Introduction to IBM Edge Application Manager - IEAM

Introduction to IBM Edge Application Manager - Think 2021 Lab #2122 workbook

Successfully deploy models to the edge using IBM Edge Application Manager workload orchestration

Introduction

Gain hands-on experience with the IBM Edge Application Manager web management console. Install the Horizon agent, then deploy and manage a Horizon edge service on your edge device.

This Code@Think lab session is available for IBM Think 2021 conference attendees (Sign up for Free!)

- Wednesday, May 12, 2021 3:00 PM to 5:00 PM EDT

Prerequistes

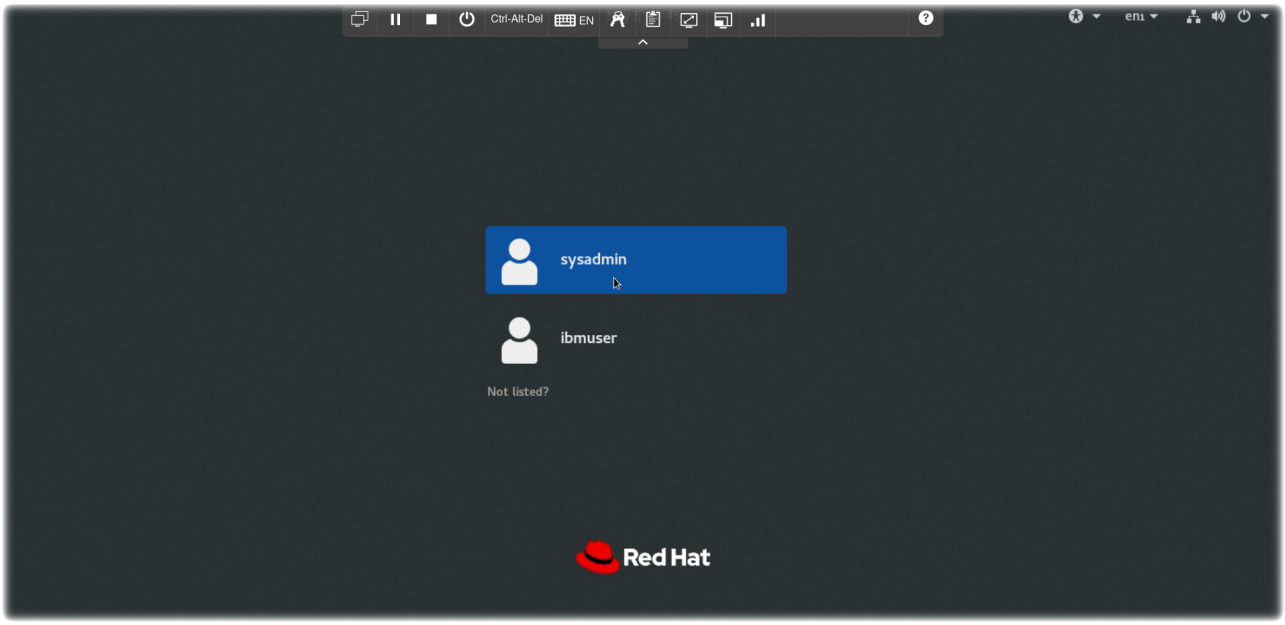
- Create a free IBM Cloud account
- Sign up for Think 2021
- Register for the Introduction to IEAM lab
- Join us on Wednesday, May 12, 2021 3:00 PM to 5:00 PM EDT by logging into the Think portal

Lab Objectives

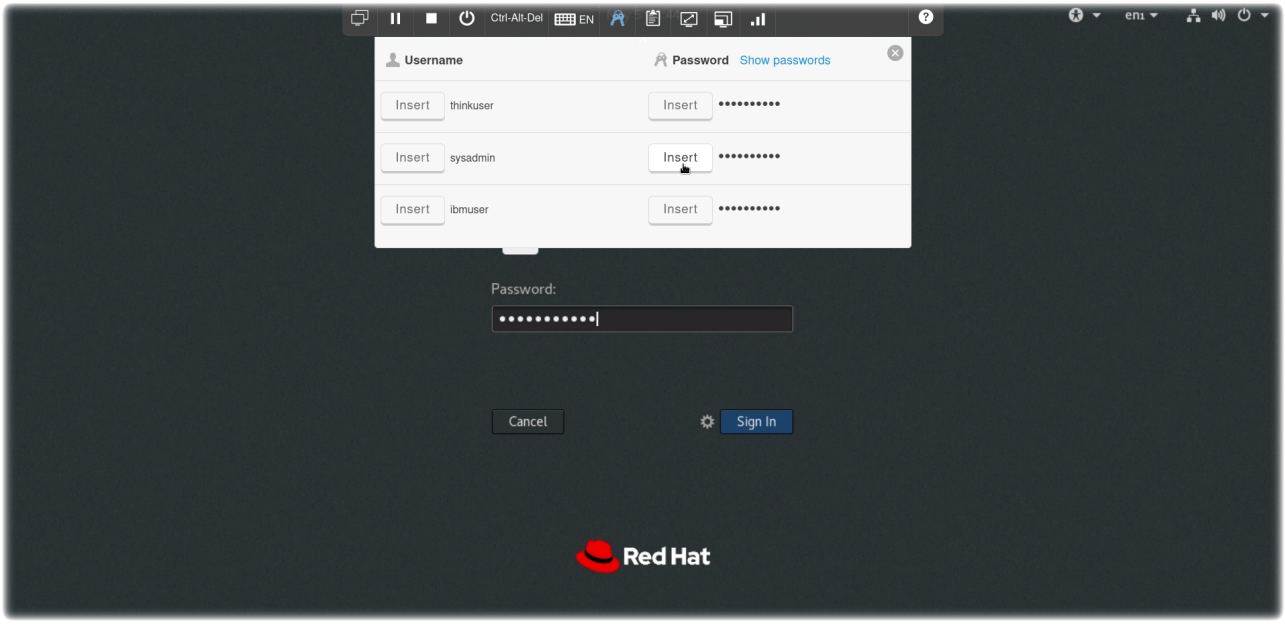
- Access the Think 2021 Lab virtual machine environment
- Learn about IEAM components
- Explore the IEAM web console
- Register an Edge node
- Experiment with the hzn command line interface
- Configure IEAM Services and Policies
- Deploy containerized workloads to your edge device

Access to the Think 2021 Lab Environment

- Select **sysadmin**



- Login Credentials for the virtual machine are stored in the toolbar
- Click on the **Keys** icon in the top toolbar

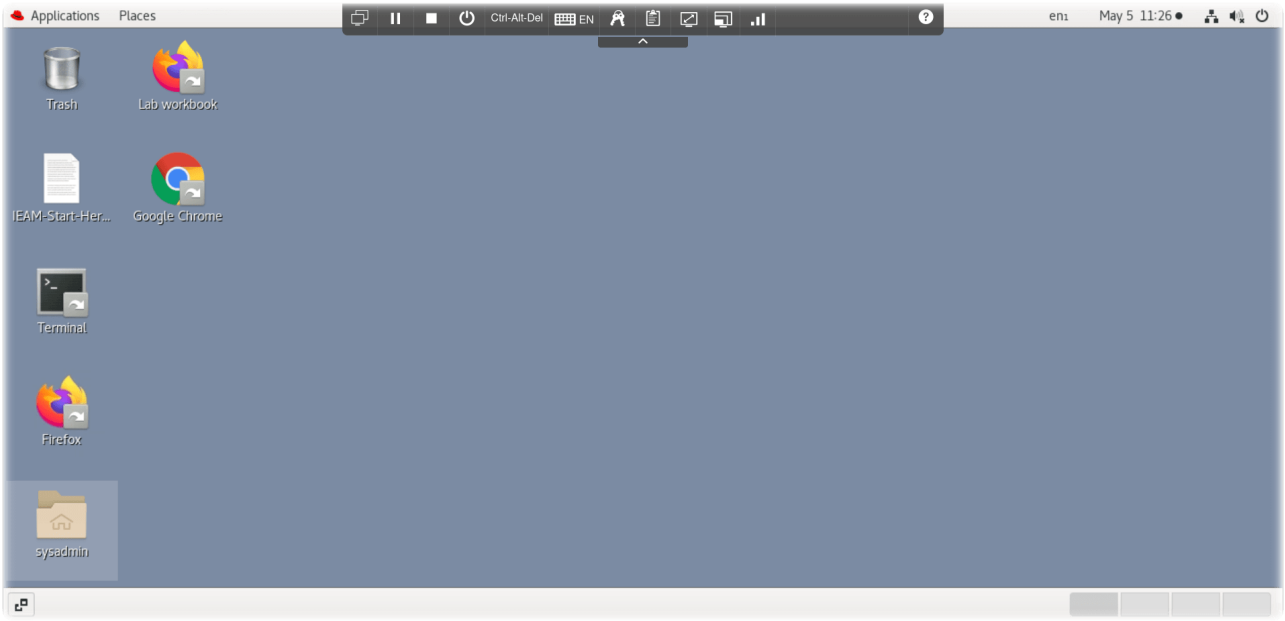


- A drop down dialog will appear with various credentials used in this lab
- Click on the **Insert** button for the sysadmin password
- Click on the **Sign In** button

Navigating the Virtual Machine

This lab will use several components within the RHEL Gnome desktop environment:

- Review the desktop



- a terminal window to execute hzn Horizon edge agent commands
- a browser window with access to the IBM Edge Application Manager web console
- this GitHub repository / PDF which provides exercises to learn about IEAM, the Horizon agent, and containerized edge workloads
- Select the Lab workbook Firefox icon (if you prefer Chrome, select the Chrome icon and then open the 'Lab Workbook' from the bookmark bar)
- The browser will open to the Introduction to IBM Edge Application Manager - IEAM github repo

IEAM Component Definitions

Before proceeding to the lab instructions, let's review some IEAM component definitions.

- *IEAM Management Hub* - The web UI used by IEAM administrators to view and manage the other components of IBM Edge Application Manager.
- *Node* - Typically, **edge devices** have a prescriptive purpose, provide (often limited) compute capabilities, and are located near or at the data source. Currently supported IEAM edge device OS and architectures:
  - x86_64
    - Linux x86_64 devices or virtual machines that run Ubuntu 20.x (focal), Ubuntu 18.x (bionic), Debian 10 (buster), Debian 9 (stretch)
    - Red Hat Enterprise Linux® 8.2
    - Fedora Workstation 32
    - CentOS 8.2
    - SuSE 15 SP2
  - ppc64le (support starting Horizon version 2.28)

        - Red Hat Enterprise Linux® 7.9
    - ARM (32-bit)
        - Linux on ARM (32-bit), for example Raspberry Pi, running Raspberry Pi OS buster or stretch
    - ARM (64-bit)
        - Linux on ARM (64-bit), for example NVIDIA Jetson Nano, TX1, or TX2, running Ubuntu 18.x (bionic)
    - Mac
        - macOS
- *Containerized Workload* - Any Docker/OCI containerized service, microservice, or piece of software that does meaningful work when it runs on an edge node.
- *Service* - A service that is designed specifically to be deployed on an edge cluster, edge gateway, or edge device. Visual recognition, acoustic insights, and speech recognition are all examples of potential edge services.
- *properties* - name/value pairs, often used to describe attributes of nodes (like model, serial number, role, capabilities, attached hardware. etc.) or attributes of a service or deployment (see “policy”).
- *constraints* - logical expressions in terms of “properties”. Constraints are used to control and manage software deployment to edge nodes.
- *policy* - a collection of zero or more properties and zero or more constraints, sometimes with additional data fields (see “node policy”, “service policy”, and “deployment policy”
- *node policy* - a set of properties and constraints related to an edge node (either a stand-alone Linux edge node or a Kubernetes cluster node)
- *service policy* - a set of properties and constraints related to a specific deployable service
- *business policy* - (deprecated) the former name for “deployment policy”
- *deployment policy* - a set of properties and constraints related to the deployment of a specific service together with an identifier for the service version to deploy, and other informatuion such as how rollbacks should be handled when failures occur.
- *pattern* - another name for “deployment pattern”
- *deployment pattern* - a list of specific deployable services. Patterns are a simplification of the more general, and more capable, “policy” mechanism. Edge nodes can register with a deployment pattern to cause the pattern’s set of services to be deployed.
- *IEAM Edge Cluster* - IBM Edge Application Manager (IEAM) edge cluster capability helps you manage and deploy workloads from a management hub cluster to remote instances of OpenShift® Container Platform or other Kubernetes-based clusters. Edge clusters are IEAM edge nodes that are Kubernetes clusters. An edge cluster enables use cases at the edge, which require colocation of compute with business operations, or that require more scalability, availability, and compute capability than what can be supported by an edge device. IEAM edge cluster configuration is outside the scope of this introduction lab.

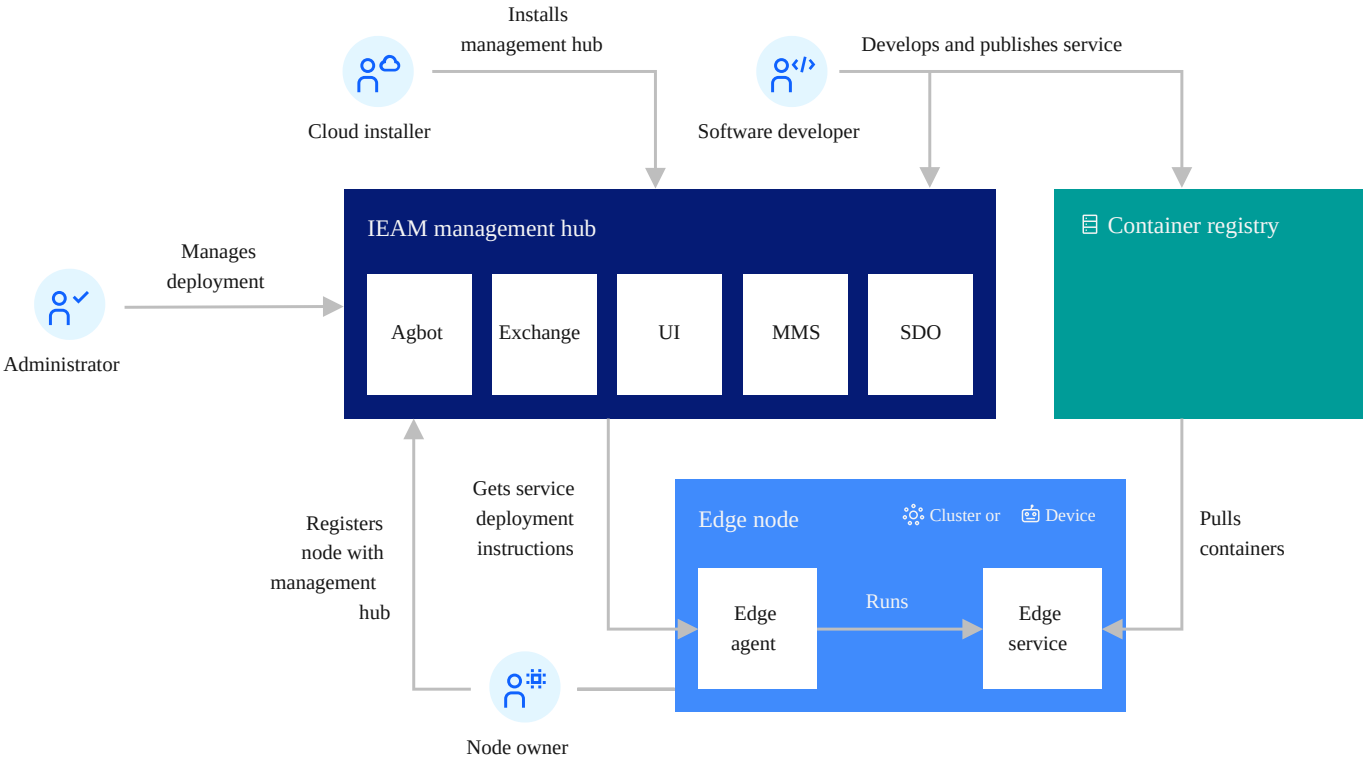
Architecture

The goal of edge computing is to harness the disciplines that have been created for hybrid cloud computing to support remote operations of edge computing facilities. IEAM is designed for that purpose.

The deployment of IEAM includes the management hub that runs in an instance of OpenShift Container Platform installed in your data center. The management hub is where the management of all of your remote edge nodes (edge devices and edge clusters) occurs.

These edge nodes can be installed in remote on-premises locations to make your application workloads local to where your critical business operations physically occur, such as at your factories, warehouses, retail outlets, distribution centers, and more.

The following diagram depicts the high-level topology for a typical edge computing setup:

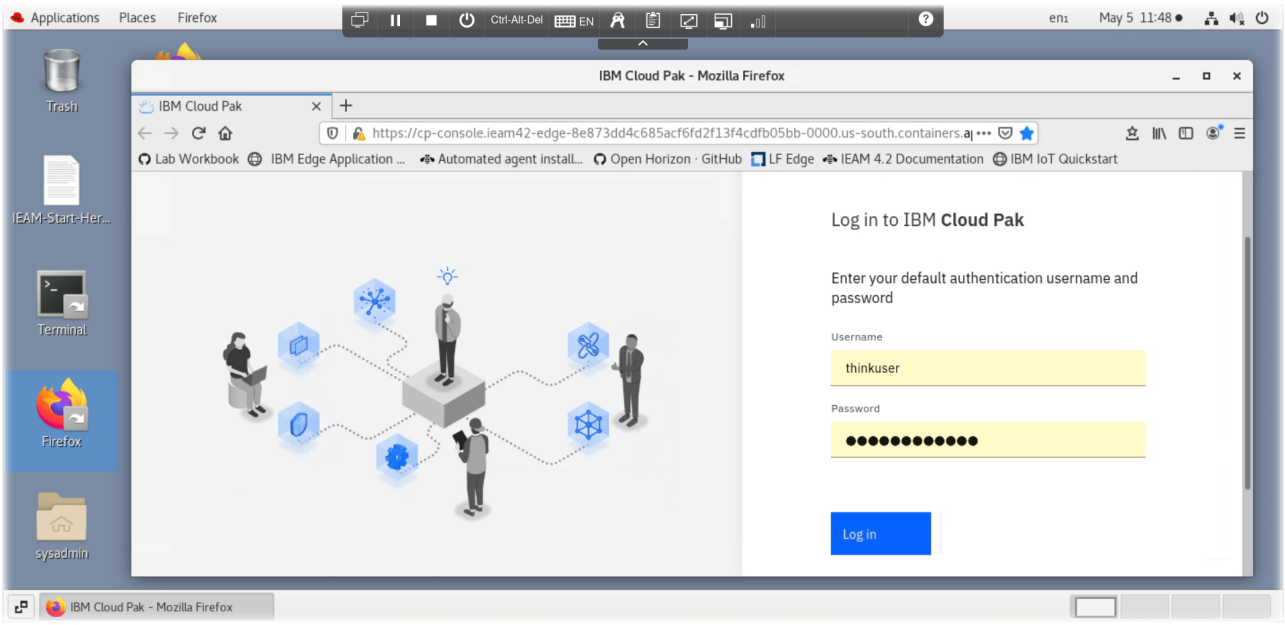


Let's Get Started

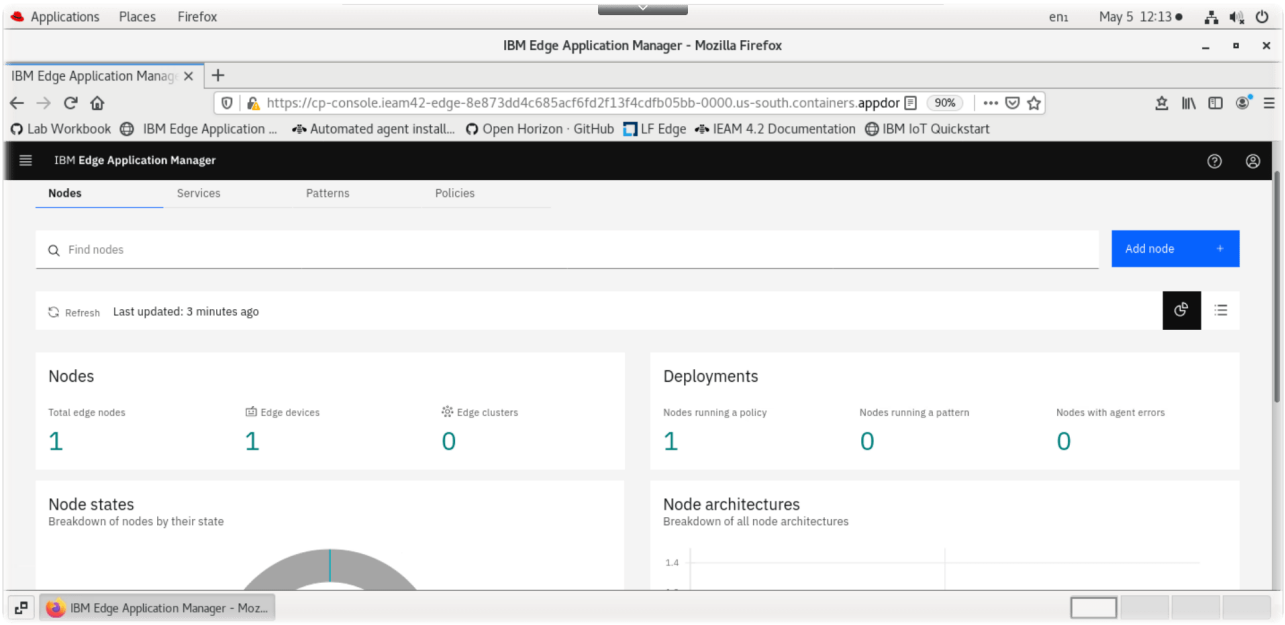
Explore the IEAM web console

In this section, explore the IEAM mgmt web console.

- Click either the Firefox or Chrome browser icon on your desktop
  - **Note** if you select Chrome, launch IBM Edge Application Manager from the browser toolbar. You will also need to click on the Advanced button and proceed to the self-signed certificate. Choose Default Authenication.
- Login into the IEAM console

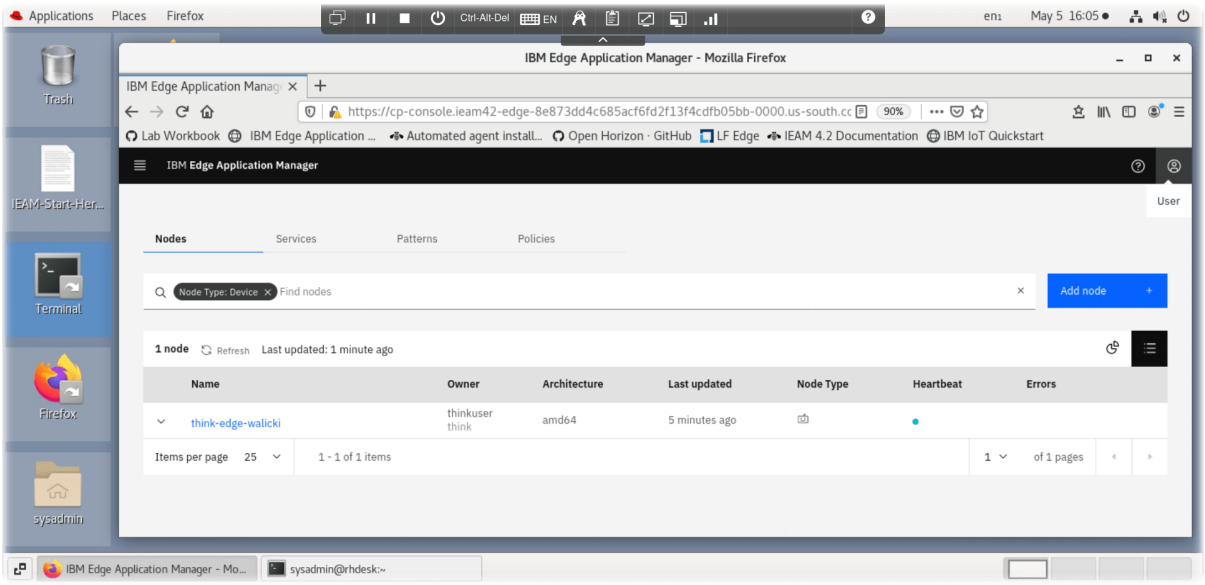


- The `thinkuser` userid / password are already saved in your browser keystore but if you need to, use the **Keys** toolbar to insert the password.

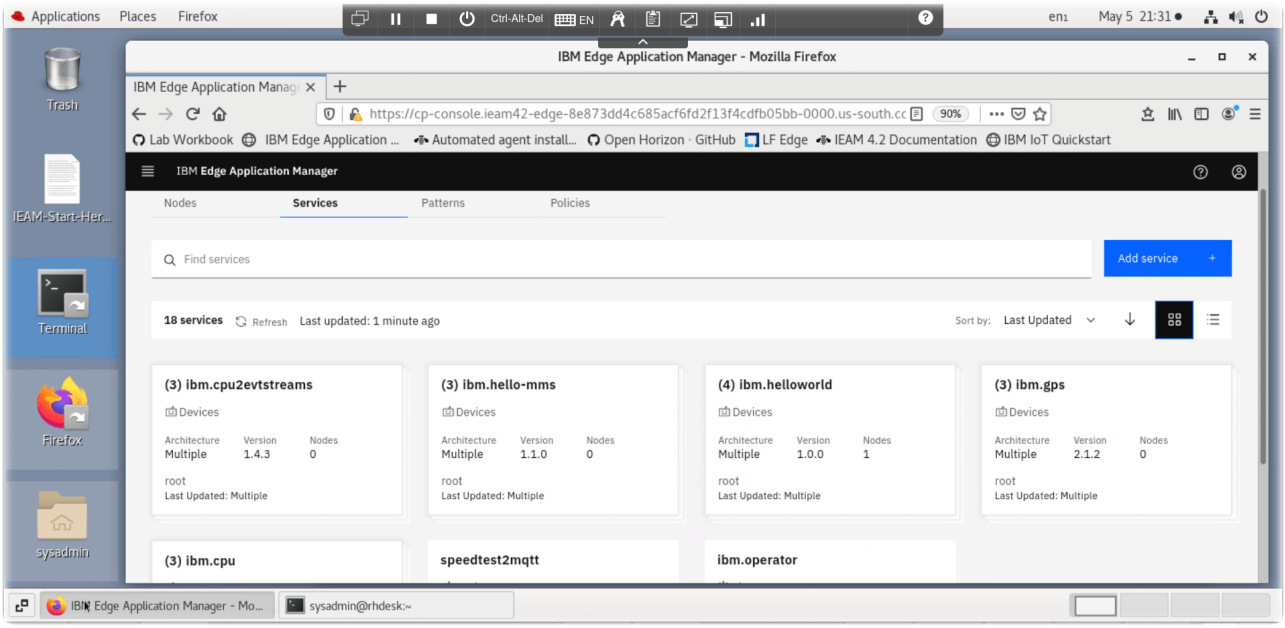


- **Note:** If you receive a `403` error, modify the browser URL in the address bar, remove the `common-nav/403` and replace it with `edge`
- Nodes
  - Manage the node properties and constraints

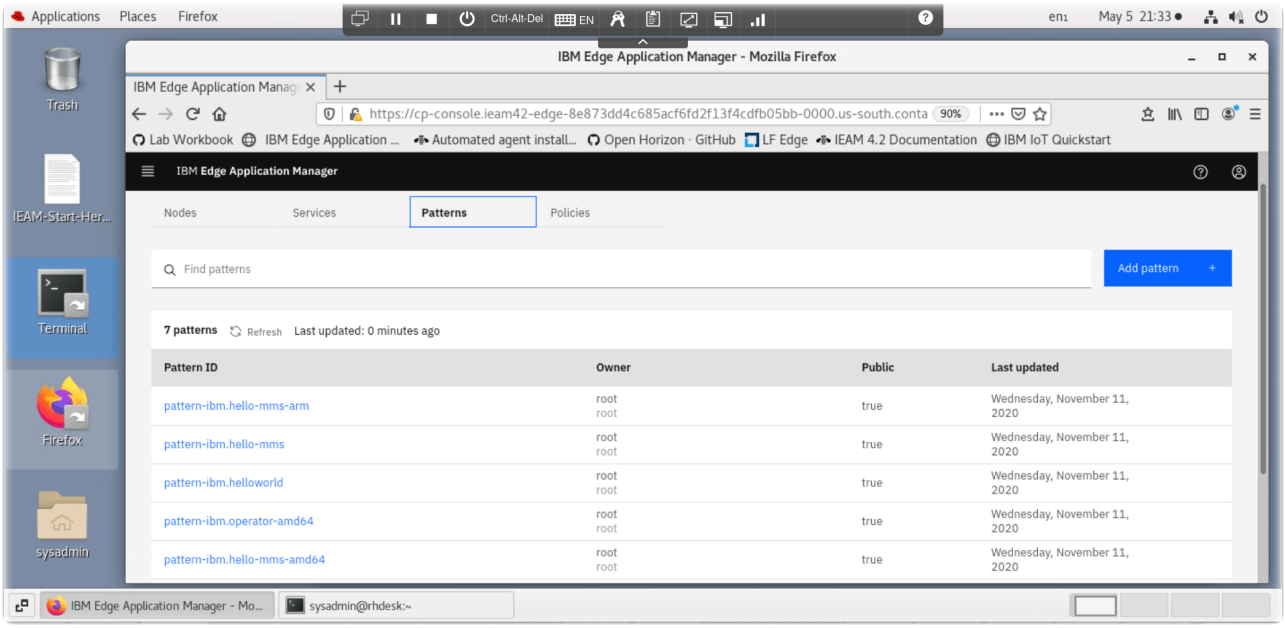
- **Note** there are only a few edge nodes registered. In a subsequent section of the lab, your edge node will appear here.



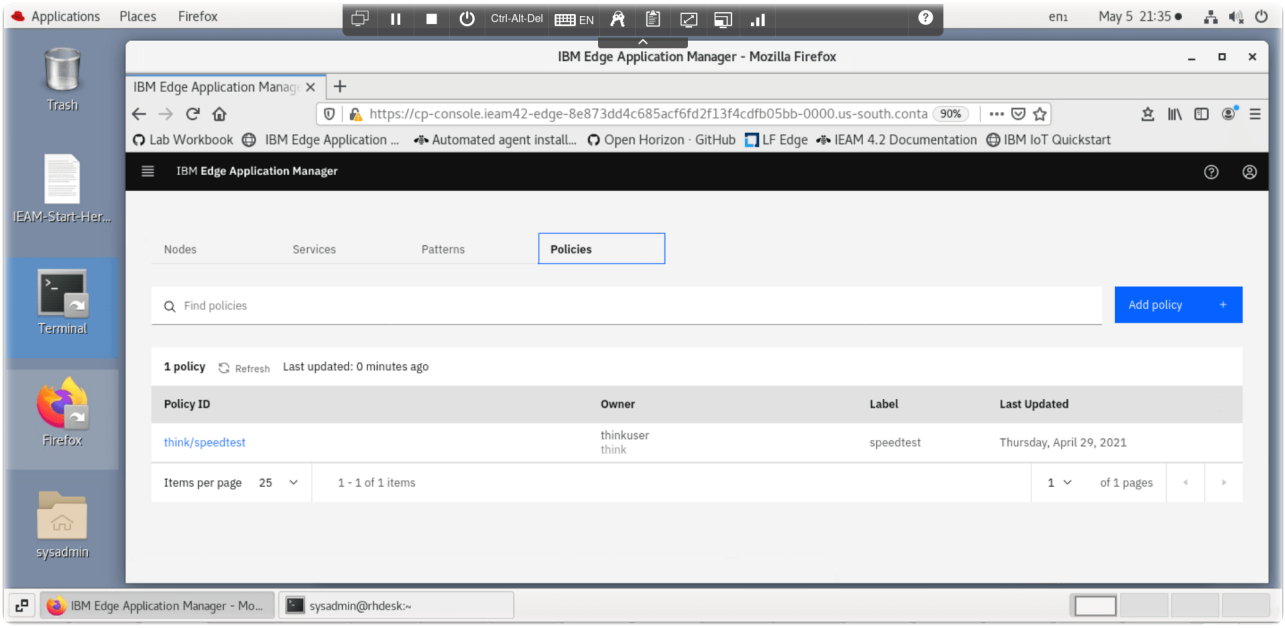
- Services



- Patterns



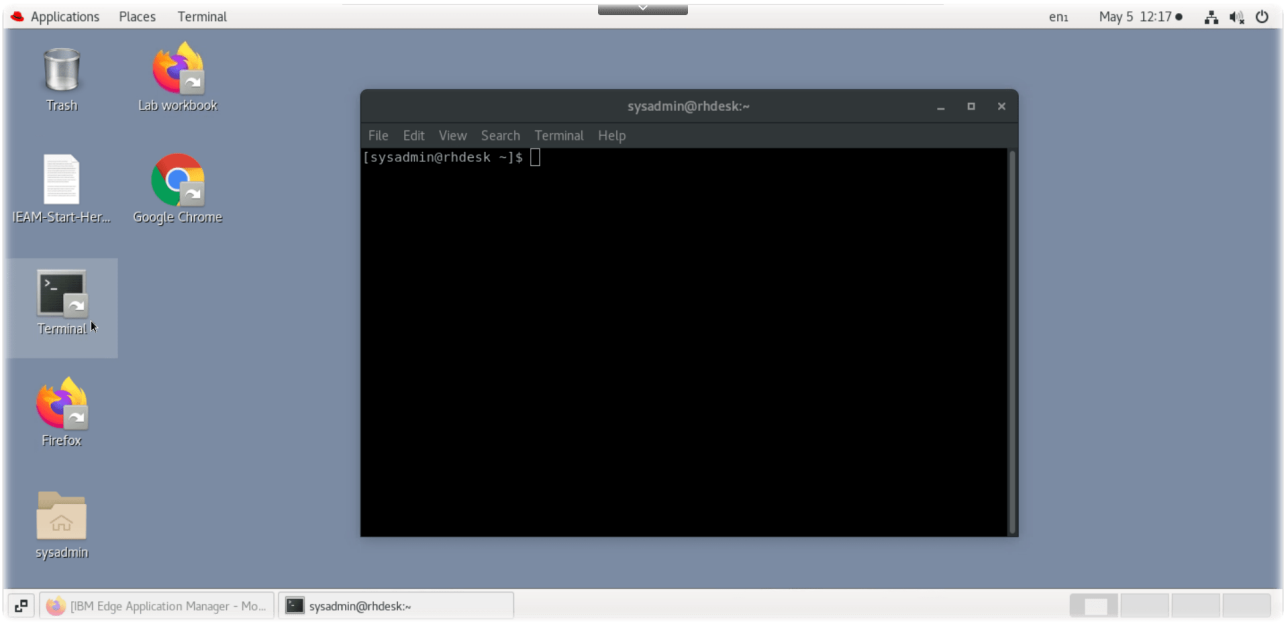
- Policies



Explore the "Edge" Device

The virtual machine lab environment provided during the Think 2021 lab will be your "edge device" today. Normally, your edge device might be an industrial computer, small board computer, etc.

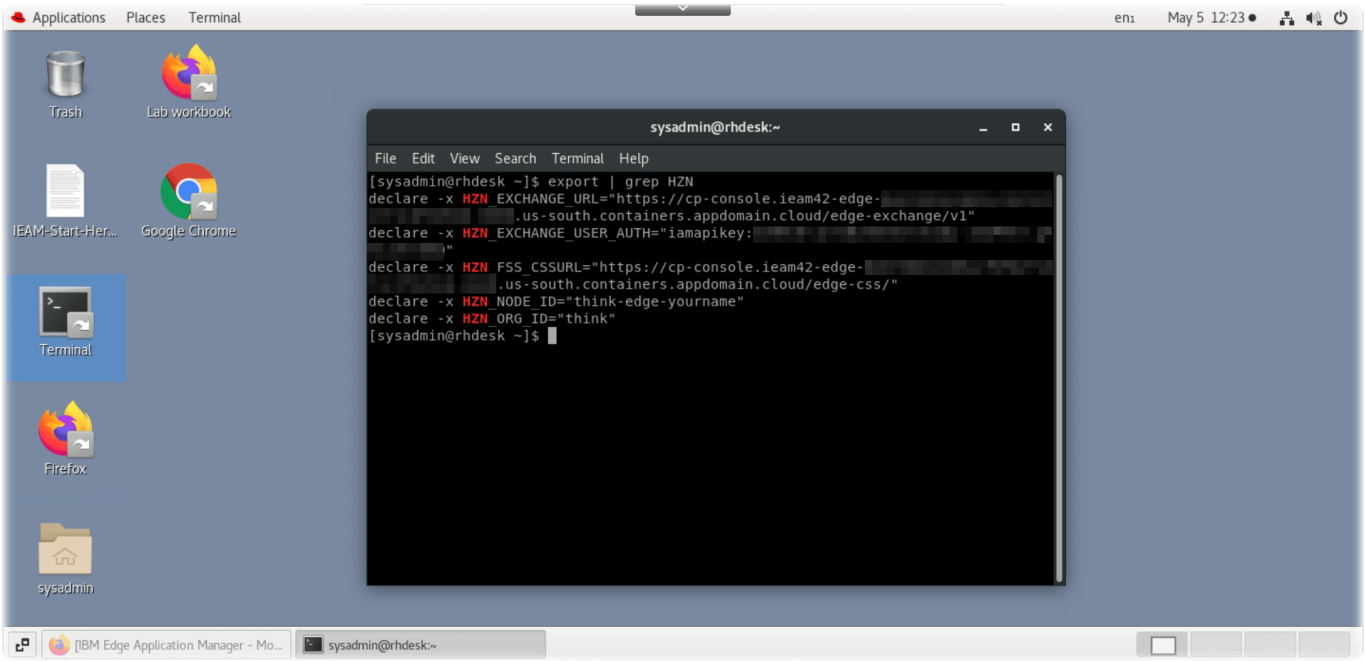
- Click on the **Terminal** icon and launch a console window.



- To simplify the edge lab setup, the HZN environment variables have been pre-configured in the ~/.bashrc profile

- Inspect the HZN environment variables by entering this shell pipeline

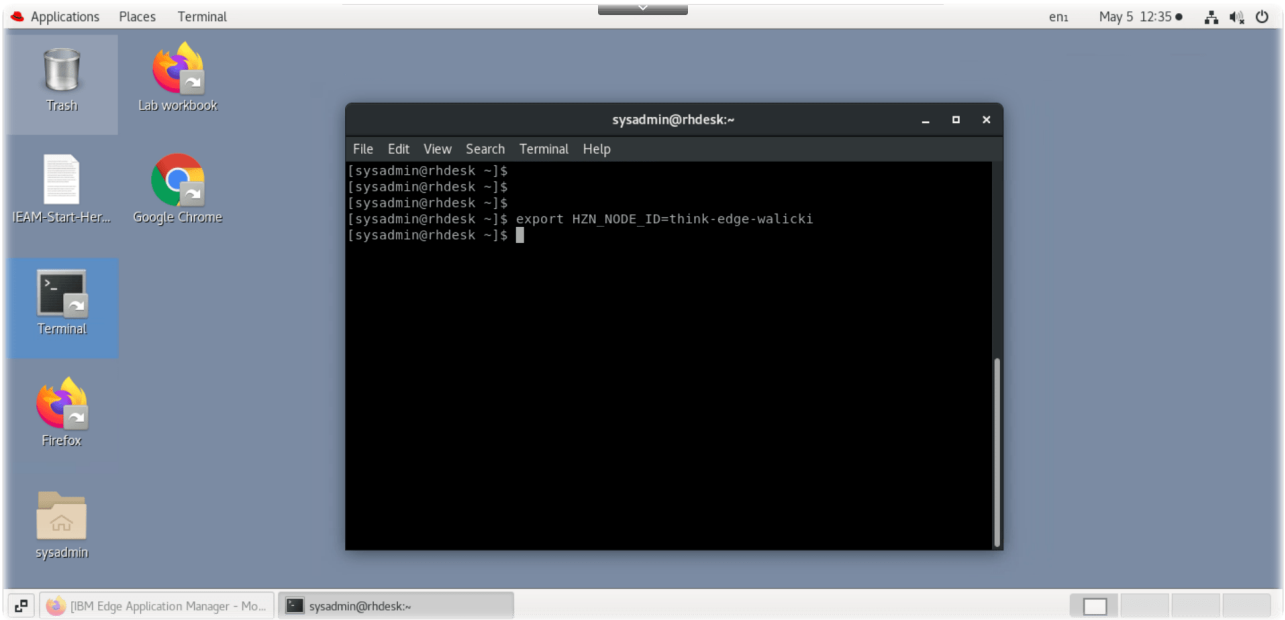
```
export | grep HZN
```



Note: (the urls and iamapikey have been blurred in the public screenshots)

- Set your HZN_NODE_ID environment variable to something unique which will allow you to find your "edge" device in the IEAM management console. eg `think-edge-<yourname>`

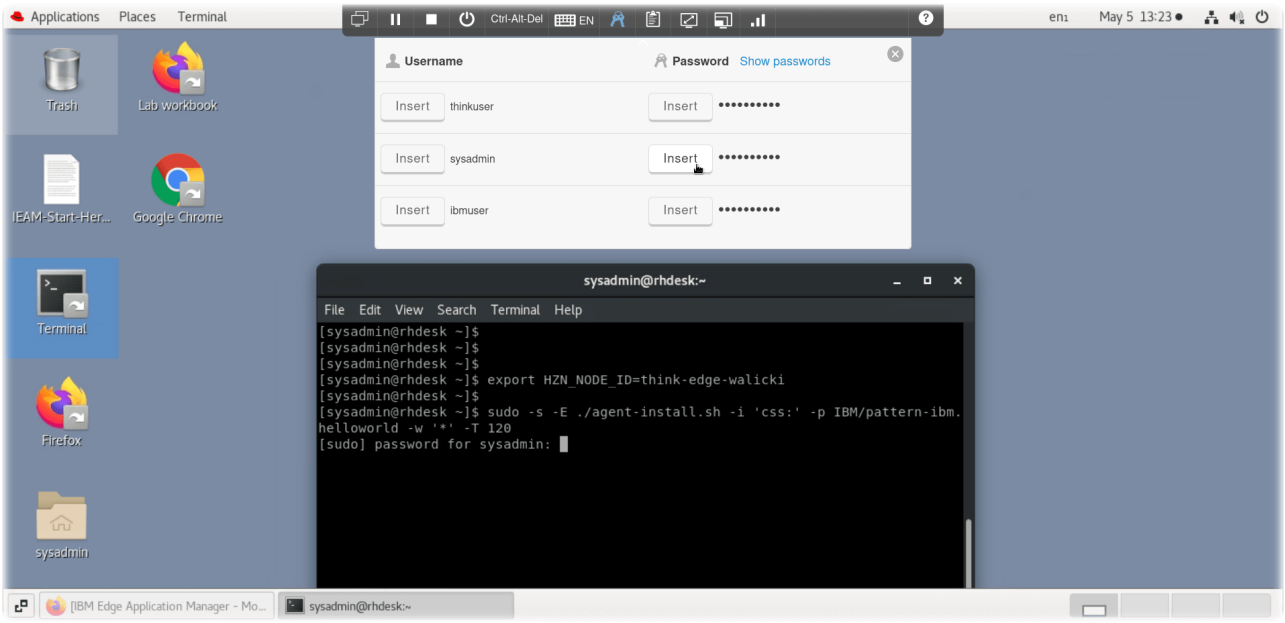
  ```
  export HZN_NODE_ID=think-edge-<yourname>
  ```



- **Stop** : Did you `export HZN_NODE_ID=think-edge-<yourname>` as required in the above step? You won't be able to find your device in the IEAM mgmt console if you don't set a node name.

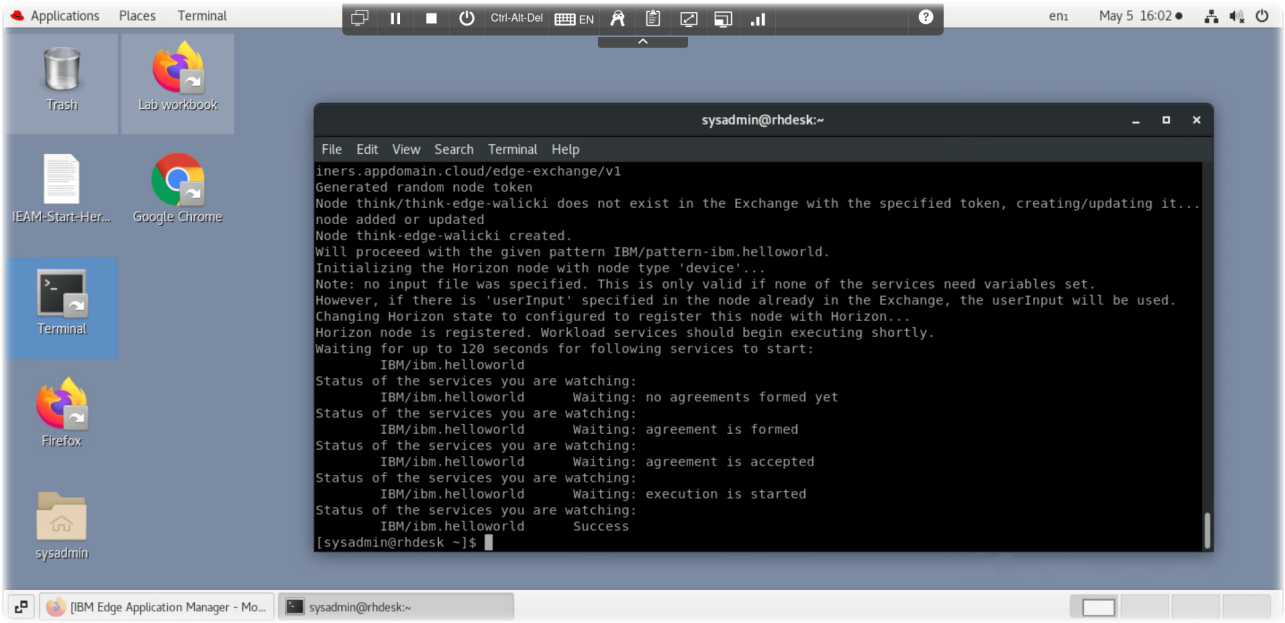
- Install / Register your edge node

```
sudo -s -E ./agent-install.sh -i 'css:' -p IBM/pattern-ibm.helloworld
-w '*' -T 120
```



Use the **Keys** toolbar to insert the `sysadmin` password at the `sudo` prompt

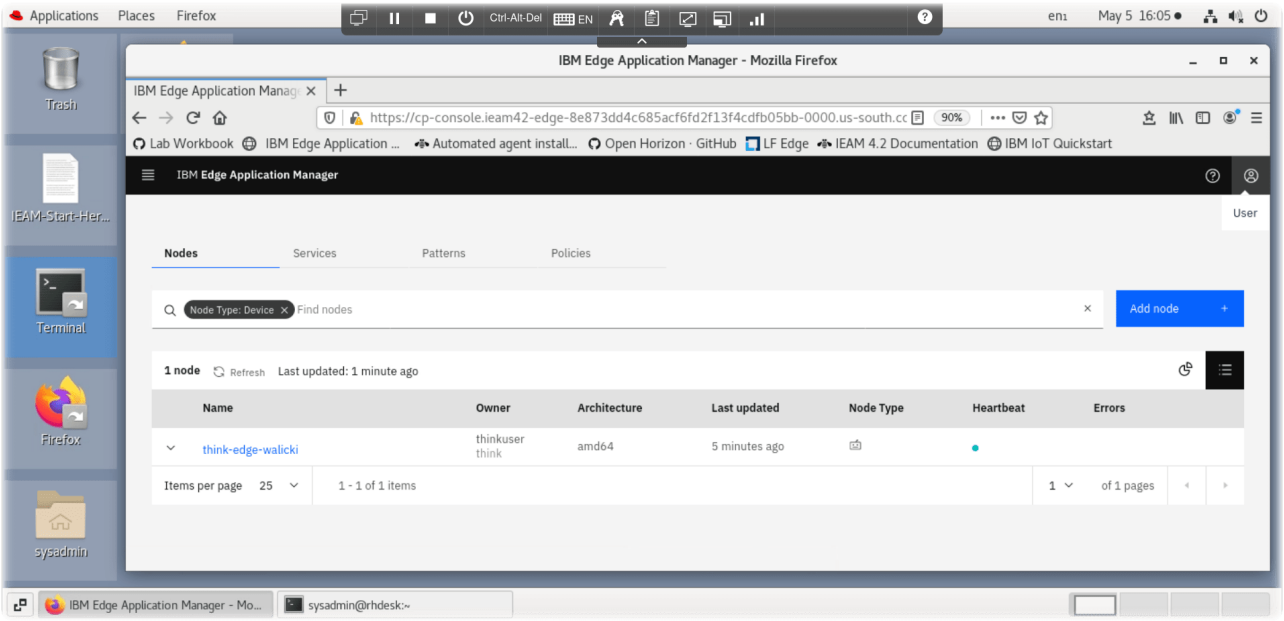
- When the node registration completes, it will have also configured an `IBM/ibm.helloworld` pattern to execute on this node.



- Query the version that was installed

```
hzn version
Horizon CLI version: 2.28.0-338
Horizon Agent version: 2.28.0-338
```

- Return to the IEAM web console. Refresh, Find your device

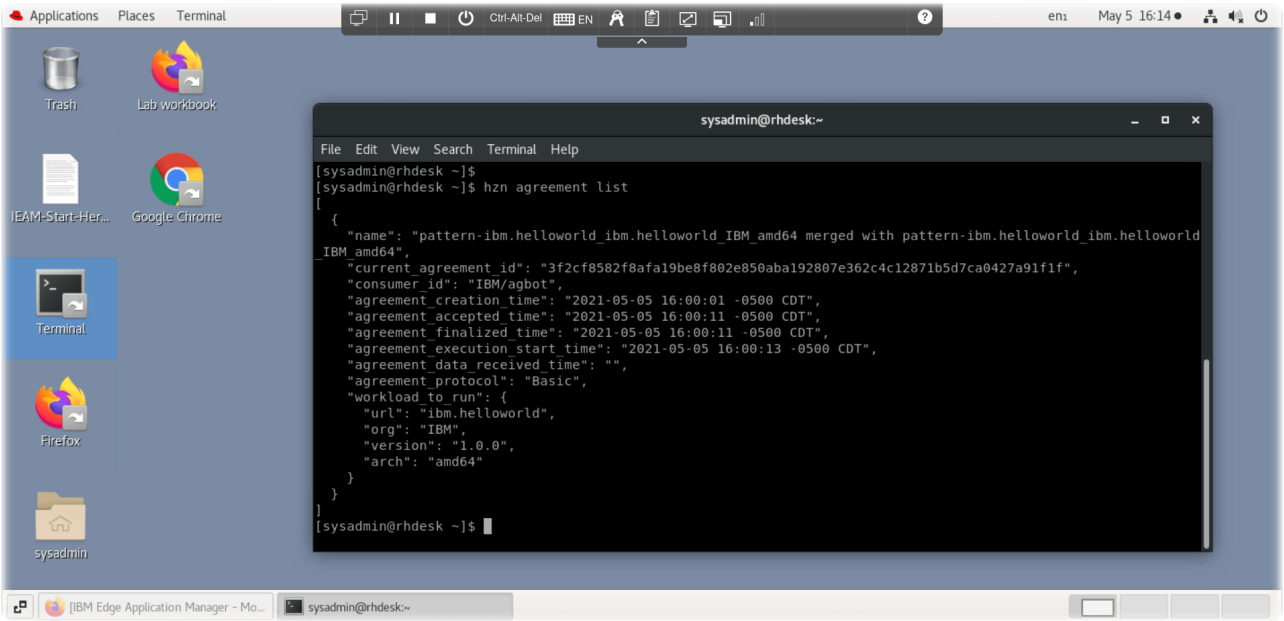


- If you don't see your node, or you skipped the step which sets the name of your edge node, `export HZN_NODE_ID=think-edge-<yourname>` , you can unregister the node and try again.

```
sudo hzn unregister -vrfD
export HZN_NODE_ID=think_edge_<yourname>
sudo -s -E ./agent-install.sh -i 'css:' -p IBM/pattern-ibm.helloworld
-w '*' -T 120
```

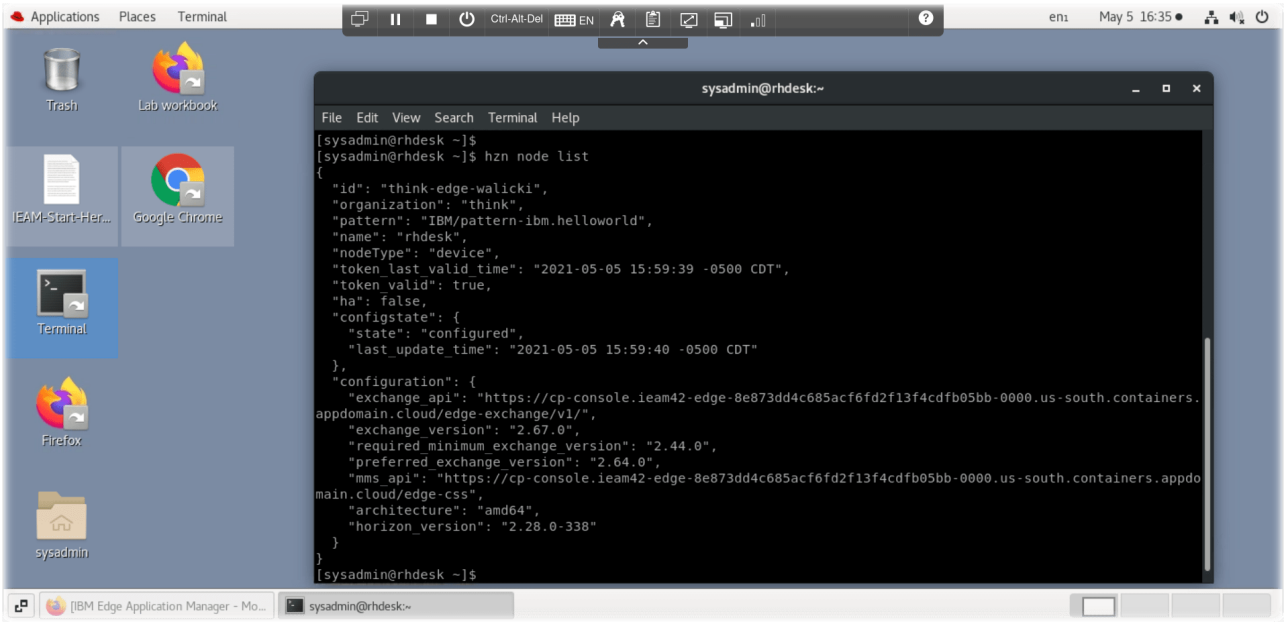
- Return to the Terminal window. See what agreements are running.

```
watch hzn agreement list
```



- Query details about your node

```
hzn node list
```



Using the Horizon HZN command line

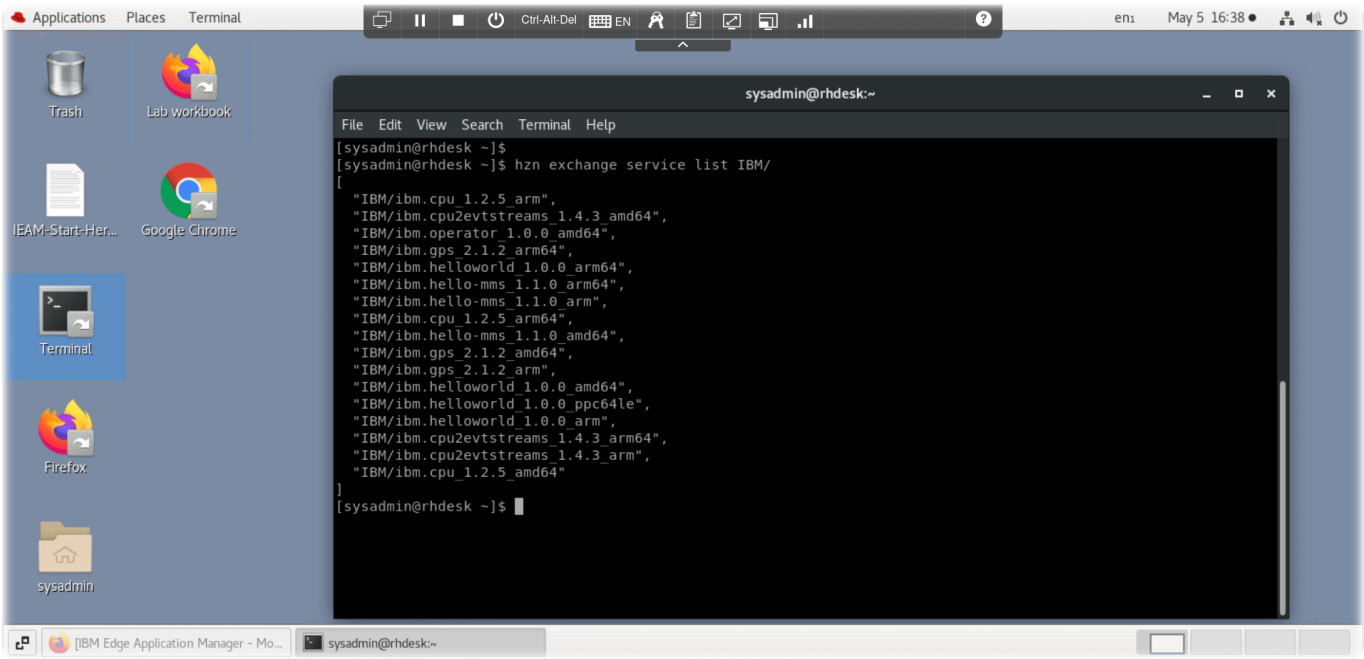
There are many horizon command line interface parameters. This section will explore some of them.

```
hzn help
```

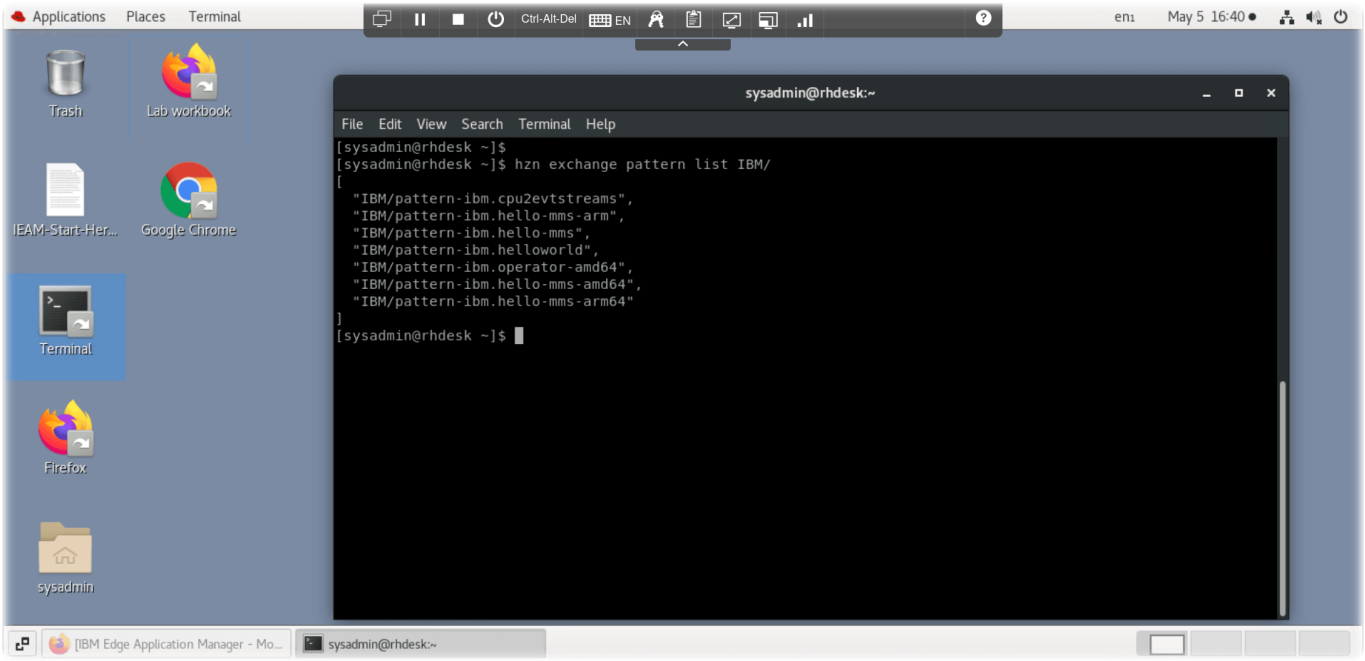
The IEAM management hub installation includes several services which have been published into the exchange automatically. The following commands will list the services, patterns, and deployment policies available in your exchange:

Note: The following commands assume you have the Horizon environment variables `HZN_ORG_ID` and `HZN_EXCHANGE_USER_AUTH` set

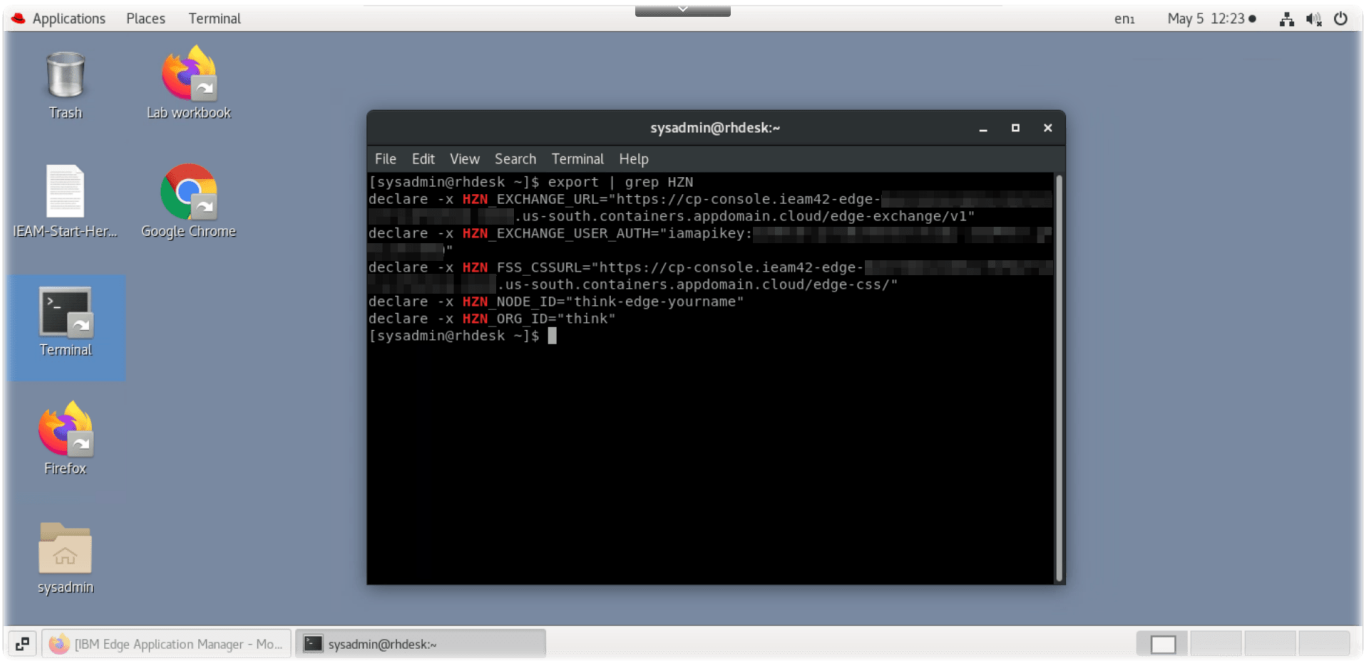
```
hzn exchange service list IBM/
```



```
hzn exchange pattern list IBM/
```

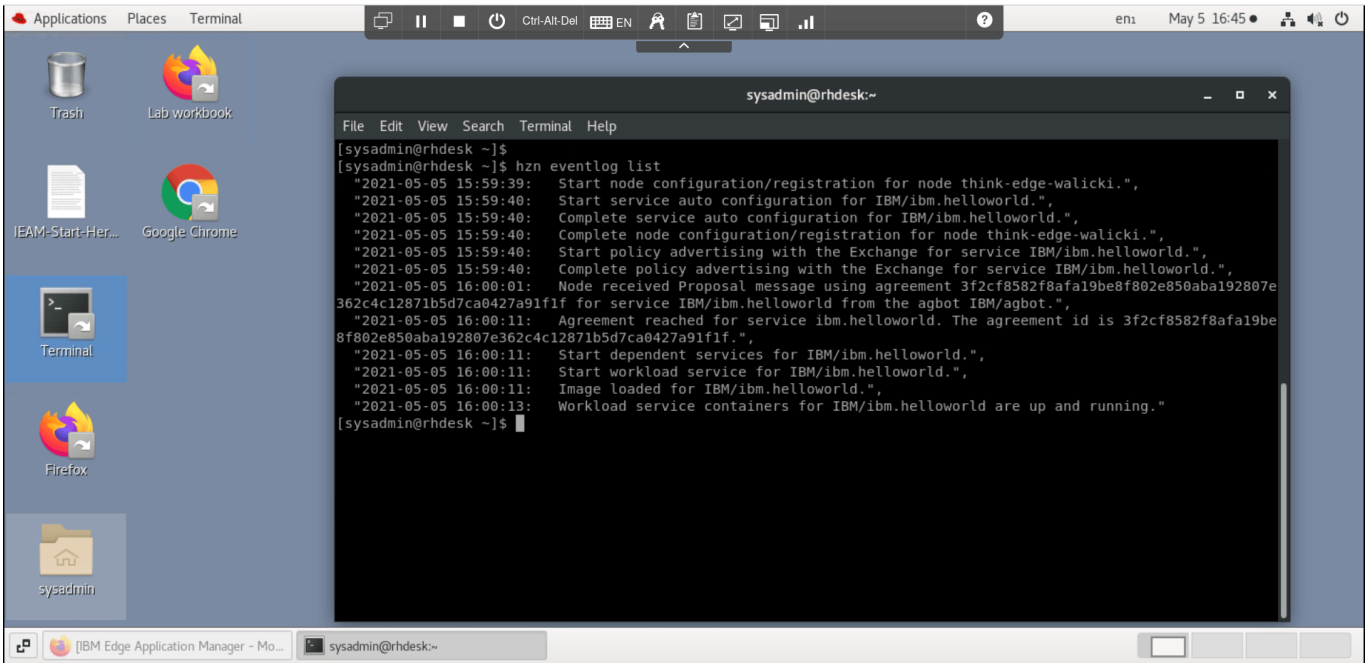



```
hzn exchange deployment listpolicy
```



- Review the event log for this node.

```
hzn eventlog list
```



Web Hello containerized workload

In this exercise, we will create a Web Hello service that runs a containerized HTTP server workload on your edge device. The web-hello-python example is an extremely simple HTTP server (written in Python) that responds on port 8000 with a hello message.

The core principle of IBM Edge Application Manager is to orchestrate containizered workloads, at scale, on edge devices. Edge developers should be familiar with creating Docker containers optimized for (CPU/Memory/Power) constrained devices. This example builds a small container, pushes it to your Docker Hub registry and creates a service and pattern in the IEAM exchange hub.

The source / instructions to build the container are posted in the web-hello-python github repository, but are copied here for the lab exercise.

- Open a Terminal window in your virtual machine
- Clone the github repository:

```
cd
git clone https://github.com/TheMosquito/web-hello-python
cd web-hello-python
```

- Log into your Docker Hub account so the container can be hosted for edge deployment

```
docker login
```

- Edit the variables at the top of the Makefile as desired. If you plan to push it to a Docker registry, make sure you give your docker ID. You may also want to create unique names for your service and pattern (necessary for this lab in the multi-tenant IEAM instance that is shared with other lab participants and you are all publishing this service).
    - gedit / vi / nano editors are available
    - Change the following `Makefile` lines:

```
DOCKERHUB_ID:=<your docker registry account>
SERVICE_NAME:="web-hello-python-<yourname>"
SERVICE_VERSION:="1.0.0"
PATTERN_NAME:="pattern-web-hello-python-<yourname>"
```

- Create a hzn cryptographic signing keys. All edge services and deployment patterns must be cryptographically signed. If you do not already have an appropriate asymmetric cryptographic signing key pair, you must create a key pair that you can use for signing. If you need to create a new key pair, run the following command:

```
hzn key create **yourcompany** **youremail**
```

This command creates the following two files:

```
~/.hzn/keys/service.private.key
~/.hzn/keys/service.public.pem
```

- Build the container, push the container to the Docker registry, publish the service and publish the pattern

```
make build
make push
make publish-service
make publish-pattern
```

- Before we can register this containerized workload to run on your specific Horizon edge node, you will need to unregister the prior IEAM pattern. Remember, only one deployment pattern can be registered on an edge node.

```
hzn unregister -v
make register-pattern
watch hzn agreement list
...
make test
```

- Test your containerize workload by opening a browser session to http://localhost:8000
- Test the container on the command line

```
curl -sS http://localhost:8000

<!DOCTYPE html>

<html>
<head>

<meta charset="utf-8">

<title>WebHello</title>

</head>
<body>
Hello, "172.17.0.1".
```

```
</body>
</html>
```

SpeedTest to MQTT containerized workload

In this exercise, we will create a Docker containerized workload that can be deployed to an edge node running the Open Horizon agent. The workload can be configured as a managed service and pattern in IBM Edge Application Manager. It runs Speedtest periodically and sends the download bandwidth results over MQTT to Watson IoT Platform Quickstart and plots the results in a chart. This example builds a small container, pushes it to your Docker Hub registry and creates a service and pattern in the IEAM exchange hub.

The source / instructions to build the container are posted in the ieam-speedtest-mqtt github repository, but are copied here for the lab exercise.

- Open a Terminal window in your virtual machine
- Clone the github repository:

```
cd
git clone https://github.com/johnwalicki/ieam-speedtest-mqtt
cd ieam-speedtest-mqtt
```

- Log into your Docker Hub account so the container can be hosted for edge deployment

```
docker login
```

- Edit the variables at the top of the Makefile as desired. If you plan to push it to a Docker registry, make sure you give your docker ID. You may also want to create unique names for your service and pattern (necessary for this lab in the multi-tenant IEAM instance that is shared with other lab participants and you are all publishing this service).
    - gedit / vi / nano editors are available
    - Change the following `Makefile` lines:

```
DOCKERHUB_ID:=<your docker registry account>
SERVICE_NAME:="speedtest-mqtt-example-<yourname>"
SERVICE_VERSION:="1.0.0"
PATTERN_NAME:="pattern-speedtest-mqtt-example-<yourname>"
```

- The prior section created hzn cryptographic signing keys. If you need to create a new key pair, run the following command:

```
hzn key create **yourcompany** **youremail**
```

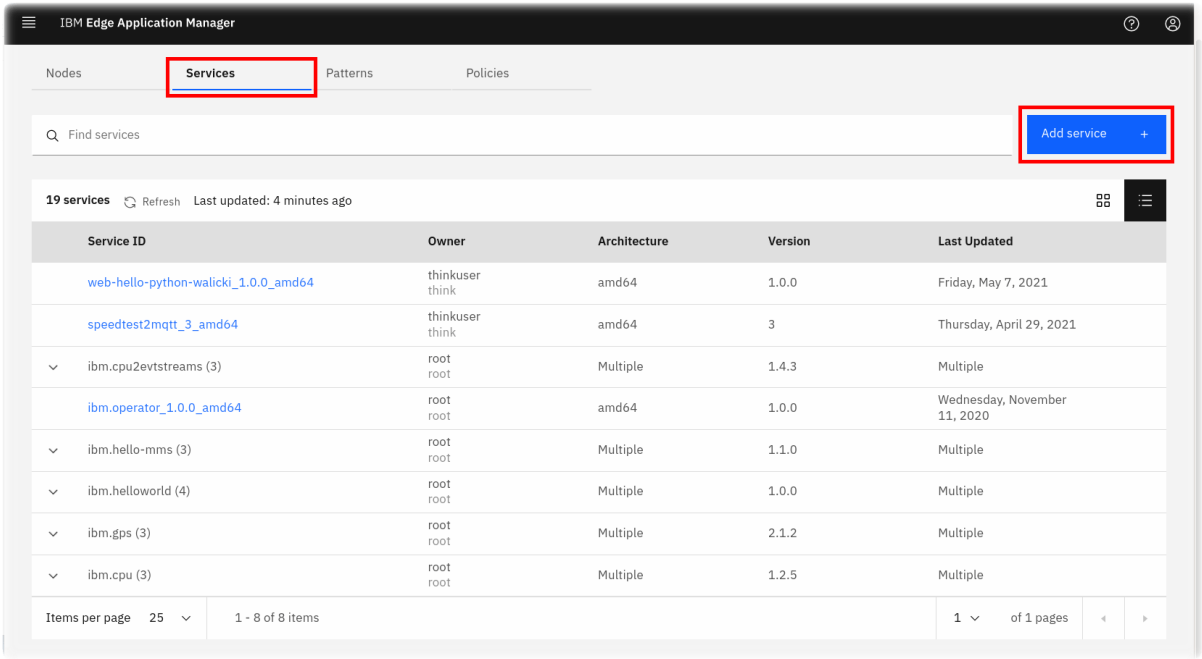
This command creates the following two files:

```
~/.hzn/keys/service.private.key
~/.hzn/keys/service.public.pem
```

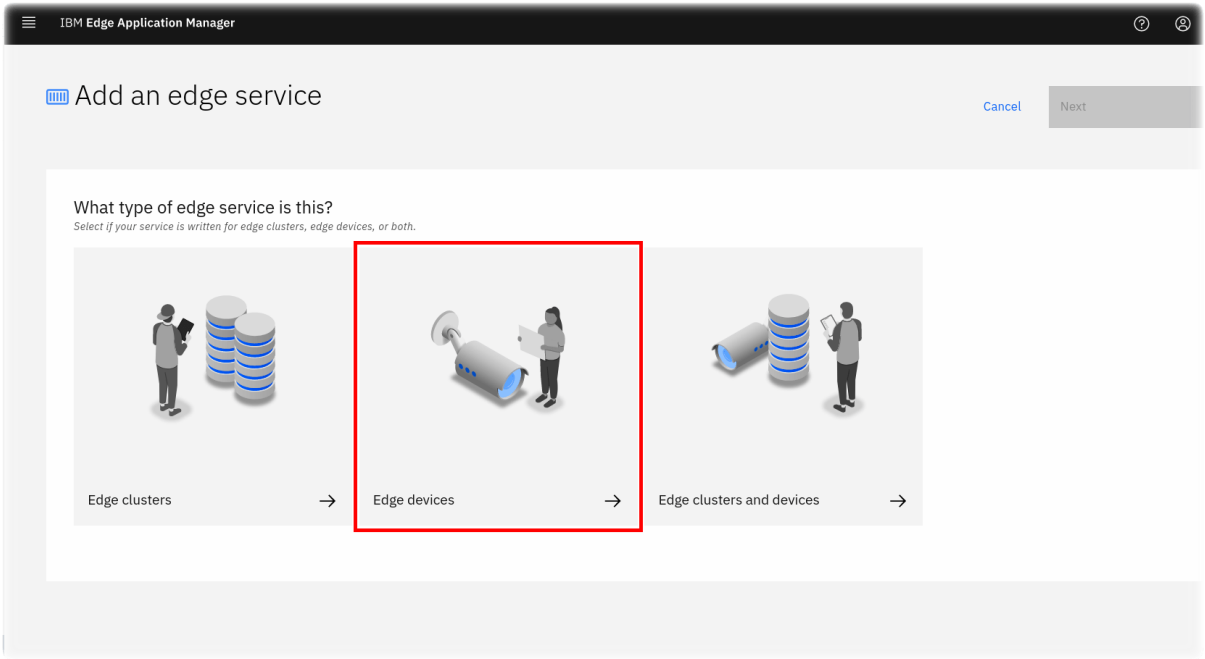
- Build the container, push the container to the Docker registry, publish the service and publish the pattern

```
make build
make push
```

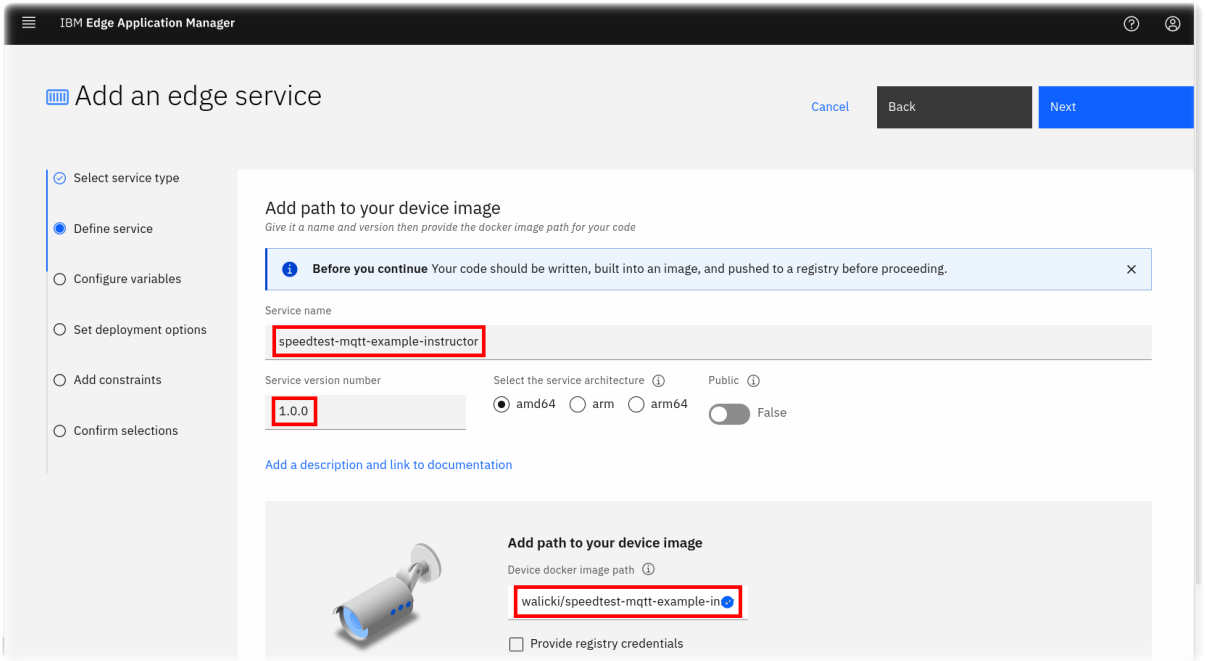
- Instead of publishing the service and pattern from the `hzn` command line interface using `make publish-service` and `make publish-pattern`, in this excerise, open the IEAM web console browser page again from the bookmark bar.
    - Select the **Services** tab and click on the **Add service** button.



- Select the **Edge devices** tile.

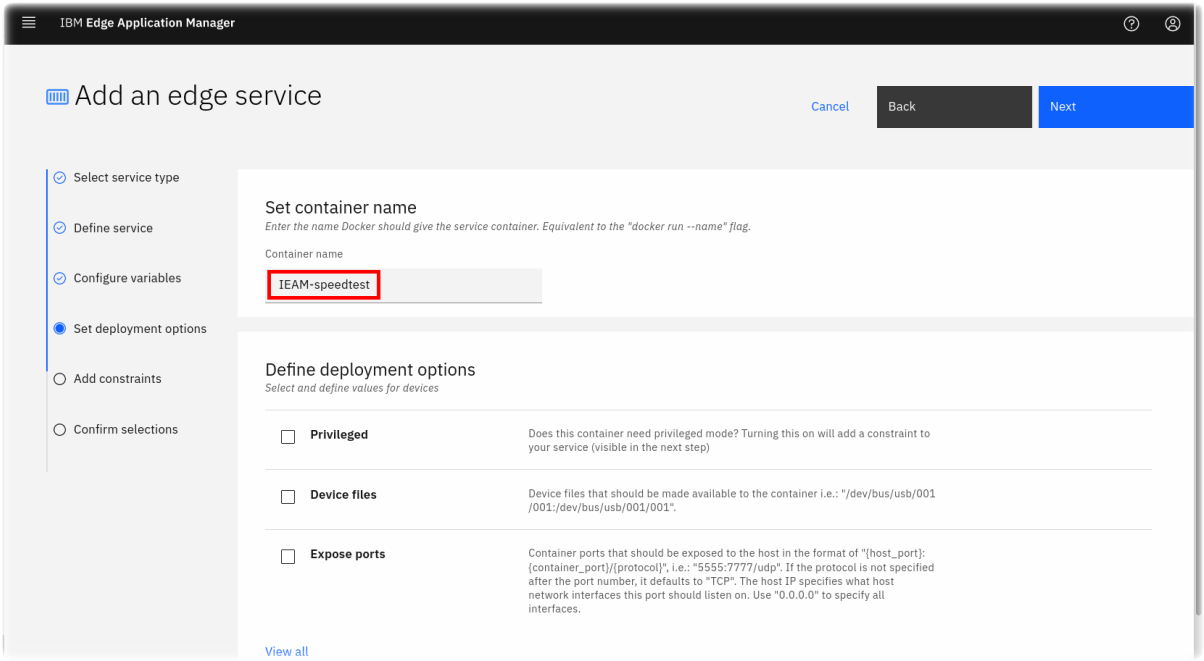


- Give your service a name, eg `speedtest-mqtt-example-<yourname>`, a Service version number, and a path to the Docker container that was pushed, eg `walicki/speedtest-mqtt-example-instructor:1.0.0`

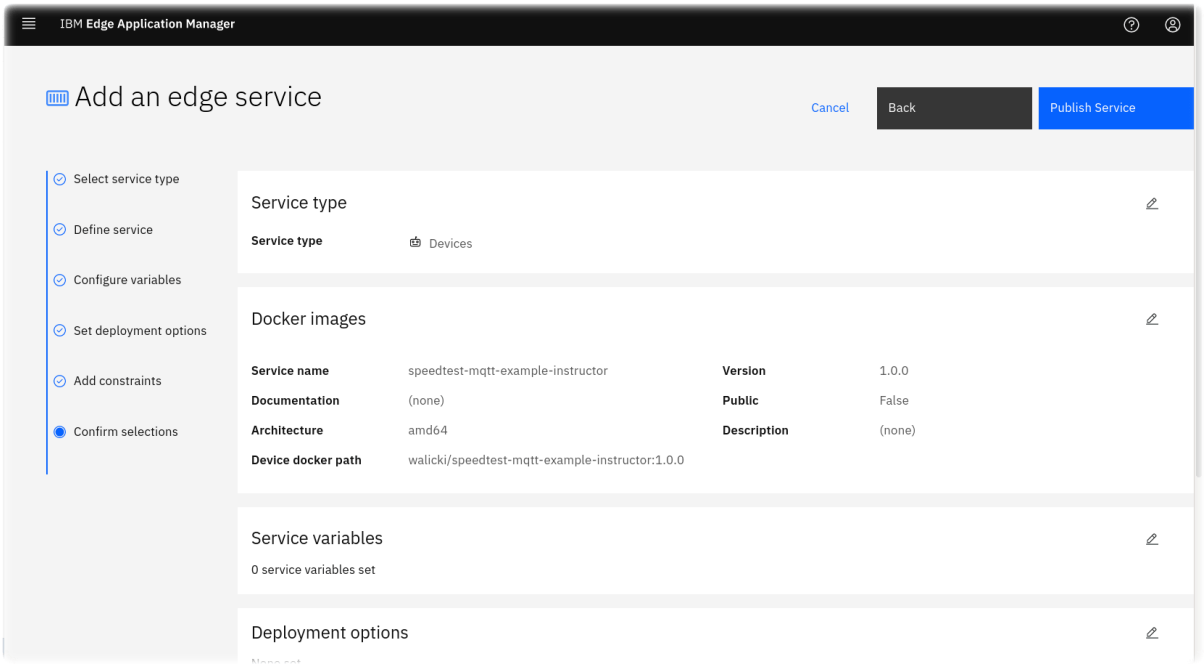


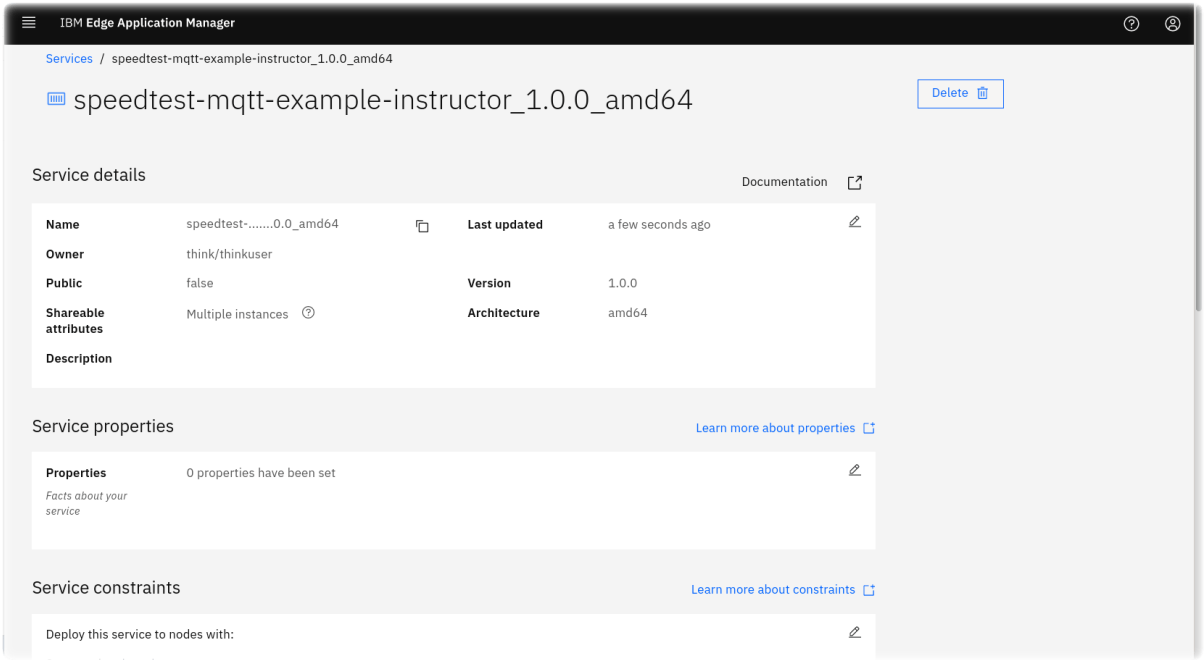
- Press the **Next** button to skip the service variables panel.

- Name your container eg `IEAM-speedtest` and press the **Next** button.

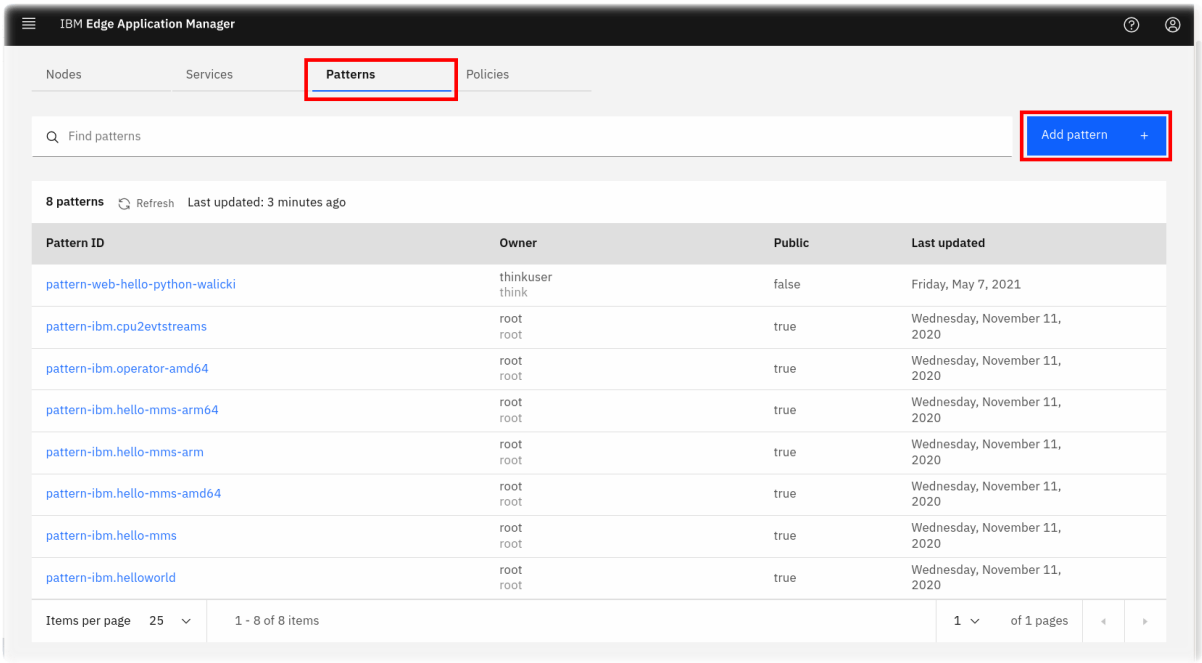


- Press the **Next** button to skip adding constraints to your service.
- Press the **Publish Service** button to publish your service.

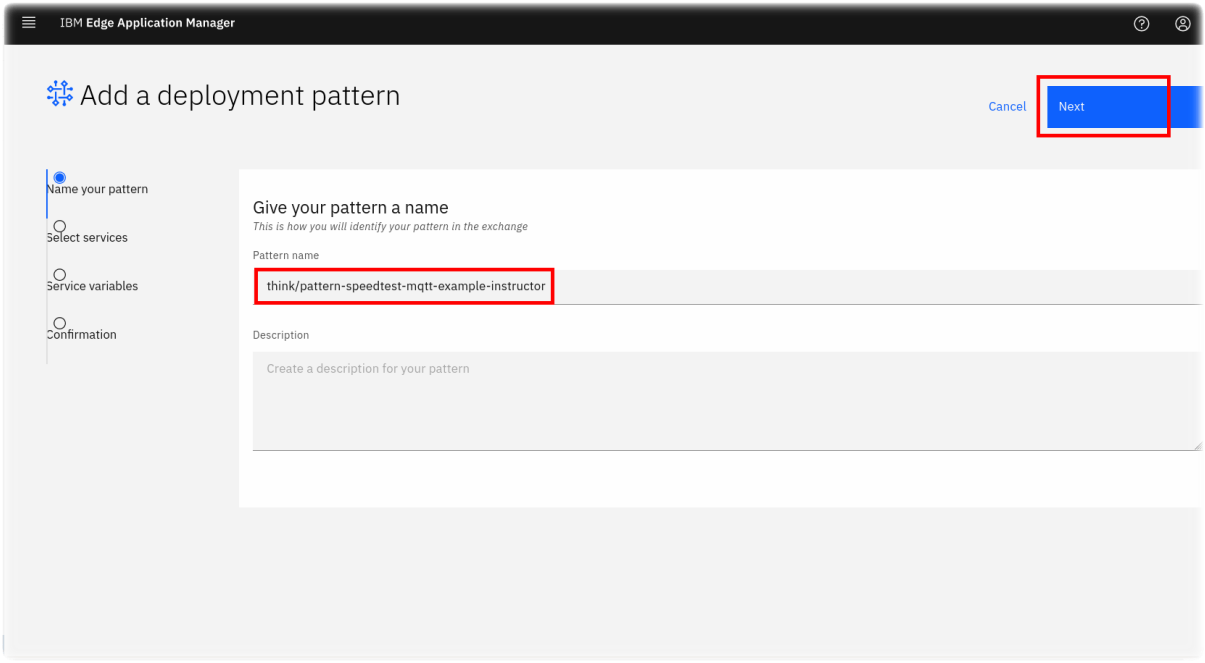




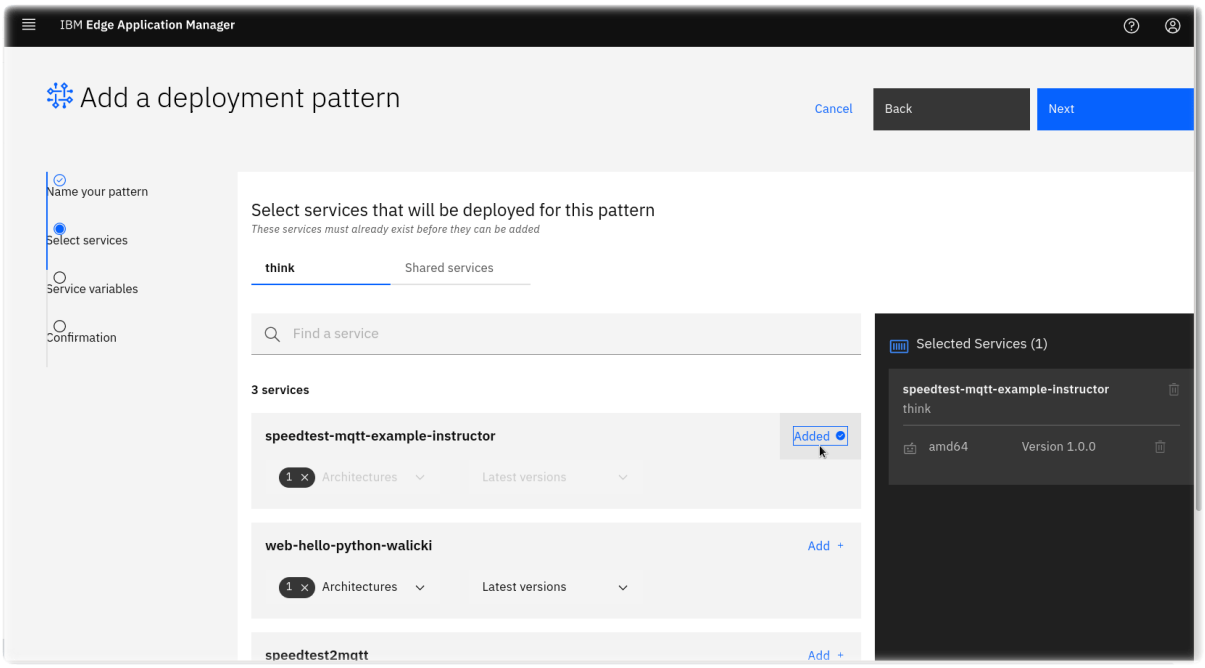
- Now that the Service is defined, add a pattern
  - Select the **Patterns** tab and click on the **Add pattern** button.



  - Give your pattern a name, eg `think/pattern-speedtest-mqtt-example-instructor` and click the **Next** button.

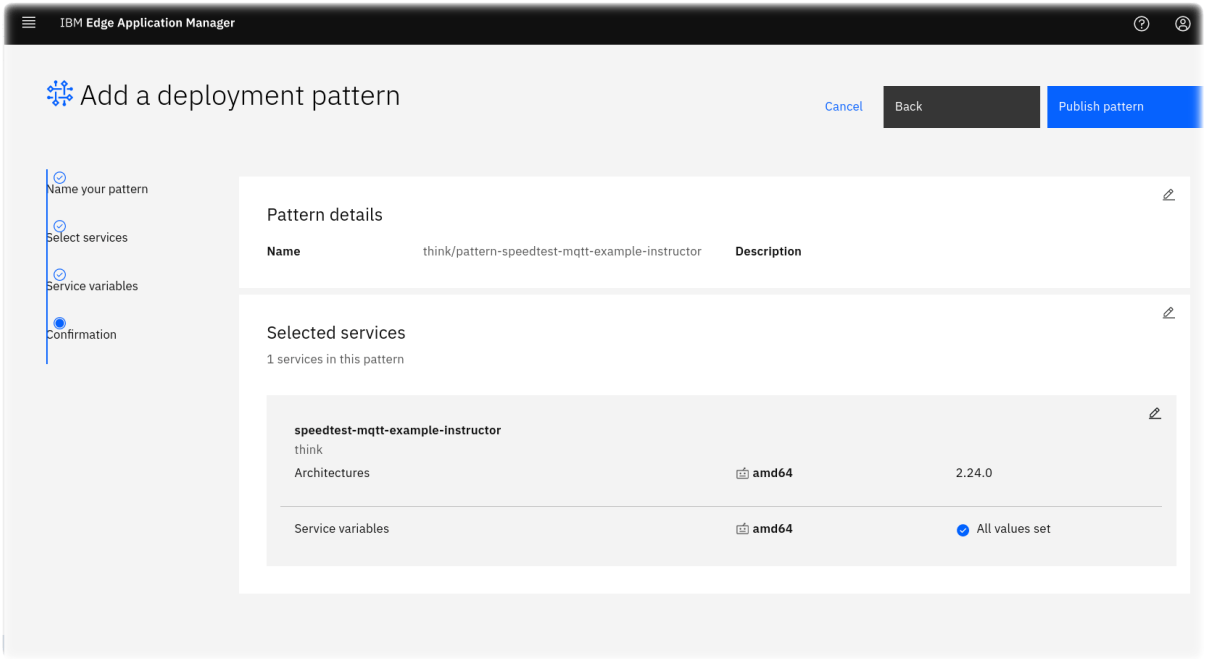


- Search/Find the `speedtest-mqtt-example-<yourname>` service, press the **Add +** button and the **Next** button.



- Press the **Next** button to skip setting service variables for the pattern.

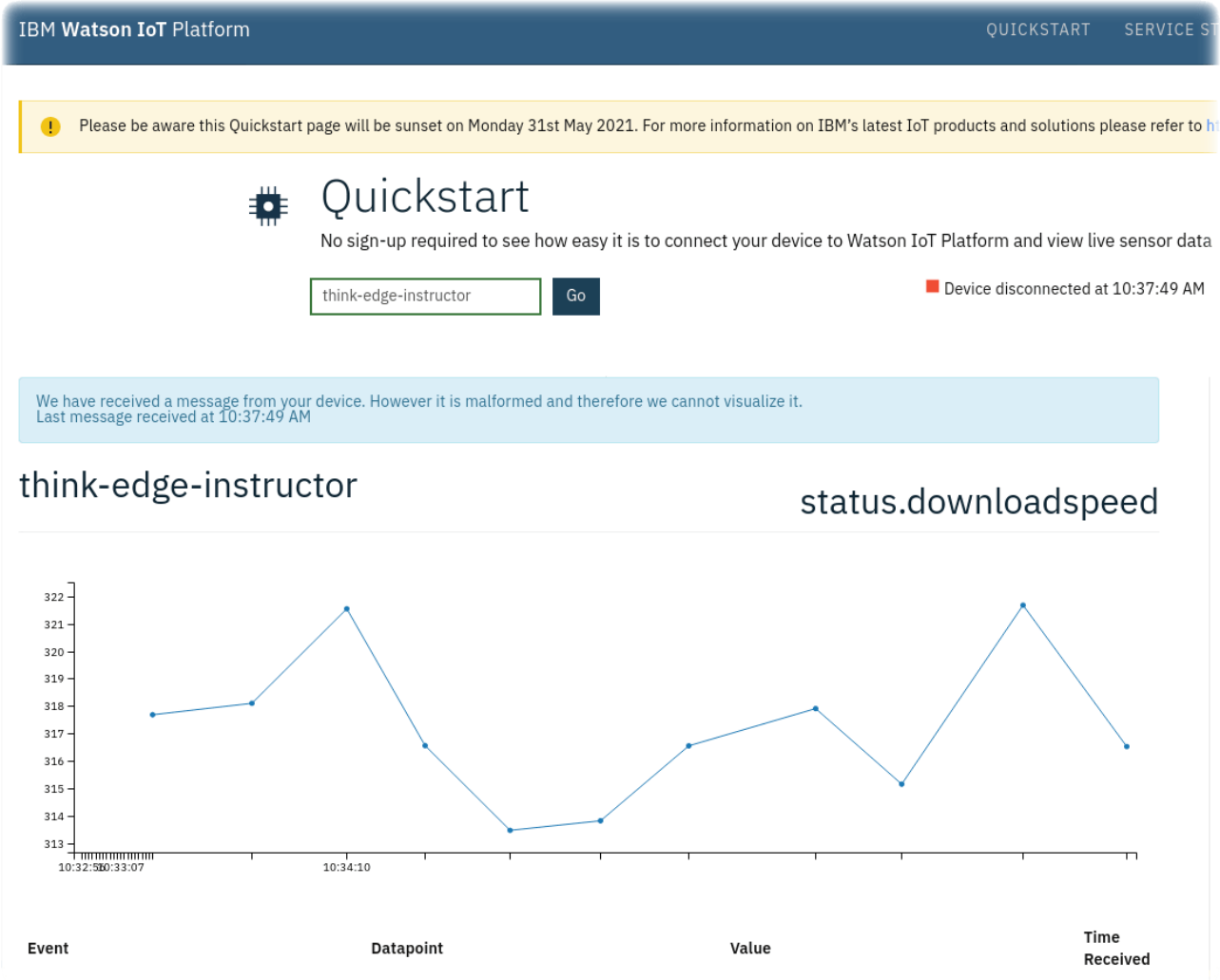
- Press the **Publish pattern** button on the confirmation panel.



- Before we can register this containerized workload to run on your specific Horizon edge node, you will need to unregister the prior IEAM pattern. Remember, only one deployment pattern can be registered on an edge node.

```
hzn unregister -v
make register-pattern
watch hzn agreement list
...
make test
```

- Observe the Speedtest bandwidth measured by your containerize workload by visiting Watson IoT Platform Quickstart. Enter your unique HZN_DEVICE_ID environment variable to something unique which will allow you to observe your "edge" device in the QuickStart chart page eg `think-edge-<yourname>`, then press the **Next** button.


IBM Watson IoT Platform
QUICKSTART
SERVICE ST
Please be aware this Quickstart page will be sunset on Monday 31st May 2021. For more information on IBM's latest IoT products and solutions please refer to ht
Quickstart
No sign-up required to see how easy it is to connect your device to Watson IoT Platform and view live sensor data
think-edge-instructor
Go
Device disconnected at 10:37:49 AM
We have received a message from your device. However it is malformed and therefore we cannot visualize it.
Last message received at 10:37:49 AM
think-edge-instructor
status.downloadspeed
322
321
320
319
318
317
316
315
314
313
10:33:07
10:34:10
Event
Datapoint
Value
Time Received

Additional Resources

- https://www.lfedge.org/projects/openhorizon/
- https://github.com/open-horizon/
- https://www.ibm.com/docs/en/edge-computing/4.2
- https://www.ibm.com/docs/en/edge-computing/4.2?topic=agent-automated-installation-registration
- https://github.com/open-horizon/examples

Author

- John Walicki

Enjoy! Give me feedback if you have suggestions on how to improve this lab workbook.

License