How to create a Linux Virtual Machine (Ubuntu Server) with Azure SDK for .NET. After creating the VM we will install VSCode, Google Chrome and .NET 8

NOTE: for more information about VM with Azure SDK for .NET visit the URL

https://github.com/Azure/azure-sdk-for-net/blob/main/sdk/compute/Azure.ResourceManager.Compute/samples/Sample2_ManagingVirtualMachines.md

0. Prerequisites

Install .NET 8 SDK: https://dotnet.microsoft.com/en-us/download/dotnet/8.0

Install Azure CLI: https://learn.microsoft.com/en-us/cli/azure/install-azure-cli

Install VSCode: https://code.visualstudio.com/download

1. Create a new C# console .Net 8 application in VSCode

Open VSCode and run the command:

```
dotnet new console --framework net8.0
```

2. Load the Azure SDK libraries.

From the Nuget web page copy the commands to load the libraries: https://www.nuget.org/

Run these commands to load the libraries:

```
dotnet add package Azure.Identity --version 1.10.4
dotnet add package Azure.ResourceManager --version 1.9.0
dotnet add package Azure.ResourceManager.Network --version 1.6.0
dotnet add package Azure.ResourceManager.Compute --version 1.2.1
```

Now run the command:

```
dotnet restore
```

3. Input the C# source code.

```
using System;
using System.Threading.Tasks;
using Azure;
using Azure.Core;
using Azure.Identity;
using Azure.ResourceManager;
using Azure.ResourceManager.Network.Models;
using Azure.ResourceManager.Network;
using Azure.ResourceManager.Resources;
using Azure.ResourceManager.Resources.Models;
using Azure.ResourceManager.Compute;
using Azure.ResourceManager.Compute.Models;

class Program
{
    static async Task Main(string[] args)
    {
        ArmClient armClient = new ArmClient(new DefaultAzureCredential());
        SubscriptionResource subscription = await armClient.GetDefaultSubscriptionAsync();
```

```
ResourceGroupCollection rgCollection = subscription.GetResourceGroups();
string rgName = "myRgName";
AzureLocation location = AzureLocation.WestEurope;
ResourceGroupResource resourceGroup = await rgCollection.CreateOrUpdate(WaitUntil.Started, rgName, new ResourceGroupData(

PublicIPAddressCollection publicIPAddressCollection = resourceGroup.GetPublicIPAddresses();
string publicIPAddressName = "20.61.0.157";
PublicIPAddressData publicIPInput = new PublicIPAddressData()
{
    Location = resourceGroup.Data.Location,
    PublicIPAllocationMethod = NetworkIPAllocationMethod.Dynamic,
    DnsSettings = new PublicIPAddressDnsSettings()
    {
        DomainNameLabel = "mydomain12319741999"
    }
};
PublicIPAddressResource publicIPAddress = await publicIPAddressCollection.CreateOrUpdate(WaitUntil.Completed, publicIPAdd

VirtualNetworkCollection virtualNetworkCollection = resourceGroup.GetVirtualNetworks();
string vnetName = "myVnet";
VirtualNetworkData input = new VirtualNetworkData()
{
    Location = resourceGroup.Data.Location,
    AddressPrefixes = { "10.0.0.0/16", },
    DhcpOptionsDnsServers = { "10.1.1.1", "10.1.2.4" },
    Subnets = { new SubnetData() { Name = "mySubnet", AddressPrefix = "10.0.1.0/24", } }
};
VirtualNetworkResource vnet = await virtualNetworkCollection.CreateOrUpdate(WaitUntil.Completed, vnetName, input).WaitFor

VirtualNetworkCollection virtualNetworkCollection1 = resourceGroup.GetVirtualNetworks();
VirtualNetworkResource virtualNetwork1 = await virtualNetworkCollection1.GetAsync("myVnet");
Console.WriteLine(virtualNetwork1.Data.Name);

NetworkInterfaceCollection networkInterfaceCollection = resourceGroup.GetNetworkInterfaces();
string networkInterfaceName = "myNetworkInterface";
NetworkInterfaceData networkInterfaceInput = new NetworkInterfaceData()
```

```
{
    Location = resourceGroup.Data.Location,
    IPConfigurations = {
        new NetworkInterfaceIPConfigurationData()
        {
            Name = "ipConfig",
            PrivateIPAllocationMethod = NetworkIPAllocationMethod.Dynamic,
            PublicIPAddress = new PublicIPAddressData()
            {
                Id = publicIPAddress.Id
            },
            Subnet = new SubnetData()
            {
                Id = virtualNetwork1.Data.Subnets[0].Id
            }
        }
    }
};

NetworkSecurityGroupCollection nsgCollection = resourceGroup.GetNetworkSecurityGroups();
string nsgName = "myNetworkSecurityGroup";
NetworkSecurityGroupData nsgInput = new NetworkSecurityGroupData()
{
    Location = resourceGroup.Data.Location,
    SecurityRules =
    {
        new SecurityRuleData()
        {
            Name = "AllowSSH",
            Priority = 100,
            Access = SecurityRuleAccess.Allow,
            Direction = SecurityRuleDirection.Inbound,
            Protocol = SecurityRuleProtocol.Tcp,
            SourceAddressPrefix = "*",
            SourcePortRange = "*",
            DestinationAddressPrefix = "*",
            DestinationPortRange = "22", // SSH port
```

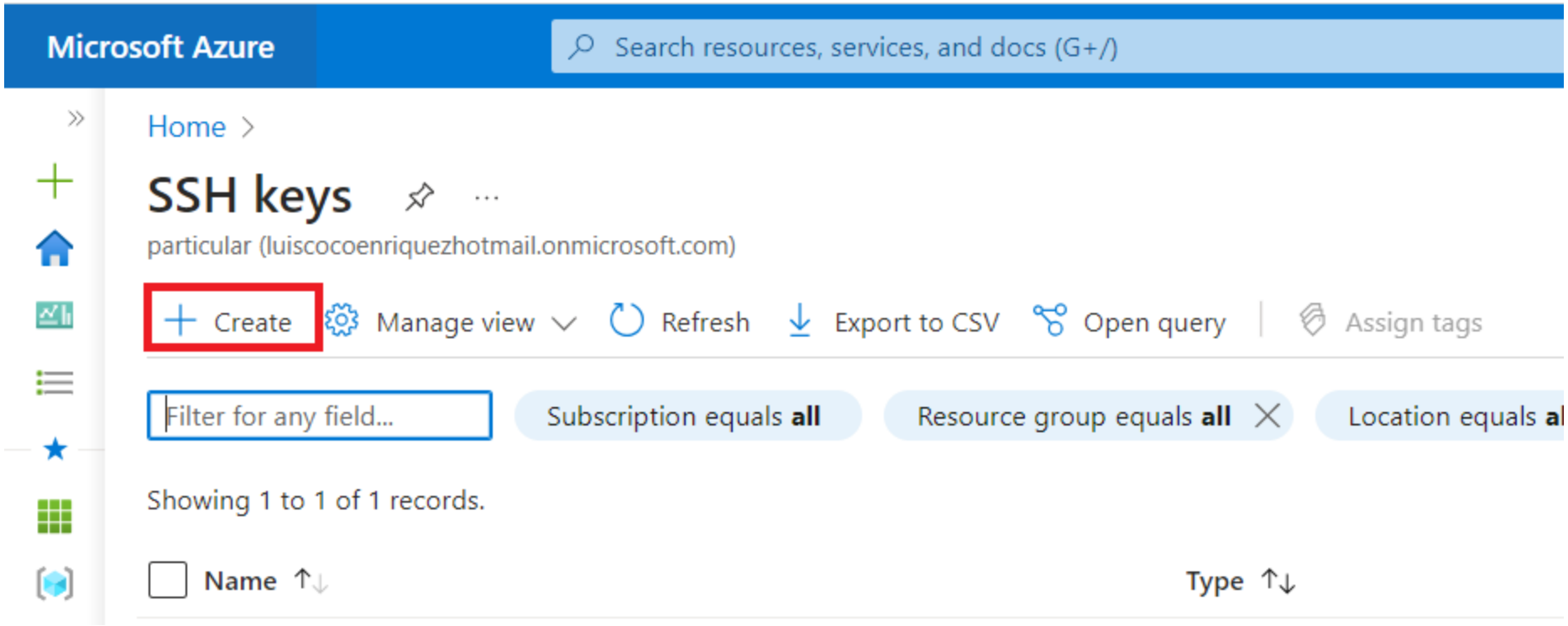
```
},
new SecurityRuleData()
{
    Name = "AllowHTTP",
    Priority = 110,
    Access = SecurityRuleAccess.Allow,
    Direction = SecurityRuleDirection.Outbound,
    Protocol = SecurityRuleProtocol.Tcp,
    SourceAddressPrefix = "*",
    SourcePortRange = "*",
    DestinationAddressPrefix = "*",
    DestinationPortRange = "80", // HTTP port
},
new SecurityRuleData()
{
    Name = "AllowHTTPS",
    Priority = 120,
    Access = SecurityRuleAccess.Allow,
    Direction = SecurityRuleDirection.Outbound,
    Protocol = SecurityRuleProtocol.Tcp,
    SourceAddressPrefix = "*",
    SourcePortRange = "*",
    DestinationAddressPrefix = "*",
    DestinationPortRange = "443", // HTTPS port
},
new SecurityRuleData()
{
    Name = "AllowRDP",
    Priority = 130,
    Access = SecurityRuleAccess.Allow,
    Direction = SecurityRuleDirection.Inbound,
    Protocol = SecurityRuleProtocol.Tcp,
    SourceAddressPrefix = "*",
    SourcePortRange = "*",
    DestinationAddressPrefix = "*",
    DestinationPortRange = "3389", // RDP port
}
```

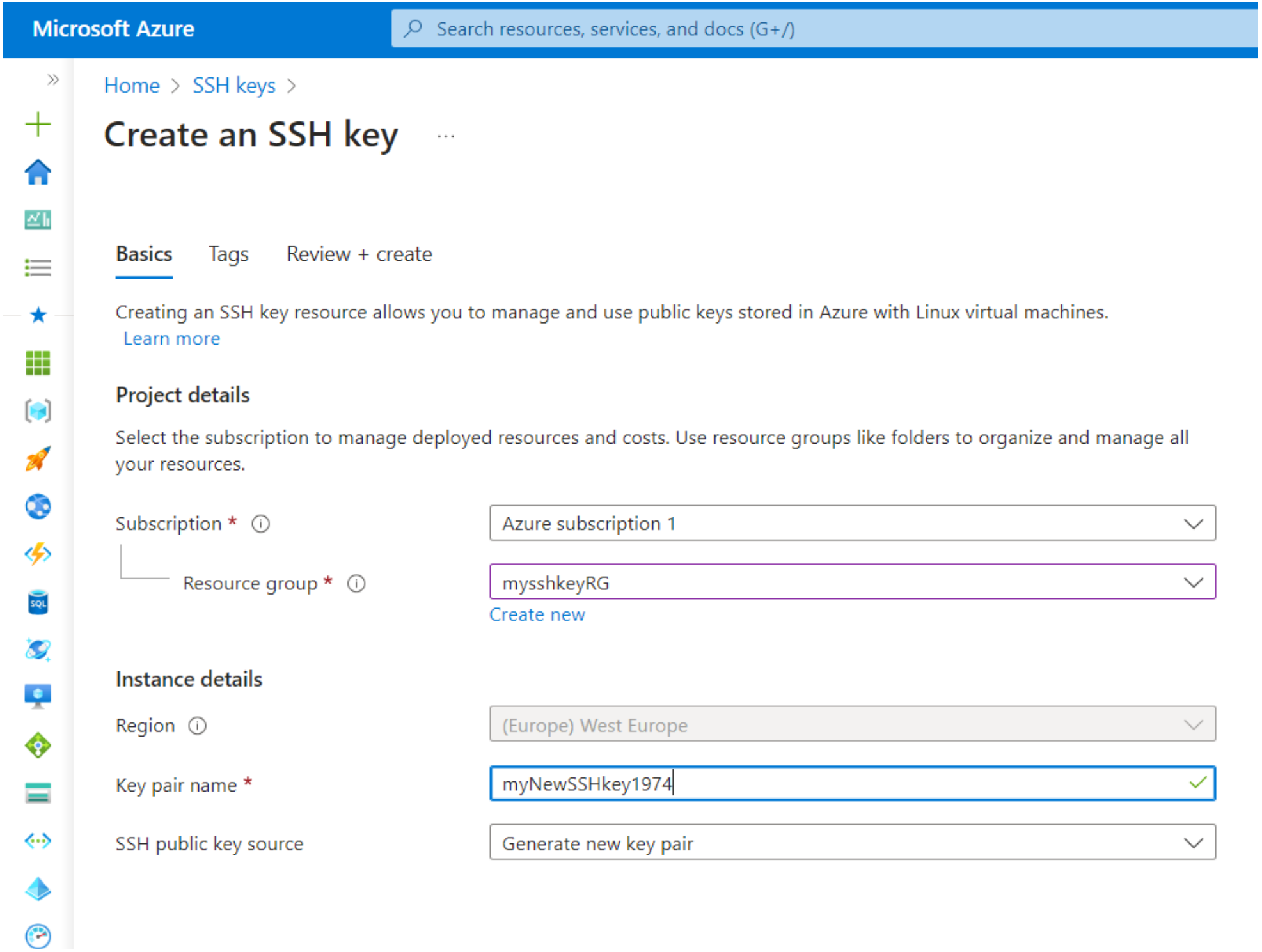
```
        }
    };

    NetworkSecurityGroupResource nsg = await nsgCollection.CreateOrUpdate(WaitUntil.Completed, nsgName, nsgInput).WaitForComp

    networkInterfaceInput.NetworkSecurityGroup = new NetworkSecurityGroupData()
    {
        Id = nsg.Id
    };

    NetworkInterfaceResource networkInterface = await networkInterfaceCollection.CreateOrUpdate(WaitUntil.Completed, networkI

    VirtualMachineCollection vmCollection = resourceGroup.GetVirtualMachines();
    string vmName = "myVM";
    string adminusername = "azureuser";
    VirtualMachineData input2 = new VirtualMachineData(resourceGroup.Data.Location)
    {
        HardwareProfile = new VirtualMachineHardwareProfile()
        {
            VmSize = VirtualMachineSizeType.StandardE2SV3
        },
        OSProfile = new VirtualMachineOSProfile()
        {
            AdminUsername = adminusername,
            ComputerName = "myVM",
            CustomData = Convert.ToBase64String(System.Text.Encoding.UTF8.GetBytes(
                    "#cloud-config\n" +
                    "write_files:\n" +
                    "  - path: /etc/systemd/resolved.conf\n" +
                    "    content: |\n" +
                    "      [Resolve]\n" +
                    "      DNS=8.8.8.8 8.8.4.4\n" +
                    "runcmd:\n" +
                    "  - systemctl restart systemd-resolved\n"
            )),
            LinuxConfiguration = new LinuxConfiguration()
            {
```

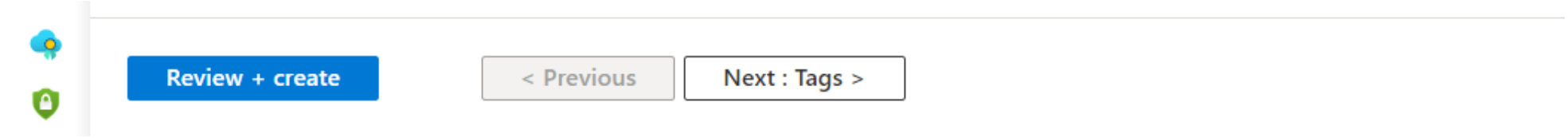
```
            DisablePasswordAuthentication = true,
            SshPublicKeys = {
                new SshPublicKeyConfiguration()
                {
                    Path = $"/home/" + adminusername + "/.ssh/authorized_keys",
                    KeyData = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC62swVPqUSWDldLCH/UelaV5hBQ7K2UjumZcVO+B4qjL3mCgN2oB
                }
            }
        }
    },
    NetworkProfile = new VirtualMachineNetworkProfile()
    {
        NetworkInterfaces =
        {
            new VirtualMachineNetworkInterfaceReference()
            {
                Id = new ResourceIdentifier($"/subscriptions/{subscription.Data.SubscriptionId}/resourceGroups/{rgName}/p
                Primary = true,
            }
        }
    },
    StorageProfile = new VirtualMachineStorageProfile()
    {
        OSDisk = new VirtualMachineOSDisk(DiskCreateOptionType.FromImage)
        {
            OSType = SupportedOperatingSystemType.Linux,
            Caching = CachingType.ReadWrite,
            ManagedDisk = new VirtualMachineManagedDisk()
            {
                StorageAccountType = StorageAccountType.StandardLrs
            }
        },
        ImageReference = new ImageReference()
        {
            Publisher = "Canonical",
            Offer = "0001-com-ubuntu-server-jammy",
            Sku = "22_04-lts-gen2",
```

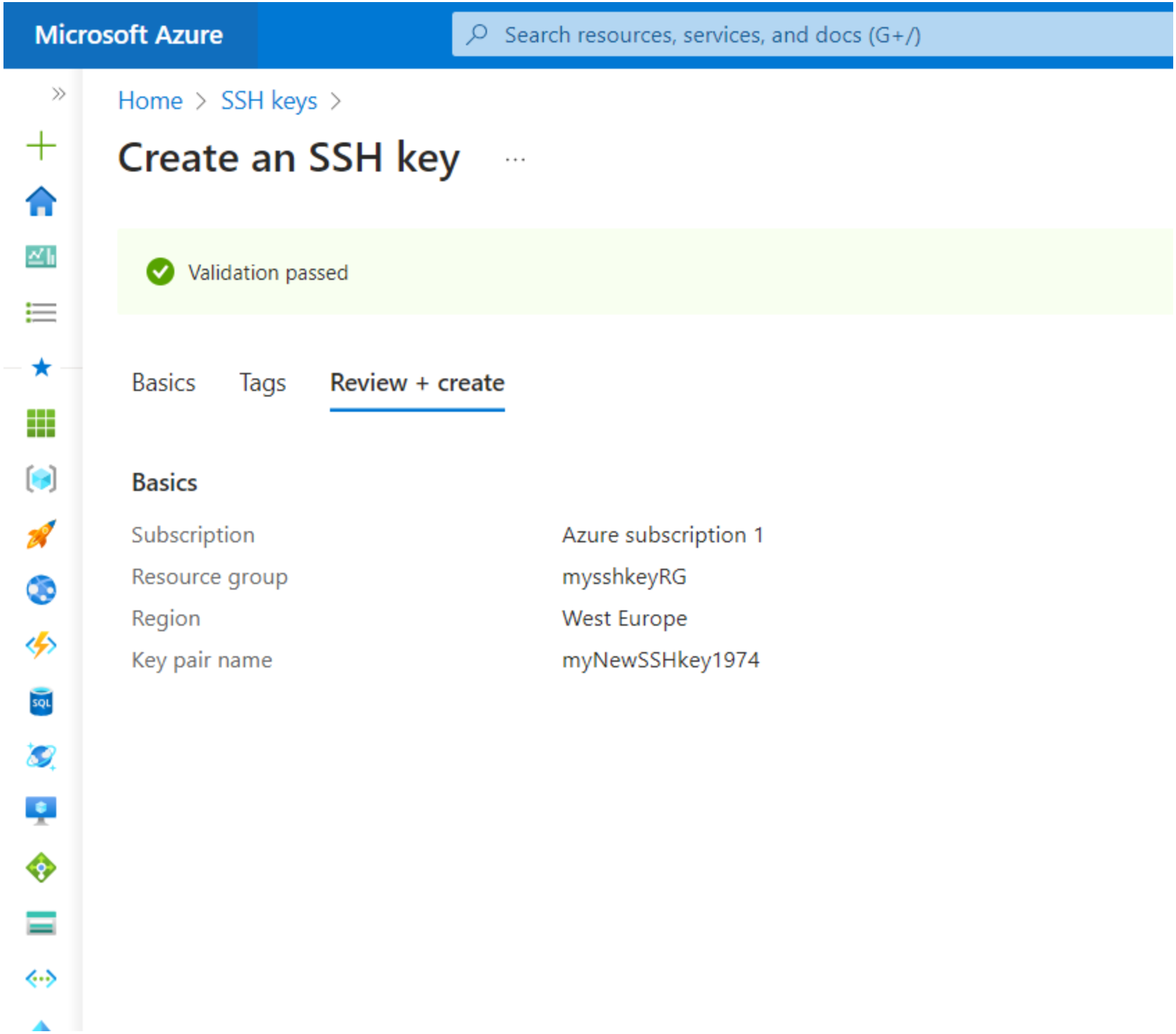
```
                    Version = "latest",
                }
            }
        };

        ArmOperation<VirtualMachineResource> lro = await vmCollection.CreateOrUpdateAsync(WaitUntil.Completed, vmName, input2);
        VirtualMachineResource vm = lro.Value;
    }
}
```

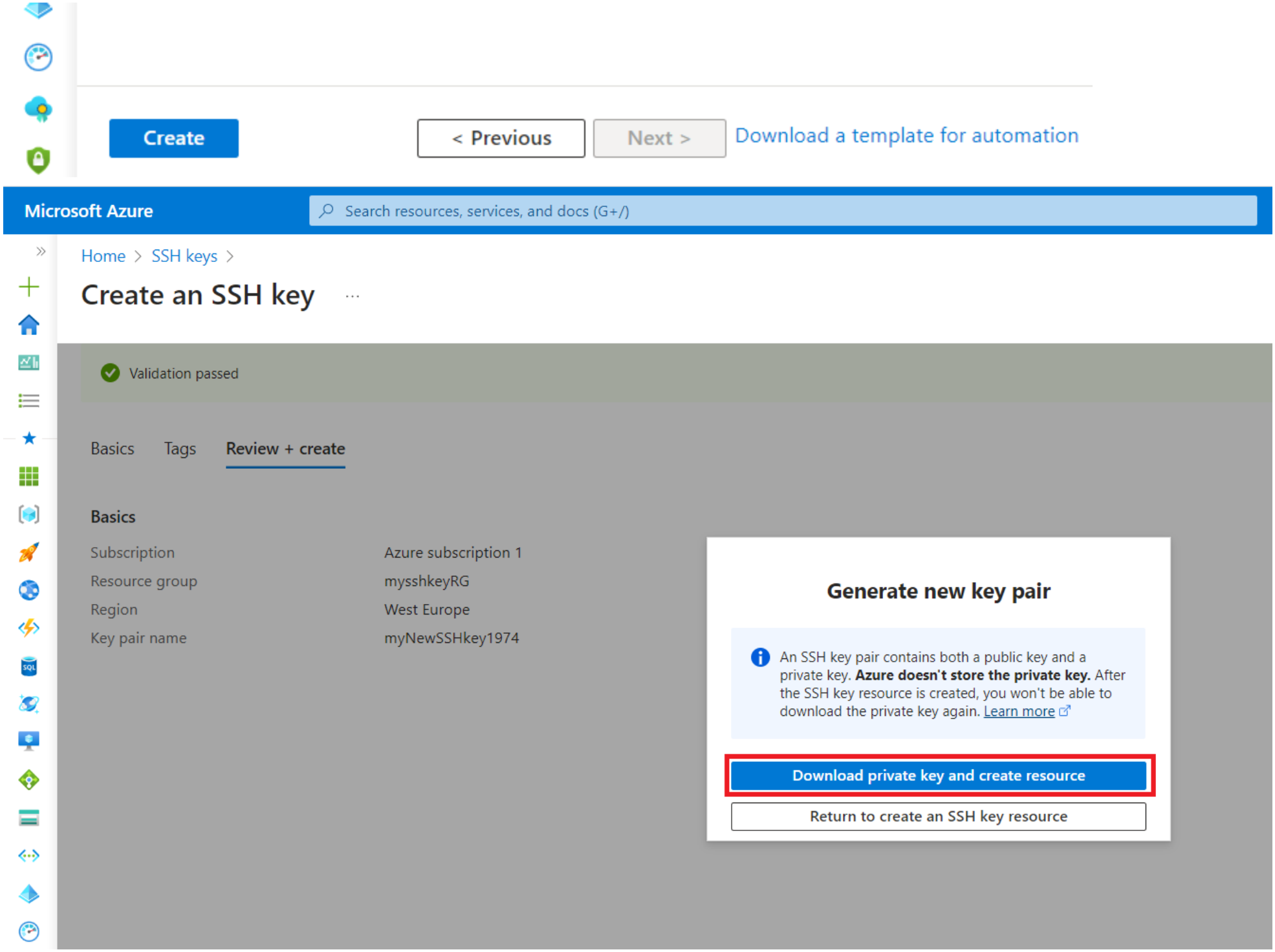
4. Create a new SSH key pair in Azure Portal



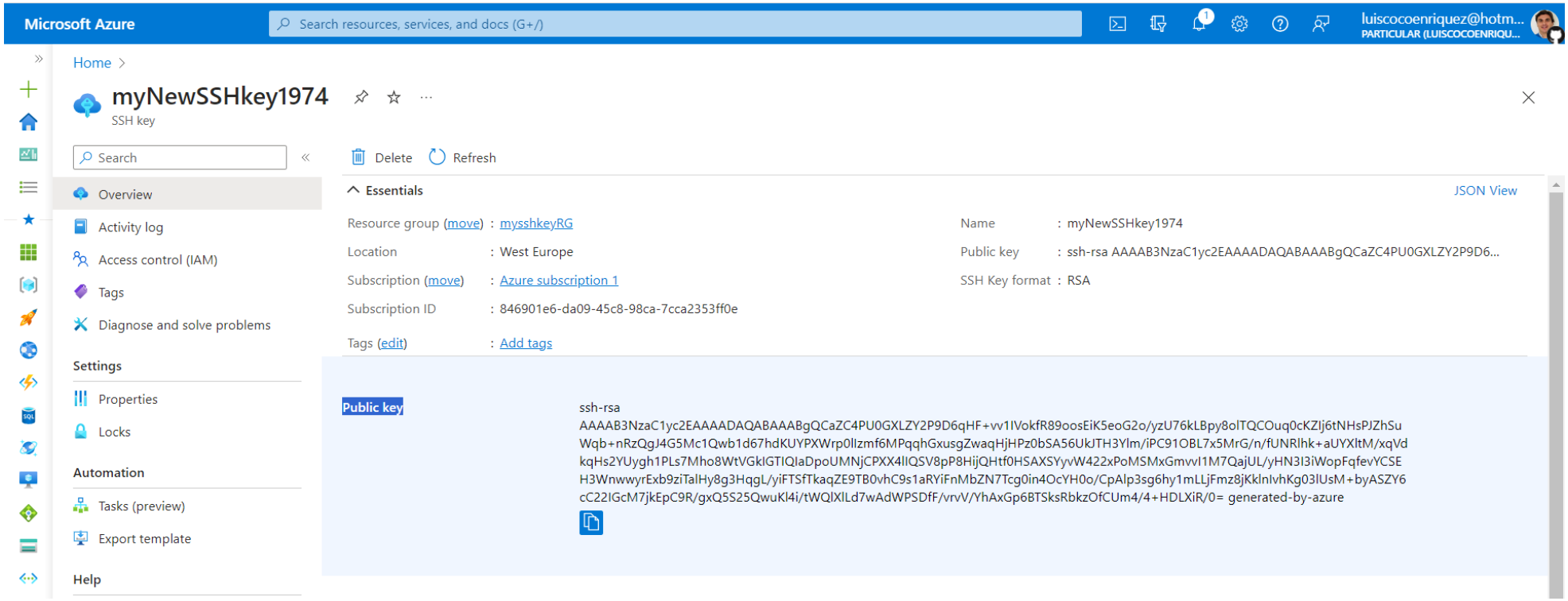

Microsoft Azure
Search resources, services, and docs (G+/)
Home > SSH keys >
Create an SSH key
Basics
Tags
Review + create
Creating an SSH key resource allows you to manage and use public keys stored in Azure with Linux virtual machines.
Learn more
Project details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.
Subscription *
Azure subscription 1
Resource group *
mysshkeyRG
Create new
Instance details
Region
(Europe) West Europe
Key pair name *
myNewSSHkey1974
SSH public key source
Generate new key pair


Review + create
< Previous
Next : Tags >

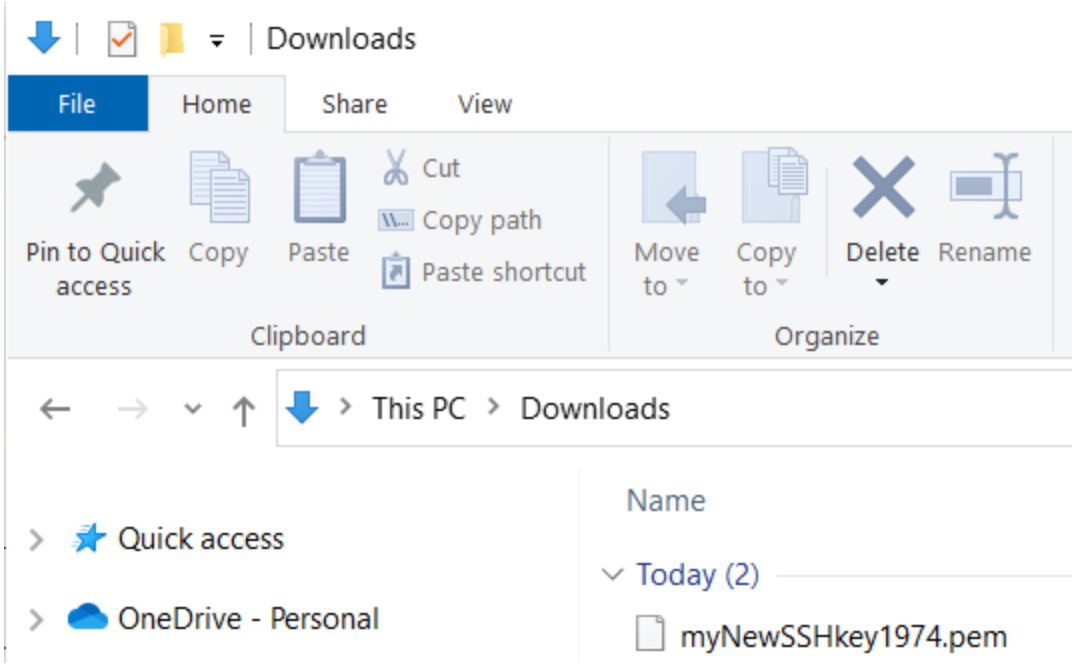

Microsoft Azure
Search resources, services, and docs (G+/)
Home > SSH keys >
Create an SSH key
Validation passed
Basics
Tags
Review + create
Basics
Subscription
Azure subscription 1
Resource group
mysshkeyRG
Region
West Europe
Key pair name
myNewSSHkey1974


Create
< Previous
Next >
Download a template for automation
Microsoft Azure
Search resources, services, and docs (G+/)
Home > SSH keys >
Create an SSH key
Validation passed
Basics
Tags
Review + create
Basics
Subscription
Azure subscription 1
Resource group
mysshkeyRG
Region
West Europe
Key pair name
myNewSSHkey1974
Generate new key pair
An SSH key pair contains both a public key and a private key. Azure doesn't store the private key. After the SSH key resource is created, you won't be able to download the private key again. Learn more
Download private key and create resource
Return to create an SSH key resource

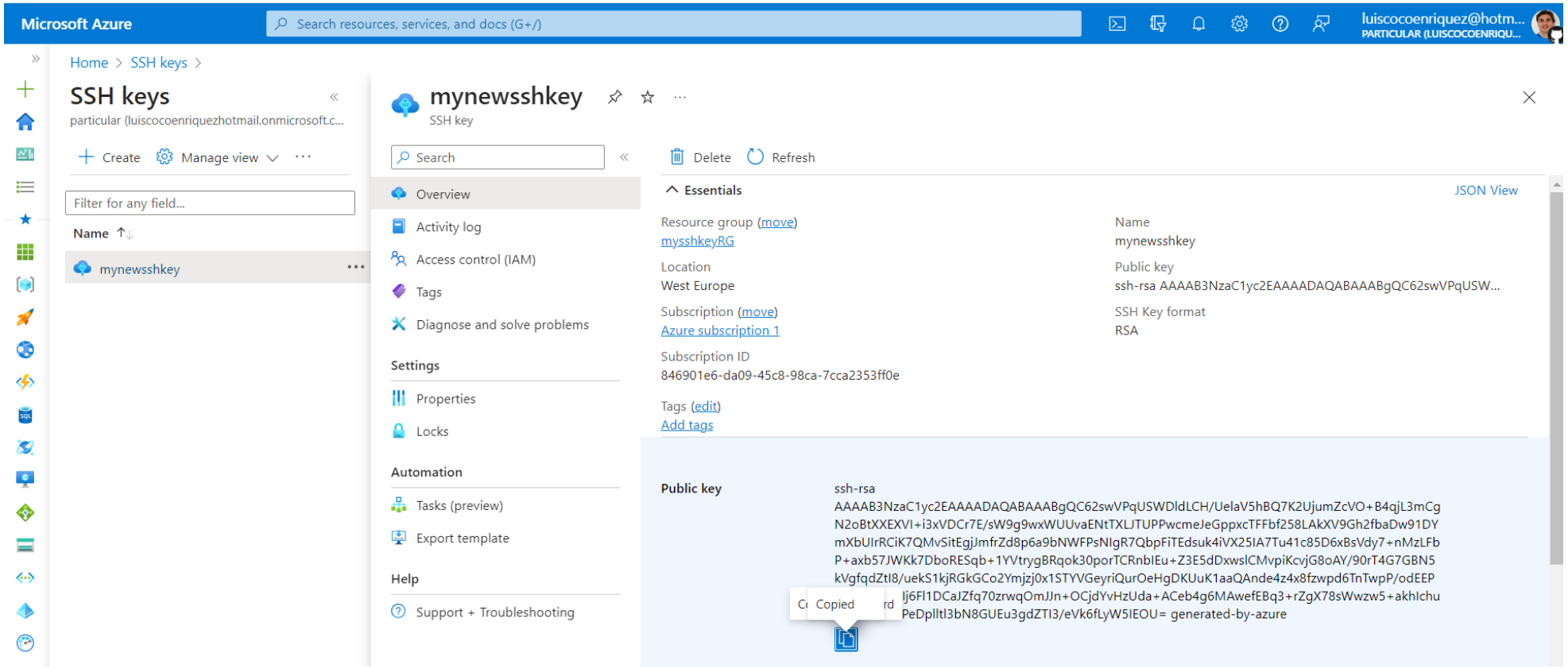
The PUBLIC key is available in the SSH key pair Azure resource

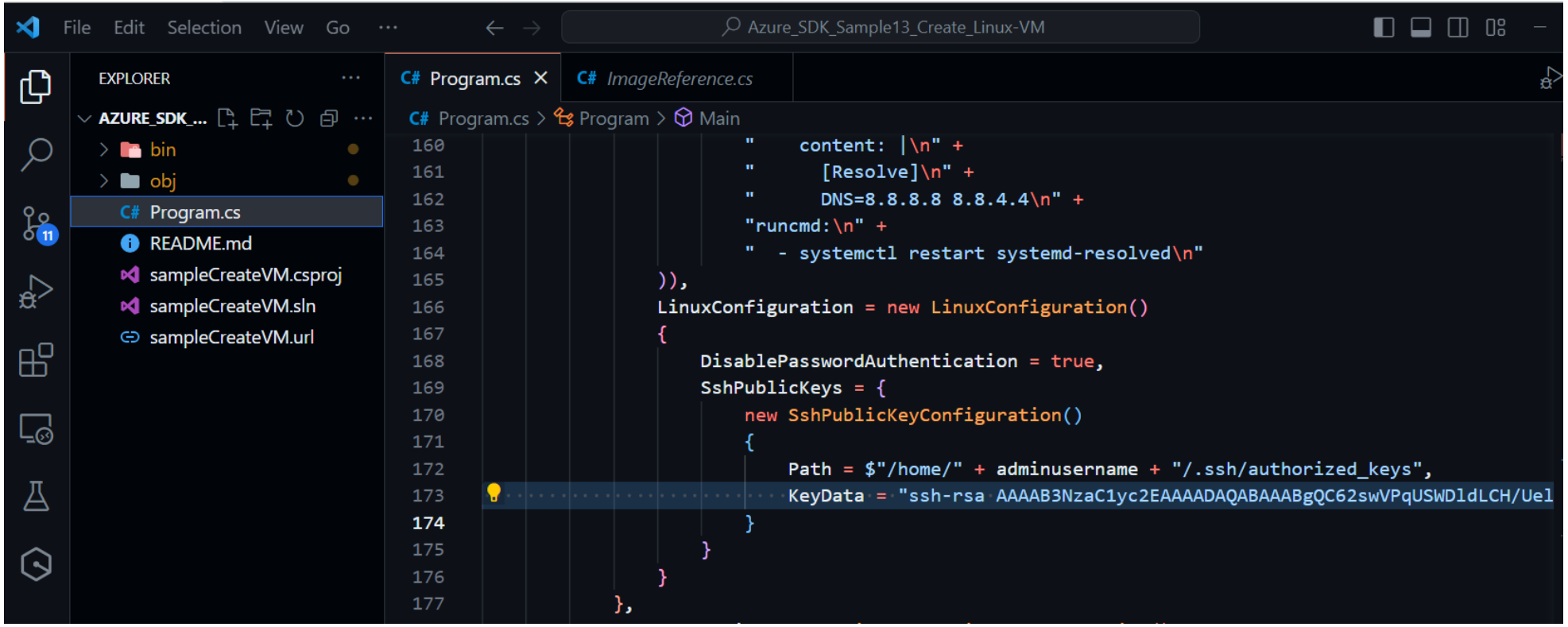


The PRIVATE key *.pem file is downloaded



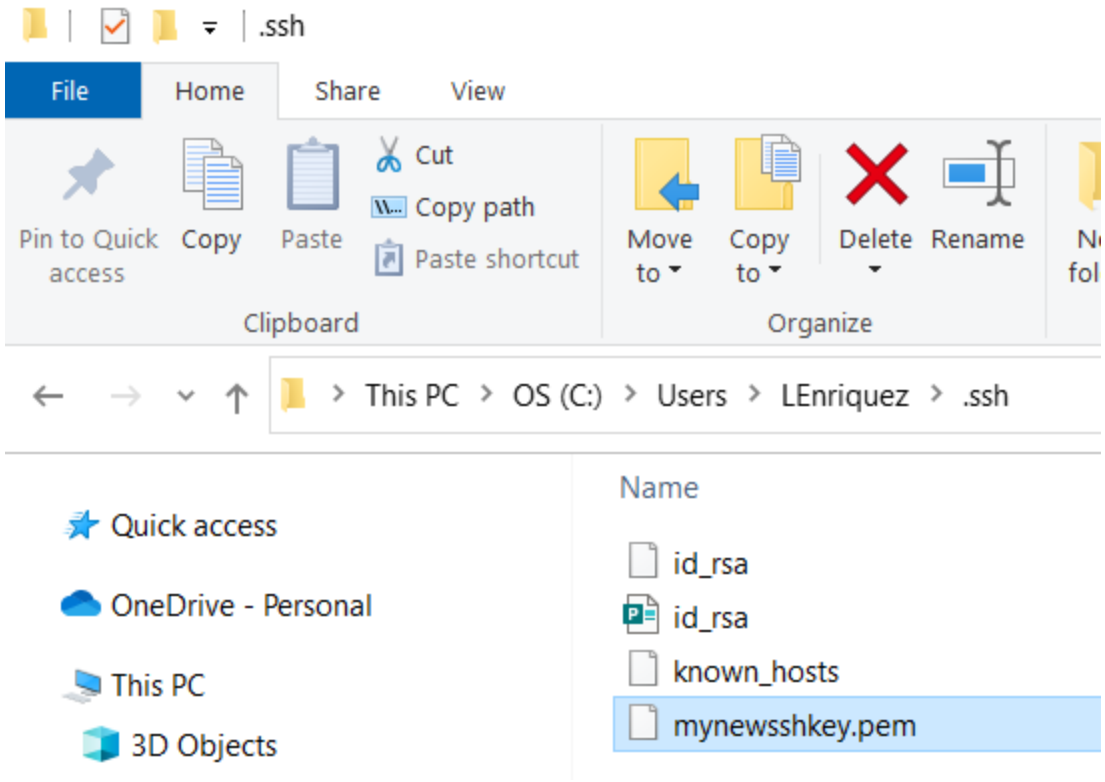
5. Copy the PUBLIC key in the C# source code

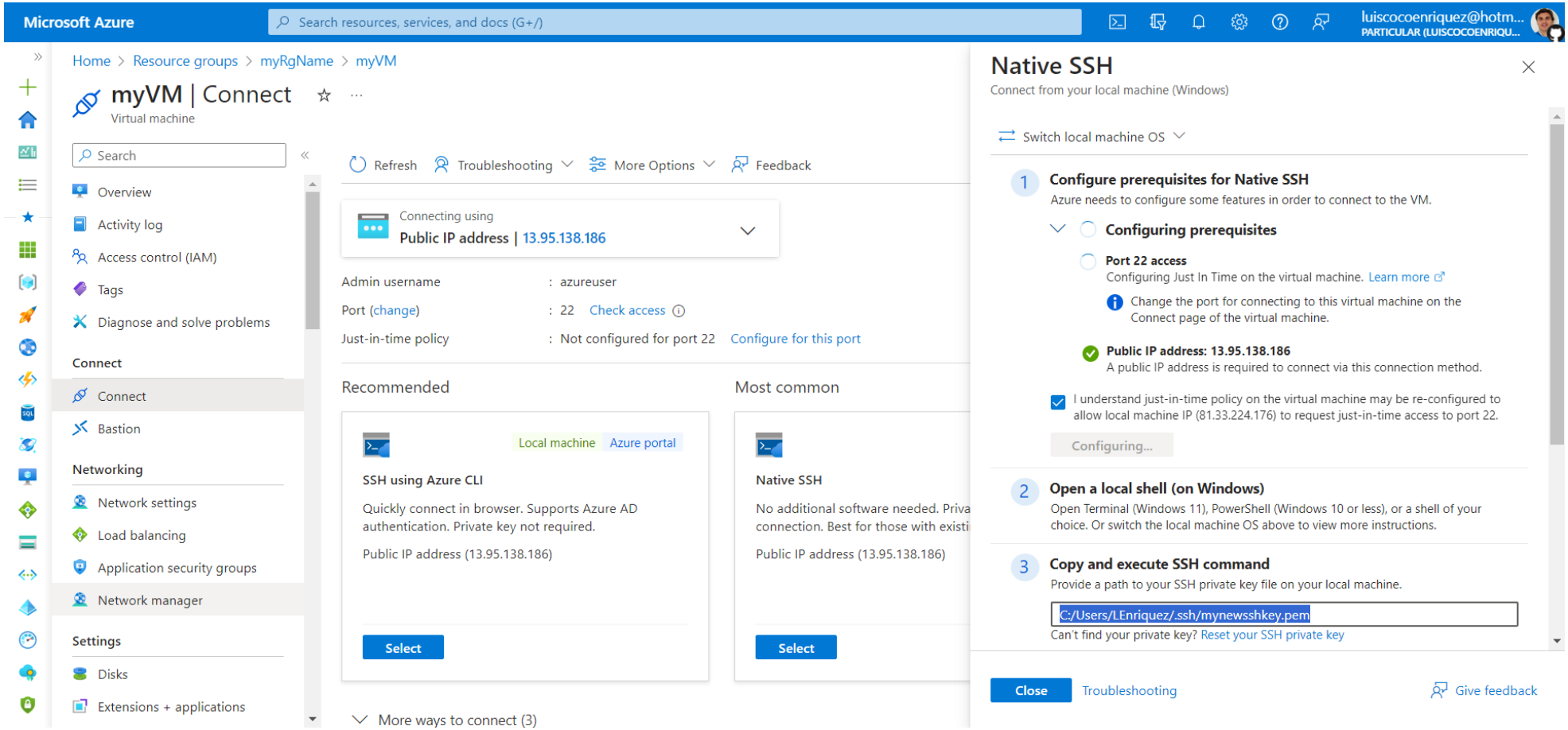

Microsoft Azure
Search resources, services, and docs (G+/)
luiscocoenriquez@hotm...
PARTICULAR (LUISCOCOENRIQU...
Home > SSH keys >
SSH keys
particular (luiscocoenriquezhotmail.onmicrosoft.c...
Create
Manage view
Filter for any field...
Name
mynewsshkey
mynewsshkey
SSH key
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Properties
Locks
Automation
Tasks (preview)
Export template
Help
Support + Troubleshooting
Delete
Refresh
Essentials
JSON View
Resource group (move)
mysshkeyRG
Location
West Europe
Subscription (move)
Azure subscription 1
Subscription ID
846901e6-da09-45c8-98ca-7cca2353ff0e
Tags (edit)
Add tags
Name
mynewsshkey
Public key
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC62swVPqUSW...
SSH Key format
RSA
Public key
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQC62swVPqUSWDldLCH/UelaV5hBQ7K2UjumZcVO+B4qjL3mCg
N2oBtXXEXVI+i3xVDCr7E/sW9g9wxWUUvaENtTXLJTUPPwcmeJeGppxcTFFbf258LAkXV9Gh2fbaDw91DY
mXbUIrRCiK7QMvSitEgjJmfrZd8p6a9bNWFPsNIgR7QbpFiTEdsuk4iVX25IA7Tu41c85D6xBsVdy7+nMzLFb
P+axb57JWKk7DboRESqb+1YVtrygBRqok30porTCRnbIEu+Z3E5dDxwslCMvpiKcvjG8oAY/90rT4G7GBN5
kVgfqdZtI8/uekS1kjRGkGCo2Ymjzj0x1STYVGeyriQurOeHgDKUuK1aaQAnde4z4x8fzwpd6TnTwpP/odEEP
lj6Fl1DCaJZfq70zrwqOmJJn+OCjdYvHzUda+ACeb4g6MAwefEBq3+rZgX78sWwzw5+akhIchu
PeDpIltl3bN8GUEu3gdZTI3/eVk6fLyW5IEOU= generated-by-azure
Copied



6. Download the PRIVATE key *.pem file

Copy the private key file (*.pem) and paste it in: "C:\Users\LEnriquez.ssh"


.ssh
File
Home
Share
View
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Copy to
Delete
Rename
Organize
This PC
OS (C:)
Users
LEnriquez
.ssh
Quick access
OneDrive - Personal
This PC
3D Objects
Name
id_rsa
id_rsa
known_hosts
mynewsshkey.pem

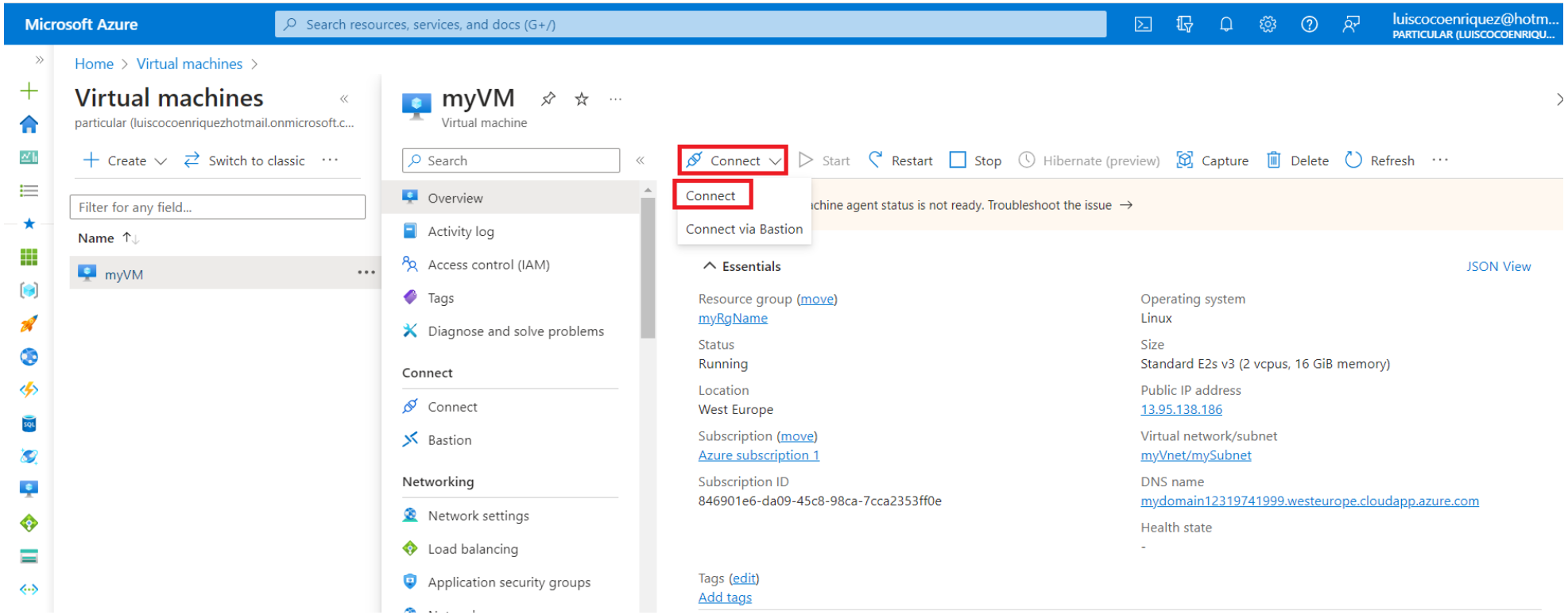


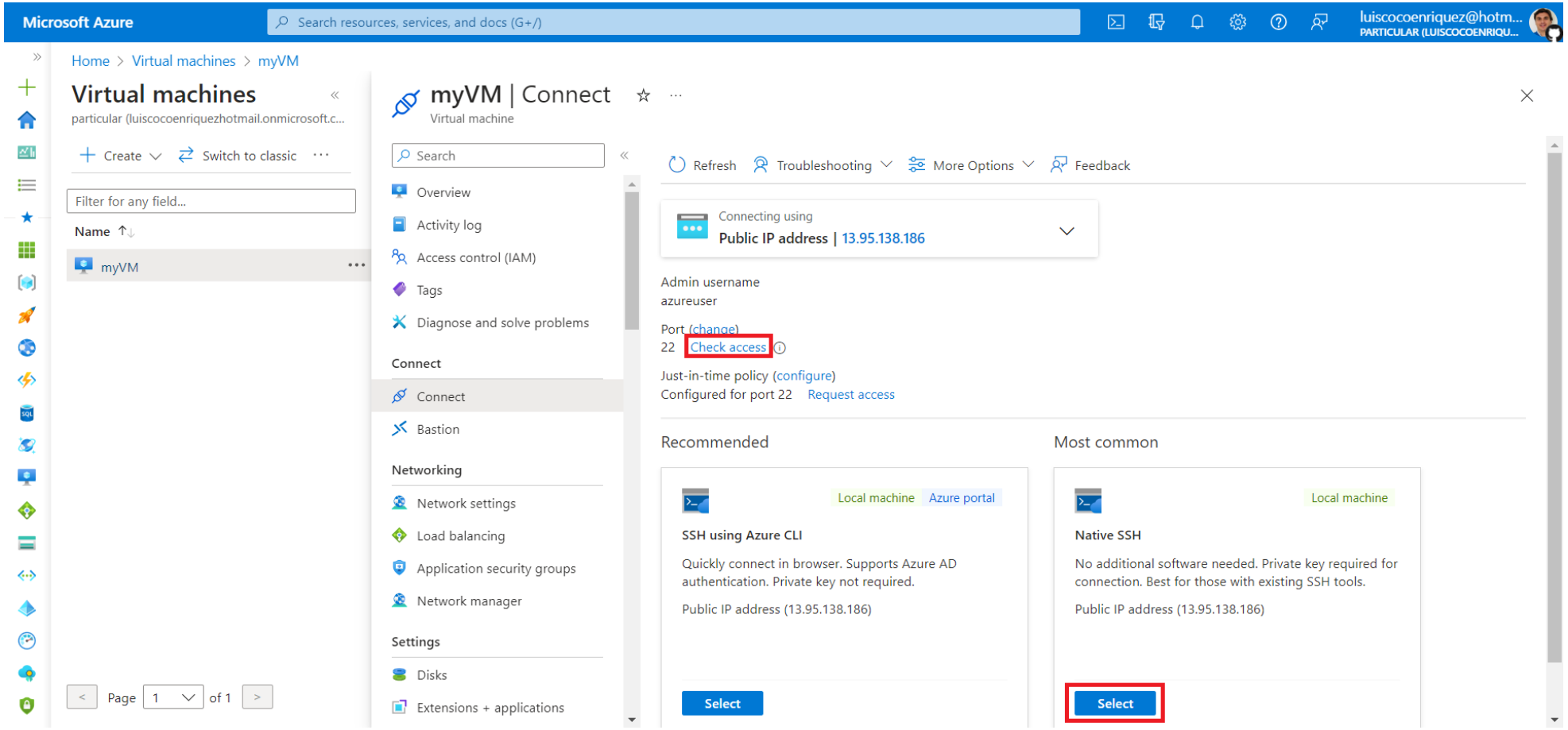
7. Build and run the application

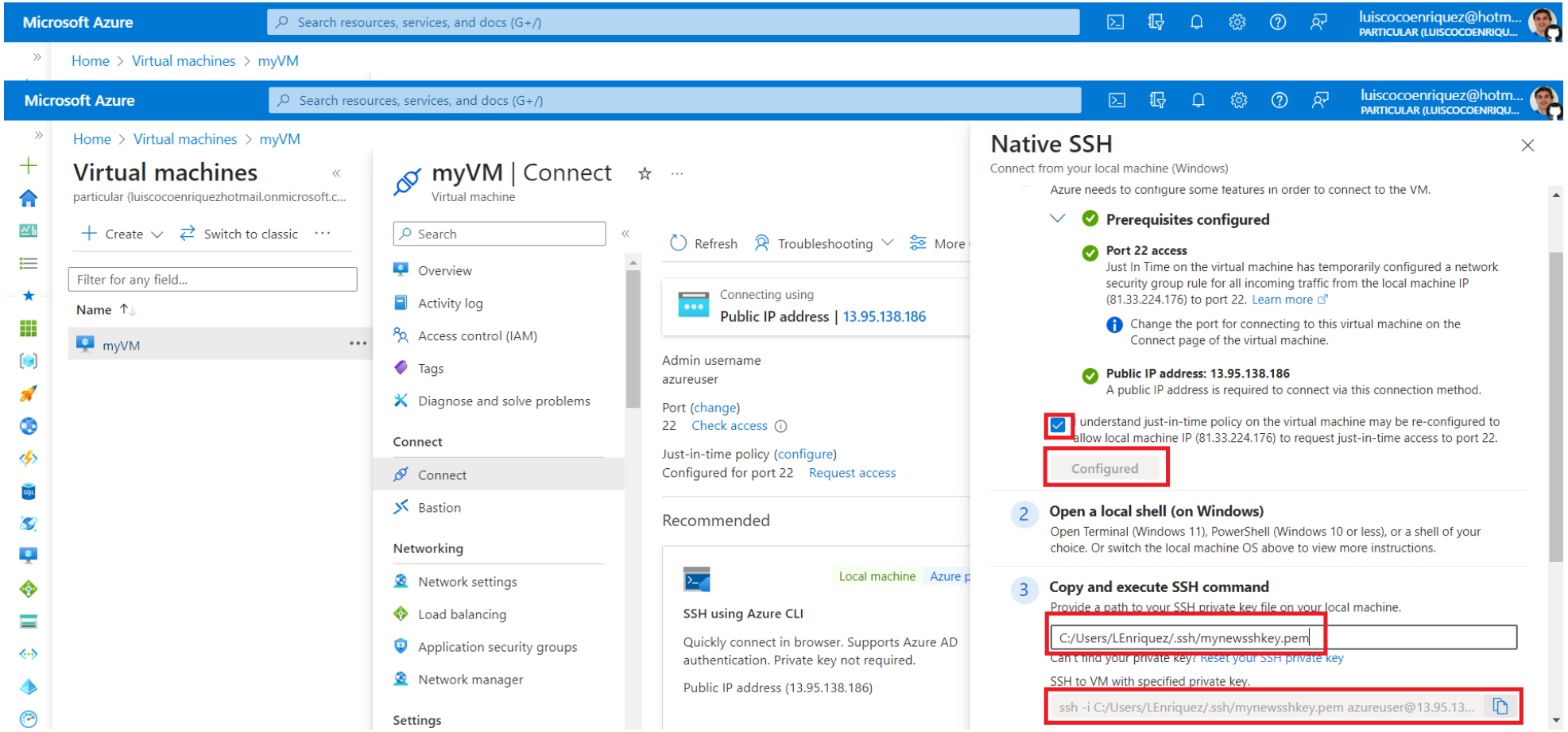
Execute the command:

```
dotnet run
```

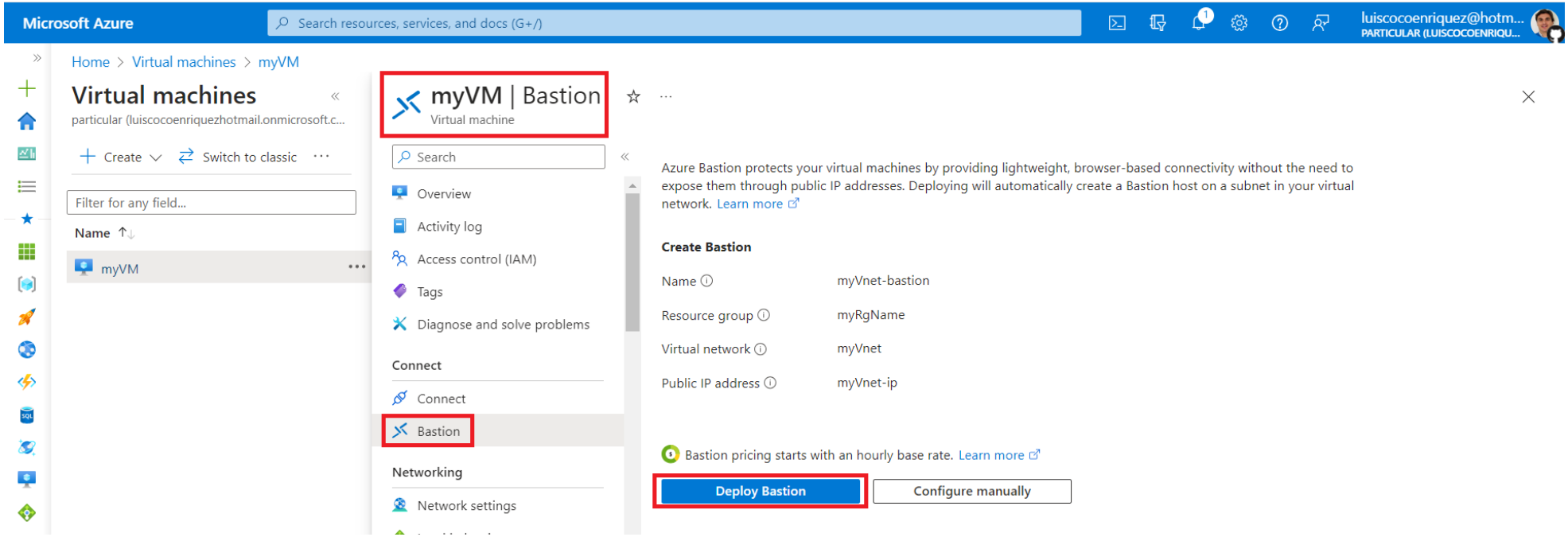
8. Access to the Virtual Machine

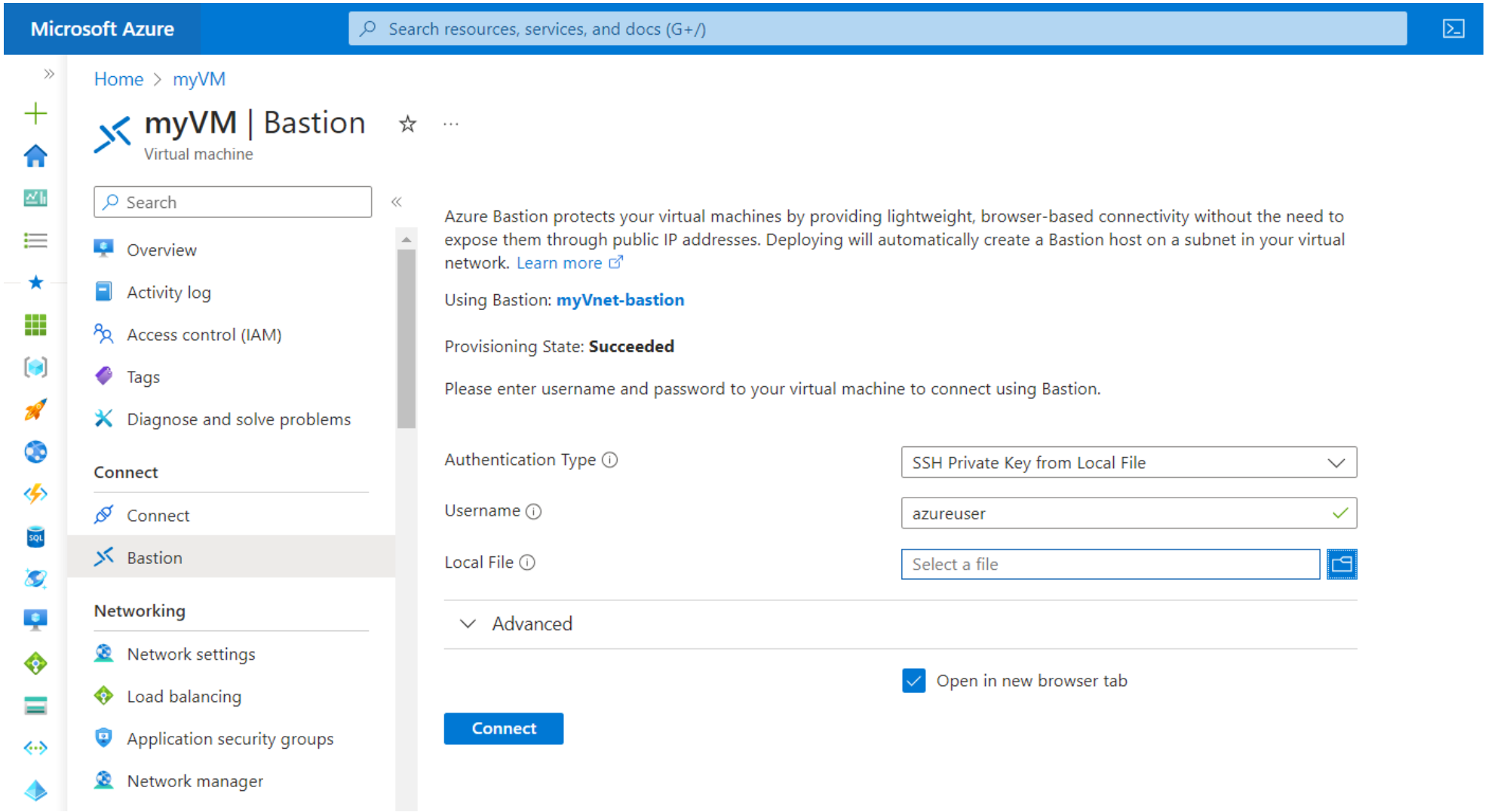

Microsoft Azure
Search resources, services, and docs (G+/)
luiscocoenriquez@hotm...
PARTICULAR (LUISCOCOENRIQU...
Home > Virtual machines >
Virtual machines
particular (luiscocoenriquezhotmail.onmicrosoft.c...
Create
Switch to classic
Filter for any field...
Name
myVM
myVM
Virtual machine
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Connect
Connect
Bastion
Networking
Network settings
Load balancing
Application security groups
Connect
Start
Restart
Stop
Hibernate (preview)
Capture
Delete
Refresh
Connect
Connect via Bastion
chine agent status is not ready. Troubleshoot the issue →
Essentials
JSON View
Resource group (move)
myRgName
Status
Running
Location
West Europe
Subscription (move)
Azure subscription 1
Subscription ID
846901e6-da09-45c8-98ca-7cca2353ff0e
Operating system
Linux
Size
Standard E2s v3 (2 vcpus, 16 GiB memory)
Public IP address
13.95.138.186
Virtual network/subnet
myVnet/mySubnet
DNS name
mydomain12319741999.westeurope.cloudapp.azure.com
Health state
-
Tags (edit)
Add tags


Microsoft Azure
Search resources, services, and docs (G+/)
luiscocoenriquez@hotm...
PARTICULAR (LUISCOCOENRIQU...
Home > Virtual machines > myVM
Virtual machines
particular (luiscocoenriquezhotmail.onmicrosoft.c...
Create
Switch to classic
Filter for any field...
Name
myVM
Page 1 of 1
myVM | Connect
Virtual machine
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Connect
Connect
Bastion
Networking
Network settings
Load balancing
Application security groups
Network manager
Settings
Disks
Extensions + applications
Refresh
Troubleshooting
More Options
Feedback
Connecting using
Public IP address | 13.95.138.186
Admin username
azureuser
Port (change)
22 Check access
Just-in-time policy (configure)
Configured for port 22 Request access
Recommended
Most common
Local machine
Azure portal
SSH using Azure CLI
Quickly connect in browser. Supports Azure AD authentication. Private key not required.
Public IP address (13.95.138.186)
Select
Local machine
Native SSH
No additional software needed. Private key required for connection. Best for those with existing SSH tools.
Public IP address (13.95.138.186)
Select

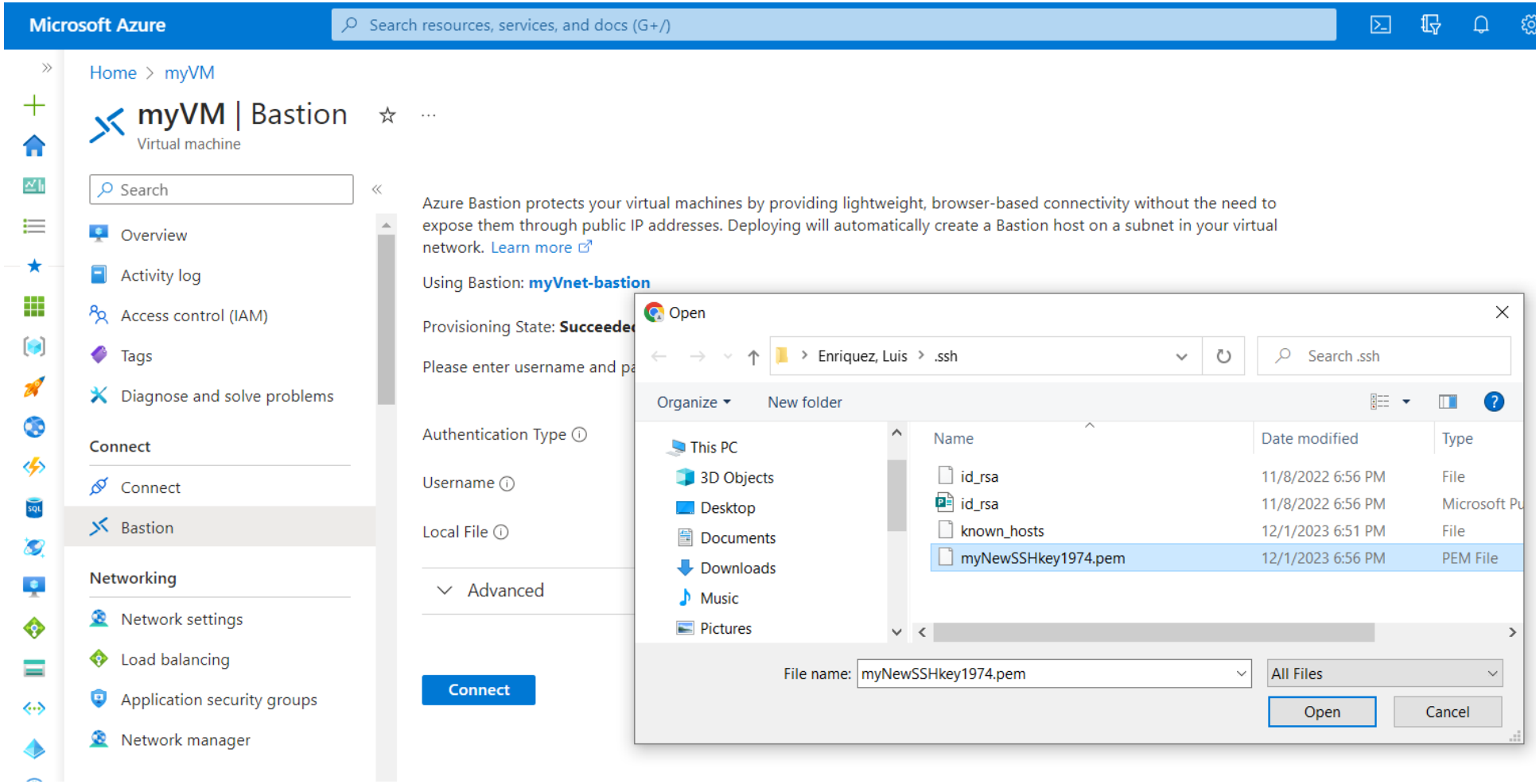


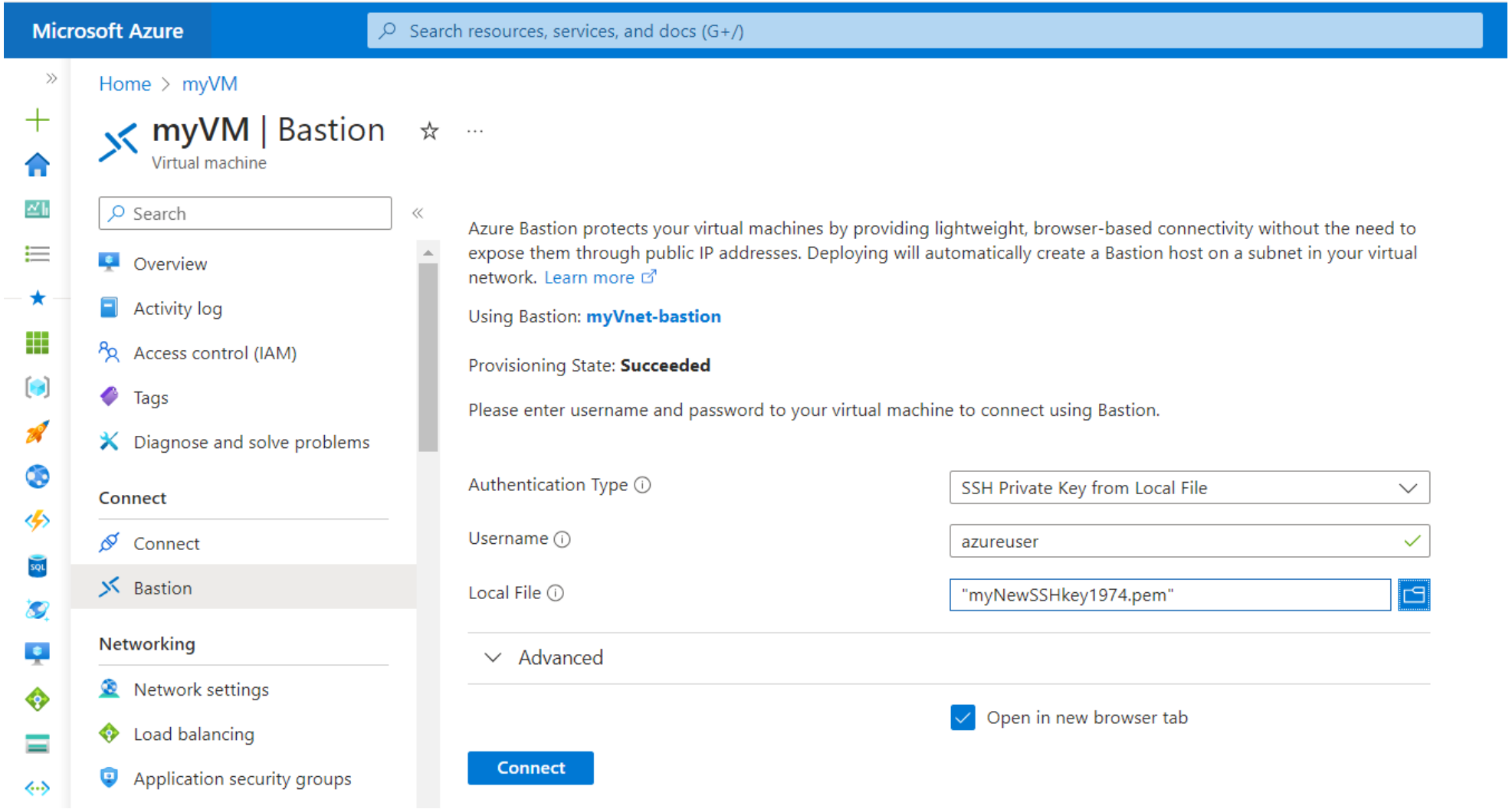
9. Access with bastion


Microsoft Azure
Search resources, services, and docs (G+/)
luiscocoenriquez@hotm...
PARTICULAR (LUISCOCOENRIQU...
Home > Virtual machines > myVM
Virtual machines
particular (luiscocoenriquezhotmail.onmicrosoft.c...
Create
Switch to classic
Filter for any field...
Name
myVM
myVM | Bastion
Virtual machine
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Connect
Connect
Bastion
Networking
Network settings
Azure Bastion protects your virtual machines by providing lightweight, browser-based connectivity without the need to expose them through public IP addresses. Deploying will automatically create a Bastion host on a subnet in your virtual network. Learn more
Create Bastion
Name
myVnet-bastion
Resource group
myRgName
Virtual network
myVnet
Public IP address
myVnet-ip
Bastion pricing starts with an hourly base rate. Learn more
Deploy Bastion
Configure manually



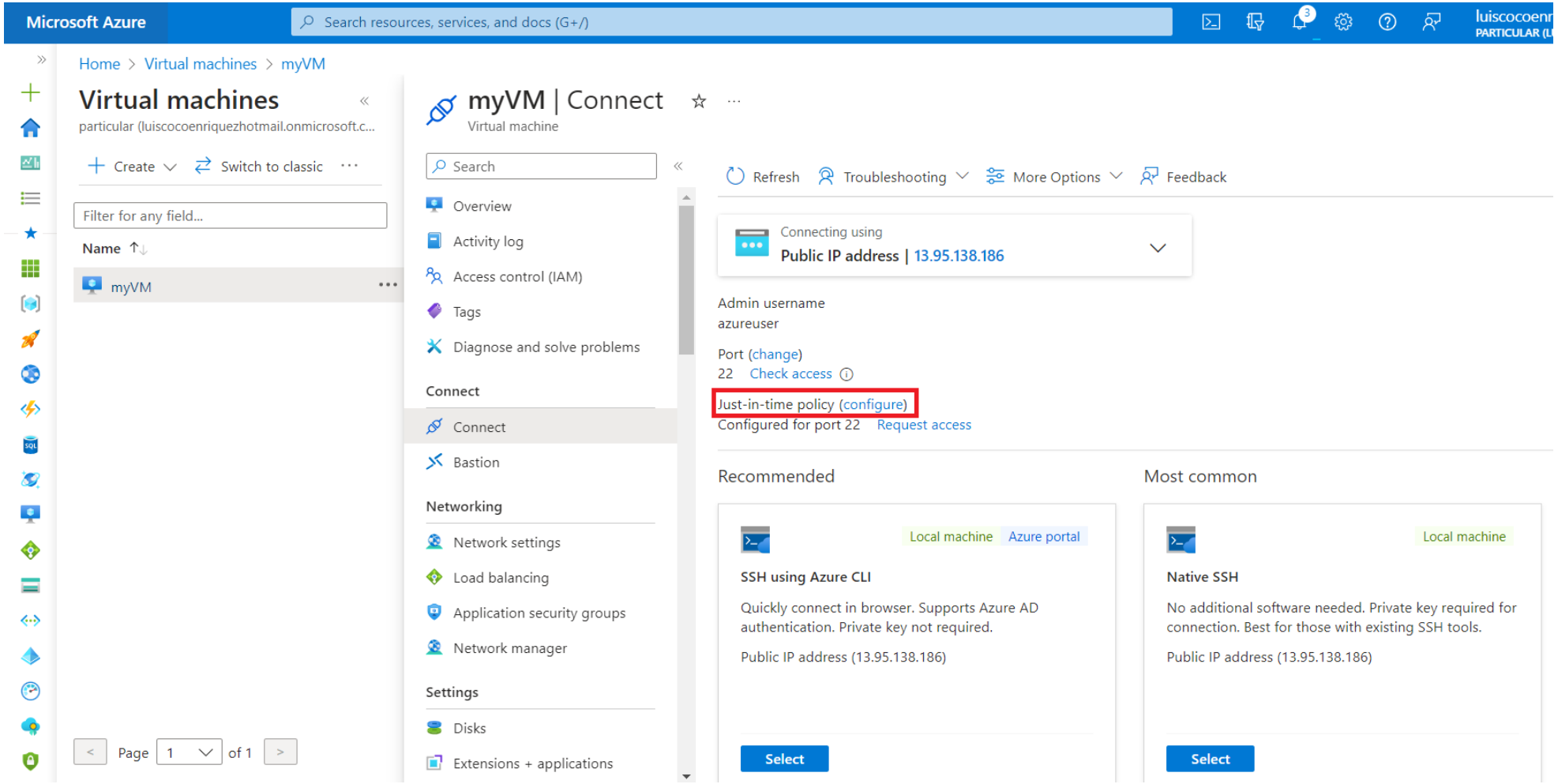
Then we upload the private key file

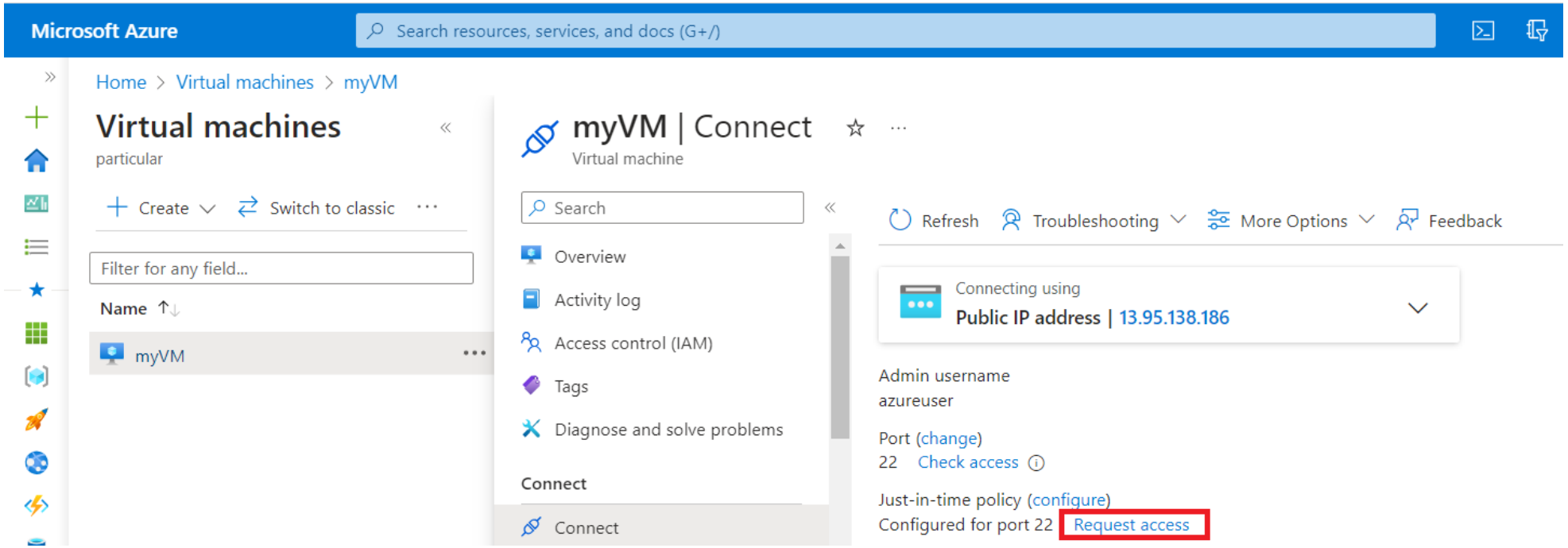

Microsoft Azure
Search resources, services, and docs (G+/)
Home > myVM
myVM | Bastion
Virtual machine
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Connect
Connect
Bastion
Networking
Network settings
Load balancing
Application security groups
Network manager
Azure Bastion protects your virtual machines by providing lightweight, browser-based connectivity without the need to expose them through public IP addresses. Deploying will automatically create a Bastion host on a subnet in your virtual network. Learn more
Using Bastion: myVnet-bastion
Provisioning State: Succeede
Please enter username and pa
Authentication Type
Username
Local File
Advanced
Connect
Open
Enriquez, Luis > .ssh
Search .ssh
Organize
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Name
Date modified
Type
id_rsa
11/8/2022 6:56 PM
File
id_rsa
11/8/2022 6:56 PM
Microsoft Pu
known_hosts
12/1/2023 6:51 PM
File
myNewSSHkey1974.pem
12/1/2023 6:56 PM
PEM File
File name: myNewSSHkey1974.pem
All Files
Open
Cancel

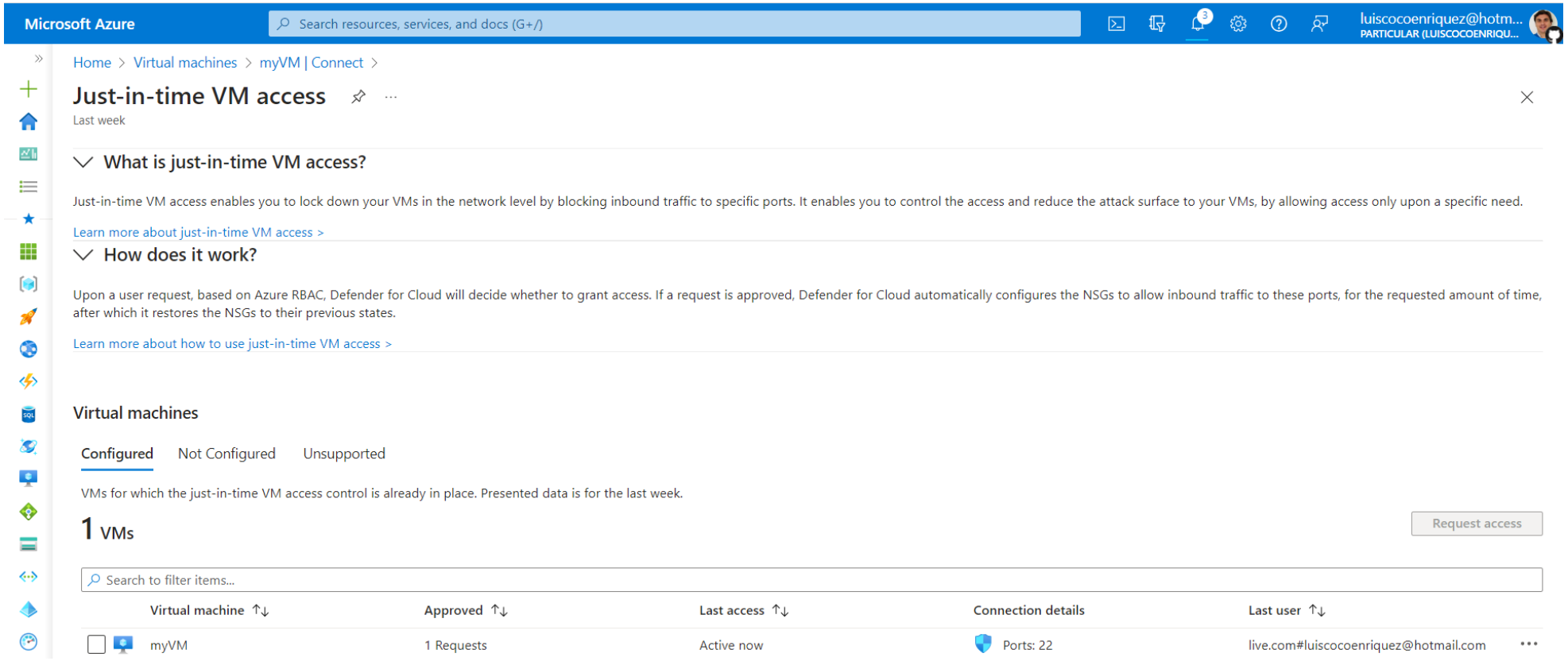


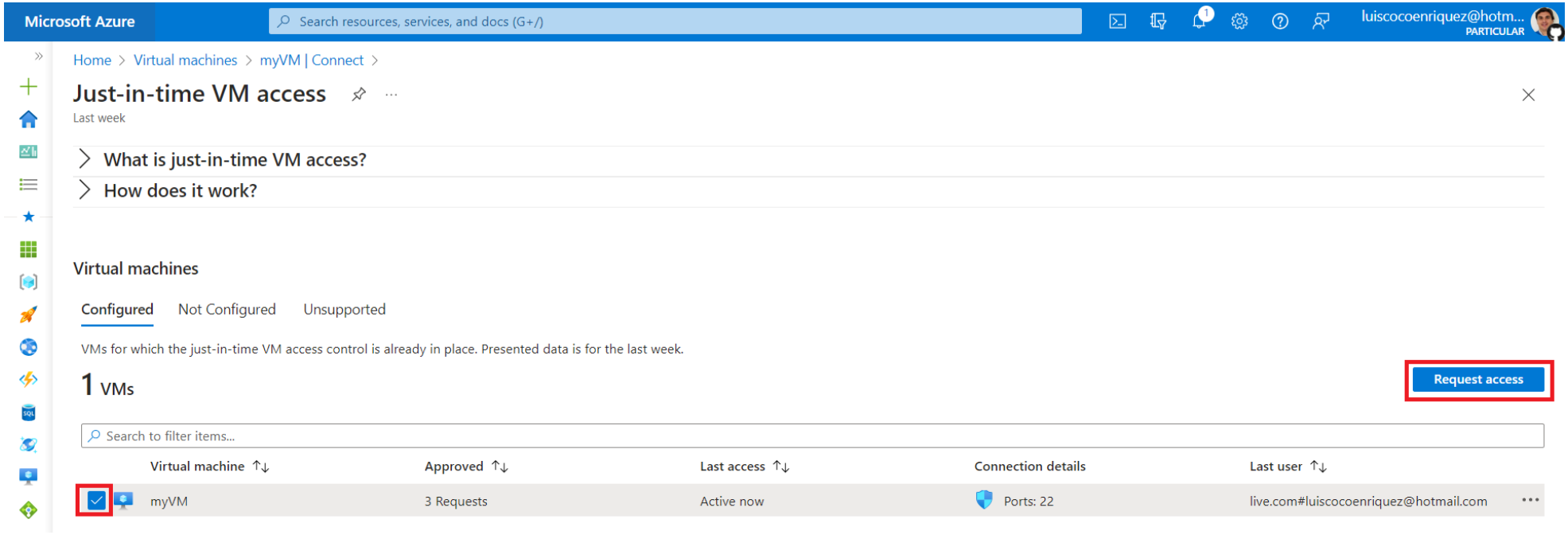
The we press the "**Connect**" button. A new internet web browser window will be opened in a new tab with the Linux Virtual Machine connection.

10. Just-in-time policy required Microsoft Defender

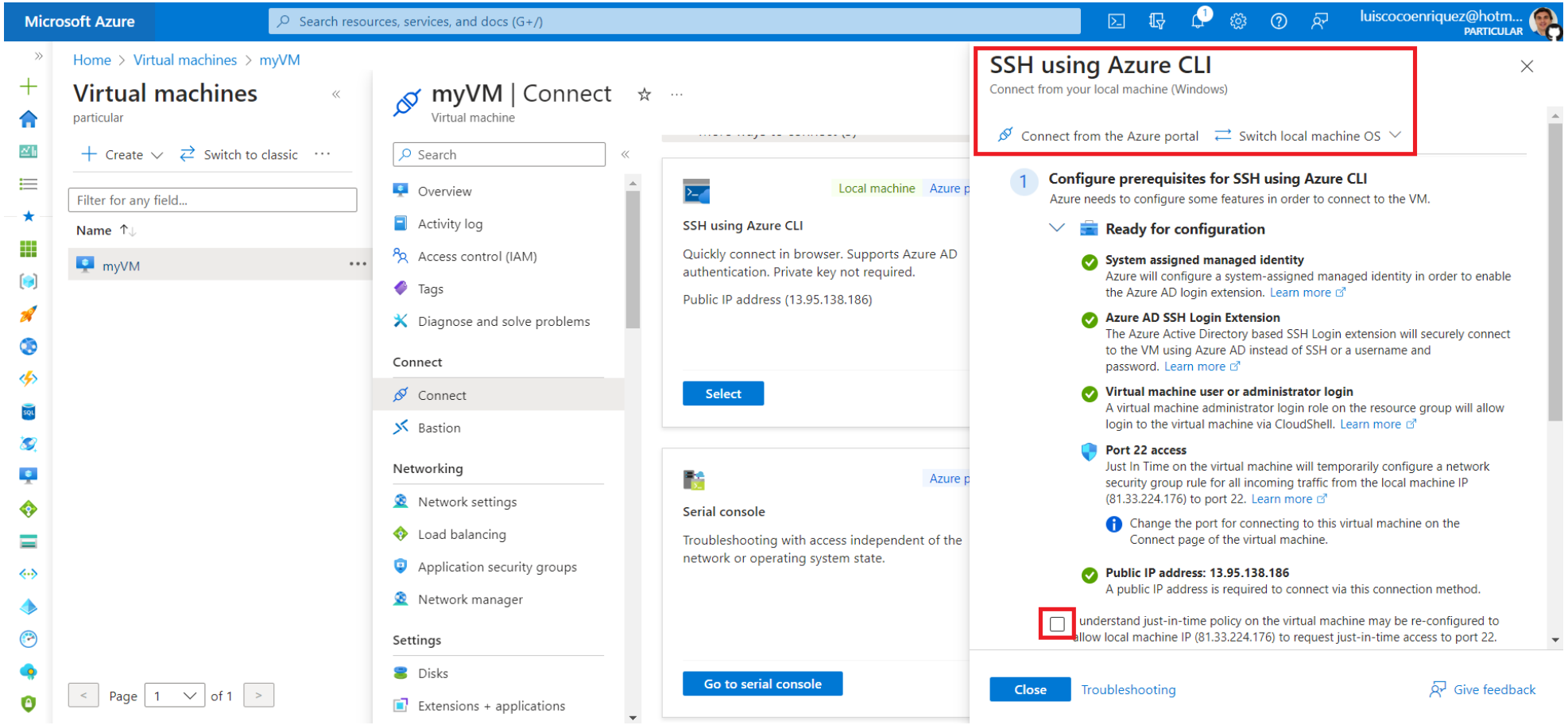

Microsoft Azure
Search resources, services, and docs (G+/)
Home > Virtual machines > myVM
Virtual machines
particular (luiscocoenriquezhotmail.onmicrosoft.c...
Create
Switch to classic
Filter for any field...
Name
myVM
Page 1 of 1
myVM | Connect
Virtual machine
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Connect
Connect
Bastion
Networking
Network settings
Load balancing
Application security groups
Network manager
Settings
Disks
Extensions + applications
Refresh
Troubleshooting
More Options
Feedback
Connecting using
Public IP address | 13.95.138.186
Admin username
azureuser
Port (change)
22 Check access
Just-in-time policy (configure)
Configured for port 22 Request access
Recommended
Most common
Local machine
Azure portal
SSH using Azure CLI
Quickly connect in browser. Supports Azure AD authentication. Private key not required.
Public IP address (13.95.138.186)
Select
Local machine
Native SSH
No additional software needed. Private key required for connection. Best for those with existing SSH tools.
Public IP address (13.95.138.186)
Select

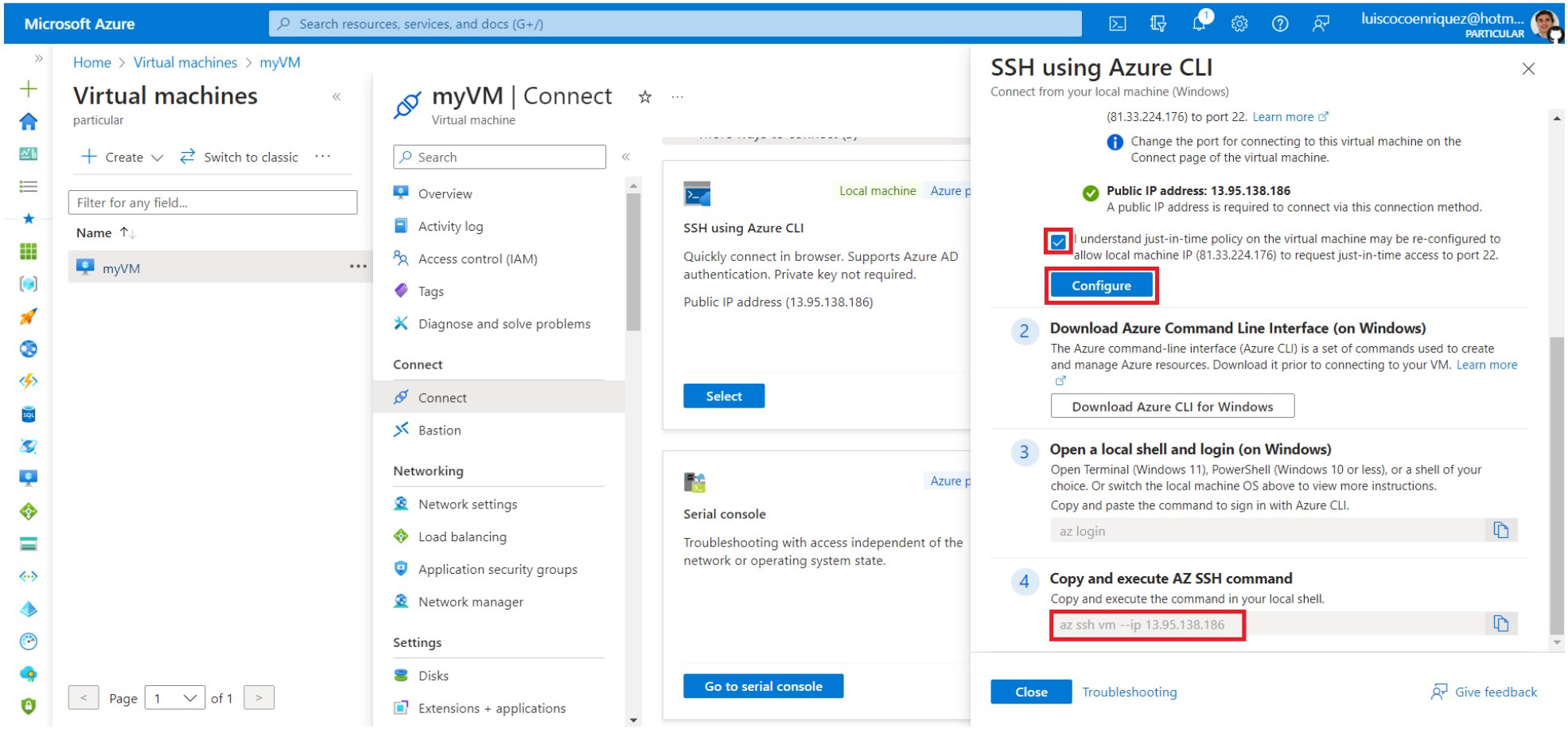

Microsoft Azure
Search resources, services, and docs (G+/)
Home > Virtual machines > myVM
Virtual machines
particular
Create
Switch to classic
Filter for any field...
Name
myVM
myVM | Connect
Virtual machine
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Connect
Connect
Refresh
Troubleshooting
More Options
Feedback
Connecting using
Public IP address | 13.95.138.186
Admin username
azureuser
Port (change)
22 Check access
Just-in-time policy (configure)
Configured for port 22 Request access


Microsoft Azure
Search resources, services, and docs (G+/)
luiscocoenriquez@hotm...
PARTICULAR (LUISCOCOENRIQU...
Home > Virtual machines > myVM | Connect >
Just-in-time VM access
Last week
What is just-in-time VM access?
Just-in-time VM access enables you to lock down your VMs in the network level by blocking inbound traffic to specific ports. It enables you to control the access and reduce the attack surface to your VMs, by allowing access only upon a specific need.
Learn more about just-in-time VM access >
How does it work?
Upon a user request, based on Azure RBAC, Defender for Cloud will decide whether to grant access. If a request is approved, Defender for Cloud automatically configures the NSGs to allow inbound traffic to these ports, for the requested amount of time, after which it restores the NSGs to their previous states.
Learn more about how to use just-in-time VM access >
Virtual machines
Configured
Not Configured
Unsupported
VMs for which the just-in-time VM access control is already in place. Presented data is for the last week.
1 VMs
Request access
Search to filter items...
Virtual machine
Approved
Last access
Connection details
Last user
myVM
1 Requests
Active now
Ports: 22
live.com#luiscocoenriquez@hotmail.com



After requesting the access with the Just-in-time policy we can use the Azure CLI for accessing the Linux VM.


Microsoft Azure
Search resources, services, and docs (G+/)
luiscocoenriquez@hotm...
PARTICULAR
Home > Virtual machines > myVM
Virtual machines
particular
Create
Switch to classic
Filter for any field...
Name
myVM
Page 1 of 1
myVM | Connect
Virtual machine
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Connect
Connect
Bastion
Networking
Network settings
Load balancing
Application security groups
Network manager
Settings
Disks
Extensions + applications
Local machine
SSH using Azure CLI
Quickly connect in browser. Supports Azure AD authentication. Private key not required.
Public IP address (13.95.138.186)
Select
Serial console
Troubleshooting with access independent of the network or operating system state.
Go to serial console
SSH using Azure CLI
Connect from your local machine (Windows)
Connect from the Azure portal
Switch local machine OS
1
Configure prerequisites for SSH using Azure CLI
Azure needs to configure some features in order to connect to the VM.
Ready for configuration
System assigned managed identity
Azure will configure a system-assigned managed identity in order to enable the Azure AD login extension. Learn more
Azure AD SSH Login Extension
The Azure Active Directory based SSH Login extension will securely connect to the VM using Azure AD instead of SSH or a username and password. Learn more
Virtual machine user or administrator login
A virtual machine administrator login role on the resource group will allow login to the virtual machine via CloudShell. Learn more
Port 22 access
Just In Time on the virtual machine will temporarily configure a network security group rule for all incoming traffic from the local machine IP (81.33.224.176) to port 22. Learn more
Change the port for connecting to this virtual machine on the Connect page of the virtual machine.
Public IP address: 13.95.138.186
A public IP address is required to connect via this connection method.
understand just-in-time policy on the virtual machine may be re-configured to allow local machine IP (81.33.224.176) to request just-in-time access to port 22.
Close
Troubleshooting
Give feedback



We can connect to the Linux VM with these commands:

```
az ssh vm --ip 13.95.138.186
```

or

```
az ssh vm --resource-group myRgName --name myVM --subscription 846901e6-da09-45c8-98ca-7cca2353ff0e
```

11. Acces from Linux Virtual Machine to Internet

Be sure your Linux Virtual Machine has access to internet. To do so run the following commands:

```
ping 8.8.8.8
```

Also try to run this command:

```
nslookup google.com
```

If you cannot connect to internet then run the command:

```
sudo nano /etc/systemd/resolved.conf
```

Then uncomment the line:

```
[Resolve]
DNS=8.8.8.8 8.8.4.4
```

The file will be like this:

```
#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or modify it under the
#  terms of the GNU Lesser General Public License as published by the Free
#  Software Foundation; either version 2.1 of the License, or (at your option)
#  any later version.
#
# Entries in this file show the compile time defaults. Local configuration
# should be created by either modifying this file, or by creating "drop-ins" in
# the resolved.conf.d/ subdirectory. The latter is generally recommended.
# Defaults can be restored by simply deleting this file and all drop-ins.
```

```
#
# Use 'systemd-analyze cat-config systemd/resolved.conf' to display the full config.
#
# See resolved.conf(5) for details.

[Resolve]
# Some examples of DNS servers which may be used for DNS= and FallbackDNS=:
# Cloudflare: 1.1.1.1#cloudflare-dns.com 1.0.0.1#cloudflare-dns.com 2606:4700:4700::1111#cloudflare-dns.com 2606:4700:4>
# Quad9:      9.9.9.9#dns.quad9.net 149.112.112.112#dns.quad9.net 2620:fe::fe#dns.quad9.net 2620:fe::9#dns.quad9.net
DNS=8.8.8.8 8.8.4.4
#FallbackDNS=
#Domains=
#DNSSEC=no
#DNSOverTLS=no
```

Then press **Ctrl + o** for saving the file, and then press **Enter** and finally press **Ctrl + x** to exit the nano editor

Then type the command:

```
sudo systemctl restart systemd-resolved
```

Then try again to connect with the command:

```
nslookup google.com
```

You should get this output:

```
azureuser@myVM:~$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
```

```
Address: 142.251.36.14
Name:   google.com
Address: 2a00:1450:400e:811::200e
```

12. How to install in the Ubuntu Server the GUI Desktop, VSCode, Google Chrome and .NET 8

12.1. Run these commands to stall "xfce" using "apt":

```
sudo apt-get update
sudo DEBIAN_FRONTEND=noninteractive apt-get -y install xfce4
sudo apt install xfce4-session
```

12.2. Install and configure a remote desktop server:

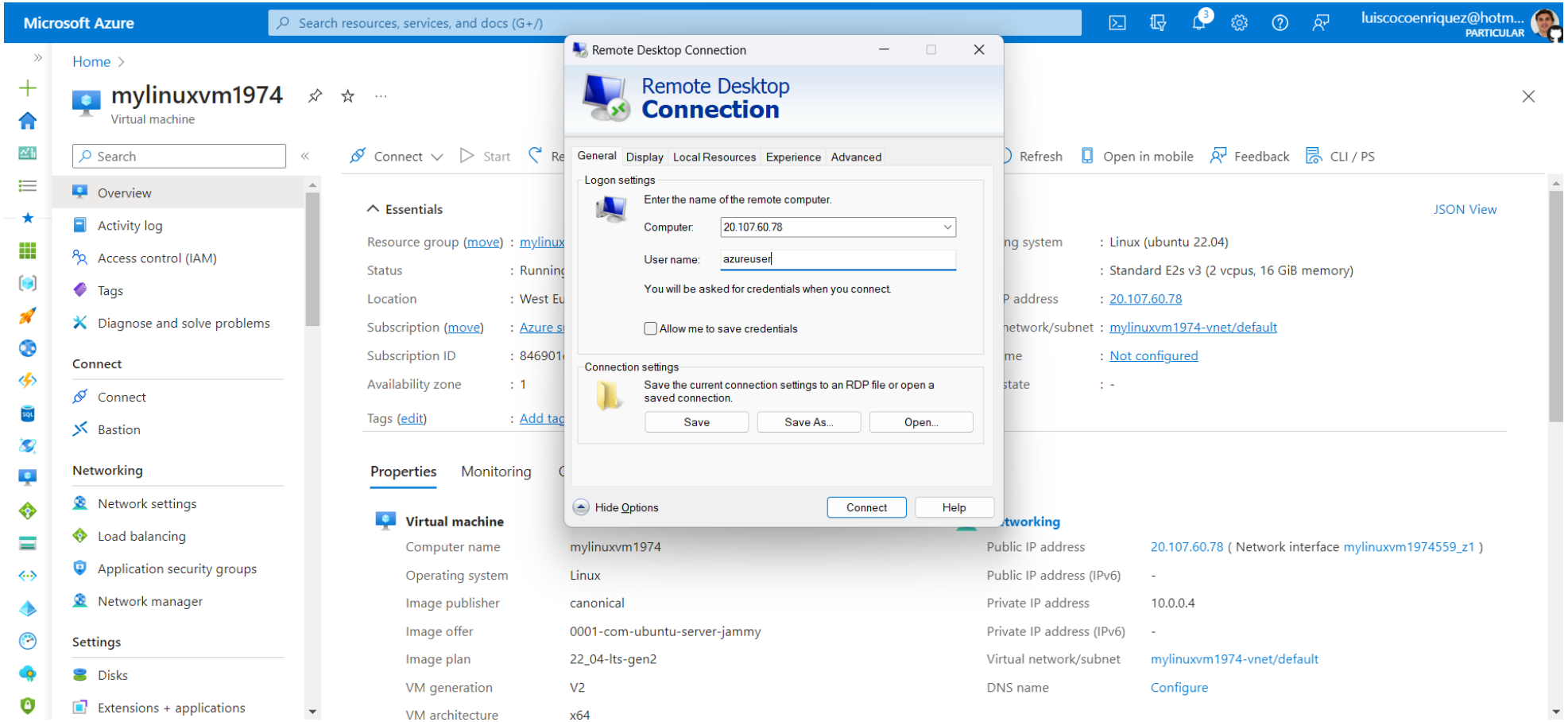
```
sudo apt-get -y install xrdp
sudo systemctl enable xrdp
sudo adduser xrdp ssl-cert
echo xfce4-session >~/.xsession
sudo service xrdp restart
```

12.3. Set a local user account password, for example:"Thismypassword123456"

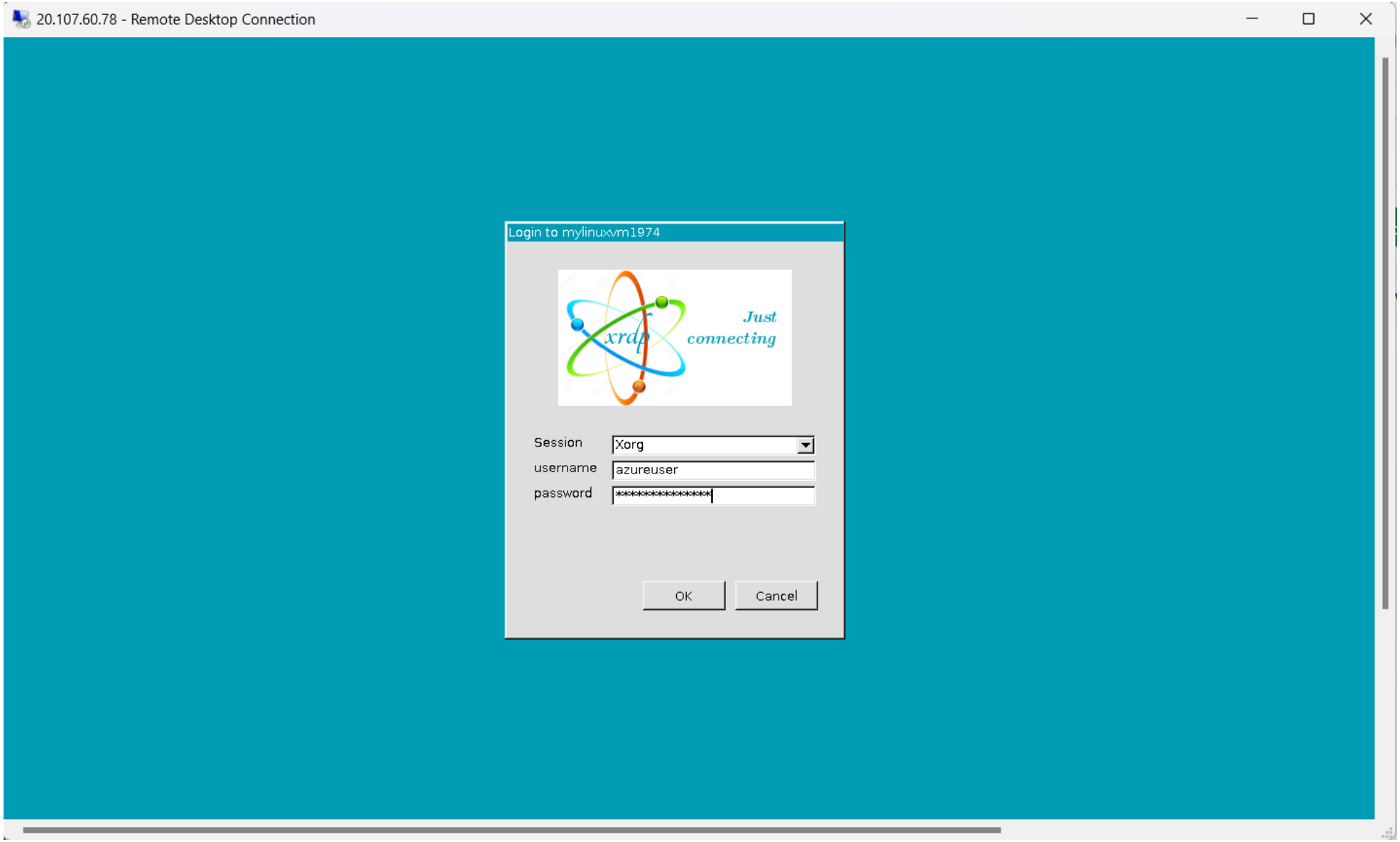
```
sudo passwd azureuser
```

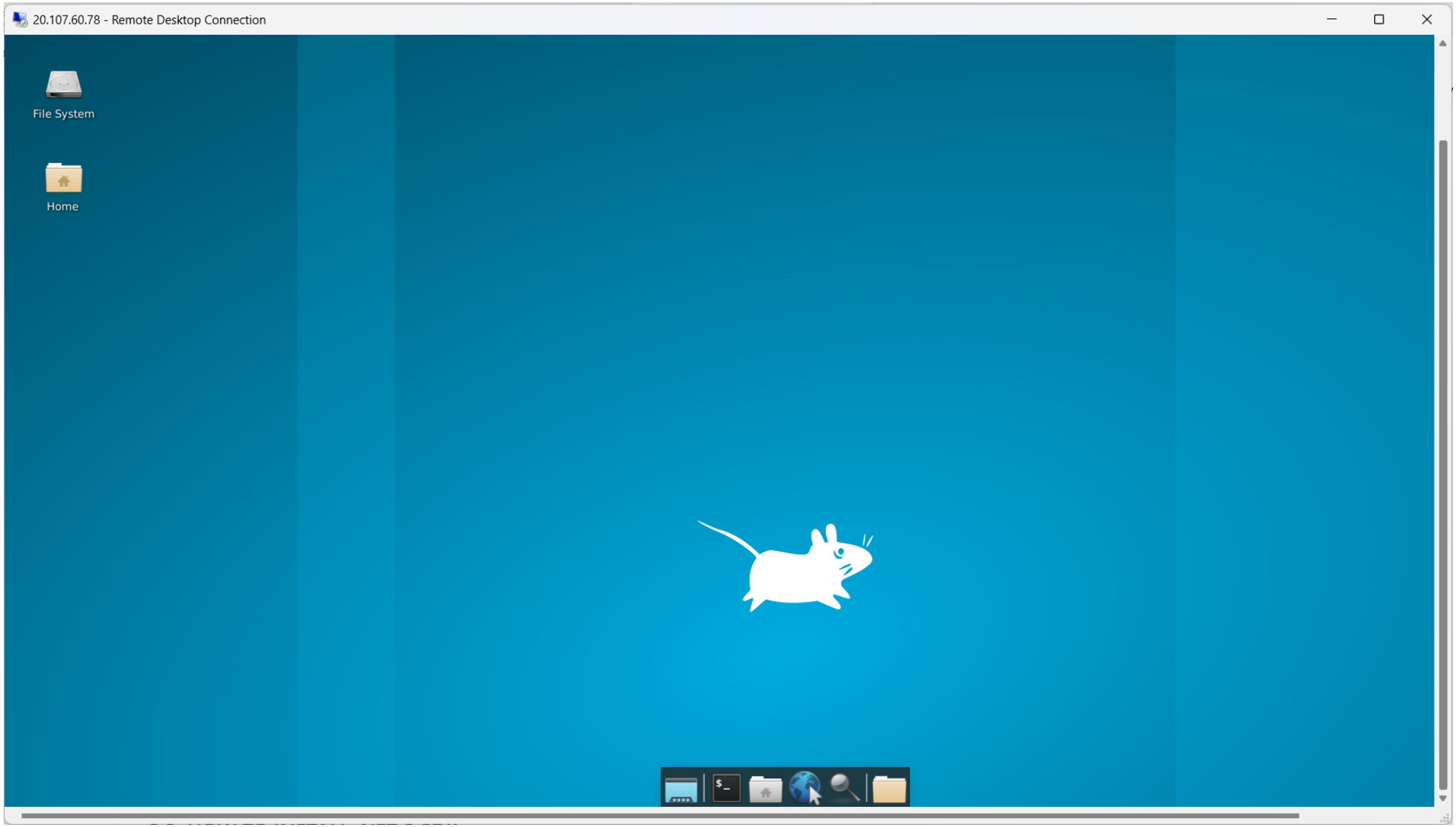
12.4. OPEN REMOTE DESKTOP CONNECTION

Now Open "**Remote Desktop Connection**" application and type the Azure VM **Public IP address** and the username "**azureuser**" and then press connect:



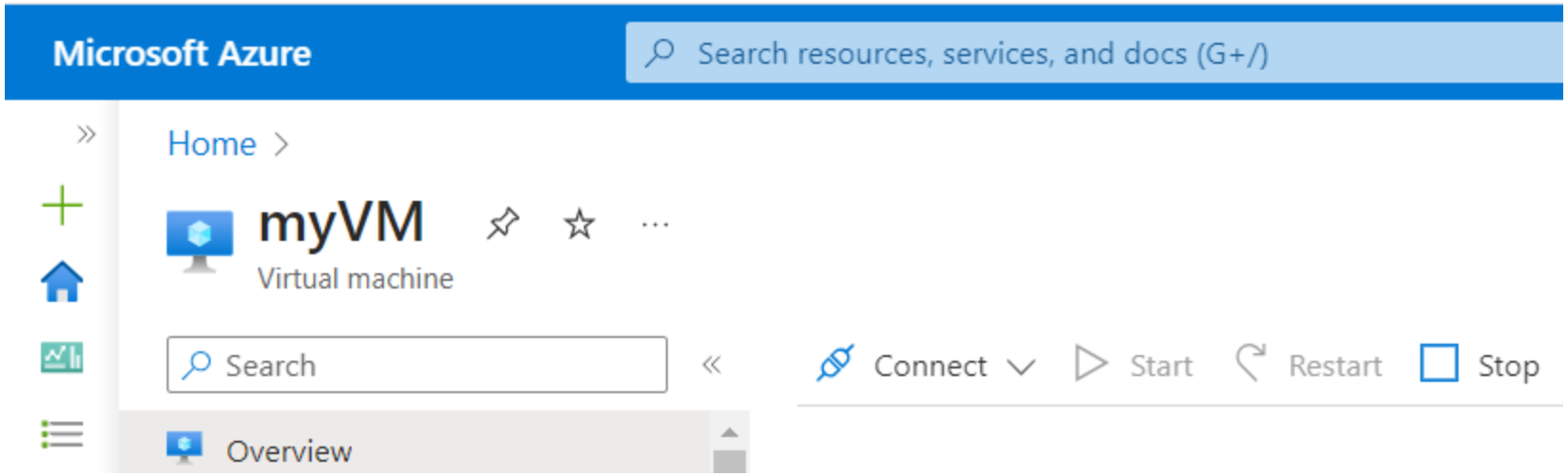
Then enter the password "**Thismypassword123456**" to access the Linux GUI Desktop, it also requires another password, set the same one as before "**Thismypassword123456**":


20.107.60.78 - Remote Desktop Connection
Login to mylinuxvm1974
xrdp
Just connecting
Session
Xorg
username
azureuser
password
OK
Cancel

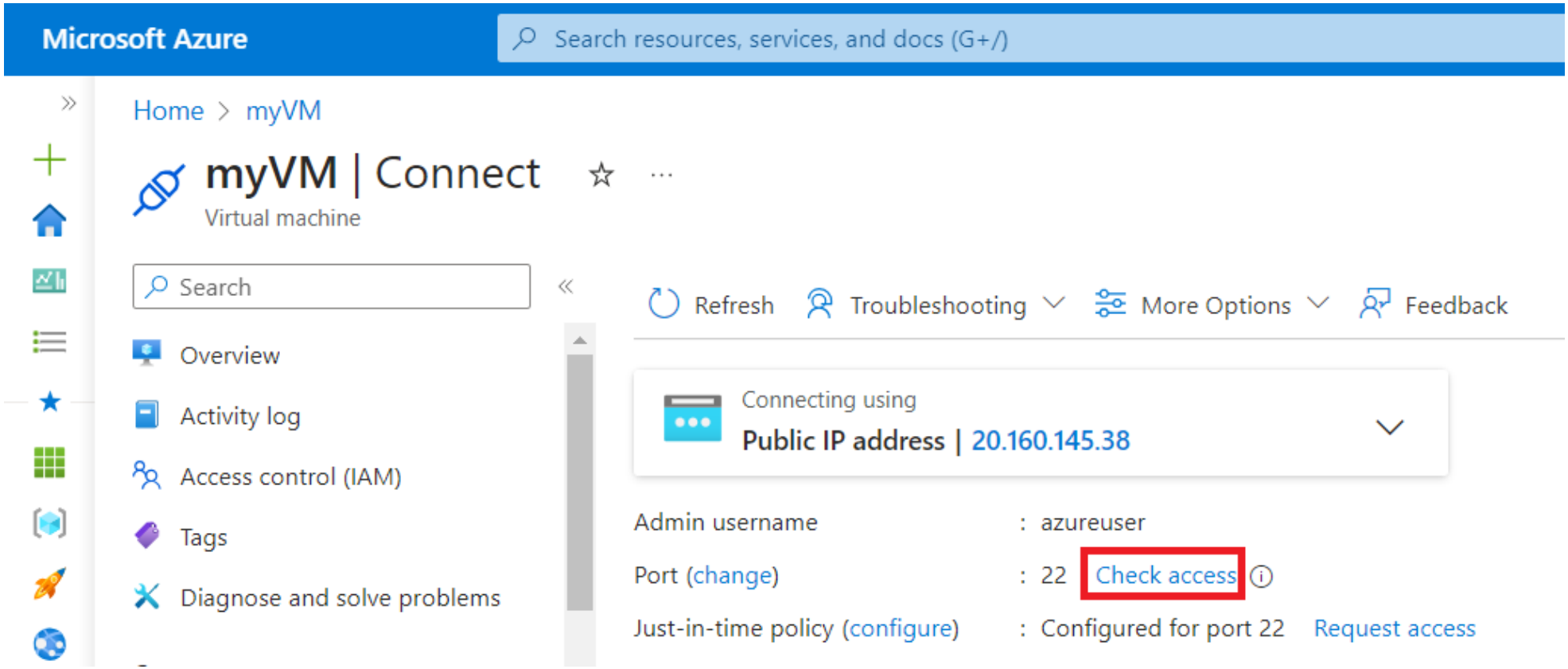


IMPORTANT NOTE: if you cannot access:

Restart the VM



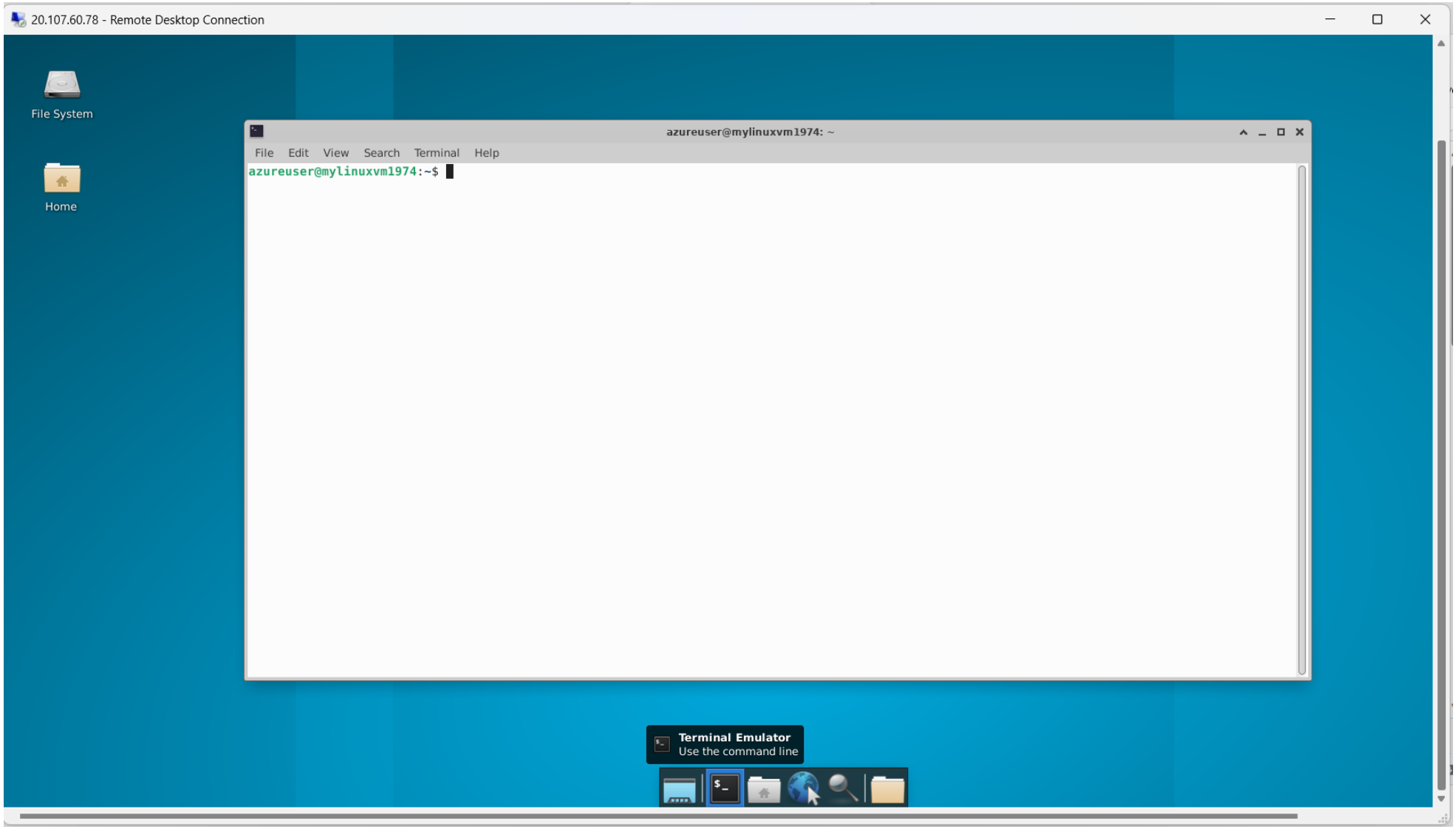
And Check the access,



After try/run again the **Remote Desktop Connection**" application.

12.5. HOW TO INSTALL VSCODE

Open a Terminal Emulator window and run the following commands to install the VSCode application



```
sudo apt install software-properties-common apt-transport-https wget
wget -q https://packages.microsoft.com/keys/microsoft.asc -O- | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://packages.microsoft.com/repos/vscode stable main"
sudo apt update
sudo apt install code
```

For accessing VSCode type the command:

```
code
```

12.6. HOW TO INSTALL GOOGLE CHROME

```
wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
sudo dpkg -i google-chrome-stable_current_amd64.deb
```

12.7. HOW TO INSTALL .NET 8 SDK

```
wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb

sudo apt update
sudo apt install -y apt-transport-https
sudo apt update
sudo apt install -y dotnet-sdk-8.0

dotnet --version
```