Use the **oc** CLI to login and authenticate to the environment.

```
oc login https://<IP Address>:8443  -u <team> -p <team>

    <IP Address>  - replace with instructor provided information
    <team>        - replace with team name
```

The instructor will provide the IP address needed to access the OCP cluster that will be used in this lab.

---

Follow the detailed step-by-step instructions if additional assistance is needed.

---

Using the **oc** CLI login to the environment.

| Item | Action |
|---|---|
| <IP Address> | Replace with instructor provided information |
| <team> | Replace with team name |

**oc login https://<IP Address>:8443 -u <team> -p <team>**

Once the command has completed a message will be displayed. The message will contain a count of projects available to the user. *XXX_* will provide the number of projects available to the user.

**Example output:**

```
Login successful.

You have access to XXX projects, the list has been suppressed. You can list all projects with 'oc projects'

Using project <team>.
```

Press to mark completed

What are the seven categories of commands for the **oc** CLI? Use the **oc --help** feature to obtain this information.

---

Use the --help feature of the oc command.

Enter the following command to view the categories of commands.

```
Command:

    oc --help


Example output:

    OpenShift Client

    This client helps you develop, build, deploy, and run your applications on any OpenShift or Kubernetes
compatible
    platform. It also includes the administrative commands for managing a cluster under the 'adm' subcommand.

    Usage:
      oc [flags]

    Basic Commands:
      types           An introduction to concepts and types
      login           Log in to a server
      new-project     Request a new project
      new-app         Create a new application
      status          Show an overview of the current project
      project         Switch to another project
      projects        Display existing projects
      explain         Documentation of resources
      cluster         Start and stop OpenShift cluster

    Build and Deploy Commands:
      rollout         Manage a Kubernetes deployment or OpenShift deployment config
      rollback        Revert part of an application back to a previous deployment
      new-build       Create a new build configuration
      start-build     Start a new build
      cancel-build    Cancel running, pending, or new builds
      import-image    Imports images from a Docker registry
      tag             Tag existing images into image streams

    Application Management Commands:
      get             Display one or many resources
      describe        Show details of a specific resource or group of resources
      edit            Edit a resource on the server
      set             Commands that help set specific features on objects
      label           Update the labels on a resource
      annotate        Update the annotations on a resource
      expose          Expose a replicated application as a service or route
      delete          Delete one or more resources
      scale           Change the number of pods in a deployment
      autoscale       Autoscale a deployment config, deployment, replication controller, or replica set
```

```
  secrets         Manage secrets
  serviceaccounts Manage service accounts in your project

Troubleshooting and Debugging Commands:
  logs            Print the logs for a resource
  rsh             Start a shell session in a pod
  rsync           Copy files between local filesystem and a pod
  port-forward    Forward one or more local ports to a pod
  debug           Launch a new instance of a pod for debugging
  exec            Execute a command in a container
  proxy           Run a proxy to the Kubernetes API server
  attach          Attach to a running container
  run             Run a particular image on the cluster
  cp              Copy files and directories to and from containers.
  wait            Experimental: Wait for one condition on one or many resources

Advanced Commands:
  adm             Tools for managing a cluster
  create          Create a resource from a file or from stdin.
  replace         Replace a resource by filename or stdin
  apply           Apply a configuration to a resource by filename or stdin
  patch           Update field(s) of a resource using strategic merge patch
  process         Process a template into list of resources
  export          Export resources so they can be used elsewhere
  extract         Extract secrets or config maps to disk
  idle            Idle scalable resources
  observe         Observe changes to resources and react to them (experimental)
  policy          Manage authorization policy
  auth            Inspect authorization
  convert         Convert config files between different API versions
  import          Commands that import applications
  image           Useful commands for managing images
  registry        Commands for working with the registry
  api-versions    Print the supported API versions on the server, in the form of "group/version"
  api-resources   Print the supported API resources on the server

Settings Commands:
  logout          End the current server session
  config          Change configuration files for the client
  whoami          Return information about the current session
  completion      Output shell completion code for the specified shell (bash or zsh)

Other Commands:
  ex              Experimental commands under active development
  help            Help about any command
  plugin          Runs a command-line plugin
  version         Display client and server versions

Use "oc <command> --help" for more information about a given command.
Use "oc options" for a list of global command-line options (applies to all commands).
```

Press to mark completed

What are the node names in the cluster? Use **oc get** to obtain this information. Additionally, use the -o wide parameter. The -o is a small letter O.

---

Get nodes and include the "-o wide" parameter.

---

Enter the following command to view the nodes in the cluster.

```
Command:

    oc get nodes   <and>
    oc get nodes -o wide


Example output:

    From: oc get nodes
    NAME                          STATUS    ROLES          AGE       VERSION
    sydney.52.117.155.20.nip.io   Ready     compute        3d        v1.11.0+d4cacc0
    sydney.52.117.155.26.nip.io   Ready     infra,master   3d        v1.11.0+d4cacc0
    sydney.52.117.155.27.nip.io   Ready     compute        3d        v1.11.0+d4cacc0
    sydney.52.117.155.29.nip.io   Ready     compute        3d        v1.11.0+d4cacc0

    From: oc get nodes -o wide
    NAME                          STATUS    ROLES          AGE       VERSION           INTERNAL-IP
EXTERNAL-IP   OS-IMAGE               KERNEL-VERSION               CONTAINER-RUNTIME
    sydney.52.117.155.20.nip.io   Ready     compute        3d        v1.11.0+d4cacc0   52.117.155.20   <none>
CentOS Linux 7 (Core)   3.10.0-957.27.2.el7.x86_64   docker://1.13.1
    sydney.52.117.155.26.nip.io   Ready     infra,master   3d        v1.11.0+d4cacc0   52.117.155.26   <none>
CentOS Linux 7 (Core)   3.10.0-957.27.2.el7.x86_64   docker://1.13.1
    sydney.52.117.155.27.nip.io   Ready     compute        3d        v1.11.0+d4cacc0   52.117.155.27   <none>
CentOS Linux 7 (Core)   3.10.0-957.27.2.el7.x86_64   docker://1.13.1
    sydney.52.117.155.29.nip.io   Ready     compute        3d        v1.11.0+d4cacc0   52.117.155.29   <none>
CentOS Linux 7 (Core)   3.10.0-957.27.2.el7.x86_64   docker://1.13.1
```

Press to mark completed

What are the pods in the namespace default? Use **oc get** to obtain this information. Be sure to include the **-n default** options to select the default namespace.

---

Be sure to include the **-n default** options to select the default namespace.

---

Get a list of pods in the 'default' namespace.

```
Command:

    oc get pods -n default


Example output:

    NAME                          READY     STATUS    RESTARTS   AGE
    dashboard-7cc4b6645c-bczzf    1/1       Running   0          17h
    docker-registry-1-gxkxh       1/1       Running   0          17h
    registry-console-1-bmw27      1/1       Running   0          17h
    router-1-xx7nq                1/1       Running   0          17h
    yarns-5fcbcfdd9d-bfl5r        1/1       Running   0          17h


Command:

    oc get pods -n default -o wide


Example output:

NAME                          READY     STATUS    RESTARTS   AGE       IP              NODE
NOMINATED NODE
dashboard-7cc4b6645c-bczzf    1/1       Running   0          17h       10.129.2.2      cp4i.149.81.85.109.nip.io
<none>
docker-registry-1-gxkxh       1/1       Running   0          17h       10.128.0.4      cp4i.149.81.85.120.nip.io
<none>
registry-console-1-bmw27      1/1       Running   0          17h       10.128.0.6      cp4i.149.81.85.120.nip.io
<none>
router-1-xx7nq                1/1       Running   0          17h       149.81.85.120   cp4i.149.81.85.120.nip.io
<none>
yarns-5fcbcfdd9d-bfl5r        1/1       Running   0          17h       10.128.4.3      cp4i.149.81.85.111.nip.io
<none>
```

> Press to mark completed

Get events from all namespaces.

---

All namespaces can be viewed by using the --all-namespaces parameter.

---

Get events from all namespaces.

```
Command:

    oc get events --all-namespaces


Example output:  (your detailed output will be differnt)

    NAMESPACE   LAST SEEN   FIRST SEEN   COUNT     NAME                                                    KIND
SUBOBJECT   TYPE      REASON                 SOURCE                                MESSAGE
    default     9m          17h          227       nfs-client-provisioner-7dd7cff46c.15ecdf49063eeb3b
ReplicaSet                          Warning   FailedCreate           replicaset-controller                Error
creating: pods "nfs-client-provisioner-7dd7cff46c-" is forbidden: error looking up service account default/nfs-
client-provisioner: serviceaccount "nfs-client-provisioner" not found
    nfsprov     4m          17h          4446      test-claim.15ecdf49111fc828
PersistentVolumeClaim               Normal    ExternalProvisioning   persistentvolume-controller
waiting for a volume to be created, either by external provisioner "myokd/nfs" or manually created by system
administrator
    default     3m          18h          216       ansible-service-broker.15ecdf40772c1b61
ClusterServiceBroker                Normal    FetchedCatalog         service-catalog-controller-manager
Successfully fetched catalog entries from broker.


    . . . additional output may be shown . . .
```

Press to mark completed

Use the **oc describe** feature to obtain detail information for the pod in your team namespace that begins with -student-ui... ( is your team name).

The output from the describe is formatted for human viewing. The describe command does not support the **-o wide** parameter.

---

You must get the pod name and then describe the pod.

Get the pod name using the **oc get pods** command.

Using the pod name shown from get pods, now use the **oc describe pod** command to obtain the pod details.

---

Describe the team ui pod

```
Command:
```

```
    oc get pods


Example output:

    NAME                                READY     STATUS    RESTARTS   AGE
    team99-student-ui-64c59b7849-lwzp9   1/1       Running   0          17h



Command:

    oc describe pod team99-student-ui-64c59b7849-lwzp9


Example output:

    Name:               team99-student-ui-64c59b7849-lwzp9
    Namespace:          team99
    Priority:           0
    PriorityClassName:  <none>
    Node:               cp4i.149.81.85.108.nip.io/149.81.85.108
    Start Time:         Fri, 24 Jan 2020 17:41:38 +0100
    Labels:             app=team99-student-ui
                        pod-template-hash=2071563405
    Annotations:        openshift.io/scc=restricted
    Status:             Running
    IP:                 10.130.2.10
    Controlled By:      ReplicaSet/team99-student-ui-64c59b7849
    Containers:
      team99-student-ui:
        Container ID:   docker://01e7955f60b6d86cfcf2d3c7c294606f0e7f4672dea9b77e55dd1fee55b05e0d
        Image:          docker.io/ibmicpcoc/collector:latest
        Image ID:       docker-
pullable://docker.io/ibmicpcoc/collector@sha256:71ff985f9e682aece6e2f462aba43c40ed12d00053b74e7630544a730ac32d9f
        Port:           3000/TCP
        Host Port:      0/TCP
        State:          Running
          Started:      Fri, 24 Jan 2020 17:42:03 +0100
        Ready:          True
        Restart Count:  0
        Requests:
          cpu:     100m
          memory:  100Mi
        Environment:
          APP_NAMESPACE:     team99 (v1:metadata.namespace)
          APP_NAME:          team99-student-ui-64c59b7849-lwzp9 (v1:metadata.name)
          COLLECTOR_CONFIG:  <set to the key 'COLLECTOR_CONFIG' of config map 'team99-collector-config'>
Optional: false
        Mounts:
          /var/run/secrets/kubernetes.io/serviceaccount from default-token-rhsbw (ro)
    Conditions:
      Type              Status
```

```
      Initialized       True
      Ready             True
      ContainersReady   True
      PodScheduled      True
    Volumes:
      default-token-rhsbw:
        Type:        Secret (a volume populated by a Secret)
        SecretName:  default-token-rhsbw
        Optional:    false
    QoS Class:       Burstable
    Node-Selectors:  node-role.kubernetes.io/compute=true
    Tolerations:     node.kubernetes.io/memory-pressure:NoSchedule
    Events:          <none>
```

Press to mark completed

View the logs for the pod that starts with “webconsole” in the openshift-web-console namespace. What IP and port are securely serving the console?

---

- Get the list of pods in the openshift-web-console namespace to determine the full pod name to view the logs.
- Review the options for the "Troubleshooting and Debugging Commands" section from **oc --help**.
- Be sure to define the namespace.

---

View the logs for the webconsole pod.

```
Command:

    oc get pods  -n openshift-web-console


Exampel output:

    NAME                          READY     STATUS    RESTARTS   AGE
    webconsole-7fc8759f7b-26v5l   1/1       Running   1          18h



Command:

    oc logs webconsole-7fc8759f7b-26v5l  -n openshift-web-console


Example output:
```

```
    W0913 21:33:08.411930       1 start.go:93] Warning: config.clusterInfo.loggingPublicURL: Invalid value: "":
required to view aggregated container logs in the console, web console start will continue.
    I0913 21:33:08.998336       1 start.go:208] OpenShift Web Console Version: v3.11.0+ea42280
    I0913 21:33:08.998683       1 serve.go:89] Serving securely on 0.0.0.0:8443.  <<<--- view this line
```

> Press to mark completed

Switch to the project for your team. RSH into the pod that starts with **"<team>-student-ui-"**

Replace **<team>** with your team name.

Explore the container and once done "exit" the RSH session.

---

- Change to the project.
- Start a shell session in a pod the pod.
- View the oc --help section labeled "Troubleshooting and Debugging Commands"

---

RSH into the student UI pod in your team project.

```
Command:

    oc project team01

Example output:

    Now using project "team01" on server "https://52.117.155.26:8443".



Command:

    oc get po

Example output:

    NAME                                READY     STATUS    RESTARTS   AGE
    team01-student-ui-7f47864588-hz7gn   1/1       Running   0          1d



Command:
```

```
    oc rsh team01-student-ui-7f47864588-hz7gn
Example output:

    /collector      <<<--- This is a command prompt


Command:

    exit           <<<--- This command is run at the command prompt opened by RSH

Example output:

    Return to a command prompt on the SSH session that the student is using.
```

Press to mark completed

Without using an interactive shell prompt or RSH session, list the files in directory /collector/lib in the pod that starts with **"<team>-student-ui-"**.

Replace **<team>** with your team name.

This will require the use of the **oc exec** command with the following **-it -- ls -la /collector/lib** sub parameters.

---

- The command syntax needs the **-it** and **--** followed by the command to list the files.
- **ls -la /collector/lib** will list the files in the directory
- Example: oc exec <pod name> -it -- ls -la /collector/lib

---

Execute command in the student UI pod.

```
Command:

    oc exec team01-student-ui-7f47864588-hz7gn  -it -- ls -la /collector/lib

    (if not in the <team> project add the -n <team> parameter)


Example output:

    total 132
    drwxrwxrwx    1 appuser  appusers      4096 Sep 13 23:44 .
    drwxrwxrwx    1 root     root          4096 Sep 13 23:44 ..
    -rwxrwxrwx    1 appuser  appusers     20786 Aug 14 20:38 cllr.js
    -rwxrwxrwx    1 appuser  appusers      4671 Aug 14 20:38 config.js
```

```
    -rwxrwxrwx    1 appuser  appusers      8030 Aug 15 15:25 courses.js
    -rwxrwxrwx    1 appuser  appusers     10738 Aug 14 20:38 insight.js
    -rwxrwxrwx    1 appuser  appusers     23135 Sep 12 01:38 parseHtmlBuffer.js
    -rwxrwxrwx    1 appuser  appusers      4405 Aug 14 20:38 printCourse.js
    -rwxrwxrwx    1 appuser  appusers      5499 Aug 17 14:59 student.js
    -rwxrwxrwx    1 appuser  appusers      2879 Aug 14 20:38 utl.js
    -rwxrwxrwx    1 appuser  appusers     28027 Aug 14 20:38 validateBuffer.js
```

Press to mark completed

Within your team project run a pod named **<team>-tools** using the "ibmicpcoc/tools" container image, and never restart the pod. Use the following command to run the pod.

oc run **<team>-tools** --image=ibmicpcoc/tools --restart=Never

Replace **<team>** with your team name.

This running pod is used in the next task.

---

- oc run **<team>**-tools --image=ibmicpcoc/tools --restart=Never

Replace **<team>** with your team name.

---

Run a pod from a single command.

```
Command: (example using team01)

    oc run team01-tools --image=ibmicpcoc/tools --restart=Never


Example output:

    deploymentconfig.apps.openshift.io/team01-tools created
```

Press to mark completed

Using the pod created in the previous task edit the pod (named **<team>-tools**) adding the following label in the metadata.labels section.

work: training

Save the changes and then describe the pod to validate the label is defined.

**NOTE** You must use proper and correct syntax or the editor will not close.

---

The default editor "vi" will open when the command is executed. If you desire to use the “nano” editor add the following parameter before using the oc edit command.

**OC_EDITOR="nano"** oc edit po <team>-tools

---

Edit the pod.

```
EDITING:

Command:

    oc edit po team01-tools

    (OR)

    OC_EDITOR="nano" oc edit po team01-tools



Example output:


    # Please edit the object below. Lines beginning with a '#' will be ignored,
    # and an empty file will abort the edit. If an error occurs while saving this file will be
    # reopened with the relevant failures.
    #
    apiVersion: v1
    kind: Pod
    metadata:
      annotations:
        openshift.io/scc: anyuid
      creationTimestamp: 2019-09-24T00:44:20Z
      labels:                                   <<<--- insert label in this area
        run: team01-tools

    . . . additional output removed . . .




Insert the following into the labels: section

    work: training


Example labels: after adding content
```

```
  labels:
    run: team01-tools
    work: training


USE MUST SAVE the edited pod for the changes to take affect




Command:

    oc describe po team01-tools

Example output:

    Name:               team01-tools
    Namespace:          default
    Priority:           0
    PriorityClassName:  <none>
    Node:               davew.169.45.224.68.nip.io/169.45.224.68
    Start Time:         Mon, 23 Sep 2019 19:44:20 -0500
    Labels:             run=team01-tools
                        work=training
    Annotations:        openshift.io/scc=anyuid
    Status:             Running

    . . . additional output removed . . .
```

Press to mark completed