# Document to create required Azure AD Server and Azure AD Client Apps in Azure AD

https://docs.microsoft.com/en-us/azure/aks/azure-ad-integration

#Variables

```
NetworkRG=”RG-Process-Fussion"

region="East US"

vnet=RG-Process-Fussion-Vnet

subnet=Frontendsubnet1

az account set --subscription "1a317aed-2ae0-467e-a97d-46ea4e51f919"

SP_ID= f0e707b4-d71e-472f-95f9-044522f4b47e

SP_PASSWORD= 8Jj3ucX9f.7h_ybbCMtxDHBPuFucxR=.

AKSRG="RG-Process-Fussion"

Clustername=frontendakscluster
```

#Get VNET and Subnet IDs

```
VNET_ID=$(az network vnet show --resource-group $NetworkRG --name $vnet --query id -o tsv)

SUBNET_ID=$(az network vnet subnet show --resource-group $NetworkRG --vnet-name $vnet --name $subnet --query id -o tsv)
```

#Assign Contributor Role to SPN on VNET

```
az role assignment create --assignee $SP_ID --scope $VNET_ID --role Contributor
```

#Variables for Azure AD Integration

```
add-server= 908122b6-46eb-48d3-b524-84b332c9fc19

aad_client= f0e707b4-d71e-472f-95f9-044522f4b47e

aad_tenant= 4f02b584-f3fb-4aa8-ad45-151db3b6a408

aks-Secret= 8Jj3ucX9f.7h_ybbCMtxDHBPuFucxR=.
```

#Create Azure AD Integrated and Managed Identities Enabled AKS Cluster

```
az aks create --resource-group $AKSRG --name $Clustername --kubernetes-version 1.15.7 --load-balancer-sku basic --node-count 2 --generate-ssh-keys --network-plugin azure --service-cidr 172.35.0.0/16  --dns-service-ip 172.35.0.10 --docker-bridge-address 172.17.0.1/16 --vnet-subnet-id $SUBNET_ID --service-principal $SP_ID --client-secret $SP_PASSWORD --aad-server-app-id ccfed16e-34ac-4d9f-81b6-c1defd719dc2 --aad-client-app-id 9c841d5e-6047-44c7-9d3e-1e327199b6df  --aad-server-app-secret d5Bg8kxG1Dti.Y@uvW=bS0:oBkI2o86H --aad-tenant-id fb52a3f0-8cec-44db-be1a-19a597ce73bc  --network-policy calico --enable-managed-identity
```

# Confirm AKS Cluster is RBAC Enabled

```
az resource show -g $AKSRG -n $Clustername --resource-type Microsoft.ContainerService/ManagedClusters --query properties.enableRBAC
```

# Configure AKS RBAC

#Create Groups

```
APPDEV_ID=$(az ad group create --display-name appdev --mail-nickname appdev --query objectId -o tsv)
```

```
OPSSRE_ID=$(az ad group create --display-name opssre --mail-nickname opssre --query objectId -o tsv)
```

```
cluster_admin=$(az ad group create --display-name aksclsadmin --mail-nickname aksclsadmin --query objectId -o tsv)
```

"Get AKS Resource ID

```
AKS_ID=$(az aks show --resource-group myaksrg --name $Clustername --query id -o tsv)
```

```
AKS_ID=$(az aks show --resource-group na-sandbox-atul --name atulaksmsi --query id -o tsv)
```

#Azure role assignment

```
az role assignment create --assignee $APPDEV_ID --role "Azure Kubernetes Service Cluster User Role" --scope $AKS_ID
az role assignment create --assignee $OPSSRE_ID --role "Azure Kubernetes Service Cluster User Role" --
scope $AKS_ID
```

#Create Users

```
AKSDEV_ID=$(az ad user create --display-name "AKS Dev" --user-principal-name
aksdev@azatlab.onmicrosoft.com --password P@ssw0rd1 --query objectId -o tsv)

AKSSRE_ID=$(az ad user create --display-name "AKS SRE" --user-principal-name
akssre@azatlab.onmicrosoft.com --password P@ssw0rd1 --query objectId -o tsv)
```

#Add Members to Groups

```
az ad group member add --group appdev --member-id $AKSDEV_ID

az ad group member add --group opssre --member-id $AKSSRE_ID
```

#Create AKS NameSpace

```
az aks get-credentials -g $AKSRG -n $Clustername --admin

kubectl create namespace dev

kubectl create namespace sre
```

#Create Role for DEV Name Space

```
kubectl apply -f role-dev-namespace.yaml
```

#Get Resource ID for AppDev Group

```
az ad group show --group appdev --query objectId -o tsv
```

#create a RoleBinding for the appdev group

```
kubectl apply -f rolebinding-dev-namespace.yaml
```

#Create Role for SRE Name Space

```
kubectl apply -f role-sre-namespace.yaml
```

#Get Resource ID for OPSSRE Group

```
az ad group show --group opssre --query objectId -o tsv
```

#create a RoleBinding for the opssre group

```
kubectl apply -f rolebinding-sre-namespace.yaml
```

# Assign Cluster Admin Role to aksclsadmin group

```
kubectl apply -f rbac-aad-group.yaml
```

#Test

```
az aks get-credentials -g $AKSRG -n $Clustername --overwrite-existing
```

#create NGINX in DEV NameSpace

```
kubectl run --generator=run-pod/v1 nginx-dev --image=nginx --namespace dev  #sign in with aksdev@azatlab.onmicrosoft.com

Kubectl get pods -n dev
```

# Trying to perform a task that user has no rights

```
kubectl get pods --all-namespaces     #Error Expected

kubectl run --generator=run-pod/v1 nginx-dev --image=nginx --namespace sre  #Error Expected
```

#create NGINX in SRE NameSpace

```
az aks get-credentials -g $AKSRG -n $Clustername --overwrite-existing

kubectl run --generator=run-pod/v1 nginx-dev --image=nginx --namespace sre  #sign in with akssre@azatlab.onmicrosoft.com
```

```
Kubectl get pods -n dev
```

```
kubectl get pods --all-namespaces     #Error Expected
```

```
kubectl run --generator=run-pod/v1 nginx-dev --image=nginx --namespace sre  #Error Expected
```

#POD Managed Identity

#Create new API and MIC (Managed Identity Container)

```
kubectl apply -f https://raw.githubusercontent.com/Azure/aad-pod-identity/master/deploy/infra/deployment-rbac.yaml
```

#create AKS Managed Identity

```
az identity create -g myaksrg -n aksuser -o json
```

#Output

```
 {
  "clientId": "1ade0be6-d13e-4de5-9e8c-87b2c1a25890",
  "clientSecretUrl": "https://control-eastus.identity.azure.net/subscriptions/6c3152be-eea2-4910-9368-288b491a5b02/resourcegroups/MC_NA-SANDBOX-ATUL_atulaksmsi_eastus/providers/Microsoft.ManagedIdentity/userAssignedIdentities/aksuser/credentials?tid=fb52a3f0-8cec-44db-be1a-19a597ce73bc&oid=5de22a50-356c-4ba4-945b-500d134291fd&aid=1ade0be6-d13e-4de5-9e8c-87b2c1a25890",
  "id": "/subscriptions/6c3152be-eea2-4910-9368-288b491a5b02/resourcegroups/MC_NA-SANDBOX-ATUL_atulaksmsi_eastus/providers/Microsoft.ManagedIdentity/userAssignedIdentities/aksuser",
  "location": "eastus",
  "name": "aksuser",
  "principalId": "5de22a50-356c-4ba4-945b-500d134291fd",
  "resourceGroup": "MC_NA-SANDBOX-ATUL_atulaksmsi_eastus",
  "tags": {},
  "tenantId": "fb52a3f0-8cec-44db-be1a-19a597ce73bc",
  "type": "Microsoft.ManagedIdentity/userAssignedIdentities"
}
```

```
kubectl apply -f aadpodidentity-lab.yaml
```

# Create AAD POD Identity Binding

```
kubectl apply -f aadpodidentitybinding.yaml

    # Get AKS Ckuster Resource Client ID

az aks show -g na-sandbox-atul -n atulaksmsi --query identityProfile.kubeletidentity.clientId -o tsv


    # Configure RBAC for AKS Managed ID for the VMSS


az role assignment create --role "Virtual Machine Contributor" --assignee 2c5b4a69-67c3-4eb2-b64e-19dba43e6bec --scope /subscriptions/6c3152be-eea2-4910-9368-288b491a5b02/resourcegroups/MC_NA-sandbox-atul_atulaksmsi_eastus

az role assignment create --role "Managed Identity Operator" --assignee 2c5b4a69-67c3-4eb2-b64e-19dba43e6bec --scope /subscriptions/6c3152be-eea2-4910-9368-288b491a5b02/resourcegroups/MC_NA-sandbox-atul_atulaksmsi_eastus

az role assignment create --role "Managed Identity Operator" --assignee 2c5b4a69-67c3-4eb2-b64e-19dba43e6bec --scope /subscriptions/6c3152be-eea2-4910-9368-288b491a5b02/resourcegroups/MC_NA-sandbox-atul_atulaksmsi_eastus/providers/Microsoft.ManagedIdentity/userAssignedIdentities/aksuser

az role assignment create --role "Virtual Machine Contributor" --assignee 2c5b4a69-67c3-4eb2-b64e-19dba43e6bec  --scope /subscriptions/6c3152be-eea2-4910-9368-288b491a5b02/resourcegroups/NA-Sandbox-Atul

az role assignment create --role "Managed Identity Operator" --assignee 2c5b4a69-67c3-4eb2-b64e-19dba43e6bec  --scope /subscriptions/6c3152be-eea2-4910-9368-288b491a5b02/resourcegroups/NA-Sandbox-Atul

# Install aks-hellO AND AKS Ingress Demo APP

helm repo add azure-samples https://azure-samples.github.io/helm-charts/

helm install azure-samples/aks-helloworld --generate-name

helm install azure-samples/aks-helloworld --set title="AKS Ingress Demo" --set serviceName="ingress-demo" --generate-name

Kubectl get pods


# Label Pod to bind POD Managed Identity


kubectl label pod <AKS_Hello Pod Name> aadpodidbinding=msienabled
```

# Checking (Troubleshooting Steps)

```
kubectl get AzureIdentity --all-namespaces -o yaml

kubectl get AzureIdentityBinding --all-namespaces -o yaml

kubectl get AzureAssignedIdentities --all-namespaces -o yaml   #shall show assigned Managed Identity to Pods <AKS -Hello>

Kubectl exec -it <AKS Hello POD> bash
```

# Get Key Vault Access Token

```
apt update

apt install jq
```

#Check get Access Token for Key Vault using POD Managed Identity

```
curl 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net' -H Metadata:true
```

# get Key Vault Access Token as a Variable

```
token=$(curl 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net' -H Metadata:true | jq -r '.access_token')

# Get Secret from Key Vault

curl https://pvt1vault.vault.azure.net//secrets/aks?api-version=2016-10-01 -H "Authorization: Bearer $token"
```

YAML Files

Yaml Files