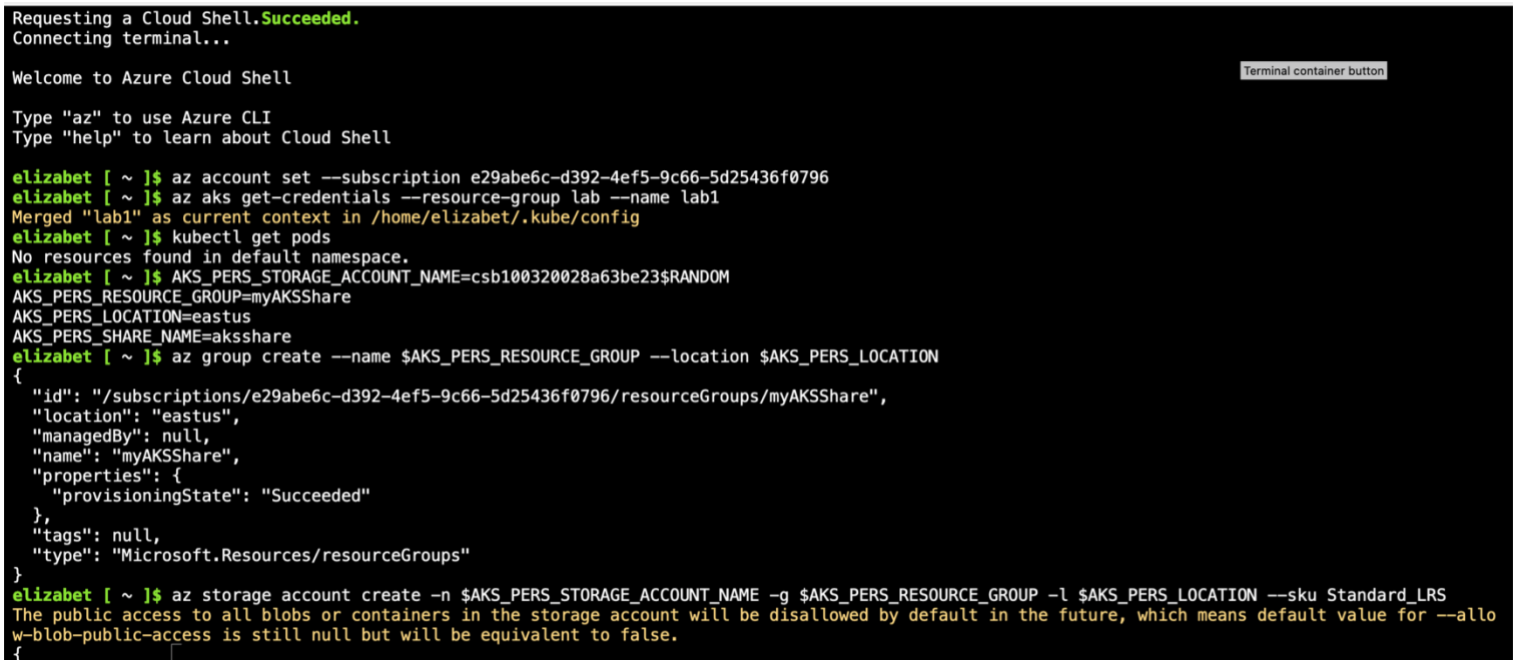
SFA Homework #17
Elizabet Ivanova

Exercise: Storage

Practice 1: Direct provisioning of Azure File storage

1. Connecting to the AKS cluster, checking for pods in the default namespace, and creating the Azure File share:



2. Getting the account name and key, creating the secret (I replaced the names, and then realized I shouldn't have), and verifying the secret has been created:

```
  "statusOfPrimary": "available",
  "statusOfSecondary": null,
  "storageAccountSkuConversionStatus": null,
  "tags": {},
  "type": "Microsoft.Storage/storageAccounts"
}
elizabet [ ~ ]$ export AZURE_STORAGE_CONNECTION_STRING=$(az storage account show-connection-string -n $AKS_PERS_STORAGE_ACCOUNT_NAME -g $AKS_PERS_RES
OURCE_GROUP -otsv)
elizabet [ ~ ]$ az storage share create -n $AKS_PERS_SHARE_NAME --connection-string $AZURE_STORAGE_CONNECTION_STRING
{
  "created": true
}
elizabet [ ~ ]$ STORAGE_KEY=$(az storage account keys list --resource-group $AKS_PERS_RESOURCE_GROUP --account-name $AKS_PERS_STORAGE_ACCOUNT_NAME --
query "[0].value" -otsv)
elizabet [ ~ ]$ echo Storage account name: $AKS_PERS_STORAGE_ACCOUNT_NAME
echo Storage account key: $STORAGE_KEY
Storage account name: csb100320028a63be2311759
Storage account key: sKtcXg2DhI+pZ94MlqFZPtpxzevI4EUa0YQ2FklHuilS9DKOuAOGjy9b69qim+HrSZdDxVuLjj7H+ASt6aGexQ==
elizabet [ ~ ]$ kubectl create secret generic azure-secret --from-literal=csb100320028a63be2311759=$AKS_PERS_STORAGE_ACCOUNT_NAME --from-literal=sKtc
Xg2DhI+pZ94MlqFZPtpxzevI4EUa0YQ2FklHuilS9DKOuAOGjy9b69qim+HrSZdDxVuLjj7H+ASt6aGexQ===$STORAGE_KEY
error: "sKtcXg2DhI+pZ94MlqFZPtpxzevI4EUa0YQ2FklHuilS9DKOuAOGjy9b69qim+HrSZdDxVuLjj7H+ASt6aGexQ" is not valid key name for a Secret a valid config key
 must consist of alphanumeric characters, '-', '_' or '.' (e.g. 'key.name',  or 'KEY_NAME',  or 'key-name', regex used for validation is '[-._a-zA-Z0
-9]+')
elizabet [ ~ ]$ kubectl create secret generic azure-secret --from-literal=azurestorageaccountname=$AKS_PERS_STORAGE_ACCOUNT_NAME --from-literal=azure
storageaccountkey=$STORAGE_KEY
secret/azure-secret created
elizabet [ ~ ]$ kubectl get secret -A
NAMESPACE     NAME                     TYPE                            DATA   AGE
default       azure-secret             Opaque                          2      19s
kube-system   ama-logs-secret          Opaque                          2      15m
kube-system   bootstrap-token-r0mzrl   bootstrap.kubernetes.io/token   4      15m
kube-system   konnectivity-certs       Opaque                          3      15m
elizabet [ ~ ]$
```

3. Creating mypod and describing it:

```
elizabet [ ~ ]$ kubectl apply -f azure-files-pod.yaml
pod/mypod created
elizabet [ ~ ]$ kubectl describe pod mypod
Name:             mypod
Namespace:        default
Priority:         0
Service Account:  default
Node:             aks-agentpool-10604515-vmss000000/10.224.0.4
Start Time:       Thu, 06 Apr 2023 09:50:28 +0000
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.0.17
IPs:
  IP:  10.244.0.17
Containers:
  mypod:
    Container ID:   containerd://baa422ae23093d038807f457d44258bd08507c7eb2651b5eed0c123f43c8c24c
    Image:          mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine
    Image ID:       mcr.microsoft.com/oss/nginx/nginx@sha256:f84780a5ad654515bcd9ba2f35e20935e1246799f198683dd2c4f74d19ae9e5e
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 06 Apr 2023 09:50:31 +0000
    Ready:          True
    Restart Count:  0
    Limits:
      cpu:     250m
      memory:  256Mi
    Requests:
      cpu:        100m
      memory:     128Mi
    Environment:  <none>
```

4. Checking that the share has been mounted successfully:

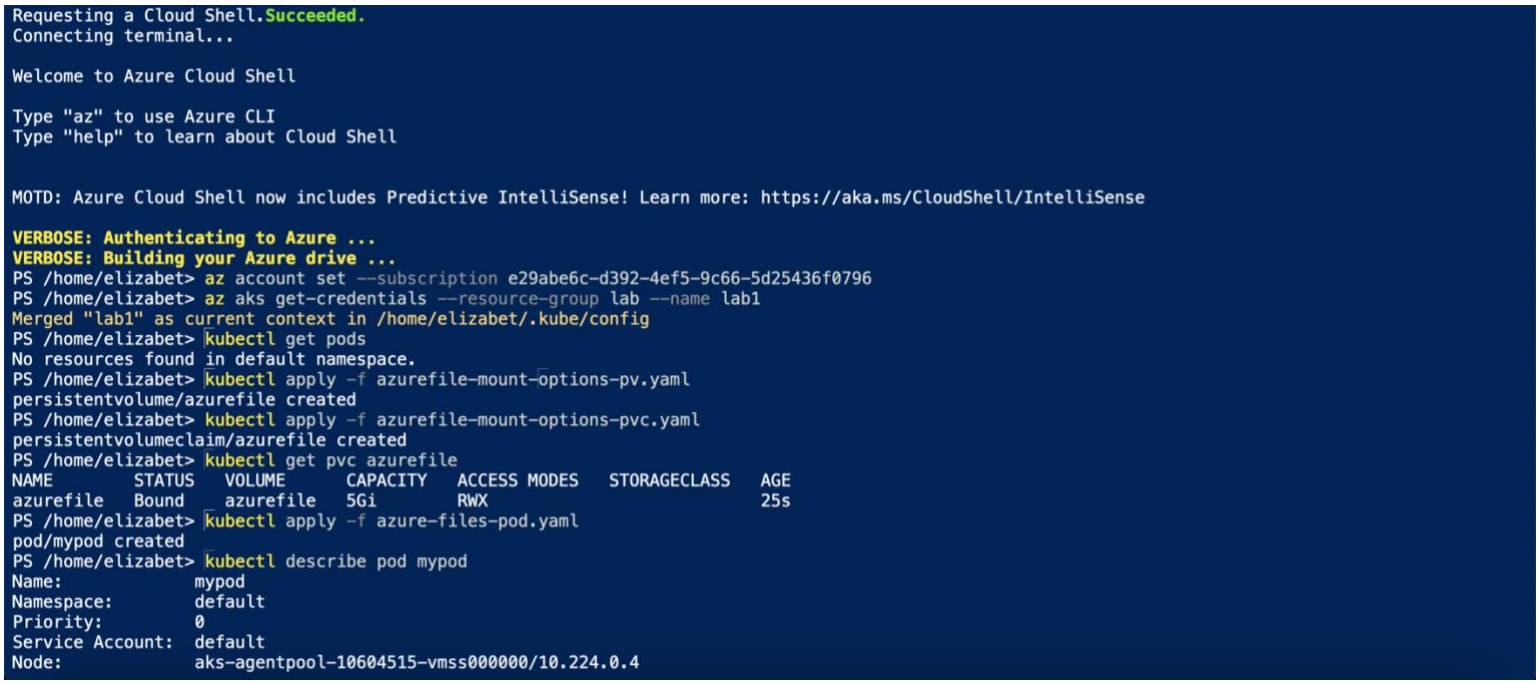
```
    Requests:
      cpu:        100m
      memory:     128Mi
    Environment:  <none>
    Mounts:
      /mnt/azure from azure (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-8k4sn (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  azure:
    Type:        AzureFile (an Azure File Service mount on the host and bind mount to the pod)
    SecretName:  azure-secret
    ShareName:   aksshare
    ReadOnly:    false
  kube-api-access-8k4sn:
    Type:                     Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:   3607
    ConfigMapName:            kube-root-ca.crt
    ConfigMapOptional:        <nil>
    DownwardAPI:              true
QoS Class:                    Burstable
Node-Selectors:               <none>
Tolerations:                  node.kubernetes.io/memory-pressure:NoSchedule op=Exists
                              node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                              node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
```

5. Creating a test file inside and checking if it is there; deleting the pod.

```
    SecretName:  azure-secret
    ShareName:   aksshare
    ReadOnly:    false
  kube-api-access-8k4sn:
    Type:                     Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:   3607
    ConfigMapName:            kube-root-ca.crt
    ConfigMapOptional:        <nil>
    DownwardAPI:              true
QoS Class:                    Burstable
Node-Selectors:               <none>
Tolerations:                  node.kubernetes.io/memory-pressure:NoSchedule op=Exists
                              node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                              node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  31s   default-scheduler  Successfully assigned default/mypod to aks-agentpool-10604515-vmss000000
  Normal  Pulling    30s   kubelet            Pulling image "mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine"
  Normal  Pulled     29s   kubelet            Successfully pulled image "mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine" in 1.090863119s
  Normal  Created    29s   kubelet            Created container mypod
  Normal  Started    28s   kubelet            Started container mypod
elizabet [ ~ ]$ kubectl exec -it mypod --bash
error: unknown flag: --bash
See 'kubectl exec --help' for usage.
elizabet [ ~ ]$ kubectl exec -it mypod -- bash
error: Internal error occurred: error executing command in container: failed to exec in container: failed to start exec "f50a3c71f95444f4e761c080172a
77f3e69a0d6edc30a36e1d1411ef943e9de4": OCI runtime exec failed: exec failed: container_linux.go:380: starting container process caused: exec: "bash":
 executable file not found in $PATH: unknown
elizabet [ ~ ]$ kubectl exec -it mypod -- sh
/ # cd /mnt/azure
/mnt/azure # touch test.txt
/mnt/azure # ls
test.txt
/mnt/azure # exit
elizabet [ ~ ]$ kubectl delete pod mypod
pod "mypod" deleted
elizabet [ ~ ]$
```

Practice 2: Provisioning Azure File storage using PVs and PVCs

1. *forgot to switch to bash so this section is in PS*
2. Applying the pv and pvc files; verifying that the PersistentVolumeClaim is created and bound to the PersistentVolume; creating and describing mypod.

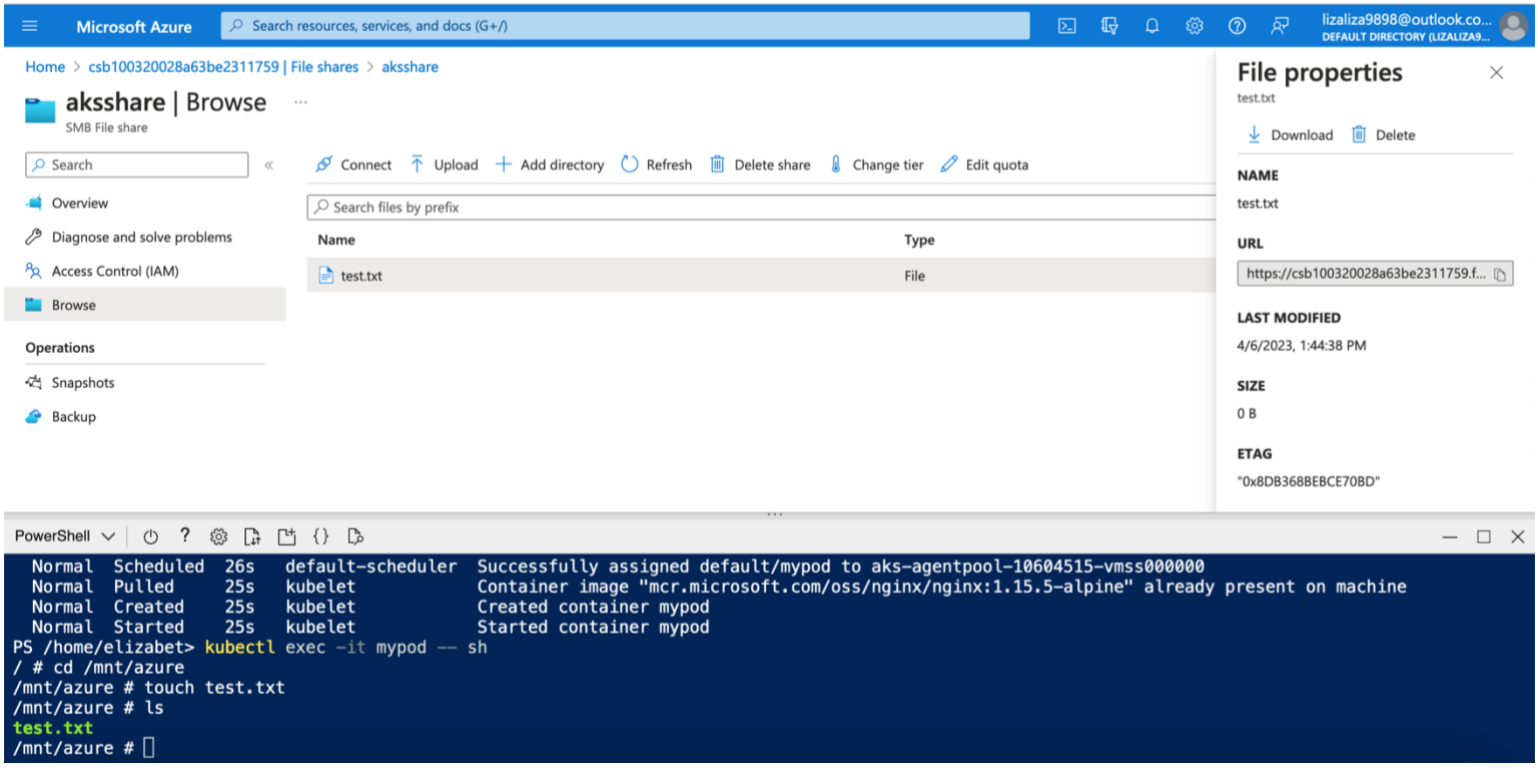


**and yes, Azure Files supports multiple access modes such as *ReadOnlyMany* and *ReadWriteOnce*.

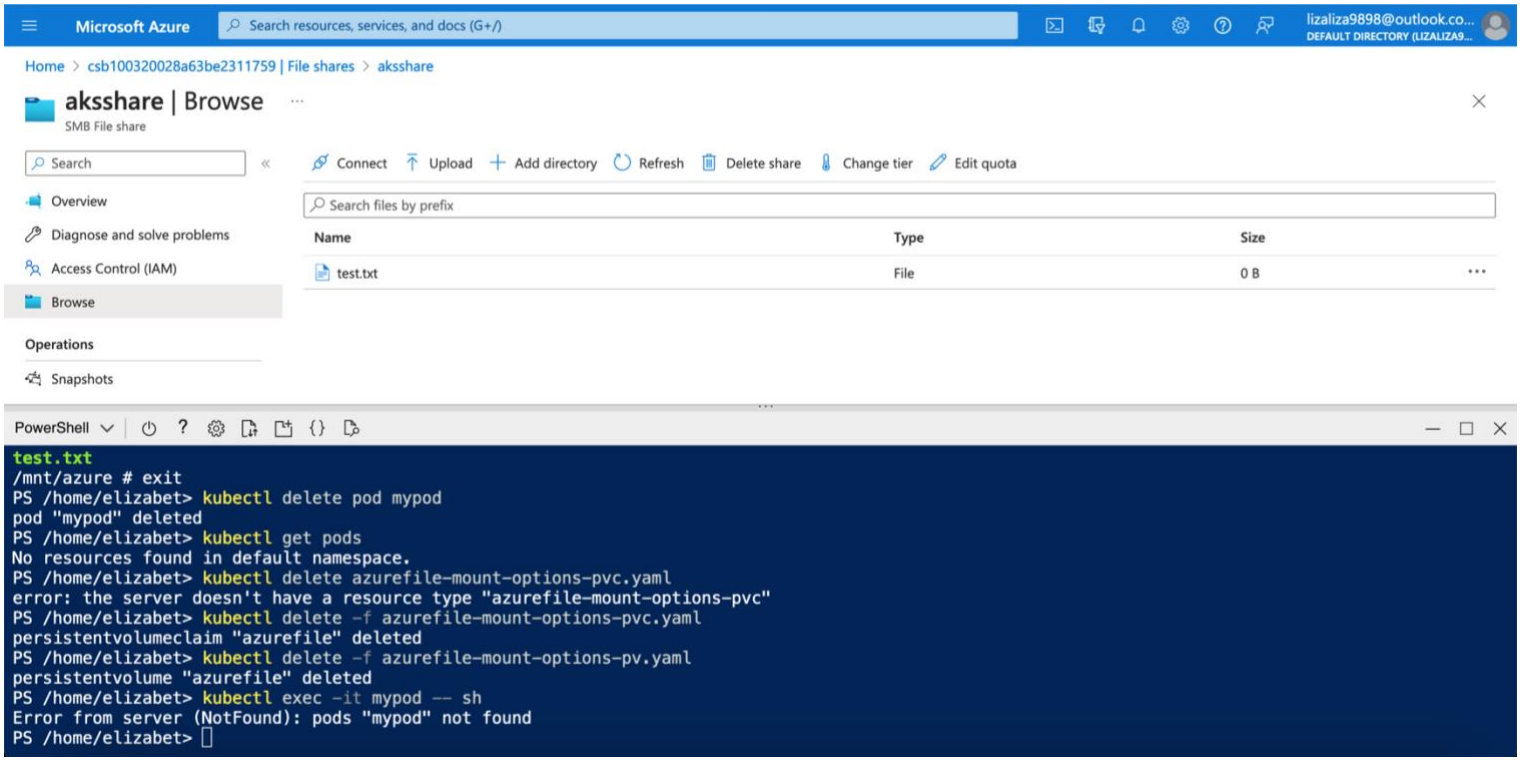
3. Verifying that the share has been mounted successfully:

```
    Mounts:
      /mnt/azure from azure (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-hrsvh (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  azure:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  azurefile
    ReadOnly:   false
  kube-api-access-hrsvh:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   Burstable
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/memory-pressure:NoSchedule op=Exists
                             node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  26s   default-scheduler  Successfully assigned default/mypod to aks-agentpool-10604515-vmss000000
  Normal  Pulled     25s   kubelet            Container image "mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine" already present on machine
  Normal  Created    25s   kubelet            Created container mypod
  Normal  Started    25s   kubelet            Started container mypod
PS /home/elizabet>
```

4. Creating the blank text file and checking if it is there:



5. Deleting everything but the test.txt is still in the portal.



Practice 3: Provisioning Azure file storage using Storage Classes

1. Applying the yaml files, creating the storage and PVC, checking the status, creating and describing mypod:

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

elizabet [ ~ ]$ az account set --subscription e29abe6c-d392-4ef5-9c66-5d25436f0796
elizabet [ ~ ]$ az aks get-credentials --resource-group lab --name lab1
Merged "lab1" as current context in /home/elizabet/.kube/config
elizabet [ ~ ]$ kubectl get pods
No resources found in default namespace.
elizabet [ ~ ]$ kubectl apply -f azure-file-sc.yaml
storageclass.storage.k8s.io/my-azurefile created
elizabet [ ~ ]$ kubectl apply -f azure-file-pvc.yaml
persistentvolumeclaim/my-azurefile created
elizabet [ ~ ]$ kubectl get pvc my-azurefile
NAME           STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
my-azurefile   Bound    pvc-f2dc9a66-484b-4c84-8ec6-607f778716cc   5Gi        RWX            my-azurefile   34s
elizabet [ ~ ]$ kubectl apply -f azure-pvc-files.yaml
pod/mypod created
elizabet [ ~ ]$ kubectl describe pod mypod
Name:             mypod
Namespace:        default
Priority:         0
Service Account:  default
Node:             aks-agentpool-10604515-vmss000000/10.224.0.4
Start Time:       Thu, 06 Apr 2023 11:11:58 +0000
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.0.19
IPs:
  IP:  10.244.0.19
Containers:
  mypod:
    Container ID:   containerd://4c66ef3e84d28558ac6a669355f29cee5cb1664394fe0fa0ef07b0aa1505c877
```

2. Checking the volumes:

```
    Mounts:
      /mnt/azure from volume (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-fxqpb (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  volume:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  my-azurefile
    ReadOnly:   false
  kube-api-access-fxqpb:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   Burstable
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/memory-pressure:NoSchedule op=Exists
                             node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  14s   default-scheduler  Successfully assigned default/mypod to aks-agentpool-10604515-vmss000000
  Normal  Pulled     13s   kubelet            Container image "mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine" already present on machine
  Normal  Created    13s   kubelet            Created container mypod
  Normal  Started    13s   kubelet            Started container mypod
elizabet [ ~ ]$
```

Practice 4: Direct provisioning of Azure Disk storage

1. Forgot to replace the name but then I fixed it; creating the disk, applying the files, and describing the pod:

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

elizabet [ ~ ]$ az account set --subscription e29abe6c-d392-4ef5-9c66-5d25436f0796
elizabet [ ~ ]$ az aks get-credentials --resource-group lab --name lab1
Merged "lab1" as current context in /home/elizabet/.kube/config
elizabet [ ~ ]$ kubectl get pods
No resources found in default namespace.
elizabet [ ~ ]$ az aks show --resource-group myResourceGroup --name myAKSCluster --query nodeResourceGroup -o tsv
(ResourceGroupNotFound) Resource group 'myResourceGroup' could not be found.
Code: ResourceGroupNotFound
Message: Resource group 'myResourceGroup' could not be found.
elizabet [ ~ ]$ az aks show --resource-group lab --name lab1 --query nodeResourceGroup -o tsv
MC_lab_lab1_eastus
elizabet [ ~ ]$ az disk create \
    --resource-group MC_lab_lab1_eastus \
    --name myAKSDisk \
    --size-gb 20 \
    --query id \
    --output tsv
/subscriptions/e29abe6c-d392-4ef5-9c66-5d25436f0796/resourceGroups/MC_lab_lab1_eastus/providers/Microsoft.Compute/disks/myAKSDisk
elizabet [ ~ ]$ kubectl apply -f azure-disk-pod.yaml
error: error parsing azure-disk-pod.yaml: error converting YAML to JSON: yaml: line 23: did not find expected '-' indicator
elizabet [ ~ ]$ kubectl apply -f azure-disk-pod.yaml
error: error parsing azure-disk-pod.yaml: error converting YAML to JSON: yaml: line 23: did not find expected '-' indicator
elizabet [ ~ ]$ kubectl apply -f azure-disk-pod.yaml
pod/mypod created
elizabet [ ~ ]$ kubectl describe pod mypod
Name:             mypod
Namespace:        default
Priority:         0
Service Account:  default
Node:             aks-agentpool-10604515-vmss000000/10.224.0.4
```

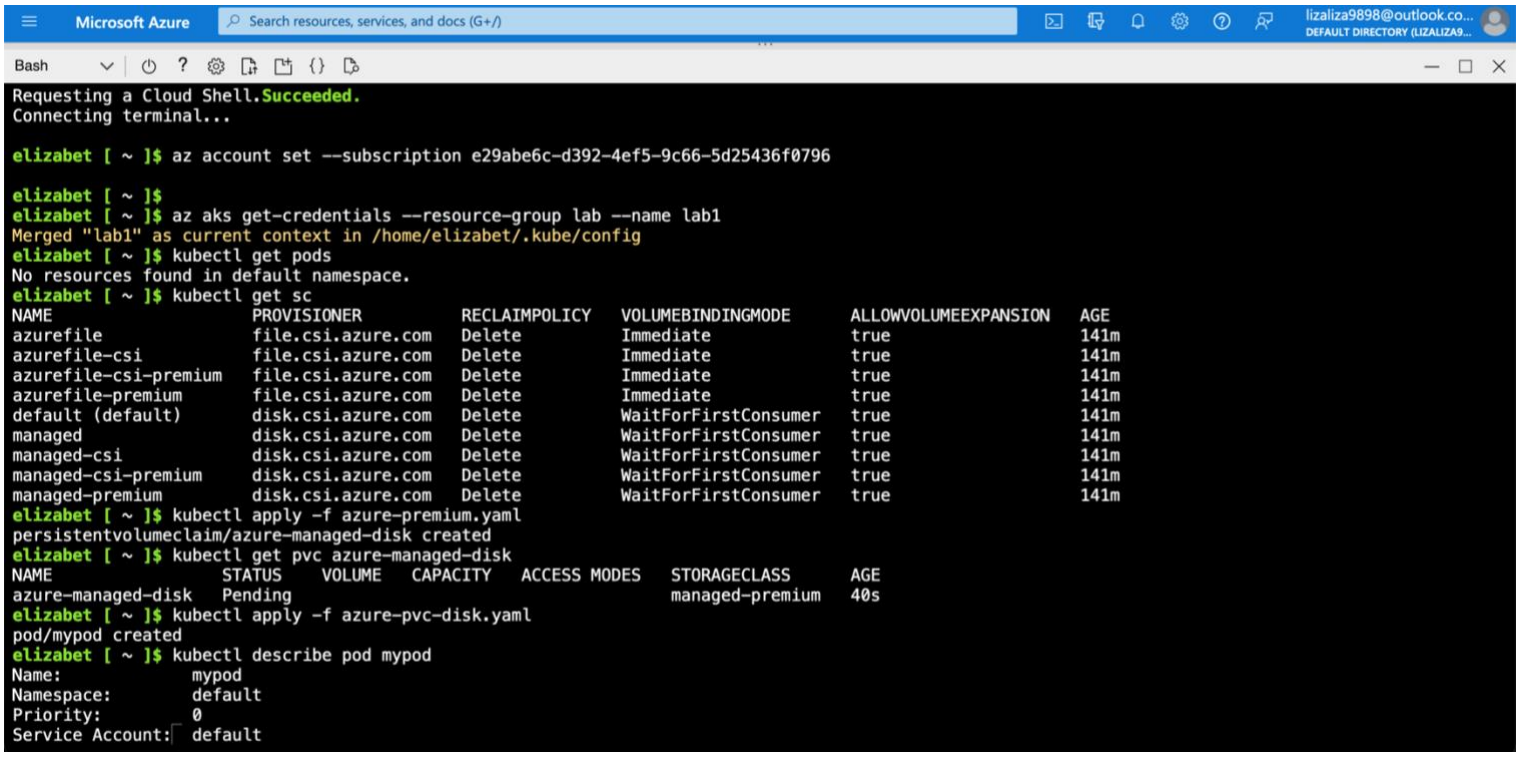
2. Checking the volume:

```
    Mounts:
      /mnt/azure from azure (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-mnpk9 (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             False
  ContainersReady   False
  PodScheduled      True
Volumes:
  azure:
    Type:         AzureDisk (an Azure Data Disk mount on the host and bind mount to the pod)
    DiskName:     myAKSDisk
    DiskURI:      /subscriptions/e29abe6c-d392-4ef5-9c66-5d25436f0796/resourceGroups/MC_lab_lab1_eastus/providers/Microsoft.Compute/disks/myAKSDisk
    Kind:         Managed
    FSType:       ext4
    CachingMode:  ReadWrite
    ReadOnly:     false
  kube-api-access-mnpk9:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   Burstable
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/memory-pressure:NoSchedule op=Exists
                             node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason            Age              From                Message
  ----     ------            ----             ----                -------
  Normal   Scheduled         28s              default-scheduler   Successfully assigned default/mypod to aks-agentpool-10604515-vmss000000
```

Terminal container button

Practice 5: Provisioning AzureDisk storage using Storage Classes

1. Connecting to AKS, getting the storage classes, applying the yaml files, and descrbing the pod:



2. Checking the volumes: