

Simple Windows Server 2016 - Datacenter with Visual Studio 2019

The intent of this README is to guide you in a deployment a **Windows Server 2016 - Datacenter Standard-D2-v2** (Two VCPUs and 7 GiB RAM) with the **Visual Studio Community 2019**.

This template was designed to be an easy and fast way to create a Virtual Machine ready for you to start coding with Visual Studio.

You can deploy this template either using the [Azure Portal](#) or [Azure CLI](#). Let's dig on the template and have some fun.

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 two parameters that we need to inform are: **adminUsername** and **adminPassword**.

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
```

Keeping that in mind, let's rock with the Deployment.

Deployment

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

For this task, we will deploy using the Portal and 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**.

Using Azure CLI with Visual Code

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

The screenshot shows a Visual Studio Code editor with a file named `azuredeploy.json` open. The file contains a Bicep resource definition for a virtual machine. The terminal window at the bottom shows the command `az login` being executed, with a message indicating that the user has been redirected to the Azure Portal for login. The terminal output shows the user's credentials and the subscription they are logged into.

```
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 }
```

PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL 1: bash

```
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": "4d98f108-8c3b-462c-b982-78d7e22f51e3",
  "isDefault": false,
  "name": "Free Trial",
  "state": "Enabled",
  "tenantId": "4d98f108-8c3b-462c-b982-78d7e22f51e3",
  "user": {
    "name": "krisnatagoras@4d98f108-8c3b-462c-b982-78d7e22f51e3",
    "type": "user"
  }
}
```

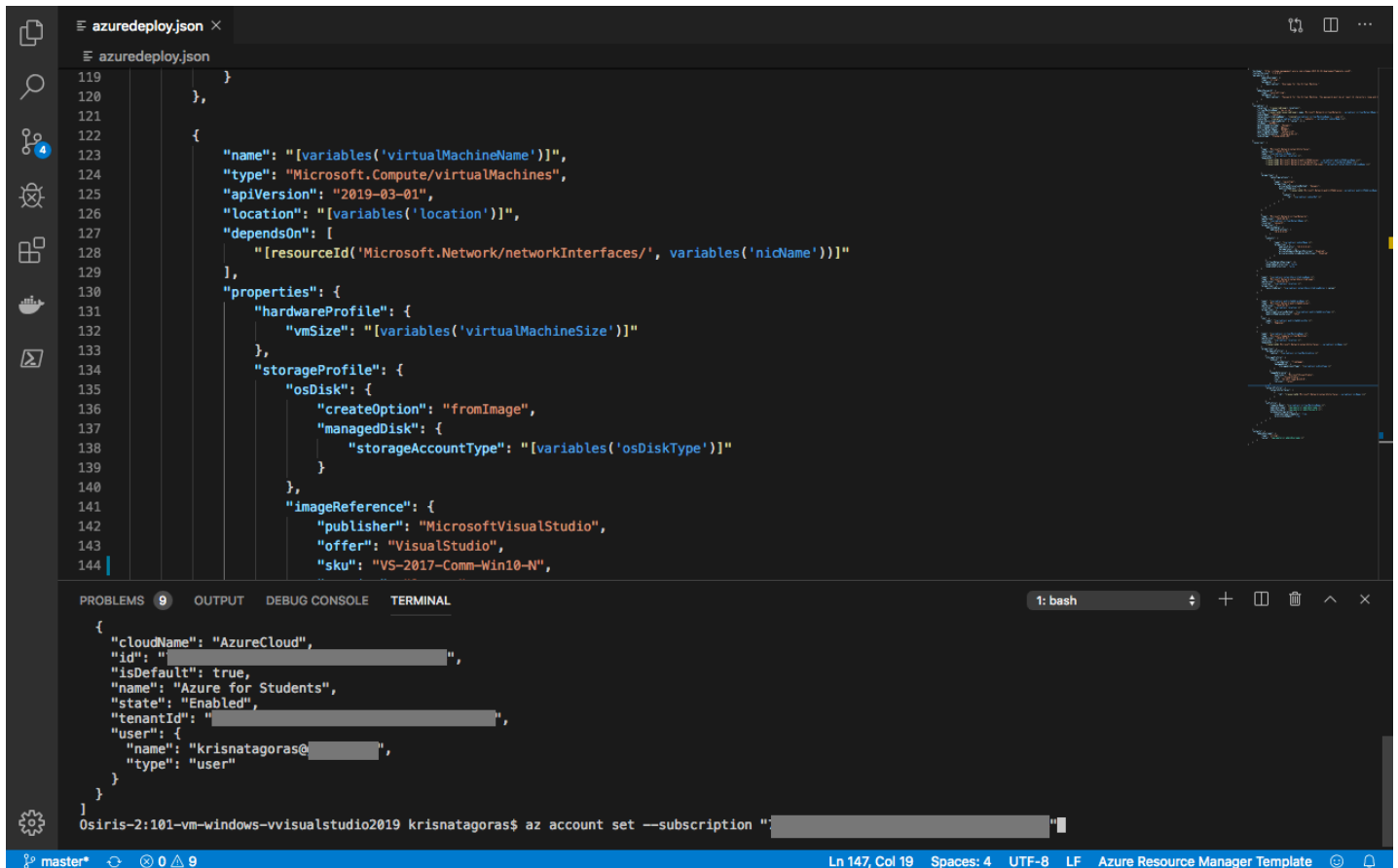
master* 0 9 Ln 147, Col 19 Spaces: 4 UTF-8 LF Azure Resource Manager Template

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 image shows a VS Code editor with a file named `azuredeploy.json` open. The file contains a JSON template for an Azure 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 current Azure account, including the cloud name, ID, name, state, tenant ID, and user information. The command `az account set --subscription` is partially visible at the bottom of the terminal.

Resource Group

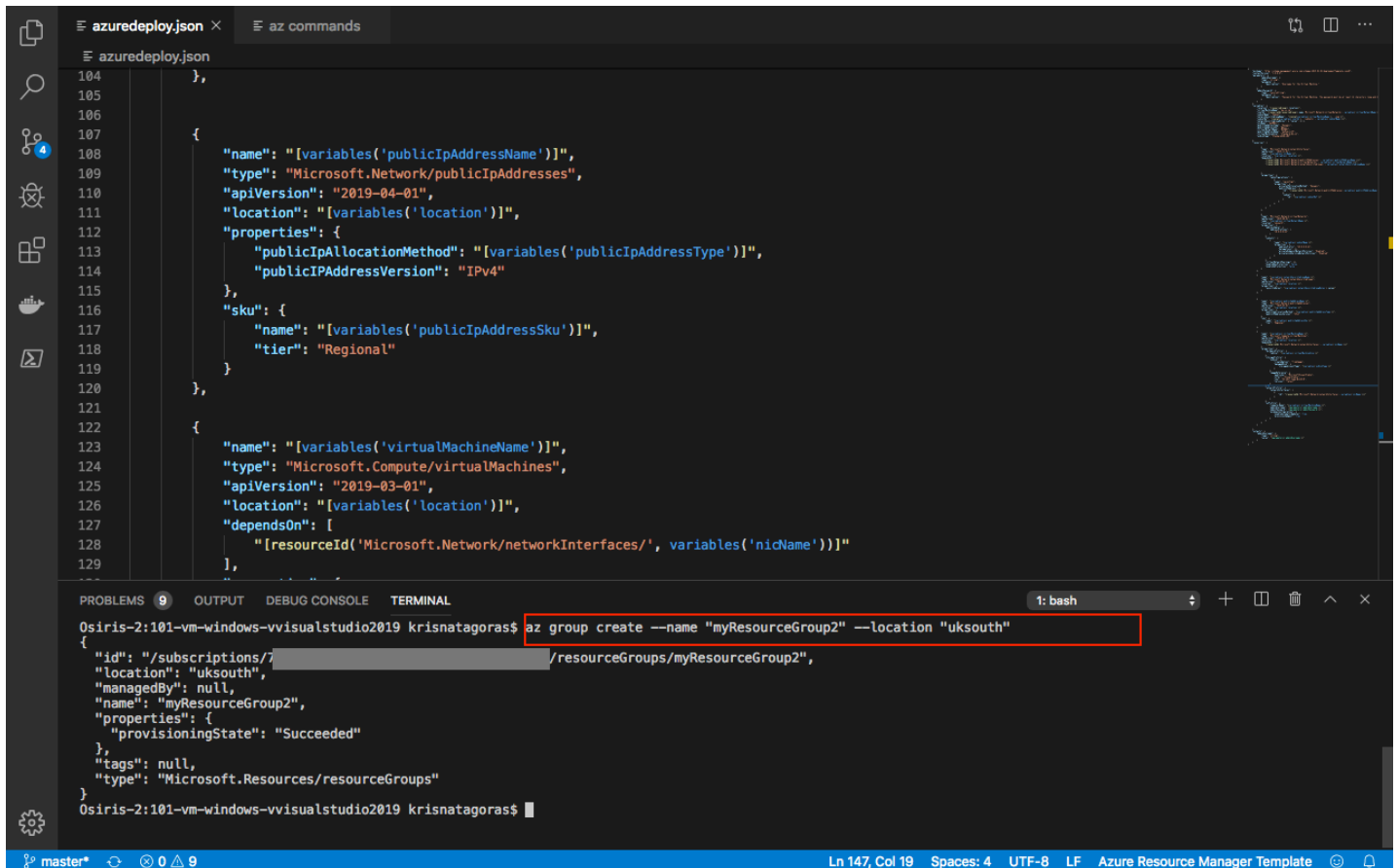
After you logged in, we gonna need to create a Resource Group for our deployment. If you haven't yet created a Resource Group, we gonna do that now! But what is a Resource Group, one might ask. Bare with me! A Resource Group is a container that holds related resources for an Azure solution. The resource group includes those resources that you want to manage as a group. Simply saying, it's like a folder that contains files. Simple as that ;-)

To create a Resource Group, you need a name and the location for your Resource Group.

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

To create the Resource group, just type the command:

az group create --name < mygroupname> --location < yourlocation >



```
104 },
105
106
107 {
108   "name": "[variables('publicIpAddressName')]",
109   "type": "Microsoft.Network/publicIpAddresses",
110   "apiVersion": "2019-04-01",
111   "location": "[variables('location')]",
112   "properties": {
113     "publicIpAllocationMethod": "[variables('publicIpAddressType')]",
114     "publicIpAddressVersion": "IPv4"
115   },
116   "sku": {
117     "name": "[variables('publicIpAddressSku')]",
118     "tier": "Regional"
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 }
```

```
Osiris-2:101-vm-windows-vvisualstudio2019 krisnatagoras$ az group create --name "myResourceGroup2" --location "uksouth"
{
  "id": "/subscriptions/[REDACTED]/resourceGroups/myResourceGroup2",
  "location": "uksouth",
  "managedBy": null,
  "name": "myResourceGroup2",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
Osiris-2:101-vm-windows-vvisualstudio2019 krisnatagoras$
```

Super simple, right? Now that we have our **Resource Group** created, let's deploy our Virtual Machine.

az group deployment create --name "name of your deployment" --resource-group "The group you created" --template-file "./azuredeploy.json"

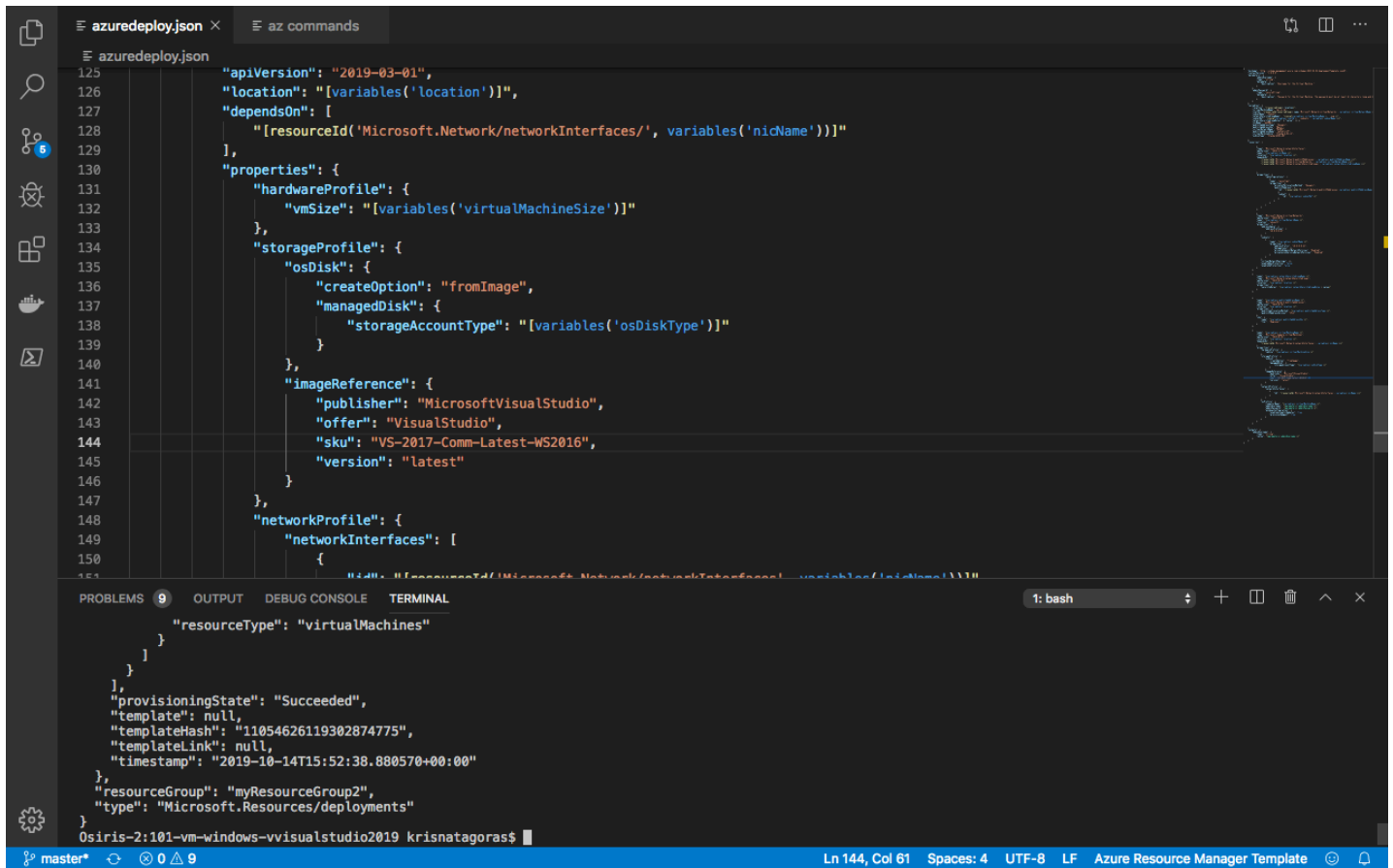
The image shows a Visual Studio Code editor with a Bicep file named `azuredeploy.json` open. The file defines an Azure resource group and a virtual machine. The terminal at the bottom shows the command to create the resource group and the output indicating success.

```
104 },
105
106
107 {
108   "name": "[variables('publicIpAddressName')]",
109   "type": "Microsoft.Network/publicIpAddresses",
110   "apiVersion": "2019-04-01",
111   "location": "[variables('location')]",
112   "properties": {
113     "publicIpAllocationMethod": "[variables('publicIpAddressType')]",
114     "publicIpAddressVersion": "IPv4"
115   },
116   "sku": {
117     "name": "[variables('publicIpAddressSku')]",
118     "tier": "Regional"
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 }
```

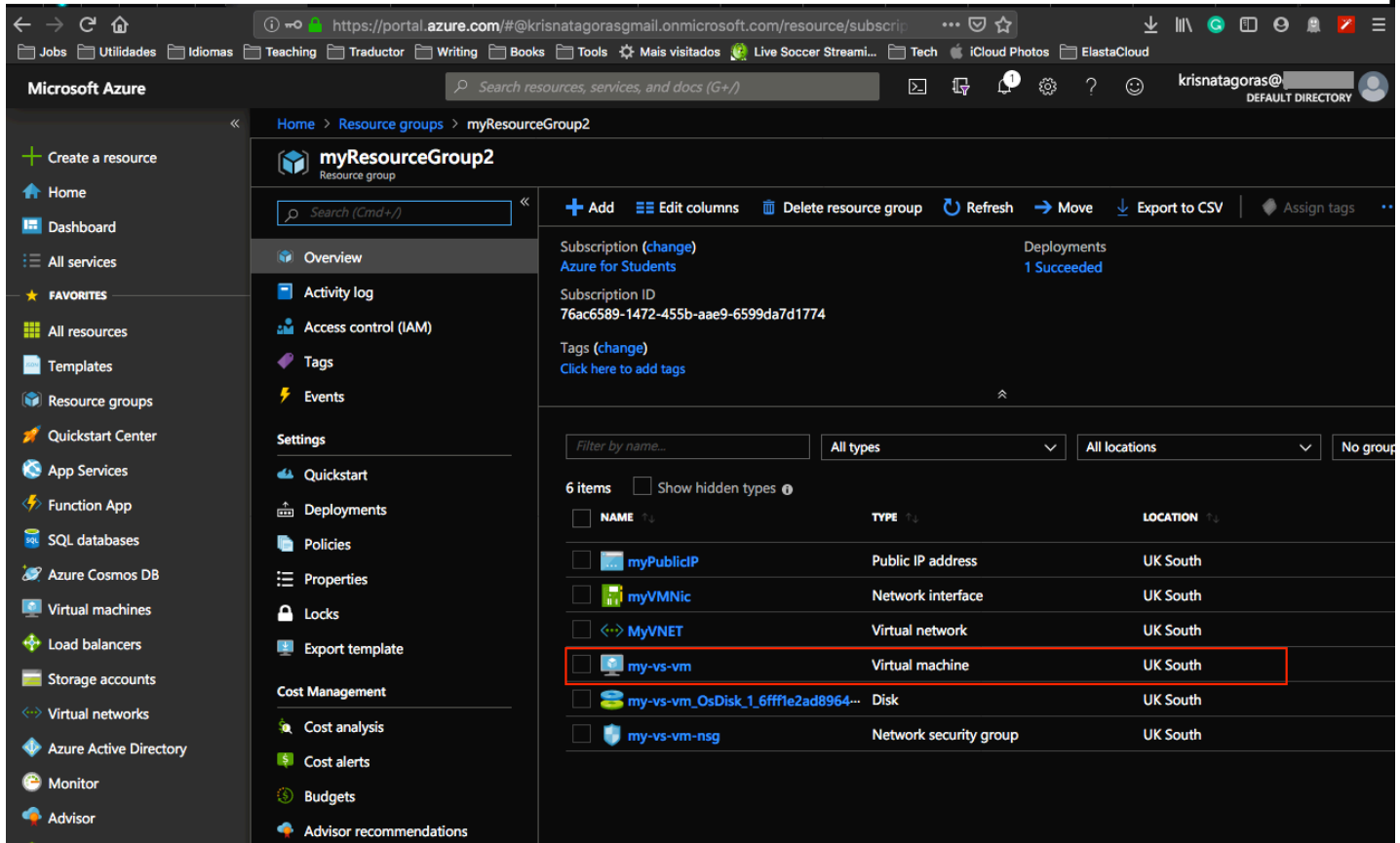
```
PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL
1: bash
{"id": "/subscriptions/[REDACTED]/resourceGroups/myResourceGroup2",
"location": "uksouth",
"managedBy": null,
"name": "myResourceGroup2",
"properties": {
  "provisioningState": "Succeeded"
},
"tags": null,
"type": "Microsoft.Resources/resourceGroups"
}
Osiris-2:101-vm-windows-vvisualstudio2019 krisnatagoras$ az group deployment create --name mydep --resource-group "myResourceGroup2" --template-file "./azuredeploy.json"
Please provide string value for 'adminUsername' (? for help): karaujo
Please provide securestring value for 'adminPassword' (? for help):
Running ..
```

As you can see, it's running. Go grab a cup of coffee, have some fresh air and I'm sure that before you come back you gonna have your Windows Virtual Machine with Visual Studio ready.

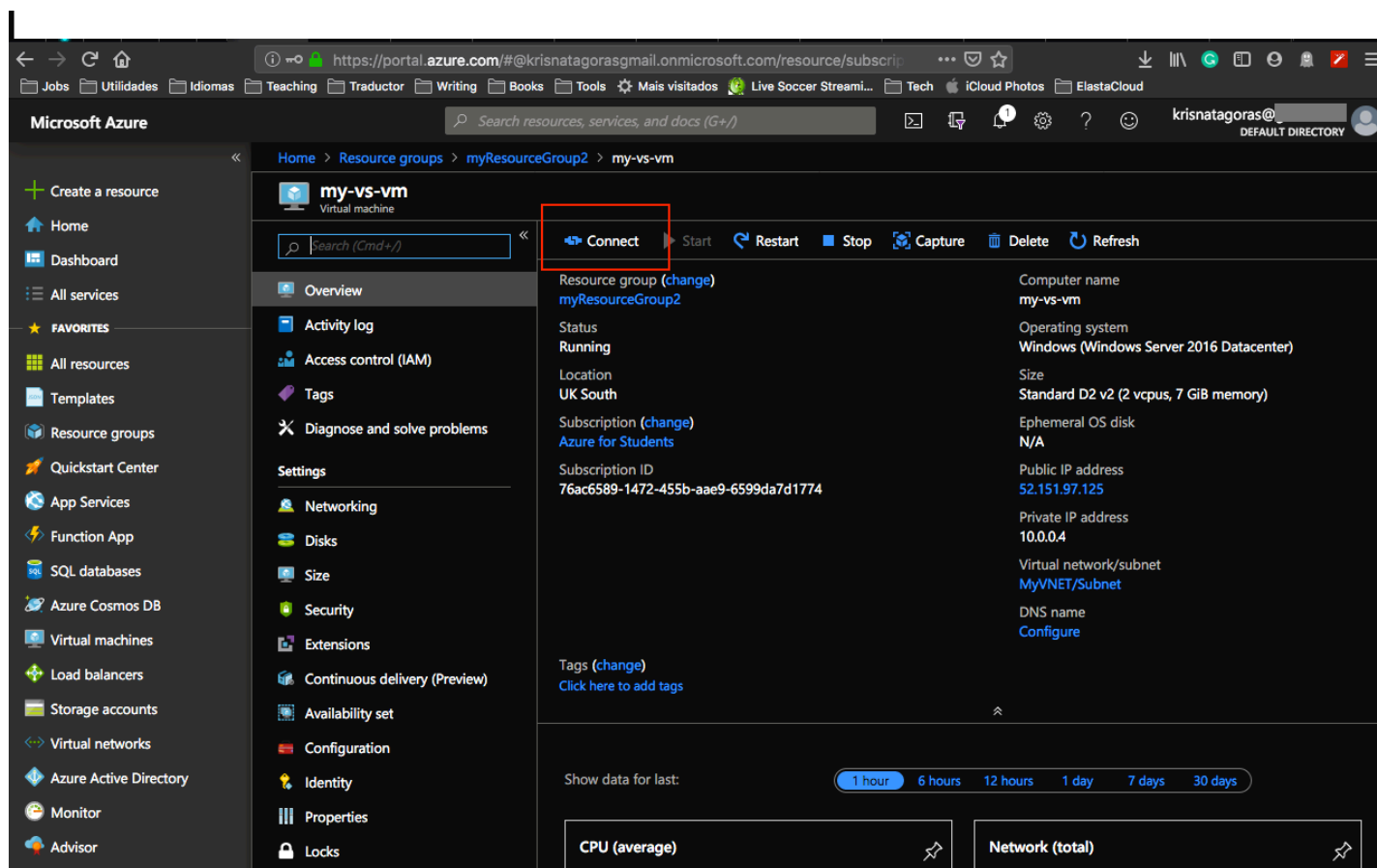
And there we go, our deploy is Succeeded:



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**:



Open your Virtual Machine and then click on the button **connect**.



Download the RDP File (you gonna need an RDP software to connect), double click on the file, insert your **login** and **password**.

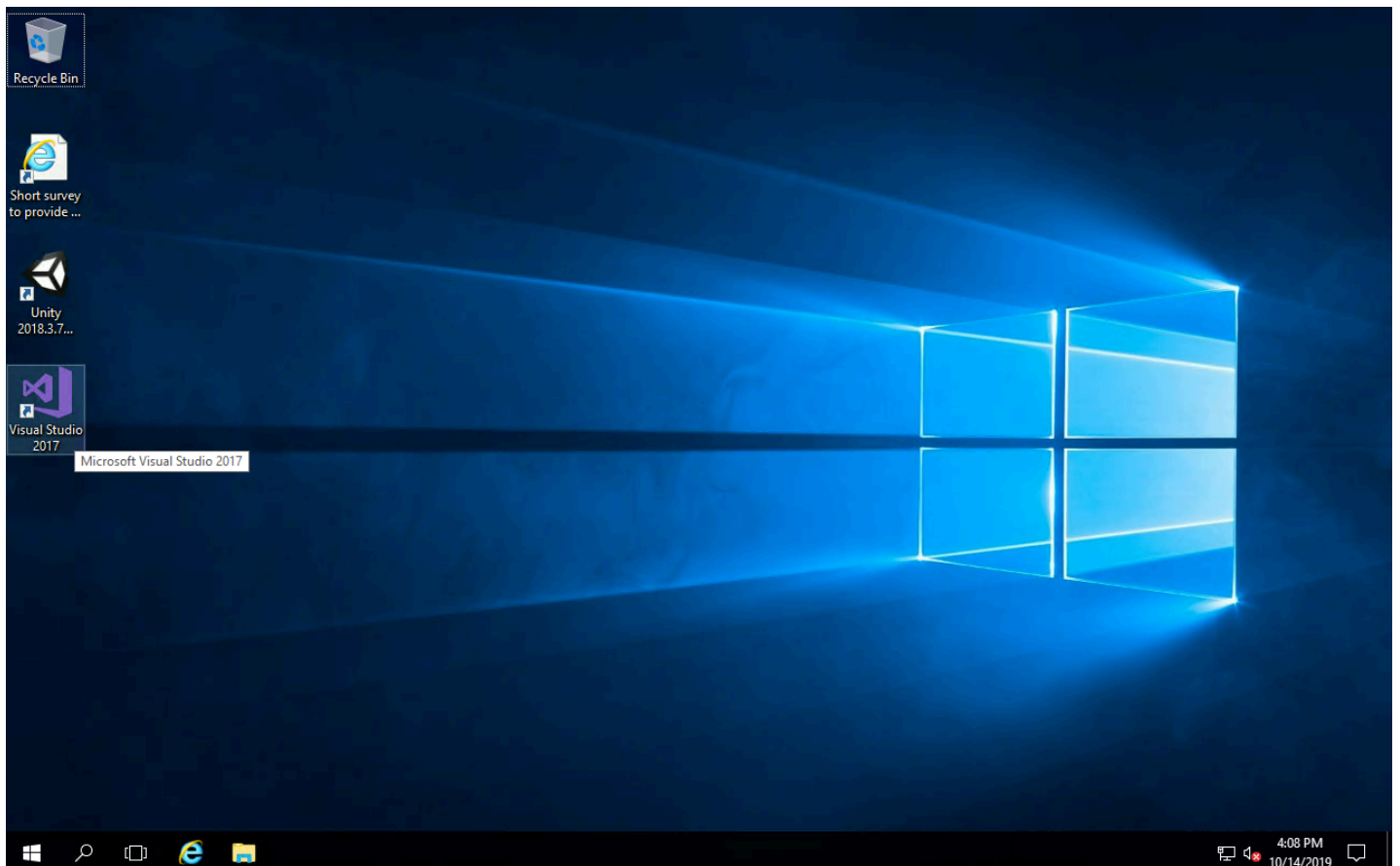
Invalid login credentials. Please re-enter.

User name

Password

Negotiating Credentials ...

And Voilà, there you have a brandy new Windows Virtual Machine with Visual Studio.



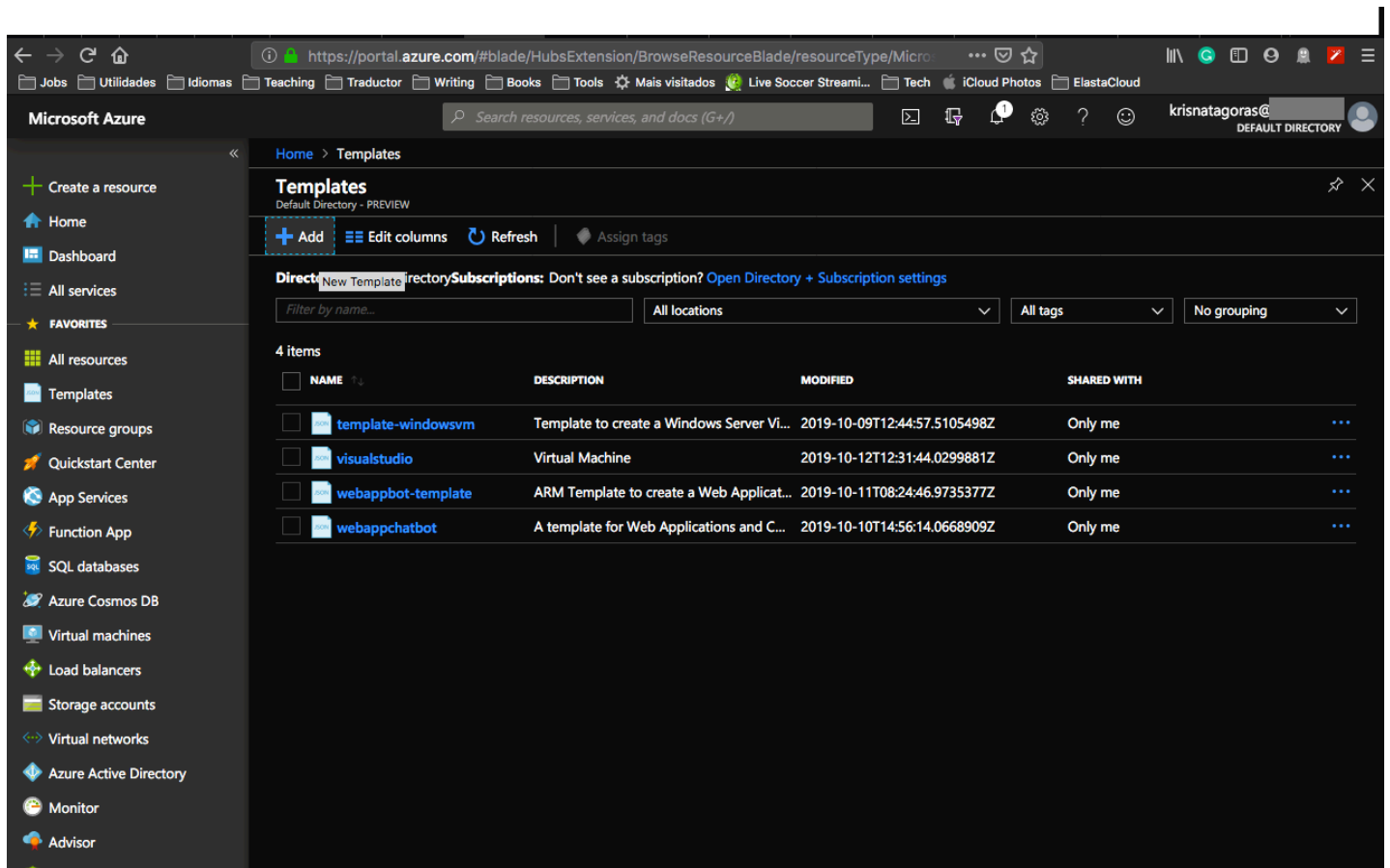
Have fun coding!

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.

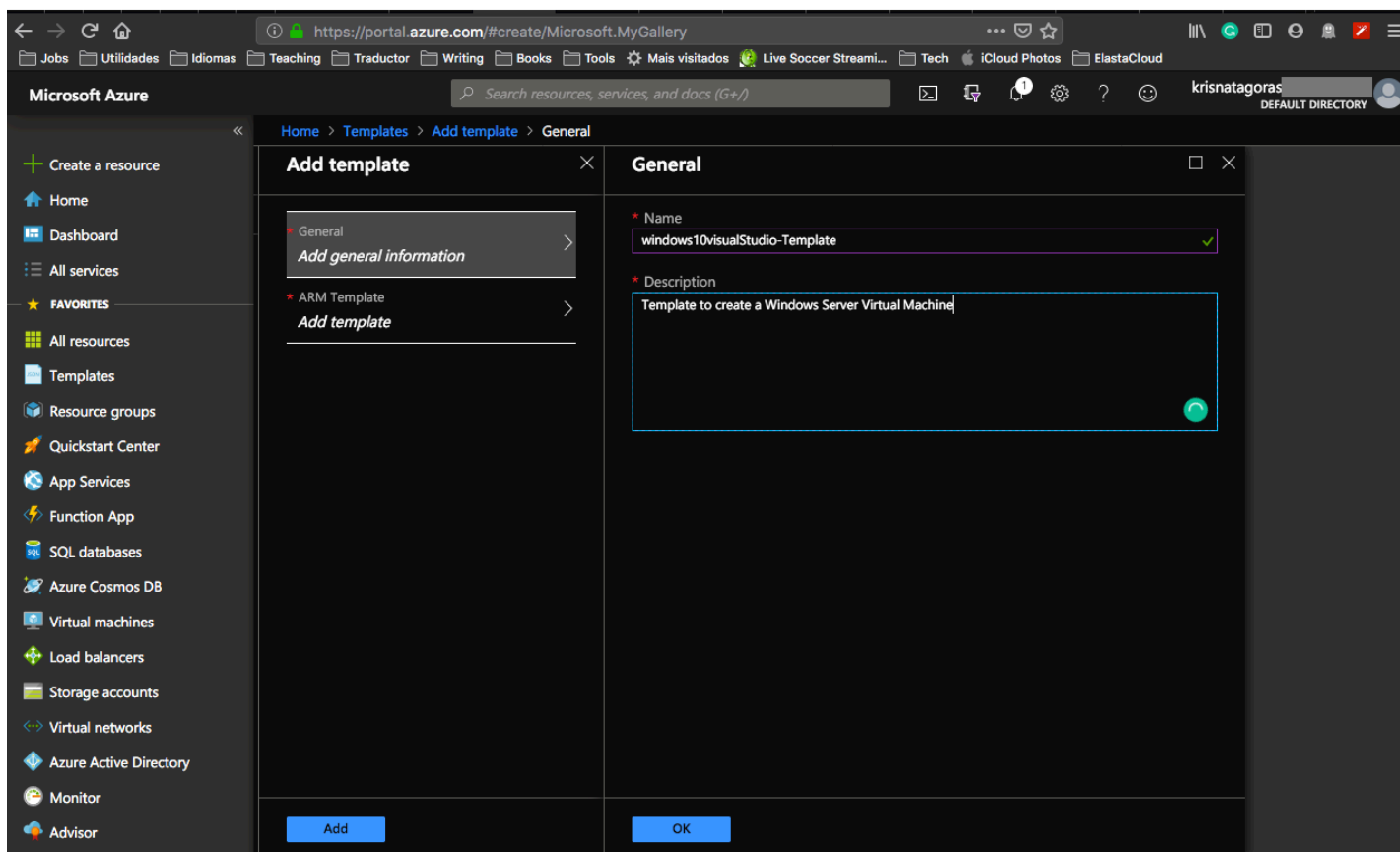
Using the Portal

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

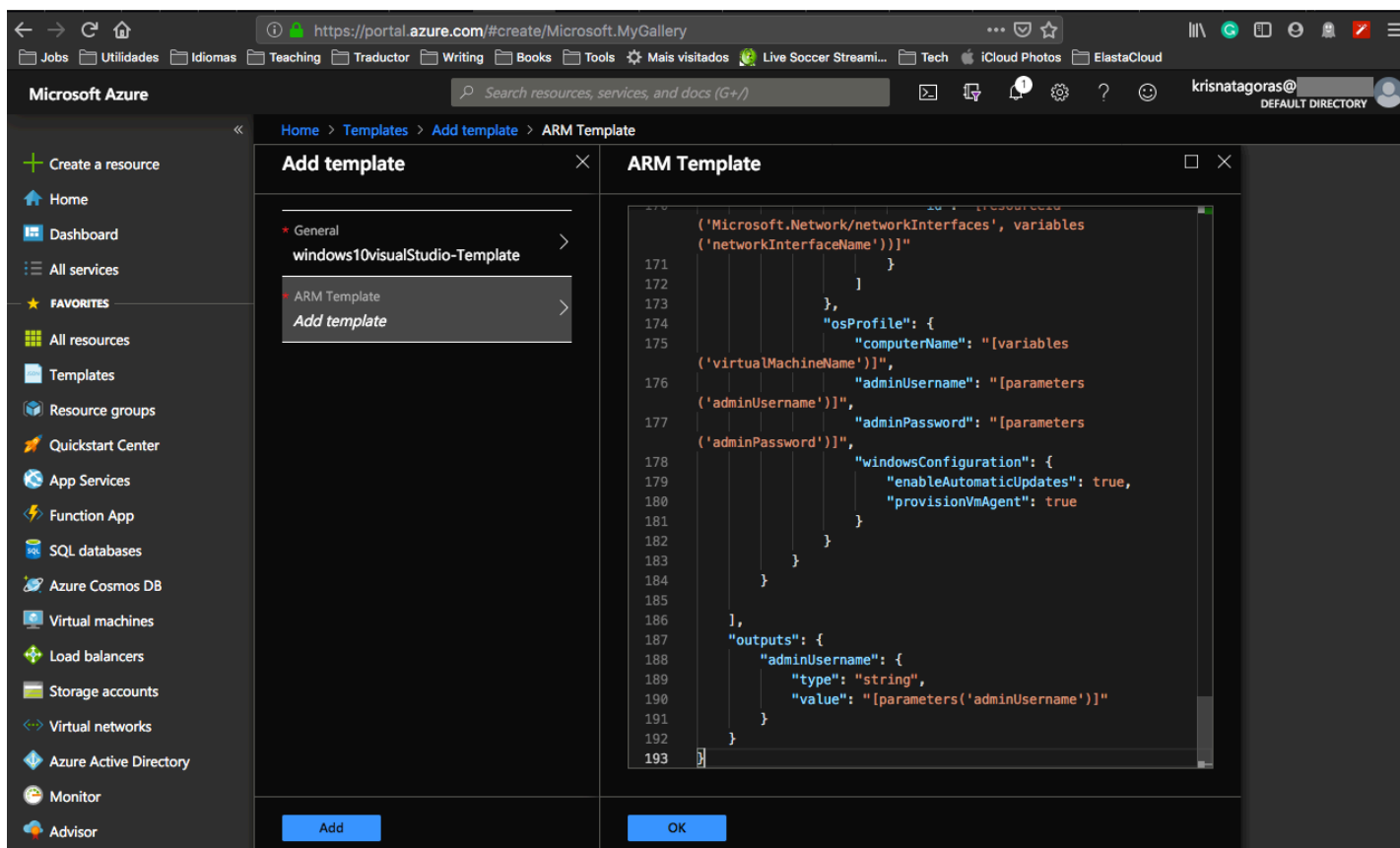
Click in **Add** to add your template:



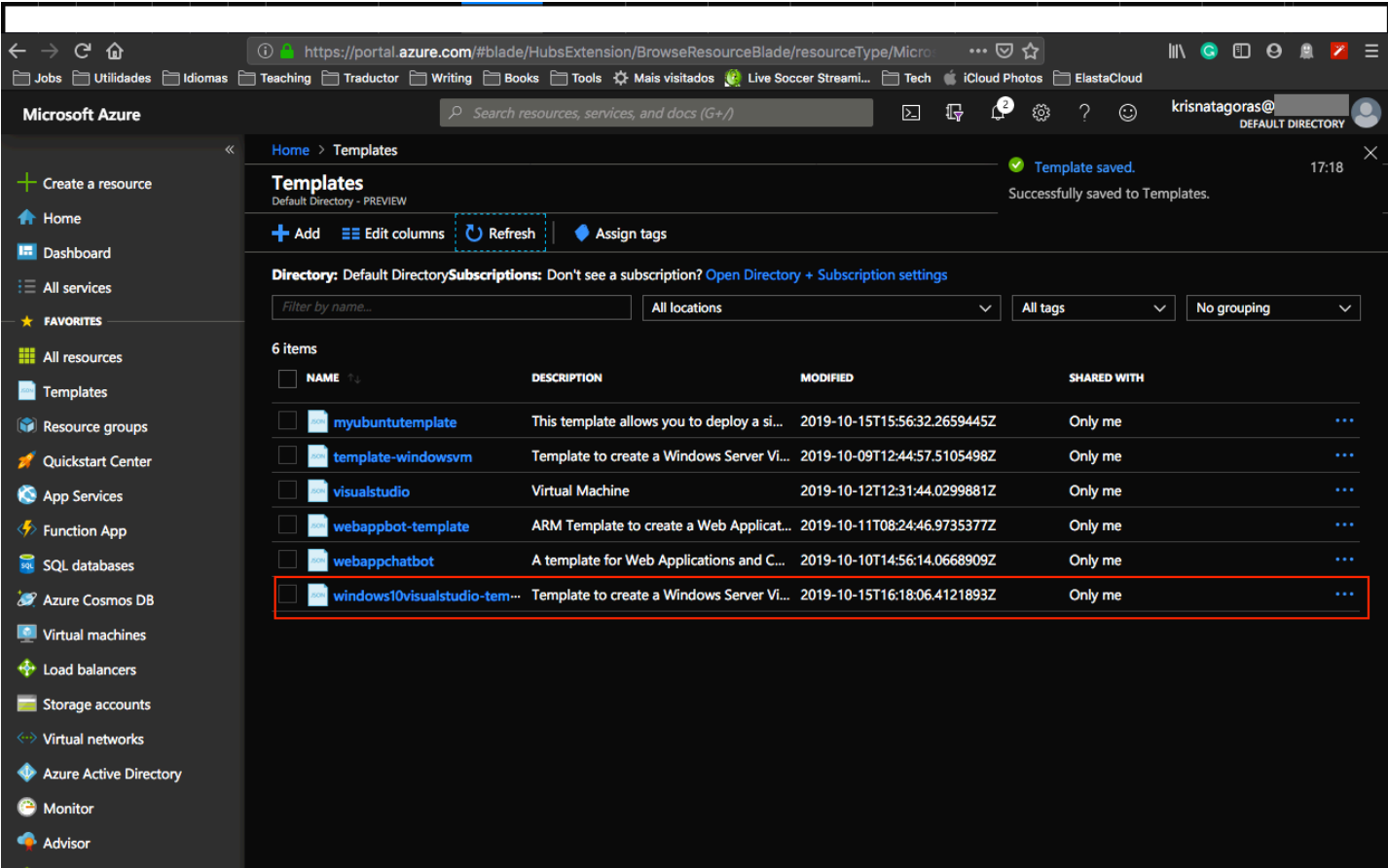
On General, type a name and a description for your template, and click on [OK].



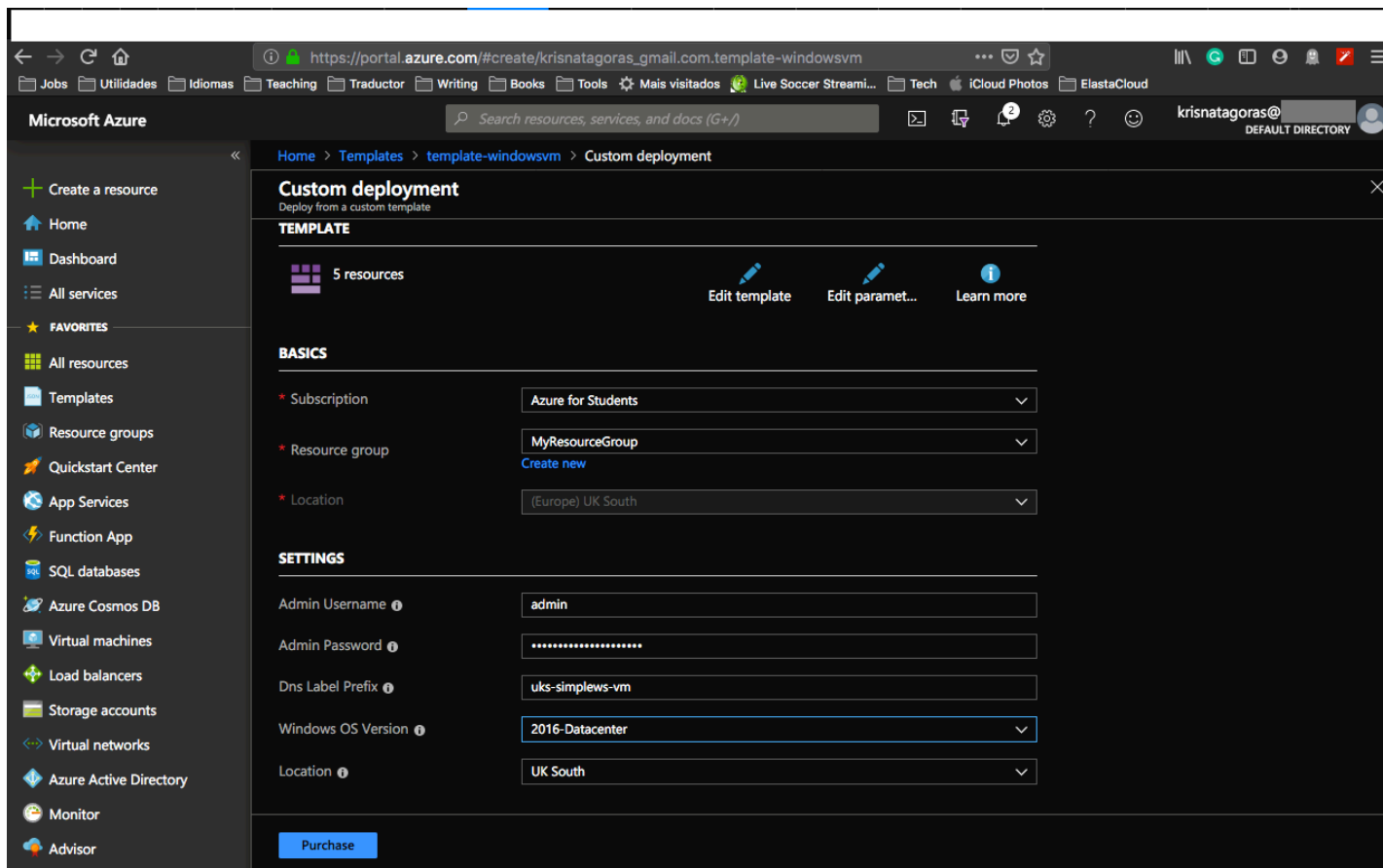
On ARM Template, replace the contents of the template with your template, and click on [OK].



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]:

Insert all the information, by now you should be familiar with it, select [I agree] and click on [Purchase].

And voilà, you have your new VM deployed. How easy was that, uhn?

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

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

p.s.: If by any chance you felt a bit overloaded with all these processes or perhaps you are asking yourself if there is a simple way to deploy your Virtual Machine? Good news for you bud! Just click on the button below and it will automatically deploy the VM on your Azure Portal.



Now that you have done the hard work, with the Portal is even easier to create our Windows Virtual Machine with Visual Studio.

Just click on this button: **Deploy to Azure**

Insert your credentials to log in to the Portal.

Create the group if you haven't yet, select the location and insert the **admin user** and **password**:

Select **I agree..** and then click in **Purchase**.

The screenshot shows the Microsoft Azure portal interface for a 'Custom deployment'. The left sidebar contains navigation links like 'Create a resource', 'Home', 'Dashboard', 'All services', and 'FAVORITES'. The main content area is titled 'Custom deployment' with the subtitle 'Deploy from a custom template'. It features three dropdown menus: 'Subscription' (Azure for Students), 'Resource group' ((New) myResourceGroup3), and 'Location' ((Europe) UK South). Below these are the 'SETTINGS' section with 'Admin Username' (karaujo) and 'Admin Password' (masked), both marked with green checkmarks. The 'TERMS AND CONDITIONS' section includes a link to 'Azure Marketplace Terms' and a checkbox for 'I agree to the terms and conditions stated above'. A 'Purchase' button is located at the bottom of the form.

And voilà, you have your new VM deployed. How easy was that, uhn?

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

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

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.

The screenshot shows the Microsoft Azure portal interface. The left sidebar contains navigation options like 'Create a resource', 'Home', 'Dashboard', 'All services', and 'FAVORITES'. The main content area displays the 'myResourceGroup2' resource group. At the top of this section, there are buttons for '+ Add', 'Edit columns', 'Delete resource group' (highlighted with a red box), 'Refresh', 'Move', and 'Export to CSV'. Below these buttons, there is a summary of the resource group, including the subscription name 'Azure for Students', the subscription ID '76ac6589-1472-455b-aae9-6599da7d1774', and the number of deployments '1 Succeeded'. A table lists the resources within the group:

Name	Type	Location
myNetInt	Network interface	UK South
myPublicIP	Public IP address	UK South
mySecGroup	Network security group	UK South
MyVNET	Virtual network	UK South
my-wsvs-vm	Virtual machine	UK South
my-wsvs-vm_OsDisk_1_c819e0079...	Disk	UK South

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

The screenshot shows the Microsoft Azure portal interface for a specific virtual machine named 'my-wsvs-vm'. The left sidebar is the same as in the previous image. The main content area displays the 'my-wsvs-vm' virtual machine. At the top of this section, there are buttons for 'Connect', 'Start', 'Restart', 'Stop' (highlighted with a red box), 'Capture', 'Delete', and 'Refresh'. Below these buttons, there is a summary of the virtual machine, including the resource group 'myResourceGroup2', the status 'Running', the location 'UK South', the subscription name 'Azure for Students', and the subscription ID '76ac6589-1472-455b-aae9-6599da7d1774'. A table lists the virtual machine's properties:

Property	Value
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	-

Just refresh your screen and you are good to go.