Microsoft Cloud Workshop

Enterprise-ready cloud

Hands-on lab step-by-step

March 2018

Contents

Enterprise-ready cloud hands-on lab step-by-step

Abstract and learning objectives

Create an Azure cloud governance plan to advise a manufacturing company of the features available in Azure. Discover features to bring governance to their Azure deployments, distributed administration, and allowance for secure remote connectivity and development work for their offshore developers. Attendees will be better able to design a governance plan to showcase the security and governance features of Azure and control costs. In addition,

- Provide for cost tracking by business unit, environment, and project
- Provide for a distributed administration model
- Put a service catalog in place to prevent deployment of unsupported Azure services
- Put controls in place to allow deployment of services only in specific regions

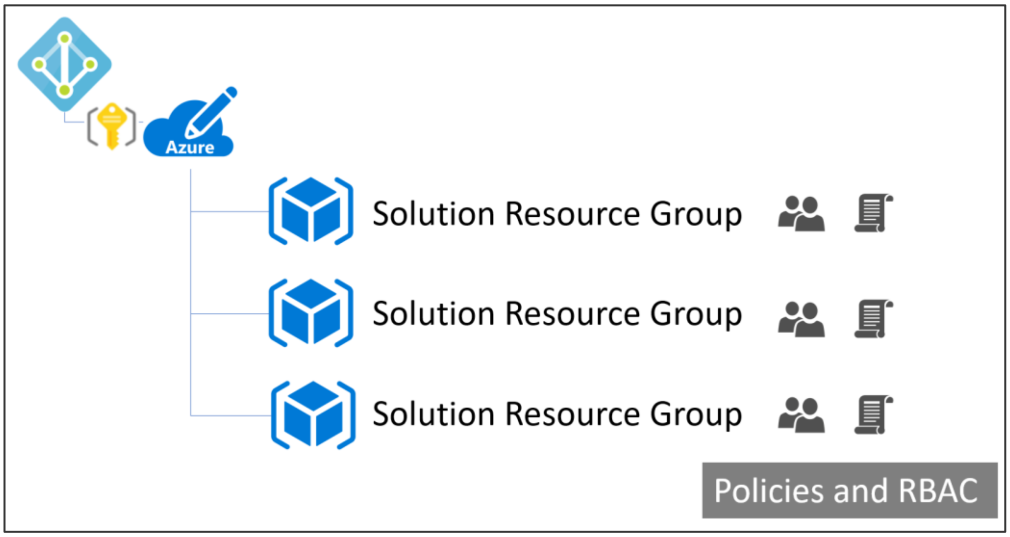
Overview

In this hands-on lab, you are working with Trey Research to setup some best practices regarding policies, permissions, and remote access to their network. Some tasks will include creating scripts that Enterprise IT will use to automatically set policy and delegate permissions when a new subscription is created. You will also help them solve a critical problem for onboarding new developers and controlling access to what they can access on the network.

Requirements

- Local machine or a virtual machine configured with:
    - Visual Studio 2015 or 2017 Community Edition or VS Code
- Full global admin access to the Azure AD tenant associated with your Azure subscription.

Solution architecture



Before the hands-on lab

Duration: 15 minutes

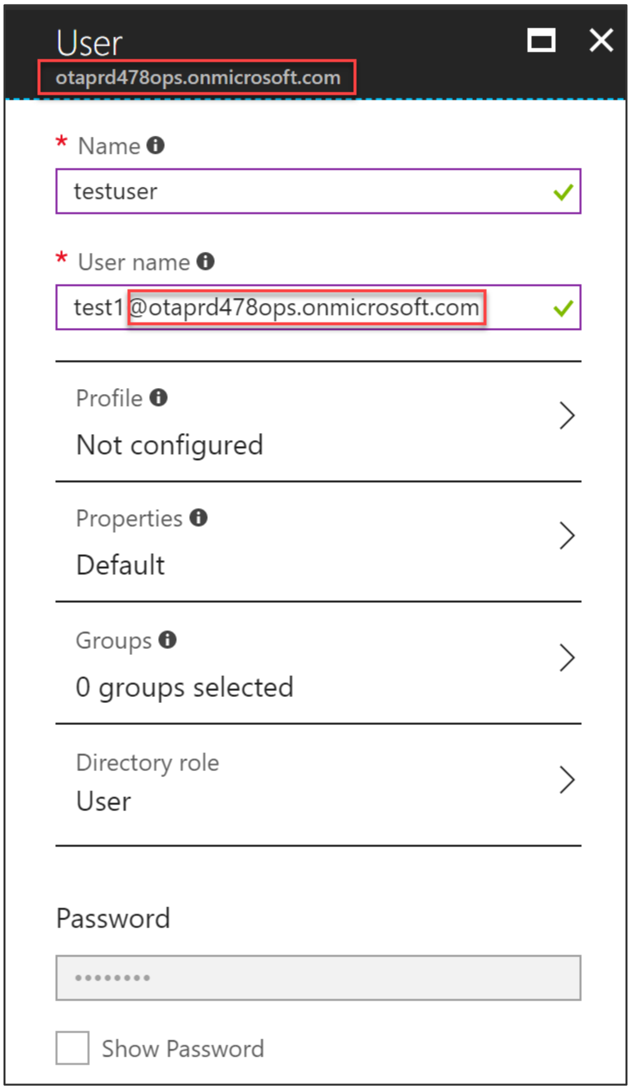
To complete this lab, you must have full global admin access to the Azure AD tenant associated with your Azure subscription.

Task 1: Validate global admin access to Azure AD tenant

1. Login to http://portal.azure.com, click on **All Services**, and type in **Azure Active Directory**.

2. Open the Azure Active Directory tenant. Click Users -> New User.

3. Create a new user called testuser/test1. Use the tenant name on the header for the domain name of the email.



User
otaprd478ops.onmicrosoft.com

* Name: testuser

* User name: test1@otaprd478ops.onmicrosoft.com

Profile: Not configured

Properties: Default

Groups: 0 groups selected

Directory role: User

Password: ••••••••

□ Show Password

If you can create the user, you will enough permissions in Azure AD to complete the lab. If you cannot, you will need more permissions before proceeding.

Task 2: Setup a development environment

If you do not have a machine set up with Visual Studio complete this task, and use this VM to complete the remainder of the Lab.

1. Create a DS2_V2 Azure Virtual Machine using the Visual Studio Community 2017 image from the Azure Marketplace.

Exercise 1: Create the policy for Enterprise IT

Duration: 60 minutes

In this exercise, you first create a Management Group for your Azure subscription(s). You will apply several of the built-in Azure Policy definitions to that Management Group to ensure that users stay within the scope of supported services for Enterprise IT. Finally, you will create a new policy initiative defining a multi-resource naming convention, and apply that initiative to the Management Group.

Help references

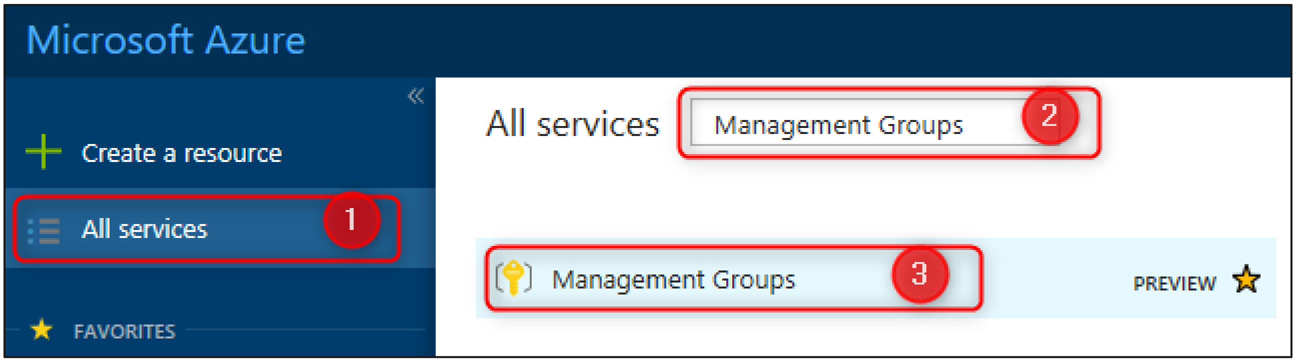
Azure Policy	https://docs.microsoft.com/azure/azure-policy/azure-policy-introduction
Azure Management Groups	https://docs.microsoft.com/azure/azure-resource-manager/management-groups-overview

Task 1: Create a Management Group

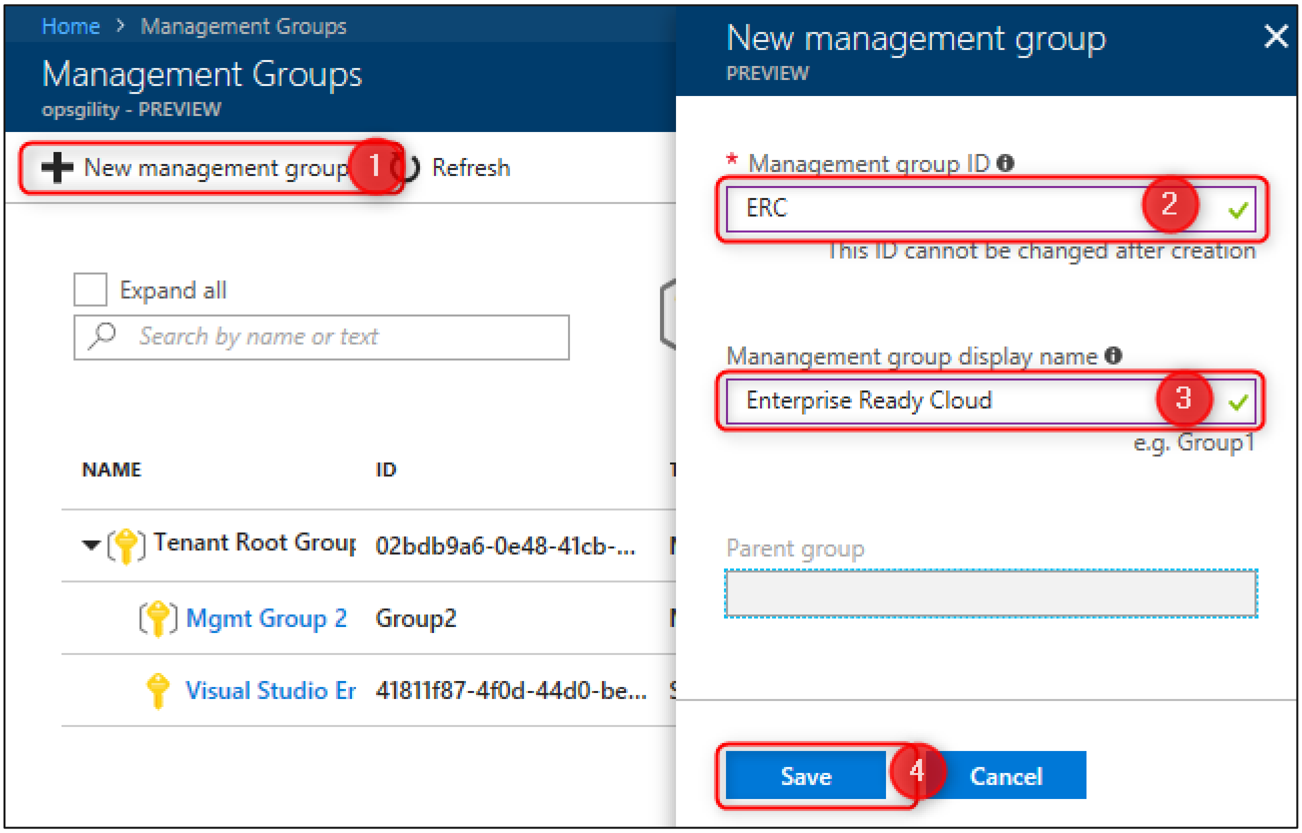
In this Task, you will create a new Management Group, and move a subscription into this Management Group. We'll later assign Azure Policy using the Management Group scope, so that it applies automatically to all subscriptions under that scope.

Note: We'll use our own Management Group, if you have permissions you could also use the Tenant Root Management Group.

1. Launch the Azure Management Portal, and navigate to **Management Groups** under **All services**:



2. Click **New management group**, then fill in the management group ID and display name (we'll use 'ERC' as the management group ID). Leave the Parent group blank (so our new group will sit under the Tenant Root Management Group), and click **Save**.



Home > Management Groups

Management Groups

opsgility - PREVIEW

NAME	ID
Tenant Root Grouį	02bdb9a6-0e48-41cb-...
Mgmt Group 2	Group2
Visual Studio Er	41811f87-4f0d-44d0-be...

New management group

PREVIEW

* Management group ID: ERC

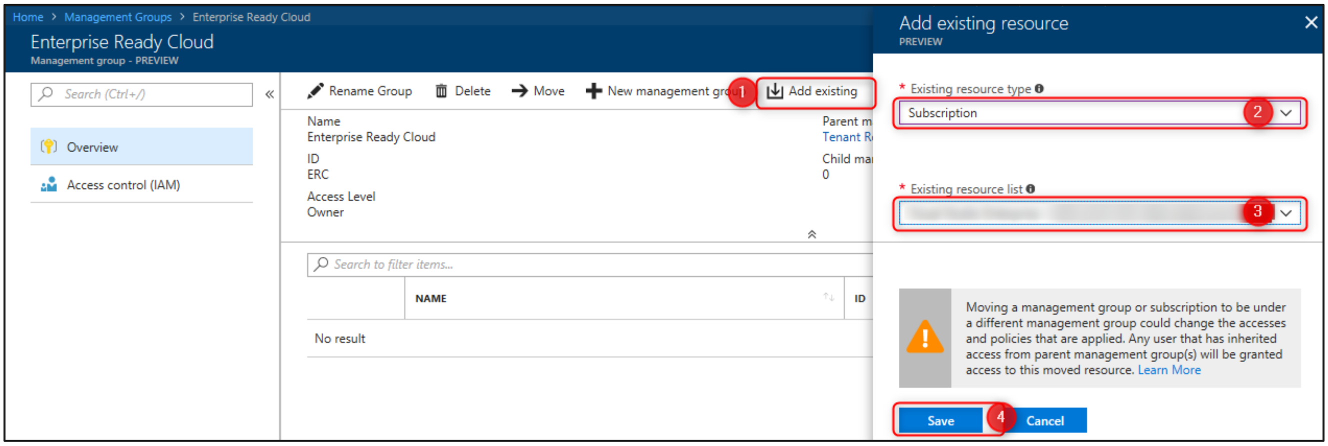
This ID cannot be changed after creation

Manangement group display name: Enterprise Ready Cloud

e.g. Group1

Parent group:

3. Click on the newly-created management group, then click **Add existing**. Select 'Subscription' as the existing resource type, and choose your subscription from the drop-down list, then click **Save**.



Home > Management Groups > Enterprise Ready Cloud

Enterprise Ready Cloud

Management group - PREVIEW

Name
Enterprise Ready Cloud

ID
ERC

Access Level
Owner

Parent m
Tenant R

Child ma
0

NAME	ID
No result	

Add existing resource

PREVIEW

* Existing resource type: Subscription

* Existing resource list: [redacted]

Moving a management group or subscription to be under a different management group could change the accesses and policies that are applied. Any user that has inherited access from parent management group(s) will be granted access to this moved resource. Learn More

Task 2: Apply the service catalog policy

In this exercise, you will apply one of the built-in Azure Policies to restrict services to the supported list provided by Trey Research.

1. First, we need to build a list of resource types which will be permitted, and their corresponding resource providers. We'll do that using PowerShell. Start PowerShell ISE, and log in to your Azure subscription:

```
Login-AzureRmAccount -Subscription "{subscription name or id}"
```

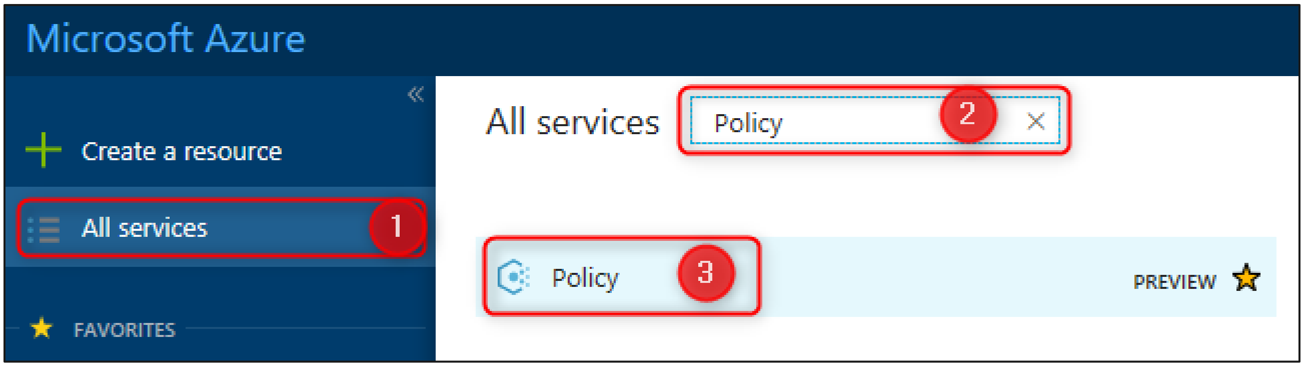
2. Enter the following script into the edit window, and run the script:

```
$FormatEnumerationLimit = -1
Get-AzureRmResourceProvider `
   | Select-Object ProviderNamespace, ResourceTypes `
   | Format-List
```

3. Review the list, and identify the resource providers and resource types for each of the following:

Resource Name
Resource Group
Virtual Machines
Disk
Network Interface
Public IP Address
Network Security Group
Virtual Networks
Virtual Network Gateways
ExpressRoute Circuits
VPN Gateways
Storage Accounts
Backup Vault
Site Recovery Vault
DevTest Labs
Key Vault
Web Apps
SQL Database

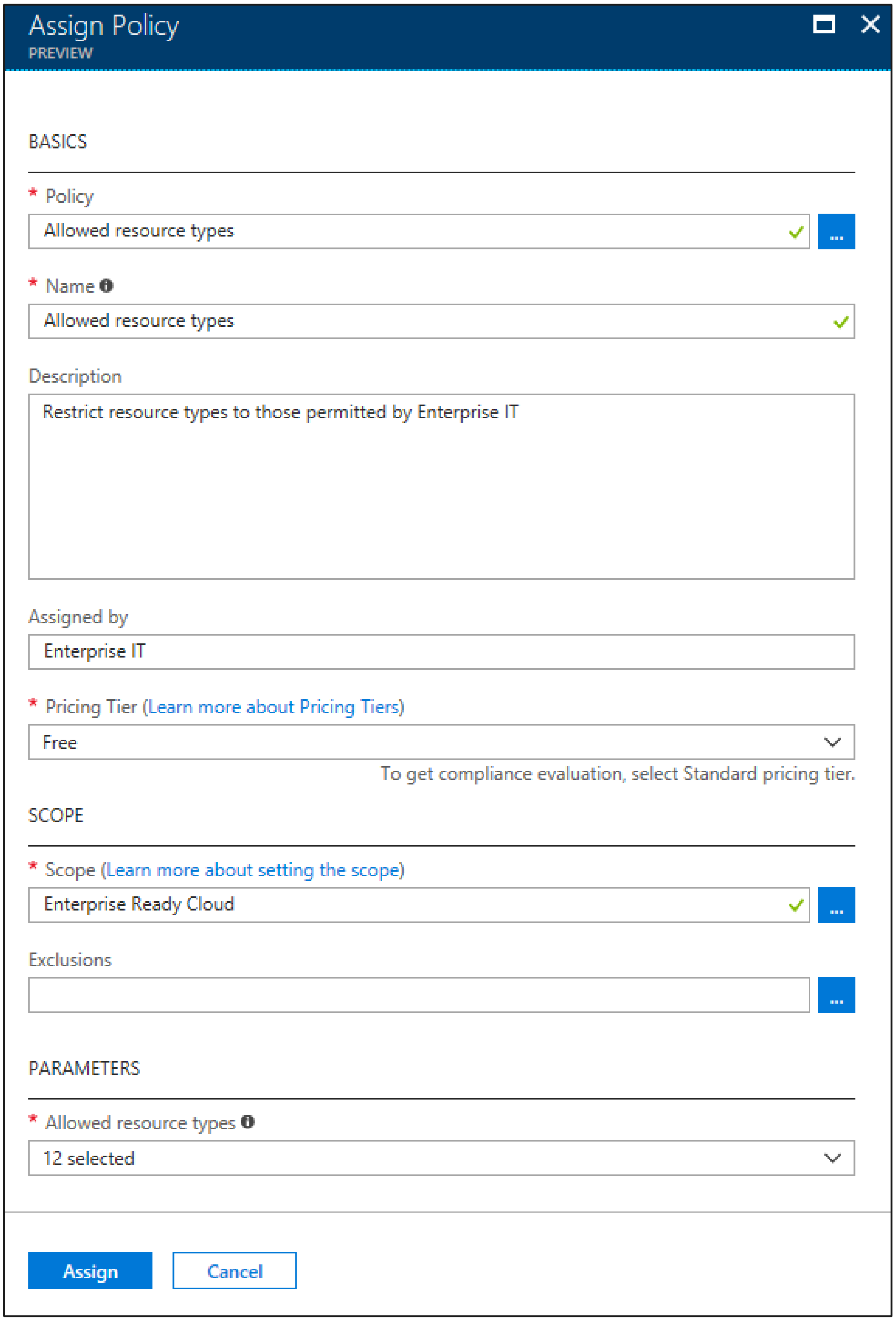
4. Launch the Azure Management portal, and navigate to **Policy** under **All services**:



5. Click **Assignments**, then **Assign Policy**. Complete the form as follows:

a. Policy: Select ‘Allowed resource types’
b. Name: Service Catalog policy
c. Description: Restrict resource types to those permitted by Enterprise IT
d. Assigned by: Enterprise IT
e. Pricing Tier: Standard
f. Scope: Enterprise Ready Cloud (ERC) management group, as created in Task 1
g. Exclusions: None
h. Parameters | Allowed resource types: Choose the resource types identified in Step 3. (You may need to include some additional types, such as for NICs, Public IP Addresses, NSGs, etc.)

The assignment form should look like this:



Assign Policy
PREVIEW

BASICS

* Policy
Allowed resource types ✓

* Name
Allowed resource types ✓

Description
Restrict resource types to those permitted by Enterprise IT

Assigned by
Enterprise IT

* Pricing Tier (Learn more about Pricing Tiers)
Free

To get compliance evaluation, select Standard pricing tier.

SCOPE

* Scope (Learn more about setting the scope)
Enterprise Ready Cloud ✓

Exclusions

PARAMETERS

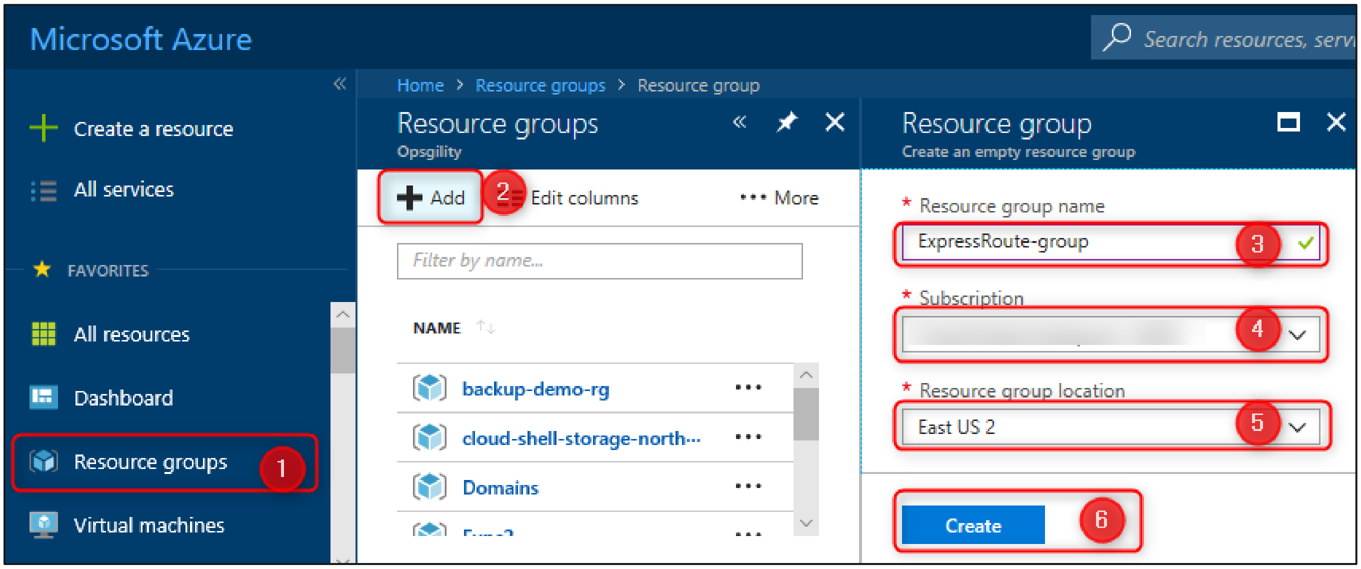
* Allowed resource types
12 selected

When complete, click **Assign** to create the policy assignment.

Task 3: Restrict the creation of ExpressRoute circuits

In this exercise, you will apply another built-in Azure policy to restrict the creation of ExpressRoute circuits. For this policy, we'll use an exclusion scope for the resource group in which Enterprise IT will create the permitted ExpressRoute circuits.

1. First, we'll create the resource group for the exclusion scope. Click **Resource groups**, then **Add**, then fill in the resource group name, select your subscription, and choose a resource group location:



Resource group name: ExpressRoute-group

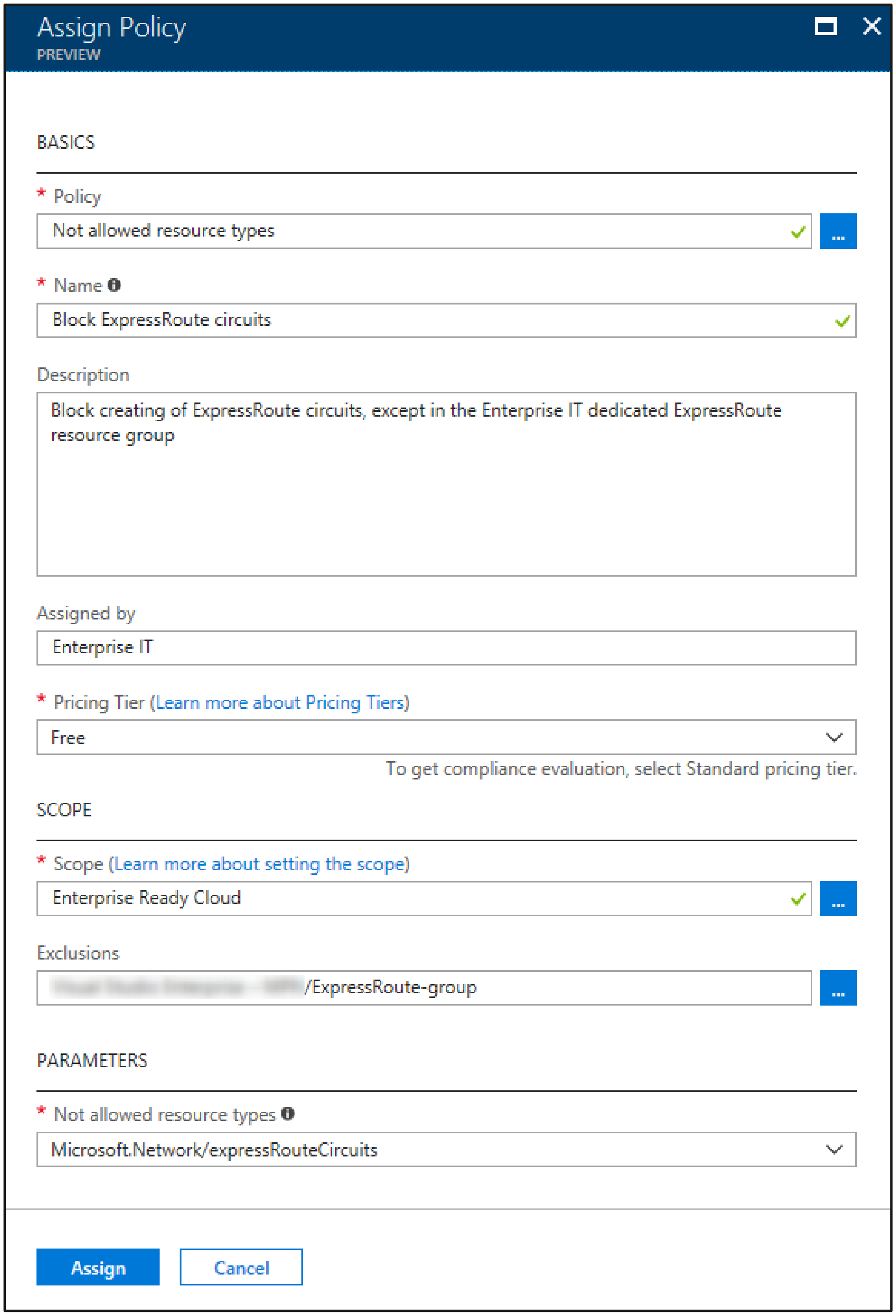
Subscription: [redacted]

Resource group location: East US 2

Once complete, click **Create**.

2. Return to the **Policy** blade in the Azure portal. Click **Assignments**, then **Assign Policy**. Complete the form as follows:
    a. Policy: Not allowed resource types
    b. Name: Block ExpressRoute circuits
    c. Description: Block creating of ExpressRoute circuits, except in the Enterprise IT dedicated ExpressRoute resource group
    d. Assigned by: Enterprise IT
    e. Pricing Tier: Standard
    f. Scope: Enterprise Ready Cloud (the management group created earlier)
    g. Exclusions: The resource group created in Step 1 above. Select the management group, subscription, and resource group.
    h. Parameters | Not allowed resource types: Microsoft.Network/expressRouteCircuits

The assignment form should look like this:



BASICS

* Policy: Not allowed resource types

* Name: Block ExpressRoute circuits

Description: Block creating of ExpressRoute circuits, except in the Enterprise IT dedicated ExpressRoute resource group

Assigned by: Enterprise IT

* Pricing Tier (Learn more about Pricing Tiers): Free

To get compliance evaluation, select Standard pricing tier.

SCOPE

* Scope (Learn more about setting the scope): Enterprise Ready Cloud

Exclusions: [redacted]/ExpressRoute-group

PARAMETERS

* Not allowed resource types: Microsoft.Network/expressRouteCircuits

When complete, click **Assign** to create the policy assignment.

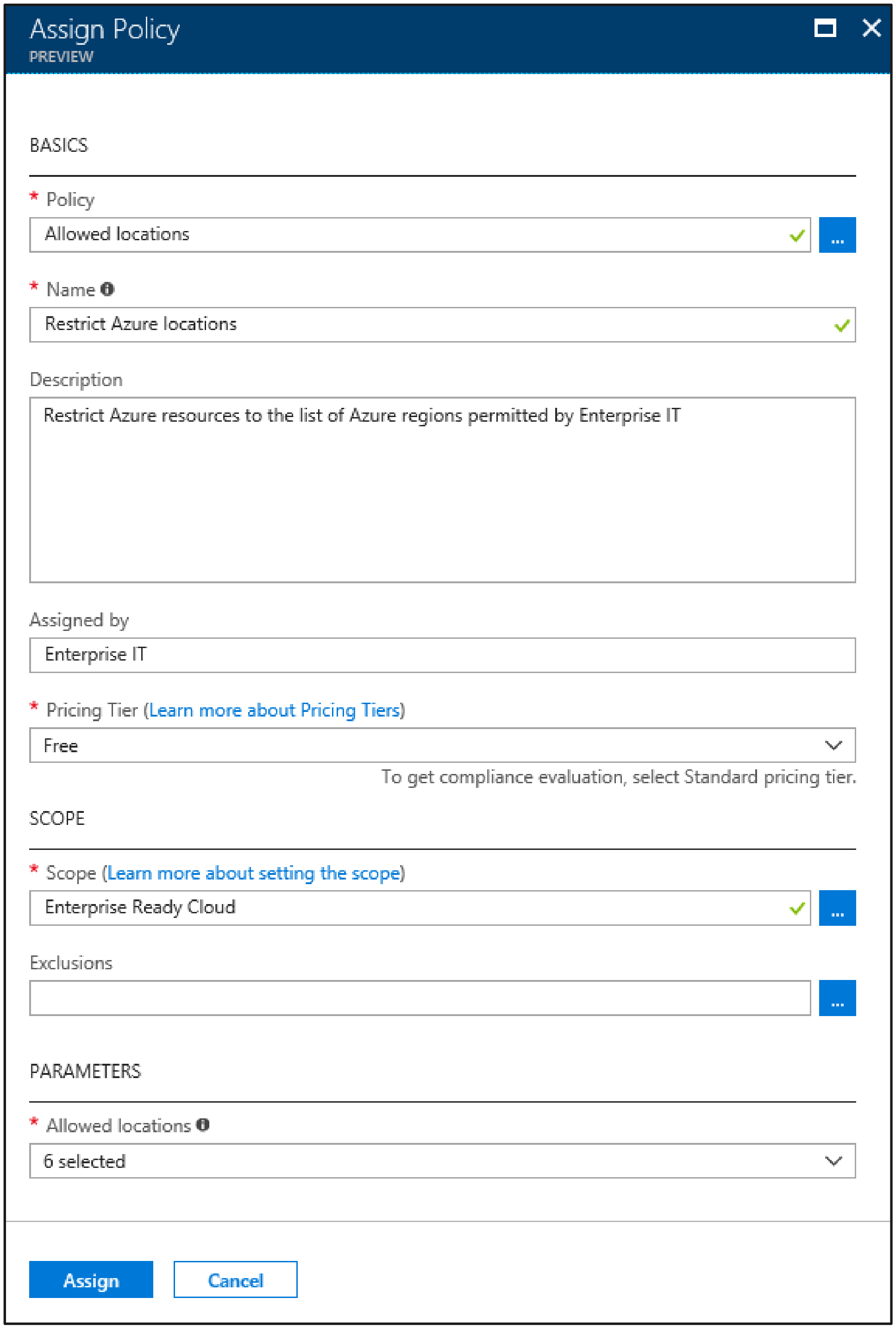
Task 4: Restrict the creation of resources in regions

In this exercise, you will create a new Azure Policy assignment that restricts which regions resources can be created in.

1. In the Azure portal, navigate to **Policy**, then click **Assignments**, then **Assign Policy**. Complete the form as follows:
    a. Policy: Allowed locations

b. Name: Restrict Azure locations
c. Description: Restrict Azure resources to the list of Azure regions permitted by Enterprise IT
d. Assigned by: Enterprise IT
e. Pricing Tier: Free
f. Scope: Enterprise Ready Cloud (the management group created earlier)
g. Exclusions: None
h. Parameters | Allowed locations: East US, West US, North Europe, West Europe, Japan East, Japan West

The assignment form should look like this:



BASICS

* Policy: Allowed locations ✓

* Name: Restrict Azure locations ✓

Description: Restrict Azure resources to the list of Azure regions permitted by Enterprise IT

Assigned by: Enterprise IT

* Pricing Tier (Learn more about Pricing Tiers): Free

To get compliance evaluation, select Standard pricing tier.

SCOPE

* Scope (Learn more about setting the scope): Enterprise Ready Cloud ✓

Exclusions:

PARAMETERS

* Allowed locations: 6 selected

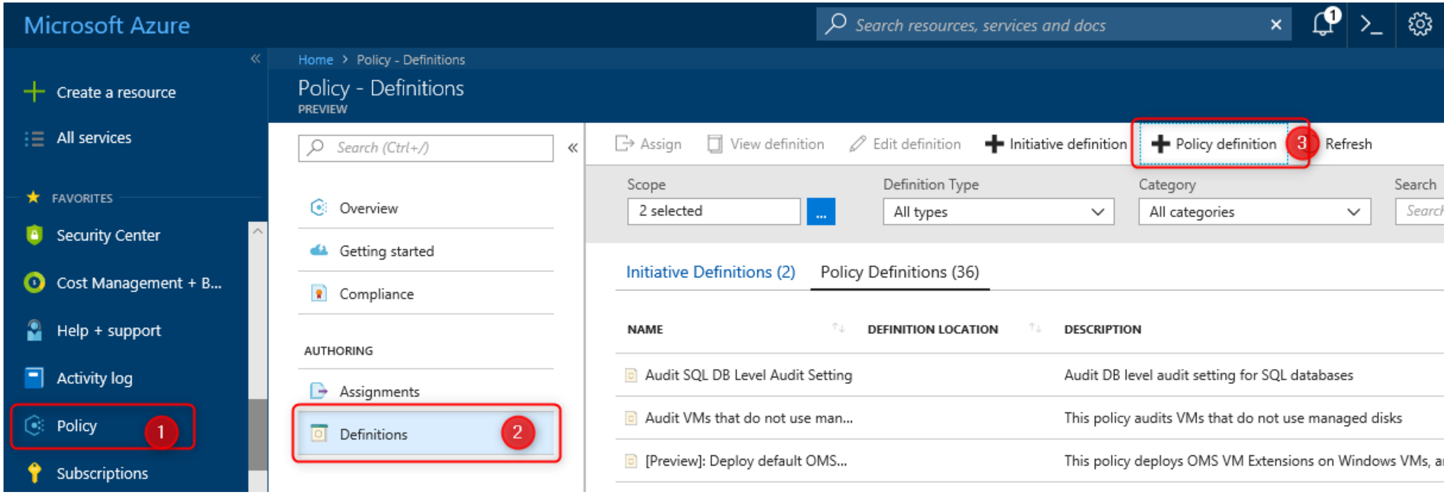
When complete, click **Assign** to create the policy assignment.

Task 5: Create and apply a naming convention

In this task, we will define a simple naming convention for Azure resources. We shall simply require that virtual machine names end with '-vm', virtual networks end with '-vnet', and so on across the various resource types. We shall implement this naming convention using custom policy definition and policy initiative, assigned at the management group scope.

First, we shall create a generic policy definition that restricts resources of a given type to have a given name suffix. The resource type and name suffix shall be specified using parameters.

1. In the Azure portal, open the **Policy** blade, then click **Definitions**, then **+ Policy definition**.



NAME	DEFINITION LOCATION	DESCRIPTION
Audit SQL DB Level Audit Setting		Audit DB level audit setting for SQL databases
Audit VMs that do not use man...		This policy audits VMs that do not use managed disks
[Preview]: Deploy default OMS...		This policy deploys OMS VM Extensions on Windows VMs, a

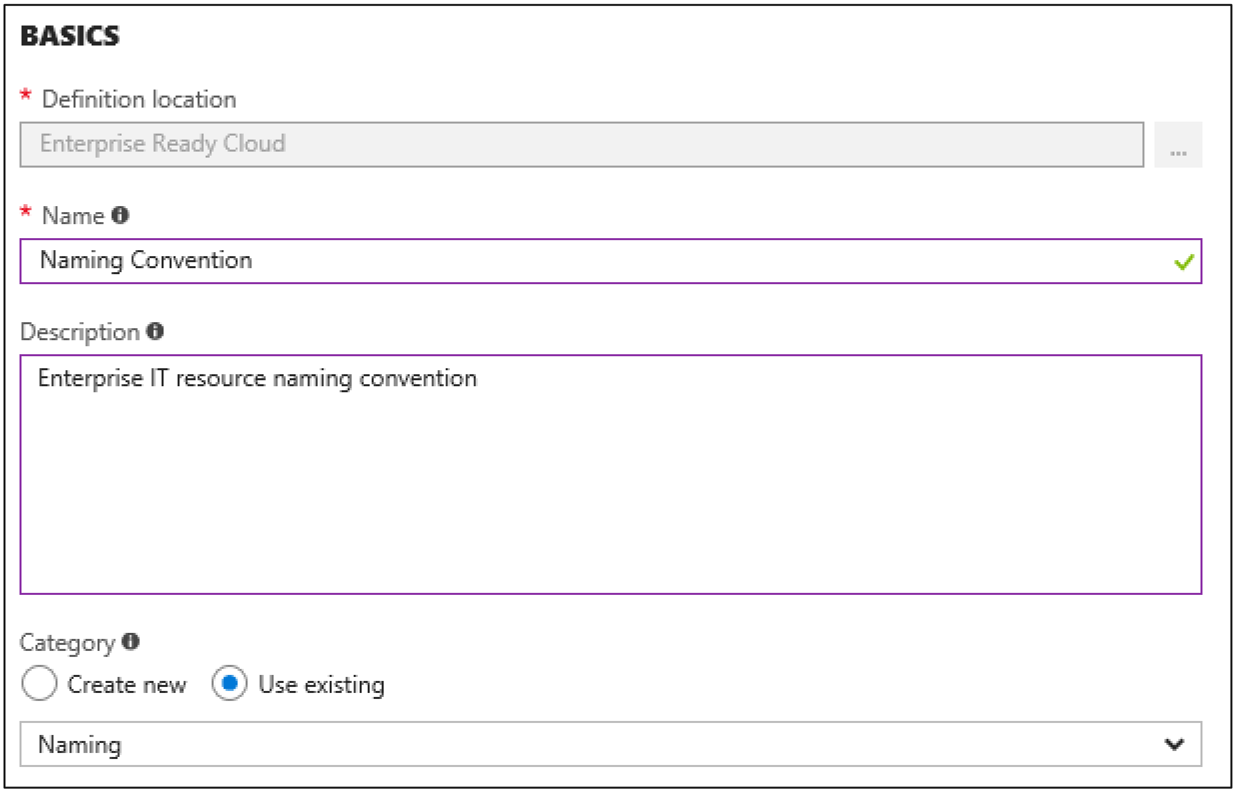
2. Complete the Policy definition form as follows:
   a. Definition location: Enterprise Ready Cloud (the Management Group created earlier)
   b. Name: Restrict Resource Name Suffix
   c. Description: Restrict resources of a given type to have a name ending with a given suffix. The resource type and suffix are parameterized.
   d. Category: Create New, "Naming"
   e. Policy rule and paramters: As shown below:

```
{
  "properties": {
    "mode": "all",
    "parameters": {
      "resourceType": {
        "type": "string",
        "metadata": {
          "displayName": "Resource Type",
          "description": "The resource type for this policy",
          "strongType": "resourceTypes"
        }
      },
      "nameSuffix": {
        "type": "string",
        "metadata": {
          "displayName": "Resource Name Suffix",
          "description": "The suffix that must be appended"
        }
      }
    },
    "policyRule": {
      "if": {
        "allof": [
          {
            "field": "type",
            "equals": "[parameters('resourceType')]"
          },
          {
            "not": {
              "field": "name",
              "like": "[concat('*-', parameters('nameSuffix'))]"
            }
          }
        ]
      },
      "then": {
        "effect": "deny"
      }
    }
  }
}
```

Once the policy definition is complete, click **Save**.

Next, we shall create a policy initiative comprising multiple instances of our policy definition (one per resource type).

3. From the **Policy** blade, on the **Definitions** panel, click **+Initiative Defintion**.
4. Fill in the Initiative Definition form as follows (but don't click Save yet)
   a. Definition location: Enterprise Ready Cloud (the Management Group created earlier)
   b. Name: Naming Convention
   c. Description: Trey Research resource naming convention
   d. Category: Use Existing | Naming



BASICS

* Definition location

Enterprise Ready Cloud

* Name

Naming Convention

Description

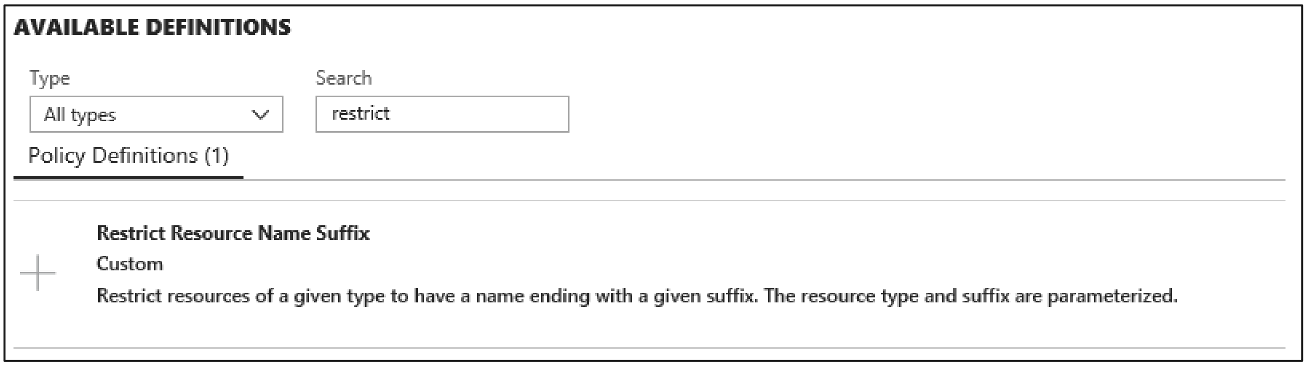
Enterprise IT resource naming convention

Category

○ Create new ● Use existing

Naming

5. Under 'Available Definitions', find the 'Restrict Resource Name Suffix' policy definition created in Step 2.



AVAILABLE DEFINITIONS

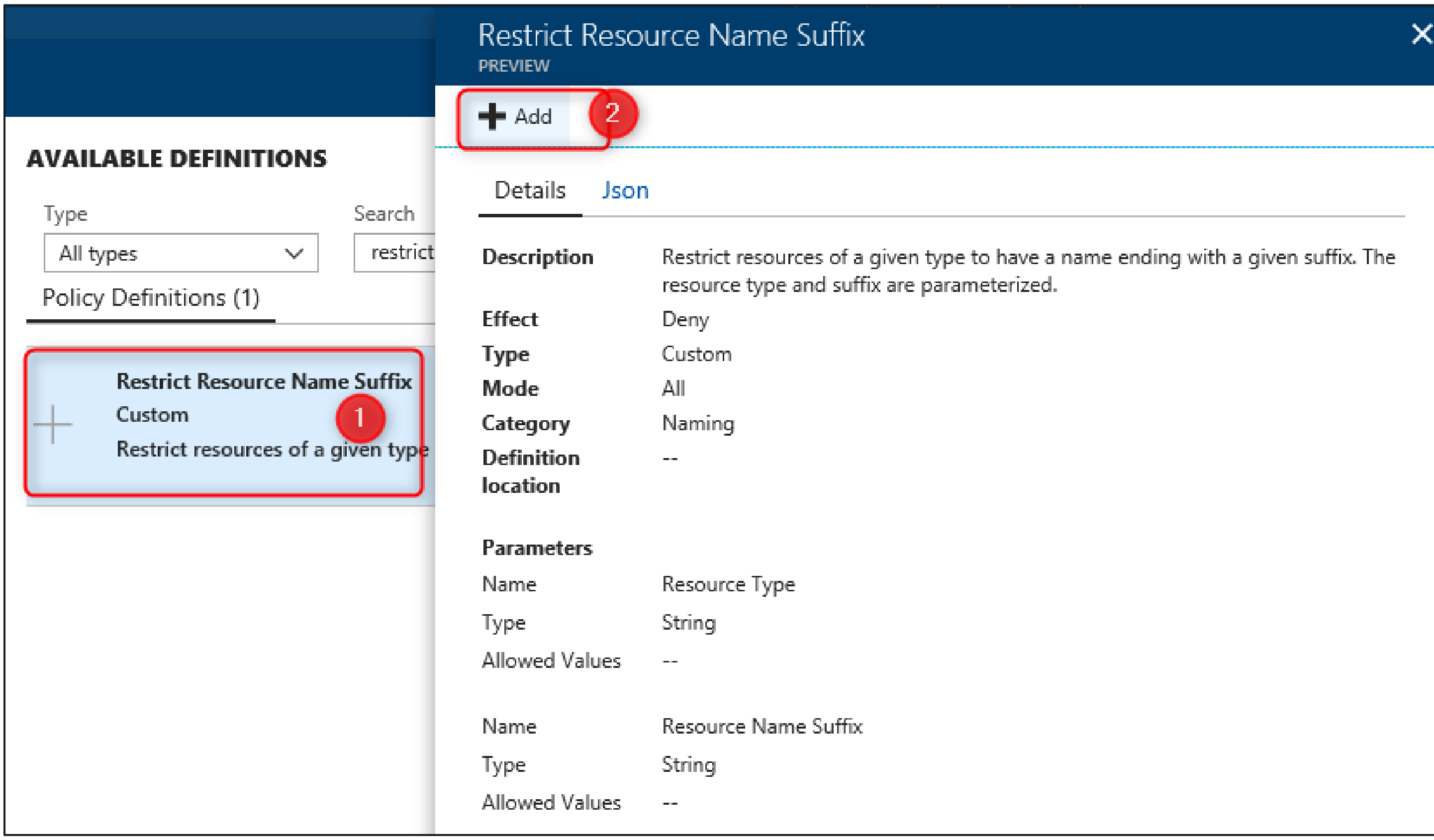
Type: All types

Search: restrict

Policy Definitions (1)

Restrict Resource Name Suffix
Custom
Restrict resources of a given type to have a name ending with a given suffix. The resource type and suffix are parameterized.

6. Click on the policy definition, then click **+Add** to add the Policy Definition to the Policy Initiative.



AVAILABLE DEFINITIONS

Type: All types

Search: restrict

Policy Definitions (1)

Restrict Resource Name Suffix
Custom
Restrict resources of a given type

Restrict Resource Name Suffix
PREVIEW

+ Add

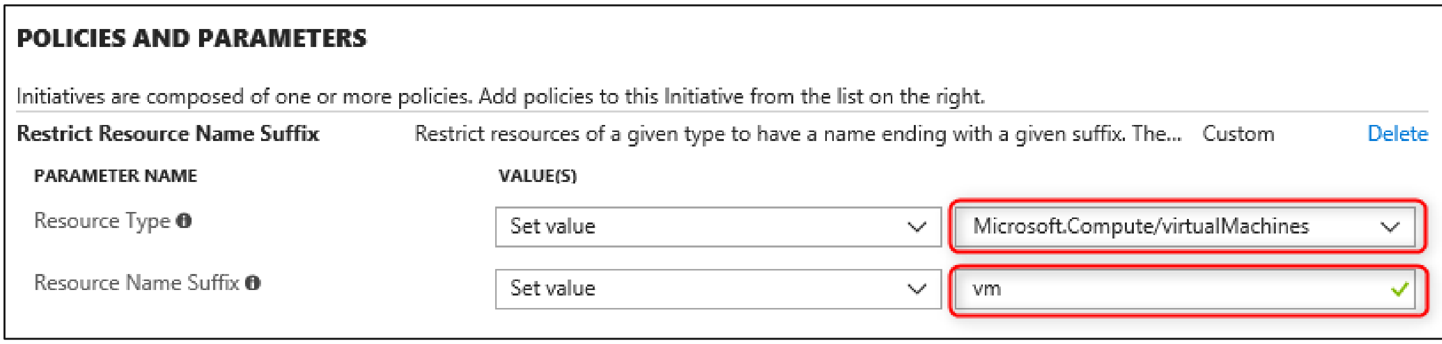
Details | Json

Description	Restrict resources of a given type to have a name ending with a given suffix. The resource type and suffix are parameterized.
Effect	Deny
Type	Custom
Mode	All
Category	Naming
Definition location	--

Parameters

Name	Resource Type
Type	String
Allowed Values	--
Name	Resource Name Suffix
Type	String
Allowed Values	--

7. Select the resource type and name suffix. In this case, we'll choose **Microsoft.Network/virtualMachines** as the resource type, and **vm** as the name suffix.



POLICIES AND PARAMETERS

Initiatives are composed of one or more policies. Add policies to this Initiative from the list on the right.

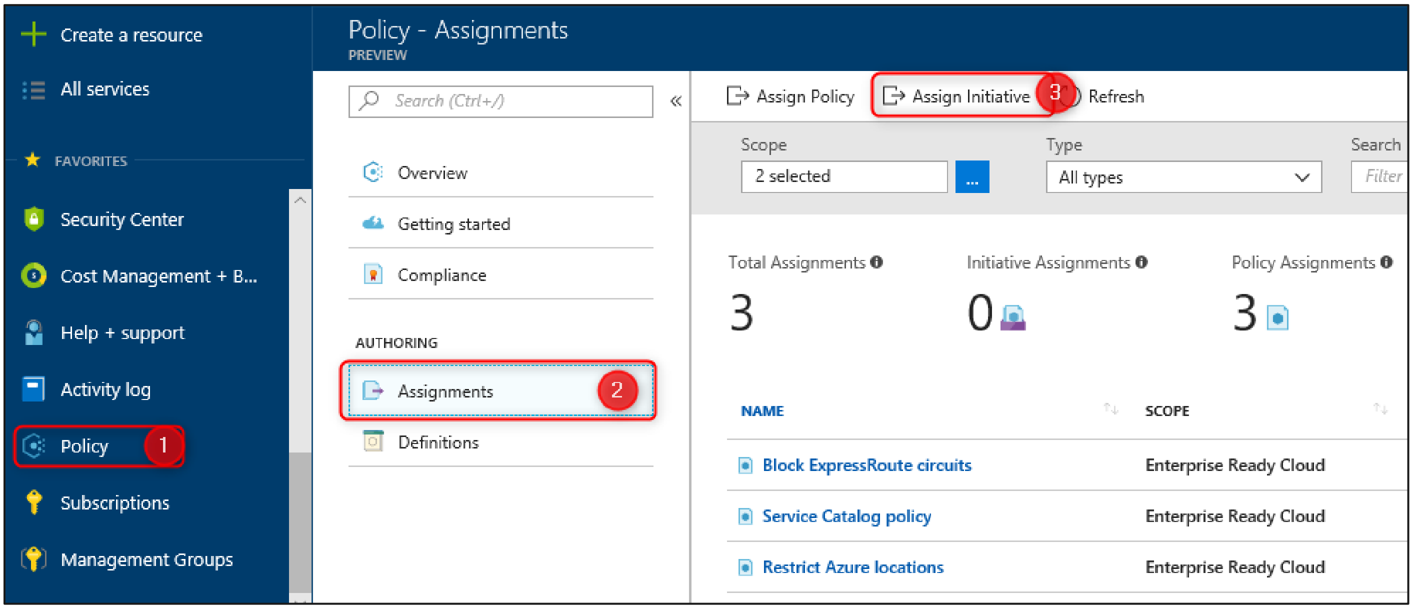
Restrict Resource Name Suffix Restrict resources of a given type to have a name ending with a given suffix. The... Custom Delete

PARAMETER NAME	VALUE(S)	
Resource Type	Set value	Microsoft.Compute/virtualMachines
Resource Name Suffix	Set value	vm ✓

8. Repeat steps 6 and 7 above for each of the resource types you wish to include in the naming convention.
9. Once you've added each resource type, click **Save**.

Finally, we will apply the policy initiative across all subscriptions in the Management Group by creating an assignment at the Management Group scope.

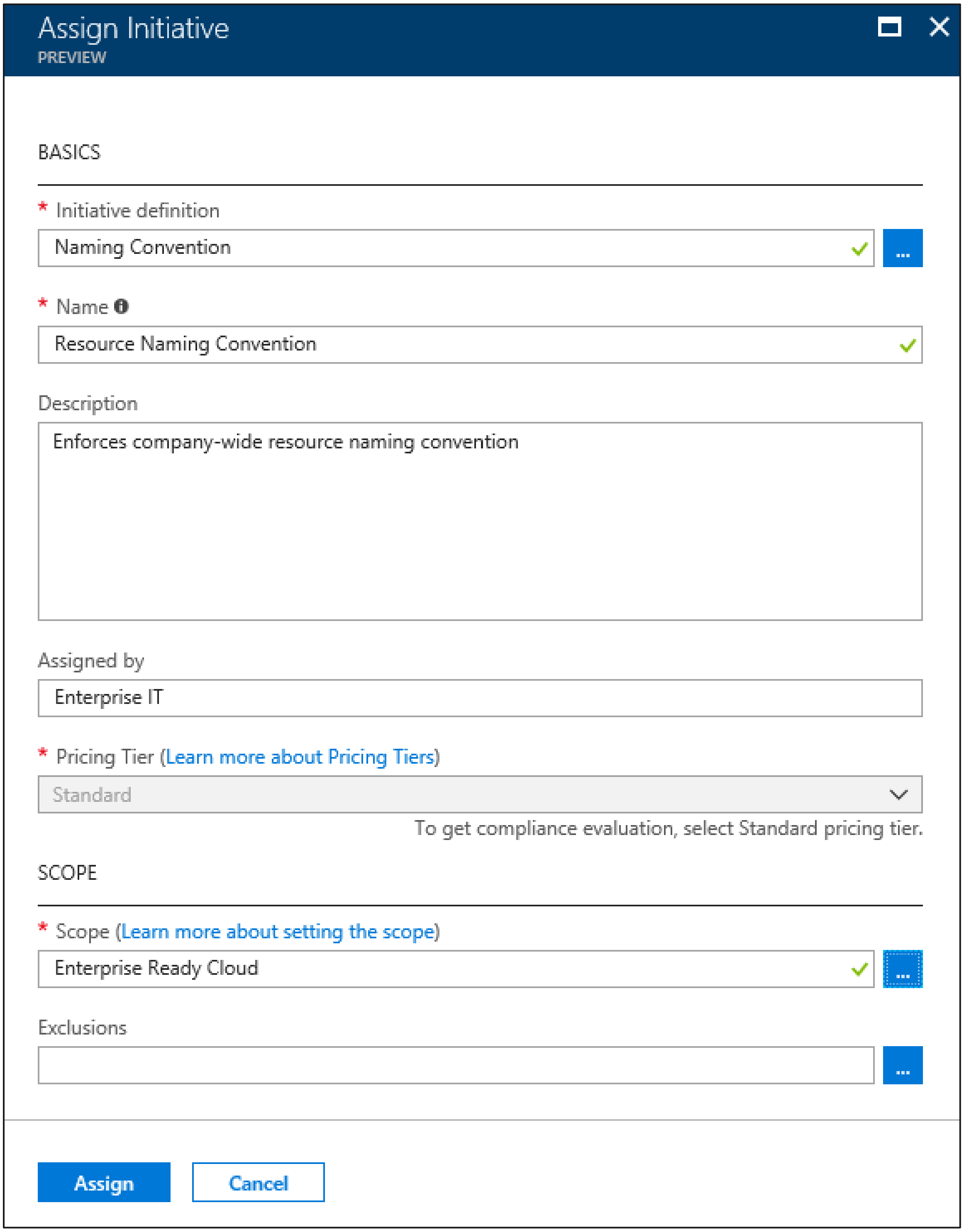
10. On the **Policy** blade, click **Assignments**, then **Assign Initiative**.



NAME	SCOPE
Block ExpressRoute circuits	Enterprise Ready Cloud
Service Catalog policy	Enterprise Ready Cloud
Restrict Azure locations	Enterprise Ready Cloud

11. Complete the Assign Initiative form as follows:
    a. Initiative definition: Naming Convention (the initiative definition we just created)
    b. Name: Resource Naming Convention
    c. Description: Enforces company-wide resource naming convention
    d. Assigned by: Enterprise IT
    e. Pricing Tier: Standard
    f. Scope: Enterprise Ready Cloud (the Management Group created earlier)
    g. Exclusions: None

The assignment form should look like this:



Assign Initiative
PREVIEW

BASICS

* Initiative definition: Naming Convention ✓

* Name: Resource Naming Convention ✓

Description: Enforces company-wide resource naming convention

Assigned by: Enterprise IT

* Pricing Tier (Learn more about Pricing Tiers): Standard

To get compliance evaluation, select Standard pricing tier.

SCOPE

* Scope (Learn more about setting the scope): Enterprise Ready Cloud ✓

Exclusions:

When complete, click **Assign** to create the policy initiative assignment.

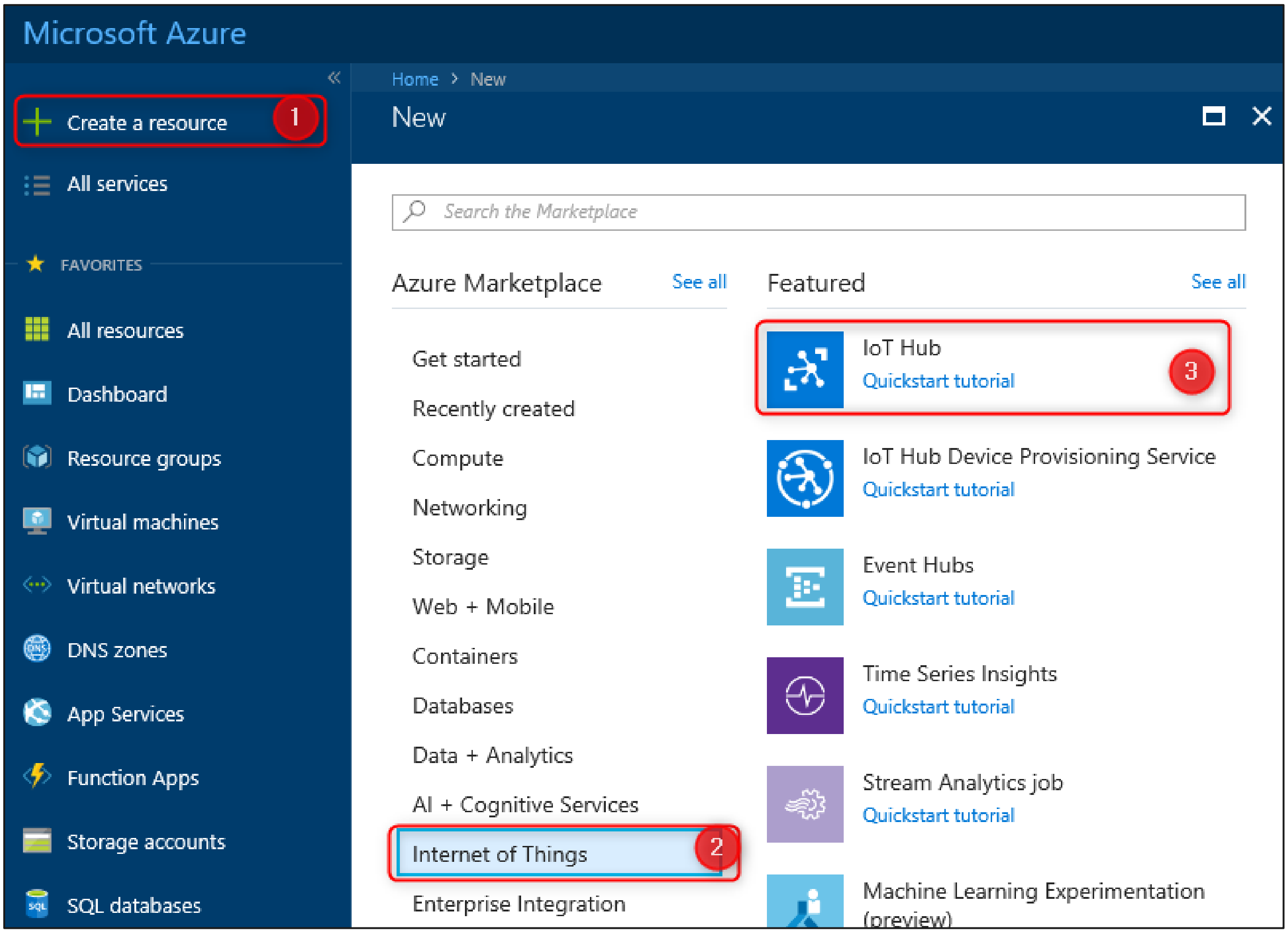
Task 6: Test the policies

In this task, you will use the Azure management portal to validate each of the policies created so far, and to understand how to identify policy events.

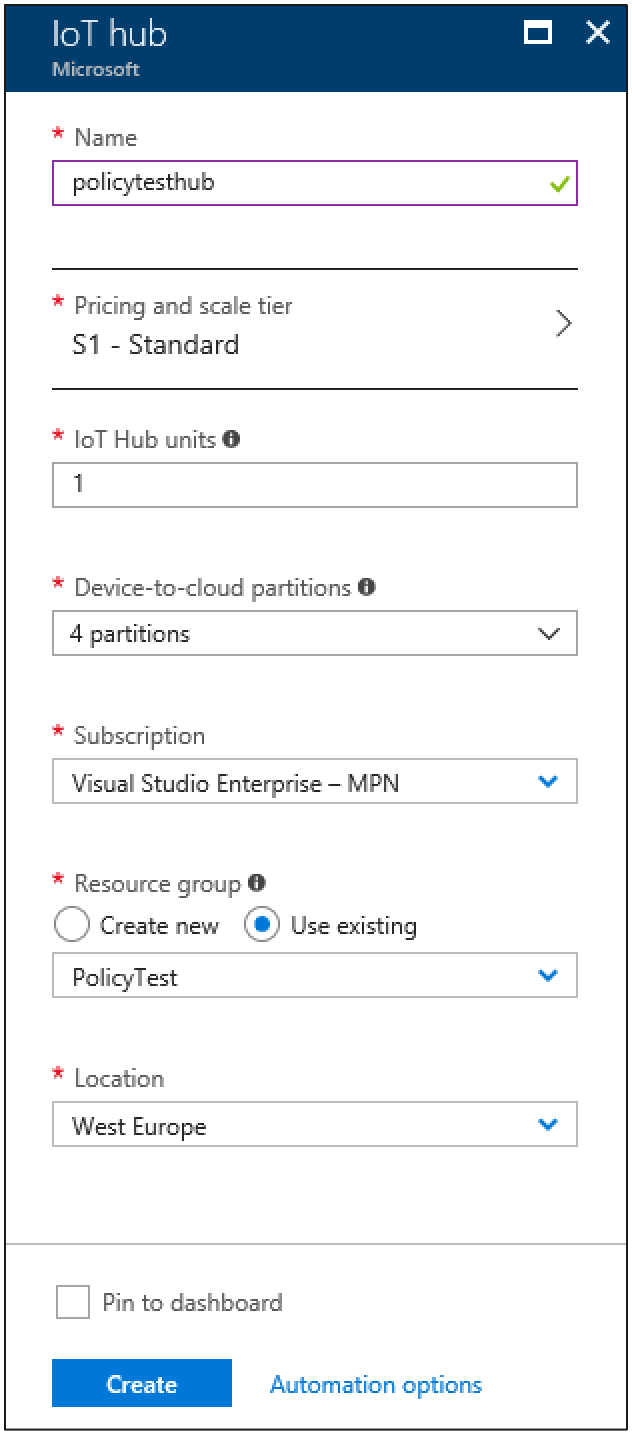
Subtask 1: Test the service catalog policy

1. Navigate to the Azure management portal in a browser http://portal.azure.com, and sign in.

2. Click **Create a Resource > Internet of Things > IoT Hub**



3. Specify a unique name for the IoT Hub, and choose an existing resource group. Choose a permitted location (we are only testing the Service Catalog policy at this time).



IoT hub
Microsoft

* Name
policytesthub

* Pricing and scale tier
S1 - Standard

* IoT Hub units
1

* Device-to-cloud partitions
4 partitions

* Subscription
Visual Studio Enterprise – MPN

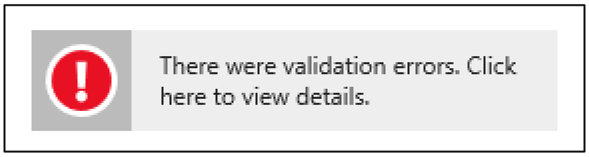
* Resource group
○ Create new ● Use existing
PolicyTest

* Location
West Europe

□ Pin to dashboard

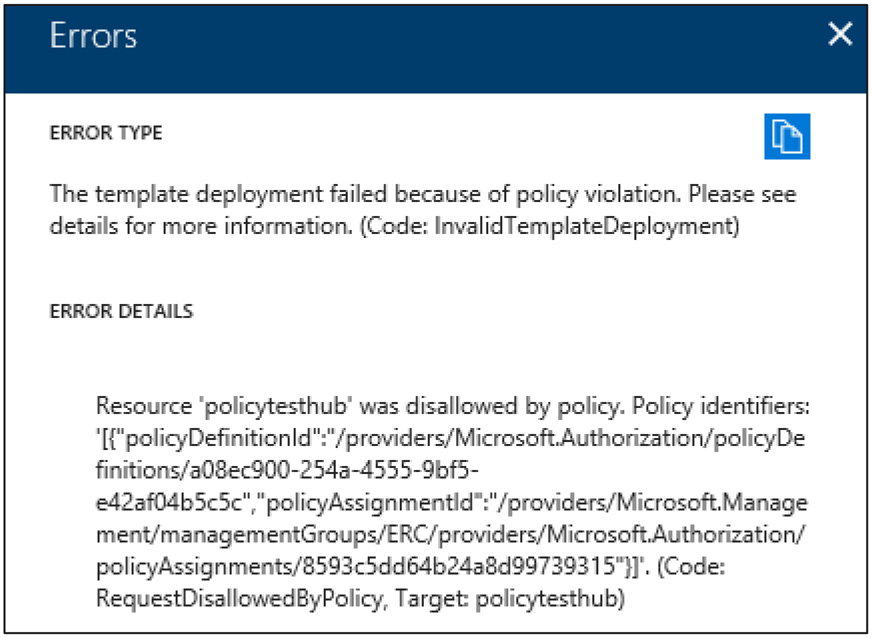
Once all the settings have been filled in, click **Create**.

4. The IoT Hub creation blade should show an error:



There were validation errors. Click here to view details.

5. Click on the error. The following error details are displayed:



Errors

ERROR TYPE

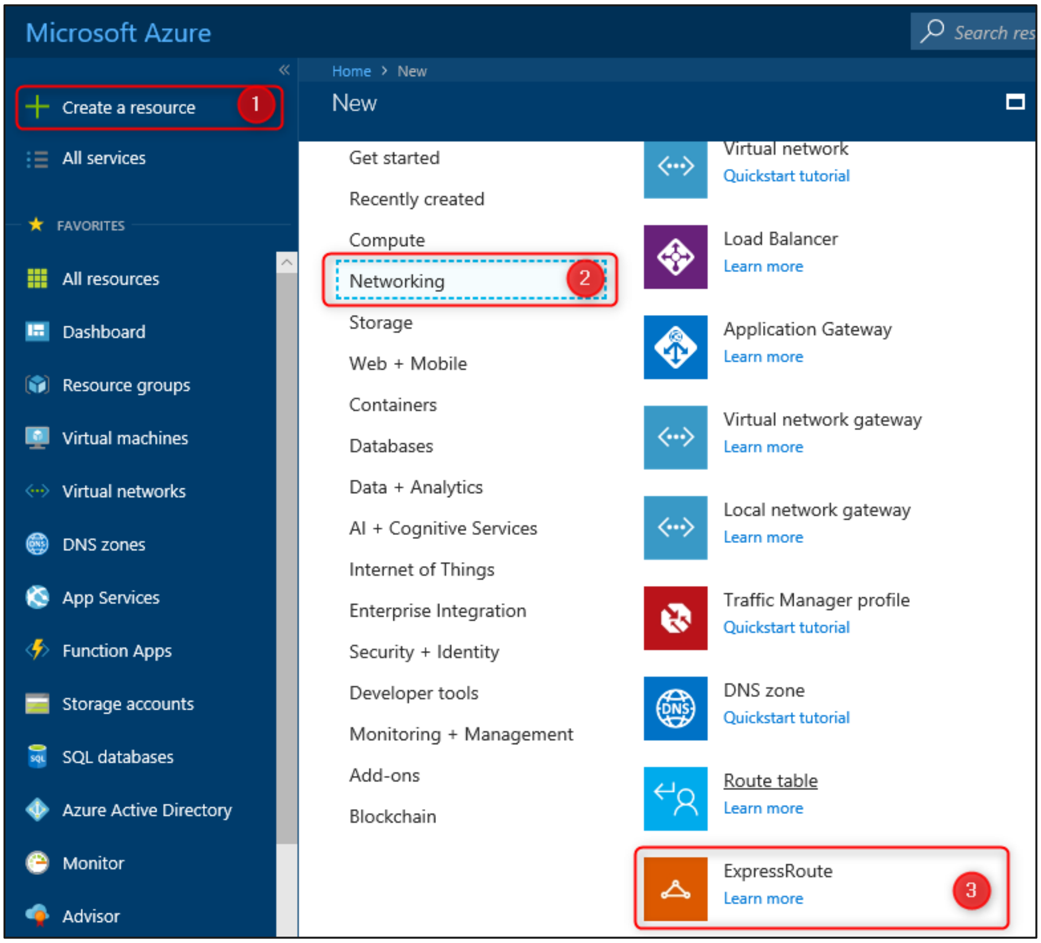
The template deployment failed because of policy violation. Please see details for more information. (Code: InvalidTemplateDeployment)

ERROR DETAILS

Resource 'policytesthub' was disallowed by policy. Policy identifiers: '[{"policyDefinitionId":"/providers/Microsoft.Authorization/policyDefinitions/a08ec900-254a-4555-9bf5-e42af04b5c5c","policyAssignmentId":"/providers/Microsoft.Management/managementGroups/ERC/providers/Microsoft.Authorization/policyAssignments/8593c5dd64b24a8d99739315"}]'. (Code: RequestDisallowedByPolicy, Target: policytesthub)

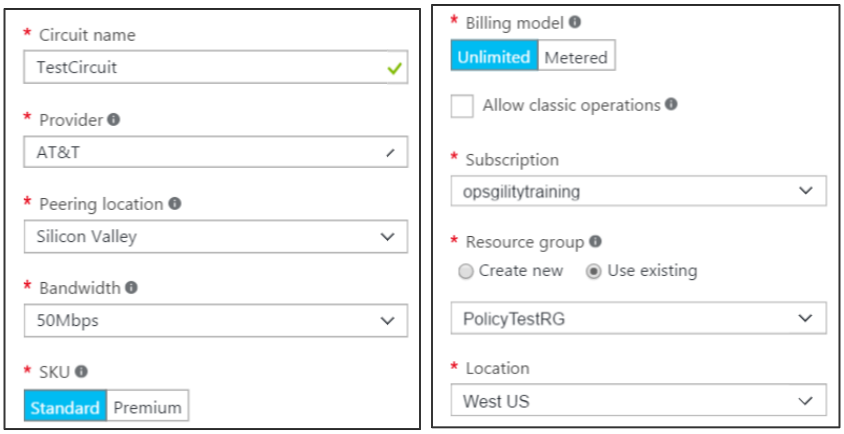
Subtask 2: Test the ExpressRoute circuit policy

6. Click **New > Networking > ExpressRoute**.



7. Specify the following configuration for the circuit and click **Create**.

> Note: you may have to specify an alternate region if West United States is not supported with your subscription.



* Circuit name: TestCircuit ✓

* Provider: AT&T

* Peering location: Silicon Valley

* Bandwidth: 50Mbps

* SKU: [✓ Standard] Premium

* Billing model: [✓ Unlimited] Metered

□ Allow classic operations

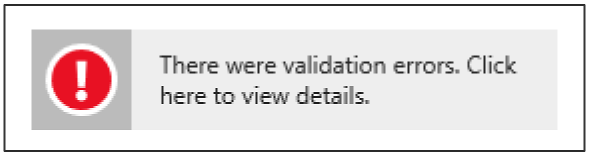
* Subscription: opsgilitytraining

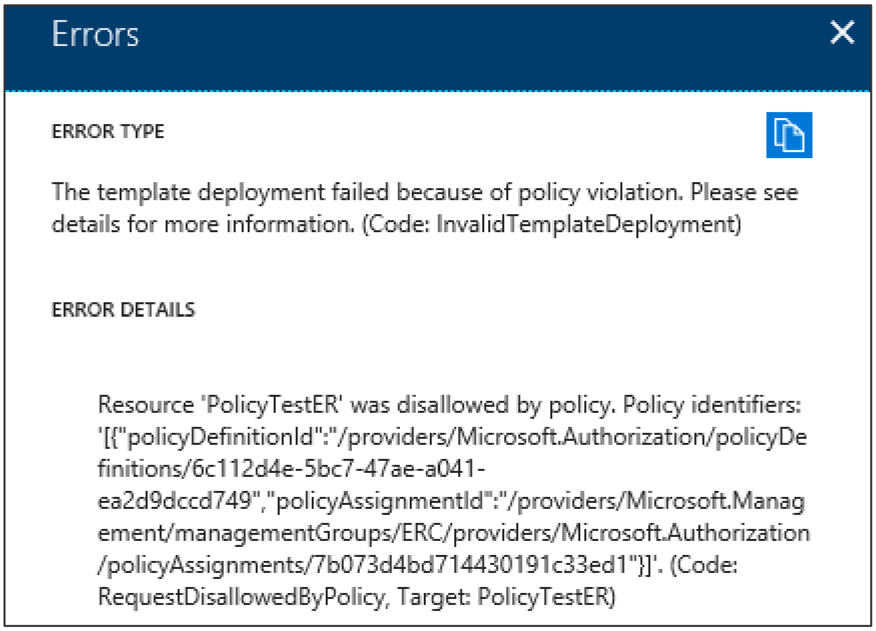
* Resource group: ○ Create new ● Use existing

PolicyTestRG

* Location: West US

8. As with the Service Catalog policy, you should see an error in the Create ExpressRoute Circuit blade, which when clicked shows error details:


There were validation errors. Click here to view details.



ERROR TYPE

The template deployment failed because of policy violation. Please see details for more information. (Code: InvalidTemplateDeployment)

ERROR DETAILS

Resource 'PolicyTestER' was disallowed by policy. Policy identifiers: '[{"policyDefinitionId":"/providers/Microsoft.Authorization/policyDefinitions/6c112d4e-5bc7-47ae-a041-ea2d9dccd749","policyAssignmentId":"/providers/Microsoft.Management/managementGroups/ERC/providers/Microsoft.Authorization/policyAssignments/7b073d4bd714430191c33ed1"}]'. (Code: RequestDisallowedByPolicy, Target: PolicyTestER)

SubTask 3: Test the resource location policy

9. Testing the resource location policy follows a similar pattern. Attempt to create a permitted resource, with a permitted name, but in a not-permitted region. For example, create a virtual network named 'erc-vnet' in South Central US. This should be rejected by the 'Restrict Azure locations' policy.
10. To test further, change to a permitted location (e.g. East US) and try again—this time, the virtual network should be created OK. Note: you may need to refresh browser to release caching on policies.

SubTask 4: Test the naming convention policy

11. Attempt to create a permitted resource, in a permitted location, with a not-permitted name. For example, create a virtual network named 'erc-network' in East US. This should be rejected by the 'Resource Naming Convention' policy.
12. To test further, change to a permitted name (e.g. 'erc-network-vnet') and try again—this time, the virtual network should be created OK. Note: you may need to refresh browser to release caching on policies.

Exercise 2: Configure delegated permissions

Duration: 60 minutes

In this exercise, you will configure delegated permissions for users in the Trey Research business unit. You will extend a PowerShell script to automatically provision a limited access user with the configuration of the subscription.

Help references

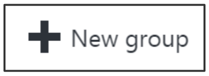
Add new users to Active Directory	https://docs.microsoft.com/azure/active-directory/add-users-azure-active-directory
How Subscriptions are associated with Azure AD	https://docs.microsoft.com/azure/active-directory/active-directory-how-subscriptions-associated-directory
Managing Azure AD Security Groups	https://docs.microsoft.com/azure/active-directory/active-directory-groups-create-azure-portal
Role Based Access Control	https://docs.microsoft.com/azure/active-directory/role-based-access-control-configure
Manage RBAC with PowerShell	https://docs.microsoft.com/azure/active-directory/role-based-access-control-manage-access-powershell

Task 1: Create groups in Azure AD for delegation

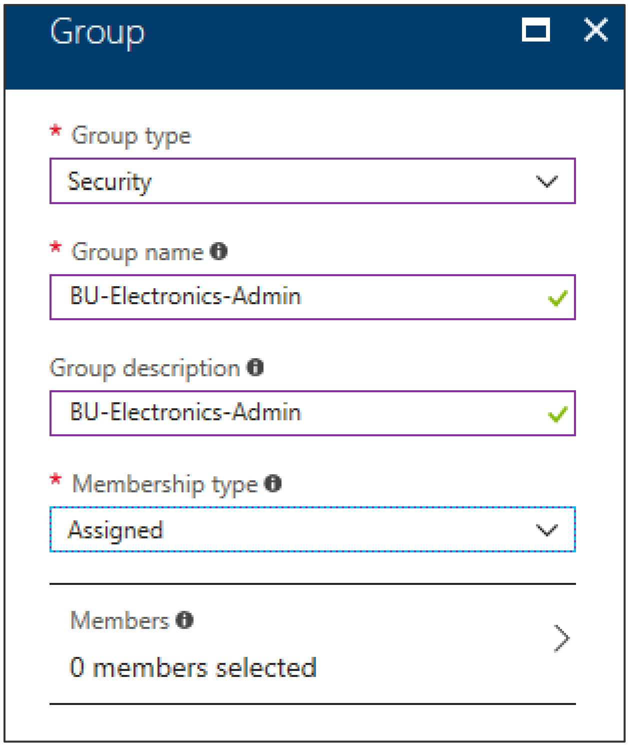
In this task, you will create two groups in Azure AD that you will use for testing delegated access control. You will add the users created in the previous task to the groups.

1. Open the Azure Active Directory console under the Azure Management portal in your browser (https://portal.azure.com).

2. Click **Groups**, and click **New group**.



3. Specify the **Security** as the Group type, **BU-Electronics-Admin** as the **Name** and **Description**, and change the Membership type to **Assigned**. Then, click **Create**.



Group

* Group type: Security

* Group name: BU-Electronics-Admin

Group description: BU-Electronics-Admin

* Membership type: Assigned

Members: 0 members selected

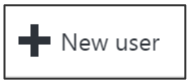
4. Repeat the process, and create another group named **BU-Electronics-Users.**

Task 2: Create user accounts in Azure AD for delegation

In this task, you will create two user accounts in Azure AD that you will use for testing delegated access control.

1. Navigate to **All services -> Azure Active Directory**, and click on **Custom domain names** to find out the name of your Azure AD tenant (this will be needed in the next step).

2. Click **Users**, and click **+New user**.



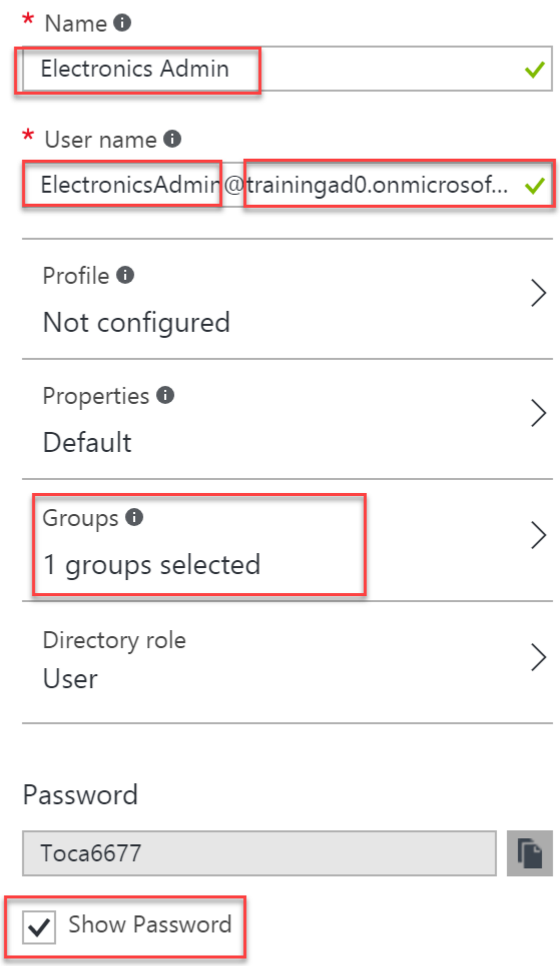
3. Specify the following configuration for the new user:

Name
Electronics Admin

User name:
ElectronicsAdmin@[yourtenant].onmicrosoft.com

Groups
Add the user to the BU-Electronics-Admin group.

Password
Check the Show password checkbox and note the password for later.



* Name: Electronics Admin ✓

* User name: ElectronicsAdmin@trainingad0.onmicrosof... ✓

Profile: Not configured

Properties: Default

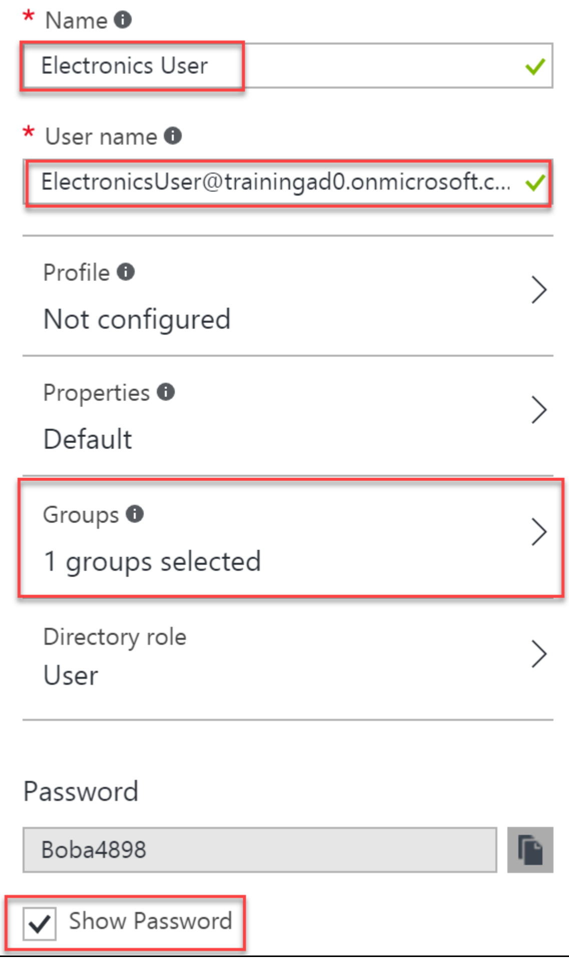
Groups: 1 groups selected

Directory role: User

Password: Toca6677

☑ Show Password

4. Create a second user with the following configuration:

Name Electronics User **User name:** ElectronicsUser@[yourtenant].onmicrosoft.com **Groups** Add the user to the BU-Electronics-User group. **Password** Check the Show password checkbox and note the password for later.	 * Name: Electronics User * User name: ElectronicsUser@trainingad0.onmicrosoft.c... Profile: Not configured Properties: Default Groups: 1 groups selected Directory role: User Password: Boba4898 ☑ Show Password

Task 3: Enable a business unit administrator for the subscription

In this task, you will update a script to automatically add a user to the contributor role of the subscription.

1. Open PowerShell ISE, and log in to your Azure account

```
Login-AzureRmAccount
```

2. Create a new script **ConfigureSubscription.ps1** in PowerShell ISE.

3. Add the following code to script, and save the file. This code will retrieve the object ID for the Active Directory group passed in and assign the group to the Contributor role on the subscription.

```
param([string]$SubscriptionId, [string]$AdGroupName)

Select-AzureRmSubscription -SubscriptionId $SubscriptionId

$scope = "/subscriptions/$SubscriptionId"

$groupObjectId = (Get-AzureRmADGroup -SearchString $AdGroupName).Id.Guid
```

```
Write-Host "Adding group to contributor role" -ForegroundColor Green

New-AzureRmRoleAssignment -Scope $scope `
                          -RoleDefinitionName "Contributor" `
                          -ObjectId $groupObjectId
```

This code will add an Azure AD security group to the contributor role at the subscription scope.

4. Create a local variable containing your Subscription ID (you can copy your subscription ID from the Azure portal, or obtain it using `Get-AzureRmSubscription`):
   Paste this under the param section of the script and save.

```
$SubscriptionId = "{your subscription id}"
```

5. Execute the script passing in the -SubscriptionID and -AdGroupName parameters:

```
.\ConfigureSubscription.ps1 -SubscriptionId $SubscriptionId -AdGroupName
"BU-Electronics-Admin"
```

6. Close all instances of your browser (or switch to a different type of browser) and re-launch In-Private or Incognito mode.

7. Navigate to the Azure management portal in a browser http://portal.azure.com, and sign in using the **ElectronicsAdmin** credentials created earlier. When prompted to change your password, specify a strong password you will remember.

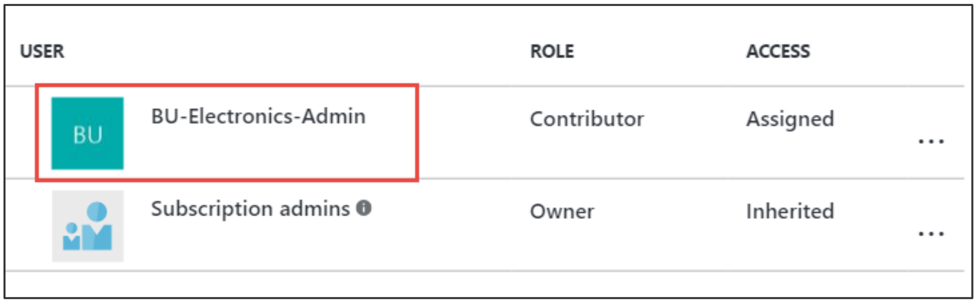
8. You will need to configure a method of resetting your account. You can choose either a phone call or email.

9. Click **All services**, then **Subscriptions**.

10. Click the name of the subscription you have been working on.

11. Click the **Access control (IAM)** tile:

12. You should see the BU-Electronics-Admin group assigned to the contributor role.



USER	ROLE	ACCESS	
BU BU-Electronics-Admin	Contributor	Assigned	...
Subscription admins	Owner	Inherited	...

Users in the contributor role scoped at the subscription have full access to all resources within the subscription but cannot grant access to others or change policies on the subscription.

Task 4: Enable project-based delegation and chargeback

In this task, you will create a script that will create a new resource group, assign 'Owner' rights over the resource group to a given AD group, and apply a policy to enforce an 'IOCode' tag with a given value.

1. Using PowerShell ISE, click **File > New**, and save the file in the **C:\Hackathon\ERC** folder. Name the file **CreateProjectResourceGroup.ps1**.

2. Add the following code to the script, and **Save** the file.

```
param(
    [string]$SubscriptionId,
    [string]$ResourceGroupName,
    [String]$Location,
    [String]$IOCode,
    [string]$AdGroupName
)

Select-AzureRmSubscription -SubscriptionId $SubscriptionId

# Create resource group
New-AzureRmResourceGroup -Name $ResourceGroupName -Location $Location

$scope = "/subscriptions/$subscriptionId/resourceGroups/$resourceGroupName"

# Assign Owner role to given group
$groupObjectId = (Get-AzureRmADGroup -SearchString $AdGroupName).Id.Guid

New-AzureRmRoleAssignment -Scope $scope `
                          -RoleDefinitionName "Owner" `
                          -ObjectId $groupObjectId

# Assign policy to apply IOCode tag
$definition = Get-AzureRmPolicyDefinition | where {$_.Properties.displayName
-eq "Apply tag and its default value"}
```

```
$parameters = @{
    tagName = 'IOCode'
    tagValue = $IOCode
    }

New-AzureRmPolicyAssignment -Name "AppendIOCode" `
                            -Scope $scope `
                            -DisplayName "Append IO Code" `
                            -PolicyDefinition $definition `
                            -PolicyParameterObject $parameters
```

This code creates a new resource group in the specified region. It then assigns the group to the owner role definition just for the resource group. It will allow users in the group to have full ownership of resources within the resource group only. This code then applies a built-in policy to append a tag with name 'IOCode' and the given tag value to any resource created in the resource group.

3. In the **Console** pane, create a new variable called **$location**, and specify a region name to deploy to the resource group to. This location must be one of the supported regions in your previously created policy.

```
$location = "West US"
```

4. In the **Console** pane, create a new variable called **$resourceGroupName**, and specify the value as **DelegatedProjectDemo**. Also, make sure you create a **$SubscriptionId** variable as you did earlier.

```
$resourceGroupName = "DelegatedProjectDemo"
$SubscriptionId = "{your subscription id}"
```

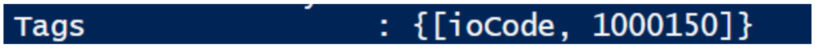
5. In the **Console** pane, execute the following command to create a new resource group with delegated permissions and IO Code policy.

```
.\CreateProjectResourceGroup.ps1 -SubscriptionId $SubscriptionId -
ResourceGroupName $resourceGroupName -Location $location -IOCode "1000150" -
AdGroupName "BU-Electronics-Admin"
```

6. Create a new storage account in the resource group (choose a unique name) to validate the ioCode tag was applied (replace uniquestorageaccount with a unique value).

```
New-AzureRmStorageAccount -ResourceGroupName $resourceGroupName `
                          -Name "uniquestorageaccount" -SkuName Standard_LRS `
                          -Location $location
```

You should see the ioCode tag applied in the output.



```
Tags                : {[ioCode, 1000150]}
```

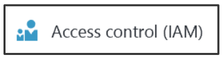
7. Switch back to the Azure Management portal using the ElectronicsAdmin credentials.

8. Click **Resource Groups.**

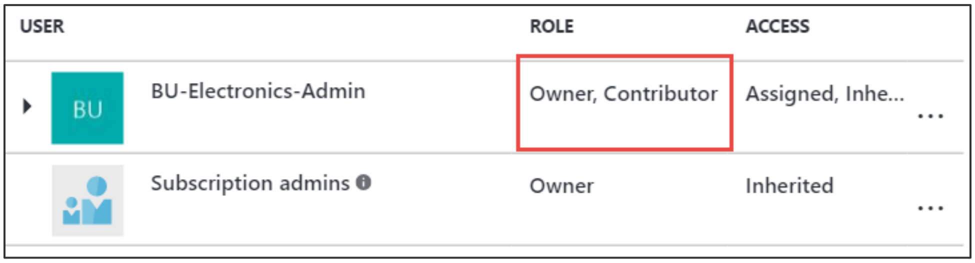
9. Click the **DelegatedProjectdemo** resource group.

DelegatedProjectDemo

10. Click the **Access** icon.

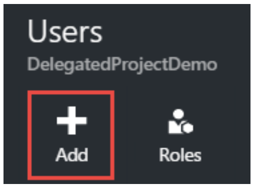


11. Note the BU-Electronics-Admin role is set as owner for the resource group.

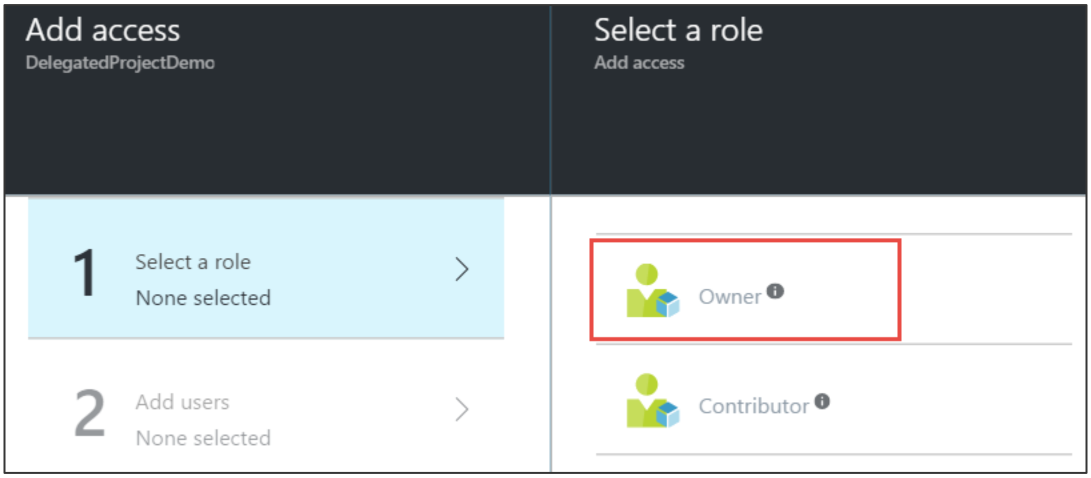


USER	ROLE	ACCESS
BU BU-Electronics-Admin	Owner, Contributor	Assigned, Inhe...
Subscription admins	Owner	Inherited

12. Click **Add**.



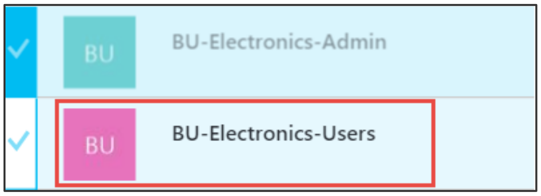
13. Select **Owner**.



1 Select a role — None selected

2 Add users — None selected

14. Click the **BU-Electronics-Users group** > **Select** > **OK** to add the group to the role.



Exercise 3: Create the environment for the e-commerce team

Duration: 75 minutes

In this exercise, you will configure a new environment for the developers of the e-commerce team. You will configure access to a subnet where other developer resources are available and provide secure access to the network for the developers.

Help references

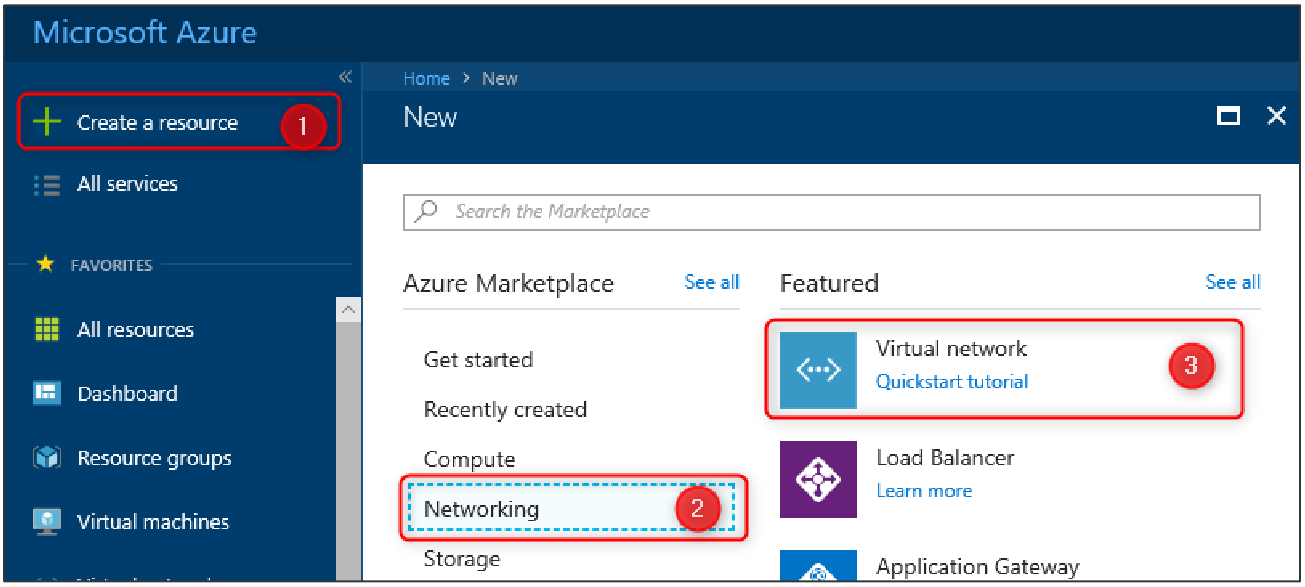
Configuring Point-to-Site Secure VPN	https://docs.microsoft.com/azure/vpn-gateway/vpn-gateway-howto-point-to-site-resource-manager-portal
Network Security Groups	https://docs.microsoft.com/azure/virtual-network/virtual-networks-nsg
Azure DevTest Labs	https://docs.microsoft.com/azure/devtest-lab/
MakeCert.exe	https://cloudworkshop.blob.core.windows.net/enterprise-ready-cloud/makecert.exe

Task 1: Create a new virtual network

In this task, you will create a new virtual network for Trey Research.

1. Sign in to the Azure Management portal using the subscription owner user account.

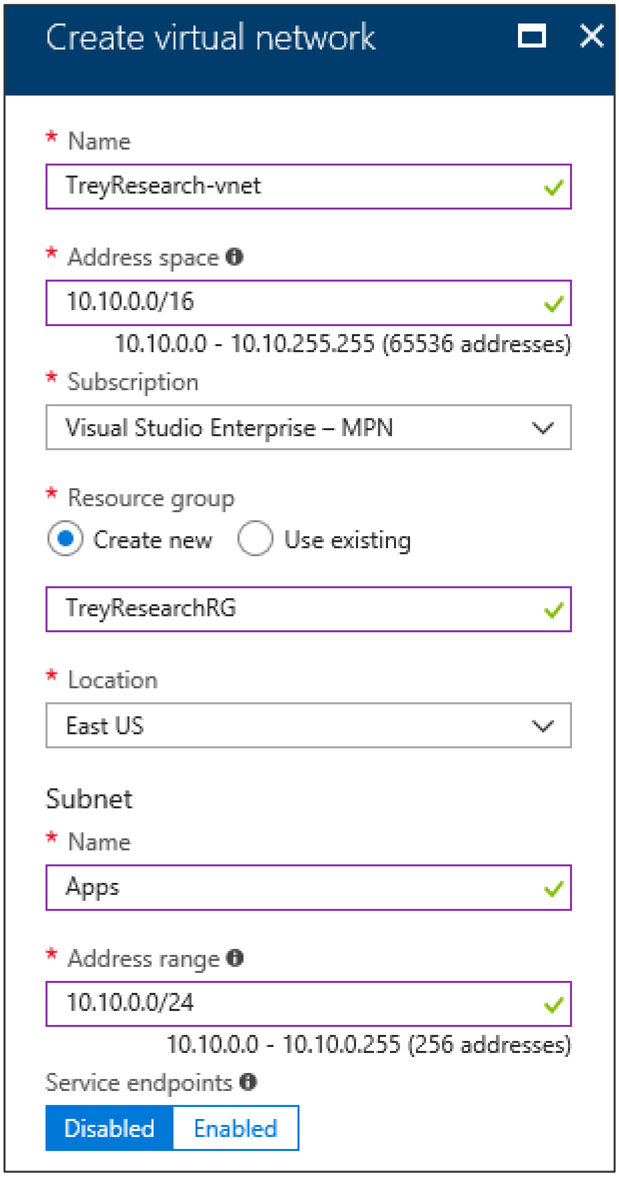
2. Click **New > Networking > Virtual Network**.



3. Specify the following configuration for the virtual network:
   a. Name: TreyResearch-vnet
   b. Address Space: 10.10.0.0/16
   c. Resource Group: TreyResearchRG (Create New)
   d. Location: Choose one of the supported regions.
   e. Subnet Name: Apps

f. Subnet Address Range: 10.10.0.0/24

g. Service endpoints: Disabled



Create virtual network

* Name

TreyResearch-vnet ✓

* Address space

10.10.0.0/16 ✓

10.10.0.0 - 10.10.255.255 (65536 addresses)

* Subscription

Visual Studio Enterprise – MPN

* Resource group

● Create new ○ Use existing

TreyResearchRG ✓

* Location

East US

Subnet

* Name

Apps ✓

* Address range

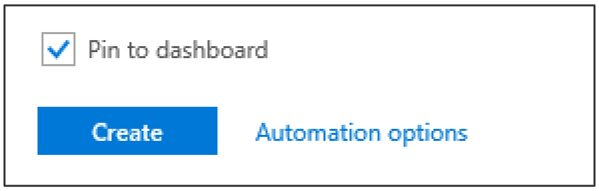
10.10.0.0/24 ✓

10.10.0.0 - 10.10.0.255 (256 addresses)

Service endpoints

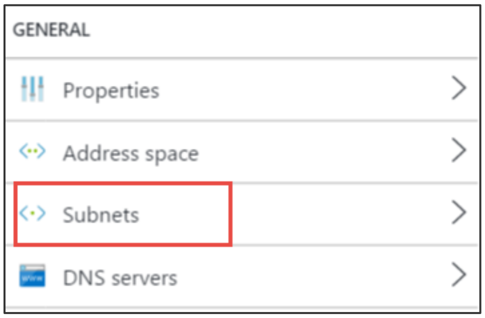
[✓ Disabled] Enabled

4. Select Pin to dashboard, and click **Create**.

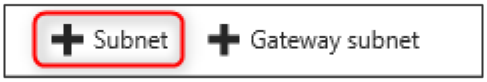


☑ Pin to dashboard

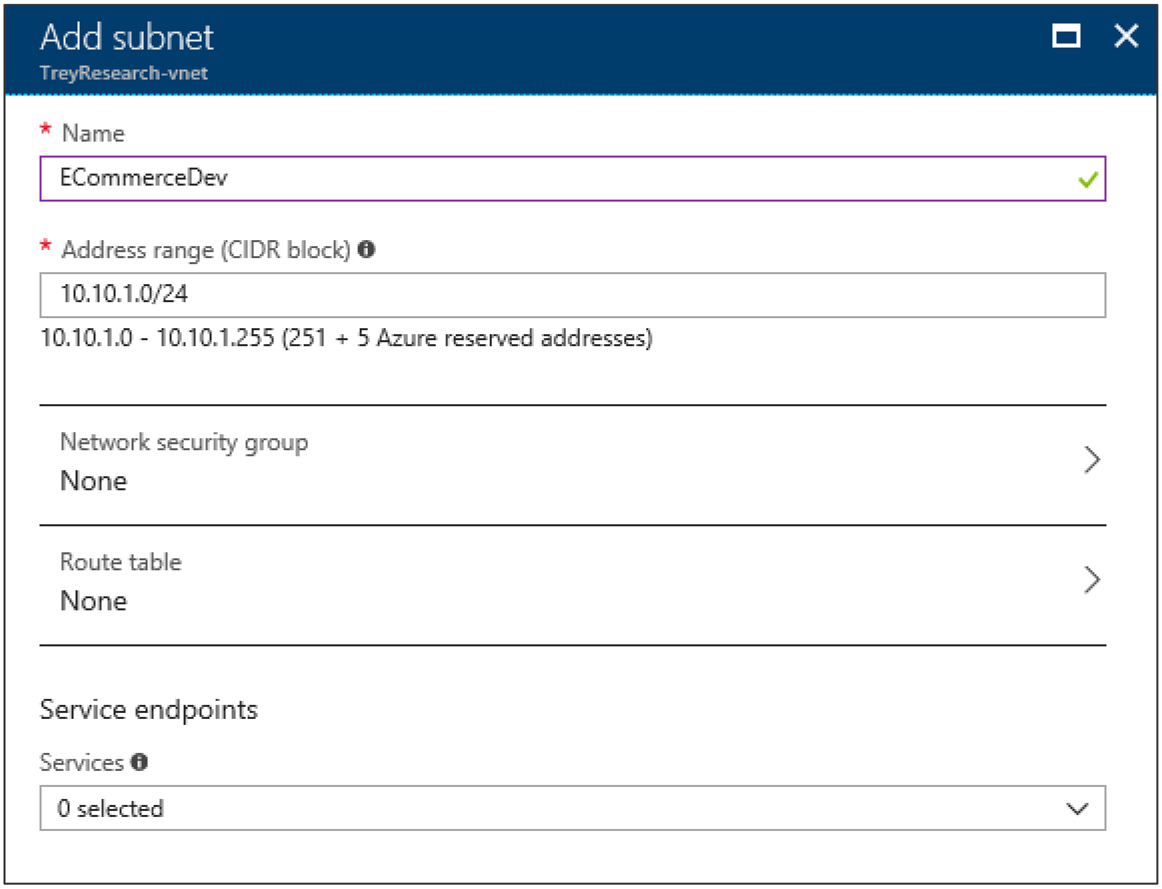
5. After the virtual network is open, click **Subnets.**



6. Click **+Subnet**.



7. Name the subnet **ECommerceDev**, and specify the Address Range as **10.10.1.0/24**.



Add subnet

TreyResearch-vnet

* Name

ECommerceDev

* Address range (CIDR block)

10.10.1.0/24

10.10.1.0 - 10.10.1.255 (251 + 5 Azure reserved addresses)

Network security group

None

Route table

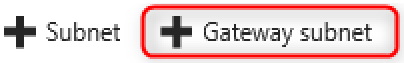
None

Service endpoints

Services

0 selected

8. Click **+ Gateway subnet** to add a gateway subnet to the virtual network. Note: VPN gateway and ExpressRoute gateway could be co-existing at Gateway subnet. We recommend that you create a gateway subnet of /27 or larger (/27, /26, /25 etc.)

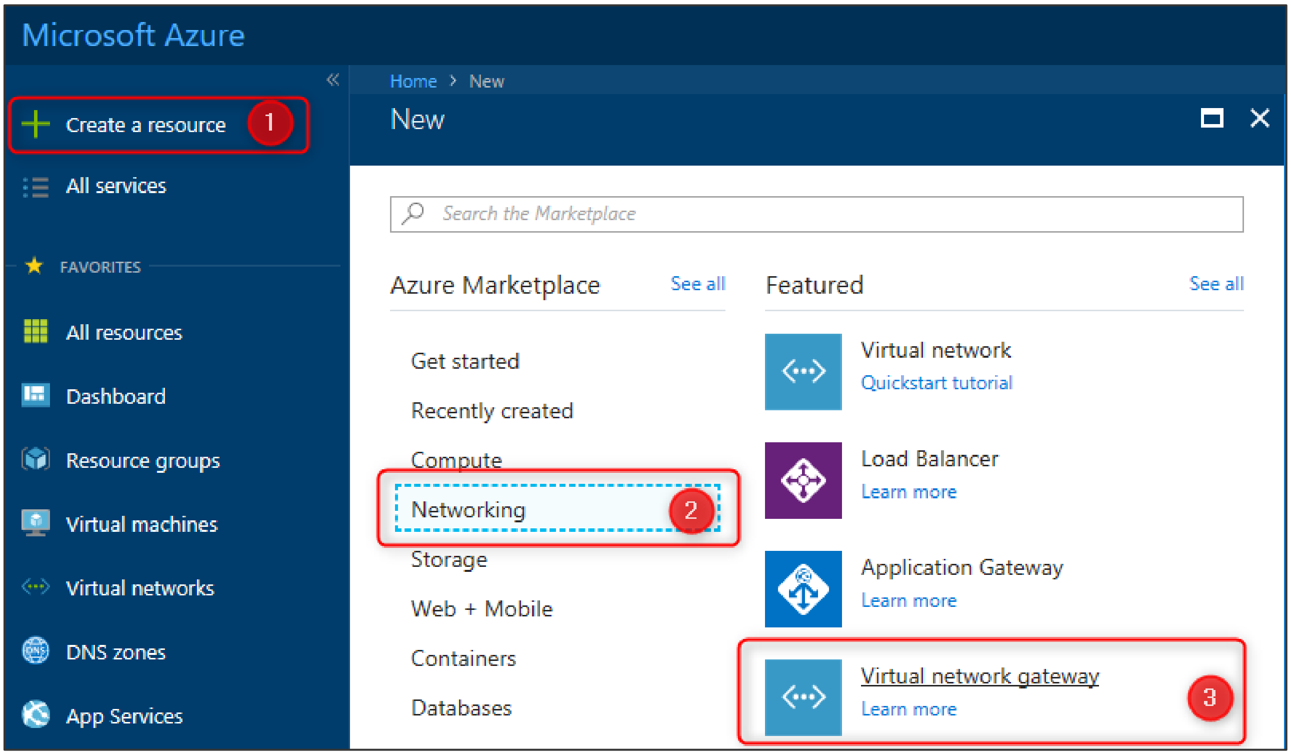


9. Click **OK** on the new blade that opens to create the Gateway subnet with default settings.

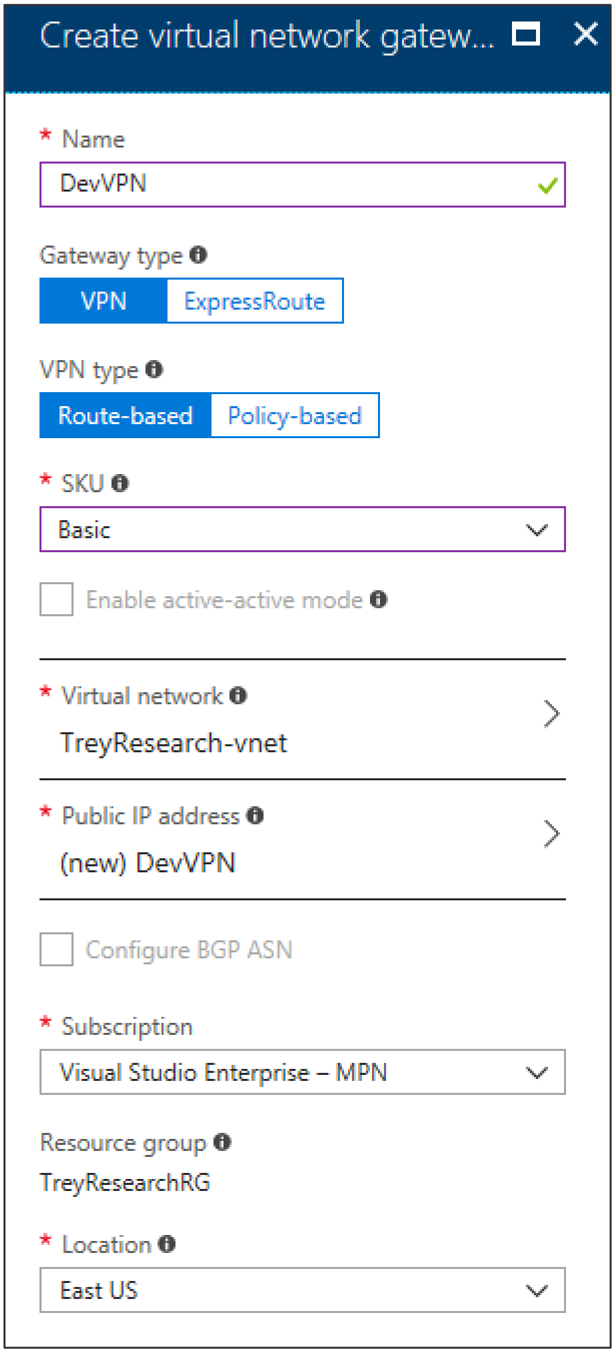
Task 2: Configure secure VPN for connectivity

In this task, you will start provisioning of a VPN gateway that will be used for secure connectivity for Trey Research.

1. Click **Crate a resource** > **Networking > Virtual network gateway.**



2. Name the VPN Gateway **DevVPN**, select the existing **TreyResearchVNET** virtual network, specify the Basic SKU, and specify a new Public IP address named **DevVPN** (again, with the Basic SKU). Note: The Basic SKU is considered a legacy SKU. The Basic SKU has certain feature limitations, it is for Dev-test or proof of concept. You can't resize a gateway that uses a Basic SKU to one of the new gateway SKUs.



Create virtual network gatew...

* Name
DevVPN

Gateway type
VPN | ExpressRoute

VPN type
Route-based | Policy-based

* SKU
Basic

□ Enable active-active mode

* Virtual network
TreyResearch-vnet

* Public IP address
(new) DevVPN

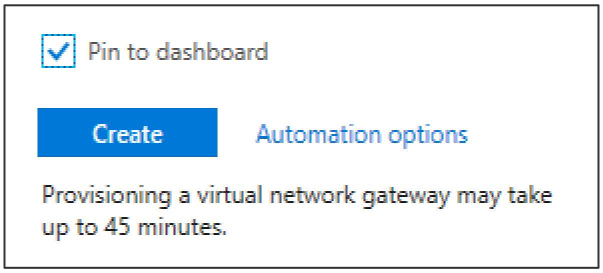
□ Configure BGP ASN

* Subscription
Visual Studio Enterprise – MPN

Resource group
TreyResearchRG

* Location
East US

3. Select **Pin to dashboard**, and click **Create** to start provisioning the VPN gateway.



☑ Pin to dashboard

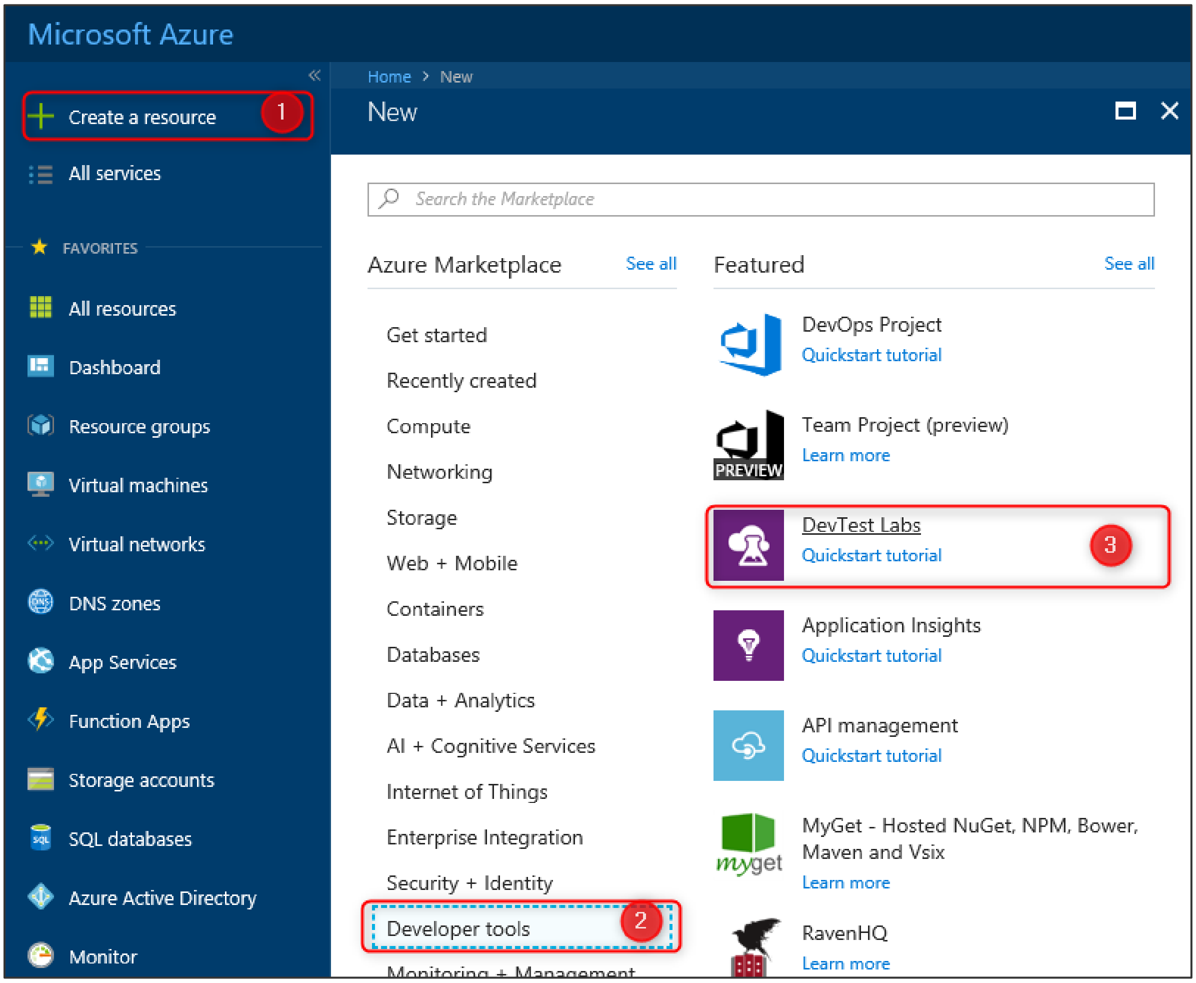
Provisioning a virtual network gateway may take up to 45 minutes.

This step will take up to 45 minutes to complete. Continue to the next task. Gateway configuration will be continued in a later task.

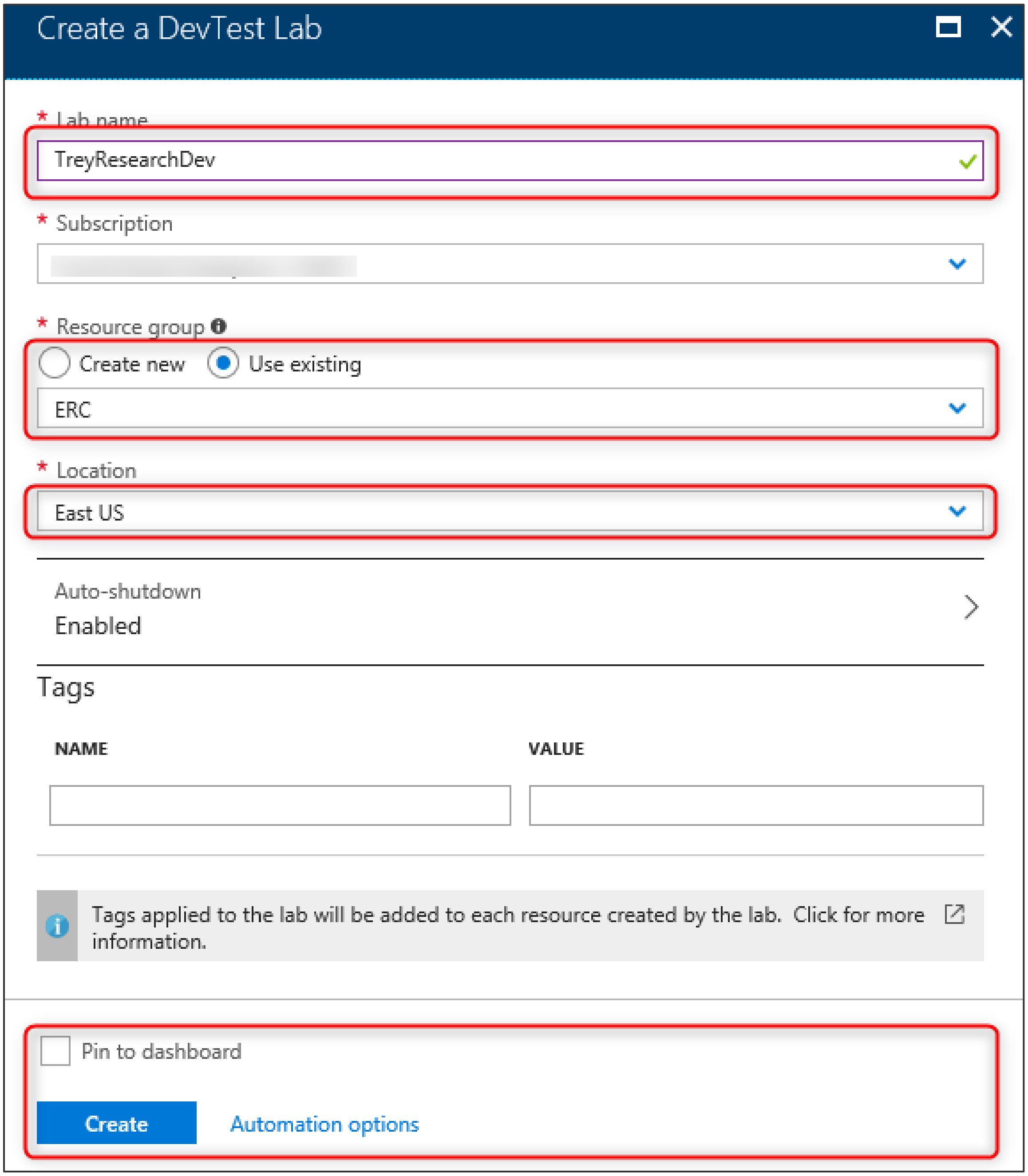
Task 3: Create an Azure DevTest lab environment

In this task, you will create and configure a new development environment for Trey Research developers and contingent staff.

1. Click **Create a resource** > **Developer tools** > **DevTest Labs**.



2. Name the lab **TreyResearchDev-vnet**, and specify the same region you deployed the virtual network to.



Create a DevTest Lab

* Lab name: TreyResearchDev

* Subscription

* Resource group: ○ Create new ● Use existing

ERC

* Location: East US

Auto-shutdown
Enabled

Tags

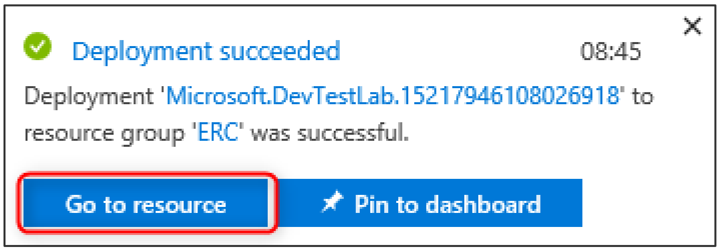
NAME	VALUE

Tags applied to the lab will be added to each resource created by the lab. Click for more information.

□ Pin to dashboard

When done, click **Create**.

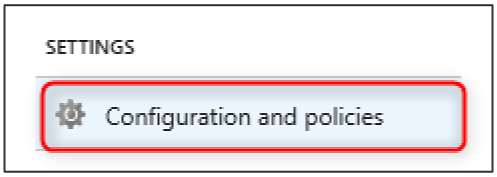
3. Open the DevTest lab environment following the completion of provisioning.



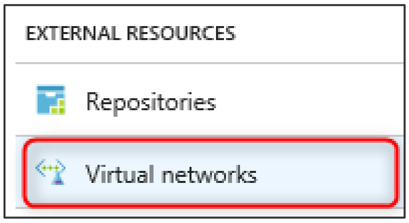
Deployment succeeded 08:45

Deployment 'Microsoft.DevTestLab.15217946108026918' to resource group 'ERC' was successful.

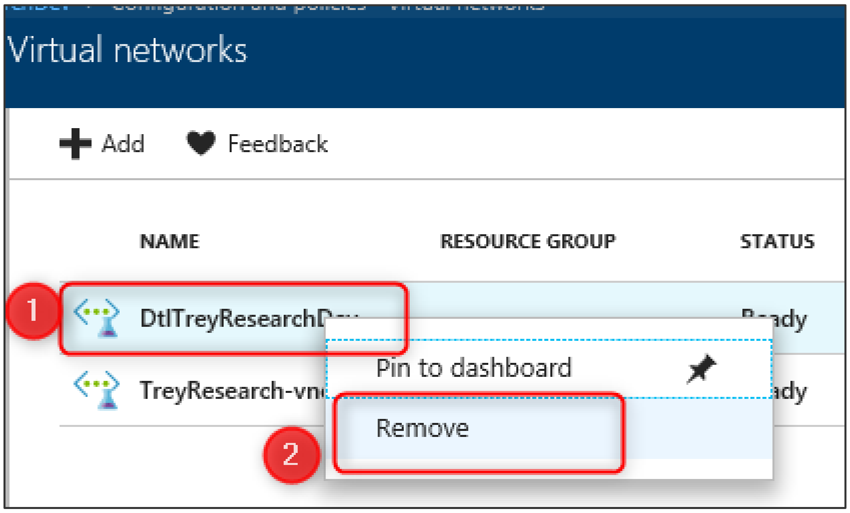
4. Open **Settings** > **Configuration and policies**.



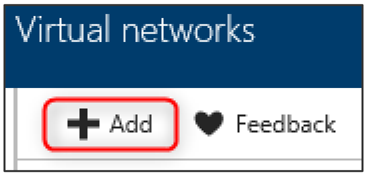
5. Click **Virtual Networks**.



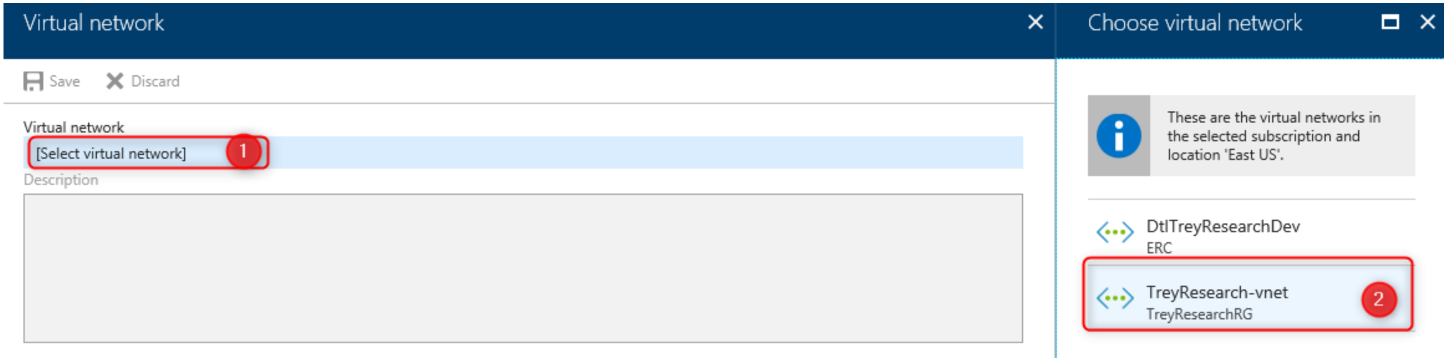
Right-click the default virtual network created for the DevTest lab environment, and click **Remove**.



6. Click the **Add** button.

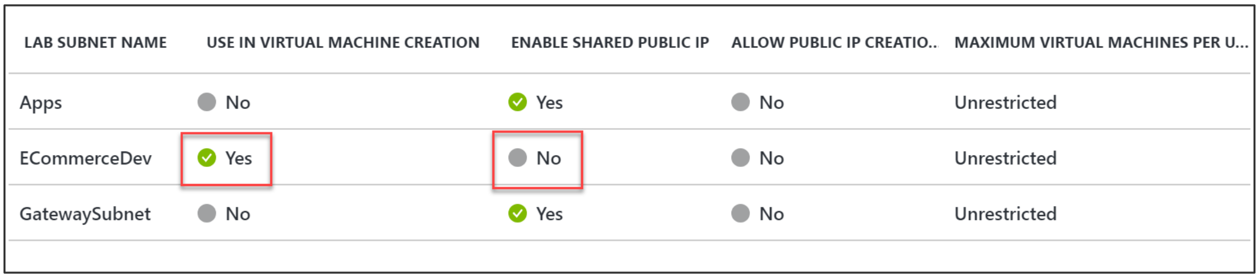


7. Click **[Select virtual network]**, and select **TreyResearch-vnet**



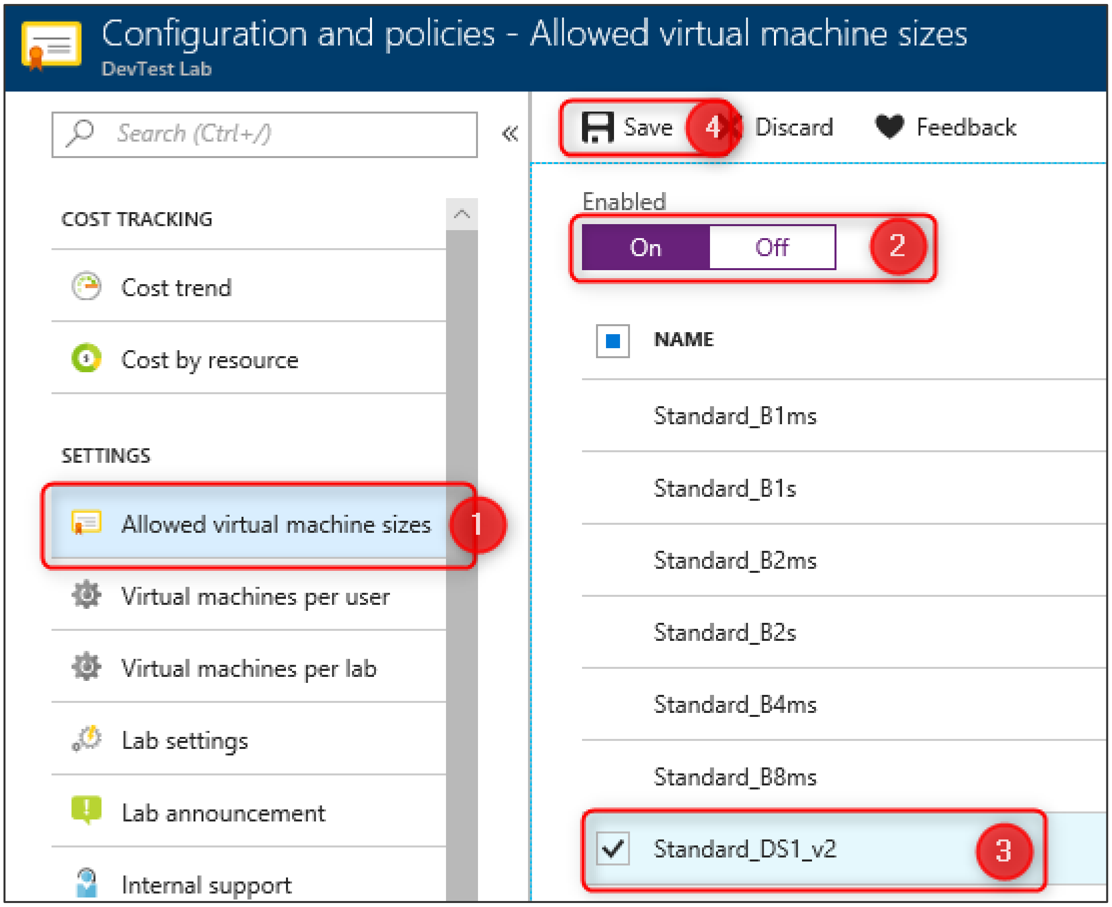
These are the virtual networks in the selected subscription and location 'East US'.

8. Configure the ECommerceDev subnet to **allow USE IN VIRTUAL MACHINE CREATION**, and to **disable ENABLE SHARED PUBLIC IP**. Then, click **Save** and close the blade.



LAB SUBNET NAME	USE IN VIRTUAL MACHINE CREATION	ENABLE SHARED PUBLIC IP	ALLOW PUBLIC IP CREATIO..	MAXIMUM VIRTUAL MACHINES PER U...
Apps	● No	✓ Yes	● No	Unrestricted
ECommerceDev	✓ Yes	● No	● No	Unrestricted
GatewaySubnet	● No	✓ Yes	● No	Unrestricted

9. Configure a virtual machine policy for this DevTest lab by clicking **Allowed virtual machine sizes**, select **Standard_DS2_v2 (or Standard_DS2)**, and click **Save**.

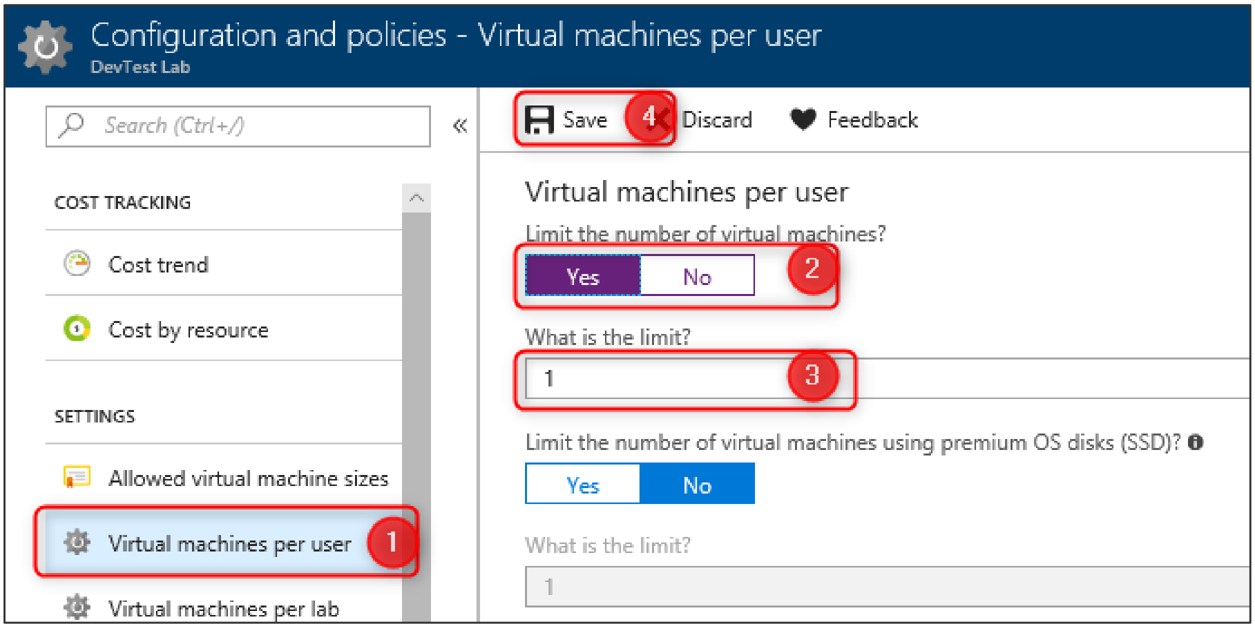


Configuration and policies - Allowed virtual machine sizes
DevTest Lab

Enabled: On | Off (On selected)

	NAME
	Standard_B1ms
	Standard_B1s
	Standard_B2ms
	Standard_B2s
	Standard_B4ms
	Standard_B8ms
✓	Standard_DS1_v2

10. Enable the virtual machines per user policy. Set a maximum number of virtual machines per user to one, and click **Save**.



Configuration and policies - Virtual machines per user
DevTest Lab

Virtual machines per user

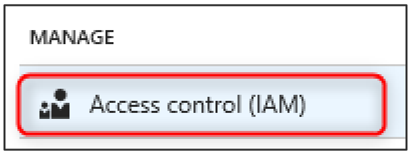
Limit the number of virtual machines? [✓ Yes] No

What is the limit? 1

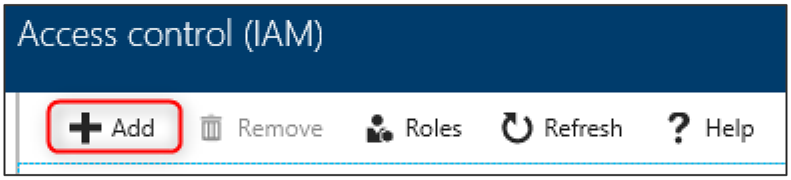
Limit the number of virtual machines using premium OS disks (SSD)? Yes [✓ No]

What is the limit? 1

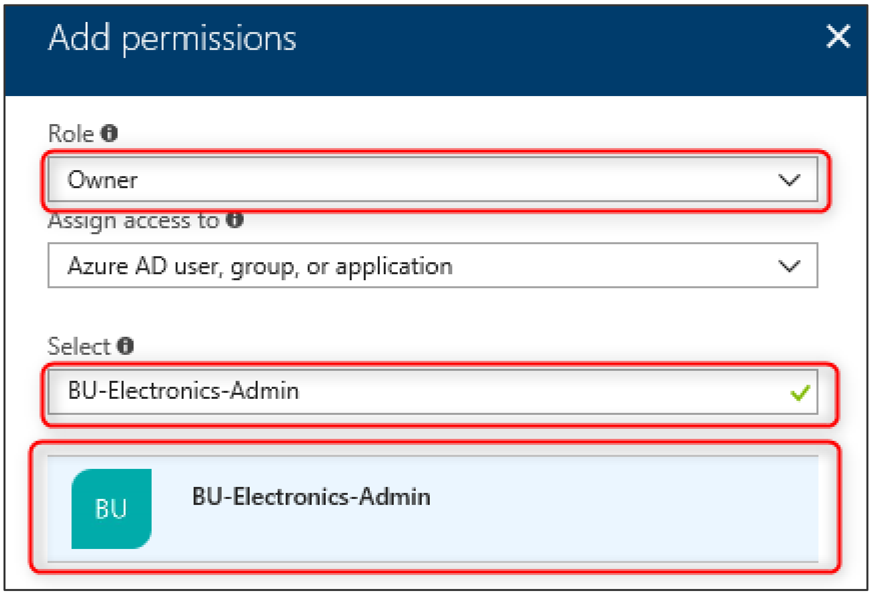
11. Allow access to the DevTest labs users by **Access control** icon within Configuration and Policies.



12. Click **+Add**



13. Select **Role** as **Owner**, type 'BU-Electronics-Admin into the search field and select the **BU-Electronics-Admin** group.



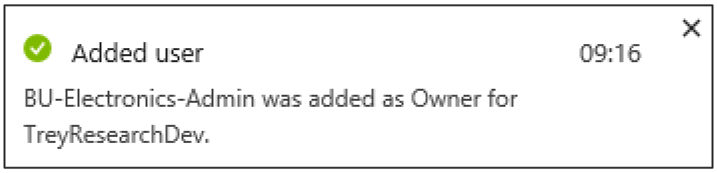
Add permissions

Role: Owner

Assign access to: Azure AD user, group, or application

Select: BU-Electronics-Admin ✓

Then click **Save**.



✓ Added user 09:16

BU-Electronics-Admin was added as Owner for TreyResearchDev.

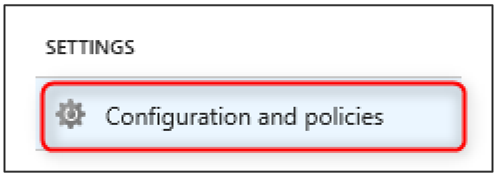
Task 4: Test access to the DevTest labs environment

In this task, you will use the ElectronicsAdmin user account to grant access to the developer environment. Then, you will validate as a user whether access was successfully granted.

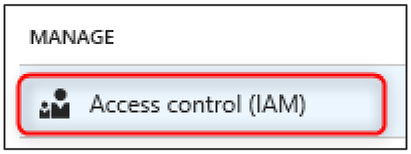
1. Sign in to the Azure Management Portal as the **ElectronicsAdmin** user account.

2. Open the DevTest labs environment by clicking **All services** > **DevTest Labs** > **TreyResearchDev**.

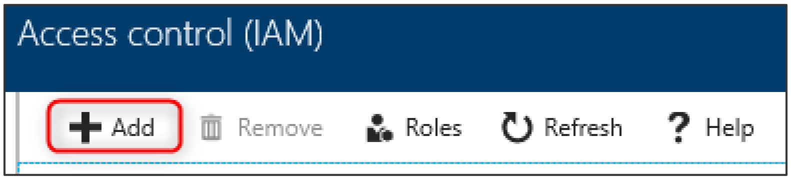
3. Click **Configuration and policies.**



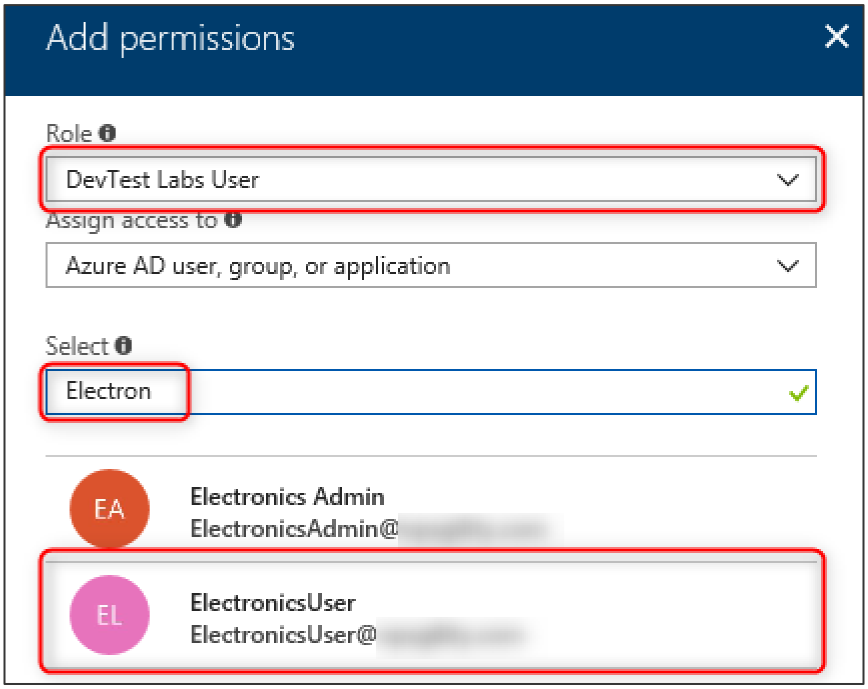
4. Allow access to the DevTest Labs users by clicking the **Access control** icon within Configuration and Policies.



14. Click **+Add**



5. Specify the **Role** as **DevTest Labs User**. Use the search field to find the **ElectronicsUser** account, and select it. Then **Save**.



Role: DevTest Labs User

Assign access to: Azure AD user, group, or application

Select: Electron ✓

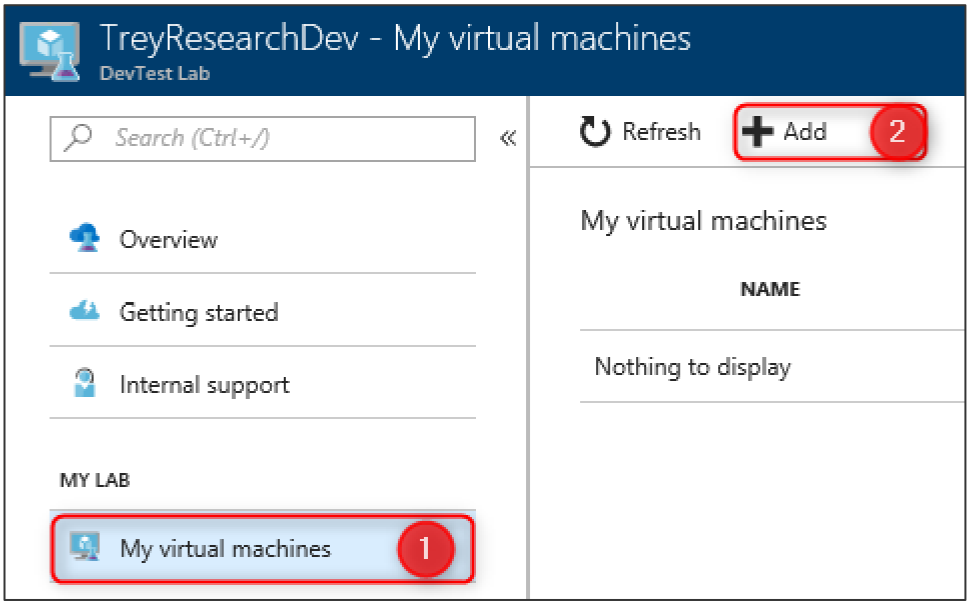
Electronics Admin
ElectronicsAdmin@[redacted]

ElectronicsUser
ElectronicsUser@[redacted]

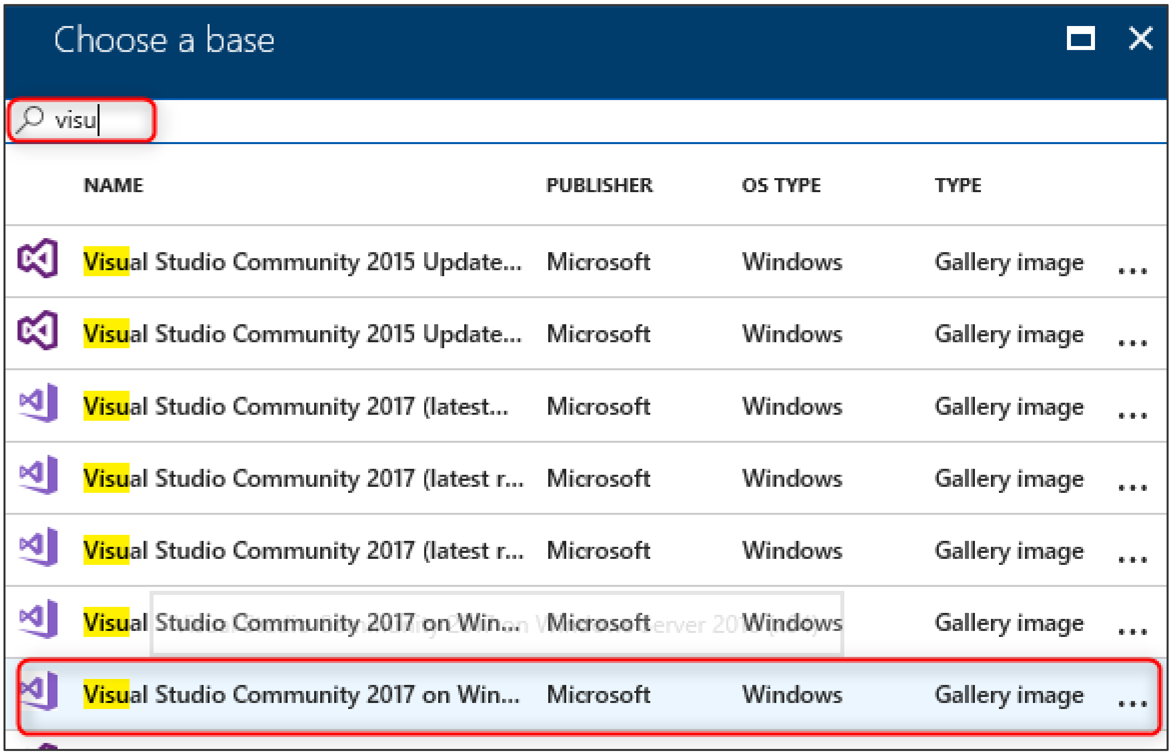
6. Close your browser and sign in with the **ElectronicsUser** account. You will have to change your password and setup a recovery mechanism with this account.

7. Open the DevTest labs environment by clicking **All services** > **DevTest Labs** > **TreyResearchDev**.

8. Click **My virtual machines**, then **+Add** provision a virtual machine for the developer.

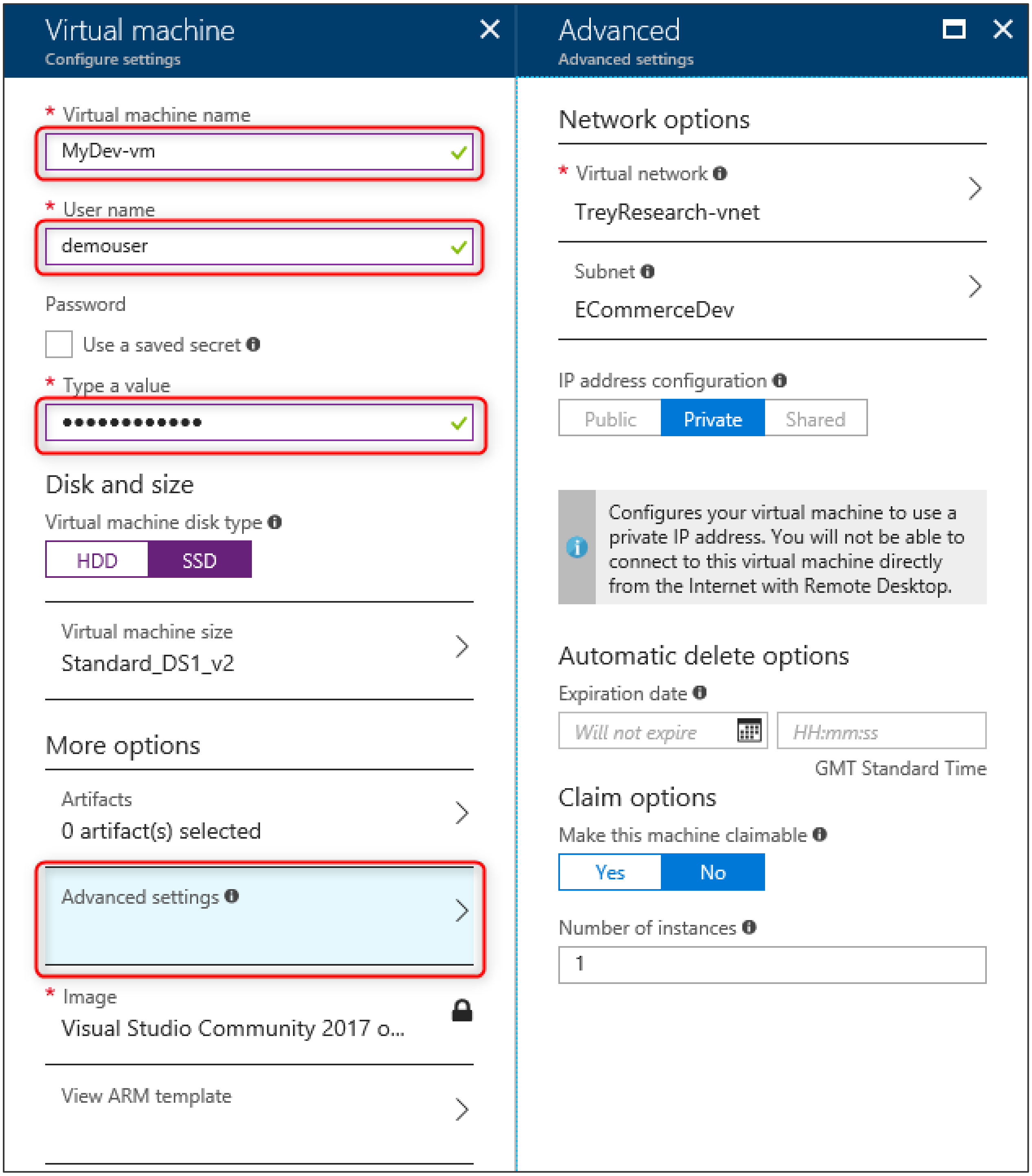


9. On the **Choose a Base** blade, select the **Visual Studio Community 2017 on Windows Server 2016 (x64)** image.



NAME	PUBLISHER	OS TYPE	TYPE	
Visual Studio Community 2015 Update...	Microsoft	Windows	Gallery image	...
Visual Studio Community 2015 Update...	Microsoft	Windows	Gallery image	...
Visual Studio Community 2017 (latest...	Microsoft	Windows	Gallery image	...
Visual Studio Community 2017 (latest r...	Microsoft	Windows	Gallery image	...
Visual Studio Community 2017 (latest r...	Microsoft	Windows	Gallery image	...
Visual Studio Community 2017 on Win...	Microsoft	Windows	Gallery image	...
Visual Studio Community 2017 on Win...	Microsoft	Windows	Gallery image	...

10. Specify the **virtual machine name** as well as a **user name** and **password**. Click **Advanced Settings** and note the VM size, IP address configuration, virtual network, and subnet are not changeable by the user.



Virtual machine
Configure settings

* Virtual machine name
MyDev-vm ✓

* User name
demouser ✓

Password
□ Use a saved secret

* Type a value
●●●●●●●●●●●●● ✓

Disk and size
Virtual machine disk type
HDD [✓ SSD]

Virtual machine size
Standard_DS1_v2

More options

Artifacts
0 artifact(s) selected

Advanced settings

* Image
Visual Studio Community 2017 o...

View ARM template

Advanced
Advanced settings

Network options

* Virtual network
TreyResearch-vnet

Subnet
ECommerceDev

IP address configuration
Public [✓ Private] Shared

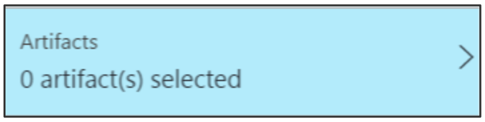
Configures your virtual machine to use a private IP address. You will not be able to connect to this virtual machine directly from the Internet with Remote Desktop.

Automatic delete options
Expiration date
Will not expire HH:mm:ss
GMT Standard Time

Claim options
Make this machine claimable
Yes [✓ No]

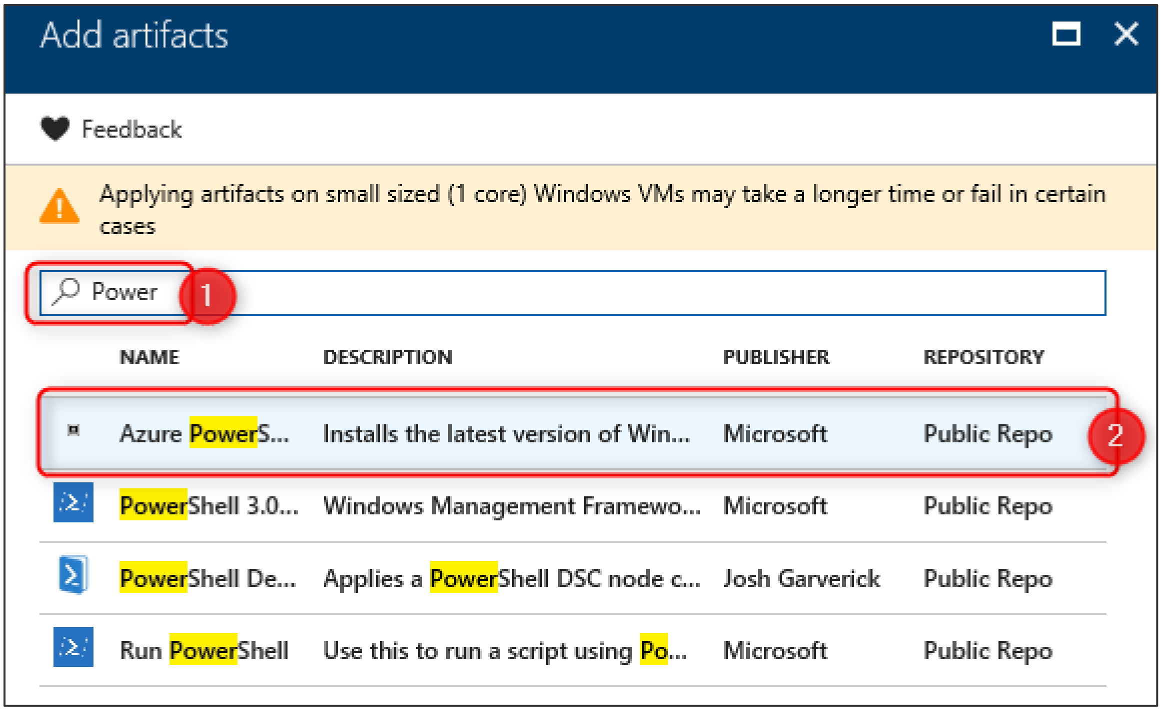
Number of instances
1

11. Click **Artifacts**.



Artifacts
0 artifact(s) selected

12. Add **Azure PowerShell** to the artifacts of the VM by selecting their names and clicking **ADD**. Click **OK** at the bottom of the **Add artifacts** blade when complete.



Add artifacts

Applying artifacts on small sized (1 core) Windows VMs may take a longer time or fail in certain cases

NAME	DESCRIPTION	PUBLISHER	REPOSITORY
Azure PowerS...	Installs the latest version of Win...	Microsoft	Public Repo
PowerShell 3.0...	Windows Management Framewo...	Microsoft	Public Repo
PowerShell De...	Applies a PowerShell DSC node c...	Josh Garverick	Public Repo
Run PowerShell	Use this to run a script using Po...	Microsoft	Public Repo

13. Click **Create** to provision the virtual machine.

NOTE: If you receive a Policy Error make sure that the VM Auto Shutdown Policy is enabled in your Allowed Resources.

Task 5: Finish configuring secure connectivity

In this task, you will configure certificates for the VPN gateway and for the end users as well as complete configuration of the VPN gateway. You will then configure and test access to the development environment.

Subtask 1: Create certificates for point-to-site VPN

1. Download makecert.exe from: https://cloudworkshop.blob.core.windows.net/enterprise-ready-cloud/makecert.exe and save it to the C:\Hackathon\ERC folder.

2. Launch a command prompt (run cmd.exe), and navigate to the **C:\Hackathon\ERC** folder by typing in the following command:

```
CD C:\Hackathon\ERC
```

3. Execute the following command to generate a root certificate for configuring a point-to-site VPN gateway.

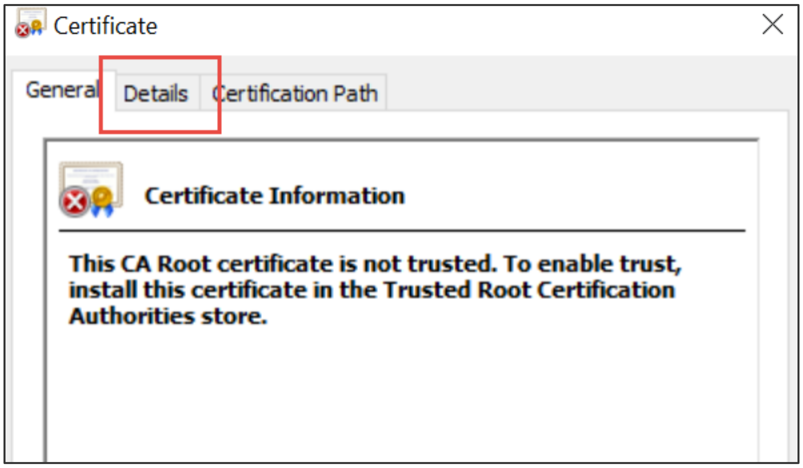
```
makecert -sky exchange -r -n "CN=P2SROOT" -pe -a sha1 -len 2048 -ss My .\P2SRoot.cer
```

4. Execute the following command to generate a client certificate:

```
makecert.exe -n "CN=P2SClient" -pe -sky exchange -m 96 -ss My -in "P2SRoot" -is my -a sha1
```

5. Using **File Explorer**, navigate to the C:\Hackathon\ERC folder, and double-click the **P2SRoot.cer** file.

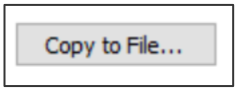
6. Click the **Details** tab of the certificate.



Certificate Information

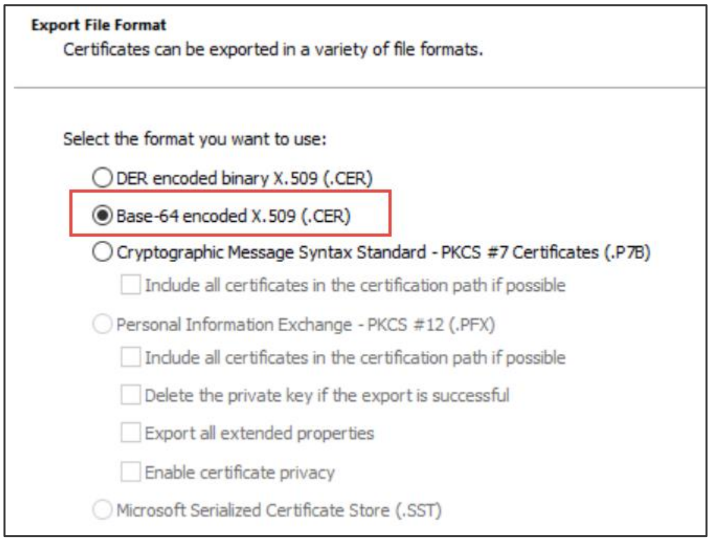
This CA Root certificate is not trusted. To enable trust, install this certificate in the Trusted Root Certification Authorities store.

7. Click **Copy to File**.



Click **NEXT**

8. Change the encoding type to **Base-64 encoded X.509 (.CER)**, and click **Next**.

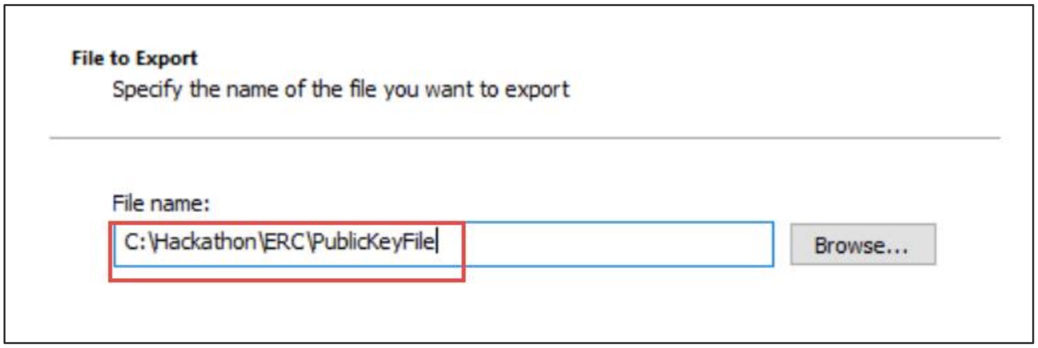


Export File Format
Certificates can be exported in a variety of file formats.

Select the format you want to use:

○ DER encoded binary X.509 (.CER)
● Base-64 encoded X.509 (.CER)
○ Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)
□ Include all certificates in the certification path if possible
○ Personal Information Exchange - PKCS #12 (.PFX)
□ Include all certificates in the certification path if possible
□ Delete the private key if the export is successful
□ Export all extended properties
□ Enable certificate privacy
○ Microsoft Serialized Certificate Store (.SST)

9. Specify the filename as **C:\Hackathon\ERC\PublicKeyFile**. Click **Next** and **Finish**.



File to Export
Specify the name of the file you want to export

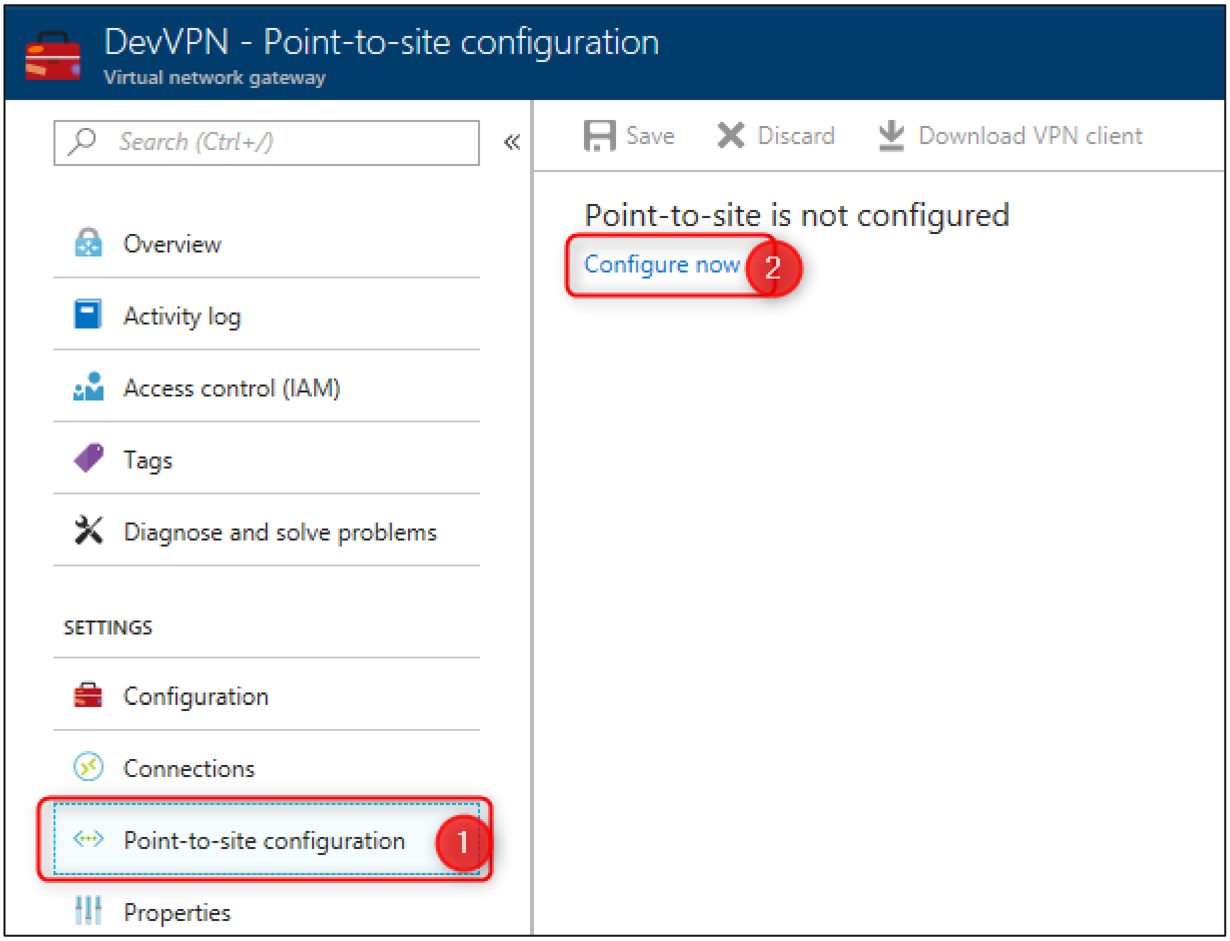
File name:
C:\Hackathon\ERC\PublicKeyFile

10. Open the newly created PublicKeyFile in Notepad, and copy the certificate text to the clipboard. Do NOT copy the first and last lines (containing -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----).

Subtask 2: Configure the VPN gateway

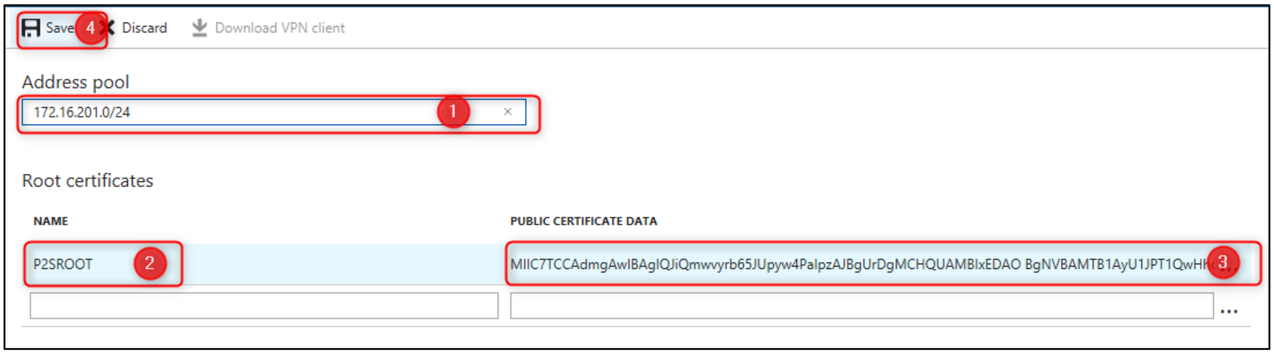
1. In the Azure Portal, make sure you are logged in as ElectronicsAdmin. Then, navigate to **All services** > **Virtual Network Gateways**, and click on the **DevVPN** gateway created earlier

2. Click **Point to site configuration**, and then **Configure Now**



Point-to-site is not configured

3. In the **DevVPN – Point to site configuration** blade, enter the following details:
   a. Address pool: 172.16.201.0/24
   b. Under Root Certificates, enter
      i. Name: P2SROOT
      ii. Public Certificate Data: Paste the certificate data copied to the clipboard earlier



Address pool: 172.16.201.0/24

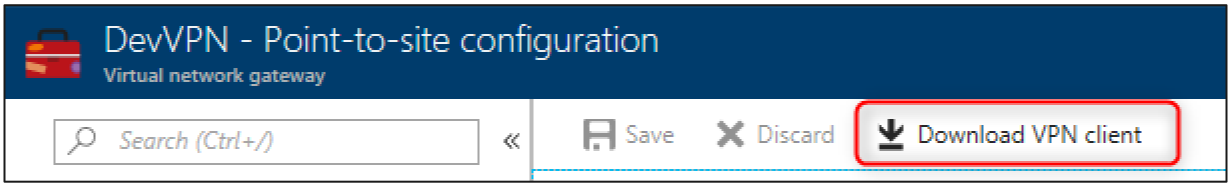
Root certificates

NAME	PUBLIC CERTIFICATE DATA
P2SROOT	MIIC7TCCAdmgAwIBAgIQJiQmwvyrb65JUpyw4PalpzAJBgUrDgMCHQUAMBIxEDAO BgNVBAMTB1AyU1JPT1QwHH

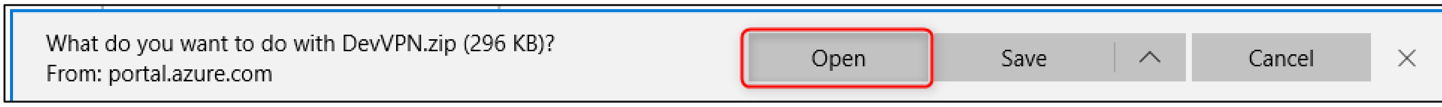
Once complete, click **Save**.

Subtask 3: Configure and test the client

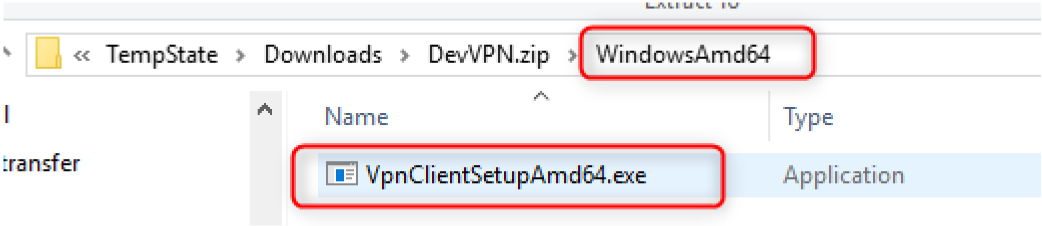
1. Once the P2S configuration has been saved, click **Download VPN Client**.



2. Click **Open**

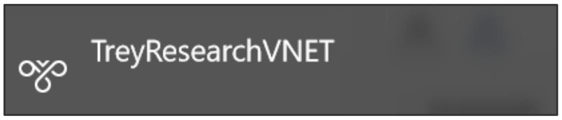

What do you want to do with DevVPN.zip (296 KB)?
From: portal.azure.com

3. Windows Explorer should open, showing the contents of the downloaded ZIP file. Open the **WindowsAmd64folder**, and copy the **VpnClientSetupAmd64.exe** file to the desktop.

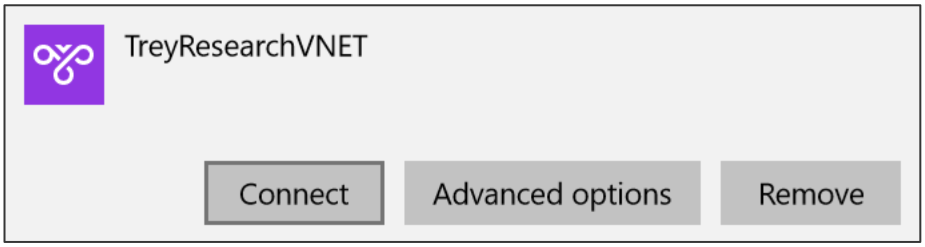


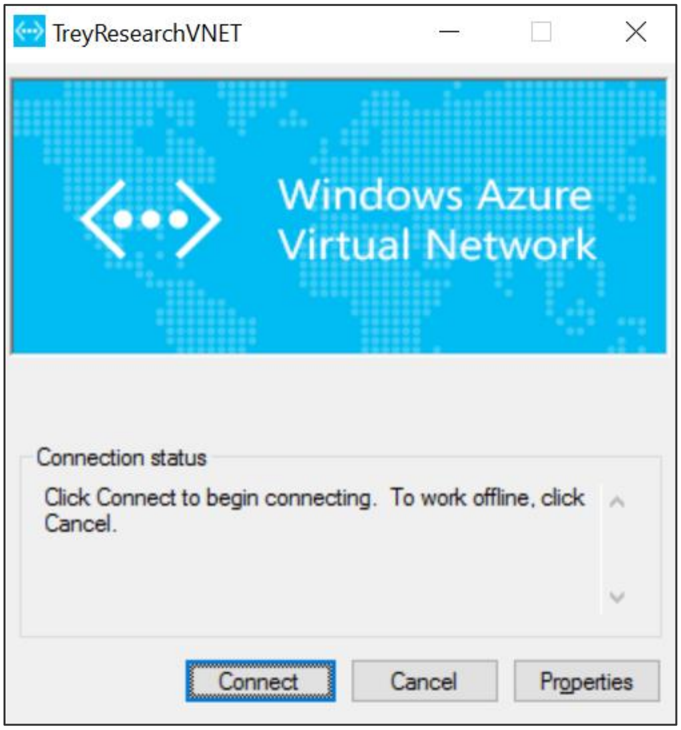
4. Run the **VpnClientSetupAmd64.exe** installer from the desktop. Accept any confirmation prompts.

5. The client computer should now have a new connection option in the same location as new wireless connections. Click the **TreyResearchVNET** icon to launch the connection.



6. Click **Connect** to initiate connection.

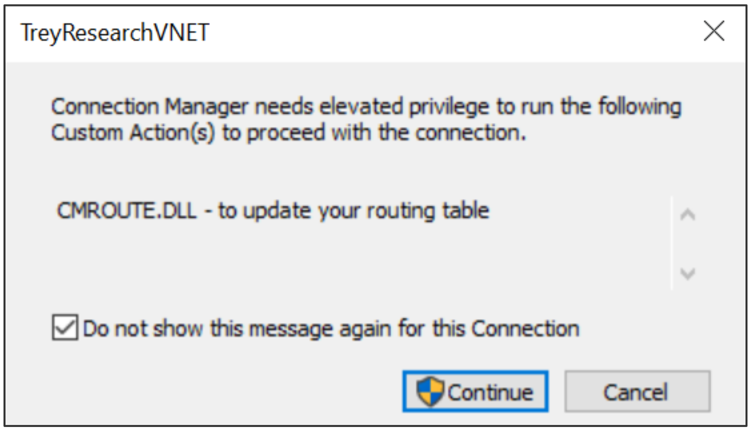




Connection status

Click Connect to begin connecting. To work offline, click Cancel.

7. Select **Do not show this message again for this Connection**, and click **Continue**.



Connection Manager needs elevated privilege to run the following Custom Action(s) to proceed with the connection.

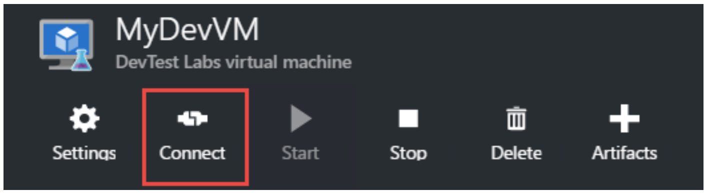
CMROUTE.DLL - to update your routing table

☑ Do not show this message again for this Connection

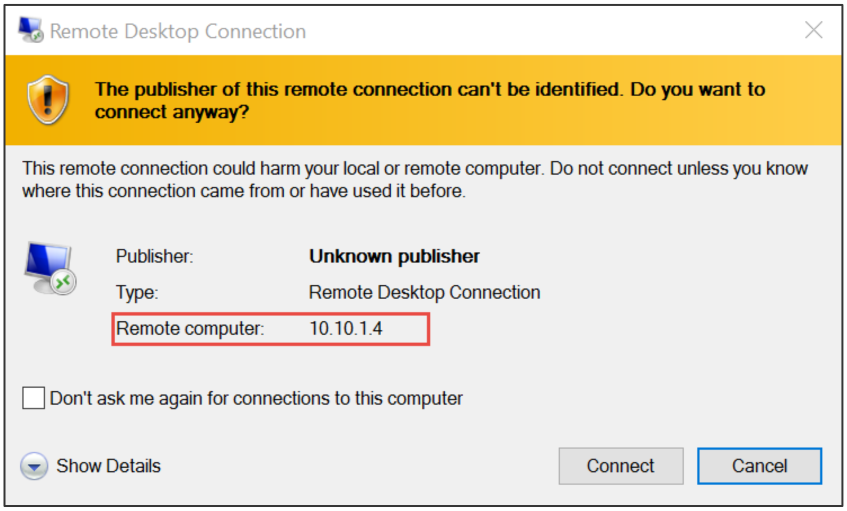
8. After you are successfully connected, switch back to the Azure Management Portal using the **ElectronicsUser** account.

9. Browse to **DevTest Labs**, open **TreyResearchLab**, and click **MyDevVM**.

10. Click **Connect** to initiate a connection with the virtual machine.



11. Note the remote computer is connecting over a Private IP address.



The publisher of this remote connection can't be identified. Do you want to connect anyway?

This remote connection could harm your local or remote computer. Do not connect unless you know where this connection came from or have used it before.

Publisher:	**Unknown publisher**
Type:	Remote Desktop Connection
Remote computer:	10.10.1.4

□ Don't ask me again for connections to this computer

After the hands-on lab

Duration: 10 minutes

After completing the hands-on lab, you will remove the policies on your subscription.

Task 1: Remove resources and configuration created during this lab

You should follow all steps provided *after* attending the hands-on lab.

1. Log in to the Azure portal.
2. Navigate to the **Policy** blade.
3. Click on **Assignments** and delete all policy assignments created during this lab
4. Click on **Definitions** and delete any policy definitions or initiative definitions created during this lab
5. Navigate to the **Management Groups** blade.
6. Remove any Management Groups crated during this lab.
7. Navigate to the **Resource Groups** blade.
8. Remove any resource groups created during this lab. This will also delete any resources in those resource groups.
9. Navigate to the **Azure Active Directory** blade
10. Remove any users and groups created during this lab.