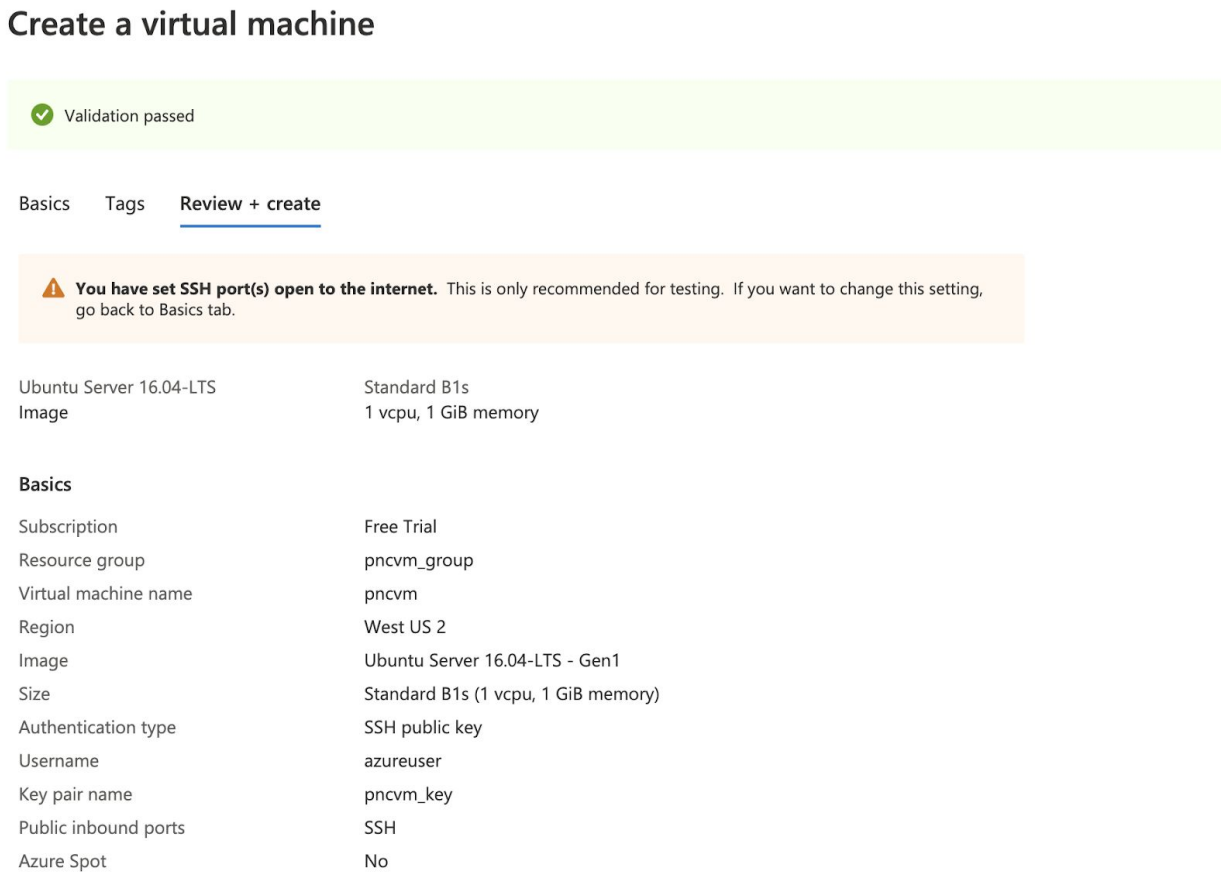
Creating a VM using Azure

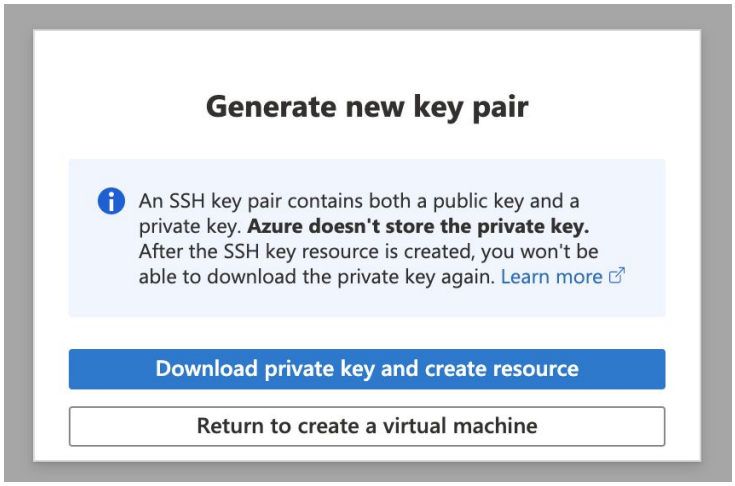
Creating the VM manually on Azure

After creating an account on Azure with my personal email (while waiting for my PNC credentials to get processed), I used the free services page on azure portal (https://go.microsoft.com/fwlink/?linkid=859151) to create a Linux Virtual Machine.

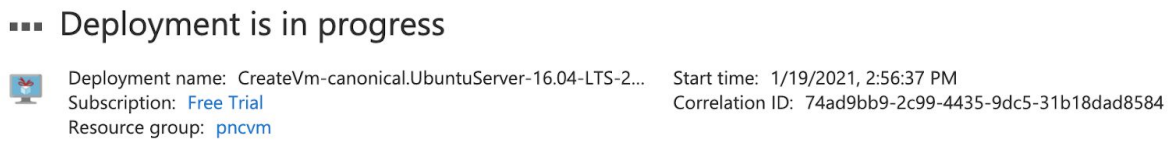
The details of the machine I created are shown below:



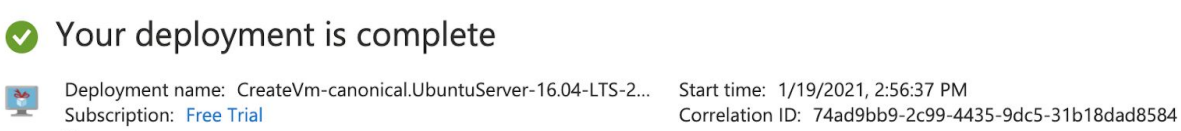
After selecting “create”, I was prompted to generate and save the key-value pair to my computer

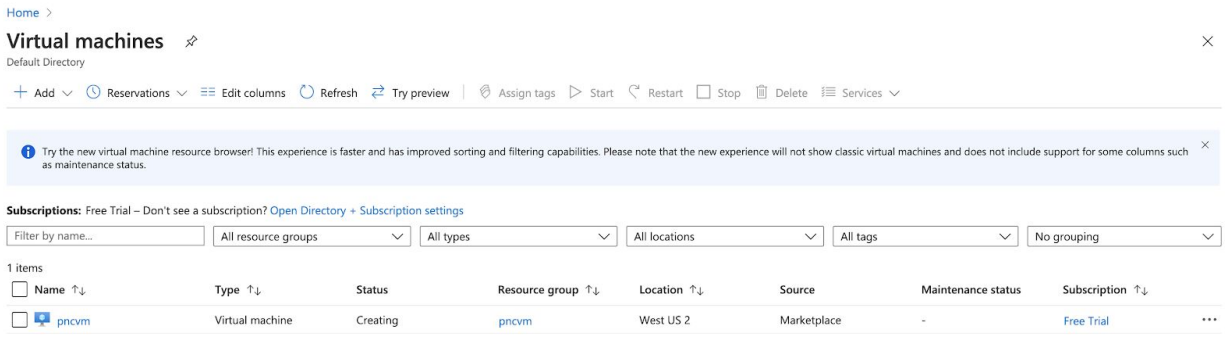


After this the VM began to deploy.



And as soon as it was complete, I could see it in my dashboard of VMs





Connecting to the Virtual Machine from my computer

Initially I tried to use the instructions found here https://docs.microsoft.com/en-us/azure/virtual-machines/ssh-keys-portal to use my SSH key to connect to my VM using 'ssh -i <path-to-key-file> <username>@<VM's-IP-address>.

I got this error:

```
priyakanipakam@Priyas-MBP-2 virtualmachine % ssh -i pncvm_key_0119.pem azureuser@52.183.85.79
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'pncvm_key_0119.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "pncvm_key_0119.pem": bad permissions
azureuser@52.183.85.79: Permission denied (publickey).
```

This was because you need to switch the file with the key to a read only file for security reasons. I did this using the command "chmod 400 <file-name>.pem. After this, I was able to connect to my VM.

```
priyakanipakam@Priyas-MBP-2 virtualmachine % ssh -i pncvm_key_0119.pem azureuser@52.183.85.79
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.15.0-1103-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 of these updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@pncvm:~$
```

Creating a VM using terraform

In order to explore how we could use automation to create VMs in azure, I will create a VM using terraform.

About terraform

- It is used for provisioning and managing cloud infrastructure. It codifies infrastructure in configuration files that describe the topology of cloud resources.

- Terraform CLI provides a simple mechanism to deploy and version the configuration files to Azure.

Instructions:
https://docs.microsoft.com/en-us/azure/developer/terraform/create-linux-virtual-machine-with-infrastructure

First I installed terraform into my system (MacOS)

1. `brew tap hashicorp/tap`
2. `brew install hashicorp/tap/terraform`

To verify installation, run "terraform -help"

```
priyakanipakam@Priyas-MBP-2 azure-docs % terraform -help
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console       Try Terraform expressions at an interactive command prompt
  fmt           Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get           Install or upgrade remote Terraform modules
  graph         Generate a Graphviz graph of the steps in an operation
  import        Associate existing infrastructure with a Terraform resource
  login         Obtain and save credentials for a remote host
  logout        Remove locally-stored credentials for a remote host
  output        Show output values from your root module
  providers     Show the providers required for this configuration
  refresh       Update the state to match remote systems
  show          Show the current state or a saved plan
  state         Advanced state management
  taint         Mark a resource instance as not fully functional
  untaint       Remove the 'tainted' state from a resource instance
  version       Show the current Terraform version
  workspace     Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR    Switch to a different working directory before executing the
                given subcommand.
  -help         Show this help output, or the help for a specified subcommand.
  -version      An alias for the "version" subcommand.
```

Next, I installed azure cli into my system

1. `brew update && brew install azure-cli`
2. To set up account - `az login`

```
priyakanipakam@Priyas-MBP-2 azure-docs % az login
The default web browser has been opened at https://login.microsoftonline.com/common/oauth2/authorize. Please continue the login in the web browser. If no web browser is available or if the web browser fails to open, use device code flow with `az login --use-device-code`.
You have logged in. Now let us find all the subscriptions to which you have access...
The following tenants don't contain accessible subscriptions. Use 'az login --allow-no-subscriptions' to have tenant level access.
c10cd415-1635-4d79-9e32-262b326d1f28 'my team'
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "273e4da0-c196-4c65-b234-56a0dc18c5a5",
    "id": "3fa40d5c-dbaa-49f0-8620-d88fea24f13b",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Free Trial",
    "state": "Enabled",
    "tenantId": "273e4da0-c196-4c65-b234-56a0dc18c5a5",
    "user": {
      "name": "priyarkan@gmail.com",
      "type": "user"
    }
  }
]
```

Then, I created the file shown in these instructions. I then moved into the directory with the file terraform_azure.tf.

Then I ran the following

1. `terraform init`
2. `terraform plan`
3. `terraform apply - this step builds the infrastructure in azure`

Below you can see the vm “myVM” shown in the Azure portal.

I found the public IP address through the interface on the portal (in myVM/overview). However, I was getting the error “Permission denied (publickey)”

```
priyakanipakam@Priyas-MBP-2 azure-docs % ssh azureuser@52.170.62.155
The authenticity of host '52.170.62.155 (52.170.62.155)' can't be established.
ECDSA key fingerprint is SHA256:P///cWZ2WmKXTys2hZUETiQyatm5gEscM5JhwTFqXkE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.170.62.155' (ECDSA) to the list of known hosts.
azureuser@52.170.62.155: Permission denied (publickey).
priyakanipakam@Priyas-MBP-2 azure-docs % ssh azureuser@52.170.62.155
azureuser@52.170.62.155: Permission denied (publickey).
```

When creating the VM using terraform, I couldn't find the private key file stored anywhere. However, when I ran "terraform apply" the output had the following private key:

```
tls_private_key = <<EOT
-----BEGIN RSA PRIVATE KEY-----
MIIJKQIBAAKCAgEAmo9HFsg9ZZ9zLxXPNRr6V0/Bo4vZ8Z22CgAc2ceG3eAqeDxH
CcoX36qdAq00ZrIsey4H79jeVBASG5kBzOycCLyKACnrazu7icTcqEoKRCwGlNw7
3yFDfkyhcEv83n67yKFCBWQWqnrPL7/udMEZG2LJM/hoRoZuWNITFaY4KbfIddPK
kEfdcfUhcqE2N8OWsLZmwmIu/HpHrzJ5jUmNlVDtfDkkUrc6MijhlEoIZ9vqjcbA
HQEXy52dVvS+voGNypU5utZddLKiB8iLSqJp3LjL6k7x+NC9gVMH/j273anhDY4D
f3UZlgulWbINZYbE2RvnMLxdYzVUNn5juIol55QBUx9oXcoejPkF7nngcXOdigMH
ofSQneyhn/CxZmn/87090kNfxbbsXovpyglQvL/cN/quyiAyEAnaEen1Gda2zVT1
Z+2JloBUVLZTwh2WJM57DbIhxsabQsHsb2xJIOCadmXBDKNN+444flYUZmBswo0Z
VYtuiyvqXiN7Zze8HSZQFlV0mt4csoPZlNd//c5sp+kZJg4fm6UnyttSLc8FhVF1
VHd677HDCC0J4eIfkepVosFXkx82KJYX/Qq3RaoeiDTOMepQH+jGhm3jgzRodu93
rBSc25GXA61JeBnghId4MmpEJf7jufiv44ylqtO6js8VFJnxkhjQUvmksskCAwEA
AQKCAgARhv4Az2RnCdX+GPMOiqipDhpfeZSVmdTy/WBIMiBROQKX7/3+VjSu4KI9
k3bFzp7MuUOdJ3LPBaOIDanum0Z2lyrvKdHQr01tKi37x6myYWu9pdwESbllsTr9
iSB7+tNAxUi/Y8JOQu8L8zAPfXZjbKh0hE8O8F/DIs+XPOpRkM3izsnB4uRx7+KD
JVUIrd2A3K56iLfXWeaXL5onM5YiwLKtv60a1Dx9O7WRI3OgkwfeXBOdNmqFZLNb
```

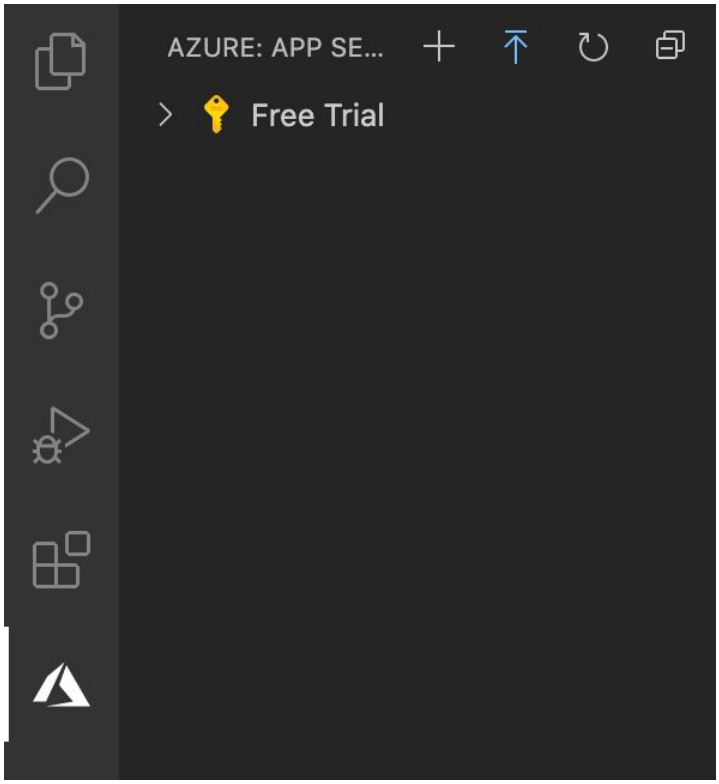
I copied it into a file in the directory I was in and called it key.pem and changed it to read only (using chmod 400 key.pem).
Then to ssh to connect to the VM using the following command:

```
ssh -i key.pem azureuser@52.170.62.155
```

```
priyakanipakam@Priyas-MBP-2 azure-docs % ssh -i key.pem azureuser@52.170.62.155
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1036-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Jan 20 23:49:05 UTC 2021

  System load:  0.0               Processes:           109
  Usage of /:   4.5% of 28.90GB   Users logged in:     0
  Memory usage: 5%                IP address for eth0: 10.0.1.4
  Swap usage:   0%


0 packages can be updated.
0 of these updates are security updates.

New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Wed Jan 20 23:41:31 2021 from 23.114.170.181
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@myvm:~$
```

Creating a web application

1. I set up node and npm on my system using ‘brew install node’
2. On VSCode I added the Azure App Service extension
3. Then I signed into my Azure account on VSCode through



4. I created a basic express application using the command: npx express-generator myExpressApp --view pug --git
5. Then I followed these instructions to run the app

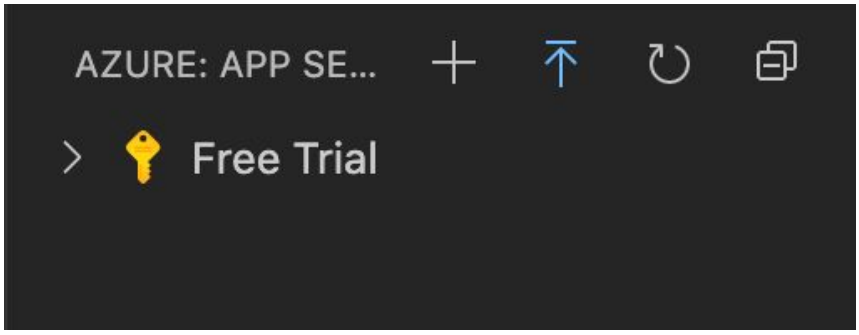
```
change directory:
  $ cd myExpressApp

install dependencies:
  $ npm install

run the app:
  $ DEBUG=myexpressapp:* npm start
```

6. The app is now up and running locally

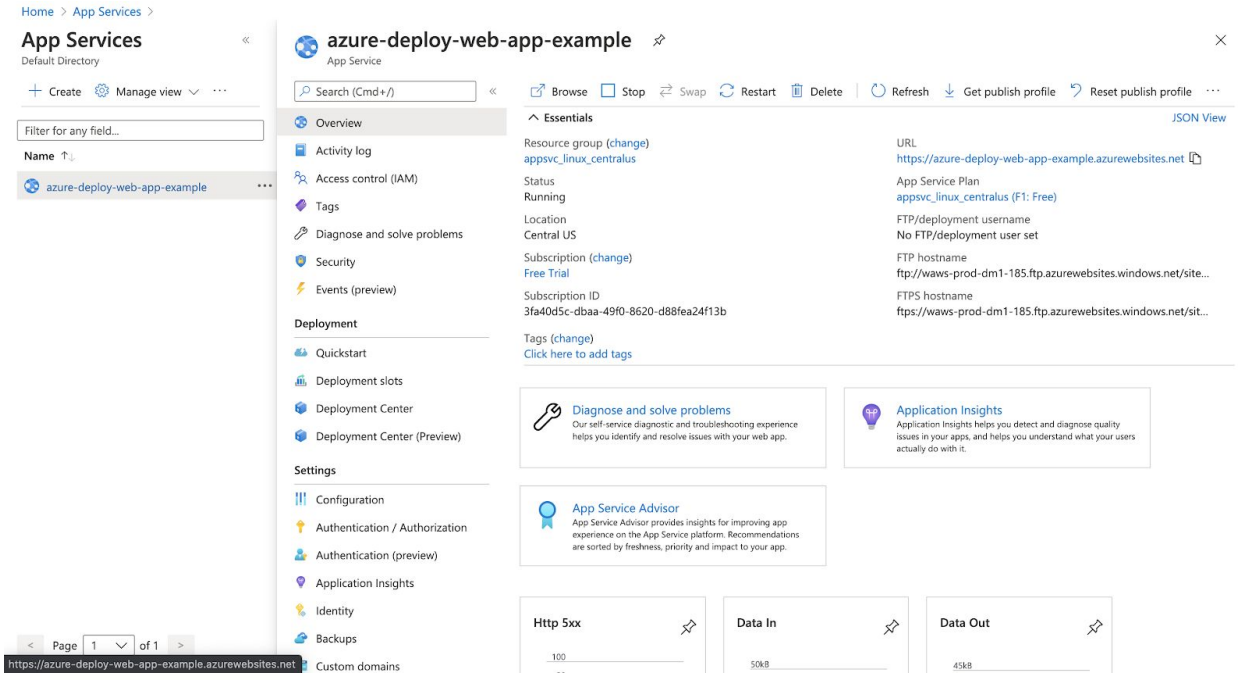
Express

Welcome to Express

7. To deploy it on Azure I clicked the blue arrow shown below and entered the name for my web app and the Node version I used.



8. Once deployment was complete I was able to see my deployed website on the App Serviced section of the Azure Portal



9. The App was deployed

Example App

Welcome to Example App that was deployed on Azure