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

```
#!/bin/bash

# There are 3 pods from version 1 and 1 from version 2
# The desired pods for the service is 4.
# For the v2 to take 50% of the traffic means that both
# v1 and v2 will have 2 replicas each.

echo "Starting deployment"

while [ "$(kubectl get po -n udacity | grep -c canary-v1)" -ne "$(kubectl get po -n udacity | grep -c canary-v2)" ]; do
  PODS_V1=$(kubectl get po -n udacity | grep -c canary-v1)
  PODS_V2=$(kubectl get po -n udacity | grep -c canary-v2)

  echo "$((PODS_V1 - 1))"
  echo "$((PODS_V2 - 1))"

  kubectl scale deploy canary-v1 --replicas="$((PODS_V1 - 1))"
  kubectl scale deploy canary-v2 --replicas="$((PODS_V2 + 1))"

  until kubectl rollout status deployment/canary-v2 -n udacity; do
    echo "Deploying..."
    sleep 1
  done
done

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  kubernetes_res..  dns.tf  green.yml  blue-green.sh
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>
```

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>
```

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
QoS 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  60s (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-znggn  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

Metrics

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

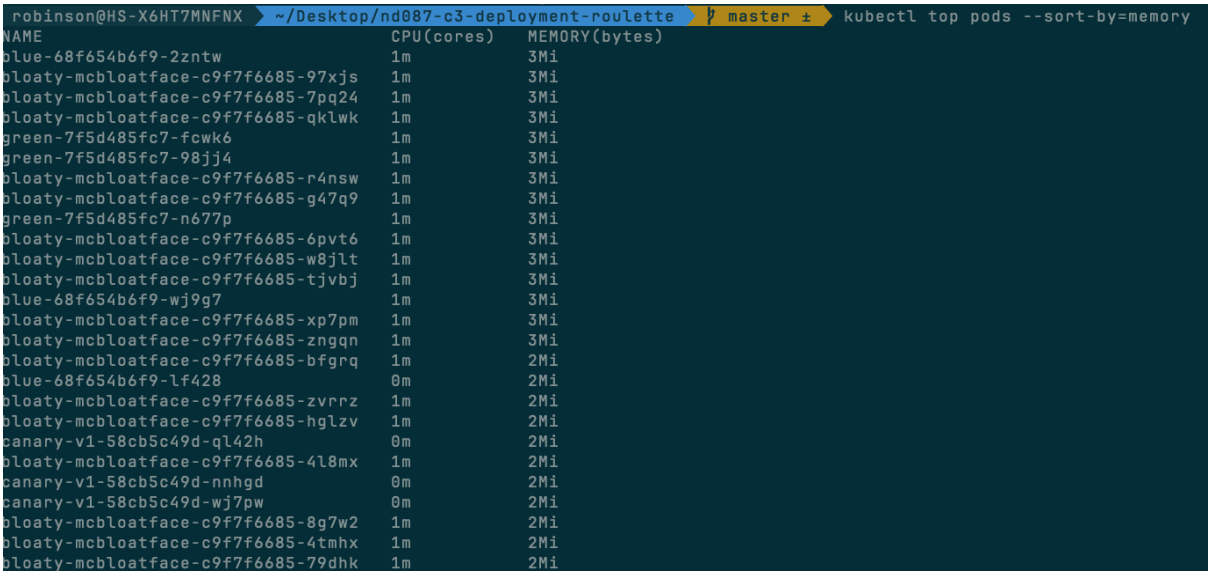
before installing the metrics server

```
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-c9f7f6685-79dhk  1m           2Mi
```

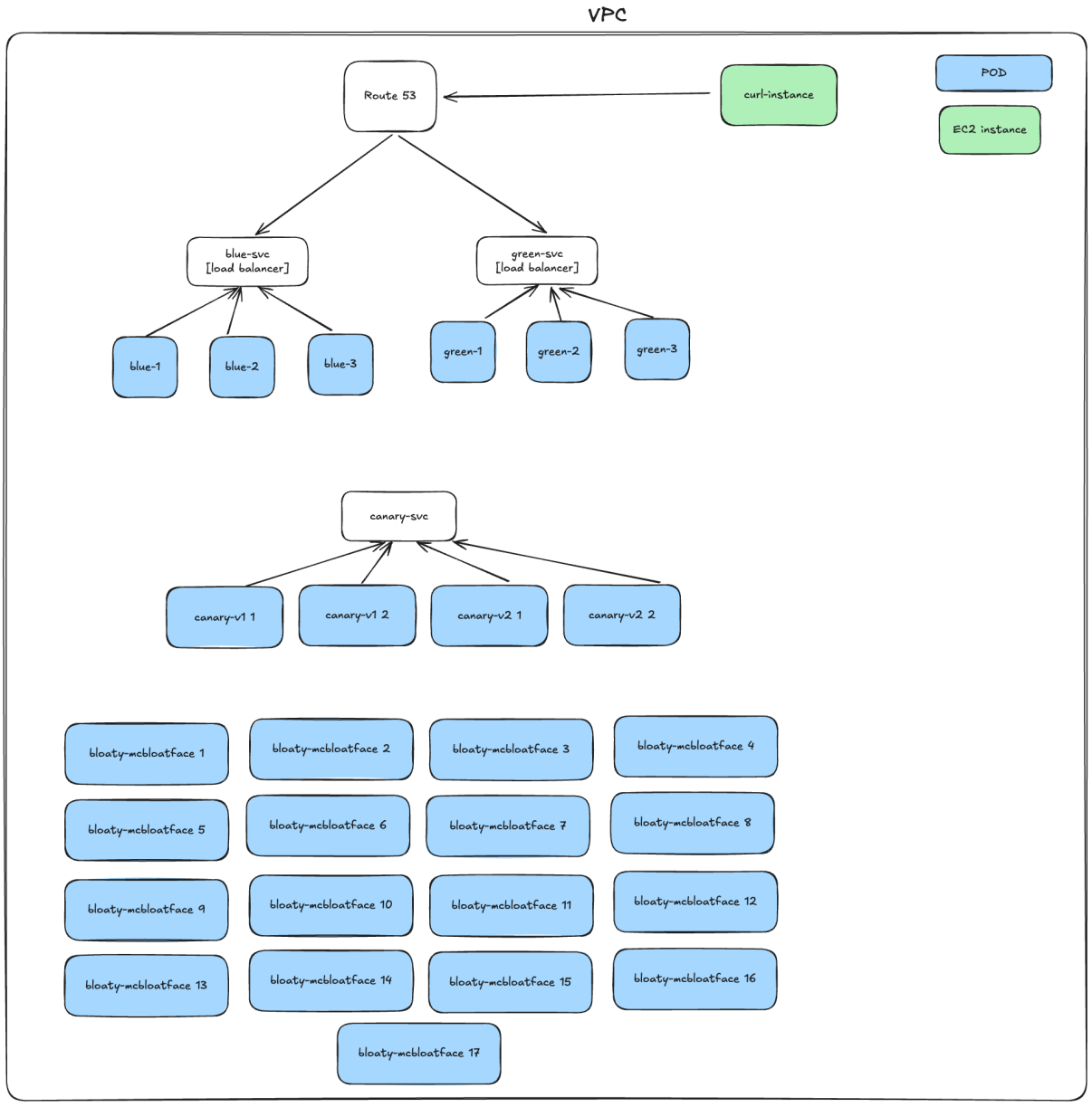
The service that uses the most memory is hello-world

```
✘ 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



The pod is no longer present


VPC
Route 53
curl-instance
POD
EC2 instance
blue-svc
[load balancer]
green-svc
[load balancer]
blue-1
blue-2
blue-3
green-1
green-2
green-3
canary-svc
canary-v1 1
canary-v1 2
canary-v2 1
canary-v2 2
bloaty-mcbloatface 1
bloaty-mcbloatface 2
bloaty-mcbloatface 3
bloaty-mcbloatface 4
bloaty-mcbloatface 5
bloaty-mcbloatface 6
bloaty-mcbloatface 7
bloaty-mcbloatface 8
bloaty-mcbloatface 9
bloaty-mcbloatface 10
bloaty-mcbloatface 11
bloaty-mcbloatface 12
bloaty-mcbloatface 13
bloaty-mcbloatface 14
bloaty-mcbloatface 15
bloaty-mcbloatface 16
bloaty-mcbloatface 17