Enterprise-ready cloud

Hands-on lab step-by-step

October 2017

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2017 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <https://www.microsoft.com/en-us/legal/intellectualproperty/Trademarks/Usage/General.aspx> are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners

Contents

[Enterprise-ready cloud hands-on lab step-by-step 1](#_Toc505864692)

[Abstract and learning objectives 1](#_Toc505864693)

[Overview 1](#_Toc505864694)

[Requirements 1](#_Toc505864695)

[Before the hands-on lab 2](#_Toc505864696)

[Task 1: Validate global admin access to Azure AD tenant 2](#_Toc505864697)

[Exercise 1: Create the policy for Enterprise IT 4](#_Toc505864698)

[Help references 4](#_Toc505864699)

[Task 1: Create the service catalog policy 4](#_Toc505864700)

[Task 2: Restrict the creation of ExpressRoute circuits 6](#_Toc505864701)

[Task 3: Restrict the creation of resources in regions 7](#_Toc505864702)

[Task 4: Apply the policies 7](#_Toc505864703)

[Task 5: Test the policies 9](#_Toc505864704)

[Exercise 2: Configure delegated permissions 15](#_Toc505864705)

[Help references 15](#_Toc505864706)

[Task 1: Create groups in Azure AD for delegation 15](#_Toc505864707)

[Task 2: Create user accounts in Azure AD for delegation 16](#_Toc505864708)

[Task 3: Enable a business unit administrator for the subscription 18](#_Toc505864709)

[Task 4: Enable project based delegation and chargeback 19](#_Toc505864710)

[Exercise 3: Create the environment for the e-commerce team 24](#_Toc505864711)

[Help references 24](#_Toc505864712)

[Task 1: Create a new virtual network 24](#_Toc505864713)

[Task 2: Configure limited network access 27](#_Toc505864714)

[Task 3: Configure secure VPN for connectivity 29](#_Toc505864715)

[Task 4: Create an Azure DevTest lab environment 30](#_Toc505864716)

[Task 5: Test access to the DevTest labs environment 34](#_Toc505864717)

[Task 6: Finish configuring secure connectivity 37](#_Toc505864718)

[After the hands-on lab 43](#_Toc505864719)

[Task 1: Remove Policies 43](#_Toc505864720)

## 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
  + Azure SDK 2.9 or later for Visual Studio <http://go.microsoft.com/fwlink/?linkid=518003&clcid=0x409>
* Full global admin access to the Azure AD tenant associated with your Azure subscription.

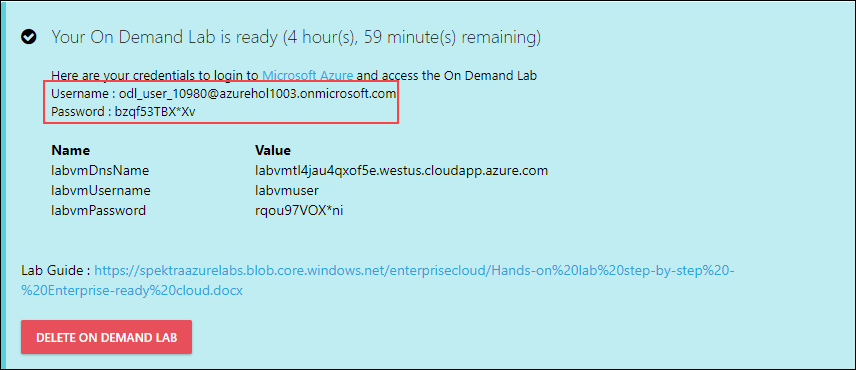
## Before the hands-on lab

Duration: 15 minutes

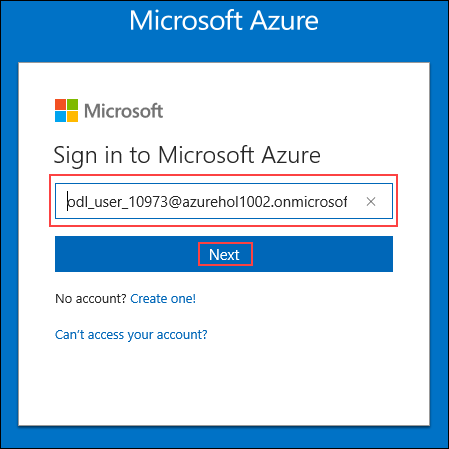
To complete this lab, you must have full global admin access to the Azure AD tenant associated with your Azure subscription.

#### Task 1: Login to the Azure Portal

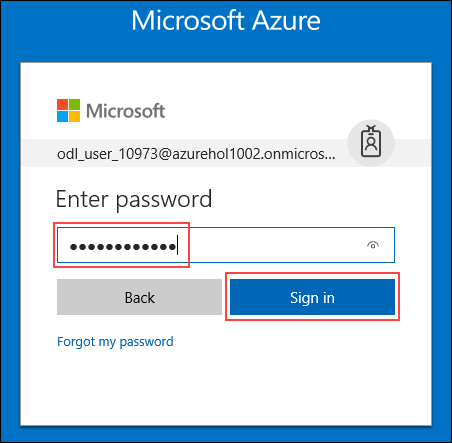
1. Take a note of the **Username** and **Password**. These credentials will be used to login to the Azure Portal.



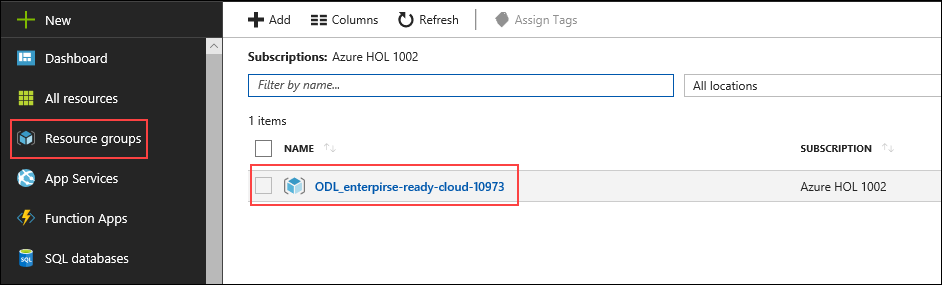
1. Navigate to <https://portal.azure.com>.
2. Enter the **Username** and click on **Next**.



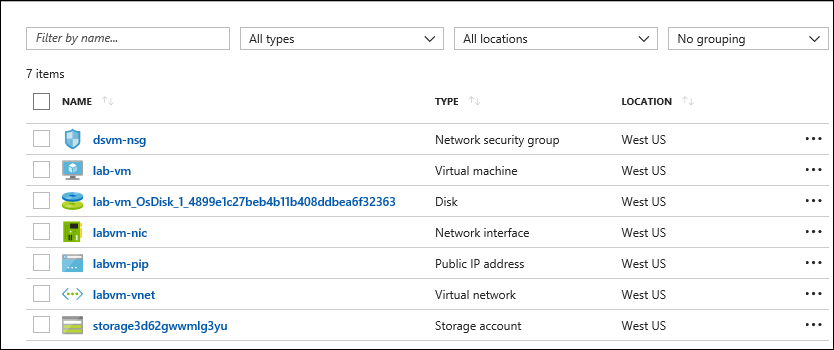
1. Enter the **Password** and click on **Sign in**.



1. **Click** on the **Resource groups** and click on the Resource group that is displayed.

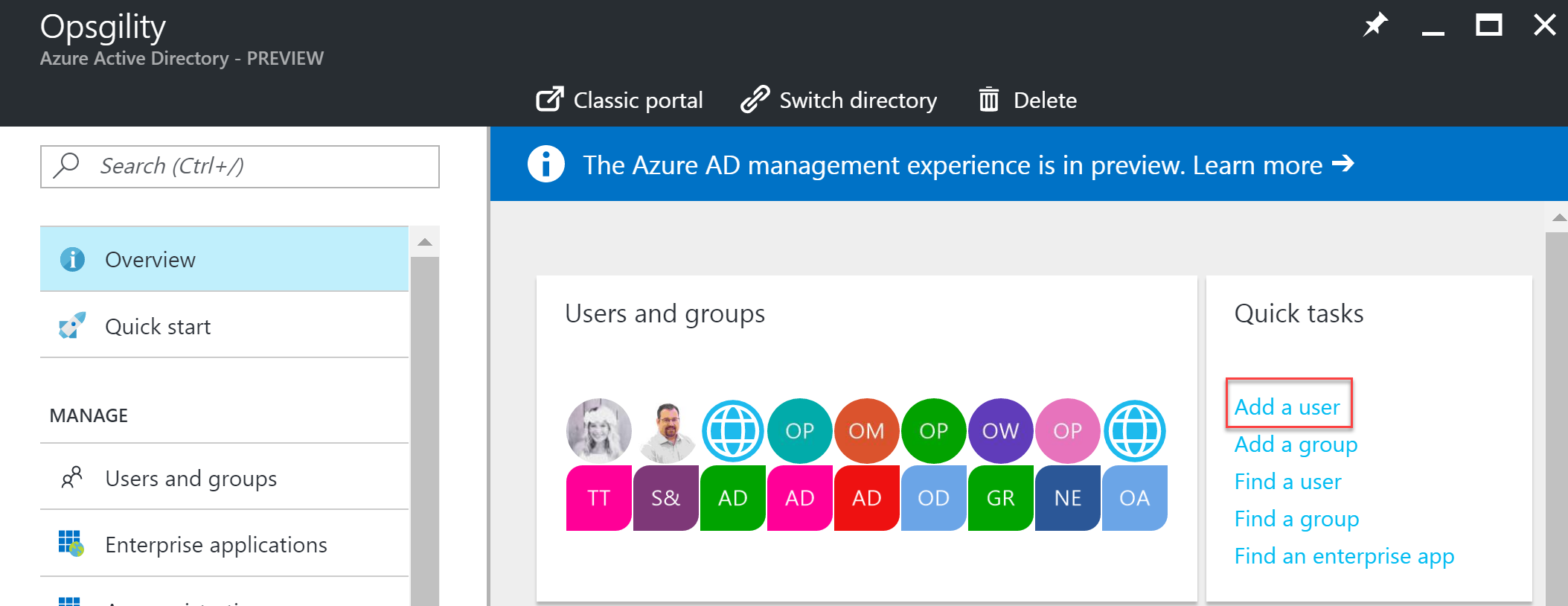


1. Make sure that all the necessary resources are deployed in the Resource group.



#### Task 2: Validate global admin access to Azure AD tenant

1. Login to <http://portal.azure.com>, click on **More Services**, and type in **Azure Active Directory**.
2. 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.



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

## Exercise 1: Create the policy for Enterprise IT

Duration: 60 minutes

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.

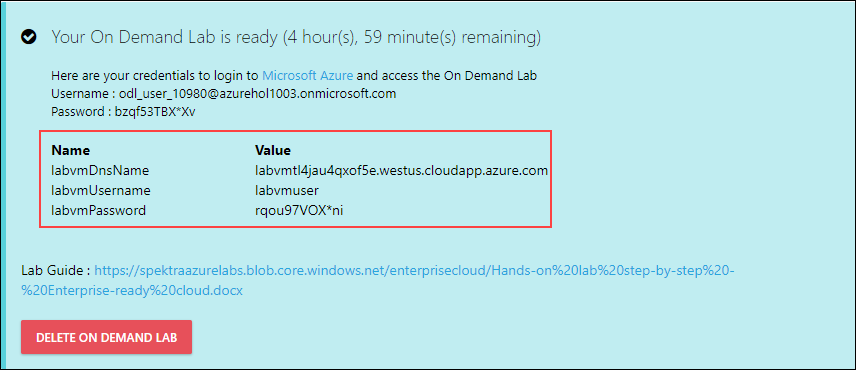
#### Help references

|  |  |
| --- | --- |
| 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/ |

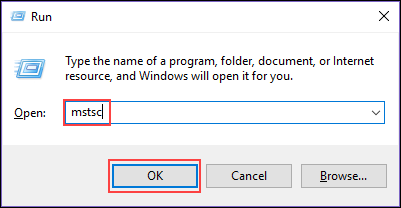
#### Task 1: Create the service catalog policy

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

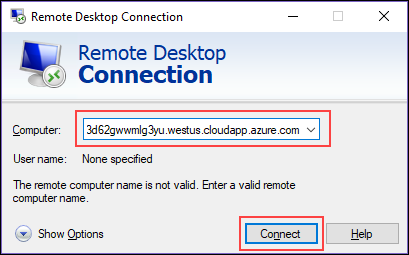
1. Take a note of the **labvmDnsName, labvmUsername** and the **labvmPassword**.



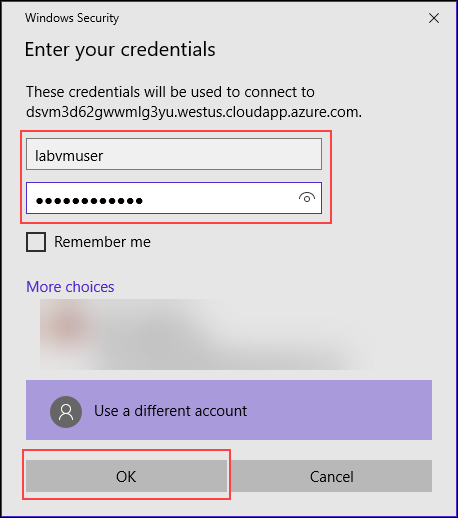
1. Press **Win+R** to get into the **Run** window.
2. In the **Run** window, type ‘**mstsc’.**



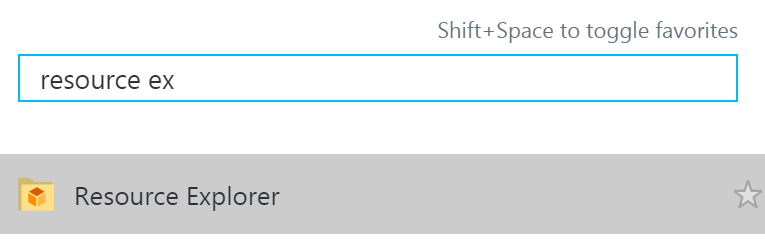
1. Enter the Dnsname and click on **Connect.**



1. Enter the **Username** and **Password** and click on **OK.**

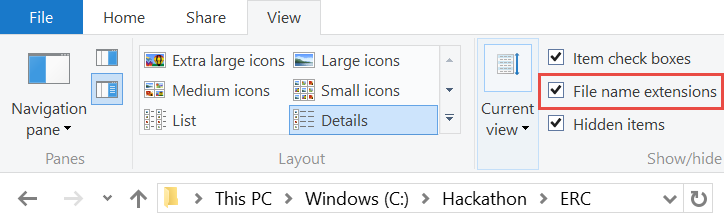


1. Open the Internet Explorer and navigate to <https://portal.azure.com> and use the credentials provided to login to the azure portal.
2. Launch the Azure Management portal, and navigate to Resource Explorer under more services.



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**.
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. Use Resource Explorer to identify the provider namespaces for the following Azure services and features:

|  |
| --- |
| Resource Name |
| Resource Group |
| Virtual Machines |
| ExpressRoute Circuits |
| VPN Gateways |
| Storage |
| Virtual Networks |
| Backup Vault |
| Site Recovery Vault |
| DevTest Labs |
| Key Vault |
| Web Apps |
| SQL Database |

1. This section of the policy will create a whitelist of services you will be able to create. 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.
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. Add the following code to the allOf block to restrict writing to ExpressRoute circuits. This allows a user to link a virtual network to an existing circuit but not create a new one.

"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.
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 which region names are available. Launch a PowerShell console, and execute the following code:

Login-AzureRmAccount

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, and replace the placeholder region names with all 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 to 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

1. 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.

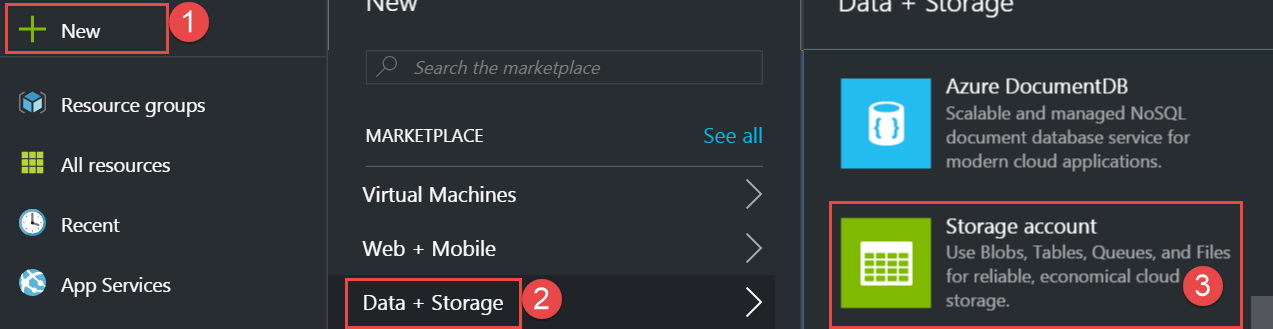
.\ConfigureSubscription.ps1 -SubscriptionId $subscriptionId

#### Task 5: Test the policies

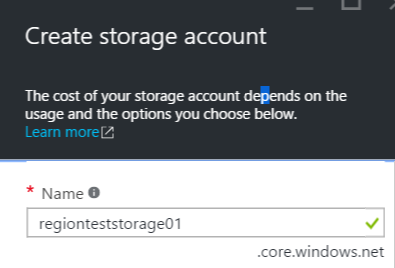
In this task, you will use the Azure management portal to validate the policies work and understand how to identify policy events.

Subtask 1: Test the regional policy

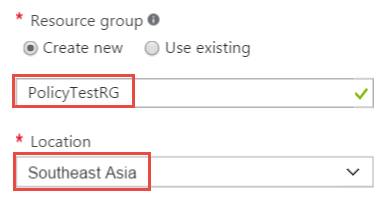
1. Navigate to the Azure management portal in a browser <http://portal.azure.com>, and sign in using the same credentials you logged in with PowerShell.
2. Select **New > Data + Storage > Storage account**.



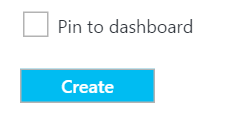
1. Specify a unique name for the storage account.



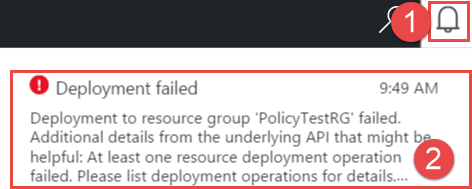
1. In the **Storage Account Creation** blade, ensure the subscription is the same as the subscription you applied the policy to, specify **PolicyTestRG** as the name of the resource group, and specify the location as a region that is not part of the supported regions of your policy.



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

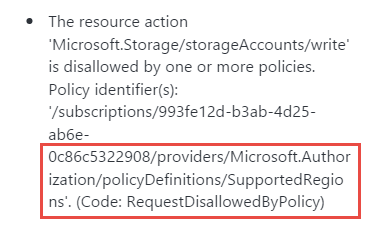


1. In the **Notification** hub, you should see an error that a policy blocked the creation of the storage account. Click the error to view the details as well as the actual policy that blocked the creation.

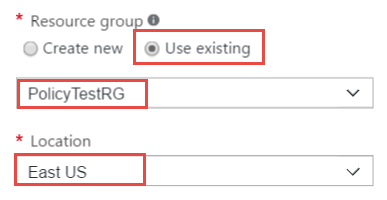


1. Click **Failed. Click here for details** to view the failure details.



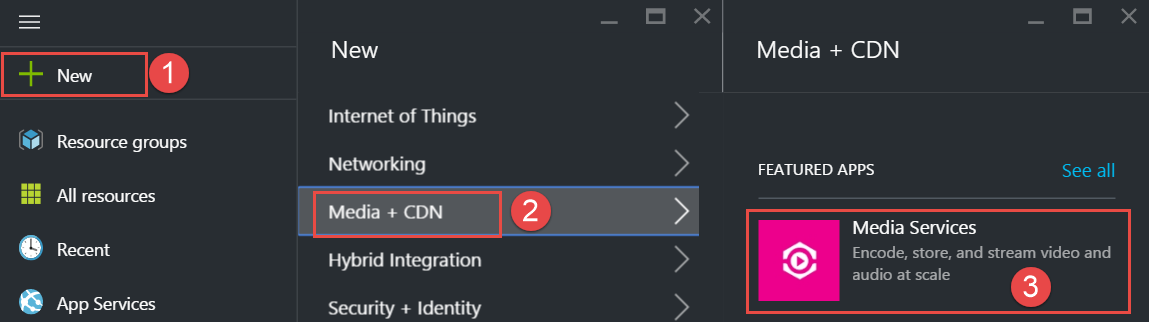


1. Repeat the previous steps, but change the location to a supported region to ensure you can actually create the resource. Specify the existing PolicyTestRG resource group.

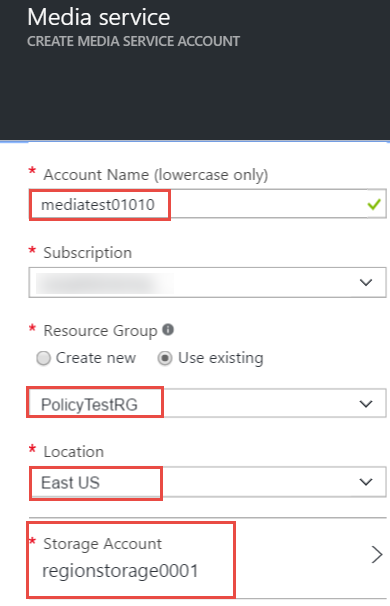


Subtask 2: Test the service catalog policy

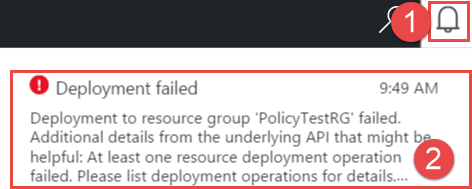
1. Navigate to the Azure management portal in a browser <http://portal.azure.com>, and sign in using the same credentials you used with PowerShell.
2. Click **New > Media + CDN > Media Services**.



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).

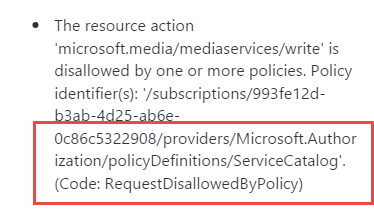


1. In the Notification hub, you should see an error that a policy blocked the creation of the media service. Click the error to view the details as well as the actual policy that blocked the creation.



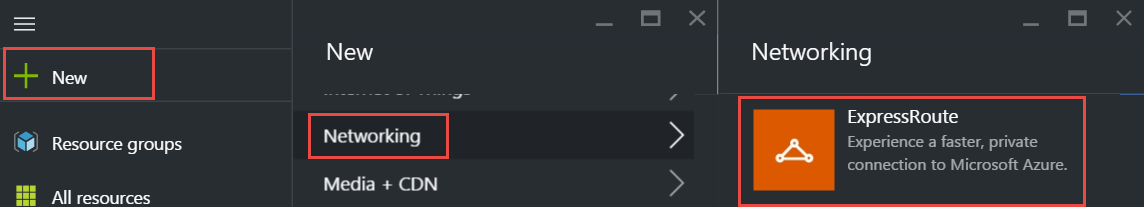
1. Click **Failed. Click here for details** to view the failure details.





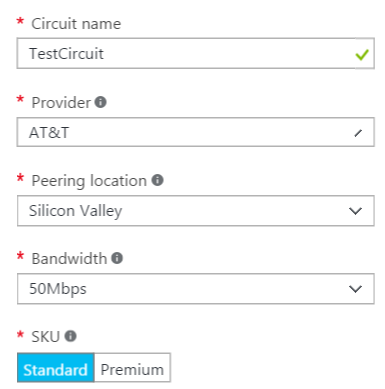
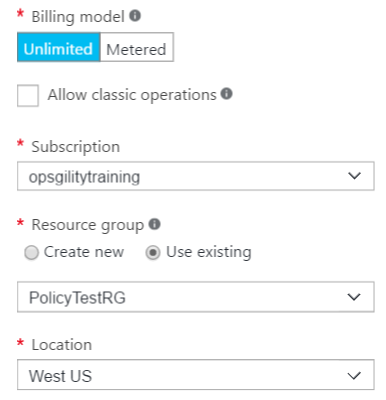
Subtask 3: Test the ExpressRoute circuit policy

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

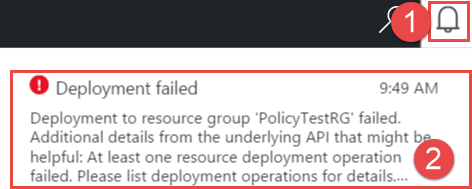


1. Specify the following configuration for the circuit, and click **Create**.

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

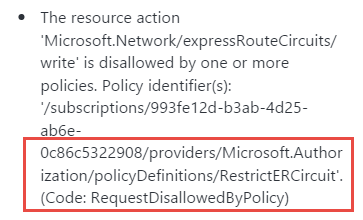
 

1. In the **Notification** hub, you should see an error that a policy blocked the creation of the ExpressRoute circuit. Click the error to view the details as well as the actual policy that blocked the creation.



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





## 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.

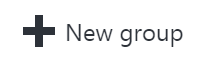
#### Help references

|  |  |
| --- | --- |
| Add new users to Active Directory | https://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/ |

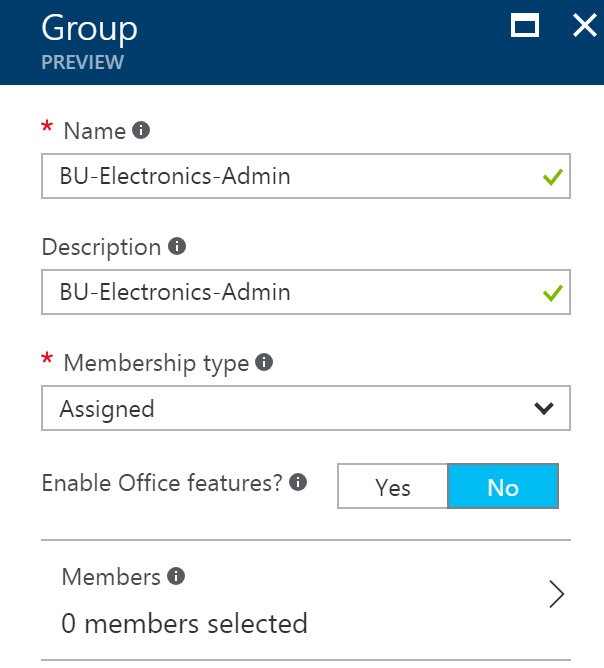
#### Task 1: Create groups in Azure AD for delegation

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 Management portal in your browser (<https://portal.azure.com>).
2. Click **Users and Groups -> All Groups**, and click **New group**.



1. Specify the name: **BU-Electronics-Admin** as the **Name** and **Description**, and change the Membership type to Assigned. Then, click **Create**.

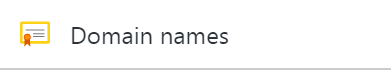


1. Repeat the process, and create another group named **BU-Electronics-Users.**

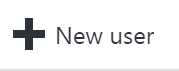
#### Task 2: Create user accounts in Azure AD for delegation

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

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



1. Click **Users and Groups -> All Users**, and click **New user**.



1. Specify the following configuration for the new user:

|  |  |
| --- | --- |
| **Name**  Electronics Admin  **User name:** [ElectronicsAdmin@[yourtenant].onmicrosoft.com](mailto: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. |  |

1. Create a second user with the following configuration:

|  |  |
| --- | --- |
| **Name**  Electronics User  **User name:** [ElectronicsUser@[yourtenant].onmicrosoft.com](mailto: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. |  |

#### 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, and this time, specify the new parameter –AdGroupName.

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

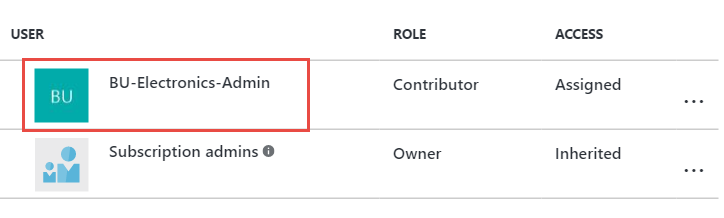
1. Close all instances of your browser (or launch a different type of browser) in in-private or incognito mode.
2. Navigate to the Azure management portal in a browser <http://portal.azure.com>, and sign in using the **ElectronicsAdmin** credentials created earlier. When prompted to change your password, 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**,and add the following JSON code. Then, save the file.

{

"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 it 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.



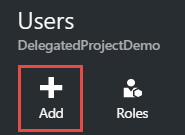
1. Click the **Access** icon.



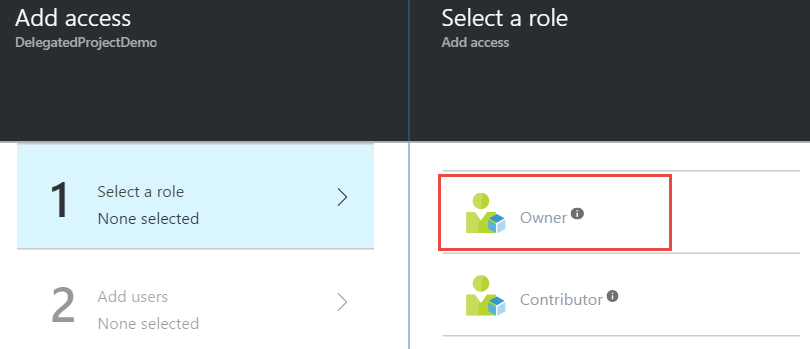
1. Note the BU-Electronics-Admin role is set as owner for the resource group.



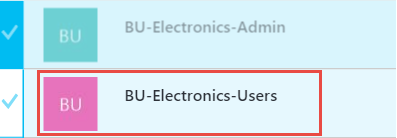
1. Click **Add**.



1. Select **Owner**.



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



## Exercise 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.

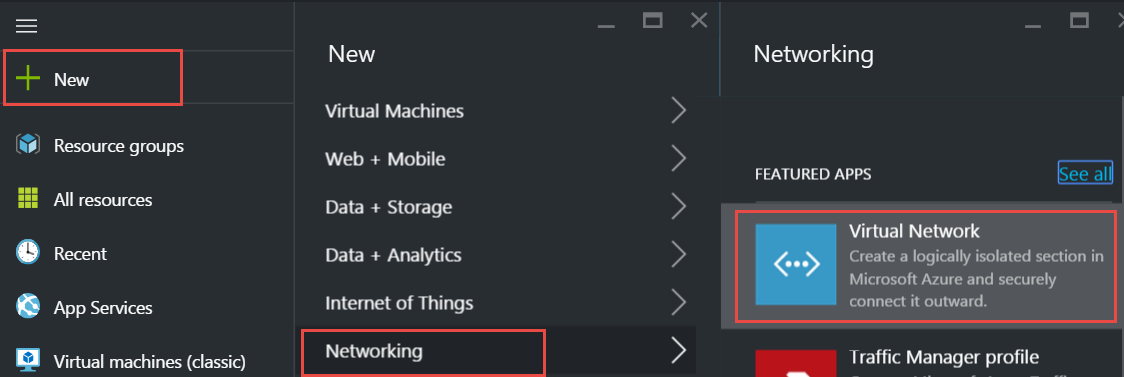
#### Help references

|  |  |
| --- | --- |
| Configuring Point-to-Site Secure VPN | <https://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/ |
| MakeCert.exe | <https://cloudworkshop.blob.core.windows.net/enterprise-ready-cloud/makecert.exe> |

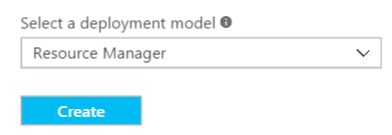
#### Task 1: Create a new virtual network

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

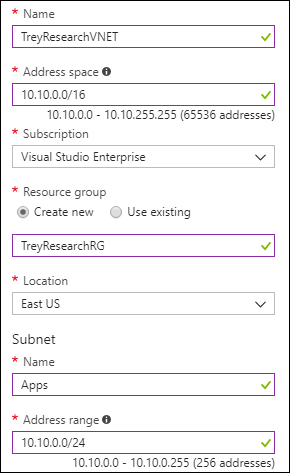
1. Sign in to the Azure Management portal using the subscription owner user account.
2. Click **New** **> Networking >** **Virtual Network**.



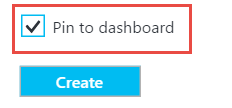
1. Leave the deployment model set to **Resource Manager**, and click **Create**.



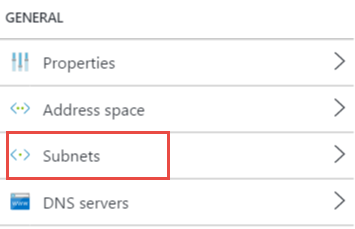
1. Specify the following configuration for the virtual network:
   1. Name: TreyResearchVNET
   2. Address Space: 10.10.0.0/16
   3. Resource Group: TreyResearchRG (Create New)
   4. Location: Choose one of the supported regions.
   5. Subnet Name: Apps
   6. Subnet Address Range: 10.10.0.0/24



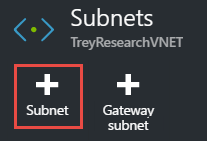
1. Select Pin to dashboard, and click **Create**.



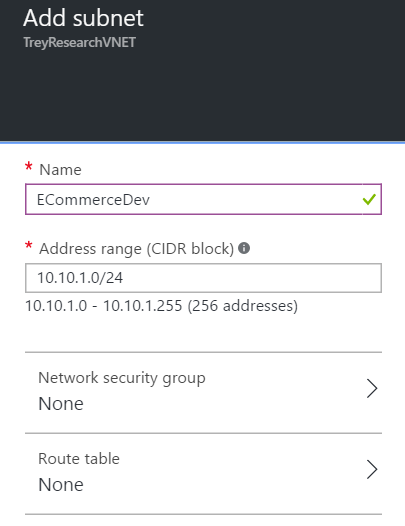
1. After the virtual network is open, click **Subnets.**



1. Click **+ Subnet**.



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



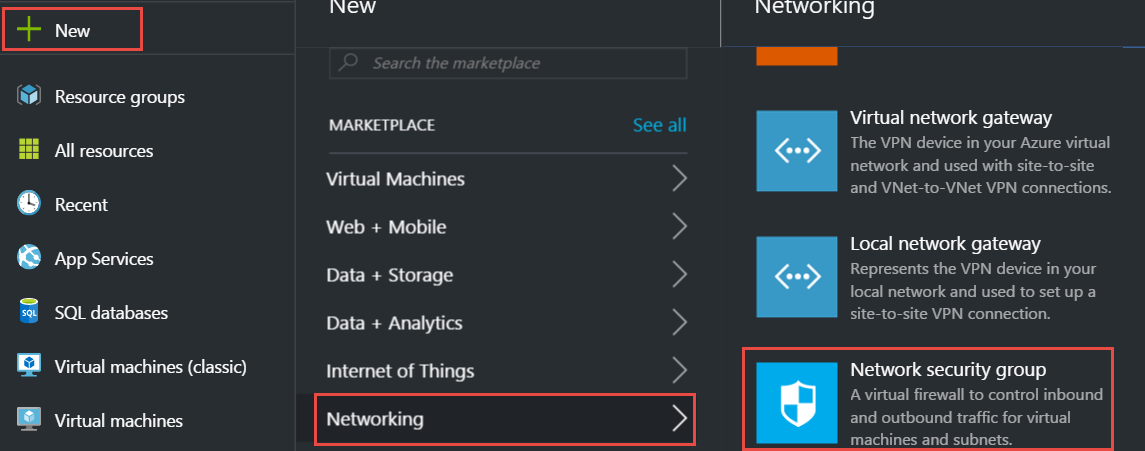
1. Click **+ Gateway subnet** to add a gateway subnet to the virtual network. Click **OK** on the new blade that opens.



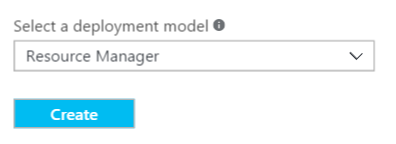
#### Task 2: Configure limited network access

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

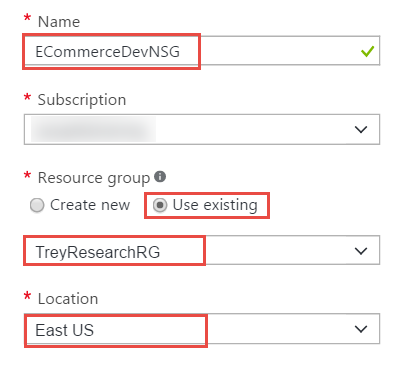
1. Click **New** **>** **Networking** **>** **Network Security Group.**



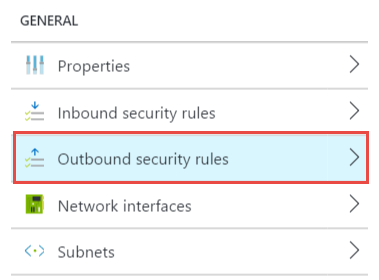
1. Leave the deployment model set to **Resource Manager** and click **Create**.



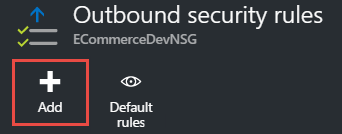
1. Name the NSG **ECommerceDevNSG**, specify the existing TreyResearchRG resource group, and ensure you choose the same region you created the virtual network in.



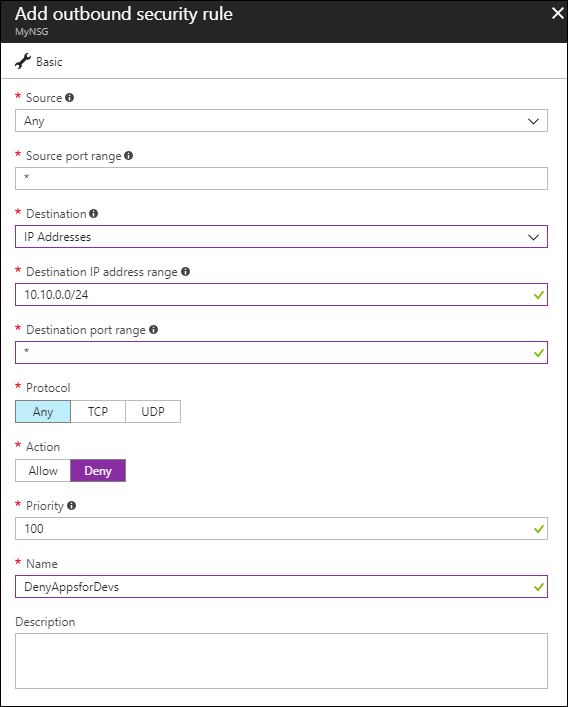
1. Select **Pin to Dashboard**, and click **Create**.
2. After the Network Security Group’s configuration blade opens, click **Outbound security rules**.



1. Click **Add.**



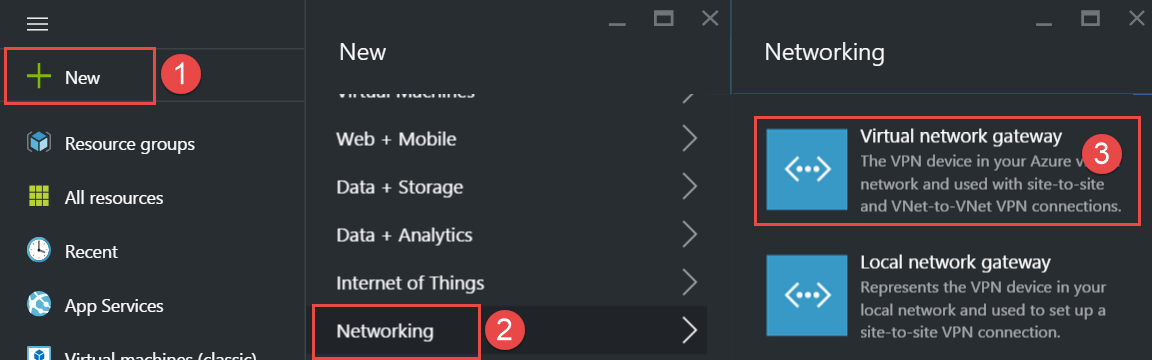
1. Specify the following configuration to restrict access from the developer subnet.
   * Source: Any
   * Source port range: \*
   * Destination: IP Addresses
   * Destination IP Range: 10.10.0.0/24
   * Destination Port Range: \*
   * Protocol: Any
   * Action: Deny
   * Priority: 100
   * Name: DenyAppsforDevs



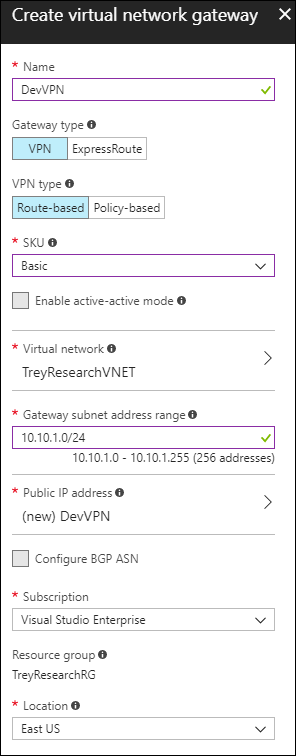
#### Task 3: Configure secure VPN for connectivity

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

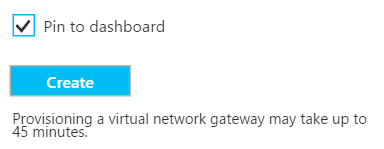
1. Click **New** > **Networking** **>** **virtual network gateway.**



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



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

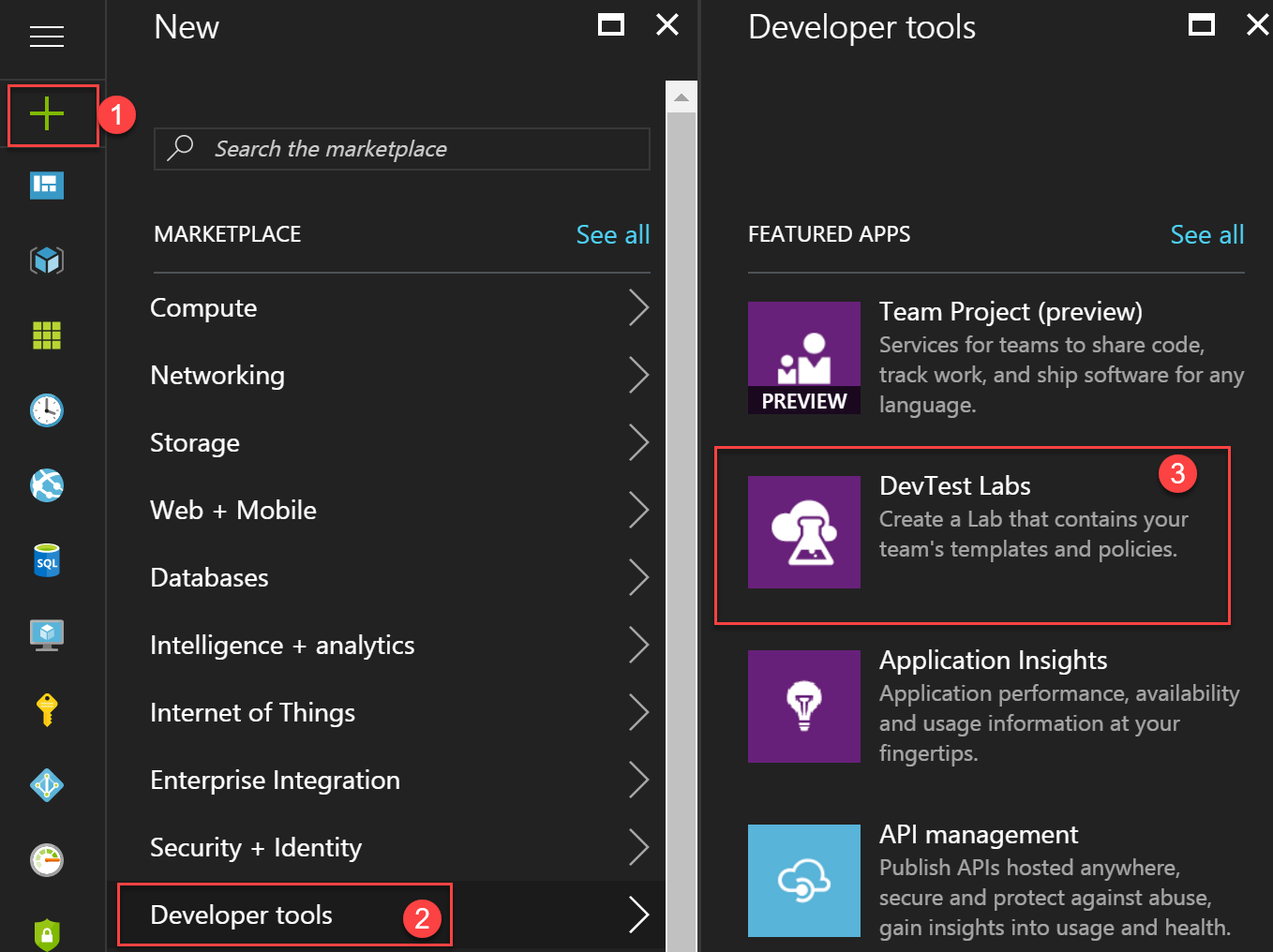


This step will take up to 45 minutes to complete. Continue to the next task. Gateway configuration will be continued in a later task.

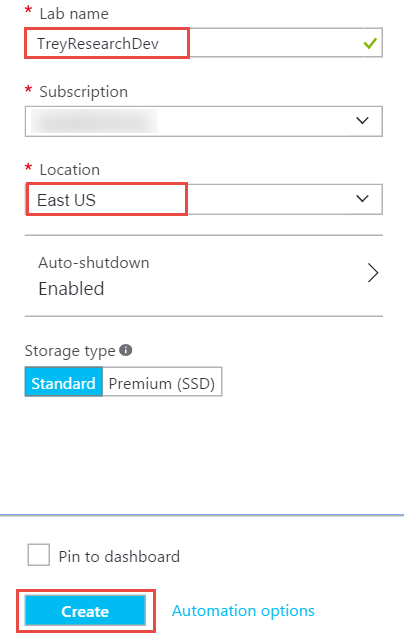
#### Task 4: Create an Azure DevTest lab environment

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

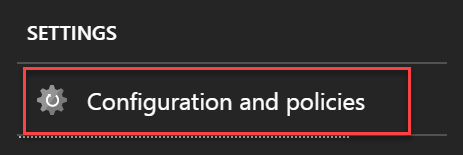
1. Click **New** **>** **Developer tools >** **DevTest Labs**.



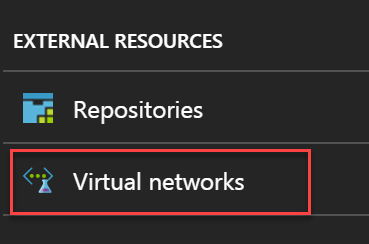
1. Name the lab **TreyResearchDev**, and specify the same region you deployed the virtual network to.



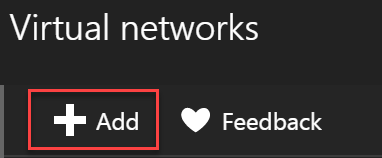
1. Open the DevTest lab environment following the completion of provisioning.
2. Open **Settings > Configuration and policies**.



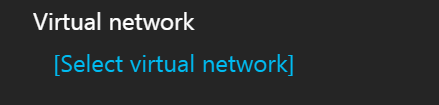
1. Click **Virtual Networks**.



1. Click the **Add** button.



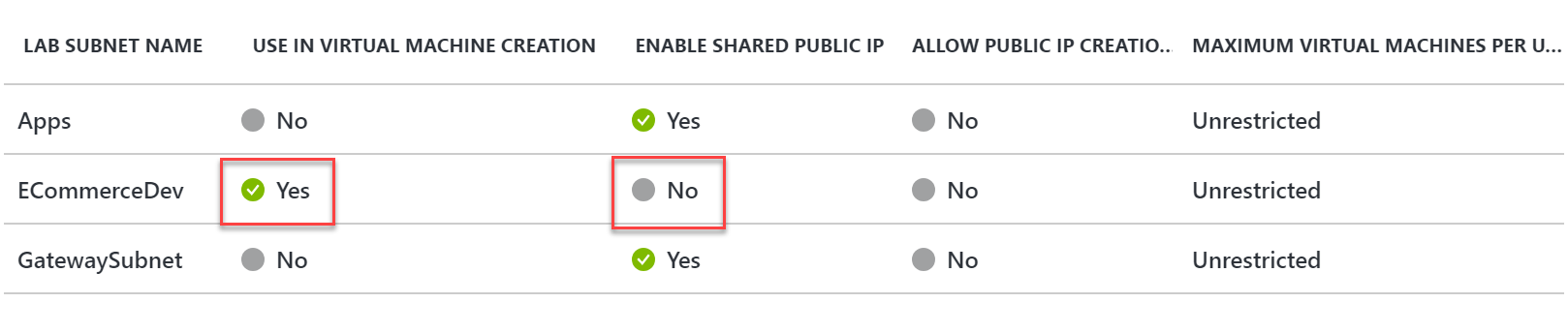
1. Click the [Select virtual network] dropdown.



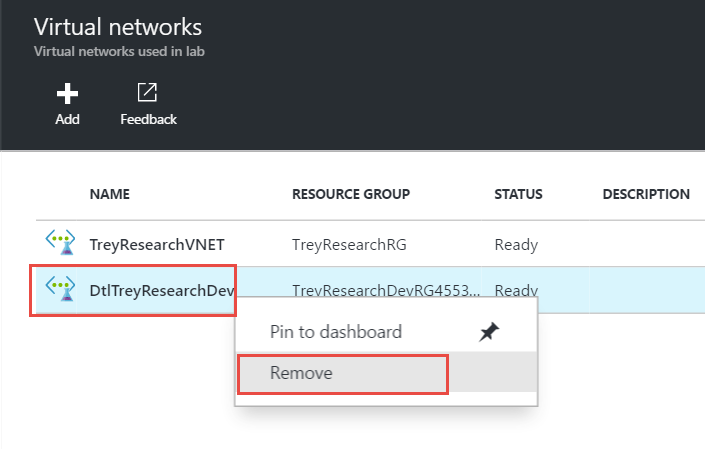
1. Select **TreyResearchVNET**.



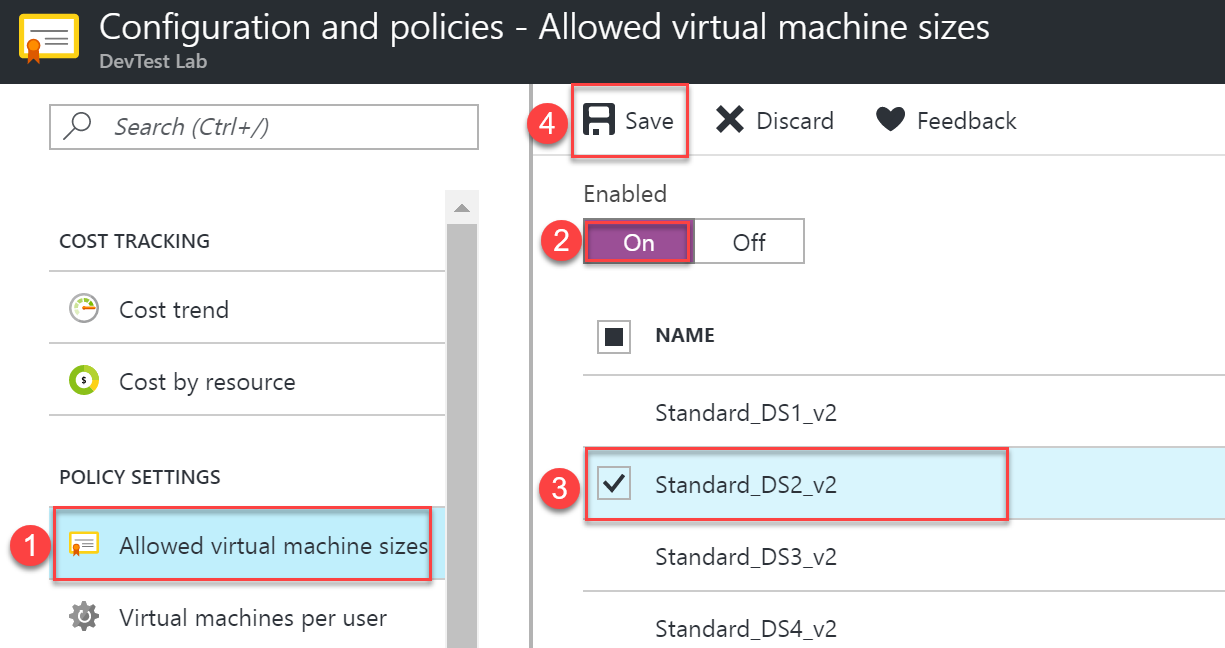
1. Configure the ECommerceDev subnet to **allow** **USE IN VIRTUAL MACHINE CREATION**, and to **disable ENABLE SHARED PUBLIC IP**. Then, click **Save.**



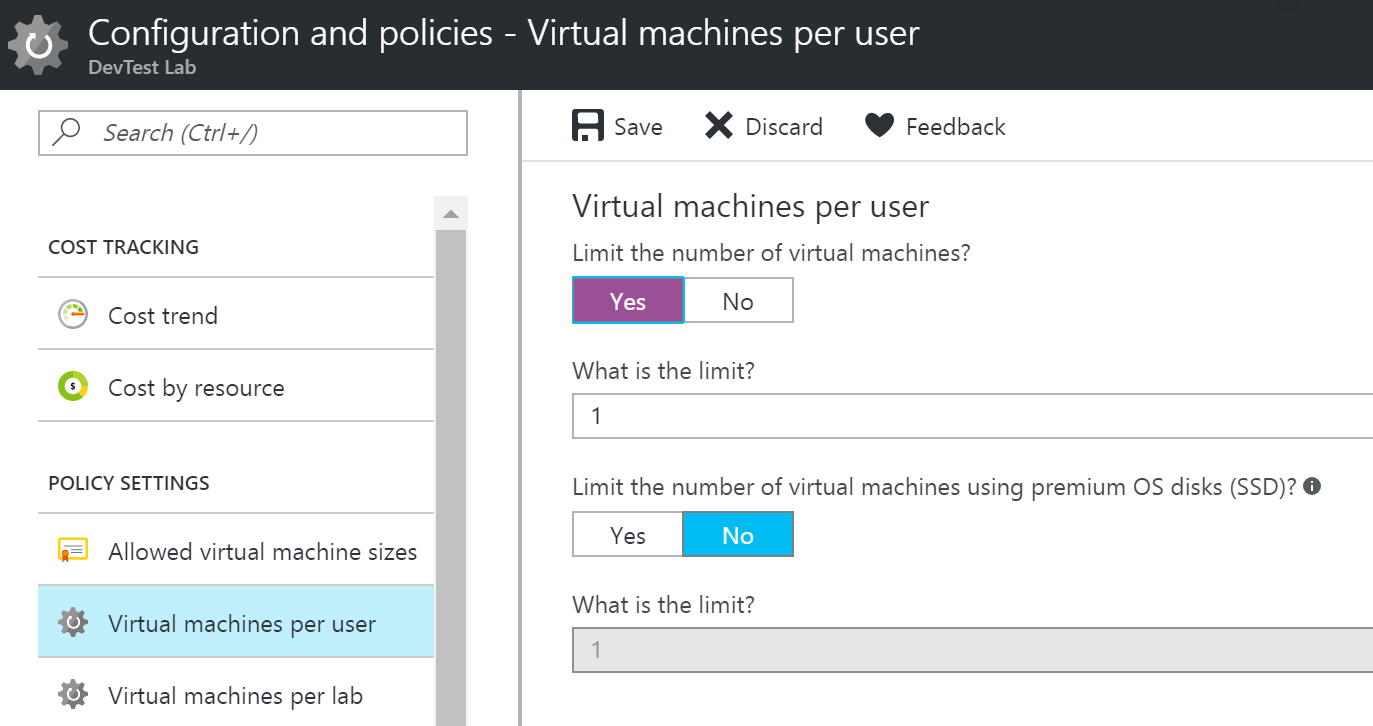
1. Right-click the default virtual network created for the DevTest lab environment, and click **Remove**.



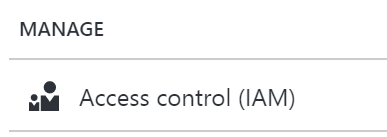
1. Configure a virtual machine policy for this DevTest lab by clicking **Allowed virtual machine sizes**, select **Standard\_DS2\_v2 (or Standard\_DS2)**, and click **Save**.



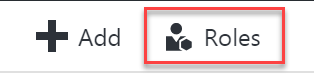
1. Enable the virtual machines per user policy. Set a maximum number of virtual machines per user to one, and click **Save**.



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



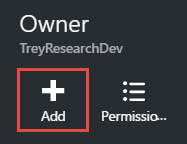
1. Click **Roles**.



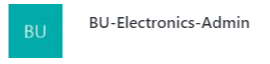
1. Select **Owner**.



1. Click **Add**.



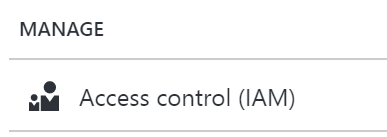
1. Add the **BU-Electronics-Admin** group. Now the user will be able to invite users to the DevTest lab.



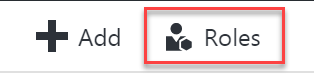
#### Task 5: Test access to the DevTest labs environment

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



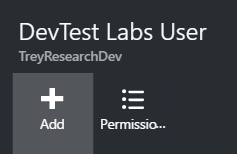
1. Click **Roles**.



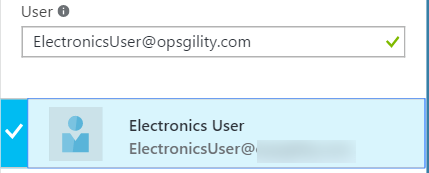
1. Click **DevTest Labs User**.



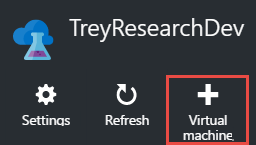
1. Click **Add**.



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



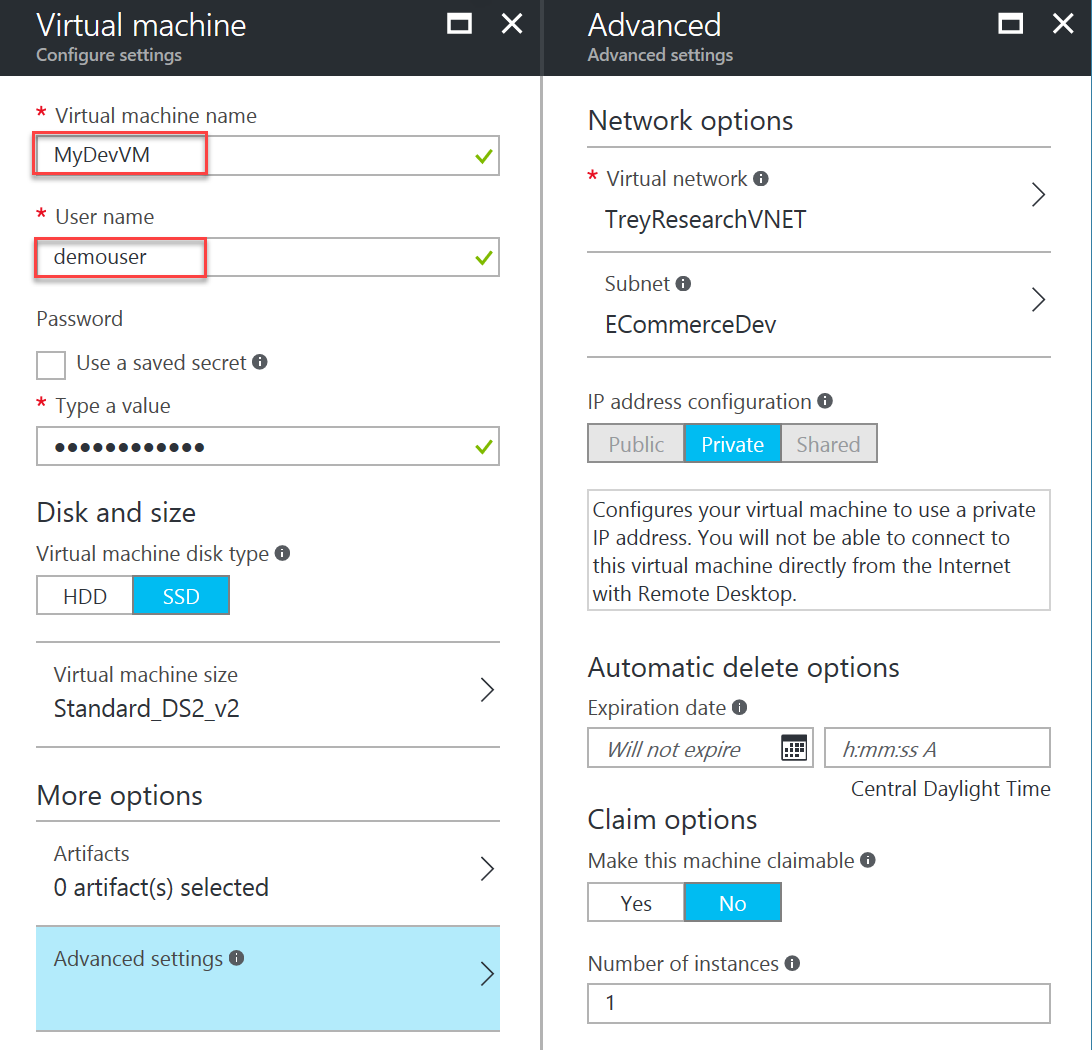
1. Close your browser and sign in with the **ElectronicsUser** account. You will have to change your password and setup a recovery mechanism with this account.
2. Open the DevTest labs environment by clicking **Browse** **>** **DevTest Labs** **>** **TreyResearchDev**.
3. Click **+ Virtual machine**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



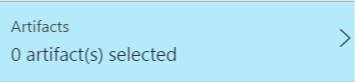
1. Select the Visual Studio Community 2017 on Windows Server image.



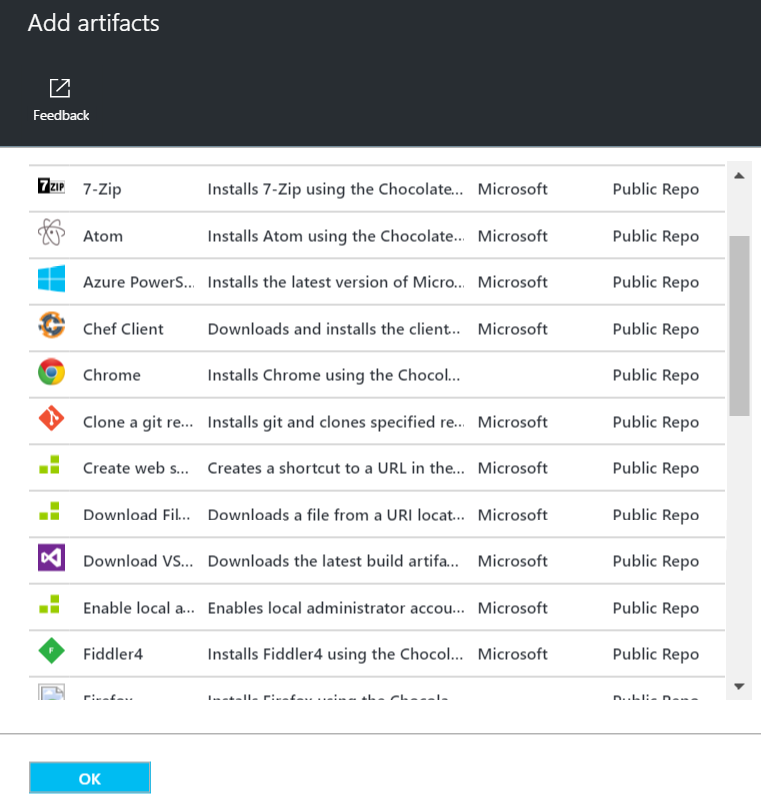
1. Specify a name for the virtual machine as well as a user name and password. Note the VM size, IP address configuration, virtual network, and subnet are not changeable for the user.



1. Click **Artifacts**.



1. Add Azure PowerShell and Fiddler 4 to the artifacts of the VM by selecting 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 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

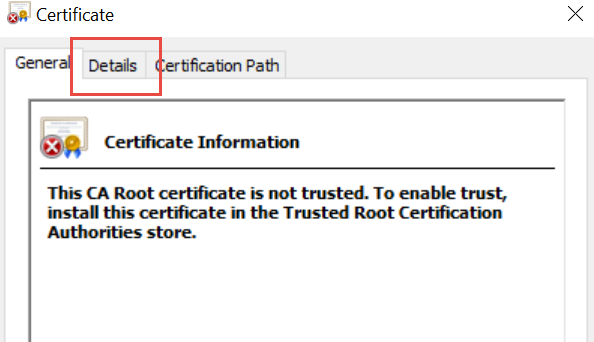
1. Execute the following command to generate a root certificate for configuring a point-to-site VPN gateway.

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

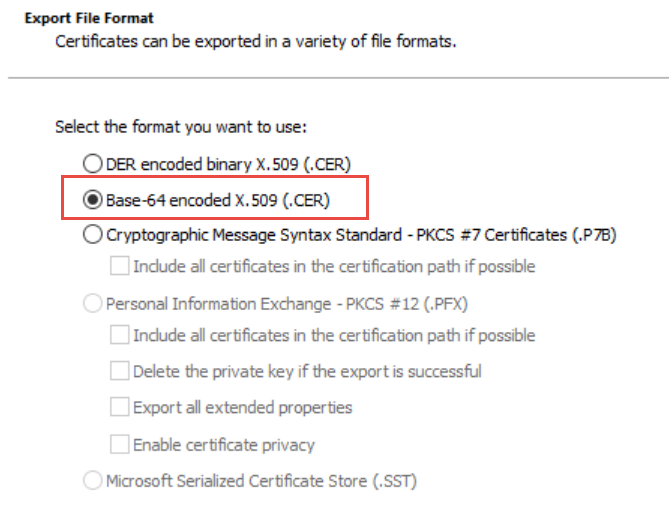
1. Using **File Explorer**, navigate to the C:\Hackathon\ERC folder, and double-click the **P2SRoot.cer** file.
2. Click the **Details** tab of the certificate.



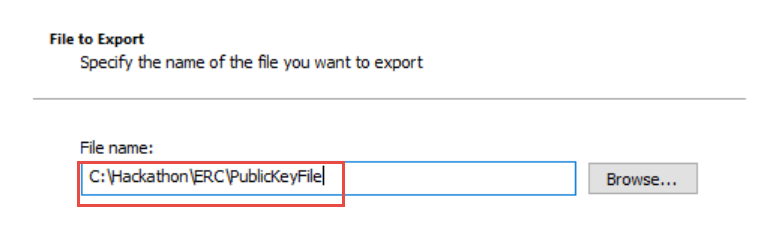
1. Click **Copy to File**.



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. Launch PowerShell, and login to your Azure subscription:

Login-AzureRmAccount

1. In the **PowerShell Console**, execute the following commands to setup variables for the point-to-site gateway configuration:

$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. Execute the following command to copy the value from the cer file to a PowerShell variable.

$publicCertData = Get-Content PublicKeyFile.cer | Out-String

# Strip out the delimieters

$publicCertData = $publicCertData.Replace("-----BEGIN CERTIFICATE-----","")

$publicCertData = $publicCertData.Replace("-----END CERTIFICATE-----","")

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.

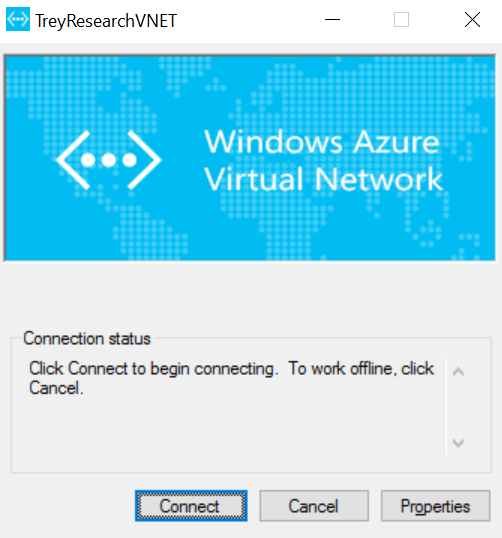
.\VPNclient.exe

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.

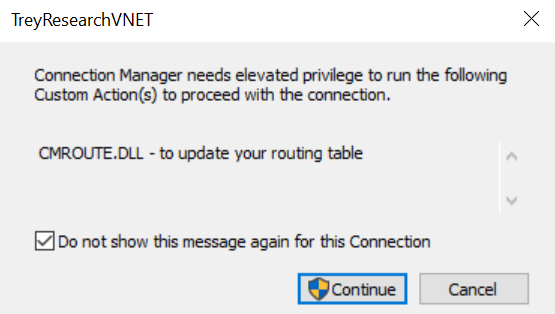


1. Click **Connect** to initiate connection.

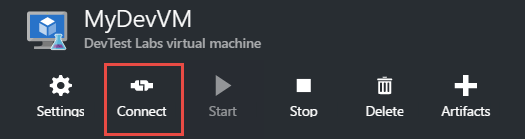




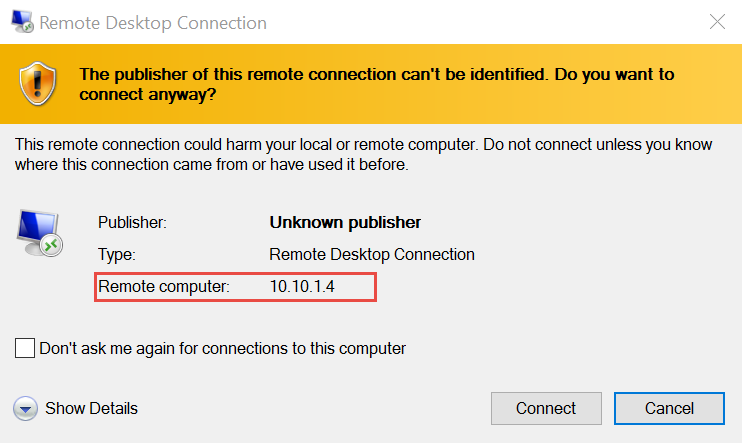
1. Select **Do not show this message again for this Connection**,and click **Continue**.



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.



## 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 Policies

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"

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