Contents

Note the changes!!!

The AZ-100 and AZ-101 certifications have been replaced by a new AZ-103 Microsoft Azure Administrator exam! You can read more about this announcement on Liberty Munson's blog at https://www.microsoft.com/en-

us/learning/community-blog-post.aspx?BlogId=8&Id=375217

To support the new exam there is a new AZ-103 GitHub repository, available since May 3 2019. At that time, all the AZ-100 and AZ-101 labs in their respective repositories have been moved to the AZ-103 repository. Those labs are being reused in AZ-103 and we will be maintaining only one repository. The AZ-100 and AZ-101 lab numbering system has be retained, so if you are still teaching the AZ-100 or AZ-101 courses you will be able to easily identify the labs. You will also be able to get the latest version of the labs, and submit any issues you find.

What are we doing?

- We are publishing the lab instructions and lab files on GitHub to allow for interaction between the course authors and MCTs. We hope this will help keep the content current as the Azure platform changes.
- This is a GitHub repository for the AZ-100, Microsoft Azure Infrastructure Deployment course.
- You can access the repositories from https://github.com/orgs/MicrosoftLearning/dashboard
- Within each repository there are lab guides in the Markdown format in the Instructions folder. The lab guides in the PDF format are available from the MCT Download Center, however they are not being regularly updated. If appropriate, there are also additional files that are needed to complete the lab within the Allfiles\Labfiles folder. Not every course has corresponding lab files.
- For each delivery, trainers should download the latest files from GitHub. Trainers should also check the Issues tab to see if other MCTs have reported any errors.
- Lab timing estimates are provided but trainers should check to ensure this is accurate based on the audience.
- The lab content has been placed at the end of each course for consistency and convenience. However, as the instructor, you are the best judge to determine when the lab should be offered.
- To conduct you will need an internet connection and an Azure subscription. Please read the Instructor Prep Guide for more information on using the Cloud Shell.
- It is recommended that you provide these materials directly to your students rather than point them to the GitHub repository.

How are we doing?

- If as you are teaching these courses, you identify areas for improvement, please use the Issues tab to provide feedback. We will periodically create new files to incorporate the changes.

General comments regarding the AZ-100 and AZ-101 courses

- PowerShell scripts in all labs use the current version of Azure PowerShell Az module
- Although not required, it is a good idea to deprovision any existing resources when you have completed each lab. This will help mitigate the risk of exceeding the default vCPU quota limits and minimize usage charges.
- Availability of Azure regions and resources in these regions depends to some extent on the type of subscription you are using. To identify Azure regions available in your subscription, refer to https://azure.microsoft.com/en-us/regions/offers/ . To identify resources available in these regions, refer to https://azure.microsoft.com/en-us/global-infrastructure/services/ . These restrictions might result in failures during template validation or template deployment, in particular when provisioning Azure VMs. If this happens, review error messages and retry deployment with a different VM size or a different region.
- When launching Azure Cloud Shell for the first time, you will likely be prompted to create an Azure file share to persist Cloud Shell files. If so, you can typically accept the defaults, which will result in creation of a storage account in an automatically generated resource group. Note that this might happen again if you delete that storage account.
- Before you perform a template based deployments, you might need to register providers that handle provisioning of resource types referenced in the template. This is a one-time operation (per subscription) required when using Azure Resource Manager templates to deploy resources managed by these resource providers (if these resource providers have not been yet registered). You can perform registration from the subscription's Resource Providers blade in the Azure portal or by using Cloud Shell to run Register-AzResourceProvider PowerShell cmdlet or az provider Azure CLI command.

We hope using this GitHub repository brings a sense of collaboration to the labs and improves the overall quality of the lab experience.

Regards, Azure Administrator Courseware Team

AZ 100 Module 1 - Manage Azure Subscriptions and Resources

1 Lab 2: Manage Azure Subscriptions and Resources - Implementing governance and compliance with Azure initiatives and resource locks

Estimated Time: 90 minutes

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?view=azps-1.2.0

Lab files:

- **Allfiles/Labfiles/AZ-100.1/az-100-01b_azuredeploy.json**
- **Allfiles/Labfiles/AZ-100.1/az-100-01b_azuredeploy.parameters.json**

1.0.1 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.

1.0.2 Objectives

After completing this lab, you will be able to:

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

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

Estimated Time: 45 minutes

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.

1.1.0.1 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 **New** blade.
3. From the **New** 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.

1.1.0.2 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 **Enforce tag and its value** policy definition.
4. From the **Enforce 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: the existing policy rule with the **effect** set to **audit**, such that the policy definition has the following content:

```
{
  "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**
   - POLICIES AND PARAMETERS: **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.

1.1.0.3 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 **Enforce 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.

1.1.0.4 Task 4: Evaluate tagging enforcement and tagging compliance.

1. In the Azure portal, navigate to the **New** 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 **Enforce 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 reviwew 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.

1.1.0.5 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 **Apply 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 **Enforce 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.

   ```
   Get-AzResource -ResourceGroupName 'az1000101b-RG' | ForEach-Object {Set-AzResource -ResourceId $_.
   ```

   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.

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

1. In the Azure portal, navigate to the **New** 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.

1.2 Exercise 2: Implement Azure resource locks

Estimated Time: 15 minutes

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

1.2.0.1 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**

1.2.0.2 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 - Locks** 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. # AZ 100 Module 1 - Manage Azure Subscriptions and Resources

2 Lab: Manage Azure Subscriptions and Resources

Estimated Time: 30 minutes

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?view=azps-1.2.0

Lab files: none

2.0.1 Scenario

Adatum Corporation wants to use Azure Role Based Access Control and Azure Policy to control provisioning and management of their Azure resources. It also wants to be able to automate and track provisioning and management tasks.

2.0.2 Objectives

After completing this lab, you will be able to:

- Configure delegation of provisioning and management of Azure resources by using built-in Role-Based Access Control (RBAC) roles and built-in Azure policies
- Verify delegation by provisioning Azure resources as a delegated admin and auditing provisioning events

2.1 Exercise 1: Configure delegation of provisioning and management of Azure resources by using built-in Role-Based Access Control (RBAC) roles and built-in Azure policies

Estimated Time: 15 minutes

The main tasks for this exercise are as follows:

1. Create Azure Active Directory (AD) users and groups
2. Create Azure resource groups
3. Delegate management of an Azure resource group via a built-in RBAC role
4. Assign a built-in Azure policy to an Azure resource group

2.1.0.1 Task 1: Create Azure AD users and groups

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 and is a Global Administrator of the Azure AD tenant associated with that subscription.

2. In the Azure portal, navigate to the **Azure Active Directory** blade
3. From the **Azure Active Directory** blade, navigate to the **Custom domain names** blade and identify the primary DNS domain name associated the Azure AD tenant. Note its value - you will need it later in this task.
4. From the Azure AD **Custom domain names** blade, navigate to the **Users - All users** blade.
5. From the **Users - All users** blade, create a new user with the following settings:
   - Name: **aaduser100011**
   - User name: **aaduser100011@** where represents the primary DNS domain name you identified earlier in this task.
   - Profile: **Not configured**
   - Properties: **Default**
   - Groups: **0 groups selected**
   - Directory role: **User**
   - Password: select the checkbox **Show Password** and note the string appearing in the **Password** text box. You will need it later in this lab.
6. From the **Users - All users** blade, navigate to the **Groups - All groups** blade.
7. From the **Groups - All groups** blade, create a new group with the following settings:
   - Group type: **Security**
   - Group name: **az1001 Contributors**
   - Group description: **az1001 Contributors**
   - Membership type: **Assigned**
   - Members: **aaduser100011**

2.1.0.2 Task 2: Create Azure resource groups

1. In the Azure portal, navigate to the **Resource groups** blade.
2. From the **Resource groups** blade, create the first resource group with the following settings:
   - Resource group name: **az1000101-RG**
   - Subscription: the name of the subscription you are using in this lab
   - Resource group location: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs.

     Note: To identify Azure regions available in your subscription, refer to **https://azure.microsoft.com/en-us/regions/offers/**
3. From the **Resource groups** blade, create the second resource group with the following settings:
   - Resource group name: **az1000102-RG**
   - Subscription: the name of the subscription you selected in the previous step
   - Resource group location: the name of the Azure region you selected in the previous step

2.1.0.3 Task 3: Delegate management of an Azure resource group via a built-in RBAC role

1. In the Azure portal, from the **Resource groups** blade, navigate to the **az1000101-RG** blade.
2. From the **az1000101-RG** blade, display its **Access control (IAM)** blade.
3. From the **az1000101-RG - Access control (IAM)** blade, display the **Role assignments** blade.
4. From the **Role assignments** blade, create the following **role assignment**:
   - Role: **Contributor**
   - Assign access to: **Azure AD user, group, or service principal**

- Select: **az1001 Contributors**

2.1.0.4 Task 4: Assign a built-in Azure policy to an Azure resource group

1. From the **az1000101-RG** blade, display its **Policies** blade.
2. From the **Policy - Compliance** blade, display the **Assign policy** blade.
3. Assign the policy with the following settings:
   - Scope: **az1000101-RG**
   - Exclusions: leave the entry blank
   - Policy definition: **Allowed virtual machine SKUs**
   - Assignment name: **Allowed virtual machine SKUs**
   - Description: **Allowed selected virtual machine SKUs (Standard_DS1_v2)**
   - Assigned by: leave the entry set to its default value
   - Allowed SKUs: **Standard_DS1_v2**
   - Create a Managed Identity: leave the entry blank

   Result: After you completed this exercise, you have created an Azure AD user and an Azure AD group, created two Azure resource groups, delegated management of the first Azure resource group via the built-in Azure VM Contributor RBAC role, and assigned to the same resource group the built-in Azure policy restricting SKUs that can be used for Azure VMs.

2.2 Exercise 2: Verify delegation by provisioning Azure resources as a delegated admin and auditing provisioning events

Estimated Time: 15 minutes

The main tasks for this exercise are as follows:

1. Identify an available DNS name for an Azure VM deployment
2. Attempt an automated deployment of a policy non-compliant Azure VM as a delegated admin
3. Perform an automated deployment of a policy compliant Azure VM as a delegated admin
4. Review Azure Activity Log events corresponding to Azure VM deployments

2.2.0.1 Task 1: Identify an available DNS name for an Azure VM deployment

1. 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.

2. In the Cloud Shell pane, run the following command, substituting the placeholder `<custom-label>` with any string which is likely to be unique and the placeholder `<location-of-az1000101-RG>` with the name of the Azure region in which you created the **az1000101-RG** resource group.

   ```
   Test-AzDnsAvailability -DomainNameLabel <custom-label> -Location '<location-of-az1000101-RG>'
   ```

3. Verify that the command returned **True**. If not, rerun the same command with a different value of the `<custom-label>` until the command returns **True**.
4. Note the value of the `<custom-label>` that resulted in the successful outcome. You will need it in the next task
5. Run these commands:

   ```
   Register-AzResourceProvider -ProviderNamespace Microsoft.Network
   ```

   ```
   Register-AzResourceProvider -ProviderNamespace Microsoft.Compute
   ```

Note: These cmdlets register the Azure Resource Manager Microsoft.Network and Microsoft.Compute resource providers. This is a one-time operation (per subscription) required when using Azure Resource Manager templates to deploy resources managed by these resource providers (if these resource providers have not been yet registered).

2.2.0.2 Task 2: Attempt an automated deployment of a policy non-compliant Azure VM as a delegated admin

1. Launch another browser window in the Private mode.
2. In the new browser window, navigate to the Azure portal and sign in using the user account you created in the previous exercise. When prompted, change the password to a new value.
3. In the Azure portal, navigate to the **Resource groups** blade and note that you can view only the resource group **az1000101-RG**.
4. In the Azure portal, navigate to the **Create a resource** blade.
5. From the **Create a resource** blade, search Azure Marketplace for **Template deployment**.
6. Use the list of search results to navigate to the **Deploy a custom template** blade.
7. On the **Custom deployment** blade, in the **Load a GitHub quickstart template** drop-down list, select the **101-vm-simple-linux** entry and navigate to the **Edit template** blade.
8. On the **Edit template** blade, navigate to the **Variables** section and locate the **vmSize** entry.
9. Note that the template is using hard-coded **Standard_A1** VM size.
10. Discard any changes you might have made to the template and navigate to the **Deploy a simple Ubuntu Linux VM** blade.
11. From the **Deploy a simple Ubuntu Linux VM** blade, initiate a template deployment with the following settings:
    - Subscription: the same subscription you selected in the previous exercise
    - Resource group: **az1000101-RG**
    - Location: the name of the Azure region which you selected in the previous exercise
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Dns Label Prefix: the `<custom-label>` you identified in the previous task
    - Ubuntu OS Version: accept the default value
    - Location: accept the default value
12. Note that the intiation of the deployment fails. Navigate to the **Errors** blade and note that the deployment of the resource is not allowed by the policy **Allowed virtual machine SKUs**.

2.2.0.3 Task 3: Perform an automated deployment of a policy compliant Azure VM as a delegated admin

1. From the **Deploy a simple Ubuntu Linux VM** blade, navigate to the **Edit template** blade.
2. On the **Edit template** blade, navigate back to the **Variables** section and locate the **vmSize** entry.
3. Replace the value **Standard_A1** with **Standard_DS1_v2** and save the change.
4. Initiate a deployment again. Note that this time validation is successful.
5. Do not wait for the deployment to complete but proceed to the next task.

2.2.0.4 Task 4: Review Azure Activity Log events corresponding to Azure VM deployments

1. Switch to the browser window that you used in the previous exercise.
2. In the Azure portal, navigate to the **az1000101-RG** resource group blade.
3. From the **az1000101-RG** resource group blade, display its **Activity log** blade.

4. In the list of operations, note the ones corresponding to the failed and successful validation events.
5. Refresh the view of the blade and observe events corresponding to the Azure VM provisioning, including the final one representing the successful deployment.

   Result: After you completed this exercise, you have identified an available DNS name for an Azure VM deployment, attempted an automated deployment of a policy non-compliant Azure VM as a delegated admin, performed an automated deployment of a policy compliant Azure VM as the same delegated admin, and reviewed Azure Activity Log entries corresponding to both Azure VM deployments. # AZ 100 Module 2 - Implement and Manage Storage

3 Lab 2: Implement Azure File Sync

Estimated Time: 90 minutes

All tasks in this lab are performed from the Azure portal, except for steps in Exercise 1 and Exercise 2 performed within a Remote Desktop session to an Azure VM.

Lab files:

- **Allfiles/Labfiles/AZ-100.2/az-100-02b_azuredeploy.json**
- **Allfiles/Labfiles/AZ-100.2/az-100-02b_azuredeploy.parameters.json**

3.0.1 Scenario

Adatum Corporation hosts its file shares in on-premises file servers. Considering its plans to migrate majority of its workloads to Azure, Adatum is looking for the most efficient method to replicate its data to file shares that will be available in Azure. To implement it, Adatum will use Azure File Sync.

3.0.2 Objectives

After completing this lab, you will be able to:

- Deploy an Azure VM by using an Azure Resource Manager template
- Prepare Azure File Sync infrastructure
- Implement and validate Azure File Sync

3.0.3 Exercise 0: Prepare the lab environment

Estimated Time: 10 minutes

The main tasks for this exercise are as follows:

1. Deploy an Azure VM by using an Azure Resource Manager template

3.0.3.1 Task 1: Deploy an Azure VM 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 **New** blade.
3. From the **New** 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-100-02b_azuredeploy.json**.

   Note: Review the content of the template and note that it defines deployment of an Azure VM hosting Windows Server 2016 Datacenter with a single data disk.
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-100-02b_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 **az1000201b-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: **az1000201b-vm1**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1000201b-vnet1**

      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 but proceed to the next exercise. You will use the virtual machine included in this deployment in the next exercise of this lab.

      Note: Keep in mind that the purpose of Azure VM **az1000201b-vm1** is to emulate an on-premises file server in our scenario.

Result: After you completed this exercise, you have initiated a template deployment of an Azure VM **az1000201b-vm1** that you will use in the next exercise of this lab.

3.1 Exercise 1: Prepare Azure File Sync infrastructure

Estimated Time: 35 minutes

The main tasks for this exercise are as follows:

1. Create an Azure Storage account and a file share
2. Prepare Windows Server 2016 for use with Azure File Sync
3. Run Azure File Sync evaluation tool

3.1.0.1 Task 1: Create an Azure Storage account and a file share

1. In the Azure portal, navigate to the **New** blade.
2. From the **New** blade, search Azure Marketplace for **Storage account - blob, file, table, queue**.
3. Use the list of search results to navigate to the **Create storage account** blade.
4. From the **Create storage account** blade, create a new storage account with the following settings:
    - Subscription: the same subscription you selected in the previous task
    - Resource group: the name of a new resource group **az1000202b-RG**
    - Storage account name: any valid, unique name between 3 and 24 characters consisting of lowercase letters and digits
    - Location: the name of the Azure region which you selected in the previous task
    - Performance: **Standard**
    - Account kind: **Storage (general purpose v1)**
    - Replication: **Locally-redundant storage (LRS)**
    - Secure transfer required: **Disabled**
    - Allow access from: **All networks**
    - Hierarchical namespace: **Disabled**

Note: Wait for the storage account to be provisioned but proceed to the next step.

5. In the Azure portal, navigate to the blade representing the newly provisioned storage account.
6. From the storage account blade, display the properties of its File Service.
7. From the storage account **Files** blade, create a new file share with the following settings:
   - Name: **az10002bshare1**
   - Quota: none

3.1.0.2 Task 2: Prepare Windows Server 2016 for use with Azure File Sync

Note: Before you start this task, ensure that the template deployment you started in Exercise 0 has completed.

1. In the Azure portal, navigate to the **az1000201b-vm1** blade.
2. From the **az1000201b-vm1** blade, connect to the Azure VM via the RDP protocol and, when prompted to sign in, provide the following credentials:
   - Admin Username: **Student**
   - Admin Password: **Pa55w.rd1234**
3. Within the RDP session to the Azure VM, in Server Manager, navigate to **File and Storage Services**, locate the data disk attached to the Azure VM, initialize it as a **GPT** disk, and use **New Volume Wizard** to create a single volume occupying entire disk with the following settings:
   - Drive letter: **S**
   - File system: **NTFS**
   - Allocation unit size: **Default**
   - Volume label: **Data**
4. Within the RDP session, start a Windows PowerShell session as administrator.
5. From the Windows PowerShell console, set up a file share by running the following:

```
$directory = New-Item -Type Directory -Path 'S:\az10002bShare'

New-SmbShare -Name $directory.Name -Path $directory.FullName -FullAccess 'Administrators' -ReadAcc

Copy-Item -Path 'C:\WindowsAzure\*' -Destination $directory.FullName -Recurse
```

   Note: To populate the file share with sample data, we use content of the C:\WindowsAzure folder, which should contain about 100 MB worth of files

6. From the Windows PowerShell console, install the latest AzureRM module by running the following:

```
Install-Module -Name AzureRM
```

   Note: When prompted, confirm that you want to proceed with the installation from PSGallery repository.

3.1.0.3 Task 3: Run Azure File Sync evaluation tool

1. Within the RDP session to the Azure VM, from the Windows PowerShell console, install the latest version of Package Management and PowerShellGet by running the following :

```
Install-Module -Name PackageManagement -Repository PSGallery -Force

Install-Module -Name PowerShellGet -Repository PSGallery -Force
```

   Note: When prompted, confirm that you want to proceed with the installation of the NuGet provider.

2. Restart the PowerShell session.
3. From the Windows PowerShell console, install the Azure File Sync PowerShell module by running the following:

```
Install-Module -Name Az.StorageSync -AllowPrerelease -AllowClobber -Force
```

4. From the Windows PowerShell console, install the Azure File Sync PowerShell module by running the following:

   ```
   Invoke-AzStorageSyncCompatibilityCheck -Path 'S:\az10002bShare'
   ```

5. Review the results and verify that no compatibility issues have been found.

   Result: After you completed this exercise, you have created an Azure Storage account and a file share, prepare Windows Server 2016 for use with Azure File Sync, and run Azure File Sync evaluation tool

3.2 Exercise 2: Prepare Azure File Sync infrastructure

Estimated Time: 45 minutes

The main tasks for this exercise are as follows:

1. Deploy the Storage Sync Service
2. Install the Azure File Sync Agent
3. Register the Windows Server with Storage Sync Service
4. Create sync groups and a cloud endpoint
5. Create a server endpoint
6. Validate Azure File Sync operations

3.2.0.1 Task 1: Deploy the Storage Sync Service

1. Within the RDP session to the Azure VM, in Server Manager, navigate to the Local Server view and turn off temporarily **IE Enhanced Security Configuration**.
2. Within the RDP session to the Azure VM, start Internet Explorer, browse to the Azure portal at **http://portal.azure.com** and sign in by using the same Microsoft account you used previously in this lab.
3. In the Azure portal, navigate to the **New** blade.
4. From the **New** blade, search Azure Marketplace for **Azure File Sync**.
5. Use the list of search results to navigate to the **Deploy Storage Sync** blade.
6. From the **Deploy Storage Sync** blade, create a Storage Sync Service with the following settings:
   - Name: **az1000202b-ss**
   - Subscription: the same subscription you selected in the previous task
   - Resource group: the name of a new resource group **az1000203b-RG**
   - Location: the name of the Azure region in which you created the storage account earlier in this exercise

3.2.0.2 Task 2: Install the Azure File Sync Agent.

1. Within the RDP session, start another instance of Internet Explorer, browse to Microsoft Download Center at **https://go.microsoft.com/fwlink/?linkid=858257** and download the Azure File Sync Agent Windows Installer file **StorageSyncAgent_V5_WS2016.msi**.
2. Once the download completes, run the Storage Sync Agent Setup wizard with the default settings to install Azure File Sync Agent.
3. After the Azure File Sync agent installation completes, the **Azure File Sync - Server Registration** wizard will automatically start.

3.2.0.3 Task 3: Register the Windows Server with Storage Sync Service

1. From the initial page of the **Azure File Sync - Server Registration** wizard, sign in by using the same Microsoft account you used previously in this lab.
2. On the **Choose a Storage Sync Service** page of the **Azure File Sync - Server Registration** wizard, specify the following settings to register:
   - Azure Subscription: the name of the subscription you are using in this lab
   - Resource group: **az1000203b-RG**
   - Storage Sync Service: **az1000202b-ss**
3. When prompted, sign in again by using the same Microsoft account you used previously in this lab.

3.2.0.4 Task 4: Create a sync group and a cloud endpoint

1. Within the RDP session to the Azure VM, in the Azure portal, navigate to the **az1000202b-ss** Storage Sync Service blade.
2. From the **az1000202b-ss** Storage Sync Service blade, navigate to the **Sync group** blade and create a new sync group with the following settings:
   - Sync group name: **az1000202b-syncgroup1**
   - Azure Subscription: the name of the subscription you are using in this lab
   - Storage account: the resource id of the storage account you created in the previous exercise
   - Azure File Share: **az10002bshare1**

3.2.0.5 Task 5: Create a server endpoint

1. Within the RDP session to the Azure VM, in the Azure portal, from the **az1000202b-ss** Storage Sync Service blade, navigate to the **az1000202b-syncgroup1** blade.
2. From the **az1000202b-syncgroup1** blade, navigate to the **Add server endpoint** blade and create a new server endpoint with the following settings:
   - Registered server: **az1000201b-vm1**
   - Path: **S:\az10002bShare**
   - Cloud Tiering: **Enabled**
     - Always preserve the specified percentage of free space on the volume: **15**
     - Only cache files that were accessed or modified within the specified number of days: **30**
   - Offline Data Transfer: **Disabled**

3.2.0.6 Task 6: Validate Azure File Sync operations

1. Within the RDP session to the Azure VM, in the Azure portal, monitor the health status of the server endpoint **az100021b-vm1** on the **az1000202b-syncgroup1** blade, as it changes from **Provisioning** to **Pending** and, eventually, to a green checkmark.

   Note: You should be able to proceed to the next step after a few minutes.
2. In the Azure portal, navigate to the **az10002bshare1** blade and display the **Connect** blade.
3. From the **Connect** blade, copy into Clipboard the PowerShell commands that connect to the file share from a Windows computer.
4. Within the RDP session, start a Windows PowerShell ISE session.
5. From the Windows PowerShell ISE session, open the script pane and paste into it the content of your local Clipboard.
6. Execute the script and verify that its output confirms successful mapping of the Z: drive to the Azure Storage File Service share.

7. Within the RDP session, start File Explorer, navigate to the Z: drive, and verify that it contains the same content as S:\az10002bShare
8. Display the Properties window of individual folders on the Z: drive, review the Security tab, and note that the entries represent NTFS permissions assigned to the corresponding folders on the S: drive.

   Result: After you completed this exercise, you have deployed the Storage Sync Service, installed the Azure File Sync Agent, registered the Windows Server with Storage Sync Service, created a sync group and a cloud endpoint, created a server endpoint, and validated Azure File Sync operations.
   # AZ 100 Module 2 - Implement and Manage Storage

4 Lab: Implement and Manage Storage

Estimated Time: 30 minutes

All tasks in this lab are performed from the Azure portal (including a PowerShell Cloud Shell session) except for Excercise 2 Task 2, which includes steps performed from a Remote Desktop session to an Azure VM

> **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?view=azps-1.2.0

Lab files:

- **Labfiles\AZ100\Mod02\az-100-02_azuredeploy.json**
- **Labfiles\AZ100\Mod02\az-100-02_azuredeploy.parameters.json**

4.0.1 Scenario

Adatum Corporation wants to leverage Azure Storage for hosting its data

4.0.2 Objectives

After completing this lab, you will be able to:

- Deploy an Azure VM by using an Azure Resource Manager template
- Implement and use Azure Blob Storage
- Implement and use Azure File Storage

4.0.3 Exercise 0: Prepare the lab environment

Estimated Time: 5 minutes

The main tasks for this exercise are as follows:

1. Deploy an Azure VM by using an Azure Resource Manager template

4.0.3.1 Task 1: Deploy an Azure VM 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 **Subscriptions** blade.
3. From the **Subscriptions** blade, navigate to the blade displaying properties of your Azure subscription.
4. From the blade displaying the properties of your subscription, navigate to its **Resource providers** blade.
5. On the **Resource providers** blade, register the following resource providers (if these resource providers have not been yet registered):

- Microsoft.Network
- Microsoft.Compute
- Microsoft.Storage

Note: This step registers the Azure Resource Manager Microsoft.Network, Microsoft.Compute, and Microsoft.Storage resource providers. This is a one-time operation (per subscription) required when using Azure Resource Manager templates to deploy resources managed by these resource providers (if these resource providers have not been yet registered).

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Template deployment**.
3. Use the list of search results to navigate to the **Deploy a custom template** 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 **Labfiles\AZ100\Mod02\az-100-02_azuredeploy.json**.

   Note: Review the content of the template and note that it defines deployment of an Azure VM hosting Windows Server 2016 Datacenter.
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 **Labfiles\AZ100\Mod02\az-100-02_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 **az1000201-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: **az1000201-vm1**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1000201-vnet1**

    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 but proceed to the next exercise. You will use the virtual machine **az1000201-vm1** in the second exercise of this lab.

Result: After you completed this exercise, you have initiated template deployment of an Azure VM **az1000201-vm1** that you will use in the second exercise of this lab.

4.1 Exercise 1: Implement and use Azure Blob Storage

Estimated Time: 15 minutes

The main tasks for this exercise are as follows:

1. Create Azure Storage accounts
2. Review configuration settings of Azure Storage accounts
3. Manage Azure Storage Blob Service
4. Copy a container and blobs between Azure Storage accounts
5. Use a Shared Access Signature (SAS) key to access a blob

4.1.0.1 Task 1: Create Azure Storage accounts

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Storage account**.
3. Use the list of search results to navigate to the **Create storage account** blade.
4. From the **Create storage account** blade, create a new storage account with the following settings:
   - Subscription: the same subscription you selected in the previous task
   - Resource group: the name of a new resource group **az1000202-RG**
   - Storage account name: any valid, unique name between 3 and 24 characters consisting of lowercase letters and digits
   - Location: the name of the Azure region which you selected in the previous task
   - Performance: **Standard**
   - Account kind: **Storage (general purpose v1)**
   - Replication: **Locally-redundant storage (LRS)**
   - Secure transfer required: **Disabled**
   - Allow access from: **All networks**
   - **Data Lake Storage Gen2** Hierarchical namespace: **Disabled**
5. Do not wait for the storage account to be provisioned but proceed to the next step.
6. In the Azure portal, navigate to the **Create a resource** blade.
7. From the **Create a resource** blade, search Azure Marketplace for **Storage account**.
8. Use the list of search results to navigate to the **Create storage account** blade.
9. From the **Create storage account** blade, create a new storage account with the following settings:
   - Subscription: the same subscription you selected in the previous task
   - Resource group: the name of a new resource group **az1000203-RG**
   - Storage account name: any valid, unique name between 3 and 24 characters consisting of lowercase letters and digits
   - Location: the name of an Azure region different from the one you chose when creating the first storage account
   - Performance: **Standard**
   - Account kind: **Storage (general purpose v2)**
   - Access tier: **Hot**
   - Replication: **Geo-redundant storage (GRS)**
   - Secure transfer required: **Disabled**
   - Allow access from: **All networks**
   - **Data Lake Storage Gen2** Hierarchical namespace: **Disabled**
10. Wait for the storage account to be provisioned. This should take less than a minute.

4.1.0.2 Task 2: Review configuration settings of Azure Storage accounts

1. In Azure Portal, navigate to the blade of the first storage account you created.
2. With your storage account blade open, review the storage account configuration in the **Overview** section, including the performance, replication, and account kind settings.
3. Display the **Access keys** blade. Note that you have the option of copying the values of storage account name, as well as the values of key1 and key2. You also have the option to regenerate each of the keys.
4. Display the **Configuration** blade of the storage account.

5. On the **Configuration** blade, note that you have the option of performing an upgrade to **General Purpose v2** account, enforcing secure transfer, and changing the replication settings to either **Geo-redundant storage (GRS)** or **Read-access geo-redundant storage (RA-GRS)**. However, you cannot change the performance setting (this setting can only be assigned when the storage account is created).
6. Display the **Encryption** blade of the storage account. Note that encryption is enabled by default and that you have the option of using your own key.

   Note: Do not change the configuration of the storage account.
7. In Azure Portal, navigate to the blade of the second storage account you created.
8. With your storage account blade open, review the storage account configuration in the **Overview** section, including the performance, replication, and account kind settings.
9. Display the **Configuration** blade of the storage account.
10. On the **Configuration** blade, note that you have the option of disabling the secure transfer requirement, setting the default access tier to **Cool**, and changing the replication settings to either **Locally-redundant storage (LRS)** or **Read-access geo-redundant storage (RA-GRS)**. In this case, you also cannot change the performance setting.
11. Display the **Encryption** blade of the storage account. Note that in this case encryption is also enabled by default and that you have the option of using your own key.

4.1.0.3 Task 3: Manage Azure Storage Blob Service

1. In the Azure portal, navigate to the **Blobs** blade of the first storage account.
2. From the **Blobs** blade of the first storage account, create a new container named **az1000202-container** with the **Public access level** set to **Private (no anonymous access)**.
3. From the **az1000202-container** blade, upload **Labfiles\AZ100\Mod02\az-100-02_azuredeploy.json** and **Labfiles\AZ100\Mod02\az-100-02_azuredeploy.parameters.json** into the container.

4.1.0.4 Task 4: Copy a container and blobs between Azure Storage accounts

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

   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.
2. In the Cloud Shell pane, run the following commands:

```
$storageAccount1Name = (Get-AzStorageAccount -ResourceGroupName 'az1000202-RG')[0].StorageAccountN
$storageAccount2Name = (Get-AzStorageAccount -ResourceGroupName 'az1000203-RG')[0].StorageAccountN
$storageAccount1Key1 = (Get-AzStorageAccountKey -ResourceGroupName 'az1000202-RG' -StorageAccountN
$storageAccount2Key1 = (Get-AzStorageAccountKey -ResourceGroupName 'az1000203-RG' -StorageAccountN
```

   Note: These commands set the values of variables representing the names of each storage account and their corresponding keys. You will use these values to copy blobs between storage accounts by using the AZCopy command line utility in the next step.
3. In the Cloud Shell pane, run the following command:

```
azcopy --source https://$storageAccount1Name.blob.core.windows.net/az1000202-container/ --destinat
```

   Note: This command uses the AzCopy utility to copy the content of the container between the two storage accounts.
4. Verify that the command returned the results confirming that the two files were transferred.
5. Navigate to the **Blobs** blade of the second storage account and verify that it includes the entry representing the newly created **az1000202-container** and that the container includes two copied blobs.

4.1.0.5 Task 5: Use a Shared Access Signature (SAS) key to access a blob

1. From the **Blobs** blade of the second storage account, navigate to the container **az1000202-container**, and then open the **az-100-02_azuredeploy.json** blade.
2. On the **az-100-02_azuredeploy.json** blade, copy the value of the **URL** property.
3. Open another Microsoft Edge window and navigate to the URL you copied in the previous step.

   Note: The browser will display the **ResourceNotFound**. This is expected since the container has the **Public access level** set to **Private (no anonymous access)**.
4. On the **az-100-02_azuredeploy.json** blade, generate a shared access signature (SAS) and the corresponding URL with the following settings:
   - Permissions: **Read**
   - Start date/time: specify the current date/time in your current time zone
   - Expiry date/time: specify the date/time 24 hours ahead of the current time
   - Allowed IP addresses: leave blank
   - Allowed protocols: **HTTP**
   - Signing key: **Key 1**
5. On the **az-100-02_azuredeploy.json** blade, copy **Blob SAS URL**.
6. From the previously opened Microsoft Edge window, navigate to the URL you copied in the previous step.

   Note: This time, you will be prompted whether you want to open or save **az-100-02_azuredeploy.json**. This is expected as well, since this time you are no longer accessing the container anonymously, but instead you are using the newly generated SAS key, which is valid for the next 24 hours.
7. Close the Microsoft Edge window displaying the prompt.

Result: After you completed this exercise, you have created two Azure Storage accounts, reviewed their configuration settings, created a blob container, uploaded blobs into the container, copied the container and blobs between the storage accounts, and used a SAS key to access one of the blobs.

4.2 Exercise 2: Implement and use Azure File Storage

Estimated Time: 10 minutes

The main tasks for this exercise are as follows:

1. Create an Azure File Service share
2. Map a drive to the Azure File Service share from an Azure VM

4.2.0.1 Task 1: Create an Azure File Service share

1. In the Azure portal, navigate to the blade displaying the properties of the second storage account you created in the previous exercise.
2. From the storage account blade, display the properties of its File Service.
3. From the storage account **Files** blade, create a new file share with the following settings:
   - Name: **az10002share1**
   - Quota: **5 GB**

4.2.0.2 Task 2: Map a drive to the Azure File Service share from an Azure VM

Note: Before you start this task, ensure that the template deployment you started in Exercise 0 has completed.

1. Navigate to the **az10002share1** blade and display the **Connect** blade.
2. From the **Connect** blade, copy into Clipboard the PowerShell commands that connect to the file share from a Windows computer.

3. In the Azure portal, navigate to the **az1000201-vm1** blade.
4. From the **az1000201-vm1** blade, connect to the Azure VM via the RDP protocol and, when prompted to sign in, provide the following credentials:
   - Admin Username: **Student**
   - Admin Password: **Pa55w.rd1234**
5. Within the RDP session, start a Windows PowerShell ISE session.
6. From the Windows PowerShell ISE session, open the script pane and paste into it the content of your local Clipboard.
7. Execute the script and verify that its output confirms successful mapping of the Z: drive to the Azure Storage File Service share.
8. Start File Explorer, navigate to the Z: drive and create a folder named **Folder1**.
9. In the File Explorer window, navigate to **Folder1** and create a text document named **File1.txt**.

   > **Note**: Make sure that you take into account the default configuration of File Explorer that does not display known file extensions in order to avoid creating a file named **File1.txt.txt**.

   Result: After you completed this exercise, you have created an Azure File Service share, mapped a drive to the file share from an Azure VM, and used File Explorer from the Azure VM to create a folder and a file in the file share. # AZ 100 Module 3 - Deploy and Manage Virtual Machines

5 Lab 2: Configure compute and storage resources of Azure VMs and Azure VM scale sets

Estimated Time: 90 minutes

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?view=azps-1.2.0

Lab files:

- **Allfiles/Labfiles/AZ-100.3/az-100-03b_01_azuredeploy.json**
- **Allfiles/Labfiles/AZ-100.3/az-100-03b_01_azuredeploy.parameters.json**
- **Allfiles/Labfiles/AZ-100.3/az-100-03b_02_azuredeploy.json**
- **Allfiles/Labfiles/AZ-100.3/az-100-03b_02_azuredeploy.parameters.json**

5.0.1 Scenario

Adatum Corporation wants to scale compute and storage resources for workloads running on Azure VMs and Azure VM scale sets

5.0.2 Objectives

After completing this lab, you will be able to:

- Deploy Azure VMs and Azure VM scale sets by using Azure Resource Manager templates
- Configure compute and storage resources of Azure VMs
- Configure compute and storage resources of Azure VM scale sets

5.0.3 Exercise 0: Prepare the lab environment

Estimated Time: 20 minutes

The main tasks for this exercise are as follows:

1. Deploy an Azure VM by using an Azure Resource Manager template

2. Deploy an Azure VM scale set by using an Azure Resource Manager template

5.0.3.1 Task 1: Deploy an Azure VM 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 **New** blade.
3. From the **New** 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-100-03b_01_azuredeploy.json**.

   Note: Review the content of the template and note that it defines deployment of an Azure VM hosting Windows Server 2016 Datacenter.
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-100-03b_01_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 **az1000301b-RG**
    - Location: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs
    - Vm Name: **az1000301b-vm1**
    - Vm Size: **Standard_DS1_v2**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**

    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 but proceed to the next task of this exercise. You will use the virtual machine included in this deployment in the next exercise of this lab.

5.0.3.2 Task 2: Deploy an Azure VM scale set by using an Azure Resource Manager template

1. On the lab virtual machine, in the Azure portal, navigate to the **New** 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-100-03b_02_azuredeploy.json**.

   Note: Review the content of the template and note that it defines deployment of an Azure VM scale set hosting Windows Server 2016 Datacenter.
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-100-03b_02_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 **az1000302b-RG**
- Location: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs
- Vmss Name: **az1000302bvmss1**
- Vm Size: **Standard_DS1_v2**
- Admin Username: **Student**
- Admin Password: **Pa55w.rd1234**
- Instance Count: **1**

  Note: Do not wait for the deployment to complete but proceed to the next exercise. You will use the virtual machine scale set included in this deployment in the last exercise of this lab.

Result: After you completed this exercise, you have initiated a template deployment of an Azure VM **az1000301b-vm1** and an Azure VM scale set **az1000302bvmss1** that you will use in the next exercise of this lab.

5.1 Exercise 1: Configure compute and storage resources of Azure VMs

Estimated Time: 40 minutes

The main tasks for this exercise are as follows:

1. Scale vertically compute resources of the Azure VM by using the Azure portal
2. Scale vertically compute resources of the Azure VM by using an ARM template
3. Attach data disks to the Azure VM
4. Configure data volumes within an Azure VM

5.1.0.1 Task 1: Scale vertically compute resources of the Azure VM by using the Azure portal

1. From the lab virtual machine, in the Azure portal, navigate to the **az1000301b-RG** resource group blade.
2. From the **az1000301b-RG** resource group blade, navigate to the **az1000301b-vm1** virtual machine blade.
3. From the **az1000301b-vm1** virtual machine blade, navigate to the **az1000301b-vm1 - Size** virtual machine blade.
4. From the **az1000301b-vm1 - Size** virtual machine blade, increase the size of the virtual machine to **DS2_v2 Standard**.

   Note: If this size is not available, choose another size. To identify Azure VM sizes that you can choose from, refer to https://azure.microsoft.com/en-us/global-infrastructure/services/

   Note: Keep in mind that resizing an Azure VM requires restarting its operating system.

5.1.0.2 Task 2: Scale vertically compute resources of the Azure VM by using an ARM template

1. From the lab virtual machine, in the Azure portal, navigate to the **az1000301b-RG** resource group blade.
2. From the **az1000301b-RG** resource group blade, navigate to the **az1000301b-RG - Deployments** blade.
3. From the **az1000301b-RG - Deployments** blade, navigate to the **Microsoft.Template - Overview** blade showing the most recent successful deployment.
4. On the **az1000301b-RG - Deployments** blade, use the **Redeploy** option to navigate to the **Custom deployment** blade.
5. From the **Custom deployment** blade, navigate to the **Edit parameters** blade.

6. On the **Edit parameters** blade, review the values of the original parameters used for the most recent successful deployment.

   Note: The value of the **adminPassword** parameter is null because that parameter has the **securestring** data type.

7. Save the parameters and return to the **Custom deployment** blade.
8. 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: **az1000301b-RG**
   - Location: the name of the same Azure region you chose previously
   - Vm Name: **az1000301b-vm1**
   - Vm Size: **Standard_DS1_v2**
   - Admin Username: **Student**
   - Admin Password: **Pa55w.rd1234**

   Note: Wait for the deployment to complete before you proceed to the next task of this exercise. The deployment will change the size of the Azure VM back to its original value, effectively scaling it down.

5.1.0.3 Task 3: Attach data disks to the Azure VM

1. In the Azure portal, navigate to the **az1000301b-vm1** virtual machine blade.
2. From the **az1000301b-vm1** virtual machine blade, navigate to the **az1000301b-vm1 - Disks** blade.
3. From the **az1000301b-vm1 - Disks** blade, use the **+ Add data disk** option to navigate to the **Create managed disk** blade.
4. From the **Create managed disk** blade, create a new data disk with the following settings:
   - Name: **az1000301b-vm1-DataDisk0**
   - Resource group: **az1000301b-RG**
   - Account type: **Standard HDD**
   - Source type: **None (empty disk)**
   - Size (GiB): **1023**
5. Back on the **az1000301b-vm1 - Disks** blade, configure the following settings for the newly created disk:
   - LUN: **0**
   - HOST CACHING: **None**
6. From the **az1000301b-vm1 - Disks** blade, use the **+ Add data disk** option to navigate to the **Create managed disk** blade.
7. From the **Create managed disk** blade, create a new data disk with the following settings:
   - Name: **az1000301b-vm1-DataDisk1**
   - Resource group: **az1000301b-RG**
   - Account type: **Standard HDD**
   - Source type: **None (empty disk)**
   - Size (GiB): **1023**
8. Back on the **az1000301b-vm1 - Disks** blade, configure the following settings for the newly created disk:
   - LUN: **1**
   - HOST CACHING: **None**

5.1.0.4 Task 4: Configure data volumes within an Azure VM

1. In the Azure portal, navigate to the **az1000301b-vm1** blade.
2. From the **az1000301b-vm1** blade, connect to the Azure VM via the RDP protocol and, when prompted to sign in, provide the following credentials:
   - Admin Username: **Student**
   - Admin Password: **Pa55w.rd1234**
3. Within the RDP session, in Server Manager, navigate to **File and Storage Services**, use New Storage Pool Wizard to create a new storage pool named **StoragePool1** consisting of two disks you attached in the previous task, use New Virtual Disk Wizard to create a new virtual disk named **VirtualDisk1** with **Simple** layout, **Fixed** provisioning type, and maximum size, and use the New Volume Wizard to create a single volume occupying entire virtual disk with the following settings:
   - Drive letter: **F**
   - File system: **NTFS**
   - Allocation unit size: **Default**
   - Volume label: **Data**

   Result: After you completed this exercise, you have scaled vertically compute resources of the Azure VM by using the Azure portal and by using an ARM template, attached data disks to the Azure VM, and configured data volumes within an Azure VM.

5.2 Exercise 2: Configure compute and storage resources of Azure VM scale sets

Estimated Time: 30 minutes

The main tasks for this exercise are as follows:

1. Scale vertically compute resources of the Azure VM scale set by using an Azure Resource Manager template
2. Attach data disks to the Azure VM scale set
3. Configure data volumes in the Azure VM scale set
4. Scale horizontally compute resources of the Azure VM scale set

5.2.0.1 Task 1: Scale vertically compute resources of the Azure VM scale set by using an Azure Resource Manager template

1. From the lab virtual machine, in the Azure portal, navigate to the **az1000302b-RG** resource group blade.
2. From the **az1000302b-RG** resource group blade, navigate to the **az1000302b-RG - Deployments** blade.
3. From the **az1000302b-RG - Deployments** blade, navigate to the **Microsoft.Template - Overview** blade showing the most recent successful deployment.
4. On the **az1000302b-RG - Deployments** blade, use the **Redeploy** option to navigate to the **Custom deployment** blade.
5. 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: **az1000302b-RG**
   - Location: the name of the same Azure region you chose previously
   - Vm Name: **az1000302bvmss1**
   - Vm Size: the name of the VM size you used to scale up the Azure VM in the previous exercise of this lab
   - Admin Username: **Student**
   - Admin Password: **Pa55w.rd1234**

Note: Wait for the deployment to complete before you proceed to the next task of this exercise. The deployment will increase the size of Azure VM instances of the Azure VM scale set.

6. In the Azure portal, navigate to the **az1000302bvmss1** blade and verify that the size of the VM instances of the VM scale set has changed.

5.2.0.2 Task 2: Attach data disks to the Azure VM scale set

1. 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.

2. In the Cloud Shell pane, run the following in order to attach one 128 GB disk to each VM instance in the VM scale set:

```
$vmss = Get-AzVmss -ResourceGroupName 'az1000302b-RG' -VMScaleSetName 'az1000302bvmss1'

Add-AzVmssDataDisk -VirtualMachineScaleSet $vmss -CreateOption Empty -Lun 1 -DiskSizeGB 128 -Storag

Update-AzVmss -ResourceGroupName $vmss.ResourceGroupName -VirtualMachineScaleSet $vmss -VMScaleSetN
```

3. In the Azure portal, navigate to the **az1000302bvmss1 - Storage** blade and verify that the data disk has been added.

5.2.0.3 Task 3: Configure data volumes in the Azure VM scale set

1. In the Cloud Shell pane, run the following in order to configure a simple volume on the newly added disk by using Custom Script Extension:

```
$vmss = Get-AzVmss -ResourceGroupName 'az1000302b-RG' -VMScaleSetName 'az1000302bvmss1'

$publicSettings = @{"fileUris" = (,"https://raw.githubusercontent.com/Azure-Samples/compute-automat

Add-AzVmssExtension -VirtualMachineScaleSet $vmss -Name "customScript" -Publisher "Microsoft.Comput

Update-AzVmss -ResourceGroupName $vmss.ResourceGroupName -VirtualMachineScaleSet $vmss -VMScaleSetN
```

   Note: To confirm that the volume has been configured, you will connect to the VM instance in the VM scale set via RDP.

2. To identify the public IP address of the Azure load balancer in front of the VM scale set, in the Azure portal, navigate to the **az1000302bvmss1** blade and note the value of the **Public IP address** entry.
3. To identify the port on which you should connect, navigate to the **az1000302b-RG** resource group blade.
4. From the **az1000302b-RG** resource group blade, navigate to the **az1000302bvmss1-lb** load balancer blade.
5. On the **az1000302bvmss1-lb** load balancer blade, note that the value of the **Public IP address** setting matches the value of the IP address used by the VM scale set you identified earlier in this task.
6. From the **az1000302bvmss1-lb** load balancer blade, navigate to the **az1000302bvmss1-lb - Inbound NAT rules** blade.
7. From the **az1000302bvmss1-lb - Inbound NAT rules** blade, navigate to the **natpool.0** blade and note that the port mappings start with **50000**.
8. From the lab computer, initiate the Remote Desktop Connection to the first VM instance in the VM scale set by running the following from **Start -> Run** text box (where `<public_IP_address>` represents the public IP address you identified earlier in this task):

```
mstsc /f /v:<public_IP_address>:50000
```

   Note: Make sure to replace the placeholder `<public_IP_address>` with the value of the public IP address you identified earlier in this task.

9. When prompted to sign in via RDP to a VM instance of the VM scale set, provide the following credentials:

- Admin Username: **Student**
- Admin Password: **Pa55w.rd1234**

  Note: Ignore any messages regarding signing in with temporary profile.

10. Within the RDP session, launch File Explorer and verify that the VM has an extra volume of 127 GB in size with F: drive letter assigned to it.
11. Sign out from the RDP session.

5.2.0.4 Task 4: Scale horizontally compute resources of the Azure VM scale set

1. From the lab virtual machine, in the Azure portal, navigate to the **az1000302bvmss1** VM scale set blade.
2. From the **az1000302bvmss1** VM scale set blade, navigate to the **az1000302bvmss1 - Scaling** blade.
3. On the **az1000302bvmss1 - Scaling** blade, use the **Override condition** setting to increase the instance count to 2.
4. Navigate to the **az1000302bvmss1 - Instances** blade and verify that the number of instances has increased to 2.

   Note: You might need to refresh the display on the **az1000302bvmss1 - Instances** blade to view the process of provisioning the additional instance.

   Result: After you completed this exercise, you have scaled vertically compute resources of the Azure VM scale set by using an Azure Resource Manager template, attached data disks to the Azure VM scale set, configured data volumes in the Azure VM scale set, and scaled horizontally compute resources of the Azure VM scale set. # AZ 100 Module 3 - Deploy and Manage Virtual Machines

6 Lab: Deploy and Manage Virtual Machines

Estimated Time: 60 minutes

All tasks in this lab are performed from the Azure portal (including a PowerShell Cloud Shell session) except for Excercise 2 Task 2 and Exercise 2 Task 3, which include steps performed from a Remote Desktop session to an Azure VM

> **Note**: When not using Cloud Shell, the lab virtual machine must have Azure PowerShell module installed **https://docs.microsoft.com/en-us/powershell/azure/install-azurerm-ps?view=azurermps-6.12.0**

Lab files:

- **Labfiles\AZ100\Mod03\az-100-03_deploy_azure_vm.ps1**
- **Labfiles\AZ100\Mod03\az-100-03_azuredeploy.json**
- **Labfiles\AZ100\Mod03\az-100-03_azuredeploy.parameters.json**
- **Labfiles\AZ100\Mod03\az-100-03_install_iis_vmss.zip**

6.0.1 Scenario

Adatum Corporation wants to implement its workloads by using Azure virtual machines (VMs) and Azure VM scale sets

6.0.2 Objectives

After completing this lab, you will be able to:

- Deploy Azure VMs by using the Azure portal, Azure PowerShell, and Azure Resource Manager templates
- Configure networking settings of Azure VMs running Windows and Linux operating systems
- Deploy and configure Azure VM scale sets

6.1 Exercise 1: Deploy Azure VMs by using the Azure portal, Azure PowerShell, and Azure Resource Manager templates

Estimated Time: 25 minutes

The main tasks for this exercise are as follows:

1. Deploy an Azure VM running Windows Server 2016 Datacenter into an availability set by using the Azure portal
2. Deploy an Azure VM running Windows Server 2016 Datacenter into the existing availability set by using Azure PowerShell
3. Deploy two Azure VMs running Linux into an availability set by using an Azure Resource Manager template

6.1.0.1 Task 1: Deploy an Azure VM running Windows Server 2016 Datacenter into an availability set by using the Azure portal

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 **Windows Server 2016 Datacenter**.
4. Use the list of search results to navigate to the **Create a virtual machine** blade for a deployment of the Windows Server 2016 Datacenter Azure Marketplace image.
5. Use the **Create a virtual machine** blade to deploy a virtual machine 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 **az1000301-RG**
   - Virtual machine name: **az1000301-vm0**
   - Region: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs
   - Availability options: **Availability set**
   - Availability set: the name of a new availability set **az1000301-avset0** with **2** fault domains and **5** update domains.
   - Image: **Windows Server 2016 Datacenter**
   - Size: **Standard DS1 v2**
   - Username: **Student**
   - Password: **Pa55w.rd1234**
   - Public inbound ports: **None**
   - Already have a Windows license?: **No**
   - OS disk type: **Standard HDD**
   - Virtual network: the name of a new virtual network **az1000301-vnet0** with the following settings:
     - Address space: **10.103.0.0/16**
     - Subnet name: **subnet0**
     - Subnet address range: **10.103.0.0/24**
   - Public IP: the name of a new public IP address **az1000301-vm0-ip**
   - Network security group: **Basic**
   - Public inbound ports: **None**
   - Accelerated networking: **Off**

- Boot diagnostics: **Off**
- OS guest diagnostics: **Off**
- System assigned managed identity: **Off**
- Enable auto-shutdown: **Off**
- Enable backup: **Off**

Note: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**

Note: You will configure the network security group you create in this task in the second exercise of this lab

Note: Wait for the deployment to complete before you proceed to the next task. This should take about 5 minutes.

6.1.0.2 Task 2: Deploy an Azure VM running Windows Server 2016 Datacenter into the existing availability set by using Azure PowerShell

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

   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.

2. In the Cloud Shell pane, run the following command:

```
$vmName = 'az1000301-vm1'
$vmSize = 'Standard_DS1_v2'
```

   Note: This sets the values of variables designating the Azure VM name and its size

3. In the Cloud Shell pane, run the following commands:

```
$resourceGroup = Get-AzResourceGroup -Name 'az1000301-RG'
$location = $resourceGroup.Location
```

   Note: These commands set the values of variables designating the target resource group and its location

4. In the Cloud Shell pane, run the following commands:

```
$availabilitySet = Get-AzAvailabilitySet -ResourceGroupName $resourceGroup.ResourceGroupName -Name
$vnet = Get-AzVirtualNetwork -Name 'az1000301-vnet0' -ResourceGroupName $resourceGroup.ResourceGro
$subnetid = (Get-AzVirtualNetworkSubnetConfig -Name 'subnet0' -VirtualNetwork $vnet).Id
```

   Note: These commands set the values of variables designating the availability set, virtual network, and subnet into which you will deploy the new Azure VM

5. In the Cloud Shell pane, run the following commands:

```
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $resourceGroup.ResourceGroupName -Location $l
$pip = New-AzPublicIpAddress -Name "$vmName-ip" -ResourceGroupName $resourceGroup.ResourceGroupNam
$nic = New-AzNetworkInterface -Name "$($vmName)$(Get-Random)" -ResourceGroupName $resourceGroup.Re
```

   Note: These commands create a new network security group, public IP address, and network interface that will be used by the new Azure VM

   Note: You will configure the network security group you create in this task in the second exercise of this lab

6. In the Cloud Shell pane, run the following commands:

```
$adminUsername = 'Student'
$adminPassword = 'Pa55w.rd1234'
$adminCreds = New-Object PSCredential $adminUsername, ($adminPassword | ConvertTo-SecureString -As
```

   Note: These commands set the values of variables designating credentials of the local Administrator account of the new Azure VM

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

```
$publisherName = 'MicrosoftWindowsServer'
$offerName = 'WindowsServer'
$skuName = '2016-Datacenter'
```

> **Note**: These commands set the values of variables designating the properties of the Azure Marketplace image that will be used to provision the new Azure VM

8. In the Cloud Shell pane, run the following command:

```
$osDiskType = (Get-AzResource -ResourceGroupName $resourceGroup.ResourceGroupName -ResourceType Mi
```

> **Note**: This command sets the values of a variable designating the operating system disk type of the new Azure VM

9. In the Cloud Shell pane, run the following commands:

```
$vmConfig = New-AzVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $availabilitySet.Id
Add-AzVMNetworkInterface -VM $vmConfig -Id $nic.Id
Set-AzVMOperatingSystem -VM $vmConfig -Windows -ComputerName $vmName -Credential $adminCreds
Set-AzVMSourceImage -VM $vmConfig -PublisherName $publisherName -Offer $offerName -Skus $skuName -
Set-AzVMOSDisk -VM $vmConfig -Name "$($vmName)_OsDisk_1_$(Get-Random)" -StorageAccountType $osDisk
Set-AzVMBootDiagnostics -VM $vmConfig -Disable
```

> **Note**: These commands set up the properties of the Azure VM configuration object that will be used to provision the new Azure VM, including the VM size, its availability set, network interface, computer name, local Administrator credentials, the source image, the operating system disk, and boot diagnostics settings.

10. In the Cloud Shell pane, run the following command:

```
New-AzVM -ResourceGroupName $resourceGroup.ResourceGroupName -Location $location -VM $vmConfig
```

> **Note**: This command initiates deployment of the new Azure VM
>
> **Note**: Do not wait for the deployment to complete but instead proceed to the next task.

6.1.0.3 Task 3: Deploy two Azure VMs running Linux into an availability set by using an Azure Resource Manager template

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Template deployment**.
3. Use the list of search results to navigate to the **Deploy a custom template** 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 **Labfiles\AZ100\Mod03\az-100-03_azuredeploy.json**.

> **Note**: Review the content of the template and note that it defines deployment of two Azure VMs hosting Linux Ubuntu into an availability set and into the existing virtual network **az1000301-vnet0**.

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 **Labfiles\AZ100\Mod03\az-100-03_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 **az1000302-RG**
    - Location: the same Azure region you chose earlier in this exercise
    - Vm Name Prefix: **az1000302-vm**

- Nic Name Prefix: **az1000302-nic**
- Pip Name Prefix: **az1000302-ip**
- Admin Username: **Student**
- Admin Password: **Pa55w.rd1234**
- Virtual Network Name: **az1000301-vnet0**
- Image Publisher: **Canonical**
- Image Offer: **UbuntuServer**
- Image SKU: **16.04.0-LTS**
- Vm Size: **Standard_DS1_v2**

  Note: Wait for the deployment to complete before you proceed to the next task. This should take about 5 minutes.

Result: After you completed this exercise, you have deployed an Azure VM running Windows Server 2016 Datacenter into an availability set by using the Azure portal, deployed another Azure VM running Windows Server 2016 Datacenter into the same availability set by using Azure PowerShell, and deployed two Azure VMs running Linux Ubuntu into an availability set by using an Azure Resource Manager template.

Note: You could certainly use a template to deploy two Azure VMs hosting Windows Server 2016 datacenter in a single task (just as this was done with two Azure VMs hosting Linux Ubuntu server). The reason for deploying these Azure VMs in two separate tasks was to give you the opportunity to become familiar with both the Azure portal and Azure PowerShell-based deployments.

6.2 Exercise 2: Configure networking settings of Azure VMs running Windows and Linux operating systems

Estimated Time: 10 minutes

The main tasks for this exercise are as follows:

1. Configure static private and public IP addresses of Azure VMs
2. Connect to an Azure VM running Windows Server 2016 Datacenter via a public IP address
3. Connect to an Azure VM running Linux Ubuntu Server via a private IP address

6.2.0.1 Task 1: Configure static private and public IP addresses of Azure VMs

1. In the Azure portal, navigate to the **az1000301-vm0** blade.
2. From the **az1000301-vm0** blade, navigate to the **az1000301-vm0-ip - Configuration** blade, displaying the configuration of the public IP address **az1000301-vm0-ip**, assigned to its network interface.
3. From the **az1000301-vm0-ip - Configuration** blade, change the assignment of the public IP address to **Static**.

   Note: Take a note of the public IP address assigned to the network interface of **az1000301-vm0**. You will need it later in this exercise.
4. In the Azure portal, navigate to the **az1000302-vm0** blade.
5. From the **az1000302-vm0** blade, display the **az1000302-vm0 - Networking** blade.
6. From the **az1000302-vm0 - Networking** blade, navigate to the blade displaying the properties of its network interface.
7. From the blade displaying the properties of the network interface of **az1000302-vm0**, navigate to its **ipconfig1** blade.
8. On the **ipconfig1** blade, configure the private IP address to be static and set it to **10.103.0.100**.

   Note: Changing the private IP address assignment requires restarting the Azure VM.

Note: It is possible to connect to Azure VMs via either statically or dynamically assigned public and private IP addresses. Choosing static IP assignment is commonly done in scenarios where these IP addresses are used in combination with IP filtering, routing, or if they are assigned to network interfaces of Azure VMs that function as DNS servers.

6.2.0.2 Task 2: Connect to an Azure VM running Windows Server 2016 Datacenter via a public IP address

1. In the Azure portal, navigate to the **az1000301-vm0** blade.
2. From the **az1000301-vm0** blade, navigate to the **az1000301-vm0 - Networking** blade.
3. On the **az1000301-vm0 - Networking** blade, review the inbound port rules of the network security group assigned to the network interface of **az1000301-vm0**.

   Note: The default configuration consisting of built-in rules block inbound connections from the internet (including connections via the RDP port TCP 3389)
4. Add an inbound security rule to the existing network security group with the following settings:
   - Source: **Any**
   - Source port ranges: *****
   - Destination: **Any**
   - Destination port ranges: **3389**
   - Protocol: **TCP**
   - Action: **Allow**
   - Priority: **100**
   - Name: **AllowInternetRDPInBound**
5. In the Azure portal, display the **Overview** pane of the **az1000301-vm0** blade.
6. From the **Overview** pane of the **az1000301-vm0** blade, generate an RDP file and use it to connect to **az1000301-vm0**.
7. When prompted, authenticate by specifying the following credentials:
   - User name: **Student**
   - Password: **Pa55w.rd1234**

6.2.0.3 Task 3: Connect to an Azure VM running Linux Ubuntu Server via a private IP address

1. Within the RDP session to **az1000301-vm0**, start **Command Prompt**.
2. From the Command Prompt, run the following:

   ```
   nslookup az1000302-vm0
   ```
3. Examine the output and note that the name resolves to the IP address you assigned in the first task of this exercise (**10.103.0.100**).

   Note: This is expected. Azure provides built-in DNS name resolution within a virtual network.
4. Within the RDP session to **az1000301-vm0**, from Server Manager, disable temporarily **IE Enhanced Security Configuration**.
5. Within the RDP session to **az1000301-vm0**, start Internet Explorer and download **putty.exe** from **https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html**
6. Use **putty.exe** to verify that you can successfully connect to **az1000302-vm0** on its private IP address via the **SSH** protocol (TCP 22).
7. When prompted, authenticate by specifying the following values:
   - User name: **Student**
   - Password: **Pa55w.rd1234**

8. Once you successfully authenticated, terminate the RDP session to **az1000301-vm0**.
9. On the lab virtual machine, in the Azure portal, navigate to the **az1000302-vm0** blade.
10. From the **az1000302-vm0** blade, navigate to the **az1000302-vm0 - Networking** blade.
11. On the **az1000302-vm0 - Networking** blade, review the inbound port rules of the network security group assigned to the network interface of **az1000301-vm0** to determine why your SSH connection via the private IP address was successsful.

    Note: The default configuration consisting of built-in rules allows inbound connections within the Azure virtual network environment (including connections via the SSH port TCP 22).

    Result: After you completed this exercise, you have configured static private and public IP addresses of Azure VMs, connected to an Azure VM running Windows Server 2016 Datacenter via a public IP address, and connect to an Azure VM running Linux Ubuntu Server via a private IP address

6.3 Exercise 3: Deploy and configure Azure VM scale sets

Estimated Time: 25 minutes

The main tasks for this exercise are as follows:

1. Identify an available DNS name for an Azure VM scale set deployment
2. Deploy an Azure VM scale set
3. Install IIS on a scale set VM by using DSC extensions

6.3.0.1 Task 1: Identify an available DNS name for an Azure VM scale set deployment

1. From the Azure Portal, start a PowerShell session in the Cloud Shell pane.
2. In the Cloud Shell pane, run the following command, substituting the placeholder `<custom-label>` with any string which is likely to be unique and the placeholder `<location-of-az1000301-RG>` with the name of the Azure region in which you created the **az1000301-RG** resource group.

   ```
   Test-AzDnsAvailability -DomainNameLabel <custom-label> -Location '<location-of-az1000301-RG>'
   ```

3. Verify that the command returned **True**. If not, rerun the same command with a different value of the `<custom-label>` until the command returns **True**.
4. Note the value of the `<custom-label>` that resulted in the successful outcome. You will need it in the next task

6.3.0.2 Task 2: Deploy an Azure VM scale set

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Virtual machine scale set**.
3. Use the list of search results to navigate to the **Create virtual machine scale set** blade.
4. Use the **Create virtual machine scale set** blade to deploy a virtual machine scale set with the following settings:
   - Virtual machine scale set name: **az1000303vmss0**
   - Operating system disk image: **Windows Server 2016 Datacenter**
   - Subscription: the name of the subscription you are using in this lab
   - Resource group: the name of a new resource group **az1000303-RG**
   - Location: the same Azure region you chose in the previous exercises of this lab
   - Availability zone: **None**
   - Username: **Student**
   - Password: **Pa55w.rd1234**
   - Instance count: **1**

- Instance size: **DS1 v2**
- Deploy as low priority: **No**
- Use managed disks: **Yes**
- Autoscale: **Disabled**
- Choose Load balancing options: **Load balancer**
- Public IP address name: **az1000303vmss0-ip**
- Domain name label: type in the value of the `<custom-label>` you identified in the previous task
- Virtual network: the name of a new virtual network **az1000303-vnet0** with the following settings:
  - Address space: **10.203.0.0/16**
  - Subnet name: **subnet0**
  - Subnet address range: **10.203.0.0/24**
- Public IP address per instance: **Off**

  Note: Wait for the deployment to complete before you proceed to the next task. This should take about 5 minutes.

6.3.0.3 Task 3: Install IIS on a scale set VM by using DSC extensions

1. In the Azure portal, navigate to the **az1000303vmss0** blade.
2. From the **az1000303vmss0** blade, display its Extension blade.
3. From the **az1000303vmss0 - Extension** blade, add the **PowerShell Desired State Configuration** extension with the following settings:

   Note: The DSC configration module is available for upload from **Labfiles\AZ100\Mod03\az-100-03_install_iis_vmss.zip**. The module contains the DSC configuration script that installs the Web Server (IIS) role.

   - Configuration Modules or Script: **"az-100-03_install_iis_vmss.zip"**
   - Module-qualified Name of Configuration: **az-100-03_install_iis_vmss.ps1\IISInstall**
   - Configuration Arguments: leave blank
   - Configuration Data PSD1 File: leave blank
   - WMF Version: **latest**
   - Data Collection: **Disable**
   - Version: **2.76**
   - Auto Upgrade Minor Version: **Yes**
4. Navigate to the **az1000303vmss0 - Instances** blade and initiate the upgrade of the **az1000303vmss0_0** instance.

   Note: The update will trigger application of the DSC configration script. Wait for upgrade to complete. This should take about 5 minutes. You can monitor the progress from the **az1000303vmss0 - Instances** blade.
5. Once the upgrade completes, navigate to the **az1000303vmss0-ip** blade.
6. On the **az1000303vmss0-ip** blade, note the public IP address assigned to **az1000303vmss0**.
7. Start Microsoft Edge and navigate to the public IP address you identified in the previous step.
8. Verify that the browser displays the default IIS home page.

Result: After you completed this exercise, you have identified an available DNS name for an Azure VM scale set deployment, deployed an Azure VM scale set, and installed IIS on a scale set VM by using the DSC extension. # AZ 100 Module 4 - Configure and Manage Virtual Networks

7 Lab 2: Configure Azure DNS

Estimated Time: 90 minutes

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?view=azps-1.2.0

Lab files:

- **Allfiles/Labfiles/AZ-100.4/az-100-04b_01_azuredeploy.json**
- **Allfiles/Labfiles/AZ-100.4/az-100-04b_02_azuredeploy.json**
- **Allfiles/Labfiles/AZ-100.4/az-100-04_azuredeploy.parameters.json**

7.0.1 Scenario

Adatum Corporation wants to implement public and private DNS service in Azure without having to deploy its own DNS servers.

7.0.2 Objectives

After completing this lab, you will be able to:

- Configure Azure DNS for public domains
- Configure Azure DNS for private domains

7.1 Exercise 1: Configure Azure DNS for public domains

Estimated Time: 25 minutes

The main tasks for this exercise are as follows:

1. Create a public DNS zone
2. Create a DNS record in the public DNS zone
3. Validate Azure DNS-based name resolution for the public domain

7.1.0.1 Task 1: Create a public DNS zone

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 **New** blade.
3. From the **New** blade, search Azure Marketplace for **DNS zone**.
4. Use the list of search results to navigate to the **Create DNS zone** blade.
5. From the to **Create DNS zone** blade, create a new DNS zone with the following settings:
   - Name: any unique, valid DNS domain name in the **.com** namespace
   - Subscription: the name of the Azure subscription you are using in this lab
   - Resource group: the name of a new resource group **az1000401b-RG**
   - Resource group location: the name of the Azure region which is closest to the lab location and where you can provision Azure DNS zones

7.1.0.2 Task 2: Create a DNS record in the public DNS zone

1. 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.

2. In the Cloud Shell pane, run the following in order to identify the public IP address of your lab computer:

   ```
   Invoke-RestMethod http://ipinfo.io/json | Select-Object -ExpandProperty IP
   ```

   Note: Take a note of this IP address. You will use it later in this task.

3. In the Cloud Shell pane, run the following in order to create a public IP address resource:

   ```
   $rg = Get-AzResourceGroup -Name az1000401b-RG

   New-AzPublicIpAddress -ResourceGroupName $rg.ResourceGroupName -Sku Basic -AllocationMethod Static
   ```

4. In the Azure portal, navigate to the **az1000401b-RG** resource group blade.

5. From the **az1000401b-RG** resource group blade, navigate to the blade displaying newly created public DNS zone.

6. From the DNS zone blade, navigate to the **Add record set** blade and create a DNS record with the following settings:

   - Name: **mylabvmpip**
   - Type: **A**
   - Alias record set: **No**
   - TTL: **1**
   - TTL unit: **Hours**
   - IP ADDRESS: the public IP address of your lab computer you identified earlier in this task

7. From the **Add record set** blade, create another record with the following settings:

   - Name: **myazurepip**
   - Type: **A**
   - Alias record set: **Yes**
   - Alias type: **Azure resource**
   - Choose a subscription: the name of the Azure subscription you are using in this lab
   - Azure resource: **az1000401b-pip**
   - TTL: **1**
   - TTL unit: **Hours**

7.1.0.3 Task 3: Validate Azure DNS-based name resolution for the public domain

1. On the DNS zone blade, note the list of the name servers that host the zone you created. You will use the first of them named in the next step.

2. From the lab virtual machine, start Command Prompt and run the following to validate the name resolution of the two newly created DNS records (where `<custom_DNS_domain>` represents the custom DNS domain you created in the first task of this exercise and `<name_server>` represents the name of the DNS name server you identified in the previous step):

   ```
   nslookup mylabvmpip.<custom_DNS_domain> <name_server>

   nslookup myazurepip.<custom_DNS_domain> <name_server>
   ```

3. Verify that the IP addresses returned match those you identified earlier in this task.

Result: After you completed this exercise, you have created a public DNS zone, created a DNS record in the public DNS zone, and validated Azure DNS-based name resolution for the public domain.

7.2 Exercise 2: Configure Azure DNS for private domains

Estimated Time: 65 minutes

The main tasks for this exercise are as follows:

1. Provision a multi-virtual network environment
2. Create a private DNS zone
3. Deploy Azure VMs into virtual networks
4. Validate Azure DNS-based name reservation and resolution for the private domain

7.2.0.1 Task 1: Provision a multi-virtual network environment

1. From the Azure Portal, start a PowerShell session in the Cloud Shell.
2. In the Cloud Shell pane, run the following in order to create a resource group:

```
$rg1 = Get-AzResourceGroup -Name 'az1000401b-RG'

$rg2 = New-AzResourceGroup -Name 'az1000402b-RG' -Location $rg1.Location
```

3. In the Cloud Shell pane, run the following in order to create two Azure virtual networks:

```
$subnet1 = New-AzVirtualNetworkSubnetConfig -Name subnet1 -AddressPrefix '10.104.0.0/24'

$vnet1 = New-AzVirtualNetwork -ResourceGroupName $rg2.ResourceGroupName -Location $rg2.Location -Na

$subnet2 = New-AzVirtualNetworkSubnetConfig -Name subnet1 -AddressPrefix '10.204.0.0/24'

$vnet2 = New-AzVirtualNetwork -ResourceGroupName $rg2.ResourceGroupName -Location $rg2.Location -Na
```

7.2.0.2 Task 2: Create a private DNS zone

1. In the Cloud Shell pane, run the following in order to create a private DNS zone with the first virtual network supporting registration and the second virtual network supporting resolution:

```
New-AzDnsZone -Name adatum.local -ResourceGroupName $rg2.ResourceGroupName -ZoneType Private -Regis
```

   Note: Virtual networks that you assign to an Azure DNS zone cannot contain any resources.

2. In the Cloud Shell pane, run the following in order to verify that the private DNS zone was successfully created:

```
Get-AzDnsZone -ResourceGroupName $rg2.ResourceGroupName
```

7.2.0.3 Task 3: Deploy Azure VMs into virtual networks

1. In the Cloud Shell pane, upload **az-100-04b__01__azuredeploy.json**, **az-100-04b__02__azuredeploy.json**, and **az-100-04__azuredeploy.parameters.json** files.
2. In the Cloud Shell pane, run the following in order to deploy an Azure VM into the first virtual network:

```
New-AzResourceGroupDeployment -ResourceGroupName $rg2.ResourceGroupName -TemplateFile "$home/az-10
```

3. In the Cloud Shell pane, run the following in order to deploy an Azure VM into the second virtual network:

```
New-AzResourceGroupDeployment -ResourceGroupName $rg2.ResourceGroupName -TemplateFile "$home/az-10
```

   Note: Wait for both deployments to complete before you proceed to the next task. You can identify the state of the jobs by running the `Get-Job` cmdlet in the Cloud Shell pane.

7.2.0.4 Task 4: Validate Azure DNS-based name reservation and resolution for the private domain

1. In the Azure portal, navigate to the blade of the **az1000402b-vm2** Azure VM.
2. From the **Overview** pane of the **az1000402b-vm2** blade, generate an RDP file and use it to connect to **az1000402b-vm2**.
3. When prompted, authenticate by specifying the following credentials:
   - User name: **Student**
   - Password: **Pa55w.rd1234**
4. Within the Remote Desktop session to **az1000402b-vm2**, start a Command Prompt window and run the following:

   ```
   nslookup az1000402b-vm1.adatum.local
   ```

5. Verify that the name is successfully resolved.
6. Switch back to the lab virtual machine and, in the Cloud Shell pane of the Azure portal window, run the following in order to create an additional DNS record in the private DNS zone:

   ```
   New-AzDnsRecordSet -ResourceGroupName $rg2.ResourceGroupName -Name www -RecordType A -ZoneName ada
   ```

7. Switch again to the Remote Desktop session to **az1000402b-vm2** and run the following from the Command Prompt window:

   ```
   nslookup www.adatum.local
   ```

8. Verify that the name is successfully resolved.

   Result: After completing this exercise, you have provisioned a multi-virtual network environment, created a private DNS zone, deployed Azure VMs into virtual networks, and validated Azure DNS-based name reservation and resolution for the private domain # AZ 100 Module 4 - Configure and Manage Virtual Networks

8 Lab: Configure VNet peering and service chaining

Estimated Time: 45 minutes

All tasks in this lab are performed from the Azure portal except for Excercise 2 Task 3, Exercise 3 Task 1, and Exercise 3 Task 2, which include steps performed from a Remote Desktop session to an Azure VM

Lab files:

- **Labfiles\AZ100\Mod04\az-100-04_01_azuredeploy.json**
- **Labfiles\AZ100\Mod04\az-100-04_02_azuredeploy.json**
- **Labfiles\AZ100\Mod04\az-100-04_azuredeploy.parameters.json**

8.0.1 Scenario

ADatum Corporation wants to implement service chaining between Azure virtual networks in its Azure subscription.

8.0.2 Objectives

After completing this lab, you will be able to:

- Create Azure virtual networks and deploy Azure VM by using Azure Resource Manager templates.
- Configure VNet peering.
- Implement custom routing
- Validate service chaining

8.0.3 Exercise 0: Prepare the Azure environment

Estimated Time: 10 minutes

The main tasks for this exercise are as follows:

1. Create the first virtual network hosting two Azure VMs by using an Azure Resource Manager template
2. Create the second virtual network in the same region hosting a single Azure VM by using an Azure Resource Manager template

8.0.3.1 Task 1: Create the first virtual network hosting two Azure VMs 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 **Deploy a custom template** 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 **Labfiles\AZ100\Mod04\az-100-04_01_azuredeploy.json**.

   Note: Review the content of the template and note that it defines deployment of an Azure VM hosting Windows Server 2016 Datacenter.
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 **Labfiles\AZ100\Mod04\az-100-04_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 **az1000401-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**
    - Vm1Name: **az1000401-vm1**
    - Vm2Name: **az1000401-vm2**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1000401-vnet1**

    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 but proceed to the next task. You will use the network and the virtual machines included in this deployment in the second exercise of this lab.

8.0.3.2 Task 2: Create the second virtual network in the same region hosting a single Azure VM by using an Azure Resource Manager template

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Template deployment**.
3. Use the list of search results to navigate to the **Deploy a custom template** 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 **Labfiles\AZ100\Mod04\az-100-04_02_azuredeploy.json**.

   Note: Review the content of the template and note that it defines deployment of an Azure VM hosting Windows Server 2016 Datacenter.
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 **Labfiles\AZ100\Mod04\az-100-04_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 **az1000402-RG**
    - Location: the name of the Azure region which you selected in the previous task
    - Vm Size: **Standard_DS1_v2**
    - VmName: **az1000402-vm3**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1000402-vnet2**

    Note: Do not wait for the deployment to complete but proceed to the next task. You will use the network and the virtual machines included in this deployment in the second exercise of this lab.

Result: After you completed this exercise, you have created two Azure virtual networks and initiated deployments of three Azure VM by using Azure Resource Manager templates.

8.1 Exercise 1: Configure VNet peering

Estimated Time: 5 minutes

The main tasks for this exercise are as follows:

1. Configure VNet peering for the first virtual network
2. Configure VNet peering for the second virtual network

8.1.0.1 Task 1: Configure VNet peering for the first virtual network

1. In the Azure portal, navigate to the **az1000401-vnet1** virtual network blade.
2. From the **az1000401-vnet1** virtual network blade, display its **Peerings** blade.
3. From the **az1000401-vnet1 - Peerings** blade, create a VNet peering with the following settings:
   - Name: **az1000401-vnet1-to-az1000402-vnet2**
   - Virtual network deployment model: **Resource manager**
   - Subscription: the name of the Azure subscription you are using in this lab
   - Virtual network: **az1000402-vnet2**

- Allow virtual network access: **Enabled**
- Allow forwarded traffic: disabled
- Allow gateway transit: disabled
- Use remote gateways: disabled

8.1.0.2 Task 2: Configure VNet peering for the second virtual network

1. In the Azure portal, navigate to the **az1000402-vnet2** virtual network blade.
2. From the **az1000402-vnet2** virtual network blade, display its **Peerings** blade.
3. From the **az1000402-vnet2 - Peerings** blade, create a VNet peering with the following settings:
   - Name: **az1000402-vnet2-to-az1000401-vnet1**
   - Virtual network deployment model: **Resource manager**
   - Subscription: the name of the Azure subscription you are using in this lab
   - Virtual network: **az1000401-vnet1**
   - Allow virtual network access: **Enabled**
   - Allow forwarded traffic: disabled
   - Allow gateway transit: disabled
   - Use remote gateways: disabled

   Result: After you completed this exercise, you have configured virtual network peering between the two virtual networks.

8.2 Exercise 2: Implement custom routing

Estimated Time: 20 minutes

The main tasks for this exercise are as follows:

1. Enable IP forwarding for a network interface of an Azure VM
2. Configure user defined routing
3. Configure routing in an Azure VM running Windows Server 2016

8.2.0.1 Task 1: Enable IP forwarding for a network interface of an Azure VM

Note: Before you start this task, ensure that the template deployments you started in Exercise 0 have completed.

1. In the Azure portal, navigate to the blade of the second Azure VM **az1000401-vm2**.
2. From the **az1000401-vm2** blade, display its **Networking** blade.
3. From the **az1000401-vm2 - Networking** blade, display the blade of the network adapter (**az1000401-nic2**) of the Azure VM.
4. From the **az1000401-nic2** blade, display its **IP configurations** blade.
5. From the **az1000401-nic2 - IP configurations** blade, enable **IP forwarding**.

   Note: The Azure VM **az1000401-vm2**, which network interface you configured in this task, will function as a router, facilitating service chaining between the two virtual networks.

8.2.0.2 Task 2: Configure user defined routing

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Route table**.
3. Use the list of search results to navigate to the **Create route table** blade.
4. From the **Create route table** blade, create a new route table with the following settings:

- Name: **az1000402-rt1**
- Subscription: the name of the Azure subscription you use for this lab
- Resource group: **az1000402-RG**
- Location: the same Azure region in which you created the virtual networks
- BGP route propagation: **Disabled**

5. In the Azure portal, navigate to the **az1000402-rt1** blade.
6. From the **az1000402-rt1** blade, display its **Routes** blade.
7. From the **az1000402-rt1 - Routes** blade, add to the route table a route with the following settings:
   - Route name: **custom-route-to-az1000401-vnet1**
   - Address prefix: **10.104.0.0/16**
   - Next hop type: **Virtual appliance**
   - Next hop address: **10.104.1.4**

     Note: **10.104.1.4** is the IP address of the network interface of **az1000401-vm2**, which will provide service chaining between the two virtual networks.
8. From the **az1000402-rt1** blade, display its **Subnets** blade.
9. From the **az1000402-rt1 - Subnets** blade, associate the route table **az1000402-rt1** with **subnet0** of **az1000402-vnet2**.

8.2.0.3 Task 3: Configure routing in an Azure VM running Windows Server 2016

1. In the Azure portal, navigate to the blade of the **az1000401-vm2** Azure VM.
2. From the **Overview** pane of the **az1000401-vm2** blade, generate an RDP file and use it to connect to **az1000401-vm2**.
3. When prompted, authenticate by specifying the following credentials:
   - User name: **Student**
   - Password: **Pa55w.rd1234**
4. Within the Remote Desktop session to **az1000401-vm2**, from **Server Manager**, use the **Add Roles and Features Wizard** to add the **Remote Access** server role with the **Routing** role service and all required features.

   Note: If you receive an error message **There may be a version mismatch between this computer and the destination server or VHD** once you select the **Remote Access** checkbox on the **Server Roles** page of the **Add Roles and Features Wizard**, clear the checkbox, click **Next**, click **Previous** and select the **Remote Access** checkbox again.
5. Within the Remote Desktop session to **az1000401-vm2**, from Server Manager, start the **Routing and Remote Access** console.
6. In the **Routing and Remote Access** console, run **Routing and Remote Access Server Setup Wizard**, use the **Custom configuration** option, enable **LAN routing**, and start **Routing and Remote Access** service.
7. Within the Remote Desktop session to **az1000401-vm2**, start the **Windows Firewall with Advanced Security** console and enable **File and Printer Sharing (Echo Request - ICMPv4-In)** inbound rule for all profiles.

   Result: After completing this exercise, you have implemented custom routing between peered Azure virtual networks.

8.3 Exercise 3: Validating service chaining

Estimated Time: 10 minutes

The main tasks for this exercise are as follows:

1. Configure Windows Firewall with Advanced Security on the target Azure VM
2. Test service chaining between peered virtual networks

8.3.0.1 Task 1: Configure Windows Firewall with Advanced Security on the target Azure VM

1. In the Azure portal, navigate to the blade of the **az1000401-vm1** Azure VM.
2. From the **Overview** pane of the **az1000401-vm1** blade, generate an RDP file and use it to connect to **az1000401-vm1**.
3. When prompted, authenticate by specifying the following credentials:
   - User name: **Student**
   - Password: **Pa55w.rd1234**
4. Within the Remote Desktop session to **az1000401-vm1**, open the **Windows Firewall with Advanced Security** console and enable **File and Printer Sharing (Echo Request - ICMPv4-In)** inbound rule for all profiles.

8.3.0.2 Task 2: Test service chaining between peered virtual networks

1. In the Azure portal, navigate to the blade of the **az1000402-vm3** Azure VM.
2. From the **Overview** pane of the **az1000402-vm3** blade, generate an RDP file and use it to connect to **az1000402-vm3**.
3. When prompted, authenticate by specifying the following credentials:
   - User name: **Student**
   - Password: **Pa55w.rd1234**
4. Once you are connected to **az1-1000402-vm3** via the Remote Desktop session, start **Windows PowerShell**.
5. In the **Windows PowerShell** window, run the following:

   ```
   Test-NetConnection -ComputerName 10.104.0.4 -TraceRoute
   ```

   Note: **10.104.0.4** is the IP address of the network interface of the first Azure VM **az1000401-vm1**
6. Verify that test is successful and note that the connection was routed over **10.104.1.4**

   Note: Without custom routing in place, the traffic would flow directly between the two Azure VMs.

   Result: After you completed this exercise, you have validated service chaining between peered Azure virtual networks. # AZ 100 Module 5 - Implement and Manage Hybrid Identities

9 Lab 2: Manage Azure AD Premium tenants

Estimated Time: 90 minutes

All tasks in this lab are performed from the Azure portal

Lab files: none

9.0.1 Scenario

Adatum Corporation wants to take advantage of Azure AD Premium features

9.0.2 Objectives

After completing this lab, you will be able to:

- Manage Azure AD users and groups
- Manage Azure AD-integrated SaaS applications

9.1 Exercise 1: Manage Azure AD users and groups

Estimated Time: 55 minutes

The main tasks for this exercise are as follows:

1. Create a new Azure AD tenant
2. Activate Azure AD Premium v2 trial
3. Create and configure Azure AD users
4. Assign Azure AD Premium v2 licenses to Azure AD users
5. Manage Azure AD group membership
6. Configure self-service password reset functionality
7. Validate self-service password reset functionality

9.1.0.1 Task 1: Create a new Azure AD tenant

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 **New** blade.
3. From the **New** blade, search Azure Marketplace for **Azure Active Directory**.
4. Use the list of search results to navigate to the **Create directory** blade.
5. From the **Create directory** blade, create a new Azure AD tenant with the following settings:

- Organization name: **AdatumLab100-5b**
- Initial domain name: a unique name consisting of a combination of letters and digits.
- Country or region: **United States**

Note: Take a note of the initial domain name. You will need it later in this lab.

9.1.0.2 Task 2: Activate Azure AD Premium v2 trial

1. In the Azure portal, set the **Directory + subscription** filter to the newly created Azure AD tenant.

   Note: The **Directory + subscription** filter appears to the right of the Cloud Shell icon in the toolbar of the Azure portal

   Note: You might need to refresh the browser window if the **AdatumLab100-5b** entry does not appear in the **Directory + subscription** filter list.
2. In the Azure portal, navigate to the **AdatumLab100-5b - Overview** blade.
3. From the **AdatumLab100-5b - Overview** blade, navigate to the **Licenses - Overview** blade.
4. From the **Licenses - Overview** blade, navigate to the **Products** blade.
5. From the **Products** blade, navigate to the **Activate** blade and activate **Azure AD Premium P2** free trial.

9.1.0.3 Task 3: Create and configure Azure AD users

1. In the Azure portal, navigate to the **Users - All users** blade of the AdatumLab100-5b Azure AD tenant.
2. From the **Users - All users** blade, create a new user with the following settings:
   - Name: **aaduser1**
   - User name: **aaduser1@**.onmicrosoft.com** where represents the initial domain name you specified in the first task of this exercise.

     Note: Take a note of this user name. You will need it later in this lab.
   - Profile:

- Department: **Sales**

- Properties: **Default**
- Groups: **0 groups selected**
- Directory role: **User**
- Password: select the checkbox **Show Password** and note the string appearing in the **Password** text box. You will need it later in this lab.

3. From the **Users - All users** blade, create a new user with the following settings:
   - Name: **aaduser2**
   - User name: **aaduser2@**.onmicrosoft.com** where represents the initial domain name you specified in the first task of this exercise.

     Note: Take a note of this user name. You will need it later in this lab.
   - Profile:
     - Department: **Finance**
   - Properties: **Default**
   - Groups: **0 groups selected**
   - Directory role: **User**
   - Password: select the checkbox **Show Password** and note the string appearing in the **Password** text box. You will need it later in this lab.

9.1.0.4 Task 4: Assign Azure AD Premium v2 licenses to Azure AD users

Note: In order to assign Azure AD Premium v2 licenses to Azure AD users, you first have to set their location attribute.

1. From the **Users - All users** blade, navigate to the **aaduser1 - Profile** blade and set the **Usage location** to **United States**.
2. From the **aaduser1 - Profile** blade, navigate to the **aaduser1 - Licenses** blade and assign to the user an Azure Active Directory Premium P2 license with all licensing options enabled.
3. Return to the **Users - All users** blade, navigate to the **aaduser2 - Profile** blade, and set the **Usage location** to **United States**.
4. From the **aaduser2 - Profile** blade, navigate to the **aaduser2 - Licenses** blade and assign to the user an Azure Active Directory Premium P2 license with all licensing options enabled.
5. Return to the **Users - All users** blade, navigate to the Profile entry of your user account and set the **Usage location** to **United States**.
6. Navigate to **Licenses** blade of your user account and assign to it an Azure Active Directory Premium P2 license with all licensing options enabled.
7. Sign out from the portal and sign back in using the same account you are using for this lab.

   Note: This step is necessary in order for the license assignment to take effect.

9.1.0.5 Task 5: Manage Azure AD group membership

1. In the Azure portal, navigate to the **Groups - All groups** blade.
2. From the **Groups - All groups** blade, navigate to the **Group** blade and create a new group with the following settings:
   - Group type: **Security**
   - Group name: **Sales**
   - Group description: **All users in the Sales department**
   - Membership type: **Dynamic User**

- Dynamic user members:
  - Simple rule
  - Add users where: **department Equals Sales**

3. From the **Groups - All groups** blade, navigate to the **Group** blade and create a new group with the following settings:
   - Group type: **Security**
   - Group name: **Sales and Finance**
   - Group description: **All users in the Sales and Finance departments**
   - Membership type: **Dynamic User**
   - Dynamic user members:
     - Advanced rule: **(user.department -eq "Sales") -or (user.department -eq "Finance")**
4. From the **Groups - All groups** blade, navigate to the blades of **Sales** and **Sales and Finance** groups, and note that the group membership evaluation is in progress. Wait until the evalution completes, then navigate to the **Members** blade, and verify that the group membership is correct.

9.1.0.6 Task 6: Configure self-service password reset functionality

1. In the Azure portal, navigate to the **AdatumLab100-5b - Overview** blade.
2. From the **AdatumLab100-5b - Overview** blade, navigate to the **Password reset - Properties** blade.
3. On the **Password reset - Properties** blade, configure the following settings:
   - Self service password reset enabled: **Selected**
   - Selected group: **Sales**
4. From the **Password reset - Properties** blade, navigate to the **Password reset - Auhentication methods** blade and configure the following settings:
   - Number of methods required to reset: **1**
   - Methods available to users:
     - **Email**
     - **Mobile phone**
     - **Office phone**
     - **Security questions**
     - Number of security questions required to register: **3**
     - Number of security questions required to reset: **3**
     - Select security questions: select **Predefined** and add any combination of 5 predefined security questions
5. From the **Password reset - Authentication methods** blade, navigate to the **Password reset - Registration** blade, and ensure that the following settings are configured:
   - Require users to register when signing in?: **Yes**
   - Number of days before users are asked to re-confirm their authentication information: **180**

9.1.0.7 Task 7: Validate self-service password reset functionality

1. Open an InPrivate Microsoft Edge window.
2. In the new browser window, navigate to the Azure portal and sign in using the **aaduser1** user account. When prompted, change the password to a new value.

   Note: You will need to provide a fully qualified name of the **aaduser1** user account, including the Azure AD tenant DNS domain name, as noted earlier in this lab.

3. When prompted with the **More information required** message, continue to the **don't lose access to your account** page.
4. On the **don't lose access to your account** page, note that you need to set up at least one of the following options:
   - **Office phone**
   - **Authentication Phone**
   - **Authentication Email**
   - **Security Questions**
5. From the **don't lose access to your account** page, configure answers to 5 security questions you selected in the previous task
6. Verify that you successfully signed in to the Azure portal.
7. Sign out as **aaduser1** and close the InPrivate browser window.
8. Open an InPrivate Microsoft Edge window.
9. In the new browser window, navigate to the Azure portal and, on the **Pick an account** page, type in the **aaduser1** user account name.
10. On the **Enter password** page, click the **Forgot my password** link.
11. On the **Get back into your account** page, verify the **User ID**, enter the characters in the picture or the words in the audio, and proceed to the next page.
12. On the next page, provide answers to thre security questions using answers you specified in the previous task.
13. On the next page, enter twice a new password and complete the password reset process.
14. Verify that you can sign in to the Azure portal by using the newly reset password.

    Result: After you completed this exercise, you have created a new Azure AD tenant, activated Azure AD Premium v2 trial, created and configured Azure AD users, assigned Azure AD Premium v2 licenses to Azure AD users, managed Azure AD group membership, as well as configured and validated self-service password reset functionality

9.2 Exercise 2: Manage Azure AD-integrated SaaS applications

Estimated Time: 35 minutes

The main tasks for this exercise are as follows:

1. Add an application from the Azure AD gallery
2. Configure the application for a single sign-on
3. Assign users to the application
4. Validate single sign-on for the application

9.2.0.1 Task 1: Add an application from the Azure AD gallery

1. In the Azure portal, navigate to the **AdatumLab100-5b - Overview** blade.
2. From the **AdatumLab100-5b - Overview** blade, navigate to the **Enterprise applications - All applications** blade.
3. From the **Enterprise applications - All applications** blade, navigate to the **Add an application** blade.
4. On the **Add an application** blade, search the application gallery for the **Microsoft OneDrive**.
5. Use the list of search results to navigate to the **Microsoft OneDrive** add app blade and add the app.

9.2.0.2 Task 2: Configure the application for a single sign-on

1. From the **Microsoft OneDrive - Overview** blade, navigate to the **Microsoft OneDrive - Getting started** blade.
2. On the **Microsoft OneDrive - Getting started** blade, use the **Configure single sign-on (required)** option to navigate to the **Microsoft OneDrive - Single sign-on** blade.
3. On the **Microsoft OneDrive - Single sign-on** blade, select the **Password-based** option and save the configuration.

9.2.0.3 Task 3: Assign users to the application

1. Navigate back to the **Microsoft OneDrive - Getting started** blade.
2. On the **Microsoft OneDrive - Getting started** blade, use the **Assign a user for testing (required)** option to navigate to the **Users and groups** blade for **Microsoft OneDrive**.
3. From the **Users and groups** blade for **Microsoft OneDrive**, navigate to the **Add Assignment** blade and add the following assignment:
   - Users and groups: **Sales and Finance**
   - Select role: **Default access**
   - Assign Credentials:
     - Assign credentials to be shared among all group members: **Yes**
     - Email Address: the name of the Microsoft Account you are using for this lab
     - Password: the password of the Microsoft Account you are using for this lab
4. Sign out from the Azure portal and close the Microsoft Edge window.

9.2.0.4 Task 4: Validate single sign-on for the application

1. Open a Microsoft Edge window.
2. In the Microsoft Edge window, navigate to the Application Access Panel at **http://myapps.microsoft.com** and sign in by using the **aaduser2** user account. When prompted, change the password to a new value.

   > **Note**: You will need to provide a fully qualified name of the **aaduser2** user account, including the Azure AD tenant DNS domain name, as noted earlier in this lab.
3. On the Access Panel Applications page, click the **Microsoft OneDrive** icon.
4. When prompted, add the My Apps Secure Sign-in Extension and enable it, including the **Allow for InPrivate browsing** option.
5. Navigate again to the Application Access Panel at **http://myapps.microsoft.com** and sign in by using the **aaduser2** user account.
6. On the Access Panel Applications page, click the **Microsoft OneDrive** icon.
7. Verify that you have successfully accessed the Microsoft OneDrive application without having to re-authenticate.
8. Sign out from the Application Access Panel and close the Microsoft Edge window.

   > **Note**: Make sure to launch Microsoft Edge again, browse to the Azure portal, sign in by using the Microsoft account that has the Owner role in the Azure subscription you were using in this lab, and use the **Directory + subscription** filter to switch to your default Azure AD tenant once you complete this lab.

Result: After you completed this exercise, you have added an application from the Azure AD gallery, configured the application for a single sign-on, assigned users to the application, and validated single sign-on for the application. # AZ 100 Module 5 - Implement and Manage Hybrid Identities

10 Lab: Implement Directory Synchronization

Estimated Time: 60 minutes

All tasks in this lab are performed from the Azure portal (including a PowerShell Cloud Shell session) except for Excercise 3 Task 1, Exercise 3 Task 2, and Exercise 3 Task 3, which include steps performed from a Remote Desktop session to an Azure VM

> **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?view=azps-1.2.0

Lab files: none

10.0.1 Scenario

Adatum Corporation wants to integrate its Active Directory with Azure Active Directory

10.0.2 Objectives

After completing this lab, you will be able to:

- Deploy an Azure VM hosting an Active Directory domain controller
- Create and configure an Azure Active Directory tenant
- Synchronize Active Directory forest with an Azure Active Directory tenant

10.1 Exercise 1: Deploy an Azure VM hosting an Active Directory domain controller

Estimated Time: 25 minutes

The main tasks for this exercise are as follows:

1. Identify an available DNS name for an Azure VM deployment
2. Deploy an Azure VM hosting an Active Directory domain controller by using an Azure Resource Manager template

10.1.0.1 Task 1: Identify an available DNS name for an Azure VM deployment

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 and is a Global Administrator of the Azure AD tenant associated with that subscription.
2. From the Azure Portal, start a PowerShell session in the Cloud Shell pane.

   > **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.

3. In the Cloud Shell pane, run the following command, substituting the placeholder `<custom-label>` with any string which is likely to be unique and the placeholder `<location>` with the name of the Azure region into which you want to deploy the Azure VM that will host an Active Directory domain controller.

   > **Note**: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**

   ```
   Test-AzDnsAvailability -DomainNameLabel <custom-label> -Location '<location>'
   ```

4. Verify that the command returned **True**. If not, rerun the same command with a different value of the `<custom-label>` until the command returns **True**.
5. Note the value of the `<custom-label>` that resulted in the successful outcome. You will need it in the next task

10.1.0.2 Task 2: Deploy an Azure VM hosting an Active Directory domain controller by using an Azure Resource Manager template

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Template deployment**.
3. Use the list of search results to navigate to the **Deploy a custom template** blade.
4. On the **Custom deployment** blade, in the **Load a GitHub quickstart template** drop-down list and select the **active-directory-new-domain** entry.
5. On the **Create an Azure VM with a new AD Forest** 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 **az1000501-RG**
   - Location: the name of the Azure region which you used in the previous task
   - Admin Username: **Student**
   - Admin Password: **Pa55w.rd1234**
   - Domain Name: **adatum.com**
   - Dns Prefix: the `<custom-label>` you identifed in the previous task
   - _artifacts Location: accept the default value
   - _artifacts Location Sas Token: leave blank
   - Location: accept the default value

     Note: Do not wait for the deployment to complete but proceed to the next exercise. You will use the virtual machine deployed in this task in the third exercise of this lab.

   Result: After you completed this exercise, you have initiated deployment of an Azure VM that will host an Active Directory domain controller by using an Azure Resource Manager template

10.2 Exercise 2: Create and configure an Azure Active Directory tenant

Estimated Time: 10 minutes

The main tasks for this exercise are as follows:

1. Create an Azure Active Directory (AD) tenant
2. Add a custom DNS name to the new Azure AD tenant
3. Create an Azure AD user with the Global Administrator role

10.2.0.1 Task 1: Create an Azure Active Directory (AD) tenant

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Azure Active Directory**.
3. Use the list of search results to navigate to the **Create directory** blade.
4. From the **Create directory** blade, create a new Azure AD tenant with the following settings:

- Organization name: **AdatumSync**
- Initial domain name: a unique name consisting of a combination of letters and digits.
- Country or region: **United States**

  Note: The green check mark in the **Initial domain name** text box will indicate whether the domain name you typed in is valid and unique.

10.2.0.2 Task 2: Add a custom DNS name to the new Azure AD tenant

1. In the Azure portal, set the **Directory + subscription** filter to the newly created Azure AD tenant.

   Note: The **Directory + subscription** filter appears to the left of the notification icon in the toolbar of the Azure portal

   Note: You might need to refresh the browser window if the **AdatumSync** entry does not appear in the **Directory + subscription** filter list.

2. In the Azure portal, navigate to the **AdatumSync - Overview** blade.
3. From the **AdatumSync - Overview** blade, display the **AdatumSync - Custom domain names** blade.
4. On the **AdatumSync - Custom domain names** blade, identify the primary, default DNS domain name associated with the Azure AD tenant. Note its value - you will need it in the next task.
5. From the **AdatumSync - Custom domain names** blade, add the **adatum.com** custom domain.
6. On the **adatum.com** blade, review the information necessary to perform verification of the Azure AD domain name.

   Note: You will not be able to complete the validation process because you do not own the **adatum.com** DNS domain name. This will not prevent you from synchronizing the **adatum.com** Active Directory domain with the Azure AD tenant. You will use for this purpose the default primary DNS name of the Azure AD tenant (the name ending with the **onmicrosoft.com** suffix), which you identified earlier in this task. However, keep in mind that, as a result, the DNS domain name of the Active Directory domain and the DNS name of the Azure AD tenant will differ. This means that Adatum users will need to use different names when signing in to the Active Directory domain and when signing in to Azure AD tenant.

10.2.0.3 Task 3: Create an Azure AD user with the Global Administrator role

1. In the Azure portal, navigate to the **Users - All users** blade of the **AdatumSync** Azure AD tenant.
2. From the **Users - All users** blade, create a new user with the following settings:
   - Name: **syncadmin**
   - User name: **syncadmin@** where represents the default primary DNS domain name you identified in the previous task. Take a note of this user name. You will need it later in this lab.
   - Profile: **Not configured**
   - Properties: **Default**
   - Groups: **0 groups selected**
   - Directory role: **Global administrator**
   - Password: select the checkbox **Show Password** and note the string appearing in the **Password** text box. You will need it later in this task.

     Note: An Azure AD user with the Global Administrator role is required in order to implement Azure AD Connect.
3. Open an InPrivate Microsoft Edge window.
4. In the new browser window, navigate to the Azure portal and sign in using the **syncadmin** user account. When prompted, change the password to a new value.

   Note: You will need to provide the fully qualified name of the **syncadmin** user account, including the Azure AD tenant DNS domain name.

5. Sign out as **syncadmin** and close the InPrivate browser window.

Result: After you completed this exercise, you have created an Azure AD tenant, added a custom DNS name to the new Azure AD tenant, and created an Azure AD user with the Global Administrator role.

10.3 Exercise 3: Synchronize Active Directory forest with an Azure Active Directory tenant

Estimated Time: 35 minutes

The main tasks for this exercise are as follows:

1. Configure Active Directory in preparation for directory synchronization
2. Install Azure AD Connect
3. Verify directory synchronization

10.3.0.1 Task 1: Configure Active Directory in preparation for directory synchronization

Note: Before you start this task, ensure that the template deployment you started in Exercise 1 has completed.

1. In the Azure portal, set the **Directory + subscription** filter back to the Azure AD tenant associated with the Azure subscription you used in the first exercise of this lab.

   Note: The **Directory + subscription** filter appears to the left of the notification icon in the toolbar of the Azure portal

2. In the Azure portal, navigate to the **adVM** blade, displaying the properties of the Azure VM hosting an Active Directory domain controller that you deployed in the first exercise of this lab.
3. From the **Overview** pane of the **adVM** blade, generate an RDP file and use it to connect to **adVM**.
4. When prompted, authenticate by specifying the following credentials:
   - User name: **Student**
   - Password: **Pa55w.rd1234**
5. Within the Remote Desktop session to **adVM**, open the **Active Directory Administrative Center**.
6. From **Active Directory Administrative Center**, create a root level organizational unit named **ToSync**.
7. From **Active Directory Administrative Center**, in the organizational unit **ToSync**, create a new user account with the following settings:
   - Full name: **aduser1**
   - User UPN logon: **aduser1@adatum.com**
   - User SamAccountName logon: **adatum\aduser1**
   - Password: **Pa55w.rd1234**
   - Other password options: **Password never expires**

10.3.0.2 Task 2: Install Azure AD Connect

1. Within the RDP session to **adVM**, from Server Manager, disable temporarily **IE Enhanced Security Configuration**.
2. Within the RDP session to **adVM**, start Internet Explorer and download **Azure AD Connect** from **https://www.microsoft.com/en-us/download/details.aspx?id=47594**
3. Start **Microsoft Azure Active Directory Connect** wizard, accept the licensing terms, and, on the **Express Settings** page, select the **Customize** option.
4. On the **Install required components** page, leave all optional configuration options deselected and start the installation.
5. On the **User sign-in** page, ensure that only the **Password Hash Synchronization** is enabled.
6. When prompted to connect to Azure AD, authenticate by using the credentials of the **syncadmin** account you created in the previous exercise.
7. When prompted to connect your directories, add the **adatum.com** forest, choose the option to **Create new AD account**, and authenticate by using the following credentials:

- User name: **ADATUM\Student**
- Password: **Pa55w.rd1234**

8. On the **Azure AD sign-in configuration** page, note the warning stating **Users will not be able to sign-in to Azure AD with on-premises credentials if the UPN suffix does not match a verified domain name** and enable the checkbox **Continue without matching all UPN suffixes to verified domain**.

   Note: As explained earlier, this is expected, since you could not verify the custom Azure AD DNS domain **adatum.com**.

9. On the **Domain and OU filtering** page, ensure that only the **ToSync** OU is selected.
10. On the **Uniquely identifying your users** page, accept the default settings.
11. On the **Filter users and devices** page, accept the default settings.
12. On the **Optional features** page, accept the default settings.
13. On the **Ready to configure** page, ensure that the **Start the synchronization process when configuration completes** checkbox is selected and continue with the installation process.

    Note: Installation should take about 2 minutes.

14. Close the Microsoft Azure Active Directory Connect window once the configuration is completed.

10.3.0.3 Task 3: Verify directory synchronization

1. Within the RDP session to **adVM**, start Internet Explorer, browse to the Azure portal at **http://portal.azure.com** and sign in by using the **syncadmin** account that you created in the previous exercise.
2. In the Azure portal, navigate to the **AdatumSync - Overview** blade.
3. From the **AdatumSync - Overview** blade, display the **Users - All users** blade of the AdatumSync Azure AD tenant.
4. On the **Users - All users** blade, note that the list of user objects includes the **aduser1** account, with the **Windows Server AD** appearing in the **SOURCE** column.
5. From the **Users - All users** blade, display the **aduser1 - Profile** blade. Note that the **Department** attribute is not set.
6. Within the RDP session to **adVM**, switch to the **Active Directory Administrative Center**, open the window displaying properties of the **aduser1** user account, and set the value of its **Department** attribute to **Sales**.
7. Within the RDP session to **adVM**, start **Windows PowerShell** as Administrator.
8. From the Windows PowerShell prompt, start Azure AD Connect delta synchronization by running the following:

   ```
   Start-ADSyncSyncCycle -PolicyType Delta
   ```

9. Within the RDP session to **adVM**, switch to the Internet Explorer window displaying the Azure portal.
10. In the Azure portal, navigate back to the **Users - All users** blade and refresh the page.
11. From the **Users - All users** blade, display the **aduser1 - Profile** blade. Note that the **Department** attribute is now set to **Sales**.

    Note: You might need to wait for another minute and refresh the page again if the **Department** attribute remains not set.

    Result: After you completed this exercise, you have configured Active Directory in preparation for directory synchronization, installed Azure AD Connect, and verified directory synchronization.