## Setting up Azure

### Step 1: Create a service principal:

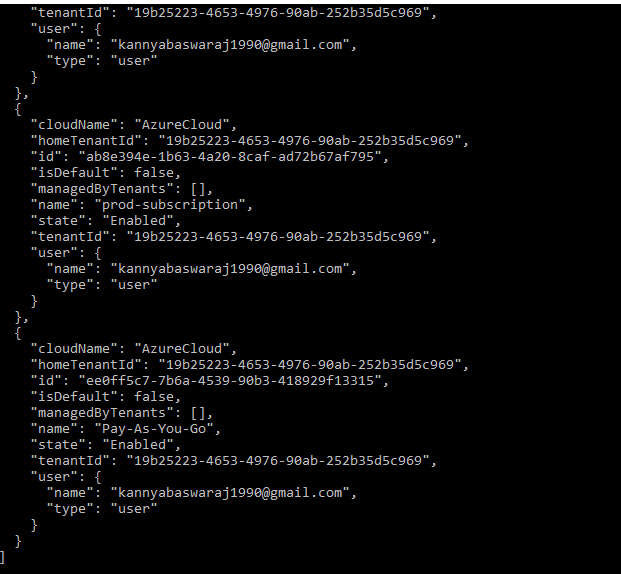
You can create a service principle in two ways. Either using the Azure CLI or using the portal.

#### Option 1: Creating using Azure CLI

1. Go to <https://docs.microsoft.com/en-us/cli/azure/install-azure-cli>
2. Download and install the installer file based on your operating system
3. Open the command prompt or terminal and type ***az login --use-device-code***

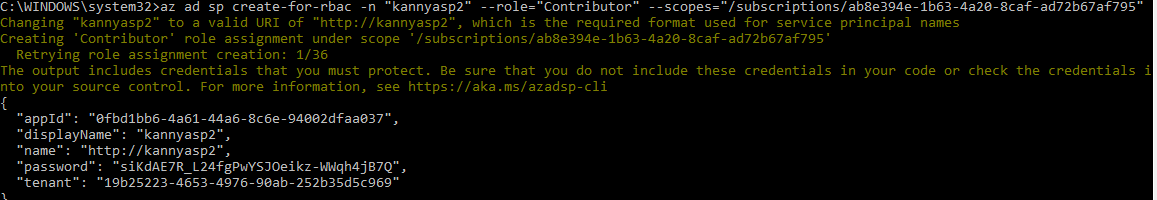
******

1. Navigate to the url <https://microsoft.com/devicelogin> and enter the code which is shown in the console.
2. After successful authentication, Close the browser window and come back to the command prompt.
3. If you get the list of active subscriptions like below image, Authentication got successful.



1. To create the service principal, Run the below command with your service principal name and subscription id.

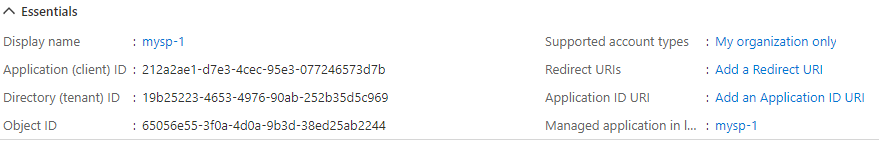
***#az ad sp create-for-rbac -n "<serviceprincipal-name>" --role="Contributor" --scopes="/subscriptions/<subscription-id>***

**

This would create a service principal using the Azure CLI.

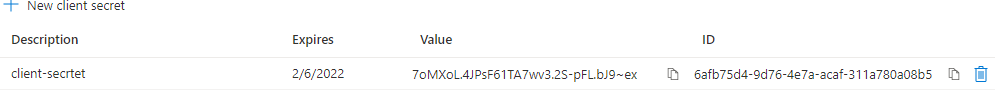
#### Option 2: Creating service principal using Azure portal:

1. Goto <https://portal.azure.com/>
2. Search for “Active Directory”.
3. Navigate to “App registration” in the left side bar menu.
4. Click on “New registration”
5. Give the display name for the service principal.
6. Click on register.



This would create a service principal, Note down the client id and tenant id.

Now, Goto “*Certificates & secrets”* in the left hand side menu and click on *new client secret*, Enter the description and click on add.



Note down the Value here as we will be using it while creating our terraform script.

### 

### Step 2: Download and initialize terraform

1. Go to <https://www.terraform.io/downloads.html>
2. Download and extract the terraform zip file depending upon the os type.
3. Update the terraform directory path in the system path environment.
4. Run below commands in sequence.

***#terraform --version*** (To check the terraform version)

**Note:** If you are using the Azure CLI, You don’t have to follow the above steps as Azure CLI comes preinstalled with terraform. If you want to install terraform on a local pc you follow the above step2.

### Step 3: Creating terraform configuration files to provisioning a linux vms in azure

Create below 4 files.

1. ***“terraform.tfvars”***

*resource\_group\_name = "sk\_terraform\_rg"*

*resource\_group\_location = "West Europe"*

*virtual\_network\_name = "vnet38"*

*subnet\_name = "subnet38"*

*public\_ip\_name = "publicip38"*

*network\_security\_group\_name = "nsg38"*

*network\_interface\_name = "nic38"*

*linux\_virtual\_machine\_name = "linuxvm38"*

*subscription\_id = "ab8e394e-1b63-4a20-8caf-ad72b67af795"*

*client\_id = "b1c046a3-ee83-490a-826a-731488f834db"*

*client\_secret = "~3yi-Y8N2teK..z\_g-Q\_~y0e9zx2UhNSui"*

*tenant\_id = "19b25223-4653-4976-90ab-252b35d5c969"*

Update the *subscription\_id, client\_id, client\_secret, tenant\_id*, with the values you got while creating the service principal and client secret.

You can also update the values of other variables such as linux machine name etc..

1. ***“variables.tf”***

*variable "resource\_group\_name" {*

*type = string*

*description = "RG name in Azure"*

*}*

*variable "resource\_group\_location" {*

*type = string*

*description = "RG location in Azure"*

*}*

*variable "virtual\_network\_name" {*

*type = string*

*description = "VNET name in Azure"*

*}*

*variable "subnet\_name" {*

*type = string*

*description = "Subnet name in Azure"*

*}*

*variable "public\_ip\_name" {*

*type = string*

*description = "Public IP name in Azure"*

*}*

*variable "network\_security\_group\_name" {*

*type = string*

*description = "NSG name in Azure"*

*}*

*variable "network\_interface\_name" {*

*type = string*

*description = "NIC name in Azure"*

*}*

*variable "linux\_virtual\_machine\_name" {*

*type = string*

*description = "Linux VM name in Azure"*

*}*

*variable "subscription\_id" {*

*type = string*

*description = "subscription id"*

*}*

*variable "client\_id" {*

*type = string*

*description = "client id"*

*}*

*variable "client\_secret" {*

*type = string*

*description = "client secret"*

*}*

*variable "tenant\_id" {*

*type = string*

*description = "tenant id"*

*}*

1. **“*outputs.tf”***

*output "vm\_id" {*

*value = azurerm\_linux\_virtual\_machine.linuxvm.id*

*}*

*output "vm\_ip" {*

*value = azurerm\_linux\_virtual\_machine.linuxvm.public\_ip\_address*

*}*

*output "tls\_private\_key" {*

*value = tls\_private\_key.example\_ssh.private\_key\_pem*

*}*

*output "admin\_username" {*

*value = azurerm\_linux\_virtual\_machine.linuxvm.admin\_username*

*}*

1. **“main.tf”**

*provider "azurerm" {*

*subscription\_id = var.subscription\_id*

*client\_id = var.client\_id*

*client\_secret = var.client\_secret*

*tenant\_id = var.tenant\_id*

*features {}*

*}*

*# Create a resource group if it doesn't exist*

*resource "azurerm\_resource\_group" "rg" {*

*name = var.resource\_group\_name*

*location = var.resource\_group\_location*

*tags = {*

*environment = "production"*

*}*

*}*

*# Create virtual network*

*resource "azurerm\_virtual\_network" "vnet" {*

*name = var.virtual\_network\_name*

*address\_space = ["10.0.0.0/16"]*

*location = azurerm\_resource\_group.rg.location*

*resource\_group\_name = azurerm\_resource\_group.rg.name*

*tags = {*

*environment = "production"*

*}*

*}*

*# Create subnet*

*resource "azurerm\_subnet" "subnet" {*

*name = var.subnet\_name*

*resource\_group\_name = azurerm\_resource\_group.rg.name*

*virtual\_network\_name = azurerm\_virtual\_network.vnet.name*

*address\_prefixes = ["10.0.1.0/24"]*

*}*

*# Create public IPs*

*resource "azurerm\_public\_ip" "public\_ip" {*

*name = var.public\_ip\_name*

*location = azurerm\_resource\_group.rg.location*

*resource\_group\_name = azurerm\_resource\_group.rg.name*

*allocation\_method = "Dynamic"*

*tags = {*

*environment = "production"*

*}*

*}*

*# Create Network Security Group and rule*

*resource "azurerm\_network\_security\_group" "nsg" {*

*name = var.network\_security\_group\_name*

*location = azurerm\_resource\_group.rg.location*

*resource\_group\_name = azurerm\_resource\_group.rg.name*

*security\_rule {*

*name = "SSH"*

*priority = 1001*

*direction = "Inbound"*

*access = "Allow"*

*protocol = "Tcp"*

*source\_port\_range = "\*"*

*destination\_port\_range = "22"*

*source\_address\_prefix = "\*"*

*destination\_address\_prefix = "\*"*

*}*

*tags = {*

*environment = "production"*

*}*

*}*

*# Create network interface*

*resource "azurerm\_network\_interface" "nic" {*

*name = var.network\_interface\_name*

*location = azurerm\_resource\_group.rg.location*

*resource\_group\_name = azurerm\_resource\_group.rg.name*

*ip\_configuration {*

*name = "myNicConfiguration"*

*subnet\_id = azurerm\_subnet.subnet.id*

*private\_ip\_address\_allocation = "Dynamic"*

*public\_ip\_address\_id = azurerm\_public\_ip.public\_ip.id*

*}*

*tags = {*

*environment = "production"*

*}*

*}*

*# Connect the security group to the network interface*

*resource "azurerm\_network\_interface\_security\_group\_association" "association" {*

*network\_interface\_id = azurerm\_network\_interface.nic.id*

*network\_security\_group\_id = azurerm\_network\_security\_group.nsg.id*

*}*

*# Generate random text for a unique storage account name*

*resource "random\_id" "randomId" {*

*keepers = {*

*# Generate a new ID only when a new resource group is defined*

*resource\_group = azurerm\_resource\_group.rg.name*

*}*

*byte\_length = 8*

*}*

*# Create storage account for boot diagnostics*

*resource "azurerm\_storage\_account" "storage" {*

*name = "diag${random\_id.randomId.hex}"*

*resource\_group\_name = azurerm\_resource\_group.rg.name*

*location = azurerm\_resource\_group.rg.location*

*account\_tier = "Standard"*

*account\_replication\_type = "LRS"*

*tags = {*

*environment = "production"*

*}*

*}*

*# Create (and display) an SSH key*

*resource "tls\_private\_key" "example\_ssh" {*

*algorithm = "RSA"*

*rsa\_bits = 4096*

*}*

*# Create virtual machine*

*resource "azurerm\_linux\_virtual\_machine" "linuxvm" {*

*name = var.linux\_virtual\_machine\_name*

*location = azurerm\_resource\_group.rg.location*

*resource\_group\_name = azurerm\_resource\_group.rg.name*

*network\_interface\_ids = [azurerm\_network\_interface.nic.id]*

*size = "Standard\_DS1\_v2"*

*os\_disk {*

*name = "myOsDisk"*

*caching = "ReadWrite"*

*storage\_account\_type = "Premium\_LRS"*

*}*

*source\_image\_reference {*

*publisher = "Canonical"*

*offer = "UbuntuServer"*

*sku = "18.04-LTS"*

*version = "latest"*

*}*

*computer\_name = "myvm"*

*admin\_username = "azureuser"*

*disable\_password\_authentication = true*

*admin\_ssh\_key {*

*username = "azureuser"*

*public\_key = tls\_private\_key.example\_ssh.public\_key\_openssh*

*}*

*boot\_diagnostics {*

*storage\_account\_uri = azurerm\_storage\_account.storage.primary\_blob\_endpoint*

*}*

*tags = {*

*environment = "production"*

*}*

*}*

*resource "local\_file" "cloud\_pem" {*

*filename = "${path.module}/privatekey.pem"*

*content = tls\_private\_key.example\_ssh.private\_key\_pem*

*}*

Update the client\_id with the app\_id which you got in the above output. Similarly, Update client\_secret and tenant\_id with password and tenant values respectively.

Get the subscription id from the azure portal and update it in the below file as well.

### Step 4: Execution:

Navigate to the directory where you have placed all the files and run below commands in sequence to provision the infrastructure in azure

***#terraform init*** (initializing terraform)

***#terraform validate*** (validate the code)

***#terraform plan*** (performs dry run before provisioning)

***#terraform apply*** (provision the infrastructure on azure)

After running the above commands vms is deployed successfully and one pem file with name “privatekey.pem” is generated in the same directory.

### Step 5: Accessing the VMS

#### Using the Azure CLI

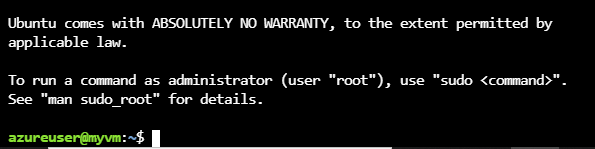
1. First, Change the permissions of the private key file to make it read only using the below command.

***#chmod 400 privatekey.pem***

1. Run the below command to access the vms using the SSH and enter “yes” when prompted

***#ssh -i privatekey.pem azureuser@publicip***

**Note:** You will get the public ip of the vms after running the *terraform apply* command

******

You would be connected to the linux vms which you have just created.