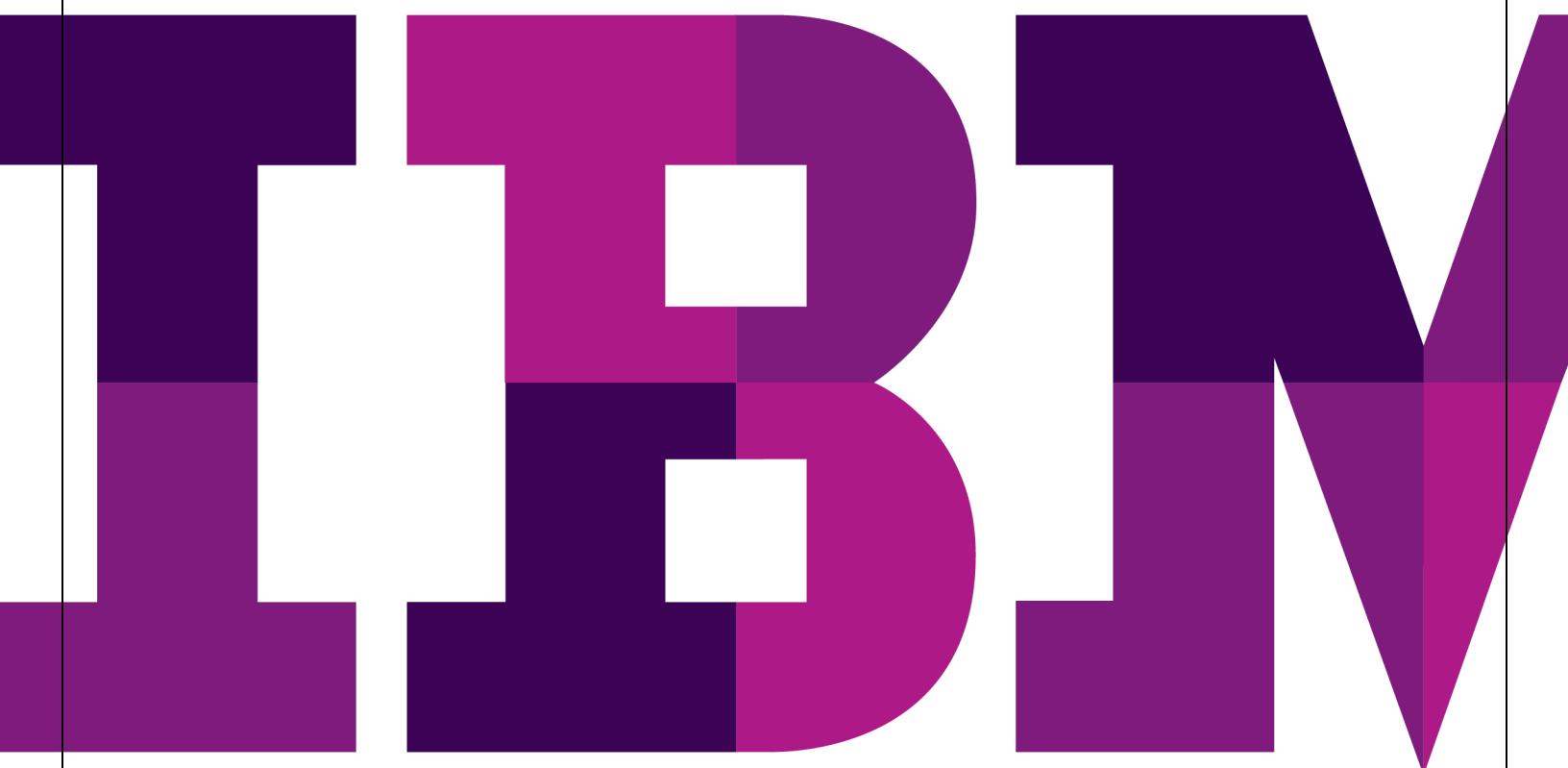


App Modernization using WAS Cloud Pak



Contents

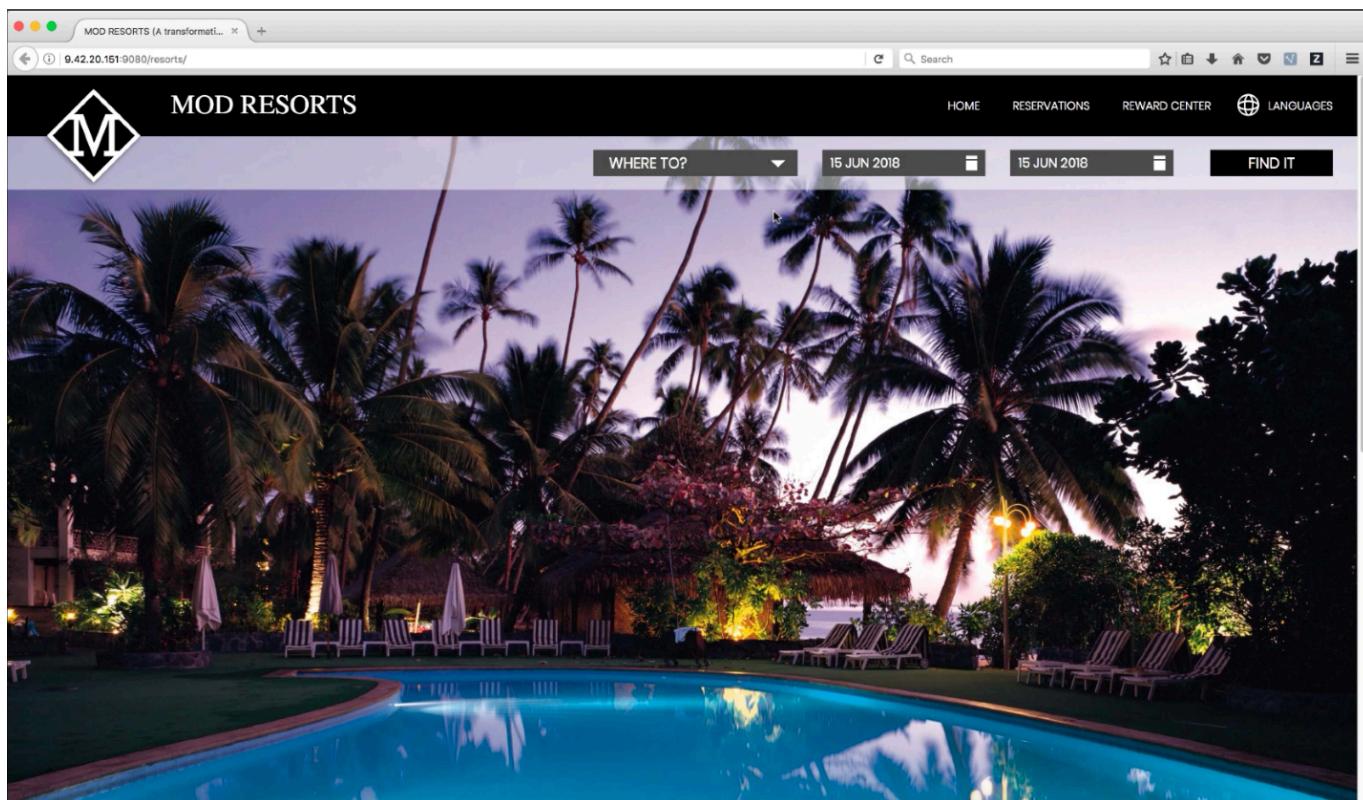
| | | |
|-------------|---|----|
| LAB9. | APP MODERNIZATION USING WAS CLOUD PAK | 3 |
| 1. | BUSINESS SCENARIO | 3 |
| 2. | OBJECTIVE..... | 3 |
| 3. | PREREQUISITES..... | 4 |
| 4. | WHAT IS ALREADY COMPLETED | 4 |
| 5. | LAB TASKS..... | 5 |
| 6. | EXECUTE LAB TASKS | 6 |
| APPENDIX A. | NOTICES | 34 |
| APPENDIX B. | TRADEMARKS AND COPYRIGHTS | 36 |

Lab9. App Modernization using WAS Cloud Pak

In this lab, we will use WebSphere Application Server (WAS) Cloud Pak (WAS Cloud Pak) to create a WAS container in IBM Cloud Private (ICP) and deploy a websphere application to it.

1. Business Scenario

As shown in image below, your company has a web app called **Mod Resorts**, a WebSphere app showing the weather in various locations. Your company wants to move this app from on-premises to the cloud.



As a tech lead, you are planning to use the WAS Base Cloud Pack to move this app to a WAS instance running on ICP. In this lab, you are going to implement this solution. The **Mod Resorts** app is currently deployed on-premises in the local WAS server. You will create a WAS container in ICP using the WAS Base Cloud Pack helm chart and deploy the app to the WAS container.

2. Objective

When you have completed this lab, you will:

- understand how easy it is to move a WebSphere app from an on-premises environment to the ICP without any changes to the app code, while utilizing existing WebSphere administration skills.
- learn the process to create a WAS docker container.
- get familiar with the WAS Base container on ICP deployment process.

3. Prerequisites

The following prerequisites must be completed prior to beginning this lab:

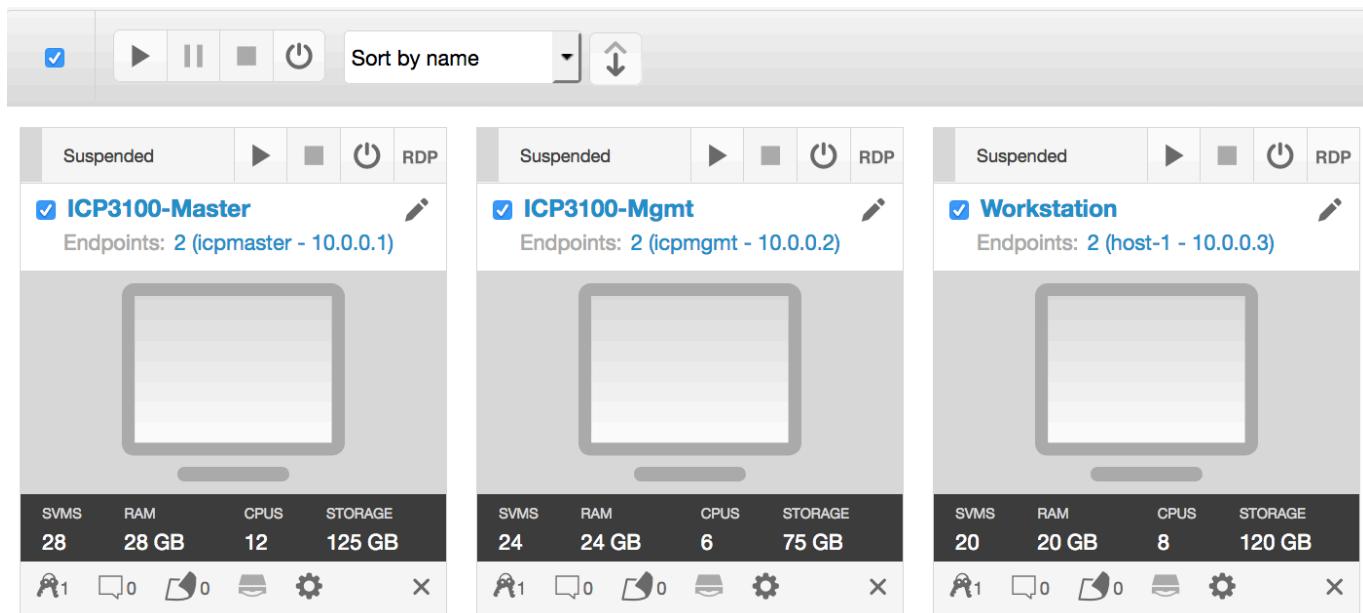
- Familiarity with basic Linux commands

The following symbols appear in this document at places where additional guidance is available.

| Icon | Purpose | Explanation |
|---|------------------|--|
|  | Important! | This symbol calls attention to a particular step or command. For example, it might alert you to type a command carefully because it is case sensitive. |
|  | Information | This symbol indicates information that might not be necessary to complete a step, but is helpful or good to know. |
|  | Trouble-shooting | This symbol indicates that you can fix a specific problem by completing the associated troubleshooting information. |

4. What is Already Completed

Three Ubuntu Linux VMs have been provided for this lab. The ICP cluster is deployed to ICP3100-Master VM with ICP master node, work node and proxy node and ICP3100-Mgmt VM with ICP management node. The Workstation VM is the one you will use to access and work with the ICP cluster.



The login credentials for the Workstation VM are:

User ID: **ibmdemo**

Password: **passw0rd**

Note: Use the Password above in the Workstation VM Ubuntu Terminal for **sudo** in the Lab.

Included components

- IBM WAS Cloud Pak: A solution to move traditional WAS workloads to containerized WAS environment on ICP. Customers will find this solution appealing since their WebSphere apps can be deployed to the cloud environment without making code changes, while utilizing their existing WebSphere administration skills and automation scripts. In this tour, you will experience how seamless and easy it is to move an existing WebSphere app to the cloud using the WAS Base Cloud Pak.

Featured technologies

- IBM Cloud Private: Drive innovation, transform your enterprise, speed of public, control of private.
- Cloud: Accessing computer and information technology resources through the Internet.
- Containers: Virtual software objects that include all the elements that an app needs to run.
- Java: A secure, object-oriented programming language for creating applications.

5. Lab Tasks

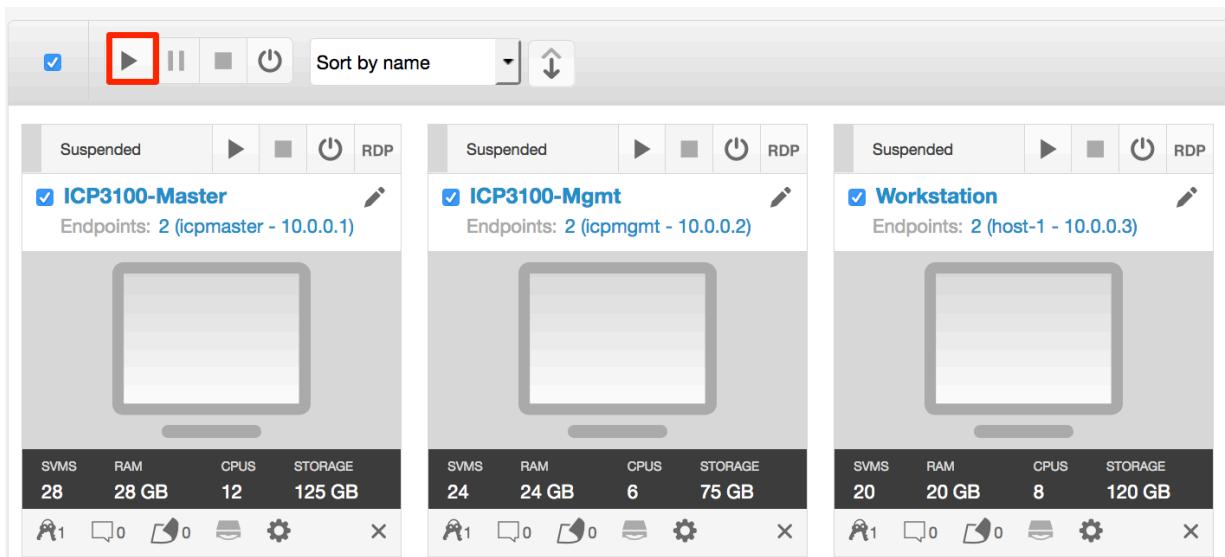
During this lab, you will complete the following tasks:

- review the **Mod Resorts** app on local WAS server.
- build a WAS Base server container image.
- push the WAS Base server container image to ICP image repository.
- deploy the WAS Base server container to ICP using WAS Base Cloud Pak helm chart.
- verify WAS server deployment.
- test and verify the **Mod Resorts** app on WAS container

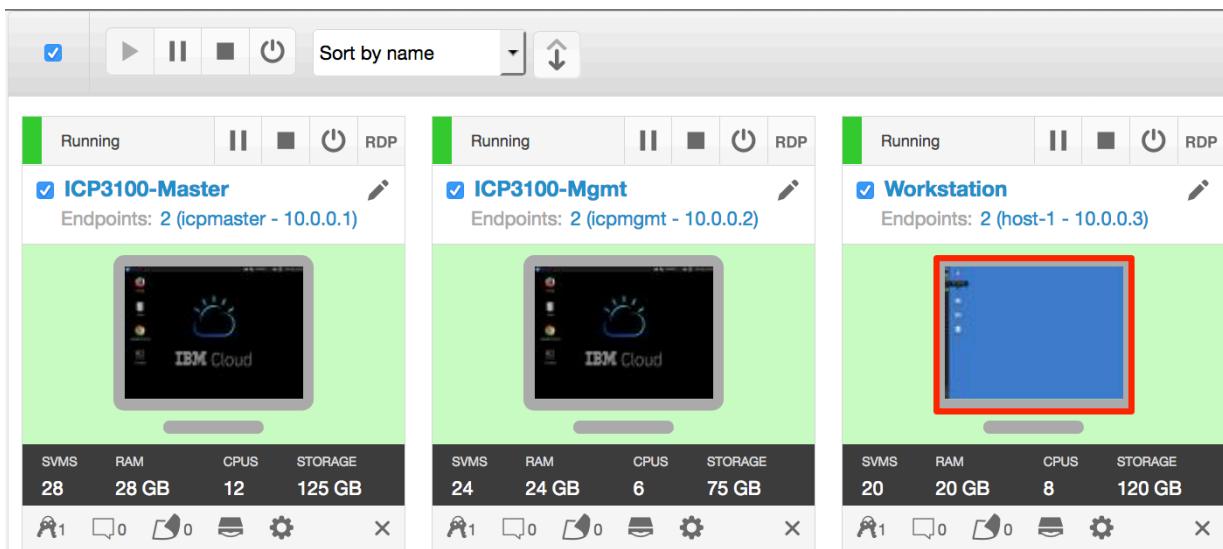
6. Execute Lab Tasks

6.1 Log in to the workstation VM and get started

1. Launch the three lab VMs by clicking the **start** button.



2. After the VMs are start, click the Workstation VM icon to access it.



The Workstation Ubuntu Desktop will be displayed.

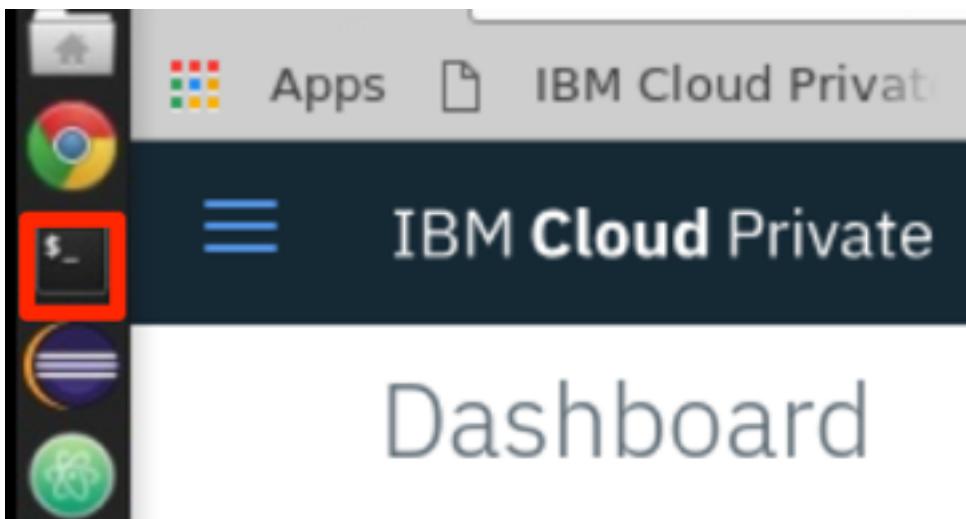
6.2 Review the on-prem WebSphere app

In this task, you will take a look at **Mod Resorts** app deployed to the local WebSphere Application Server (WAS) environment. You are going to move this app to the cloud using WAS Cloud Pak later.

1. Start WebSphere Application Server

In the **workstation** VM, we have a local WebSphere Application Server V8.5.5 which hosts the **Mod Resorts** app. To start the WAS server:

- _a. Open a terminal window by clicking its icon on the lab VM desktop tool bar.



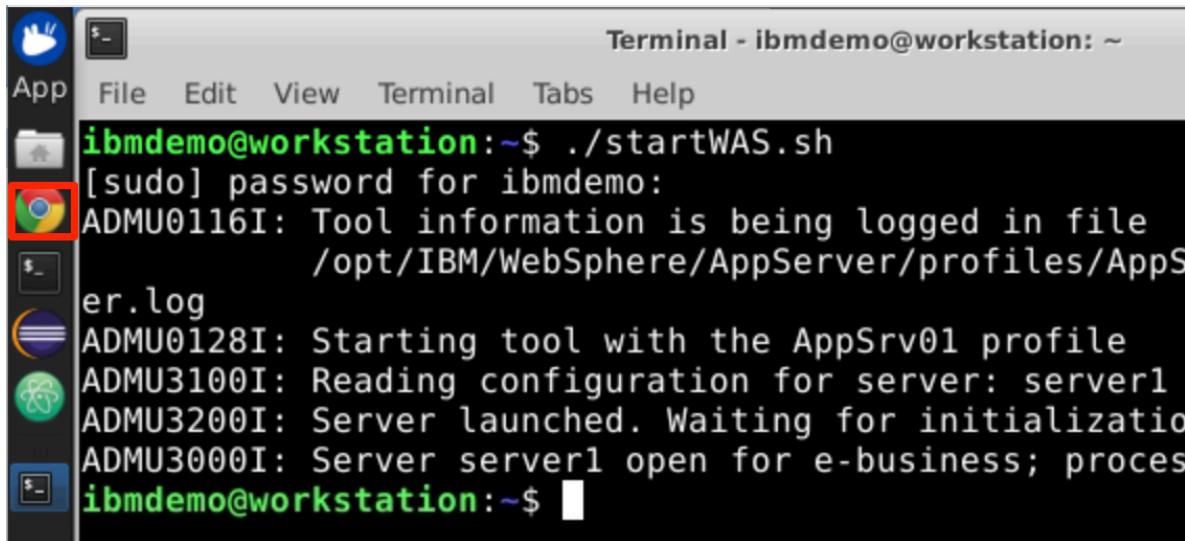
_b. In the terminal window, issue the commands below to start the WAS server.

./startWAS.sh

when prompted, enter the **sudo** user password as: **passw0rd**.

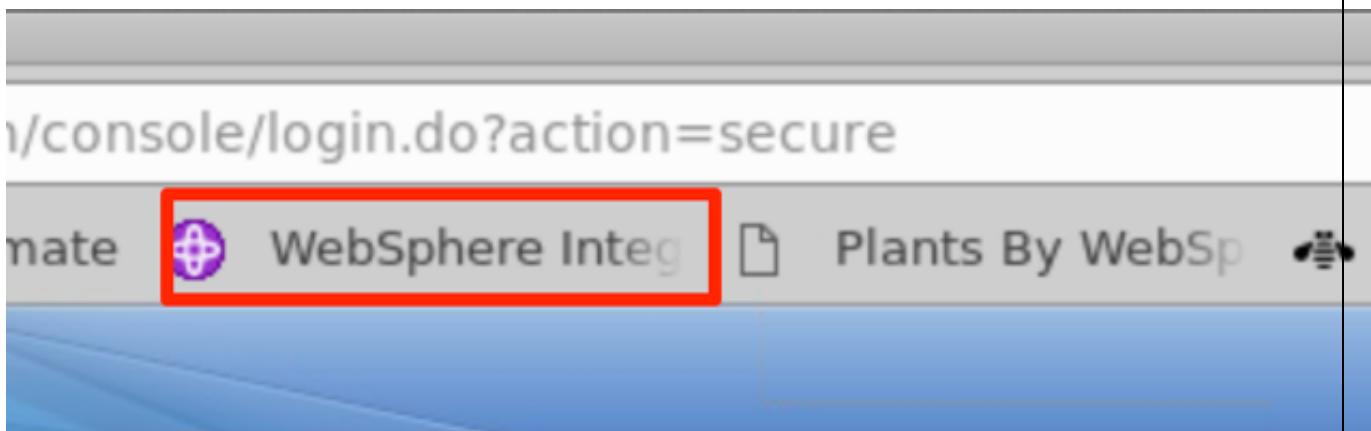
The WAS server container will get started. Within a few minutes, WAS server will be ready.

_c. Click the web browser icon desktop tool bar to open a browser window.

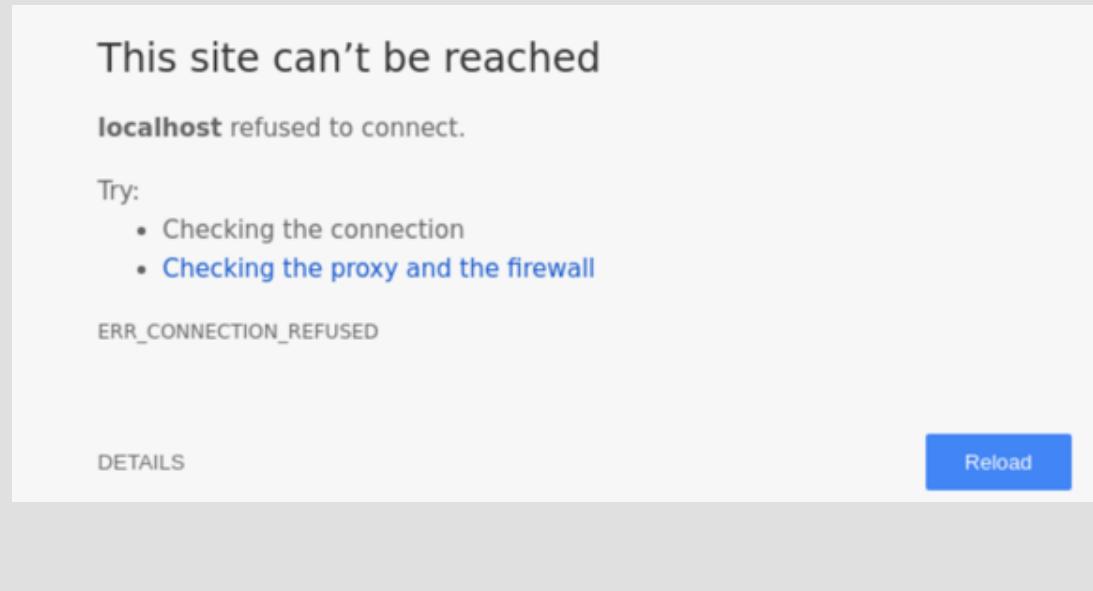


```
Terminal - ibmdemo@workstation: ~
App File Edit View Terminal Tabs Help
ibmdemo@workstation:~$ ./startWAS.sh
[sudo] password for ibmdemo:
ADMU0116I: Tool information is being logged in file
            /opt/IBM/WebSphere/AppServer/profiles/AppS
er.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initializatio
ADMU3000I: Server server1 open for e-business; proces
ibmdemo@workstation:~$
```

_d. Go back to web browser window and click **WebSphere Integrated Solution Console** bookmark to launch the WAS console.



Note: If you see the following message, that means the WAS server is not started yet, just wait a few minutes and try again, you should be able to access the WAS Console page.



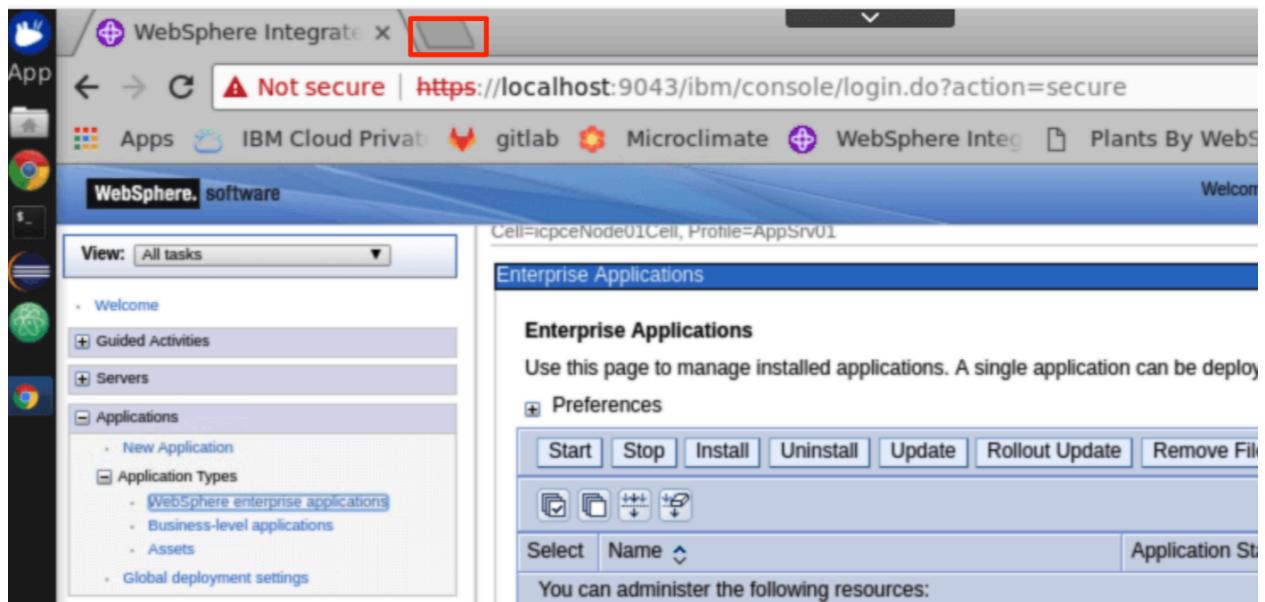
- _e. If you see “Your connection is not private” message, click **ADVANCED -> Proceed to localhost** to continue.
- _f. In the WAS Console login page, enter the User ID and Password as: **wsadmin/passw0rd** and click **Login**.
- _g. On the WAS Console page, click **Applications -> Application Types -> WebSphere enterprise applications** to view the apps deployed.

In the **Enterprise Applications** list, you can see all websphere apps deployed. Currently the **Mod Resorts** app is running.

| Select | Name | Application Status |
|--------------------------|--------------------|--------------------|
| <input type="checkbox"/> | DefaultApplication | |
| <input type="checkbox"/> | lvApp | |
| <input type="checkbox"/> | modresorts-1_0.war | |
| <input type="checkbox"/> | query | |

2. View **Mod Resorts** app

- a. From the web browser window, click new Tab to open a new browser window with the **Modresorts** app URL: <http://localhost:9080/resorts/>.



WebSphere Integrated Solutions Console

Not secure | https://localhost:9043/ibm/console/login.do?action=secure

Cell=icpceNode01Cell, Profile=AppSrv01

Enterprise Applications

Enterprise Applications

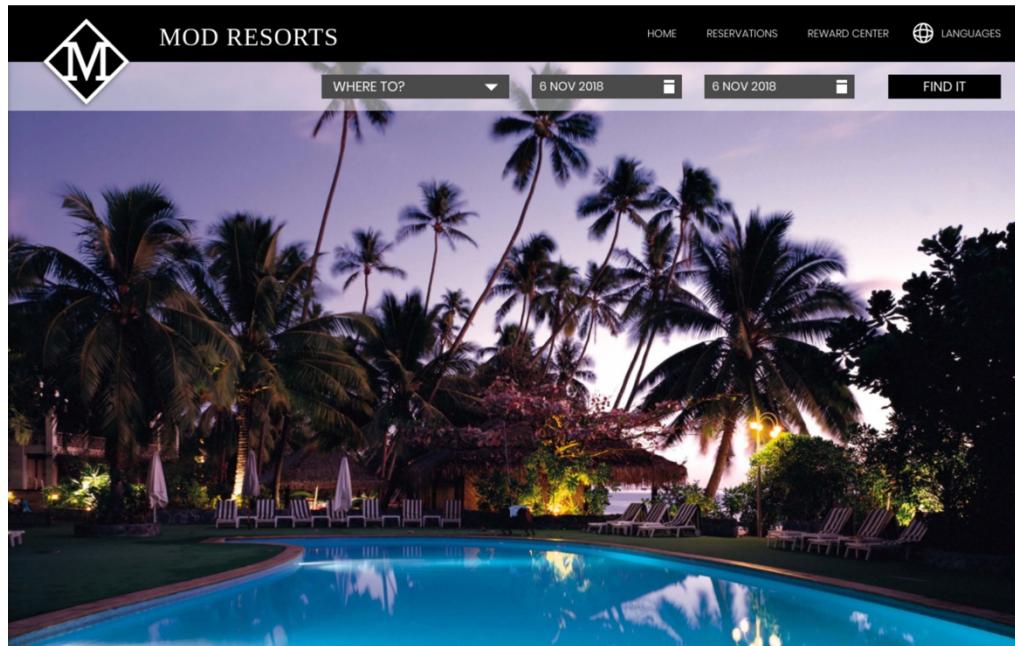
Use this page to manage installed applications. A single application can be deployed to multiple servers.

Start Stop Install Uninstall Update Rollout Update Remove File

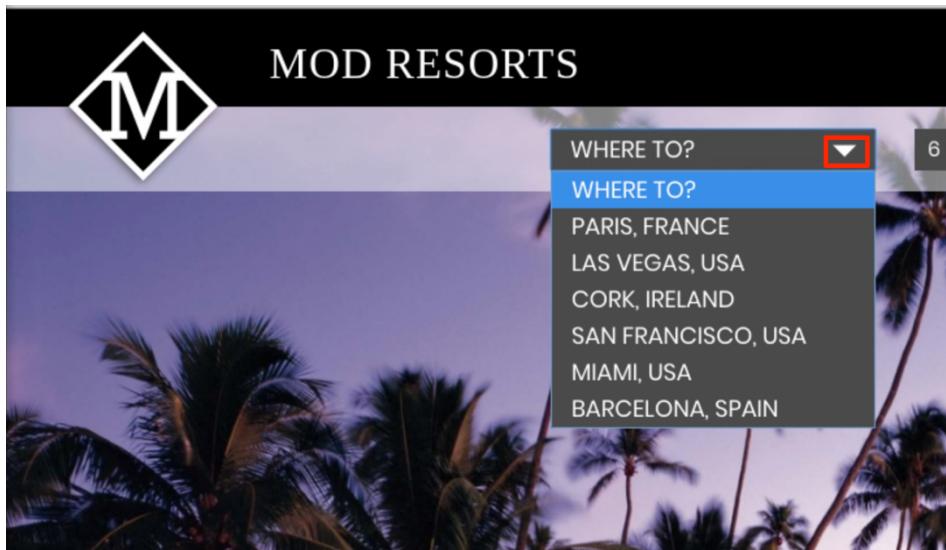
Select Name Application Status

You can administer the following resources:

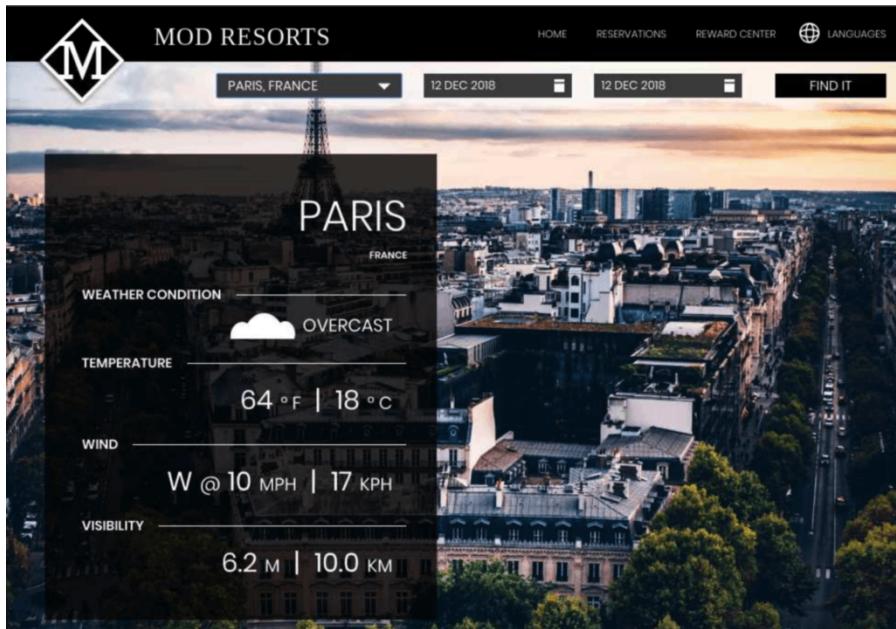
The **Mod Resorts** app home page is shown.



b. Click **WHERE TO?** dropdown menu to see the city list.



- ___c. Click **PARIS, FRANCE** from the list, it will show the weather of the city.



You have reviewed the app, next we will shift our discussion to how the **WAS Cloud Pak** will help to move this app to the cloud.

6.3 Build a WAS Base Server Container Image.

In this task, you are going to build a WAS Base server Docker container image with the **Mod Resorts** app deployed.

Note: according to Docker's best practices you should create a new WAS Base image which adds a single application and the corresponding configuration. You should avoid configuring the image manually

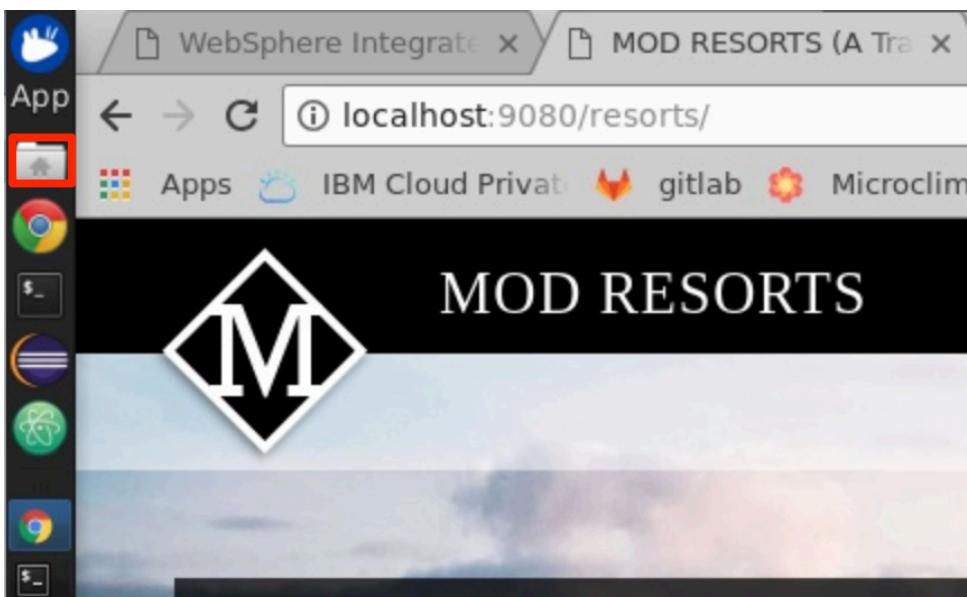
(after it started) via Admin Console or wsadmin unless it is for debugging purposes, because such changes won't be present if you spawn a new container from the image.

There are four key files you needed to build your WAS Base Server container image:

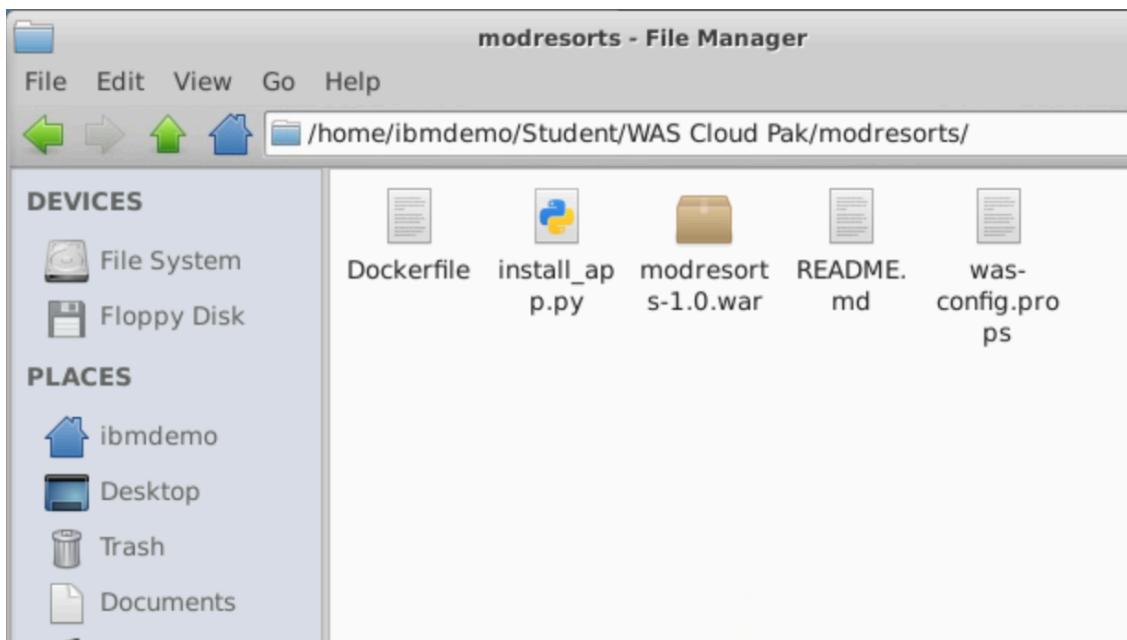
- Dockerfile – defines how the Docker image that has your app and configuration pre-loaded will be built.
- App runtime – the ear or war file of your app.
- was-config.props – defines additional WAS server configurations.
- Install_app.py – your app deployment and advanced configuration Jython script file.

__1. Review Dockerfile, was-config.props file and install_app.py file.

__a. From the Desktop tool bar, click the **File Manager** icon to open it.



__b. Navigate to **/home/ibmdemo/Student/WAS Cloud Pak/modresorts** directory.



__c. Double click the **Dockerfile** to open it in **Text editor** for reviewing.

```

FROM ibmcom/websphere-traditional
COPY modresorts-1.0.war /work/config/modresorts-1.0.war
COPY install_app.py /work/config/install_app.py
COPY was-config.props /work/config/was-config.props
RUN /work/configure.sh

```

As you can see, we have these three files opened for reviewing. The first is the **Dockerfile** file which defines the following activities to create the WAS Base container image:

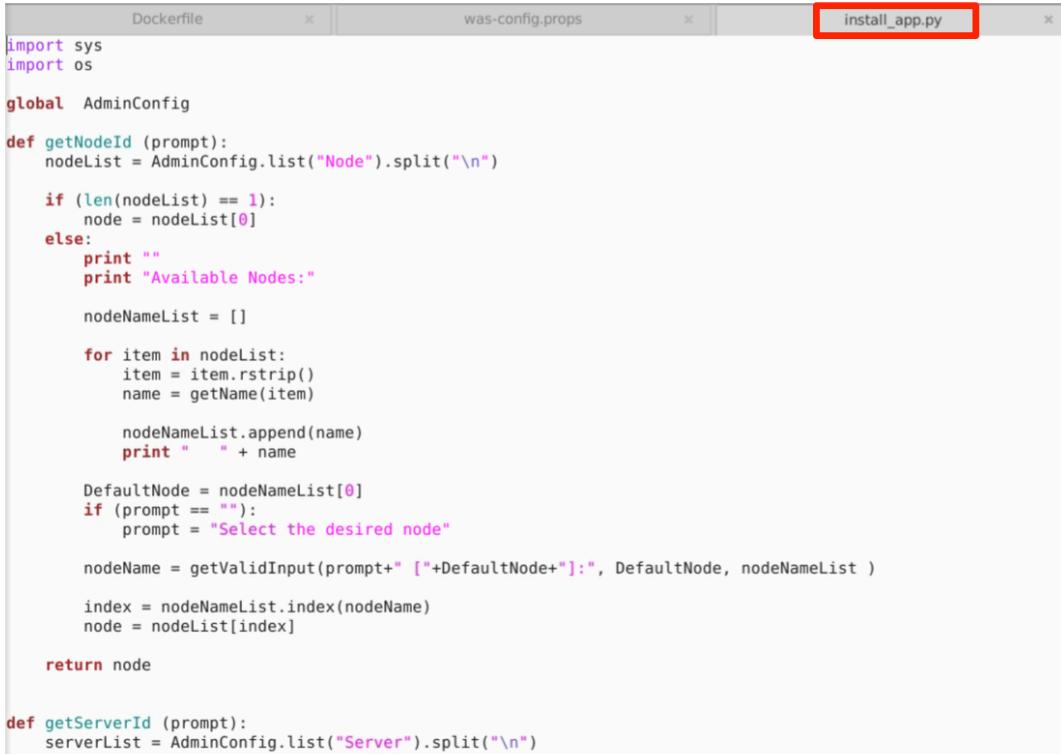
- Get the base WAS image from Docker Hub
- Add the app runtime file to the base image
- Add the app deployment script to the base image
- Add the WAS configure script to the base image
- Run the configuration script to config the WAS server instance inside the container

- ___d. Go back to File Manager and double click **was-config.props** file to review it. This is the script file you use to configure your WAS server instance. In this lab, we have specified to increases the WAS server instance thread pool to 100.



```
ResourceType=ThreadPool
ImplementingResourceType=Server
ResourceId=Cell={!{cellName}:Node={!{nodeName}:Server={!{serverName}:ThreadPoolManager=:ThreadPool=
maximumSize=100
name=WebContainer
minimumSize=100
inactivityTimeout=60000
```

- ___e. In the File Manager window, double click **install_app.py** file to review its contents. This is a standard Jython script for deploying the **Mod Resorts** app to WAS server container.



```
import sys
import os

global AdminConfig

def getNodeID (prompt):
    nodeList = AdminConfig.list("Node").split("\n")

    if (len(nodeList) == 1):
        node = nodeList[0]
    else:
        print ""
        print "Available Nodes:"

        nodeNameList = []

        for item in nodeList:
            item = item.rstrip()
            name = getName(item)

            nodeNameList.append(name)
            print " " + name

        DefaultNode = nodeNameList[0]
        if (prompt == ""):
            prompt = "Select the desired node"

        nodeName = getValidInput(prompt+ " ["+DefaultNode+"]:", DefaultNode, nodeNameList )

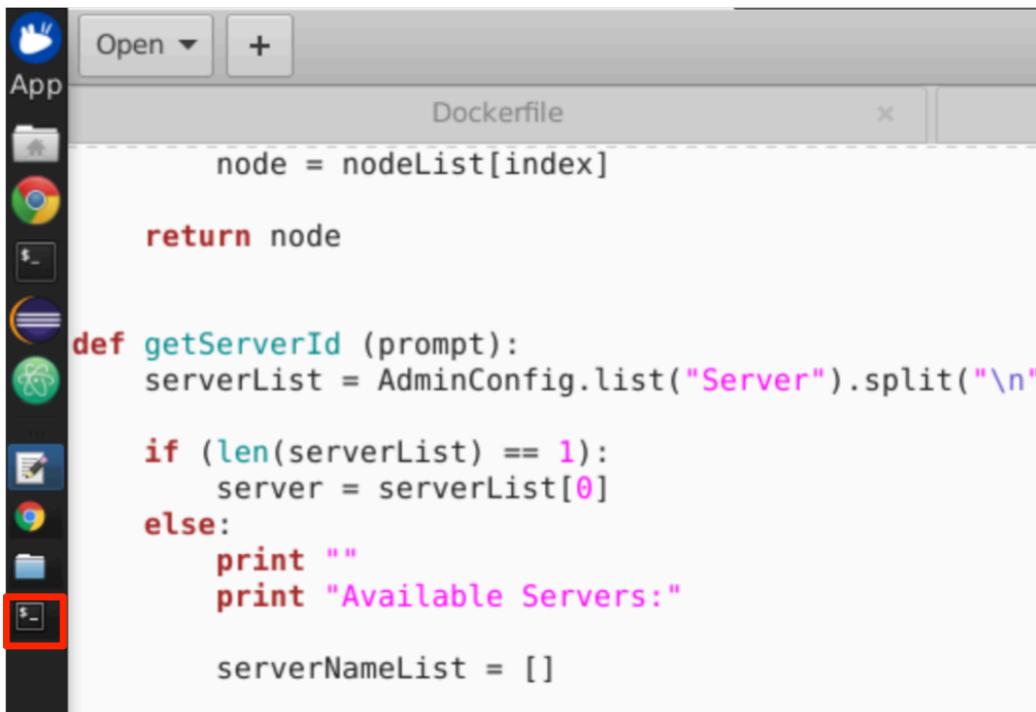
        index = nodeNameList.index(nodeName)
        node = nodeList[index]

    return node

def getServerId (prompt):
    serverList = AdminConfig.list("Server").split("\n")
```

___2. Build the WAS Base server container image

- ___a. Go back to the terminal window by clicking its icon on the Desktop too bar.



```

App Open + Dockerfile x
node = nodeList[index]

return node

def getServerId (prompt):
    serverList = AdminConfig.list("Server").split("\n")

    if (len(serverList) == 1):
        server = serverList[0]
    else:
        print ""
        print "Available Servers:"

    serverNameList = []

```

- ___b. Navigate to the **/home/ibmdemo/Student/WAS Cloud Pak/modresorts** directory with command

```
cd /home/ibmdemo/Student/WAS Cloud Pak/modresorts
```

- ___c. Execute the following command to build the docker container image with the Dockerfile you just reviewed:

```
docker build . -t mycluster.icp:8500/lab/wasv9:latest
```

This will create a docker image called

mycluster.icp:8500/lab/wasv9:latest where **mycluster.icp:8500** is the ICP hostname and port and **lab** is the Kubernetes namespace.

- ___d. After the docker container image is created, you can issue the command below to check it:

```
docker images |grep wasv9
```

You will see your docker image.

6.4 Push WAS Container Image to ICP

After the WAS container image is built, you need to push it to the ICP image registry, so you can use it to deploy the WAS Base server container to ICP.

- ___1. From the terminal window, issue the command below to login to ICP.

```
docker login mycluster.icp:8500
```

when prompted, enter the username and password as: **admin/admin**.

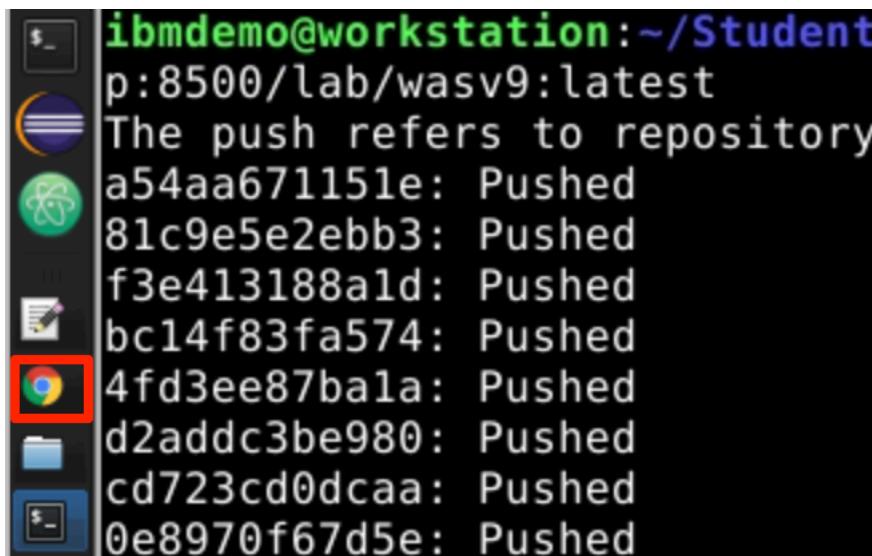
- __2. Execute the following command to push your docker image to ICP image repository.

docker push mycluster.icp:8500/lab/wasv9:latest

When it is done, your docker image **mycluster.icp:8500/lab/wasv9:latest** is pushed to ICP.

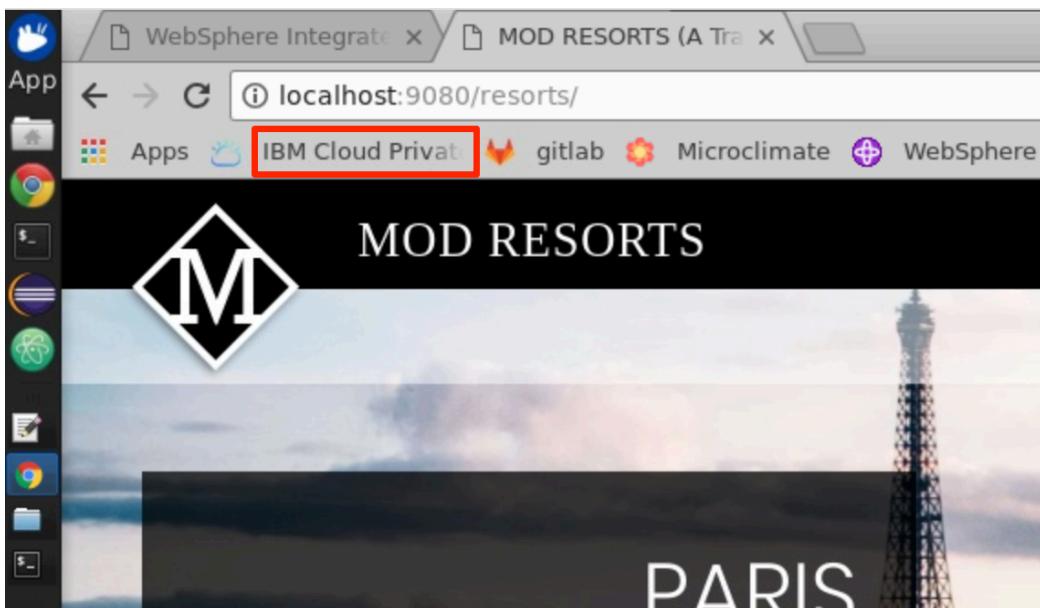
- __3. Verify the pushed docker image in ICP.

- __a. Click the web browser icon on the desktop tool bar to go back to the browser window.

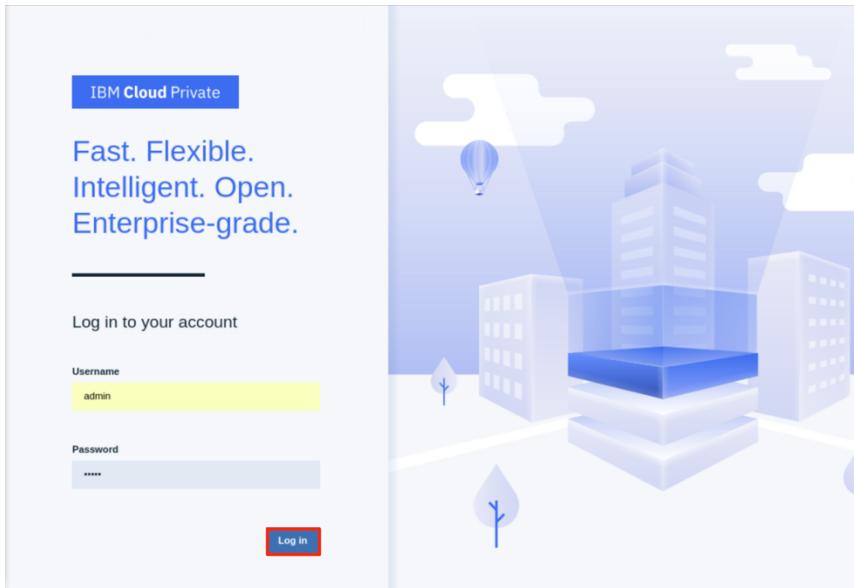


```
ibmdemo@workstation:~/Student
p:8500/lab/wasv9:latest
The push refers to repository
a54aa671151e: Pushed
81c9e5e2ebb3: Pushed
f3e413188a1d: Pushed
bc14f83fa574: Pushed
4fd3ee87bala: Pushed
d2addc3be980: Pushed
cd723cd0dcaa: Pushed
0e8970f67d5e: Pushed
```

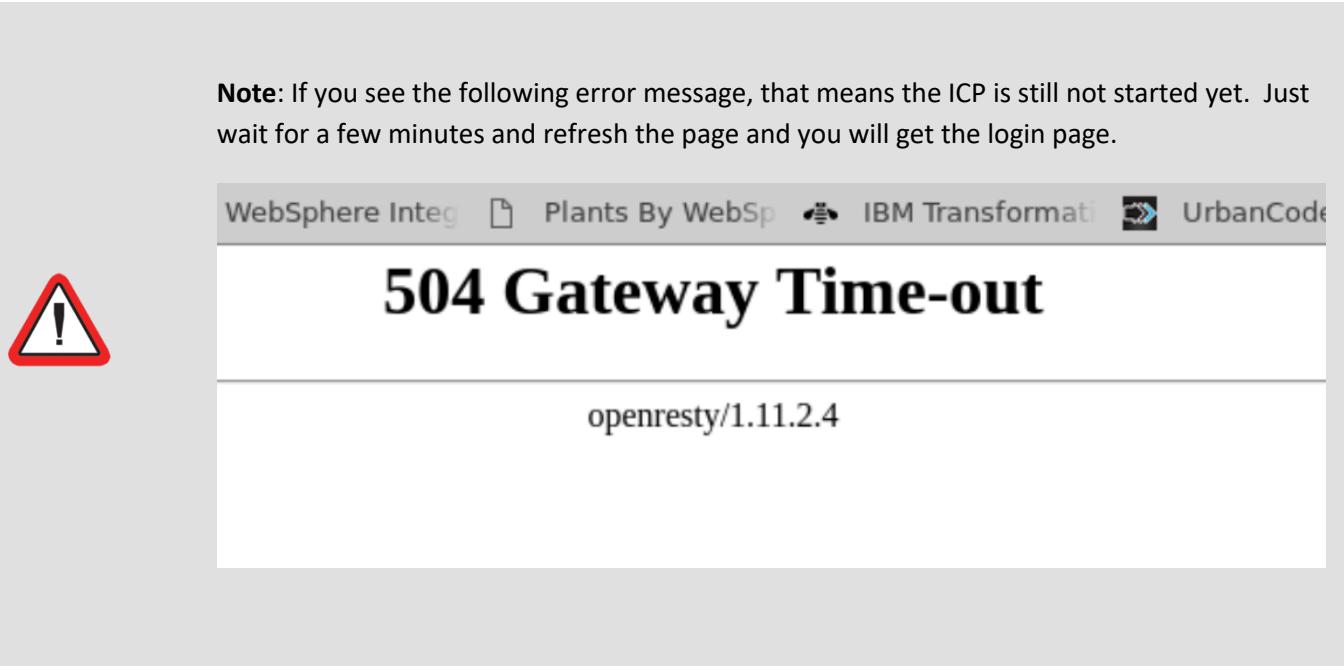
- __b. Click **IBM Cloud Private** bookmark to open the ICP console.



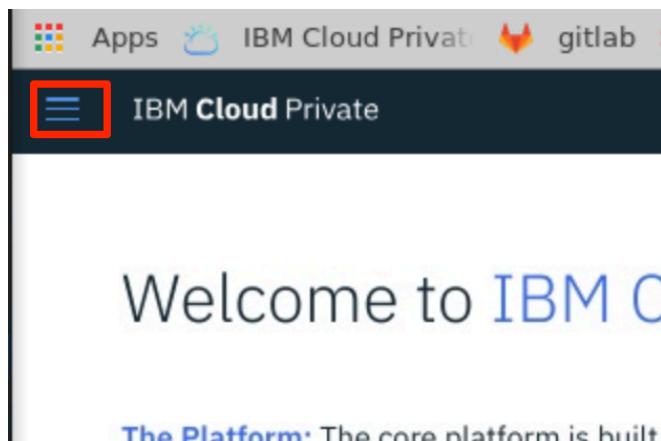
- __c. Click **Log in** to login to ICP console with ICP username/password: **admin/admin**.



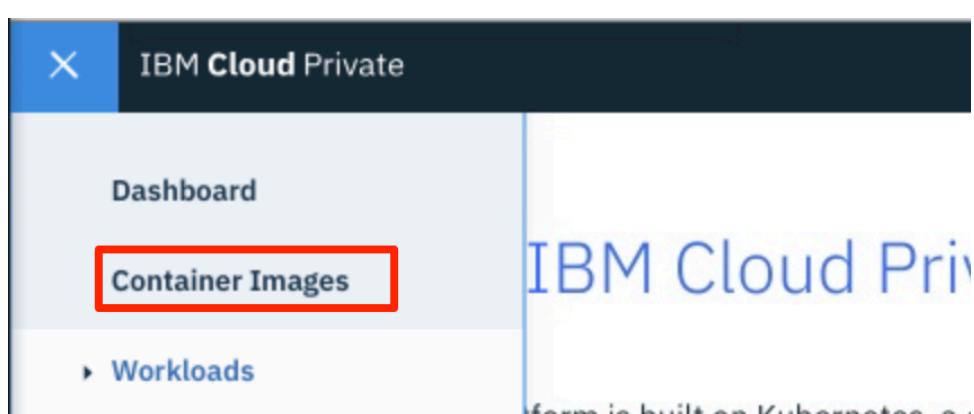
Note: Since we do not set SSL cert in Chrome web browser in this lab, if you see **“Your connection is not private”** message, click **Advanced->Proceed to 10.0.01.** to continue.



___d. Click ICP **Menu** icon on the top left corner.



___e. Click **Container Images** to access the image list.



___f. You will see the image you just pushed is in the list.

| Container Images | | | |
|--|-----------------------|--------|-----------|
| ibmcom/secure-config-annotator-amd64 | ● Passed | ibmcom | namespace |
| ibmcom/service-catalog-service-catalog-amd64 | ● Passed | ibmcom | namespace |
| ibmcom/statsd-amd64 | ● Passed | ibmcom | namespace |
| ibmcom/tiller-amd64 | ● Passed | ibmcom | namespace |
| ibmcom/ucaro-amd64 | ● Passed | ibmcom | namespace |
| ibmcom/unified-router-amd64 | ● Passed | ibmcom | namespace |
| ibmcom/usncrawler-amd64 | ● Passed | ibmcom | namespace |
| ibmcom/vulnerability-annotator-amd64 | ● Passed | ibmcom | namespace |
| lab/catalog | ● Passed | lab | namespace |
| lab/db2_developer_c | ● Passed | lab | global |
| lab/ibmcom/ibm-http-server-v1 | ● Passed | lab | namespace |
| lab/orders | ● Passed | lab | namespace |
| lab/ui | ● Passed | lab | namespace |
| lab/wasy9 | ● Passed | lab | namespace |

g. Click the image link to view its detail info.

| Type | Detail |
|-------|-----------|
| Name | lab/wasv9 |
| Owner | lab |
| Scope | namespace |
| Tags | latest |

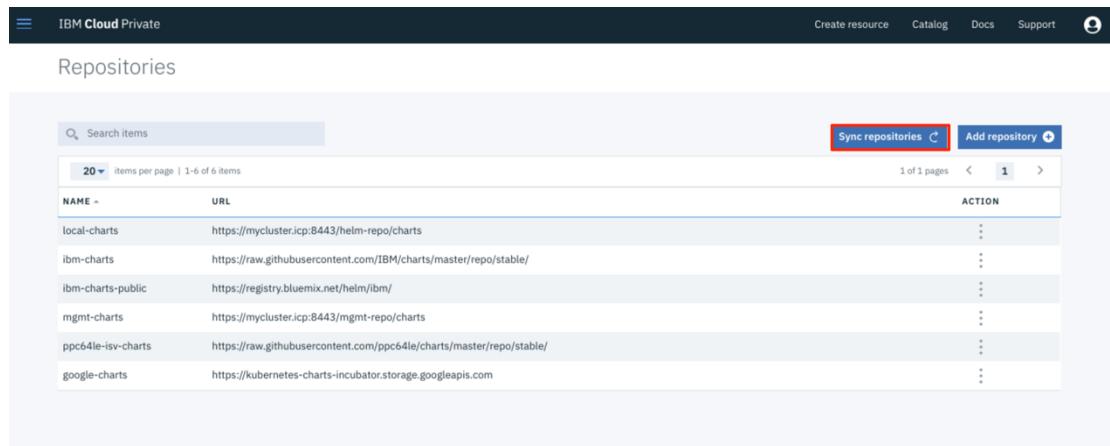
6.5 Deploy WAS Container to ICP using WAS Cloud Pak Helm Chart

In this task, you are going to create a WAS container in ICP using WAS Cloud Pak helm chart and the WAS container image you created. The helm chart in WAS Cloud Pak provides an easy way to quickly provision a WAS container in a Kubernetes cluster environment.

1. From the ICP console menu, select **Manage->Helm Repositories**.

The screenshot shows the IBM Cloud Private interface. The left sidebar has a 'Manage' section with a 'Helm Repositories' link, which is highlighted with a red box. The main content area displays a brief description of the platform and its management features, with a large blue and white graphic of a cloud and a bee.

2. Click **Sync repositories** to synchronize the helm chart repositories.



IBM Cloud Private

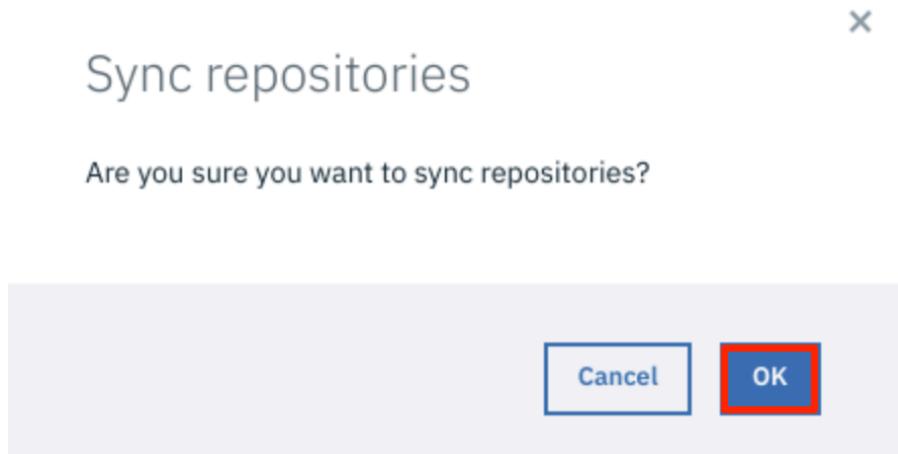
Repositories

Search items

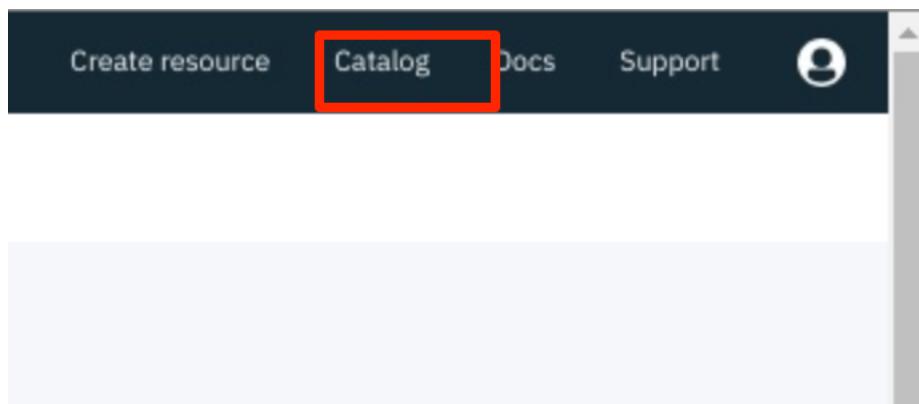
20 items per page | 1-6 of 6 items

| NAME | URL | ACTION |
|--------------------|--|--------|
| local-charts | https://mycluster.icp:8443/helm-repo/charts | ⋮ |
| ibm-charts | https://raw.githubusercontent.com/IBM/charts/master/repo/stable/ | ⋮ |
| ibm-charts-public | https://registry.blueix.net/helm/ibm/ | ⋮ |
| mgmt-charts | https://mycluster.icp:8443/mgmt-repo/charts | ⋮ |
| ppc64le-isv-charts | https://raw.githubusercontent.com/ppc64le/charts/master/repo/stable/ | ⋮ |
| google-charts | https://kubernetes-charts-incubator.storage.googleapis.com | ⋮ |

3. Click OK to continue.



4. After the helm chart repositories are synchronized, click **Catalog** to access ICP catalog.



Create resource Catalog Docs Support

The Catalog page lists all product and service helm charts available in the ICP environment,

IBM Cloud Private

Catalog

All Categories >

Search Items

Filter

Blockchain

Business Automation

Data

Data Science & Analytics

DevOps

Integration

IoT

Network

Operations

Runtimes & Frameworks

Security

Storage

Tools

Other

ibm-ace-dev

ibm-aspera-cll

ibm-blockchain-platform-remote-peer

ibm-cam

ibm-cem

ibm-csi-nfs

ibm-datapower-dev

ibm-db2otp-dev

ibm-dsm-dev

ibm-dsx-dev

ibm-eventstreams-dev

ibm-voice-gateway-dev

ibm-was-vm-quickstarter-dev

ibm-websphere-liberty

ibm-websphere-traditional

5. Enter **was** in the Search Filter field and all **was** related helm charts will show up. Click **ibm-websphere-traditional** helm chart to open it.

Search Items

Filter

was

ibm-voice-gateway-dev

ibm-was-vm-quickstarter-dev

ibm-websphere-liberty

ibm-websphere-traditional

The Overview page of the WAS helm chart is shown.

6. The **Overview** page lists all system prerequisite and installation information you needed to install the chart. Read through the info before you deploy the chart.

- The **Chart Details** section shows all service components to be deployed.
- The **Prerequisites** section highlights the requirements needed to deploy this chart. The section also presents the configuration scripts which can be used to create or clean up the required resources.
- The **Installing Chart** section describes the chart deployment steps.
- The **Configuration** section lists all data required for the chart installation, including their description and default values.

7. Next click **Configure** to go to the chart Configuration page.

Configuration

The following tables lists the configurable parameters of the IBM WebSphere Application Server traditional chart and their default values.

| Parameter | Description | Default |
|------------------|--|------------------------------|
| replicaCount | The number of desired replica pods that run simultaneously | 1 |
| image.repository | Docker image repository | ibmcom/websphere-traditional |
| image.pullPolicy | Docker image pull policy. Defaults to Always when the latest tag is specified. | IfNotPresent |
| image.tag | Docker image tag | 9.0.0.9-profile |
| service.type | Kubernetes service type exposing ports | NodePort |
| service.name | Kubernetes service name for HTTP | https-was |
| service.port | The abstracted service port for | 9443 |

Configure

8. In the **Configuration** page, Click **Helm release name** field to enter a unique release name like: **was01**.

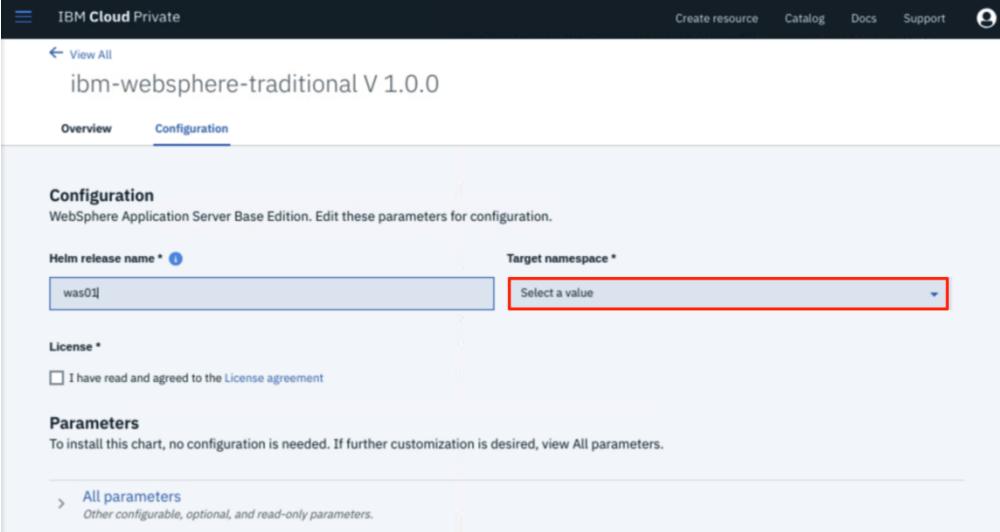
ibm-websphere-traditional V 1.0.0

Overview Configuration

Configuration
WebSphere Application Server Base Edition. Edit these parameters for configuration.

Helm release name * **Target namespace ***

9. Click **Select a value** dropdown list under the **Target namespace** to select a namespace for your WAS container.



IBM Cloud Private

View All

ibm-websphere-traditional V 1.0.0

Overview Configuration

Configuration
WebSphere Application Server Base Edition. Edit these parameters for configuration.

Helm release name *

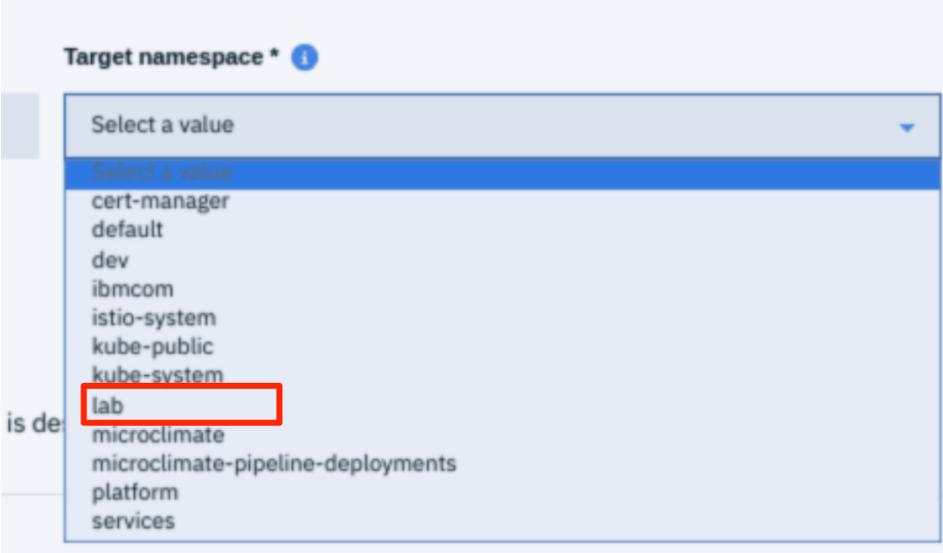
Target namespace *

License * I have read and agreed to the License agreement

Parameters
To install this chart, no configuration is needed. If further customization is desired, view All parameters.

All parameters

10. From the dropdown list, click to select **lab** namespace which is the one you are going to use to deploy the WAS helm chart to.



Target namespace *

- Select a value
- cert-manager
- default
- dev
- ibmcom
- istio-system
- kube-public
- kube-system
- lab
- microclimate
- microclimate-pipeline-deployments
- platform
- services

11. Next click to check the License agreement box.

12. Now click **All parameters** to expand the section, you are going to make a few changes to the default values.

13. Scroll down to Image section. Click **Repository** field to set its value and the **Tag** value to **mycluster.icp:8500/lab/wasv9** and **latest** which are the docker image name you created before.

Number of replicas *

Image
Specify Docker image

Repository * Tag *

Docker image pull policy *

14. Scroll down to Ingress section and click the **Enable Ingress** box to check it.

Ingress
Configure ingress rules that allow inbound connections to reach the cluster services

Enable Ingress * i

Rewrite target * Path *

Configuration Properties
Configure WebSphere Application Server environment

ConfigMap name

By doing this, the form of your **Mod Resorts** app URL running in the WAS instance on the ICP will look like this: **<https://<kubernetes cluster ip>/resorts>**

15. Continue to scroll down to the Persistence section and uncheck the **Use dynamic provision**. The WAS helm chart allows you to set persistent volume to store the log data. In the lab, we are not going to store the log data in the persistent volume.

IBM Cloud Private

Persistence

Configure persistent storage

Name * Size *

File system group ID

Use dynamic provisioning * ⓘ

Storage class name Selector label

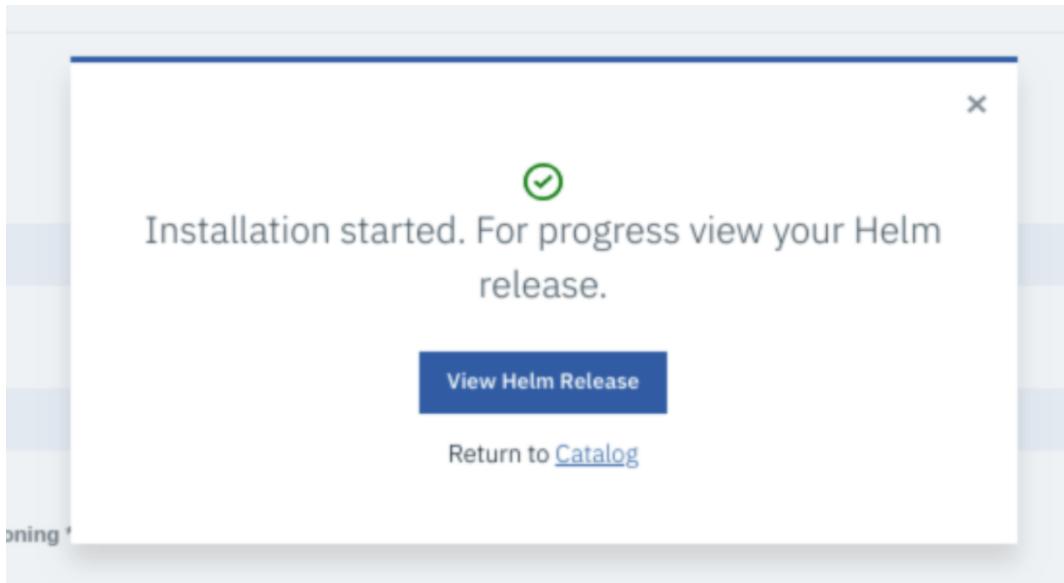
16. Now click **Install** to deploy the WAS Base helm chart to ICP.

Use dynamic provisioning * ⓘ

Storage class name Selector label

Selector value

The WAS Base helm chart is deployed to ICP cluster and you will see a screen like this.

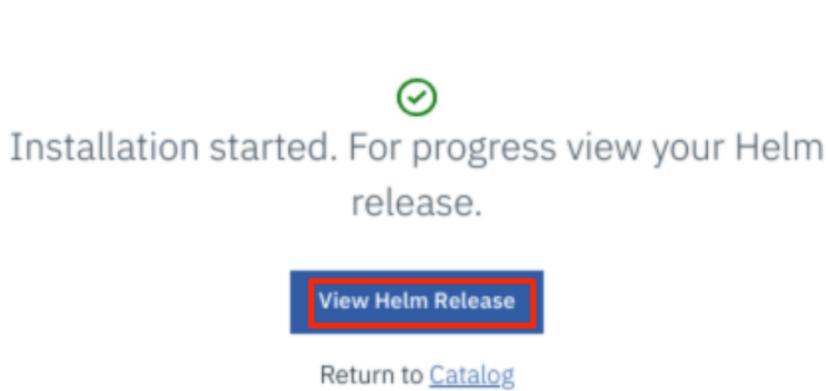


You have created a WAS Base instance running Kubernetes pod on ICP.

6.6 Verify WAS Container Deployment

After the WAS helm chart is deployed, you can go to the ICP console to take a look at the deployment details.

1. Click **View Helm Release** to access the helm chart release page in ICP.



2. In the helm char release page, you can see the details deployment info, including chart namespace, current version, available version, deployment name and Ingress name, pod name and service name.

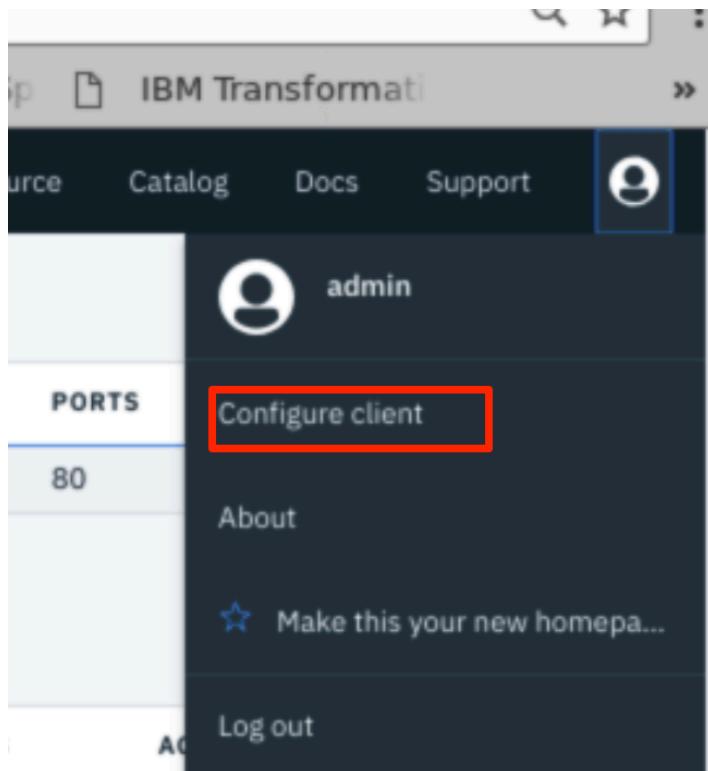
The screenshot shows the IBM Cloud Private interface for the 'was01' application. At the top, it says 'was01' is 'Deployed' and was 'UPDATED: December 13, 2018 at 5:36 AM'. There is a 'Launch' button. Below this, the 'Details and Upgrades' section shows the chart name 'was01' in 'NAMESPACE lab', the current version '1.0.0' (released on December 13, 2018), and the available version '1.0.0' (released on November 16, 2018). It includes 'Upgrade' and 'Rollback' buttons. The 'Deployment' section shows one pod named 'was01-ibm-websphere-trad' with 1 desired, 1 current, and 1 up-to-date pod, all available and 10m old. The 'Ingress' section shows one ingress named 'was01-ibm-websphere-trad' with 1 host, 10.0.0.1 address, and port 80, 10m old. A blue circular icon with a right-pointing arrow is located to the right of the ingress table.

3. Scroll down to the Notes section in the bottom, the chart provides commands to obtain app URL prefix in Kubernetes command line. To run the Kubernetes commands, we need to configure the Kubernetes client first. You can configure it by clicking the **ICP user** icon on the top right corner.

The screenshot shows the IBM Cloud Private interface for the 'was01' application. It includes sections for 'Ingress', 'Pod', 'Service', and 'Notes'. The 'Ingress' section shows the same details as the previous screenshot. The 'Pod' section shows the pod 'was01-ibm-websphere-trad-58fc5bd59f-pjq7c' with 1/1 ready, running, 0 restarts, and 10m age. The 'Service' section shows the service 'was01-ibm-websphere-trad' with type NodePort, cluster IP 10.1.0.217, and port 9443:31003/TCP, 10m old. The 'Notes' section contains the following text:

```
+ Get the application URL by running these commands:
export INGRESS_IP=$(kubectl get nodes -l proxy=true -o jsonpath=".items[0].status.addresses[?(@.type=='Ingress')].address")
export APP_PATH=/sample
echo https://$INGRESS_IP$APP_PATH
```

4. From the user dropdown menu, click **Configure client**.



5. The configure Kubernetes client commands will show up. Click the **copy** icon to copy them to clipboard.

Configure client

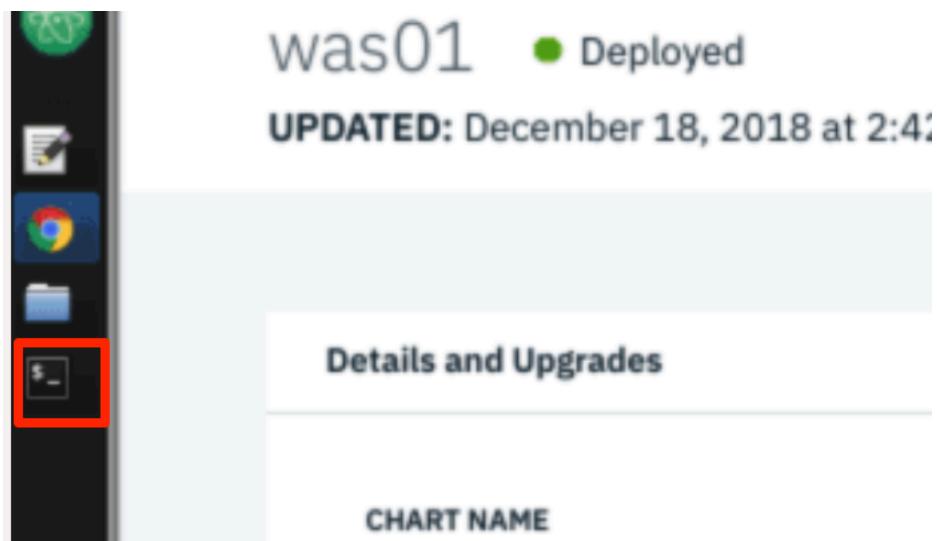
Before you run commands in the kubectl command line interface for this cluster, you must configure the client.

Prerequisites:
Install the kubectl CLI: [kubectl](#)

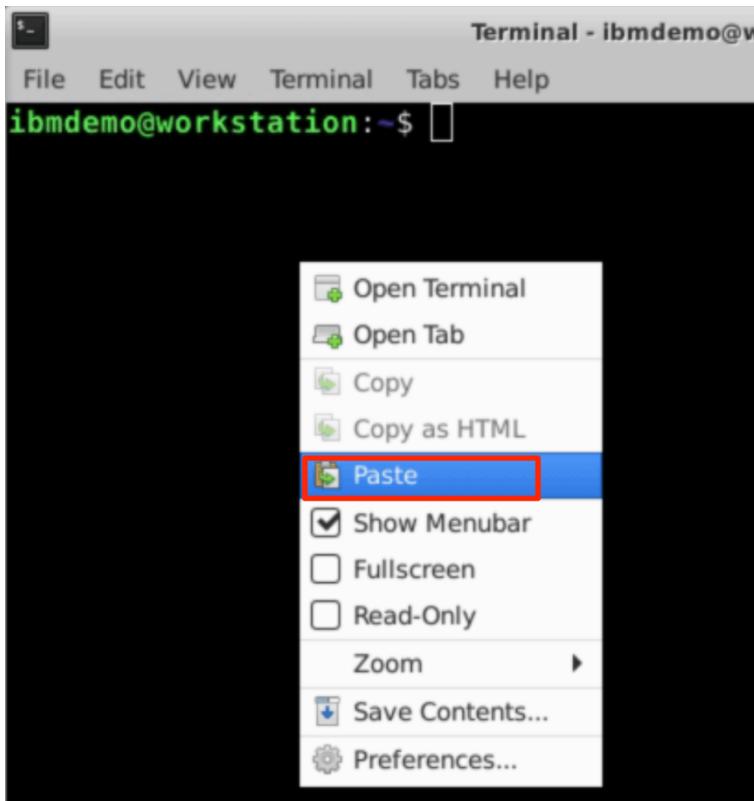
To configure the CLI, paste the displayed configuration commands into your terminal window and run them:

```
kubectl config set-cluster cluster.local --server=https://10.0.0.1:8001
kubectl config set-context cluster.local-context --cluster=cluster.local
kubectl config set-credentials admin --token=eyJ0eXAiOiJKV1QiLCJhb
kubectl config set-context cluster.local-context --user=admin --names
kubectl config use-context cluster.local-context
```

6. Next, go back to the terminal window you worked before by clicking its icon on the desktop tool bar.



7. In the terminal windows, click **Paste** to paste and run the Kubernetes client configuration commands.



You have the Kubernetes client configured.

8. Execute the commands listed in the Notes section of the WAS helm chart release page, you can get the app URL prefix as shown below.

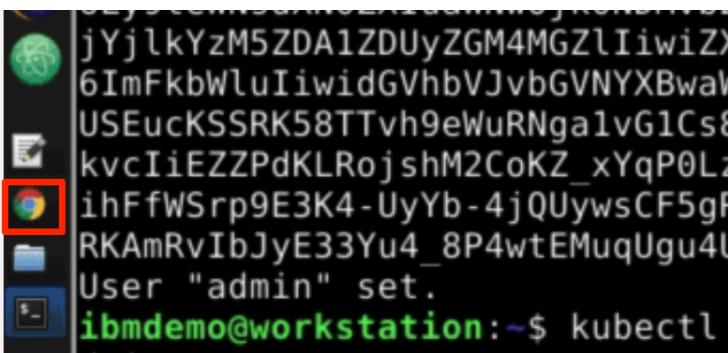
```

INFSWSRp9E3K4-UyYb-4jQUywsCF5gRa1x5S100ttx0Nq_KbLdEmCg1gQLovXes1C19gR0
RKAmRvIbJyE33Yu4_8P4wtEMuqUgu4UbzerNq5Us-FXjA7bj1_T8IH7V8f3EejFg
User "admin" set.
ibmdemo@workstation:~$ kubectl config set-context cluster.local-context
dmin --namespace=cert-manager
Context "cluster.local-context" modified.
ibmdemo@workstation:~$ kubectl config use-context cluster.local-context
Switched to context "cluster.local-context".
ibmdemo@workstation:~$ export INGRESS_IP=$(kubectl get nodes -l proxy=
onpath=".items[0].status.addresses[?(@.type==\"Hostname\")].address"
ibmdemo@workstation:~$ export APP_PATH=/
ibmdemo@workstation:~$ echo https://$INGRESS_IP$APP_PATH
https://10.0.0.1/
ibmdemo@workstation:~$ 

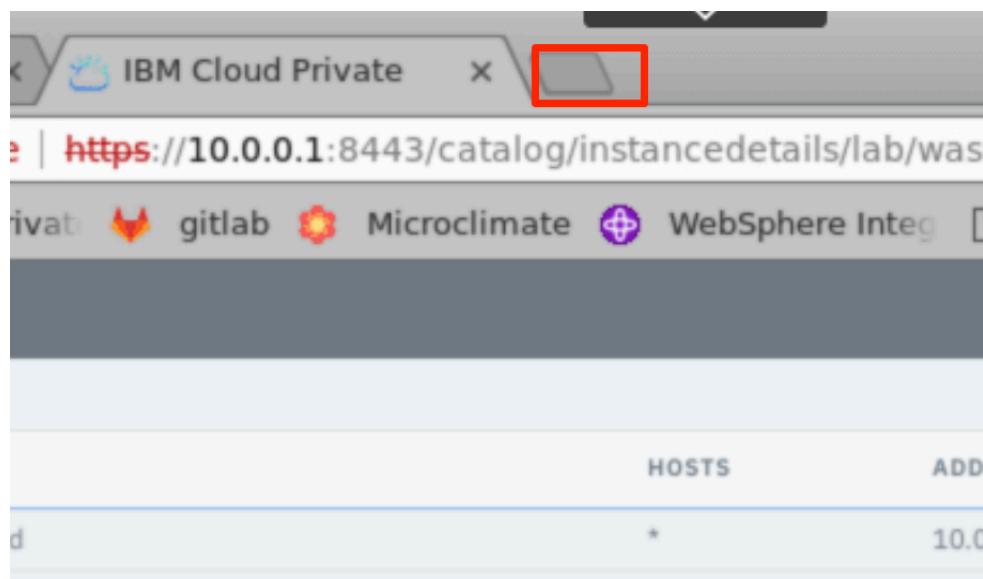
```

Now you have the app URL prefix.

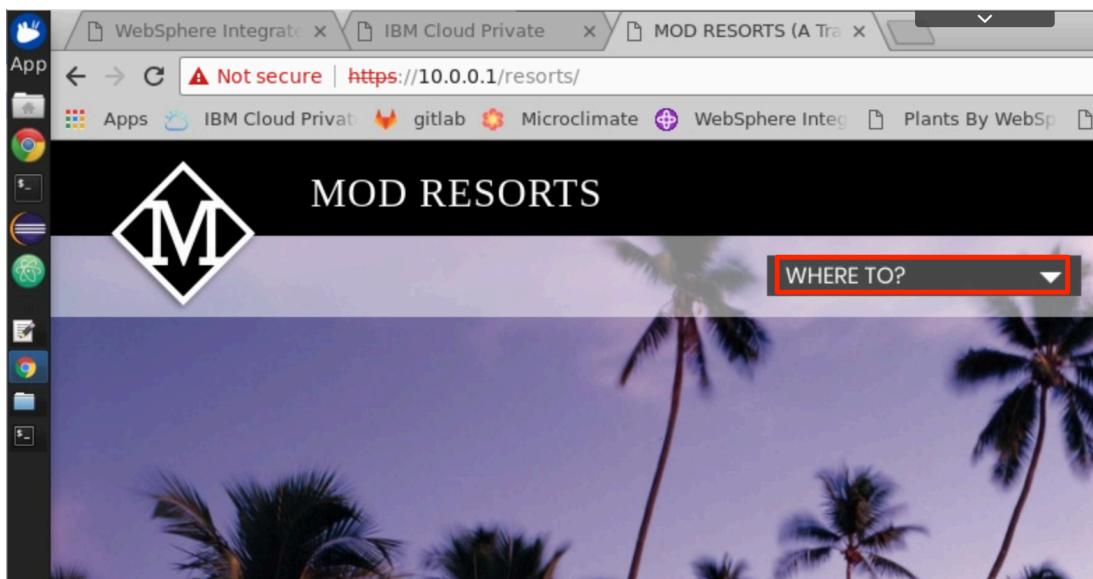
9. Go back to the web browser window by clicking its icon on the desktop tool bar.



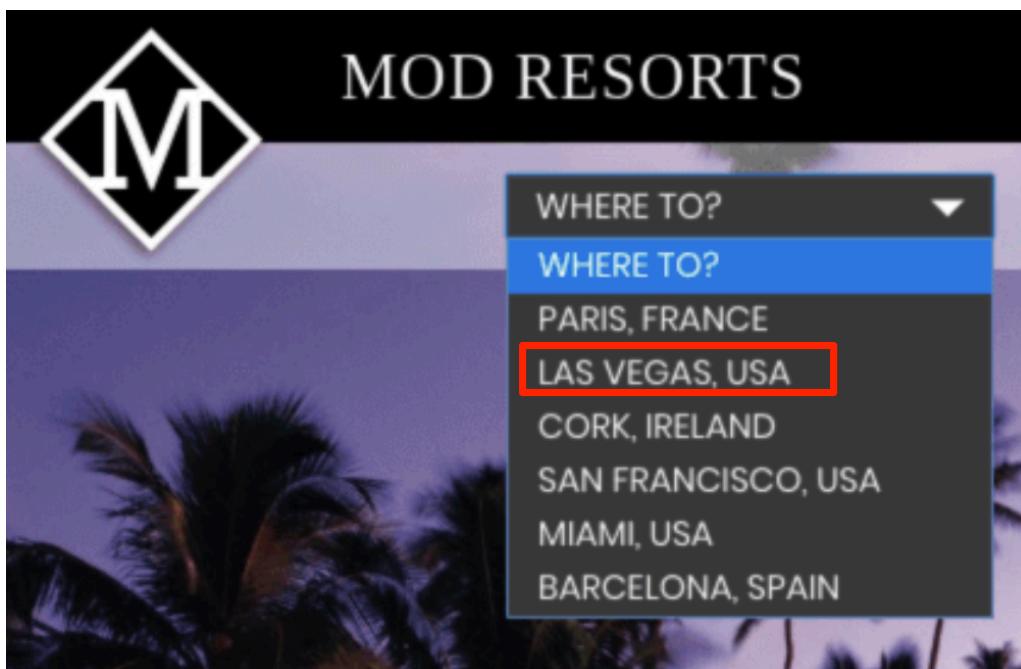
10. Click to open a new browser tab and launch the **Mod Resorts** app with its URL: <https://10.0.0.1/resorts>.



11. The app home page will be shown. Click **WHERE TO?** to view city list.



12. Click **LAS VEGAS, USA** from the list, it will show the weather of the city.



Congratulations! You have successfully completed App Modernization using WAS Cloud Pak lab!

Appendix A. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have

been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Appendix B. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

| | | | | | |
|-----------------|---------------|------------------|------------------|-----------------|------------|
| IBM | AIX | CICS | ClearCase | ClearQuest | Cloudscape |
| Cube Views | DB2 | developerWorks | DRDA | IMS | IMS/ESA |
| Informix | Lotus | Lotus Workflow | MQSeries | OmniFind | |
| Rational | Redbooks | Red Brick | RequisitePro | StrongLoop | System i |
| <i>System z</i> | <i>Tivoli</i> | <i>WebSphere</i> | <i>Workplace</i> | <i>System p</i> | |

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

NOTES

NOTES



© Copyright IBM Corporation 2019.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.



Please Recycle
