

Lab_06

To create a PowerShell script that deploys an IIS web server on a remote Windows server and modifies the default web page to display the server's public IP address, you can use a combination of PowerShell Desired State Configuration (DSC) for setting up IIS and additional scripting to dynamically fetch and display the public IP address on the default web page.

PowerShell Script

Here's the complete script:

Define the configuration for deploying IIS and modifying the default page

Configuration DeployIISWithIP

```
{  
param (  
[Parameter(Mandatory)]  
[string]$NodeName  
)
```

```
# Import necessary DSC Resources  
Import-DscResource -ModuleName 'PSDesiredStateConfiguration'  
Import-DscResource -ModuleName 'xWebAdministration'  
  
# Node configuration for the remote server  
Node $NodeName  
{  
    # Ensure IIS is installed  
    WindowsFeature InstallIIS  
    {  
        Ensure = "Present"  
        Name    = "Web-Server"  
    }  
  
    # Ensure the Default Website is running  
    xWebsite DefaultWebsite  
    {
```

```

        Ensure      = "Present"
        Name        = "Default Web Site"
        State       = "Started"
        PhysicalPath = "C:\inetpub\wwwroot"
        DependsOn   = "[WindowsFeature]InstallIIS"
    }

    # Script to fetch public IP and update the index.html
    Script UpdateIndexPage
    {
        GetScript = {
            @{ Result = (Get-Content -Path "C:\inetpub\wwwroot\index.html") }
        }
        SetScript = {
            $publicIp = Invoke-RestMethod -Uri 'http://ipinfo.io/json' |
Select-Object -ExpandProperty ip
            $content = "<html><body><h1>Your IP Address is: $publicIp</h1>
</body></html>"
            Set-Content -Path "C:\inetpub\wwwroot\index.html" -Value $content
        }
        TestScript = {
            $currentContent = Get-Content -Path
"C:\inetpub\wwwroot\index.html"
            return $currentContent -match "Your IP Address is:"
        }
        DependsOn = "[xWebsite]DefaultWebsite"
    }
}

```

```

}

```

Nodes where this configuration will be applied

`$nodeName = "yourRemoteServerName" # Replace with your actual server name or IP address`

Generate the MOF configuration file

`DeployIISWithIP -NodeName $nodeName -OutputPath "C:\DSC\DeployIISWithIP"`

Start the deployment on the remote node

Start-DscConfiguration -Path "C:\DSC\DeployIISWithIP" -Wait -Verbose -Force

Explanation:

1. **Configuration Definition:** This script defines a DSC configuration called `DeployIISWithIP`. It accepts a parameter `$NodeName` which specifies the target remote server.
2. **WindowsFeature and xWebsite:** These resources ensure that IIS is installed and the default website is running.
3. **Script Resource:** This dynamically updates `index.html` with the server's public IP. It fetches the IP using an external API (`ipinfo.io`) and writes the IP address to `index.html`.
4. **Applying the Configuration:** The configuration is applied by generating a MOF file and starting the DSC configuration.

Requirements:

- Ensure that the `xWebAdministration` DSC module is installed on the machine where the script is run.
- The target server must have PowerShell remoting enabled.
- The account running the script must have administrative privileges on the remote server.

Security Considerations:

- Using HTTP to fetch the IP might introduce vulnerabilities or reveal sensitive information. Consider using HTTPS.
- Ensure that PowerShell remoting and DSC are securely configured, especially in a production environment.

To enable PowerShell Remoting on an Azure VM, use the PowerShell cmdlets available in the Azure PowerShell module. Then, deploy and configure IIS on the VM using PowerShell Remoting. Follow these steps to accomplish this:

Prerequisites

- Ensure you have the Azure PowerShell module installed. If not, you can install it using:
`Install-Module -Name Az -AllowClobber -Scope CurrentUser`
- You should have administrative access to the Azure subscription and the virtual machine.
- The VM should have a network security group (NSG) rule allowing HTTP traffic on port 80 and PowerShell Remoting traffic.

Step 1: Enable PowerShell Remoting on the Azure VM

First, authenticate with Azure using `Connect-AzAccount` , then run the following command to enable PSRemoting.

```
$resourceGroupName = "YourResourceGroupName" # Replace with your resource group name
$vmName = "YourVMName" # Replace with your VM name
Enable-AzVMPSRemoting -ResourceGroupName $resourceGroupName -Name $vmName -Protocol Https
```

Step 2: Deploy and Configure IIS on the VM

Once PSRemoting is enabled, use the `Invoke-AzVMCommand` to run scripts on the VM. Here's how to execute the DSC configuration script to install and configure IIS.

Define the script to configure IIS

```
$scriptContent = @"
Configuration DeployIISWithIP
{
    Import-DscResource -ModuleName 'PSDesiredStateConfiguration'
    Import-DscResource -ModuleName 'xWebAdministration'
    Node localhost
    {
        WindowsFeature InstallIIS
        {
            Ensure = "Present"
            Name = "Web-Server"
        }
        xWebsite DefaultWebsite
        {
            Ensure = "Present"
            Name = "Default Web Site"
            State = "Started"
            PhysicalPath = "C:\inetpub\wwwroot"
        }
        Script UpdateIndexPage
        {
            GetScript = @{ Result = (Get-Content -Path "C:\inetpub\wwwroot\index.html") }
            SetScript = {
                $publicIp = Invoke-RestMethod -Uri 'http://ipinfo.io/json' | Select-Object -ExpandProperty ip
                $content = "
```

Your IP Address is: \$publicip

```
"  
Set-Content -Path "C:\inetpub\wwwroot\index.html" -Value $content  
}  
TestScript = { $currentContent = Get-Content -Path "C:\inetpub\wwwroot\index.html"; return  
$currentContent -match "Your IP Address is:" }  
}  
}  
}  
DeployIISWithIP  
Start-DscConfiguration -Path DeployIISWithIP -Wait -Force -Verbose  
"@
```

Run the script on the VM

```
Invoke-AzVMCommand -ResourceGroupName  
resourceGroupName -VMName $vmName -ScriptBlock { param(  
script) Invoke-Expression $script } -  
ArgumentList $scriptContent
```