Troubleshooting Cloud Run for Anthos on Google Cloud

This page provides troubleshooting strategies as well as solutions for some common errors.

When troubleshooting Cloud Run for Anthos on Google Cloud, first confirm that you can run your container image locally (https://cloud.google.com/run/docs/testing/local).

If your application is not running locally, you will need to diagnose and fix it. You should use Stackdriver logging (https://cloud.google.com/run/docs/logging) to help debug a deployed project.

When troubleshooting Cloud Run for Anthos on Google Cloud, consult the following sections for possible solutions to the problem.

Checking command line output

If you use the gcloud command line, check your command output to see if it succeeded or not. For example if your deployment terminated unsuccessfully, there should be an error message describing the reason for the failure.

Deployment failures are most likely due to either a misconfigured manifest or an incorrect command. For example, the following output says that you must configure route traffic percent to sum to 100

```
Error from server (InternalError): error when applying patch:</p><pre>{"metadata":{"
to:
&{0xc421d98240 0xc421e77490 default route-example STDIN 0xc421db0488 264682 false}
for: "STDIN": Internal error occurred: admission webhook "webhook.knative.dev" denie
ERROR: Non-zero return code '1' from command: Process exited with status 1
```

Checking logs for your service

You can use Stackdriver Logging or the Cloud Run page in the Cloud Console to check request logs and container logs. For complete details, read Logging and viewing logs (https://cloud.google.com/run/docs/logging).

If you use Stackdriver Logging, the resource you need to filter on is **Kubernetes Container**.

Checking Route status

Run the following command to get the status of the Route object you used to deploy your application:

```
kubectl get route
```

The conditions in `status` provide the reason for the failure.

Checking Istio routing

Compare your Istio Route object's configuration, obtained by checking Route status, to the Istio RouteRule object's configuration.

Enter the following, replacing with the appropriate value:

```
kubectl get routerule  -o yaml
```

If you don't know the name of your route rule, use `kubectl get routerule` to find it.

The command returns the configuration of your route rule. Compare the domains between your route and route rule; they should match.

Checking ingress status

Check ingress status using:

```
kubectl get ingress
```

The command returns the status of the ingress. You can see the name, age, domains, and IP address.

Checking Revision status

If you configure your Route with Configuration, run the following command to get the name of the Revision created for you deployment, and look up the configuration name in the Route's `.yaml` file:

```
kubectl get configuration  -o jsonpath="{.status.latestCreatedRevisionName}"
```

If you configure your Route with Revision directly, look up the revision name in the Route yaml file.

Then run

```
kubectl get revision  -o yaml
```

A ready Revision should has the following condition in status:

```
conditions:
  - reason: ServiceReady
    status: "True"
    type: Ready
```

If you see this condition, check the following to continue debugging:

- Check Pod status
- Check application logs
- Check Istio routing

If you see other conditions, to debug further:

Checking Pod status

To get the Pods for all your deployments:

```
kubectl get pods
```

This should list all Pods with brief status. For example:

```
NAME                                                  READY     STATUS
configuration-example-00001-deployment-659747ff99-9bvr4   2/2       Running
configuration-example-00002-deployment-5f475b7849-gxcht   1/2       CrashLoopBackOff
```

Choose one and use the following command to see detailed information for its status. Some useful fields are conditions and containerStatuses:

```
kubectl get pod  -o yaml
```

Checking Build status

If you are using Build to deploy, run the following command to get the Build for your Revision:

```
kubectl get build $(kubectl get revision  -o jsonpath="{.spec.buildName}") -o yaml
```

The conditions in status describe the failure.

EXTERNAL-IP is <pending> for a long time

Note: For cluster versions `1.15.3-gke.19` and greater, `1.14.3-gke.12` and greater, and `1.13.10-gke.8` and greater, `istio-ingress` is the Kubernetes service created by Istio as the shared gateway for all traffic incoming to the cluster. For all other cluster versions, `istio-ingressgateway` is the service for this.

Sometimes, you may not get an external IP address immediately after you create a cluster, but instead see the external IP as `pending`. For example you could see this by invoking the command:

To get the external IP for the Istio ingress gateway:

```
kubectl get svc ISTIO-GATEWAY -n NAMESPACE
```

Replace ***ISTIO-GATEWAY*** and ***NAMESPACE*** as follows:

Cluster version	ISTIO-GATEWAY	NAMESPACE
1.15.3-gke.19 and greater **1.14.3-gke.12** and greater **1.13.10-gke.8** and greater	**istio-ingress**	**gke-system**
All other versions	**istio-ingressgateway**	**istio-system**

where the resulting output looks something like this:

```
NAME              TYPE           CLUSTER-IP     EXTERNAL-IP  PORT(S)
ISTIO-GATEWAY     LoadBalancer   XX.XX.XXX.XX   pending      80:32380/TCP,443:32390/TC
```

The `EXTERNAL-IP` for the Load Balancer is the IP address you must use.

This may mean that you have run out of external IP address quota in Google Cloud. You can check the possible cause by invoking:

```
kubectl describe svc ISTIO-GATEWAY -n NAMESPACE
```

replacing ***ISTIO-GATEWAY*** and ***NAMESPACE*** with the values from the table above. This yields output similar to the following:

```
stem
Name:                     ISTIO-GATEWAY
Namespace:                NAMESPACE
Labels:                   addonmanager.kubernetes.io/mode=Reconcile
                          app=ISTIO-GATEWAY
                          chart=gateways-1.0.3
                          heritage=Tiller
                          istio=ingressgateway
                          k8s-app=istio
                          kubernetes.io/cluster-service=true
                          release=istio
Annotations:              kubectl.kubernetes.io/last-applied-configuration={"apiVers
Selector:                 app=ISTIO-GATEWAY,istio=ingressgateway
Type:                     LoadBalancer
IP:                       10.XX.XXX.XXX
LoadBalancer Ingress:     35.XXX.XXX.188
Port:                     http2  80/TCP
TargetPort:               80/TCP
NodePort:                 http2  31380/TCP
```

```
Endpoints:                XX.XX.1.6:80
Port:                     https  443/TCP
TargetPort:               443/TCP
NodePort:                 https  3XXX0/TCP
Endpoints:                XX.XX.1.6:XXX
Port:                     tcp  31400/TCP
TargetPort:               3XX00/TCP
NodePort:                 tcp  3XX00/TCP
Endpoints:                XX.XX.1.6:XXXXX
Port:                     tcp-pilot-grpc-tls  15011/TCP
TargetPort:               15011/TCP
NodePort:                 tcp-pilot-grpc-tls  32201/TCP
Endpoints:                XX.XX.1.6:XXXXX
Port:                     tcp-citadel-grpc-tls  8060/TCP
TargetPort:               8060/TCP
NodePort:                 tcp-citadel-grpc-tls  31187/TCP
Endpoints:                XX.XX.1.6:XXXX
Port:                     tcp-dns-tls  853/TCP
TargetPort:               XXX/TCP
NodePort:                 tcp-dns-tls  31219/TCP
Endpoints:                10.52.1.6:853
Port:                     http2-prometheus  15030/TCP
TargetPort:               XXXXX/TCP
NodePort:                 http2-prometheus  30944/TCP
Endpoints:                10.52.1.6:15030
Port:                     http2-grafana  15031/TCP
TargetPort:               XXXXX/TCP
NodePort:                 http2-grafana  31497/TCP
Endpoints:                XX.XX.1.6:XXXXX
Session Affinity:         None
External Traffic Policy:  Cluster
Events:
  Type    Reason                Age                  From                Message
  ----    ------                ----                 ----                -------
  Normal  EnsuringLoadBalancer  7s (x4318 over 15d)  service-controller  Ensuring lo
```

If your output contains an indication that the `IN_USE_ADDRESSES` quota was exceeded, you can request additional quota by navigating to the IAM & Admin page (https://console.cloud.google.com/iam-admin/iam) in the Google Cloud Console to request additional quota.

The gateway will continue to retry until an external IP address is assigned. This may take a few minutes

Last updated December 4, 2019.