Microsoft Cloud Workshops

Enterprise-ready cloud hackathon

Lab guide

February 2017

© 2017 Microsoft Corporation. All rights reserved. This document is confidential and proprietary to Microsoft. Internal use only. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples are for illustration only and are fictitious. No real association is intended or inferred.

## Contents

[Enterprise-ready cloud hackathon 1](#_Toc473899302)

[Enterprise-ready cloud hackathon answers 11](#_Toc473899303)

# Enterprise-ready cloud hackathon

## Overview

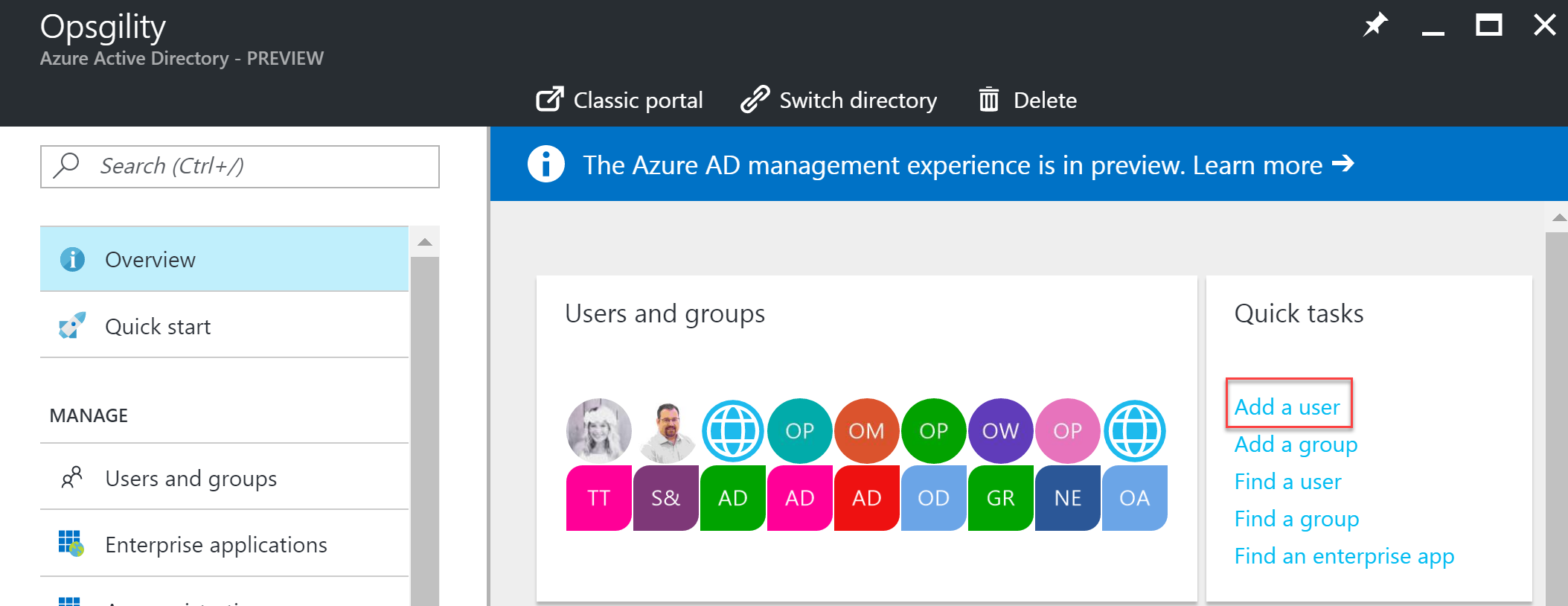
In this hackathon, you are working with Trey Research to setup some best practices regarding policies, permissions, and remote access to their network. Some tasks will include creating scripts that Enterprise IT will use to automatically set policy and delegate permissions when a new subscription is 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 Community Edition
  + Azure SDK 2.9 for Visual Studio
    - <http://go.microsoft.com/fwlink/?linkid=518003&clcid=0x409>

## **Read Before Continuing**

To complete this lab, you must have full global admin access to the Azure AD tenant associated with your Azure subscription. To validate, login to <http://portal.azure.com>, click on **More Services** and type in **Azure Active Directory.** If you can open the Azure Active Directory tenant, click **Add a User** and attempt to add a test user. If you do not have permissions you will not be able to complete this lab using your current subscription.



## Lab structure

This lab has two sets of instructions. The first is a high-level set of instructions that is designed for attendees that have previous experience with managing Microsoft Azure governance features including management tools like the Azure PowerShell cmdlets.

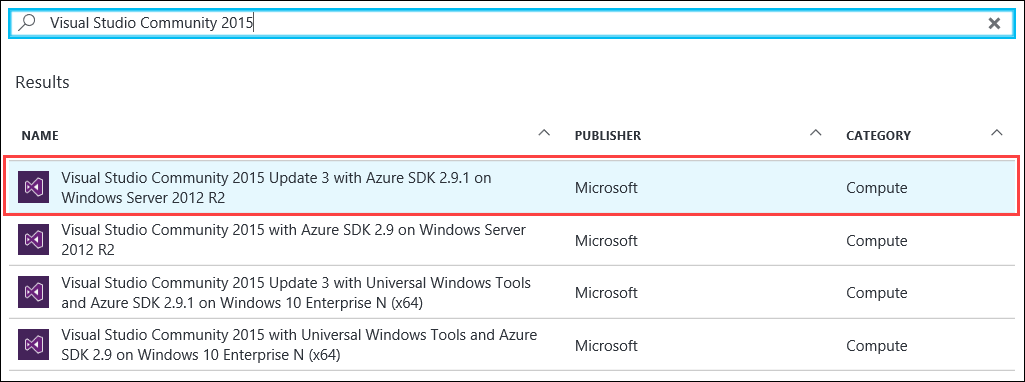
## Exercise 1: Environment setup

In this exercise, you will set up your environment for use for the rest of the exercises. This exercise will involve creating a virtual machine with the correct tools in place.

### Task 1: Setup a development environment

If you do not have a machine setup with Visual Studio 2015 Community Update 3 and Azure SDK 2.9.+, complete this task.

1. Create a virtual machine in Azure using the Visual Studio Community 2015 Update 3 and SDK 2.9.+ on Windows Server 2012 R2 image.



We *highly* recommended using a DS2 or D2 instance size for this virtual machine (VM).

### Task 2: Upgrade Azure PowerShell

1. Use Web Platform Installer to upgrade the Azure PowerShell to latest available version.

* <https://www.microsoft.com/web/handlers/webpi.ashx/getinstaller/WindowsAzurePowershellGet.3f.3f.3fnew.appids>

## Exercise 2: Create the policy for Enterprise IT

In this exercise, you will create several policy files that can be applied to a subscription to ensure that users stay within the scope of supported services for Enterprise IT. You will create a PowerShell script to apply the policies.

### Help references

|  |  |
| --- | --- |
| Azure Resource Manager Policy | https://azure.microsoft.com/en-us/documentation/articles/resource-manager-policy/ |
| PowerShell and Azure Resource Manager | https://azure.microsoft.com/en-us/documentation/articles/powershell-azure-resource-manager/ |

### Task 1: Create the service catalog policy

In this exercise, you will create a new Azure Resource Manager policy to restrict services to the supported list provided by Trey Research.

#### Tasks to complete

* Create an ARM policy that only allows the following resources
  + Virtual machines
  + ExpressRoute
  + VPN gateways
  + Storage
  + virtual networks
  + Backup
  + Site recovery
  + DevTest labs
  + Key vault
  + Azure App Services
  + SQL Database

#### Exit criteria

* Create a file called ServiceCatalog.json that has the appropriate code to restrict the services. This file should be in the C:\Hackathon\ERC folder.

### Task 2: Restrict the creation of ExpressRoute circuits

In this exercise, you will create a new Azure Resource Manager policy to restrict the creation of an ExpressRoute circuit.

#### Tasks to complete

* Create an ARM policy that blocks the creation of ExpressRoute circuits (must still be able to use one).

#### Exit criteria

* Create a file called RestrictERCircuit.json that has the appropriate code to restrict the ability to create an ExpressRoute circuit. This file should be in the C:\Hackathon\ERC folder.

### Task 3: Restrict the creation of resources in regions

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

#### Tasks to complete

* Create an ARM policy that only allows the creation of resources in the following regions:
  + Primary: East United States, Failover: West United States
  + Primary: West Europe, Failover: North Europe
  + Primary: Japan West, Failover: Japan East

#### Exit criteria

* Create a file called RestrictRegions.json that has the appropriate code to restrict the available regions. This file should be in the C:\Hackathon\ERC folder.

### Task 4: Apply the Policies

In this task, you will create a reusable script that can be used to apply policy to a new subscription in Trey Research.

#### Tasks to complete

* Create a reusable PowerShell script called ConfigureSubscription.ps1 that can take the subscription ID as a parameter and apply the policy files to the specified subscription.

#### Exit criteria

* A PowerShell named ConfigureSubscription.ps1 that can apply the policy files in the previous tasks to a subscription. This script should take the subscription ID as a parameter.

### Task 5: Test the Policies

In this task, you will use the Azure management portal to validate the policies work and understand how to identify policy events.

#### Tasks to complete

* Apply the ARM policies to the subscription and ensure that the policies work as expected.

#### Exit criteria

* You should receive an error if you create a storage account in an unsupported region.
* You should be able to create a storage account (or another supported service) in a supported region.
* You should receive an error if you create a media service account due to it not being in the supported service list.
* You should not be able to create an ExpressRoute circuit

## Exercise 3: Configure delegated permissions

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://azure.microsoft.com/en-us/documentation/articles/active-directory-create-users/ |
| How Subscriptions are associated with Azure AD | https://azure.microsoft.com/en-us/documentation/articles/active-directory-how-subscriptions-associated-directory/ |
| Managing Azure AD Security Groups | https://azure.microsoft.com/en-us/documentation/articles/active-directory-accessmanagement-manage-groups/ |
| Role Based Access Control | https://azure.microsoft.com/en-us/documentation/articles/role-based-access-control-configure/ |
| Manage RBAC with PowerShell | https://azure.microsoft.com/en-us/documentation/articles/role-based-access-control-manage-access-powershell/ |

### Task 1: Create user accounts in Azure AD

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

#### Tasks to complete

* Within Azure AD create two users
  + ElectronicsAdmin@[your tenant name].onmicrosoft.com
  + ElectronicsUser@[your tenant name].onmicrosoft.com

#### Exit criteria

* Two users that will be used to test delegated permissions using role based access control (RBAC).

### Task 2: 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.

#### Tasks to complete

* Within Azure AD create two groups
  + BU-Electronics-Admin– add the ElectronicsAdmin user to the group
  + BU-Electronics-Users – add the ElectronicsUser user to this group

#### Exit criteria

* Two groups with users that will be used to test delegated permissions using role based access control (RBAC).

### Task 3: Configure delegated permissions

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

#### Tasks to complete

* Extend ConfigureSubscription.ps1 to accept the Azure AD Group to add to the subscriptions contributor role whenever the subscription is configured.
* Use the modified script (not the portal) to add the BU-Electronics-Admin group to your subscription in the contributor role.

#### Exit criteria

* A modified script that will add the BU-Electronics-Admin group to the contributor role automatically.

### Task 4: Enable project based delegation and chargeback

In this task, you will create a script that can create a new resource group as the owner of the subscription and delegate owner permissions to the BU-Electronics-Admin for a specific project by applying permissions at the resource group level. This task will also include automatically applying a tag to all resources in the resource group to track them by their IO code.

#### Tasks to complete

* Create a new ARM policy that will apply a new tag named ioCode with a project IO code placeholder value that will be added all resources for the project.
* Create a new PowerShell script called CreateProjectResourceGroup.ps1 that will create a new resource group with the following settings applied:
  + Accepts an Azure AD Group Name as a parameter that will be assigned to the owner role of the new resource group.
  + Accepts a project IO code value (for example 1000150) that will add the ioCode tag to all resources created within the resource group.
* Using the new script, create a new resource group called DelegatedProjectDemo that has the BU-Electronics-Admin group be added to the subscription in the owner role and has the policy applied and has the ioCode policy applied with the ioCode parameter set to 1000150.
* Create a new storage account within the resource group and validate the new tag is applied.

#### Exit criteria

* A reusable script that will accept the subscription ID, an IOCode value, and the group to add as parameters. When the script is executed, it should create the resource group and automatically add the group to the owner role.

## Exercise 4: Create the environment for the e-commerce team

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://azure.microsoft.com/en-us/documentation/articles/vpn-gateway-howto-point-to-site-rm-ps/ |
| Network Security Groups | https://azure.microsoft.com/en-us/documentation/articles/virtual-networks-nsg/ |
| Azure DevTest Labs | https://azure.microsoft.com/en-us/documentation/services/devtest-lab/ |

### Task 1: Create a new virtual network

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

#### Tasks to complete

* Create a new virtual network with the following properties:
  + Name: TreyResearchVNET
  + Address Space: 10.10.0.0/16
  + Subnet Name: Apps
  + Subnet address Range: 10.10.0.0/24
  + Resource Group: TreyResearchRG
* Add an additional subnet to the virtual network
  + Name: ECommerceDev
  + Address Space: 10.10.1.0/24
* Add a gateway subnet

#### Exit criteria

* A virtual network named TreyResearchVNET with the correct settings noted above.

### Task 2: Configure limited network access

In this task, you will configure traffic from the ECommerceDev subnet to be restricted from the Apps subnet.

#### Tasks to complete

* Configure outbound traffic from the ECommerceDev subnet to be restricted from the Apps subnet.

#### Exit criteria

* Virtual machines booted in the ECommerceDev subnet should not be able to access virtual machines or other services in the Apps subnet.

### Task 3: 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.

#### Tasks to complete

* Configure a dynamic VPN gateway on the virtual network that will allow individual developers to connect securely and access virtual machines by their private IP address.

#### Exit criteria

* A VPN gateway (dynamic) in the provisioning state.

### Task 4: Create an Azure DevTest labs environment

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

#### Tasks to complete

* Create an environment that developers can use to provision and access virtual machines.

#### Exit criteria

* The environment should only allow virtual machines to deploy to the ECommerceDev subnet.
* Virtual machines should not be allowed to be deployed on any subnet other than ECommerceDev
* The only virtual machines that can be provisioned are Standard\_DS2.
* Each user of the environment should only be able to deploy one VM.
* The BU-Electronics-Admin group should be allowed to grant access to the environment to other users.

### Task 5: Test access to the DevTest labs environment

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

#### Tasks to complete

* Login as the ElectronicsAdmin user and grant access to the environment to the ElectronicsUser user.
* Login as the ElectronicsUser user and provision a virtual machine that has Azure PowerShell and Fiddler 4 installed at provision time.

#### Exit criteria

* The ElectronicsUser account should only be allowed to use the environment you assigned permissions to (nothing further).
* The exit criteria put in place in task 4 should apply to this user.

### Task 6: Finish configure secure connectivity

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

#### Tasks to complete

* Complete the configuration of the secure VPN solution.
* Provision a certificate that can be used to authenticate a client to the VPN solution.
* Connect to the virtual network using the VPN solution.
* Sign in to the Azure Management Portal using the ElectronicsUser user and connect to the previously created virtual machine using its private IP address.

#### Exit criteria

* The ElectronicsUser user should be able to login to the portal, find his or her virtual machine, click **Connect** and successfully connect to its private IP address (no public IPs allowed).

# Enterprise-ready cloud hackathon answers

## Overview

In this hackathon, 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.

## Exercise 1: Environment setup

In this exercise, you will set up an environment to use for the rest of the exercises.

### Task 1: Setup a development environment

If you do not have a machine setup with Visual Studio 2015 Community Update 3 and Azure SDK 2.9.+, complete this task.

1. Create a virtual machine in Azure using the Visual Studio Community 2015 Update 3 and SDK 2.9.+ on Windows Server 2012 R2 image.

We *highly* recommend using a DS2 or D2 instance size for this VM.

### Task 2: Upgrade Azure PowerShell

1. Use Web Platform Installer to upgrade the Azure PowerShell to latest available version.

* <https://www.microsoft.com/web/handlers/webpi.ashx/getinstaller/WindowsAzurePowershellGet.3f.3f.3fnew.appids>

## Exercise 2: Create the policy for Enterprise IT

In this exercise, you will create several policy files that can be applied to a subscription to ensure that users stay within the scope of supported services for Enterprise IT. You will create a PowerShell script to apply the policies.

### Help references

|  |  |
| --- | --- |
| Azure Resource Manager Policy | https://azure.microsoft.com/en-us/documentation/articles/resource-manager-policy/ |
| PowerShell and Azure Resource Manager | https://azure.microsoft.com/en-us/documentation/articles/powershell-azure-resource-manager/ |

### Task 1: Create the service catalog policy

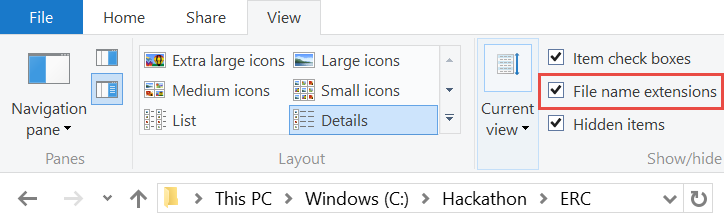
In this exercise, you will create a new Azure Resource Manager policy to restrict services to the supported list provided by Trey Research.

1. Launch the PowerShell ISE by entering **powershell\_ise** in the **Run** box.
2. Within the Console pane, execute the following command. When prompted enter your credentials.

Login-AzureRmAccount

1. Create a folder on the C: drive of your lab machine called Hackathon. Create a subfolder within it called ERC (**C:\Hackathon\ERC**).
2. Right-click in the ERC folder and create a new Text file. Rename the file to **ServiceCatalog.json**.

Note: You will need to select **File name extensions** in the **View** tab of File Explorer to specify the .json extension.



1. Right-click the file and select **Open with Visual Studio 2015**.
2. Within the file, paste the following template code. This template will deny the creation of any resource that is not part of the anyOf array.

{

"if": {

"not": {

"anyOf": [

]

}

},

"then": {

"effect": "deny"

}

}

1. Execute the following command to view the available resource providers for your subscription.

Get-AzureRmResourceProvider -ListAvailable

1. Identify the provider namespaces for each of the services that will go in the policy file.
   1. Resource Groups
   2. Virtual machines
   3. ExpressRoute
   4. VPN gateways
   5. Storage
   6. virtual networks
   7. Backup
   8. Site recovery
   9. DevTest labs
   10. Key vault
   11. Azure App Services
   12. SQL Database

Tip: You can view the resource types of a specific resource provider by running the following code:

Get-AzureRmResourceProvider -ProviderNamespace "Microsoft.*NameSpace*" | select -ExpandProperty ResourceTypes

1. Add the following code for each namespace identified to the JSON file in between the [ ] brackets of the anyOf code block. Ensure you replace the NameSpace placeholder with the real name of the namespace. Remove the comma from the last item in the list.

{

"source": "action",

"like": "Microsoft.*NameSpace*/\*"

},

For example, if you were only adding the Microsoft.Resources and Microsoft.Compute namespaces (not likely) your JSON code would look like this:



### Task 2: Restrict the creation of ExpressRoute circuits

In this exercise, you will create a new Azure Resource Manager policy to restrict the creation of an ExpressRoute circuit.

1. Within the ERC folder, create a new text/json file named **RestrictERCircuit.json**. Open the file in Visual Studio 2015.
2. Within the file, paste the following template code. This template will deny any operation that is part of the allOf array.

{

"if": {

"allOf": [

{

}

]

},

"then": {

"effect": "deny"

}

}

1. To identify the ExpressRoute operation we want to restrict, execute the following code in the PowerShell ISE console pane:

Get-AzureRmProviderOperation -OperationSearchString "Microsoft.Network/\*"

In the output, you should see the write operation for expressRouteCircuits

BUG: Doesn’t work for ExpressRoute

1. Add the following code to the allOf block to restrict writing to ExpressRoute circuits.

"source": "action",

"equals": "Microsoft.Network/expressRouteCircuits/write"

### Task 3: Restrict the creation of resources in regions

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

1. Within the ERC folder, create a new text/json file named **RestrictRegions.json**. Open the file in Visual Studio 2015.
2. Add the following code to the template. This code will deny operation that does not match in the not code block.

{

"if": {

"not": {

}

},

"then": {

"effect": "deny"

}

}

1. To restrict regions, you need to know the region names available. Within the PowerShell console, execute the following code:

Get-AzureRmLocation

1. Within the output from the PowerShell code, identify the location names for the following regions:
   1. Primary: East United States, Failover: West United States
   2. Primary: West Europe, Failover: North Europe
   3. Primary: Japan West, Failover: Japan East
2. To create the policy add the following code to the not block, replace the placeholder region names with all of the regions identified from the PowerShell output.

"field": "location",

"in": [ "region1", "region2"]

### Task 4: Apply the policies

In this task, you will create a reusable script that can be used to apply policy to a new subscription in Trey Research.

1. Using PowerShell ISE select **File > New** to create a new PowerShell script. Save the script in the ERC folder and name if **ConfigureSubscription.ps1**.
2. Add the following code to the top of the script:

param([string]$SubscriptionId)

This code defines a parameter for executing the script.

1. Add the following code the script to have the script execute in the correct subscription context when applying the policy.

Select-AzureRmSubscription -SubscriptionId $SubscriptionId

1. Add the following code to the script:

$scope = "/subscriptions/$SubscriptionId"

This code defines a variable that will hold the scope of where policies will be applied. It could be scoped on the subscription, resource group, or a resource.

1. Add the following code to the script.

$expressRoutePolicy = "$PSScriptRoot\RestrictERCircuit.json"

$supportedRegionPolicy = "$PSScriptRoot\RestrictRegions.json"

$serviceCatalogPolicy = "$PSScriptRoot\ServiceCatalog.json"

This code defines the variables to the policy file paths.

1. Add the following code to apply the service catalog policy to the subscription.

Write-Host "Applying Service Catalog Policy" -ForegroundColor Green

New-AzureRmPolicyDefinition -Name "ServiceCatalog" `

-DisplayName "Service Catalog Policy" `

-Policy $serviceCatalogPolicy `

-Description "Trey Research Service Catalog Policy"

$scPolicy = Get-AzureRmPolicyDefinition -Name "ServiceCatalog"

New-AzureRmPolicyAssignment -Name "ServiceCatalog" `

-Scope $scope `

-DisplayName "Trey Research - Service Catalog Policy" `

-PolicyDefinition $scPolicy

This code will create the policy definition for the service catalog using the New-AzureRmPolicyDefinition cmdlet. It will then assign the policy at the scope defined in the $scope variable.

1. Add the following code at the end of the file. This code will apply the other two policy files to the subscription.

Write-Host "Applying ExpressRoute Restriction Policy" -ForegroundColor Green

New-AzureRmPolicyDefinition -Name "RestrictERCircuit" `

-DisplayName "RestrictERCircuit" `

-Policy $expressRoutePolicy `

-Description "Restrict ExpressRoute Circuit"

$erPolicy = Get-AzureRmPolicyDefinition -Name "RestrictERCircuit"

New-AzureRmPolicyAssignment -Name "RestrictERCircuit" `

-Scope $scope `

-DisplayName "RestrictERCircuit" `

-PolicyDefinition $erPolicy

Write-Host "Applying Supported Regions Policy" -ForegroundColor Green

New-AzureRmPolicyDefinition -Name "SupportedRegions" `

-DisplayName "SupportedRegions" `

-Policy $supportedRegionPolicy `

-Description "Trey Research Supported Regions"

$regionsPolicy = Get-AzureRmPolicyDefinition -Name "SupportedRegions"

New-AzureRmPolicyAssignment -Name "SupportedRegions" `

-Scope $scope `

-DisplayName "Trey Research Supported Regions" `

-PolicyDefinition $regionsPolicy

1. Press **CTRL + S** to save the file.
2. In the **Console** pane, change the directory to C:\Hackathon\ERC by entering CD C:\Hackathon\ERC.
3. Execute the following command in the **Console** Pane.

Get-AzureRmSubscription

1. In the **Console** pane, create a new variable that contains your subscription ID by typing in the following code (ensure you replace the placeholder text with the subscription ID returned from Get-AzureRmSubscription).

$subscriptionId = "your subscription id"

1. In the **Console** pane, execute the following code to apply the policy to your subscription.

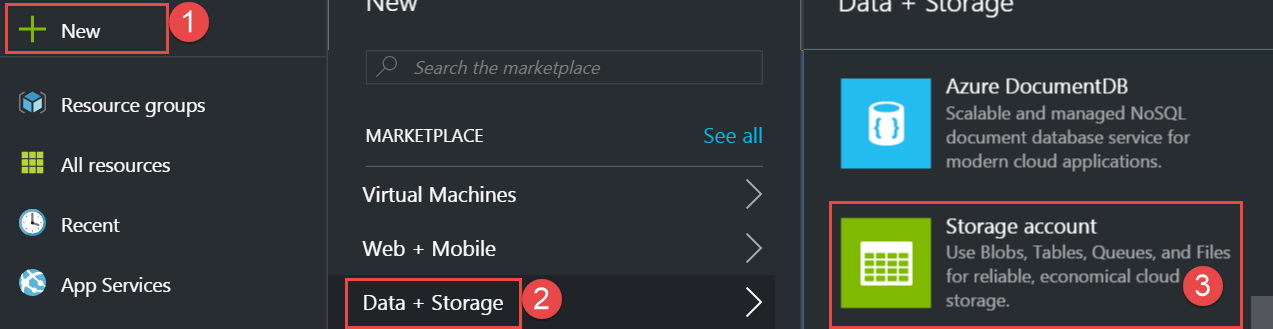
.\ConfigureSubscription.ps1 -SubscriptionId $subscriptionId

### Task 5: Test the policies

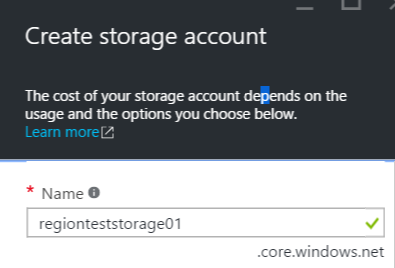
In this task, you will use the Azure management portal to validate the policies work and understand how to identify policy events.

#### Subtask 1: Test the regional policy

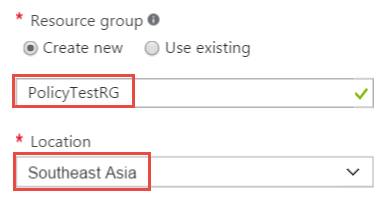
1. Navigate to the Azure management portal in a browser <http://portal.azure.com> and sign in using the same credentials you logged in with PowerShell.
2. Select **New > Data + Storage > Storage account**.



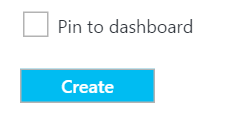
1. Specify a unique name for the storage account.



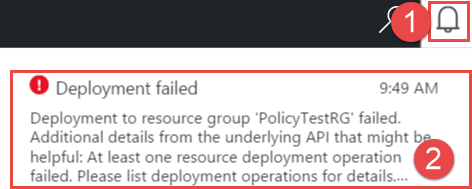
1. In the **Storage Account Creation** blade, ensure the subscription is the same as the subscription you applied the policy to, specify **PolicyTestRG** as the name of the resource group, and specify the location as a region that is not part of the supported regions of your policy.



1. Click **Create** to start provisioning the storage account.

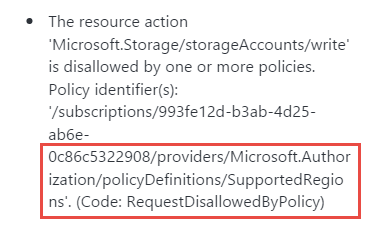


1. In the **Notification** hub, you should see an error that a policy blocked the creation of the storage account. Click the error to view the details and the actual policy that blocked the creation.

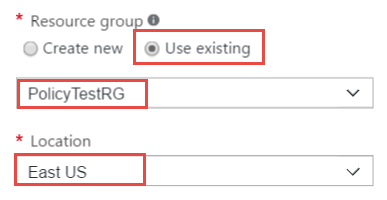


1. Click **Failed. Click here for details** to view the failure details.



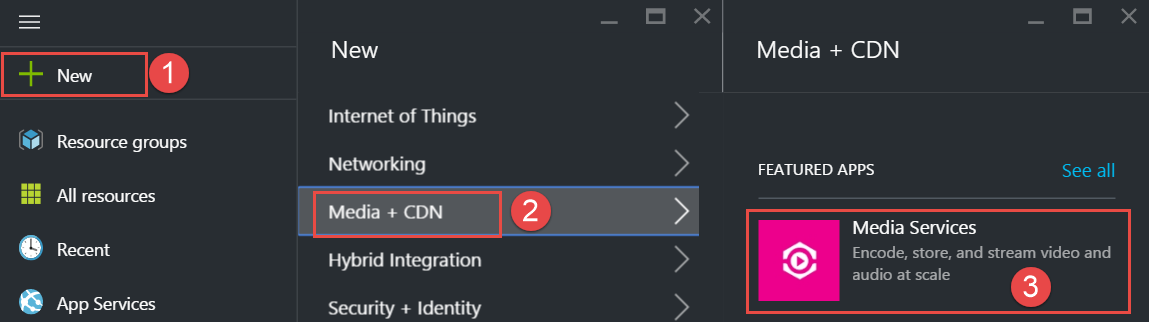


1. Repeat the previous steps but change the location to a supported region to ensure that you can actually create the resource. Specify the existing PolicyTestRG resource group.

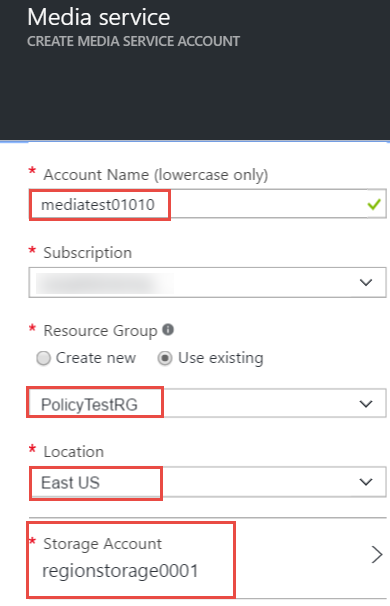


#### Subtask 2: Test the service catalog policy

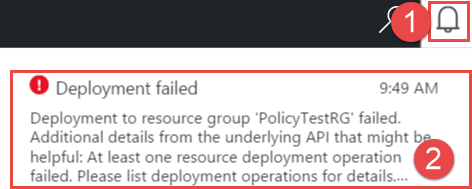
1. Navigate to the Azure management portal in a browser <http://portal.azure.com> and sign in using the same credentials you used with PowerShell.
2. Click **New > Media + CDN > Media Services**.



1. Specify a unique name for the media service account. Specify the existing **PolicyTestRG** resource group. Select a supported region and the previously created storage account (assuming they are in the same region).

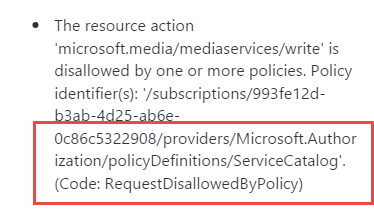


1. In the Notification hub, you should see an error that a policy blocked the creation of the media service. Click the error to view the details and the actual policy that blocked the creation.



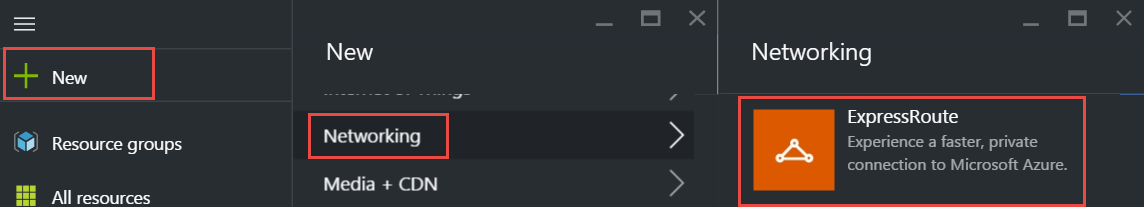
1. Click **Failed. Click here for details** to view the failure details.





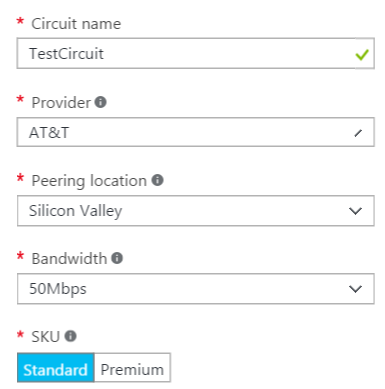
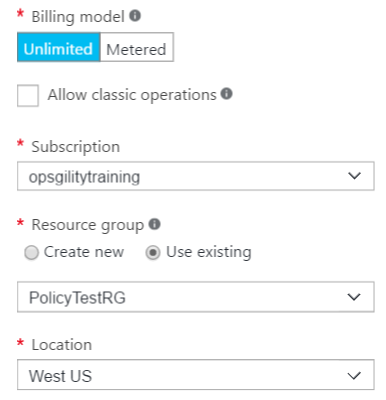
#### Subtask 3: Test the ExpressRoute circuit policy

1. Click **New** **>** **Networking** **>** **ExpressRoute**.

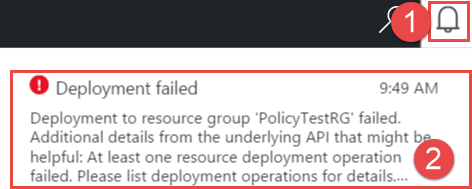


1. Specify the following configuration for the circuit and click **Create**.

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

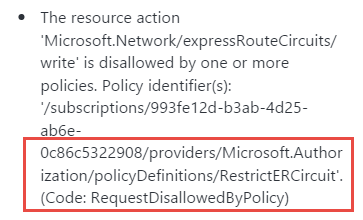
 

1. In the **Notification** hub, you should see an error that a policy blocked the creation of the ExpressRoute circuit. Click the error to view the details and the actual policy that blocked the creation.



1. Click **Failed. Click here for details** to view the failure details.





## Exercise 3: Configure delegated permissions

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://azure.microsoft.com/en-us/documentation/articles/active-directory-create-users/ |
| How Subscriptions are associated with Azure AD | https://azure.microsoft.com/en-us/documentation/articles/active-directory-how-subscriptions-associated-directory/ |
| Managing Azure AD Security Groups | https://azure.microsoft.com/en-us/documentation/articles/active-directory-accessmanagement-manage-groups/ |
| Role Based Access Control | https://azure.microsoft.com/en-us/documentation/articles/role-based-access-control-configure/ |
| Manage RBAC with PowerShell | https://azure.microsoft.com/en-us/documentation/articles/role-based-access-control-manage-access-powershell/ |

### Task 1: 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 the classic Azure management portal using a browser <http://manage.windowsazure.com>.
2. Expand Active Directory on the left navigation and click the directory associated with your subscription.

If you have multiple directories, click **Settings** to view which directory is associated with your subscription.

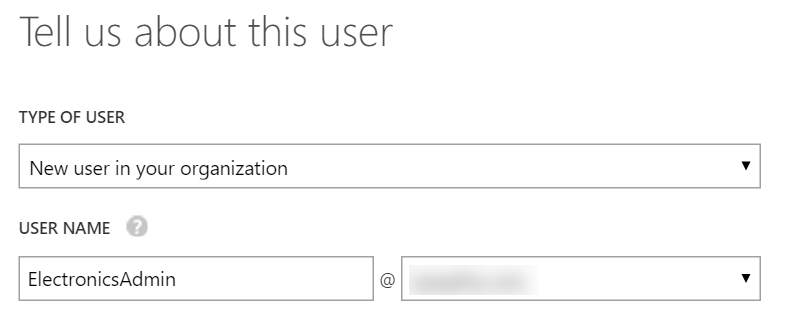
1. Click **USERS**.



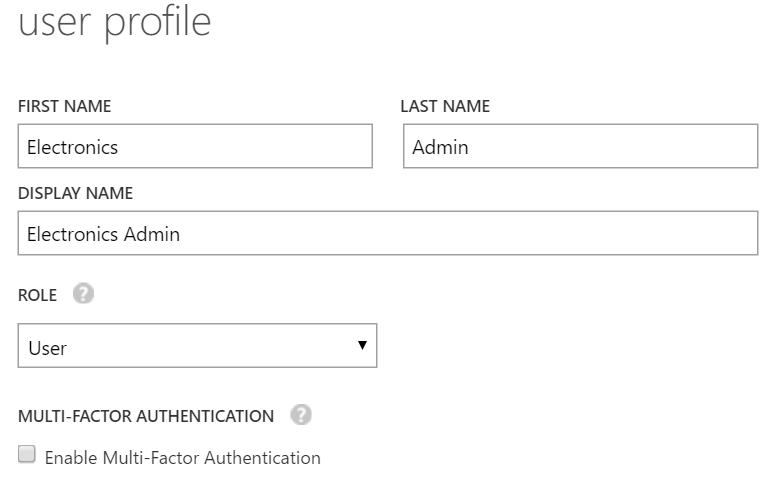
1. Click **ADD USER**.



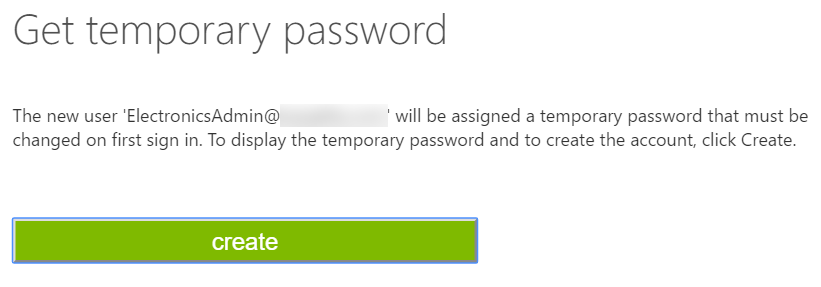
1. Enter the USER NAME as ElectronicsAdmin and click **>**.



1. Complete the users profile and click **>**.



1. Click **create** and note the email and temporary password generated in notepad.



1. Repeat the previous steps and create another user. Name this user **ElectronicsUser**. Note the email and temporary password in notepad.

### Task 2: 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.

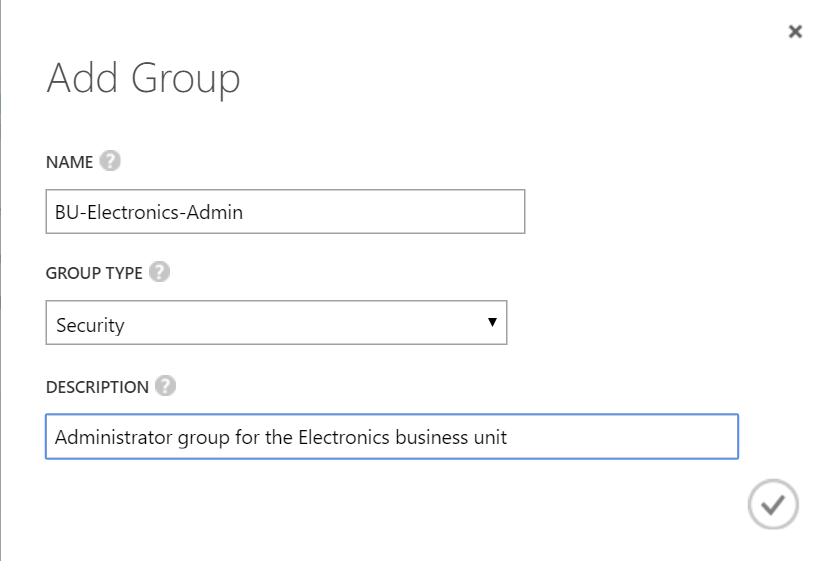
1. Click **GROUPS**.



1. Click **ADD GROUP**.



1. Create a new security group that will be used for the Electronics business group. Name the group **BU-Electronics-Admin**.



1. Repeat the steps and name the second group **BU-Electronics-Users**.
2. Click the **BU-Electronics-Admin group**.



1. Click **ADD MEMBERS**.



1. Add the **ElectronicsAdmin** user to the group.

C:\Users\Michael\AppData\Local\Temp\SNAGHTML39baecc9.PNG

1. Repeat the steps and add the **ElectronicsUser** to the **BU-Electronics-User group**.

### 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 the **ConfigureSubscription.ps1** script using the PowerShell ISE.
2. Replace the first line of the script with the following.

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

1. Add the following code to the very end of the 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.

$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.

1. Execute the script again, this time specify the new parameter **–GroupObjectId**.

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

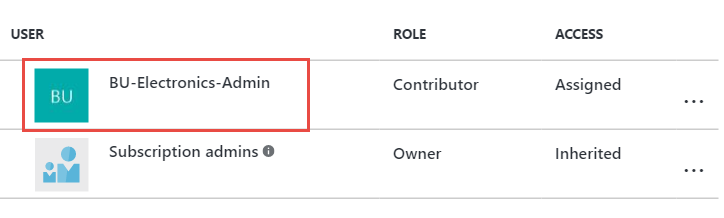
1. Close all instances of your browser (or launch a different type of browser) in in-private or incognito mode.
2. Navigate to the Azure management portal in a browser <http://portal.azure.com> and sign in using the **ElectronicsAdmin** credentials created earlier. When prompted to change your password specify a strong password you will remember.
3. You will need to configure a method of resetting your account. You can choose either a phone call or email.
4. Click **Subscriptions**.



1. Click the name of the subscription you have been working on.
2. Click the **Access control** icon



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



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 can create a new resource group as the owner of the subscription and delegate owner permissions to the BU-Electronics-Admin for a specific project by applying permissions at the resource group level.

1. Create a new JSON file in the C:\Hackathon\ERC folder named **AppendIOCode.json**.
2. Open the new file in **Visual Studio 2015** and add the following JSON code to the file and then save.

{

"if": {

"not": {

"anyOf": [

{

"source": "action",

"like": "Microsoft.Resources/[ProjectResourceGroup]/\*"

}

]

}

},

"then": {

"effect": "append",

"details": [

{

"field": "tags",

"value": { "ioCode": "[IOCODEVALUE]" }

}

]

}

}

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 top of the script. This code will require three parameters, the name of the resource group, the region/location, and the group to add as an owner.

param(

[string]$SubscriptionId,

[string]$ResourceGroupName,

[String]$Location,

[String]$IOCode,

[string]$AdGroupName

)

1. Add the following code to the script and save the file.

Select-AzureRmSubscription -SubscriptionId $SubscriptionId

New-AzureRmResourceGroup -Name $ResourceGroupName -Location $Location

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

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

New-AzureRmRoleAssignment -Scope $scope `

-RoleDefinitionName "Owner" `

-ObjectId $groupObjectId

Write-Host "Creating Project Specific Policy File" -ForegroundColor Green

$ioCodePolicy = "$PSScriptRoot\appendIOCode.json"

$policyContent = Get-Item -Path $ioCodePolicy | Get-Content

$policyContent = $policyContent.Replace("[ProjectResourceGroup]", $ResourceGroupName)

$policyContent = $policyContent.Replace("[IOCODEVALUE]", $IOCode)

$ioCodePolicySpecific = "$PSScriptRoot\appendIOCode$ioCode.json"

$policyContent | Out-File -FilePath $ioCodePolicySpecific

Write-Host "Applying IO Code Policy" -ForegroundColor Green

New-AzureRmPolicyDefinition -Name "AppendIOCode" `

-DisplayName "Append IO Code" `

-Policy $ioCodePolicySpecific `

-Description "Append IO Code"

$scPolicy = Get-AzureRmPolicyDefinition -Name "AppendIOCode"

New-AzureRmPolicyAssignment -Name "AppendIOCode" `

-Scope $scope `

-DisplayName "Append IO Code" `

-PolicyDefinition $scPolicy

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 creates a copy of the original appendIOCode.json file and replaces the placeholder values in it for the resource group scope and the IO code specified in the script.

1. 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"

1. In the **Console** pane, create a new variable called **$resourceGroupName** and specify the value as **DelegatedProjectDemo**.

$resourceGroupName = "DelegatedProjectDemo"

1. 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"

1. 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.



1. Switch back to the Azure Management portal using the ElectronicsAdmin credentials.
2. Click **Resource Groups.**
3. Click the **DelegatedProjectdemo** resource group.



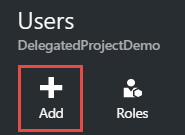
1. Click the **Access** icon.



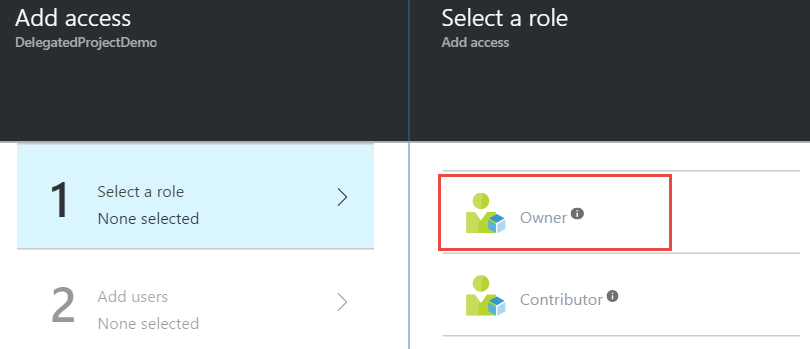
1. Note that the BU-Electronics-Admin role is set as owner for the resource group.



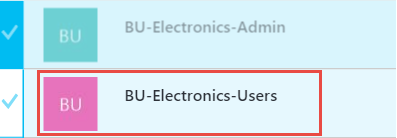
1. Click **Add**.



1. Select **Owner**.



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



## Exercise 4: Create the environment for the e-commerce team

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.

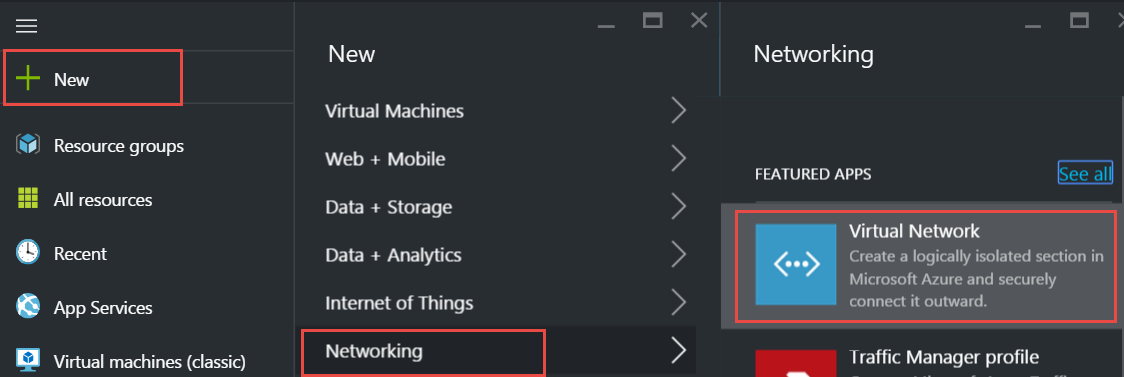
### Help references

|  |  |
| --- | --- |
| Configuring Point-to-Site Secure VPN | https://azure.microsoft.com/en-us/documentation/articles/vpn-gateway-howto-point-to-site-rm-ps/ |
| Network Security Groups | https://azure.microsoft.com/en-us/documentation/articles/virtual-networks-nsg/ |
| Azure DevTest Labs | https://azure.microsoft.com/en-us/documentation/services/devtest-lab/ |

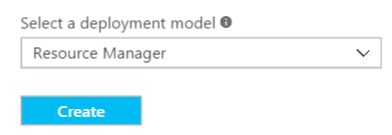
### Task 1: Create a new virtual network

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

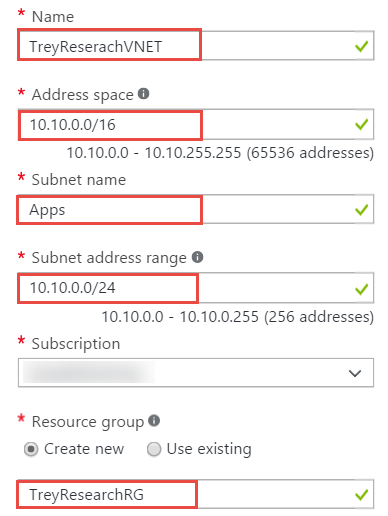
1. Sign in to the Azure Management portal using the subscription owner user account.
2. Click **New** **> Networking >** **Virtual Network**.



1. Leave the deployment model set to **Resource Manager** and click **Create**.

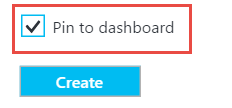


1. Specify the following configuration for the virtual network:
   1. Name: TreyResearchVNET
   2. Address Space: 10.10.0.0/16
   3. Subnet Name: Apps
   4. Subnet address Range: 10.10.0.0/24
   5. Resource Group: TreyResearchRG (Create New)
   6. Location: Choose one of the supported regions.

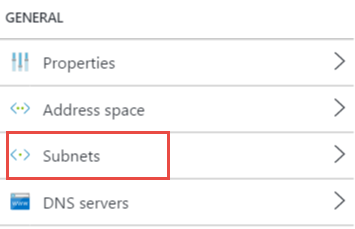




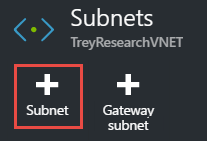
1. Select Pin to dashboard and click **Create**.



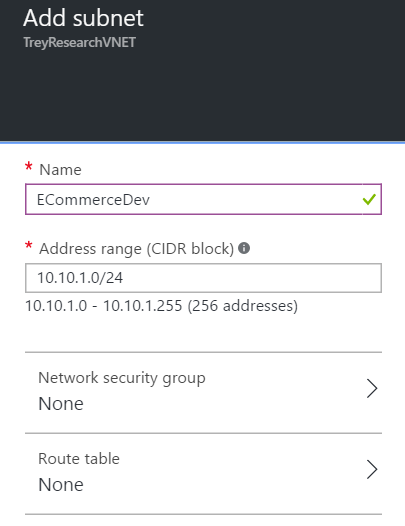
1. After the virtual network is open, click **Subnets.**



1. Click **+ Subnet**.



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



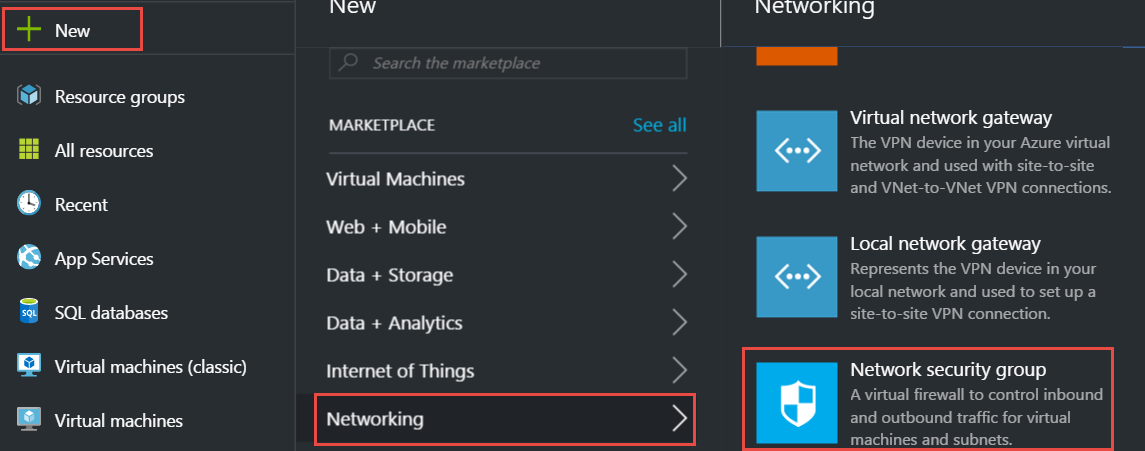
1. Click **+ Gateway subnet** to add a gateway subnet to the virtual network. Click **OK** on the new blade that opens.



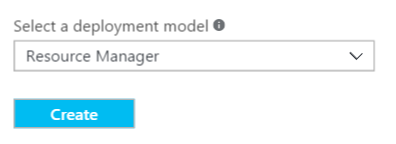
### Task 2: Configure limited network access

In this task, you will configure traffic from the ECommerceDev subnet to be restricted from the Apps subnet.

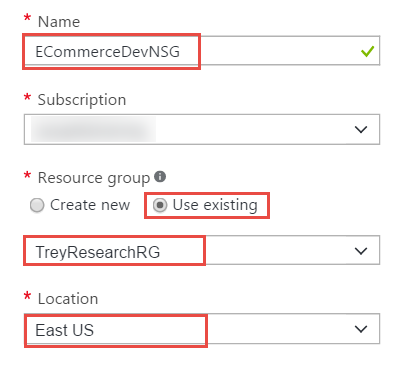
1. Click **New** **>** **Networking** **>** **Network Security Group.**



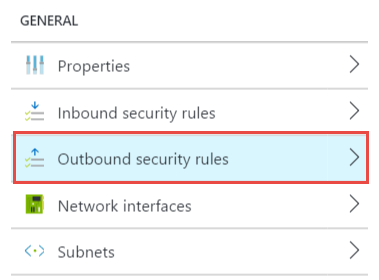
1. Leave the deployment model set to **Resource Manager** and click **Create**.



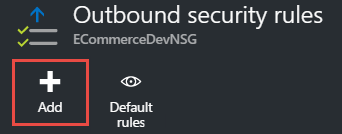
1. Name the NSG **ECommerceDevNSG**, specify the existing TreyResearchRG resource group, and ensure you choose the same region you created the virtual network in.



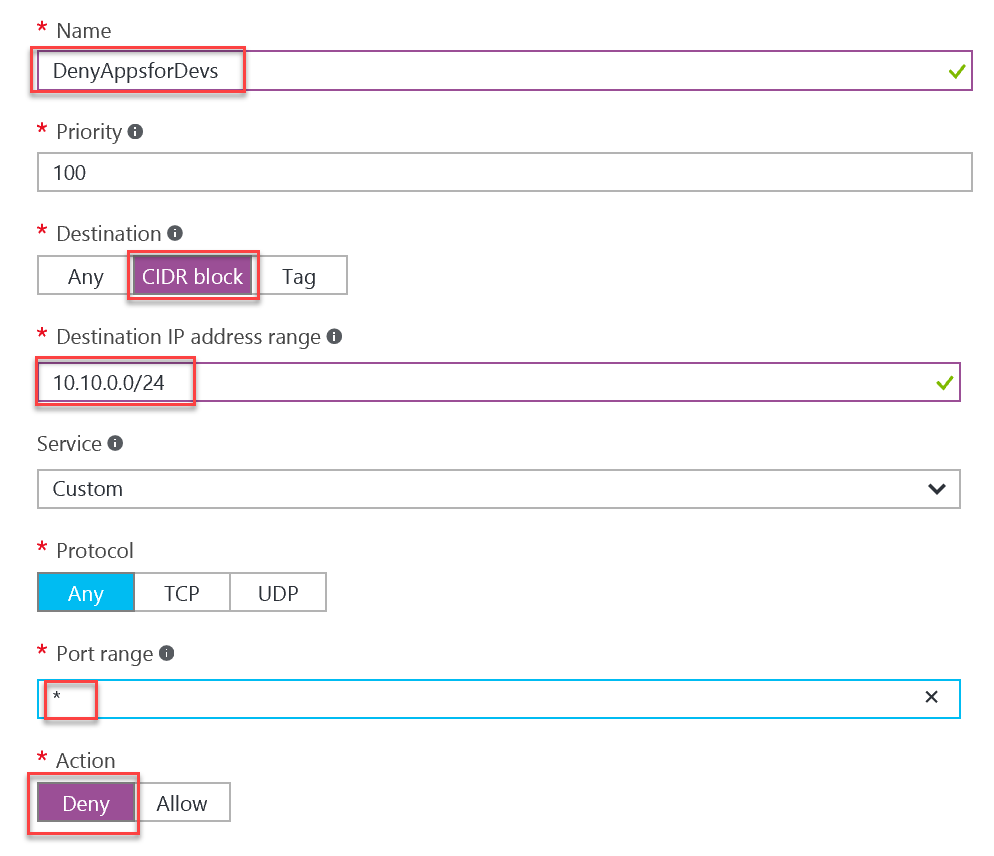
1. Select **Pin to Dashboard** and click **Create**.
2. After the Network Security Group’s configuration blade opens, click **Outbound security rules**.



1. Click **Add.**



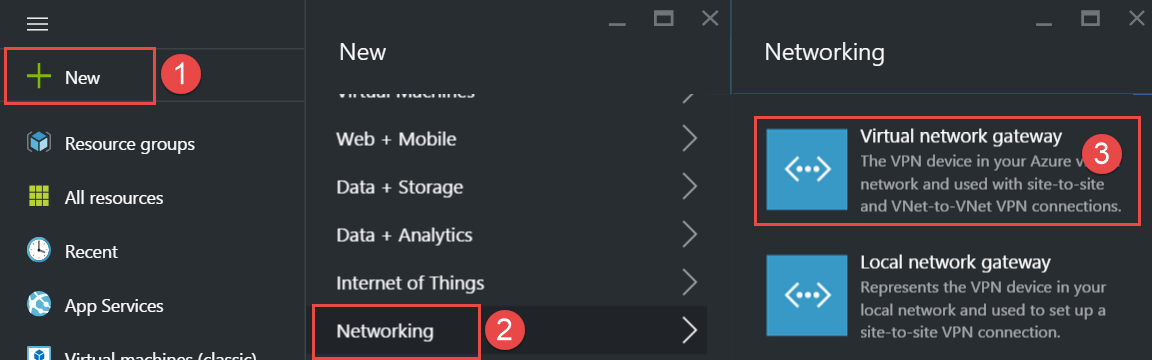
1. Specify the following configuration to restrict access from the developer subnet.
   * Name: DenyAppsforDevs
   * Priority: 100
   * Destination: CIDR block
   * Destination IP Range: 10.10.0.0/24
   * Port Range: \*
   * Action: Deny



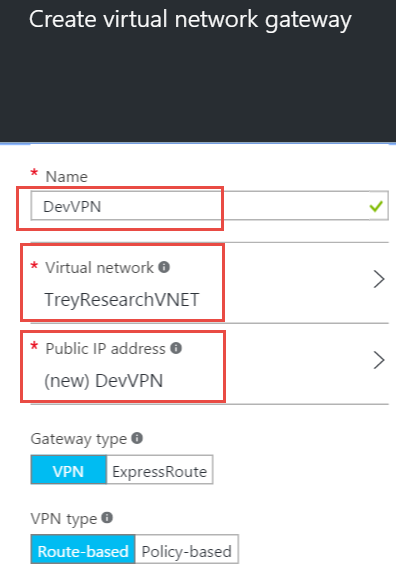
### Task 3: Configure secure VPN for connectivity

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

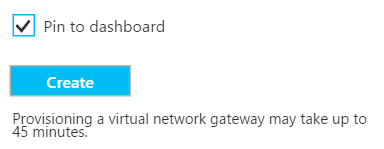
1. Click **New** > **Networking** **>** **virtual network gateway.**



1. Name the VPN Gateway **DevVPN**, select the existing **TreyResearchVNET** virtual network, and specify a new Public IP address.



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

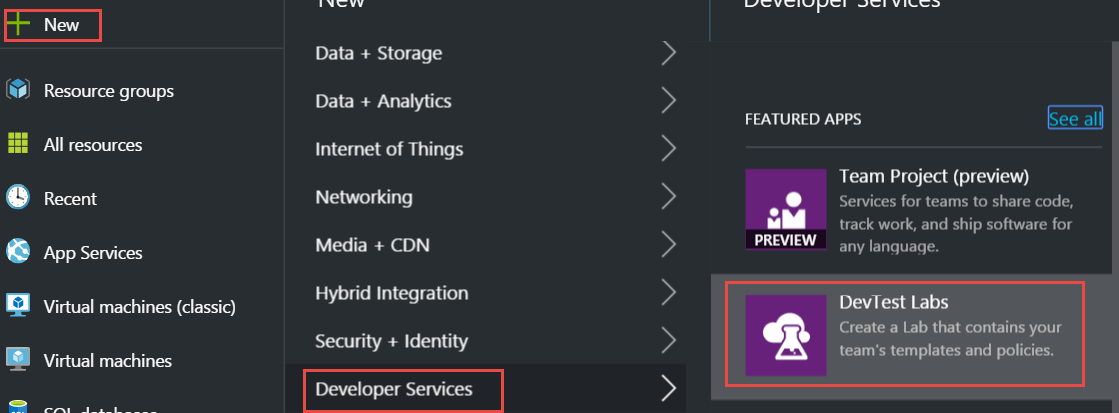


This step will take up to 45 minutes to complete. Continue to the next task. Gateway configuration will be continued in a later task.

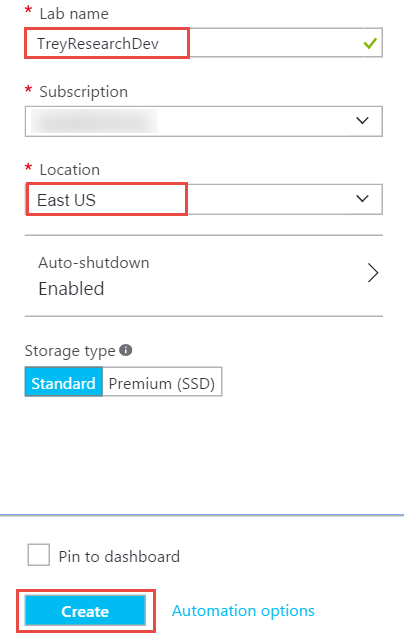
### Task 4: Create an Azure DevTest lab environment

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

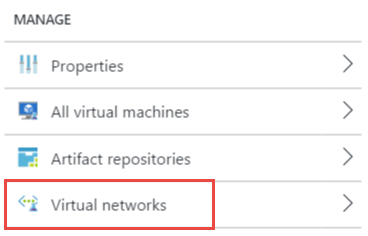
1. Click **New** **>** **Developer Services >** **DevTest Labs**.



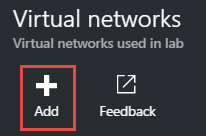
1. Name the lab **TreyResearchDev** and specify the same region that you deployed the virtual network to.



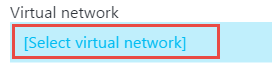
1. Open the DevTest lab environment after it has completed provisioning.
2. Open **Settings > Virtual Networks**.



1. Click **Add**.



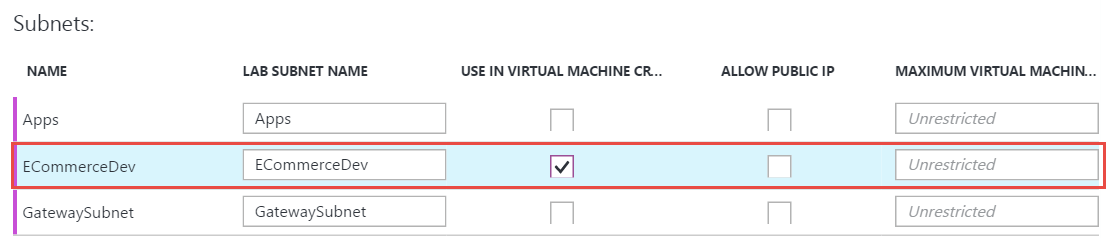
1. Click the **Select virtual network.**



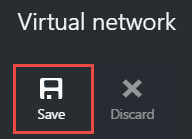
1. Select **TreyResearchVNET**.



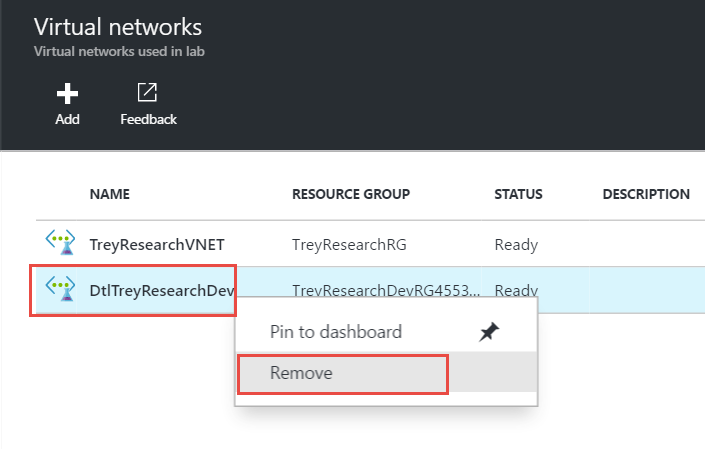
1. Select **USE IN VIRTUAL MACHINE CREATION** next to the **ECommerceDev** subnet.



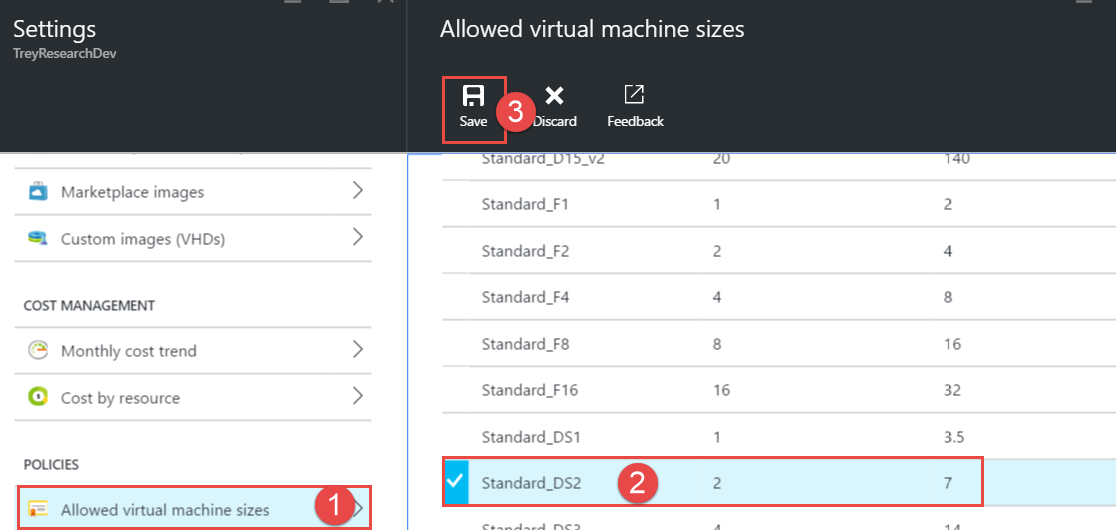
1. Click **Save.**



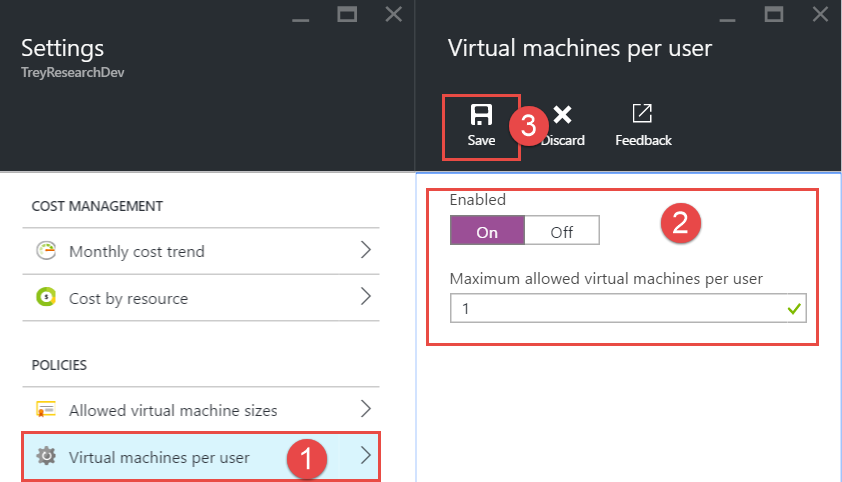
1. Right-click the default virtual network created for the DevTest lab environment and click **Remove**.



1. Configure a virtual machine policy for this DevTest lab by clicking **Allowed virtual machine sizes**, select **Standard\_DS2**, and click **Save**.



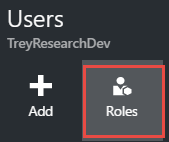
1. Enable the virtual machines per user policy. Set a maximum number of virtual machines per user to one and click **Save**.



1. Allow access to the DevTest labs users by clicking the **Access** icon.



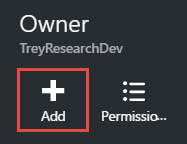
1. Click **Roles**.



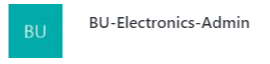
1. Select **Owner**.



1. Click **Add**.



1. Add the **BU-Electronics-Admin** group. Now the user will be able to invite users to the DevTest lab.



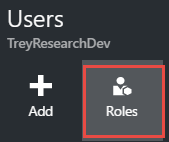
### Task 5: Test access to the DevTest labs environment

In this task, you will use the ElectronicsAdmin user account to grant access to the developer environment and then 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 **Browse** **>** **DevTest Labs** **>** **TreyResearchDev**.
3. Allow access to the DevTest Labs users by clicking the **Access** icon.



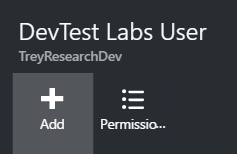
1. Click **Roles**.



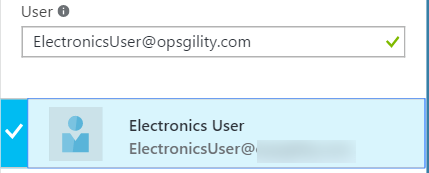
1. Click **DevTest Labs User**.



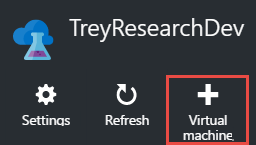
1. Click **Add**.



1. Specify the **ElectronicsUser** account and click **Select**.



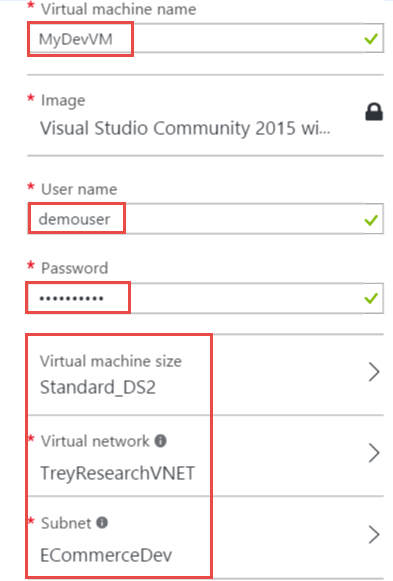
1. Close your browser and sign in with the **ElectronicsUser** account. You will have to change your password and setup a recovery mechanism with this account.
2. Open the DevTest labs environment by clicking **Browse** **>** **DevTest Labs** **>** **TreyResearchDev**.
3. Click **+ Virtual machine** to provision a virtual machine for the developer.



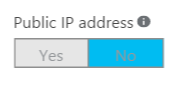
1. Select the Visual Studio Community 2015 with the latest Azure SDK.



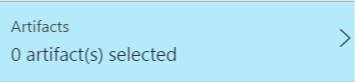
1. Specify a name for the virtual machine as well as a user name and password. Note that the VM size, virtual network, and subnet are not changeable for the user.



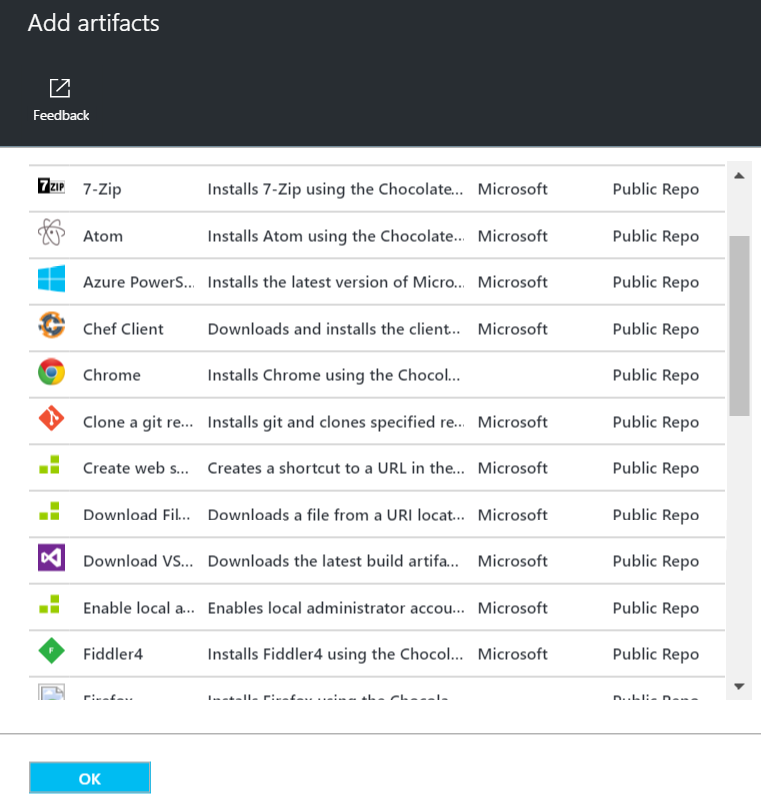
1. Note that a Public IP address cannot be set either, it means the user will have to access the virtual machine by its private IP address.



1. Click **Artifacts**.



1. Add Azure PowerShell and Fiddler 4 to the artifacts of the VM by clicking their names and clicking **OK**. Click **OK** at the bottom of the **Add artifacts** blade when complete.



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

### Task 6: Finish configuring secure connectivity

In this task, you will configure certificates for the VPN gateway and for the end users and 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: <http://opsgilityweb.blob.core.windows.net/makecert/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

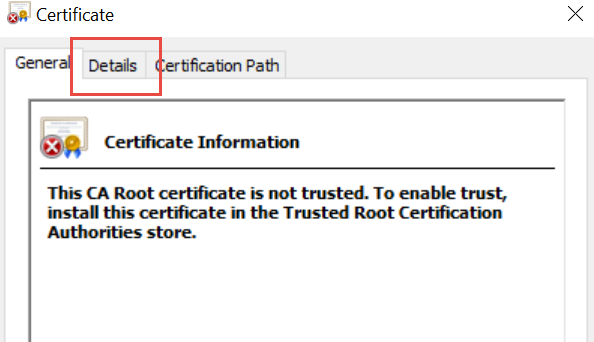
1. Execute the following command to generate a root certificate for configuring a point-to-site VPN gateway.

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

1. 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

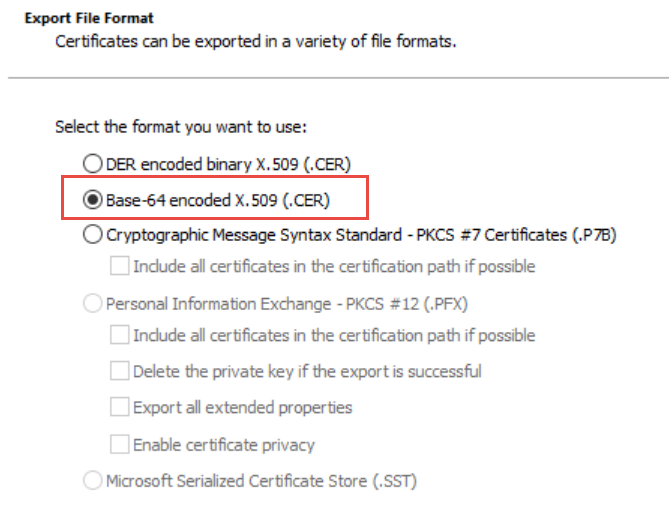
1. Using **File Explorer**, navigate to the C:\Hackathon\ERC folder and double-click the **P2SRoot.cer** file.
2. Click the **Details** tab of the certificate.



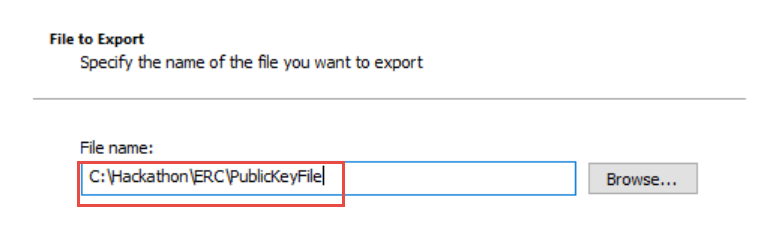
1. Click **Copy to File**



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



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



#### Subtask 2: Configure the VPN gateway

1. In the **PowerShell ISE Console** pane, execute the following commands to setup variables for the gateway creation:

$vpnClientPool = "172.16.201.0/24"

$vpnGWName = "DevVPN"

$rgName = "TreyResearchRG"

$rootCertName = "P2SRoot.cer"

1. Execute the following command to configure the VPN Gateway for point-to-site:

$vpnGW = Get-AzureRmVirtualNetworkGateway -ResourceGroupName $rgName `

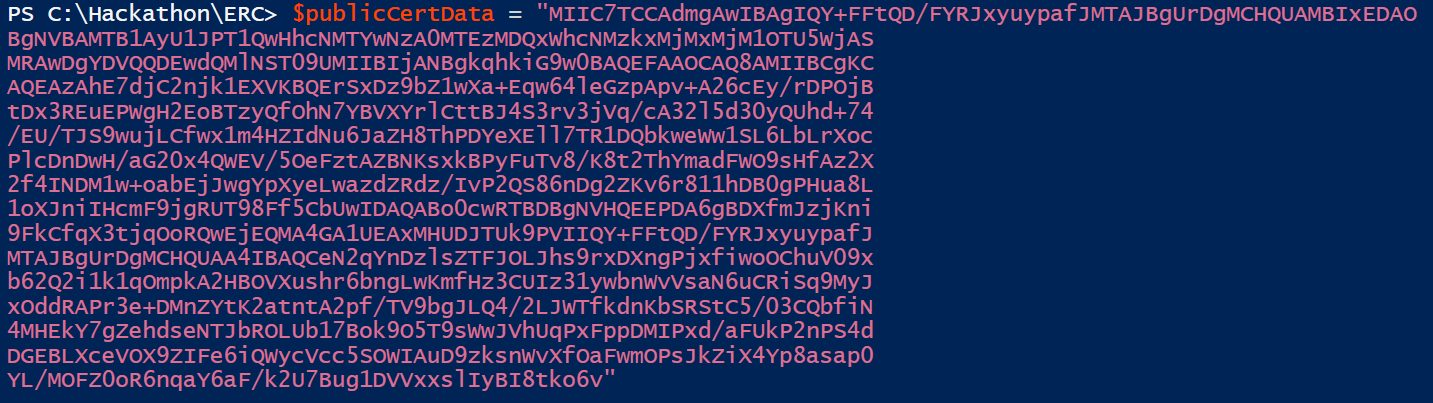
-Name $vpnGWName

Set-AzureRmVirtualNetworkGatewayVpnClientConfig `

-VirtualNetworkGateway $vpnGW `

-VpnClientAddressPool $vpnClientPool

1. Open the **PublicKeyFile.cer** created earlier in notepad.
2. Copy ALL of the values between “-----BEGIN CERTIFICATE-----“ and “-----END CERTIFICATE-----“ to the clipboard.
3. Create a new variable by entering **$publicCertData = "** **and paste in the value from the clipboard and end with a "**.



1. Execute the following code to upload the root certificate to the gateway.

Add-AzureRmVpnClientRootCertificate `

-VpnClientRootCertificateName $rootCertName `

-VirtualNetworkGatewayName $vpnGWName `

-ResourceGroupName $rgName `

-PublicCertData $publicCertData

#### Subtask 3: Configure and test the client

1. In the PowerShell ISE Console pane, execute the following command to generate the VPN package for the virtual network. This command will generate a link to the created package.

$downloadUrl = Get-AzureRmVpnClientPackage `

-ResourceGroupName $rgName `

-VirtualNetworkGatewayName $vpnGWName `

-ProcessorArchitecture Amd64

1. Execute the following code to download the VPN package to your local machine.

Invoke-WebRequest -Uri $downloadUrl.Replace("""","") -OutFile "C:\Hackathon\ERC\VPNclient.exe"

1. In the **PowerShell ISE Console** pane, execute the following command to install the VPN client (the current directory most be set to C:\Hackathon\ERC). Accept any prompts to install the client.

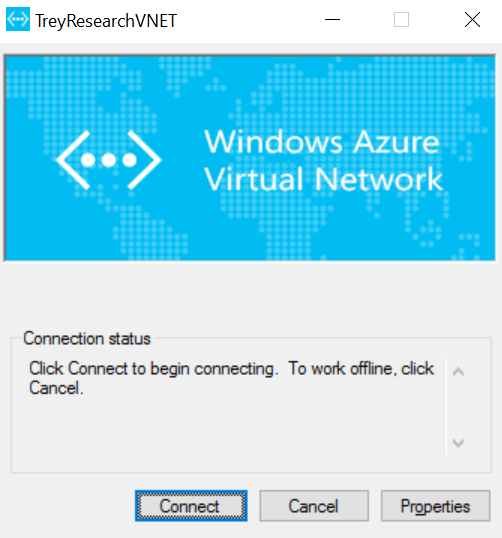
.\VPNclient.exe

1. 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.

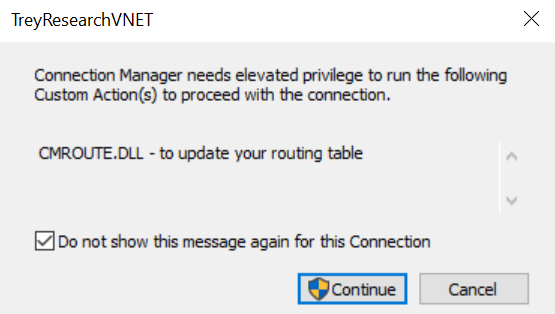


1. Click **Connect** to initiate connection.

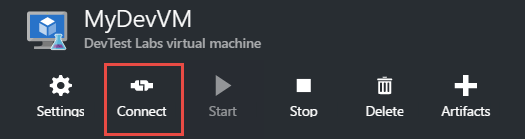




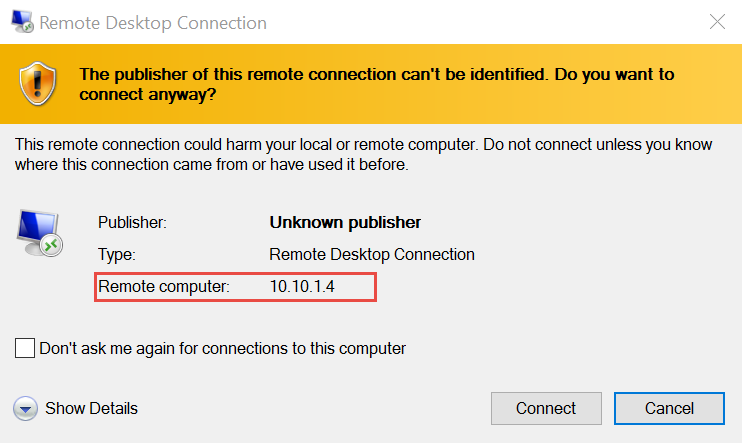
1. Select **Do not show this message again for this Connection** and click **Continue**.



1. After you are successfully connected, switch back to the Azure Management Portal using the **ElectronicsUser** account.
2. Browse to **DevTest Labs**, open **TreyResearchLab**, and click **MyDevVM**.
3. Click **Connect** to initiate a connection with the virtual machine.



1. Note the remote computer is connecting over a private IP address.



## Exercise 5: Remove the policies from your subscription

In this exercise, you will remove the policies on your subscription.

1. Run the following code in the **PowerShell ISE Console** pane:

# Retrieve with Get-AzureRmSubscription

$SubscriptionId = ""

$scope = "/subscriptions/$SubscriptionId"

Remove-AzureRmPolicyAssignment -Name "ServiceCatalog" -Scope $scope

Remove-AzureRmPolicyDefinition -Name "ServiceCatalog"

Remove-AzureRmPolicyAssignment -Name "RestrictERCircuit" -Scope $scope

Remove-AzureRmPolicyDefinition -Name "RestrictERCircuit"

Remove-AzureRmPolicyAssignment -Name "SupportedRegions" -Scope $scope

Remove-AzureRmPolicyDefinition -Name "SupportedRegions"