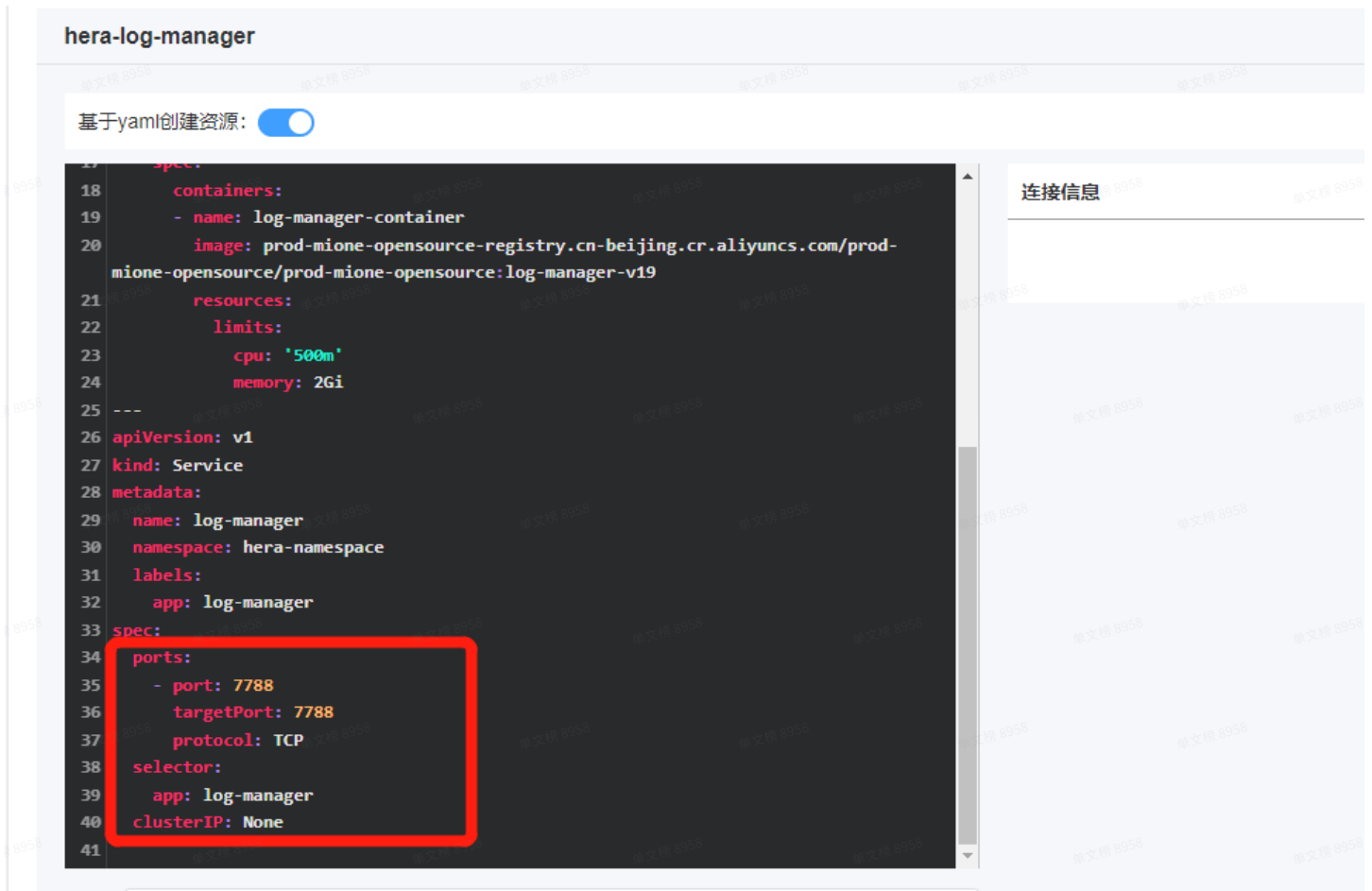
备注: load hera-app

Hera-log-manager

hera-log-manager主要负责页面应用日志的接入，以及各种元数据的配置下发

需要注意的是：

1、由于该服务需要对外提供http服务，因此需要开一个端口，端口默认来自与项目中的配置文件，默认为7788



hera-log-stream

hera-log-stream负责消费mq中的应用日志信息，然后负责应用日志解析，最后存入存储空间(ES)

需要注意的是：

1、该服务是StatefulSet类型服务

2、需要注入一个环境变量MONE_CONTAINER_S_POD_NAME，值为容器pod的名称

基于yaml创建资源: ☒

```
2 kind: StatefulSet
3 metadata:
4   name: log-stream
5   namespace: hera-namespace
6   labels:
7     app: log-stream
8 spec:
9   replicas: 1
10  selector:
11    matchLabels:
12      app: log-stream
13  template:
14    metadata:
15      labels:
16        app: log-stream
17    spec:
18      containers:
19        - name: log-stream-container
20          image: prod-mione-opensource-registry.cn-beijing.cr.aliyuncs.com/prod-
mione-opensource/prod-mione-opensource:log-stream-v7
21          env:
22            - name: MONE_CONTAINER_S_POD_NAME
23              valueFrom:
24                fieldRef:
25                  apiVersion: v1
26                  fieldPath: metadata.name
```

备注: load log-stream

连接信息

暂无连接

2.4.5 集群部署

保存配置

确保2.4.4步骤完毕后，点击保存配置，该步骤会完成:1. 整个配置的保持；2.nacos变量替换（nacos配置中有\${变量}配置的，会自动完成一轮替换，替换值来源于输入的连接信息、第二步生成的访问方式ip:port）

集群生效

确保"保存配置"已完成后，可点击"集群生效"进行整个hera集群的部署。