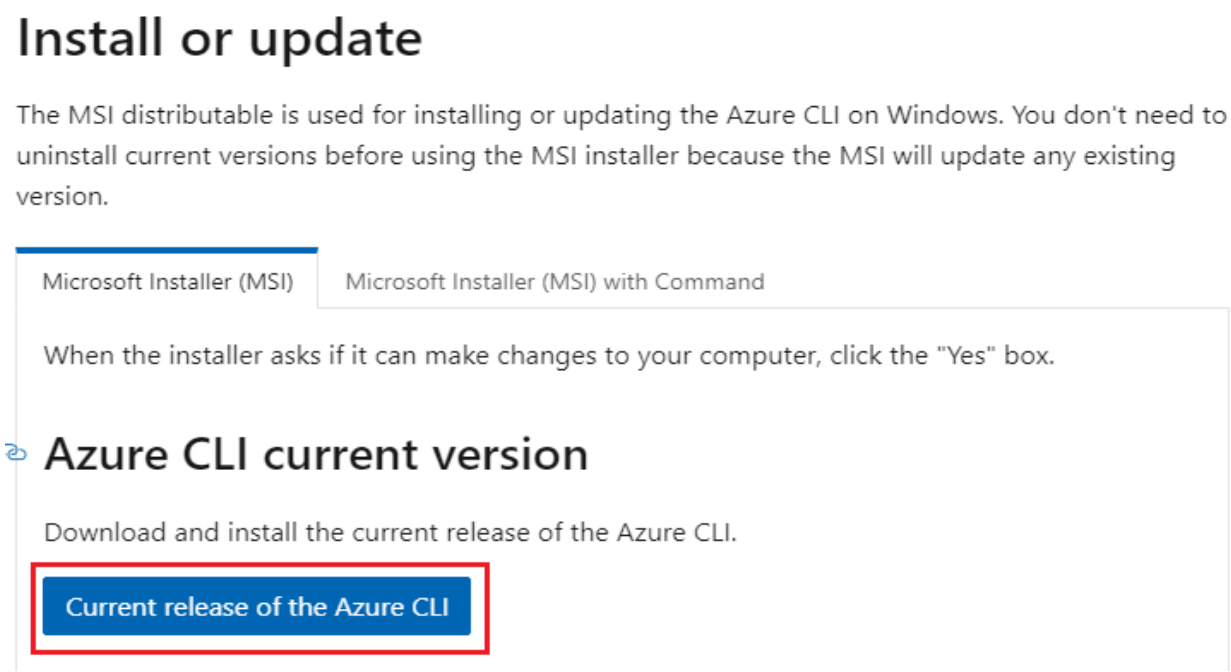
Azure CLI Installation and Service Principal Creation

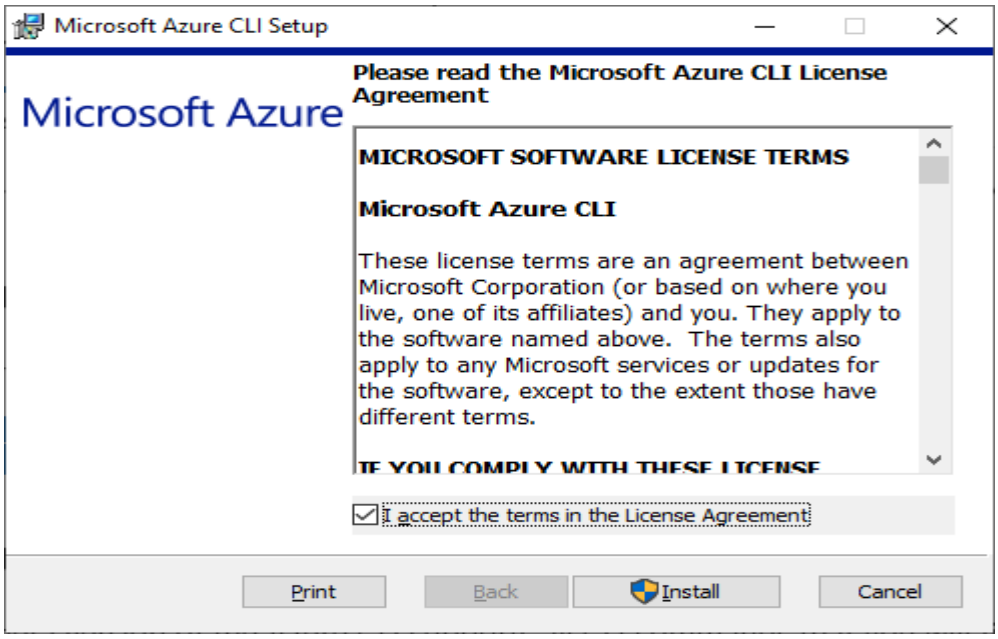
Azure CLI enables to manage resources in Azure from your PC.

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:

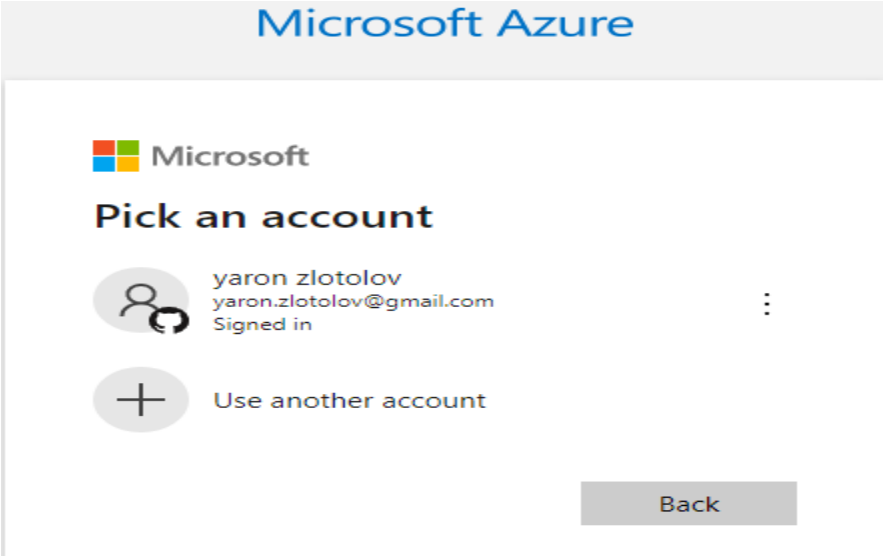


You can also install the Azure CLI using PowerShell. Start PowerShell as administrator and run the following command:

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 in variable for later use:

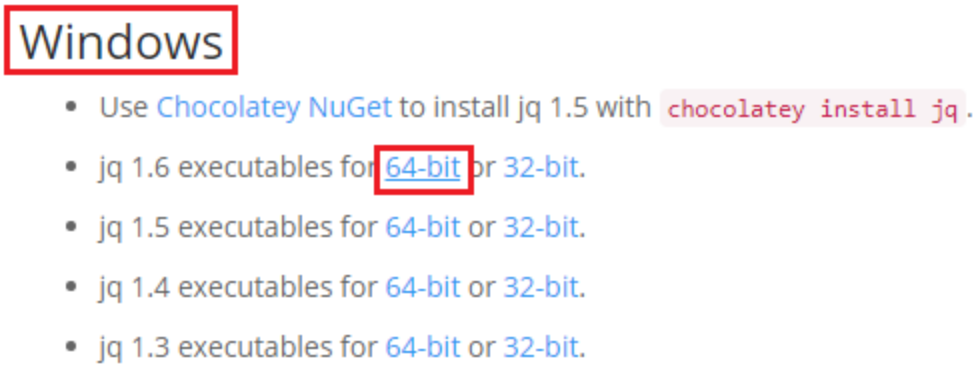
$SUBSCRIPTION="000000-0000-0000-0000-000000000"

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/



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:\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' 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:

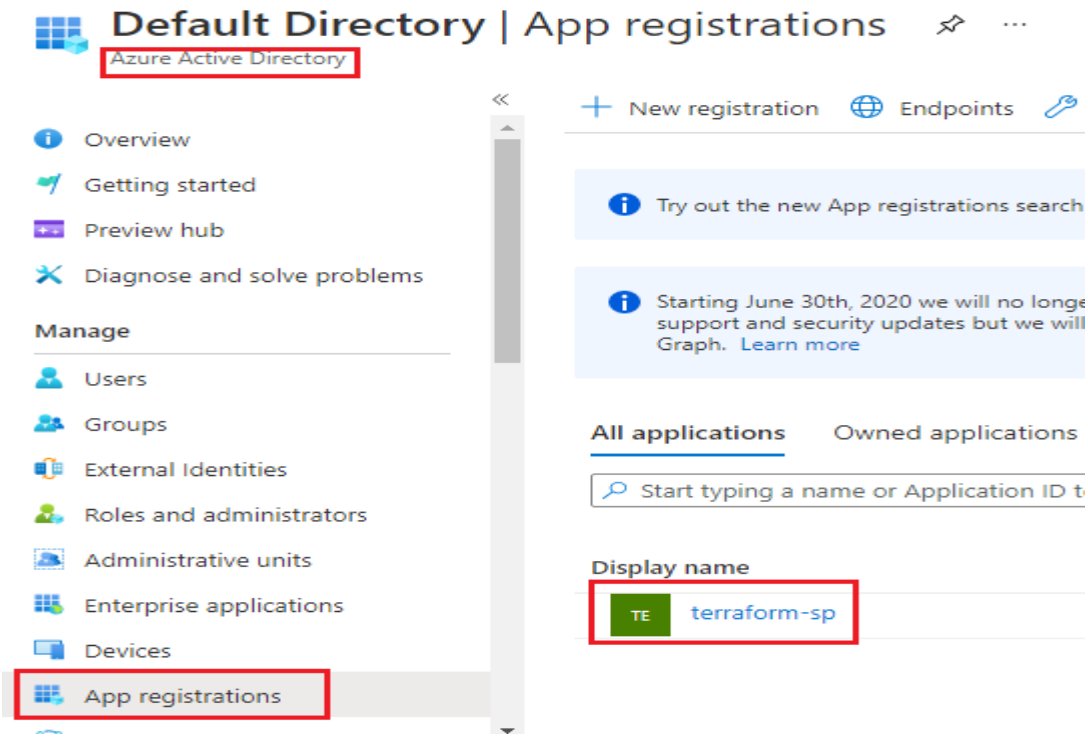
az role assignment create --assignee $SERVICE_PRINCIPAL --scope "/subscriptions/$SUBSCRIPTION" --role Contributor

```
PS C:\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!

Review the Service Principle in Azure Portal

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



List role assignments for the Service Principal:

az role assignment list --assignee $SERVICE_PRINCIPAL


```
[
  {
    "canDelegate": null,
    "condition": null,
    "conditionVersion": null,
    "description": null,
    "id": "/subscriptions/[redacted]/provid
4774-bb5c-714d1e6c2e8d",
    "name": "30ae4ccd-7d5e-4774-bb5c-714d1e6c2e8d",
    "principalId": "e92bf080-3bcd-4898-8486-6515d0b23204",
    "principalName": "http://terraform-sp",
    "principalType": "ServicePrincipal",
    "roleDefinitionId": "/subscriptions/133bc1d5-767e-4628-a9d1-0914f
b24988ac-6180-42a0-ab88-20f7382dd24c",
    "roleDefinitionName": "Contributor",
    "scope": "/subscriptions/[redacted]",
    "type": "Microsoft.Authorization/roleAssignments"
  }
]
```


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>

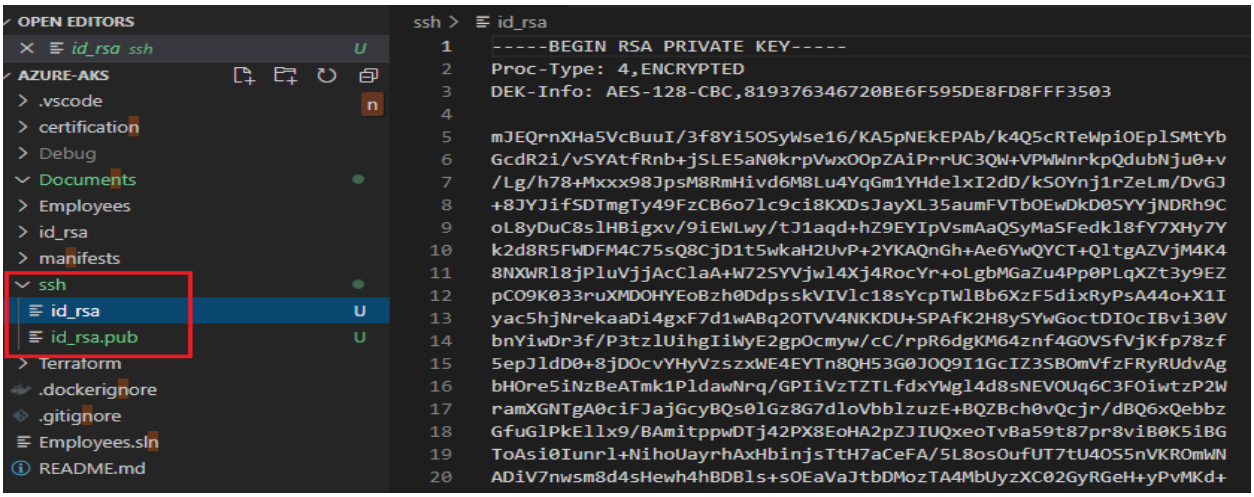
setx ARM_CLIENT_SECRET <password>

Generate SSH key

Later we will need SSH key for connecting to Kuberneties 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:\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>
```


OPEN EDITORS
id_rsa ssh
AZURE-AKS
.vscode
certification
Debug
Documents
Employees
id_rsa
manifests
ssh
id_rsa
id_rsa.pub
Terraform
.dockerignore
.gitignore
Employees.sln
README.md
ssh > id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,819376346720BE6F595DE8FD8FFF3503
mJEQrnXHa5VcBuuI/3f8Yi5OSyWse16/KA5pNEkEPAb/k4Q5cRTeWpiOEplSMtYb
GcdR2i/vSYAtfRnb+jSLE5aN0krpVwxOOpZAiPrrUC3QW+VPWWnrkpQdubNju0+v
/Lg/h78+Mxxx98JpsM8RmHivd6M8Lu4YqGm1YHdelxI2dD/kSOYnj1rZeLm/DvGJ
+8JYJifSDTmgTy49FzCB6o7lc9ci8KXDsJayXL35aumFVTbOEwDkD0SYYjNDRh9C
oL8yDuC8slHBigxv/9iEWLwy/tJ1aqd+hZ9EYIpVsmAaQSyMaSFedkl8fY7XHy7Y
k2d8R5FWDFM4C75sQ8CjD1t5wkaH2UvP+2YKAQnGh+Ae6YwQYCT+QltgAZVjM4K4
8NXWRl8jPluVjjAcClaA+W72SYVjwl4Xj4RocYr+oLgbMGaZu4Pp0PLqXZt3y9EZ
pCO9K033ruXMDOHYEoBzh0DdpsskVIVlc18sYcpTWlBb6XzF5dixRyPsA44o+X1I
yac5hjNrekaaDi4gxF7d1wABq2OTVV4NKKDU+SPAfK2H8ySYwGoctDIOcIBvi30V
bnYiwDr3f/P3tzlUihgIiWyE2gpOcmyw/cC/rpR6dgKM64znf4GOVSfVjKfp78zf
5epJldD0+8jDOcvYHyVzszxWE4EYTn8QH53G0JOQ9I1GcIZ3SBOmVfzFRyRUdvAg
bHOre5iNzBeATmk1PldawNrq/GPIiVzTZTLfdxYWgl4d8sNEVOUq6C3FOiwtzP2W
ramXGNTgA0ciFJajGcyBQs0lGz8G7dloVbblzuzE+BQZBch0vQcjr/dBQ6xQebbz
GfuGlPkEllx9/BAmitppwDTj42PX8EoHA2pZJIUQxeoTvBa59t87pr8viB0K5iBG
ToAsi0Iunrl+NihoUayrhAxHbinjsTtH7aCeFA/5L8osOufUT7tU4OS5nVKROmWN
ADiV7nwsm8d4sHewh4hBDBls+sOEaVaJtbDMozTA4MbUyzXC02GyRGeH+yPvMKd+

Done!