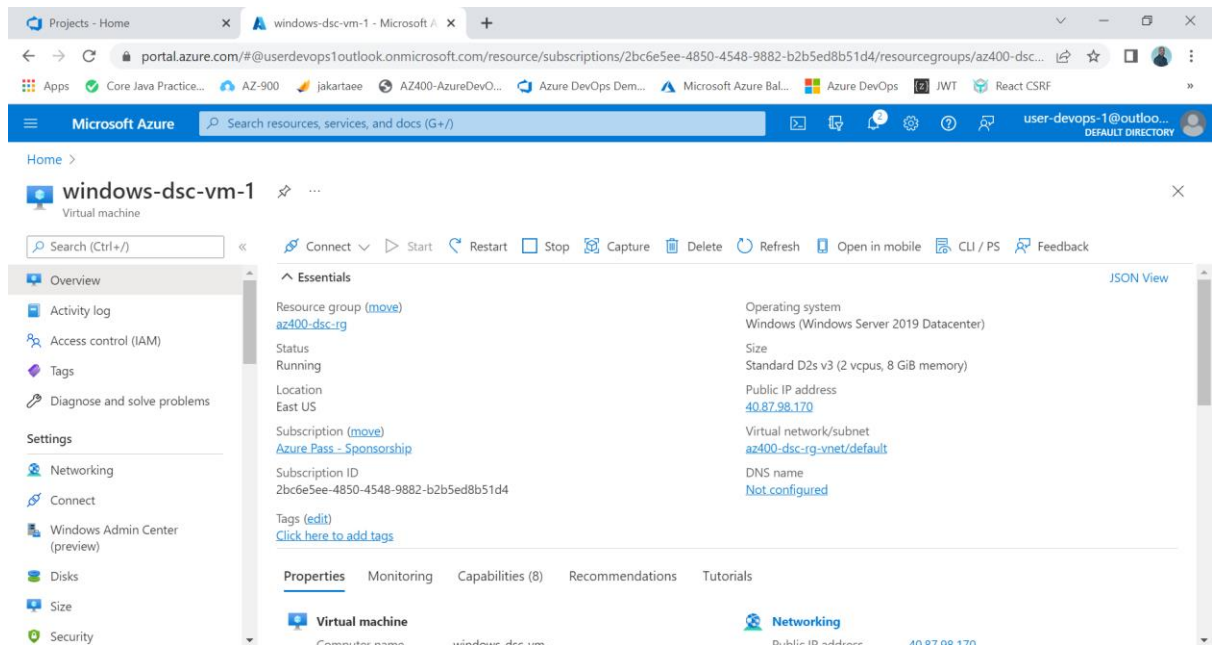


1. Create new windows virtual machine & add inbound port 80



2. Now to install IIS we will use two custom scripts, save it with .ps1 extension

Configuration DSCConfiguration {

Node \$AllNodes.Where{\$_.Role -eq "WebServer"}.NodeName

{

WindowsFeature IISInstall {

Ensure = 'Present'

Name = 'Web-Server'

}

}}

And save following file with .psd1 extension

@{

AllNodes = @(

@{

NodeName = "localhost"

Role = "WebServer"

}

)

}

3. Create a storage account

The screenshot shows the 'Create a storage account' page in the Microsoft Azure portal. The page has a breadcrumb trail: Home > Storage accounts > Create a storage account. Below the breadcrumb, there are tabs for 'Basics', 'Advanced', 'Networking', 'Data protection', 'Encryption', 'Tags', and 'Review + create'. The 'Basics' tab is selected. The form contains the following fields:

- Storage account name:** az400dscstorageaccount
- Region:** (US) East US
- Performance:** ☒ Standard: Recommended for most scenarios (general-purpose v2 account). ☐ Premium: Recommended for scenarios that require low latency.
- Redundancy:** Locally-redundant storage (LRS)

At the bottom, there are three buttons: 'Review + create' (in blue), '< Previous', and 'Next: Advanced >'.

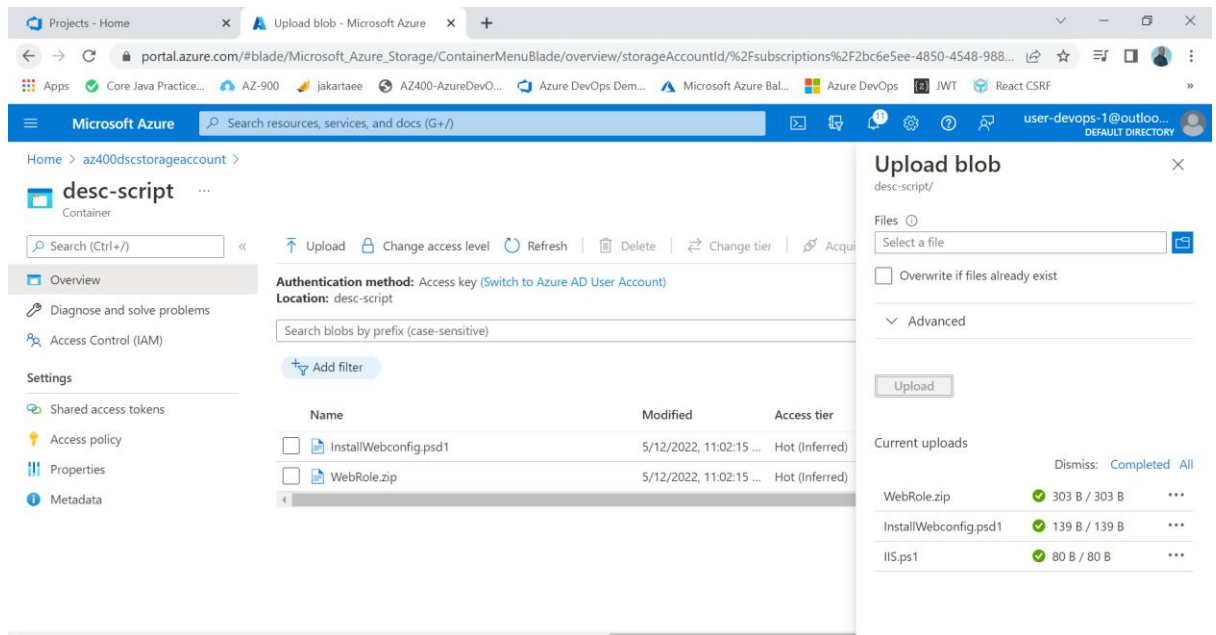
4. Create a container →

The screenshot shows the Microsoft Azure portal with a storage account named 'az400dscstorageaccount' selected. The 'Containers' section is active in the left sidebar. A 'New container' dialog is open on the right. The dialog has the following fields:

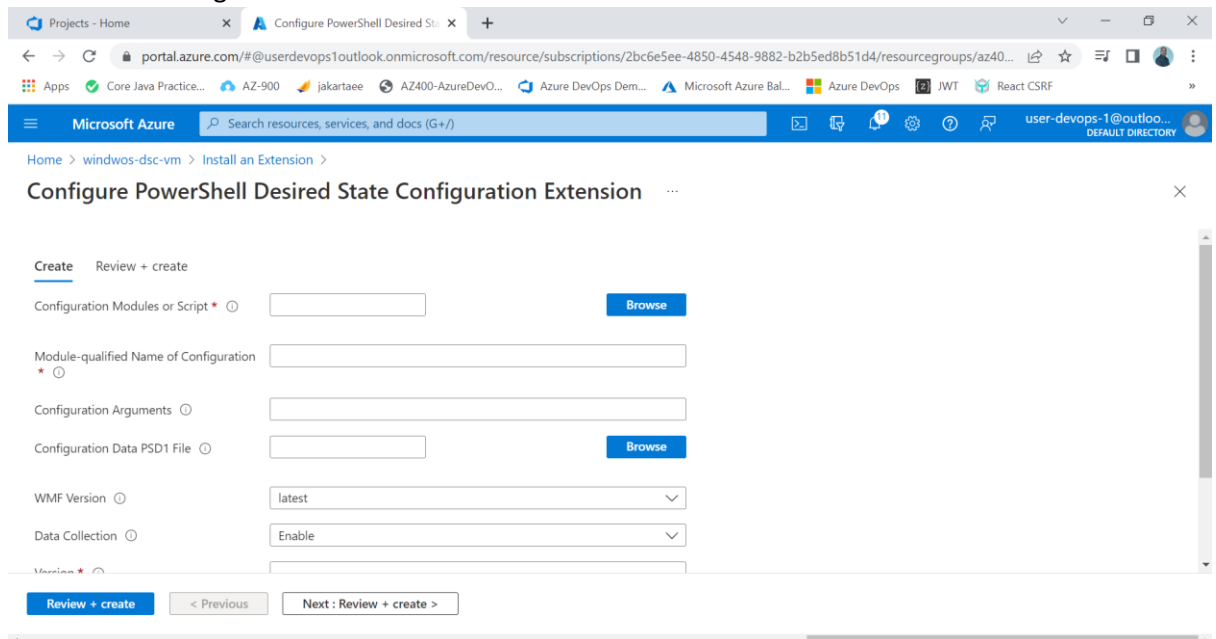
- Name:** custom-script
- Public access level:** Blob (anonymous read access for blobs only)

Below these fields, there is a warning message: 'Blobs within the container can be read by anonymous request, but container data is not available. Anonymous clients cannot enumerate the blobs within the container.' At the bottom of the dialog, there are 'Create' and 'Discard' buttons.

5. upload .ps1 & .psd1 file to newly create container



- Go back to virtual machine → click on Extension + application → search for Powershell desire state configuration. If not visible then down below click Load more extension



- Fill all the data → click on Review + create

Configure PowerShell Desired State Configuration Extension

Create Review + create

Configuration Modules or Script *

Module-qualified Name of Configuration *

Configuration Arguments

Configuration Data PSD1 File

WMF Version

Data Collection

Version *

8. Finally once deployment is successful → take public IP address of vm → test IIS

