AERO-WEBB
Technical Direction Docs

How to configure Azure

How to configure Azure

Prerequisites / Configuration

Service Principal

The provisioning of VMs requires the creation of a technical account called a service principal.
To do this, follow the DEP documentation:
https://dep-docs.apps.ocp4.innershift.sodigital.io/docs/platforms/service-principal/

Variable Definitions

Identification information must never be shared publicly ❗

```
# Authentication (by service principal)
export AZ_CLIENT_ID="9c20ac13-8196-4c6c-a76f-33923f08437e"
export AZ_CLIENT_SECRET="<define it locally>"
export AZ_TENANT_ID="steria.onmicrosoft.com"
export AZ_SUBSCRIPTION_ID="d2829e59-bf97-42ec-b5a8-3f9b23a61a2a"

# Location (do not change it)
export AZ_LOCATION="francecentral"
# CIDR (Soprasteria network)
export AZ_CIDR="81.80.72.0/27 81.80.17.0/27 115.117.111.92 14.140.238.18"

# Resources (have to be defined)
export AZ_RESOURCE_GROUP="aerowebb-dev-group"
export AZ_SUBNETWORK="aerowebb-dev-subnet"
export AZ_VIRTUAL_NETWORK="aerowebb-dev-vnet"
export AZ_NETWORK_SECURITY_GROUP="aerowebb-dev-nsg"
export AZ_STORAGE_ACCOUNT="aerowebbdevstorage" # alphanumeric, lowercase, no special characters
export AZ_KEY_VAULT_NAME="aerowebb-dev-vault"
export AZ_MANAGED_IDENTITY="aerowebb-dev-identity"
export AZ_BASTION="aerowebb-dev-bastion"

# Configuration (can be customized)
export AZ_STORAGE_CONTAINER="postgres-dump"
export AZ_MOUNT_POINT="/mnt/$AZ_STORAGE_CONTAINER"
export AZ_SFTP_DEVELOPER="developer"
export AZ_SFTP_MAINTAINER="maintainer"

# Miscellaneous (doesn't need to be changed)
export AZ_STORAGE_KEY="storage-key1"

# Example of VM name, must start with "slaz" and end with number
# See: https://dep-docs.apps.ocp4.innershift.sodigital.io/docs/platforms/azure/restriction-policies/namingrules/
export AZ_VM_NAME="slazaerowebbdev01"
```

Connection via Service Principal

```
az login \
  --service-principal \
  --tenant $AZ_TENANT_ID \
  --username $AZ_CLIENT_ID \
  --password $AZ_CLIENT_SECRET
```

Create a resource group

```
az group create \
  --name $AZ_RESOURCE_GROUP \
  --location $AZ_LOCATION
```

Network Configuration

> *Traffic between resources in the same virtual network is allowed by default*

Create a Virtual Network and a Subnet

```
az network vnet create \
  --resource-group $AZ_RESOURCE_GROUP \
  --name $AZ_VIRTUAL_NETWORK \
  --address-prefix 10.0.0.0/16 \
  --subnet-name $AZ_SUBNETWORK \
  --subnet-prefix 10.0.1.0/24
```

Create a Network Security Group (NSG)

```
az network nsg create \
  --resource-group $AZ_RESOURCE_GROUP \
  --name $AZ_NETWORK_SECURITY_GROUP
```

Associate NSG to Virtual Network

```
az network vnet subnet update \
  --vnet-name $AZ_VIRTUAL_NETWORK \
  --resource-group $AZ_RESOURCE_GROUP \
  --name $AZ_SUBNETWORK \
  --network-security-group $AZ_NETWORK_SECURITY_GROUP
```

Allow connections from Sopra Steria network

The details of the CIDR are defined in Whitelist CIDR

```
az network nsg rule create \
  --resource-group $AZ_RESOURCE_GROUP \
  --nsg-name $AZ_NETWORK_SECURITY_GROUP \
  --name AllowAllFromSopraSteria \
  --priority 1000 \
  --protocol Tcp \
  --direction Inbound \
  --source-address-prefixes $AZ_CIDR \
  --destination-port-ranges '*' \
  --access Allow
```

Create a shared storage space

Create a Key Vault that meets DEP requirements

```
az keyvault create \
  --name $AZ_KEY_VAULT_NAME \
  --resource-group $AZ_RESOURCE_GROUP \
  --location $AZ_LOCATION \
  --enable-purge-protection true \
  --enable-rbac-authorization false \
  --network-acls-vnets $AZ_VIRTUAL_NETWORK/$AZ_SUBNETWORK
```

Allow Service endpoint connection to Key Vault for VNet/Subnet

Blob Storage

Create a Storage Account

> With DEP requirements, with SFTP and hierarchical namespace enabled

```
az storage account create \
    --name $AZ_STORAGE_ACCOUNT \
    --resource-group $AZ_RESOURCE_GROUP \
    --vnet-name $AZ_VIRTUAL_NETWORK \
    --subnet $AZ_SUBNETWORK \
    --location $AZ_LOCATION \
    --sku Standard_LRS \
    --kind StorageV2 \
    --https-only true \
    --min-tls-version TLS1_2 \
    --enable-sftp true \
    --enable-local-user true \
    --enable-hierarchical-namespace true

az storage account network-rule add \
    --resource-group $AZ_RESOURCE_GROUP \
    --account-name $AZ_STORAGE_ACCOUNT \
    --vnet-name $AZ_VIRTUAL_NETWORK \
    --subnet $AZ_SUBNETWORK \
    --ip-address $AZ_CIDR

az storage container create \
  --account-name $AZ_STORAGE_ACCOUNT \
  --name $AZ_STORAGE_CONTAINER \
  --auth-mode login
```

Create a developer SFTP account

Permissions: Read, List, Create

```
az storage account local-user create \
  --resource-group $AZ_RESOURCE_GROUP \
  --account-name $AZ_STORAGE_ACCOUNT \
  --user-name $AZ_SFTP_DEVELOPER \
  --permission-scope permissions=rlc service=blob resource-name=$AZ_STORAGE_CONTAINER \
  --home-directory $AZ_STORAGE_CONTAINER \
  --has-ssh-password

# Force to regenerate the password (copy it and store it securely !)
az storage account local-user regenerate-password \
  --resource-group $AZ_RESOURCE_GROUP \
  --account-name $AZ_STORAGE_ACCOUNT \
  --user-name $AZ_SFTP_DEVELOPER
```

Create a maintainer SFTP account

Permissions: Read, List, Create **& Write, Delete**

```
# Create the user "maintainer"
az storage account local-user create \
  --resource-group $AZ_RESOURCE_GROUP \
  --account-name $AZ_STORAGE_ACCOUNT \
  --user-name $AZ_SFTP_MAINTAINER \
  --permission-scope permissions=rlcwd service=blob resource-name=$AZ_STORAGE_CONTAINER \
  --home-directory $AZ_STORAGE_CONTAINER \
  --has-ssh-password

# Regenerate the password (copy it and store it securely !)
az storage account local-user regenerate-password \
  --resource-group $AZ_RESOURCE_GROUP \
  --account-name $AZ_STORAGE_ACCOUNT \
  --user-name $AZ_SFTP_MAINTAINER
 ```

Activate Storage Account Firewall

Activate the firewall at the end of the configuration to avoid any blocking. :exclamation:

```bash
az storage account update \
  --resource-group $AZ_RESOURCE_GROUP \
  --name $AZ_STORAGE_ACCOUNT \
  --default-action Deny
```

Create a Virtual Machine

Example of manual creation, can be integrated into an automated provisioning solution (e.g., Terraform)

Generate cloud-init.yml file

```
storageAccountSecret=$(az storage account keys list \
  --resource-group $AZ_RESOURCE_GROUP \
  --account-name $AZ_STORAGE_ACCOUNT \
  --query '[0].value' \
  --output tsv)

cat <<EOF > cloud-init.yml
#cloud-config

package_update: true
package_upgrade: true
packages:
  - blobfuse2

write_files:
  - path: /etc/blobfuse2.yaml
    content: |
      components:
        - libfuse
      libfuse:
        mountPath: ${AZ_MOUNT_POINT}
        configFilePath: /etc/blobfuse2_connection.cfg
    permissions: '0644'
  - path: /etc/blobfuse2_connection.cfg
    content: |
      authType: key
      accountName: ${AZ_STORAGE_ACCOUNT}
      containerName: ${AZ_STORAGE_CONTAINER}
      accountKey: ${storageAccountSecret}
    permissions: '0644'

runcmd:
  - mkdir -p /mnt/blob
  - blobfuse2 mount --config-file=/etc/blobfuse2.yaml

mounts:
  - [ "blobfuse2#${AZ_STORAGE_ACCOUNT}#${AZ_STORAGE_CONTAINER}", "${AZ_MOUNT_POINT}", "fuse",
"_netdev,config_file=/etc/blobfuse2.yaml", "0", "0" ]
EOF
```

Create the VM that meets DEP requirements

```
az vm create \
  --resource-group $AZ_RESOURCE_GROUP \
  --name $AZ_VM_NAME \
  --location $AZ_LOCATION \
  --vnet-name $AZ_VIRTUAL_NETWORK \
  --subnet $AZ_SUBNETWORK \
  --nsg $AZ_NETWORK_SECURITY_GROUP \
  --image Ubuntu2204 \
  --size Standard_B4ms \
  --storage-sku StandardSSD_LRS \
  --os-disk-size-gb 80 \
  --admin-username azureuser \
  --generate-ssh-keys \
  --custom-data cloud-init.yml
```

Automation

Automatic VM shutdown

Configure automatic shutdown of the VM at 7:00 PM Europe/Paris time. In case of forgetting, DEP will add automatic shutdown of the VM within 24 hours.

```
az vm auto-shutdown --resource-group $AZ_RESOURCE_GROUP \
  --name $AZ_VM_NAME \
  --time 19:00 \
  --time-zone "Europe/Paris"
```

Automatic VM startup

TODO

Enable boot diagnostics

Enable boot diagnostics for the VM using the previously created storage account:

```
az vm boot-diagnostics enable \
  --resource-group $AZ_RESOURCE_GROUP \
  --name $AZ_VM_NAME \
  --storage $AZ_STORAGE_ACCOUNT
```

Limitations

We use a `service principal` to manage our resources, but this *technical account* has the role `Contributor`, which has limited access for security reasons. The `service principal` doesn't have the authorization `Microsoft.Authorization/roleAssignments/write'`, usually used to grant access with `managed identity` resource.

To do

- Use SAS service token (with automatic renewal) to access to Storage Account by VMs