Azure in a Day

Lab Workbook

Table of Contents

Lab 1

Creating a Resource Group

Introduction

Description

This lab introduces the concept of a resource group in Azure, which acts as a container for managing all related services and assets. You will create a resource group that will house the resources provisioned in subsequent labs. It must be created first so all other resources can be assigned to it.

Lab Objectives

- Organize cloud resources by creating a resource group
- Select an appropriate Azure region for deployment

Create a Resource Group

1. Navigate to the Azure Portal (https://portal.azure.com/) and login.

2. Search for “**Resource groups**” in the top search bar.

3. Click **+ Create**.

4. Set the following:

   a. **Subscription:** Your active subscription

   b. **Resource group name**: rg-azure-labs-{your-initials-here}

   c. **Region:** East US (or region closest to you)

5. Click **Review + create > Create**.

Verification

1. Confirm the resource appears in the **Resource groups** blade of the Azure Portal.
2. Ensure the resource group shows your specified region and has no deployment errors.
3. If the resource doesn’t appear, try refreshing the Azure Portal or check the region filter.

Lab 2

Creating a Storage Account

Introduction

Description

In this lab, you will create a storage account and enable static website hosting. This service will host a simple HTML-based website that can be publicly accessed via a URL. You will upload both an index.html file and a 404.html file to demonstrate default routing and error handling behavior.

Lab Objectives

- Provision a storage account in Azure
- Enable static website hosting and upload HTML files

Create a Storage Account

1. Search for “**Storage accounts**” in the top search bar of the Azure Portal.

2. Click **+ Create.**

3. Set the following:

   a. **Resource group:** rg-azure-labs-{**your-initials-here}**

   b. **Storage account name:** Must be globally unique (e.g. mystaticwebsite1234)

   c. **Region:** Same as Resource Group

   d. **Primary service:** Azure Blob Storage

   e. **Performance:** Standard

   f. **Redundancy:** Locally redundant storage (LRS)

4. Click **Review + create > Create**.

Configure Static Hosting

1. After deployment, navigate to the storage account.

2. In the left menu, select **Data management > Static website.**

3. Click Enable, then set:

    a. **Index document name:** index.html

    b. **Error document path:** 404.html

4. Click **Save**.

Upload and Serve Web Content

1. In the left menu, select **Overview**

2. Click **Upload**

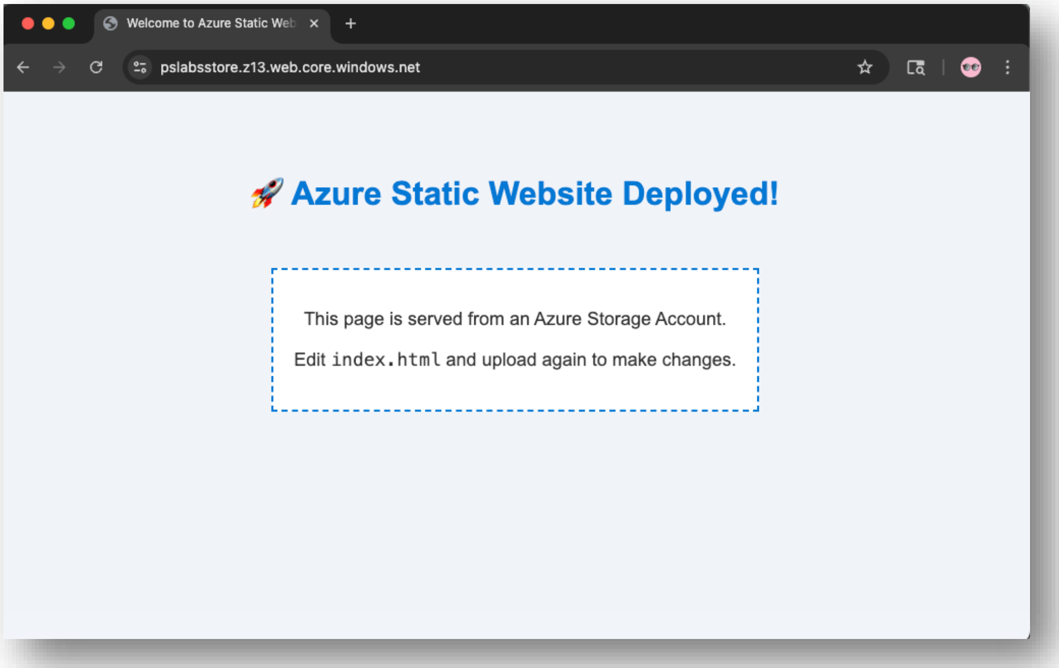
3. Under containers, select **$web**.

4. Browse to or drag and drop the **index.html** and **404.html** files from the **code-samples** folder in the courseware directory and click **Upload**

5. Go back to the **Static website** blade to find the **public primary endpoint URL**.

Verification

1. Open the Primary endpoint URL in your browser and verify the static HTML page loads correctly.

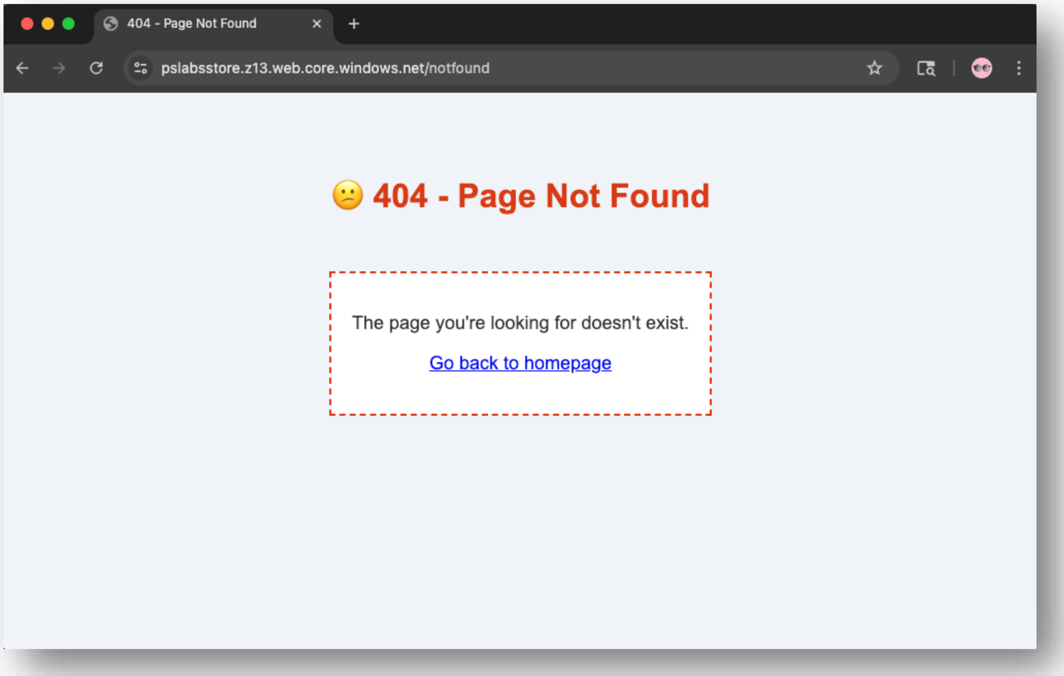

Welcome to Azure Static Web
pslabsstore.z13.web.core.windows.net

🚀 Azure Static Website Deployed!

This page is served from an Azure Storage Account.

Edit `index.html` and upload again to make changes.

2. Try navigating to a non-existent subpath (e.g. /notfound) and confirm the 404.html page appears.



😕 404 - Page Not Found

The page you're looking for doesn't exist.

Go back to homepage

3. If the page doesn’t load:

- Ensure the index.html and 404.html files are uploaded to the $web container
- Confirm static hosting is enabled and saved

Lab 3

Creating a SQL Server and Database

Introduction

Description

In this lab, you will provision an Azure SQL Server and a database to serve as the backend for the application you'll deploy later. You will use the Query Editor to run a SQL script (provided in the courseware) that creates a Products table and inserts test data. This database will be queried by the web app you will deploy in Lab 4.

Lab Objectives

- Deploy an Azure SQL Server and database
- Configure firewall settings to allow client access
- Use Query Editor to create a table and seed test data

Create a SQL Server and Database

1. Search for "**SQL databases**" in the top search bar of the Azure Portal.

2. Click **+ Create**.

3. Set the following:

   a. **Resource group:** {your-resource-group}

   b. **Database name:** ProductsDB

   c. **Server:** Click **Create new:**

      - **Server name:** sql-server-lab-{your-initials} *(must be unique)*
      - **Location:** Same as resource group
      - **Authentication method**: Use SQL authentication
      - **Server admin login**: sqladmin
      - **Password**: P@ssword123! (or other, just remember it!)

   d. **Workload environment:** Development

   e. **Compute + Storage:** General Purpose - Serverless

   f. **Backup storage redundancy:** Locally-redundant backup storage

4. Click **Review + create > Create**.

Configure Firewall

1. Navigate to the SQL Server resource (**sql-server-lab-{your-initials}**, not the database).
2. In the left menu, click **Security > Networking**.
3. Enable **“Selected networks”** under public network access.
4. Click **“Add your client IPv4 address (your.ip.address.here)”** under **Firewall rules**.
5. Check the box for **“Allow Azure services and resources to access this server”** under **Exceptions**.
6. Click **Save**.

Use Query Editor to Create and Seed a Table

1. Navigate to the **ProductsDB** resource

2. In the left menu, click **Query editor (preview)**

3. Login using the server admin username and password configured in the prior step

4. In the editor, paste the contents of the **script.sql** file from the **code-samples** folder in the courseware directory and click **Run**.

script.sql

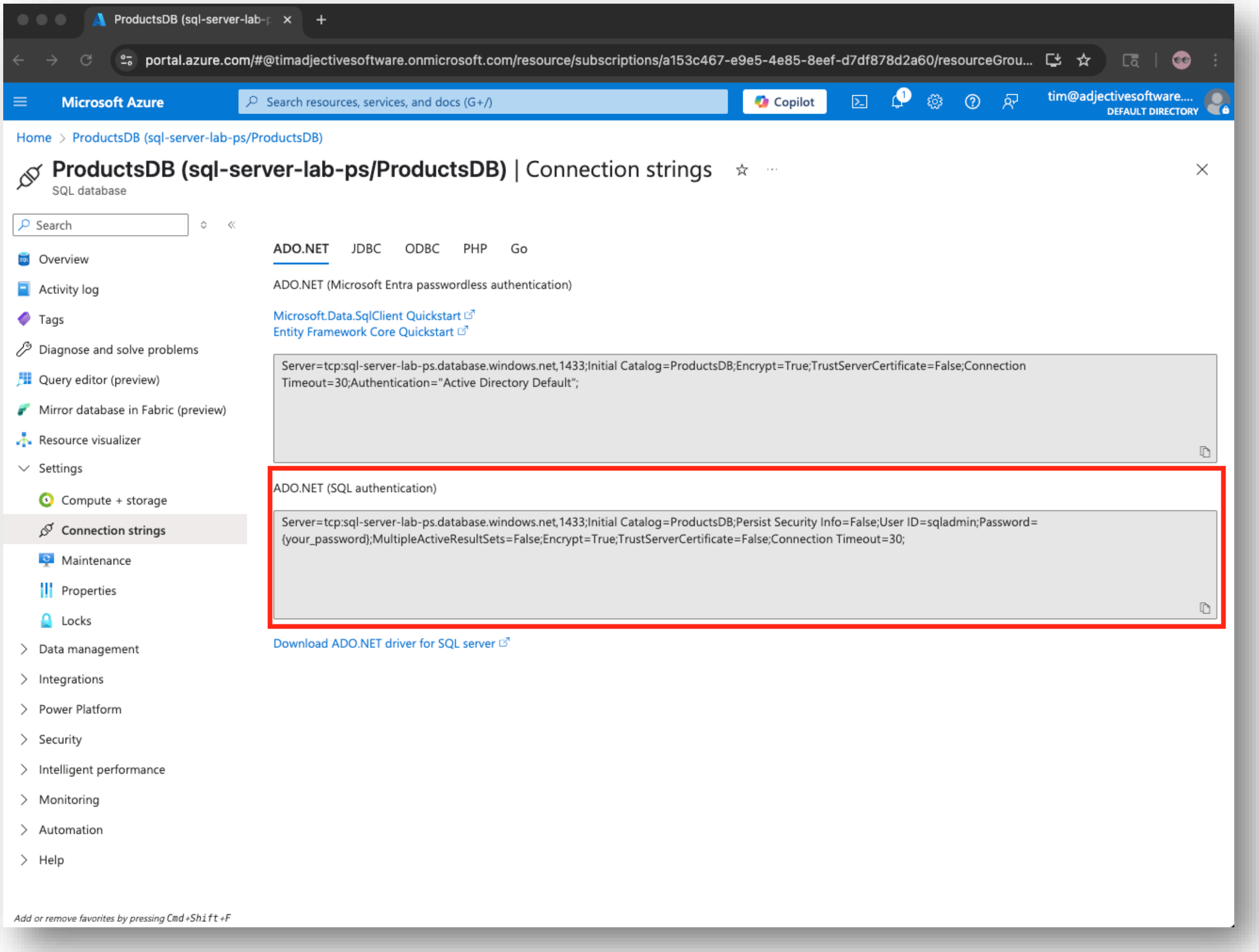
```
CREATE TABLE Products (
    Id INT PRIMARY KEY
IDENTITY(1,1),
    Name NVARCHAR(50),
    Price DECIMAL(10,2)
);

INSERT INTO Products (Name,
Price)
VALUES
  ('Azure Hat', 14.99),
  ('Cloud Mug', 9.49),
  ('Dev Hoodie', 39.95),
  ('Laptop Sticker Pack', 4.99),
  ('Notebook', 6.75);

SELECT * FROM Products;
```

Access Database Connection String

1. Navigate to the **ProductsDB** resource.

2. In the left menu, click Settings > Connecting strings.

3. Copy the **ADO.NET (SQL authentication)** connection string. You will need it in the next lab.


ProductsDB (sql-server-lab-p
portal.azure.com/#@timadjectivesoftware.onmicrosoft.com/resource/subscriptions/a153c467-e9e5-4e85-8eef-d7df878d2a60/resourceGrou...
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
tim@adjectivesoftware....
DEFAULT DIRECTORY
Home > ProductsDB (sql-server-lab-ps/ProductsDB)
ProductsDB (sql-server-lab-ps/ProductsDB) | Connection strings
SQL database
Search
Overview
Activity log
Tags
Diagnose and solve problems
Query editor (preview)
Mirror database in Fabric (preview)
Resource visualizer
Settings
Compute + storage
Connection strings
Maintenance
Properties
Locks
Data management
Integrations
Power Platform
Security
Intelligent performance
Monitoring
Automation
Help
Add or remove favorites by pressing Cmd+Shift+F
ADO.NET
JDBC
ODBC
PHP
Go
Microsoft.Data.SqlClient Quickstart
Entity Framework Core Quickstart
Download ADO.NET driver for SQL server

ADO.NET (Microsoft Entra passwordless authentication)

```
Server=tcp:sql-server-lab-ps.database.windows.net,1433;Initial Catalog=ProductsDB;Encrypt=True;TrustServerCertificate=False;Connection
Timeout=30;Authentication="Active Directory Default";
```

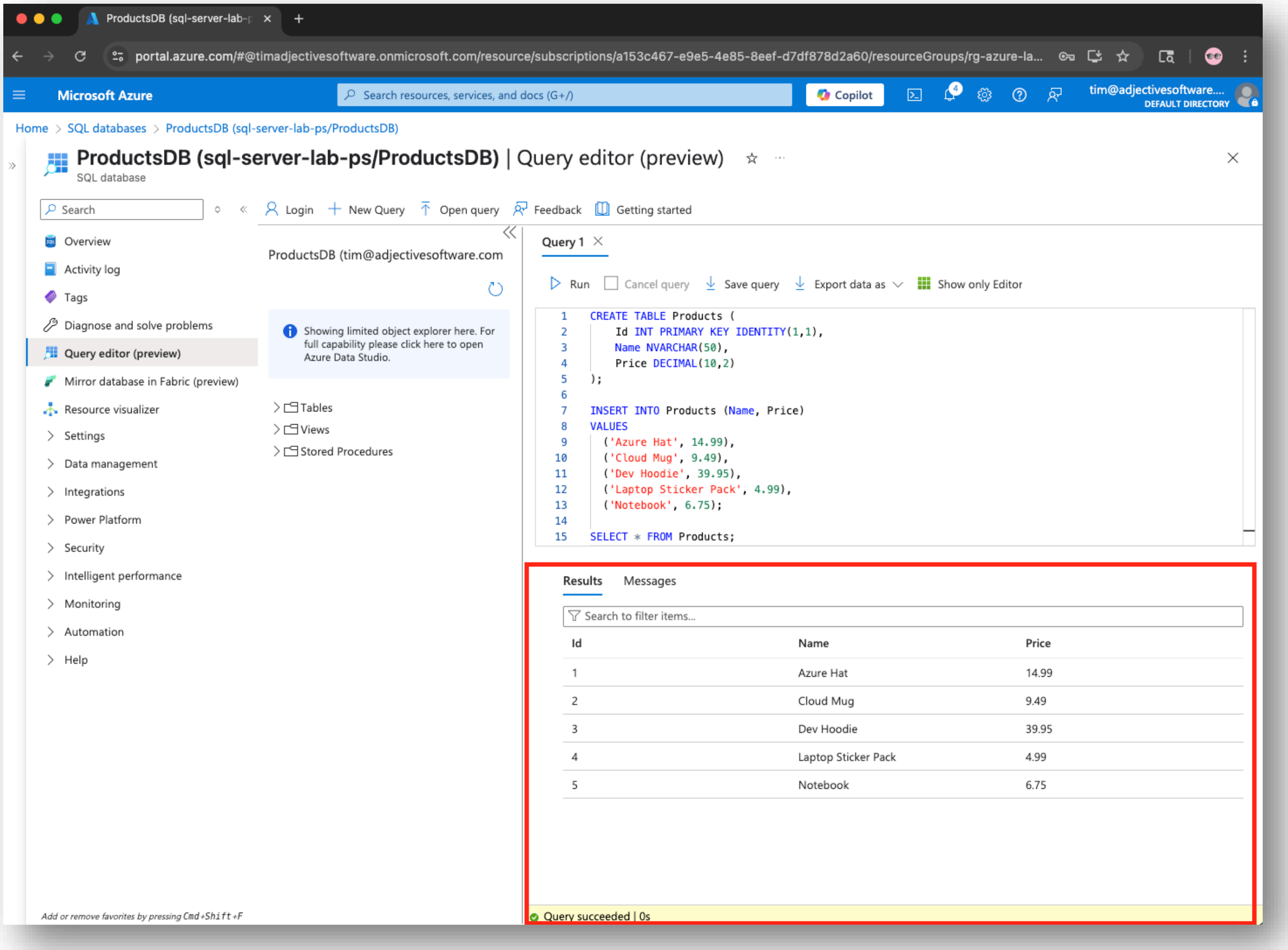
ADO.NET (SQL authentication)

```
Server=tcp:sql-server-lab-ps.database.windows.net,1433;Initial Catalog=ProductsDB;Persist Security Info=False;User ID=sqladmin;Password=
{your_password};MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;
```

- *Note: You will need to paste your SQL Server password into the connection string* ***(replace {your_password}***

Verification

1. Confirm the sample rows appear in the results.


ProductsDB (sql-server-lab-
portal.azure.com/#@timadjectivesoftware.onmicrosoft.com/resource/subscriptions/a153c467-e9e5-4e85-8eef-d7df878d2a60/resourceGroups/rg-azure-la...
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
tim@adjectivesoftware....
DEFAULT DIRECTORY
Home > SQL databases > ProductsDB (sql-server-lab-ps/ProductsDB)
ProductsDB (sql-server-lab-ps/ProductsDB) | Query editor (preview)
SQL database
Search
Login
New Query
Open query
Feedback
Getting started
Overview
Activity log
Tags
Diagnose and solve problems
Query editor (preview)
Mirror database in Fabric (preview)
Resource visualizer
Settings
Data management
Integrations
Power Platform
Security
Intelligent performance
Monitoring
Automation
Help
Add or remove favorites by pressing Cmd+Shift+F
ProductsDB (tim@adjectivesoftware.com
Tables
Views
Stored Procedures
Query 1
Run
Cancel query
Save query
Export data as
Show only Editor
Results
Messages
Search to filter items...

Showing limited object explorer here. For full capability please click here to open Azure Data Studio.

```sql
CREATE TABLE Products (
    Id INT PRIMARY KEY IDENTITY(1,1),
    Name NVARCHAR(50),
    Price DECIMAL(10,2)
);

INSERT INTO Products (Name, Price)
VALUES
  ('Azure Hat', 14.99),
  ('Cloud Mug', 9.49),
  ('Dev Hoodie', 39.95),
  ('Laptop Sticker Pack', 4.99),
  ('Notebook', 6.75);

SELECT * FROM Products;
```

Id	Name	Price
1	Azure Hat	14.99
2	Cloud Mug	9.49
3	Dev Hoodie	39.95
4	Laptop Sticker Pack	4.99
5	Notebook	6.75

Query succeeded | 0s

2. If the query doesn’t run:

- Confirm your IP address is added to the SQL Server firewall
- Verify SQL admin login credentials

Lab 4

Creating an App Service and Key Vault

Introduction

Description

In this lab, you will create a Key Vault to store the SQL connection string generated in Lab 3. You'll then build and deploy a .NET web application (provided in the courseware) that retrieves the secret from Key Vault and queries the Azure SQL Database. You'll also configure Key Vault role assignments so that your user account can add and edit secrets in the portal, and the App Service's managed identity has permission to retrieve secrets securely. This lab depends on the database and credentials provisioned in Lab 3.

Lab Objectives

- Create an Azure Key Vault and store secrets securely
- Create an App Service and configure managed identity to retrieve secrets
- Deploy a .NET app using Kudu Zip Push Deploy

Create a Key Vault

1. Search for "**Key vaults**" in the top search bar of the Azure Portal.
2. Click **+ Create**.
3. Set the following:
   a. **Name:** key-vault-lab-{your-initials-here} *(must be unique)*
   b. **Resource group**: rg-azure-labs-{your-initials-here}
   c. **Region:** Same as resource group
   d. **Pricing tier:** Standard
4. Click **Review + create > Create**.

Create an App Service

1. Search for “**App Services**” in the top search bar of the Azure Portal.
2. Click **+ Create > Web App**
3. Set the following:
    a. **Resource group**: rg-azure-labs-{your-initials-here}
    b. **Name:** web-app-lab-{your-initials-here} *(must be unique)*
    c. **Runtime stack:** .NET 8 (LTS)
    d. **Region:** Same as resource group
    e. **Pricing plan:** Basic B1
4. Click Review + create > Create.

Set Key Vault Name in App Environment Variables

1. Navigate to the created App Service resource.
2. In the left menu, click **Settings > Environment Variables**.
3. Click **+ Add**.
4. Set the following:
    a. **Name:** KEY_VAULT_NAME
    b. **Value:** key-vault-lab-{your-initials} *(or whatever name you chose previously)*
5. Click Apply.
6. Click Apply (again!) and **Confirm**.

Enable Managed Identity

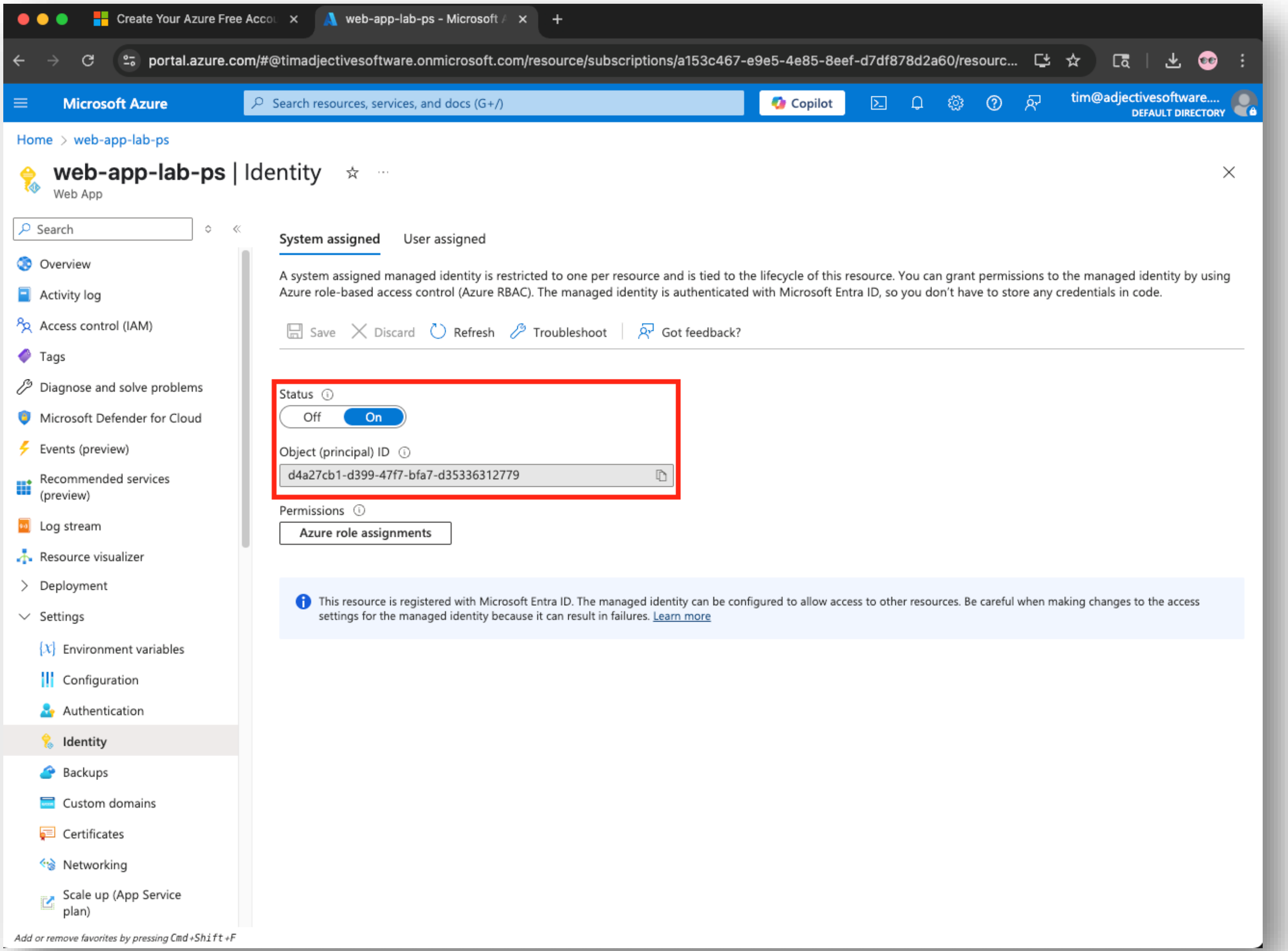
1. Navigate to the App Service resource.

2. In the left menu, click **Settings > Identity**.

3. Under **System assigned**, set **Status** to **On**.

4. Click Save.

5. Copy the **Object ID** – you will need it in the next step.



Create Your Azure Free Accou
web-app-lab-ps - Microsoft
portal.azure.com/#@timadjectivesoftware.onmicrosoft.com/resource/subscriptions/a153c467-e9e5-4e85-8eef-d7df878d2a60/resourc...
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
tim@adjectivesoftware....
DEFAULT DIRECTORY
Home > web-app-lab-ps
web-app-lab-ps | Identity
Web App
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Microsoft Defender for Cloud
Events (preview)
Recommended services (preview)
Log stream
Resource visualizer
Deployment
Settings
Environment variables
Configuration
Authentication
Identity
Backups
Custom domains
Certificates
Networking
Scale up (App Service plan)
Add or remove favorites by pressing Cmd+Shift+F
System assigned
User assigned
Save
Discard
Refresh
Troubleshoot
Got feedback?
Permissions
Azure role assignments

A system assigned managed identity is restricted to one per resource and is tied to the lifecycle of this resource. You can grant permissions to the managed identity by using Azure role-based access control (Azure RBAC). The managed identity is authenticated with Microsoft Entra ID, so you don't have to store any credentials in code.

Status: Off [✓ On]

Object (principal) ID: d4a27cb1-d399-47f7-bfa7-d35336312779

This resource is registered with Microsoft Entra ID. The managed identity can be configured to allow access to other resources. Be careful when making changes to the access settings for the managed identity because it can result in failures. Learn more

Add Role Assignments

Add Key Vault Administrator Assignment to User Account

1. Navigate to the key vault resource.
2. In the left side menu, click **Access control (IAM).**
3. Click **+ Add > Add role assignment**.
4. Select **Key Vault Secrets Administrator**.
5. Click Next.
6. Set the following:
    a. **Assign access to:** User, group, of service principal
    b. **Members:** Click + Select members
    c. Select your currently logged-in user account
7. Click **Review + assign**.

Add Key Vault Secrets User Assignment to App Service

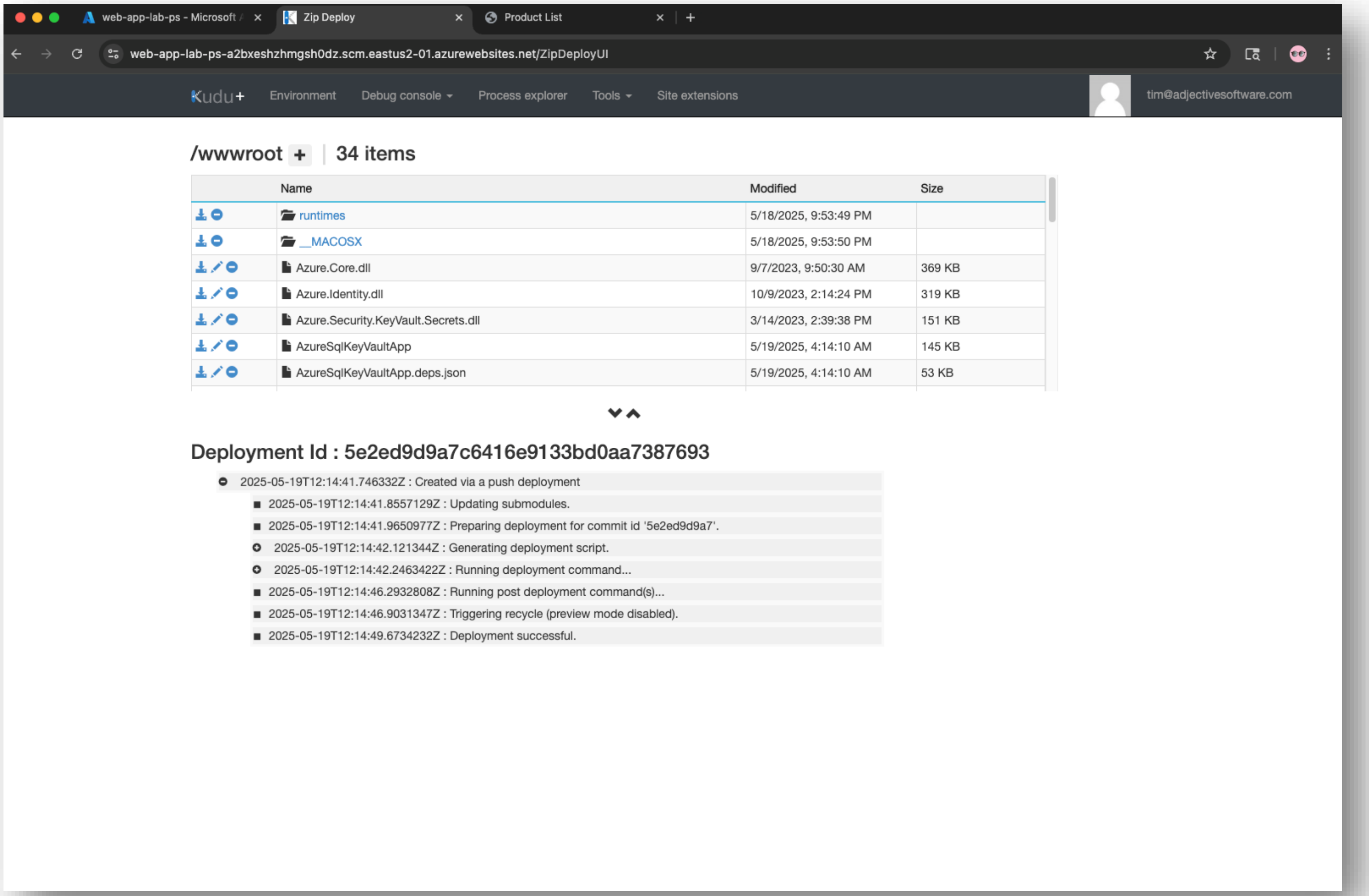
1. In the left side menu, click **Access control (IAM).**
2. Click **+ Add > Add role assignment**.
3. Select **Key Vault Secrets User**.
4. Click Next.
5. Set the following:
    a. **Assign access to:** Managed identity
    b. **Members:** Click + Select members
    c. Locate and **select** your App Service resource by **name** or **Object ID**
6. Click **Review + assign**.

Add SQL Connection String to Key Vault

1. Navigate to the created key vault resource.
2. In the left menu, click **Objects > Secrets**.
3. Click + **Generate/Import**.
4. Set the following:
    a. **Name:** sql-conn-string
    b. **Value:** {your-database-connection-string} *(see Lab 3: Accessing Database Connection String)*
5. Click Save.

Deploy .NET App using Kudu Zip Push Deploy

1. Navigate to the App Service resource.
2. Click **Development Tools > Advanced Tools**.
3. Click **Go ->.** The **Kudu Services** console will open in a new tab.
4. In the top menu, click **Tools > Zip Push Deploy**.
5. Click the **"-"** icon on the left side of the **hostingstart.html** file to **delete** it.
6. Drag the prebuilt-app.zip folder from the courseware directory to the **/wwwroot folder** (where the hostingstart.html file was deleted) to deploy the app.


web-app-lab-ps - Microsoft
Zip Deploy
Product List
web-app-lab-ps-a2bxeshzhmgsh0dz.scm.eastus2-01.azurewebsites.net/ZipDeployUI
Kudu+
Environment
Debug console
Process explorer
Tools
Site extensions
tim@adjectivesoftware.com

/wwwroot + | 34 items

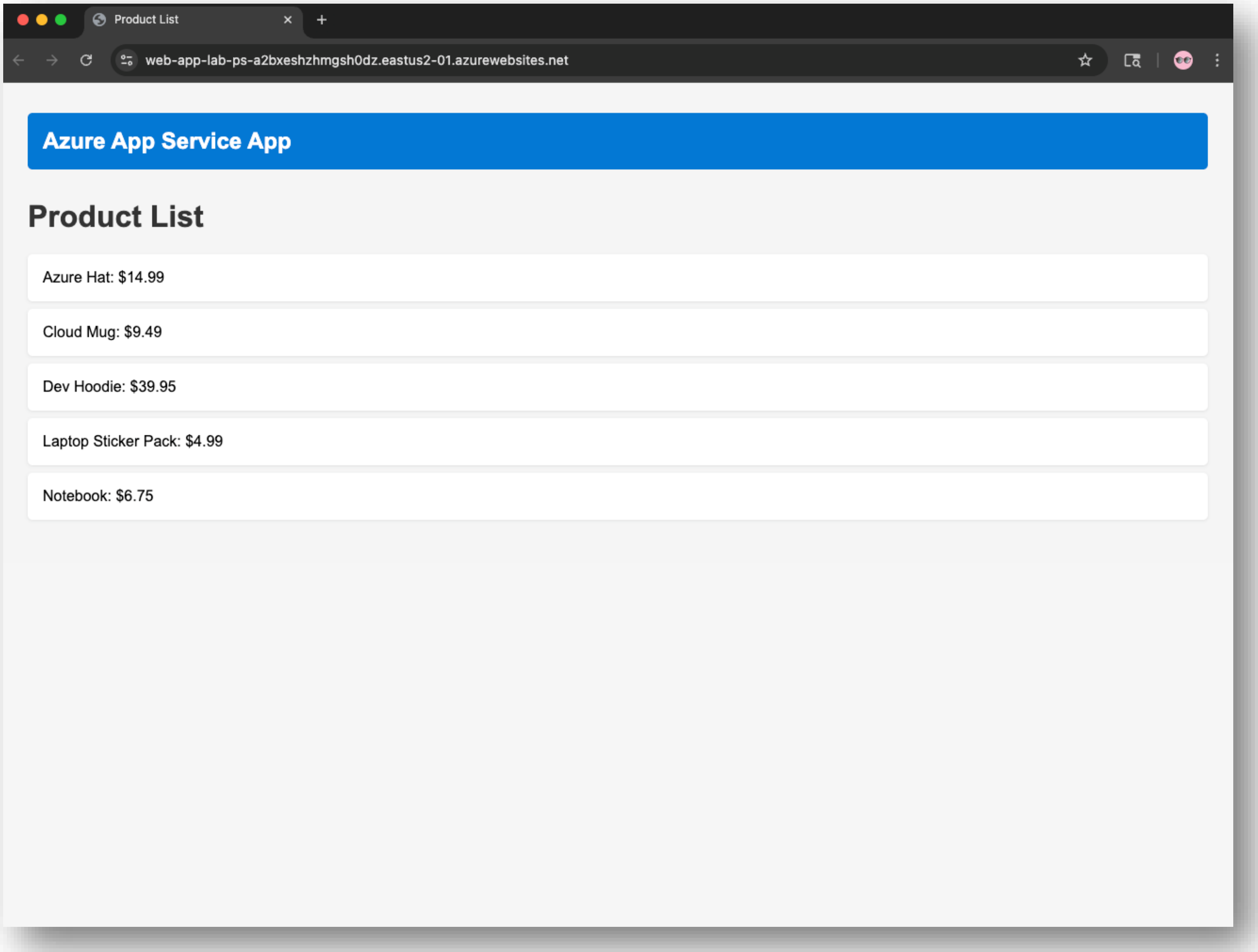
	Name	Modified	Size
	runtimes	5/18/2025, 9:53:49 PM	
	__MACOSX	5/18/2025, 9:53:50 PM	
	Azure.Core.dll	9/7/2023, 9:50:30 AM	369 KB
	Azure.Identity.dll	10/9/2023, 2:14:24 PM	319 KB
	Azure.Security.KeyVault.Secrets.dll	3/14/2023, 2:39:38 PM	151 KB
	AzureSqlKeyVaultApp	5/19/2025, 4:14:10 AM	145 KB
	AzureSqlKeyVaultApp.deps.json	5/19/2025, 4:14:10 AM	53 KB

Deployment Id : 5e2ed9d9a7c6416e9133bd0aa7387693

```
2025-05-19T12:14:41.746332Z : Created via a push deployment
    2025-05-19T12:14:41.8557129Z : Updating submodules.
    2025-05-19T12:14:41.9650977Z : Preparing deployment for commit id '5e2ed9d9a7'.
    2025-05-19T12:14:42.121344Z : Generating deployment script.
    2025-05-19T12:14:42.2463422Z : Running deployment command...
    2025-05-19T12:14:46.2932808Z : Running post deployment command(s)...
    2025-05-19T12:14:46.9031347Z : Triggering recycle (preview mode disabled).
    2025-05-19T12:14:49.6734232Z : Deployment successful.
```

Verification

1. Browse to the App Service URL and confirm the page loads successfully.
2. Ensure the page displays the data retrieved from the SQL database (the list of products).
3. If you see an error, verify that the App Service has permission to access Key Vault and that the SQL connection string is being properly stored.


Product List
web-app-lab-ps-a2bxeshzhmgsh0dz.eastus2-01.azurewebsites.net

Azure App Service App

Product List

Azure Hat: $14.99

Cloud Mug: $9.49

Dev Hoodie: $39.95

Laptop Sticker Pack: $4.99

Notebook: $6.75

4. If the app fails to load:

- Ensure the App Service has the **Key Vaults Secret User** role
- Confirm **KEY VAULT_NAME** is correctly set in environment variables

- Verify the SQL connection string was saved as a secret in the Key Vault

Lab 5

Creating a Virtual Machine

Introduction

Description

In this final lab, you will create a Windows Virtual Machine (VM) inside a custom Virtual Network (VNet). You will configure a Network Security Group (NSG) add a custom security rule to allow a Remote Desktop connection from Windows Admin Center over the public internet. This lab introduces basic network isolation and access control.

Lab Objectives

- Create a virtual network and network security group for isolation
- Deploy a Windows virtual machine with Remote Desktop access
- Use Windows Admin Center to manage and connect to the VM

Create a Virtual Network

1. Search for “**Virtual networks**” in the top search bar.
2. Click **+ Create**.
3. Set the following:
   a. **Resource group name**: rg-azure-labs-{your-initials-here}
   b. **Virtual network name**: vnet-lab-{your-initials-here}
   c. **Region:** Same as resource group
4. Click **Review + create > Create**.

Create a Network Security Group

1. Search for “**Network security groups**” in the top search bar.

2. Click **+ Create**.

3. Set the following:

   a. **Resource group name**: rg-azure-labs-{your-initials-here}

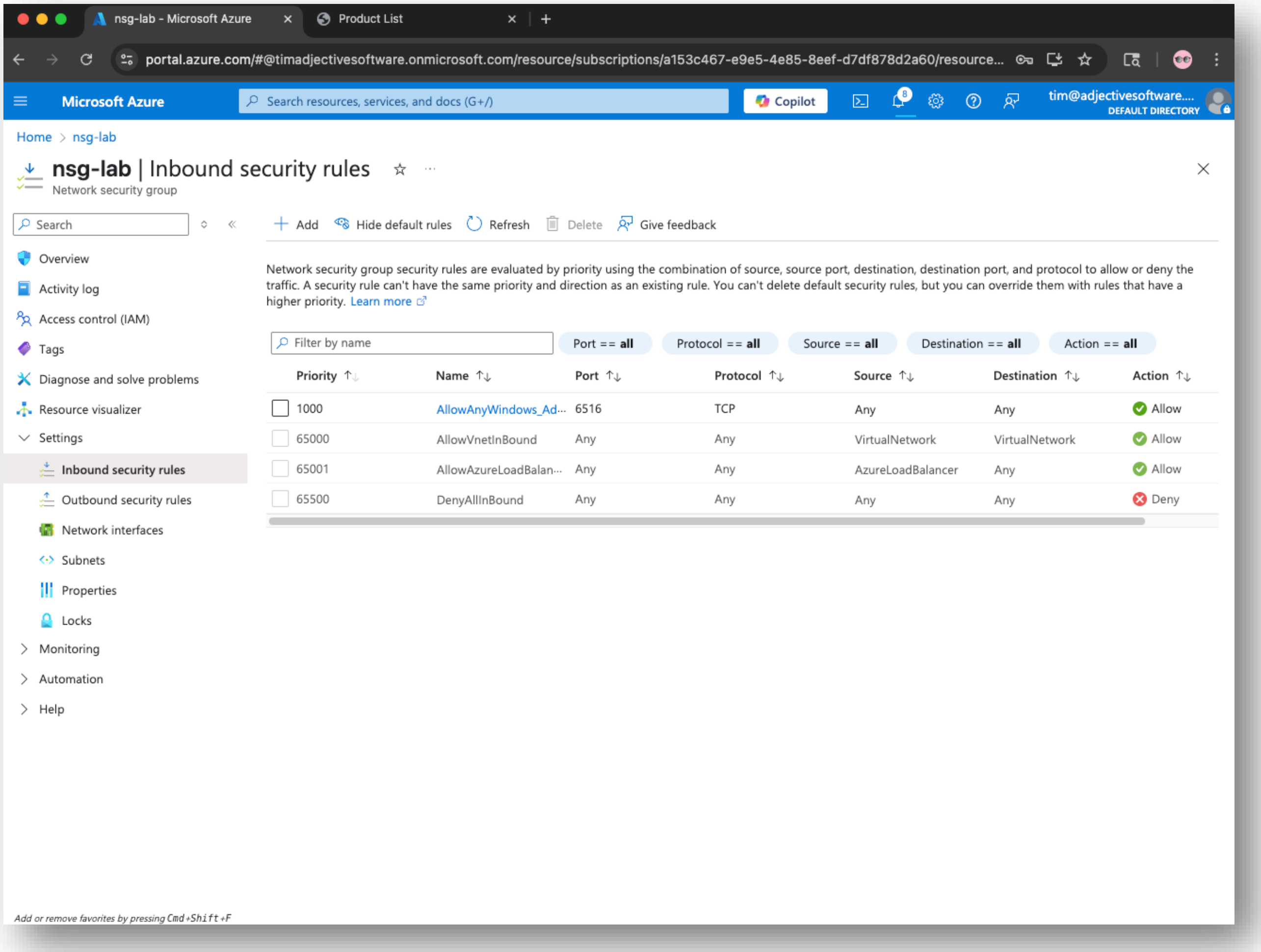
   b. **Name:** nsg-lab-{your-initials-here}

   c. **Region:** Same as resource group

4. Click **Review + create > Create**.

Configure NSG Inbound Security Rule

1. Navigate to the created network security group resource.
2. In the left menu, click **Settings > Inbound security rules**.
3. Click **+ Add**.
4. Add rule:
   a. **Source:** Any
   b. **Service:** Windows Admin Center
   c. **Action:** Allow
   d. **Priority:** 1000
5. Click Add.



Network security group security rules are evaluated by priority using the combination of source, source port, destination, destination port, and protocol to allow or deny the traffic. A security rule can't have the same priority and direction as an existing rule. You can't delete default security rules, but you can override them with rules that have a higher priority. Learn more

	Priority	Name	Port	Protocol	Source	Destination	Action
□	1000	AllowAnyWindows_Ad...	6516	TCP	Any	Any	✓ Allow
□	65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	✓ Allow
□	65001	AllowAzureLoadBalan...	Any	Any	AzureLoadBalancer	Any	✓ Allow
□	65500	DenyAllInBound	Any	Any	Any	Any	✗ Deny

Create a Virtual Machine

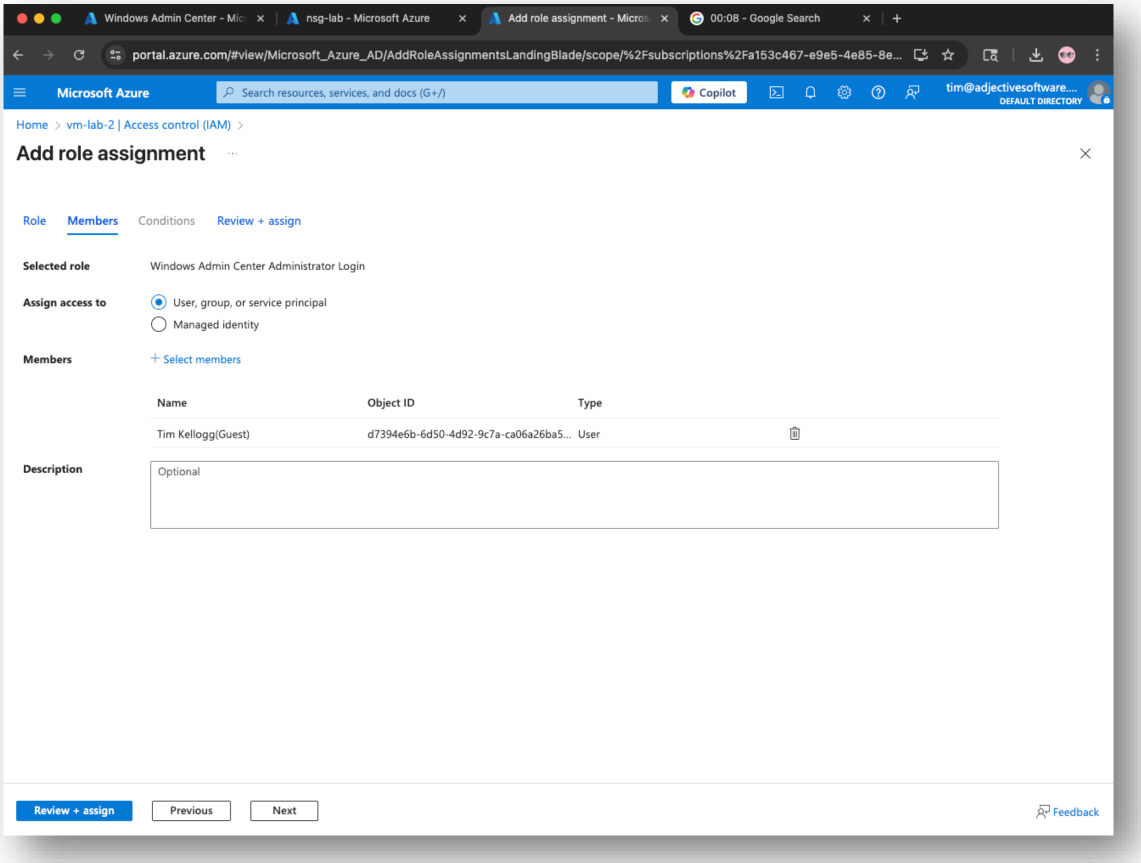
1. Search for “**Virtual machines”** in the top search bar.
2. Click **+ Create > Azure virtual machine**.
3. Set the following:
    a. **Resource group:** rg-azure-labs-{your-initials-here}
    b. **Virtual machine name:** vm-demo
    c. **Region:** Same as resource group
    d. **Availability options:** No infrastructure redundancy required
    e. **Image:** Windows Server 2022 Datacenter: Azure Edition
    f. **Size:** Standard D2v_v3
    g. **Username:** vmadmin
    h. **Password:** P@ssword123!
4. Click the **Networking** tab at the top.
5. Set the following:
    a. **Virtual network:** vnet-lab-{your-initials-here}
    b. **NIC network security group:** Advanced
    c. **Configure network security group:** nsg-lab-{your-initials-here}
    d. Enable **“Delete public IP and NIC when VM is deleted”**
6. Click **Review + create > Create**.

Install Windows Admin Center and Add Role Assignment

1. Navigate to the created Virtual Machine resource.
2. In the left side menu, click **Connect > Windows Admin Center**.
3. Confirm Inbound Port is pre-configured **6516**.
4. Check **“Open an outbound port for Windows Admin Center to install”**.
5. Click **Install**.

   - Note: This may take about 5 minutes to be configured on Azure’s end. Be patient and click the “Refresh button” to check the status. You can continue with the remaining steps in this section while waiting.

6. In the left side menu, click **Access control (IAM)**.
7. Click **Add role assignment**.
8. In the list, search for and select **“Windows Admin Center Administrator”** and click **Next**.
9. Click **+ Select Members**.
10. Type your username in the search box, select your account and click **Select**.


Windows Admin Center - Mic
nsg-lab - Microsoft Azure
Add role assignment - Micros
00:08 - Google Search
portal.azure.com/#view/Microsoft_Azure_AD/AddRoleAssignmentsLandingBlade/scope/%2Fsubscriptions%2Fa153c467-e9e5-4e85-8e...
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
tim@adjectivesoftware....
DEFAULT DIRECTORY
Home > vm-lab-2 | Access control (IAM) >

Add role assignment

Role
Members
Conditions
Review + assign

Selected role: Windows Admin Center Administrator Login

Assign access to:
● User, group, or service principal
○ Managed identity

Members: + Select members

Name	Object ID	Type
Tim Kellogg(Guest)	d7394e6b-6d50-4d92-9c7a-ca06a26ba5...	User

Description: Optional

Review + assign
Previous
Next
Feedback

11. Click **Review + assign**.

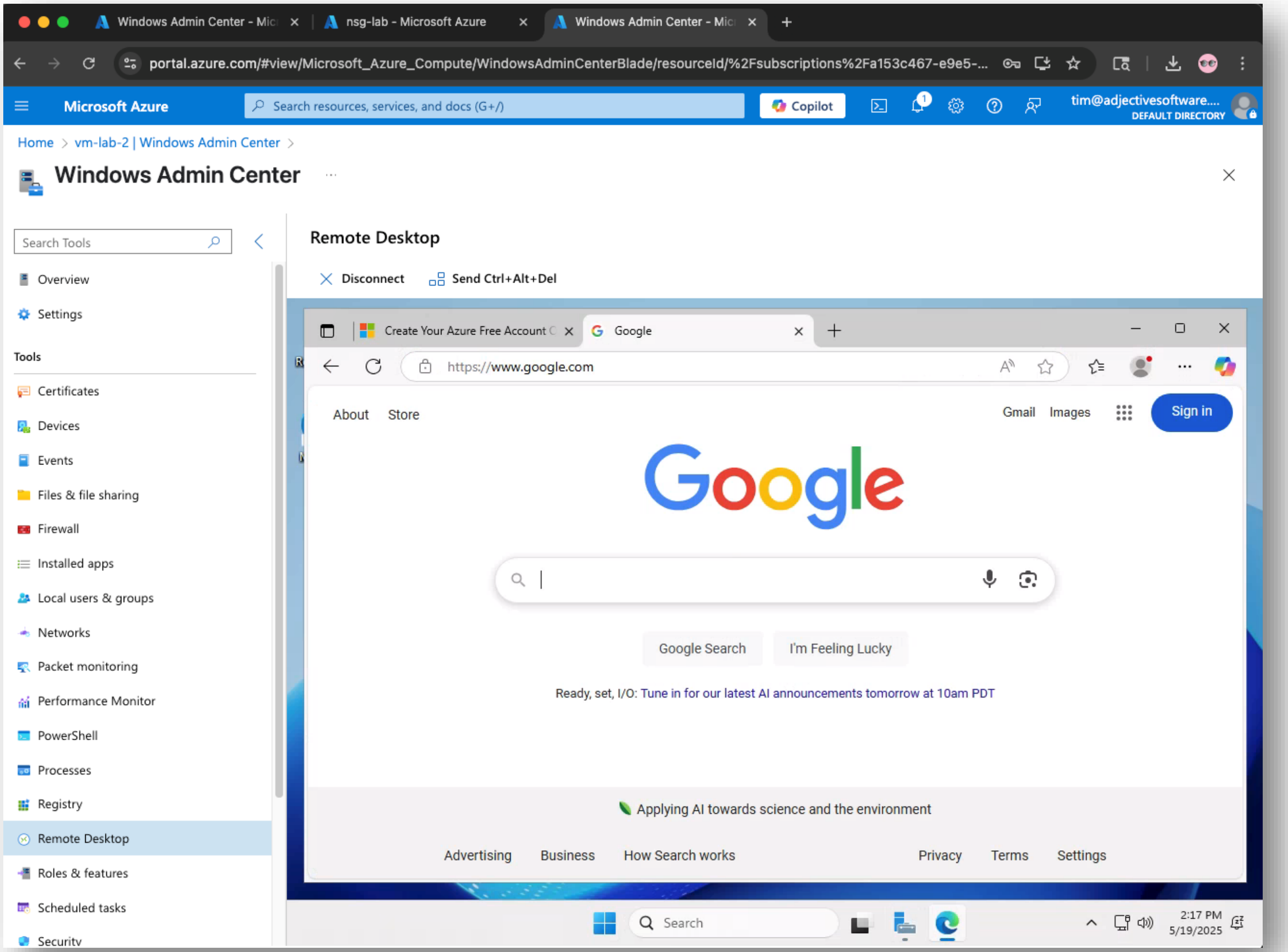
Connect to VM via Windows Admin Center

1. Navigate to the created Virtual Machine resource.
2. In the left side menu, click **Connect > Windows Admin Center**.
3. Confirm IP Address is set to **Public IP address (your-ip-address-here)**.
4. Click **Connect**. This will navigate you to the Windows Admin Center blade.
5. In the top menu, click **Connect > Remote Desktop**.
6. Enter the username and password created in **“Create a Virtual Machine”** step.
7. Check **“Automatically connect with the certificate presented by this machine *”**.
8. Click **Connect**.

Verification

1. Successfully connect to the VM via **Windows Admin Center > Remote Desktop**.

2. On the VM, open a browser and verify internet access.



3. If you're unable to connect:

- Ensure port **6516** is open for Windows Admin Center in the NSG
- Verify the VM has public IP and that the IP is correct
- Confirm you've been added as a **Windows Admin Center Administrator**