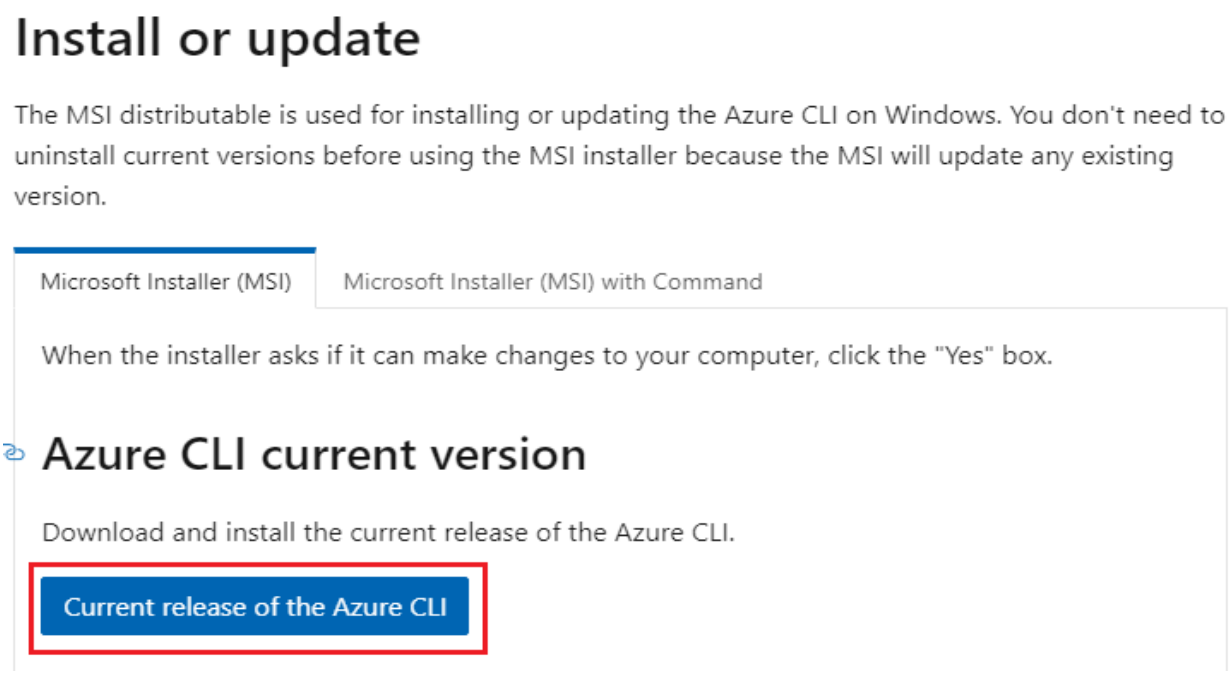
Azure CLI Installation and Service Principal Creation

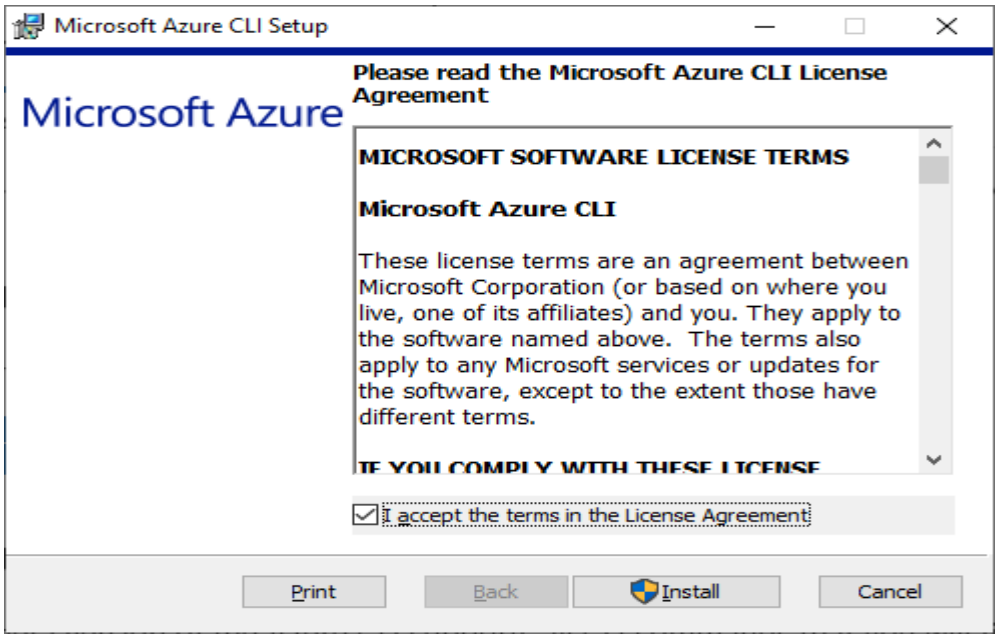
Azure CLI enables to manage resources in Azure from your PC.

Install Azure CLI on Windows

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



Run the downloaded MSI:

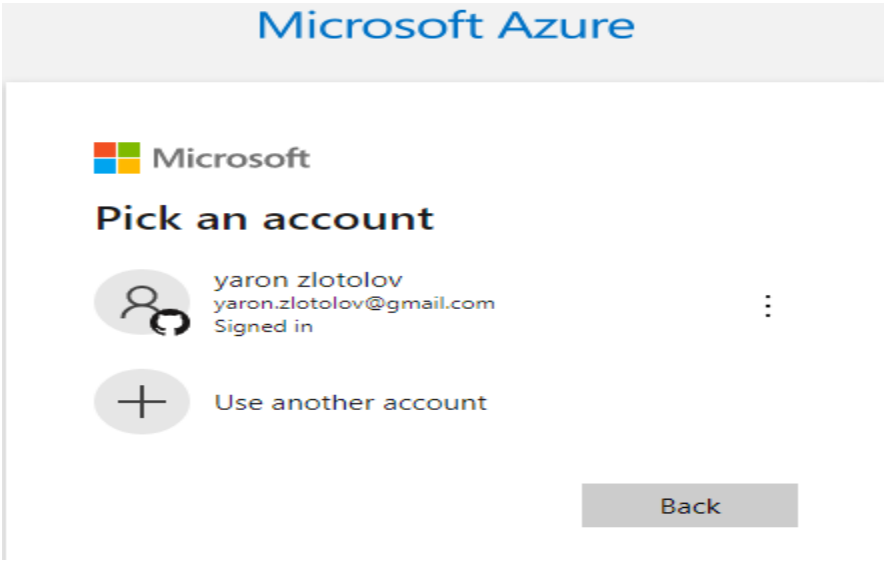


You can also install the Azure CLI using PowerShell. Start PowerShell as administrator and run the following command:

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

Login to Azure CLI:

Open VS Code and run **az login** and **s**elect 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:\Kuberneties\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  [redacted]9665c  Enabled  True
PS C:\Kuberneties\Azure-AKS>
```

Keep the subscription for later use:

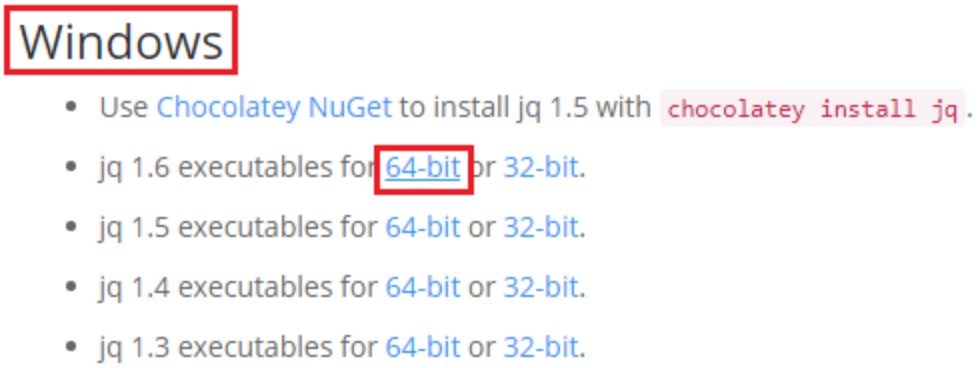
$SUBSCRIPTION="000000-0000-0000-0000-000000000"

In case we have subscriptions for testing and production we need to specify the subscription to use via the following command so we don’t accidentally change things to 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/



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

Create Service Principal

Service principal with contributor right allows terraform the manage infrastructure on the Azure subscription.

Create service principal:

$SERVICE_PRINCIPAL_JSON=(az ad sp create-for-rbac --skip-assignment --name terraform-sp -o json)

```
PS C:\Kuberneties\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 cre
entials 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’ 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:

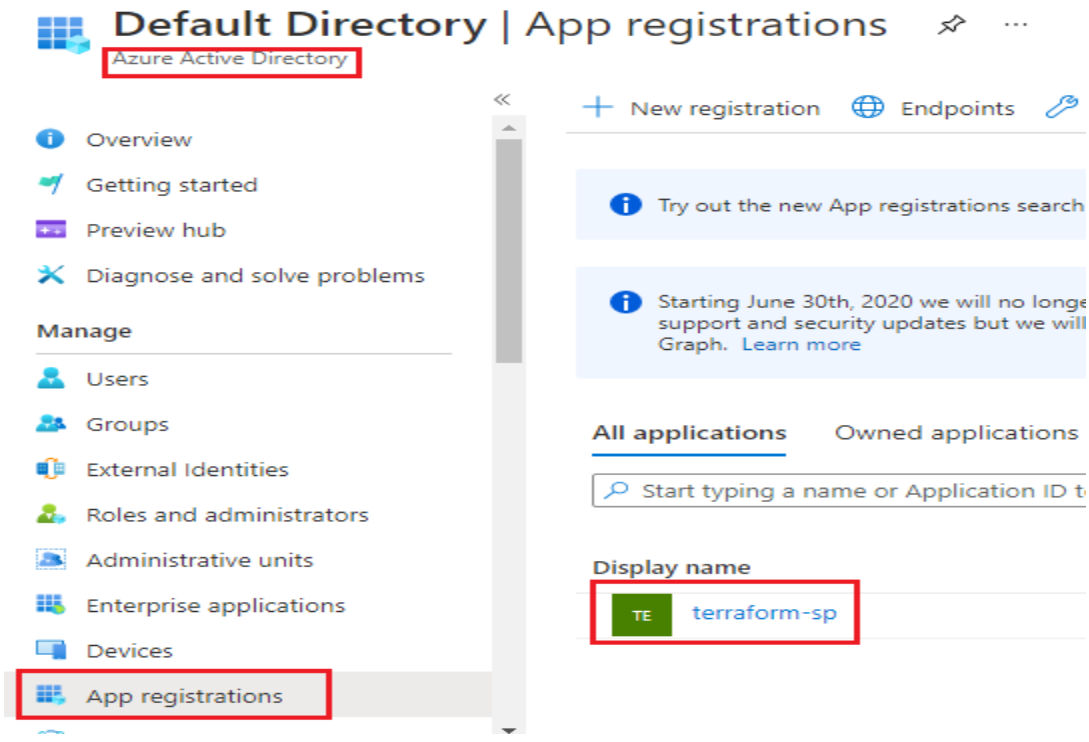
az role assignment create --assignee $SERVICE_PRINCIPAL --scope "/subscriptions/$SUBSCRIPTION" --role Contributor

```
PS C:\Kuberneties\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!

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>

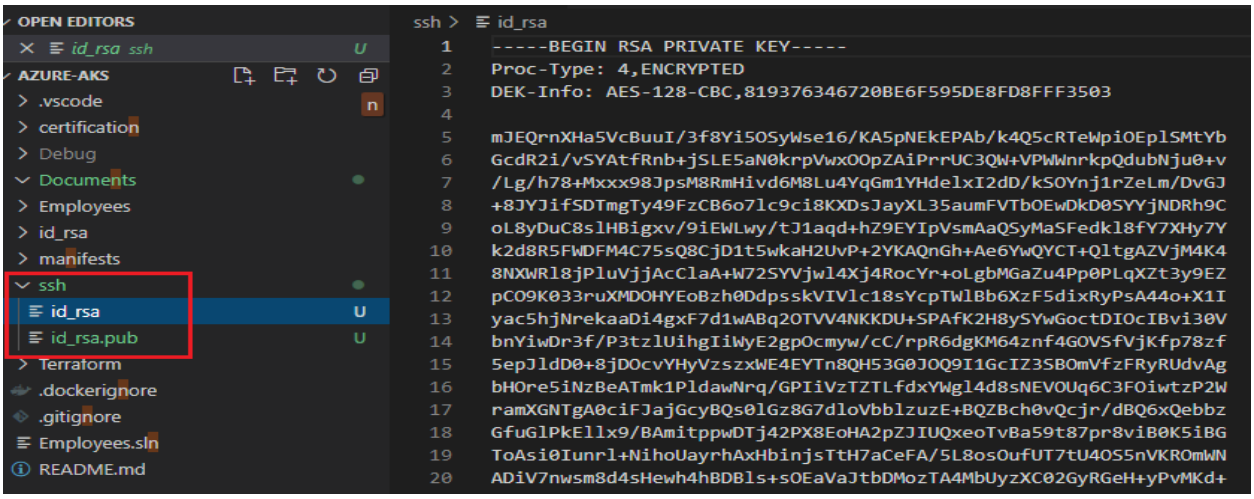
setx ARM_CLIENT_SECRET <password>

Generate SSH key

Later on we will need SSH key for connection to Kuberneties cluster in case of problems.

ssh-keygen -t rsa -b 4096 -N "VeryStrongSecret123!" -C "your_email@example.com" -q -f .\ssh\id_rsa

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



Done!