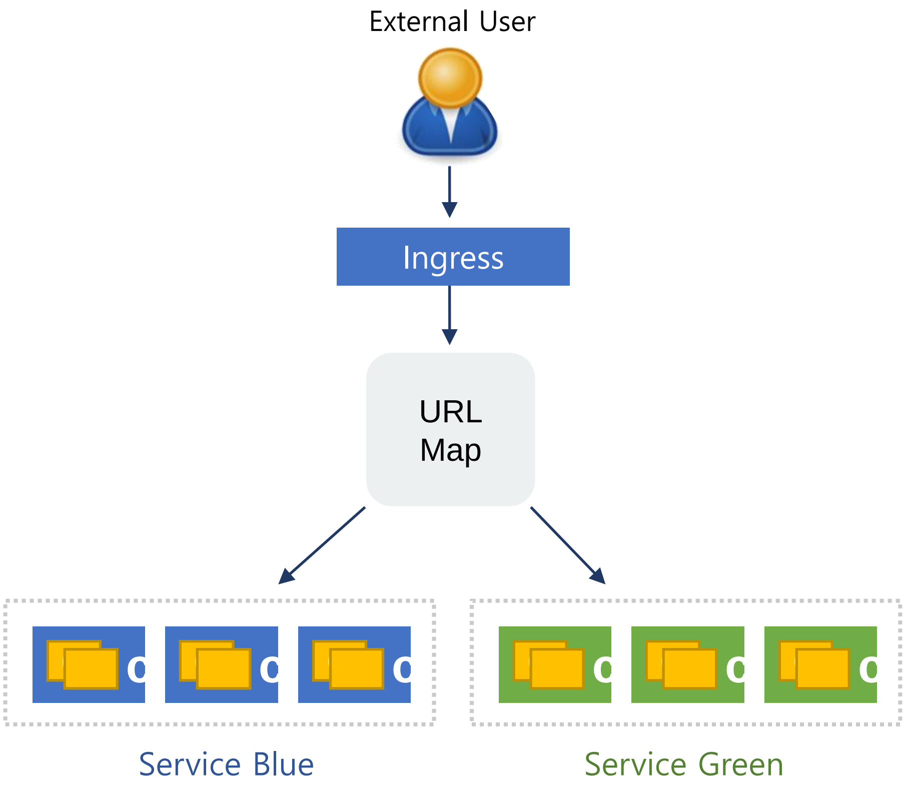
Ingress

- MSA 서비스간 라우팅을 위해 API Gateway를 두는 경우가 많은데 관리포인트가 생김
- URL기반의 라우팅 정도라면 L7 로드밸런서 정도로 위의 기능을 충족
- Kubernetes에서 제공하는 L7 로드밸런싱 컴포넌트를 'Ingress' 라고 함

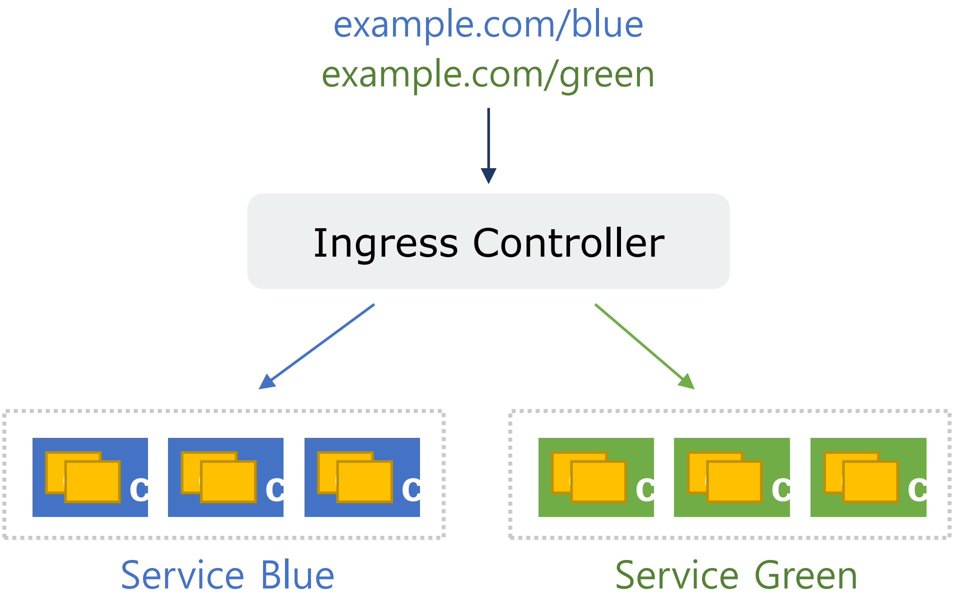
아래와 같은 Service들의 Inbound Connection을 지원하기 위해
Ingress는 Layer7의 HTTP Load balancer 기능 제공

- TLS (SSL)
- Name-based virtual hosting
- Path-based routing
- Custom rules



Ingress

URL Path based



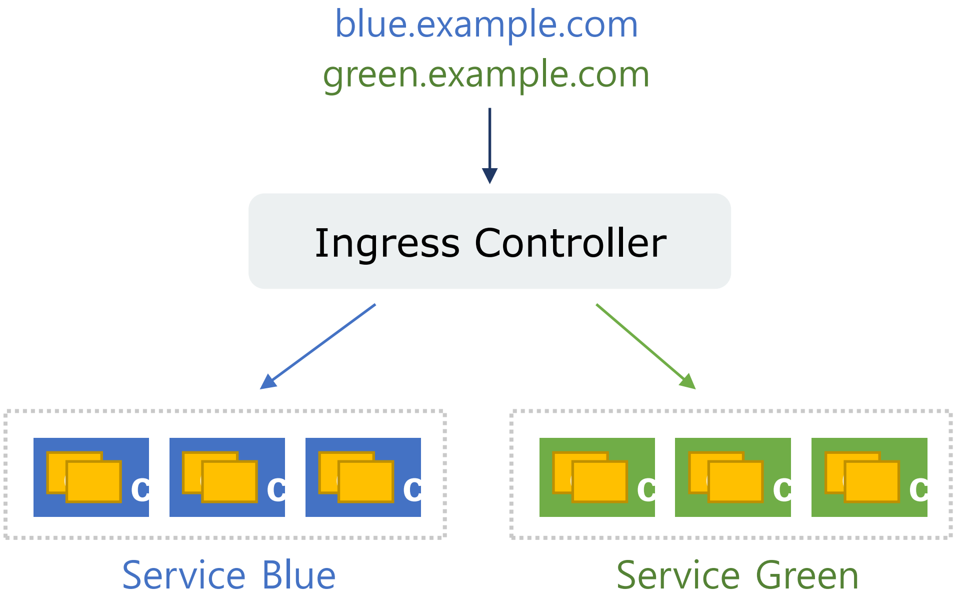
example.com/blue
example.com/green

Ingress Controller

Service Blue

Service Green

Virtual Hosting



blue.example.com
green.example.com

Ingress Controller

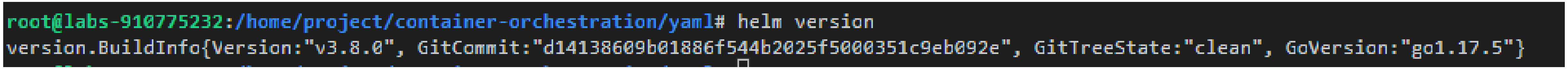
Service Blue

Service Green

- 사용자들은 직접 Service에 접속하지 않는다.
- 유저는 Ingress에 먼저 접근하고, 요청은 해당 Service로 포워드 된다.
- Ingress 요청은 Ingress Controller에 의해 처리된다.

"Ingress Controller"는 Ingress 리소스의 변경 사항을 마스터 노드의 API 서버에서 감시하고,
그에 따라 Layer 7로드 밸런서를 업데이트하는 응용 프로그램

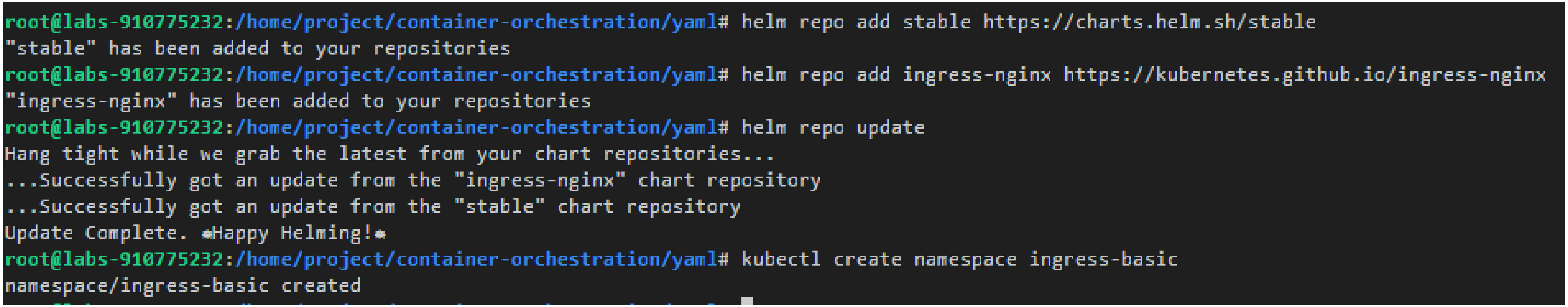
Ingress



```
root@labs-910775232:/home/project/container-orchestration/yaml# helm version
version.BuildInfo{Version:"v3.8.0", GitCommit:"d14138609b01886f544b2025f5000351c9eb092e", GitTreeState:"clean", GoVersion:"go1.17.5"}
```

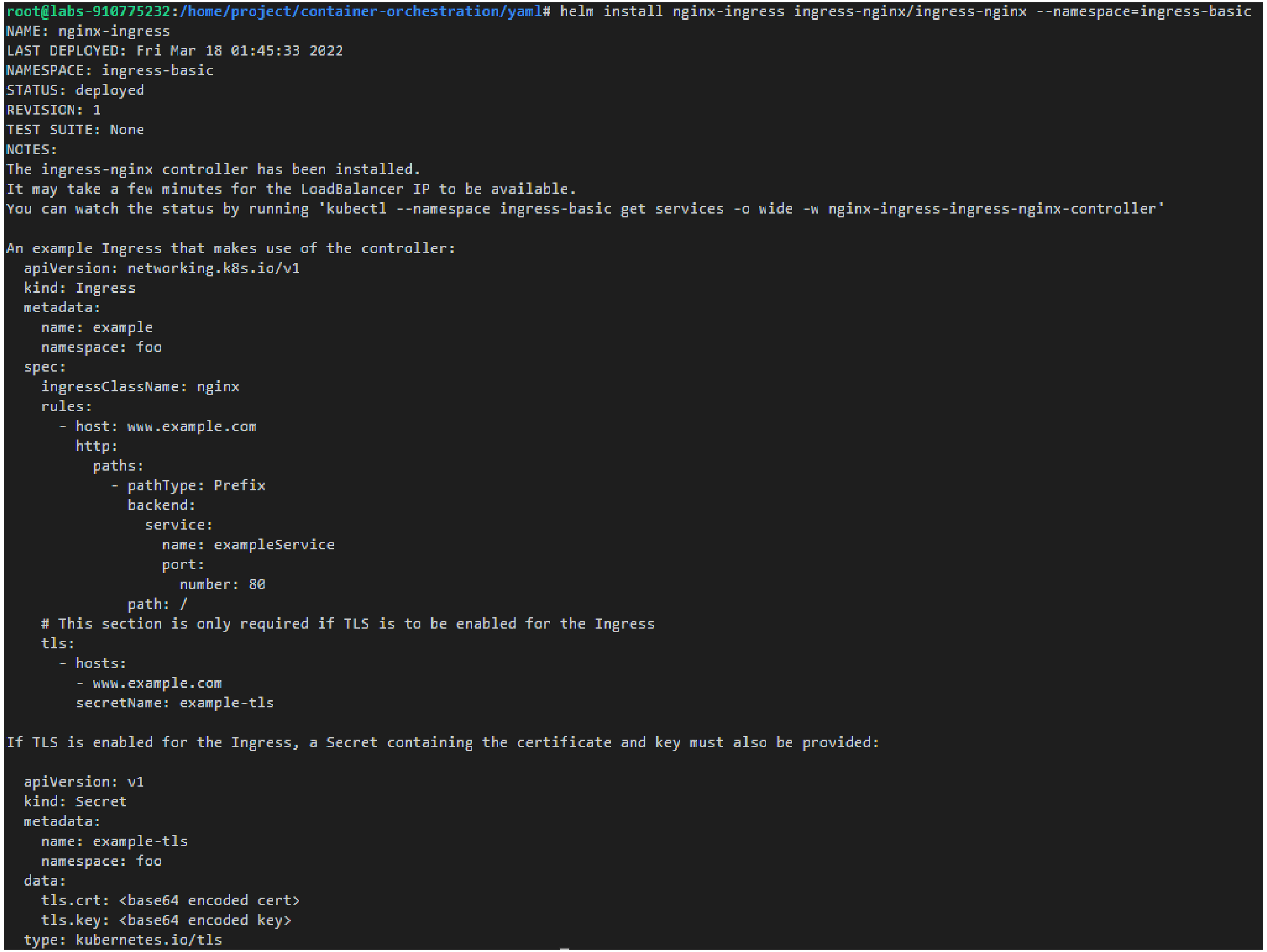
Helm으로 Ingress Controller 설치

helm repo add stable https://charts.helm.sh/stable
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
helm repo update
kubectl create namespace ingress-basic



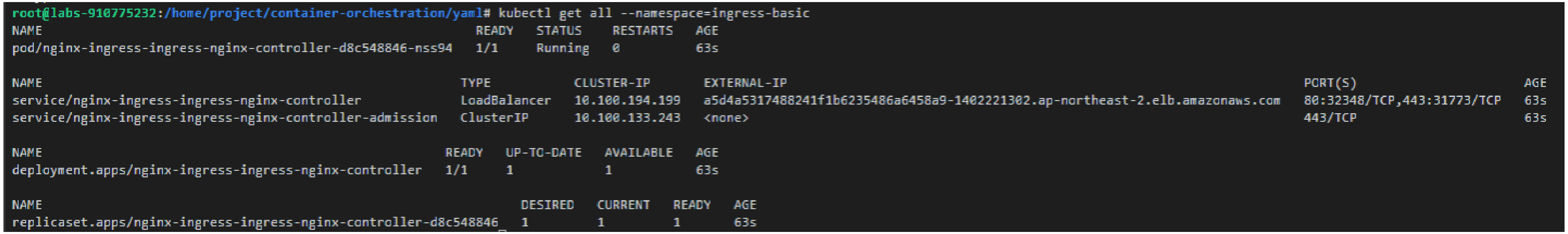
```
root@labs-910775232:/home/project/container-orchestration/yaml# helm repo add stable https://charts.helm.sh/stable
"stable" has been added to your repositories
root@labs-910775232:/home/project/container-orchestration/yaml# helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
"ingress-nginx" has been added to your repositories
root@labs-910775232:/home/project/container-orchestration/yaml# helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "ingress-nginx" chart repository
...Successfully got an update from the "stable" chart repository
Update Complete. ⎈Happy Helming!⎈
root@labs-910775232:/home/project/container-orchestration/yaml# kubectl create namespace ingress-basic
namespace/ingress-basic created
```

helm install nginx-ingress ingress-nginx/ingress-nginx --namespace=ingress-basic



```
root@labs-910775232:/home/project/container-orchestration/yaml# helm install nginx-ingress ingress-nginx/ingress-nginx --namespace=ingress-basic
NAME: nginx-ingress
LAST DEPLOYED: Fri Mar 18 01:45:33 2022
NAMESPACE: ingress-basic
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The ingress-nginx controller has been installed.
It may take a few minutes for the LoadBalancer IP to be available.
You can watch the status by running 'kubectl --namespace ingress-basic get services -o wide -w nginx-ingress-ingress-nginx-controller'

An example Ingress that makes use of the controller:
  apiVersion: networking.k8s.io/v1
  kind: Ingress
  metadata:
    name: example
    namespace: foo
  spec:
    ingressClassName: nginx
    rules:
      - host: www.example.com
        http:
          paths:
            - pathType: Prefix
              backend:
                service:
                  name: exampleService
                  port:
                    number: 80
              path: /
    # This section is only required if TLS is to be enabled for the Ingress
    tls:
      - hosts:
        - www.example.com
        secretName: example-tls

If TLS is enabled for the Ingress, a Secret containing the certificate and key must also be provided:

  apiVersion: v1
  kind: Secret
  metadata:
    name: example-tls
    namespace: foo
  data:
    tls.crt: <base64 encoded cert>
    tls.key: <base64 encoded key>
  type: kubernetes.io/tls
```

kubectl get all --namespace=ingress-basic



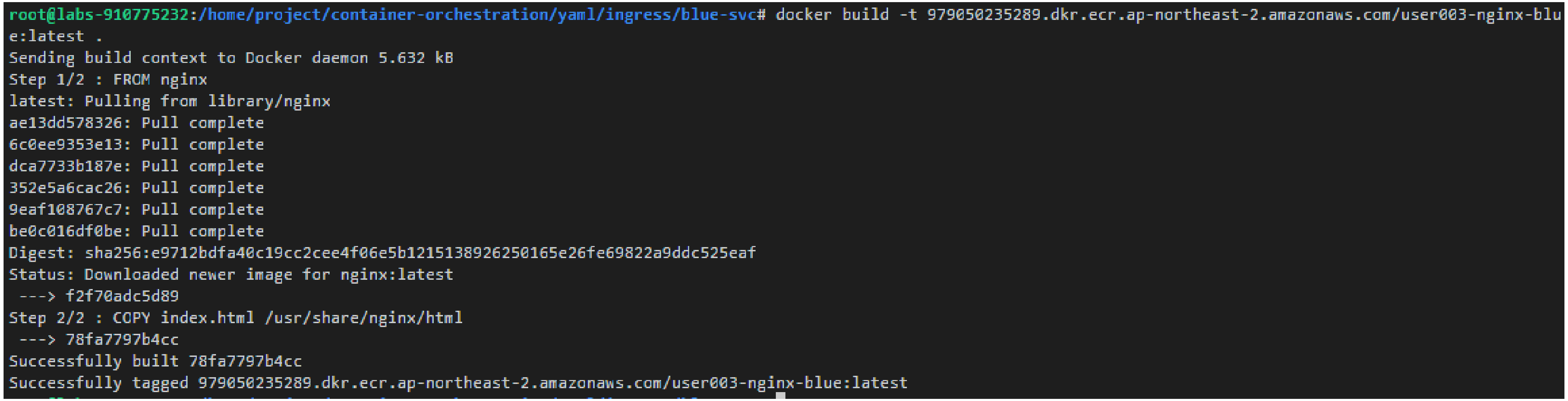
```
root@labs-910775232:/home/project/container-orchestration/yaml# kubectl get all --namespace=ingress-basic
NAME                                                READY   STATUS    RESTARTS   AGE
pod/nginx-ingress-ingress-nginx-controller-d8c548846-nss94   1/1     Running   0          63s

NAME                                               TYPE           CLUSTER-IP       EXTERNAL-IP                                                                 PORT(S)                      AGE
service/nginx-ingress-ingress-nginx-controller             LoadBalancer   10.100.194.199   a5d4a5317488241f1b6235486a6458a9-1402221302.ap-northeast-2.elb.amazonaws.com   80:32348/TCP,443:31773/TCP   63s
service/nginx-ingress-ingress-nginx-controller-admission   ClusterIP      10.100.133.243   <none>                                                                      443/TCP                      63s

NAME                                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-ingress-ingress-nginx-controller   1/1     1            1           63s

NAME                                                                DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-ingress-ingress-nginx-controller-d8c548846   1         1         1       63s
```

EXTERNAL-IP: a5d4a5317488241f1b6235486a6458a9-1402221302.ap-northeast-2.elb.amazonaws.com

External-IP가 API Gateway 엔드포인트

Ingress 대상 서비스(BLUE) 생성

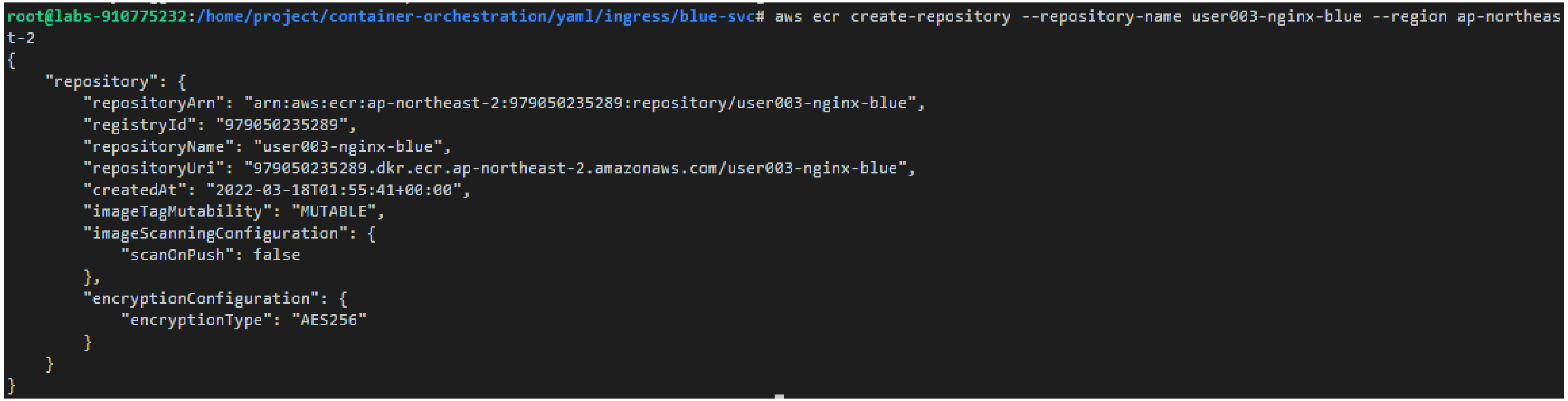
docker build -t 979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-blue:latest .



```
root@labs-910775232:/home/project/container-orchestration/yaml/ingress/blue-svc# docker build -t 979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-blu
e:latest .
Sending build context to Docker daemon 5.632 kB
Step 1/2 : FROM nginx
latest: Pulling from library/nginx
ae13dd578326: Pull complete
6c0ee9353e13: Pull complete
dca7733b187e: Pull complete
352e5a6cac26: Pull complete
9eaf108767c7: Pull complete
be0c016df0be: Pull complete
Digest: sha256:e9712bdfa40c19cc2cee4f06e5b1215138926250165e26fe69822a9ddc525eaf
Status: Downloaded newer image for nginx:latest
 ---> f2f70adc5d89
Step 2/2 : COPY index.html /usr/share/nginx/html
 ---> 78fa7797b4cc
Successfully built 78fa7797b4cc
Successfully tagged 979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-blue:latest
```

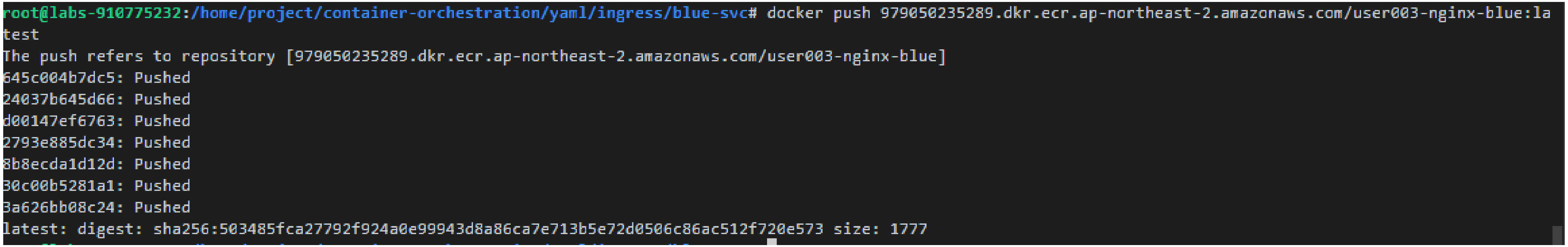
Ingress 대상 서비스(BLUE) 생성

aws ecr create-repository --repository-name user003-nginx-blue --region ap-northeast-2



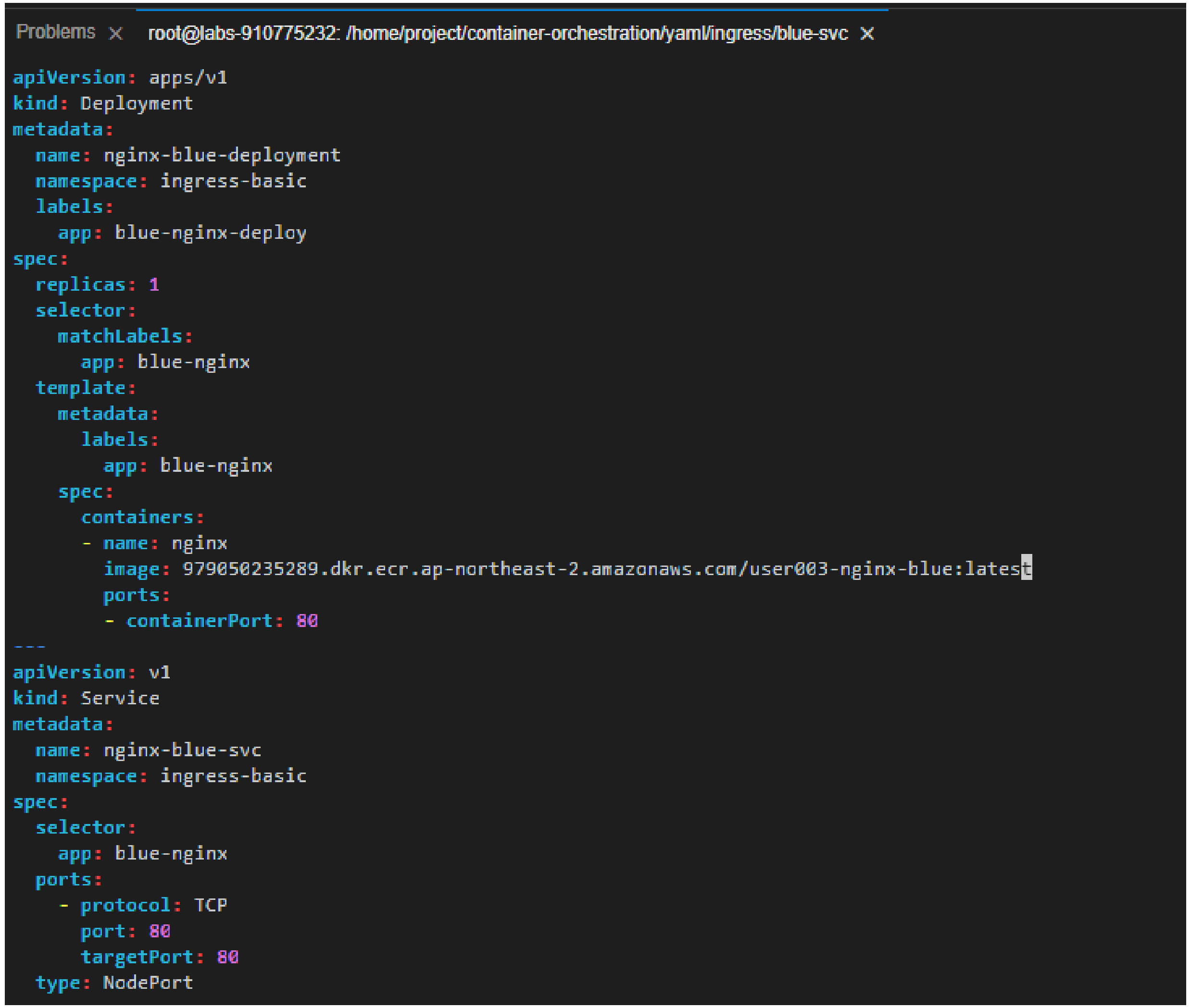
```
root@labs-910775232:/home/project/container-orchestration/yaml/ingress/blue-svc# aws ecr create-repository --repository-name user003-nginx-blue --region ap-northeas
t-2
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:ap-northeast-2:979050235289:repository/user003-nginx-blue",
        "registryId": "979050235289",
        "repositoryName": "user003-nginx-blue",
        "repositoryUri": "979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-blue",
        "createdAt": "2022-03-18T01:55:41+00:00",
        "imageTagMutability": "MUTABLE",
        "imageScanningConfiguration": {
            "scanOnPush": false
        },
        "encryptionConfiguration": {
            "encryptionType": "AES256"
        }
    }
}
```

docker push 979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-blue:latest



```
root@labs-910775232:/home/project/container-orchestration/yaml/ingress/blue-svc# docker push 979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-blue:la
test
The push refers to repository [979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-blue]
645c004b7dc5: Pushed
24037b645d66: Pushed
d00147ef6763: Pushed
2793e885dc34: Pushed
8b8ecda1d12d: Pushed
30c00b5281a1: Pushed
3a626bb08c24: Pushed
latest: digest: sha256:503485fca27792f924a0e99943d8a86ca7e713b5e72d0506c86ac512f720e573 size: 1777
```

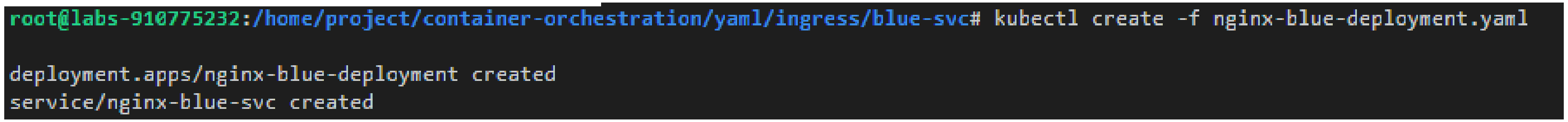
EDIT nginx-blue-deployment.yaml



Problems ×
root@labs-910775232: /home/project/container-orchestration/yaml/ingress/blue-svc ×

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-blue-deployment
  namespace: ingress-basic
  labels:
    app: blue-nginx-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      app: blue-nginx
  template:
    metadata:
      labels:
        app: blue-nginx
    spec:
      containers:
      - name: nginx
        image: 979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-blue:latest
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: nginx-blue-svc
  namespace: ingress-basic
spec:
  selector:
    app: blue-nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: NodePort
```

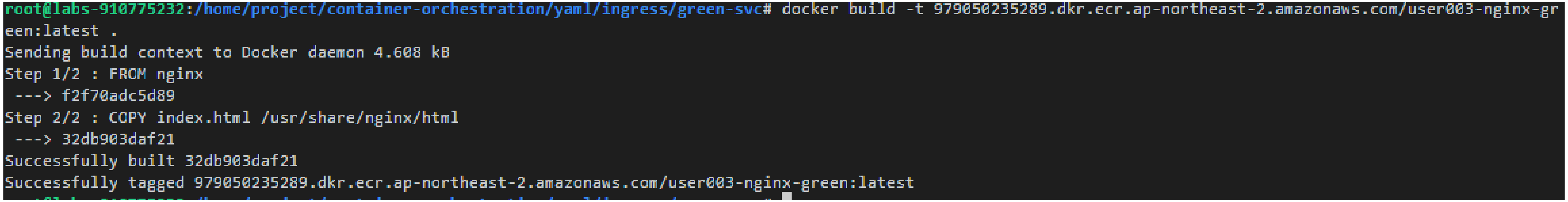
kubectl create -f nginx-blue-deployment.yaml



```
root@labs-910775232:/home/project/container-orchestration/yaml/ingress/blue-svc# kubectl create -f nginx-blue-deployment.yaml

deployment.apps/nginx-blue-deployment created
service/nginx-blue-svc created
```

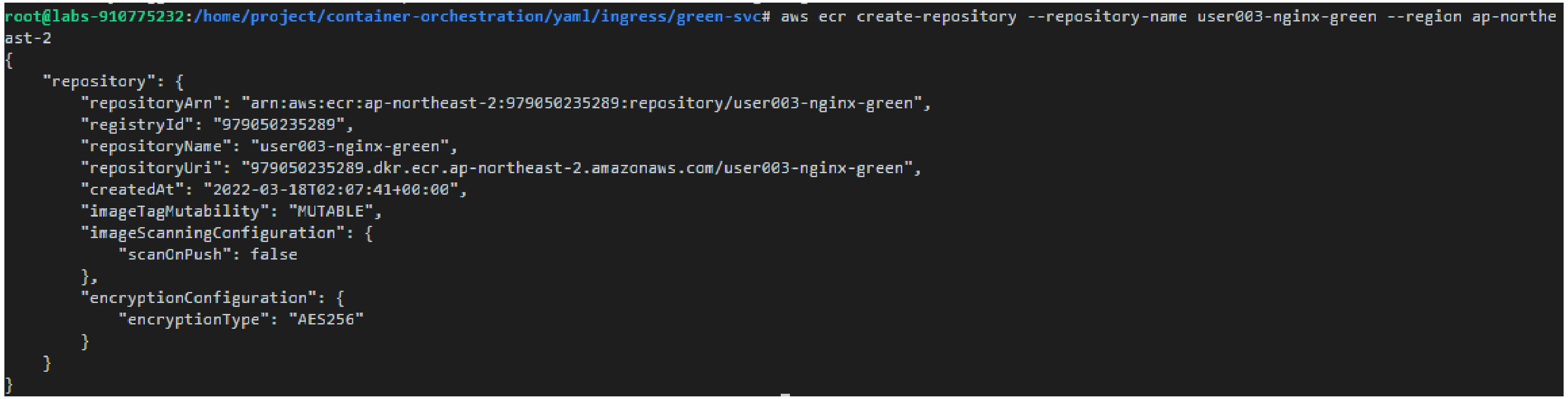
Ingress 대상 서비스(GREEN) 생성

```
docker build -t 979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-green:latest .
```



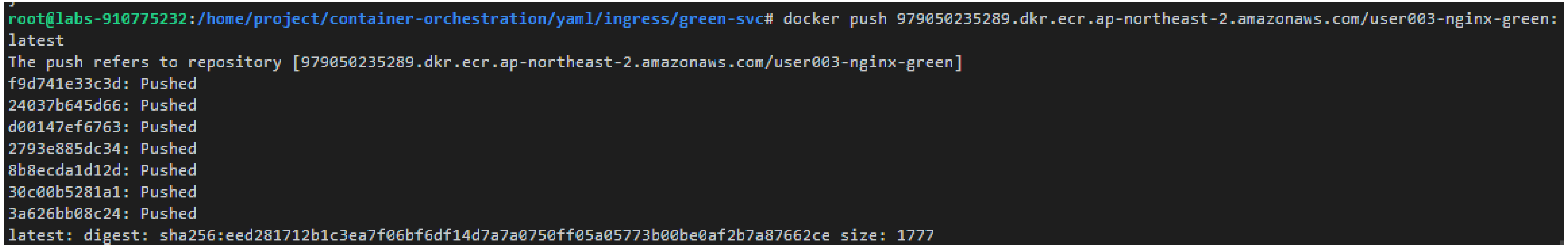
```
root@labs-910775232:/home/project/container-orchestration/yaml/ingress/green-svc# docker build -t 979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-gr
een:latest .
Sending build context to Docker daemon 4.608 kB
Step 1/2 : FROM nginx
 ---> f2f70adc5d89
Step 2/2 : COPY index.html /usr/share/nginx/html
 ---> 32db903daf21
Successfully built 32db903daf21
Successfully tagged 979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-green:latest
```

```
aws ecr create-repository --repository-name user003-nginx-green --region ap-northeast-2
```



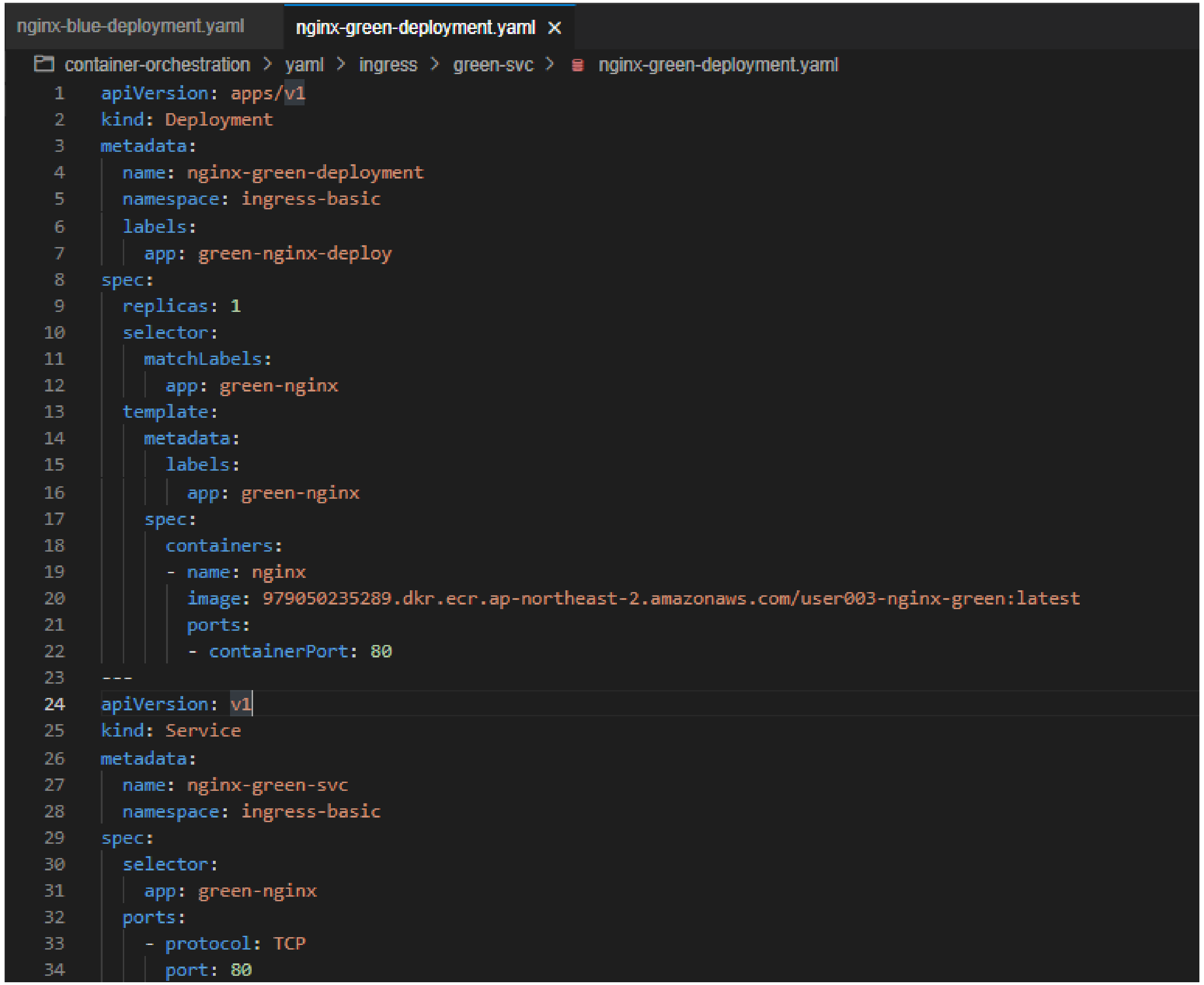
```
root@labs-910775232:/home/project/container-orchestration/yaml/ingress/green-svc# aws ecr create-repository --repository-name user003-nginx-green --region ap-northe
ast-2
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:ap-northeast-2:979050235289:repository/user003-nginx-green",
        "registryId": "979050235289",
        "repositoryName": "user003-nginx-green",
        "repositoryUri": "979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-green",
        "createdAt": "2022-03-18T02:07:41+00:00",
        "imageTagMutability": "MUTABLE",
        "imageScanningConfiguration": {
            "scanOnPush": false
        },
        "encryptionConfiguration": {
            "encryptionType": "AES256"
        }
    }
}
```

```
docker push 979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-green:latest
```



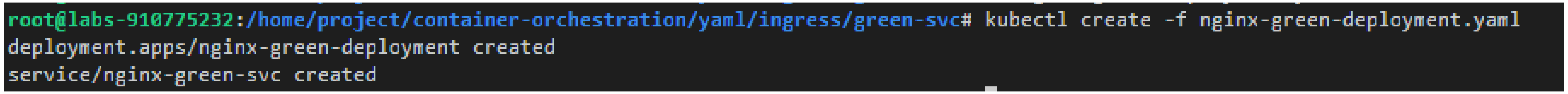
```
root@labs-910775232:/home/project/container-orchestration/yaml/ingress/green-svc# docker push 979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-green:
latest
The push refers to repository [979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-green]
f9d741e33c3d: Pushed
24037b645d66: Pushed
d00147ef6763: Pushed
2793e885dc34: Pushed
8b8ecda1d12d: Pushed
30c00b5281a1: Pushed
3a626bb08c24: Pushed
latest: digest: sha256:eed281712b1c3ea7f06bf6df14d7a7a0750ff05a05773b00be0af2b7a87662ce size: 1777
```

EDIT nginx-green-deployment.yaml


nginx-blue-deployment.yaml
nginx-green-deployment.yaml ✕
container-orchestration > yaml > ingress > green-svc > nginx-green-deployment.yaml

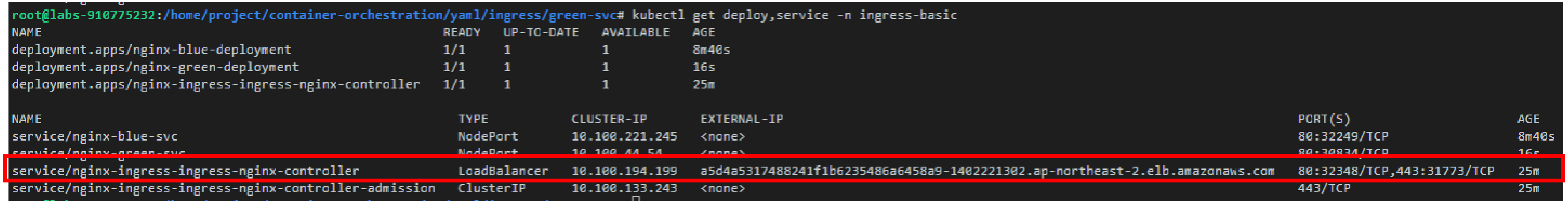
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-green-deployment
  namespace: ingress-basic
  labels:
    app: green-nginx-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      app: green-nginx
  template:
    metadata:
      labels:
        app: green-nginx
    spec:
      containers:
      - name: nginx
        image: 979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-green:latest
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: nginx-green-svc
  namespace: ingress-basic
spec:
  selector:
    app: green-nginx
  ports:
    - protocol: TCP
      port: 80
```

kubectl create -f nginx-green-deployment.yaml



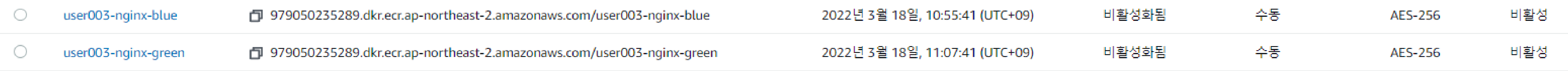
```
root@labs-910775232:/home/project/container-orchestration/yaml/ingress/green-svc# kubectl create -f nginx-green-deployment.yaml
deployment.apps/nginx-green-deployment created
service/nginx-green-svc created
```

kubectl get deploy,service -n ingress-basic



```
root@labs-910775232:/home/project/container-orchestration/yaml/ingress/green-svc# kubectl get deploy,service -n ingress-basic
NAME                                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-blue-deployment             1/1     1            1           8m40s
deployment.apps/nginx-green-deployment            1/1     1            1           16s
deployment.apps/nginx-ingress-ingress-nginx-controller   1/1     1            1           25m

NAME                                                 TYPE           CLUSTER-IP       EXTERNAL-IP                                                                    PORT(S)                      AGE
service/nginx-blue-svc                               NodePort       10.100.221.245   <none>                                                                         80:32249/TCP                 8m40s
service/nginx-green-svc                              NodePort       10.100.44.54     <none>                                                                         80:30834/TCP                 16s
service/nginx-ingress-ingress-nginx-controller       LoadBalancer   10.100.194.199   a5d4a5317488241f1b6235486a6458a9-1402221302.ap-northeast-2.elb.amazonaws.com   80:32348/TCP,443:31773/TCP   25m
service/nginx-ingress-ingress-nginx-controller-admission   ClusterIP      10.100.133.243   <none>                                                                         443/TCP                      25m
```

아마존 ECR에 레파지토리 생성 확인

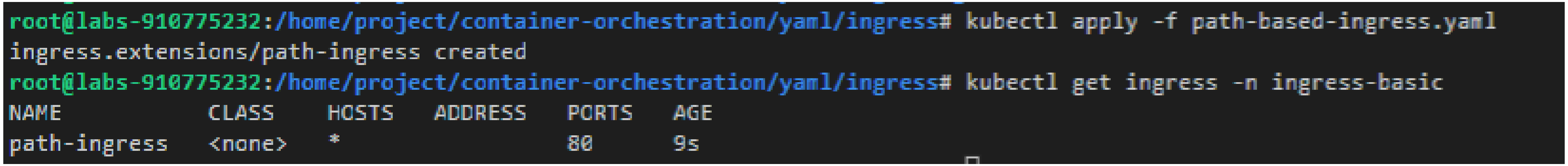


○	user003-nginx-blue	979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-blue	2022년 3월 18일, 10:55:41 (UTC+09)	비활성화됨	수동	AES-256	비활성
○	user003-nginx-green	979050235289.dkr.ecr.ap-northeast-2.amazonaws.com/user003-nginx-green	2022년 3월 18일, 11:07:41 (UTC+09)	비활성화됨	수동	AES-256	비활성

EXTERNAL-IP : a5d4a5317488241f1b6235486a6458a9-1402221302.ap-northeast-2.elb.amazonaws.com

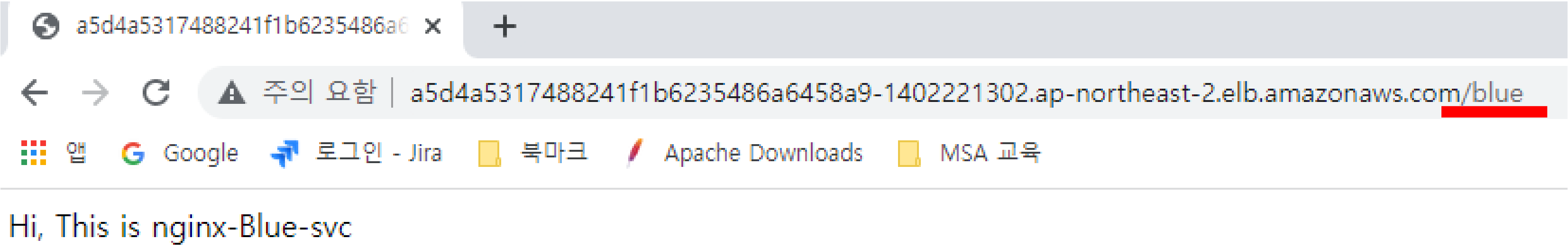
Ingress Routing Rule 생성

kubectl apply -f path-based-ingress.yaml
kubectl get ingress -n ingress-basic



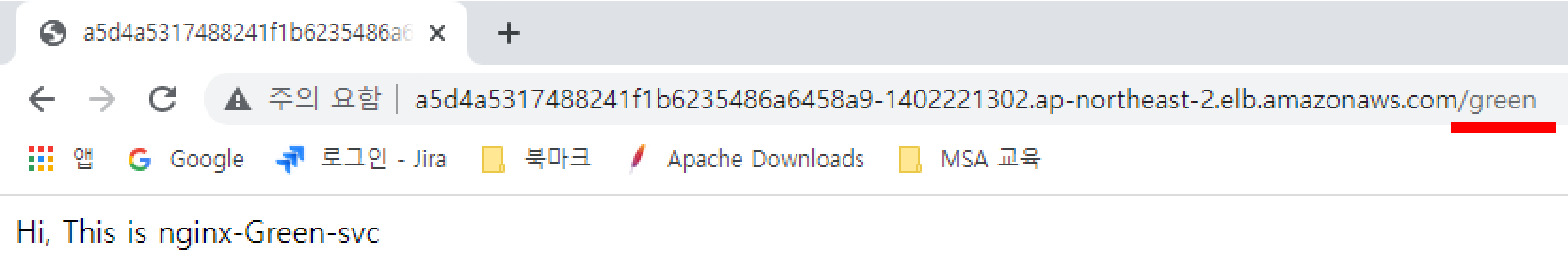
```
root@labs-910775232:/home/project/container-orchestration/yaml/ingress# kubectl apply -f path-based-ingress.yaml
ingress.extensions/path-ingress created
root@labs-910775232:/home/project/container-orchestration/yaml/ingress# kubectl get ingress -n ingress-basic
NAME           CLASS    HOSTS   ADDRESS   PORTS   AGE
path-ingress   <none>   *                 80      9s
```

Ingress Routing 테스트



a5d4a5317488241f1b6235486a6458a9-1402221302.ap-northeast-2.elb.amazonaws.com/blue

Hi, This is nginx-Blue-svc

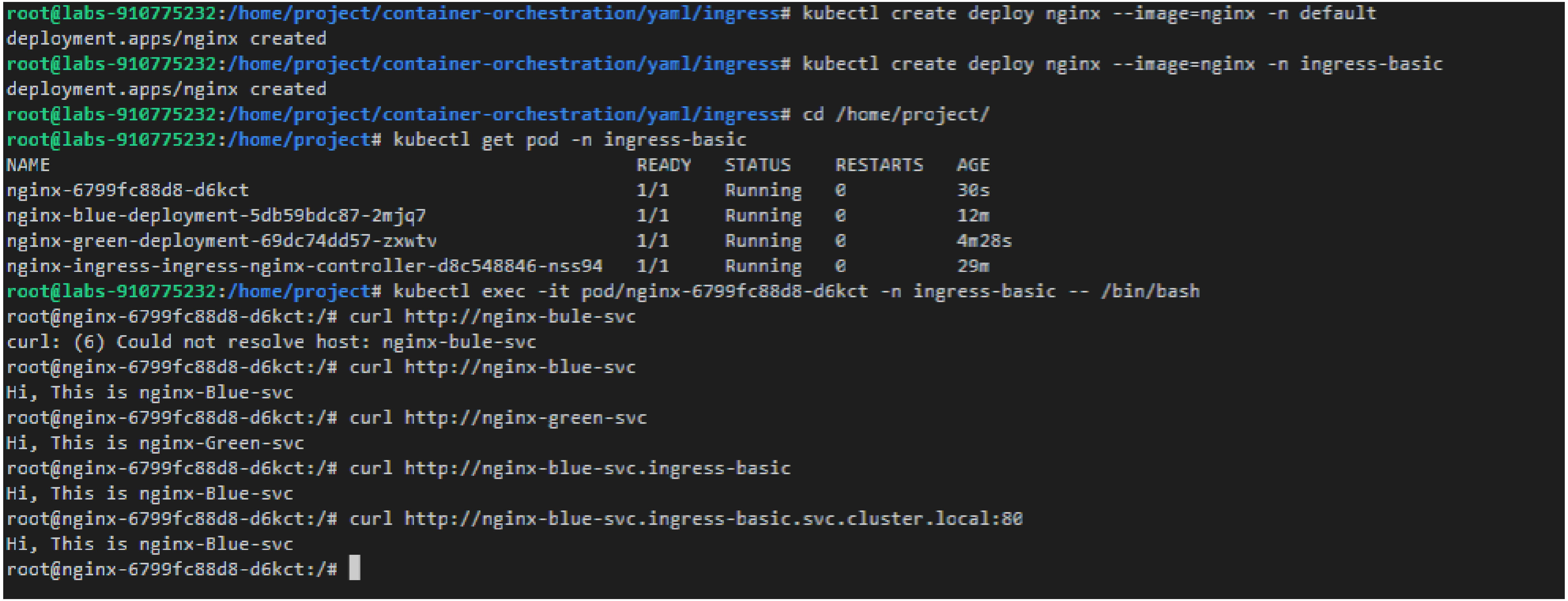


a5d4a5317488241f1b6235486a6458a9-1402221302.ap-northeast-2.elb.amazonaws.com/green

Hi, This is nginx-Green-svc

Cluster에서의 Web-URL 원리

- nginx 이미지를 가지고, default 및 ingress-basic 네임스페이스 2곳에 배포(deploy 생성).
- 각 namespace에서 서비스 호출하기



```
root@labs-910775232:/home/project/container-orchestration/yaml/ingress# kubectl create deploy nginx --image=nginx -n default
deployment.apps/nginx created
root@labs-910775232:/home/project/container-orchestration/yaml/ingress# kubectl create deploy nginx --image=nginx -n ingress-basic
deployment.apps/nginx created
root@labs-910775232:/home/project/container-orchestration/yaml/ingress# cd /home/project/
root@labs-910775232:/home/project# kubectl get pod -n ingress-basic
NAME                                                     READY   STATUS    RESTARTS   AGE
nginx-6799fc88d8-d6kct                                   1/1     Running   0          30s
nginx-blue-deployment-5db59bdc87-2mjq7                   1/1     Running   0          12m
nginx-green-deployment-69dc74dd57-zxwtv                  1/1     Running   0          4m28s
nginx-ingress-ingress-nginx-controller-d8c548846-nss94   1/1     Running   0          29m
root@labs-910775232:/home/project# kubectl exec -it pod/nginx-6799fc88d8-d6kct -n ingress-basic -- /bin/bash
root@nginx-6799fc88d8-d6kct:/# curl http://nginx-bule-svc
curl: (6) Could not resolve host: nginx-bule-svc
root@nginx-6799fc88d8-d6kct:/# curl http://nginx-blue-svc
Hi, This is nginx-Blue-svc
root@nginx-6799fc88d8-d6kct:/# curl http://nginx-green-svc
Hi, This is nginx-Green-svc
root@nginx-6799fc88d8-d6kct:/# curl http://nginx-blue-svc.ingress-basic
Hi, This is nginx-Blue-svc
root@nginx-6799fc88d8-d6kct:/# curl http://nginx-blue-svc.ingress-basic.svc.cluster.local:80
Hi, This is nginx-Blue-svc
root@nginx-6799fc88d8-d6kct:/# 
```