Lab: Implementing governance and compliance with Azure initiatives and resource locks

All tasks in this lab are performed from the Azure portal (including a PowerShell Cloud Shell session)

Note: When not using Cloud Shell, the lab virtual machine must have the Azure PowerShell 1.2.0 module (or newer) installed https://docs.microsoft.com/en-us/powershell/azure/install-az-ps

Lab files:

- **Labfiles\Module_11\Governance_and_Compliance\AZ-100.1\az-100-01b_azuredeploy.json**
- **Labfiles\Module_11\Governance_and_Compliance\az-100-01b_azuredeploy.parameters.json**

Scenario

Adatum Corporation wants to use Azure policies and initiatives in order to enforce resource tagging in its Azure subscription. Once the environment is compliant, Adatum wants to prevent unintended changes by implementing resource locks.

Objectives

After completing this lab, you will be able to:

- Implement Azure tags by using Azure policies and initiatives
- Implement Azure resource locks

Exercise 1: Implement Azure tags by using Azure policies and initiatives

The main tasks for this exercise are as follows:

1. Provision Azure resources by using an Azure Resource Manager template.
2. Implement an initiative and policy that evaluate resource tagging compliance.
3. Implement a policy that enforces resource tagging compliance.

4. Evaluate tagging enforcement and tagging compliance.
5. Implement remediation of resource tagging non-compliance.
6. Evaluate effects of the remediation task on compliance.

Task 1: Provision Azure resources by using an Azure Resource Manager template.

1. From the lab virtual machine, start Microsoft Edge, browse to the Azure portal at **http://portal.azure.com** and sign in by using a Microsoft account that has the Owner role in the Azure subscription you intend to use in this lab.
2. In the Azure portal, navigate to the **Create a resource** blade.
3. From the **Create a resource** blade, search Azure Marketplace for **Template deployment**.
4. Use the list of search results to navigate to the **Custom deployment** blade.
5. On the **Custom deployment** blade, select the **Build your own template in the editor**.
6. From the **Edit template** blade, load the template file **az-100-01b_azuredeploy.json**.

> **Note**: Review the content of the template and note that it defines deployment of an Azure VM hosting Windows Server 2016 Datacenter, including tags on some of its resources.

7. Save the template and return to the **Custom deployment** blade.
8. From the **Custom deployment** blade, navigate to the **Edit parameters** blade.
9. From the **Edit parameters** blade, load the parameters file **az-100-01b_azuredeploy.parameters.json**.
10. Save the parameters and return to the **Custom deployment** blade.
11. From the **Custom deployment** blade, initiate a template deployment with the following settings:
    - Subscription: the name of the subscription you are using in this lab
    - Resource group: the name of a new resource group **az1000101b-RG**
    - Location: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs
    - Vm Size: **Standard_DS1_v2**
    - Vm Name: **az1000101b-vm1**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1000101b-vnet1**
    - Environment Name: **lab**

> **Note**: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**
>
> **Note**: Do not wait for the deployment to complete before you proceed to the next step.

12. In the Azure portal, navigate to the **Tags** blade.
13. From the **Tags** blade, display all resources with the **environment** tag set to the value **lab**. Note that only some of the resources deployed in the previous task have this tag assigned.

> **Note**: At this point, only some of the resources have been provisioned, however, you should see at least a few without tags assigned to them.

Task 2: Implement a policy and an initiative that evaluate resource tagging compliance.

1. In the Azure portal, navigate to the **Policy** blade.
2. From the **Policy** blade, navigate to the **Policy - Definitions** blade.
3. From the **Policy Definitions** blade, display the **Require tag and its value** policy definition.
4. From the **Require tag and its default value** policy definition blade, use the duplicate the definition feature to create a new policy with the following settings:
    - Definition location: the name of the subscription you are using in this lab
    - Name: **az10001b - Audit tag and its value**
    - Description: **Audits a required tag and its value. Does not apply to resource groups.**
    - Category: the name of a new category **Lab**
    - Policy rule: in the existing policy rule, change the **effect** from **deny** to **audit**, such that the policy definition has the following content:

CodeCopy

```
{
  "mode": "indexed",
  "policyRule": {
    "if": {
      "not": {
        "field": "[concat('tags[', parameters('tagName'), ']')]",
        "equals": "[parameters('tagValue')]"
      }
    },
    "then": {
```

```
        "effect": "audit"
      }
    },
    "parameters": {
      "tagName": {
        "type": "String",
        "metadata": {
          "displayName": "Tag Name",
          "description": "Name of the tag, such as 'environment'"
        }
      },
      "tagValue": {
        "type": "String",
        "metadata": {
          "displayName": "Tag Value",
          "description": "Value of the tag, such as 'production'"
        }
      }
    }
}
```

5. From the **Policy - Definitions** blade, navigate to the **New Initiative definition** blade.
6. From the **New Initiative definition** blade, create a new initiative definition with the following settings:
    - Definition location: the name of the subscription you are using in this lab
    - Name: **az10001b - Tagging initiative**
    - Description: **Collection of tag policies.**
    - Category: **Lab**
    - AVAILABLE DEFINITIONS: search for and select **az10001b - Audit tag and its value**
        - Tag Name: **environment**
        - Tag Value: **lab**
7. Navigate to the **Policy - Assignments** blade.
8. From the **Policy - Assignments** blade, navigate to the **Assign initiative** blade and create a new initative assignment with the following settings:
    - Scope: the name of the subscription you are using in this lab
    - Exclusions: none
    - Initiative definition: **az10001b - Tagging initiative**
    - Assignment name: **az10001b - Tagging initiative assignment**
    - Description: **Assignment of az10001b - Tagging initiative**
    - Assigned by: the default value
    - Create a Managed Identity: **unchecked**
9. Navigate to the **Policy - Compliance** blade. Note that **COMPLIANCE STATE** is set to either **Not registered** or **Not started**.

> **Note**: On average, it takes about 10 minutes for a compliance scan to start. Rather than waiting for the compliance scan, proceed to the next task. You will review the compliance status later in this exercise.

Task 3: Implement a policy that enforces resource tagging compliance.

1. Navigate to the **Policy - Definitions** blade.
2. From the **Policy - Definitions** blade, navigate to the **az10001b - Tagging initiative** blade.
3. From the **az10001b - Tagging initiative** blade, navigate to its **Edit initiative definition** blade.
4. Add the built-in policy definition named **Require tag and its value** to the initiative and set its parameters to the following values:
    - Tag Name: **environment**
    - Tag Value: **lab**

> **Note**: At this point, your initiative contains two policies. The first of them evaluates the compliance status and the second one enforces tagging during deployment.

Task 4: Evaluate tagging enforcement and tagging compliance.

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **New** blade, search Azure Marketplace for **Template deployment**.
3. Use the list of search results to navigate to the **Custom deployment** blade.
4. On the **Custom deployment** blade, select the **Build your own template in the editor**.
5. From the **Edit template** blade, load the template file **az-100-01b_azuredeploy.json**.

> **Note**: This is the same template that you used for deployment in the first task of this exercise.

6. Save the template and return to the **Custom deployment** blade.
7. From the **Custom deployment** blade, navigate to the **Edit parameters** blade.
8. From the **Edit parameters** blade, load the parameters file **az-100-01b_azuredeploy.parameters.json**.
9. Save the parameters and return to the **Custom deployment** blade.
10. From the **Custom deployment** blade, initiate a template deployment with the following settings:
    - Subscription: the name of the subscription you are using in this lab

- Resource group: the name of a new resource group **az1000102b-RG**
- Location: the name of the Azure region which you chose in the first task of this exercise
- Vm Size: **Standard_DS1_v2**
- Vm Name: **az1000102b-vm1**
- Admin Username: **Student**
- Admin Password: **Pa55w.rd1234**
- Virtual Network Name: **az1000102b-vnet1**
- Environment Name: **lab**

> **Note**: The deployment will fail. This is expected.

11. You will be presented with the message indicating validation erors. Review the error details, indicating that deployment of resource **az1000102b-vnet1** was disallowed by the policy **Require tag and its value** which is included in the **az10001b - Tagging initiative assignment**.
12. Navigate to the **Policy - Compliance** blade. Identify the entry in the **COMPLIANCE STATE** column.
13. Navigate to the **az10001b - Tagging initiative assignment** blade and review the summary of the compliance status.
14. Display the listing of resource compliance and note which resources have been identified as non-compliant.

> **Note**: You might need to click **Refresh** button on the **Policy - Compliance** blade in order to see the update to the compliance status.

Task 5: Implement remediation of resource tagging non-compliance.

1. In the Azure portal, navigate to the **az10001b - Tagging initiative** blade.
2. From the **az10001b - Tagging initiative** blade, navigate to its **Edit initiative definition** blade.
3. Add the built-in policy definition named **Append tag and its default value** to the initiative and set its parameters to the following values:
   - Tag Name: **environment**
   - Tag Value: **lab**
4. Delete the custom policy definition named **az10001b - Audit tag and its value** from the initiative.
5. Delete the built-in policy definition named **Require tag and its value** from the initiative and save the changes.

> **Note**: At this point, your initiative contains a single policy that automatically remediates tagging non-compliance during deployment of new resources and provides evaluation of compliance status.

6. From the Azure Portal, start a PowerShell session in the Cloud Shell.

> **Note**: If this is the first time you are launching the Cloud Shell in the current Azure subscription, you will be asked to create an Azure file share to persist Cloud Shell files. If so, accept the defaults, which will result in creation of a storage account in an automatically generated resource group.

7. In the Cloud Shell pane, run the following commands.

   CodeCopy

```
Get-AzResource -ResourceGroupName 'az1000101b-RG' | ForEach-Object {Set-
AzResource -ResourceId $_.ResourceId -Tag @{environment="lab"} -Force }
```

> **Note**: These commands assign the **environment** tag with the value **lab** to each resource in the resource group **az1000101b-RG**, overwriting any already assigned tags.
>
> **Note**: Wait until the commands successfully complete.

8. In the Azure portal, navigate to the **Tags** blade.
9. From the **Tags** blade, display all resources with the **environment** tag set to the value **lab**. Verify that all resources in the resource group **az1000101b-RG** are listed.

Task 6: Evaluate effects of the remediation task on compliance.

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **New** blade, search Azure Marketplace for **Template deployment**.
3. Use the list of search results to navigate to the **Custom deployment** blade.
4. On the **Custom deployment** blade, select the **Build your own template in the editor**.
5. From the **Edit template** blade, load the template file **az-100-01b_azuredeploy.json**.

> **Note**: This is the same template that you used for deployment in the first task of this exercise.

6. Save the template and return to the **Custom deployment** blade.

7. From the **Custom deployment** blade, navigate to the **Edit parameters** blade.
8. From the **Edit parameters** blade, load the parameters file **az-100-01b_azuredeploy.parameters.json**.
9. Save the parameters and return to the **Custom deployment** blade.
10. From the **Custom deployment** blade, initiate a template deployment with the following settings:
    - Subscription: the name of the subscription you are using in this lab
    - Resource group: **az1000102b-RG**
    - Location: the name of the Azure region which you chose in the first task of this exercise
    - Vm Size: **Standard_DS1_v2**
    - Vm Name: **az1000102b-vm1**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1000102b-vnet1**
    - Environment Name: **lab**

    Note: The deployment will succeed this time. This is expected.

    Note: Do not wait for the deployment to complete before you proceed to the next step.

11. In the Azure portal, navigate to the **Tags** blade.
12. From the **Tags** blade, display all resources with the **environment** tag set to the value **lab**. Note that all the resources deployed to the resource group **az1000102b-RG** have this tag with the same value automatically assigned.

    Note: At this point, only some of the resources have been provisioned, however, you should see that all of them have tags assigned to them.

13. Navigate to the **Policy - Compliance** blade. Identify the entry in the **COMPLIANCE STATE** column.
14. Navigate to the **az10001b - Tagging initiative assignment** blade. Identify the entry in the **COMPLIANCE STATE** column. If the column contains the **Not started** entry, wait until it the compliance scan runs.

    Note: You might need to wait for up to 10 minutes and click **Refresh** button on the **Policy - Compliance** blade in order to see the update to the compliance status.

    Note: Do not wait until the status is listed as compliant but instead proceed to the next exercise.

Result: After you completed this exercise, you have implemented an initiative and policies that evaluate, enforce, and remediate resource tagging compliance. You also evaluated the effects of policy assignment.

Exercise 2: Implement Azure resource locks

The main tasks for this exercise are as follows:

1. Create resource group-level locks to prevent accidental changes
2. Validate functionality of the resource group-level locks

Task 1: Create resource group-level locks to prevent accidental changes

1. In the Azure portal, navigate to the **az1000101b-RG** resource group blade.
2. From the **az1000101b-RG** resource group blade, display the **az1000101b-RG - Locks** blade.
3. From the **az1000101b-RG - Locks** blade, add a lock with the following settings:
    - Lock name: **az1000101b-roLock**
    - Lock type: **Read-only**

Task 2: Validate functionality of the resource group-level locks

1. In the Azure portal, navigate to the **az1000102b-vm1** virtual machine blade.
2. From the **az1000102b-vm1** virtual machine blade, navigate to the **az1000102b-vm1 - Tags** blade.
3. Try setting the value of the **environment** tag to **dev**. Note that the operation is successful.
4. In the Azure portal, navigate to the **az1000101b-vm1** virtual machine blade.
5. From the **az1000101b-vm1** virtual machine blade, navigate to the **az1000101b-vm1 - Tags** blade.
6. Try setting the value of the **environment** tag to **dev**. Note that this time the operation fails. The resulting error message indicates that the resource refused tag assignment, with resource lock being the likely reason.
7. Navigate to the blade of the storage account created in the **az1000101b-RG** resource group.
8. From the storage account blade, navigate to its **Access keys** blade. Note the resulting error message stating that you cannot access the data plane because a read lock on the resource or its parent.
9. In the Azure portal, navigate to the **az1000101b-RG** resource group blade.
10. From the **az1000101b-RG** resource group blade, navigate to its **Tags** blade.
11. From the **Tags** blade, attempt assigning the **environment** tag with the value **lab** to the resource group and note the error message.

Result: After you completed this exercise, you have created a resource group-level lock to prevent accidental changes and validated its functionality.

Exercise 3: Remove lab resources

Task 1: Delete the resource group-level lock.

1. In the Azure portal, navigate to the **az1000101b-RG** resource group blade.
2. In the Azure portal, navigate to the **az1000101b-RG** resource group blade.
3. From the **az1000101b-RG** resource group blade, display the **az1000101b-RG - Locks** blade.
4. On the **az1000101b-RG - Locks** blade, delete the **az1000101b-roLock**.

Task 2: Delete the policy assignment and definition.

1. In the Azure portal, navigate to the **Policy** blade.
2. From the **Policy**, blade navigate to the **Policy - Assignments** blade.
3. From the **Policy - assignments** blade, remove the assignment you created earlier in this lab.
4. From the **Policy**, blade navigate to the **Policy - Definitions** blade.
5. From the **Policy - Definitions** blade, delete all definitions you created earlier in this lab.

Task 3: Open Cloud Shell

1. At the top of the portal, click the **Cloud Shell** icon to open the Cloud Shell pane.
2. At the Cloud Shell interface, select **Bash**.
3. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to list all resource groups you created in this lab:

   ShellCopy

   ```
   az group list --query "[?starts_with(name,'az100010')].name" --output tsv
   ```

4. Verify that the output contains only the resource groups you created in this lab. These groups will be deleted in the next task.

Task 4: Delete resource groups

1. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to delete the resource groups you created in this lab

   ShellCopy

```
az group list --query "[?starts_with(name,'az100010')].name" --output tsv |
xargs -L1 bash -c 'az group delete --name $0 --no-wait --yes'
```

2. Close the **Cloud Shell** prompt at the bottom of the portal.