



Azure Section 11 - Weather app Connecting it to Readit app.

1. There we can see a link in the read it app. You can see the link in the top right corner and that link is of the weather app. So we added this extra link just in case the customer wants to know the weather condition before buying the book.
2. Now for this we will create a virtual machine with standard SSD and with the iso image as Linux.
3. We will also download the putty app and then we will connect the machine using putty app.
4. Then going forward we will import the code from the git repo and then we will download nodejs.
5. Then we will run the app on this Linux server and once it is run successfully then we will again go to the catalog server and then check from there if we are able to communicate with the weather app or not.
6. Now if we go to the catalog/ readit app then we will see that there is weather link. Click on it and there will be a page which will appear saying that we need to enter the ip of the weather server.
7. So in this step we need to memorize the private IP of the server and then we need to enter it into the weather app followed by the port number 8080

Subscription *

Resource group * 1

Instance details

Virtual machine name *

Region *

Availability options

Availability zone *

You can now select multiple zones. Selecting multiple zones will create one VM per zone. [Learn more](#)

Creating the Vm with the Linux image for the weather app. We will create this VM under the same resource group.

Security type

Image * 2

VM architecture ☐ ☒

Run with Azure Spot discount ☐

You are in the free trial period. Costs associated with this VM can be covered by any remaining credits on your subscription. [Learn more](#)

Size *

Took the linux iso and the size should also be small we do not need some heavy server for this purpose.

Authentication type ☐ ☒ 3

Username *

Password *

Confirm password *

Now here we need to choose the password instead of SSH coz it is easy to remember. so give the user name and password.

Virtual network *

Subnet *

Public IP

NIC network security group ☐ ☒ 4

Public inbound ports * ☐ ☒

Select inbound ports *

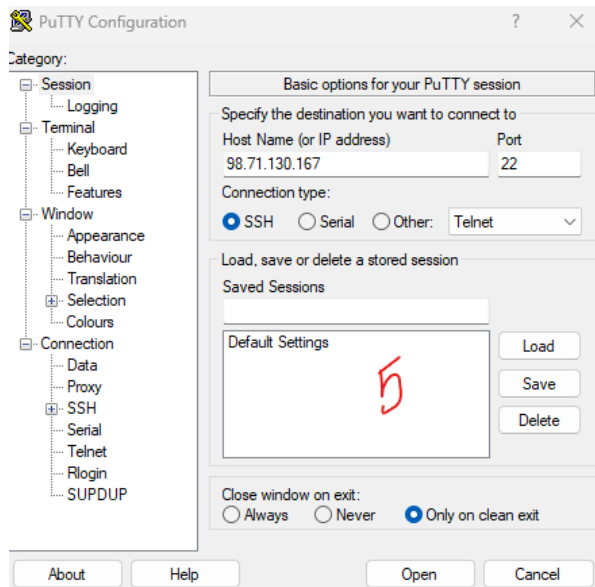
So here under the network settings we need to choose the virtual network which is same as that we created initially. Also under the public IP section choose the static IP option for the Virtual machine.

Under the management please choose the auto shutdown.

So to connect to the Linux machine we need a tool which is called as putty tool.

Download it and install it.

Copy the public IP of the VM of Linux and past it in the putty tool.



Here we will paste the public IP address of the VM of Linux. And hit open then we need to enter the user name and password.

So we will install git on the server to pull the code of the app from the git repository.

```
Abhinav@weather-app-vm:~$ sudo apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.25.1-lubuntu3.11).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Abhinav@weather-app-vm:~$ git --version
git version 2.25.1
Abhinav@weather-app-vm:~$
```

So we will install a random weather app on the server which will give a random forecast of the weather. the app is developed in node js. So we need to do the installation for the application.

First we will install the git package to pull the code from the git hub.

Next we will be installing the Node Js for that we need to use the following command.

`sudo apt install nodejs`

We will now pull the git code with following command.

`sudo git clone https://github.com/memilavi/weatherAPI.git` `sudo apt install npm`

```

Abhinav@weather-app-vm:~$ sudo git clone https://github.com/memilavi/weatherAPI.
git
Cloning into 'weatherAPI'...
remote: Enumerating objects: 406, done.
remote: Counting objects: 100% (406/406), done.
remote: Compressing objects: 100% (323/323), done.
remote: Total 406 (delta 74), reused 400 (delta 73), pack-reused 0
Receiving objects: 100% (406/406), 565.19 KiB | 17.66 MiB/s, done.
Resolving deltas: 100% (74/74), done.
Abhinav@weather-app-vm:~$ cd weatherAPI/
Abhinav@weather-app-vm:~/weatherAPI$ ls
index.js  node_modules  package-lock.json  package.json
Abhinav@weather-app-vm:~/weatherAPI$ 

```

There are some additional packages that we need to install so that the app will run properly.

sudo apt install npm

Use the above command to install the packages. And then run the package with the command **start npm**.

So we can see that npm is getting downloaded.

```

Preparing to unpack .../397-node-combined-stream_1.0.8-1_all.deb ...
Unpacking node-combined-stream (1.0.8-1) ...
Selecting previously unselected package node-debug.
Preparing to unpack .../398-node-debug_4.1.1-2_all.deb ...
Unpacking node-debug (4.1.1-2) ...
Selecting previously unselected package node-encoding.
Preparing to unpack .../399-node-encoding_0.1.12-3_all.deb ...
Unpacking node-encoding (0.1.12-3) ...
Selecting previously unselected package node-prr.
Preparing to unpack .../400-node-prr_1.0.1-1_all.deb ...
Unpacking node-prr (1.0.1-1) ...
Selecting previously unselected package node-errno.
Preparing to unpack .../401-node-errno_0.1.7-1_all.deb ...
Unpacking node-errno (0.1.7-1) ...
Progress: [ 47%] [#####.....]

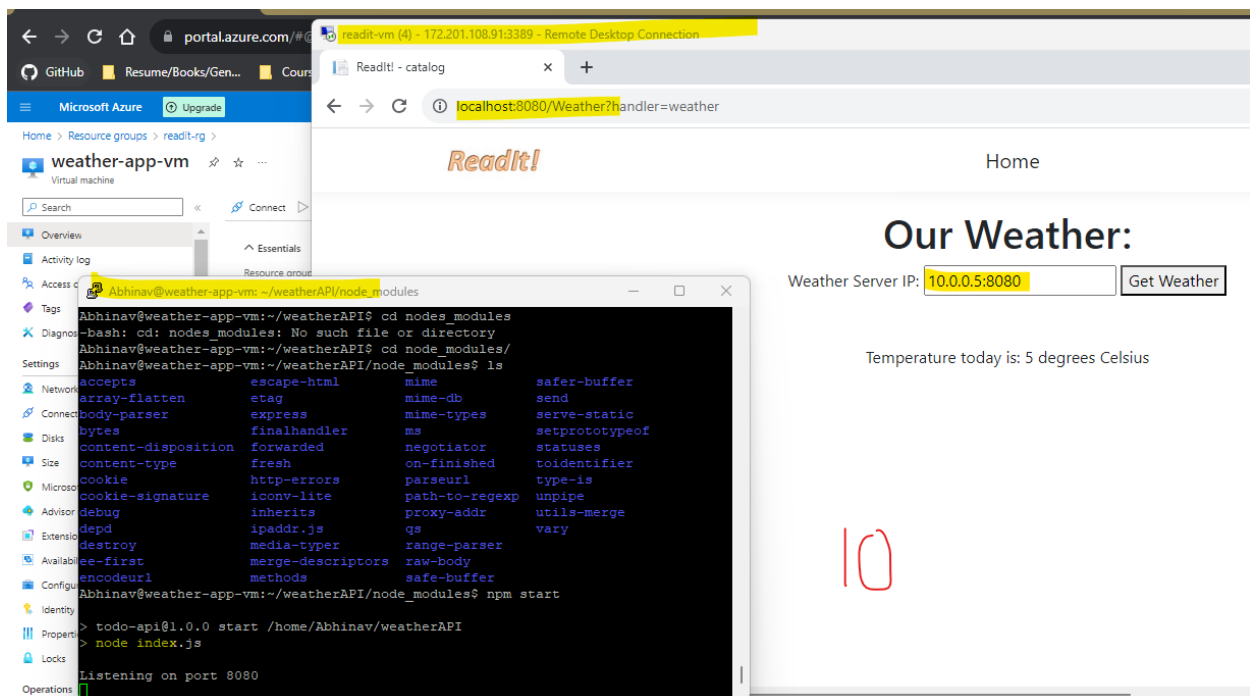
```

Now once this is completed we can start the npm

```
Abhinav@weather-app-vm:~/weatherAPI/node_modules$ npm start
> todo-api@1.0.0 start /home/Abhinav/weatherAPI
> node index.js
Listening on port 8080
```

So the NPM has now started. Listening on port 8080.

Now we need to go to the catalog app and enter the private IP address of the Linux machine in the catalog app server under the weather page.



Here we have to enter the private IP address followed by the port number because the machine is listening at the port 8080.

Here the hands on of the weather app connectivity is completed.