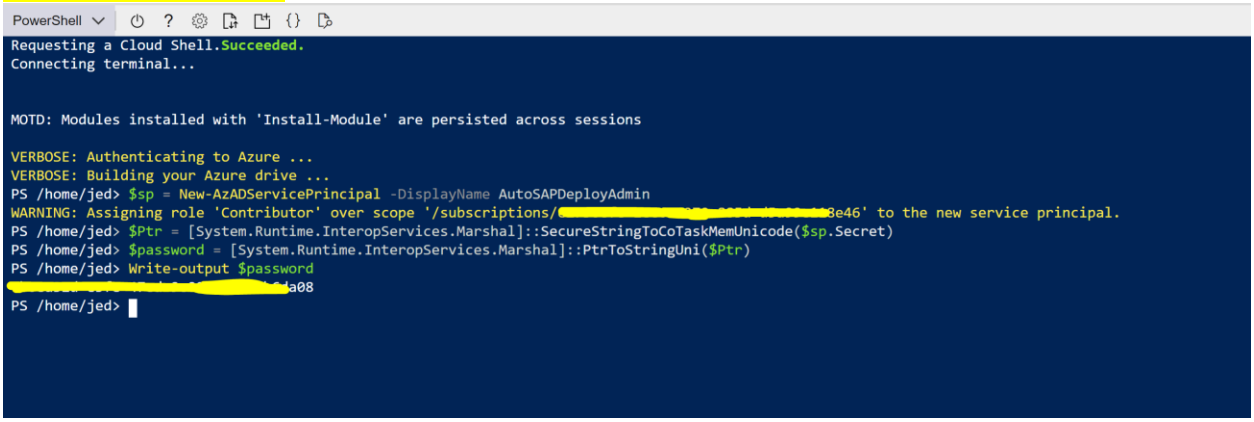
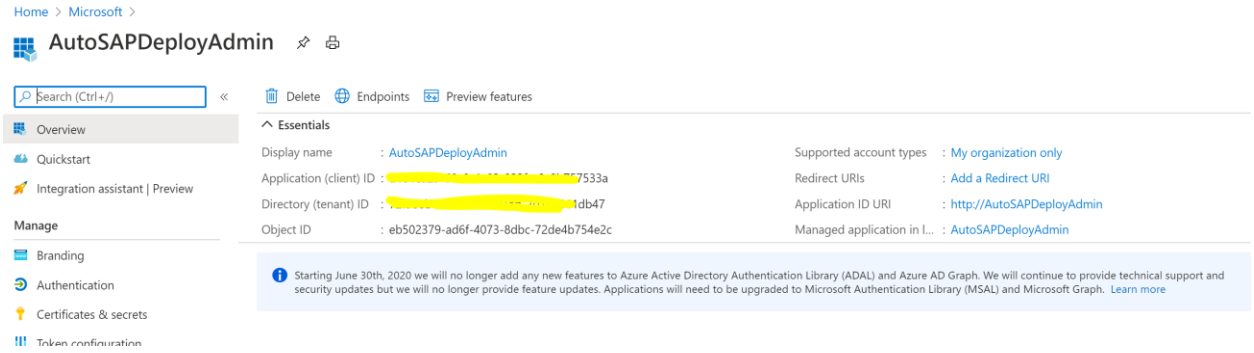
Openhack SAP deployment steps

1. Create a service principle for the deployment in your subscription:
   Powershell command: (note: “AutoSAPDeployAdmin” will be the principle name to be used. You should change it to somename unique like “AutoSAPDeployAdmin_johnsmith”)

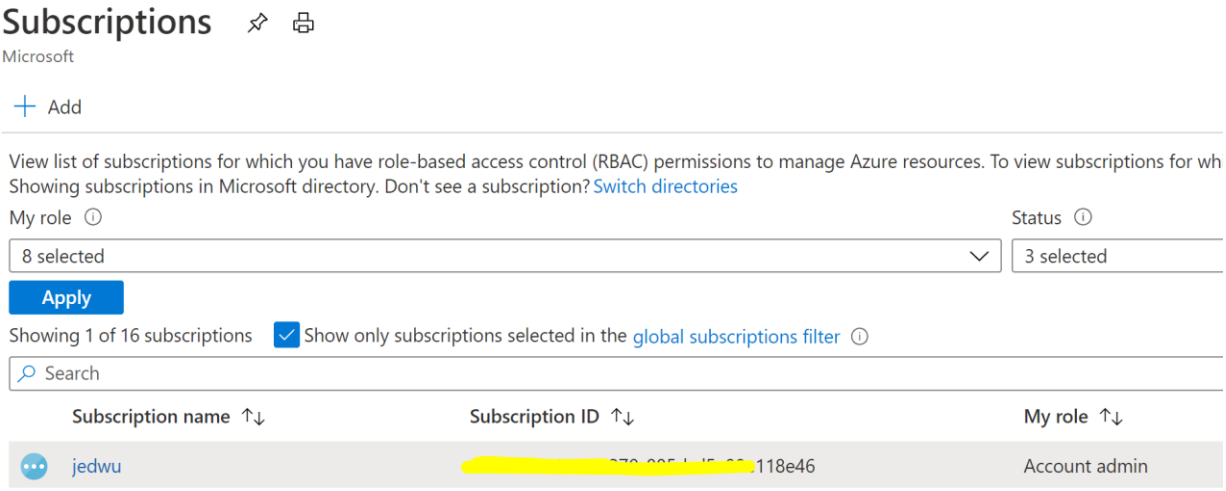
```
$sp = New-AzADServicePrincipal -DisplayName AutoSAPDeployAdmin
$Ptr =
[System.Runtime.InteropServices.Marshal]::SecureStringToCoTaskMemUnicode($sp.Secret)
$password = [System.Runtime.InteropServices.Marshal]::PtrToStringUni($Ptr)
Write-output $password
```



   Take down the $password value. Also take down the following info from Azure portal → Azure Active Directoy→App registrations and select the principle “AutoSAPDeployAdmin”. Note down the **Application (client) ID** and **Directory (tenant) ID** field.



   Finally take down your **subscription ID**:



2. Provision an ubuntu linux server through Azure portal (18.04 LTS, SKU: Standard DS1 v2) with named user "azureuser". Login to the server as the named user "**azureuser**"

```
% mkdir TST200/
% cd TST200/
% wget "https[package_url]"  -O ophk.tar.gz
```

⇨ Coach will provide the package_url

```
% gzip -d  ophk.tar .gz
% tar xf  ophk.tar
% ./local_setup_env.sh
```

3. Edit the "main.inputs" parameter file within this folder for the following settings:
   - [required] Subscription and principle information → replace all the "xxxxx" with the data taken down from step 1.

```
azure_login:
  subscription_id: "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
  client_id: "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
  client_secret:  "xxxxxxxxxxxxxxxxxxxxxxxxxx"
  tenant_id:  "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
```

   - [required] Resource prefix setting: please change the "teamxx" to represent your team number. Eg. "team00"

```
resource_prefix:  "teamxx"
```

   - [required] Resource group information: you can change the "**saprg_ophk_teamxx**" to the name representing your team.

```
resource_group:
  name: "saprg_ophk_teamxx"
  state: "new"
  region: "westus2"
```

   - [optional] SAP system/app build information: you can change the SAP system name to a three-letter name.

```
SAP_system_name:          "s4p"
```

Now save the main.inputs

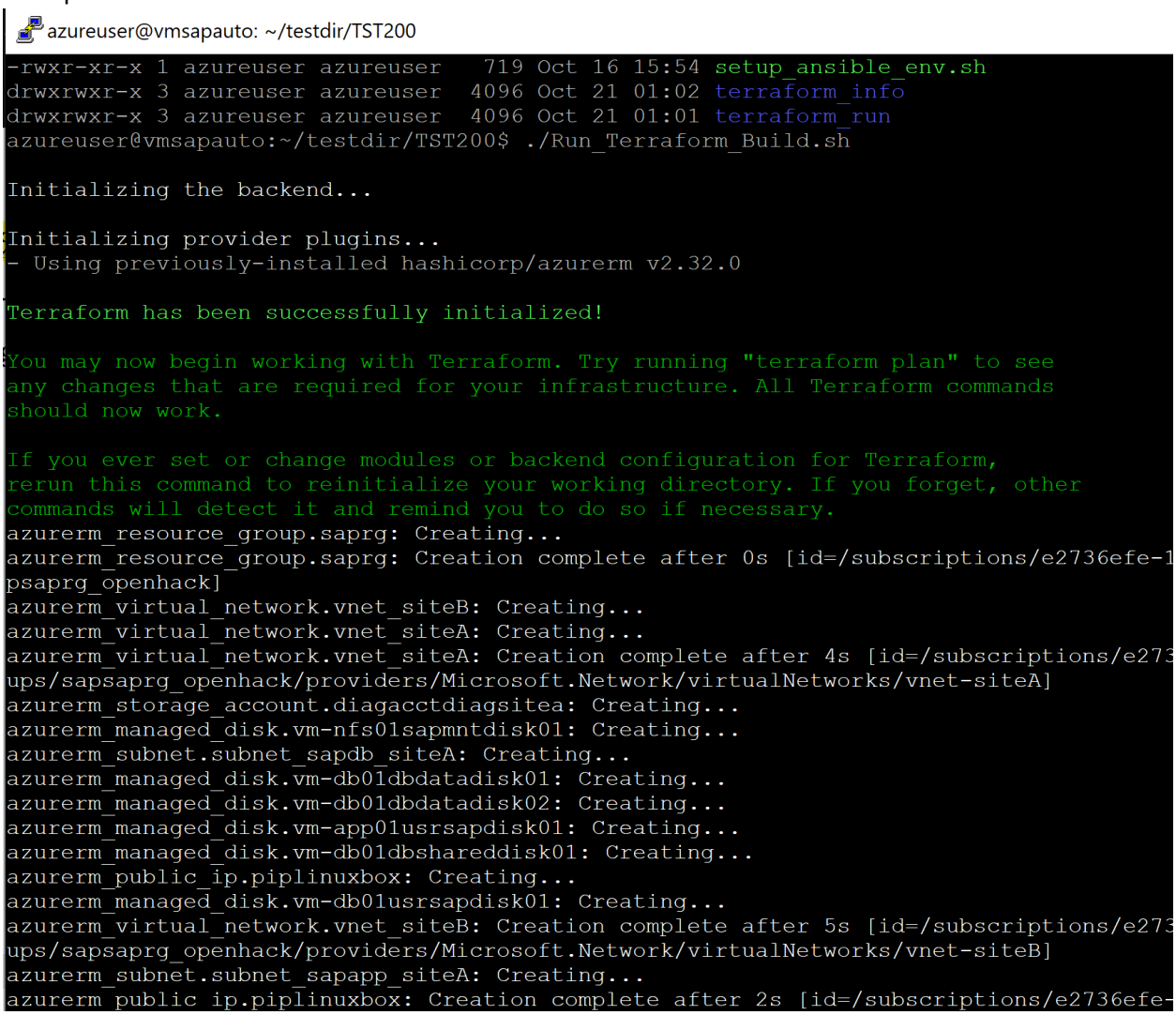
4. Generate runnable terraform scripts
   % python3 gen_terraform_script.py

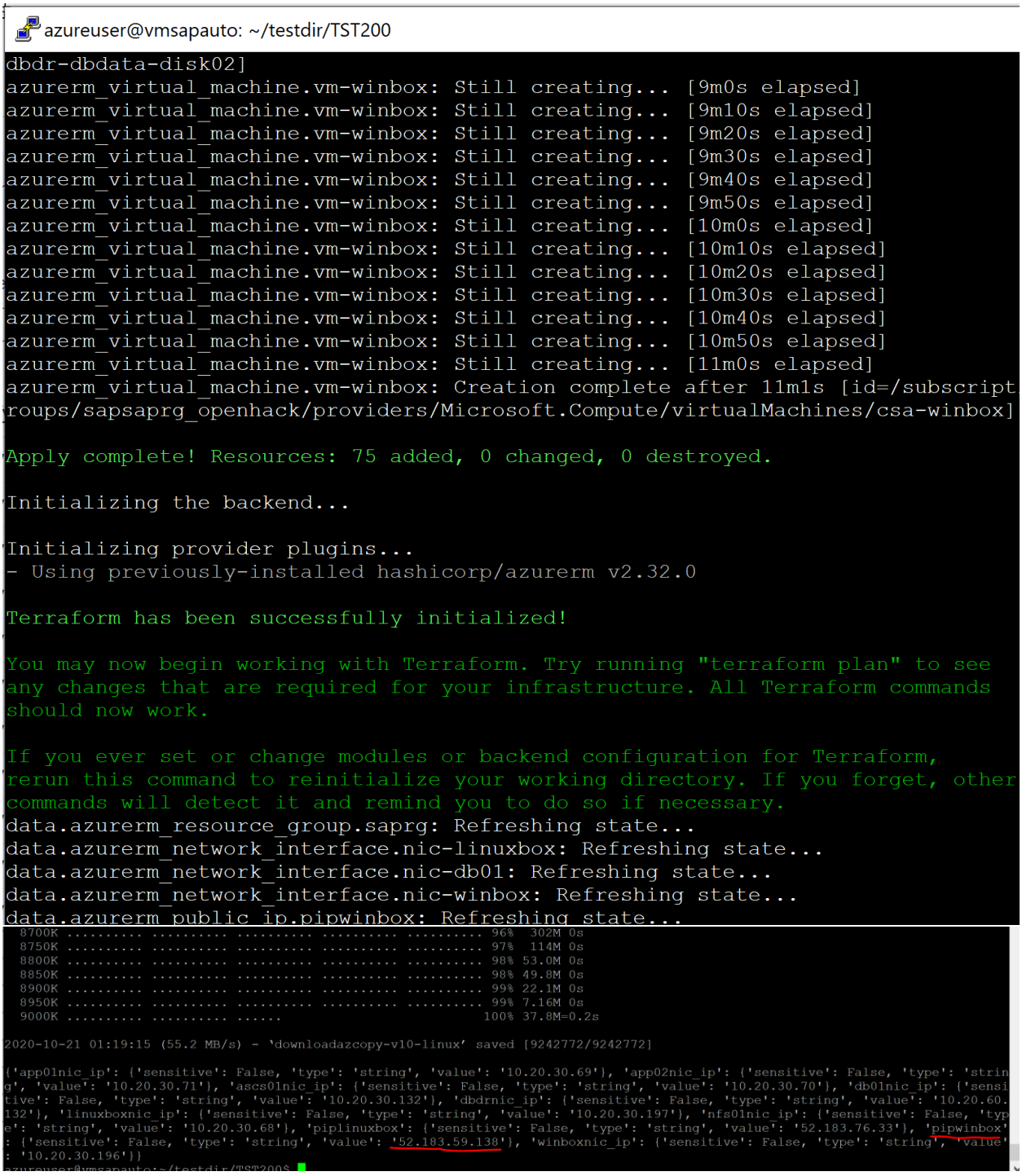
5. Run terraform script to build the Azure infrastructure
   % ./Run_Terraform_Build.sh

   This script typically run 15-20 min and will provision VNET/subnet/PPG/AVset/VM/.. within a resource group named “saprg_*ophk_teamxx*” (which is configurable within “main.inputs”)

   Note: **Please make sure that you are running this script under “azureuser” account as we observed some permission issue while using root.**

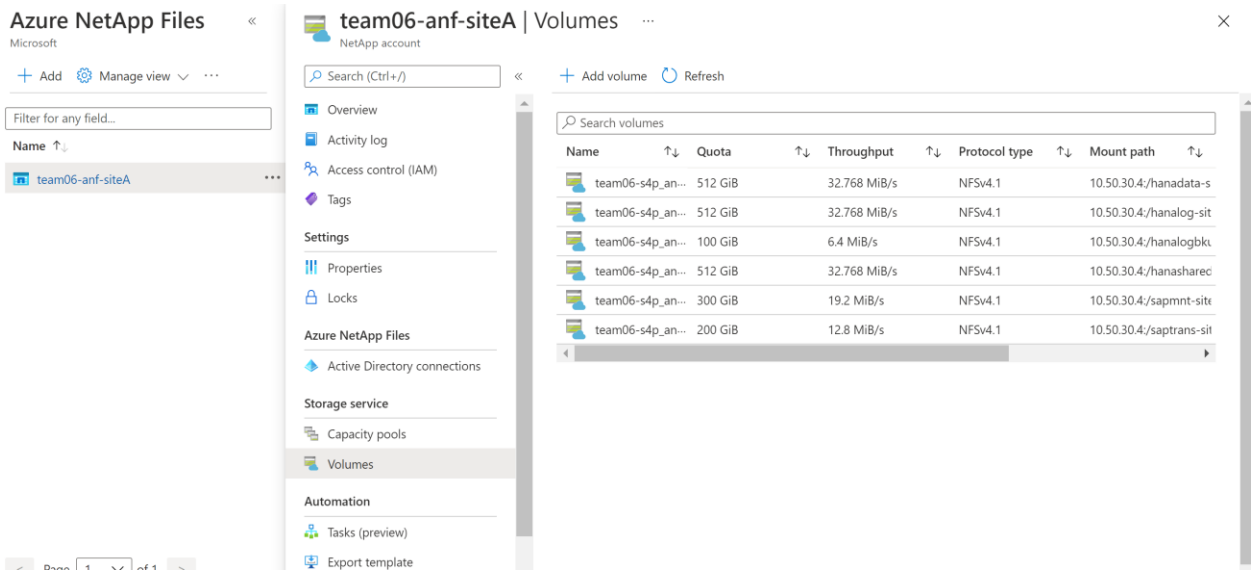
   Example of a successful run:

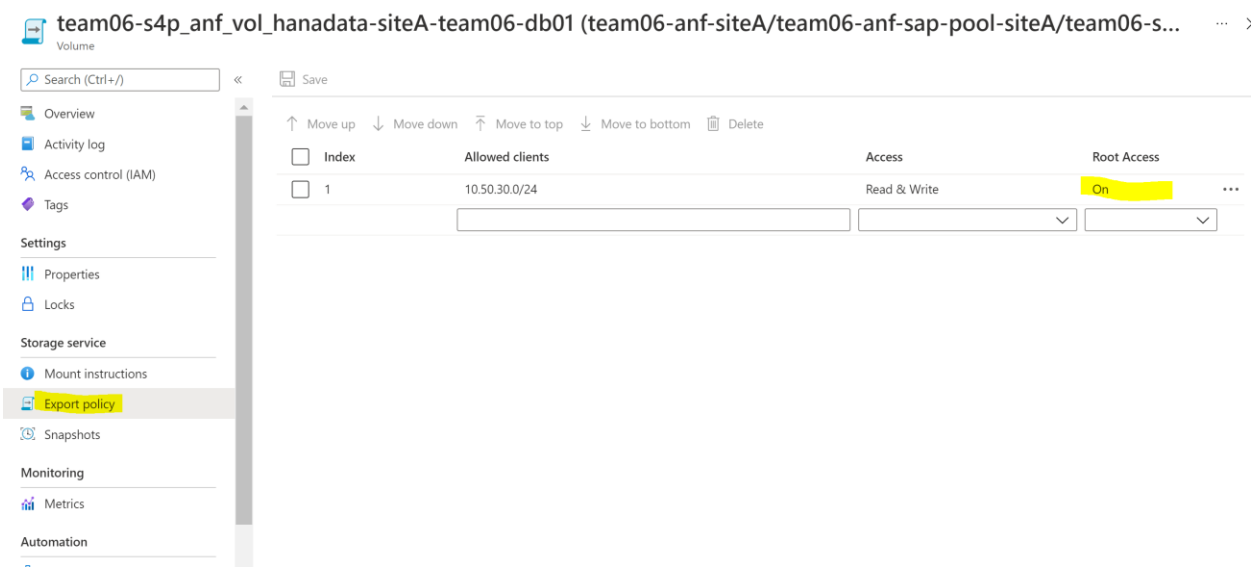




At the end of the execution find out the public IP of the window jumpbox and RDP to the window jumpbox (get the public IP show in the redline field from "**pipwinbox**"): login credential: azureuser/Welcome!2345

6. [**Additional Manual checking step**] This step is introduced because of some changes from Azure Netapp Filesystem flip the default access policy and Terraform script has not caught up with the update.
   Logon to portal: go to ANF account created and display each ANF volumes and check the export policy for every volume has “Root Access” to be “On”. If it shows “Off” then change it to “On” and save – for each netapp file volumes.





7. Logon to the window jumpbox. Download the following tools and SAP packages: Note, you might want to install and switch to some other browser to download these as the default browser with window defender will block the direct download.
   Putty.exe: **coach will provide the link**

   SAP GUI 7.60: **coach will provide the link**

   HANA studio 2.0: **coach will provide the link**

8. From the window jumpbox, logon to the linux jumpbox:

   Putty session to “team**xx**-linux-jumpbox” with the credential “azureuser/Welcome!2345” – please replace “xx” with your team number chosen in step 3.

   % cd ~azureuser/Current_Deployment

   % cd ansible

   % ./SAP_Ansible_Deploy.sh

   This script will config and install a complete SAP system and last several hours (typically 2-7 hours) session of server configuration and SAP installations (you can leave the session running in window jumpbox and reconnect late time). For a S4Hana instance fresh install it could take much longer.

   While the ansible script is running, you can continue with step 5 on install SAPGUI and SAP HANA studio on Window jumpbox.

9. Login to SAP system or HANA DB through SAPGUI or hana studio and complete the rest of the challenges.