Blue–Green Deployment on AWS using kOps

1. Project Overview

This project demonstrates a **production-style Blue–Green deployment strategy** on **AWS using Kubernetes (kOps)**. The setup uses **two separate Deployments (Blue and Green)** and a **single Service of type LoadBalancer** to control live traffic. Traffic switching and rollback are achieved instantly by updating the Service selector, without downtime.

---

2. Objectives

- Create a Kubernetes cluster on AWS using **kOps**
- Deploy two application versions (Blue and Green)
- Expose the application using **Service type LoadBalancer**
- Implement **Blue–Green deployment** with instant rollback
- Debug real-world issues related to ports, selectors, ELB, and networking

---

3. Architecture

Components Used

- **AWS EC2** – Worker and master nodes
- **AWS S3** – kOps state store
- **AWS ELB** – External LoadBalancer
- **kOps** – Kubernetes cluster lifecycle management
- **Kubernetes Deployments** – Blue and Green versions
- **Kubernetes Service** – Traffic routing

Traffic Flow

Client → AWS ELB (Service LoadBalancer) → NodePort → Pod (Blue or Green)

---

4. Cluster Setup

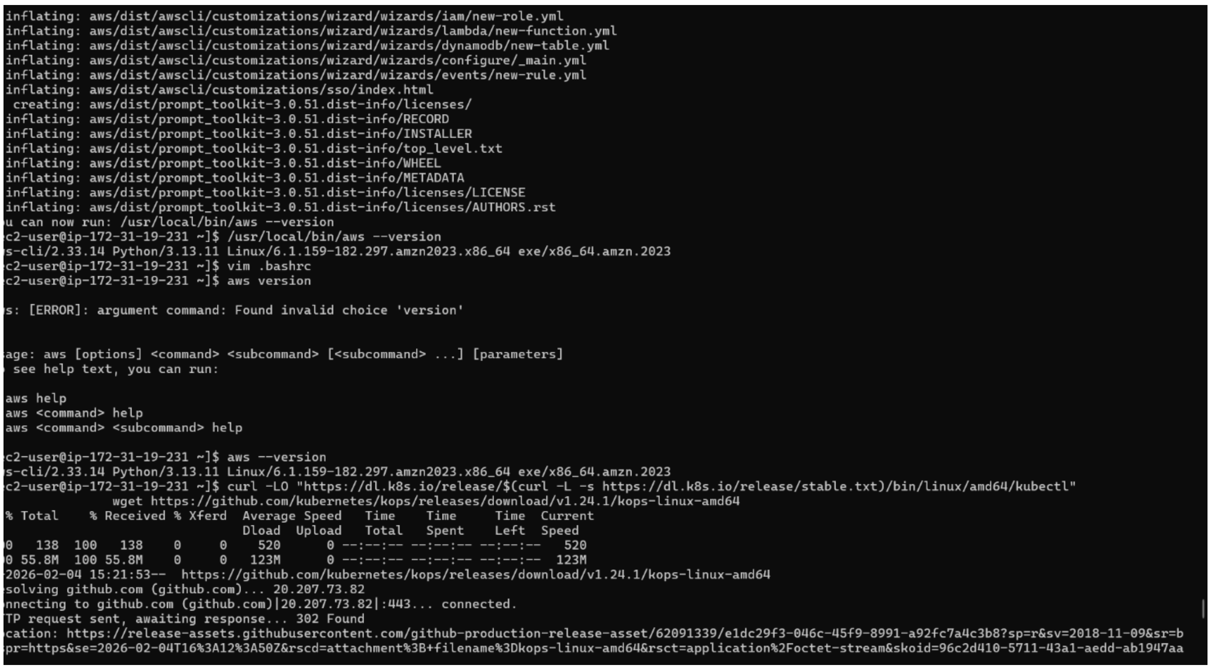
S3 State Store

An S3 bucket is created and versioning is enabled. This bucket stores all kOps cluster state.

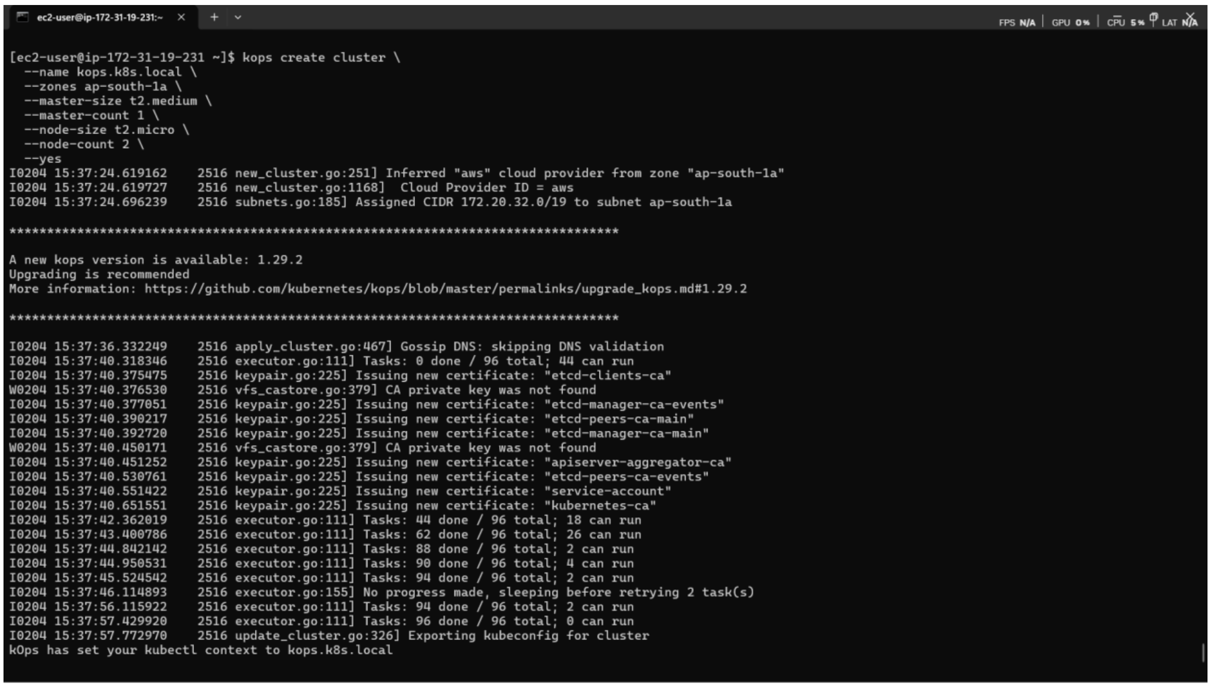
Cluster Creation

- Cluster name: kops.k8s.local
- Region: ap-south-1
- Networking: Public topology

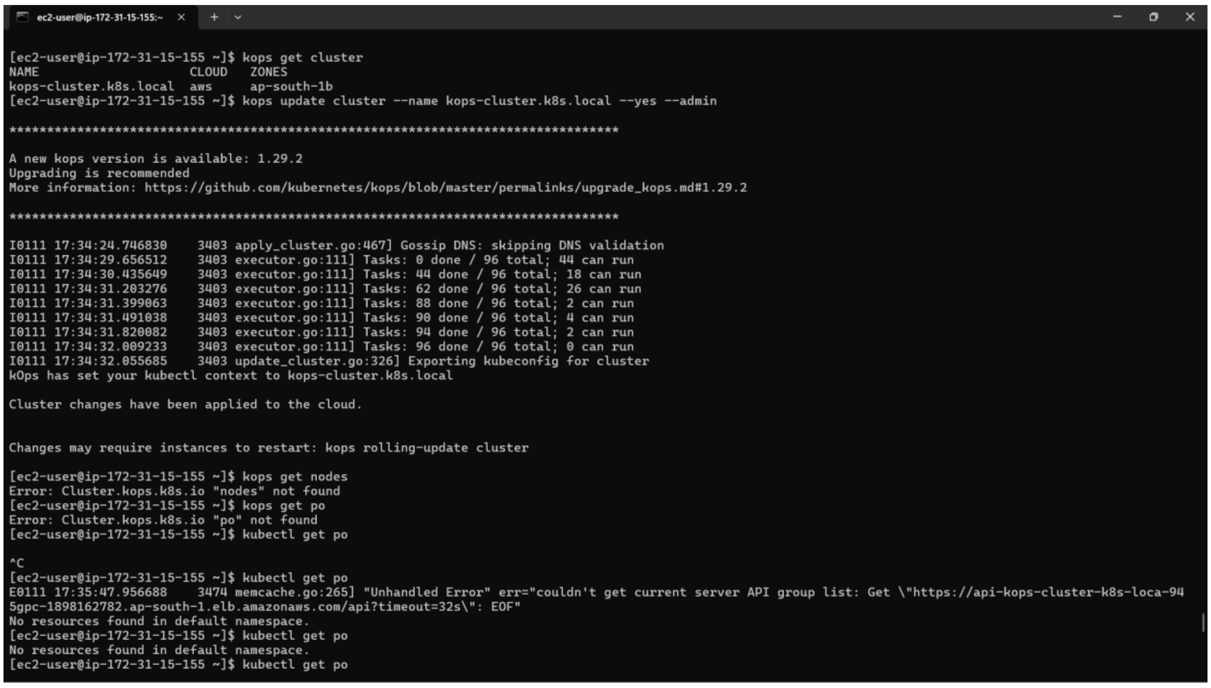
- Node size: t2.micro
- Master size: t2.medium



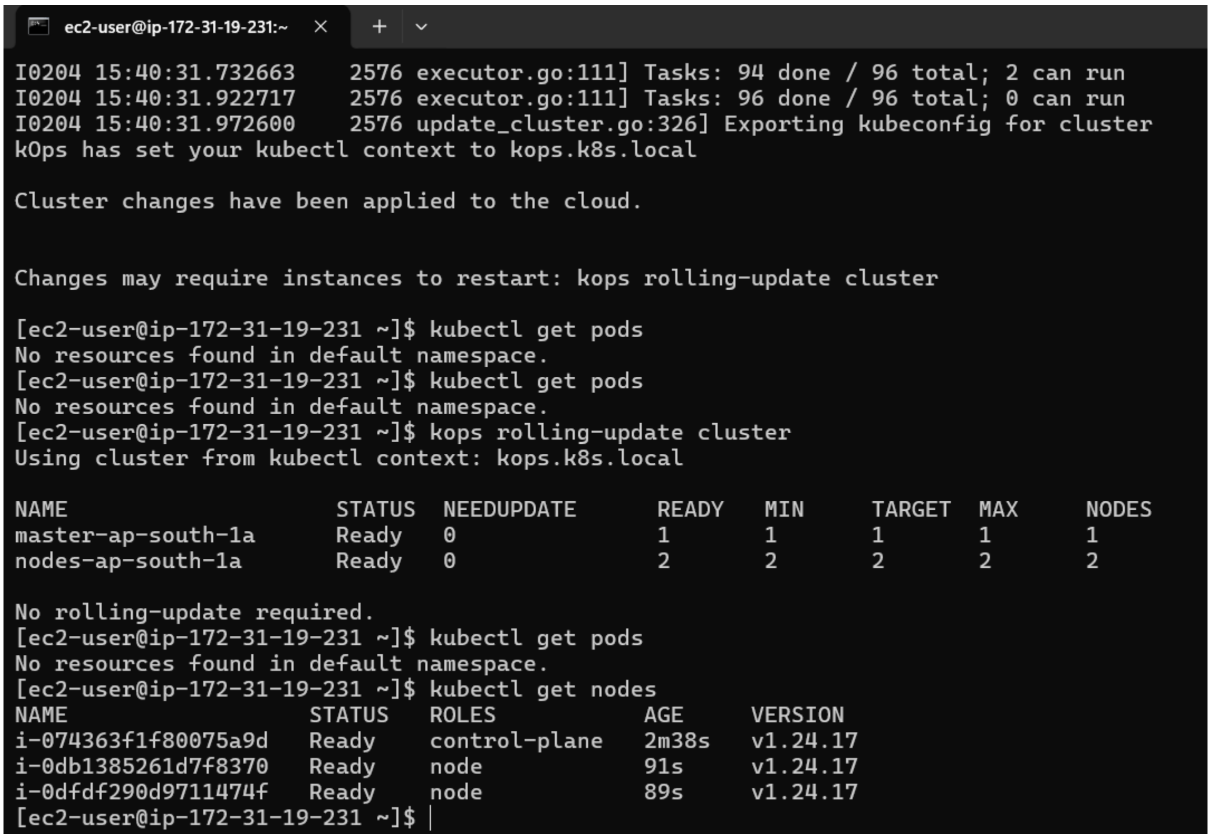
```
inflating: aws/dist/awscli/customizations/wizard/wizards/iam/new-role.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/lambda/new-function.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/dynamodb/new-table.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/configure/_main.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/events/new-rule.yml
inflating: aws/dist/awscli/customizations/sso/index.html
 creating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/RECORD
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/INSTALLER
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/top_level.txt
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/WHEEL
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/METADATA
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/LICENSE
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/AUTHORS.rst
u can now run: /usr/local/bin/aws --version
c2-user@ip-172-31-19-231 ~]$ /usr/local/bin/aws --version
s-cli/2.33.14 Python/3.13.11 Linux/6.1.159-182.297.amzn2023.x86_64 exe/x86_64.amzn.2023
c2-user@ip-172-31-19-231 ~]$ vim .bashrc
c2-user@ip-172-31-19-231 ~]$ aws version

s: [ERROR]: argument command: Found invalid choice 'version'


age: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
 see help text, you can run:

aws help
aws <command> help
aws <command> <subcommand> help

c2-user@ip-172-31-19-231 ~]$ aws --version
s-cli/2.33.14 Python/3.13.11 Linux/6.1.159-182.297.amzn2023.x86_64 exe/x86_64.amzn.2023
c2-user@ip-172-31-19-231 ~]$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
              wget https://github.com/kubernetes/kops/releases/download/v1.24.1/kops-linux-amd64
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                               Dload  Upload   Total   Spent    Left  Speed
0   138  100   138    0     0    520      0 --:--:-- --:--:-- --:--:--   520
0 55.8M  100 55.8M    0     0   123M      0 --:--:-- --:--:-- --:--:--  123M
2026-02-04 15:21:53--  https://github.com/kubernetes/kops/releases/download/v1.24.1/kops-linux-amd64
solving github.com (github.com)... 20.207.73.82
nnecting to github.com (github.com)|20.207.73.82|:443... connected.
TP request sent, awaiting response... 302 Found
cation: https://release-assets.githubusercontent.com/github-production-release-asset/62091339/e1dc29f3-046c-45f9-8991-a92fc7a4c3b8?sp=r&sv=2018-11-09&sr=b
pr=https&se=2026-02-04T16%3A12%3A50Z&rscd=attachment%3B+filename%3Dkops-linux-amd64&rsct=application%2Foctet-stream&skoid=96c2d410-5711-43a1-aedd-ab1947aa
```


ec2-user@ip-172-31-19-231:~
FPS N/A | GPU 0% | CPU 5% LAT N/A

```
[ec2-user@ip-172-31-19-231 ~]$ kops create cluster \
  --name kops.k8s.local \
  --zones ap-south-1a \
  --master-size t2.medium \
  --master-count 1 \
  --node-size t2.micro \
  --node-count 2 \
  --yes
I0204 15:37:24.619162    2516 new_cluster.go:251] Inferred "aws" cloud provider from zone "ap-south-1a"
I0204 15:37:24.619727    2516 new_cluster.go:1168]  Cloud Provider ID = aws
I0204 15:37:24.696239    2516 subnets.go:185] Assigned CIDR 172.20.32.0/19 to subnet ap-south-1a

*********************************************************************************

A new kops version is available: 1.29.2
Upgrading is recommended
More information: https://github.com/kubernetes/kops/blob/master/permalinks/upgrade_kops.md#1.29.2

*********************************************************************************

I0204 15:37:36.332249    2516 apply_cluster.go:467] Gossip DNS: skipping DNS validation
I0204 15:37:40.318346    2516 executor.go:111] Tasks: 0 done / 96 total; 44 can run
I0204 15:37:40.375475    2516 keypair.go:225] Issuing new certificate: "etcd-clients-ca"
W0204 15:37:40.376530    2516 vfs_castore.go:379] CA private key was not found
I0204 15:37:40.377051    2516 keypair.go:225] Issuing new certificate: "etcd-manager-ca-events"
I0204 15:37:40.390217    2516 keypair.go:225] Issuing new certificate: "etcd-peers-ca-main"
I0204 15:37:40.392720    2516 keypair.go:225] Issuing new certificate: "etcd-manager-ca-main"
W0204 15:37:40.450171    2516 vfs_castore.go:379] CA private key was not found
I0204 15:37:40.451252    2516 keypair.go:225] Issuing new certificate: "apiserver-aggregator-ca"
I0204 15:37:40.530761    2516 keypair.go:225] Issuing new certificate: "etcd-peers-ca-events"
I0204 15:37:40.551422    2516 keypair.go:225] Issuing new certificate: "service-account"
I0204 15:37:40.651551    2516 keypair.go:225] Issuing new certificate: "kubernetes-ca"
I0204 15:37:42.362019    2516 executor.go:111] Tasks: 44 done / 96 total; 18 can run
I0204 15:37:43.400786    2516 executor.go:111] Tasks: 62 done / 96 total; 26 can run
I0204 15:37:44.842142    2516 executor.go:111] Tasks: 88 done / 96 total; 2 can run
I0204 15:37:44.950531    2516 executor.go:111] Tasks: 90 done / 96 total; 4 can run
I0204 15:37:45.524542    2516 executor.go:111] Tasks: 94 done / 96 total; 2 can run
I0204 15:37:46.114893    2516 executor.go:155] No progress made, sleeping before retrying 2 task(s)
I0204 15:37:56.115922    2516 executor.go:111] Tasks: 94 done / 96 total; 2 can run
I0204 15:37:57.429920    2516 executor.go:111] Tasks: 96 done / 96 total; 0 can run
I0204 15:37:57.772970    2516 update_cluster.go:326] Exporting kubeconfig for cluster
kOps has set your kubectl context to kops.k8s.local
```



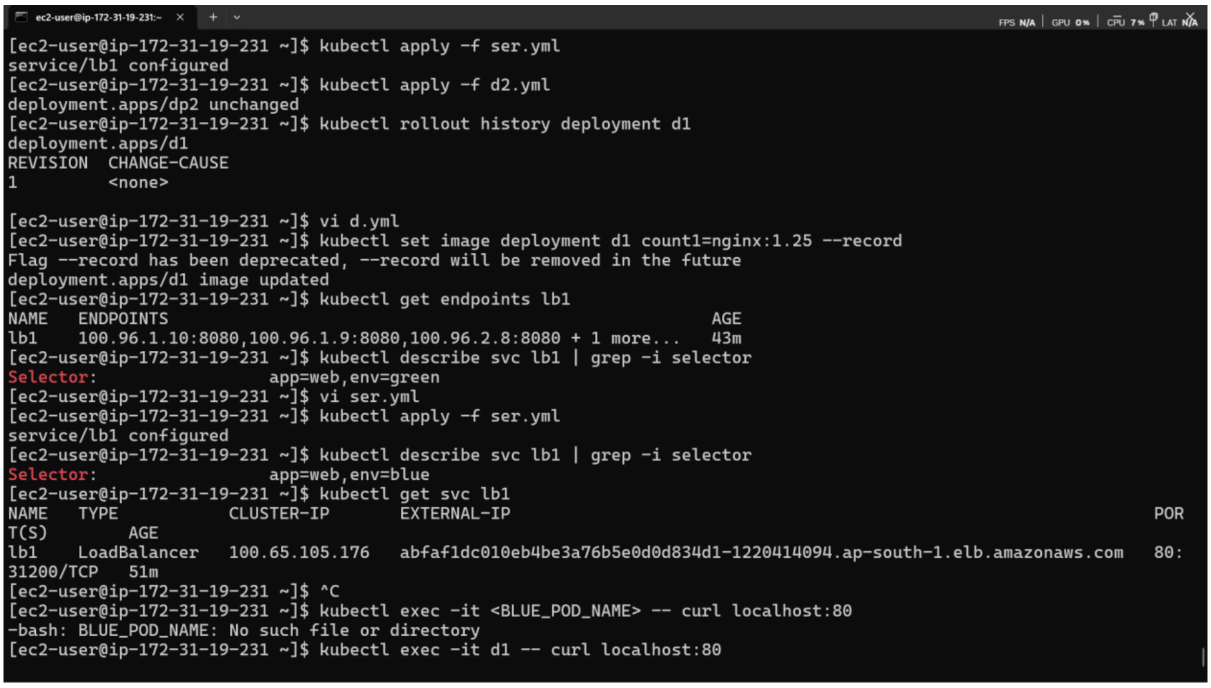
```
[ec2-user@ip-172-31-15-155 ~]$ kops get cluster
NAME                    CLOUD   ZONES
kops-cluster.k8s.local  aws     ap-south-1b
[ec2-user@ip-172-31-15-155 ~]$ kops update cluster --name kops-cluster.k8s.local --yes --admin

*********************************************************************************

A new kops version is available: 1.29.2
Upgrading is recommended
More information: https://github.com/kubernetes/kops/blob/master/permalinks/upgrade_kops.md#1.29.2

*********************************************************************************

I0111 17:34:24.746830    3403 apply_cluster.go:467] Gossip DNS: skipping DNS validation
I0111 17:34:29.656512    3403 executor.go:111] Tasks: 0 done / 96 total; 44 can run
I0111 17:34:30.435649    3403 executor.go:111] Tasks: 44 done / 96 total; 18 can run
I0111 17:34:31.203276    3403 executor.go:111] Tasks: 62 done / 96 total; 26 can run
I0111 17:34:31.399063    3403 executor.go:111] Tasks: 88 done / 96 total; 2 can run
I0111 17:34:31.491038    3403 executor.go:111] Tasks: 90 done / 96 total; 4 can run
I0111 17:34:31.820082    3403 executor.go:111] Tasks: 94 done / 96 total; 2 can run
I0111 17:34:32.009233    3403 executor.go:111] Tasks: 96 done / 96 total; 0 can run
I0111 17:34:32.055685    3403 update_cluster.go:326] Exporting kubeconfig for cluster
kOps has set your kubectl context to kops-cluster.k8s.local

Cluster changes have been applied to the cloud.


Changes may require instances to restart: kops rolling-update cluster

[ec2-user@ip-172-31-15-155 ~]$ kops get nodes
Error: Cluster.kops.k8s.io "nodes" not found
[ec2-user@ip-172-31-15-155 ~]$ kops get po
Error: Cluster.kops.k8s.io "po" not found
[ec2-user@ip-172-31-15-155 ~]$ kubectl get po

^C
[ec2-user@ip-172-31-15-155 ~]$ kubectl get po
E0111 17:35:47.956688    3474 memcache.go:265] "Unhandled Error" err="couldn't get current server API group list: Get \"https://api-kops-cluster-k8s-loca-94
5gpc-1898162782.ap-south-1.elb.amazonaws.com/api?timeout=32s\": EOF"
No resources found in default namespace.
[ec2-user@ip-172-31-15-155 ~]$ kubectl get po
No resources found in default namespace.
[ec2-user@ip-172-31-15-155 ~]$ kubectl get po
```



```
I0204 15:40:31.732663    2576 executor.go:111] Tasks: 94 done / 96 total; 2 can run
I0204 15:40:31.922717    2576 executor.go:111] Tasks: 96 done / 96 total; 0 can run
I0204 15:40:31.972600    2576 update_cluster.go:326] Exporting kubeconfig for cluster
kOps has set your kubectl context to kops.k8s.local

Cluster changes have been applied to the cloud.


Changes may require instances to restart: kops rolling-update cluster

[ec2-user@ip-172-31-19-231 ~]$ kubectl get pods
No resources found in default namespace.
[ec2-user@ip-172-31-19-231 ~]$ kubectl get pods
No resources found in default namespace.
[ec2-user@ip-172-31-19-231 ~]$ kops rolling-update cluster
Using cluster from kubectl context: kops.k8s.local

NAME                    STATUS  NEEDUPDATE      READY   MIN     TARGET  MAX     NODES
master-ap-south-1a      Ready   0               1       1       1       1       1
nodes-ap-south-1a       Ready   0               2       2       2       2       2

No rolling-update required.
[ec2-user@ip-172-31-19-231 ~]$ kubectl get pods
No resources found in default namespace.
[ec2-user@ip-172-31-19-231 ~]$ kubectl get nodes
NAME                  STATUS   ROLES           AGE     VERSION
i-074363f1f80075a9d   Ready    control-plane   2m38s   v1.24.17
i-0db1385261d7f8370   Ready    node            91s     v1.24.17
i-0dfdf290d9711474f   Ready    node            89s     v1.24.17
[ec2-user@ip-172-31-19-231 ~]$
```

---

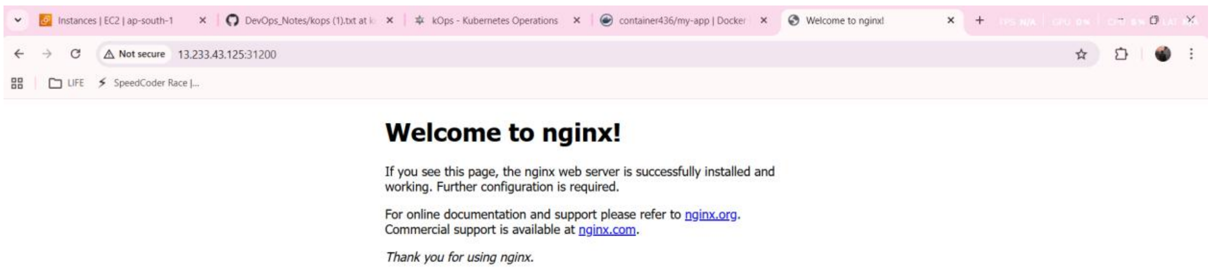
5. Application Deployments



```
[ec2-user@ip-172-31-19-231 ~]$ kubectl apply -f ser.yml
service/lb1 configured
[ec2-user@ip-172-31-19-231 ~]$ kubectl apply -f d2.yml
deployment.apps/dp2 unchanged
[ec2-user@ip-172-31-19-231 ~]$ kubectl rollout history deployment d1
deployment.apps/d1
REVISION  CHANGE-CAUSE
1         <none>

[ec2-user@ip-172-31-19-231 ~]$ vi d.yml
[ec2-user@ip-172-31-19-231 ~]$ kubectl set image deployment d1 count1=nginx:1.25 --record
Flag --record has been deprecated, --record will be removed in the future
deployment.apps/d1 image updated
[ec2-user@ip-172-31-19-231 ~]$ kubectl get endpoints lb1
NAME   ENDPOINTS                                                 AGE
lb1    100.96.1.10:8080,100.96.1.9:8080,100.96.2.8:8080 + 1 more...   43m
[ec2-user@ip-172-31-19-231 ~]$ kubectl describe svc lb1 | grep -i selector
Selector:                 app=web,env=green
[ec2-user@ip-172-31-19-231 ~]$ vi ser.yml
[ec2-user@ip-172-31-19-231 ~]$ kubectl apply -f ser.yml
service/lb1 configured
[ec2-user@ip-172-31-19-231 ~]$ kubectl describe svc lb1 | grep -i selector
Selector:                 app=web,env=blue
[ec2-user@ip-172-31-19-231 ~]$ kubectl get svc lb1
NAME   TYPE           CLUSTER-IP       EXTERNAL-IP                                                               POR
T(S)        AGE
lb1    LoadBalancer   100.65.105.176   abfaf1dc010eb4be3a76b5e0d0d834d1-1220414094.ap-south-1.elb.amazonaws.com   80:
31200/TCP    51m
[ec2-user@ip-172-31-19-231 ~]$ ^C
[ec2-user@ip-172-31-19-231 ~]$ kubectl exec -it <BLUE_POD_NAME> -- curl localhost:80
-bash: BLUE_POD_NAME: No such file or directory
[ec2-user@ip-172-31-19-231 ~]$ kubectl exec -it d1 -- curl localhost:80
```

Blue Deployment (nginx)

- Label: env=blue
- Container: nginx
- Listening port: 80
- Purpose: Stable / previous version



13.233.43.125:31200

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

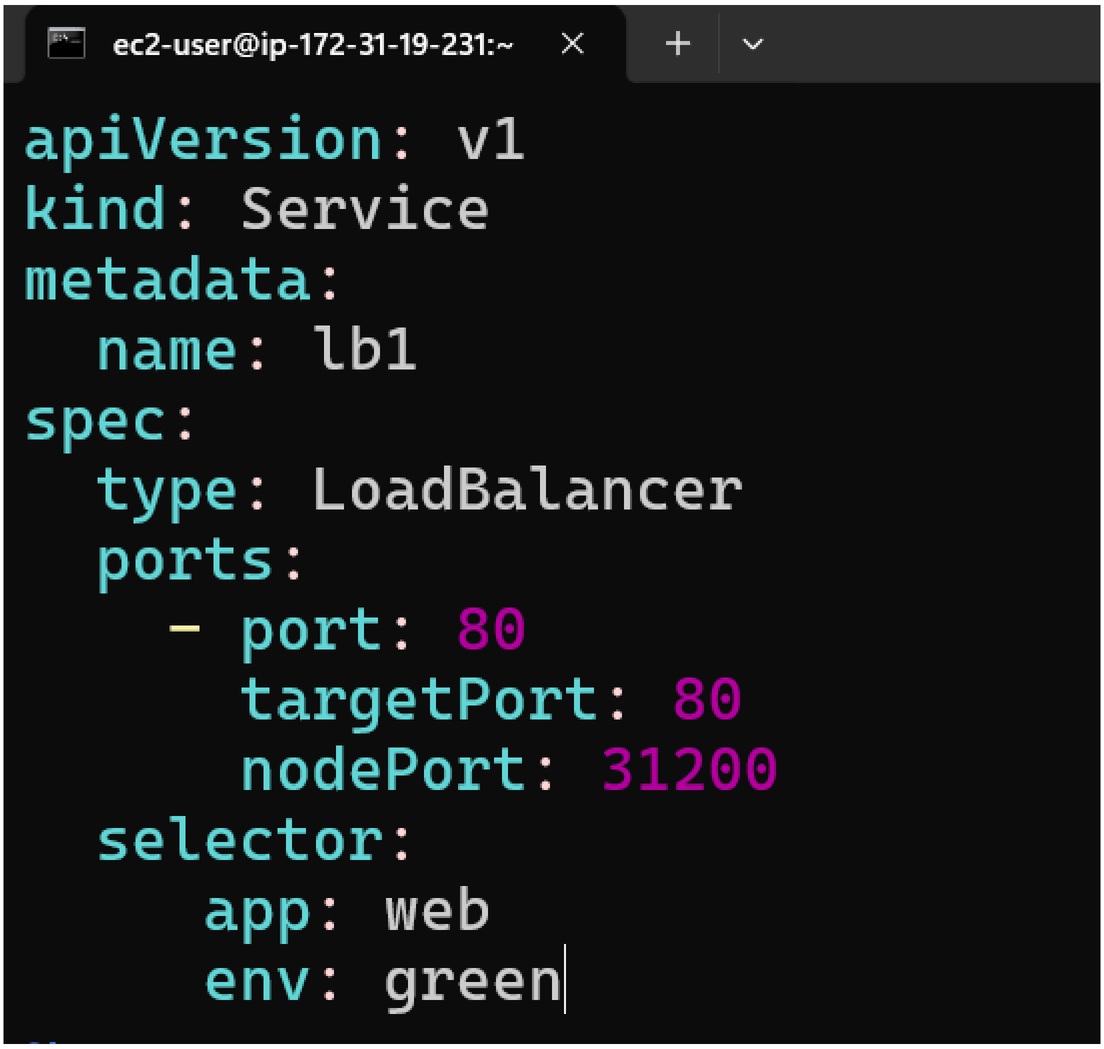
-

Green Deployment (Tomcat application)

- Label: env=green

- Container: Custom Tomcat-based image
- Listening port: 8080
- Purpose: New version

Each Deployment manages its own ReplicaSet and Pods.



```yaml
apiVersion: v1
kind: Service
metadata:
  name: lb1
spec:
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 80
      nodePort: 31200
  selector:
     app: web
     env: green
```

---

6. Service Configuration

Service Type

- LoadBalancer

Service Role

- Exposes application publicly using AWS ELB
- Routes traffic based on **label selectors**

Key Concept

The Service **does not talk to Deployments**. It selects **Pods directly via labels**.

---

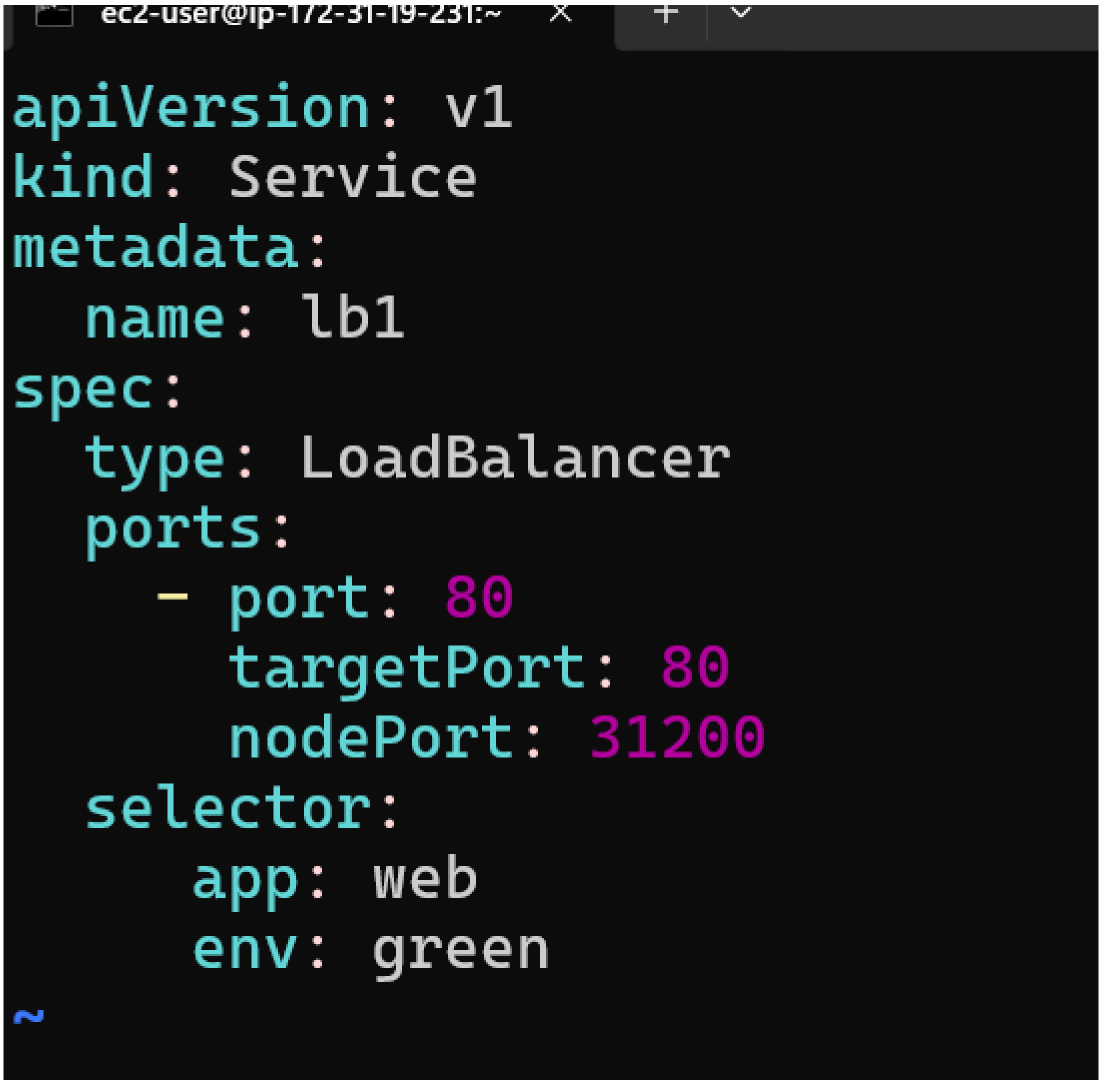
7. Blue–Green Deployment Strategy

Traffic Switching

Traffic is controlled entirely by the Service selecto

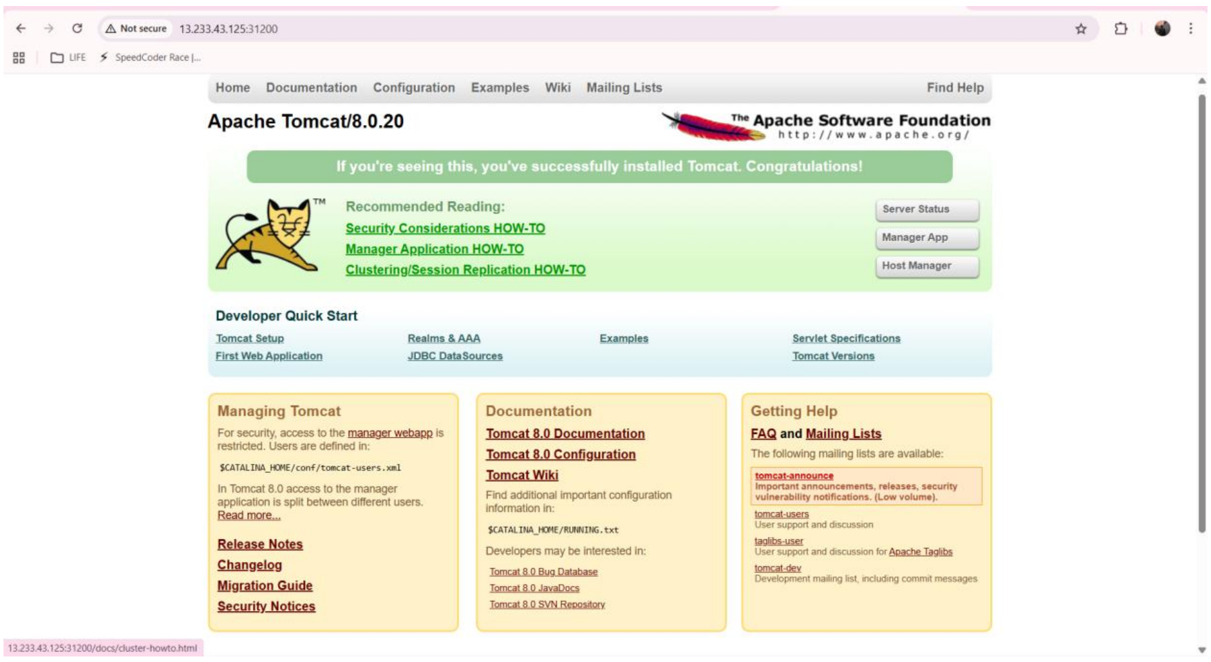
- env=blue → traffic goes to Blue (nginx)
- env=green → traffic goes to Green (Tomcat)

No pod restart or redeployment is required.



ec2-user@ip-172-31-19-231:~

```
apiVersion: v1
kind: Service
metadata:
  name: lb1
spec:
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 80
      nodePort: 31200
  selector:
     app: web
     env: green
~
```


Not secure 13.233.43.125:31200

LIFE SpeedCoder Race |...

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/8.0.20

The Apache Software Foundation http://www.apache.org/

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Server Status
Manager App
Host Manager

Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:
```
$CATALINA_HOME/conf/tomcat-users.xml
```
In Tomcat 8.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation
Tomcat 8.0 Documentation
Tomcat 8.0 Configuration
Tomcat Wiki
Find additional important configuration information in:
```
$CATALINA_HOME/RUNNING.txt
```
Developers may be interested in:
Tomcat 8.0 Bug Database
Tomcat 8.0 JavaDocs
Tomcat 8.0 SVN Repository

Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).
tomcat-users
User support and discussion
taglibs-user
User support and discussion for Apache Taglibs
tomcat-dev
Development mailing list, including commit messages

13.233.43.125:31200/docs/cluster-howto.html

Rollback

Rollback is instant by switching the selector back to the previous version.

---

8. Debugging & Challenges Faced

1. Immutable Selectors

- Deployment selectors cannot be changed
- Fix: Delete and recreate Deployment

2. Pod Naming Issue

- Pods must not have fixed names in Deployments
- Fix: Remove metadata.name from pod template

3. Port Mismatch

- Green app was listening on 8080 while Service targeted 80
- Fix: Update targetPort to match actual container port

4. NodePort Access Failure

- NodePort blocked by AWS Security Groups
- Fix: Use ELB DNS instead of Node IP

5. ELB Empty Response

- ELB had no healthy backends
- Fix: Align Service selector and ports, recreate Service

6. DNS Errors

- Incorrect or incomplete ELB DNS name
- Fix: Use full .elb.amazonaws.com DNS

---

9. Verification Commands

- Check Deployments:
  kubectl get deploy
- Check Pods:
  kubectl get pods --show-labels
- Check Service:
  kubectl get svc lb1
- Check Endpoints (source of truth):
  kubectl get endpoints lb1

---

10. Cluster Cleanup

Cluster deletion is performed using kOps and S3 state store with correct region configuration.

Key requirement:

- AWS region must be specified when deleting the cluster

---

11. Key Learnings

- Service selector controls traffic, not Deployments
- Blue–Green rollback is faster and safer than rollout undo
- targetPort must match the actual application port
- NodePort is internal plumbing on AWS
- Endpoints object always shows the real traffic destination

---

12. Conclusion

This project demonstrates a **real-world Kubernetes Blue–Green deployment on AWS** using kOps. It covers not only deployment but also real debugging scenarios encountered in production environments, making it a strong practical DevOps project.

---

13. Future Improvements

- Add readiness and liveness probes

- Use Ingress with ALB
- Automate deployments via CI/CD
- Implement Canary deployments

End of Document