Lab environment

In this lab you will use the following two machines.

- A machine known as a Node will host the PU MRP app. The only task you will perform on the node is to install the Puppet Agent. The Puppet Agent can run on Linux or Windows. For this lab, we will configure the Node in a Linux Ubuntu Virtual Machine (VM).
- A Puppet Master machine. The rest of the configuration will be applied by instructing Puppet how to configure the Node through Puppet Programs, on the Puppet Master. The Puppet Master *must* be a Linux machine. For this lab, we will configure the Puppet Master in a Linux Ubuntu VM.

Instead of manually creating the VMs in Azure, we will use an Azure Resource Management (ARM) template.

Task 1: Provision a Puppet Master and Node in Azure using Azure Resource Manager templates (both inside Linux Ubuntu Virtual Machines)

1. To provision the required VMs in Azure using an ARM template, select the **Deploy to Azure** button, and follow the wizard. You will need to log in to the Azure Portal.

   Deploy to Azure **Visualize**

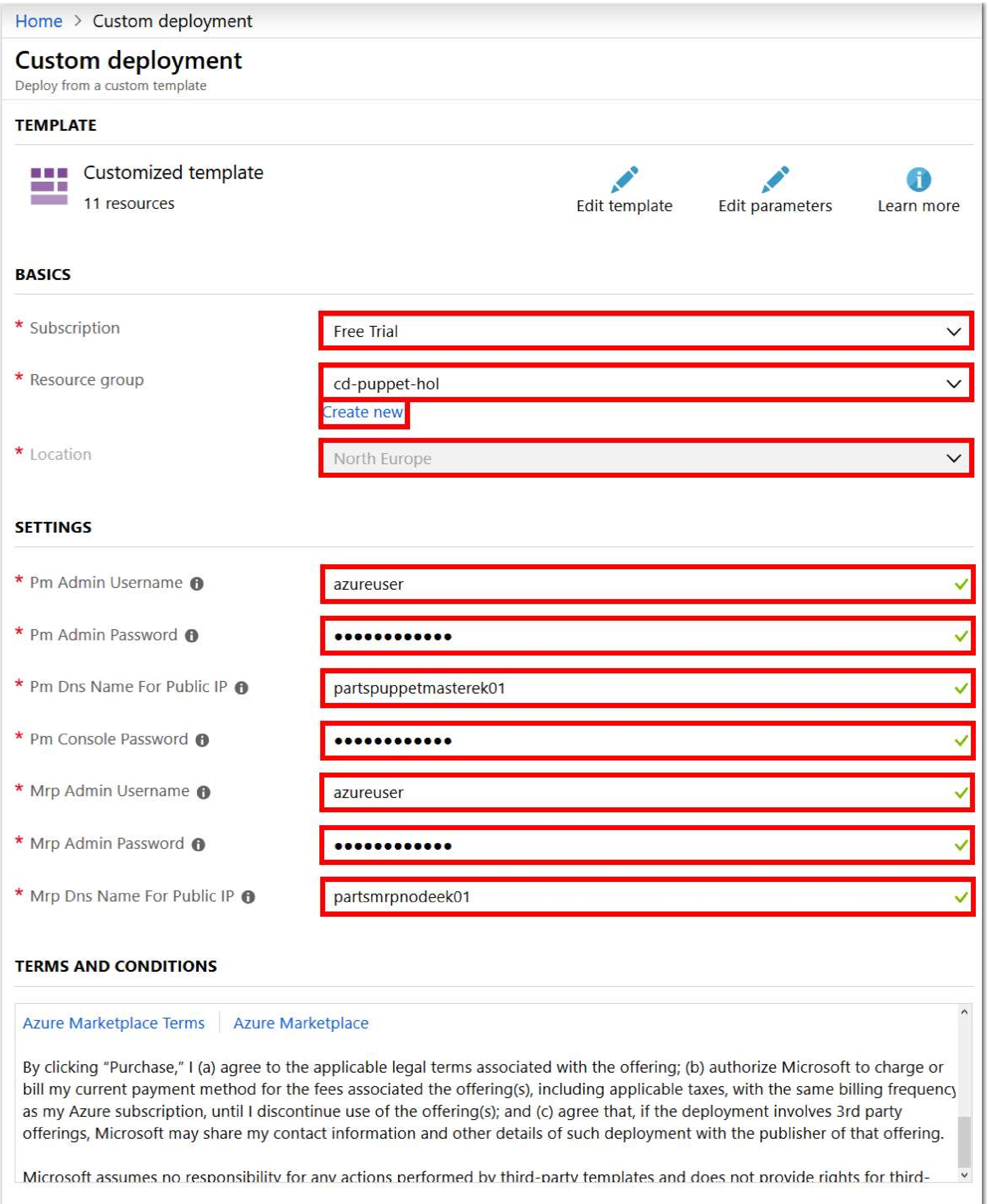
   The VMs will be deployed to a Resource Group along with a Virtual Network (VNET), and some other required resources.

   Note: You can review the JSON template and script files used to provision the VMs in the Puppet lab files folder on GitHub.

2. When prompted specify a **Subscription**, **Location**, and **Resource Group** for deploying your VM resources. Provide admin **usernames** and **passwords**, as well as a unique **Public DNS Names** for both machines.

   Consider the following guidelines.

   - **Subscription**. `< your Azure subscription >`.
   - **Resource group**. `< a unique resource group name >`. For example `puprgek01`. Create a new Resource Group during the deployment process, to demarcate the resources that you set up in this lab. Remove the resources created in this lab by deleting the Resource Group.
   - **Location**. Select a region to deploy the VMs to. For example `West Europe` or `East US`.
   - **Pm Admin Username**. The **Pm** refers to the Puppet Master VM, and **Mrp** refers to the Node VM. Choose the same Admin Username for both **Pm** and **Mrp** VMs. For example, `azureuser`.
   - **Pm Admin Password**. Set the same Admin Password for both VMs, and for the **Pm Console**. For example, `Passw0rd0134`
   - **Pm Dns Name For Public IP**. Include the word *master* in the Puppet Master DNS name, to distinguish it from the Node VM. You could include the word *node* in the Node DNS name, if you wish, for example `partsmrpnode`. Create unique DNS names for both VMs by adding your initials as a suffix. For example, `partspuppetmasterek01` and `partsmrpnodeek01`



Home > Custom deployment

Custom deployment

Deploy from a custom template

TEMPLATE

Customized template

11 resources

BASICS

* Subscription	Free Trial
* Resource group	cd-puppet-hol Create new
* Location	North Europe

SETTINGS

* Pm Admin Username	azureuser
* Pm Admin Password	•••••••••••••
* Pm Dns Name For Public IP	partspuppetmasterek01
* Pm Console Password	•••••••••••••
* Mrp Admin Username	azureuser
* Mrp Admin Password	•••••••••••••
* Mrp Dns Name For Public IP	partsmrpnodeek01

TERMS AND CONDITIONS

Azure Marketplace Terms | Azure Marketplace

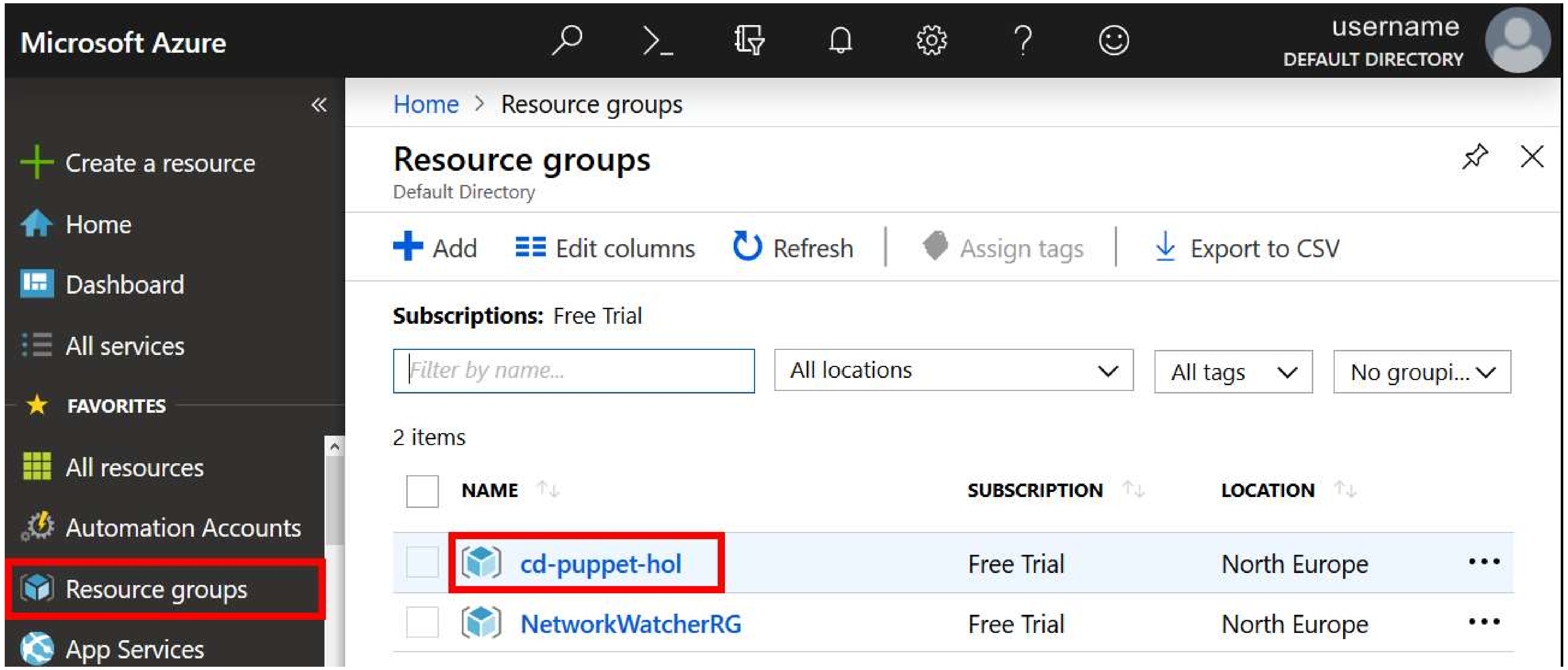
By clicking “Purchase,” I (a) agree to the applicable legal terms associated with the offering; (b) authorize Microsoft to charge or bill my current payment method for the fees associated the offering(s), including applicable taxes, with the same billing frequenc as my Azure subscription, until I discontinue use of the offering(s); and (c) agree that, if the deployment involves 3rd party offerings, Microsoft may share my contact information and other details of such deployment with the publisher of that offering.

Microsoft assumes no responsibility for any actions performed by third-party templates and does not provide rights for third-

Make a note of the location region, as well as any usernames and passwords you set for the VMs.

Use the **checkbox** to agree to the Azure Marketplace terms and conditions. Select the **Purchase** button. Allow about 10 minutes for each deployment to complete.

3. When the deployment completes, select the **Resource Groups** blade from the Azure Portal menu. Choose the name of Resource Group that you created, **cd-puppet-hol** for example.



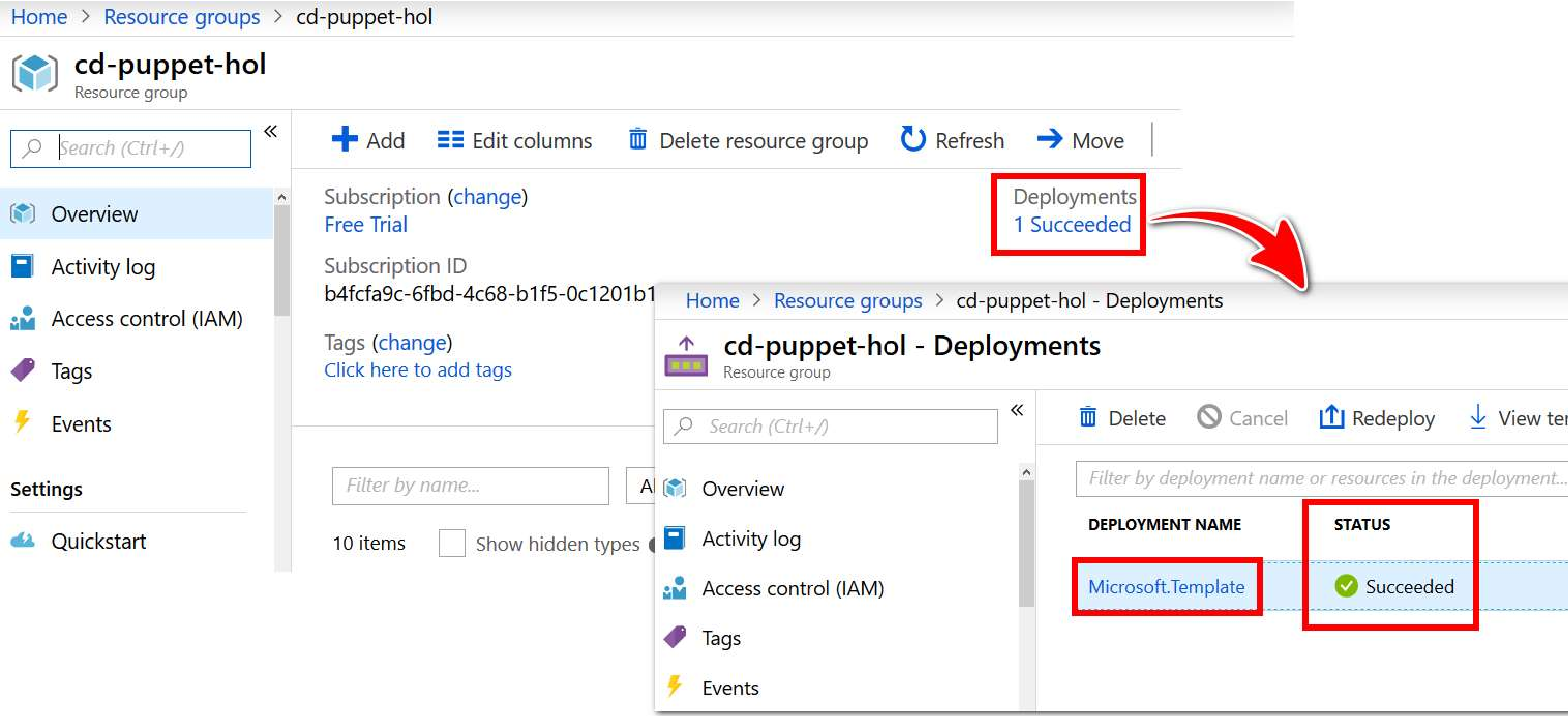
Subscriptions: Free Trial

2 items

NAME	SUBSCRIPTION	LOCATION
cd-puppet-hol	Free Trial	North Europe
NetworkWatcherRG	Free Trial	North Europe

4. Within the **Resource Group Overview** pane, beside **Deployments**, select the **1 succeeded** message to view the deployment details. The status of the `Microsoft.Template` resource should indicate that the deployment has **succeeded**.

   Choose the `Microsoft.Template` resource, listed in the **Deployment Name** column.



Subscription (change)
Free Trial

Subscription ID
b4fcfa9c-6fbd-4c68-b1f5-0c1201b1

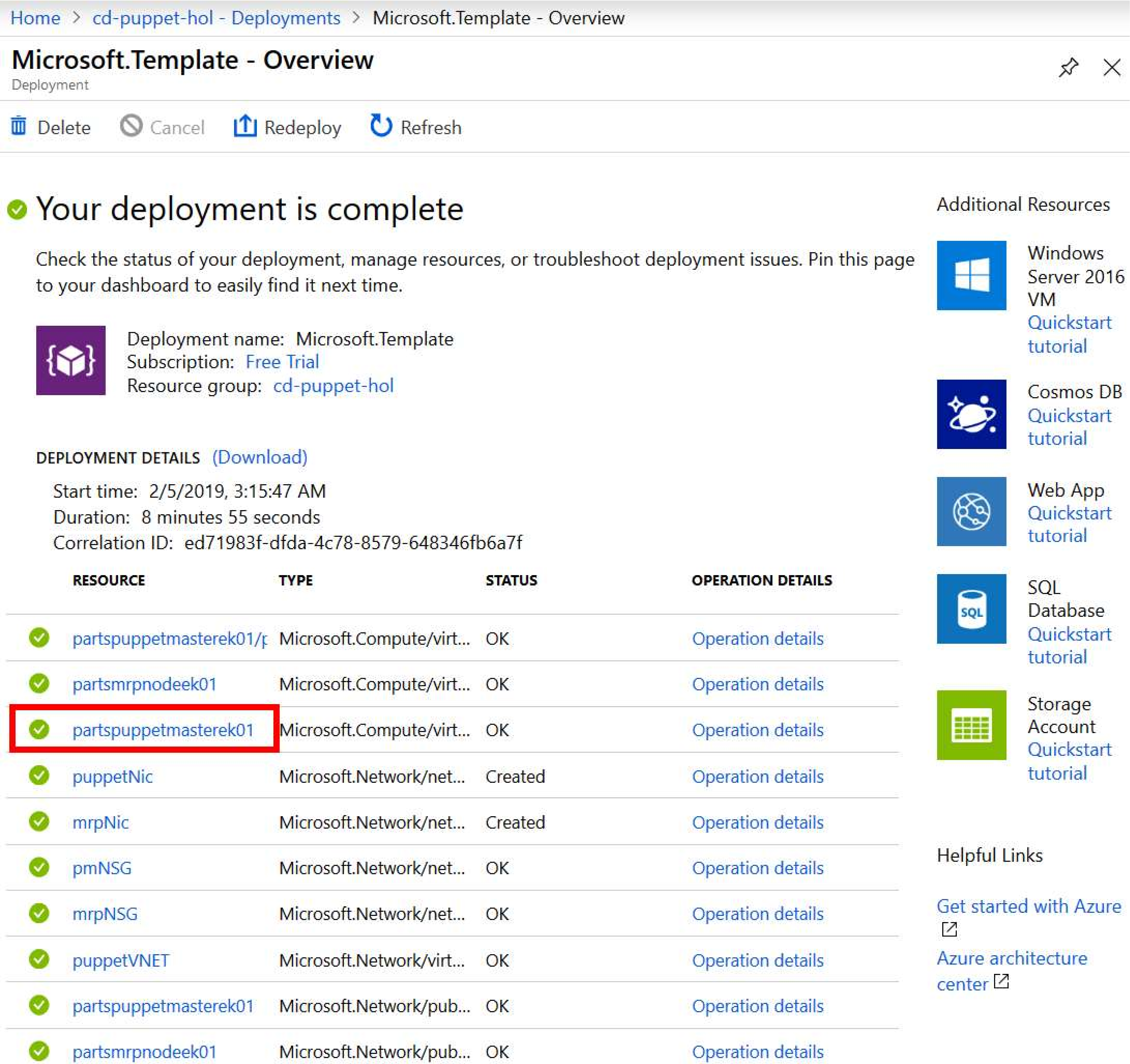
Tags (change)
Click here to add tags

Deployments
1 Succeeded

10 items

DEPLOYMENT NAME	STATUS
Microsoft.Template	✓ Succeeded

5. There is a list of the deployed resources in the **Microsoft.Template - Overview** pane, below the **Your deployment is complete** message. Select the **Puppet Master VM** from the list, for example `partspuppetmasterek01`.



Home > cd-puppet-hol - Deployments > Microsoft.Template - Overview

Microsoft.Template - Overview

Deployment

Delete
Cancel
Redeploy
Refresh

Your deployment is complete

Check the status of your deployment, manage resources, or troubleshoot deployment issues. Pin this page to your dashboard to easily find it next time.

Deployment name: Microsoft.Template
Subscription: Free Trial
Resource group: cd-puppet-hol

DEPLOYMENT DETAILS (Download)

Start time: 2/5/2019, 3:15:47 AM
Duration: 8 minutes 55 seconds
Correlation ID: ed71983f-dfda-4c78-8579-648346fb6a7f

	RESOURCE	TYPE	STATUS	OPERATION DETAILS
✓	partspuppetmasterek01/p	Microsoft.Compute/virt...	OK	Operation details
✓	partsmrpnodeek01	Microsoft.Compute/virt...	OK	Operation details
✓	partspuppetmasterek01	Microsoft.Compute/virt...	OK	Operation details
✓	puppetNic	Microsoft.Network/net...	Created	Operation details
✓	mrpNic	Microsoft.Network/net...	Created	Operation details
✓	pmNSG	Microsoft.Network/net...	OK	Operation details
✓	mrpNSG	Microsoft.Network/net...	OK	Operation details
✓	puppetVNET	Microsoft.Network/virt...	OK	Operation details
✓	partspuppetmasterek01	Microsoft.Network/pub...	OK	Operation details
✓	partsmrpnodeek01	Microsoft.Network/pub...	OK	Operation details

Additional Resources

Windows Server 2016 VM
Quickstart tutorial

Cosmos DB
Quickstart tutorial

Web App
Quickstart tutorial

SQL Database
Quickstart tutorial

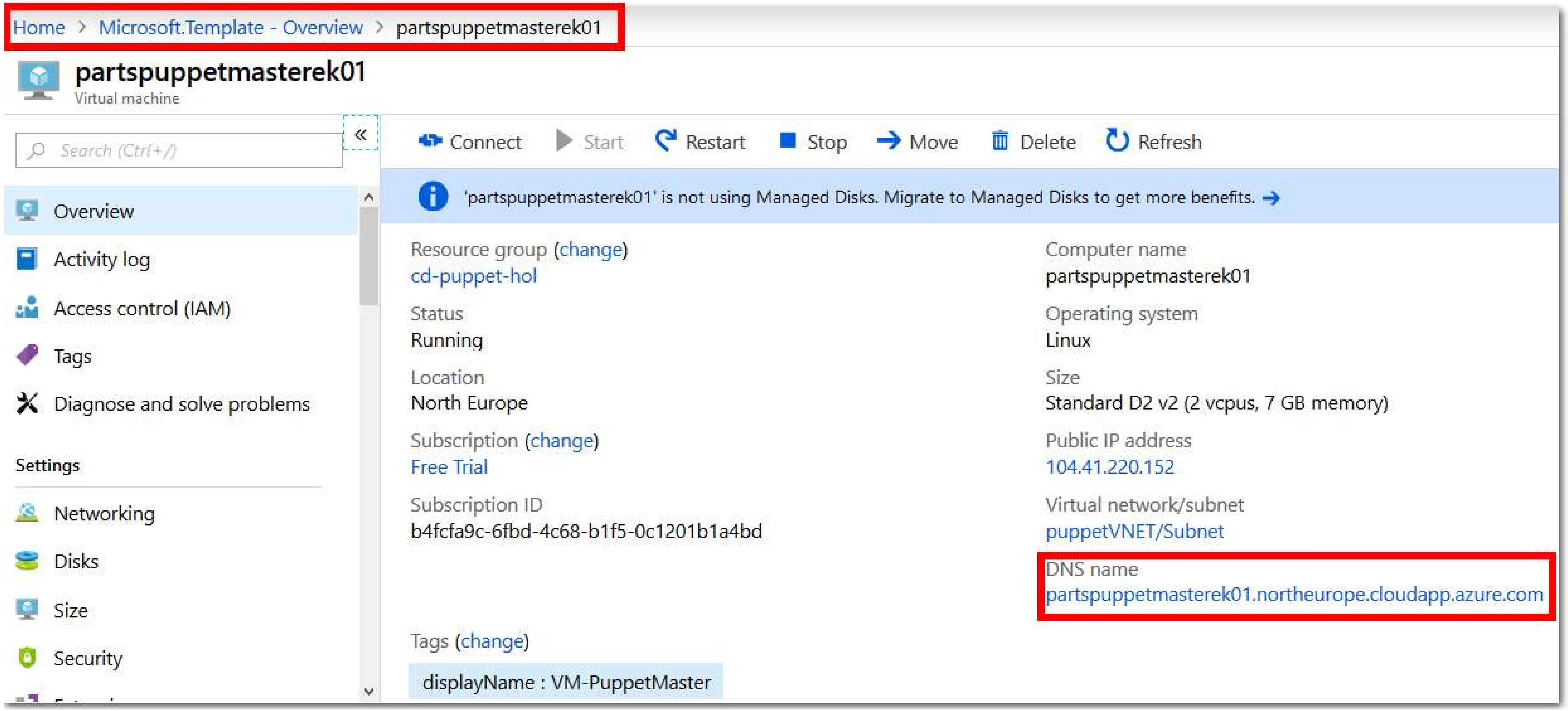
Storage Account
Quickstart tutorial

Helpful Links

Get started with Azure

Azure architecture center

6. Make a note of the **DNS name** value, listed inside the **Overview** pane for the **Puppet Master Virtual Machine**. The DNS name value is in the format `machinename.region.cloudapp.azure.com`. For example, `partspuppetmasterek01.northeurope.cloudapp.azure.com`.



Home > Microsoft.Template - Overview > partspuppetmasterek01

partspuppetmasterek01
Virtual machine

'partspuppetmasterek01' is not using Managed Disks. Migrate to Managed Disks to get more benefits. →

Resource group (change): cd-puppet-hol

Status: Running

Location: North Europe

Subscription (change): Free Trial

Subscription ID: b4fcfa9c-6fbd-4c68-b1f5-0c1201b1a4bd

Computer name: partspuppetmasterek01

Operating system: Linux

Size: Standard D2 v2 (2 vcpus, 7 GB memory)

Public IP address: 104.41.220.152

Virtual network/subnet: puppetVNET/Subnet

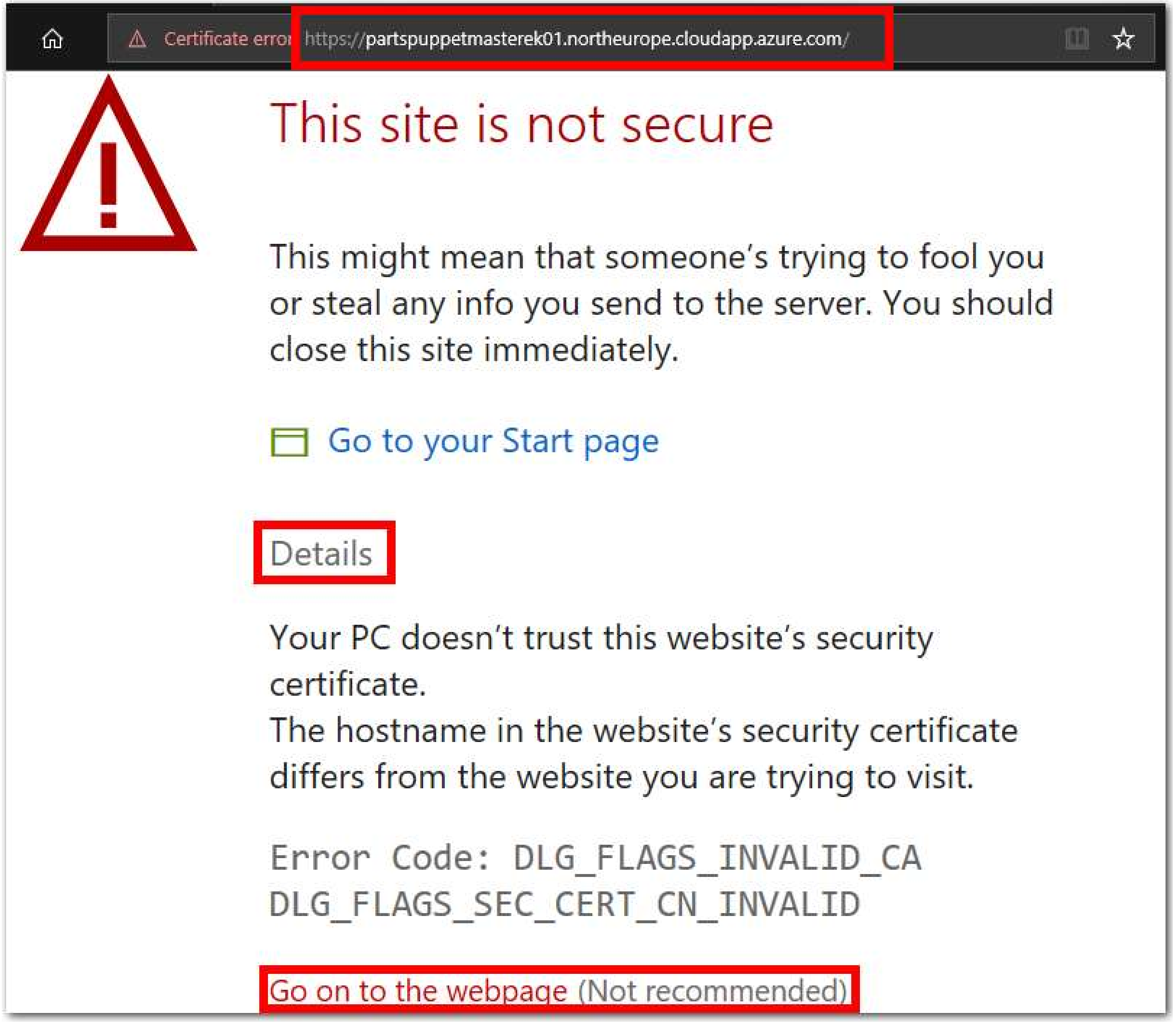
DNS name: partspuppetmasterek01.northeurope.cloudapp.azure.com

Tags (change): displayName : VM-PuppetMaster

7. Append `https://` to the beginning of the DNS name to create a URL for the Puppet Master's public DNS Address. The resulting URL should be in the format `https://machinename.region.cloudapp.azure.com`. For example, `https://partspuppetmasterek01.northeurope.cloudapp.azure.com`.

   Using `https`, *not* `http`, visit the URL in a web browser.

   Override the certificate error warning messages, and visit the webpage. It is safe to ignore these error messages for the purposes of this lab. The prompts that you see, and steps required to access the URL, may depend on the web browser you use.



Certificate error https://partspuppetmasterek01.northeurope.cloudapp.azure.com/

This site is not secure

This might mean that someone's trying to fool you or steal any info you send to the server. You should close this site immediately.

Go to your Start page

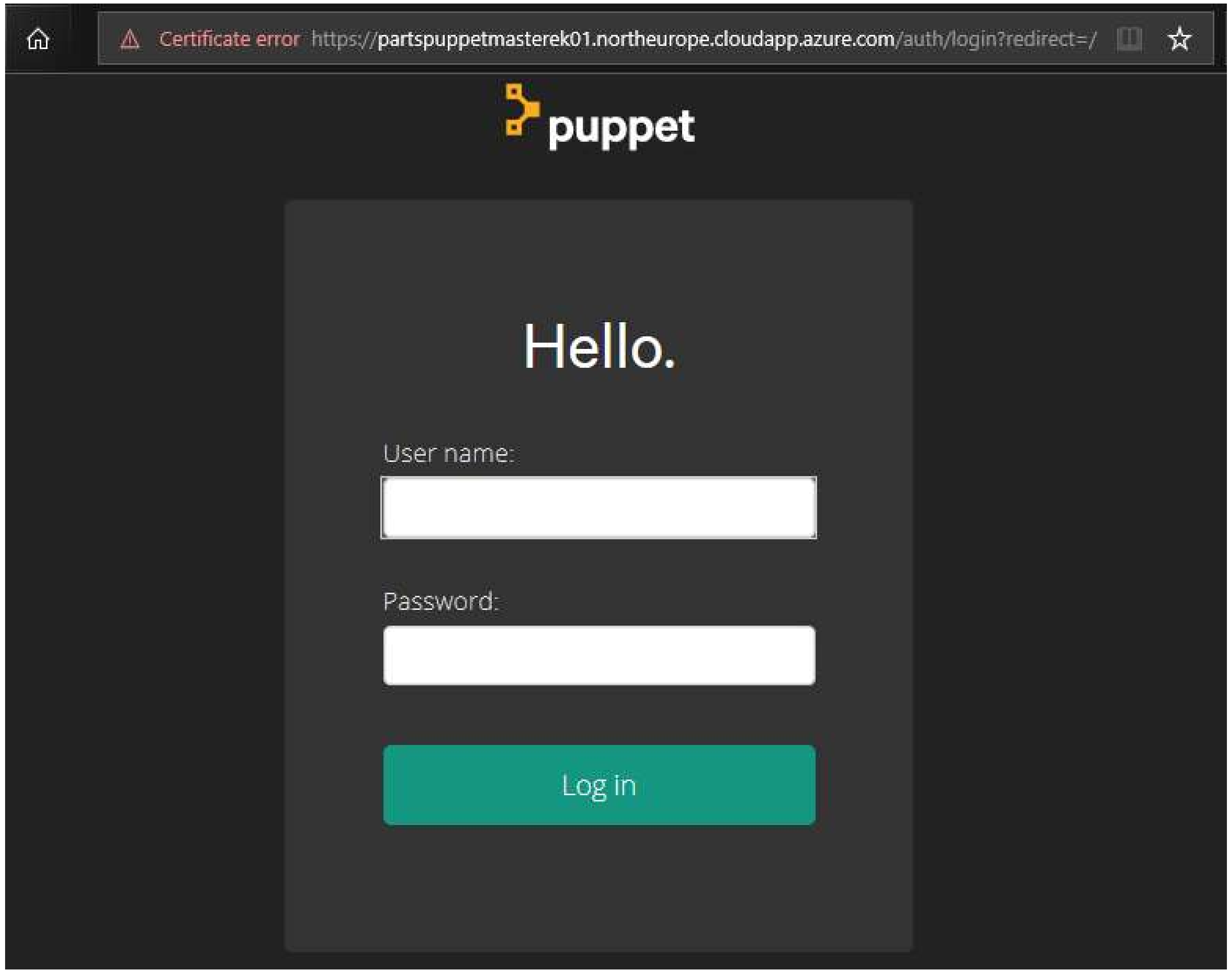
Details

Your PC doesn't trust this website's security certificate.
The hostname in the website's security certificate differs from the website you are trying to visit.

```
Error Code: DLG_FLAGS_INVALID_CA
DLG_FLAGS_SEC_CERT_CN_INVALID
```

Go on to the webpage (Not recommended)

8. If the Puppet configuration has succeeded, you should see the **Puppet Master Console sign in webpage**.


Certificate error https://partspuppetmasterek01.northeurope.cloudapp.azure.com/auth/login?redirect=/
puppet
Hello.
User name:
Password:
Log in

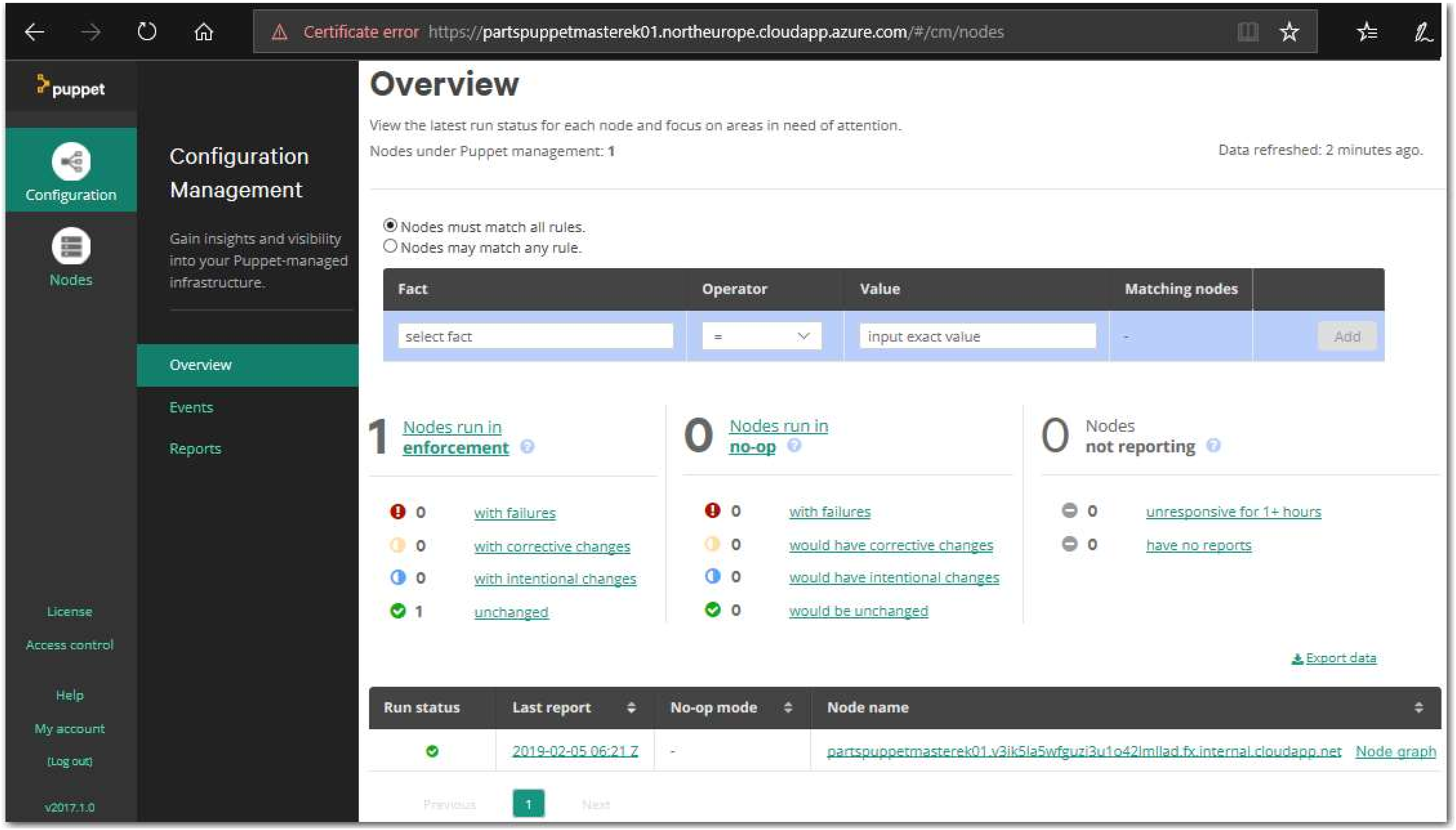
Note: The lab requires several ports to be open, such as the Puppet Server port, the Puppet Console port, SSH ports, and the PU MRP app port on the `partsmrp` VM. The ARM template opens these ports on the VMs for you. You can look through the deployment JSON template to view the port configuration details in the PuppetPartsUnlimitedMRP.json file on GitHub.

9. Log in to the **Puppet Master Console** with the following credentials.

   - **user name** = `admin`
   - **Password** = Pm Console Password you specified earlier in this lab. For example, `Passw0rd0134`

   Note: You cannot log into the Puppet Master Console with the username and password you specified earlier in this lab. *You must login using the built in `admin` account* instead, as shown.

10. If your log in is successful, you will be redirected to the **Puppet Configuration Management Console** webpage which is similar in appearance to the following screenshot.


Certificate error https://partspuppetmasterek01.northeurope.cloudapp.azure.com/#/cm/nodes
puppet
Configuration
Nodes
License
Access control
Help
My account
(Log out)
v2017.1.0
Configuration Management
Gain insights and visibility into your Puppet-managed infrastructure.
Overview
Events
Reports

Overview

View the latest run status for each node and focus on areas in need of attention.

Nodes under Puppet management: **1**

Data refreshed: 2 minutes ago.

● Nodes must match all rules.
○ Nodes may match any rule.

Fact	Operator	Value	Matching nodes	
select fact	=	input exact value	-	Add

1 Nodes run in **enforcement**

❗	0	with failures
◐	0	with corrective changes
◐	0	with intentional changes
✓	1	unchanged

0 Nodes run in **no-op**

❗	0	with failures
◐	0	would have corrective changes
◐	0	would have intentional changes
✓	0	would be unchanged

0 Nodes **not reporting**

⊖	0	unresponsive for 1+ hours
⊖	0	have no reports

Export data

Run status	Last report	No-op mode	Node name
✓	2019-02-05 06:21 Z	-	partspuppetmasterek01.v3ik5la5wfguzi3u1o42lmllad.fx.internal.cloudapp.net Node graph

Previous 1 Next

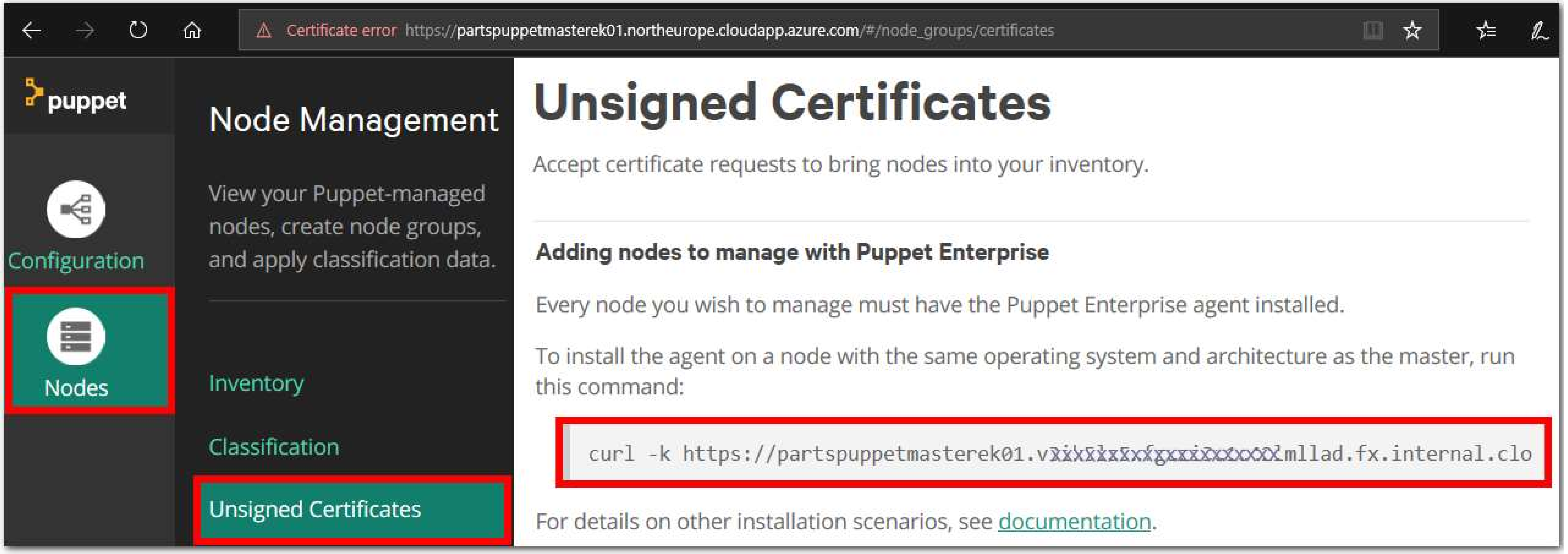
Task 2: Install Puppet Agent on the node

You are now ready to add the Node to the Puppet Master. Once the Node is added, the Puppet Master will be able to configure the Node.

1. Get the Puppet Master internal DNS name. On the **Puppet Configuration Management Console** webpage, go to **Nodes** > **Unsigned Certificates**. The page that loads will show an `Add node` command. Make a note of the command, we will run it in Step 4 of this lab task.

   For example, the Puppet Master machine name is `partspuppetmasterek01.northeurope.cloudapp.azure.com`, and the command we need to run will be similar to the following command.

```
curl -k https://partspuppetmasterek01.irblmudbrloe5hz001blu2g34f.ax.internal.cloudapp.net:8140/packages/current/install.bash | sudo bash
```


Certificate error https://partspuppetmasterek01.northeurope.cloudapp.azure.com/#/node_groups/certificates
puppet
Configuration
Nodes
Inventory
Classification
Unsigned Certificates

Node Management

View your Puppet-managed nodes, create node groups, and apply classification data.

Unsigned Certificates

Accept certificate requests to bring nodes into your inventory.

Adding nodes to manage with Puppet Enterprise

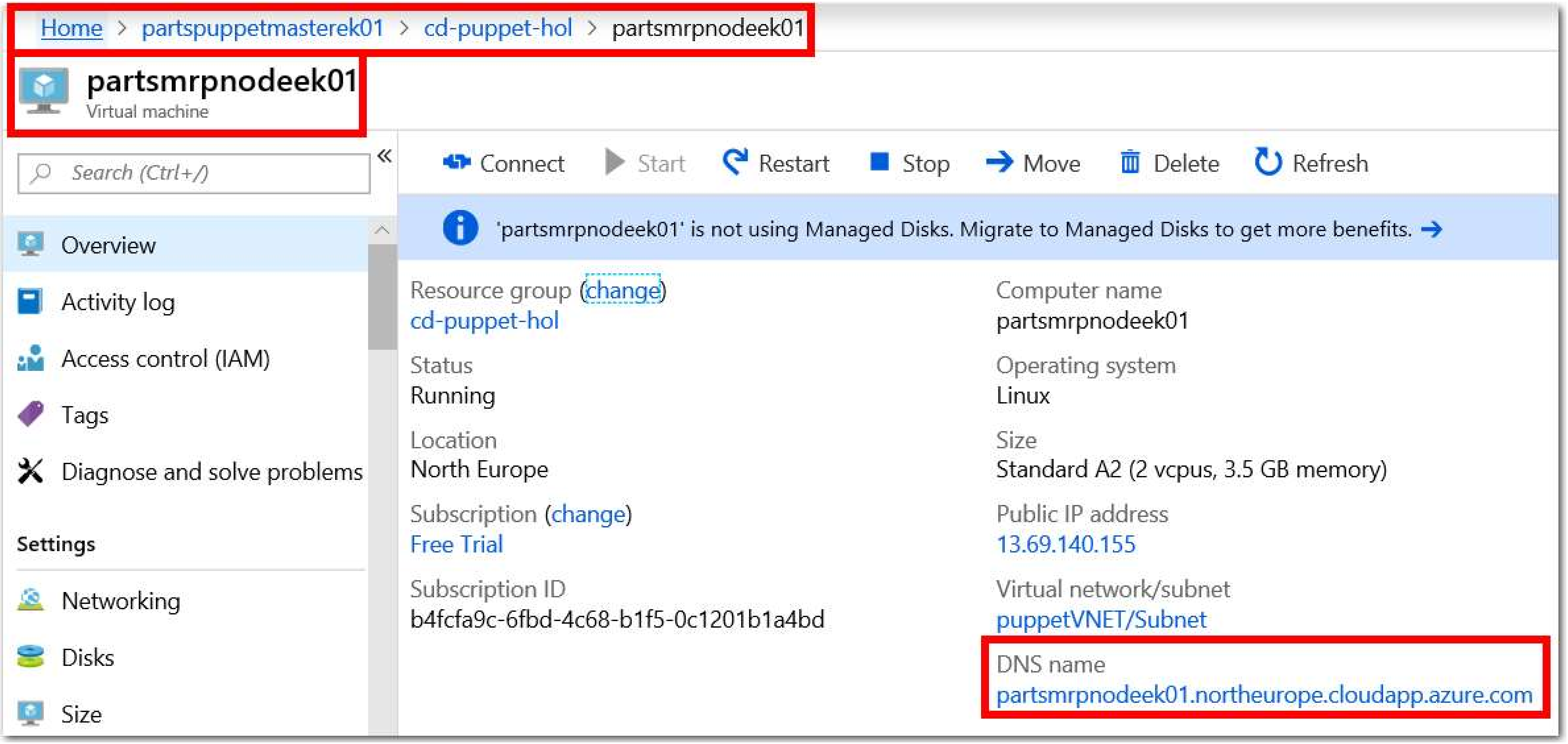
Every node you wish to manage must have the Puppet Enterprise agent installed.

To install the agent on a node with the same operating system and architecture as the master, run this command:

```
curl -k https://partspuppetmasterek01.v[redacted]mllad.fx.internal.clo
```

For details on other installation scenarios, see documentation.

2. In Azure Portal, from the **left menu**, choose **All Resources** > **partsmrpnodeek01** (or whatever name you specified for the Node / partsmrp VM). Make a note of the **DNS name** value for the Node VM.



partsmrpnodeek01
Virtual machine

'partsmrpnodeek01' is not using Managed Disks. Migrate to Managed Disks to get more benefits. →

Resource group (change)
cd-puppet-hol

Status
Running

Location
North Europe

Subscription (change)
Free Trial

Subscription ID
b4fcfa9c-6fbd-4c68-b1f5-0c1201b1a4bd

Computer name
partsmrpnodeek01

Operating system
Linux

Size
Standard A2 (2 vcpus, 3.5 GB memory)

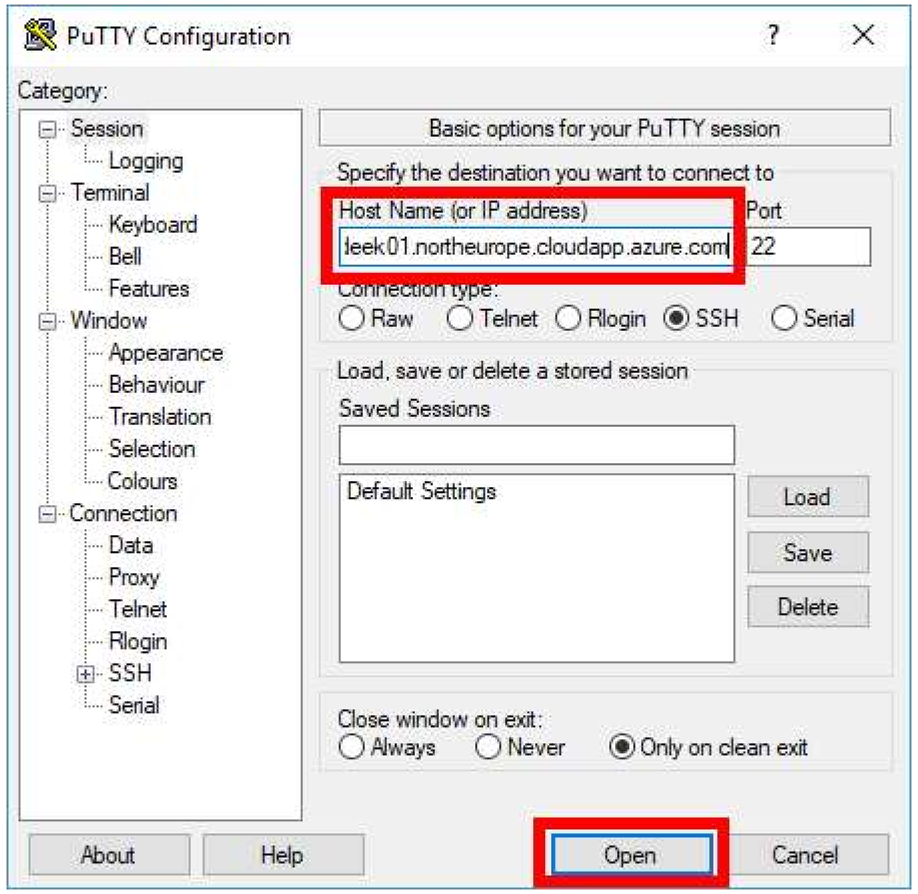
Public IP address
13.69.140.155

Virtual network/subnet
puppetVNET/Subnet

DNS name
partsmrpnodeek01.northeurope.cloudapp.azure.com

3. Establish an SSH connection to the Node VM. In the following example, we will connect to the Node VM using the PuTTy SSH client.

   Specify the Node DNS name as the destination **Host Name**.



Specify the destination you want to connect to

Host Name (or IP address): deek01.northeurope.cloudapp.azure.com

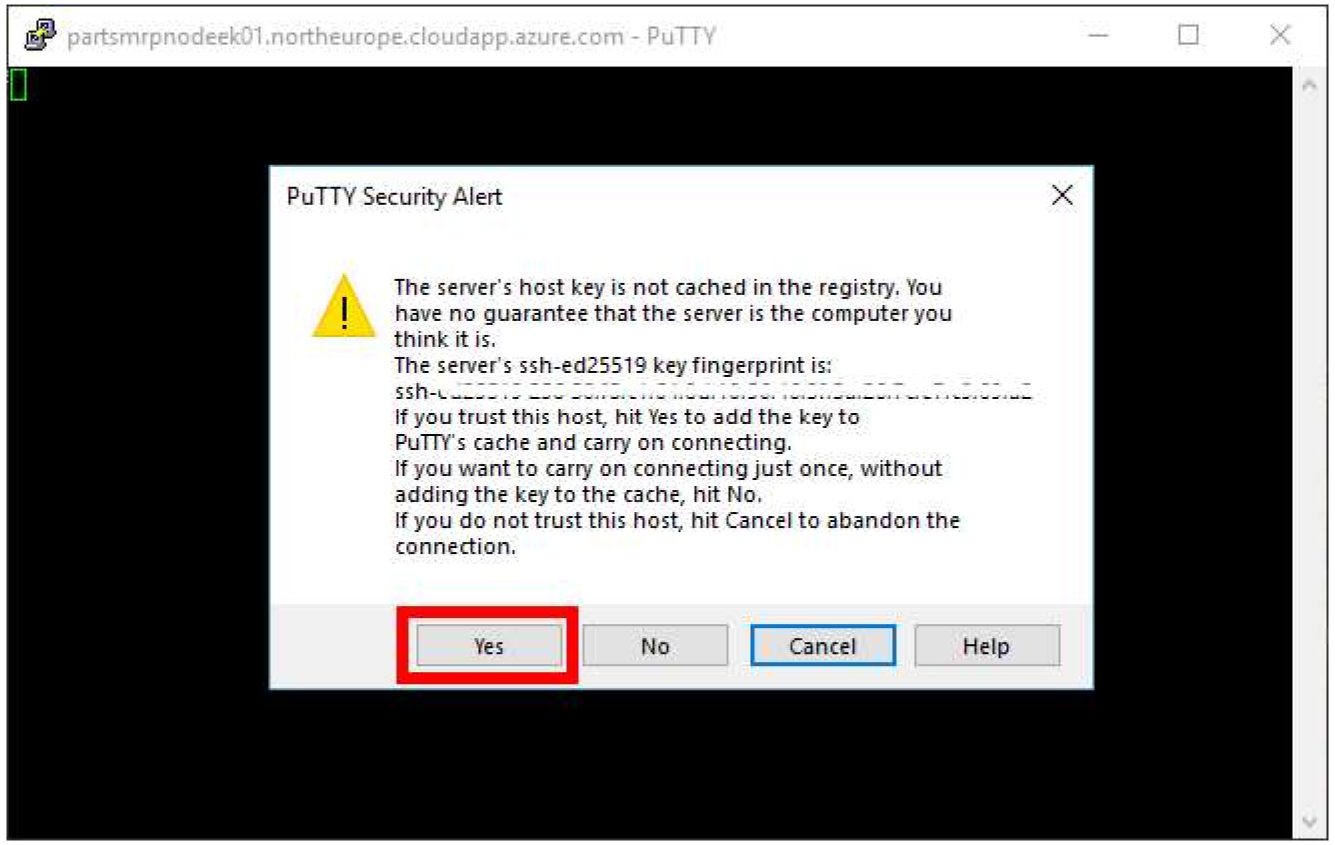
Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions: Default Settings

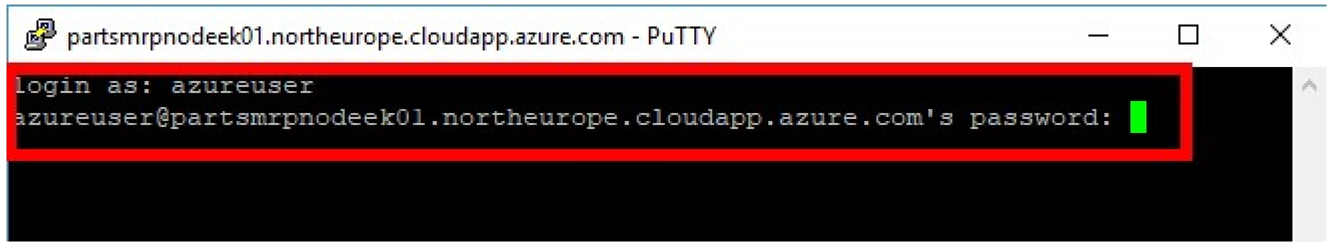
Close window on exit: ○ Always ○ Never ● Only on clean exit

If prompted, choose **Yes** to add the SSH key to PuTTy’s cache.



The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-[redacted]
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

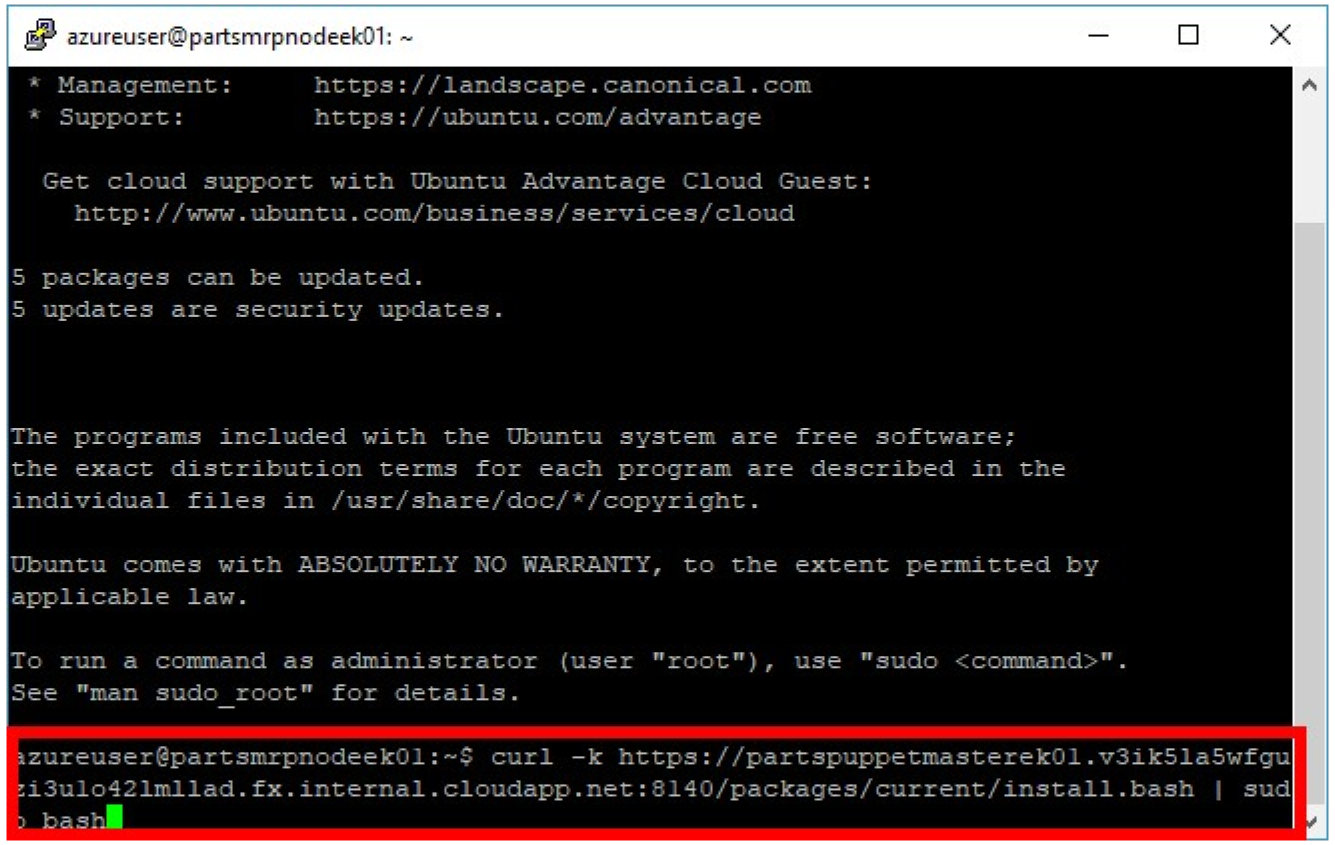
Log in with username and password credentials that you specified in Task 1. For example, `azureuser` and `Passw0rd0134`.



```
login as: azureuser
azureuser@partsmrpnodeek01.northeurope.cloudapp.azure.com's password:
```

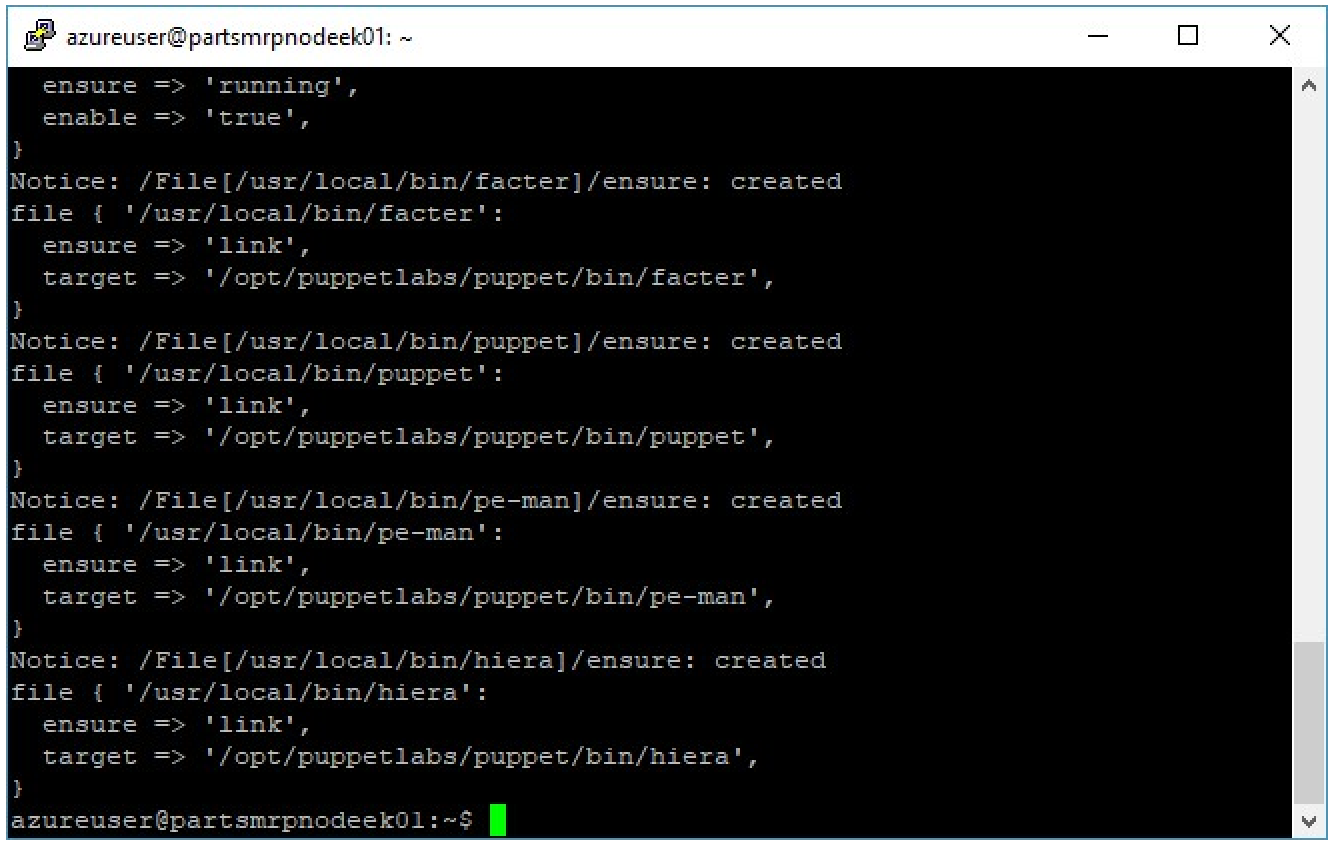
4. Run the `Add Node` command on the node.

   Enter the `Add Node` command into the SSH terminal, which you noted earlier in Step 1. The command begins with `curl...`. Run the command.



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

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

5 packages can be updated.
5 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@partsmrpnodeek01:~$ curl -k https://partspuppetmasterek01.v3ik5la5wfgu
zi3ulo421mllad.fx.internal.cloudapp.net:8140/packages/current/install.bash | sud
o bash
```

Wait for the command to install the Puppet Agent and any dependencies on the Node. The command takes two or three minutes to complete.



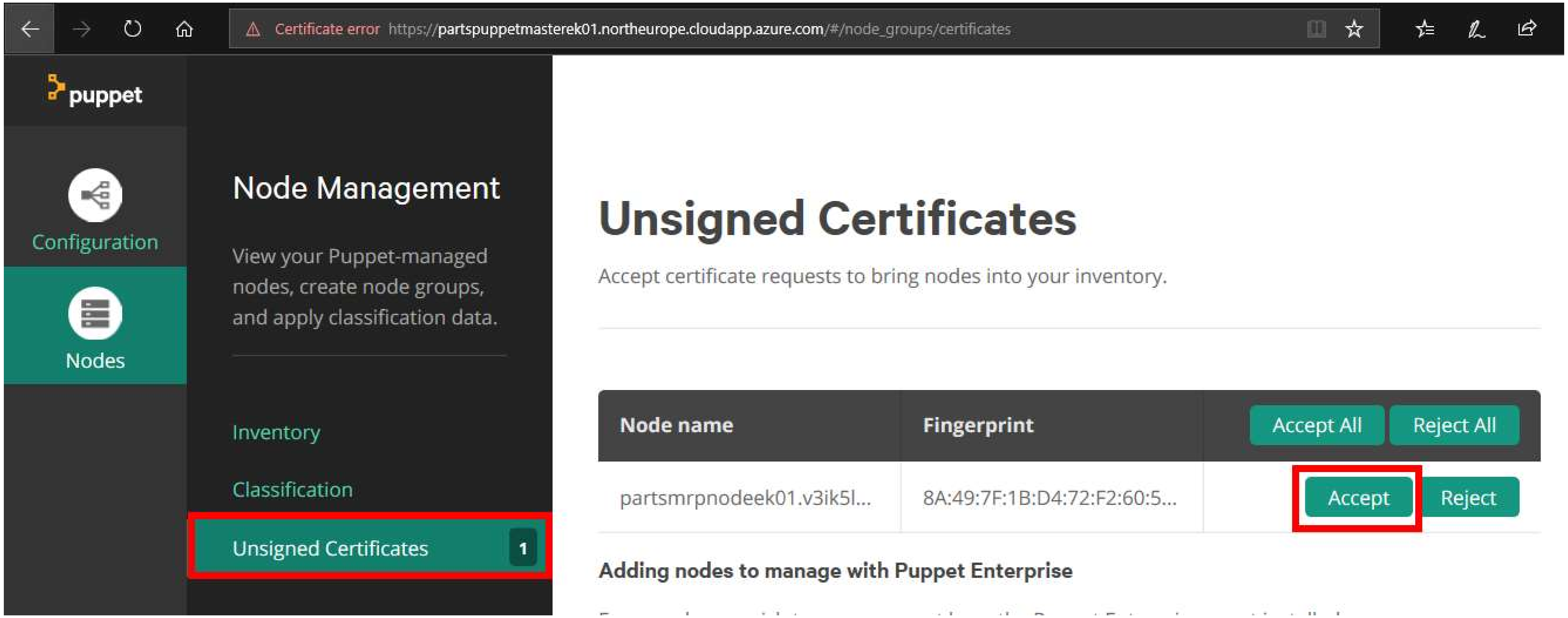
```
  ensure => 'running',
  enable => 'true',
}
Notice: /File[/usr/local/bin/facter]/ensure: created
file { '/usr/local/bin/facter':
  ensure => 'link',
  target => '/opt/puppetlabs/puppet/bin/facter',
}
Notice: /File[/usr/local/bin/puppet]/ensure: created
file { '/usr/local/bin/puppet':
  ensure => 'link',
  target => '/opt/puppetlabs/puppet/bin/puppet',
}
Notice: /File[/usr/local/bin/pe-man]/ensure: created
file { '/usr/local/bin/pe-man':
  ensure => 'link',
  target => '/opt/puppetlabs/puppet/bin/pe-man',
}
Notice: /File[/usr/local/bin/hiera]/ensure: created
file { '/usr/local/bin/hiera':
  ensure => 'link',
  target => '/opt/puppetlabs/puppet/bin/hiera',
}
azureuser@partsmrpnodeek01:~$
```

From here onwards, you will configure the Node from the Puppet Master only.

5. Accept the pending Node request.

   Return to the **Puppet Configuration Management Console**. **Refresh** the **Unsigned Certificates** webpage (where you previously got the Node install command). You should see a pending unsigned certificate request. Choose **Accept** to approve the node.

   This is a request to authorize the certificate between the Puppet Master and the Node, so that they can communicate securely.


Certificate error https://partspuppetmasterek01.northeurope.cloudapp.azure.com/#/node_groups/certificates
puppet
Configuration
Nodes
Inventory
Classification
Unsigned Certificates 1
Accept All
Reject All
Accept
Reject

Node Management

View your Puppet-managed nodes, create node groups, and apply classification data.

Unsigned Certificates

Accept certificate requests to bring nodes into your inventory.

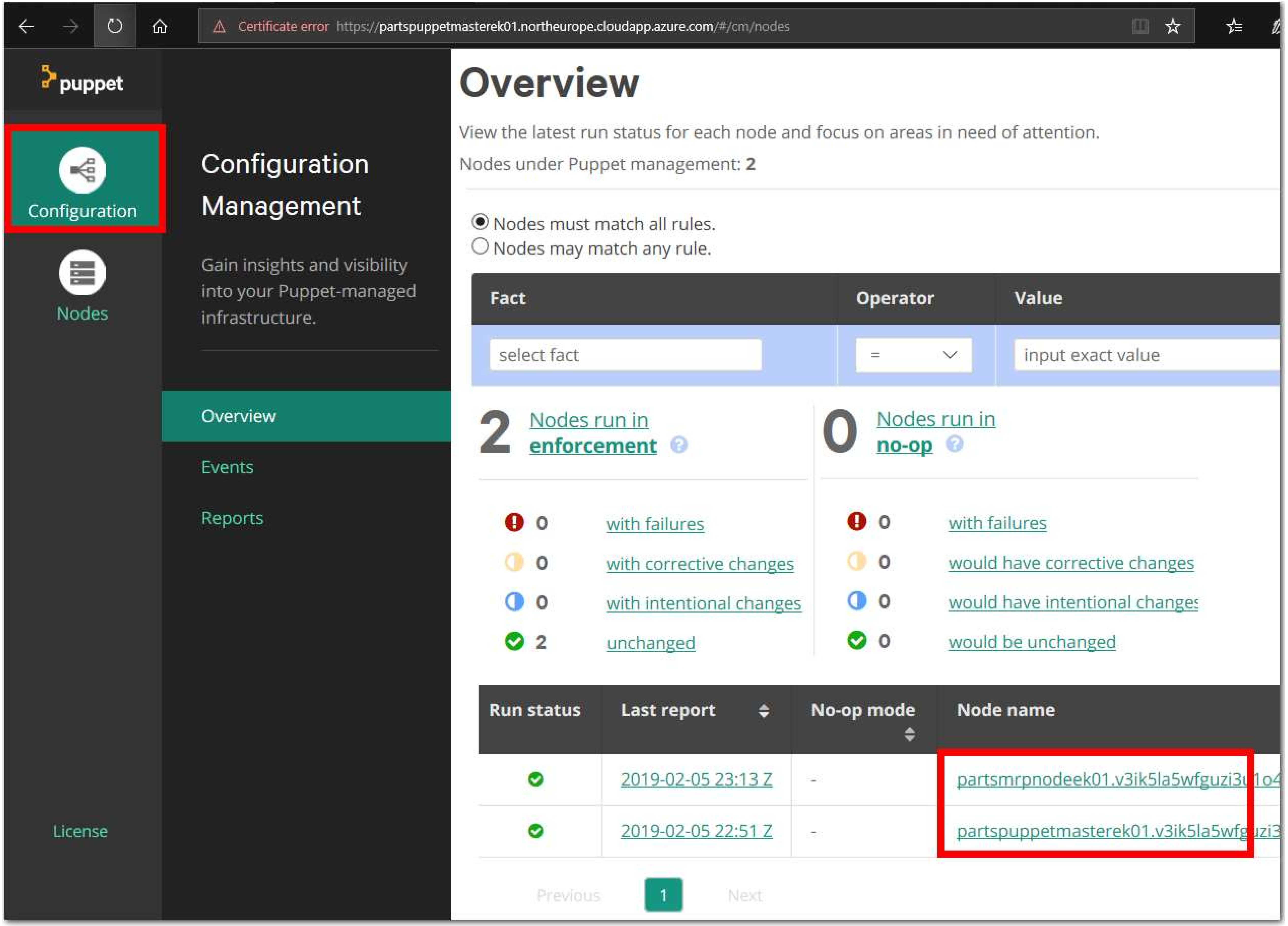
Node name	Fingerprint	
partsmrpnodeek01.v3ik5l...	8A:49:7F:1B:D4:72:F2:60:5...	Accept Reject

Adding nodes to manage with Puppet Enterprise

6. Goto the **Nodes** tab in the Puppet Configuration Management Console. It may take a few minutes to configure the Node / partsmrp VM, before it is visible in the Puppet Configuration Management Console. When the Node is ready, you should see the following nodes listed in the Puppet Configuration Management Console.

   - Puppet Master. For example, `partspuppetmasterek01`
   - Node / partsmrp. For example, `partsmrpnodeek01`

   The nodes are also listed in the Puppet Configuration Management Console under **Configuration** > **Overview**.


Certificate error https://partspuppetmasterek01.northeurope.cloudapp.azure.com/#/cm/nodes
puppet
Configuration
Nodes
License
Configuration Management
Gain insights and visibility into your Puppet-managed infrastructure.
Overview
Events
Reports

Overview

View the latest run status for each node and focus on areas in need of attention.

Nodes under Puppet management: **2**

● Nodes must match all rules.
○ Nodes may match any rule.

Fact	Operator	Value
select fact	=	input exact value

2 Nodes run in **enforcement**

❗	0	with failures
◐	0	with corrective changes
◑	0	with intentional changes
✓	2	unchanged

0 Nodes run in **no-op**

❗	0	with failures
◐	0	would have corrective changes
◑	0	would have intentional changes
✓	0	would be unchanged

Run status	Last report	No-op mode	Node name
✓	2019-02-05 23:13 Z	-	partsmrpnodeek01.v3ik5la5wfguzi3u1o4
✓	2019-02-05 22:51 Z	-	partspuppetmasterek01.v3ik5la5wfguzi3

> **Note:** You can automate the Puppet Agent installation and configuration process on an Azure VM using the Puppet Agent extension from the Azure Marketplace.
>
> There are also a series of PowerShell cmdlets for provisioning, enabling, and disabling the Puppet Extension Handler on Windows VMs. This provides a command-line interface for deploying Puppet Enterprise Agents to Windows VMs in Azure. For details see the Puppet PowerShell Cmdlets for Azure Guide.

Puppet modules and programs

The Parts Unlimited MRP application (PU MRP App) is a Java application. The PU MRP App requires you to install and configure MongoDB and Apache Tomcat on the Node / partsmrp VM. Instead of installing and configuring MongoDB and Tomcat manually, we will write a Puppet Program that will instruct the Node to configure itself.

Puppet Programs are stored in a particular directory on the Puppet Master. Puppet Programs are made up of *manifests* that describe the desired state of the Node(s). The manifests can consume *modules*, which are pre-packaged Puppet Programs. Users can create their own modules or consume modules from a marketplace that is maintained by Puppet Labs, known as The Forge.

Some modules on The Forge are supported officially, others are open-source modules uploaded from the community. Puppet Programs are organized by environment, which allows you to manage Puppet Programs for different environments such as *Dev*, *Test* and *Production*.

For this lab, we will treat the Node as if it were in the Production environment. We will also download modules from The Forge, which we will consume in order to configure the Node.

Task 3: Configure the Puppet Production Environment

The template we used to deploy Puppet to Azure also configured a directory on the Puppet Master for managing the Production environment. The Production Directory is in `/etc/puppetlabs/code/environments/production`.

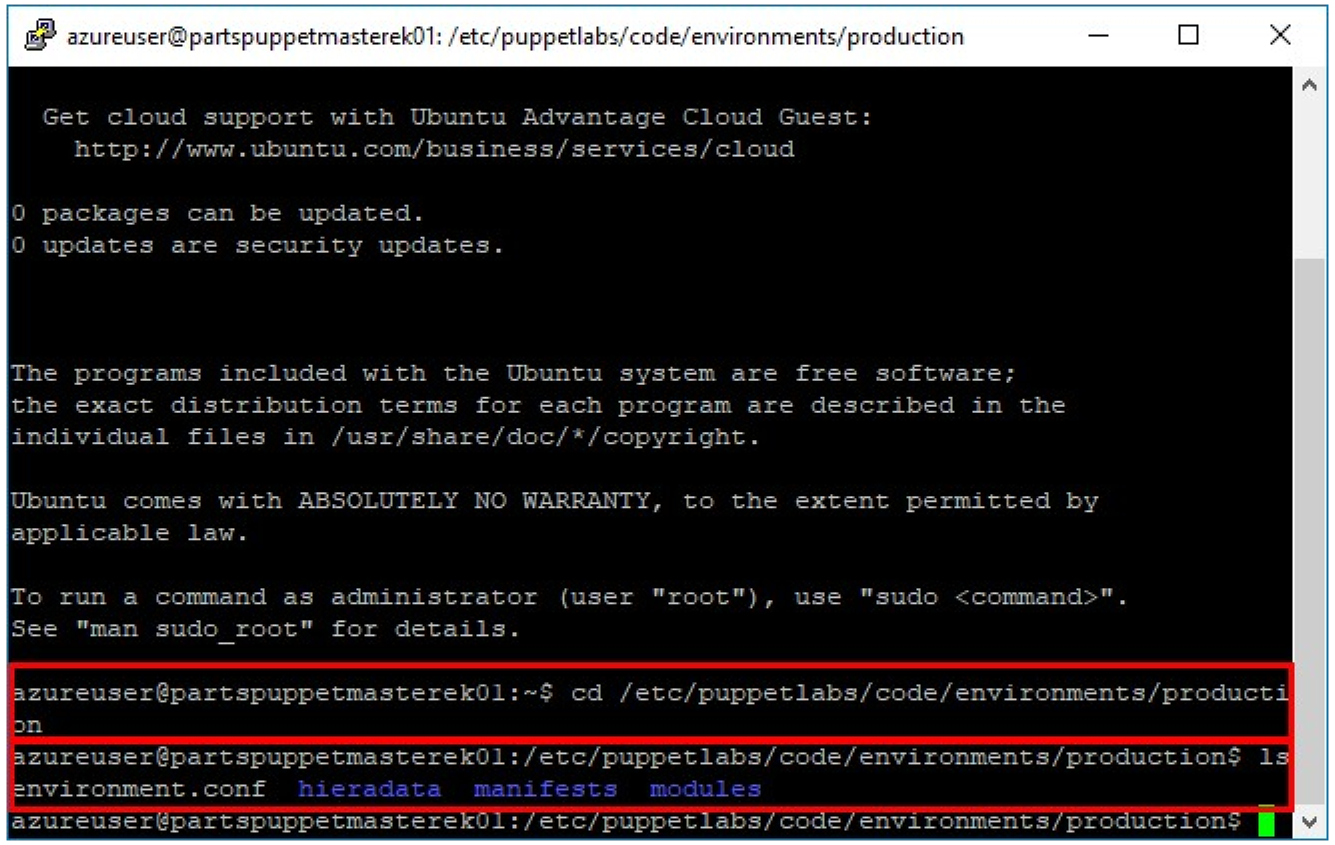
1. Inspect the Production modules.

   Connect to the to the Puppet Master via SSH, with the PuTTy client for example. Use the Change Directory command `cd` to change into the Production Directory `/etc/puppetlabs/code/environments/production`.

   ```
   cd /etc/puppetlabs/code/environments/production
   ```

   Use the list command `ls` to list the contents of the Production Directory. You will see directories named `manifests` and `modules`.

   - The `manifests` directory contains descriptions of machines that we will apply to Nodes later in this lab.
   - The `modules` directory contains any modules that are referenced by the manifests.



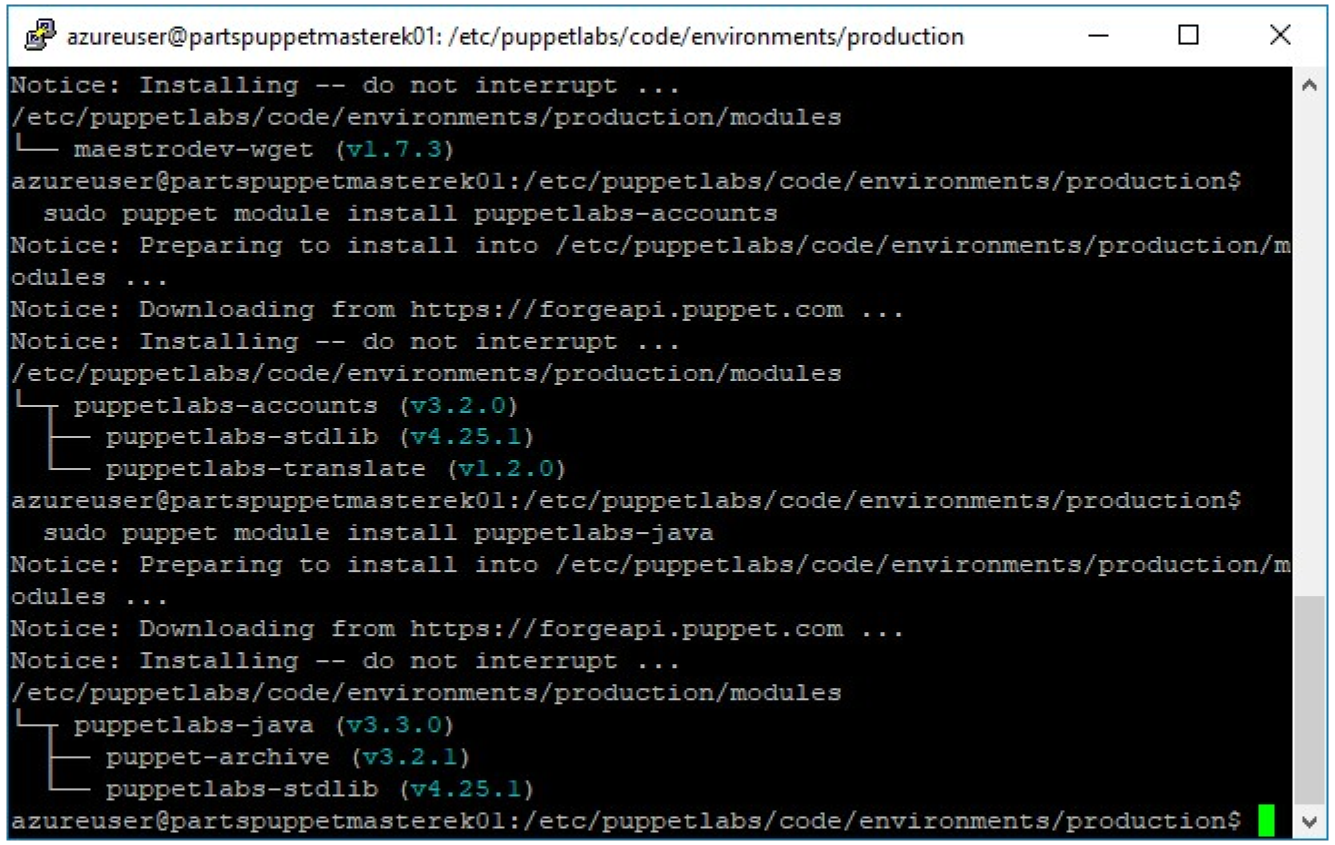
azureuser@partspuppetmasterek01: /etc/puppetlabs/code/environments/production

```
  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

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



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@partspuppetmasterek01:~$ cd /etc/puppetlabs/code/environments/producti
on
azureuser@partspuppetmasterek01:/etc/puppetlabs/code/environments/production$ ls
environment.conf  hieradata  manifests  modules
azureuser@partspuppetmasterek01:/etc/puppetlabs/code/environments/production$
```

2. Install additional Puppet Modules from The Forge.

   We will install modules from The Forge that are needed to configure the Node / partsmrp. Run the following commands in a terminal with an SSH connection to the Puppet Master.

```
sudo puppet module install puppetlabs-mongodb

sudo puppet module install puppetlabs-tomcat

sudo puppet module install maestrodev-wget

sudo puppet module install puppetlabs-accounts

sudo puppet module install puppetlabs-java
```



azureuser@partspuppetmasterek01: /etc/puppetlabs/code/environments/production

```
Notice: Installing -- do not interrupt ...
/etc/puppetlabs/code/environments/production/modules
└── maestrodev-wget (v1.7.3)
azureuser@partspuppetmasterek01:/etc/puppetlabs/code/environments/production$
  sudo puppet module install puppetlabs-accounts
Notice: Preparing to install into /etc/puppetlabs/code/environments/production/m
odules ...
Notice: Downloading from https://forgeapi.puppet.com ...
Notice: Installing -- do not interrupt ...
/etc/puppetlabs/code/environments/production/modules
└─┬ puppetlabs-accounts (v3.2.0)
  ├── puppetlabs-stdlib (v4.25.1)
  └── puppetlabs-translate (v1.2.0)
azureuser@partspuppetmasterek01:/etc/puppetlabs/code/environments/production$
  sudo puppet module install puppetlabs-java
Notice: Preparing to install into /etc/puppetlabs/code/environments/production/m
odules ...
Notice: Downloading from https://forgeapi.puppet.com ...
Notice: Installing -- do not interrupt ...
/etc/puppetlabs/code/environments/production/modules
└─┬ puppetlabs-java (v3.3.0)
  ├── puppet-archive (v3.2.1)
  └── puppetlabs-stdlib (v4.25.1)
azureuser@partspuppetmasterek01:/etc/puppetlabs/code/environments/production$
```

Note: The `mongodb` and `tomcat` modules from The Forge are supported officially. The `wget` module is a user module, and is not supported officially. The `accounts` module provides Puppet with *Classes* for managing and creating users and groups in our Linux VMs. Finally, the `java` module provides Puppet with additional Java functionality.

3. Create a custom module.

   Create a custom module named `mrpapp` in the Production/ Modules Directory on the Puppet Master. The custom module will configure the PU MRP app. Run the following commands in a terminal with an SSH connection to the Puppet Master.

   Use the Change Directory command `cd` to change into the Production/ Modules Directory `/etc/puppetlabs/code/environments/production/modules`.

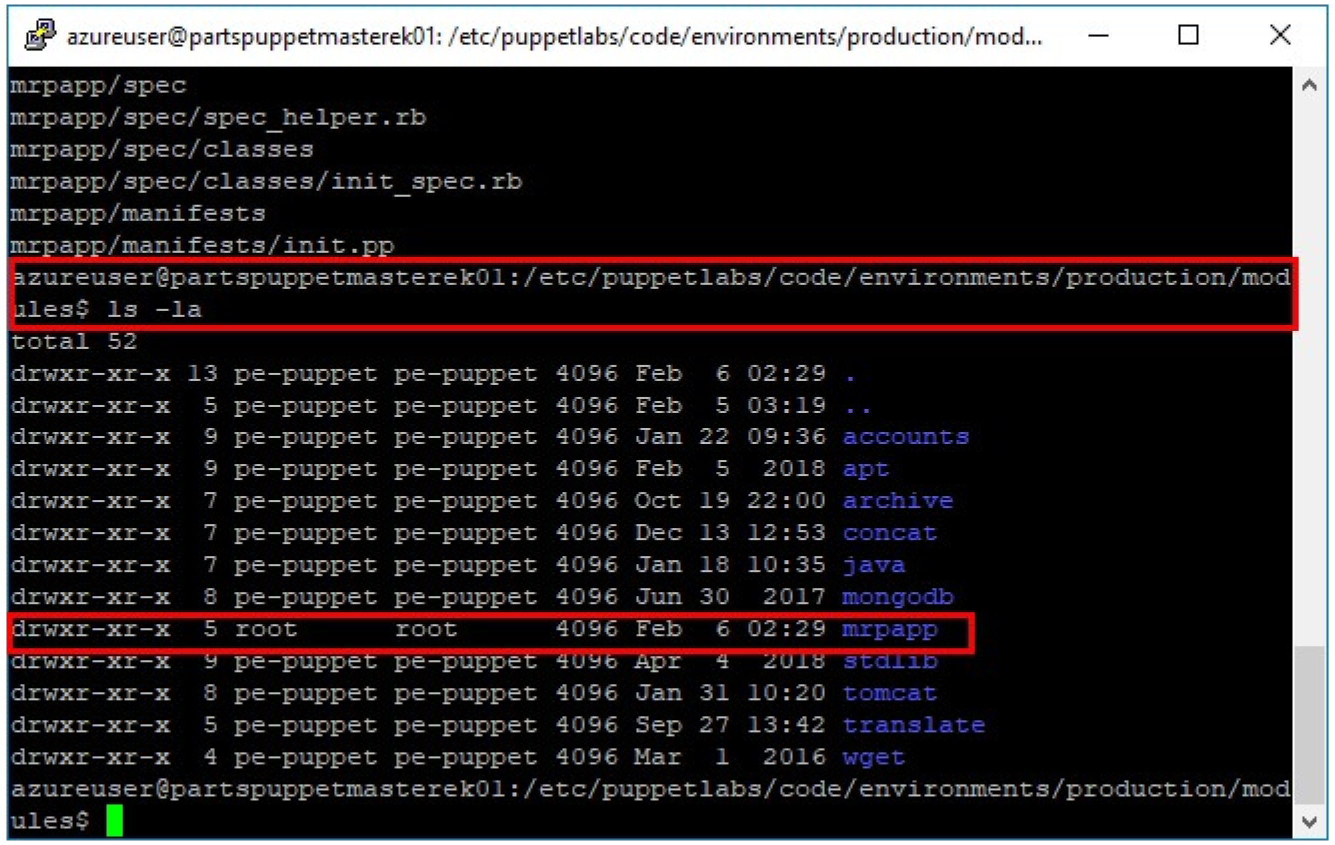
   ```
   cd /etc/puppetlabs/code/environments/production/modules
   ```

   Run the `module generate` commands to create the `mrpapp` module.

   ```
   sudo puppet module generate partsunlimited-mrpapp
   ```

   This will start a wizard that will ask a series of questions as it scaffolds the module. Simply press `enter` for each question (accepting blank or default) until the wizard completes.

   Running list command `ls -la` should show a list of the modules in the directory `~/production/modules`, including the new `mrpapp` module.



azureuser@partspuppetmasterek01: /etc/puppetlabs/code/environments/production/mod...

```
mrpapp/spec
mrpapp/spec/spec_helper.rb
mrpapp/spec/classes
mrpapp/spec/classes/init_spec.rb
mrpapp/manifests
mrpapp/manifests/init.pp
azureuser@partspuppetmasterek01:/etc/puppetlabs/code/environments/production/mod
ules$ ls -la
total 52
drwxr-xr-x 13 pe-puppet pe-puppet 4096 Feb  6 02:29 .
drwxr-xr-x  5 pe-puppet pe-puppet 4096 Feb  5 03:19 ..
drwxr-xr-x  9 pe-puppet pe-puppet 4096 Jan 22 09:36 accounts
drwxr-xr-x  9 pe-puppet pe-puppet 4096 Feb  5  2018 apt
drwxr-xr-x  7 pe-puppet pe-puppet 4096 Oct 19 22:00 archive
drwxr-xr-x  7 pe-puppet pe-puppet 4096 Dec 13 12:53 concat
drwxr-xr-x  7 pe-puppet pe-puppet 4096 Jan 18 10:35 java
drwxr-xr-x  8 pe-puppet pe-puppet 4096 Jun 30  2017 mongodb
drwxr-xr-x  5 root      root      4096 Feb  6 02:29 mrpapp
drwxr-xr-x  9 pe-puppet pe-puppet 4096 Apr  4  2018 stdlib
drwxr-xr-x  8 pe-puppet pe-puppet 4096 Jan 31 10:20 tomcat
drwxr-xr-x  5 pe-puppet pe-puppet 4096 Sep 27 13:42 translate
drwxr-xr-x  4 pe-puppet pe-puppet 4096 Mar  1  2016 wget
azureuser@partspuppetmasterek01:/etc/puppetlabs/code/environments/production/mod
ules$
```

Note: The `ls -la` combined commands will list the contents of a directory (i.e. `ls`), using a long list format (i.e. `-l`), with hidden files shown (i.e. `-a`).

4. The `mrpapp` module will define our Node's configuration.

The configuration of Nodes in the Production environment is defined in a `site.pp` file. The `site.pp` file is located in the Production \ Manifests directory. The `.pp` filename extension is short for *Puppet Program*.

We will edit the `site.pp` file by adding a configuration for our Node.

On your PuTTy session of Master node please run the following.

mkdir /tmp/cem

#git pull

cd /tmp/cem

git clone https://github.com/cemvarol/AZ-400-PuppetLab

cd /tmp/cem/AZ-400-PuppetLab

After download completed please run this.

sudo cp /tmp/cem/AZ-400-PuppetLab/site.pp /etc/puppetlabs/code/environments/production/manifests/site.pp

Note: This will download the edited files for the necessary steps, and set site.pp file as expected.

Task 4: Test the Production Environment Configuration

Before we describe the PU MRP app for the Node fully, test that everything is hooked up correctly by configuring a *dummy* file in the `mrpapp` module. If Puppet executes and creates the dummy file successfully, then everything is configured and working correctly. We can then set up the `mrpapp` module properly.

1. Edit the `init.pp` file.

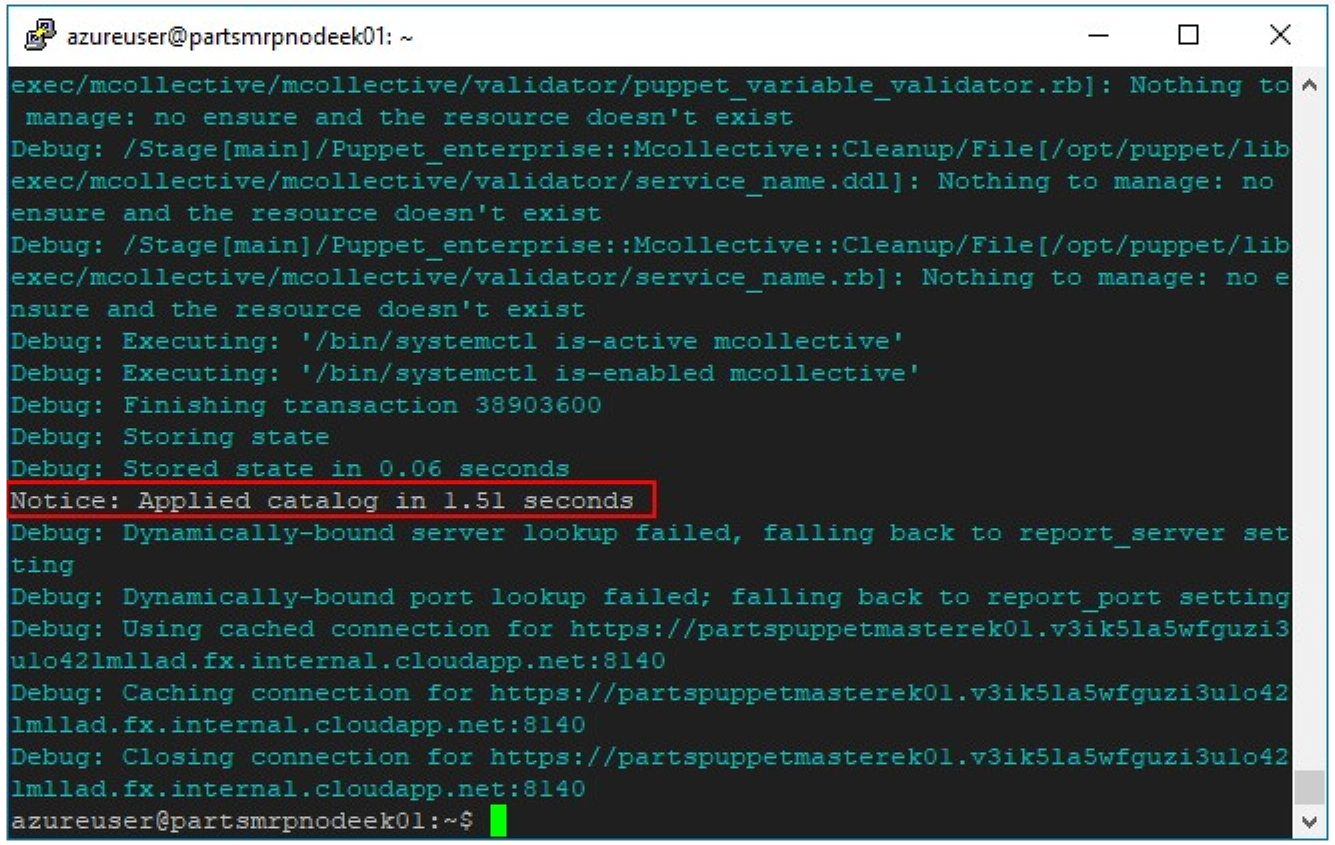
Please run the command below.

sudo cp /tmp/cem/AZ-400-PuppetLab/init.pp /etc/puppetlabs/code/environments/production/modules/mrpapp/manifests/init.pp

2. Test the dummy file.

   To test our setup, establish an SSH connection to the Node / partsmrp VM (using the PuTTy client, for example). Run the following command in an SSH terminal to the Node.

```
sudo puppet agent --test --debug
```



```
exec/mcollective/mcollective/validator/puppet_variable_validator.rb]: Nothing to
 manage: no ensure and the resource doesn't exist
Debug: /Stage[main]/Puppet_enterprise::Mcollective::Cleanup/File[/opt/puppet/lib
exec/mcollective/mcollective/validator/service_name.ddl]: Nothing to manage: no
ensure and the resource doesn't exist
Debug: /Stage[main]/Puppet_enterprise::Mcollective::Cleanup/File[/opt/puppet/lib
exec/mcollective/mcollective/validator/service_name.rb]: Nothing to manage: no e
nsure and the resource doesn't exist
Debug: Executing: '/bin/systemctl is-active mcollective'
Debug: Executing: '/bin/systemctl is-enabled mcollective'
Debug: Finishing transaction 38903600
Debug: Storing state
Debug: Stored state in 0.06 seconds
Notice: Applied catalog in 1.51 seconds
Debug: Dynamically-bound server lookup failed, falling back to report_server set
ting
Debug: Dynamically-bound port lookup failed; falling back to report_port setting
Debug: Using cached connection for https://partspuppetmasterek01.v3ik5la5wfguzi3
ulo42lmllad.fx.internal.cloudapp.net:8140
Debug: Caching connection for https://partspuppetmasterek01.v3ik5la5wfguzi3ulo42
lmllad.fx.internal.cloudapp.net:8140
Debug: Closing connection for https://partspuppetmasterek01.v3ik5la5wfguzi3ulo42
lmllad.fx.internal.cloudapp.net:8140
azureuser@partsmrpnodeek01:~$
```

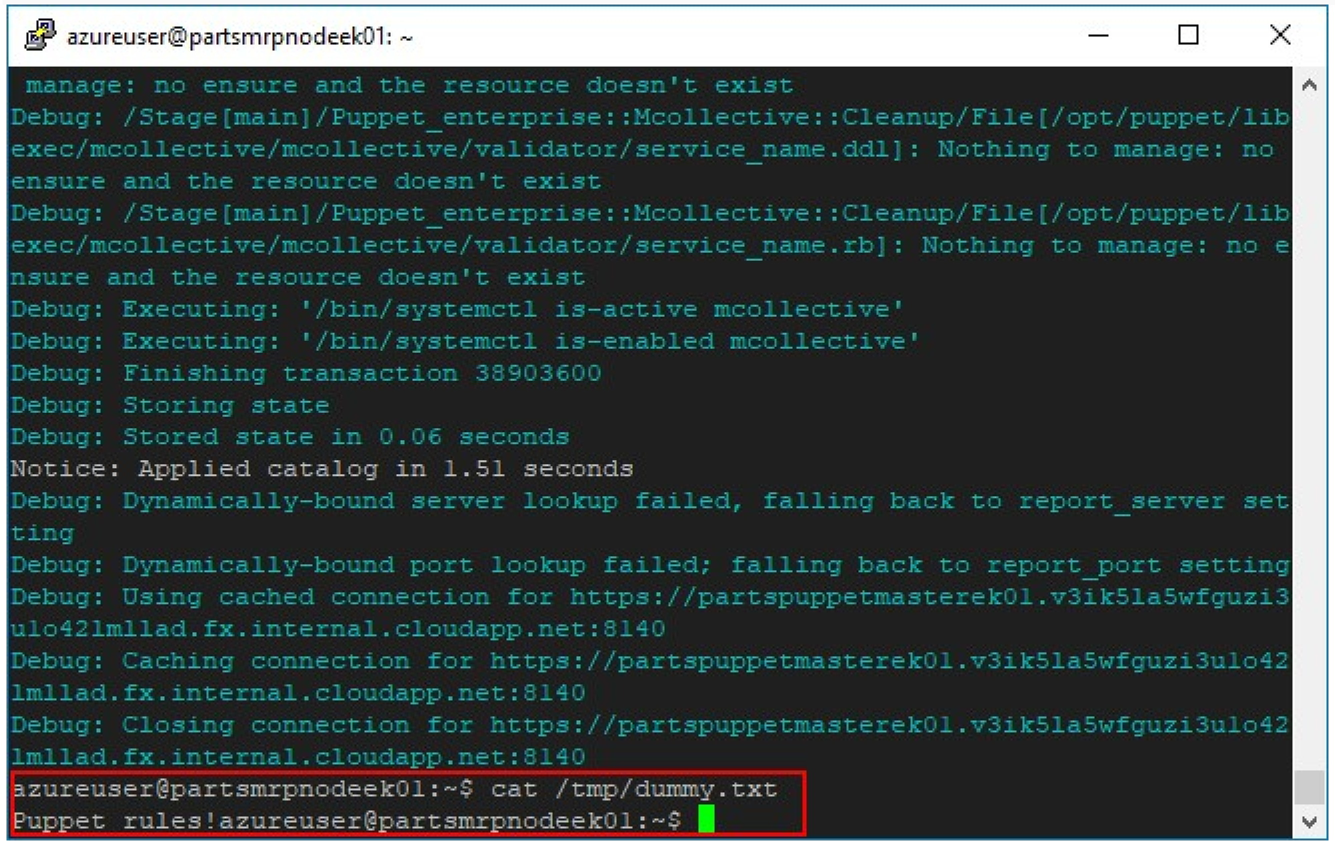
   By default, Puppet Agents query the Puppet Master for their configuration every 30 minutes. The Puppet Agent then tests its current configuration against the configuration specified by the Puppet Master. If necessary, the Puppet Agent modifies its configuration to match the configuration specified by the Puppet Master.

   The command you entered forces the Puppet Agent to query the Puppet Master for its configuration immediately. In this case, the configuration requires the `/tmp/dummy.txt` file, so the Node creates the file accordingly.

You may see more output in your terminal than is shown in the previous screenshot. We used the `--debug` switch for learning purposes, to display more information as the command executes. You can remove the `--debug` switch to receive less text output in the terminal if you wish.

You can also use the `cat` command on the Node, to verify the presence of the file `/tmp/dummy.txt` on the Node, and to inspect the file's contents. The "Puppet rules!" message should be displayed in the terminal.

```
cat /tmp/dummy.txt
```



```
 manage: no ensure and the resource doesn't exist
Debug: /Stage[main]/Puppet_enterprise::Mcollective::Cleanup/File[/opt/puppet/lib
exec/mcollective/mcollective/validator/service_name.ddl]: Nothing to manage: no
ensure and the resource doesn't exist
Debug: /Stage[main]/Puppet_enterprise::Mcollective::Cleanup/File[/opt/puppet/lib
exec/mcollective/mcollective/validator/service_name.rb]: Nothing to manage: no e
nsure and the resource doesn't exist
Debug: Executing: '/bin/systemctl is-active mcollective'
Debug: Executing: '/bin/systemctl is-enabled mcollective'
Debug: Finishing transaction 38903600
Debug: Storing state
Debug: Stored state in 0.06 seconds
Notice: Applied catalog in 1.51 seconds
Debug: Dynamically-bound server lookup failed, falling back to report_server set
ting
Debug: Dynamically-bound port lookup failed; falling back to report_port setting
Debug: Using cached connection for https://partspuppetmasterek01.v3ik5la5wfguzi3
ulo42lmllad.fx.internal.cloudapp.net:8140
Debug: Caching connection for https://partspuppetmasterek01.v3ik5la5wfguzi3ulo42
lmllad.fx.internal.cloudapp.net:8140
Debug: Closing connection for https://partspuppetmasterek01.v3ik5la5wfguzi3ulo42
lmllad.fx.internal.cloudapp.net:8140
azureuser@partsmrpnodeek01:~$ cat /tmp/dummy.txt
Puppet rules!azureuser@partsmrpnodeek01:~$
```

3. Correct configuration drift.

By default, Puppet Agent runs every 30 minutes on the Nodes. Each time the Agent runs, Puppet determines if the environment is in the correct state. If it is not in the correct state, Puppet reapplies Classes as necessary. This process allows Puppet to detect *Configuration Drift*, and fix it.

Simulate Configuration Drift by deleting the dummy file `dummy.txt` from the Node. Run the following command in a terminal connected to the Node to delete the file.

```
sudo rm /tmp/dummy.txt
```

Confirm that the file was deleted from the Node successfully by running the following command on the Node. The command should produce a *No such file or directory* warning message.

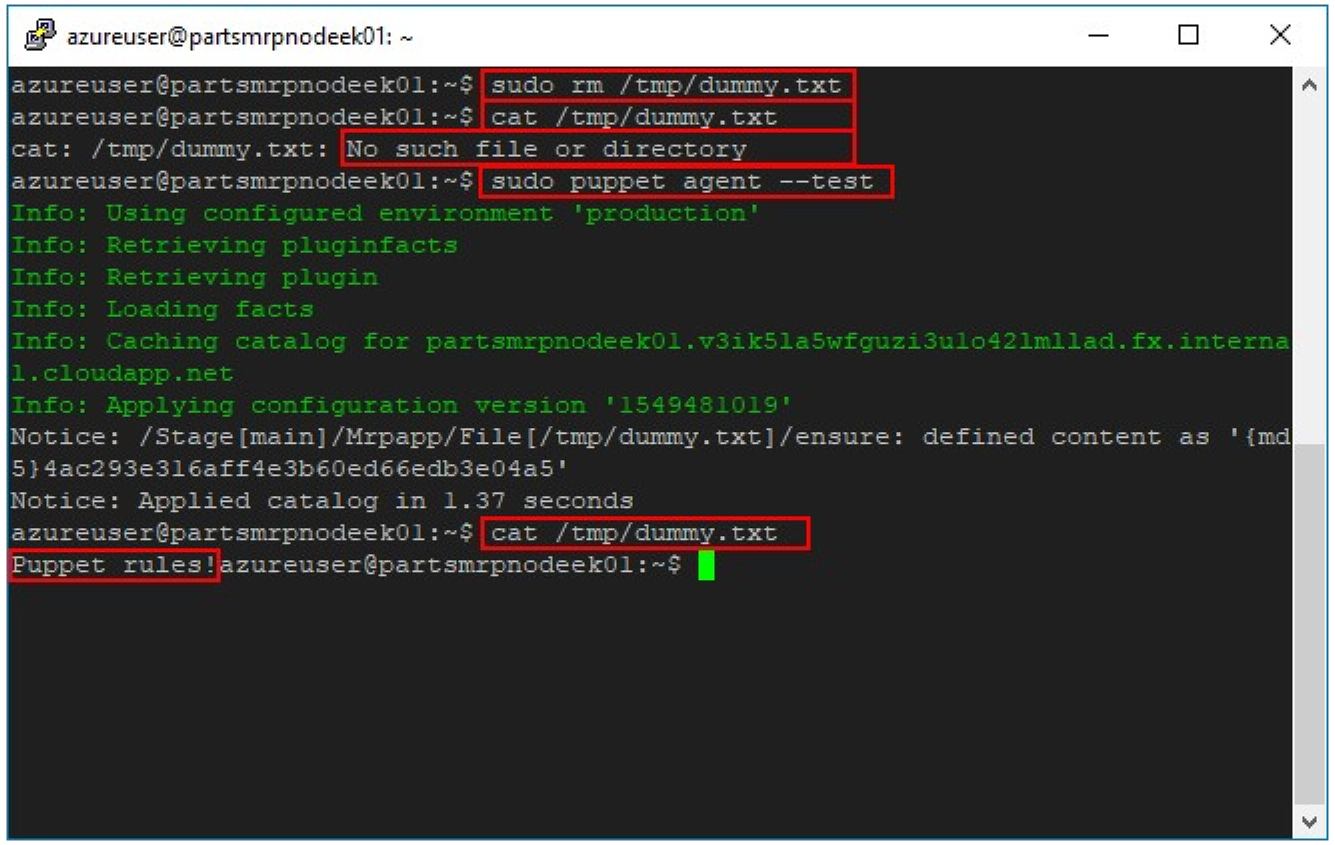
```
cat /tmp/dummy.txt
```

Re-run the Puppet Agent on the Node with the following command.

```
sudo puppet agent --test
```

The re-run should complete successfully, and the file should now exist on the Node again. Verify that the file is present on the Node by running the following command on the Node. Confirm that the "Puppet rules!" message is displayed in the terminal.

```
cat /tmp/dummy.txt
```



```
azureuser@partsmrpnodeek01:~$ sudo rm /tmp/dummy.txt
azureuser@partsmrpnodeek01:~$ cat /tmp/dummy.txt
cat: /tmp/dummy.txt: No such file or directory
azureuser@partsmrpnodeek01:~$ sudo puppet agent --test
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Loading facts
Info: Caching catalog for partsmrpnodeek01.v3ik5la5wfguzi3ulo42lmllad.fx.interna
l.cloudapp.net
Info: Applying configuration version '1549481019'
Notice: /Stage[main]/Mrpapp/File[/tmp/dummy.txt]/ensure: defined content as '{md
5}4ac293e316aff4e3b60ed66edb3e04a5'
Notice: Applied catalog in 1.37 seconds
azureuser@partsmrpnodeek01:~$ cat /tmp/dummy.txt
Puppet rules!azureuser@partsmrpnodeek01:~$
```

You can also edit the contents of the file `dummy.txt` on the Node. Re-run the `sudo puppet agent --test` command, and verify that the contents of the file `dummy.txt` have been reverted to match the configuration specified on the Puppet Master.

Task 5: Create a Puppet Program to describe the prerequisites for the PU MRP app

We have hooked the Node (partsmrp) up to the Puppet Master. Now we can write the Puppet Program to describe the prerequisites for the PU MRP app.

In practice, the different parts of a large configuration solution are typically split into multiple manifests or modules. Splitting the configuration across multiple files is a form of *Modularization*, and promotes better organization and reuse of code.

For simplicity, in this lab, we will describe our entire configuration in a single Puppet Program file `init.pp`, from inside the `mrpapp` module that we created earlier. In Task 5, we will build up our `init.pp` step-by-step.

Task 5.1 Configure MongoDB

Task 5.2 Configure Java

Task 5.3 Create User and Group

Task 5.4 Configure Tomcat

Task 5.5 Deploy a WAR File

Task 5.6 Start the Ordering Service

Task 5.7 Complete the mrpapp Resource

Please run the command below for all 8 tasks above on Master NODE…

sudo cp /tmp/cem/AZ-400-PuppetLab/init2.pp /etc/puppetlabs/code/environments/production/modules/mrpapp/manifests/init.pp

Task 5.8 Configure .war file extracton permissions

Please run the command below for task 5.8

sudo cp /tmp/cem/AZ-400-PuppetLab/war.pp /etc/puppetlabs/code/environments/production/modules/tomcat/manifests/war.pp

Task 6: Run the Puppet Configuration on the Node

1. Re-run the Puppet Agent.

   Return to, or re-establish, your SSH session on the Node/ partsmrp VM. Force Puppet to update the Node’s configuration with the following command.

   ```
   sudo puppet agent --test
   ```

   Note: This first run may take a few moments, as there is a lot to download and install. The next time that you run the Puppet Agent, it will verify that the existing environment is configured correctly. This verification process will take less time than the first run, because the services will be installed and configured already.

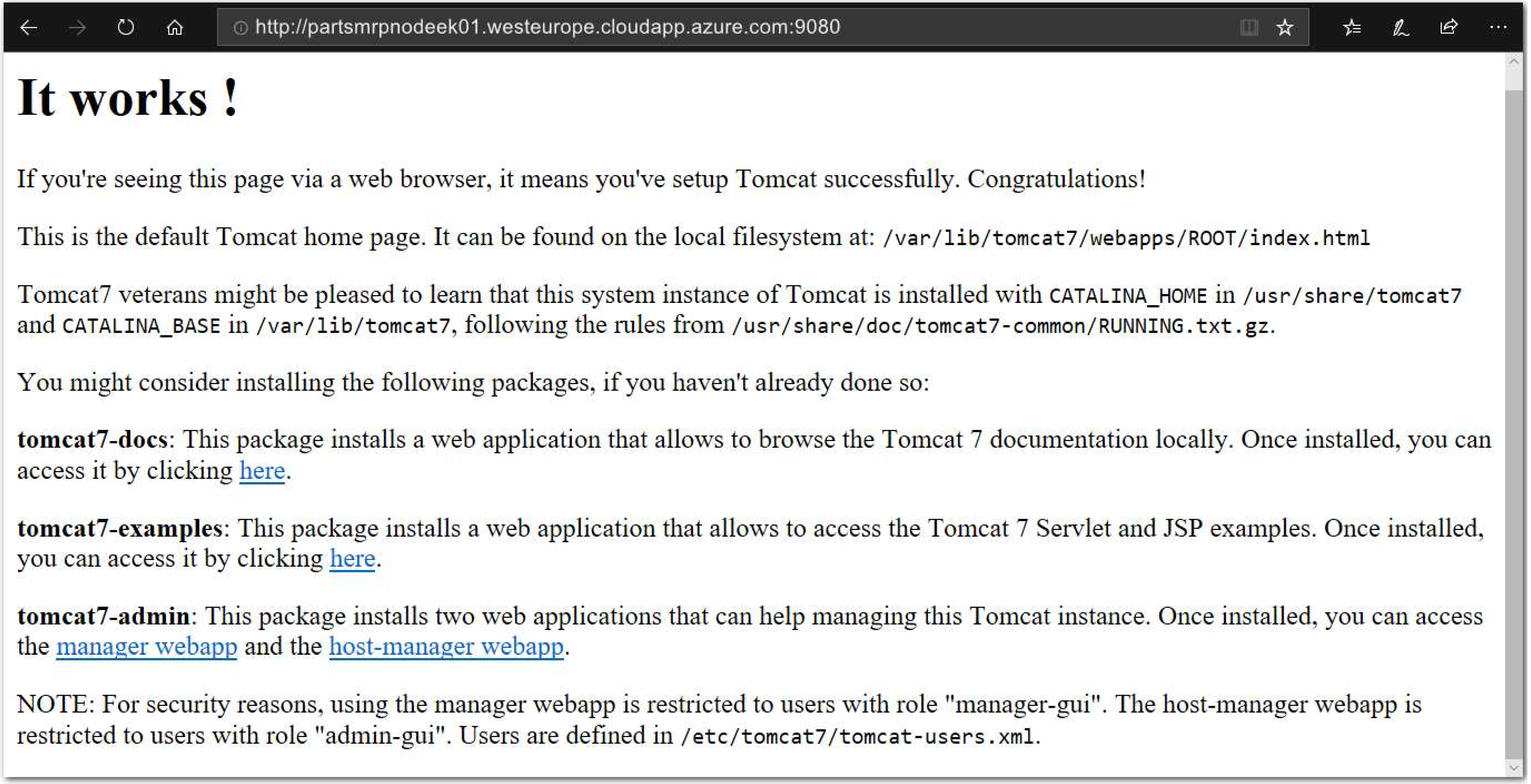
2. Verify that Tomcat is running correctly.

   Append the port number `9080` to the DNS address URL for the Node/ partsmrp VM, for example `http://partsmrpnodeek01.westeurope.cloudapp.azure.com:9080`

   You can get the DNS address URL from the **Public IP resource** for the Node, in **Azure Portal** (just as you did when you got the URL of the Puppet Master earlier).

   Open a web browser and browse to port `9080` on the Node/ partsmrp VM. Once open in the web browser, you should see the **Tomcat Confirmation** webpage.

   Note: Use the `http` protocol, and *not* `https`.



It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat7/webapps/ROOT/index.html`

Tomcat7 veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat7` and `CATALINA_BASE` in `/var/lib/tomcat7`, following the rules from `/usr/share/doc/tomcat7-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

tomcat7-docs: This package installs a web application that allows to browse the Tomcat 7 documentation locally. Once installed, you can access it by clicking here.

tomcat7-examples: This package installs a web application that allows to access the Tomcat 7 Servlet and JSP examples. Once installed, you can access it by clicking here.

tomcat7-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the manager webapp and the host-manager webapp.

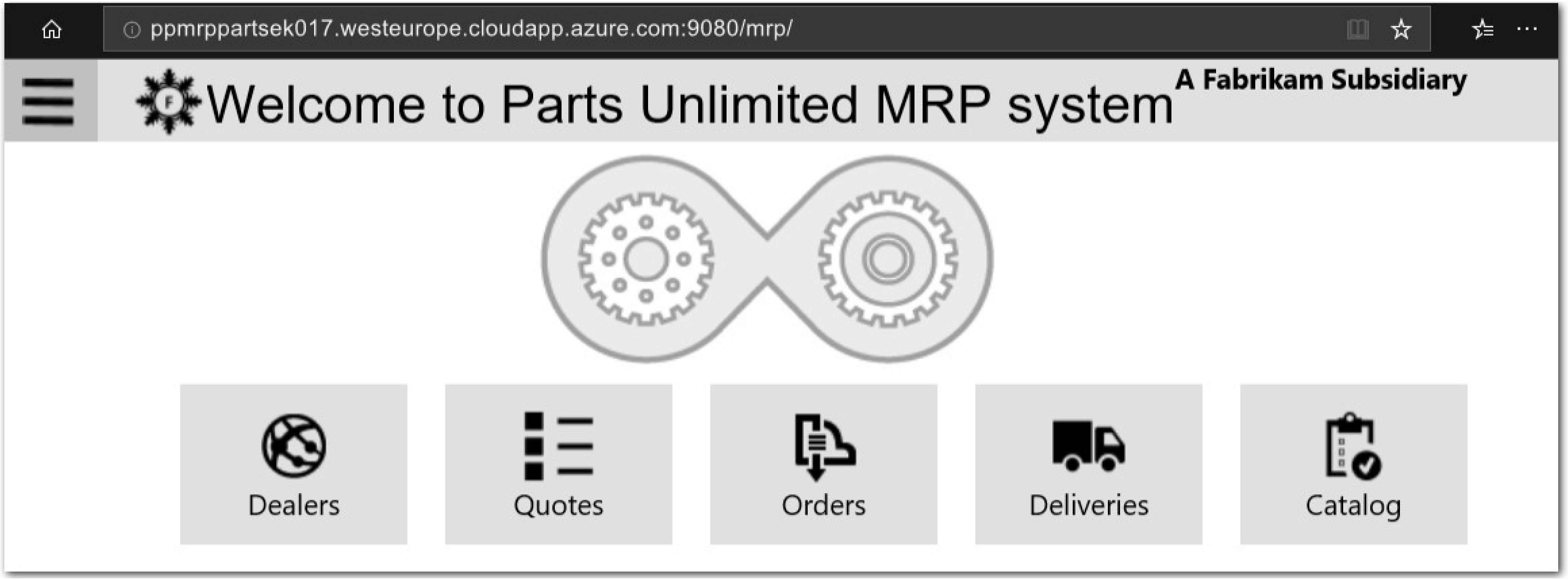
NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted to users with role "admin-gui". Users are defined in `/etc/tomcat7/tomcat-users.xml`.

3. Verify that the PU MRP app is running correctly.

   Check that the configuration is correct by opening a web browser to the PU MRP app. In your web browser, append `/mrp` to the end of DNS address URL you used in Step 2. For example, `http://partsmrpnodeek01.westeurope.cloudapp.azure.com:9080/mrp`.

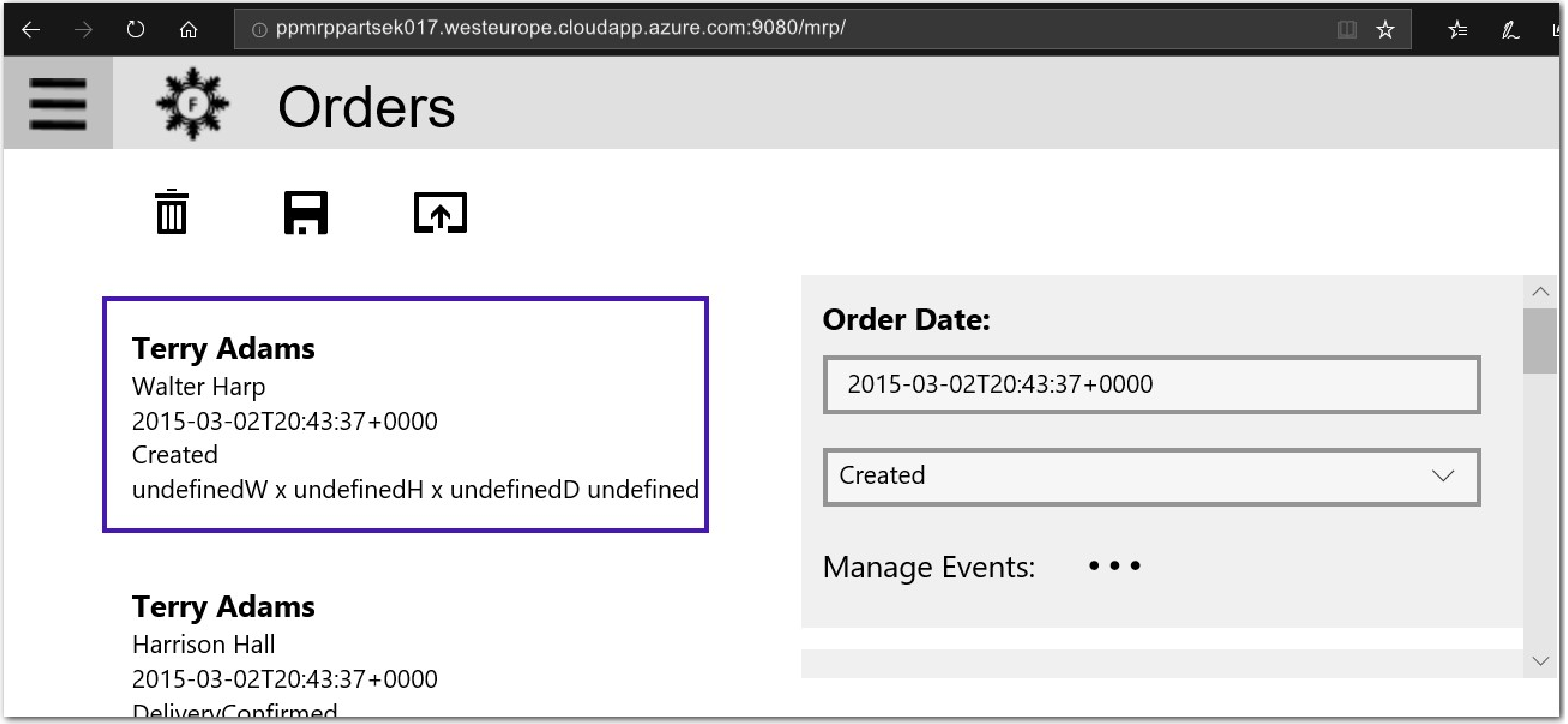
   You can also get the DNS name for the Node/ partsmrp VM in **Azure Portal**.

   The **PU MRP app Welcome** webpage should be displayed in your web browser.



ppmrppartsek017.westeurope.cloudapp.azure.com:9080/mrp/
Welcome to Parts Unlimited MRP system
A Fabrikam Subsidiary
Dealers
Quotes
Orders
Deliveries
Catalog

Explore the PU MRP app to confirm that it functions as intended. For example, select the **Orders** button, in your web browser, to view the **Orders page**.



Terry Adams
Walter Harp
2015-03-02T20:43:37+0000
Created
undefinedW x undefinedH x undefinedD undefined

Terry Adams
Harrison Hall
2015-03-02T20:43:37+0000
DeliveryConfirmed

Order Date:

2015-03-02T20:43:37+0000

Created

Manage Events: • • •