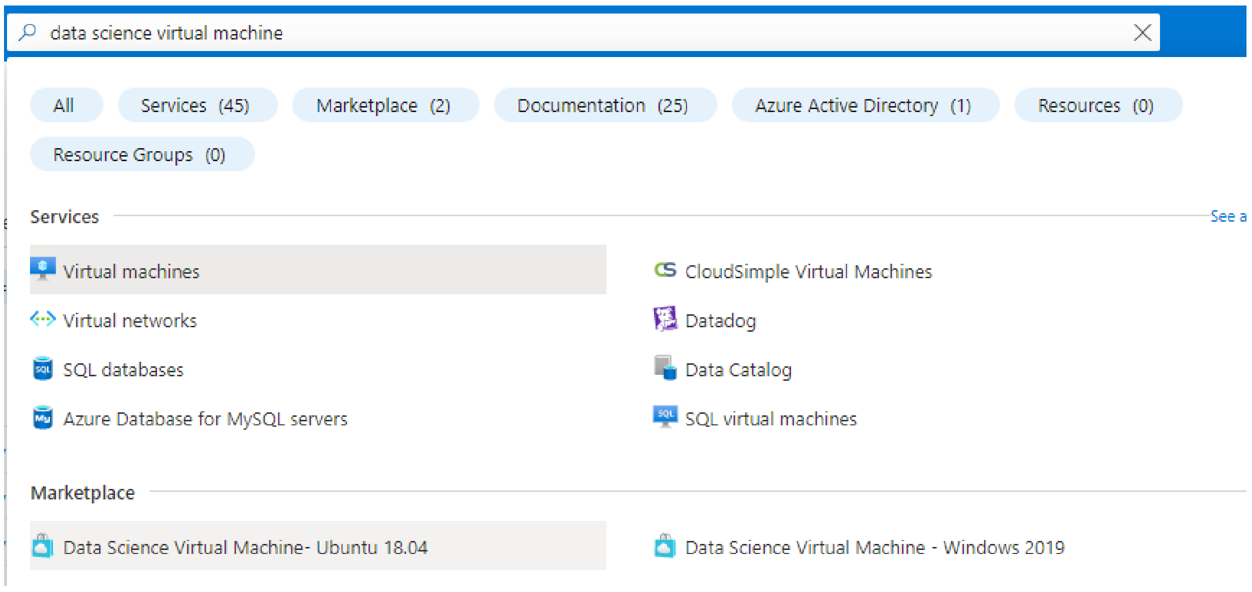
CIDA Hackathon 2022

Quickstart

1. Set up the Data Science Virtual Machine for Windows or Linux

Your first step is to set a Data Science VM on Azure.
Prerequisite
To create a Windows Data Science Virtual Machine, you must have an Azure subscription, which is provided for the Hackathon. Please note Azure free accounts do not support GPU enabled virtual machine SKUs.
Create your DSVM
To create a DSVM instance:

1. Go to the Azure portal You might be prompted to sign in to your Azure account if you're not already signed in.
2. Find the virtual machine listing by typing in "data science virtual machine" and selecting "Data Science Virtual Machine - Windows 2019" or "Data Science Virtual Machine – Ubuntu 18.04" in the Marketplace:


data science virtual machine
All
Services (45)
Marketplace (2)
Documentation (25)
Azure Active Directory (1)
Resources (0)
Resource Groups (0)
Services
See a
Virtual machines
CloudSimple Virtual Machines
Virtual networks
Datadog
SQL databases
Data Catalog
Azure Database for MySQL servers
SQL virtual machines
Marketplace
Data Science Virtual Machine- Ubuntu 18.04
Data Science Virtual Machine - Windows 2019

3. Select the **Create** button at the bottom.
4. You should be redirected to the "Create a virtual machine" blade.
5. Fill in the **Basics** tab:

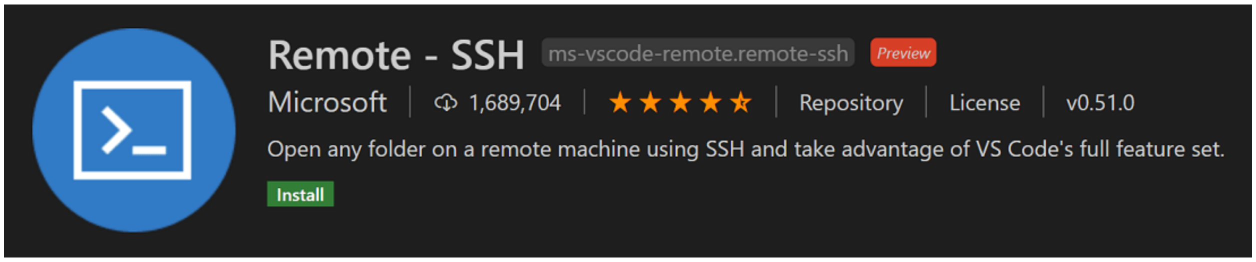
- **Subscription**:
- **Resource group**: Create a new group or use an existing one.
- **Virtual machine name**: Enter the name of the virtual machine. This is how it will appear in your Azure portal.
- **Location**: Select the datacenter that's most appropriate. Is this case: East.
- **Image**: Leave the default value.
- **Size**: This should auto-populate with a size that is appropriate for general workloads. Read more about Windows VM sizes in Azure.
- **Username**: Enter the administrator username. This is the username you will use to log into your virtual machine and need not be the same as your Azure username.
- **Password**: Enter the password you will use to log into your virtual machine.

6. Select **Review + create**.
7. **Review+create**
    - Verify that all the information you entered is correct.
    - Select **Create**.

2. How to access the VM using VS Code

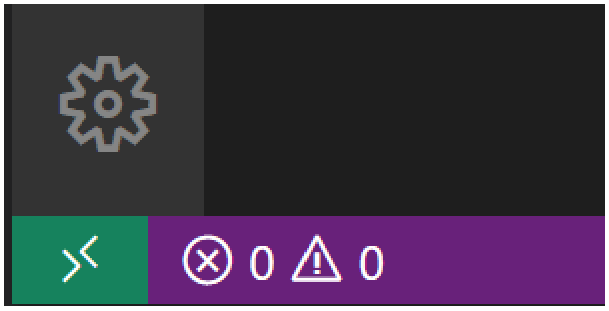
To get started, you need to:

1. Install an OpenSSH compatible SSH client (PuTTY is not supported). You can install the extension the Remote - SSH extension, which is used to connect to SSH hosts.

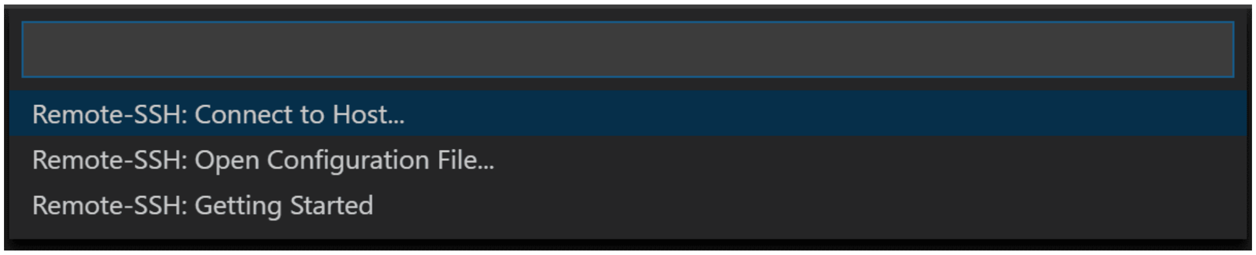


Remote - SSH ms-vscode-remote.remote-ssh Preview
Microsoft | 1,689,704 | ★★★★★ | Repository | License | v0.51.0
Open any folder on a remote machine using SSH and take advantage of VS Code's full feature set.
Install

2. With the Remote - SSH extension installed, you will see a new Status bar item at the far left.



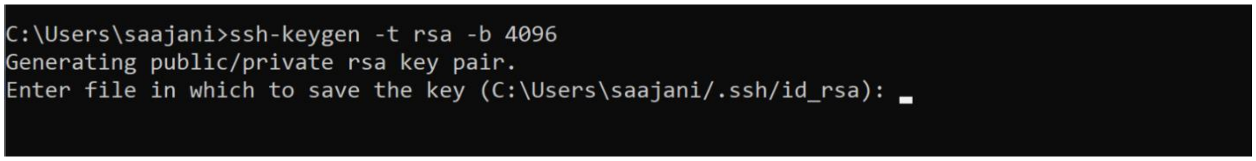
The Remote Status bar item can quickly show you in which context VS Code is running (local or remote) and clicking on the item will bring up the Remote - SSH commands. You will need an SSH key to access your VM. You can create a key (item 4) or do that within Azure (item 5).


Remote-SSH: Connect to Host...
Remote-SSH: Open Configuration File...
Remote-SSH: Getting Started

3. Set up SSH. There are several authentication methods into a VM, including an SSH public/private key pair or a username and password. We recommend using key-based authentication (if you use a username/password, you'll be prompted to enter your credentials more than once by the extension). If you're on Windows and have already created keys using PuttyGen, you can reuse them.
4. Create an SSH key. If you don't have an SSH key pair, open a bash shell or the command line and type in:

```
ssh-keygen -t rsa -b 2048
```

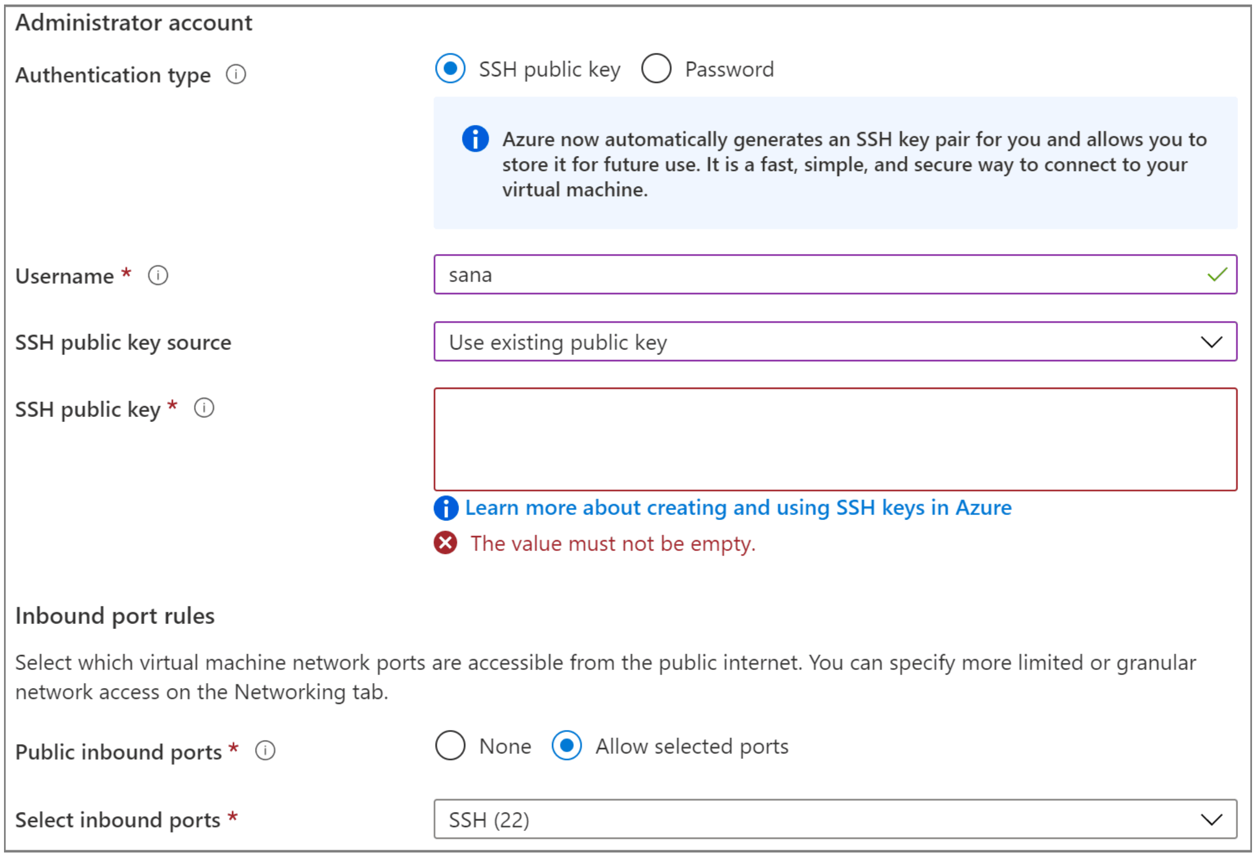
This will generate the SSH key. Press Enter at the following prompt to save the key in the default location (under your user directory as a folder named `.ssh`).



```
C:\Users\saajani>ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\saajani/.ssh/id_rsa):
```

You will then be prompted to enter a secure passphrase, but you can leave that blank. You should now have a `id_rsa.pub` file which contains your new public SSH key. When creating your VM you can also allow SSH access and create a key pair. Remember to download and store that.

5. Add SSH key to your VM. In the previous step, you generated an SSH key pair. Select **Use existing public key** in the dropdown for **SSH public key source** so that you can use the public key you just generated. Take the public key and paste it into your VM setup, by copying the entire contents of the `id_rsa.pub` in the **SSH public key**. You also want to allow your VM to accept inbound SSH traffic by selecting **Allow selected ports** and choosing **SSH (22)** from the **Select inbound ports** dropdown list. If you want to create a key pair, this is where you need to select "Create new key" under "SSH public key source".



Administrator account

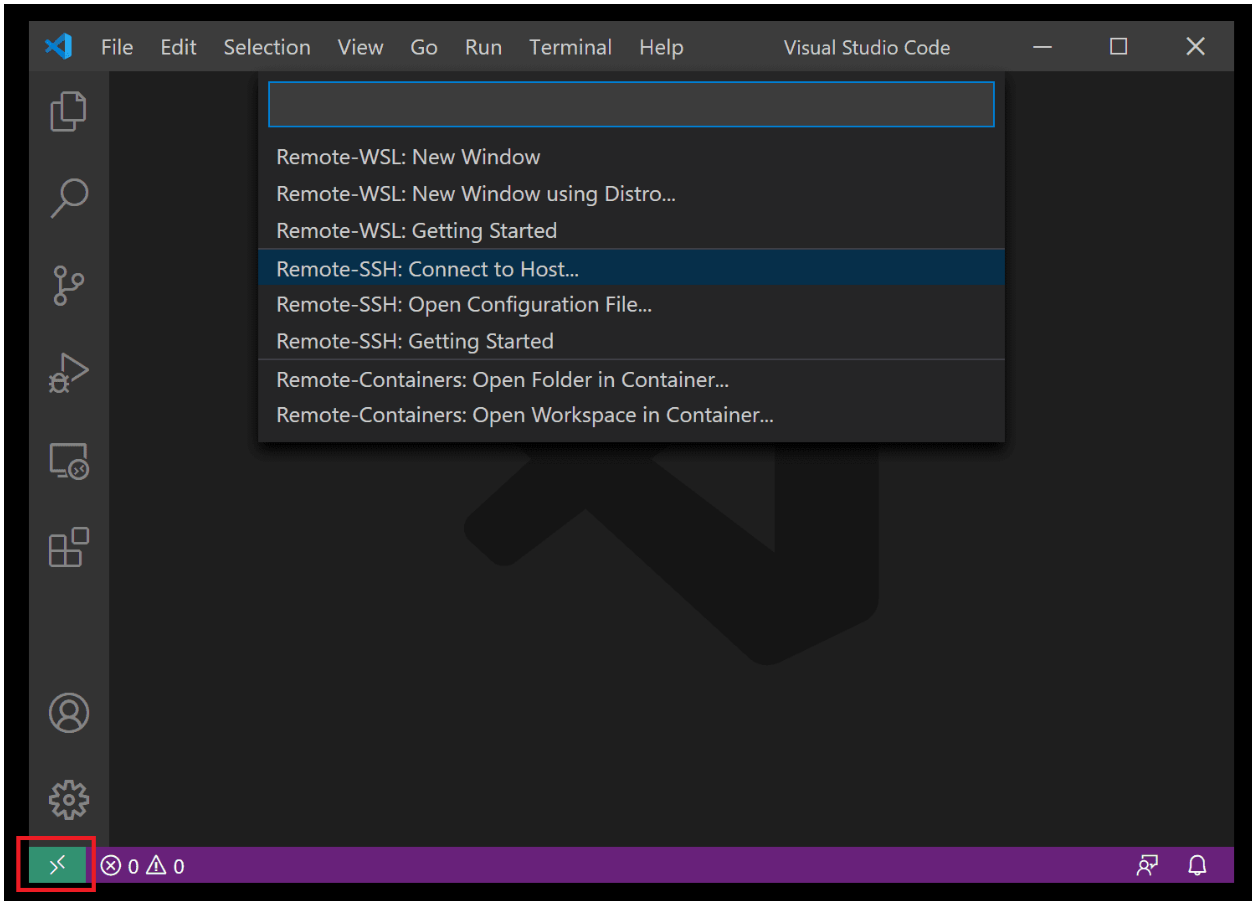
Authentication type	[✓ SSH public key] Password
	Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.
Username *	sana
SSH public key source	Use existing public key
SSH public key *	
	Learn more about creating and using SSH keys in Azure
	The value must not be empty.

Inbound port rules

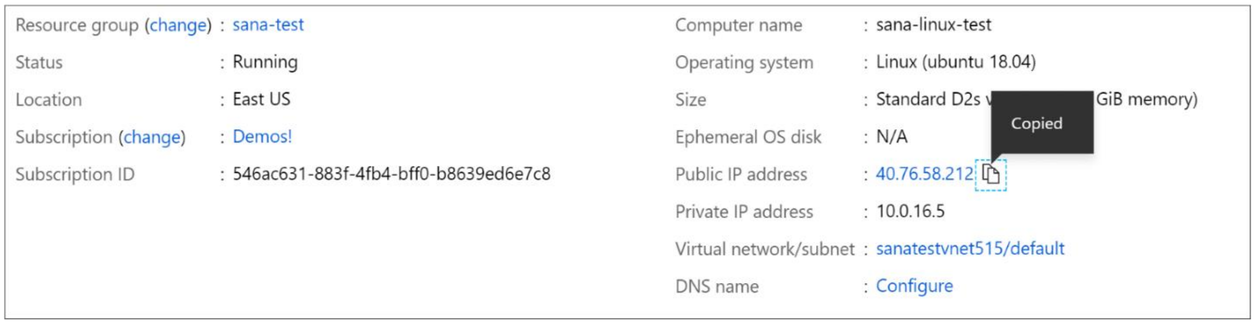
Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports *	None [✓ Allow selected ports]
Select inbound ports *	SSH (22)

6. Connect using SSH. Now that you've created an SSH host, let's connect to it! You'll have noticed an indicator on the bottom-left corner of the Status bar. This indicator tells you in which context VS Code is running (local or remote). Click on the indicator to bring up a list of Remote extension commands.


File
Edit
Selection
View
Go
Run
Terminal
Help
Visual Studio Code
Remote-WSL: New Window
Remote-WSL: New Window using Distro...
Remote-WSL: Getting Started
Remote-SSH: Connect to Host...
Remote-SSH: Open Configuration File...
Remote-SSH: Getting Started
Remote-Containers: Open Folder in Container...
Remote-Containers: Open Workspace in Container...
0
0

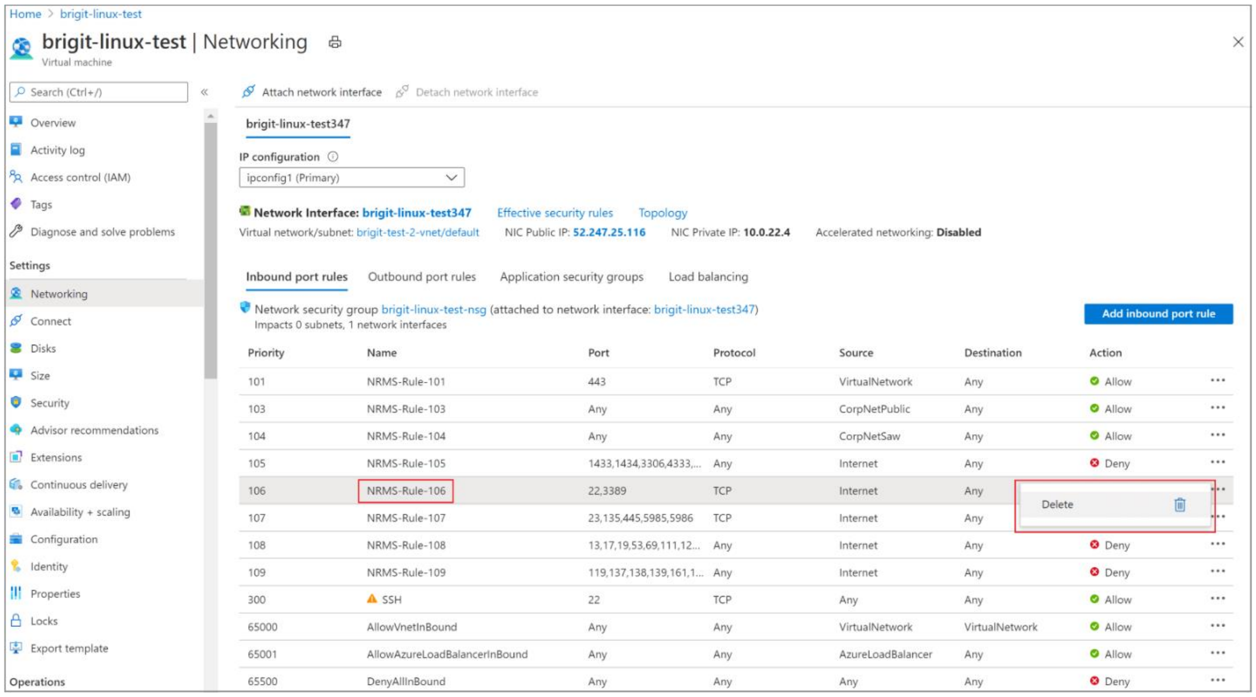
Choose the **Remote-SSH: Connect to Host** command and connect to the host by entering connection information for your VM in the following format: `user@hostname`. The `user` is the username you set when adding the SSH public key to your VM. For the `hostname`, go back to the Azure portal and in the **Overview** pane of the VM you created, copy the **Public IP address**.



Resource group (change)	: sana-test	Computer name	: sana-linux-test
Status	: Running	Operating system	: Linux (ubuntu 18.04)
Location	: East US	Size	: Standard D2s v GiB memory)
Subscription (change)	: Demos!	Ephemeral OS disk	: N/A
Subscription ID	: 546ac631-883f-4fb4-bff0-b8639ed6e7c8	Public IP address	: 40.76.58.212
		Private IP address	: 10.0.16.5
		Virtual network/subnet	: sanatestvnet515/default
		DNS name	: Configure

Before connecting in Remote - SSH, you can verify you're able to connect to your VM via a command prompt using `ssh user@hostname`.

Note: If you run into an error `ssh: connect to host <host ip> port 22: Connection timed out`, you may need to delete NRMS-Rule-106 from the Networking tab of your VM:

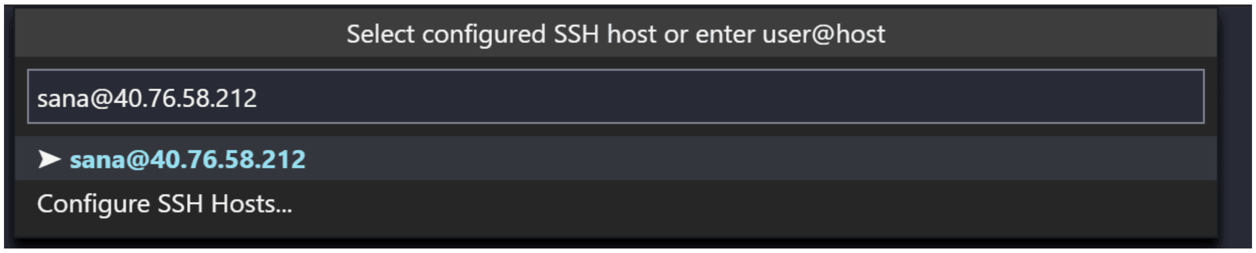


Network Interface: brigit-linux-test347
Virtual network/subnet: brigit-test-2-vnet/default NIC Public IP: 52.247.25.116 NIC Private IP: 10.0.22.4 Accelerated networking: Disabled

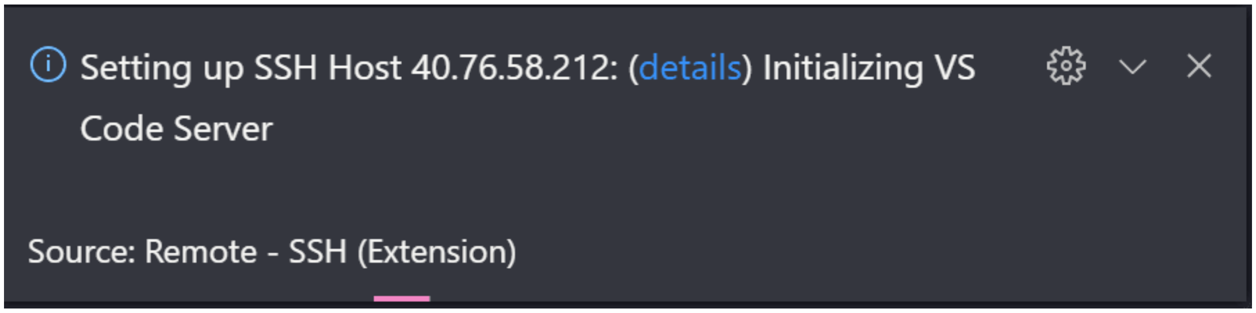
Network security group brigit-linux-test-nsg (attached to network interface: brigit-linux-test347)
Impacts 0 subnets, 1 network interfaces

Priority	Name	Port	Protocol	Source	Destination	Action
101	NRMS-Rule-101	443	TCP	VirtualNetwork	Any	Allow
103	NRMS-Rule-103	Any	Any	CorpNetPublic	Any	Allow
104	NRMS-Rule-104	Any	Any	CorpNetSaw	Any	Allow
105	NRMS-Rule-105	1433,1434,3306,4333,...	Any	Internet	Any	Deny
106	NRMS-Rule-106	22,3389	TCP	Internet	Any	
107	NRMS-Rule-107	23,135,445,5985,5986	TCP	Internet	Any	
108	NRMS-Rule-108	13,17,19,53,69,111,12...	Any	Internet	Any	Deny
109	NRMS-Rule-109	119,137,138,139,161,1...	Any	Internet	Any	Deny
300	SSH	22	TCP	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

7. Set the user and hostname in the connection information text box.

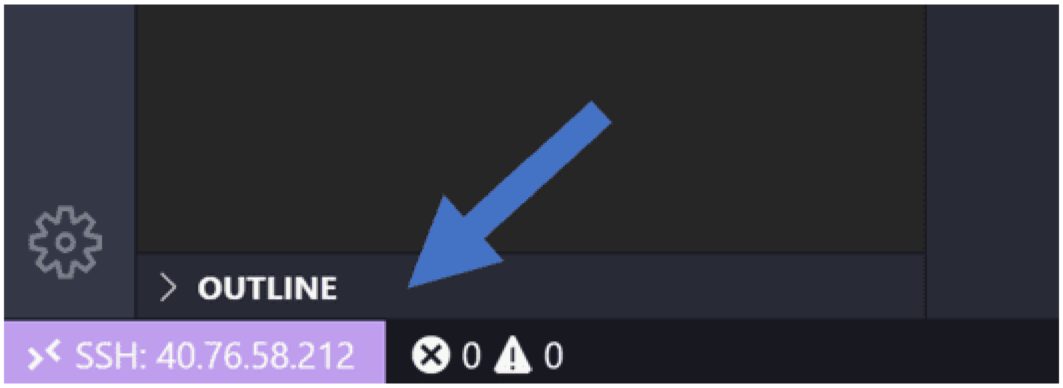


VS Code will now open a new window (instance). You'll then see a notification that the "VS Code Server" is initializing on the SSH Host. Once the VS Code Server is installed on the remote host, it can run extensions and talk to your local instance of VS Code.

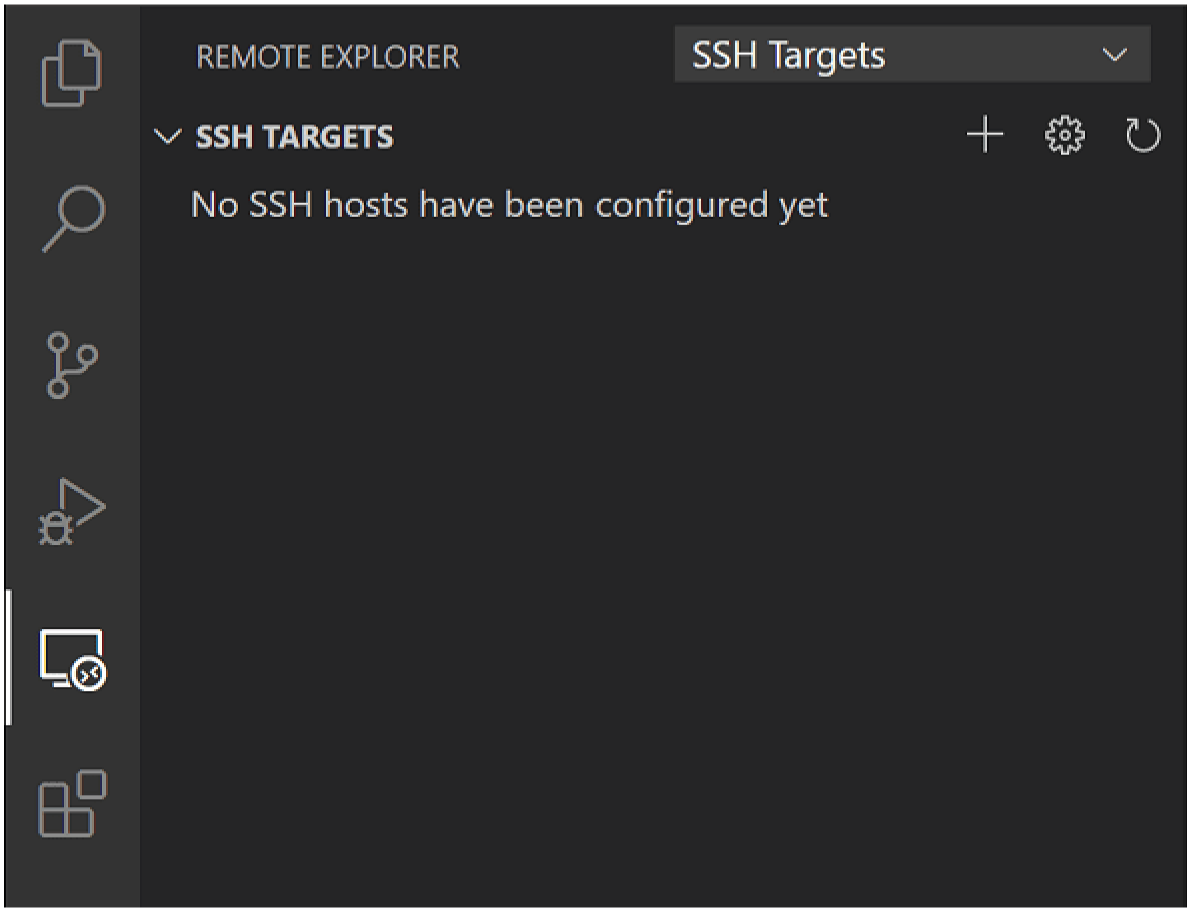

Setting up SSH Host 40.76.58.212: (details) Initializing VS Code Server

Source: Remote - SSH (Extension)

You'll know you're connected to your VM by looking at the indicator in the Status bar. It shows the hostname of your VM.

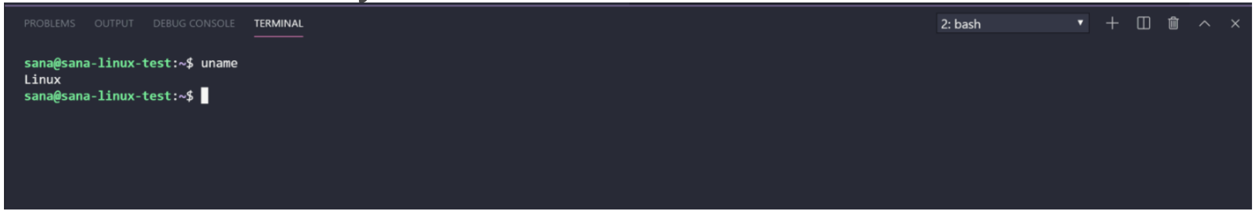


The Remote - SSH extension also contributes a new icon on your Activity bar, and clicking on it will open the Remote explorer. From the dropdown, select **SSH Targets**, where you can configure your SSH connections. For instance, you can save the hosts you connect to the most and access them from here instead of entering the user and hostname.



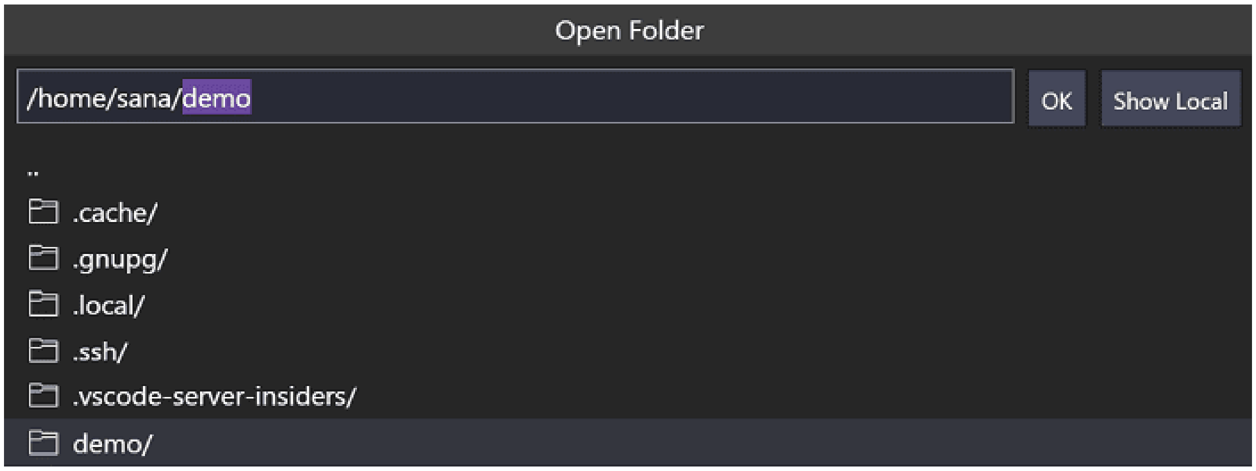
No SSH hosts have been configured yet

Once you're connected to your SSH host, you can interact with files and open folders on the remote machine. If you open the integrated terminal (Ctrl+`), you'll see you're working inside a bash shell **while you're on Windows**.



```
sana@sana-linux-test:~$ uname
Linux
sana@sana-linux-test:~$
```

You can use the bash shell to browse the file system on the VM. You can also browse and open folders on the remote home directory with **File** > **Open Folder**.



If you have any questions, you can check these references: Use SSH keys to connect to Linux VMs - Azure Virtual Machines | Microsoft Docs and Connect over SSH with Visual Studio Code

Download data from blob storage to your account

1. To download Space Eye data you will need to access the spaceye container in the blob storage. You can use azcopy to do that.
2. To install AzCopy on Windows, you can run the following PowerShell script, or you can download the zip file and run it from wherever you want. This script will add the AzCopy folder location to your system path so that you can run the AzCopy command from anywhere.

```
#Download AzCopy
Invoke-WebRequest -Uri "https://aka.ms/downloadazcopy-v10-windows" -OutFile AzCopy.zip -
UseBasicParsing
#Curl.exe option (Windows 10 Spring 2018 Update (or later))
curl.exe -L -o AzCopy.zip https://aka.ms/downloadazcopy-v10-windows
#Expand Archive
```

```
Expand-Archive ./AzCopy.zip ./AzCopy -Force
#Move AzCopy to the destination you want to store it
Get-ChildItem ./AzCopy/*/azcopy.exe | Move-Item -Destination
"C:\Users\thmaure\AzCopy\AzCopy.exe"
#Add your AzCopy path to the Windows environment PATH (C:\Users\thmaure\AzCopy in this
example), e.g., using PowerShell:
$userenv = [System.Environment]::GetEnvironmentVariable("Path", "User")
[System.Environment]::SetEnvironmentVariable("PATH", $userenv + ";C:\Users\thmaure\AzCopy",
"User")
```

If you are using Linux, you can install that by running:

```
#Download AzCopy
wget https://aka.ms/downloadazcopy-v10-linux
#Expand Archive
tar -xvf downloadazcopy-v10-linux
#(Optional) Remove existing AzCopy version
sudo rm /usr/bin/azcopy
#Move AzCopy to the destination you want to store it
sudo cp ./azcopy_linux_amd64_*/azcopy /usr/bin/

It is important to notice that if you cannot
```

3. After doing that you will need the Blob SAS URL. To download the pre-process NDVI data, you need to run to download the ndvi folder into your VM. If you cannot run the following command, try to add sudo before `azcopy` or run `sudo chmod 755 /usr/bin/azcopy`:

```
azcopy copy "https://digitalag22.blob.core.windows.net/ndvi?<SAS-Token>" . --recursive
```

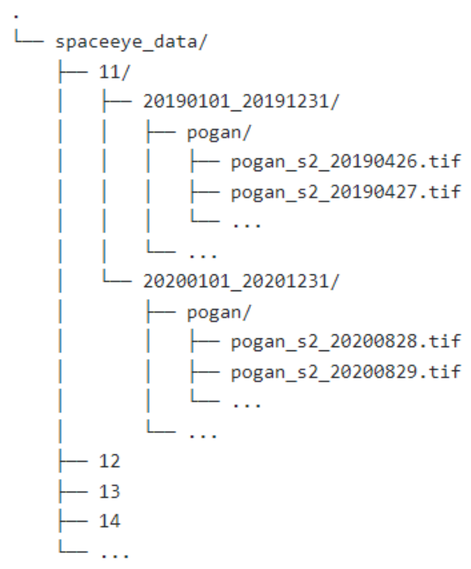
Each .tiff file has around 16 GB. At a connection speed of 100 Mbps, you will take around 1 minute to download both.

4. To download full original Space Eye data, you need to run:

```
azcopy copy "https://digitalag22.blob.core.windows.net/spaceeye-data?<SAS-Token>" . --recursive
```

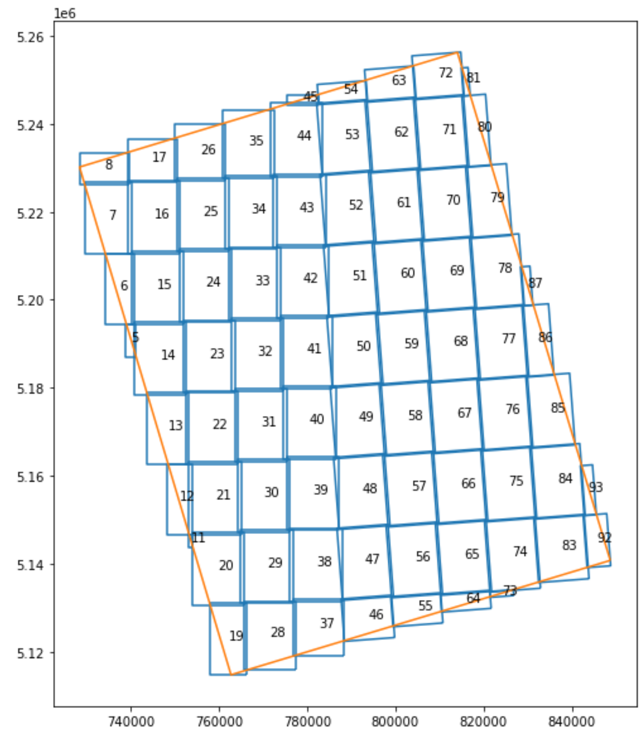
Each .tiff file has around 2.9 TB. At a connection speed of 100 Mbps, you will take around 2:30 minutes to download. At a connection speed of 10 Mbps, you will need around 30 minutes.

In this case, you will download a folder `spaceeye_data.` Inside the folder you will find several directories representing parts of the full region. Inside each numbered directory, you will find two folders representing data from the years 2019 and 2020. Inside each folder, there are files for this region every day in this specific year:

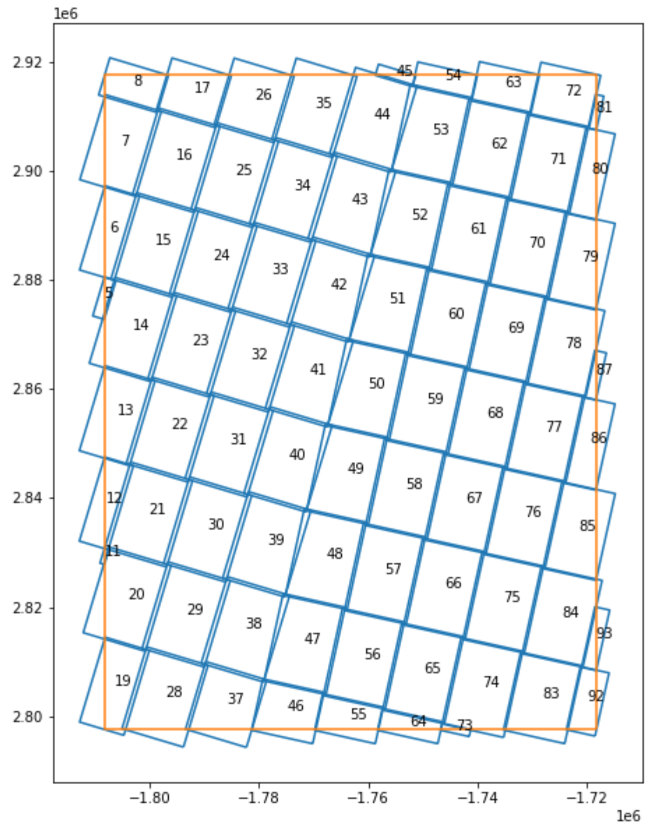


```
.
└── spaceeye_data/
    ├── 11/
    │   ├── 20190101_20191231/
    │   │   ├── pogan/
    │   │   │   ├── pogan_s2_20190426.tif
    │   │   │   ├── pogan_s2_20190427.tif
    │   │   │   └── ...
    │   │   └── ...
    │   └── 20200101_20201231/
    │       ├── pogan/
    │       │   ├── pogan_s2_20200828.tif
    │       │   ├── pogan_s2_20200829.tif
    │       │   └── ...
    │       └── ...
    ├── 12
    ├── 13
    ├── 14
    └── ...
```

Bellow you will find the numbering of the grids in UTM (be aware that different grids may have different UTM):

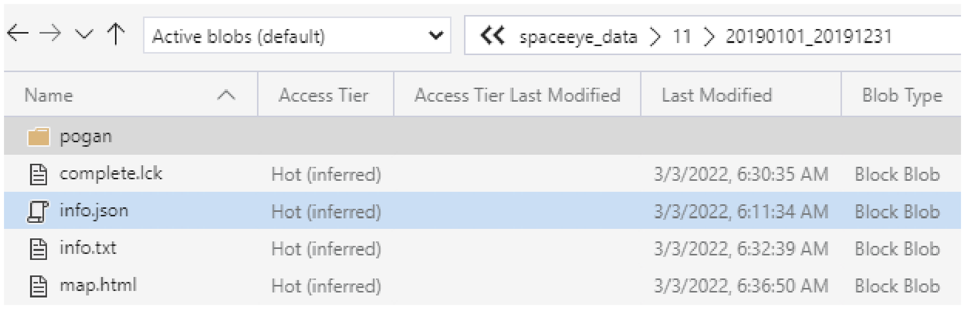


Or the grids in EPSG:5070, the same CRS used in CDL (and NDVI):



You will find the geojson file containing that in the GitHub repository under the name grid_epsg_5070.geojson and grid_epsg_4326.geojson.

Inside each directory, these are the kind of files you will find:



Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type
pogan				
complete.lck	Hot (inferred)		3/3/2022, 6:30:35 AM	Block Blob
info.json	Hot (inferred)		3/3/2022, 6:11:34 AM	Block Blob
info.txt	Hot (inferred)		3/3/2022, 6:32:39 AM	Block Blob
map.html	Hot (inferred)		3/3/2022, 6:36:50 AM	Block Blob

info.json and info.txt – It contains the information of geometry used to predict the tiff file.

map.**html** – You can view the predicted tiff file on the geospatial base map.

complete.lck – Ignore, internal processing file.