### Load balancing for A/B testing

The user sets up a route with multiple services. Each service handles a version of the application.

Each service is assigned a weight and the portion of requests to each service is the service\_weight divided by the sum\_of\_weights. The weight for each service is distributed to the service’s endpoints so that the sum of the endpoint weights is the service weight.

The route can have up to four services. The weight for the service can be between 0 and 256. When the weight is 0, the service does not participate in load-balancing but continues to serve existing persistent connections. When the service weight is not 0, each endpoint has a minimum weight of 1. Because of this, a service with a lot of endpoints can end up with higher weight than desired. In this case, reduce the number of Pods to get the desired load balance weight.

Procedure

To set up the A/B environment:

1. Create the two applications and give them different names. Each creates a DeploymentConfig. The applications are versions of the same program; one is usually the current production version and the other the proposed new version:
2. $ oc new-app openshift/deployment-example --name=ab-example-a

$ oc new-app openshift/deployment-example --name=ab-example-b

Both applications are deployed and services are created.

1. Make the application available externally via a route. At this point, you can expose either. It can be convenient to expose the current production version first and later modify the route to add the new version.

$ oc expose svc/ab-example-a

Browse to the application at ab-example-<project>.<router\_domain> to verify that you see the desired version.

1. When you deploy the route, the router balances the traffic according to the weights specified for the services. At this point, there is a single service with default weight=1 so all requests go to it. Adding the other service as an alternateBackends and adjusting the weights brings the A/B setup to life. This can be done by the oc set route-backends command or by editing the route.

Setting the oc set route-backend to 0 means the service does not participate in load-balancing, but continues to serve existing persistent connections.

|  |  |
| --- | --- |
|  | Changes to the route just change the portion of traffic to the various services. You might have to scale the DeploymentConfigs to adjust the number of Pods to handle the anticipated loads. |

To edit the route, run:

$ oc edit route <route\_name>

...

metadata:

name: route-alternate-service

annotations:

haproxy.router.openshift.io/balance: roundrobin

spec:

host: ab-example.my-project.my-domain

to:

kind: Service

name: ab-example-a

weight: 10

alternateBackends:

- kind: Service

name: ab-example-b

weight: 15

...

#### Managing weights using the web console

Procedure

1. Navigate to the Route details page (Applications/Routes).
2. Select **Edit** from the Actions menu.
3. Check **Split traffic across multiple services**.
4. The **Service Weights** slider sets the percentage of traffic sent to each service.

For traffic split between more than two services, the relative weights are specified by integers between 0 and 256 for each service.

Traffic weightings are shown on the **Overview** in the expanded rows of the applications between which traffic is split.

#### Managing weights using the CLI

Procedure

1. To manage the services and corresponding weights load balanced by the route, use the oc set route-backends command:
2. $ oc set route-backends ROUTENAME \

[--zero|--equal] [--adjust] SERVICE=WEIGHT[%] [...] [options]

For example, the following sets ab-example-a as the primary service with weight=198 and ab-example-b as the first alternate service with a weight=2:

$ oc set route-backends ab-example ab-example-a=198 ab-example-b=2

This means 99% of traffic is sent to service ab-example-a and 1% to service ab-example-b.

This command does not scale the DeploymentConfigs. You might be required to do so to have enough Pods to handle the request load.

1. Run the command with no flags to verify the current configuration:
2. $ oc set route-backends ab-example
3. NAME KIND TO WEIGHT
4. routes/ab-example Service ab-example-a 198 (99%)

routes/ab-example Service ab-example-b 2 (1%)

1. To alter the weight of an individual service relative to itself or to the primary service, use the --adjust flag. Specifying a percentage adjusts the service relative to either the primary or the first alternate (if you specify the primary). If there are other backends, their weights are kept proportional to the changed.

For example:

$ oc set route-backends ab-example --adjust ab-example-a=200 ab-example-b=10

$ oc set route-backends ab-example --adjust ab-example-b=5%

$ oc set route-backends ab-example --adjust ab-example-b=+15%

The --equal flag sets the weight of all services to 100:

$ oc set route-backends ab-example --equal

The --zero flag sets the weight of all services to 0. All requests then return with a 503 error.

|  |  |
| --- | --- |
|  | Not all routers may support multiple or weighted backends. |

#### One service, multiple DeploymentConfigs

Procedure

1. Create a new application, adding a label ab-example=true that will be common to all shards:

$ oc new-app openshift/deployment-example --name=ab-example-a

The application is deployed and a service is created. This is the first shard.

1. Make the application available via a route (or use the service IP directly):

$ oc expose svc/ab-example-a --name=ab-example

1. Browse to the application at ab-example-<project>.<router\_domain> to verify you see the v1 image.
2. Create a second shard based on the same source image and label as the first shard, but with a different tagged version and unique environment variables:
3. $ oc new-app openshift/deployment-example:v2 \
4. --name=ab-example-b --labels=ab-example=true \

SUBTITLE="shard B" COLOR="red"

1. At this point, both sets of Pods are being served under the route. However, because both browsers (by leaving a connection open) and the router (by default, through a cookie) attempt to preserve your connection to a back-end server, you might not see both shards being returned to you.

To force your browser to one or the other shard:

* 1. Use the oc scale command to reduce replicas of ab-example-a to 0.

$ oc scale dc/ab-example-a --replicas=0

Refresh your browser to show v2 and shard B (in red).

* 1. Scale ab-example-a to 1 replica and ab-example-b to 0:

$ oc scale dc/ab-example-a --replicas=1; oc scale dc/ab-example-b --replicas=0

Refresh your browser to show v1 and shard A (in blue).

1. If you trigger a deployment on either shard, only the Pods in that shard are affected. You can trigger a deployment by changing the SUBTITLE environment variable in either DeploymentConfig:

$ oc edit dc/ab-example-a

or

$ oc edit dc/ab-example-b