

# Azure CLI Installation and Service Principal Creation

Azure CLI enables to manage resources in Azure from your PC.

## Install Azure CLI on Windows

Open the link <https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows?tabs=azure-cli>

## Install or update

The MSI distributable is used for installing or updating the Azure CLI on Windows. You don't need to uninstall current versions before using the MSI installer because the MSI will update any existing version.

Microsoft Installer (MSI)

Microsoft Installer (MSI) with Command

When the installer asks if it can make changes to your computer, click the "Yes" box.

## Azure CLI current version

Download and install the current release of the Azure CLI.

**Current release of the Azure CLI**

Run the downloaded MSI:

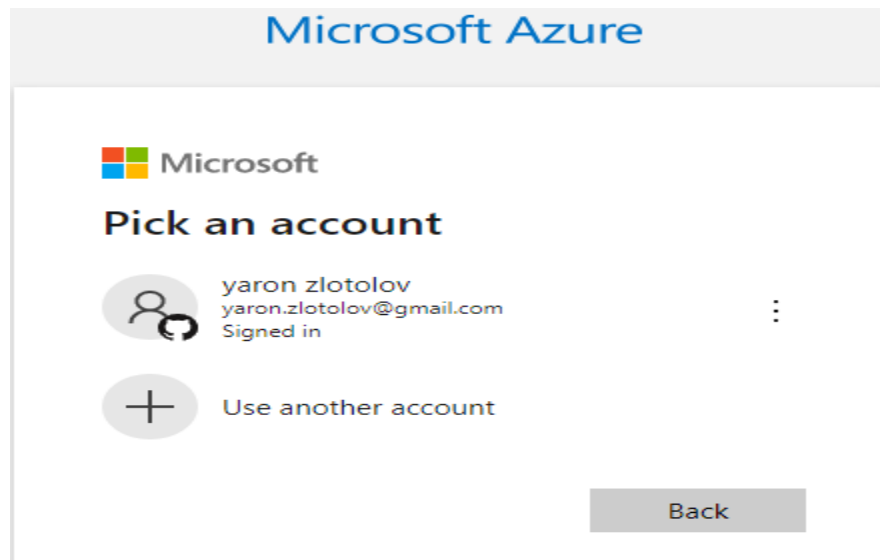


You can also install the Azure CLI using PowerShell. Start PowerShell as administrator and run the following command:

```
Invoke-WebRequest -Uri https://aka.ms/installazurecliwindows -OutFile .\AzureCLI.msi; Start-Process msixexec.exe -Wait -ArgumentList '/I AzureCLI.msi /quiet'; rm .\AzureCLI.msi
```

Login to Azure CLI:

Open VS Code and run **az login** and select your account as below.



**You have logged into Microsoft Azure!**

You can close this window, or we will redirect you to the [Azure CLI documents](#) in 10 seconds.

List the Subscriptions associated with the account and get the subscription id:

**az account list -o table**

```
PS C:\Kubernetes\Azure-AKS> az account list -o table
A few accounts are skipped as they don't have 'Enabled' state. Use '--all' to display them.
Name                        CloudName  SubscriptionId                State  IsDefault
-----
Azure subscription 1       AzureCloud  000000-0000-0000-0000-000000000000  Enabled  True
PS C:\Kubernetes\Azure-AKS>
```

Keep the subscription in variable for later use:

```
$SUBSCRIPTION="000000-0000-0000-0000-000000000000"
```

In case we have subscription for testing and subscription for production we need to specify which subscription we are using via the following command so we don't accidentally change things in production subscription.

```
az account set --subscription=$SUBSCRIPTION
```

### Install JQ tool:

JQ is a command-line tool for parsing JSON. Download from - <https://stedolan.github.io/jq/download/>

## Windows

- Use [Chocolatey NuGet](#) to install jq 1.5 with `chocolatey install jq`.
- jq 1.6 executables for [64-bit](#) or 32-bit.
- jq 1.5 executables for 64-bit or 32-bit.
- jq 1.4 executables for 64-bit or 32-bit.
- jq 1.3 executables for 64-bit or 32-bit.

rename jq-win64.exe to jq.exe and copy the file to Terraform folder.

### Create Service Principal

Service principal with Contributor permission allows terraform the manage infrastructure over the Azure subscription.

Create service principal:

**\$SERVICE\_PRINCIPAL\_JSON=(az ad sp create-for-rbac --skip-assignment --name terraform-sp -o json)**

```
PS C:\Kubernetes\Azure-AKS\Terraform> $SERVICE_PRINCIPAL_JSON=(az ad sp create-for-rbac --skip-assignment --name terraform-sp -o json)
WARNING: Changing "terraform-sp" to a valid URI of "http://terraform-sp", which is the required format used for service principal names
WARNING: The output includes credentials that you must protect. Be sure that you do not include these credentials in your code or check the credentials into your source control. For more information, see https://aka.ms/azadsp-cli
```

**echo \$SERVICE\_PRINCIPAL\_JSON**

```
{
  "appId": "00000000-0000-0000-0000-000000000000",
  "displayName": "azure-cli-2021-06-05-10-41-15",
  "name": "http://azure-cli-2021-06-05-10-41-15",
  "password": "0000-0000-0000-0000-000000000000",
  "tenant": "00000000-0000-0000-0000-000000000000"
}
```

The values that will be used to terraform are:

**appId** - is the **client\_id** defined above.

**password** - is the **client\_secret** defined above.

**tenant** - is the **tenant\_id** defined above.

Keep the 'appId', 'password' and 'tenant' in variables for later use:

```
$SERVICE_PRINCIPAL=(echo $SERVICE_PRINCIPAL_JSON | jq -r '.appId')
```

```
$SERVICE_PRINCIPAL_SECRET=(echo $SERVICE_PRINCIPAL_JSON | jq -r '.password')
```

```
$TENANT_ID=(echo $SERVICE_PRINCIPAL_JSON | jq -r '.tenant')
```

NOTE: reset the credential if you have any single or double quote on password:

```
echo $SERVICE_PRINCIPAL_SECRET
```

```
az ad sp credential reset --name "terraform-sp"
```

Grant contributor role over the subscription to our service principal:

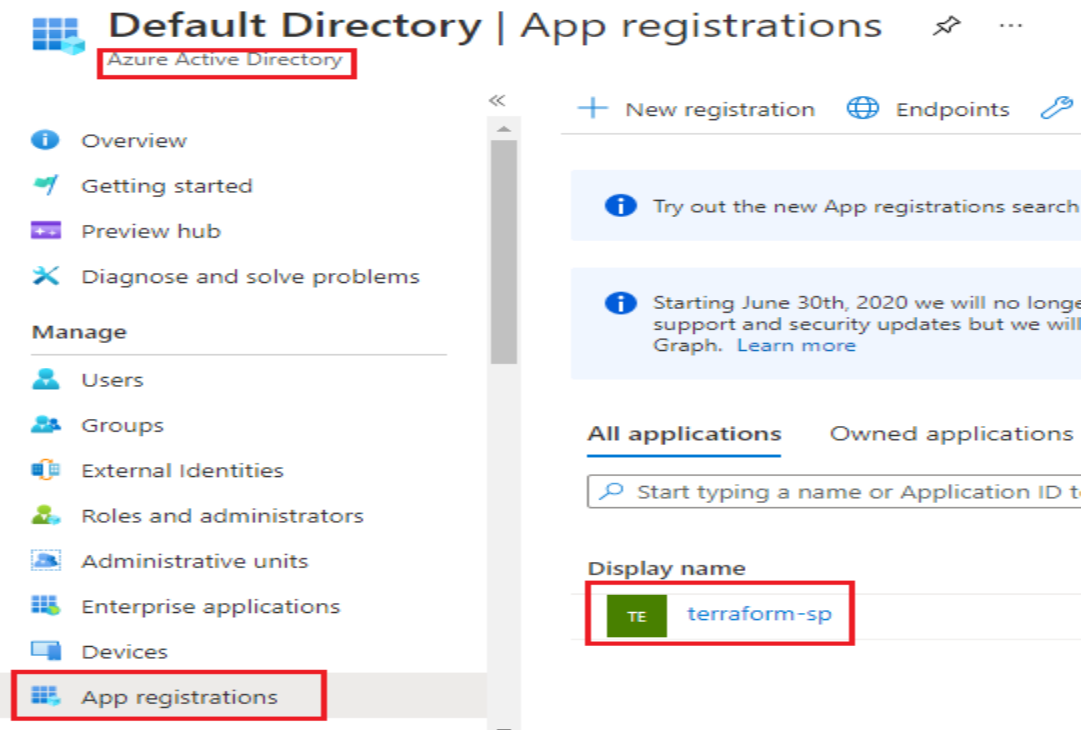
```
az role assignment create --assignee $SERVICE_PRINCIPAL --scope "/subscriptions/$SUBSCRIPTION" --role Contributor
```

```
PS C:\Kubernetes\Azure-AKS\Terraform> az role assignment create --assignee $SERVICE_PRINCIPAL --scope "/subscriptions/$SUBSCRIPTION" --role Contributor
```

Now terraform has permission to manage infrastructure on the Azure subscription!

### Review the Service Principle in Azure Portal

Home -> Azure Active Directory -> App registrations > View All Application in directory.



### Configuring the Service Principal in Terraform

Storing the credentials as Environment Variables so there is no need to do **az login** for terraform:

**setx ARM\_CLIENT\_ID <Application (client) ID>**

**setx ARM\_SUBSCRIPTION\_ID <subsciprion>**

**setx ARM\_TENANT\_ID <Directory (tenant) ID>**

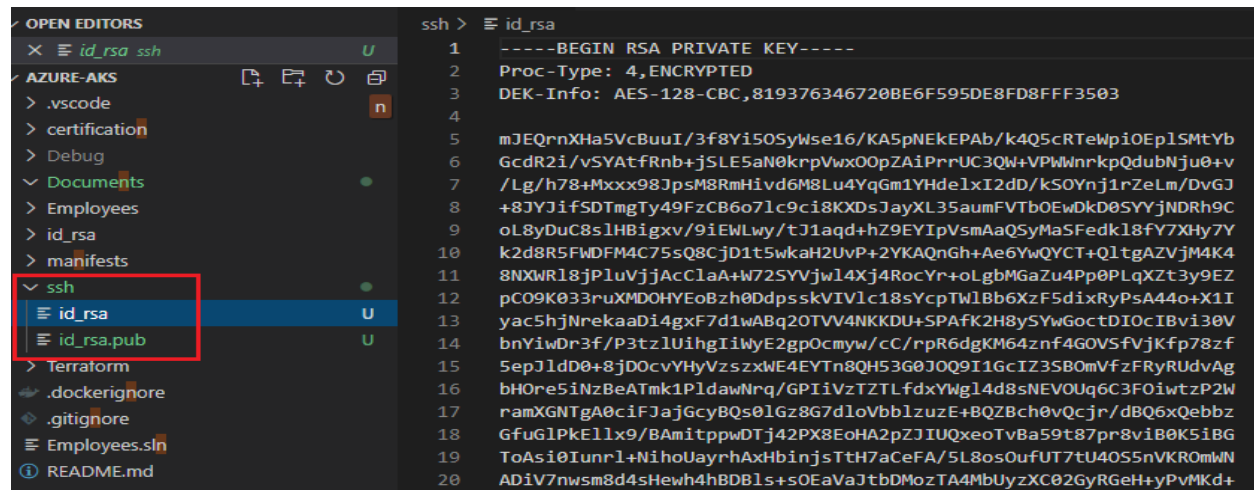
**setx ARM\_CLIENT\_SECRET <password>**

### Generate SSH key

Later we will need SSH key for connecting to Kubernetes cluster for investigation and troubleshoot.

**ssh-keygen -t rsa -b 4096 -N "VeryStrongSecret123!" -C "your\_email@example.com" -q -f .\ssh\id\_rsa**

```
PS C:\Kubernetes\Azure-AKS> ssh-keygen -t rsa -b 4096 -N "VeryStrongSecret123!" -C "yaron.zlotolov@gmail.com" -q -f .\ssh\id_rsa
PS C:\Kubernetes\Azure-AKS>
```



The screenshot shows the Visual Studio Code interface. On the left, the 'EXPLORER' sidebar is open, showing the file structure of the project. The 'ssh' folder is expanded, and the files 'id\_rsa' and 'id\_rsa.pub' are highlighted. On the right, the 'TERMINAL' pane shows the output of the 'ssh-keygen' command. The output indicates that a new RSA private key has been generated and encrypted, with a passphrase of 'VeryStrongSecret123!'. The key is saved to the file '.\ssh\id\_rsa'.

```
ssh > id_rsa
1  -----BEGIN RSA PRIVATE KEY-----
2  Proc-Type: 4, ENCRYPTED
3  DEK-Info: AES-128-CBC,819376346720BE6F595DE8FD8FFF3503
4
5  mJEQrnXHs5VcBuuI/3f8Yi50SyWse16/KA5pNEkEPAb/k4Q5cRTeWpi0Ep1SMtYb
6  GcdR2i/vSYatfRnb+jSLE5aN0krpVwx0OpZAIprrUC3QW+VPWWnrkpQdubNju0+v
7  /Lg/h78+Mxxx98JpsM8RmHivd6M8Lu4YqGm1YHdelxI2dD/kSOYnj1rZeLm/DvGJ
8  +8JYJifSDTmgTy49FzCB6o7lc9ci8KXD5JayXL35aumFVTbOEwDkD0SYyjNDRh9C
9  oL8yDuC8slHBixv/9iEWLwy/tJ1aqd+hZ9EYIpVsmAaQSYMaSFedk18fY7XH7Y
10 k2d8R5FwDFM4C75sQ8CjD1t5wkaH2UvP+2YKAQnGh+Ae6YwQYCT+Q1tgAZVjM4K4
11 8NXWRl8jPluvjjAcClAa+W72SYVjw14Xj4RocYr+oLgbMGaZu4Pp0PLqXZt3y9EZ
12 pC09K033ruXMD0HYEoBzh0DdpskVIV1c18sYcpTW1Bb6XzF5dixRyPsA44o+X1I
13 yac5hjNrekaaDi4gxF7d1wABq20TVV4NKKDU+SPAFK2H8ySYwGoctDIOcIBvi30V
14 bnYiwDr3f/P3tz1UiHgIiWyE2gp0cmYw/cC/rpR6dgKM64znf4G0VSfVjKfp78zf
15 5epJld0+8jD0cvYHyVzszxWE4EYtn8QH53G0J0Q9I1GcIZ3S80mVfzFRyRUDvAg
16 bH0re5iNzBeATmk1PldawNrQ/GPIiVzTZTLfdxYwG14d8sNEVOUq6C3F0iwtzP2W
17 ramXGNTgA0ciF3ajGcyBQs0lGz8G7dloVbb1zuzE+BQZBch0vQcjr/dBQ6xQebbz
18 GfuGlPkEllx9/BAmittppwDTj42PX8EoHA2pZJIUQxeoTvBa59t87pr8v1B0K5iBG
19 ToAsi0Iunr1+NihoUayrhAxHbinjsTtH7aCeFA/5L8os0ufUT7tU40S5nVKR0mWn
20 ADiV7nwsM8d4sHewh4hBDB1s+s0EaVaJtbdMozTA4MbUyzXC02GyRGeH+yPvMKd+
```

Done!