Hands-on Labs Security Guidelines

Permissions

In the interest of security, the default behavior is to provide read-only access to the resource group. You may use PowerShell to modify the role-based access of the user if your lab requires additional permissions. However, you must use the minimal set of roles required to create the resources for the lab. The full list of available roles and the actions allowed to each can be retrieved from Azure with the PowerShell command `Get-AzureRmRoleDefinition`. See Appendix B for the suggested subset of roles recommended for Hands-on Labs.

Valorem will reject scripts that elevate the user to "Contributor" or higher as these roles pose a significant risk to the platform.

† Except for some limited cases in facilitator-led sessions

Important: users get *access* to a resource group, not a subscription. This means the old portal is not available…

Hands-on Labs Role Based Access

By default, the Hands-on Labs user is given Reader access to the resource group in which the lab is deployed. If additional access is required for your content package, such as virtual machine creation, network modification, or other higher level tasks; we recommend using the resource specific contributor roles. See the below table for the most often used roles.

Role	**Virtual Machine Contributor**
Lets you manage virtual machines, but not access to them, and not the virtual network or storage account they're connected to.	
Role	**SQL Server Contributor**
Lets you manage SQL servers and databases, but not access to them, and not their security -related policies.	
Role	**Storage Account Contributor**
Lets you manage storage accounts, but not access to them.	
Role	**Web Plan Contributor**
Lets you manage the web plans for web sites, but not access to them.	
Role	**Web Site Contributor**
Lets you manage websites (not web plans), but not access to them.	

Hands-on Labs Security Guidelines

In addition to the standard RBAC roles available in Azure, we have created the following roles specifically for Hands-on Labs.

Role	**[Hands-on Labs] Data Factory Contributor Additional Permissions**
Additional "write resource group" permission missing from Data Factory Contributor role.	

Actions

Microsoft.Resources/subscriptions/resourceGroups/write

NoActions

Role	**[Hands-on Labs] Data Lake Contributor**
Can create and modify Data Lake resources	

Actions

Microsoft.DataLakeStore/accounts/*
Microsoft.DataLakeAnalytics/accounts/*

NoActions

Role	**[Hands-on Labs] Sql Server Data Masking Contributor**
Can configure data masking rules	

Actions

Microsoft.Sql/servers/databases/dataMaskingPolicies/*

NoActions

Hands-on Labs Security Guidelines

Role	**[Hands-on Labs] Virtual Machine Operator**
Can restart virtual machines	

Actions

Microsoft.Compute/virtualMachines/start/action
Microsoft.Compute/virtualMachines/restart/action

NoActions

Creating a Local User for the HOL Jumphost

Due to security concerns, we ask that you implement a pseudo-random password pattern for any local user accounts that will be granted remote access to the virtual machine. This would include the administrative user used at VM creation and any local user account that is added to the “Remote Users” group. An example of a pattern that meets the Windows password requirements is shown below. Note the use of a special character, and uppercase conversion at the start of the concatenation command. This pattern can be adapted as desired for your ARM template, so long as it continues to meet the requirements for Windows passwords and randomness.

Example Pattern

```
    "vmLocalUserPassword":{

      "type": "string",

      "defaultValue":
"[concat(toUpper(substring(uniqueString(resourceGroup().id,subscription().id),0,6)),'#',
substring(uniqueString(resourceGroup().id, subscription().id),6,6))]"

    }
```