

Deployment Troubleshooting

```
robinson@HS-X6HT7MNFNX ~/Desktop/nd087-c3-deployment-roulette master ± k logs hello-world-6464549999-mhs8l
Ready to receive requests on 9000
* Serving Flask app 'main' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://10.100.1.250:9000/ (Press CTRL+C to quit)
Failed health check you want to ping /healthz
10.100.1.184 - - [15/Jun/2025 23:33:41] "GET /nginx_status HTTP/1.1" 500 -
Failed health check you want to ping /healthz
10.100.1.184 - - [15/Jun/2025 23:33:43] "GET /nginx_status HTTP/1.1" 500 -
Failed health check you want to ping /healthz
10.100.1.184 - - [15/Jun/2025 23:33:45] "GET /nginx_status HTTP/1.1" 500 -
```

```
robinson@HS-X6HT7MNFNX ~/Desktop/nd087-c3-deployment-roulette master ± k logs hello-world-844c8ccbb-x44wz
Ready to receive requests on 9000
* Serving Flask app 'main' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://10.100.1.111:9000/ (Press CTRL+C to quit)
Healthy!
10.100.1.184 - - [15/Jun/2025 23:36:36] "GET /healthz HTTP/1.1" 200 -
Healthy!
10.100.1.184 - - [15/Jun/2025 23:36:38] "GET /healthz HTTP/1.1" 200 -
Healthy!
10.100.1.184 - - [15/Jun/2025 23:36:40] "GET /healthz HTTP/1.1" 200 -
Healthy!
10.100.1.184 - - [15/Jun/2025 23:36:42] "GET /healthz HTTP/1.1" 200 -
Healthy!
10.100.1.184 - - [15/Jun/2025 23:36:44] "GET /healthz HTTP/1.1" 200 -
Healthy!
10.100.1.184 - - [15/Jun/2025 23:36:46] "GET /healthz HTTP/1.1" 200 -
Healthy!
10.100.1.184 - - [15/Jun/2025 23:36:48] "GET /healthz HTTP/1.1" 200 -
Healthy!
```

The hello-world application shows the log message `healthy!` when running

Canary deployment

```
1 #!/bin/bash
2
3 # There are 3 pods from version 1 and 1 from version 2
4 # The desired pods for the service is 4.
5 # For the v2 to take 50% of the traffic means that both
6 # v1 and v2 will have 2 replicas each.
7
8 echo "Starting deployment"
9
10 while [ "$(kubectl get po -n udacity | grep -c canary-v1)" -ne "$(kubectl get po -n udacity | grep -c canary-v2)" ]; do
11   PODS_V1=$(kubectl get po -n udacity | grep -c canary-v1)
12   PODS_V2=$(kubectl get po -n udacity | grep -c canary-v2)
13
14   echo "$((PODS_V1 - 1))"
15   echo "$((PODS_V2 - 1))"
16
17   kubectl scale deploy canary-v1 --replicas="$((PODS_V1 - 1))"
18   kubectl scale deploy canary-v2 --replicas="$((PODS_V2 + 1))"
19
20 until kubectl rollout status deployment/canary-v2 -n udacity; do
21   echo "Deploying..."
22   sleep 1
23 done
24 done
25
26 echo "Deployed"
```

creation of `canary.sh` script that will canary deploy the `canary-v2` successfully.

Curl the service 10 times and save the results to `canary.txt`

```
debug:~# for i in {1..10}; do curl 172.20.57.102 --silent; done
<html>
<h1>This is version 2</h1>
</html>
<html>
<h1>This is version 1</h1>
</html>
<html>
<h1>This is version 2</h1>
</html>
<html>
<h1>This is version 2</h1>
</html>
<html>
<h1>This is version 1</h1>
</html>
<html>
<h1>This is version 1</h1>
</html>
<html>
<h1>This is version 2</h1>
</html>
<html>
<h1>This is version 2</h1>
</html>
<html>
<h1>This is version 1</h1>
</html>
<html>
<h1>This is version 2</h1>
</html>
debug:~#
```

Provide the output of `kubectl get pods --all-namespaces` to show deployed services and save to a file named `canary2.txt`

```
robinson@HS-X6HT7MNFNX ~ k get po
NAME                                READY    STATUS    RESTARTS    AGE
blue-68f654b6f9-2w269              1/1     Running   0            116m
blue-68f654b6f9-5hxrw              1/1     Running   0            116m
blue-68f654b6f9-w6plz              1/1     Running   0            116m
canary-v1-58cb5c49d-5ttwc           1/1     Running   0            116m
canary-v1-58cb5c49d-w5ct6           1/1     Running   0            116m
canary-v2-55647dff9d-8gtl4         1/1     Running   0            94m
canary-v2-55647dff9d-kbft6         1/1     Running   0            7m11s
hello-world-844c8ccbb-x44wz        1/1     Running   0            112m
robinson@HS-X6HT7MNFNX ~
```

Blue Green

```
canary.sh x kubernetes_res... x dns.tf x green.yml x blue-green.sh x
1 kubectl apply -f starter/apps/blue-green/green.yml
2 until kubectl rollout status deployment/green -n udacity; do
3   echo "Deploying..."
4   sleep 1
5 done
```

```
</html>
[ec2-user@ip-10-100-10-159 ~]$ for i in {1..10}; do curl blue-green.udacityproject; done
<html>
<h1>This is version BLUE</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
```

blue-green.png: A screenshot that shows curling
blue-green.udacityproject.com returning results for both blue & green
deployments.

```
[ec2-user@ip-10-100-10-159 ~]$ for i in {1..10}; do curl blue-green.udacityproject; done
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
<html>
<h1>This is version GREEN</h1>
</html>
```

green-only.png: Absence of the the blue deployment

Node Elasticity

```
Host Port: 0/TCP
Requests:
  cpu: 250m
  memory: 200Mi
Readiness: http-get http://:80/ delay=0s timeout=1s period=10s #success=1 #failure=3
Environment: <none>
Mounts:
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-tx28z (ro)
Conditions:
  Type              Status
  PodScheduled      False
Volumes:
  kube-api-access-tx28z:
    Type:              Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:      kube-root-ca.crt
    ConfigMapOptional:  <nil>
    DownwardAPI:        true
Pod Class:            Burstable
Node-Selectors:
  <none>
Tolerations:
  node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
  node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type      Reason      Age      From      Message
  ----      -
  Warning   FailedScheduling  46s (x47 over 3h51m)  default-scheduler  0/2 nodes are available: 2 Insufficient cpu, preemption: 0/2 nodes are available: 2 No preemption victims found for incoming pod.
```

Solution is to increase the EKS cluster desired nodes

```
module "project_eks" {
  source      = "./modules/eks"
  name        = local.name
  account     = data.aws_caller_identity.current.account_id
  private_subnet_ids = module.vpc.private_subnet_ids
  vpc_id      = module.vpc.vpc_id
  nodes_desired_size = 4
  nodes_max_size  = 4
  nodes_min_size  = 1

  depends_on = [
    module.vpc,
  ]
}
```

```
robinson@HS-X6HT7MNFNX ~/Desktop/nd087-c3-deployment-roulette master ± k get po
NAME                                READY   STATUS    RESTARTS   AGE
bloaty-mcbloatface-c9f7f6685-4l8mx 1/1     Running   0           9m11s
bloaty-mcbloatface-c9f7f6685-4tmhx 1/1     Running   0           5m5s
bloaty-mcbloatface-c9f7f6685-6pvt6 1/1     Running   0           4m59s
bloaty-mcbloatface-c9f7f6685-79dhk 1/1     Running   0           9m10s
bloaty-mcbloatface-c9f7f6685-7pq24 1/1     Running   0           5m21s
bloaty-mcbloatface-c9f7f6685-8g7w2 1/1     Running   0           4m54s
bloaty-mcbloatface-c9f7f6685-97xjs 1/1     Running   0           5m15s
bloaty-mcbloatface-c9f7f6685-bfgrq 1/1     Running   0           4m30s
bloaty-mcbloatface-c9f7f6685-g47q9 1/1     Running   0           4m59s
bloaty-mcbloatface-c9f7f6685-hglzv 1/1     Running   0           5m9s
bloaty-mcbloatface-c9f7f6685-qklwk 1/1     Running   0           5m9s
bloaty-mcbloatface-c9f7f6685-r4nsw 1/1     Running   0           5m5s
bloaty-mcbloatface-c9f7f6685-tjvbj 1/1     Running   0           4m25s
bloaty-mcbloatface-c9f7f6685-w8jlt 1/1     Running   0           5m2s
bloaty-mcbloatface-c9f7f6685-xp7pm 1/1     Running   0           5m13s
bloaty-mcbloatface-c9f7f6685-zngqn 1/1     Running   0           4m27s
bloaty-mcbloatface-c9f7f6685-zvrrz 1/1     Running   0           5m2s
blue-68f654b6f9-2zntw             1/1     Running   0           9h
blue-68f654b6f9-lf428             1/1     Running   0           9h
blue-68f654b6f9-wj9g7             1/1     Running   0           9h
canary-v1-58cb5c49d-nnhgd          1/1     Running   0           9h
canary-v1-58cb5c49d-ql42h          1/1     Running   0           9h
canary-v1-58cb5c49d-wj7pw          1/1     Running   0           9h
green-7f5d485fc7-98jj4            1/1     Running   0           9h
green-7f5d485fc7-fcwk6            1/1     Running   0           9h
green-7f5d485fc7-n677p            1/1     Running   0           9h
hello-world-844c8ccbb-2d4dt       1/1     Running   0           9h
```

After adding more capacity all pods are now in **Running** state

```
robinson@HS-X6HT7MNFNX ~/Desktop/nd087-c3-deployment-roulette master ± kubectl autoscale deployment green --cpu-percent=20 --min=3 --max=6
horizontalpodautoscaler.autoscaling/green autoscaled
```

created a HPA for the green service

```
green-6677669959-5z8rb 1/1 Running 0 28s
green-6677669959-h8v5b 1/1 Running 0 29s
robinson@HS-X6HT7MNFNX ~/Desktop/nd087-c3-deployment-roulette [master] k get po
NAME READY STATUS RESTARTS AGE
bloaty-mcbloatface-c9f7f6685-4l8mx 1/1 Running 0 112m
bloaty-mcbloatface-c9f7f6685-4tmhx 1/1 Running 0 107m
bloaty-mcbloatface-c9f7f6685-6pvt6 1/1 Running 0 107m
bloaty-mcbloatface-c9f7f6685-79dnh 1/1 Running 0 112m
bloaty-mcbloatface-c9f7f6685-7pq24 1/1 Running 0 108m
bloaty-mcbloatface-c9f7f6685-8g7w2 1/1 Running 0 107m
bloaty-mcbloatface-c9f7f6685-97xjs 1/1 Running 0 108m
bloaty-mcbloatface-c9f7f6685-bfgrq 1/1 Running 0 107m
bloaty-mcbloatface-c9f7f6685-g47q9 1/1 Running 0 107m
bloaty-mcbloatface-c9f7f6685-hglzv 1/1 Running 0 108m
bloaty-mcbloatface-c9f7f6685-qklwk 1/1 Running 0 108m
bloaty-mcbloatface-c9f7f6685-r4nsw 1/1 Running 0 107m
bloaty-mcbloatface-c9f7f6685-tjvbj 1/1 Running 0 107m
bloaty-mcbloatface-c9f7f6685-w8jlt 1/1 Running 0 107m
bloaty-mcbloatface-c9f7f6685-xp7pm 1/1 Running 0 108m
bloaty-mcbloatface-c9f7f6685-zngqn 1/1 Running 0 107m
bloaty-mcbloatface-c9f7f6685-zvrrz 1/1 Running 0 107m
blue-68f654b6f9-2zntw 1/1 Running 0 11h
blue-68f654b6f9-lf428 1/1 Running 0 11h
blue-68f654b6f9-wj9g7 1/1 Running 0 11h
canary-v1-58cb5c49d-nnhgd 1/1 Running 0 11h
canary-v1-58cb5c49d-ql42h 1/1 Running 0 11h
canary-v1-58cb5c49d-wj7pw 1/1 Running 0 11h
green-6677669959-2m8nd 1/1 Running 0 3m7s
green-6677669959-5z8rb 1/1 Running 0 3m8s
green-6677669959-746qh 1/1 Running 0 118s
green-6677669959-82nwx 1/1 Running 0 102s
green-6677669959-h8v5b 1/1 Running 0 3m9s
green-6677669959-p29af 1/1 Running 0 102s
robinson@HS-X6HT7MNFNX
```

```
Summary:
Total: 67.6181 secs
Slowest: 17.9343 secs
Fastest: 0.0802 secs
Average: 0.6584 secs
Requests/sec: 1361.8540

Total data: 4235864 bytes
Size/request: 46 bytes

Response time histogram:
0.080 [1] |
1.866 [85665] |
3.651 [5462] |
5.436 [688] |
7.222 [226] |
9.007 [60] |
10.793 [60] |
12.578 [9] |
14.364 [2] |
16.149 [4] |
17.934 [7] |

Latency distribution:
30% in 0.0801 secs
25% in 0.0884 secs
50% in 0.5173 secs
75% in 0.5477 secs
90% in 1.5234 secs
95% in 2.3506 secs
99% in 3.8291 secs

Details (average, fastest, slowest):
DNS+lookup: 0.0039 secs, 0.0802 secs, 17.9343 secs
DNS+lookup: 0.0006 secs, 0.0800 secs, 0.0643 secs
req write: 0.0000 secs, 0.0000 secs, 0.0093 secs
resp wait: 0.6501 secs, 0.0801 secs, 17.9343 secs
resp read: 0.0016 secs, 0.0000 secs, 3.7819 secs

Status code distribution:
[200] 92084 responses

Error distribution:
[2] Get "http://a20b6e50683da46b6b27dc65baedba9a-4a4774ef132c51se.elb.us-east-2.amazonaws.com": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
```

The green pods were auto-scaled when hitting the load balancer public url.

Metrics

```
robinson@HS-X6HT7MNFNX ~/Desktop/nd087-c3-deployment-roulette [master] k top nodes
error: Metrics API not available
```

before installing the metrics server

```
robinson@HS-X6HT7MNFNX ~/Desktop/nd087-c3-deployment-roulette [master] kubectl top pods --sort-by=memory
NAME CPU(cores) MEMORY(bytes)
hello-world-844c8ccbb-2d4dt 2m 19Mi
blue-68f654b6f9-2zntw 1m 3Mi
bloaty-mcbloatface-c9f7f6685-7pq24 1m 3Mi
bloaty-mcbloatface-c9f7f6685-97xjs 1m 3Mi
bloaty-mcbloatface-c9f7f6685-qklwk 1m 3Mi
green-7f5d485fc7-fcwk6 1m 3Mi
green-7f5d485fc7-98jj4 1m 3Mi
bloaty-mcbloatface-c9f7f6685-g47q9 1m 3Mi
bloaty-mcbloatface-c9f7f6685-r4nsw 1m 3Mi
bloaty-mcbloatface-c9f7f6685-w8jlt 1m 3Mi
bloaty-mcbloatface-c9f7f6685-6pvt6 1m 3Mi
green-7f5d485fc7-n677p 1m 3Mi
bloaty-mcbloatface-c9f7f6685-tjvbj 1m 3Mi
bloaty-mcbloatface-c9f7f6685-zngqn 1m 3Mi
blue-68f654b6f9-wj9g7 1m 3Mi
bloaty-mcbloatface-c9f7f6685-xp7pm 1m 2Mi
bloaty-mcbloatface-c9f7f6685-bfgrq 1m 2Mi
blue-68f654b6f9-lf428 1m 2Mi
bloaty-mcbloatface-c9f7f6685-zvrrz 1m 2Mi
bloaty-mcbloatface-c9f7f6685-hglzv 1m 2Mi
canary-v1-58cb5c49d-ql42h 0m 2Mi
canary-v1-58cb5c49d-nnhgd 0m 2Mi
canary-v1-58cb5c49d-wj7pw 0m 2Mi
bloaty-mcbloatface-c9f7f6685-4l8mx 1m 2Mi
bloaty-mcbloatface-c9f7f6685-4tmhx 1m 2Mi
bloaty-mcbloatface-c9f7f6685-8g7w2 1m 2Mi
bloaty-mcbloatface-c9f7f6685-79dnh 1m 2Mi
```

The service that uses the most memory is hello-world

```

robinson@HS-X6HT7MNFNX ~/Desktop/nd087-c3-deployment-roulette master ± k get po
NAME                                READY    STATUS    RESTARTS   AGE
bloaty-mcbloatface-c9f7f6685-4l8mx 1/1      Running   0           21m
bloaty-mcbloatface-c9f7f6685-4tmhx 1/1      Running   0           17m
bloaty-mcbloatface-c9f7f6685-6pvt6 1/1      Running   0           16m
bloaty-mcbloatface-c9f7f6685-79dhk 1/1      Running   0           21m
bloaty-mcbloatface-c9f7f6685-7pq24 1/1      Running   0           17m
bloaty-mcbloatface-c9f7f6685-8g7w2 1/1      Running   0           16m
bloaty-mcbloatface-c9f7f6685-97xjs 1/1      Running   0           17m
bloaty-mcbloatface-c9f7f6685-bfgrq 1/1      Running   0           16m
bloaty-mcbloatface-c9f7f6685-g47q9 1/1      Running   0           16m
bloaty-mcbloatface-c9f7f6685-hglzv 1/1      Running   0           17m
bloaty-mcbloatface-c9f7f6685-qklwk 1/1      Running   0           17m
bloaty-mcbloatface-c9f7f6685-r4nsw 1/1      Running   0           17m
bloaty-mcbloatface-c9f7f6685-tjvbj 1/1      Running   0           16m
bloaty-mcbloatface-c9f7f6685-w8jlt 1/1      Running   0           17m
bloaty-mcbloatface-c9f7f6685-xp7pm 1/1      Running   0           17m
bloaty-mcbloatface-c9f7f6685-zngqn 1/1      Running   0           16m
bloaty-mcbloatface-c9f7f6685-zvrrz 1/1      Running   0           17m
blue-68f654b6f9-2zntw              1/1      Running   0           9h
blue-68f654b6f9-lf428              1/1      Running   0           9h
blue-68f654b6f9-wj9g7              1/1      Running   0           9h
canary-v1-58cb5c49d-nnhgd           1/1      Running   0           9h
canary-v1-58cb5c49d-ql42h           1/1      Running   0           9h
canary-v1-58cb5c49d-wj7pw           1/1      Running   0           9h
green-7f5d485fc7-98jj4             1/1      Running   0           9h
green-7f5d485fc7-fcwk6             1/1      Running   0           9h
green-7f5d485fc7-n677p             1/1      Running   0           9h
hello-world-844c8ccbb-2d4dt        1/1      Terminating 0           9h

```

after deleting the service, the pod is terminating

```

robinson@HS-X6HT7MNFNX ~/Desktop/nd087-c3-deployment-roulette master ± kubectl top pods --sort-by=memory
NAME                                CPU(cores)  MEMORY(bytes)
blue-68f654b6f9-2zntw              1m          3Mi
bloaty-mcbloatface-c9f7f6685-97xjs 1m          3Mi
bloaty-mcbloatface-c9f7f6685-7pq24 1m          3Mi
bloaty-mcbloatface-c9f7f6685-qklwk 1m          3Mi
green-7f5d485fc7-fcwk6             1m          3Mi
green-7f5d485fc7-98jj4             1m          3Mi
bloaty-mcbloatface-c9f7f6685-r4nsw 1m          3Mi
bloaty-mcbloatface-c9f7f6685-g47q9 1m          3Mi
green-7f5d485fc7-n677p             1m          3Mi
bloaty-mcbloatface-c9f7f6685-6pvt6 1m          3Mi
bloaty-mcbloatface-c9f7f6685-w8jlt 1m          3Mi
bloaty-mcbloatface-c9f7f6685-tjvbj 1m          3Mi
blue-68f654b6f9-wj9g7              1m          3Mi
bloaty-mcbloatface-c9f7f6685-xp7pm 1m          3Mi
bloaty-mcbloatface-c9f7f6685-zngqn 1m          3Mi
bloaty-mcbloatface-c9f7f6685-bfgrq 1m          2Mi
blue-68f654b6f9-lf428              0m          2Mi
bloaty-mcbloatface-c9f7f6685-zvrrz 1m          2Mi
bloaty-mcbloatface-c9f7f6685-hglzv 1m          2Mi
canary-v1-58cb5c49d-ql42h           0m          2Mi
bloaty-mcbloatface-c9f7f6685-4l8mx 1m          2Mi
canary-v1-58cb5c49d-nnhgd           0m          2Mi
canary-v1-58cb5c49d-wj7pw           0m          2Mi
bloaty-mcbloatface-c9f7f6685-8g7w2 1m          2Mi
bloaty-mcbloatface-c9f7f6685-4tmhx 1m          2Mi
bloaty-mcbloatface-c9f7f6685-79dhk 1m          2Mi

```

The pod is no longer present

VPC

