Day 2

Info - SOLID Design Principle

```
S - Single Responsibility Principle (SRP)
O - Open Closed Principle (OCP)
L - Liskov Substitution Principle (LSP)
I - Interface Seggregation
D - Dependency Injection or Dependency Inversion or Inversion of
Control(IOC)
```

Single Reponsibility Principle

```
- One component should do just one thing
- One component should represent a single entity/object
```

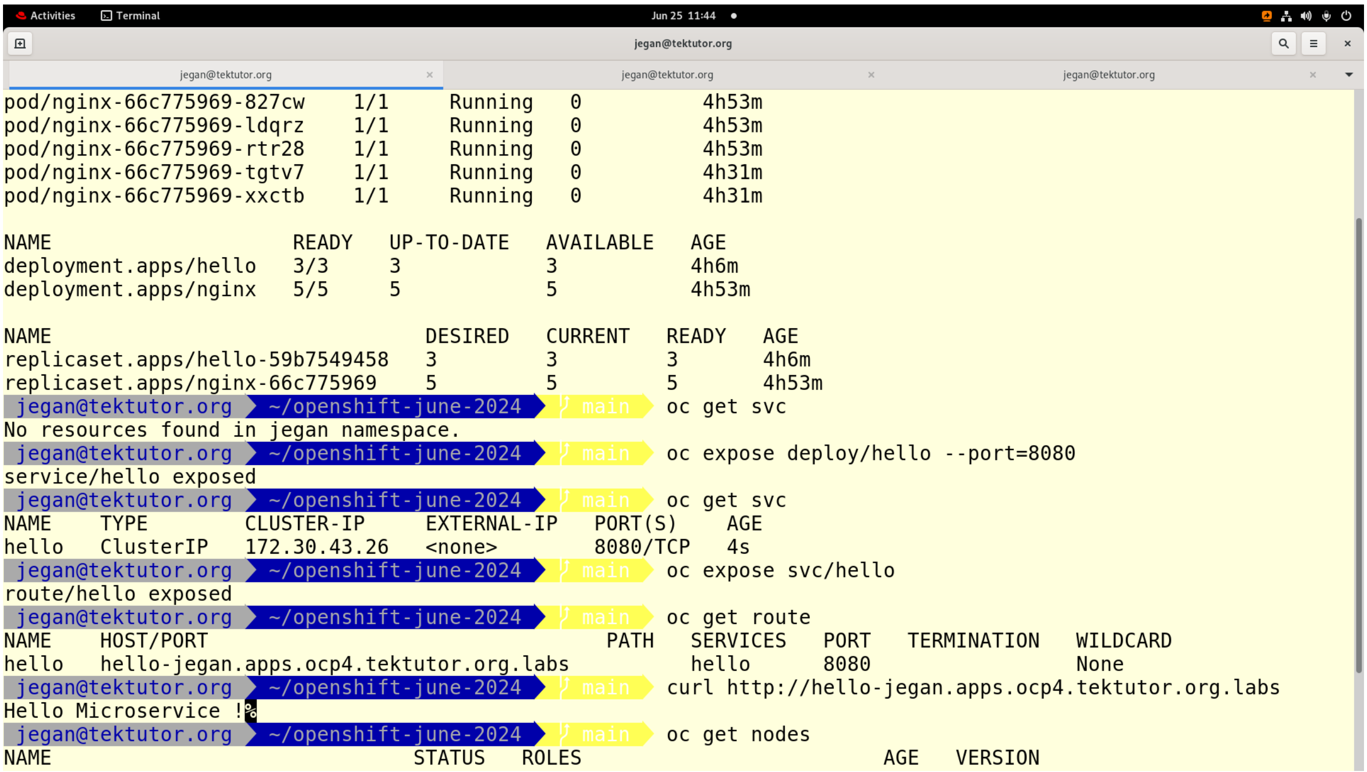
Info - ReplicationController

```
- In older versions of Kubernetes, stateless application were deployed as
ReplicationController
- ReplicationController supports
  - Rolling update
  - Scale up/down
- ReplicationController doesn't support declaratively performing scale
up/down
- ReplicationController doesn't support declaratively performing rolling
update
- For these reasons, latest versions of Kubernetes, they introducted
Deployment & ReplicaSet as an alternate to ReplicationController
- Deployment supports rolling update
- ReplicaSet supports scale up/down
- Deployment supports declaratively performing rolling update and scale
up/down
- In Openshift, before the Deployment and ReplicaSet was introducted, they
wanted to support scale up/down and rolling updte in declarative style,
hence they created DeploymentConfig
- DeploymentConfig internally used ReplicationController
- Once the Deployment & ReplicaSet was introduced in Kubernetes, Openshift
deprecated the use of DeploymentConfig
- Hence, new application deployment should avoid using DeploymentConfig and
ReplicationController.  Instead, we should consider using Deployment &
ReplicaSet
```

Lab - Create a public url using route for deployment

```
oc get deploy
oc expose deploy/hello --port=8080
oc get svc
oc expose svc/hello
oc get route
curl http://hello-jegan.apps.ocp4.tektutor.org.labs
```

Expected output



```
pod/nginx-66c775969-827cw    1/1     Running   0          4h53m
pod/nginx-66c775969-ldqrz    1/1     Running   0          4h53m
pod/nginx-66c775969-rtr28    1/1     Running   0          4h53m
pod/nginx-66c775969-tgtv7    1/1     Running   0          4h31m
pod/nginx-66c775969-xxctb    1/1     Running   0          4h31m

NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/hello    3/3     3            3           4h6m
deployment.apps/nginx    5/5     5            5           4h53m

NAME                               DESIRED   CURRENT   READY   AGE
replicaset.apps/hello-59b7549458   3         3         3       4h6m
replicaset.apps/nginx-66c775969    5         5         5       4h53m
 jegan@tektutor.org  ~/openshift-june-2024  main  oc get svc
No resources found in jegan namespace.
 jegan@tektutor.org  ~/openshift-june-2024  main  oc expose deploy/hello --port=8080
service/hello exposed
 jegan@tektutor.org  ~/openshift-june-2024  main  oc get svc
NAME    TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
hello   ClusterIP   172.30.43.26   <none>        8080/TCP   4s
 jegan@tektutor.org  ~/openshift-june-2024  main  oc expose svc/hello
route/hello exposed
 jegan@tektutor.org  ~/openshift-june-2024  main  oc get route
NAME    HOST/PORT                                   PATH   SERVICES   PORT   TERMINATION   WILDCARD
hello   hello-jegan.apps.ocp4.tektutor.org.labs            hello      8080                 None
 jegan@tektutor.org  ~/openshift-june-2024  main  curl http://hello-jegan.apps.ocp4.tektutor.org.labs
Hello Microservice !%
 jegan@tektutor.org  ~/openshift-june-2024  main  oc get nodes
NAME                                  STATUS   ROLES                          AGE   VERSION
```

Points to note

```
- Route is a new feature introduced in OpenShift
- Route is based on Kubernetes Ingress
- Route provides a user-friendly public url to access the application from outside the cluster
- This is a better alternate for Kubernetes Node Port service
```

Lab - Ingress

In case you haven't already deployed nginx, you need to deploy nginx as shown below

```
oc project jegan
oc create deployment nginx --image=bitnami/nginx:latest --replicas=3
oc expose deploy/nginx --port=8080
oc get svc
oc describe svc/nginx
```

In case you haven't already deployed hello, you need to deploy hello microservice as shown below

```
oc project jegan
oc create deployment hello --image=tektutor/hello:4.0 --replicas=3
oc expose deploy/hello --port=8080
oc get svc
oc describe svc/nginx
```

Now let's create the ingress forwarding rules to the above services based on path /nginx or /hello.

```
cd ~/openshift-june-2024
git pull
cd Day2/ingress
cat ingress.yml
oc apply -f ingress.yml
oc get ingress
oc describe ingress/tektutor
curl http://tektutor.apps.ocp4.tektutor.org.labs/nginx
curl http://tektutor.apps.ocp4.tektutor.org.labs/hello
```

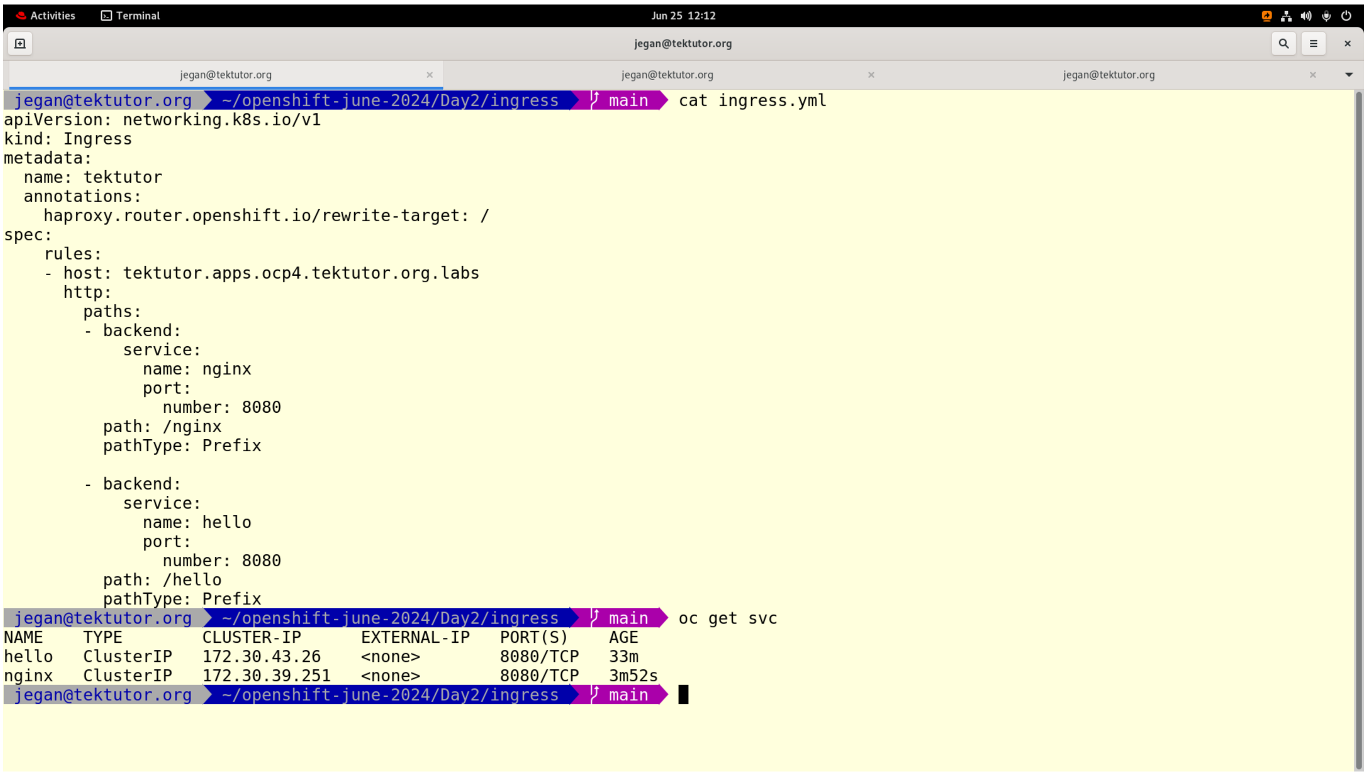
The domain must match with the registered domain in ingress controller

```
oc describe ingresscontroller default -n openshift-ingress-operator | grep Domain
```

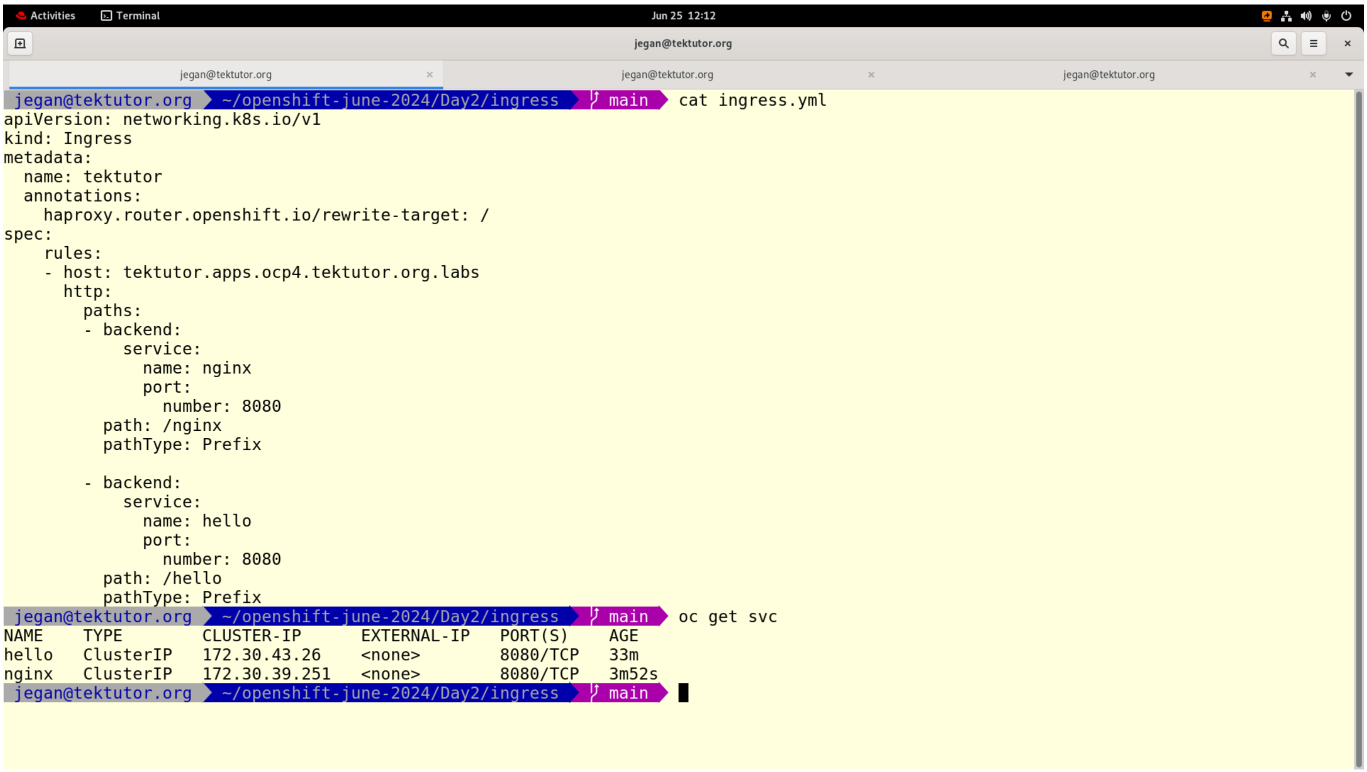
Points to note

```
- Ingress is a set of forwarding rules
- Ingress rules are picked by Ingress Controller
- There are two commonly used Ingress Controlls
  1. Nginx Ingress Controller
  2. HAProxy Ingress Controller
- For Ingress to work we need 3 major components within openshift/kubernetes cluster
  1. Ingress Forwarding rules (user-defined)
  2. Ingress Controller
  3. Load balancer
```

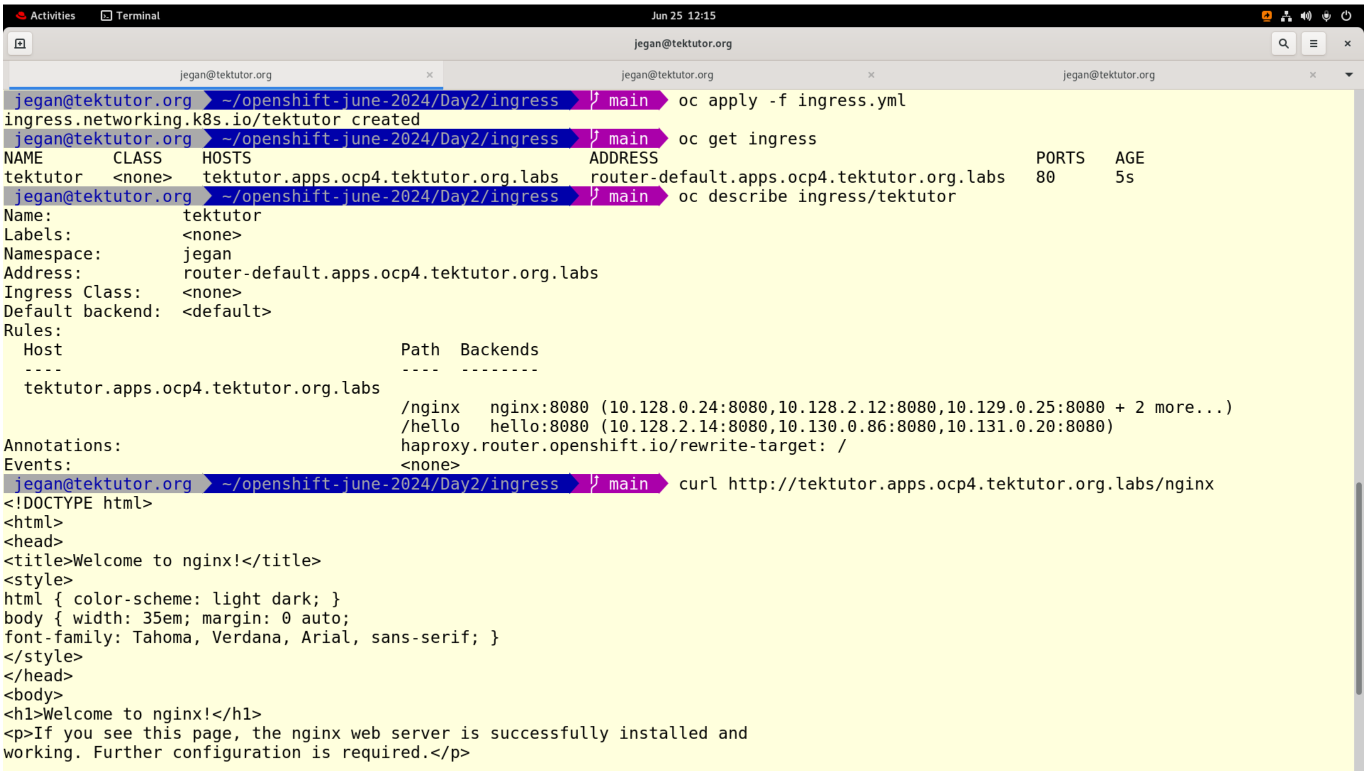
Expected output



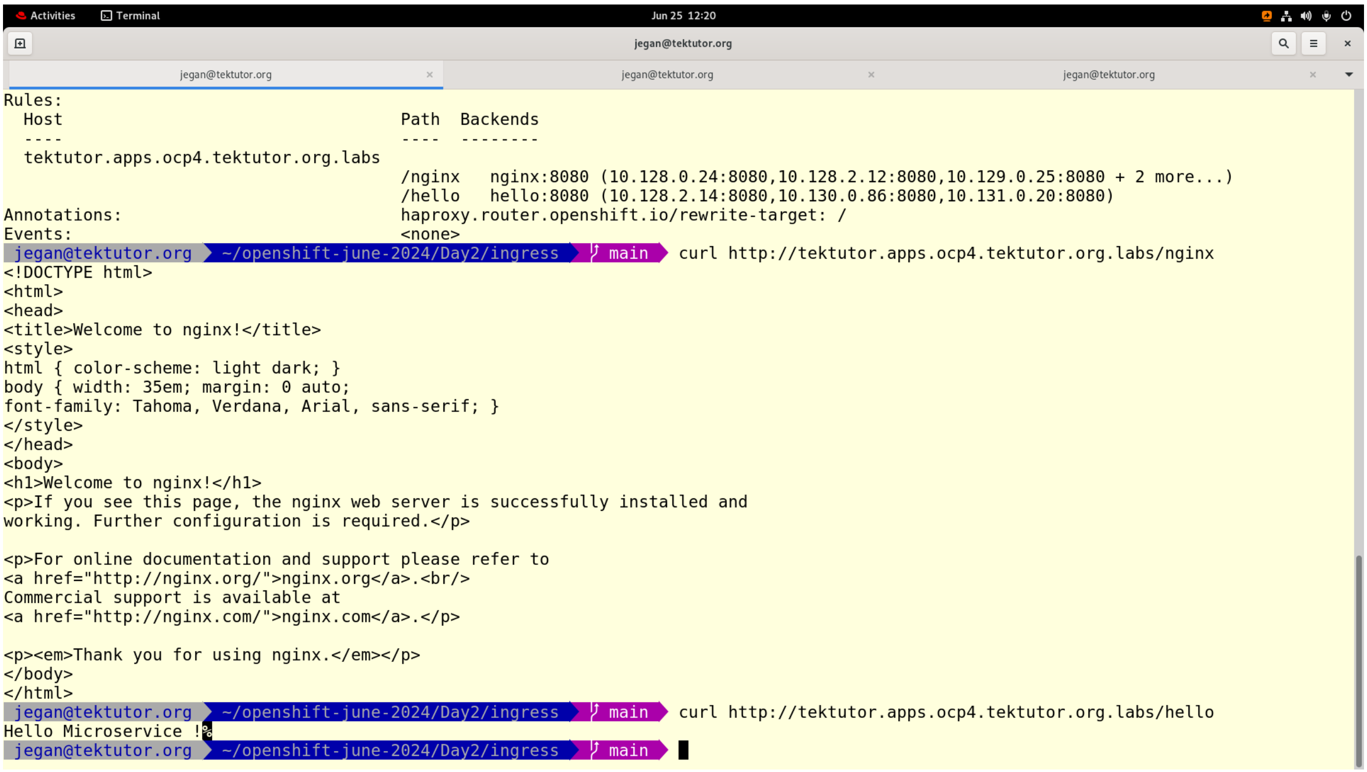
```
 jegan@tektutor.org  ~/openshift-june-2024/Day2/ingress  main  cat ingress.yml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: tektutor
  annotations:
    haproxy.router.openshift.io/rewrite-target: /
spec:
    rules:
    - host: tektutor.apps.ocp4.tektutor.org.labs
      http:
        paths:
        - backend:
            service:
              name: nginx
              port:
                number: 8080
          path: /nginx
          pathType: Prefix

        - backend:
            service:
              name: hello
              port:
                number: 8080
          path: /hello
          pathType: Prefix
 jegan@tektutor.org  ~/openshift-june-2024/Day2/ingress  main  oc get svc
NAME    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
hello   ClusterIP   172.30.43.26    <none>        8080/TCP   33m
nginx   ClusterIP   172.30.39.251   <none>        8080/TCP   3m52s
 jegan@tektutor.org  ~/openshift-june-2024/Day2/ingress  main 
```


Activities
Terminal
Jun 25 12:12
jegan@tektutor.org

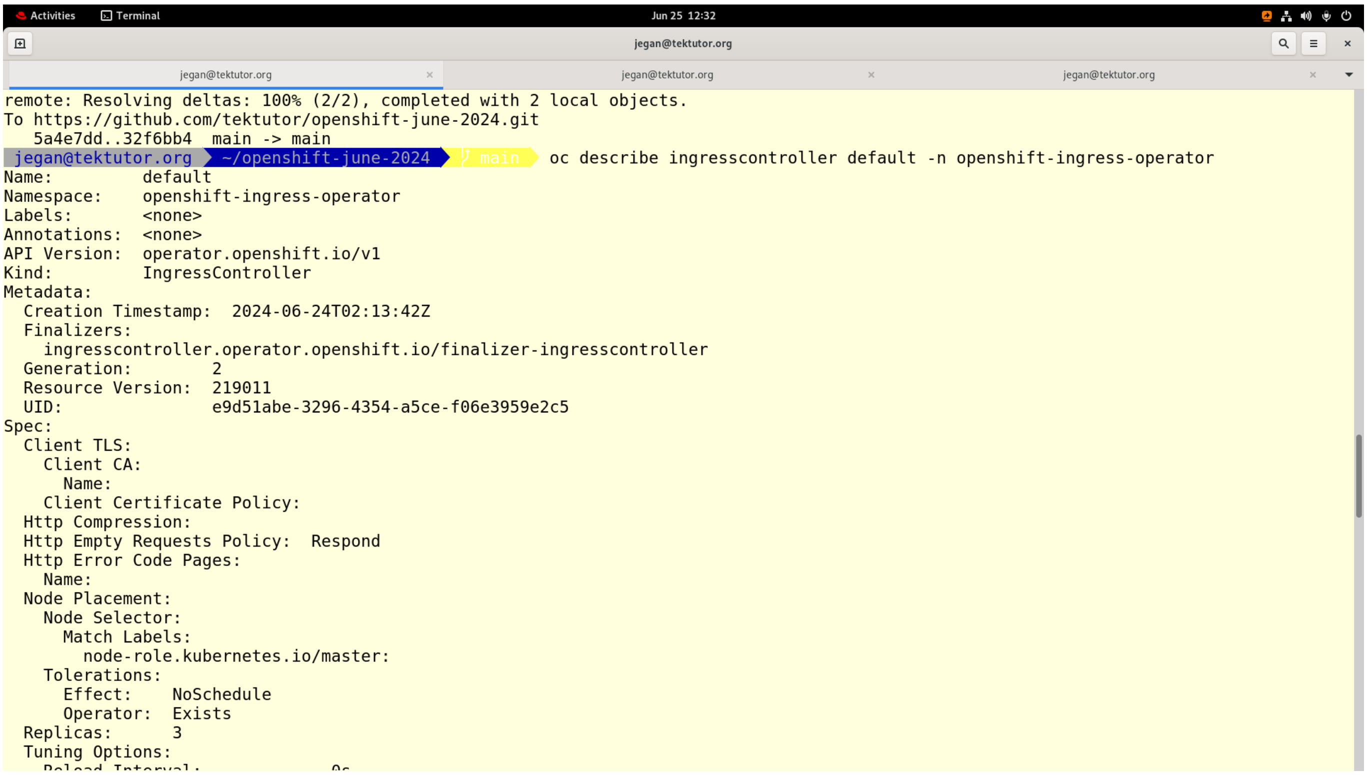
```
 jegan@tektutor.org  ~/openshift-june-2024/Day2/ingress  main  cat ingress.yml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: tektutor
  annotations:
    haproxy.router.openshift.io/rewrite-target: /
spec:
    rules:
    - host: tektutor.apps.ocp4.tektutor.org.labs
      http:
        paths:
        - backend:
            service:
              name: nginx
              port:
                number: 8080
          path: /nginx
          pathType: Prefix

        - backend:
            service:
              name: hello
              port:
                number: 8080
          path: /hello
          pathType: Prefix
 jegan@tektutor.org  ~/openshift-june-2024/Day2/ingress  main  oc get svc
NAME    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
hello   ClusterIP   172.30.43.26    <none>        8080/TCP   33m
nginx   ClusterIP   172.30.39.251   <none>        8080/TCP   3m52s
 jegan@tektutor.org  ~/openshift-june-2024/Day2/ingress  main 
```



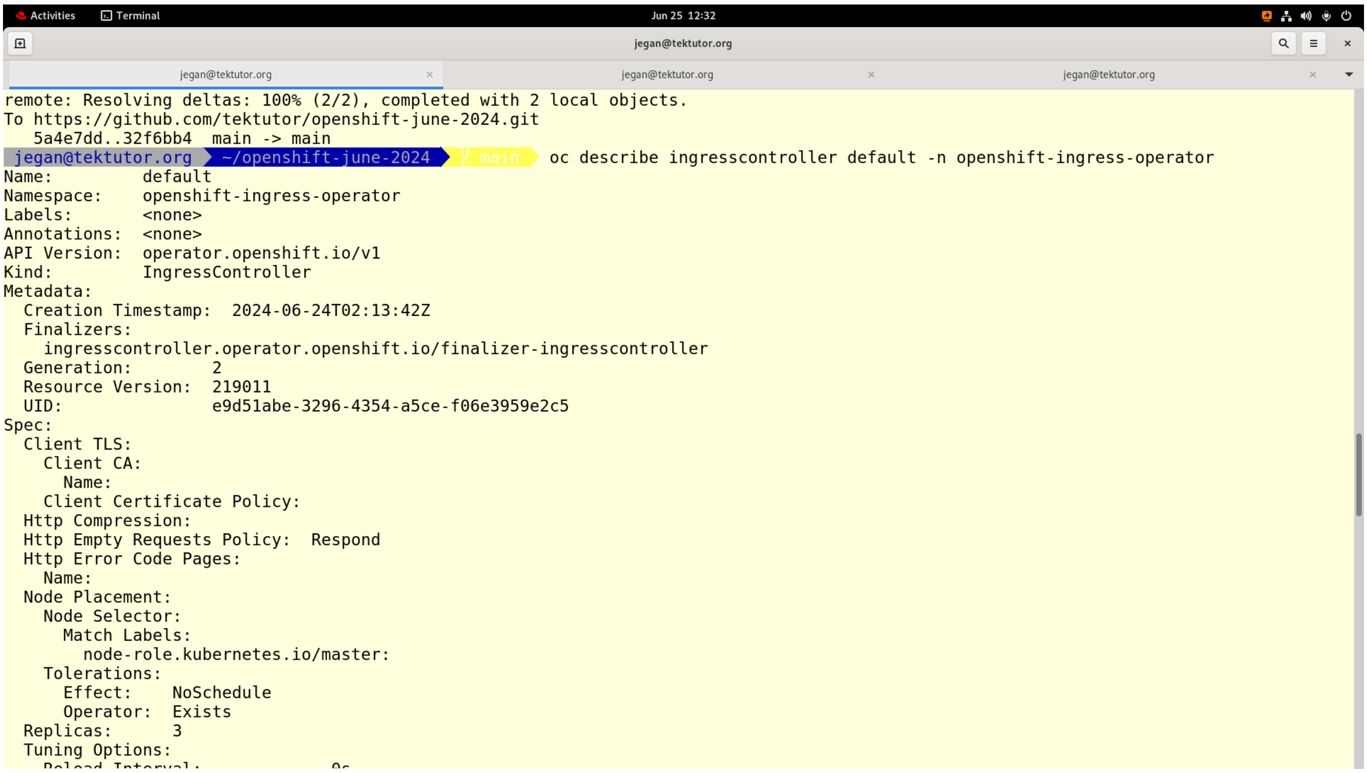
```
 jegan@tektutor.org  ~/openshift-june-2024/Day2/ingress  main  oc apply -f ingress.yml
ingress.networking.k8s.io/tektutor created
 jegan@tektutor.org  ~/openshift-june-2024/Day2/ingress  main  oc get ingress
NAME       CLASS    HOSTS                                ADDRESS                                  PORTS   AGE
tektutor   <none>   tektutor.apps.ocp4.tektutor.org.labs   router-default.apps.ocp4.tektutor.org.labs   80      5s
 jegan@tektutor.org  ~/openshift-june-2024/Day2/ingress  main  oc describe ingress/tektutor
Name:             tektutor
Labels:           <none>
Namespace:        jegan
Address:          router-default.apps.ocp4.tektutor.org.labs
Ingress Class:    <none>
Default backend:  <default>
Rules:
  Host                                 Path  Backends
  ----                                 ----  --------
  tektutor.apps.ocp4.tektutor.org.labs
                                       /nginx   nginx:8080 (10.128.0.24:8080,10.128.2.12:8080,10.129.0.25:8080 + 2 more...)
                                       /hello   hello:8080 (10.128.2.14:8080,10.130.0.86:8080,10.131.0.20:8080)
Annotations:                           haproxy.router.openshift.io/rewrite-target: /
Events:                                <none>
 jegan@tektutor.org  ~/openshift-june-2024/Day2/ingress  main  curl http://tektutor.apps.ocp4.tektutor.org.labs/nginx
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
```



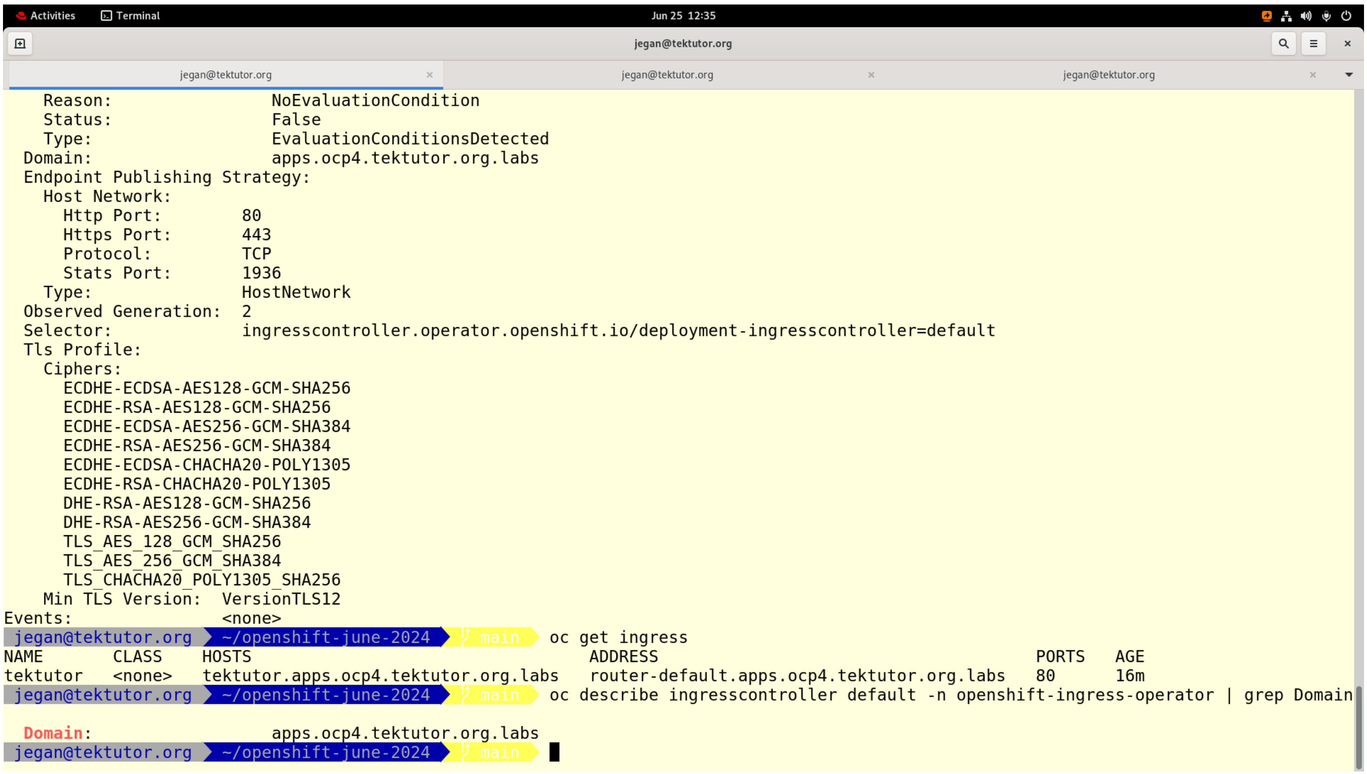
```
Rules:
  Host                                 Path  Backends
  ----                                 ----  --------
  tektutor.apps.ocp4.tektutor.org.labs
                                       /nginx   nginx:8080 (10.128.0.24:8080,10.128.2.12:8080,10.129.0.25:8080 + 2 more...)
                                       /hello   hello:8080 (10.128.2.14:8080,10.130.0.86:8080,10.131.0.20:8080)
Annotations:                           haproxy.router.openshift.io/rewrite-target: /
Events:                                <none>
 jegan@tektutor.org  ~/openshift-june-2024/Day2/ingress  main  curl http://tektutor.apps.ocp4.tektutor.org.labs/nginx
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
 jegan@tektutor.org  ~/openshift-june-2024/Day2/ingress  main  curl http://tektutor.apps.ocp4.tektutor.org.labs/hello
Hello Microservice !%
 jegan@tektutor.org  ~/openshift-june-2024/Day2/ingress  main 
```



```
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/tektutor/openshift-june-2024.git
   5a4e7dd..32f6bb4  main -> main
 jegan@tektutor.org  ~/openshift-june-2024  main  oc describe ingresscontroller default -n openshift-ingress-operator
Name:         default
Namespace:    openshift-ingress-operator
Labels:       <none>
Annotations:  <none>
API Version:  operator.openshift.io/v1
Kind:         IngressController
Metadata:
  Creation Timestamp:  2024-06-24T02:13:42Z
  Finalizers:
    ingresscontroller.operator.openshift.io/finalizer-ingresscontroller
  Generation:        2
  Resource Version:  219011
  UID:               e9d51abe-3296-4354-a5ce-f06e3959e2c5
Spec:
  Client TLS:
    Client CA:
      Name:
    Client Certificate Policy:
  Http Compression:
  Http Empty Requests Policy:  Respond
  Http Error Code Pages:
    Name:
  Node Placement:
    Node Selector:
      Match Labels:
        node-role.kubernetes.io/master:
    Tolerations:
      Effect:    NoSchedule
      Operator:  Exists
  Replicas:      3
  Tuning Options:
    Reload Interval:             0s
```



```
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/tektutor/openshift-june-2024.git
   5a4e7dd..32f6bb4  main -> main
 jegan@tektutor.org  ~/openshift-june-2024  main  oc describe ingresscontroller default -n openshift-ingress-operator
Name:         default
Namespace:    openshift-ingress-operator
Labels:       <none>
Annotations:  <none>
API Version:  operator.openshift.io/v1
Kind:         IngressController
Metadata:
  Creation Timestamp:  2024-06-24T02:13:42Z
  Finalizers:
    ingresscontroller.operator.openshift.io/finalizer-ingresscontroller
  Generation:        2
  Resource Version:  219011
  UID:               e9d51abe-3296-4354-a5ce-f06e3959e2c5
Spec:
  Client TLS:
    Client CA:
      Name:
    Client Certificate Policy:
  Http Compression:
  Http Empty Requests Policy:  Respond
  Http Error Code Pages:
    Name:
  Node Placement:
    Node Selector:
      Match Labels:
        node-role.kubernetes.io/master:
    Tolerations:
      Effect:    NoSchedule
      Operator:  Exists
  Replicas:      3
  Tuning Options:
    Reload Interval:             0s
```



```
    Reason:                NoEvaluationCondition
    Status:                False
    Type:                  EvaluationConditionsDetected
  Domain:                  apps.ocp4.tektutor.org.labs
  Endpoint Publishing Strategy:
    Host Network:
      Http Port:        80
      Https Port:       443
      Protocol:         TCP
      Stats Port:       1936
    Type:               HostNetwork
  Observed Generation:  2
  Selector:             ingresscontroller.operator.openshift.io/deployment-ingresscontroller=default
  Tls Profile:
    Ciphers:
      ECDHE-ECDSA-AES128-GCM-SHA256
      ECDHE-RSA-AES128-GCM-SHA256
      ECDHE-ECDSA-AES256-GCM-SHA384
      ECDHE-RSA-AES256-GCM-SHA384
      ECDHE-ECDSA-CHACHA20-POLY1305
      ECDHE-RSA-CHACHA20-POLY1305
      DHE-RSA-AES128-GCM-SHA256
      DHE-RSA-AES256-GCM-SHA384
      TLS_AES_128_GCM_SHA256
      TLS_AES_256_GCM_SHA384
      TLS_CHACHA20_POLY1305_SHA256
    Min TLS Version:  VersionTLS12
Events:               <none>
 jegan@tektutor.org  ~/openshift-june-2024   main  oc get ingress
NAME       CLASS    HOSTS                                  ADDRESS                                     PORTS   AGE
tektutor   <none>   tektutor.apps.ocp4.tektutor.org.labs   router-default.apps.ocp4.tektutor.org.labs   80      16m
 jegan@tektutor.org  ~/openshift-june-2024   main  oc describe ingresscontroller default -n openshift-ingress-operator | grep Domain

  Domain:                  apps.ocp4.tektutor.org.labs
 jegan@tektutor.org  ~/openshift-june-2024   main  
```

Info - Subnet

```
- If we take IPV4 IP addresses it is 32 bits(4 bytes)
- It has 4 Octets
  - A.B.C.D
  - A is 1 Byte(8 bits)
  - B is 1 Byte(8 bits)
  - C is 1 Byte(8 bits)
  - D is 1 Byte(8 bits)
- Consider this Subnet - 10.128.0.0/24 ( 256 IP Addresses are supported )
- What is IP Address in the above Subnet
  - 10.128.0.0
  - 10.128.0.1
  - 10.128.0.2
  - ...
  - 10.128.0.255
- The 24 in 10.128.0.0/24 indicates how many bits from left to right are
fixed

- From the subnet 10.244.0.0/16 compute 5 Subnets for master-1, master-
2,master-3, worker-1 and worker-2 nodes
  - Master 1 - Subnet ( 10.244.1.0/24 )
  - Master 2 - Subnet ( 10.244.2.0/24 )
  - Master 3 - Subnet ( 10.244.3.0/24 )
  - Worker 1 - Subnet ( 10.244.4.0/24 )
  - Worker 2 - Subnet ( 10.244.5.0/24 )
```

Info - Private IP

```
- Private IP are accessible only on the same machine
```

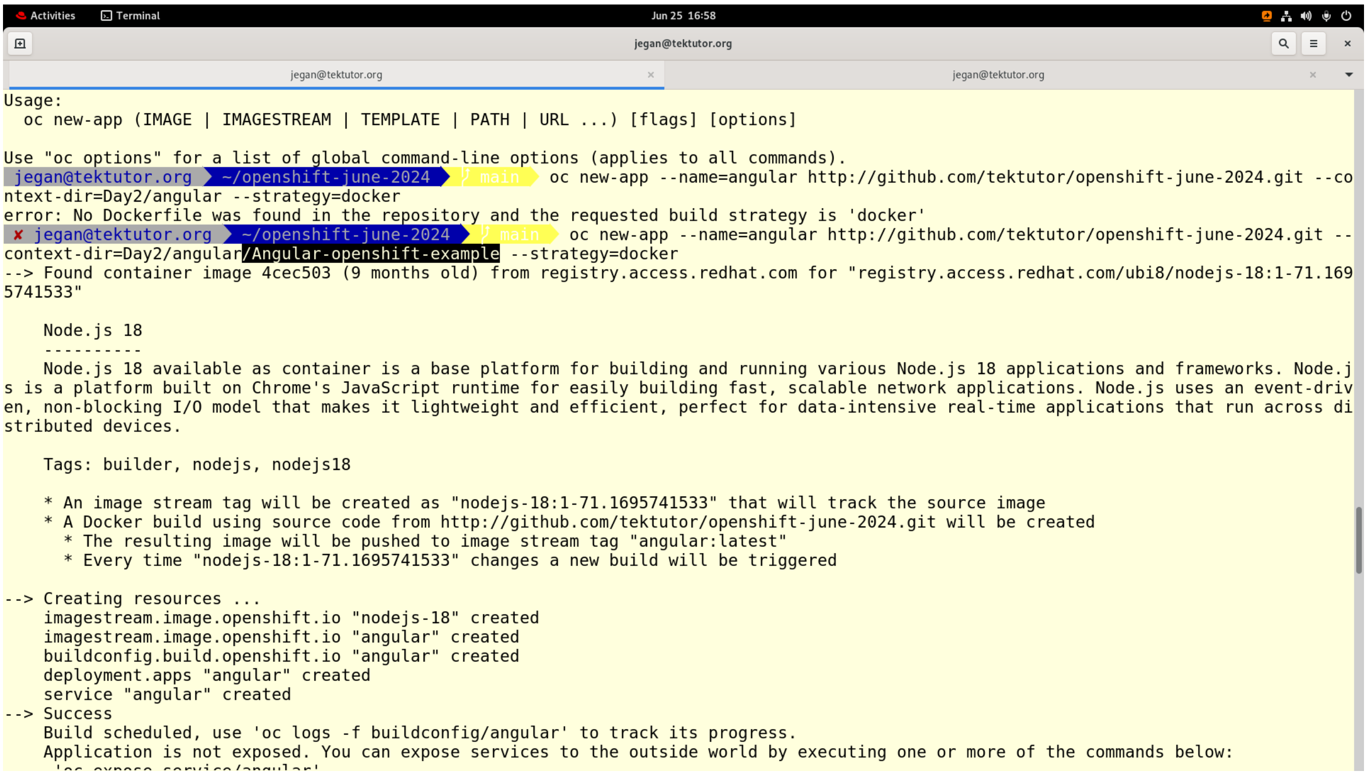
Lab - Deploying Angular application into openshift using Docker strategy cloning source from GitHub

What does the below command do?

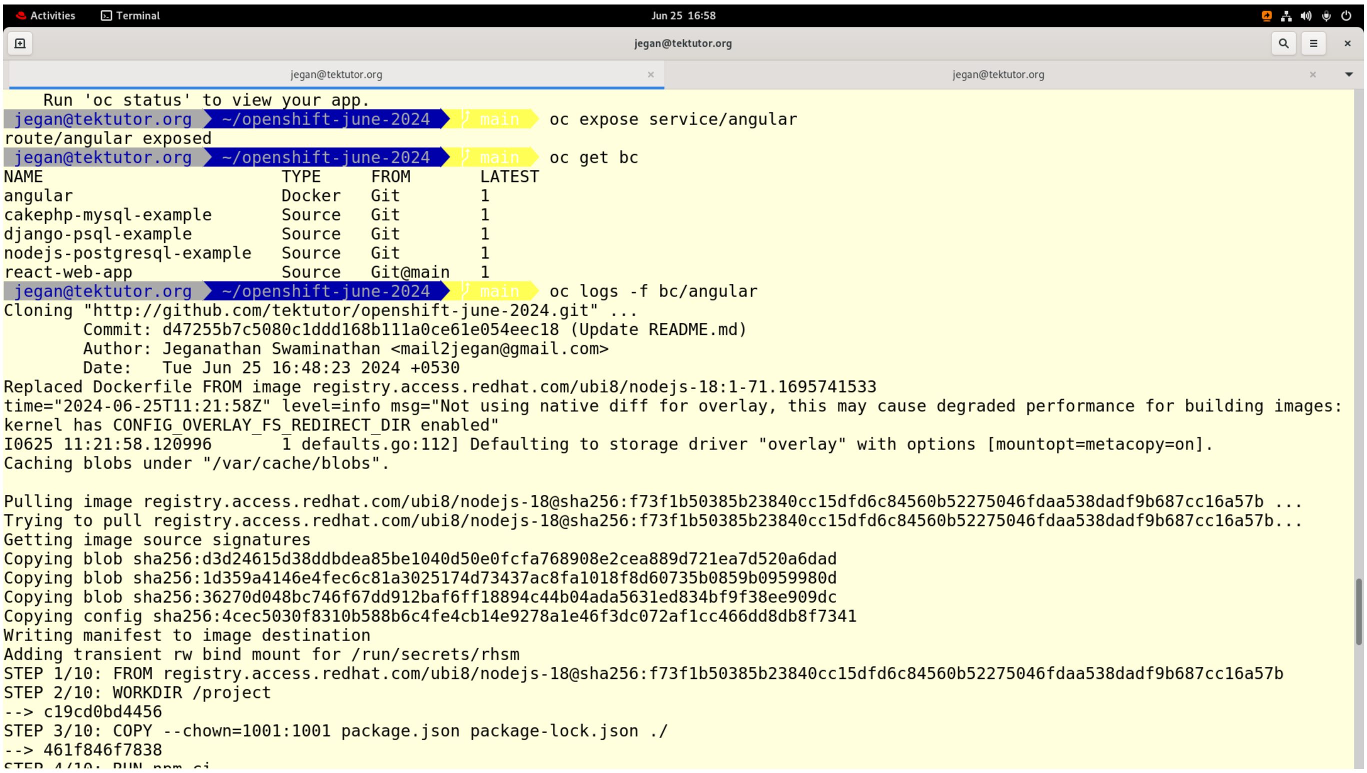
```
- The below command will deploy angularjs application into openshift by cloning the source code from GitHub repo
- navigates to Day2/angular/Angular-openshift-example folder
- since we have mentioned docker strategy, it looks for Dockerfile under Day2/angular/Angular-openshift-example folder
- Openshift creates a buildconfig with the Dockerfile, the output of the buildconfig will be a docker image which will get pushed into Openshift internal container registry
- Using the newly build image, it automatically deploys the application and creates a service for the deployment
- We need to manually create a route to access the application from outside the cluster
```

```
oc project jegan

oc new-app --name=angular https://github.com/tektutor/openshift-june-2024.git --context-dir=Day2/angular/Angular-openshift-example --strategy=docker

oc expose svc/angular
oc get buildconfigs
oc logs -f bc/angular
```

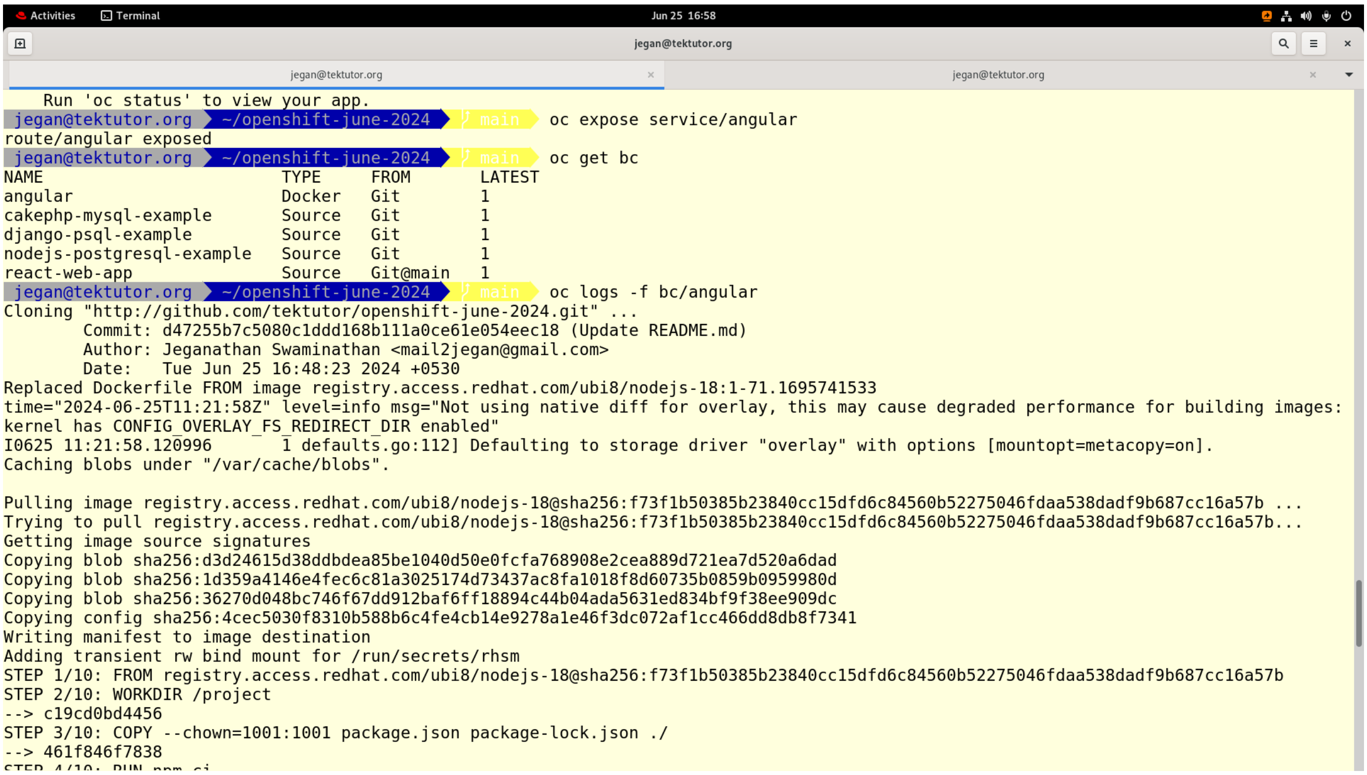
Expected output



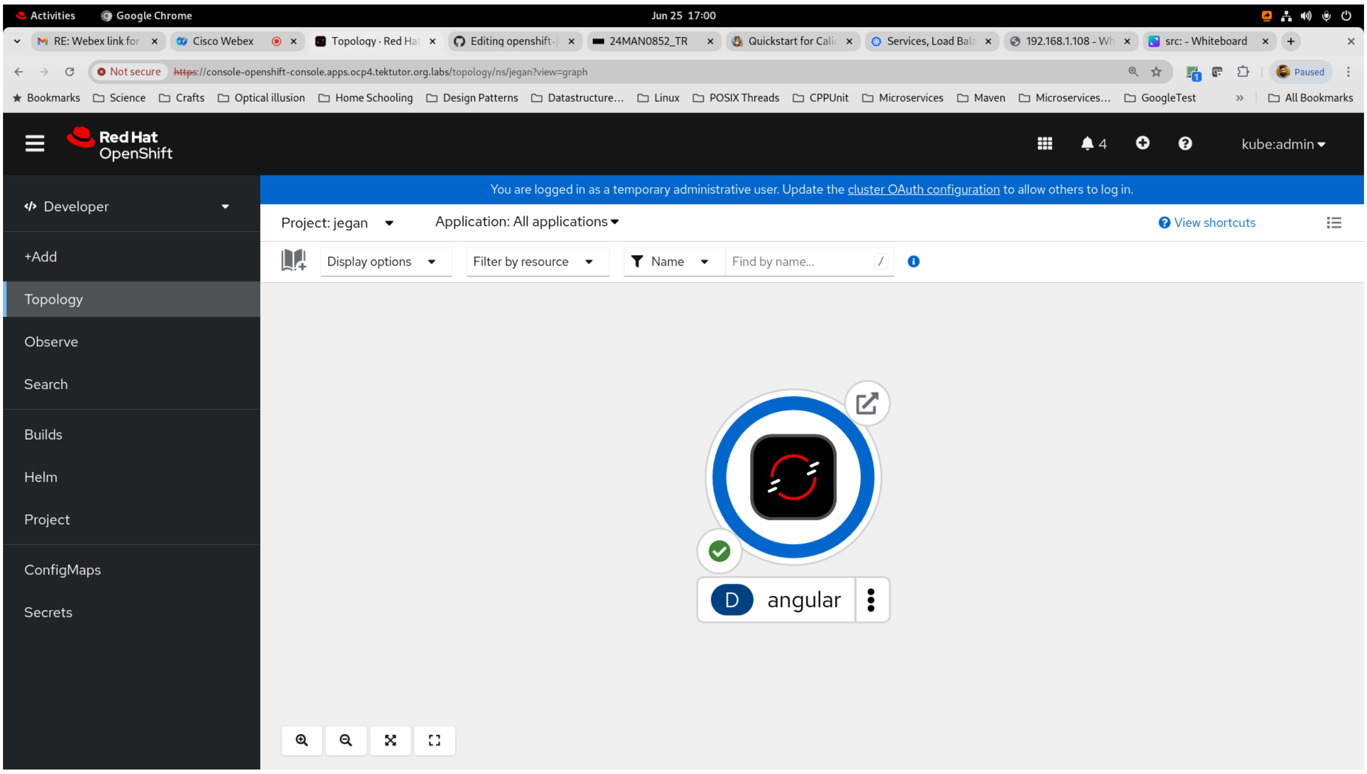
```
Usage:
  oc new-app (IMAGE | IMAGESTREAM | TEMPLATE | PATH | URL ...) [flags] [options]

Use "oc options" for a list of global command-line options (applies to all commands).
 jegan@tektutor.org  ~/openshift-june-2024  main  oc new-app --name=angular http://github.com/tektutor/openshift-june-2024.git --co
ntext-dir=Day2/angular --strategy=docker
error: No Dockerfile was found in the repository and the requested build strategy is 'docker'
 ✘ jegan@tektutor.org  ~/openshift-june-2024  main  oc new-app --name=angular http://github.com/tektutor/openshift-june-2024.git --
context-dir=Day2/angular/Angular-openshift-example --strategy=docker
--> Found container image 4cec503 (9 months old) from registry.access.redhat.com for "registry.access.redhat.com/ubi8/nodejs-18:1-71.169
5741533"

    Node.js 18
    ----------
    Node.js 18 available as container is a base platform for building and running various Node.js 18 applications and frameworks. Node.j
s is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driv
en, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across di
stributed devices.

    Tags: builder, nodejs, nodejs18

    * An image stream tag will be created as "nodejs-18:1-71.1695741533" that will track the source image
    * A Docker build using source code from http://github.com/tektutor/openshift-june-2024.git will be created
      * The resulting image will be pushed to image stream tag "angular:latest"
      * Every time "nodejs-18:1-71.1695741533" changes a new build will be triggered

--> Creating resources ...
    imagestream.image.openshift.io "nodejs-18" created
    imagestream.image.openshift.io "angular" created
    buildconfig.build.openshift.io "angular" created
    deployment.apps "angular" created
    service "angular" created
--> Success
    Build scheduled, use 'oc logs -f buildconfig/angular' to track its progress.
    Application is not exposed. You can expose services to the outside world by executing one or more of the commands below:
     'oc expose service/angular'
```



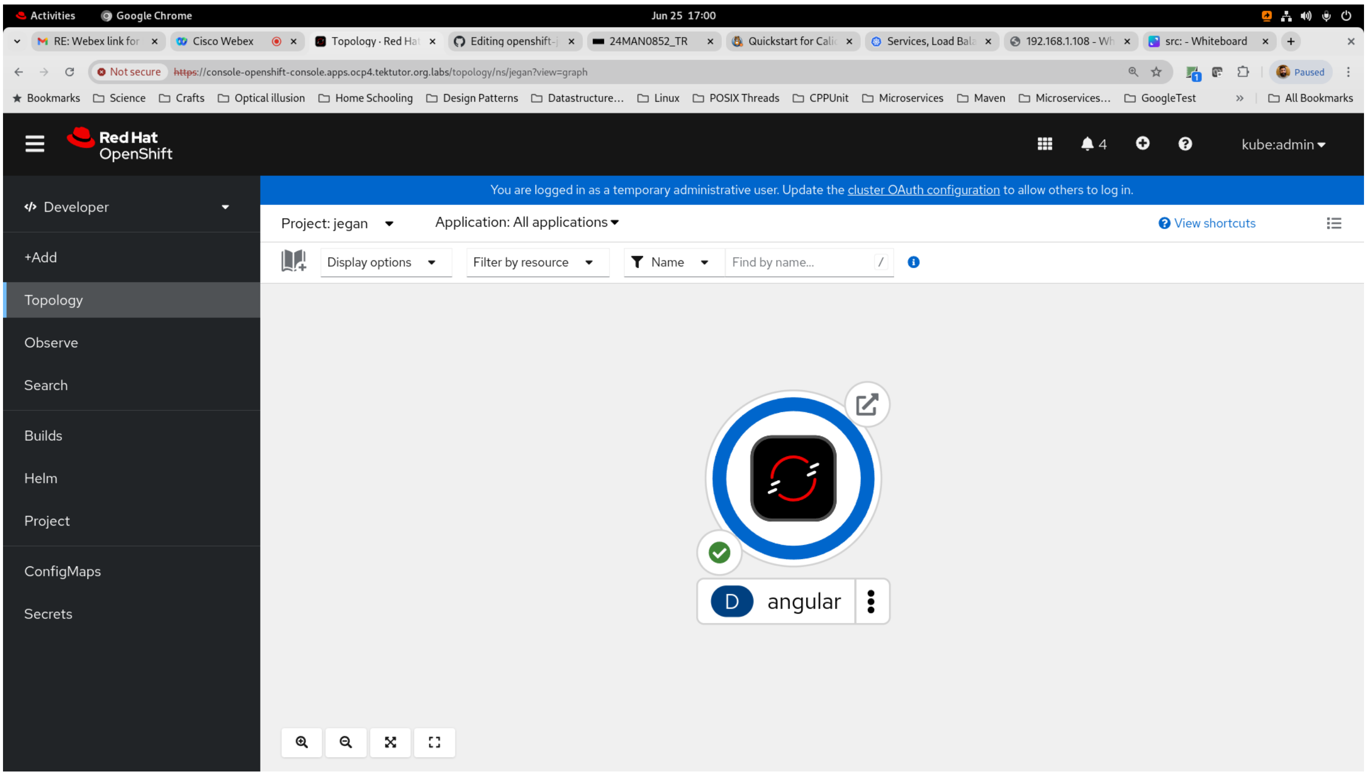
```
    Run 'oc status' to view your app.
 jegan@tektutor.org  ~/openshift-june-2024  main  oc expose service/angular
route/angular exposed
 jegan@tektutor.org  ~/openshift-june-2024  main  oc get bc
NAME                        TYPE     FROM       LATEST
angular                     Docker   Git        1
cakephp-mysql-example       Source   Git        1
django-psql-example         Source   Git        1
nodejs-postgresql-example   Source   Git        1
react-web-app               Source   Git@main   1
 jegan@tektutor.org  ~/openshift-june-2024  main  oc logs -f bc/angular
Cloning "http://github.com/tektutor/openshift-june-2024.git" ...
        Commit: d47255b7c5080c1ddd168b111a0ce61e054eec18 (Update README.md)
        Author: Jeganathan Swaminathan <mail2jegan@gmail.com>
        Date:   Tue Jun 25 16:48:23 2024 +0530
Replaced Dockerfile FROM image registry.access.redhat.com/ubi8/nodejs-18:1-71.1695741533
time="2024-06-25T11:21:58Z" level=info msg="Not using native diff for overlay, this may cause degraded performance for building images:
kernel has CONFIG_OVERLAY_FS_REDIRECT_DIR enabled"
I0625 11:21:58.120996       1 defaults.go:112] Defaulting to storage driver "overlay" with options [mountopt=metacopy=on].
Caching blobs under "/var/cache/blobs".

Pulling image registry.access.redhat.com/ubi8/nodejs-18@sha256:f73f1b50385b23840cc15dfd6c84560b52275046fdaa538dadf9b687cc16a57b ...
Trying to pull registry.access.redhat.com/ubi8/nodejs-18@sha256:f73f1b50385b23840cc15dfd6c84560b52275046fdaa538dadf9b687cc16a57b...
Getting image source signatures
Copying blob sha256:d3d24615d38ddbdea85be1040d50e0fcfa768908e2cea889d721ea7d520a6dad
Copying blob sha256:1d359a4146e4fec6c81a3025174d73437ac8fa1018f8d60735b0859b0959980d
Copying blob sha256:36270d048bc746f67dd912baf6ff18894c44b04ada5631ed834bf9f38ee909dc
Copying config sha256:4cec5030f8310b588b6c4fe4cb14e9278a1e46f3dc072af1cc466dd8db8f7341
Writing manifest to image destination
Adding transient rw bind mount for /run/secrets/rhsm
STEP 1/10: FROM registry.access.redhat.com/ubi8/nodejs-18@sha256:f73f1b50385b23840cc15dfd6c84560b52275046fdaa538dadf9b687cc16a57b
STEP 2/10: WORKDIR /project
--> c19cd0bd4456
STEP 3/10: COPY --chown=1001:1001 package.json package-lock.json ./
--> 461f846f7838
STEP 4/10: RUN npm ci
```



```
    Run 'oc status' to view your app.
 jegan@tektutor.org  ~/openshift-june-2024   main  oc expose service/angular
route/angular exposed
 jegan@tektutor.org  ~/openshift-june-2024   main  oc get bc
NAME                        TYPE     FROM       LATEST
angular                     Docker   Git        1
cakephp-mysql-example       Source   Git        1
django-psql-example         Source   Git        1
nodejs-postgresql-example   Source   Git        1
react-web-app               Source   Git@main   1
 jegan@tektutor.org  ~/openshift-june-2024   main  oc logs -f bc/angular
Cloning "http://github.com/tektutor/openshift-june-2024.git" ...
        Commit: d47255b7c5080c1ddd168b111a0ce61e054eec18 (Update README.md)
        Author: Jeganathan Swaminathan <mail2jegan@gmail.com>
        Date:   Tue Jun 25 16:48:23 2024 +0530
Replaced Dockerfile FROM image registry.redhat.com/ubi8/nodejs-18:1-71.1695741533
time="2024-06-25T11:21:58Z" level=info msg="Not using native diff for overlay, this may cause degraded performance for building images: kernel has CONFIG_OVERLAY_FS_REDIRECT_DIR enabled"
I0625 11:21:58.120996       1 defaults.go:112] Defaulting to storage driver "overlay" with options [mountopt=metacopy=on].
Caching blobs under "/var/cache/blobs".

Pulling image registry.access.redhat.com/ubi8/nodejs-18@sha256:f73f1b50385b23840cc15dfd6c84560b52275046fdaa538dadf9b687cc16a57b ...
Trying to pull registry.access.redhat.com/ubi8/nodejs-18@sha256:f73f1b50385b23840cc15dfd6c84560b52275046fdaa538dadf9b687cc16a57b...
Getting image source signatures
Copying blob sha256:d3d24615d38ddbdea85be1040d50e0fcfa768908e2cea889d721ea7d520a6dad
Copying blob sha256:1d359a4146e4fec6c81a3025174d73437ac8fa1018f8d60735b0859b0959980d
Copying blob sha256:36270d048bc746f67dd912baf6ff18894c44b04ada5631ed834bf9f38ee909dc
Copying config sha256:4cec5030f8310b588b6c4fe4cb14e9278a1e46f3dc072af1cc466dd8db8f7341
Writing manifest to image destination
Adding transient rw bind mount for /run/secrets/rhsm
STEP 1/10: FROM registry.access.redhat.com/ubi8/nodejs-18@sha256:f73f1b50385b23840cc15dfd6c84560b52275046fdaa538dadf9b687cc16a57b
STEP 2/10: WORKDIR /project
--> c19cd0bd4456
STEP 3/10: COPY --chown=1001:1001 package.json package-lock.json ./
--> 461f846f7838
STEP 4/10: RUN npm ci
```



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.