WorkshopPLUS

Microsoft Azure Infrastructure as a Service (IaaS)

Azure Resource Manager Policies & Resource Locks

Student Lab Manual

V1.0, October 28, 2016

Information in this document is subject to change without notice. The example companies, organizations, products, people, and events depicted herein are fictitious. No association with any real company, organization, product, person 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, trademarked, 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.

© 2014 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, MS, Windows, Windows NT, MSDN, Active Directory, BizTalk, SQL Server, SharePoint, Outlook, PowerPoint, FrontPage, Visual Basic, Visual C++, Visual J++, Visual InterDev, Visual SourceSafe, Visual C#, Visual J#,  and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation in the U.S.A. and/or other countries.

Other product and company names herein may be the trademarks of their respective owners.

Contents

[Prerequisites 4](#_Toc510620729)

[Exercise 1 – ARM Management Policy Using PowerShell 5](#_Toc510620730)

[Task 1 – Creating a Resource Group 5](#_Toc510620731)

[Task 2 – Creating ARM Management Policies 5](#_Toc510620732)

[Exercise 2 – Applying an ARM Management Policy Using PowerShell 7](#_Toc510620733)

[Task 1 – Getting your subscription ID 7](#_Toc510620734)

[Task 2 – Applying ARM Management Policies 8](#_Toc510620735)

[Task 3 – Testing ARM Management Policies 9](#_Toc510620736)

[Exercise 3 – Resource Locks Using the Azure Portal 10](#_Toc510620737)

[Task 1 – Creating and applying a Resource Lock 10](#_Toc510620738)

[Task 2 – Testing Resource Locks 12](#_Toc510620739)

**Introduction to Microsoft Azure Resource Manager Policies & Resource Locks**

## Prerequisites

The following is required to complete this hands-on lab:

* Microsoft Azure PowerShell v1.0 and above
* A Microsoft Azure subscription

In this lab, you’ll be creating two ARM management policies that define the **Regions** and a **Resource Naming Convention** that are allowed during resource creation. You will then apply these policies to an existing resource group and finally, you will enable a Resource Lock on the resource group. The ARM Management policies will be created and applied using PowerShell and the Resource Lock will be created and applied using the Azure portal.

You'll learn:

* The ARM policy definition language.
* How to create and apply an ARM management policy using PowerShell.

For this lab, you can copy and paste code from the lab manual into the lab interface.

# Exercise 1 – ARM Management Policy Using PowerShell

## Task 1 – Creating a Resource Group

The ARM management policies that you will be creating will be applied to a new resource group. This is because ARM policies do not apply to pre-existing resources that are in a resource group.

1. Open PowerShell ISE as an Administrator.
2. In the PowerShell ISE command prompt window, type in **Add-AzureRmAccount** and press Enter.
3. Enter your credentials into the login dialog box and click **Next** or press Enter.
4. Enter your password and click **Sign in** or press Enter.
5. Type **New-AzureRmResourceGroup -Name [YourResourceGroupName] -Location [YourResourceGroupLocation]** e.g. **New-AzureRmResourceGroup -Name PolicyRG** **-Location westeurope** and press Enter. This will create a new resource group which you will apply your policies to.

**Do not close this PowerShell session**

## Task 2 – Creating ARM Management Policies

In its simplest form, the ARM policy definition language consists of:

* **A Condition or Logical operators**: Is a condition or set of conditions which must be met in order to allow the operation to proceed (e.g., If: Location = westeurope)
* **An Effect**: Which is what happens when the condition is satisfied – either deny, audit or append.
* **Deny** generates an event in the audit log and fails the request
* **Audit** generates an event in audit log but does not fail the request
* **Append** adds the defined set of fields to the request such as appending the country code to a name.

Example:

If: **Location = westeurope**, then **Audit**.

This will allow creation of the resource and log an event in the audit log for an administrator to view later on.

1. The first policy will define the allowed **Regions (**i.e., northeurope & westeurope) that a resource can be created in. Copy & paste the code below into your PowerShell script pane.

$locationpolicy = New-AzureRmPolicyDefinition -Name regionPolicyDefinition -Description "Policy to allow resource creation only in certain regions" -Policy '{

"if" : {

"not" : {

"field" : "location",

"in" : ["northeurope" , "westeurope"]

}

},

"then" : {

"effect" : "deny"

}

}'

1. Press F5 (or click the Run Script icon in your PowerShell script pane). This has now created the above policy and stored it in a variable named **$locationpolicy**.
2. The second policy will define the allowed **Resource Naming Convention** (i.e., the resource should be prefixed with **Prod** and suffixed with **Europe)**. Copy & paste the following code into your PowerShell script pane (below the existing code).

$namingpolicy = New-AzureRmPolicyDefinition -Name NamingPolicyDefinition -Description "Policy to specify allowed naming convention" -Policy '{

"if" : {

"not" : {

"field" : "name",

"like" : "Prod\*Europe"

}

},

"then" : {

"effect" : "deny"

}

}'

1. In your PowerShell script pane select the newly pasted lines and press F8 (or click the Run Selection icon). This has now created the second policy and stored it in a variable named **$namingpolicy**.

**Do not close this PowerShell session**

# Exercise 2 – Applying an ARM Management Policy Using PowerShell

## Task 1 – Getting your subscription ID

In order to apply your ARM management policy, you will need get your subscription ID. This is because it is one of the requirements for the PowerShell command that you will use to apply your policy to your resource group.

If you don’t have an existing PowerShell session open that is logged into your Azure subscription, perform steps 1 through 4*.*  ***If your PowerShell window is still open, proceed to Step 5.***

1. Open PowerShell ISE as an Administrator.
2. In the PowerShell ISE command prompt window, type in **Add-AzureRmAccount** and press Enter.
3. Enter your credentials into the login dialog box and click **Next** or press Enter.
4. Enter your password and click **Sign in** or press Enter.
5. Type **Get-AzureRmSubscription** and press Enter, then record your subscription ID.

SubscriptionName : MySubscription

SubscriptionId : 12345678-aaaa-bbbb-cccc-123456789101

TenantId : d3452b29-6f86-45ad-bf1b-123456789876

State : Enabled

## Task 2 – Applying ARM Management Policies

Now that you’ve completed the required prerequisite tasks, you are ready to apply the policies that you’ve created earlier on. ARM Management policies can be applied at the Subscription, Resource Group or Resource level.

1. To apply the **Regions** policy, copy & paste the following code into your PowerShell script pane below the existing code.

New-AzureRmPolicyAssignment -Name locationPolicyAssignment -PolicyDefinition $locationpolicy -Scope /subscriptions/YourSubscriptionIDRecordedFromEarlier/resourceGroups/PolicyRG

1. Edit the newly pasted code to include your subscription ID.
2. Select the newly pasted code and press F8 (or click the Run Selection icon in your PowerShell script pane). This has now applied the **Regions** policy to your new resource group.
3. To apply the **Resource Naming Convention** policy, copy & paste the following code into your PowerShell script pane below the existing code.

New-AzureRmPolicyAssignment -Name namingConvPolicyAssignment -PolicyDefinition $namingpolicy -Scope /subscriptions/YourSubscriptionIDRecordedFromEarlier/resourceGroups/PolicyRG

1. Edit the newly pasted code to specify your subscription ID.
2. Select the newly pasted code and press F8 (or click the Run Selection icon in your PowerShell script pane). This has now applied the **Resource Naming Convention** policy to your new resource group.

## Task 3 – Testing ARM Management Policies

Now that you’ve applied your policies to your resource group, you can test them to confirm that they are being enforced. You will be creating a new Network Security Group with default settings during the test.

1. Copy & paste the following code into your PowerShell script pane below the existing code. Edit the newly pasted code to reflect your actual resource group name (i.e., PolicyRG).

New-AzureRmNetworkSecurityGroup -Name PolicyTestNSG -ResourceGroupName YourResourceGroupNameFromEarlier -Location westus

1. Select the newly pasted command and press F8 toselection t (it should fail)
2. Copy & paste the following code into your PowerShell script pane. Edit the newly pasted code to reflect your actual resource group name (i.e., PolicyRG).

New-AzureRmNetworkSecurityGroup -Name ProdPolicyTestNSGEurope -ResourceGroupName **YourResourceGroupNameFromEarlier** -Location westeurope

1. Select the newly pasted command and press F8 to run the selection. Observe the result (it should succeed).
2. To remove the policy assignments, run the following commands, substituting for the highlighted items, as appropriate:

Remove-AzureRmPolicyAssignment -Name locationPolicyAssignment -Scope /subscriptions/subscriptionID/resourceGroups/**YourResourceGroupNameFromEarlier**

Remove-AzureRmPolicyAssignment -Name namingConvPolicyAssignment -Scope /subscriptions/subscriptionID/resourceGroups/YourResourceGroupNameFromEarlier

# Exercise 3 – Resource Locks Using the Azure Portal

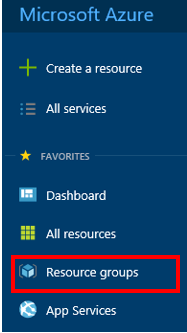
## Task 1 – Creating and applying a Resource Lock

Resources in a resource group can be accidentally deleted at any time. To help you lower the risk of this happening, a Resource Lock can be implemented. Resource Locks can be enabled at the Subscription, Resource Group or Resource level and have two settings **CanNotDelete** and **ReadOnly**:

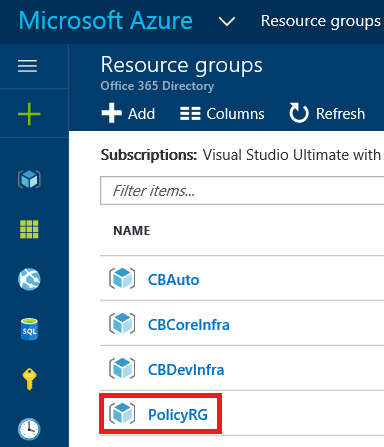
**CanNotDelete** means authorized users can read and modify a resource, but they can't delete it.

**ReadOnly** means authorized users can read from a resource, but they can't modify or delete it. The permission on the resource is restricted to the Reader role.

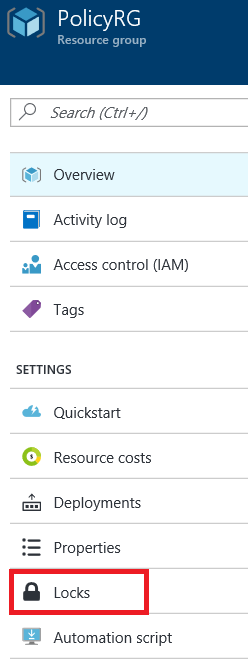
1. Login to the Azure portal (http://portal.azure.com).
2. Select **Resource groups** in the left-hand pane.



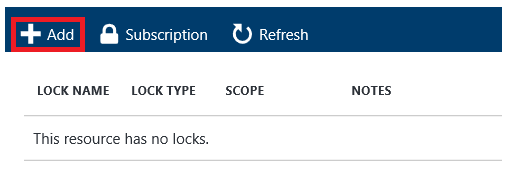
1. Select the resource group that was created during the ARM management policy exercise. In this case, **PolicyRG**.



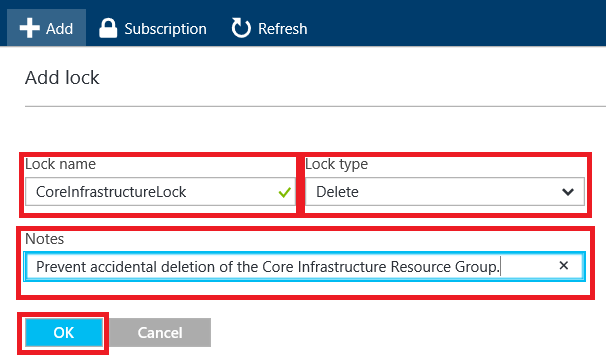
1. Click **Locks** in the *Resource group’s* configuration pane.



1. Click **+Add** in the *Locks* configuration pane.



1. To configure the lock:
   * Type in a Name
   * Select the **Delete** *Lock type*
   * Type in a description for the Resource Lock
   * Click **OK**.



## Task 2 – Testing Resource Locks

Now that you’ve created and applied your resource lock to your resource group, you can test it to confirm that it is being enforced. You will be attempting to delete the Network Security Group that was created earlier.

1. Return to the PowerShell session that was used to create your Network Security Group.
2. Create a new tab (File | New).
3. Copy & paste the following code into your PowerShell script pane.

Remove-AzureRmNetworkSecurityGroup -Name ProdPolicyTestNSGEurope -ResourceGroupName YourResourceGroupNameFromEarlier

1. Select the pasted code and press F5 to run the selection. Confirm the process by clicking **Yes**. Observe the result (you should see an error stating that the resource is locked).

This is the end of this lab. Do not delete these resources, as they may be used in later labs. However, you should ensure that any existing VMs are in a stopped (deallocated) state, to conserve costs.