Lab: Module 5 - Container Orchestrators in Azure

> Duration: 120 minutes

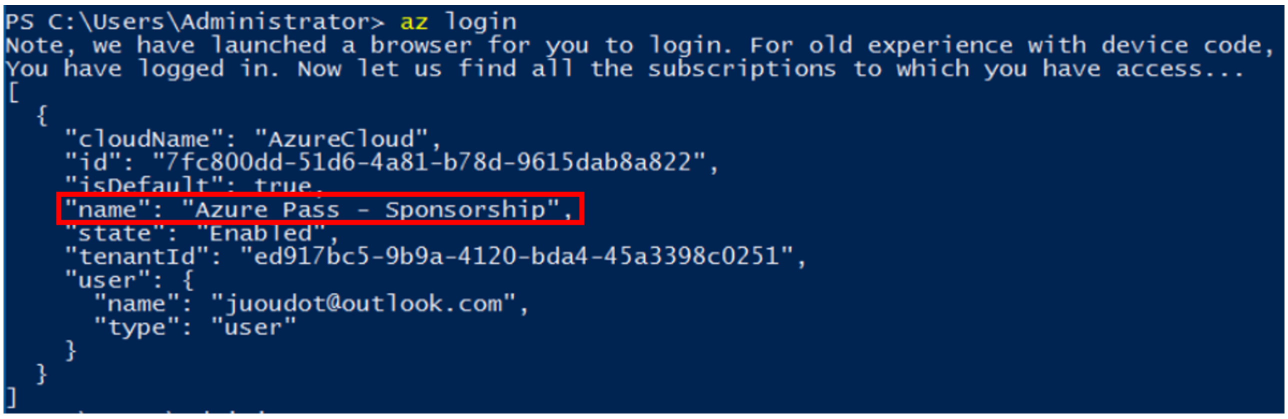
Table of Contents

Exercise 1: Create and Scale Azure Kubernetes Service (AKS) Cluster

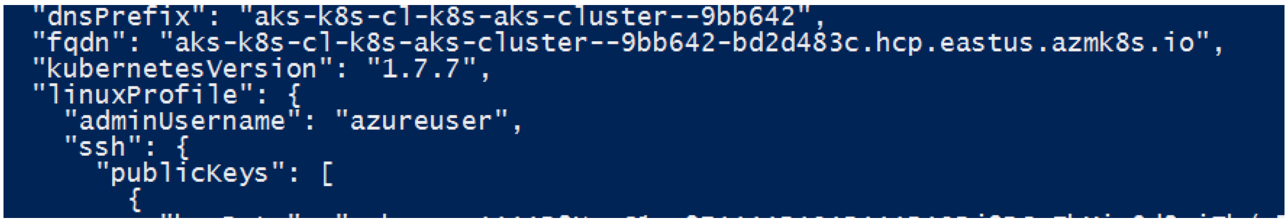
In this exercise, you will create a Kubernetes cluster on Azure Kubernetes Service and then deploy a **NGINX** container image to that cluster. The container image will be fetched from Docker Hub public registry. You will also learn how the application can be scaled using command line or dashboard.

Tasks

1. Create Linux Kubernetes Cluster on Azure Kubernetes Services

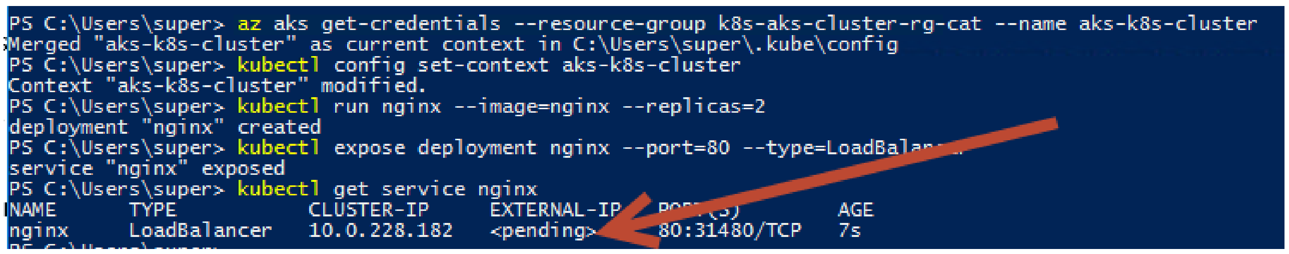
1. Sign onto your LOD **Windows VM** and open Chrome on the VM. Navigate to portal.azure.com and sign in with your **Azure Pass** account.
2. On Windows server, open PowerShell. Do not do it from Powershell ISE as the IDE won't work as expected with **az** Command Line Interface.
3. Login to Azure (the command line will automatically open your browser to login. Once you are logged in, then close the browser and go back to command line).
   ```
   az login
   ```
4. If you have multiple sunscriptions attached to your account, you will see a list of subscriptions once you are logged in. To select the correct subscription, put the name of the subscription in quotes. You can find the **name** field in the output of the `az login` executed above.
   
   ```
   PS C:\Users\Administrator> az login
   Note, we have launched a browser for you to login. For old experience with device code,
   You have logged in. Now let us find all the subscriptions to which you have access...
   [
     {
       "cloudName": "AzureCloud",
       "id": "7fc800dd-51d6-4a81-b78d-9615dab8a822",
       "isDefault": true,
       "name": "Azure Pass - Sponsorship",
       "state": "Enabled",
       "tenantId": "ed917bc5-9b9a-4120-bda4-45a3398c0251",
       "user": {
         "name": "juoudot@outlook.com",
         "type": "user"
       }
     }
   ]
   ```
   ```
   az account set --subscription "<subscription_name>"
   ```
5. Create a resource group named (replace initials with your initials so the resource group name is unique) **k8s-win-cluster-rg-INITALS**
   ```
   az group create --name=k8s-aks-cluster-rg-INITALS --location=eastus
   ```

6. Run the following command to deploy Kubernetes cluster with AKS (replace initials with your initials).
   `az aks create --resource-group k8s-aks-cluster-rg-INITALS --name aks-k8s-cluster --disable-rbac --node-count 1 --node-vm-size "Standard_A1" --generate-ssh-keys`
   *Note that this command will generate public/private ssh keys in **c:\users\Administrator.ssh** folder.*
   It may take approximately 10 minutes for the cluster to provision successfully in the US. If you are located in Europe, the provisioning may take about 30 minutes. In the interest of time, you can start the second exercise while this command is running.
7. After the cluster is provisioned successfully you will be shown JSON output describing the AKS cluster.
8. Locate and copy the value of **fdqn** attribute from the JSON output. Note that your **fdqn** value may differ from the output shown below.



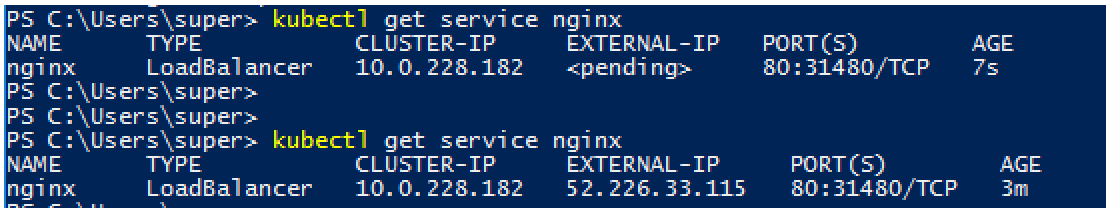
```
  "dnsPrefix": "aks-k8s-cl-k8s-aks-cluster--9bb642",
  "fqdn": "aks-k8s-cl-k8s-aks-cluster--9bb642-bd2d483c.hcp.eastus.azmk8s.io",
  "kubernetesVersion": "1.7.7",
  "linuxProfile": {
    "adminUsername": "azureuser",
    "ssh": {
      "publicKeys": [
        {
```

9. Install kubectl tool which allows you to run commands against Kubernetes cluster. Skip this step if it is already installed on your machine (run `kubectl` to see if the command is recognized).
   Windows:
   `choco install -y kubernetes-cli` (requires choco package manager)
   Linux:
   ```
   sudo apt-get -y update && sudo apt-get -y upgrade
   curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
   chmod +x ./kubectl
   sudo mv ./kubectl /usr/local/bin/kubectl
   ```
   In Linux, due to current bug with **azure-cli** you need to revert to **2.0.23-1** version.
   ```
   sudo apt-get -y purge azure-cli
   sudo apt-get -y install azure-cli=2.0.23-1
   ```
10. Run the following command to download the Kubernetes cluster configuration to the local config file **C:\users\Administrator.kube\config**
    `az aks get-credentials --resource-group k8s-aks-cluster-rg-INITALS --name aks-k8s-cluster`
    *NOTE: Be aware that you will need the content of **.kube/config** file while creating Kubernetes service endpoint on Azure DevOps in the DevOps module.*
11. Run following command to ensure context is set to the correct cluster:
    `kubectl config set-context aks-k8s-cluster`
12. You will now test the cluster by running a **nginx** container. First create a new deployment using the following command:
    `kubectl run nginx --image=nginx --replicas=2`
13. Next, make sure that you can access the container from the external (**public IP**). To do that use the **expose** command to expose port 80 and enable the external IP (**type=LoadBalancer**):
    `kubectl expose deployment nginx --port=80 --type=LoadBalancer`
14. The above command essentially creates a service with the name **nginx**. You can view the service by running following command
    `kubectl get service nginx`



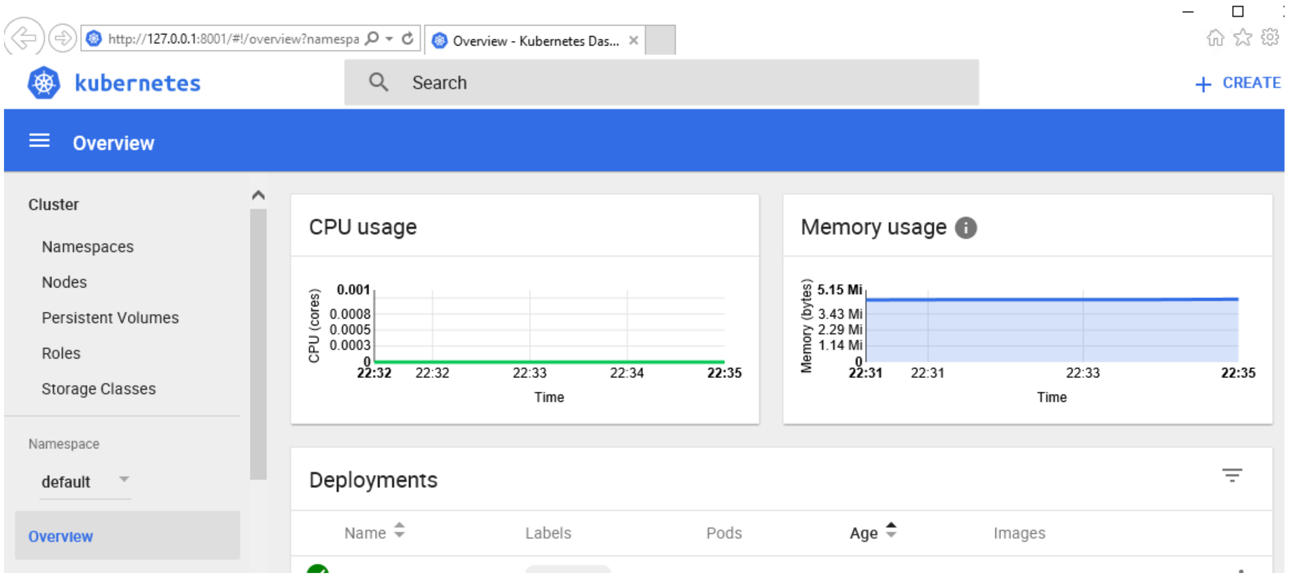
```
PS C:\Users\super> az aks get-credentials --resource-group k8s-aks-cluster-rg-cat --name aks-k8s-cluster
Merged "aks-k8s-cluster" as current context in C:\Users\super\.kube\config
PS C:\Users\super> kubectl config set-context aks-k8s-cluster
Context "aks-k8s-cluster" modified.
PS C:\Users\super> kubectl run nginx --image=nginx --replicas=2
deployment "nginx" created
PS C:\Users\super> kubectl expose deployment nginx --port=80 --type=LoadBalan
service "nginx" exposed
PS C:\Users\super> kubectl get service nginx
NAME      TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
nginx     LoadBalancer   10.0.228.182   <pending>     80:31480/TCP   7s
```

Please wait until **EXTERNAL-IP** for **nginx** service change from to a valid IP address. It may take few minutes and you may have to run the command `kubectl get service nginx` few times to probe the status of external IP assignment. When it is done, it will look like below:



```
PS C:\Users\super> kubectl get service nginx
NAME      TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
nginx     LoadBalancer   10.0.228.182   <pending>     80:31480/TCP   7s
PS C:\Users\super>
PS C:\Users\super>
PS C:\Users\super> kubectl get service nginx
NAME      TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)        AGE
nginx     LoadBalancer   10.0.228.182   52.226.33.115   80:31480/TCP   3m
```

15. You can now simply query the content hosted by nginx by using the **curl** command:
    `curl http://$( kubectl get service nginx -o=jsonpath='{.status.loadBalancer.ingress[*].ip}')`
16. Now, to access the dashboard run the following command. Please refresh the opened browser if needed after a couple of seconds.
    `az aks browse --resource-group k8s-aks-cluster-rg-INITALS --name aks-k8s-cluster`



You can always access the dashboard by browsing to the URL: http://127.0.01:8001. Although this command works with both Windows and Linux you should only run it on Windows as you have access to full browser on Windows running inside the lab virtual machine.

17. Stop running by hitting **Ctrl + C**, then **Y** to terminate the batch job.
18. Let's check total number of pods running now. You should see two pods ready with a status of running.
    `kubectl get pods`
19. You will now scale the number of pods using replicaset. You will get more details about the replica set using the following command:
    `kubectl get replicaset`
20. To scale, run the command and pass the name of deployment that was created earlier:
    `kubectl scale --replicas=3 deployment/nginx`
21. Now if you run the command to get the number of pods running after scaling, it should return three pods (earlier it was two).

```
kubectl get pods -o wide
```

22. In previous step you have successfully increased the number of pods by increasing the replica set. However, all the pods are running on a single node. You can check that by running the following command:
    ```
    kubectl get nodes
    ```
    This is because cluster was created with a single worker node. You are now going to add one more worker node to the cluster by running following command:
    ```
    az aks scale --node-count=2 --resource-group k8s-aks-cluster-rg-INITALS --name aks-k8s-cluster
    ```
    Please wait while new worker node is successfully added to the cluster. This may take few minutes to complete.
23. If you check the number of nodes again, you should see two worker nodes instead of single worker node running.
    Next time when you scale the pods using replica set, second worker node will also participate. You can also look at the maximum pod capacity of a node by running the command:
    ```
    kubectl get node NODE-NAME -o=jsonpath='{.status.capacity.pods}'
    ```
24. To scale back down to a node count of 1, run the following command:
    ```
    az aks scale --node-count=1 --resource-group k8s-aks-cluster-rg-INITALS --name aks-k8s-cluster
    ```
25. Finally, remove the deployment and service using the commands below:
    ```
    kubectl delete deployment nginx
    kubectl delete service nginx
    ```
26. Do not delete your deployment in the Azure Portal, you will need the cluster for the CI/CD portion in the next lab.

Exercise 2: Create an Azure Service Fabric Cluster

In this exercise, you will create a Windows Service Fabric cluster on Azure, and then deploy a container image to the cluster. The container image will be fetched from Docker Hub public registry.

Tasks

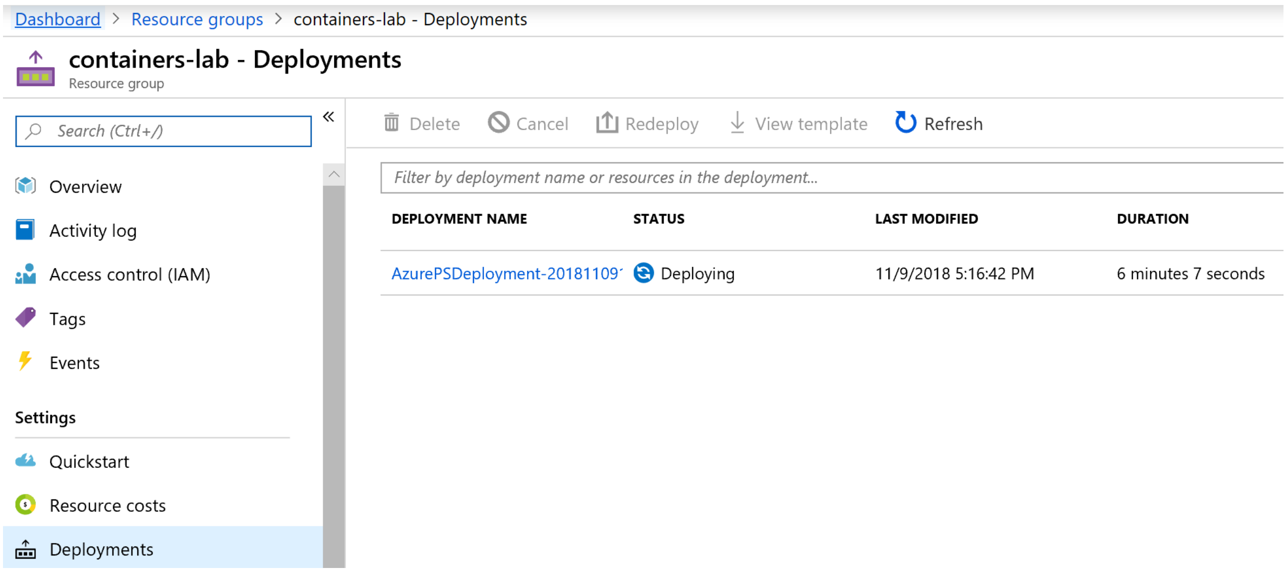
1. Create Windows Service Fabric Cluster on Azure

1. On Windows server, open PowerShell and navigate to **c:\labs\module5**.
   ```
   cd c:\labs\module5
   ```
2. Login to Azure (the command line will automatically open your browser to login. Once you are logged in, then close the browser and go back to command line).
   ```
   az login
   ```
3. If you have multiple subscriptions attached to your Microsoft account. you will see a list of subscriptions once you are logged in. Set the correct subscription, put the name of the subscription in quotes. You can find the **name** field in the output of the az login executed above.
   ```
   az account set --subscription "subscription_name"
   ```
4. Run the following command to deploy Kubernetes cluster with AKS (replace **initials** with your initials. **Note that the cluster name has to be unique, so initials might not be enough**. Instead, you might want to put your entire name).
   Make sure that initials are lower case or you will see an error saying the dns name with upper case is

invalid.

```
az sf cluster create --resource-group "containers-lab-initials" --location
"eastus"  --certificate-output-folder ".\" --vault-name "containers-lab-kv-initials
" --vault-resource-group "containers-lab-initials " --cluster-size 3 --vm-os
"WindowsServer2016DatacenterWithContainers"  --vm-password "P@ssw0rd123!" --vm-
user-name "myadmin" --certificate-subject-name "containers-lab-initials"
```

*Note that the deployment can last 20 minutes. You can follow the deployment by looking in portal.azure.com in the resource group created, under **Deployments** section.*


Dashboard > Resource groups > containers-lab - Deployments

containers-lab - Deployments
Resource group

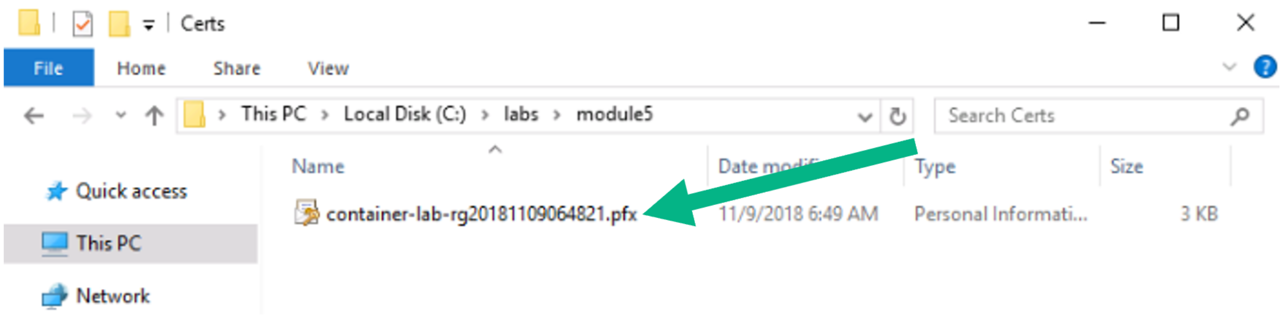
Filter by deployment name or resources in the deployment...

DEPLOYMENT NAME	STATUS	LAST MODIFIED	DURATION
AzurePSDeployment-20181109'	Deploying	11/9/2018 5:16:42 PM	6 minutes 7 seconds

2. Connect to the Service Fabric Explorer

By default, the Service Fabric cluster is set up with certificate-based security. Alternatively, Azure Active Directory can be used to protect the cluster. In our case, we need to install the required certificate to manage the cluster.

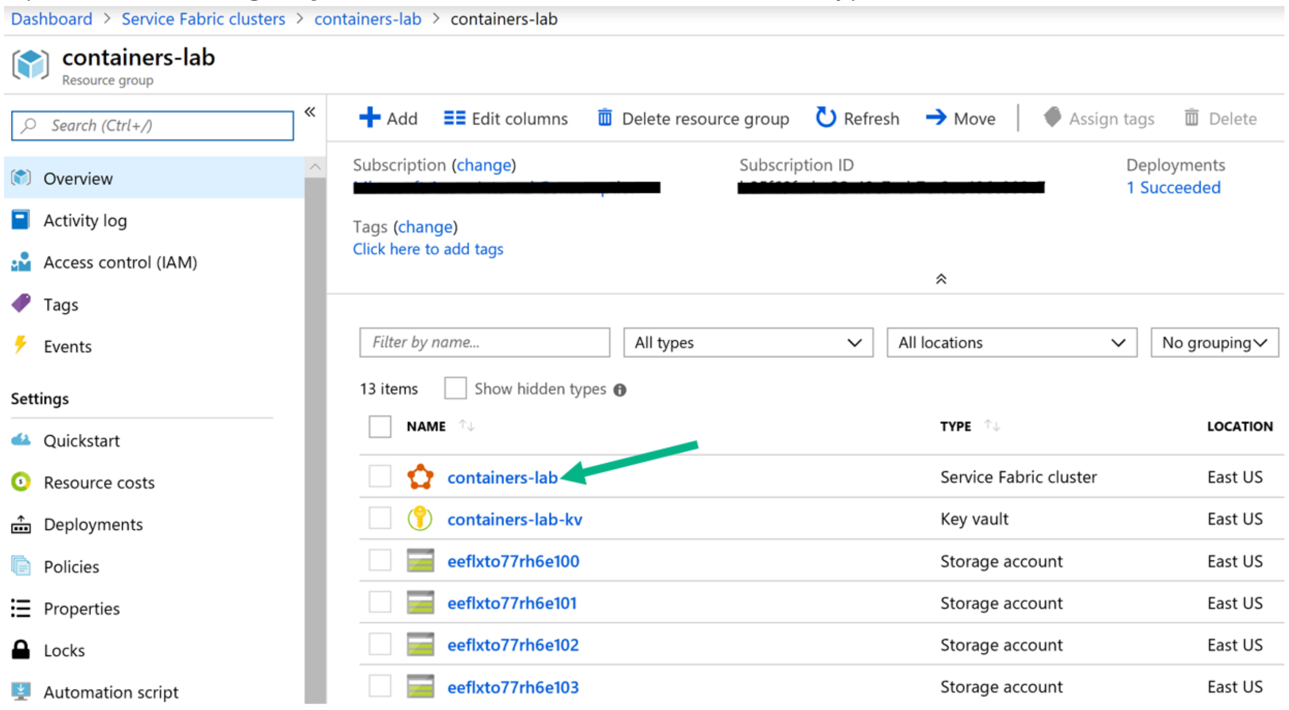
1. Double-click on the **PFX** certificate created in by the command in **C:\labs\module5\** and install it in the current user store location. Just enter an empty password. The rest of the settings can stay unchanged.


This PC > Local Disk (C:) > labs > module5

Name	Date modified	Type	Size
container-lab-rg20181109064821.pfx	11/9/2018 6:49 AM	Personal Informati...	3 KB

2. Open a browser and navigate to portal.azure.com.

3. Open the **resource group** created and click on the **Service Fabric** type of resource.



Dashboard > Service Fabric clusters > containers-lab > containers-lab

containers-lab
Resource group

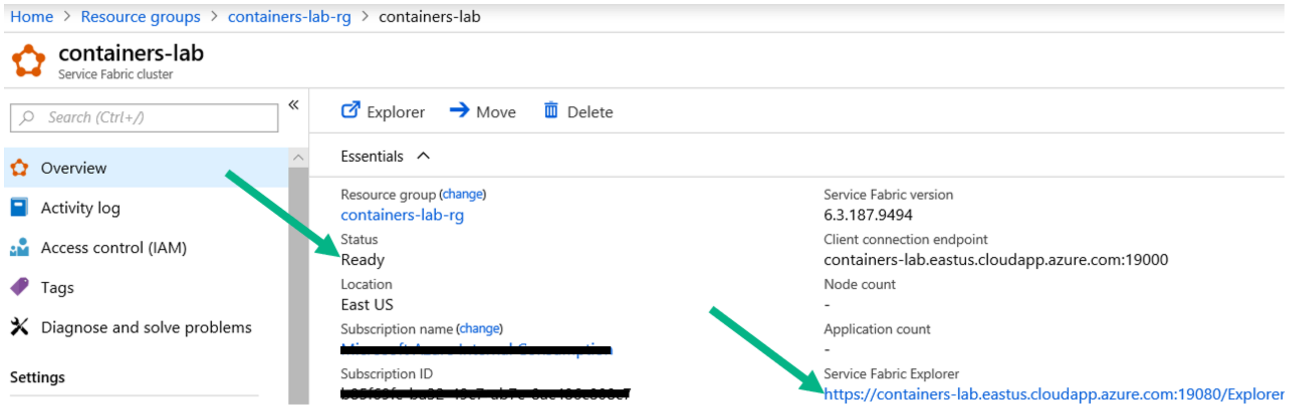
Subscription (change) [redacted]
Subscription ID [redacted]
Deployments 1 Succeeded
Tags (change) Click here to add tags

Filter by name... | All types | All locations | No grouping

13 items Show hidden types

NAME	TYPE	LOCATION
containers-lab	Service Fabric cluster	East US
containers-lab-kv	Key vault	East US
eeflxto77rh6e100	Storage account	East US
eeflxto77rh6e101	Storage account	East US
eeflxto77rh6e102	Storage account	East US
eeflxto77rh6e103	Storage account	East US

4. Once the Service Fabric cluster is in **Ready** state, click on the **Service Fabric Explorer** link. It will open the explorer. Then select the certificate you just installed to be authorized to access the dashboard.

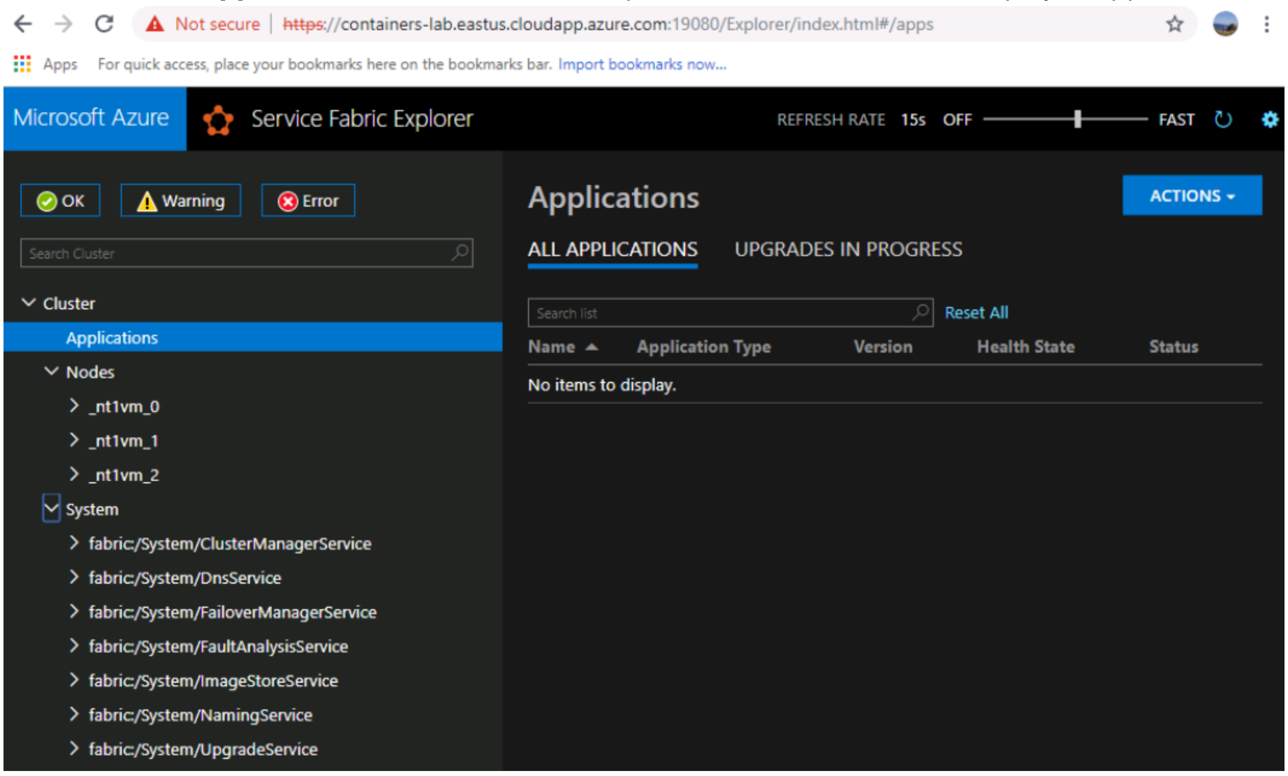


Home > Resource groups > containers-lab-rg > containers-lab

containers-lab
Service Fabric cluster

Resource group (change): containers-lab-rg
Status: Ready
Location: East US
Subscription name (change): [redacted]
Subscription ID: [redacted]
Service Fabric version: 6.3.187.9494
Client connection endpoint: containers-lab.eastus.cloudapp.azure.com:19000
Node count: -
Application count: -
Service Fabric Explorer: https://containers-lab.eastus.cloudapp.azure.com:19080/Explorer

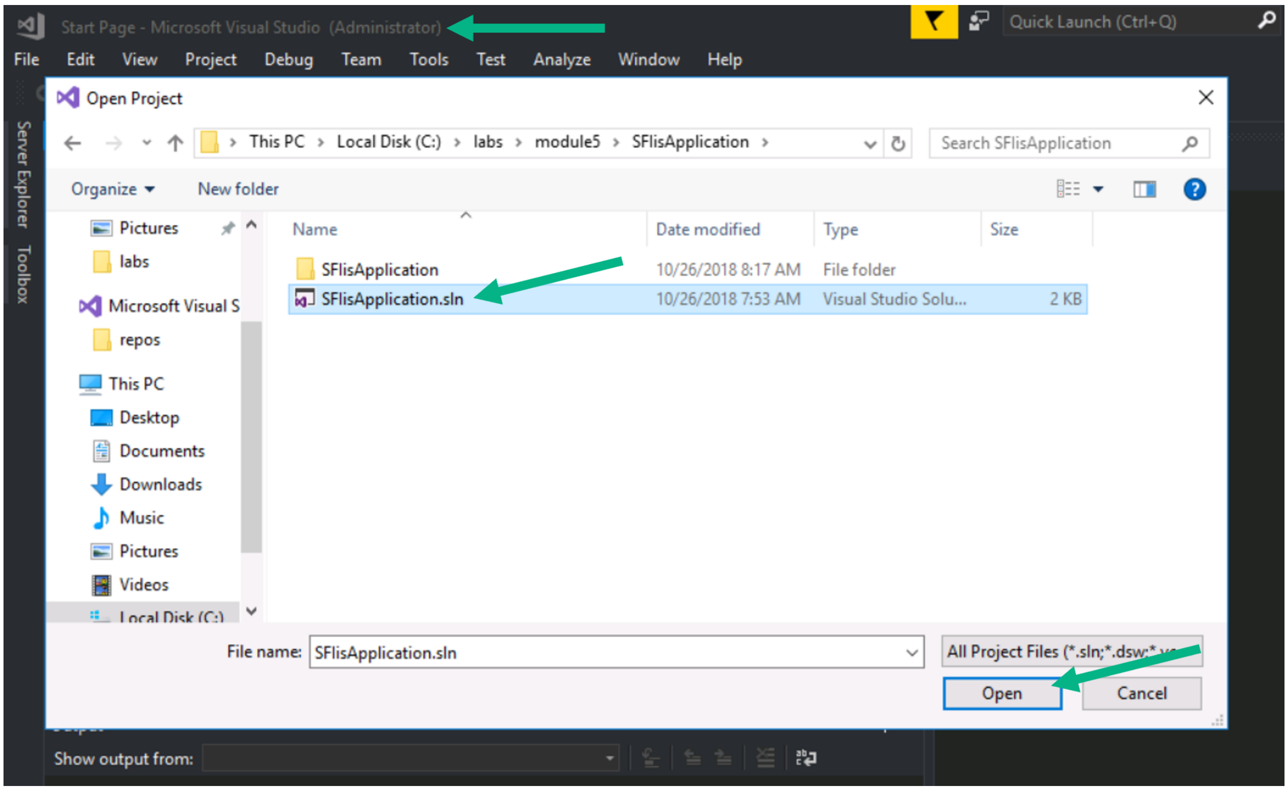
Below is the dashboard that the previous click opens. On the left-hand side, you can see the cluster **Nodes**, as well as the **System** services responsible for the orchestration and management operations in the cluster. The **Applications** section cannot be expanded because there is no deployed application.


Not secure | https://containers-lab.eastus.cloudapp.azure.com:19080/Explorer/index.html#/apps
Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Microsoft Azure
Service Fabric Explorer
REFRESH RATE 15s OFF FAST
OK
Warning
Error
Search Cluster
Cluster
Applications
Nodes
_nt1vm_0
_nt1vm_1
_nt1vm_2
System
fabric:/System/ClusterManagerService
fabric:/System/DnsService
fabric:/System/FailoverManagerService
fabric:/System/FaultAnalysisService
fabric:/System/ImageStoreService
fabric:/System/NamingService
fabric:/System/UpgradeService
Applications
ACTIONS
ALL APPLICATIONS
UPGRADES IN PROGRESS
Search list
Reset All

Name	Application Type	Version	Health State	Status
No items to display.				

3. Deploy an Application to Azure Service Fabric Cluster

1. Now that your Windows Service Fabric cluster is ready, you will deploy a containerized application located in Docker Hub. Open Visual Studio as **Admin** (right-click on Visual Studio and select **Run as Administrator**) and open the solution located in **C:\labs\module5\SFIisApplication**

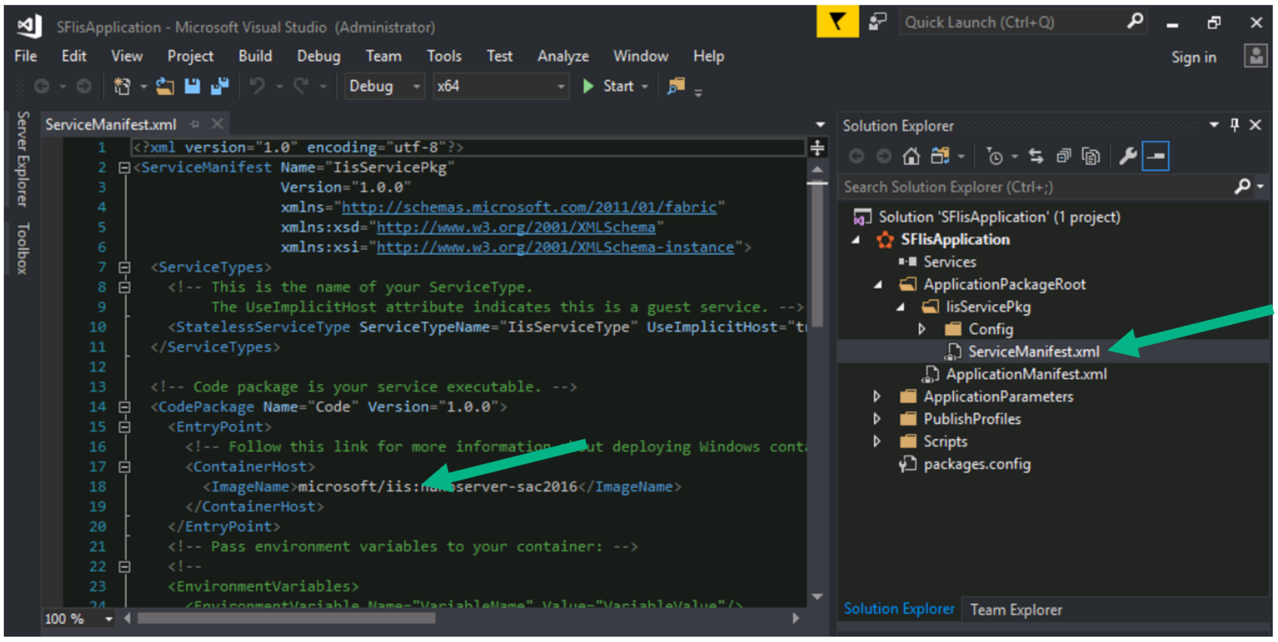


Name	Date modified	Type	Size
SFIisApplication	10/26/2018 8:17 AM	File folder	
SFIisApplication.sln	10/26/2018 7:53 AM	Visual Studio Solu...	2 KB

File name: SFIisApplication.sln

2. This project is an application, composed of a single service packaged in a docker Container. Open the **ServiceManifest.xml** under **ApplicationPackageRoot** - **IisServicePkg**. Note the image that will be

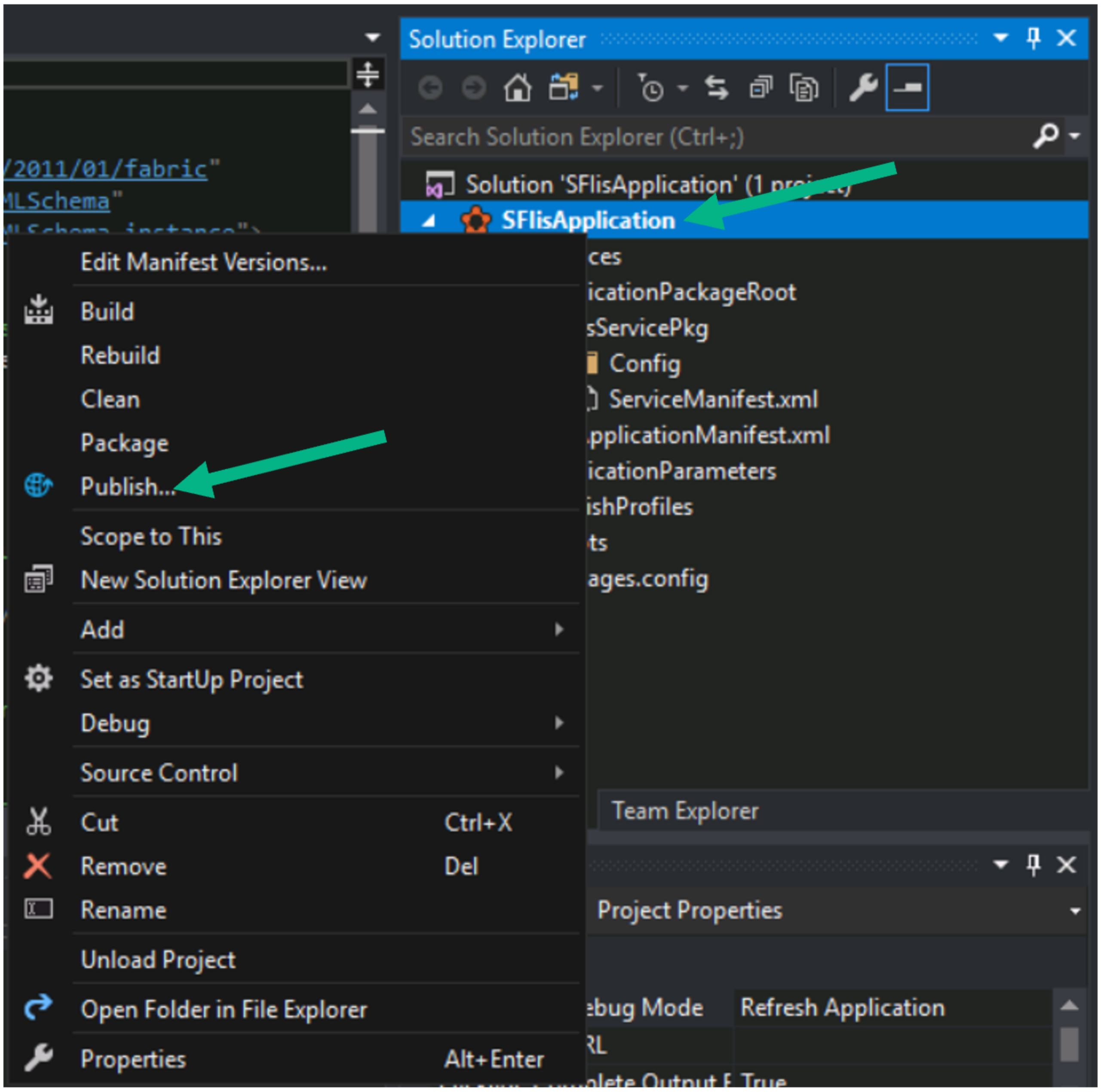
deployed as part of this application: **microsoft/iis**.



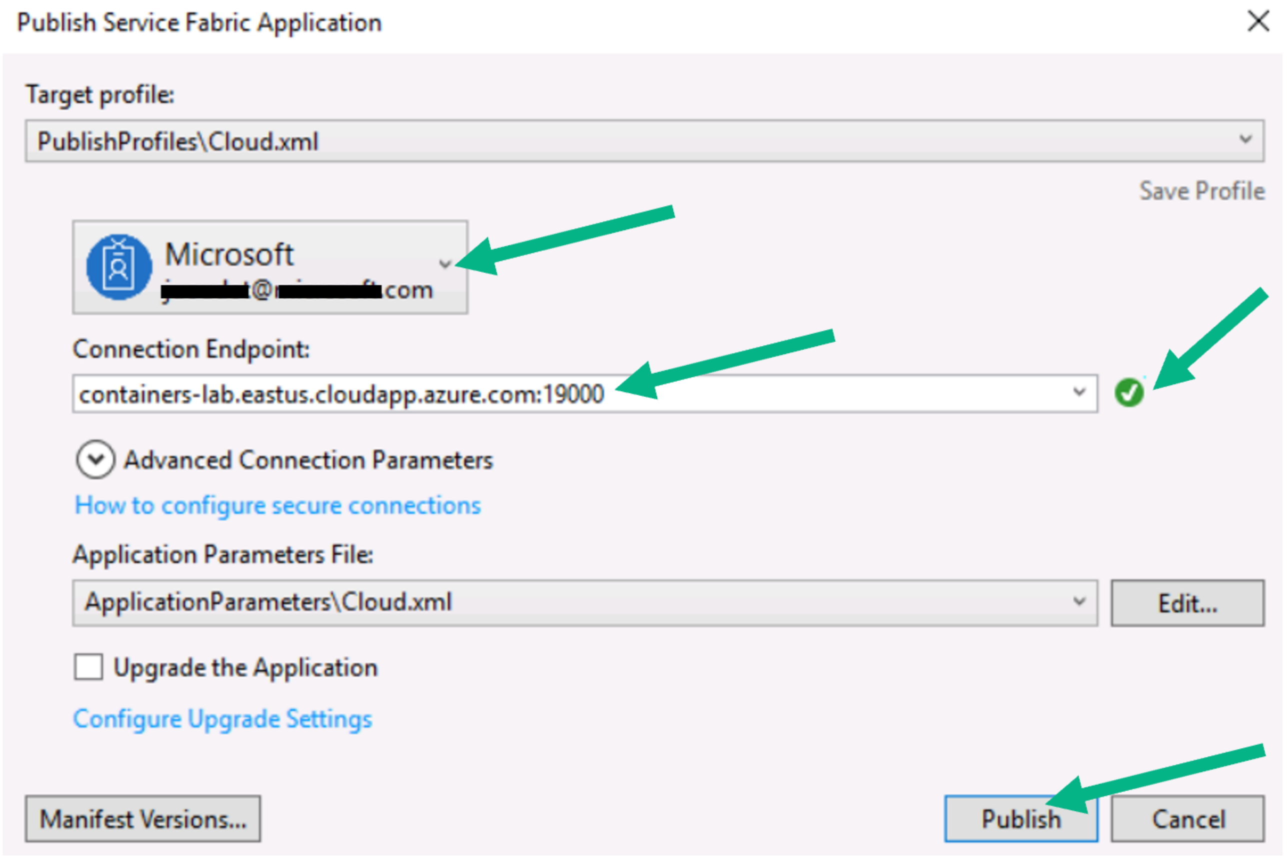
```xml
<?xml version="1.0" encoding="utf-8"?>
<ServiceManifest Name="IisServicePkg"
                 Version="1.0.0"
                 xmlns="http://schemas.microsoft.com/2011/01/fabric"
                 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ServiceTypes>
    <!-- This is the name of your ServiceType.
         The UseImplicitHost attribute indicates this is a guest service. -->
    <StatelessServiceType ServiceTypeName="IisServiceType" UseImplicitHost="t
  </ServiceTypes>

  <!-- Code package is your service executable. -->
  <CodePackage Name="Code" Version="1.0.0">
    <EntryPoint>
      <!-- Follow this link for more information about deploying Windows conta
      <ContainerHost>
        <ImageName>microsoft/iis:nanoserver-sac2016</ImageName>
      </ContainerHost>
    </EntryPoint>
    <!-- Pass environment variables to your container: -->
    <!--
    <EnvironmentVariables>
      <EnvironmentVariable Name="VariableName" Value="VariableValue"/>
```

3. Right-click on **SFIisApplication** deployment project and click **Publish…**



4. Click on the **Microsoft account** drop down and **Add an account.** Then connect with your Microsoft account.
5. Click on the **Connection Endpoint** drop down and select your remote Service Fabric cluster. You should see a green checkbox saying that the connection is valid. Then, click on **Publish.**



Publish Service Fabric Application

Target profile:
PublishProfiles\Cloud.xml

Save Profile

Microsoft
j[redacted]@[redacted]com

Connection Endpoint:
containers-lab.eastus.cloudapp.azure.com:19000 ✓

Advanced Connection Parameters

How to configure secure connections

Application Parameters File:
ApplicationParameters\Cloud.xml

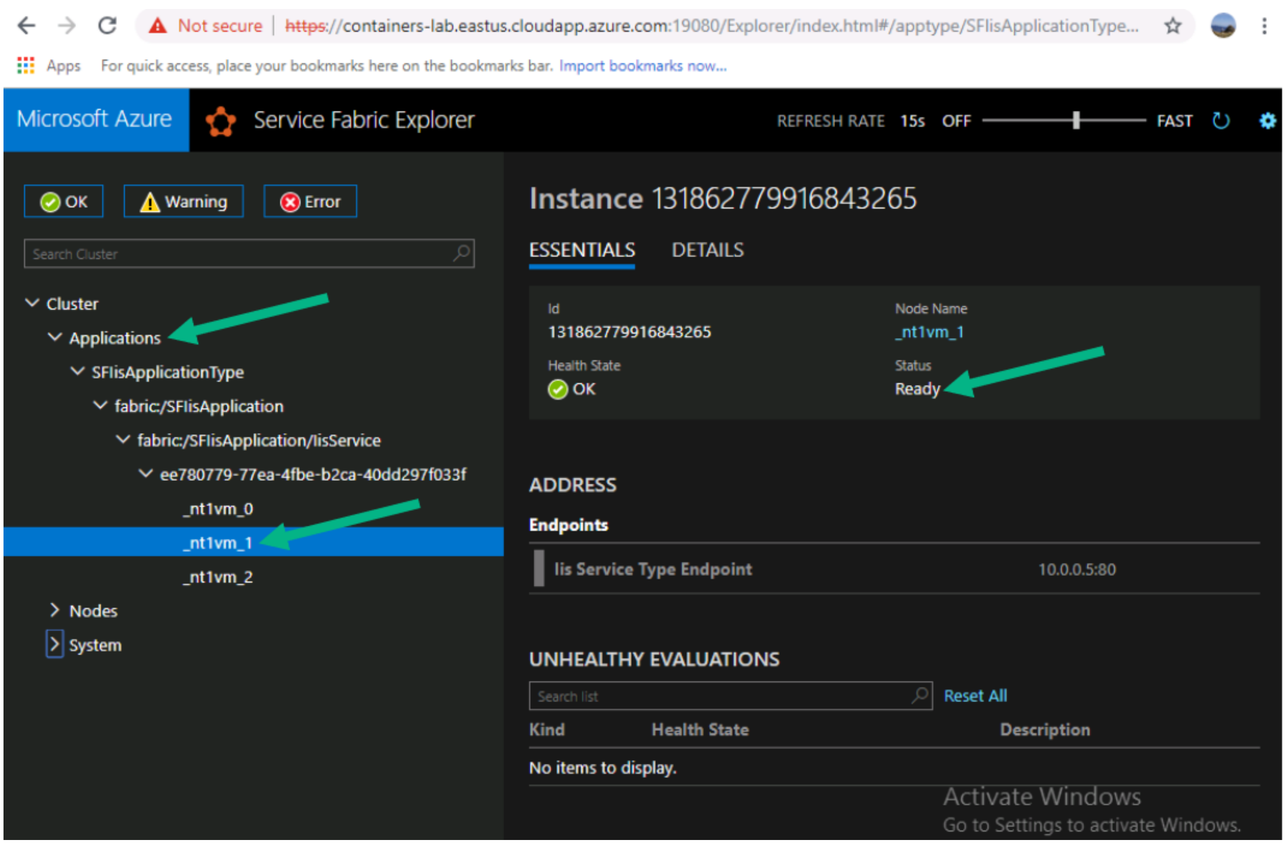
Edit...

☐ Upgrade the Application

Configure Upgrade Settings

Manifest Versions... Publish Cancel

6. Go back to the **Service Fabric Explorer** page and expand the application all the way down to the node level. While the application is being deployed, the nodes will go from **InBuild** to the **Ready** state. Once all nodes are **Ready**, access the cluster DNS on port 80. It should display the default IIS page.


Not secure | https://containers-lab.eastus.cloudapp.azure.com:19080/Explorer/index.html#/apptype/SFIisApplicationType...
Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Microsoft Azure
Service Fabric Explorer
REFRESH RATE 15s OFF FAST
OK
Warning
Error
Search Cluster
Cluster
Applications
SFIisApplicationType
fabric:/SFIisApplication
fabric:/SFIisApplication/IisService
ee780779-77ea-4fbe-b2ca-40dd297f033f
_nt1vm_0
_nt1vm_1
_nt1vm_2
Nodes
System
Activate Windows
Go to Settings to activate Windows.

Instance 131862779916843265

ESSENTIALS DETAILS

Id	Node Name
131862779916843265	_nt1vm_1
Health State	**Status**
OK	Ready

ADDRESS

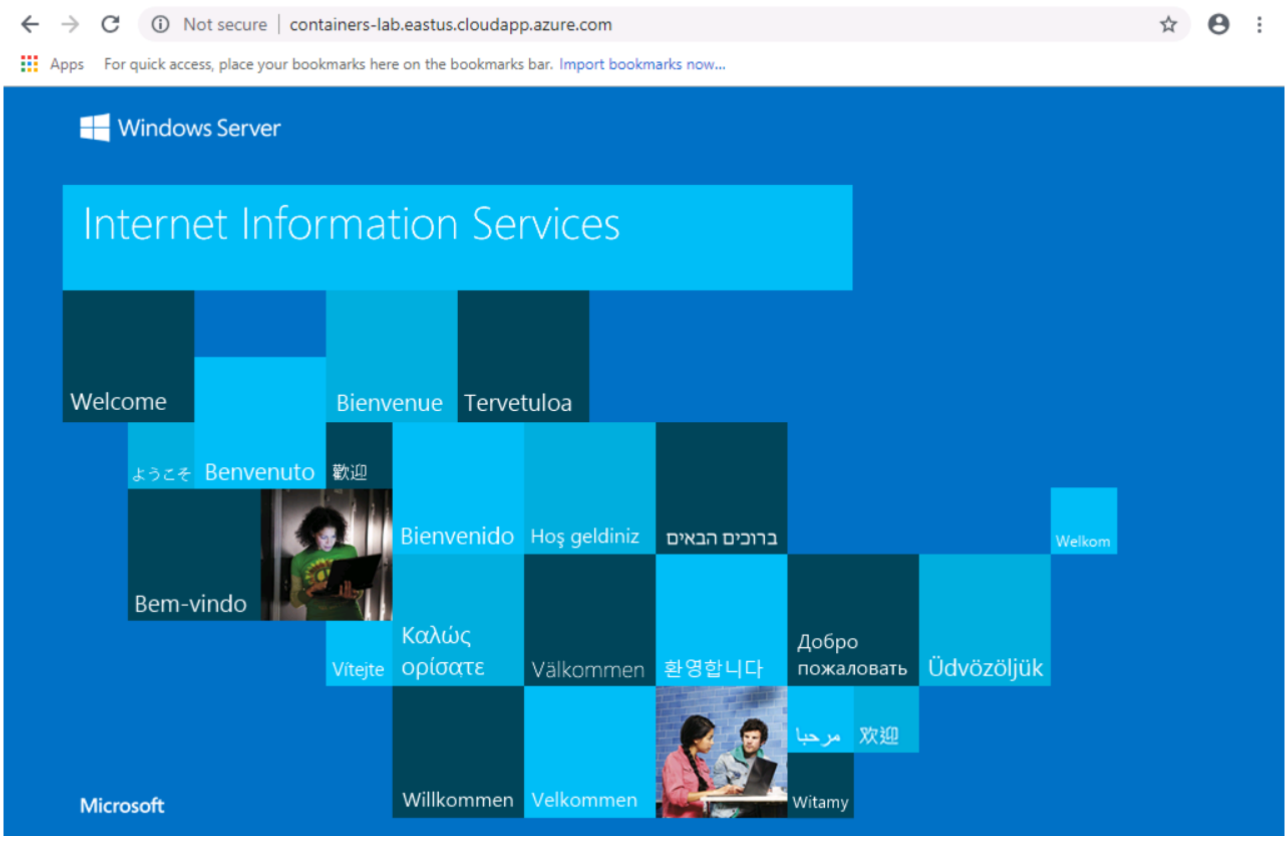
Endpoints

Iis Service Type Endpoint	10.0.0.5:80

UNHEALTHY EVALUATIONS

Kind	Health State	Description

No items to display.



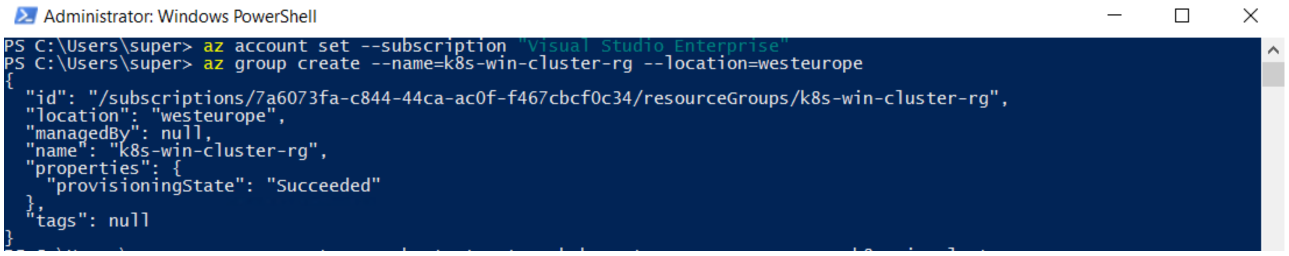
Exercise 3: Create an Azure Container Service Cluster (Optional)

In this exercise, you will create a Windows Kubernetes cluster on Azure Container Services (ACS), and then deploy a container image to the cluster. The container image will be fetched from Docker Hub public registry.

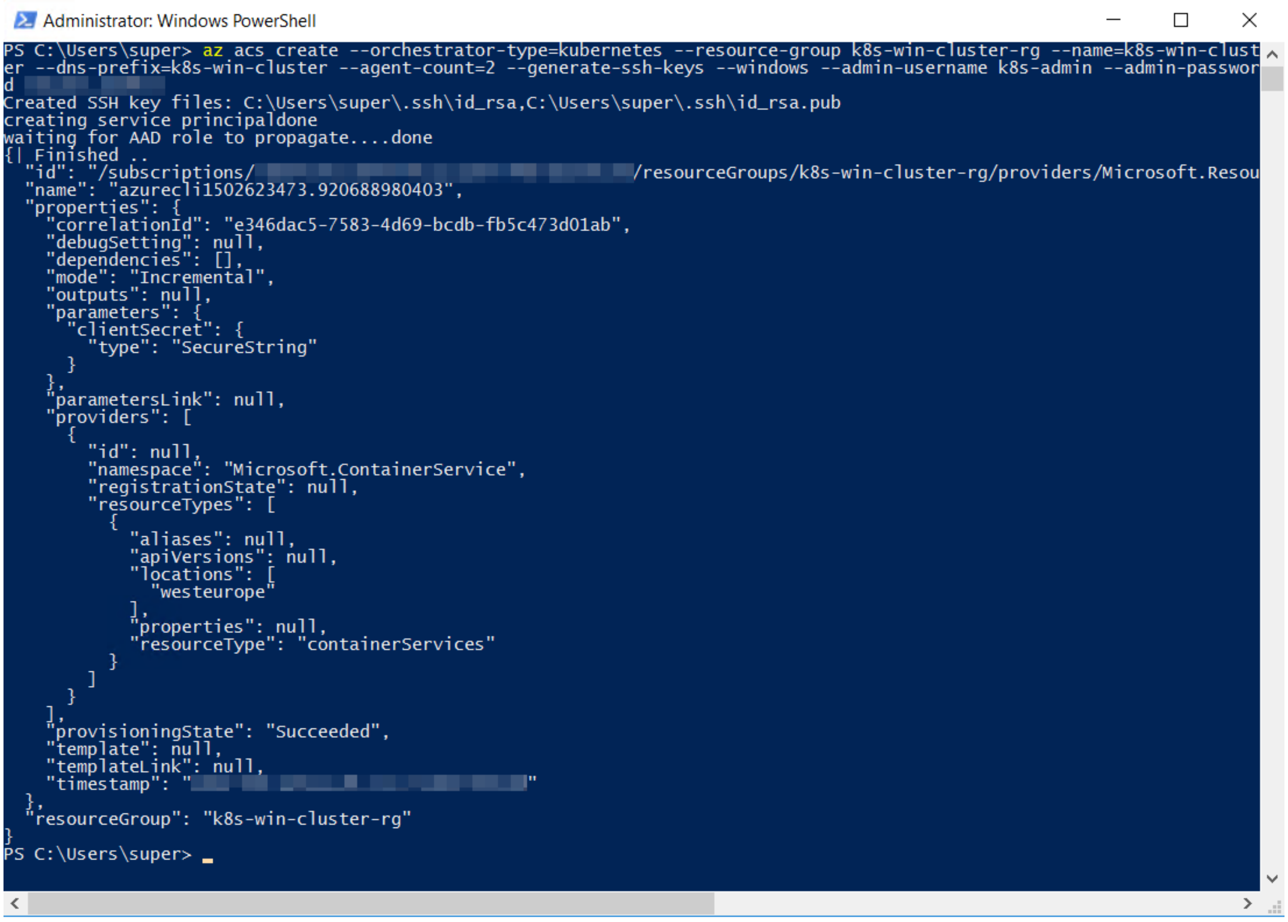
You need to choose unique DNS name for your cluster. To prevent getting **DnsRecordInUse** error, select a unique name for **--dns-prefix** parameter.

Tasks

1. Create Windows Kubernetes Cluster on Azure Container Services

1. On Windows server, open PowerShell.
2. Login to Azure and set the correct subscription if you have not already done this, skip if you have done it. You can find the "name" field in the output of the az login executed above.
   ```
   az login
   az account set --subscription "subscription_name"
   ```
3. Create a resource group named **k8s-win-cluster-rg**, and replace **INITIALS** with your initials so that the resource group name is unique.
   ```
   az group create --name=k8s-win-cluster-rg-INITIALS --location=eastus
   ```
   
   Administrator: Windows PowerShell
   ```
   PS C:\Users\super> az account set --subscription "Visual Studio Enterprise"
   PS C:\Users\super> az group create --name=k8s-win-cluster-rg --location=westeurope
   {
     "id": "/subscriptions/7a6073fa-c844-44ca-ac0f-f467cbcf0c34/resourceGroups/k8s-win-cluster-rg",
     "location": "westeurope",
     "managedBy": null,
     "name": "k8s-win-cluster-rg",
     "properties": {
       "provisioningState": "Succeeded"
     },
     "tags": null
   }
   ```
4. Run the following command to deploy the Kubernetes cluster. You will use one master and two agents in your deployment.
   You need to choose unique DNS name for your cluster. To prevent getting **DnsRecordInUse** error, select a unique name for **--dns-prefix** parameter.
   Make sure that the value of your **admin-password** is between 8-123 characters long and must contain an uppercase character, a lowercase character, a numeric digit and a special character. You can either keep the default password or change it.
5. Run the following command:
   ```
   az acs create --orchestrator-type=kubernetes --resource-group k8s-win-cluster-rg-INITIALS --name=k8s-win-cluster --dns-prefix=k8s-win-cluster-INITIALS --agent-count=1 --agent-vm-size=Standard_A1 --master-vm-size=Standard_A1 --generate-ssh-keys --windows --admin-username k8s-admin --admin-password *P@ssw0rd123!
   ```

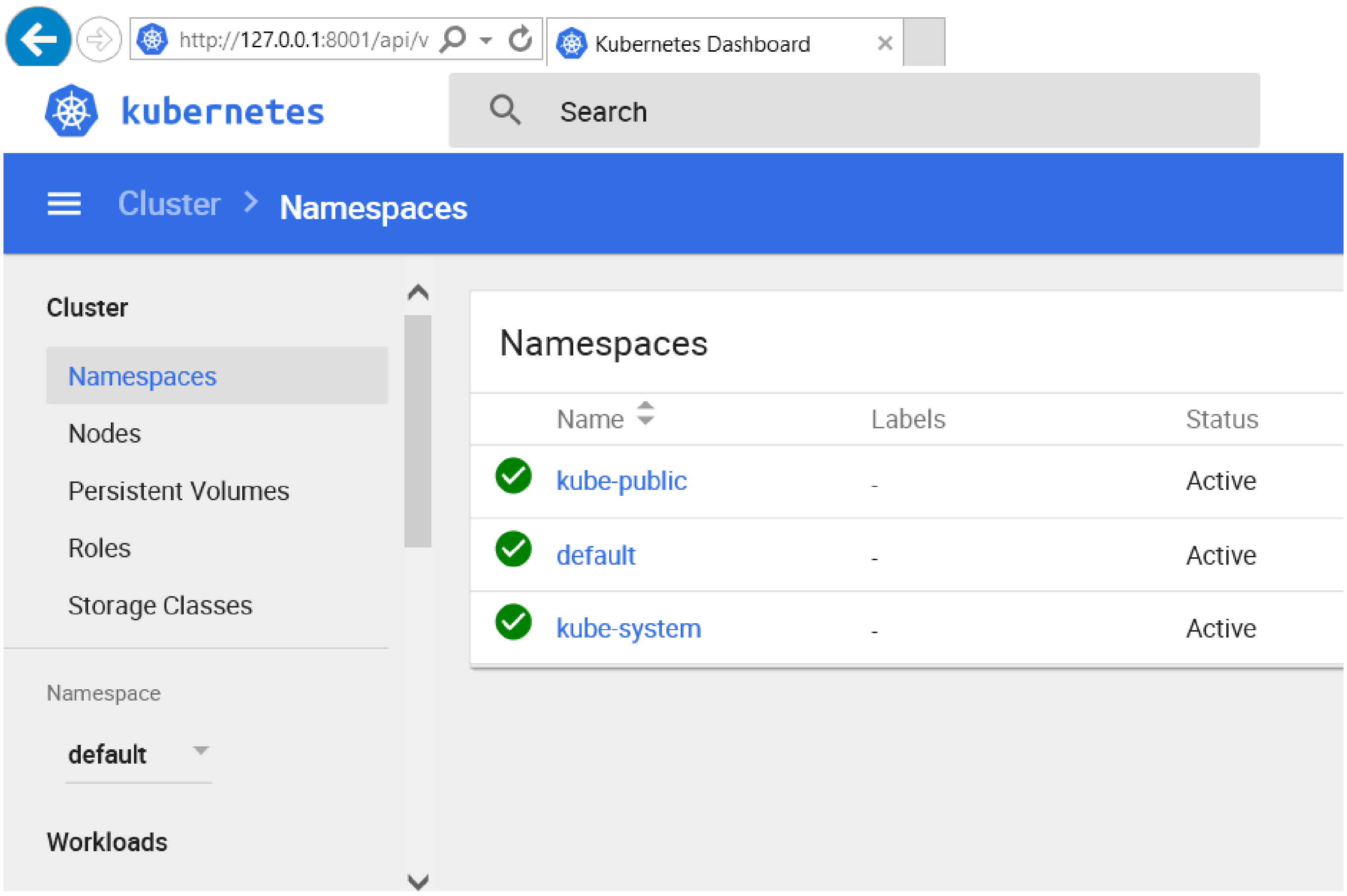
*Note that this command will generate public/private ssh keys in **c:\users\Administrator\.ssh** folder.*



```
PS C:\Users\super> az acs create --orchestrator-type=kubernetes --resource-group k8s-win-cluster-rg --name=k8s-win-clust
er --dns-prefix=k8s-win-cluster --agent-count=2 --generate-ssh-keys --windows --admin-username k8s-admin --admin-passwor
d [redacted]
Created SSH key files: C:\Users\super\.ssh\id_rsa,C:\Users\super\.ssh\id_rsa.pub
creating service principaldone
waiting for AAD role to propagate....done
{| Finished ..
  "id": "/subscriptions/[redacted]/resourceGroups/k8s-win-cluster-rg/providers/Microsoft.Resou
  "name": "azurecli1502623473.920688980403",
  "properties": {
    "correlationId": "e346dac5-7583-4d69-bcdb-fb5c473d01ab",
    "debugSetting": null,
    "dependencies": [],
    "mode": "Incremental",
    "outputs": null,
    "parameters": {
      "clientSecret": {
        "type": "SecureString"
      }
    },
    "parametersLink": null,
    "providers": [
      {
        "id": null,
        "namespace": "Microsoft.ContainerService",
        "registrationState": null,
        "resourceTypes": [
          {
            "aliases": null,
            "apiVersions": null,
            "locations": [
              "westeurope"
            ],
            "properties": null,
            "resourceType": "containerServices"
          }
        ]
      }
    ],
    "provisioningState": "Succeeded",
    "template": null,
    "templateLink": null,
    "timestamp": "[redacted]"
  },
  "resourceGroup": "k8s-win-cluster-rg"
}
PS C:\Users\super> _
```

6. Run the following command to download the Kubernetes cluster configuration to the local config file **C:\users\Administrator\.kube\config**.
   ```
   az acs kubernetes get-credentials --resource-group=k8s-win-cluster-rg-INITIALS --
   name=k8s-win-cluster
   ```
   Be aware that you will need the content of *.kube/config* file while creating Kubernetes service endpoint on VSTS in the DevOps module.
7. Now you can run the following command to open your Kubernetes dashboard.
   ```
   az acs kubernetes browse --name=k8s-win-cluster --resource-group=k8s-win-cluster-
   rg-INITIALS
   ```

> kubectl recently switched to redirecting to an 'https' protocol instead of 'http' for the '/ui' shortcut. Until the az acs command is updated, we need to manually access http://localhost:8001/api/v1/namespaces/kube-system/services/kubernetes-dashboard/proxy/#!/namespace?namespace=default instead of the default redirection URL

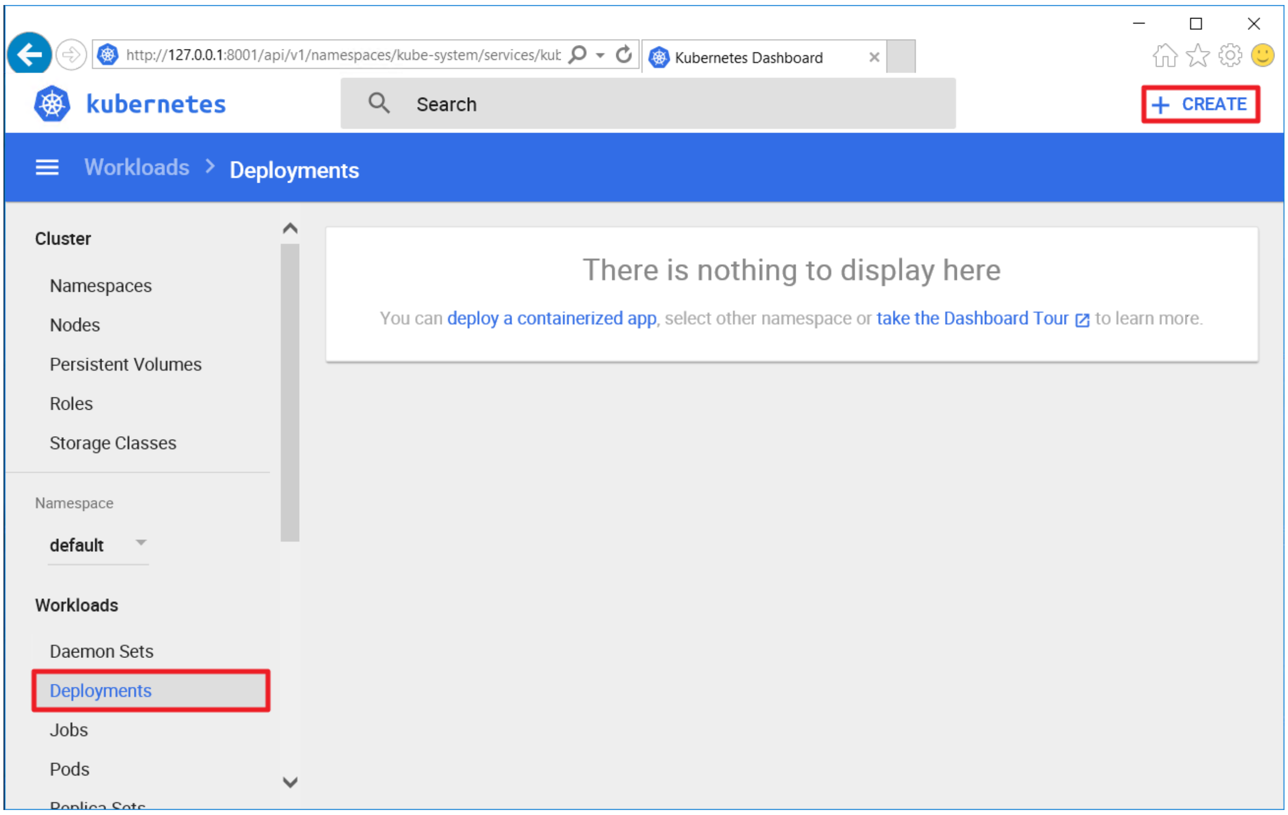


Namespaces

Name	Labels	Status
kube-public	-	Active
default	-	Active
kube-system	-	Active

2. Deploy an Application to Windows Kubernetes Cluster

1. Now your Windows Kubernetes cluster is ready, you will deploy a containerized application located in Docker Hub.
   On Kubernetes dashboard, go to **Deployment** page, and then click **Create**.

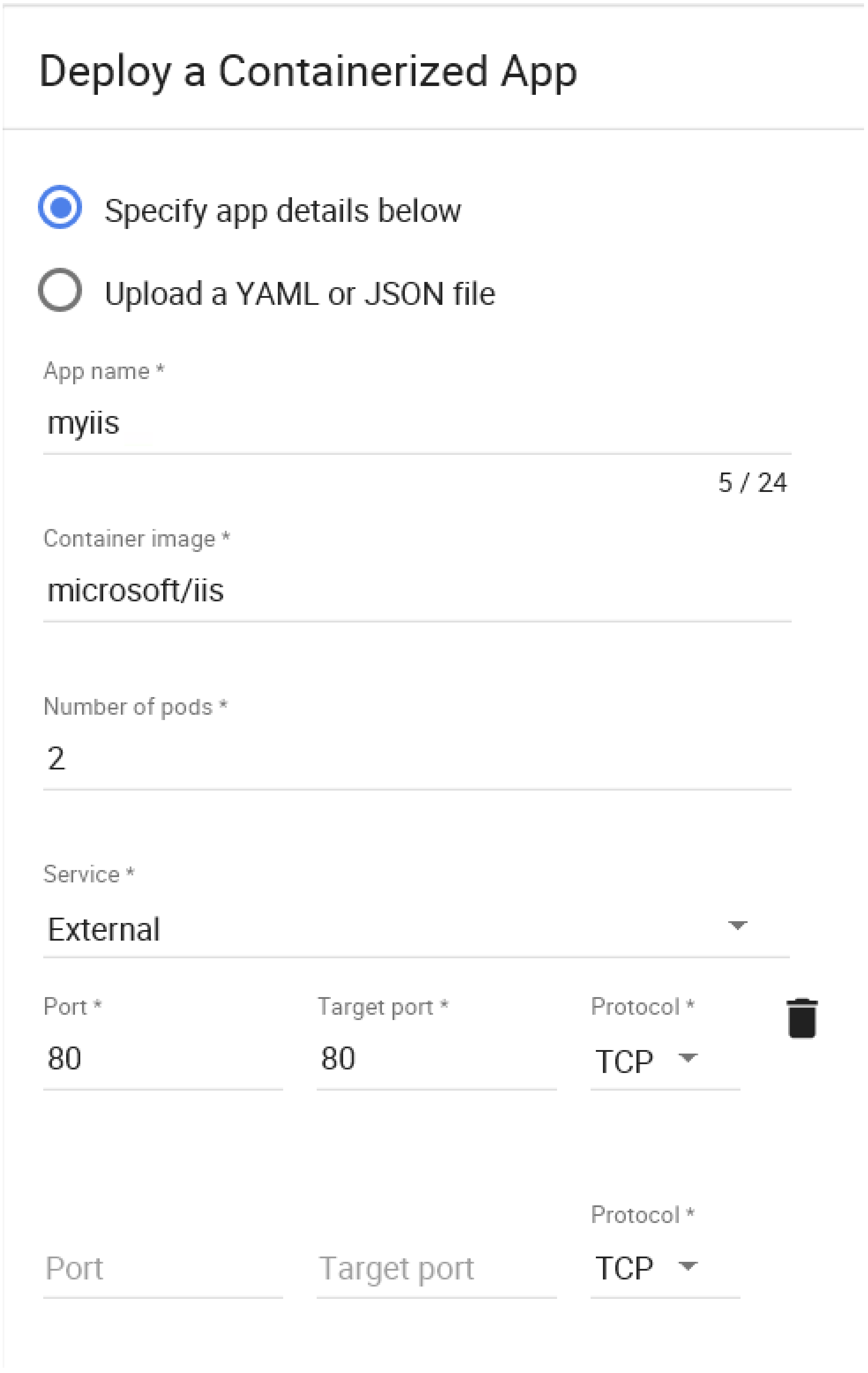


There is nothing to display here

You can deploy a containerized app, select other namespace or take the Dashboard Tour to learn more.

2. On the **Deploy a Containerized App** dialog box, select the deployment type **Specify app details below.**
   In this task, you will enter application's detail manually. In the future, you can also use YAML or JSON file to perform the deployment. Fill the parameters as follows (also please see screenshot below):

- **App name:** myiis
- **Container image**: microsoft/iis
- **Number of pods**: 1 or 2
- **Service**: External (Your application will be accessible using a public IP)
- Set **Port** (incoming port) and **Target port** to 80.
  After setting the parameters click **Deploy**.


Deploy a Containerized App

[✓ Specify app details below] Upload a YAML or JSON file

App name *
myiis
5 / 24

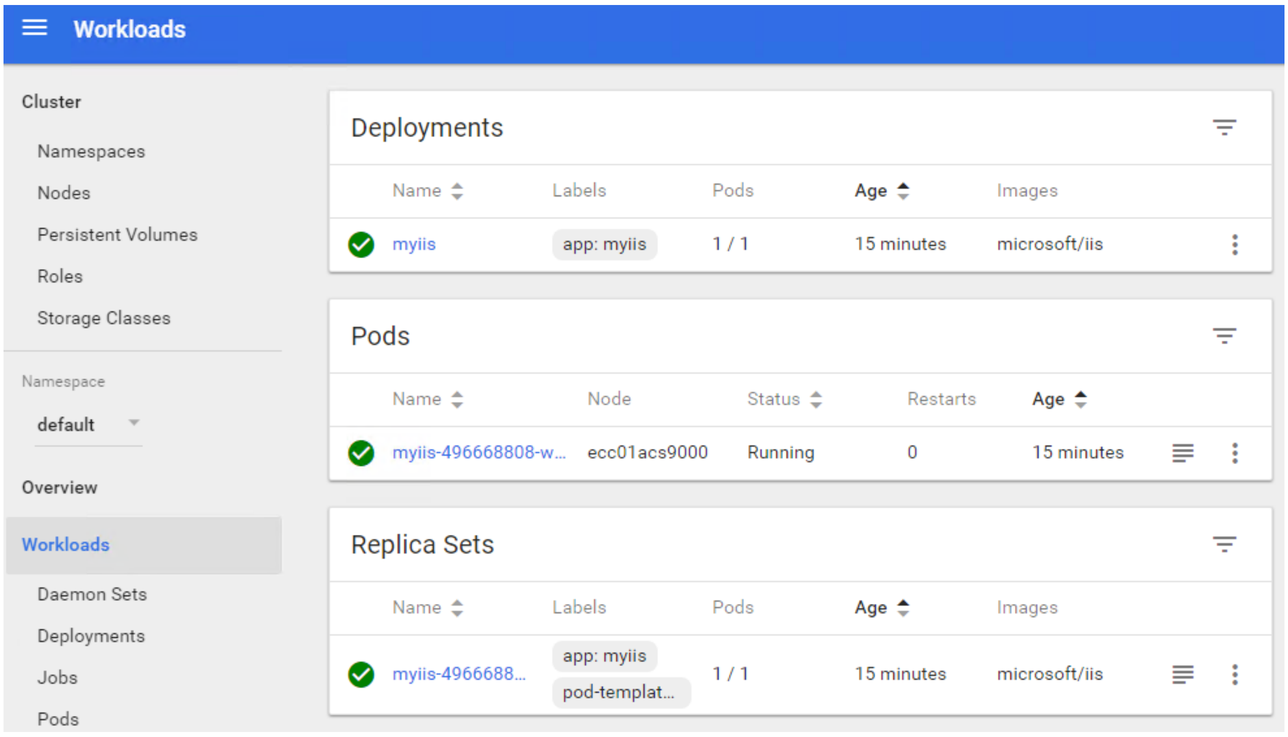
Container image *
microsoft/iis

Number of pods *
2

Service *
External

Port *	Target port *	Protocol *
80	80	TCP
Port	Target port	TCP

3. You can check the status of your deployment on **Workloads** page. You will notice that the status of your pods changes from **Pending** to **Running**. You do need to keep hitting refresh to get the new status. This deployment can take a while since the images needs to be downloaded from the Docker Hub (nothing is cached).



Deployments

Name	Labels	Pods	Age	Images
✓ myiis	app: myiis	1 / 1	15 minutes	microsoft/iis

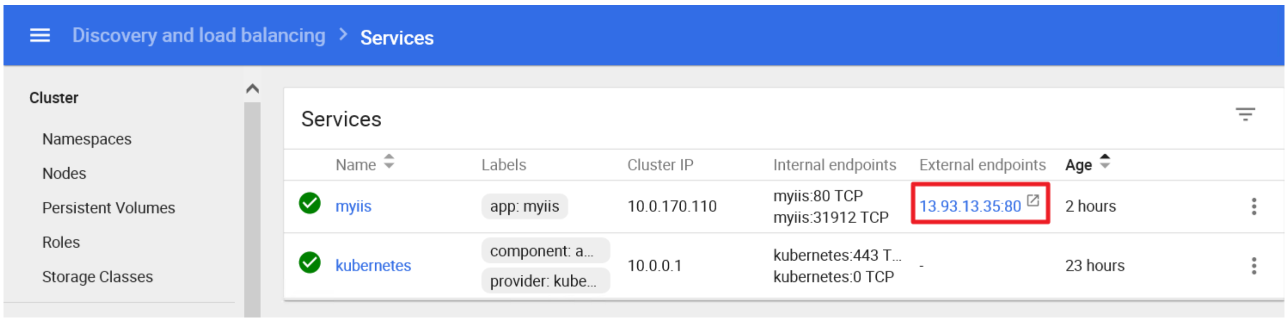
Pods

Name	Node	Status	Restarts	Age
✓ myiis-496668808-w...	ecc01acs9000	Running	0	15 minutes

Replica Sets

Name	Labels	Pods	Age	Images
✓ myiis-4966688...	app: myiis pod-templat...	1 / 1	15 minutes	microsoft/iis

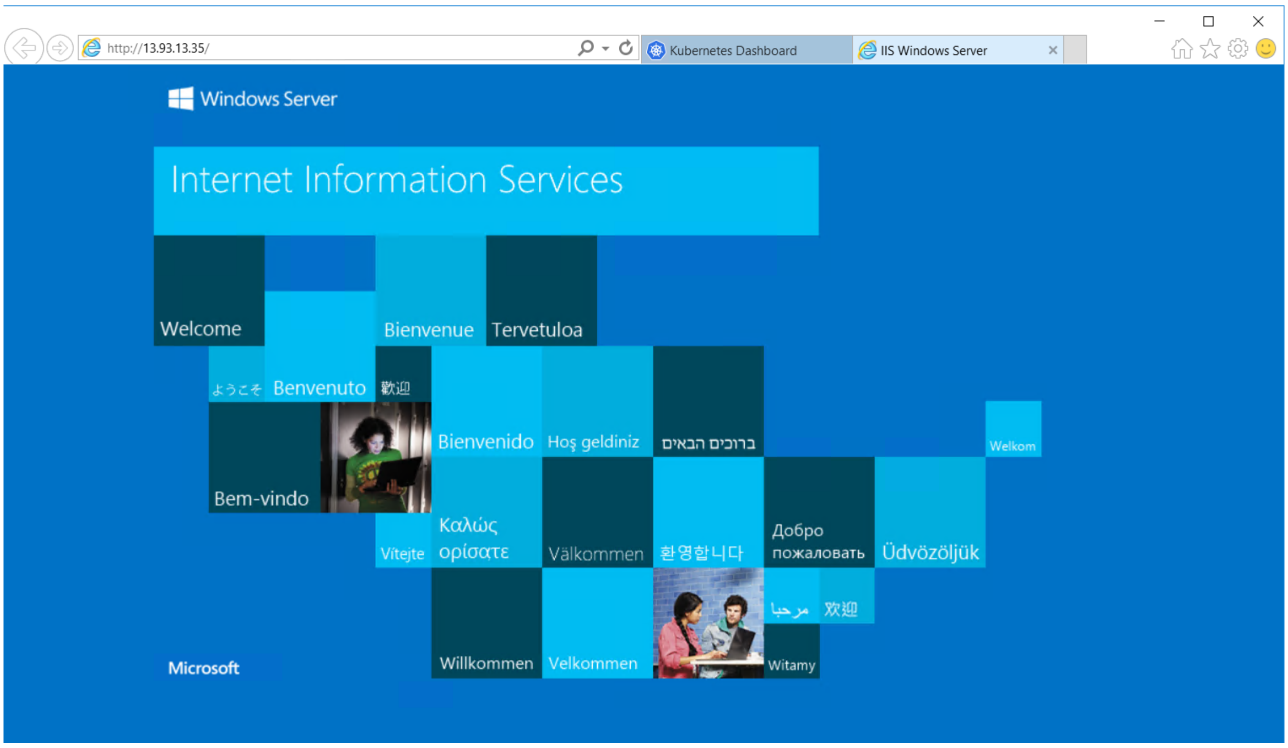
4. After the deployment is completed, go to **Service** page. Click the IP under **External endpoints** column of **myiis** service and you will see the welcome page of IIS server.
   You may need to refresh the **Services** page, if **myiis** service is not listed when you first open the page.


Discovery and load balancing > Services

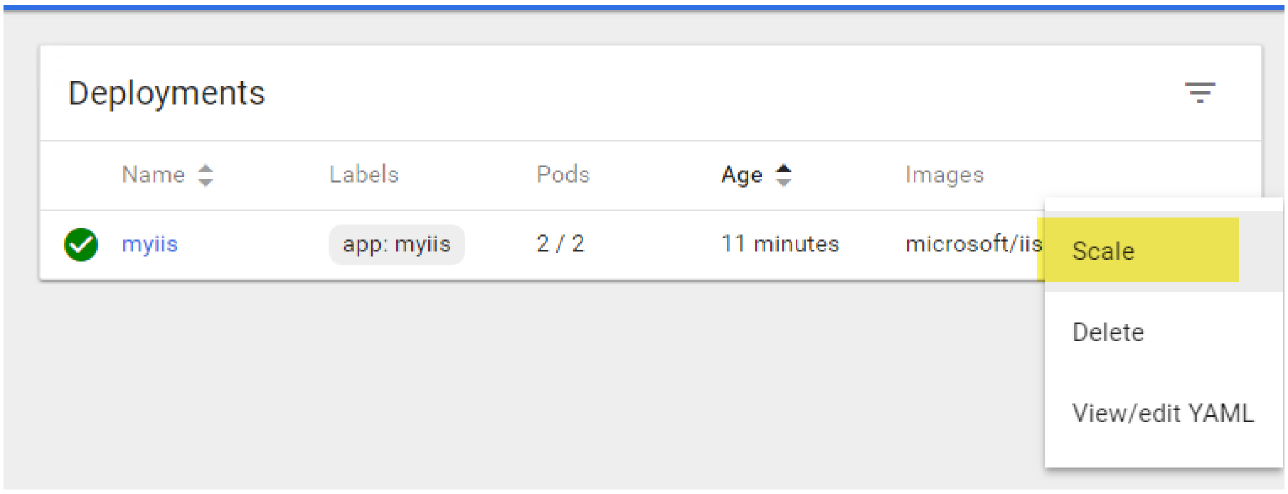
Cluster
- Namespaces
- Nodes
- Persistent Volumes
- Roles
- Storage Classes

Services

	Name	Labels	Cluster IP	Internal endpoints	External endpoints	Age
✓	myiis	app: myiis	10.0.170.110	myiis:80 TCP myiis:31912 TCP	13.93.13.35:80	2 hours
✓	kubernetes	component: a... provider: kube...	10.0.0.1	kubernetes:443 T... kubernetes:0 TCP	-	23 hours

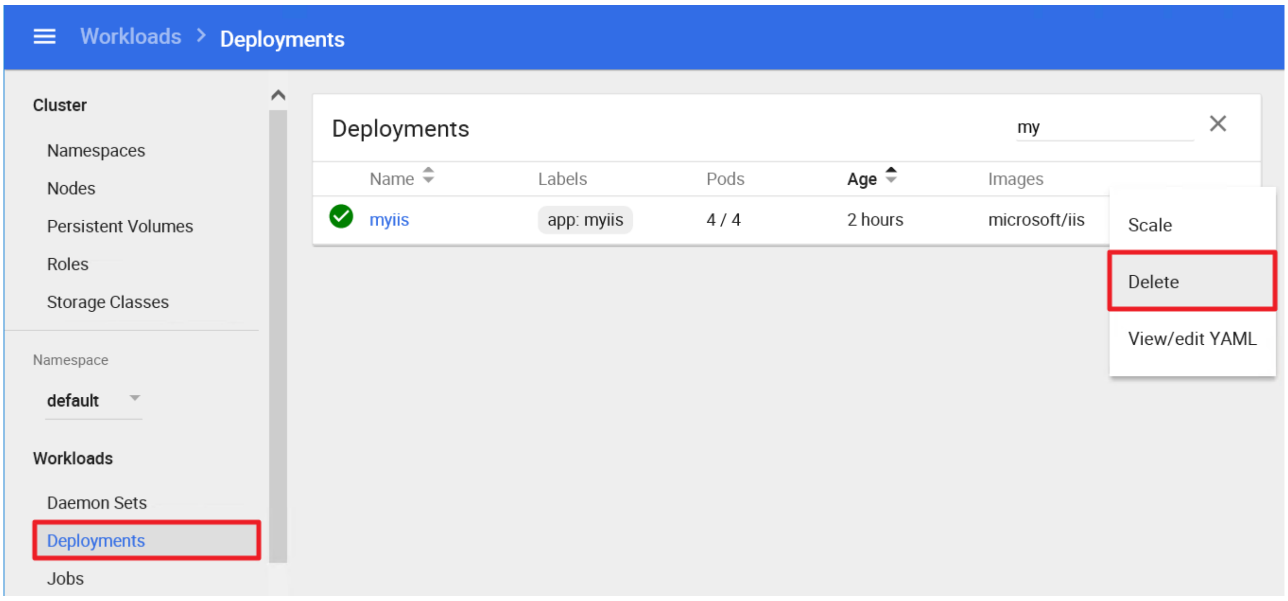

http://13.93.13.35/

5. On the deployments page, try scaling up the number of pods to **1** or **2**, depending on what you started with.


Deployments

	Name	Labels	Pods	Age	Images
✓	myiis	app: myiis	2 / 2	11 minutes	microsoft/iis

6. Now, let's delete this deployment and service. Go to the **Deployments** page and delete **myiis**.

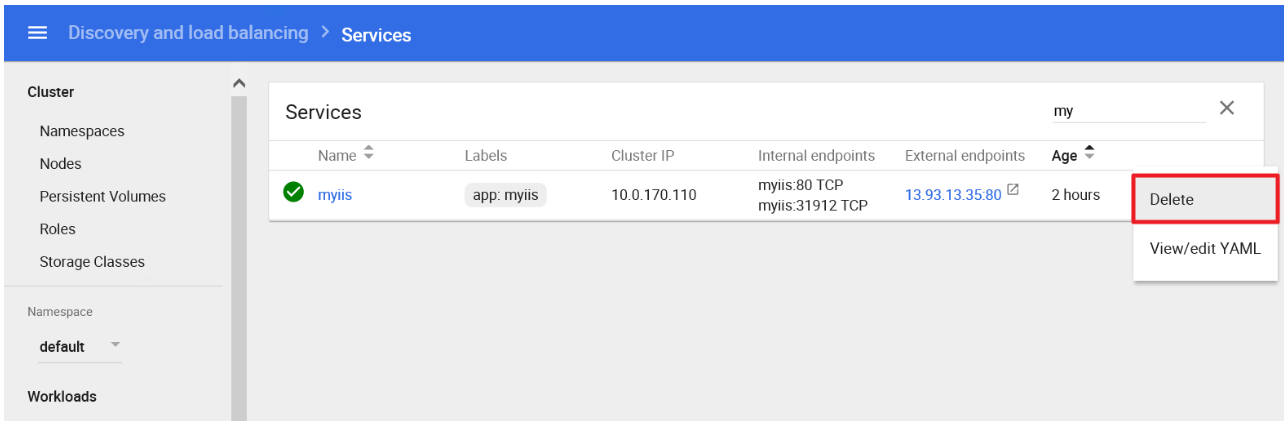


Deployments

my

Name	Labels	Pods	Age	Images
myiis	app: myiis	4 / 4	2 hours	microsoft/iis

7. Then on the **Services** page, delete **myiis** service.



Services

my

Name	Labels	Cluster IP	Internal endpoints	External endpoints	Age
myiis	app: myiis	10.0.170.110	myiis:80 TCP myiis:31912 TCP	13.93.13.35:80	2 hours

8. You have now completed all the tasks in this exercise.