

用k8s部署微服务应用

以我们之前用docker部署过的eureka应用为例，首先添加配置文件eureka-app-deployment.yaml用于创建Deployment

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: eureka-app-deployment
5   labels:
6     app: eureka-app
7 spec:
8   replicas: 1
9   selector:
10    matchLabels:
11      app: eureka-app
12   template:
13     metadata:
14       labels:
15         app: eureka-app
16     spec:
17       containers:
18         - name: eureka-app
19           # 指定Docker Hub中的镜像地址
20           image: zhuge666/microservice-eureka-server:0.0.1
21           ports:
22             - containerPort: 8761
```

通过应用配置文件来创建Deployment

```
1 kubectl apply -f eureka-app-deployment.yaml
```

查看下deployment和pod信息

```
1 kubectl get deploy,pod
```

```
[root@k8s-master k8s]# kubectl get deploy,pod
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/eureka-app-deployment 1/1      1              1            2m17s
deployment.apps/my-tomcat             3/3      3              3            10d
deployment.apps/my-tomcat-yaml         2/2      2              2            8d
deployment.apps/nginx                 1/1      1              1            11d
deployment.apps/nginx-deployment       1/1      1              1            2d17h

NAME                                READY    STATUS    RESTARTS    AGE
pod/eureka-app-deployment-7cd4b6f4d4-1kzs4 1/1      Running   0            2m17s
pod/my-tomcat-685b8fd9c9-4ngsb             1/1      Running   0            10d
pod/my-tomcat-685b8fd9c9-lrwst             1/1      Running   0            10d
pod/my-tomcat-685b8fd9c9-q6xzh             1/1      Running   0            10d
pod/my-tomcat-yaml-685b8fd9c9-8glxt         1/1      Running   0            8d
pod/my-tomcat-yaml-685b8fd9c9-ltbbf         1/1      Running   0            8d
pod/nginx-deployment-7cf97748c4-7gg28       1/1      Running   0            2d17h
pod/nginx-f89759699-ngqjl                   1/1      Running   0            11d
```

我们可以通过kubectl logs命令来查看应用的启动日志

```
1 kubectl logs -f pod/eureka-app-deployment-7cd4b6f4d4-1kzs4
```

```

2021-06-06 08:30:25.304 INFO 1 --- [main] o.s.c.n.e.s.EurekaServiceRegistry : Registering application unknown with eureka with status of
2021-06-06 08:30:25.309 INFO 1 --- [Thread-13] o.s.c.n.e.s.server.EurekaServerBootstrap : Setting the eureka configuration..
2021-06-06 08:30:25.310 INFO 1 --- [Thread-13] o.s.c.n.e.s.server.EurekaServerBootstrap : Eureka data center value eureka.datacenter is not set, defaulting to default
2021-06-06 08:30:25.310 INFO 1 --- [Thread-13] o.s.c.n.e.s.server.EurekaServerBootstrap : Eureka environment value eureka.environment is not set, defaulting to test
2021-06-06 08:30:25.326 INFO 1 --- [Thread-13] o.s.c.n.e.s.server.EurekaServerBootstrap : isAws returned false
2021-06-06 08:30:25.327 INFO 1 --- [Thread-13] o.s.c.n.e.s.server.EurekaServerBootstrap : Initialized server context
2021-06-06 08:30:25.327 INFO 1 --- [Thread-13] c.n.e.r.PeerAwareInstanceRegistryImpl : Got 1 instances from neighboring DS node
2021-06-06 08:30:25.327 INFO 1 --- [Thread-13] c.n.e.r.PeerAwareInstanceRegistryImpl : Renew threshold is: 1
2021-06-06 08:30:25.328 INFO 1 --- [Thread-13] c.n.e.r.PeerAwareInstanceRegistryImpl : Changing status to UP
2021-06-06 08:30:25.338 INFO 1 --- [Thread-13] e.s.EurekaServerInitializerConfiguration : Started Eureka Server
2021-06-06 08:30:25.356 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8761 (http) with context path ''
2021-06-06 08:30:25.357 INFO 1 --- [main] .s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 8761
2021-06-06 08:30:25.361 INFO 1 --- [main] c.tuling.cloud.study.EurekaApplication : Started EurekaApplication in 9.52 seconds (JVM running for 10.135)
2021-06-06 08:31:25.330 INFO 1 --- [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 0ms

```

如果想要从外部访问应用，需要创建Service，添加配置文件eureka-app-service.yaml用于创建Service;

```

1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: eureka-app-service
5 spec:
6   type: NodePort
7   selector:
8     app: eureka-app
9   ports:
10    - name: http
11      protocol: TCP
12      port: 8761 #service的端口
13      targetPort: 8761 #pod的端口，一般与pod内部容器的服务端口一致

```

通过应用配置文件来创建Service

```
1 kubectl apply -f eureka-app-service.yaml
```

查看服务Service信息

```
1 [macro@linux-local k8s]$ kubectl get services
```


```

[root@k8s-master k8s]# kubectl get services
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
eureka-app-service  NodePort    10.106.87.91  <none>         8761:30558/TCP   58s
kubernetes           ClusterIP   10.96.0.1     <none>         443/TCP          17d
nginx                NodePort    10.109.128.56 <none>         80:30433/TCP     11d
test-service         NodePort    10.104.189.121 <none>         80:32080/TCP,8080:32088/TCP 8d
tomcat               NodePort    10.101.176.202 <none>         8080:32224/TCP   10d
tomcat-service-yaml NodePort    10.104.55.220 <none>         8080:31524/TCP   8d

```

此时就可以通过外网来访问了: <http://192.168.65.160:30558/>

← → ↻ ⚠ 不安全 | 192.168.65.160:30558


HOME LAST

System Status

Environment	test	Current time	2021-
Data center	default	Uptime	00:11
		Lease expiration enabled	false
		Renews threshold	1
		Renews (last min)	0

EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THA INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.

DS Replicas

localhost

用k8s部署电商项目微服务

以**product服务**为例，我们来创建对应的**deployment**和**service**，做之前需要把商品服务做成镜像推到docker镜像仓库里去，参考docker的第一节课。

```
1 docker login
2 docker tag mall/tulingmall-product:0.0.1 zhuge666/tulingmall-product:0.0.1
3 docker push zhuge666/tulingmall-product:0.0.1
```

新增文件tulingmall-product-deployment.yaml，内容如下：

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: tulingmall-product-deployment
5   labels:
6     app: tulingmall-product
7 spec:
8   replicas: 1
9   selector:
10    matchLabels:
11      app: tulingmall-product
12   template:
13     metadata:
14       labels:
15         app: tulingmall-product
16     spec:
17       containers:
18         - name: tulingmall-product
19           image: zhuge666/tulingmall-product:0.0.1
20           #imagePullPolicy: Always # 1)Always 总是拉取镜像， 2)IfNotPresent(默认该值) 本地有则使用
           #本地镜像， 3)Never 只使用本地镜像，从不拉取，即使本地没有镜像
21           ports:
22             - containerPort: 8866
23           env:
24             - name: TZ
25               value: Asia/Shanghai
26             - name: spring.cloud.nacos.config.server-addr
27               value: 192.168.65.42:8848
28             - name: LOG_FILE
29               value: /var/logs
30           volumeMounts:
31             - mountPath: /var/logs
32               name: log-volume
33       volumes:
34         - name: log-volume
35           hostPath:
36             path: /mydata/k8s-app/tulingmall-product/logs
37             type: DirectoryOrCreate
38   dnsPolicy: Default #继承Pod所在宿主机的DNS设置，使pod能访问外网
```

执行如下命令创建商品服务的deployment：

```
1 kubectl apply -f tulingmall-product-deployment.yaml
```

新增文件tulingmall-product-service.yaml, 内容如下:

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: tulingmall-product-service
5 spec:
6   type: NodePort
7   selector:
8     app: tulingmall-product
9   ports:
10    - name: http
11      protocol: TCP
12      port: 8866
13      targetPort: 8866
```

执行如下命令创建商品服务的service:

```
1 kubectl apply -f tulingmall-product-service.yaml
```

查看商品服务的对外暴露端口:

```
[root@k8s-master tuling-mall]# kubectl get svc|grep product
tulingmall-product-service NodePort 10.106.36.91 <none> 8866:30911/TCP 28m
```

访问下查询商品的接口, 如果有json数据返回, 代表服务正常:

```
1 http://192.168.65.210:30911/pms/productInfo/1
```

← → ↺ ⚠ 不安全 | 192.168.65.210:30911/pms/productInfo/1

```
{"code":404,"message":"产品不存在!","data":null}
```

用相同的方法部署下order, member, gateway, authcenter等服务, 这里不一一详述了, 附上每个服务k8s部署的yaml文件供大家参考。

authcenter服务

tulingmall-authcenter-deployment.yaml

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: tulingmall-authcenter-deployment
5   labels:
6     app: tulingmall-authcenter
7 spec:
8   replicas: 1
9   selector:
10    matchLabels:
11      app: tulingmall-authcenter
12   template:
13     metadata:
14       labels:
```

```

15     app: tulingmall-authcenter
16     spec:
17       containers:
18         - name: tulingmall-authcenter
19           image: zhuge666/tulingmall-authcenter:0.0.1
20       imagePullPolicy: Always
21       ports:
22         - containerPort: 9999
23       env:
24         - name: TZ
25           value: Asia/Shanghai
26         - name: spring.cloud.nacos.config.server-addr
27           value: 192.168.65.42:8848
28         - name: LOG_FILE
29           value: /var/logs
30       volumeMounts:
31         - mountPath: /var/logs
32           name: log-volume
33       volumes:
34         - name: log-volume
35           hostPath:
36             path: /mydata/k8s-app/tulingmall-authcenter/logs
37             type: DirectoryOrCreate
38     dnsPolicy: Default #继承Pod所在宿主机的DNS设置，使pod能访问外网

```

tulingmall-authcenter-service.yaml

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: tulingmall-authcenter-service
5  spec:
6    type: NodePort
7    selector:
8      app: tulingmall-authcenter
9    ports:
10     - name: http
11       protocol: TCP
12       port: 9999
13       targetPort: 9999

```

gateway服务

tulingmall-gateway-deployment.yaml

```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:

```

```

4   name: tulingmall-gateway-deployment
5   labels:
6     app: tulingmall-gateway
7   spec:
8     replicas: 1
9     selector:
10      matchLabels:
11        app: tulingmall-gateway
12     template:
13       metadata:
14         labels:
15           app: tulingmall-gateway
16       spec:
17         containers:
18           - name: tulingmall-gateway
19             image: zhuge666/tulingmall-gateway:0.0.1
20     imagePullPolicy: Always
21     ports:
22       - containerPort: 8888
23     env:
24       - name: TZ
25         value: Asia/Shanghai
26       - name: spring.cloud.nacos.config.server-addr
27         value: 192.168.65.42:8848
28       - name: LOG_FILE
29         value: /var/logs
30     volumeMounts:
31       - mountPath: /var/logs
32         name: log-volume
33     volumes:
34       - name: log-volume
35         hostPath:
36           path: /mydata/k8s-app/tulingmall-gateway/logs
37           type: DirectoryOrCreate
38     dnsPolicy: Default #继承Pod所在宿主机的DNS设置，使pod能访问外网

```

tulingmall-gateway-service.yaml

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: tulingmall-gateway-service
5  spec:
6    type: NodePort
7    selector:
8      app: tulingmall-gateway
9    ports:
10     - name: http

```

```
11 protocol: TCP
12 port: 8888
13 targetPort: 8888
```

order服务

tulingmall-order-deployment.yaml

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: tulingmall-order-deployment
5    labels:
6      app: tulingmall-order
7  spec:
8    replicas: 1
9    selector:
10     matchLabels:
11       app: tulingmall-order
12    template:
13     metadata:
14       labels:
15         app: tulingmall-order
16     spec:
17       containers:
18         - name: tulingmall-order
19           image: zhuge666/tulingmall-order:0.0.1
20     imagePullPolicy: Always
21     ports:
22       - containerPort: 8844
23     env:
24       - name: TZ
25         value: Asia/Shanghai
26       - name: spring.cloud.nacos.config.server-addr
27         value: 192.168.65.42:8848
28       - name: LOG_FILE
29         value: /var/logs
30       - name: JAVA_TOOL_OPTIONS
31         value: -Xms1G -Xmx1G -Xmn512M -Xss512K -XX:MetaspaceSize=256M -XX:MaxMetaspaceSize=256M -javaagent:/agent/skywalking-agent.jar -DSW_AGENT_NAME=tulingmall-order -DSW_AGENT_COLLECTOR_BACKEND_SERVICES=192.168.65.204:11800
32     volumeMounts:
33       - mountPath: /var/logs
34         name: log-volume
35     volumes:
36       - name: log-volume
37         hostPath:
38           path: /mydata/k8s-app/tulingmall-order/logs
```

```
39         type: DirectoryOrCreate
40     dnsPolicy: Default #继承Pod所在宿主机的DNS设置，使pod能访问外网
```

tulingmall-order-service.yaml

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: tulingmall-order-service
5  spec:
6    type: NodePort
7    selector:
8      app: tulingmall-order
9    ports:
10     - name: http
11       protocol: TCP
12       port: 8844
13       targetPort: 8844
```

member服务

tulingmall-member-deployment.yaml

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: tulingmall-member-deployment
5    labels:
6      app: tulingmall-member
7  spec:
8    replicas: 1
9    selector:
10     matchLabels:
11       app: tulingmall-member
12    template:
13     metadata:
14       labels:
15         app: tulingmall-member
16     spec:
17       containers:
18         - name: tulingmall-member
19           image: zhuge666/tulingmall-member:0.0.1
20       imagePullPolicy: Always
21       ports:
22         - containerPort: 8877
23       env:
24         - name: TZ
25           value: Asia/Shanghai
26         - name: spring.cloud.nacos.config.server-addr
```



```

27   value: 192.168.65.42:8848
28       - name: LOG_FILE
29         value: /var/logs
30   - name: JAVA_TOOL_OPTIONS
31     value: -Xmx1g -Xms1g -XX:MaxMetaspaceSize=256m -javaagent:/agent/skywalking-agent.jar
32     -DSW_AGENT_NAME=tulingmall-order -DSW_AGENT_COLLECTOR_BACKEND_SERVICES=192.168.65.204:11800
33
34     volumeMounts:
35       - mountPath: /var/logs
36         name: log-volume
37
38     volumes:
39       - name: log-volume
40         hostPath:
41           path: /mydata/k8s-app/tulingmall-member/logs
42           type: DirectoryOrCreate
43
44   dnsPolicy: Default #继承Pod所在宿主机的DNS设置，使pod能访问外网

```

tulingmall-member-service.yaml

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: tulingmall-member-service
5  spec:
6    type: NodePort
7    selector:
8      app: tulingmall-member
9    ports:
10     - name: http
11       protocol: TCP
12       port: 8877
13       targetPort: 8877

```

创建网关的Ingress(相当于Nginx)

最后，我们来创建网关服务的Ingress，创建文件tulingmall-gateway-ingress.yaml，内容如下：

```

1  apiVersion: networking.k8s.io/v1beta1
2  kind: Ingress
3  metadata:
4    name: tulingmall-gateway-ingress
5  spec:
6    rules:
7     - host: gateway.tuling.com #转发域名
8       http:
9         paths:
10          - path: /
11            backend:
12              serviceName: tulingmall-gateway-service
13              servicePort: 8888 #service的端口

```

```
1 kubectl apply -f tulingmall-gateway-ingress.yaml
```

```
1 kubectl get ing
```

在访问机器配置host, win10客户机在目录: C:\Windows\System32\drivers\etc, 在host里增加如下host(ingress部署的机器ip对应访问的域名)

```
1 192.168.65.203 gateway.tuling.com
2 或者
3 192.168.65.210 gateway.tuling.com
```

[illegible]

1 文档: [Kubernetes电商微服务部署实战](#)

2 链接: <http://note.youdao.com/noteshare?id=dbaa1b80aafc1769a5866e693b44d08e&sub=DD4D0195FDE04F91B926AC9E064D9394>