IBM Developer
SKILLS NETWORK

Introduction to Red Hat OpenShift

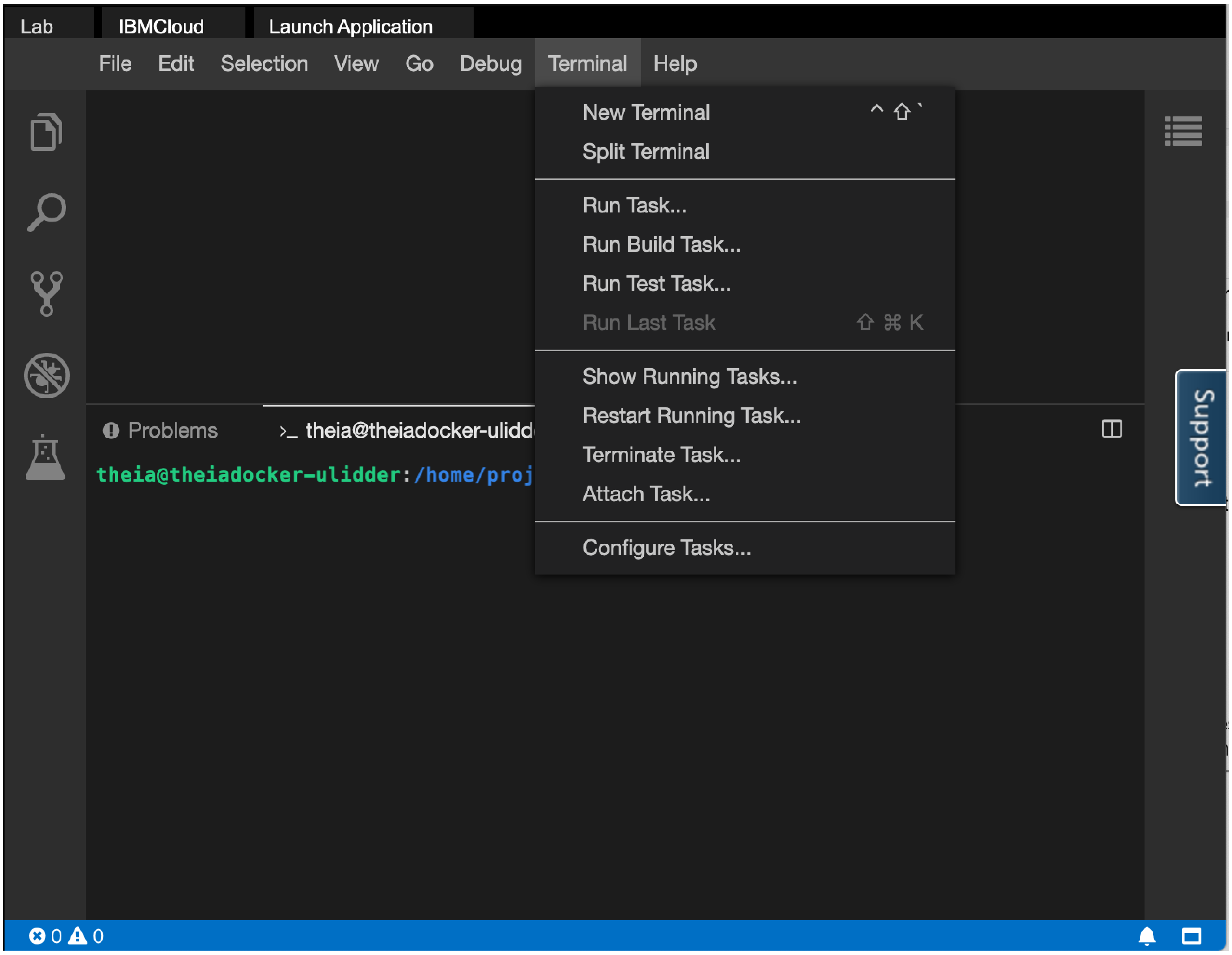
Objectives

In this lab, you will:

- Use the `oc` CLI
- Use the OpenShift web console
- Build and deploy an application using s2i
- Inspect a BuildConfig and an ImageStream

Verify the environment and command line tools

1. If a terminal is not already open, open a terminal window by using the menu in the editor: `Terminal > New Terminal`.



2. Verify that `oc` CLI is installed.

```
oc version
```

You should see output similar to this, although the versions may be different:

```
Client Version: 4.5.0-202002280431-79259a8
Kubernetes Version: v1.16.2+45a4ac4
```

3. Change to your project folder.

```
cd /home/project
```

4. Clone the git repository that contains the artifacts needed for this lab, if it doesn't already exist.

```
[ ! -d 'CC201' ] && git clone https://github.com/ibm-developer-skills-network/CC201.git
```

Use the `oc` CLI

OpenShift projects are Kubernetes namespaces with additional administrative functions. Therefore, projects also provide isolation within an OpenShift cluster. You already have access to one project in an OpenShift cluster, and `oc` is already set to target that cluster and project.

Let's look at some basic `oc` commands. Recall that `oc` comes with a copy of `kubectl`, so all the `kubectl` commands can be run with `oc`.

1. List the Pods in this namespace.

```
oc get pods
```

You will likely see a few Pods that are part of the environment. You don't need to worry about these.

2. In addition to Kubernetes objects, you can get OpenShift specific objects.

```
oc get buildconfigs
```

Because you haven't created a BuildConfig yet, this will not return any resources.

3. View the OpenShift project that is currently in use.

```
oc project
```

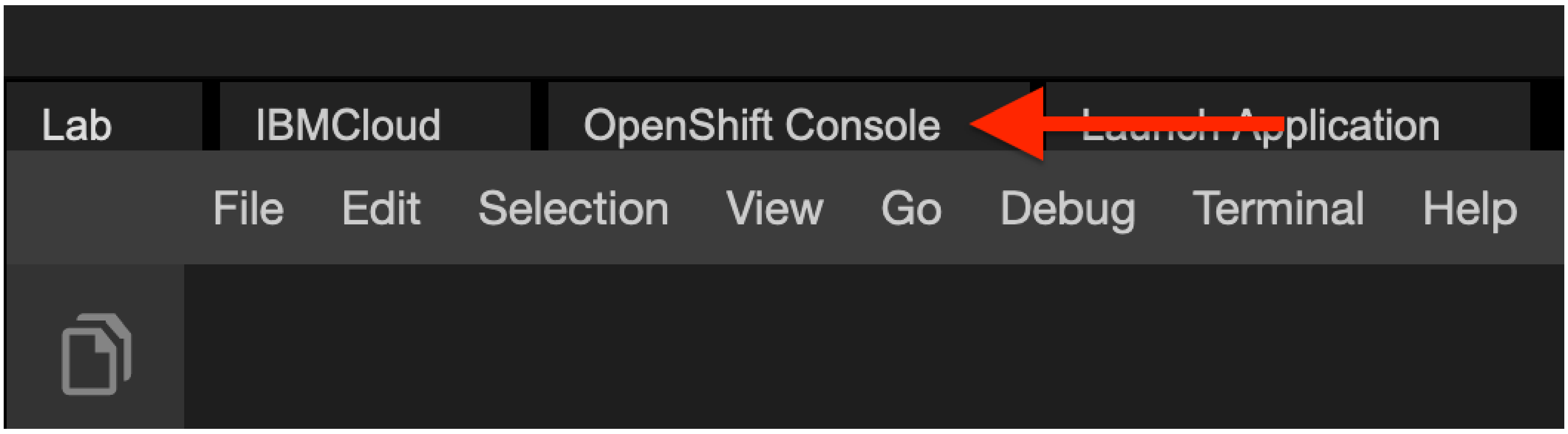
This project is specific to you and provides isolation within the cluster so that you can deploy your own applications.

Use the OpenShift web console

In addition to the CLI, OpenShift provides an intuitive web console. This is a useful and powerful feature because it enables you to deploy applications, view resources, monitor applications and view logs, and much more right in the console.

Let's open up the console and have a look around.

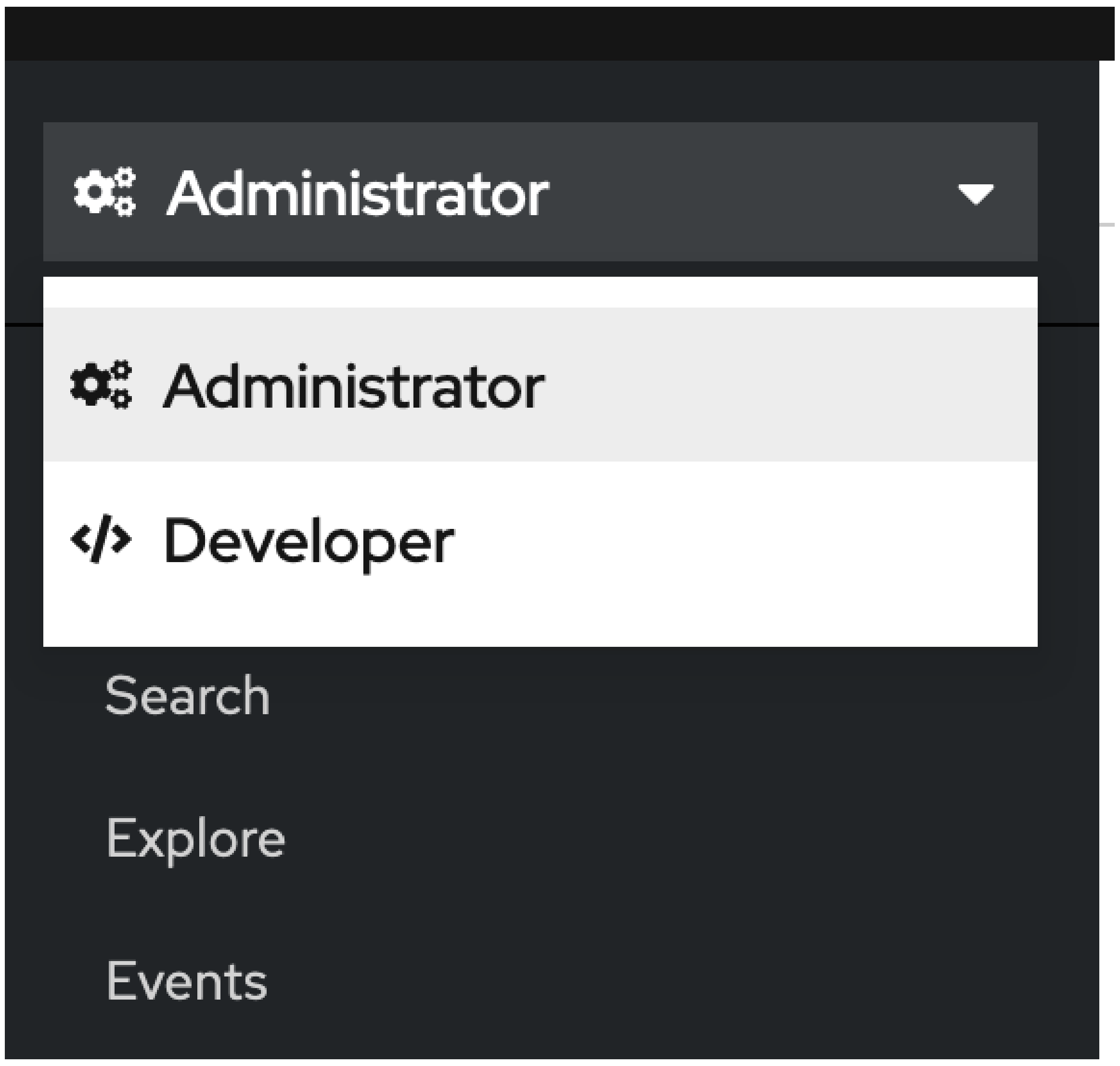
1. Open the OpenShift web console using the link at the top of the lab environment. It can take a few minutes to become available after opening the lab environment, so if you get an error, wait a minute and try again.



2. The console should open to the project details for the project you have been assigned. Take a look at all the information OpenShift provides you in an intuitive, visual manner. Click through the Dashboard, Overview, and other tabs for this project to see additional information. You should see inventory on the resources that currently exist in this project, the YAML that defines this project, and much more.

3. Familiarize yourself with the items in the left navigation menu. You can see Operators, many different Kubernetes objects, and some OpenShift-specific objects, all of which we have talked about in this course. There won't yet be many instances of these objects, but they will fill up once we deploy our application.

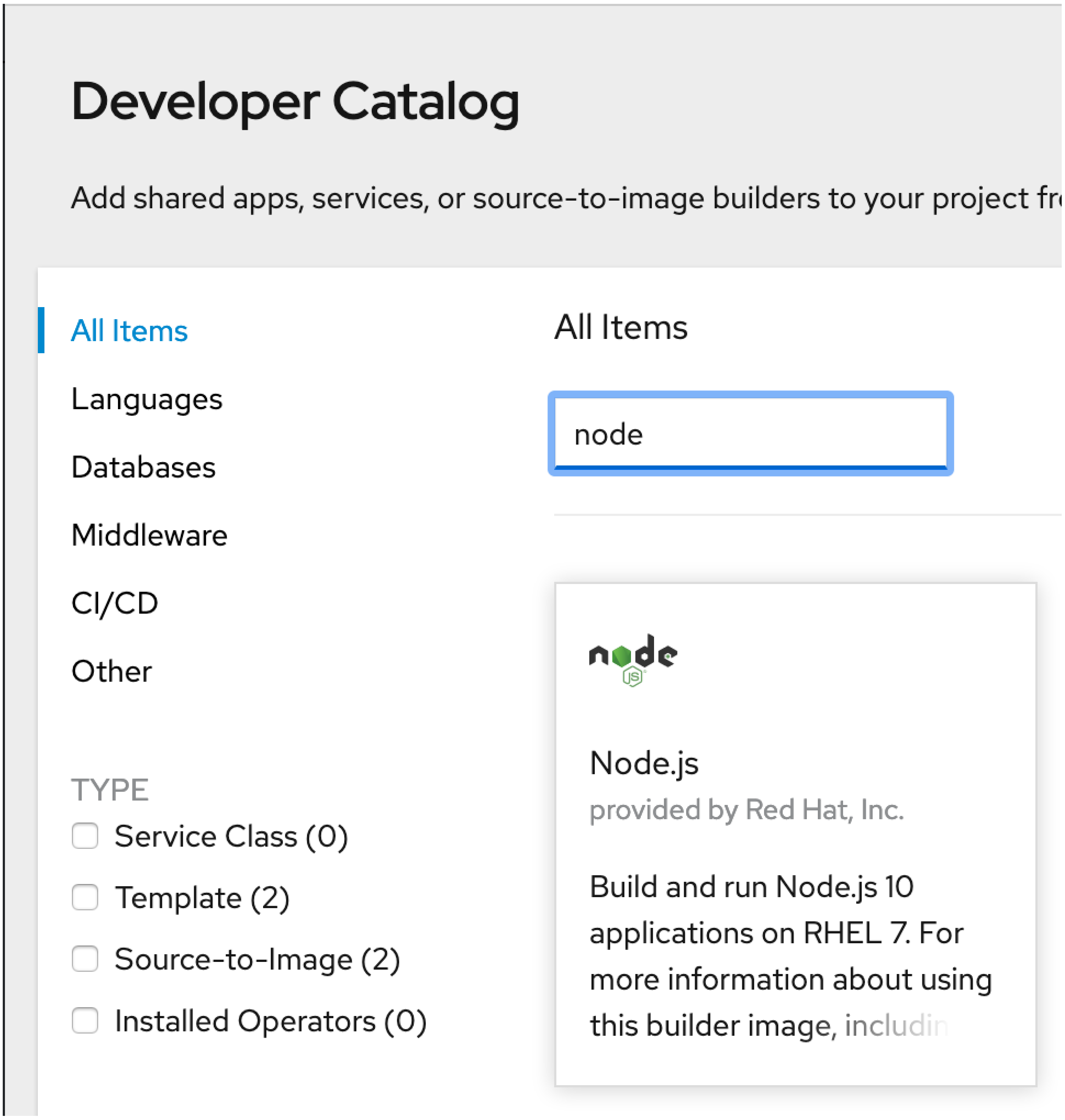
4. Notice the word "Administrator" at the top left. This indicates that you are in the Administrator perspective. There is also a Developer perspective. Each perspective provides workflows specific to that persona. Switch to the Developer perspective to begin deploying an application. (If it says "Developer" already, don't change it.)



Deploy an application in the web console

The Developer perspective provides workflows specific to developer use cases, such as the ability to create and deploy applications. Let's start here! You are likely in the "Topology" view, which provides a visual representation of applications. If not, switch to it to take a look.

1. Click the **+Add** button to add a new application to this project.
2. There are several ways to add a new application in OpenShift. Choose the **From Catalog** option.
3. Search for "node" and click the **Node.js** tile. Click **Create Application** in the window that pops up.


Developer Catalog

Add shared apps, services, or source-to-image builders to your project fr

Node.js
provided by Red Hat, Inc.

Build and run Node.js 10 applications on RHEL 7. For more information about using this builder image, includin

4. You should now be on a screen to create a source-to-image application. Recall that this means OpenShift will deploy an application using only one input from you: the application source. Beneath the **Git Repo URL** box, click **Try Sample**. This will populate the field with the URL of a sample application.

5. Keep the rest of the default options and click **Create**.

In the Topology view, you should now see your newly created application.

View application in the web console

The Topology view provides quick links to a lot of important parts of an application. You can:

- Click the outer circle to get information on the application.
- Click the inner circle with the Node.js logo to get information on the Deployment.
- Click the GitHub icon to access the code repository.
- Click the check mark to view the most recent build (you will see circular arrows if the build is in progress).
- Click the arrow coming out of a box to view the application in the browser if the application is externally available.

Let's try some specific steps:

1. Click the inner circle with the Node.js logo to bring up information on the Deployment.

2. Observe the four resources associated with this Deployment: a Pod that runs the containerized application; a Build that uses the s2i strategy to build the application into a container image; a Service that exposes the application as a network service; and a Route that provides an externally reachable hostname.

3. Click **View logs** on the line that says **Build #1**.

4. Read the logs to see a few key completed steps. The repository is cloned, a Dockerfile is generated, an image is built, and the image is pushed to the internal registry.

5. Click the **Overview** tab for this Build.

6. Click the link to the owning BuildConfig under **Owner**.

7. If you look at the **Overview** and **YAML** tabs, you'll see many concepts that we talked about in this module: triggers, build strategy, webhooks, and more.

8. From the **Overview** tab, click the link to the ImageStreamTag under **Output To**.

9. You can now see the ImageStreamTag that was created as an output of the build. Click the **History** tab to see the image in the internal registry to which this ImageStreamTag points.

10. Return to the Topology view and click on your Deployment info. Click the Route that OpenShift automatically created for you. This will open the application in the browser.

Wow! OpenShift did some pretty incredible work on your behalf. All it needed was a code repository and it was able to build the code into a container image, push that image to a registry, create a Deployment that references that image, and also expose the application to the internet with a hostname.

Congratulations! You have completed the lab for the fourth module of this course.