Installing Pod Security Policies on Azure K8S

Install aks-preview CLI extension

To use pod security policies, you need the *aks-preview* CLI extension version 0.4.1 or higher. Install the *aks-preview* Azure CLI extension using the az extension add command, then check for any available updates using the az extension update command:

```
$ az extension add --name aks-preview
The installed extension 'aks-preview' is in preview.

$ az extension update --name aks-preview
No updates available for 'aks-preview'. Use --debug for more information.

$ az feature register --name PodSecurityPolicyPreview --namespace Microsoft.ContainerService
Once the feature 'PodSecurityPolicyPreview' is registered, invoking 'az provider register -n Microsoft.
ContainerService' is required to get the change propagated
{
  "id": "/subscriptions/b63b61a0-605d-47e8-b8a6-598e188a00ed/providers/Microsoft.Features/providers/Microsoft.
ContainerService/features/PodSecurityPolicyPreview",
  "name": "Microsoft.ContainerService/PodSecurityPolicyPreview",
  "properties": {
    "state": "Registering"
  },
  "type": "Microsoft.Features/providers/features"
}
```

Register pod security policy feature provider

To create or update an AKS cluster to use pod security policies, first enable a feature flag on your subscription. To register the *PodSecurityPolicyPreview* feature flag, use the az feature register command as shown in the following example:

It takes a few minutes for the status to show *Registered*. You can check on the registration status using the az feature listcommand:

```
az feature list -o table --query "[?contains(name, 'Microsoft.ContainerService/PodSecurityPolicyPreview')].
{Name:name,State:properties.state}"
```

```
Name                                                State
--------------------------------------------------  ----------
Microsoft.ContainerService/PodSecurityPolicyPreview  Registered
```

When ready, refresh the registration of the *Microsoft.ContainerService* resource provider using the az provider registercommand:

```
az provider register --namespace Microsoft.ContainerService
```

Enable pod security policy on an AKS cluster

```
$ az aks update \
    --resource-group <RESOURCE_GROUP> \
    --name <K8S_CLUSTER> \
    --enable-pod-security-policy
```

Example:

```
$ az aks update \
    --resource-group ncyd-perftest8-rg-onprem \
    --name ncyd-perftest8-aks-cluster-onprem \
    --enable-pod-security-policy
```

```
The behavior of this command has been altered by the following extension: aks-preview
 | Running ..
```

```
...

  "workloadAutoScalerProfile": {
    "keda": null,
    "verticalPodAutoscaler": null
  }
}
```

Delete Azure PSPs

```
kubectl delete psp privileged
kubectl delete clusterrole/psp:privileged
```

Install Restrictive Pod Security Policies

```
vi psp-restrictive.yaml
```

```
---
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: privileged
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: "*"
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - "*"
  volumes:
  - "*"
  hostNetwork: true
```

```
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    rule: 'RunAsAny'
---
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: restricted
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
spec:
  privileged: false
  allowPrivilegeEscalation: false
  requiredDropCapabilities:
    - ALL
  volumes:
    - 'configMap'
    - 'emptyDir'
    - 'projected'
    - 'secret'
    - 'downwardAPI'
    - 'persistentVolumeClaim'
  hostNetwork: false
  hostIPC: false
  hostPID: false
  runAsUser:
    rule: 'MustRunAsNonRoot'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'MustRunAs'
    ranges:
      # Forbid adding the root group.
      - min: 1
        max: 65535
  fsGroup:
    rule: 'MustRunAs'
    ranges:
      # Forbid adding the root group.
      - min: 1
        max: 65535
  readOnlyRootFilesystem: false
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: psp:privileged
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
rules:
- apiGroups: ['policy']
  resources: ['podsecuritypolicies']
  verbs:     ['use']
  resourceNames:
  - privileged
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: psp:restricted
  labels:
```

```
    addonmanager.kubernetes.io/mode: EnsureExists
rules:
- apiGroups: ['policy']
  resources: ['podsecuritypolicies']
  verbs:     ['use']
  resourceNames:
  - restricted
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: default:restricted
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: psp:restricted
subjects:
- kind: Group
  name: system:authenticated
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: default:privileged
  namespace: kube-system
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: psp:privileged
subjects:
- kind: Group
  name: system:masters
  apiGroup: rbac.authorization.k8s.io
- kind: Group
  name: system:nodes
  apiGroup: rbac.authorization.k8s.io
- kind: Group
  name: system:serviceaccounts:kube-system
  apiGroup: rbac.authorization.k8s.io
```

```
kubectl apply -f psp-restrictive.yaml
```

Next

Install some helm releases and see!

References

Reference	URL
Preview - Secure your cluster using pod security policies in Azure Kubernetes Service (AKS)	https://learn.microsoft.com/en-us/azure/aks/use-pod-security-policies