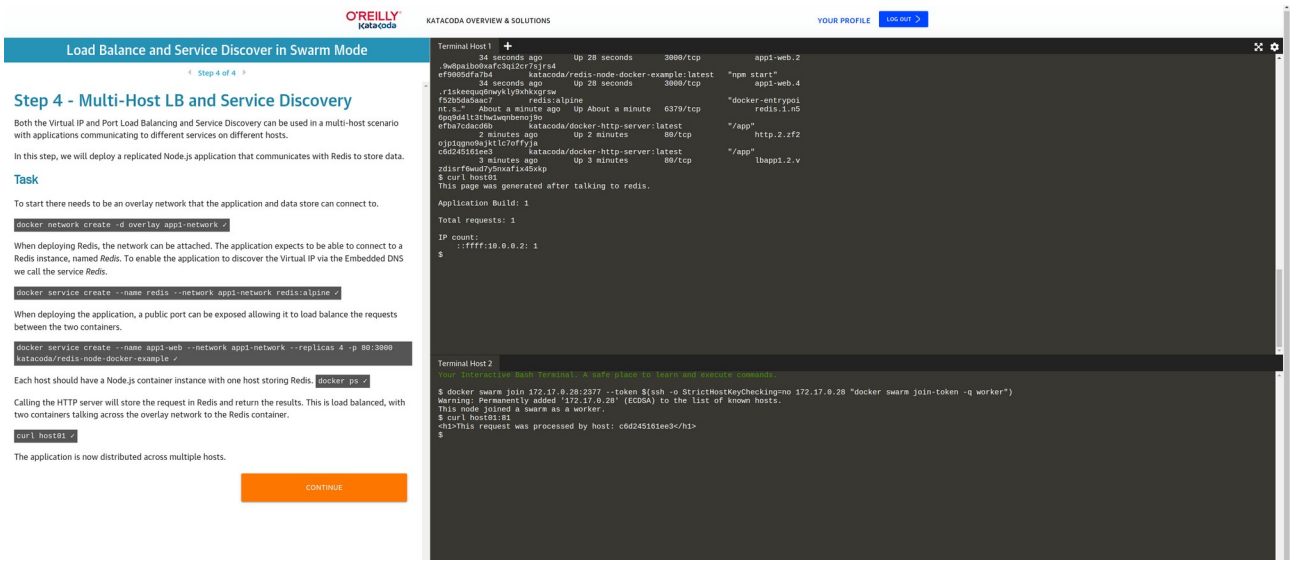
O'REILLY
katacoda
KATACODA OVERVIEW & SOLUTIONS
YOUR PROFILE
LOG OUT
Load Balance and Service Discover in Swarm Mode
Step 4 of 4
Step 4 - Multi-Host LB and Service Discovery
Both the Virtual IP and Port Load Balancing and Service Discovery can be used in a multi-host scenario with applications communicating to different services on different hosts.
In this step, we will deploy a replicated Node.js application that communicates with Redis to store data.
Task
To start there needs to be an overlay network that the application and data store can connect to.
docker network create -d overlay app1-network ✓
When deploying Redis, the network can be attached. The application expects to be able to connect to a Redis instance, named Redis. To enable the application to discover the Virtual IP via the Embedded DNS we call the service Redis.
docker service create --name redis --network app1-network redis:alpine ✓
When deploying the application, a public port can be exposed allowing it to load balance the requests between the two containers.
docker service create --name app1-web --network app1-network --replicas 4 -p 80:3000 katacoda/redis-node-docker-example ✓
Each host should have a Node.js container instance with one host storing Redis. docker ps ✓
Calling the HTTP server will store the request in Redis and return the results. This is load balanced, with two containers talking across the overlay network to the Redis container.
curl host01 ✓
The application is now distributed across multiple hosts.
CONTINUE
Terminal Host 1
34 seconds ago Up 28 seconds 3000/tcp app1-web.2
.9w0paibo0vafc3qi2cr7sjrs4
ef9005dfa7b4 katacoda/redis-node-docker-example:latest "npm start"
34 seconds ago Up 28 seconds 3000/tcp app1-web.4
.r1skeeguq6nwyk1y9xhkxgrsw
f52b5da5aac7 redis:alpine "docker-entrypoi
nt.s." About a minute ago Up About a minute 6379/tcp redis.1.n5
6pq9d4lt3thw1wqnbenoj8o
efba7cdacd6b katacoda/docker-http-server:latest "/app"
2 minutes ago Up 2 minutes 80/tcp http.2.zf2
ojp1qgno9ajktlc7offyja
c6d245161ee3 katacoda/docker-http-server:latest "/app"
3 minutes ago Up 3 minutes 80/tcp lbapp1.2.v
zdisrf6wud7y5nxafix45xkp
$ curl host01
This page was generated after talking to redis.
Application Build: 1
Total requests: 1
IP count:
::ffff:10.0.0.2: 1
$
Terminal Host 2
Your interactive Bash Terminal. A safe place to learn and execute commands.
$ docker swarm join 172.17.0.28:2377 --token $(ssh -o StrictHostKeyChecking=no 172.17.0.28 "docker swarm join-token -q worker")
Warning: Permanently added '172.17.0.28' (ECDSA) to the list of known hosts.
This node joined a swarm as a worker.
$ curl host01:81
<h1>This request was processed by host: c6d245161ee3</h1>
$