Enforce SSH key authentication for Linux Azure-VMs

Create Azure Key Vault

```
keyvault_name=kv-mvpn-prod-we-001
az keyvault create \
--resource-group MS-Tunnel-rg \
--name $keyvault_name --enabled-for-deployment
```

From inside destination Linux VM

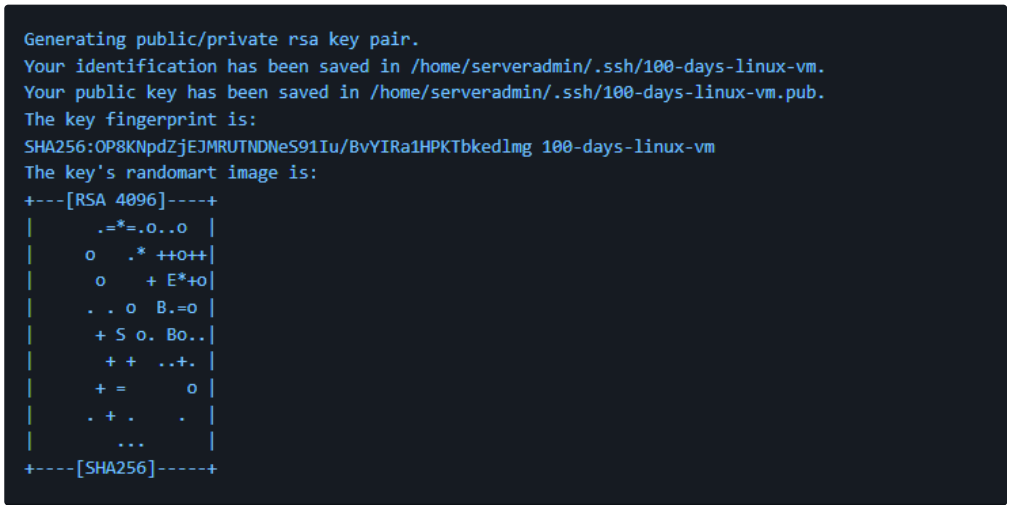
Generate Random Password

```
SSH_KEY_PASSWORD=$(openssl rand -base64 20)
```

Generate new SSH Keys

```
ssh-keygen \
-m PEM \
-t rsa \
-b 4096 \
-C "admin_local@euaz1-mvpn-38" \
-f ~/.ssh/euaz1-mvpn-38 \
-N "$SSH_KEY_PASSWORD"
```

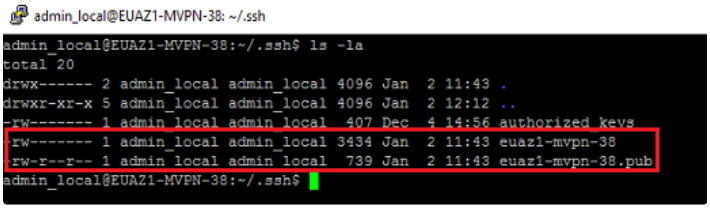
It should looks something like this (as an example)



Add Public Key to authorized_keys

```
cat ~/.ssh/euaz1-mvpn-38.pub >> ~/.ssh/authorized_keys
```

Copy following 2 generated files to your computer



From your computer

Add the SSH Keys to the Azure Key Vault

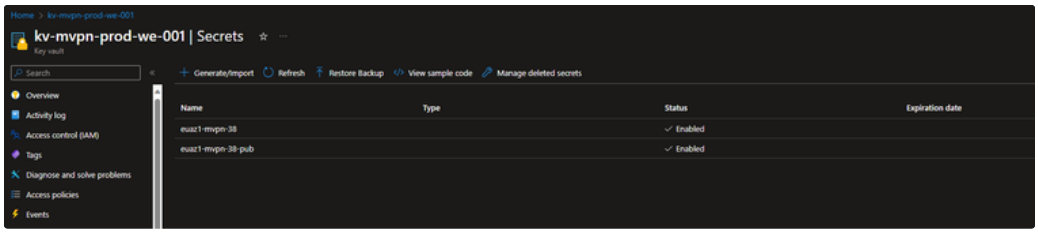
1. Copy `euaz1-mvpn-38` and `euaz1-mvpn-38.pub` keys to your local computer, normally under `C:\Users\[YOUR-USERNAME]\.ssh`
2. Log into Visual Studio, scope to the subscription where it's located the resource group
3. Create pub and priv secrets on the Key Vault

```
az keyvault secret set --name "euaz1-mvpn-38-pub" \
--vault-name "kv-mvpn-prod-we-001" \
--file ~/.ssh/euaz1-mvpn-38.pub

az keyvault secret set --name "euaz1-mvpn-38" \
--vault-name "kv-mvpn-prod-we-001" \
--file ~/.ssh/euaz1-mvpn-38
```

Output should be something similar to this (screenshot clipped)

You should see secrets inside like this:



Add the Password to the Key Vault

1. Open Azure CLI from Azure (otherwise visual studio could introduce undesired characters)
2. Get the password from the source VM and copy to your computer's clipboard > `echo $SSH_KEY_PASSWORD`
3. Set environment variable > `SSH_KEY_PASSWORD=+/YUYJoj.....`

4. Run these commands on Azure CLI from Azure

```
az keyvault secret set --name "euaz1-mvpn-38-passw" \
--vault-name "kv-mvpn-prod-we-001" \
--value "$SSH_KEY_PASSWORD"
```

Deploy a new Linux VM in Azure

```
az vm create \
--resource-group "[RESOURCE_GROUP_NAME" \
--name "[VM_NAME]" \
--image UbuntuLTS \
--admin-username "admin_local" \
--ssh-key-values "$SSH_PUBLIC_KEY" \
--output table
```

Retrieve the SSH Private Key Password from Key Vault

```
export SSHPASS=$(az keyvault secret show \
--name "euaz1-mvpn-38" \
--vault-name "kv-mvpn-prod-we-001"
--query value \
--output tsv 2>&1)
```

Use ssh-agent to store your private key passphrase

To avoid typing your private key file passphrase with every SSH sign-in, you can use `ssh-agent` to cache your private key file passphrase on your local system.

Verify and use `ssh-agent` and `ssh-add` to inform the SSH system about the key files so that you do not need to use the passphrase interactively:

```
eval "$(ssh-agent -s)"
```

Now add the private key to `ssh-agent` using the command `ssh-add`:

```
ssh-add ~/.ssh/euaz1-mvpn-38
```

The private key passphrase is now stored in `ssh-agent`.

Create and configure an SSH config file

You can create and configure an SSH config file (`~/.ssh/config`) to speed up log-ins and to optimize your SSH client behavior.

```
notepad ~/.ssh/config
```

Example Content

```
# Azure Keys
Host mvpn-90
  Hostname 10.242.142.90
  User admin_local
Host mvpn-38
```

```
  Hostname 10.242.142.11
  User admin_local
Host mvpn-39
  Hostname 10.242.142.12
  User admin_local
# ./Azure Keys
```

You can add configurations for additional hosts to enable each to use its own dedicated key pair. See SSH config file for more advanced configuration options.

```
ssh -i ~/.ssh/euaz1-mvpn-38 mvpn-38
```

Disabling Password Authentication

1. Change to the following option in the following 2 files: `etc/ssh/sshd_config` and `/etc/ssh/sshd_config.d/40-cloud-init.conf`

```
PasswordAuthentication no
```

2. Restart `ssh` server

```
sudo systemctl restart ssh
```

Further reads

https://github.com/starkfell/100DaysOfIaC/blob/master/articles/day.68.manage.access.to.linux.vms.using.key.vault.part.1.md

https://github.com/MicrosoftDocs/azure-docs/blob/main/articles/virtual-machines/linux/create-ssh-keys-detailed.md