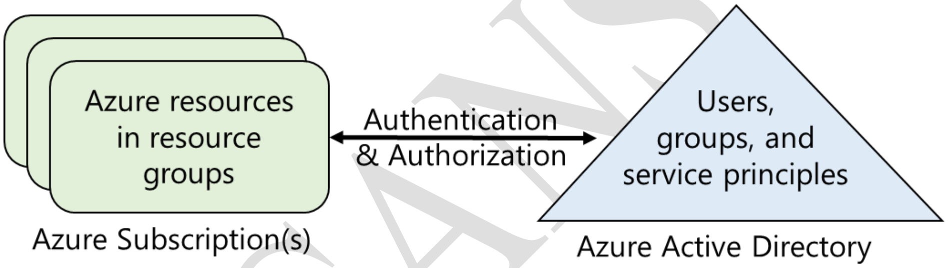
Agenda: Manage Azure Subscriptions

- Understanding Azure Subscriptions
- Configuring Role Based Access Control
- RBAC using Portal
- RBAC using PowerShell and CLI
- Custom Roles for RBAC
- Managing Subscription Policies
- Locking Resources
- Checking Resources Limits
- Resource Tags

Understanding Azure Subscriptions

An Azure subscription is a logical unit of Azure services that is linked to an Azure account. Billing for Azure services is done on a per-subscription basis.


Azure resources in resource groups
Azure Subscription(s)
Authentication & Authorization
Users, groups, and service principles
Azure Active Directory

Azure accounts

- An Azure account determines how Azure usage is reported and who the **Account Administrator** is.
- Accounts and subscriptions are created in the Azure Account Center.
- The person who creates the account is the Account Administrator for all subscriptions created in that account. That person is also the **default Service Administrator** for the subscription.
- There are three roles related to Azure accounts and subscriptions:

Administrative role	**Limit**	**Summary**
Account Administrator	1 per Azure account	Authorized to access the Account Center (create subscriptions, cancel subscriptions, change billing for a subscription, change Service Administrator). This role has full control over the subscription and is the account that is responsible for billing.

Administrative role	Limit	Summary
Service Administrator	1 per Azure subscription	Authorized to access Azure Management Portal for all subscriptions in the account. By default, same as the Account Administrator when a subscription is created. This role has control over all the services in the subscription.
Co-administrator	200 per subscription	Same as Service Administrator but can't change the association of subscriptions to Azure directories.

Access control in Azure starts from a billing perspective.

- The actual owner of an Azure account – accessed by visiting the Azure Accounts Center – is the Account Administrator (AA).
- **Subscriptions are a container for billing**, but they also act as a security boundary.
- Your Azure subscription has a trust relationship with Azure AD, which means that it trusts the directory to authenticate users, services, and devices.
- Multiple subscriptions can trust the same directory, but each subscription trusts only one directory.

For a user to access to your Azure resources, you would add them to the Azure AD directory associated with your subscription.

Azure Account Administration = sandeepsonideccansoft.onmicrosoft.com (sandeepsoni@deccansoft.com)

Hierarchy: Subscription -> Resource Group -> Resource

Tenant (Azure AD) -> Domain -> User & Groups

- Azure AD Tenant (sandeepsonideccansoft.onmicrosoft.com)
    - Domains
        - sandeepsonideccansoft.onmicrosoft.com (Primary)
        - bestazuretraining.com (verified)
    - Organization Users (only verified domains are allowed)
        - abc@sandeepsonideccansoft.onmicrosoft.com
        - xyz@sandeepsonideccansoft.onmicrosoft.com
        - abc@bestazuretraining.com
    - Guest Users (Non-Verifed Domans / Any Microsoft Account)
        - abc@hotmail.com
        - zyx@microsoft.com
        - test@contoso.com

- Azure Subscription is binding to an Azure AD.
    - FREE Trail
    - Azure Sponsorship
    - Visual Studio Subscription
    - Pay-As-You-Go
    - Enterprise Aggrement
- Permissions to AD Users
    - Permission Scopes (Users can be given access to)
        - Subscription
        - ResourceGroup
        - Resource

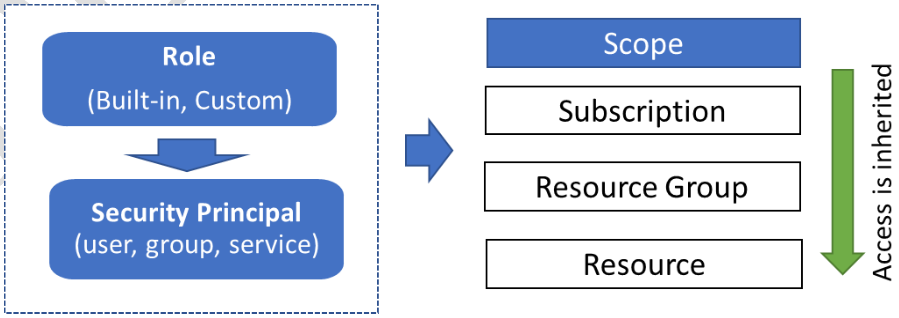
Configuring Role Based Access Control

- Managing access to resources in Azure is a critical part of an organization's security and compliance requirements. Role-based access control (RBAC) is the capability for you to grant appropriate access to Azure AD users, groups, and services.
- RBAC is configured by selecting a role (the definition of what actions are allowed and/or denied), then associating the role with a user, group or service principal.
- **Finally, this combination of role and user/group/service principal is scoped to either the entire subscription, a resource group, or specific resources within a resource group.**

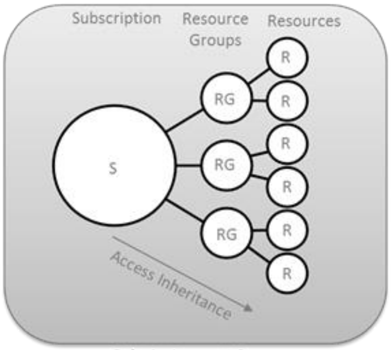
Role Assignment:

Roles can be assigned to the following types of Azure AD security principals:

- Users
- Groups
- Service principal



Resource Scope: Subscription or Resource Group or Specific Resource



Role Assignment Scopes

Role Definition:

Each role is a set of properties defined in a **JSON** file. This role definition includes **Name**, **Id**, and **Description**. It also includes the allowable permissions (**Actions**), denied permissions (**NotActions**), and **scope** (read access, etc.) for the role.

Name: Owner

ID: 8e3af657-a8ff-443c-a75c-2fe8c4bcb65

IsCustom: False

Description: Manage everything, including access to resources

Actions: {*}

NotActions: {}

AssignableScopes: {/}

In this example the Owner role means all (*) actions, no denied actions, and all (/) scopes.

Actions:

It specifies the Azure operations to which the role grants access. It is a collection of operation strings that identify securable operations of Azure resource providers.

Operation strings follow the format of **Microsoft.<ProviderName>/<ChildResourceType>/<action>**.

Examples:

- */read grants access to read operations for all resource types of all Azure resource providers.
- Microsoft.Compute/* grants access to all operations for all resource types in the Microsoft.Compute resource provider.
- Microsoft.Network/*/read grants access to read operations for all resource types in the Microsoft.Network resource provider of Azure.
- Microsoft.Compute/virtualMachines/* grants access to all operations of virtual machines and its child resource types.

- `Microsoft.Web/sites/restart` grants access to restart websites.

NotActions:

Use the **NotActions** property if the set of operations that you wish to allow is more easily defined by **excluding restricted operations**. The access granted by a custom role is computed by subtracting the **NotActions** operations from the **Actions** operations.

AssignableScopes:

This property of the role specifies the scopes (subscriptions, resource groups, or resources) within which the custom role is available for assignment.

  - /subscriptions/[subscription id]
  - /subscriptions/[subscription id]/resourceGroups/[resource group name]
  - **/subscriptions/[subscription id]/resourceGroups/[resource group name]/[resource]**

Example 1: Make a role available for assignment in **two** subscriptions.

```
"/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e", "/subscriptions/e91d47c4-76f3-4271-a796-21b4ecfe3624"
```

Example 2: Makes a role available for assignment only in the Network resource group.

```
"/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/NetworkRG"
```

Built-in Roles and their Action and NotActions

Role	**Action**	**NotActions**	**Description**
Owner	*		This role has full access to all the resources and can **delegate** access to others.
Contributor	*	Microsoft.Authorization/*/Delete, Microsoft.Authorization/*/Write,	This role can **create and manage** all types of resources, but **can't grant access** to other users and groups.
Reader	*/read		This role can **view** existing Azure resources

Walkthrough: To manage RBAC by using the Azure portal, perform the following steps:

1. In the Azure portal, locate the Users blade for the resource for which you plan to manage RBAC.
   Eg: App Services → Select the App → Settings → **Access Control (IAM)**.
   OR
   Settings → Select Subscription → Settings → Users
2. Click the Add icon on the Users blade.
   a. Select the role that you want to assign. Eg: **Reader**

b. Search for and select the user, group, or application to which you want to grant access. You can search the directory for users, groups, and applications by using display names, email addresses, and object identifiers. (User should have been created using Classic Portal in the Azure AD Directory)

c. Click OK to confirm the selection.

3. In new instance of the browser, Login using the identity of the user who has been given permissions and verify (that the user is added as a reader to your Azure subscription)

Programming using Azure PowerShell and CLI

To inspect individual operations that a role grants access to, review the Actions and NotActions properties of the role.

Get-AzureRmRoleDefinition [[-Name] <String>]
[-Scope <String>]
[-Custom]
[-DefaultProfile <IAzureContextContainer>]
[<CommonParameters>]

Example1: **Get-AzureRmRoleDefinition**

Example2: **Get-AzureRmRoleDefinition** -Name Reader

To get the operations for an Azure resource provider that are securable using Azure RBAC.

Get-AzureRmProviderOperation [[-OperationSearchString] <String>]

azure provider operations show (in Azure CLI)

You may also use these commands to verify that an operation string is valid, and to expand wildcard operation strings.

```
Get-AzureRmProviderOperation Microsoft.Compute/virtualMachines/*/action | FT
Operation, OperationName
Get-AzureRmProviderOperation Microsoft.Network/* | FT Operation, OperationName
```

Example: To Assigns the specified RBAC role to the specified principal, at the specified scope.

$resourceGroupName = "DemoRG"

$roleName = "Contributor"

$assigneeName = "existinguser@sandeepsonideccansoft.onmicrosoft.com" #must be verified domain name.

New-AzureRmRoleAssignment -RoleDefinitionName $**roleName** -SignInName $**assigneeName** -ResourceGroupName $**resourceGroupName**

Note: For external user, use **ObjectId** instead of **SignInName**

```
New-AzureRmRoleAssignment -RoleDefinitionName $roleName -ObjectId 45123af4-983e-4e29-
9586-1f6e7aedb02f -ResourceGroupName $resourceGroupName
```

OR

az role assignment create –role $roleName –assignee $assigneeName –resource-group $resourceGroupName

Walkthrough: To Assign

- **Reader role to Subscription**
- **Contributor role to Resource Group**

1. Install the Azure AD Powershell Package and Connect to Azure AD

```
install-module AzureAd (Start PowerShell in Administrator Mode)
import-module azuread
Connect-AzureAD -TenantId sandeepsonideccansoft.onmicrosoft.com
```

2. Create a password that complies with your password complexity requirements.

```
$PasswordProfile = New-Object -TypeName Microsoft.Open.AzureAD.Model.PasswordProfile
$PasswordProfile.Password = "Password@123"
```

3. Create a new user

```
New-AzureADUser -DisplayName "RBAC Tutorial User" -PasswordProfile $PasswordProfile `
 -UserPrincipalName "user1@sandeepsonideccansoft.onmicrosoft.com" -AccountEnabled $true -
MailNickName "rbacuser"
```

Note: Use `New-AzureADMSInvitation for guest user.`

4. Assign the Reader role to the user at the subscription scope

```
$subScope = "/subscriptions/<Subscription GUID as returned by Get-AzureRmSubscription>"
New-AzureRmRoleAssignment -SignInName user1@sandeepsonideccansoft.onmicrosoft.com `
 -RoleDefinitionName "Reader" `
 -Scope $subScope
```

5. Assign the Contributor role to the user at the resource group scope.

```
New-AzureRmRoleAssignment -SignInName user1@sandeepsonideccansoft.onmicrosoft.com `
        -RoleDefinitionName "Contributor" `
        -ResourceGroupName "DemoRG"
```

6. Use the **Get-AzureRMRoleAssignment** command to list the role assignments

```
Get-AzureRMRoleAssignment -ResourceGroupName DemoRG
```

Custom Roles for RBAC

Create custom roles for Azure Role-Based Access Control

The following template shows a custom role for **monitoring and restarting virtual machines**:

d:\VMOperator.json

```
 {
 "Name": "Virtual Machine Operator",
 "Id": "cadb4a5a-4e7a-47be-84db-05cad13b6769",
 "IsCustom": true,
 "Description": "Can monitor and restart virtual machines.",
 "Actions": [
   "Microsoft.Storage/*/read",
   "Microsoft.Network/*/read",
   "Microsoft.Compute/*/read",
   "Microsoft.Compute/virtualMachines/start/action",
   "Microsoft.Compute/virtualMachines/restart/action",
   "Microsoft.Authorization/*/read",
   "Microsoft.Resources/subscriptions/resourceGroups/read",
   "Microsoft.Insights/alertRules/*",
   "Microsoft.Insights/diagnosticSettings/*",
   "Microsoft.Support/*"
 ],
 "NotActions": [


 ],
 "AssignableScopes": [
   "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e",
   "/subscriptions/e91d47c4-76f3-4271-a796-21b4ecfe3624",
   "/subscriptions/34370e90-ac4a-4bf9-821f-85eeedeae1a2"
 ]
}
```

Then you use the New-AzureRmRoleDefinition or az role definition create commands to create the custom role.

```
New-AzureRmRoleDefinition -InputFile C:\Temp\VMOperator.json
```

Using Azure AAD with ARM

Create an Azure Active Directory application

1. Azure Portal → Azure Active Directory → App registrations → New application registration
2. Provide a name and URL for the application. Select **Web app / API** for the type of application you want to create. **You cannot create credentials for a Native application; therefore, that type does not work for an automated application**.
3. Get application ID and authentication key
4. Get tenant id: Azure AD → Properties → Directory ID

Assign Application to role (for a given subscription)

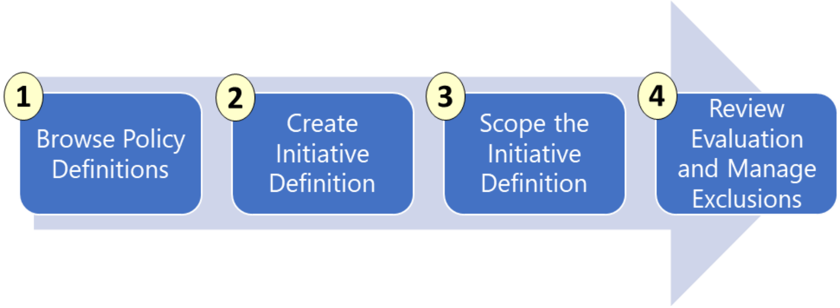
- To access resources in your subscription, you must assign the application to a role.
- You can set the scope at the level of the subscription, resource group, or resource. Permissions are inherited to lower levels of scope. For example, adding an application to the Reader role for a resource group means it can read the resource group and any resources it contains.

1. More Services → Subscription → Select the Subscription → Access Control (IAM)
2. Select + Add → Role = Reader, Select = <The application created above> → Save

Management Subscription Policies

- Azure Policy is a service in Azure that you use to create, assign and manage policy definitions.
- Policy definitions enforce different rules and actions over your resources, so those resources stay compliant with your corporate standards and service level agreements.
- Azure Policy does this by running an evaluation of your resources, scanning for those not compliant with the policy definitions you have. For example, you can have a policy to allow only certain type of virtual machines. Another requires that all resources have a particular tag. These policies are then evaluated when creating and updating resources.

To implement Azure Policies, you can follow these steps.


1
Browse Policy Definitions
2
Create Initiative Definition
3
Scope the Initiative Definition
4
Review Evaluation and Manage Exclusions

About Policy definition:

A Policy Definition expresses what to evaluate and what actions to take. Every policy definition has conditions under which it is enforced. And, it has an accompanying effect that takes place if the conditions are met.

- **Allowed Virtual Machine SKUs**: This policy enables you to specify a set of virtual machine SKUs that your organization can deploy.
- **Allowed Storage Account SKUs**: This policy definition has a set of conditions/rules that determine if a storage account that is being deployed is within a set of SKU sizes. Its action is to deny all servers that do not adhere to the set of defined SKU sizes.
- **Require SQL Server 12.0**: This policy definition has conditions/rules to ensure that all SQL servers use version 12.0. Its action is to deny all servers that do not meet these criteria.
- **Allowed Resource Type**: This policy definition has a set of conditions/rules to specify the resource types that your organization can deploy. Its action is to deny all resources that are not part of this defined list.
- **Allowed Locations**: This policy enables you to restrict the locations that your organization can specify when deploying resources. Its action is used to enforce your geo-compliance requirements.
- **Apply tag and its default value**: This policy applies a required tag and its default value, if it is not specified by the user.
- **Enforce tag and its value**: This policy enforces a required tag and its value to a resource.
- **Not allowed resource types**: This policy enables you to specify the resource types that your organization cannot deploy.

Step1: To view all Policy Definitions

Azure → All Services → Policy → **Definitions** → Filter: Search = Location → Select Allowed locations

Note the Definition (JSON)

Note: If you don't see what you need you can **add a Policy Definition**. The easiest way to do this is to Import a policy from GitHub. New Policy Definitions are added almost every day.

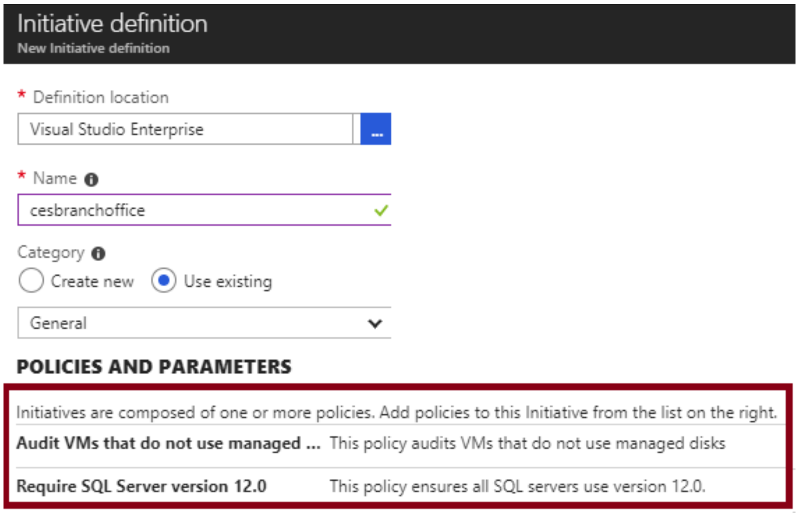
To Assign a Policy Definition to Subscription or Resource Group

Azure → All Services → Policy → Definitions → Select Any Definition → **Assign** → Select Subscription and Optionally Resource Group → Assign

About Initiative definition: An initiative definition is collection of policy definitions that are tailored towards achieving a singular goal.

Step 2: To create Initiative Definition

Azure → All Services → Policy → Definitions → **+Initiative definition**



Initiative definition
New Initiative definition

* Definition location
Visual Studio Enterprise

* Name
cesbranchoffice

Category
○ Create new ● Use existing

General

POLICIES AND PARAMETERS

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

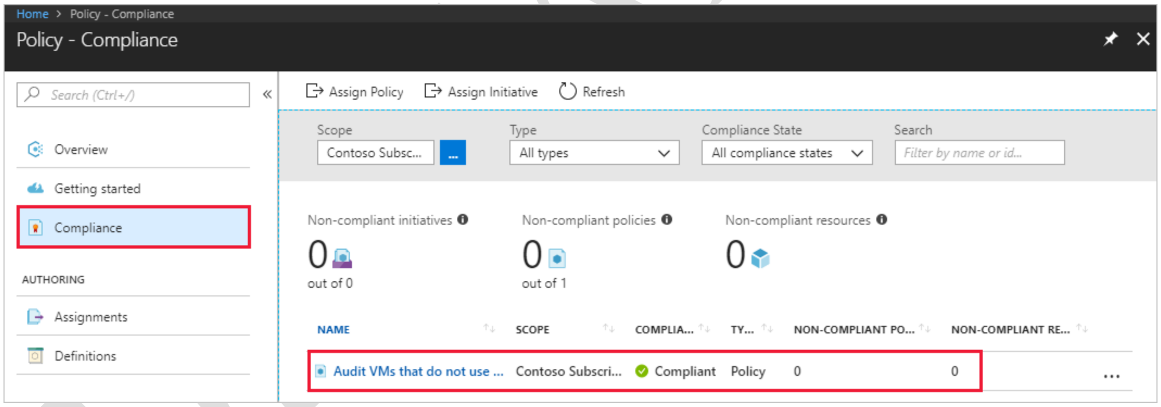
Audit VMs that do not use managed ...	This policy audits VMs that do not use managed disks
Require SQL Server version 12.0	This policy ensures all SQL servers use version 12.0.

To ASSIGN / SCOPE an Initiative Definition to Subscription or Resource Group

Azure → All Services → Policy → Definitions → Select Any Definition → **Assign** → Select Subscription and Optionally Resource Group → Assign

Step 4: Determine Compliance

Once your policy is in place you can use the Compliance blade to review non-compliant initiatives, non-compliant policies, and non-compliant resources.



Home > Policy - Compliance

Policy - Compliance

Search (Ctrl+/)

Overview
Getting started
Compliance
AUTHORING
Assignments
Definitions

Assign Policy | Assign Initiative | Refresh

Scope: Contoso Subsc... | Type: All types | Compliance State: All compliance states | Search: Filter by name or id...

Non-compliant initiatives: 0 out of 0
Non-compliant policies: 0 out of 1
Non-compliant resources: 0

NAME	SCOPE	COMPLIA...	TY...	NON-COMPLIANT PO...	NON-COMPLIANT RE...
Audit VMs that do not use ...	Contoso Subscri...	✓ Compliant	Policy	0	0

✓□ Policy evaluation happens about once an hour, which means that if you make changes to your policy definition and create a policy assignment then it will be re-evaluated over your resources within the hour.

Lock Resources

- As an administrator, you may need to lock a subscription, resource group, or resource to prevent other users in your organization from accidentally deleting or modifying critical resources. You can set the lock level to **CanNotDelete** or **ReadOnly**.
- When you apply a lock at a parent scope, all resources within that scope inherit the same lock.

- Resource changes are restricted, but resource operations are not restricted. For example, a ReadOnly lock on a SQL Database prevents you from deleting or modifying the database, but it does not prevent you from creating, updating, or deleting data in the database.

Applying Locks using Portal

1. Settings blade for the resource, resource group, or subscription → Locks → + Add
2. Lock name = DatabaseServerLock, Lock type = Delete, Notes = "Prevent deleting the database server
3. OK

Template:

The following example shows a template that creates a lock on a storage account. The storage account on which to apply the lock is provided as a parameter.

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "lockedResource": {
      "type": "string"
    }
  },
  "resources": [
    {
      "name": "[concat(parameters('lockedResource'), '/Microsoft.Authorization/myLock')]",
      "type": "Microsoft.Storage/storageAccounts/providers/locks",
      "apiVersion": "2015-01-01",
      "properties": {
        "level": "CannotDelete"
      }
    }
  ]
}
```

Using PowerShell

To lock a resource:

```
New-AzureRmResourceLock -LockLevel CanNotDelete -LockName LockSite `
 -ResourceName examplesite -ResourceType Microsoft.Web/sites `
 -ResourceGroupName exampleresourcegroup
```

To lock a Resource Group

```
New-AzureRmResourceLock -LockName LockGroup -LockLevel CanNotDelete `
 -ResourceGroupName exampleresourcegroup
```

To get all locks in a subscription

```
Get-AzureRmResourceLock
```

To get all locks for a resource:

```
Get-AzureRmResourceLock -ResourceName examplesite -ResourceType Microsoft.Web/sites `
 -ResourceGroupName exampleresourcegroup
```

To get all locks for a resource group

```
Get-AzureRmResourceLock -ResourceGroupName exampleresourcegroup
```

Using Azure CLI:

```
az lock create --name LockSite --lock-type CanNotDelete \
 --resource-group exampleresourcegroup --resource-name examplesite \
 --resource-type Microsoft.Web/sites
```

To get all the locks in your subscription:

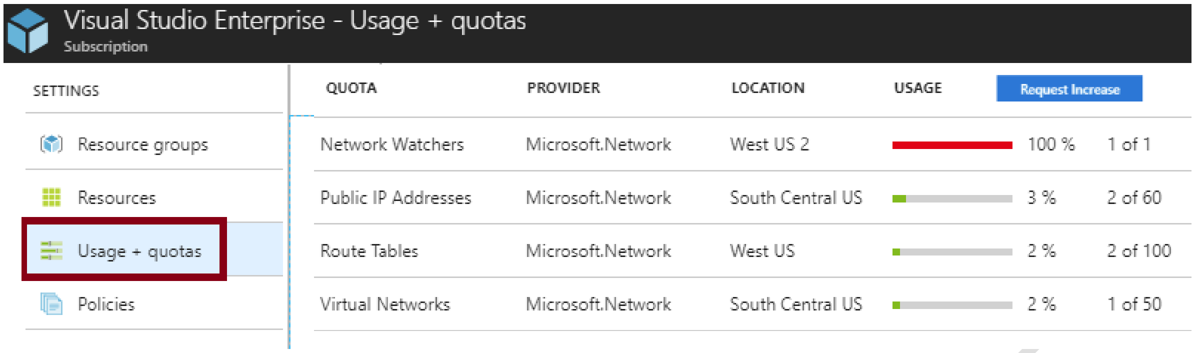
```
az lock list
```

To get all locks for a resource, use:

```
az lock list --resource-group exampleresourcegroup --resource-name examplesite \
 --namespace Microsoft.Web --resource-type sites --parent ""
```

Checking Resources Limits

Azure provides the ability to see the number of each network resource type that you've deployed in your subscription and what your subscription limits are. The ability to view resource usage against limits is helpful to track current usage, and plan for future use. In this example, there are two Public IP Addresses in South Central US and the limit is 60.



Visual Studio Enterprise - Usage + quotas
Subscription

SETTINGS
- Resource groups
- Resources
- Usage + quotas
- Policies

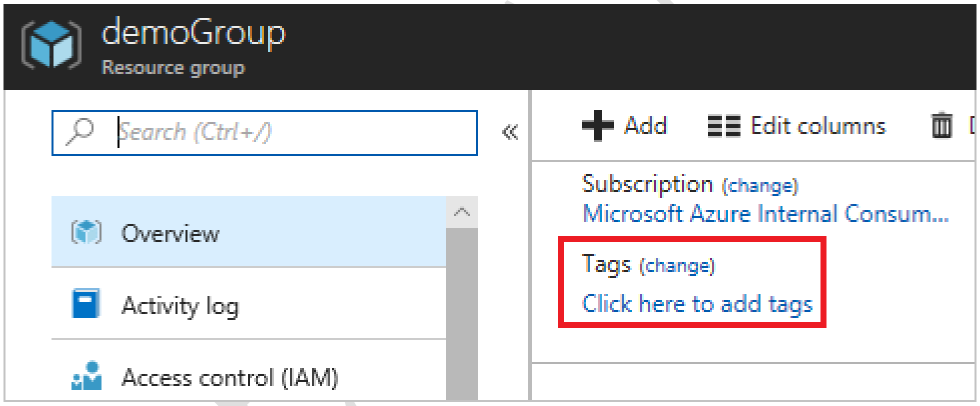
QUOTA	PROVIDER	LOCATION	USAGE	Request Increase
Network Watchers	Microsoft.Network	West US 2	100 %	1 of 1
Public IP Addresses	Microsoft.Network	South Central US	3 %	2 of 60
Route Tables	Microsoft.Network	West US	2 %	2 of 100
Virtual Networks	Microsoft.Network	South Central US	2 %	1 of 50

The limits shown are the limits for your subscription. If you need to increase a default limit, there is a Request Increase link. You will complete and submit the support request. All resources have a maximum limit listed in Azure limits. If your current limit is already at the maximum number, the limit can't be increased.

Resource Tags

You can apply tags to your Azure resources to logically organize them by categories. Each tag consists of a name and a value. For example, you can apply the name "Environment" and the value "Production" or "Development" to your resources. After creating your tags, you associate them with the appropriate resources.

With tags in place, you can retrieve all the resources in your subscription with that tag name and value. This means, you can retrieve related resources from different resource groups.



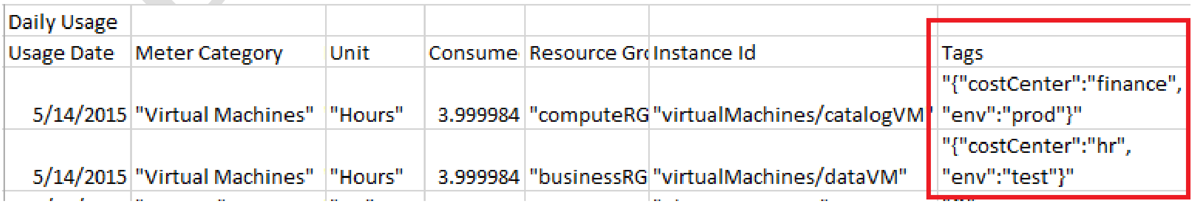
demoGroup
Resource group

Search (Ctrl+/)

Subscription (change)
Microsoft Azure Internal Consum...

Tags (change)
Click here to add tags

Perhaps one of the best uses of tags is to group billing data. When you download the usage CSV for services, the tags appear in the Tags column. For example, you could group virtual machines by cost center and production environment.



Daily Usage						
Usage Date	Meter Category	Unit	Consume	Resource Gr	Instance Id	Tags
5/14/2015	"Virtual Machines"	"Hours"	3.999984	"computeRG	"virtualMachines/catalogVM	"{"costCenter":"finance", "env":"prod"}"
5/14/2015	"Virtual Machines"	"Hours"	3.999984	"businessRG	"virtualMachines/dataVM"	"{"costCenter":"hr", "env":"test"}"

There are a few things to consider about tagging (more at the reference link):

- Each resource or resource group can have a maximum of 15 tag name/value pairs.
- Tags applied to the resource group are not inherited by the resources in that resource group.