AKS/KeyVault RUN

$  
 $ $keyVault = az keyvault create -n $keyVaultName -g $resourceGroupName -l $location --enable-soft-delete tru  
e --retention-days 7 | ConvertFrom-Json  
 $  
 $ az keyvault secret set --name $secret1Name --value "Houssem" --vault-name $keyVaultName  
{  
 "attributes": {  
 "created": "2020-05-01T17:58:07+00:00",  
 "enabled": true,  
 "expires": null,  
 "notBefore": null,  
 "recoveryLevel": "CustomizedRecoverable+Purgeable",  
 "updated": "2020-05-01T17:58:07+00:00"  
 },  
 "contentType": null,  
 "id": "https://keyvault-demo08.vault.azure.net/secrets/DatabaseLogin/dfba8679e7fe4619a1f43483242e97f1",  
 "kid": null,  
 "managed": null,  
 "name": "DatabaseLogin",  
 "tags": {  
 "file-encoding": "utf-8"  
 },  
 "value": "Houssem"  
}  
 $ az keyvault secret set --name $secret2Name --value "P@ssword123456" --vault-name $keyVaultName  
{  
 $  
 $ helm repo add secrets-store-csi-driver https://raw.githubusercontent.com/kubernetes-sigs/secrets-store-csi-  
driver/master/charts  
"secrets-store-csi-driver" has been added to your repositories  
 $  
 $ kubectl create ns csi-driver  
namespace/csi-driver created  
 $ helm install csi-secrets-store secrets-store-csi-driver/secrets-store-csi-driver --namespace csi-driver  
 $  
 $ kubectl apply -f https://raw.githubusercontent.com/Azure/secrets-store-csi-driver-provider-azure/master/dep  
loyment/provider-azure-installer.yaml --namespace csi-driver  
daemonset.apps/csi-secrets-store-provider-azure created  
 $ kubectl get pods -n csi-driver  
NAME READY STATUS RESTARTS AGE  
csi-secrets-store-provider-azure-4gdf6 0/1 ContainerCreating 0 1s  
csi-secrets-store-provider-azure-hcq2t 0/1 ContainerCreating 0 1s  
csi-secrets-store-provider-azure-qjczb 0/1 ContainerCreating 0 1s  
csi-secrets-store-secrets-store-csi-driver-82ts4 3/3 Running 0 37s  
csi-secrets-store-secrets-store-csi-driver-bc2nd 3/3 Running 0 37s  
csi-secrets-store-secrets-store-csi-driver-p6f8b 3/3 Running 0 37s  
 $  
 $ $secretProviderKV = @"  
>> apiVersion: secrets-store.csi.x-k8s.io/v1alpha1  
>> kind: SecretProviderClass  
>> metadata:  
>> name: $($secretProviderClassName)  
>> spec:  
>> provider: azure  
>> parameters:  
>> usePodIdentity: "true"  
>> useVMManagedIdentity: "false"  
>> userAssignedIdentityID: ""  
 $  
 $ kubectl apply -f https://raw.githubusercontent.com/Azure/aad-pod-identity/master/deploy/infra/deployment-rb  
ac.yaml  
serviceaccount/aad-pod-id-nmi-service-account created  
customresourcedefinition.apiextensions.k8s.io/azureassignedidentities.aadpodidentity.k8s.io crea $  
 $  
 $ az keyvault set-policy -n $keyVaultName --secret-permissions get --spn $identity.clientId  
{  
 "id": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/resourceGroups/rg-demo08/providers/Microsoft.KeyV  
ault/vaults/keyvault-demo08",  
 "location": "westeurope",  
 "name": "keyvault-demo08",  
 "properties": {  
 "accessPolicies": [  
 {  
 "applicationId": null,  
 "objectId": "9c1706eb-db28-4105-b993-f93a83f974b8",  
 "permissions": {  
 "certificates": [  
 "get",  
 "list",  
 "delete",  
 "create",  
 "import",  
 "update",  
 "managecontacts",  
 "getissuers",  
 "listissuers",  
 "setissuers",  
 "deleteissuers",  
 "manageissuers",  
 "recover"  
 ],  
 "keys": [  
 "get",  
 "create",  
 "delete",  
 "list",  
 "update",  
 "import",  
 "backup",  
 "restore",  
 "recover"  
 ],  
 "secrets": [  
 "get",  
 "list",  
 "set",  
 "delete",  
 "backup",  
 "restore",  
 "recover"  
 ],  
 "storage": [  
 "get",  
 "list",  
 "delete",  
 "set",  
 "update",  
 "regeneratekey",  
 "setsas",  
 "listsas",  
 "getsas",  
 "deletesas"  
 ]  
 },  
 "tenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47"  
 },  
 {  
 "applicationId": null,  
 "objectId": "937cfa7a-85e6-42cc-bb15-3afb0cbd3b2f",  
 "permissions": {  
 "certificates": null,  
 "keys": null,  
 "secrets": [  
 "get"  
 ],  
 "storage": null  
 },  
 "tenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47"  
 }  
 ],  
 "createMode": null,  
 "enablePurgeProtection": null,  
 "enableRbacAuthorization": false,  
 $  
 $  
 $ $nginxPod = @"  
>> kind: Pod  
>> apiVersion: v1  
>> metadata:  
>> name: nginx-secrets-store  
>> labels:  
>> aadpodidbinding: $($identitySelector)  
>> spec:  
>> containers:  
>> - name: nginx  
>> image: nginx  
>> volumeMounts:  
>> - name: secrets-store-inline  
>> mountPath: "/mnt/secrets-store"  
>> readOnly: true  
>> volumes:  
>> - name: secrets-store-inline  
>> csi:  
>> driver: secrets-store.csi.k8s.io  
>> readOnly: true  
>> volumeAttributes:  
>> secretProviderClass: $($secretProviderClassName)  
 $  
 $ kubectl exec -it nginx-secrets-store ls /mnt/secrets-store/  
DATABASE\_LOGIN DATABASE\_PASSWORD  
 $ kubectl exec -it nginx-secrets-store cat /mnt/secrets-store/DATABASE\_LOGIN  
Houssem  
 $ kubectl exec -it nginx-secrets-store cat /mnt/secrets-store/$secret1Alias  
 $  
 $ $keyVault = az keyvault create -n $keyVaultName -g $resourceGroupName -l $location --enable-soft-delete tru  
e --retention-days 7 | ConvertFrom-Json  
 $ # $keyVault = (az keyvault show -n $keyVaultName | ConvertFrom-Json) # retrieve existing KV  
 $  
 $ echo "Creating Secrets in Key Vault..."  
Creating Secrets in Key Vault...  
 $ az keyvault secret set --name $secret1Name --value "Houssem" --vault-name $keyVaultName  
{  
 "attributes": {  
 "created": "2020-05-01T20:04:10+00:00",  
 "enabled": true,  
 "expires": null,  
 "notBefore": null,  
 "recoveryLevel": "CustomizedRecoverable+Purgeable",  
 "updated": "2020-05-01T20:04:10+00:00"  
 },  
 "contentType": null,  
 "id": "https://keyvault-demo09.vault.azure.net/secrets/DatabaseLogin/4e01350b94ca42a19fadc3458e5623d8",  
 "kid": null,  
 "managed": null,  
 "name": "DatabaseLogin",  
 "tags": {  
 $  
 $ echo "Adding Helm repo for Secret Store CSI..."  
Adding Helm repo for Secret Store CSI...  
 $ helm repo add secrets-store-csi-driver https://raw.githubusercontent.com/kubernetes-sigs/secrets-store-csi-  
driver/master/charts  
"secrets-store-csi-driver" has been added to your repositories  
 $  
 $ echo "Installing Secrets Store CSI Driver using Helm..."  
Installing Secrets Store CSI Driver using Helm...  
 $  
 $  
 $ echo "Installing Secrets Store CSI Driver with Azure Key Vault Provider..."  
Installing Secrets Store CSI Driver with Azure Key Vault Provider...  
 $ kubectl apply -f https://raw.githubusercontent.com/Azure/secrets-store-csi-driver-provider-azure/master/dep  
loyment/provider-azure-installer.yaml --namespace csi-driver  
daemonset.apps/csi-secrets-store-provider-azure created  
 $ kubectl get pods -n csi-driver  
NAME READY STATUS RESTARTS AGE  
csi-secrets-store-provider-azure-4z8mw 0/1 ContainerCreating 0 1s  
csi-secrets-store-provider-azure-cj92z 0/1 ContainerCreating 0 1s  
csi-secrets-store-provider-azure-m8cbt 0/1 ContainerCreating 0 1s  
csi-secrets-store-secrets-store-csi-driver-k8wh7 3/3 Running 0 82s  
csi-secrets-store-secrets-store-csi-driver-nrrl9 3/3 Running 0 83s  
csi-secrets-store-secrets-store-csi-driver-qmcwv 3/3 Running 0 82s  
 $  
 $  
 $ echo "Using the Azure Key Vault Provider..."  
Using the Azure Key Vault Provider...  
 $ $secretProviderKV = @"  
>> apiVersion: secrets-store.csi.x-k8s.io/v1alpha1  
>> kind: SecretProviderClass  
>> metadata:  
>> name: $($secretProviderClassName)  
>> spec:  
 $  
 $ echo "Creating an Azure Identity..."  
Creating an Azure Identity...  
 $ $identity = az identity create -g $resourceGroupName -n $identityName | ConvertFrom-Json  
 $  
 $ echo "Assigning Reader Role to new Identity for Key Vault..."  
Assigning Reader Role to new Identity for Key Vault...  
 $ az role assignment create --role "Reader" --assignee $identity.principalId --scope $keyVault.id  
{  
 "canDelegate": null,  
 "id": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/resourceGroups/rg-demo09/providers/Microsoft.KeyV  
ault/vaults/keyvault-demo09/providers/Microsoft.Authorization/roleAssignments/03468c64-6b4b-407f-99c1-a91dd324  
beee",  
 "name": "03468c64-6b4b-407f-99c1-a91dd324beee",  
 "principalId": "99952e29-d98f-4b5c-968f-7d90c881027f",  
 "principalType": "ServicePrincipal",  
 "resourceGroup": "rg-demo09",  
 "roleDefinitionId": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/providers/Microsoft.Authorization/r  
oleDefinitions/acdd72a7-3385-48ef-bd42-f606fba81ae7",  
 "scope": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/resourceGroups/rg-demo09/providers/Microsoft.K  
eyVault/vaults/keyvault-demo09",  
 "type": "Microsoft.Authorization/roleAssignments"  
}  
 $  
 $ echo "Providing required permissions for MIC..."  
Providing required permissions for MIC...  
 $ az role assignment create --role "Managed Identity Operator" --assignee $aks.servicePrincipalProfile.client  
Id --scope $identity.id  
{  
 "canDelegate": null,  
 "id": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/resourcegroups/rg-demo09/providers/Microsoft.Mana  
gedIdentity/userAssignedIdentities/identity-aks-kv/providers/Microsoft.Authorization/roleAssignments/99b5b494-  
0fc0-463a-96d2-d391130b2c83",  
 "name": "99b5b494-0fc0-463a-96d2-d391130b2c83",  
 "principalId": "78e3f743-41dc-45c9-b42a-7fdf189e47b4",  
 "principalType": "ServicePrincipal",  
 "resourceGroup": "rg-demo09",  
 "roleDefinitionId": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/providers/Microsoft.Authorization/r  
oleDefinitions/f1a07417-d97a-45cb-824c-7a7467783830",  
 "scope": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/resourcegroups/rg-demo09/providers/Microsoft.M  
anagedIdentity/userAssignedIdentities/identity-aks-kv",  
 "type": "Microsoft.Authorization/roleAssignments"  
}  
 $  
 $ echo "Setting policy to access secrets in Key Vault..."  
Setting policy to access secrets in Key Vault...  
 $ az keyvault set-policy -n $keyVaultName --secret-permissions get --spn $identity.clientId  
{  
 "id": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/resourceGroups/rg-demo09/providers/Microsoft.KeyV  
ault/vaults/keyvault-demo09",  
 "location": "westeurope",  
 "name": "keyvault-demo09",  
 "properties": {  
 "accessPolicies": [  
 {  
 "applicationId": null,  
 "objectId": "9c1706eb-db28-4105-b993-f93a83f974b8",  
 "permissions": {  
 "certificates": [  
 "get",  
 "list",  
 "delete",  
 "create",  
 "import",  
 "update",  
 "managecontacts",  
 "getissuers",  
 "listissuers",  
 "setissuers",  
 "deleteissuers",  
 "manageissuers",  
 "recover"  
 ],  
 "keys": [  
 "get",  
 "create",  
 "delete",  
 "list",  
 "update",  
 "import",  
 "backup",  
 "restore",  
 "recover"  
 ],  
 "secrets": [  
 "get",  
 "list",  
 "set",  
 "delete",  
 "backup",  
 "restore",  
 "recover"  
 ],  
 "storage": [  
 "get",  
 "list",  
 "delete",  
 "set",  
 "update",  
 "regeneratekey",  
 "setsas",  
 "listsas",  
 "getsas",  
 "deletesas"  
 ]  
 },  
 "tenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47"  
 },  
 {  
 "applicationId": null,  
 "objectId": "99952e29-d98f-4b5c-968f-7d90c881027f",  
 "permissions": {  
 "certificates": null,  
 "keys": null,  
 "secrets": [  
 "get"  
 ],  
 "storage": null  
 },  
 "tenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47"  
 }  
 ],  
 "createMode": null,  
 "enablePurgeProtection": null,  
 "enableRbacAuthorization": false,  
 "enableSoftDelete": true,  
 "enabledForDeployment": false,  
 "enabledForDiskEncryption": null,  
 "enabledForTemplateDeployment": null,  
 "networkAcls": null,  
 "privateEndpointConnections": null,  
 "provisioningState": "Succeeded",  
 "sku": {  
 "name": "standard"  
 },  
 "softDeleteRetentionInDays": 7,  
 "tenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",  
 "vaultUri": "https://keyvault-demo09.vault.azure.net/"  
 },  
 "resourceGroup": "rg-demo09",  
 "tags": {},  
 "type": "Microsoft.KeyVault/vaults"  
}  
 $  
 $ echo "Assigning Reader Role to new Identity for Key Vault..."  
Assigning Reader Role to new Identity for Key Vault...  
 $ az role assignment create --role "Reader" --assignee $identity.principalId --scope $keyVault.id  
{  
 "canDelegate": null,  
 "id": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/resourceGroups/rg-demo09/providers/Microsoft.KeyV  
ault/vaults/keyvault-demo09/providers/Microsoft.Authorization/roleAssignments/03468c64-6b4b-407f-99c1-a91dd324  
beee",  
 "name": "03468c64-6b4b-407f-99c1-a91dd324beee",  
 "principalId": "99952e29-d98f-4b5c-968f-7d90c881027f",  
 "principalName": "8e98cd94-96c6-4cfa-a5e7-ca681fdeecb3",  
 "principalType": "ServicePrincipal",  
 "resourceGroup": "rg-demo09",  
 "roleDefinitionId": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/providers/Microsoft.Authorization/r  
oleDefinitions/acdd72a7-3385-48ef-bd42-f606fba81ae7",  
 "roleDefinitionName": "Reader",  
 "scope": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/resourceGroups/rg-demo09/providers/Microsoft.K  
eyVault/vaults/keyvault-demo09",  
 "type": "Microsoft.Authorization/roleAssignments"  
}  
 $  
 $ echo "Providing required permissions for MIC..."  
Providing required permissions for MIC...  
 $ az role assignment create --role "Managed Identity Operator" --assignee $aks.servicePrincipalProfile.client  
Id --scope $identity.id  
{  
 "canDelegate": null,  
 "id": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/resourcegroups/rg-demo09/providers/Microsoft.Mana  
gedIdentity/userAssignedIdentities/identity-aks-kv/providers/Microsoft.Authorization/roleAssignments/99b5b494-  
0fc0-463a-96d2-d391130b2c83",  
 "name": "99b5b494-0fc0-463a-96d2-d391130b2c83",  
 "principalId": "78e3f743-41dc-45c9-b42a-7fdf189e47b4",  
 "principalName": "http://aks-demo09SP-20200501213926",  
 "principalType": "ServicePrincipal",  
 "resourceGroup": "rg-demo09",  
 "roleDefinitionId": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/providers/Microsoft.Authorization/r  
oleDefinitions/f1a07417-d97a-45cb-824c-7a7467783830",  
 "roleDefinitionName": "Managed Identity Operator",  
 "scope": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/resourcegroups/rg-demo09/providers/Microsoft.M  
anagedIdentity/userAssignedIdentities/identity-aks-kv",  
 "type": "Microsoft.Authorization/roleAssignments"  
}  
 $  
 $ echo "Setting policy to access secrets in Key Vault..."  
Setting policy to access secrets in Key Vault...  
 $ az keyvault set-policy -n $keyVaultName --secret-permissions get --spn $identity.clientId  
{  
 "id": "/subscriptions/4b72ed90-7ca3-4e76-8d0f-31a2c0bee7a3/resourceGroups/rg-demo09/providers/Microsoft.KeyV  
ault/vaults/keyvault-demo09",  
 "location": "westeurope",  
 "name": "keyvault-demo09",  
 "properties": {  
 "accessPolicies": [  
 {  
 "applicationId": null,  
 "objectId": "9c1706eb-db28-4105-b993-f93a83f974b8",  
 "permissions": {  
 "certificates": [  
 "get",  
 "list",  
 "delete",  
 "create",  
 "import",  
 "update",  
 "managecontacts",  
 "getissuers",  
 "listissuers",  
 "setissuers",  
 "deleteissuers",  
 "manageissuers",  
 "recover"  
 ],  
 "keys": [  
 "get",  
 "create",  
 "delete",  
 "list",  
 "update",  
 "import",  
 "backup",  
 "restore",  
 "recover"  
 ],  
 "secrets": [  
 "get",  
 "list",  
 "set",  
 "delete",  
 "backup",  
 "restore",  
 "recover"  
 ],  
 "storage": [  
 "get",  
 "list",  
 "delete",  
 "set",  
 "update",  
 "regeneratekey",  
 "setsas",  
 "listsas",  
 "getsas",  
 "deletesas"  
 ]  
 },  
 "tenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47"  
 },  
 {  
 "applicationId": null,  
 "objectId": "99952e29-d98f-4b5c-968f-7d90c881027f",  
 "permissions": {  
 "certificates": null,  
 "keys": null,  
 "secrets": [  
 "get"  
 ],  
 "storage": null  
 $  
 $ echo "Deploying a Nginx Pod for testing..."  
Deploying a Nginx Pod for testing...  
 $ $nginxPod = @"  
>> kind: Pod  
>> apiVersion: v1  
>> metadata:  
>> name: nginx-secrets-store  
>> labels:  
>> aadpodidbinding: $($identitySelector)  
>> spec:  
>> containers:  
>> - name: nginx  
>> image: nginx  
>> volumeMounts:  
>> - name: secrets-store-inline  
>> mountPath: "/mnt/secrets-store"  
>> readOnly: true  
>> volumes:  
>> - name: secrets-store-inline  
>> csi:  
>> driver: secrets-store.csi.k8s.io  
>> readOnly: true  
>> volumeAttributes:  
>> secretProviderClass: $($secretProviderClassName)  
>> "@  
 $ $nginxPod | kubectl apply -f -  
pod/nginx-secrets-store created  
 $  
 $ kubectl get pods  
NAME READY STATUS RESTARTS AGE  
mic-76dd75ddf9-58vxq 1/1 Running 0 9m47s  
mic-76dd75ddf9-rtfcn 1/1 Running 0 9m47s  
nginx-secrets-store 1/1 Running 0 88s  
nmi-6s6zh 1/1 Running 0 9m48s  
nmi-r2pdr 1/1 Running 0 9m48s  
nmi-wv2sx 1/1 Running 0 9m48s  
 $  
 $ echo "Validating the pod has access to the secrets from Key Vault..."  
Validating the pod has access to the secrets from Key Vault...  
 $ kubectl exec -it nginx-secrets-store ls /mnt/secrets-store/  
DATABASE\_LOGIN DATABASE\_PASSWORD  
 $ kubectl exec -it nginx-secrets-store cat /mnt/secrets-store/DATABASE\_LOGIN  
Houssem  
 $ kubectl exec -it nginx-secrets-store cat /mnt/secrets-store/$secret1Alias  
Houssem  
 $ kubectl exec -it nginx-secrets-store cat /mnt/secrets-store/DATABASE\_PASSWORD  
P@ssword123456  
 $ kubectl exec -it nginx-secrets-store cat /mnt/secrets-store/$secret2Alias  
P@ssword123456  
 $