Setting Up a Jenkins environment in IBM Cloud Private

These exercises set up a Jenkins environment in IBM Cloud Private and verify its operation. The exercises in this module are:

- Exercise 1: Setting up Jenkins
- Exercise 2: Working with Jenkins
- Exercise 3: Building a Sample Web Application

Exercise 1: Setting up Jenkins

Perform the following steps to set up a Jenkins environment in IBM Cloud Private:

1. Set up NFS. The master node is the NFS server. The master node already has NFS server software installed.
    - Open a terminal window.
    - SSH to the master node. Log in as **root** with a password of **passw0rd**.

      `ssh root@master`
    - Create the /jenkins directory.

      `mkdir /jenkins`
    - Edit `/etc/exports` and add the following line:

      `/jenkins *(rw,sync,no_root_squash)`
    - Restart the NFS server.

      `service nfs-server restart`
    - Close the connection to the master node.

      `exit`

```
localuser@ibmcloudacademy: ~
File  Edit  View  Search  Terminal  Help
localuser@ibmcloudacademy:~$ ssh root@master
root@master's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-116-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

125 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Tue Mar 20 10:57:47 2018 from 10.10.1.10
root@master:~# mkdir /jenkins
root@master:~# vi /etc/exports
root@master:~# service nfs-server restart
root@master:~# exit
logout
Connection to master closed.
localuser@ibmcloudacademy:~$
```

2. Prepare the PersistentVolume object.
   - Set up kubectl using the `bx pr cluster-config cloudcluster` command.

```
root@ibmcloudacademy: ~
File  Edit  View  Search  Terminal  Help
root@ibmcloudacademy:~# bx pr cluster-config cloudcluster
Configuring kubectl: /home/localuser/.bluemix/plugins/icp/clusters/cloudcluster/
kube-config
Cluster "cloudcluster" set.
Cluster "cloudcluster" set.
User "cloudcluster-user" set.
Context "cloudcluster-context" modified.
Context "cloudcluster-context" modified.
Switched to context "cloudcluster-context".

OK
Cluster cloudcluster configured successfully.

root@ibmcloudacademy:~#
```

   - Create a file called **jenkinpv.yaml** in your current directory. The file should have the following contents.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: jenkins-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  nfs:
    server: 10.10.1.10
    path: "/jenkins"
```

3. Create the PersistentVolume using the command `kubectl create -f jenkinpv.yaml`. This volume will be used to store the Jenkins configuration.

```
root@ibmcloudacademy: ~
File  Edit  View  Search  Terminal  Help
root@ibmcloudacademy:~# vi jenkinpv.yaml
root@ibmcloudacademy:~# kubectl create -f jenkinpv.yaml
persistentvolume "jenkins-pv" created
root@ibmcloudacademy:~#
```

4. Install Jenkins in IBM Cloud Private. You will be using the Jenkins chart that is available from the kubernetes-charts site. There are many existing charts there that could be used. You can use the command `helm search` to see all the charts available, including the Jenkins one. Run the following command to deploy the Jenkins chart:

```
helm install --tls -n icpjenkins\
  --set Master.AdminPassword=passw0rd \
  --set Master.ServiceType=NodePort \
  --set Master.NodePort=31234 \
  --set Persistence.Size=1Gi \
  stable/jenkins
```

```
localuser@ibmcloudacademy:~$ helm install --tls -n icpjenkins --set Master.AdminPassword=passw0r
d --set Master.ServiceType=NodePort --set Master.NodePort=31234 --set Persistence.Size=1Gi stabl
e/jenkins
NAME:   icpjenkins
LAST DEPLOYED: Fri May 11 10:34:44 2018
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1beta1/Deployment
NAME        DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
icpjenkins  1        1        1           0          0s

==> v1/Pod(related)
NAME                        READY  STATUS    RESTARTS  AGE
icpjenkins-5894b7d6d8-tjgn6  0/1    Init:0/1  0         0s

==> v1/Secret
NAME        TYPE    DATA  AGE
icpjenkins  Opaque  2     0s

==> v1/ConfigMap
NAME              DATA  AGE
icpjenkins        4     0s
icpjenkins-tests  1     0s

==> v1/PersistentVolumeClaim
NAME        STATUS  VOLUME      CAPACITY  E0511 10:34:45.469231    1628 portforward.go:303] erro
r copying from remote stream to local connection: readfrom tcp4 127.0.0.1:33242->10.0.0.2:45266:
 write tcp4 127.0.0.1:33242->10.0.0.2:45266: write: broken pipe
ACCESS MODES  STORAGECLASS  AGE
icpjenkins  Bound   jenkins-pv  1Gi       RWO           0s
```

5. Check that Jenkins is deployed successfully using the command:

   `helm status icpjenkins --tls`

   Make sure that the Status of the Pod is `Running` and that there is a non-zero number of running instances.

   Note: If you are not seeing an available instance immediately, wait a few minutes and check again. It takes a few minutes for the Jenkins image to be downloaded and installed. Also, take note of the names of the 2 services that get created. You will need to configure Jenkins for these service names in a later step.

```
localuser@ibmcloudacademy:~$ helm status icpjenkins --tls
LAST DEPLOYED: Fri May 11 10:04:17 2018
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/PersistentVolumeClaim
NAME        STATUS  VOLUME      CAPACITY  ACCESS MODES  STORAGECLASS  AGE
icpjenkins  Bound   jenkins-pv  1Gi       RWO           4m

==> v1/Service
NAME                TYPE       CLUSTER-IP  EXTERNAL-IP  PORT(S)        AGE
icpjenkins-agent    ClusterIP  10.0.0.89   <none>       50000/TCP      4m
icpjenkins          NodePort   10.0.0.209  <none>       8080:31234/TCP 4m

==> v1beta1/Deployment
NAME        DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
icpjenkins  1        1        1           1          4m

==> v1/Pod(related)
NAME                          READY  STATUS   RESTARTS  AGE
icpjenkins-5894b7d6d8-zzbwr   1/1    Running  0         4m

==> v1/Secret
NAME        TYPE    DATA  AGE
icpjenkins  Opaque  2     4m

==> v1/ConfigMap
NAME              DATA  AGE
icpjenkins        4     4m
icpjenkins-tests  1     4m


NOTES:
1. Get your 'admin' user password by running:
```

Exercise 2: Working with Jenkins

Perform the following steps to start working with your Jenkins environment:

1. Collect configuration information from your Helm release:

    - Get the Jenkins admin password (command is all one line):

    ```
    kubectl get secret icpjenkins -o jsonpath="{.data.jenkins-admin-password}" | base64 --decode
    ```

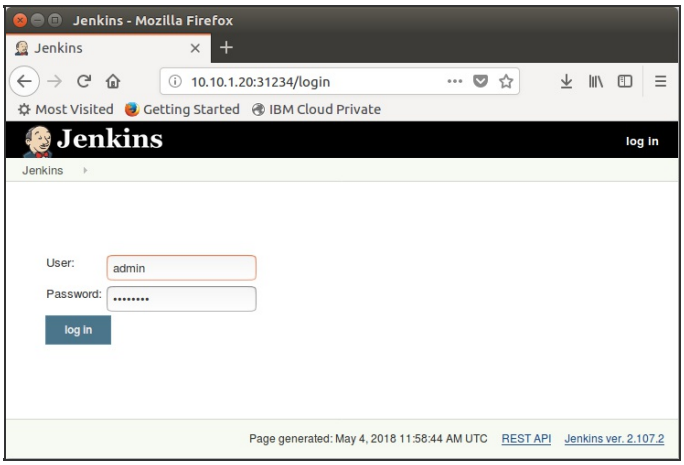
    - Get the Jenkins' port:

    ```
    kubectl get services | grep icpjenkins
    ```

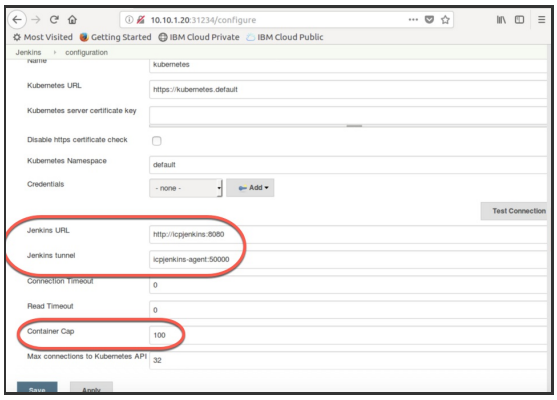
    ```
    localuser@ibmcloudacademy: ~
    File  Edit  View  Search  Terminal  Help
    localuser@ibmcloudacademy:~$ kubectl get secret icpjenkins -o jsonpath="{.data.jenkins-admin-pas
    sword}" | base64 --decode
    passw0rd localuser@ibmcloudacademy:~$
    localuser@ibmcloudacademy:~$ kubectl get services | grep icpjenkins
    icpjenkins          NodePort    10.0.0.209   <none>       8080:31234/TCP   20m
    icpjenkins-agent    ClusterIP   10.0.0.89    <none>       50000/TCP        20m
    localuser@ibmcloudacademy:~$
    ```

    The password should be `passw0rd` and the port should be `31234` as specified in the deployment parameters.

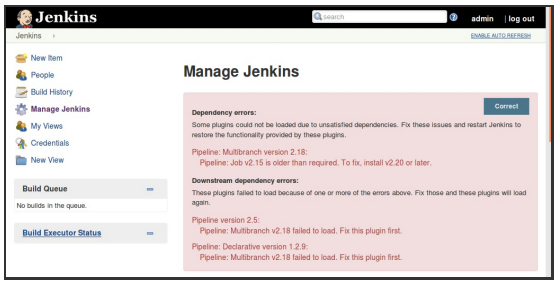
2. Open your browser to http://10.10.1.20:31234 . Log in as `admin` with the password that you retrieved from step 1.



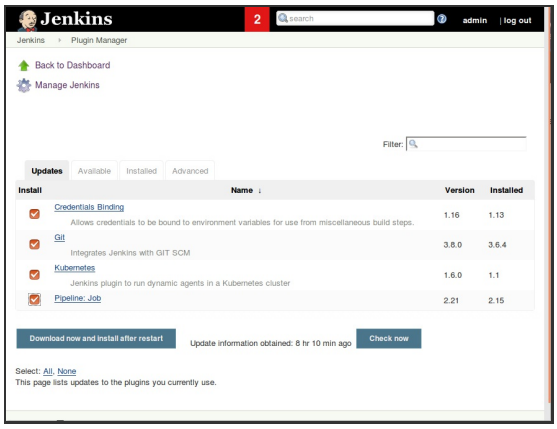
3. Configure the Jenkins server to match your installation and to increase the maximum number of containers.
    - Go to `http://10.10.1.20:31234/configure` and scroll to the **Kubernetes** area
    - Update the values for `Jenkins URL` and `Jenkins tunnel` to match the names of your services.
    - Change the Container Cap to be `100` and click **Save**.



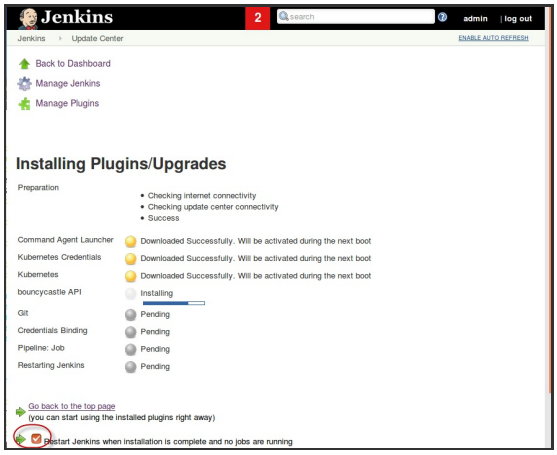
4. Update the Jenkins plugins:
    - Click **Manage Jenkins** on the Jenkins home page.



- Click the **Correct** button on the top right of the page.
- Check all of the plugins and then click **Download now and Install after restart**.



- Check `Restart Jenkins when the installation is complete`.



- Refresh the screen and log back in to Jenkins if necessary

5. Define a secret to access the IBM Cloud Private user:

- Encode the user and password in base64:

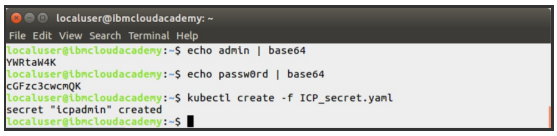
  `echo admin | base64`
  `echo passw0rd | base64`

- Create a ICP_secret.yaml file in your current directory with the following contents:

```
apiVersion: v1
kind: Secret
metadata:
  name: icpadmin
type: Opaque
data:
  username: <user-encoded>
  password: <password-encoded>
```

- Load the secret to ICP

  `kubectl create -f ICP_secret.yaml`



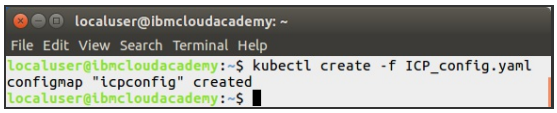
6. Set up IBM Cloud Private registry parameters as a ConfigMap (namespace, imagePullSecret and registry).

   - Create a ICP_config.yaml file in your current directory with the following contents:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: icpconfig
data:
  namespace: default
  registry: cloudcluster.icp:8500
```

   - Load the ConfigMap to ICP

     `kubectl create -f ICP_config.yaml`



7. What is the main difference between a Secret and a ConfigMap in Kubernetes?

   _____

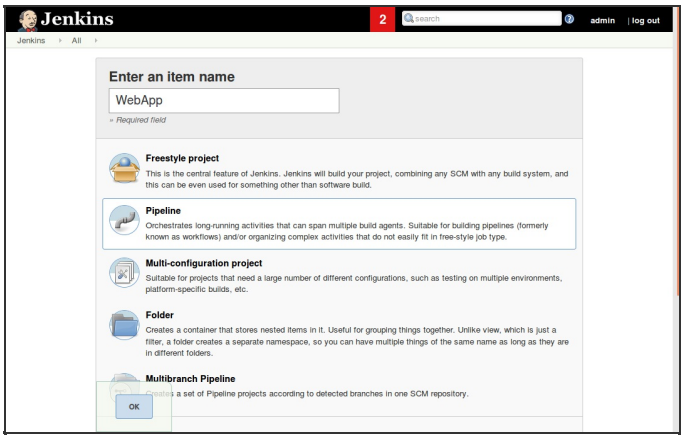
Exercise 3: Building a Sample Web Application

This exercise builds and deploys a Web application using Jenkins to verify that the Jenkins processes are working. The Web application will not be fully operational until you complete the last exercise in this course.

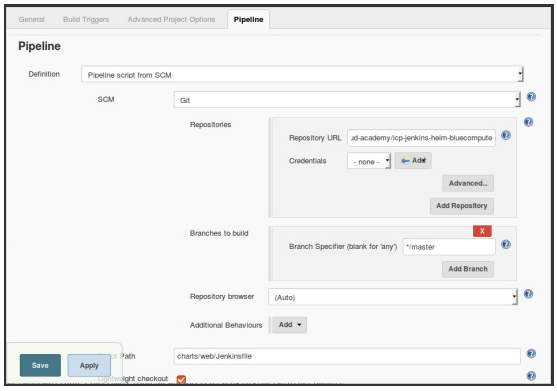
1. Login to Jenkins from the Jenkins Web UI. In your browser, go to `http://10.10.1.20:31234`. Log in as `admin` with a password of `passw0rd`. You should be in the Jenkins dashboard.

2. Click **New item** from the Jenkins menu on the left toolbar.

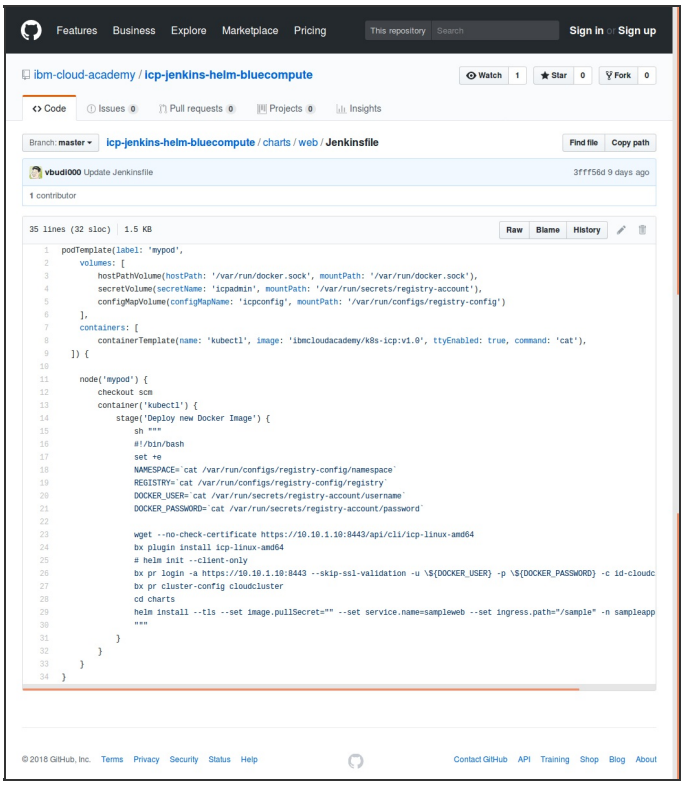
3. Enter a name of `WebApp`, select **Pipeline** and click **OK**.



4. Click the **Pipeline** tab. Scroll down to the Pipeline section and specify the following definitions:
    - Definition: **Pipeline script from SCM**
    - SCM: **Git**
    - Repository URL: **https://github.com/ibm-cloud-academy/icp-jenkins-helm-bluecompute**
    - Branch Specifier: ***/master**
    - Script Path: **charts/web/Jenkinsfile**
    - Click **Save**

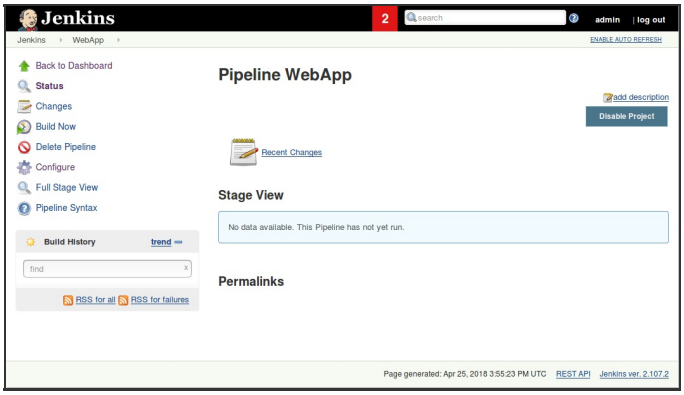


5. Open a new browser tab and navigate to the Jenkinsfile at `https://github.com/ibm-cloud-academy/icp-jenkins-helm-bluecompute`. Click to go to charts/web/Jenkinsfile.

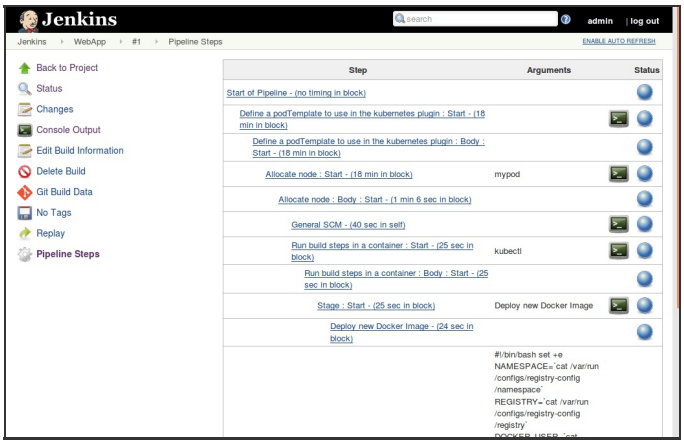


- How many stages are there in the pipeline? _____
- How does the pipeline script access the configMap and secret? _____

6. Back on the Jenkins browser tab, on the WebApp pipeline page, click **Build now**.



7. Once the pipeline is running as indicated on the lower left side, click on the Build number (`#1`), then select **Pipeline Steps**. These steps demonstrate the pipeline invocation hierarchy.

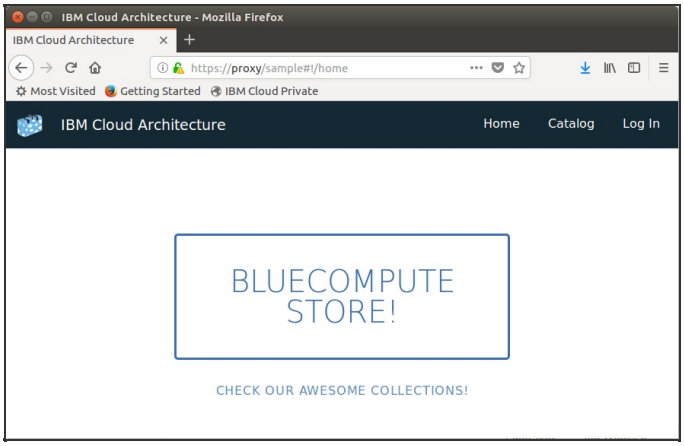


8. Again from the left navigation pane, select **Console Output**. The console should show the helm chart being deployed at the end and the pipeline finished successfully.

9. Having the helm chart deployed does not necessarily mean that the application is correctly deployed. You must check whether the actual application pod is running. This may take a couple of minutes depending on the network speed to load the container. Run `kubectl get pod | grep sampleapp` command or `helm status --tls sampleapp` command and wait until the pod status is `Running`.

```
localuser@ibmcloudacademy:~$ helm status --tls sampleapp
LAST DEPLOYED: Wed Apr 25 11:05:22 2018
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1beta1/Deployment
NAME           DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
sampleapp-web  1        1        1           1          20m

==> v1beta1/Ingress
NAME           HOSTS  ADDRESS     PORTS  AGE
sampleapp-web  *      10.10.1.20  80     20m

==> v1/Pod(related)
NAME                              READY  STATUS   RESTARTS  AGE
sampleapp-web-6dc67d7988-9pjft    1/1    Running  0         20m

==> v1/ConfigMap
NAME                  DATA  AGE
sampleapp-web-config  2     20m

==> v1/Service
NAME           TYPE      CLUSTER-IP  EXTERNAL-IP  PORT(S)       AGE
sampleapp-web  NodePort  10.0.0.51   <none>       80:30626/TCP  20m
```

10. Test the Web application using the URL `http://10.10.1.20/sample`. Note that none of the links in that page is working. This application is part of a larger microservice application. Since the other microservices are not deployed, the links are not active.



11. You can remove this Web application using the following command.

    `helm delete sampleapp --tls --purge`.

*** End of exercises ***