

# Simple Windows Virtual Machine

We gonna deploy a simple Windows Virtual Machine. In order to complete this task, all you gonna need is the azuredeploy.json file and a couple of commands if you deploy using Azure CLI.

It's good to bear in mind that there are different ways to deploy resources in Azure, here a few options if you want to dig on the Azure Universe. You can run this template either using [PowerShell](#), [Azure CLI](#), [Azure Portal](#) or your favorite SDK.

## The Template

Don't let the size of the template scares you. The structure is very intuitive and once that you get the gist of it, you gonna see how easier your life will be regarding creating resources on Azure.

The only parameters that we need to inform are: **adminUsername**, **adminPassword**, **vmName** and **windowsOSVersion**. The windows version though comes in the template with the Default Value: "2016-Datacenter" which means that your O.S. version will be a Windows Server 2016 Datacenter. You will see that there are also other options.

Don't worry about changing anything on the file, either on the portal or using Azure CLI, you gonna be asked to insert this information, but bear in mind that there is some requirement for those two parameters:

- *adminUsername*: Usernames can be a maximum of 20 characters in length and cannot end in a period (".").

The following usernames are not allowed:

administrator	admin	user	user1
test	user2	test1	user3
admin1	1	123	a
actuser	adm	admin2	aspnet
backup	console	david	guest
john	owner	root	server
sql	support	support_388945a0	sys
test2	test3	user4	user5

- *adminPassword*: There are varying password length requirements, depending on the tool you are using:

Portal - between 12 - 72 characters

PowerShell - between 8 - 123 characters

CLI - between 12 - 123

- Have lower characters
- Have upper characters
- Have a digit
- Have a special character (Regex match `[\W_]`)

. The following passwords are not allowed:

```
abc@123 iloveyou! P@$$w0rd P@ssw0rd P@ssword123 Pa$$word pass@word1 Password!  
Password1 Password22
```

- *windowsOSVersion*: The following values are allowed:
  - "2008-R2-SP1"
    - "2012-Datacenter"
    - "2012-R2-Datacenter"
    - "2016-Nano-Server"
    - "2016-Datacenter-with-Containers"
    - "2016-Datacenter"
    - "2019-Datacenter"

Keep this in mind and let's roll with the Deployment.

## Deployment

---

There are a few ways to deploy your template. You can use [PowerShell](#), [Azure CLI](#), [Azure Portal](#) or your favorite SDK.

For this task, we gonna deploy using Visual Code and the portal and a little surprise for you at the end. :D

For Azure CLI I choose to use the Visual Code with Azure CLI extensions, if you like, you can find more information [here](#). But bare in mind that you don't need to use the Visual Code, you can stick with the old good always present **Command Line** on Windows or any **bash terminal**.

## Visual Code

type on the terminal windows: **az login**

The screenshot shows the Visual Studio Code editor with a file named `azuredeploy.json` open. The JSON content defines a virtual machine resource with properties like `name`, `type`, `apiVersion`, `location`, `dependsOn`, `hardwareProfile`, `storageProfile`, and `imageReference`. Below the editor, the `TERMINAL` tab is active, showing a bash shell. The user has entered `az login`, which has triggered a login process. The terminal output includes a note about launching a browser and a JSON snippet representing the user's login information.

```
119     },
120   },
121 },
122 {
123   "name": "[variables('virtualMachineName')]",
124   "type": "Microsoft.Compute/virtualMachines",
125   "apiVersion": "2019-03-01",
126   "location": "[variables('location')]",
127   "dependsOn": [
128     "[resourceId('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
129   ],
130   "properties": {
131     "hardwareProfile": {
132       "vmSize": "[variables('virtualMachineSize')]"
133     },
134     "storageProfile": {
135       "osDisk": {
136         "createOption": "fromImage",
137         "managedDisk": {
138           "storageAccountType": "[variables('osDiskType')]"
139         }
140       },
141       "imageReference": {
142         "publisher": "MicrosoftVisualStudio",
143         "offer": "VisualStudio",
144         "sku": "VS-2017-Comm-Win10-N",

```

```
Osiris-2:101-vm-windows-visualstudio2019 krisnatagoras$ az login
Note, we have launched a browser for you to login. For old experience with device code, use "az login --use-device-code"
You have logged in. Now let us find all the subscriptions to which you have access...
{
  "cloudName": "AzureCloud",
  "id": "10000000-0000-0000-0000-000000000000",
  "isDefault": false,
  "name": "Free Trial",
  "state": "Enabled",
  "tenantId": "40000000-0000-0000-0000-000000000000",
  "user": {
    "name": "krisnatagoras@10000000-0000-0000-0000-000000000000",
    "type": "user"
  }
}
```

You gonna be redirected to the Azure Portal where you can use your credentials to login into.

After login, you gonna have your credentials.

In order to set the right subscription, you can use the following command:

**az account set --subscription "< your subscription id >"**

The screenshot shows the Visual Studio Code editor with a file named `azuredeploy.json` open. The file contains a Bicep deployment template for a virtual machine. The code is as follows:

```
119     },
120   },
121   {
122     "name": "[variables('virtualMachineName')]",
123     "type": "Microsoft.Compute/virtualMachines",
124     "apiVersion": "2019-03-01",
125     "location": "[variables('location')]",
126     "dependsOn": [
127       "[resourceId('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
128     ],
129     "properties": {
130       "hardwareProfile": {
131         "vmSize": "[variables('virtualMachineSize')]"
132       },
133       "storageProfile": {
134         "osDisk": {
135           "createOption": "fromImage",
136           "managedDisk": {
137             "storageAccountType": "[variables('osDiskType')]"
138           }
139         },
140         "imageReference": {
141           "publisher": "MicrosoftVisualStudio",
142           "offer": "VisualStudio",
143           "sku": "VS-2017-Comm-Win10-N",
144         }
145       }
146     }
147   }
148 }
```

Below the editor, the `TERMINAL` tab is active, showing a `bash` session. The terminal output displays the details of the Azure account being used:

```
{
  "cloudName": "AzureCloud",
  "id": "...",
  "isDefault": true,
  "name": "Azure for Students",
  "state": "Enabled",
  "tenantId": "...",
  "user": {
    "name": "krisnatagoras@...",
    "type": "user"
  }
}
```

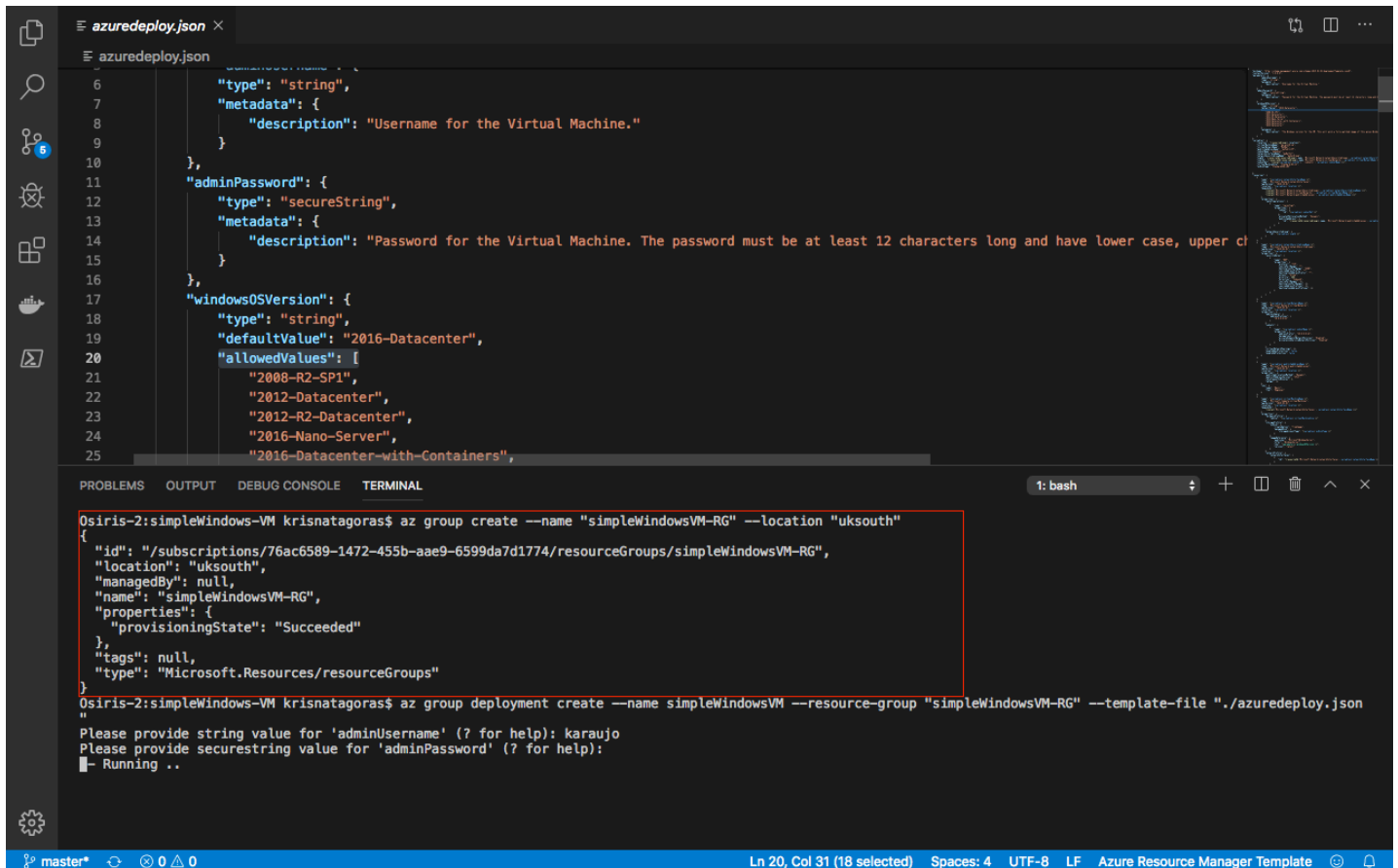
The terminal prompt shows the user is in a directory `Osiris-2:101-vm-windows-vvisualstudio2019` and has executed the command `az account set --subscription "`.

If you haven't yet created a Resource Group, we gonna do that now! A Resource Group is a container that holds related resources for an Azure solution. The resource group includes those resources that you want to manage as a group. You decide which resources belong in a resource group based on what makes the most sense for you or your organization.

That is the command sintax:

**az group create --name simpleWindowsVM-RG --location < yourlocation >**

For a list of locations, type: *az account list-locations*



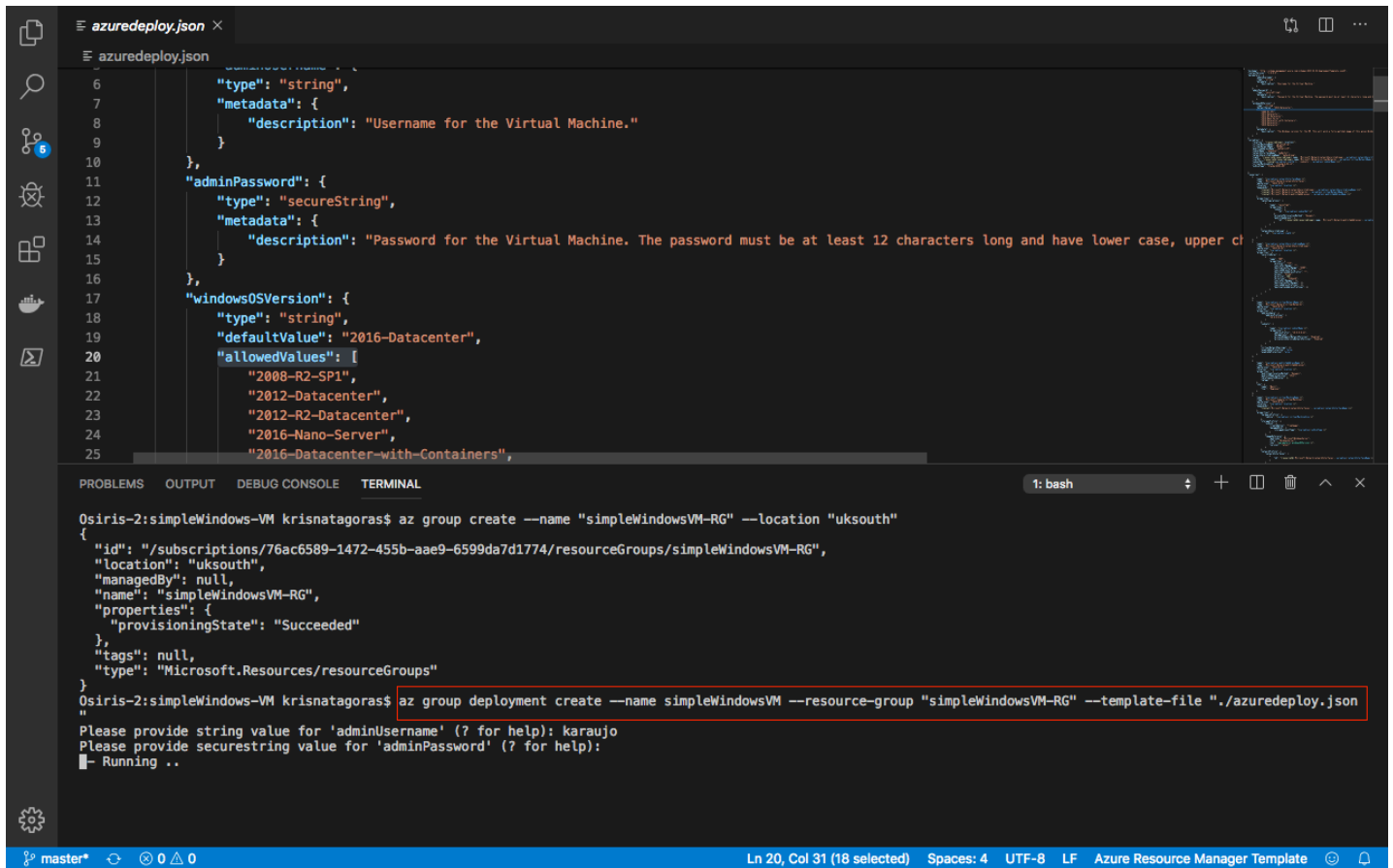
```
azuredploy.json
6      "type": "string",
7      "metadata": {
8        "description": "Username for the Virtual Machine."
9      }
10    },
11    "adminPassword": {
12      "type": "secureString",
13      "metadata": {
14        "description": "Password for the Virtual Machine. The password must be at least 12 characters long and have lower case, upper case, and numbers."
15      }
16    },
17    "windowsOSVersion": {
18      "type": "string",
19      "defaultValue": "2016-Datacenter",
20      "allowedValues": [
21        "2008-R2-SP1",
22        "2012-Datacenter",
23        "2012-R2-Datacenter",
24        "2016-Nano-Server",
25        "2016-Datacenter-with-Containers"
26      ]
27    }
28  },
29  "resources": [
30    {
31      "name": "simpleWindowsVM",
32      "type": "Microsoft.WindowsAzure.VirtualMachines",
33      "apiVersion": "2015-11-01",
34      "location": "[location]",
35      "plan": {
36        "name": "Windows Server 2016 Datacenter",
37        "product": "Microsoft.WindowsAzure.VMServer",
38        "publisher": "Microsoft",
39        "version": "2016-Datacenter",
40        "sku": "Datacenter",
41        "skuId": "00000000-0000-0000-0000-000000000000"
42      },
43      "hardware": {
44        "vmSize": "Standard_DS11",
45        "osDisk": {
46          "name": "osdisk",
47          "type": "Managed",
48          "caching": "ReadWrite",
49          "createOption": "FromImage"
50        },
51        "dataDisks": [
52          {
53            "name": "data1",
54            "type": "Managed",
55            "caching": "ReadWrite",
56            "createOption": "FromImage"
57          }
58        ]
59      },
60      "os": {
61        "type": "Windows",
62        "version": "2016-Datacenter",
63        "architecture": "x64",
64        "bootDiagnostics": {
65          "enabled": true,
66          "storageUri": "[storageUri]"
67        },
68        "adminUsername": "[adminUsername]",
69        "adminPassword": "[adminPassword]"
70      },
71      "network": {
72        "networkId": "[networkId]",
73        "subnetId": "[subnetId]",
74        "ipAddress": "[ipAddress]",
75        "port": 80
76      },
77      "provisioningState": "Succeeded"
78    }
79  ]
80}
```

```
Osiris-2:simpleWindows-VM krisnatagoras$ az group create --name "simpleWindowsVM-RG" --location "uksouth"
{
  "id": "/subscriptions/76ac6589-1472-455b-aae9-6599da7d1774/resourceGroups/simpleWindowsVM-RG",
  "location": "uksouth",
  "managedBy": null,
  "name": "simpleWindowsVM-RG",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
Osiris-2:simpleWindows-VM krisnatagoras$ az group deployment create --name simpleWindowsVM --resource-group "simpleWindowsVM-RG" --template-file "./azuredploy.json"
Please provide string value for 'adminUsername' (? for help): karaujo
Please provide securestring value for 'adminPassword' (? for help):
Running ..
```

With the Resource Group created, we can deploy our VM using the template:

**az group deployment create --name simpleWindowsVM --resource-group simpleWindowsVM-RG --template-file "azuredploy.json"**

Note that it's running:



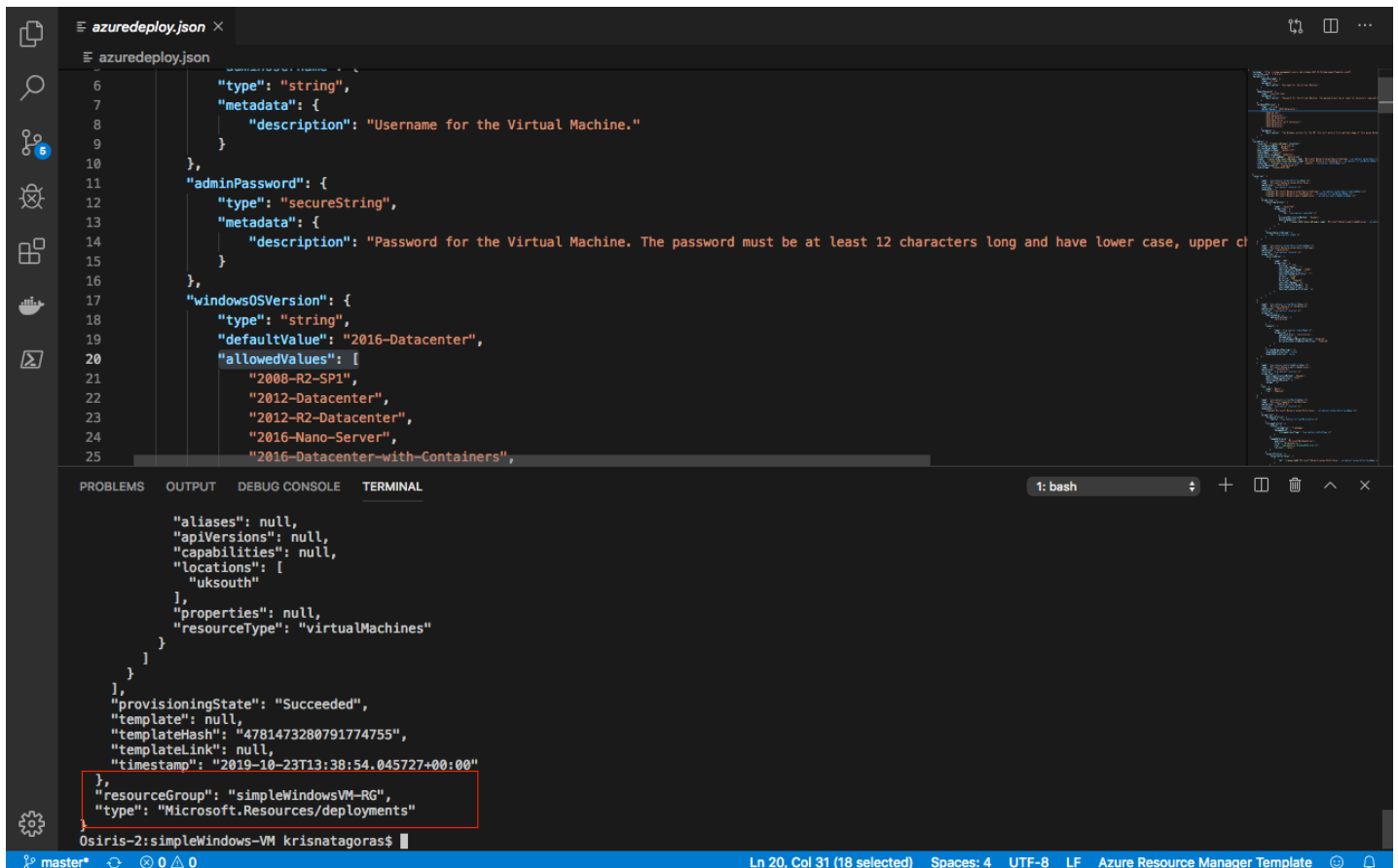
The image shows a Visual Studio Code editor with a file named `azuredeploy.json` open. The file contains a JSON configuration for an Azure Resource Manager template. The configuration includes fields for `adminUsername`, `adminPassword`, and `windowsOSVersion`. The `adminPassword` field is of type `secureString`. The `windowsOSVersion` field is of type `string` with a default value of `"2016-Datacenter"` and a list of allowed values: `"2008-R2-SP1"`, `"2012-Datacenter"`, `"2012-R2-Datacenter"`, `"2016-Nano-Server"`, and `"2016-Datacenter-with-Containers"`.

The terminal window at the bottom shows the execution of the `az group deployment create` command. The command is: `az group deployment create --name simpleWindowsVM --resource-group "simpleWindowsVM-RG" --template-file "./azuredeploy.json"`. The output shows the deployment details, including the resource ID, location, and provisioning state. The provisioning state is `"Succeeded"`.

```
6      "type": "string",
7      "metadata": {
8        "description": "Username for the Virtual Machine."
9      }
10    },
11    "adminPassword": {
12      "type": "secureString",
13      "metadata": {
14        "description": "Password for the Virtual Machine. The password must be at least 12 characters long and have lower case, upper case, and numbers."
15      }
16    },
17    "windowsOSVersion": {
18      "type": "string",
19      "defaultValue": "2016-Datacenter",
20      "allowedValues": [
21        "2008-R2-SP1",
22        "2012-Datacenter",
23        "2012-R2-Datacenter",
24        "2016-Nano-Server",
25        "2016-Datacenter-with-Containers",
26      ]
27    }
28  }
29}
```

```
Osiris-2:simpleWindows-VM krisnatagoras$ az group create --name "simpleWindowsVM-RG" --location "uksouth"
{
  "id": "/subscriptions/76ac6589-1472-455b-aae9-6599da7d1774/resourceGroups/simpleWindowsVM-RG",
  "location": "uksouth",
  "managedBy": null,
  "name": "simpleWindowsVM-RG",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
Osiris-2:simpleWindows-VM krisnatagoras$ az group deployment create --name simpleWindowsVM --resource-group "simpleWindowsVM-RG" --template-file "./azuredeploy.json"
Please provide string value for 'adminUsername' (? for help): karaujo
Please provide securestring value for 'adminPassword' (? for help):
Running ..
```

And there we go, our deploy is Succeeded:

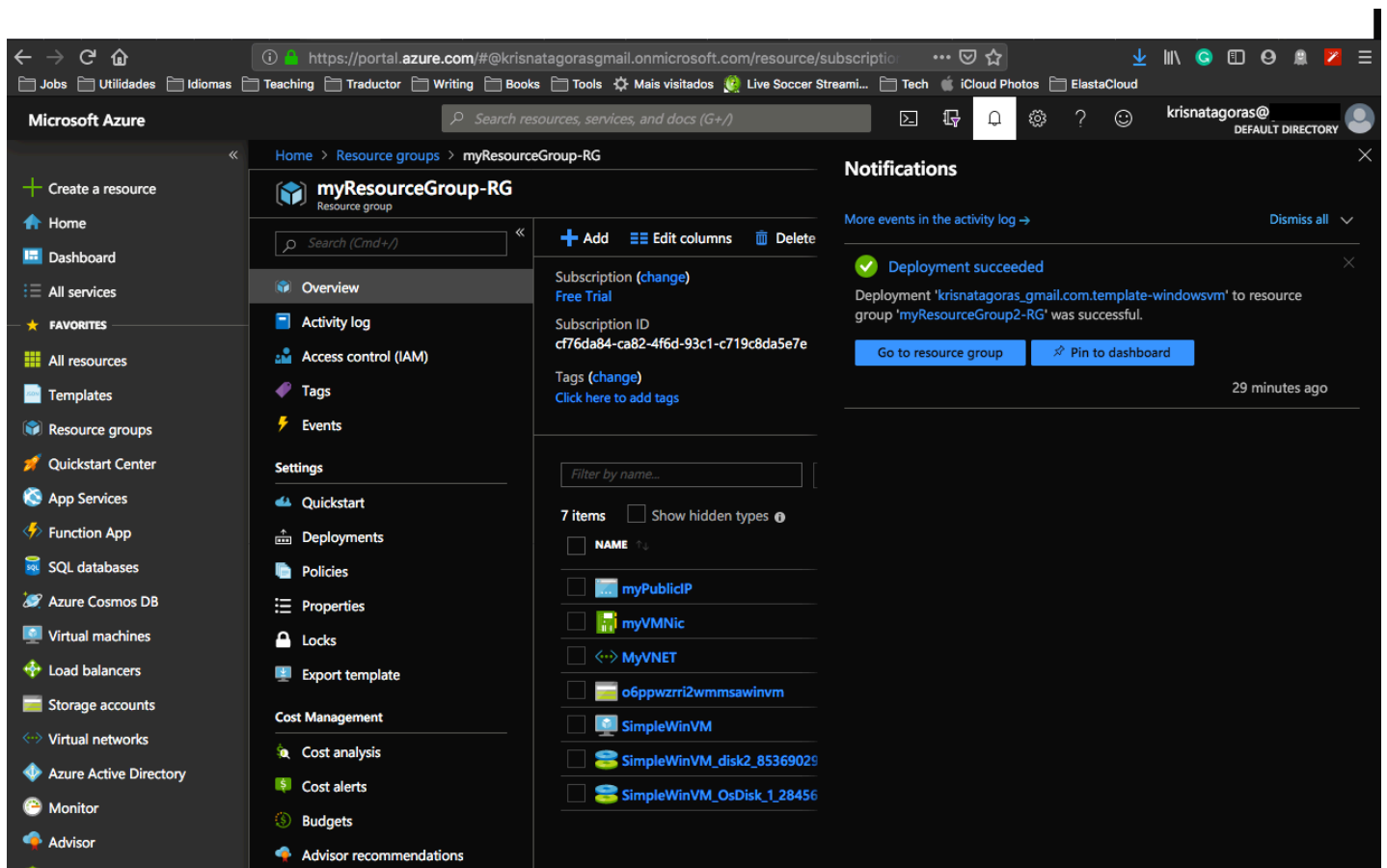


The image shows the same Visual Studio Code editor with the `azuredeploy.json` file. The terminal window now shows the output of the `az group deployment create` command, which has completed successfully. The output shows the deployment details, including the resource ID, location, and provisioning state. The provisioning state is `"Succeeded"`.

```
6      "type": "string",
7      "metadata": {
8        "description": "Username for the Virtual Machine."
9      }
10    },
11    "adminPassword": {
12      "type": "secureString",
13      "metadata": {
14        "description": "Password for the Virtual Machine. The password must be at least 12 characters long and have lower case, upper case, and numbers."
15      }
16    },
17    "windowsOSVersion": {
18      "type": "string",
19      "defaultValue": "2016-Datacenter",
20      "allowedValues": [
21        "2008-R2-SP1",
22        "2012-Datacenter",
23        "2012-R2-Datacenter",
24        "2016-Nano-Server",
25        "2016-Datacenter-with-Containers",
26      ]
27    }
28  }
29}
```

```
Osiris-2:simpleWindows-VM krisnatagoras$ az group deployment create --name simpleWindowsVM --resource-group "simpleWindowsVM-RG" --template-file "./azuredeploy.json"
{
  "aliases": null,
  "apiVersions": null,
  "capabilities": null,
  "locations": [
    "uksouth"
  ],
  "properties": null,
  "resourceType": "virtualMachines"
}
{
  "provisioningState": "Succeeded",
  "template": null,
  "templateHash": "4781473280791774755",
  "templateLink": null,
  "timestamp": "2019-10-23T13:38:54.045727+00:00"
},
{
  "resourceGroup": "simpleWindowsVM-RG",
  "type": "Microsoft.Resources/deployments"
}
Osiris-2:simpleWindows-VM krisnatagoras$
```

Let's go and check the resource at the Azure Portal: Go the Resource Group, find the Resource group you've created. And there it's your brand new **Virtual Machine**:

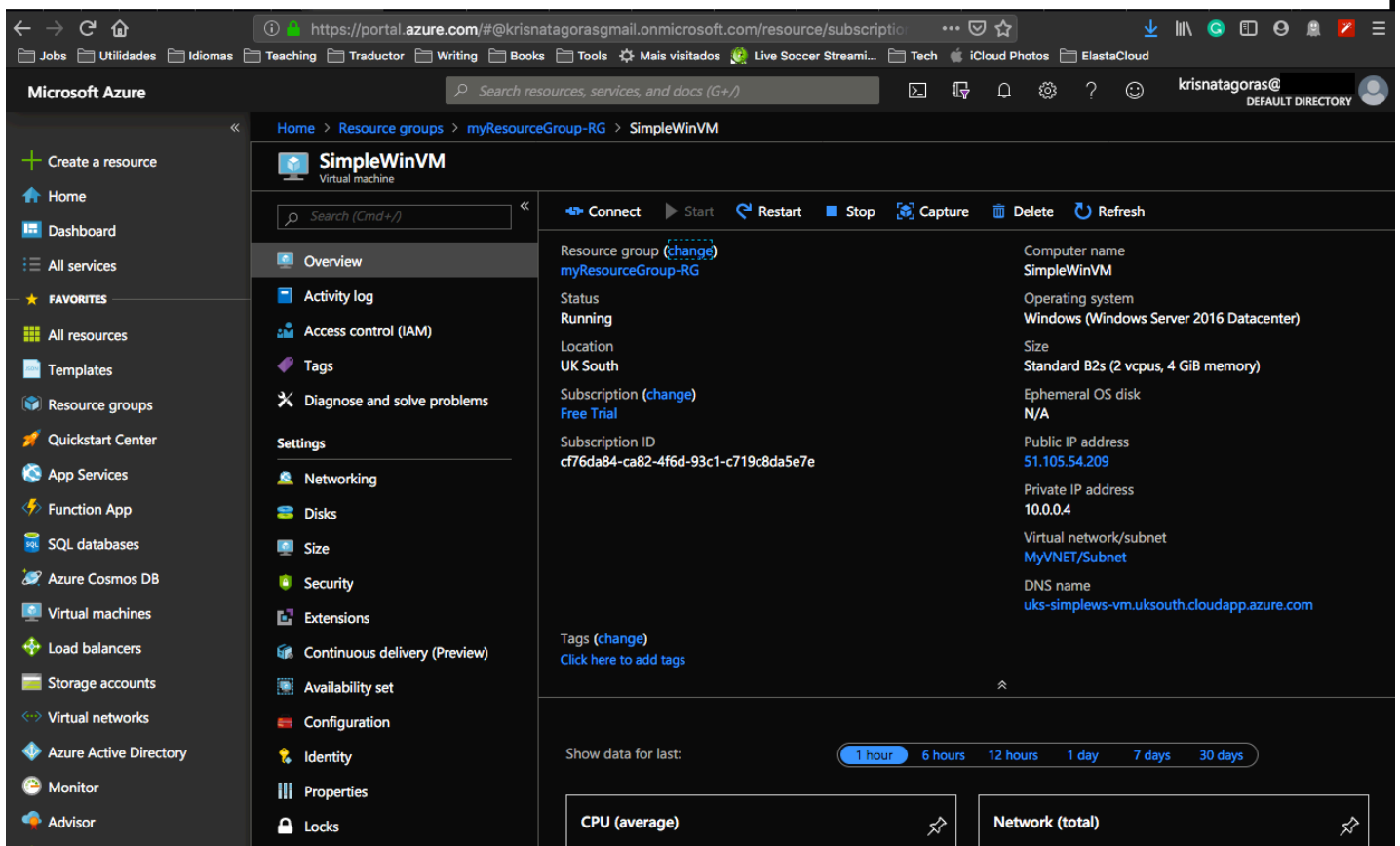


Note that beyond your Virtual Machine there are also all the resources that the VM needs in order to run:

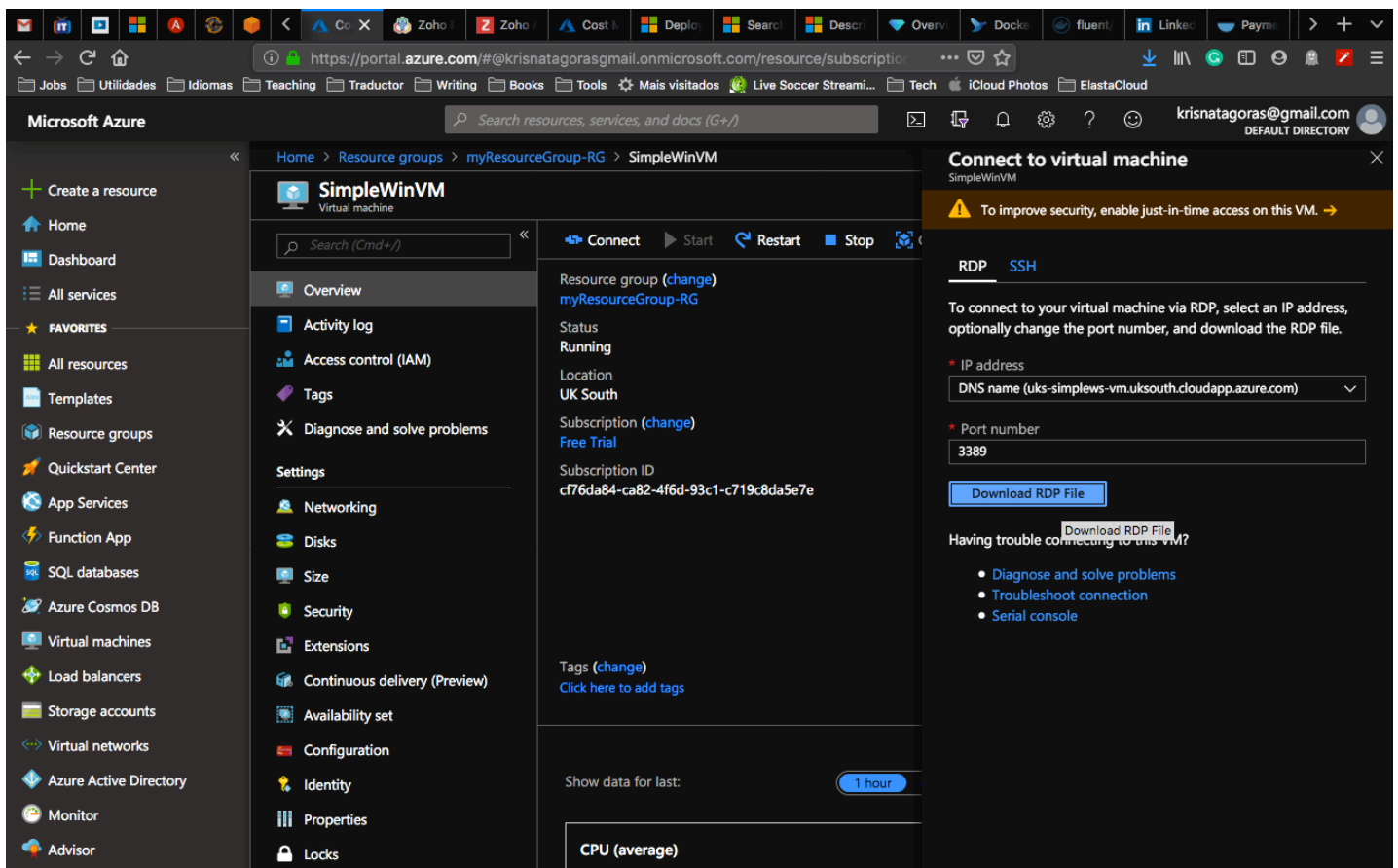
- Virtual Network Interface*
- Public IP Address*
- Storage Account*
- Virtual Network*
- Disks*

Compare these resources with your ARM Template is a good exercise to have a better understanding.

Continuing, let's open your Virtual Machine and check the details:

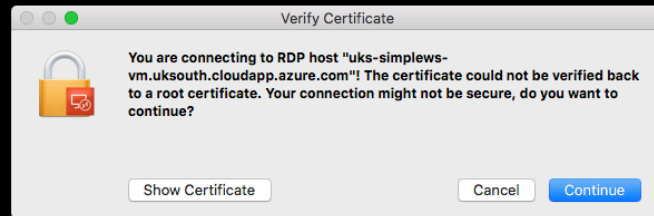


We gonna use Remote Desktop Protocol to connect into this Virtual Machine. Click on [Connect] and then download the RDP File.





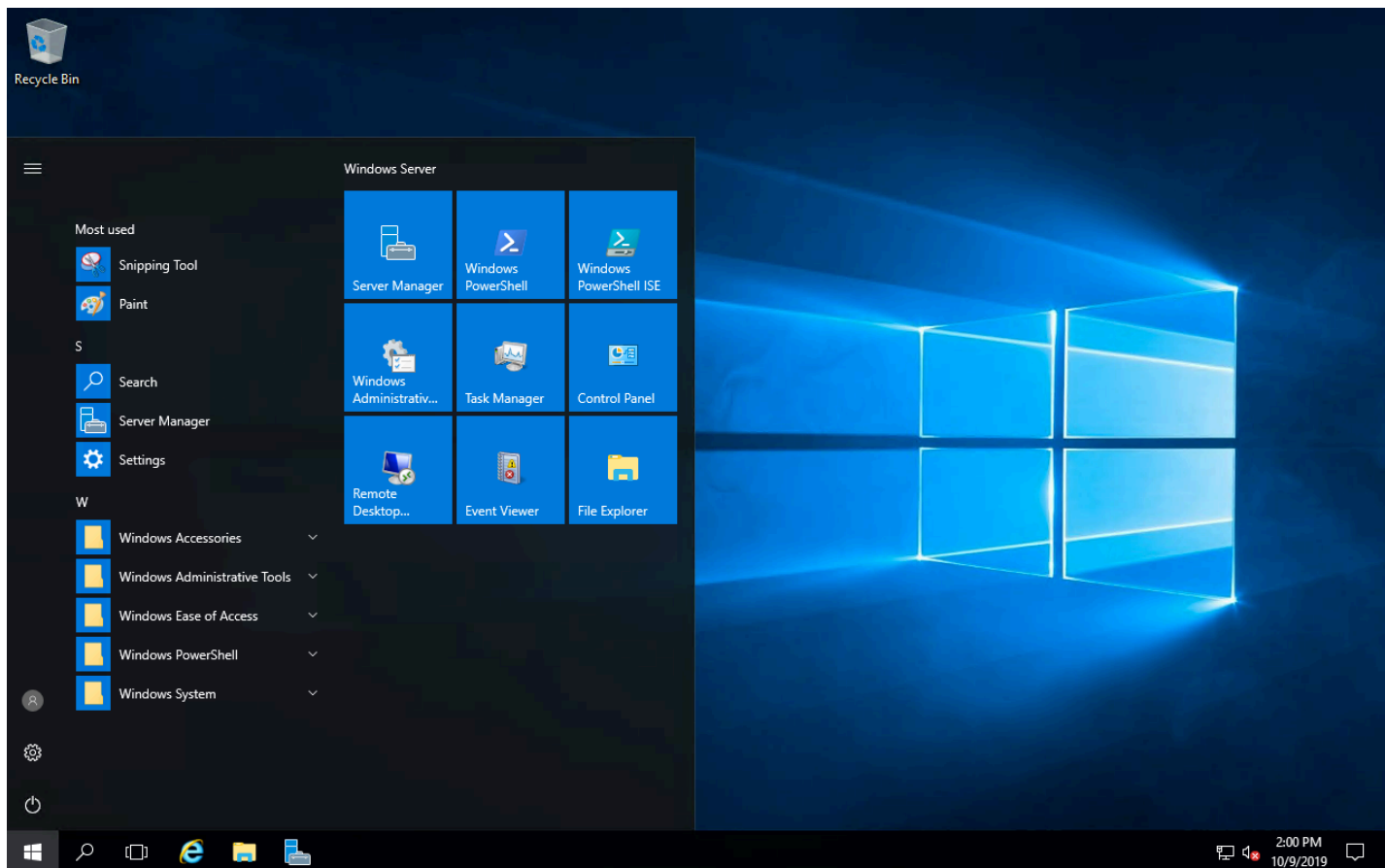
Double click on the file and you should open the RDP Software to connect to the Virtual Machine.



Negotiating Credentials ...

Click on [Continue]

Type your user and password and Voilà, you are connected in your Windows Virtual Machine... how cool is that uh?

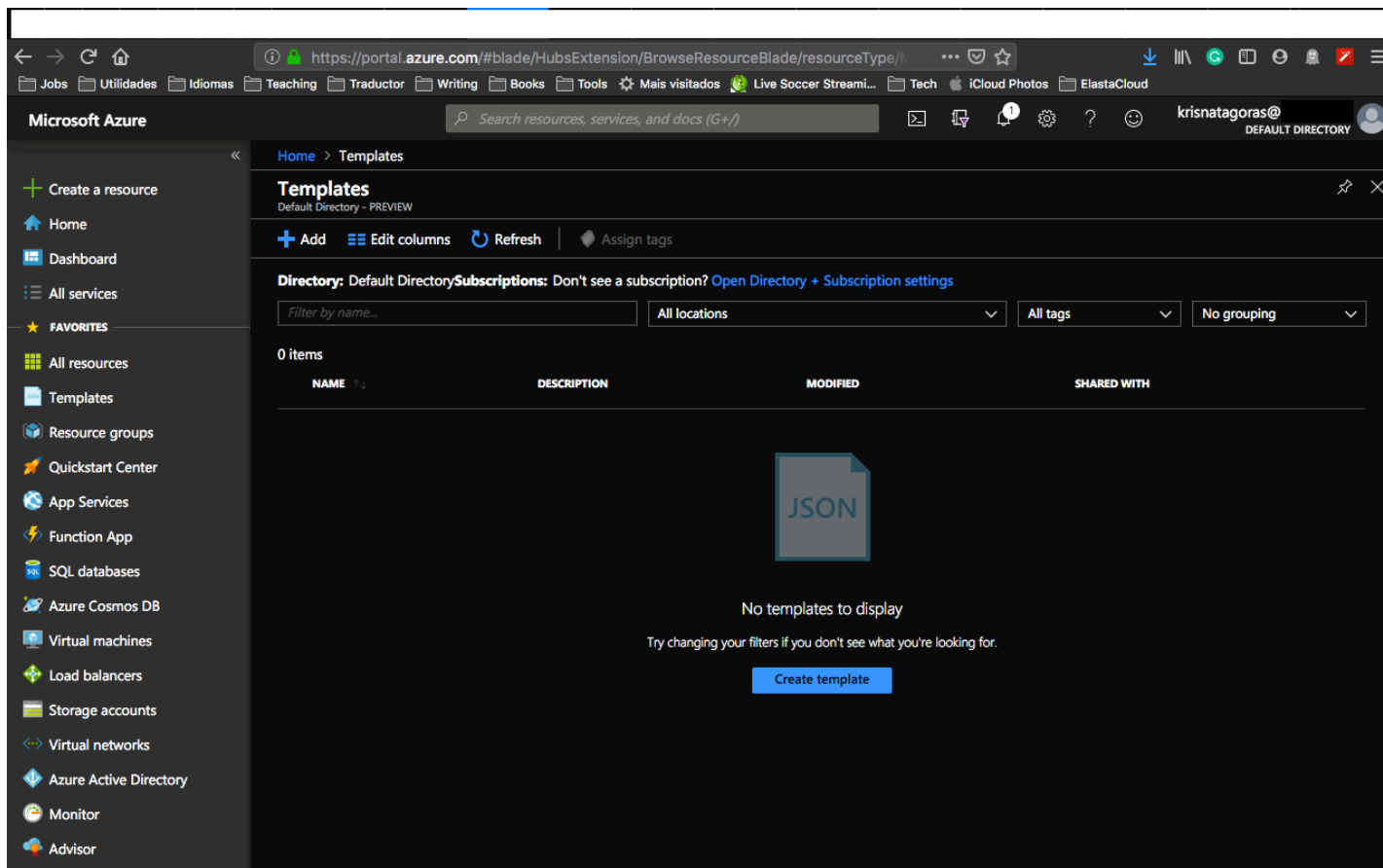


Now you are ready to practice, make your hand dirty buddy!

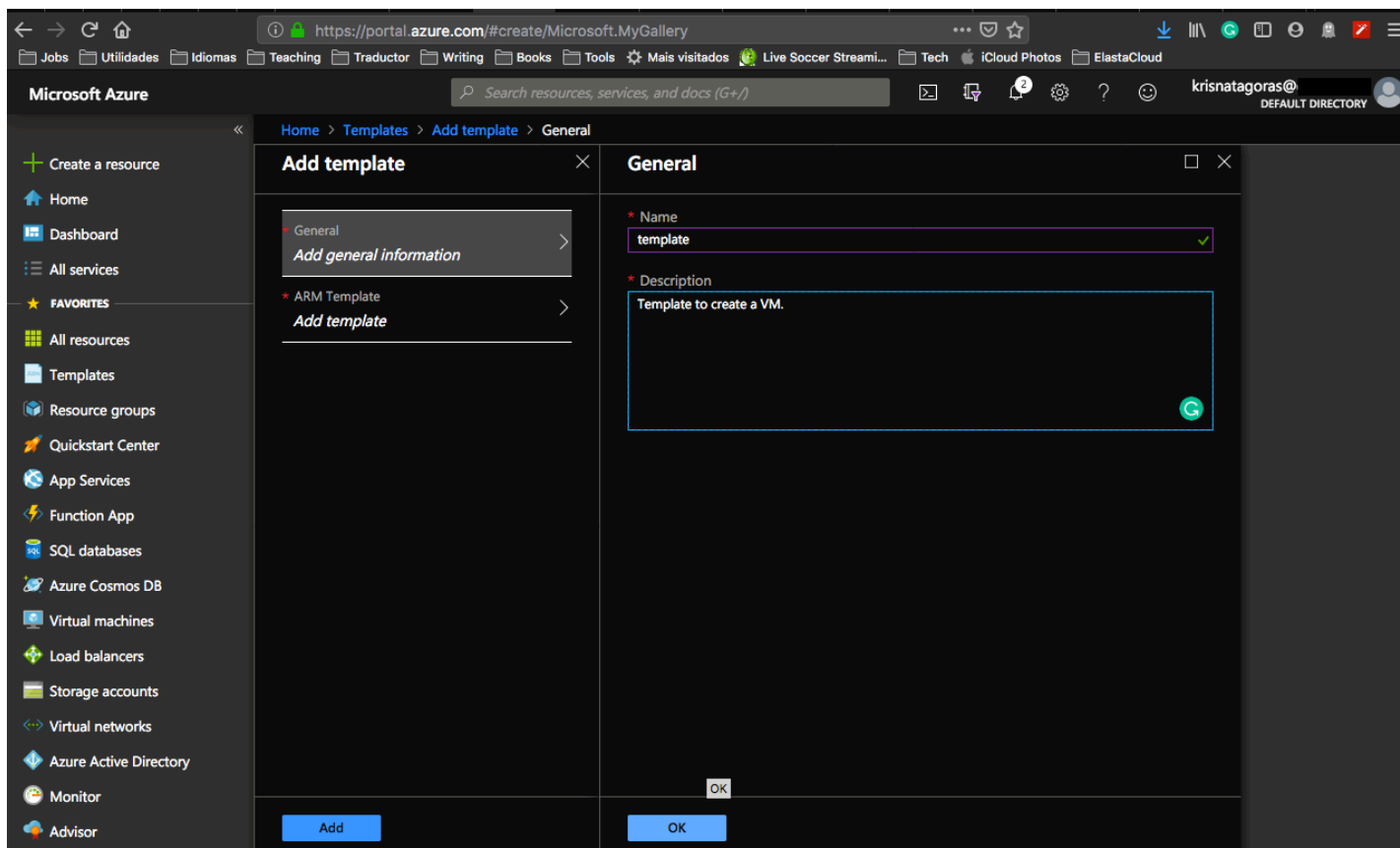
## Using the Portal

At the Portal, in All Services look for **Templates**, you can favorite this service.

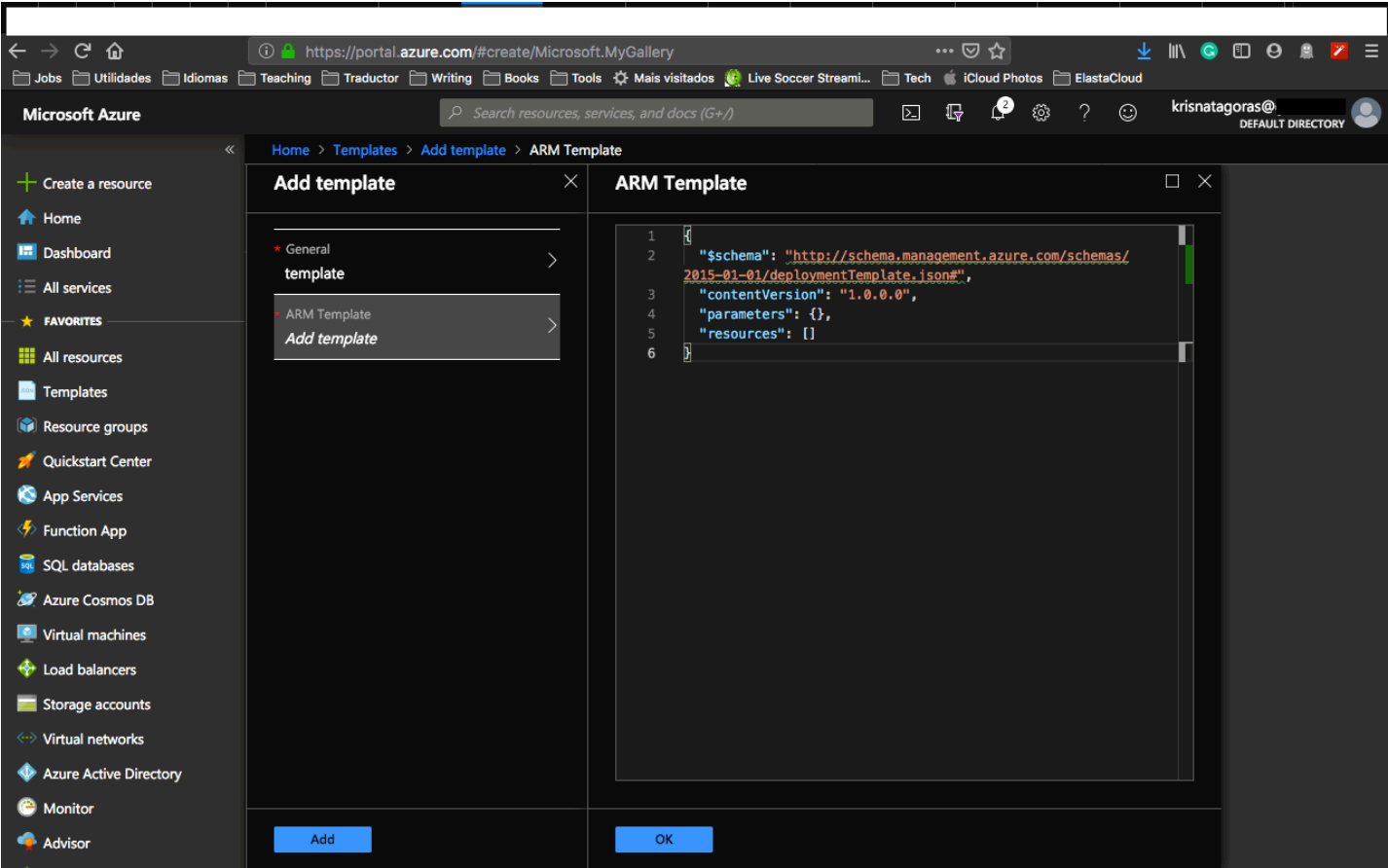
Click in **Add** to add your template:



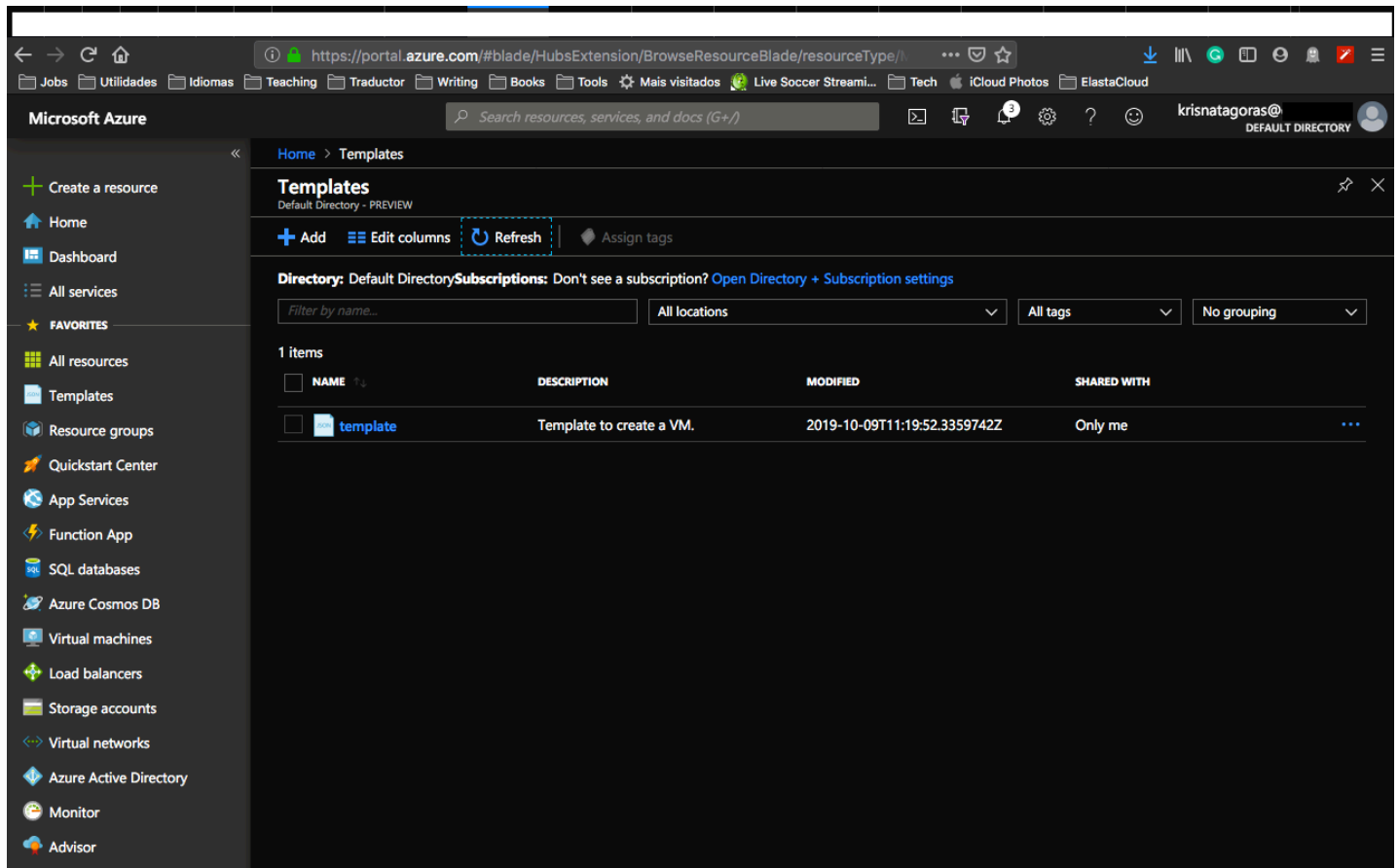
On General, type a name and a description for your template, and click on [OK].



On ARM Template, replace the contents of the template with your template, and click on [OK].



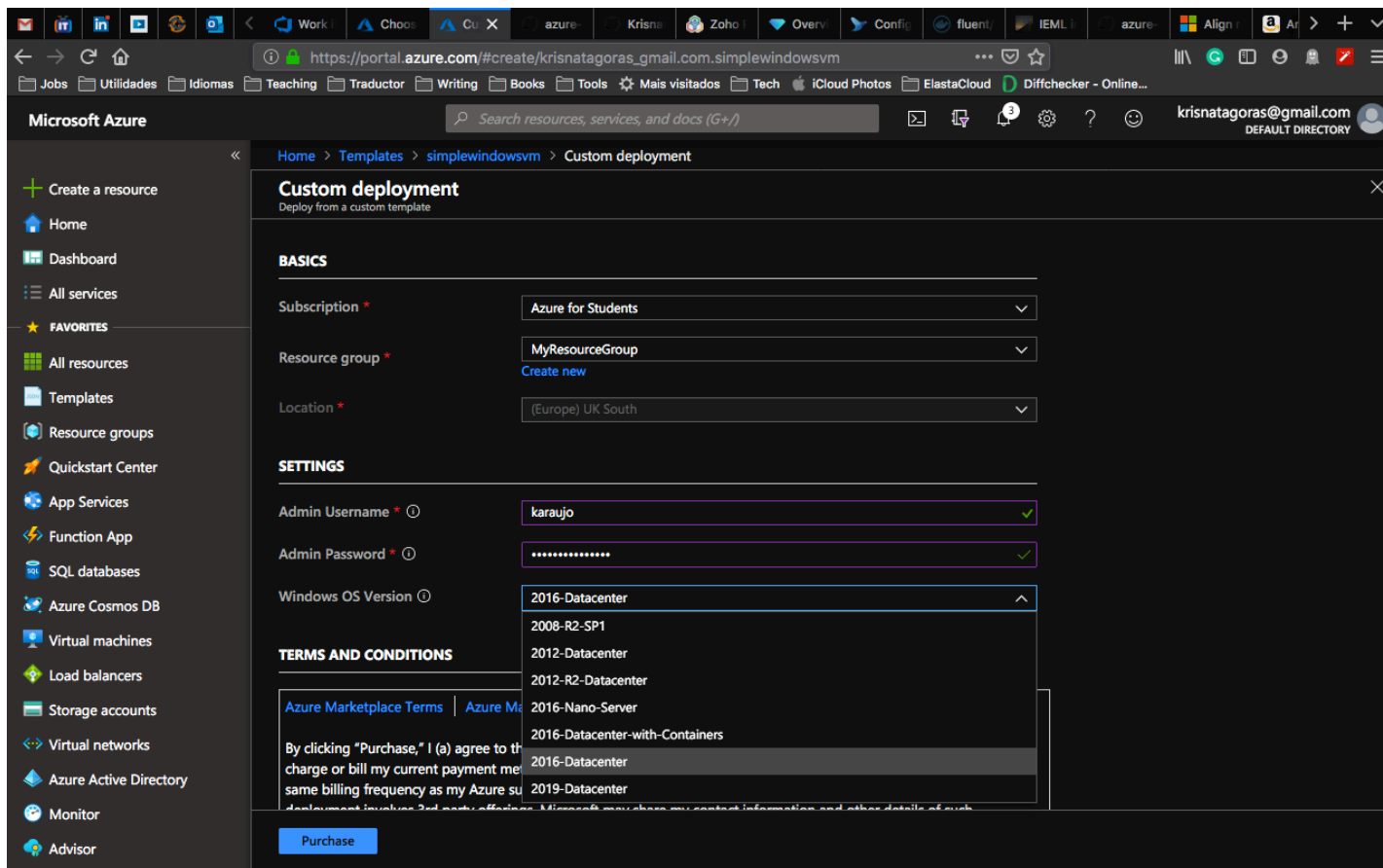
Click on the refresh button and there is your template:



Open the template and click in [Deploy]

On the screen Custom Deployment, check your information and if you don't have the Resource Group you can click and [create new]: The name of the Resource Group is: simpleWindowsVM-RG, select your location and click on [OK]. Insert the rest of the information for the parameters, note that it shows for your option to choose which Windows version you want.

By now you shall be familiar with these parameters, select [I agree] and click on [Purchase].



And voilà, you have your new VM deployed.

To connect with the Virtual Machine you can repeat the same process as before, using the RDP file.

Now is time to get your hands dirty, don't forget that you are in the cloud now :D

**p.s.: Pretty easy to create resources on Azure, right? But if you are the sort of IT guy that always looks for automating things on the extreme :D Surprise, surprise!. Just click on the button below and it will automatically deploy the VM on your Azure Portal.**



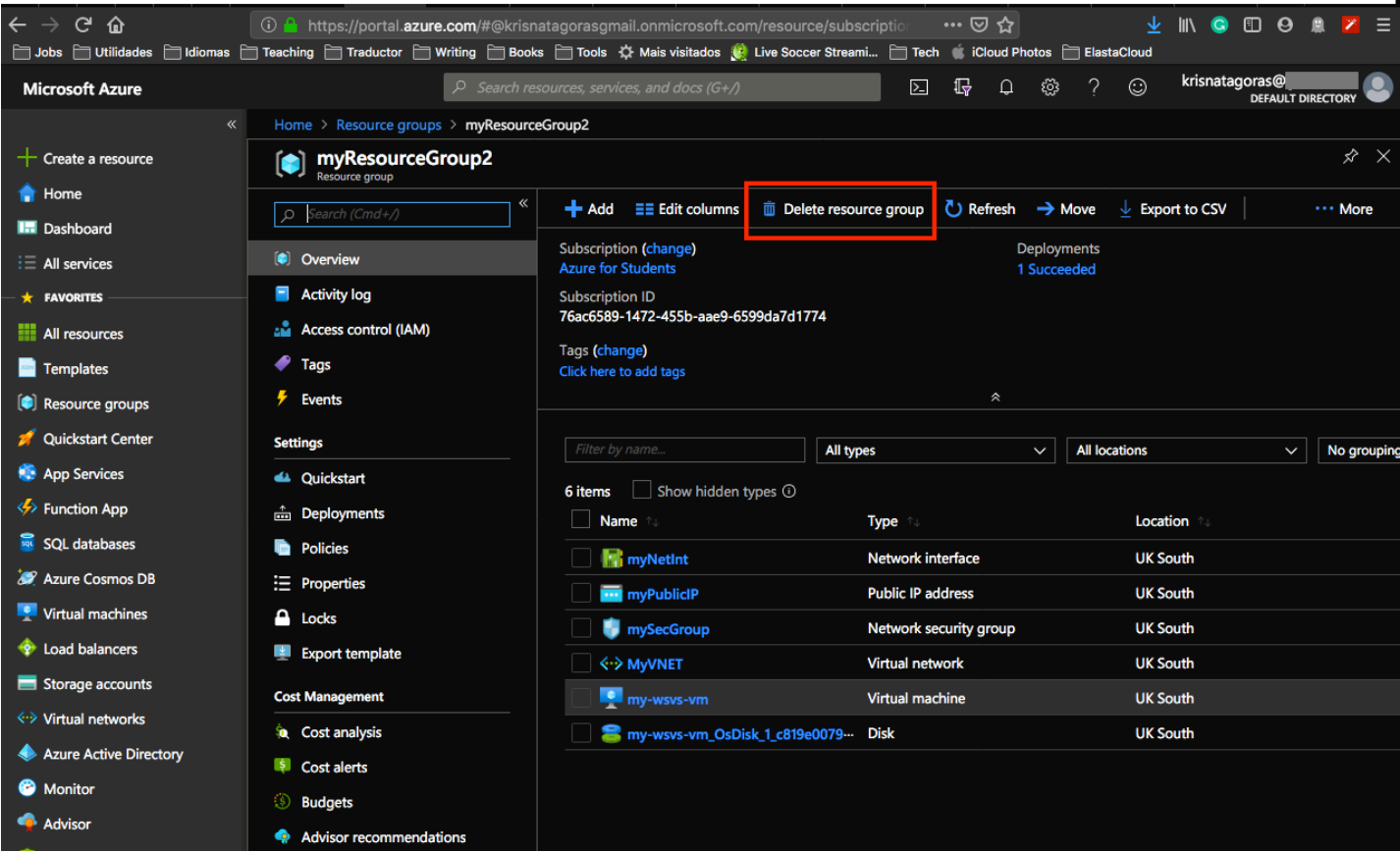
**Important disclaimer: Azure charges you for the resources you are using, and you don't want to finish all your credits at once, right? So, for not running out of credit, don't forget to stop the VM at the portal or even delete the Resource Group you create to avoid any unnecessary charges.**

## How to shutdown your resources:

### Using the portal:

On the portal, open your Resource Group, if you will not use the service or VM anymore, you can just click

on the [Delete] Button.



You can also just stop the service or Virtual Machine in case you gonna need the resource. Open the resource and click on Stop.

Microsoft Azure portal interface showing the details of a virtual machine named **my-wsvs-vm**.

The interface includes a left sidebar with navigation options (Home, Dashboard, All services, FAVORITES, All resources, Templates, Resource groups, Quickstart Center, App Services, Function App, SQL databases, Azure Cosmos DB, Virtual machines, Load balancers, Storage accounts, Virtual networks, Azure Active Directory, Monitor, Advisor) and a top navigation bar with the URL <https://portal.azure.com/#@krisnatagorasgmail.onmicrosoft.com/resource/subscriptions/76ac6589-1472-455b-aae9-6599da7d1774>.

The main content area displays the **my-wsvs-vm** details, including:

- Resource group:** [myResourceGroup2](#)
- Status:** Running
- Location:** UK South
- Subscription:** [Azure for Students](#)
- Subscription ID:** 76ac6589-1472-455b-aae9-6599da7d1774
- Computer name:** my-wsvs-vm
- Operating system:** Windows (Windows Server 2019 Datacenter)
- Size:** Standard D2 v2 (2 vcpus, 7 GiB memory)
- Ephemeral OS disk:** N/A
- Public IP address:** -
- Private IP address:** -
- Virtual network/subnet:** -
- DNS name:** -

The **Stop** button is highlighted with a red box. Other buttons visible include **Connect**, **Start**, **Restart**, **Capture**, **Delete**, and **Refresh**.

At the bottom, there are sections for **CPU (average)** and **Network (total)** metrics, with a "Show data for last:" dropdown set to **1 hour**.

Just refresh your screen and you are good to go.