Issues regarding cluster resources in Azure

Last updated by | Leslie Cardoza | Feb 23, 2024 at 2:13 PM GMT+1

Common issues for the cluster's resources in Azure and their troubleshooting.
A.K.A. All the things **not** Kubernetes.

> Errors are listed from generic to specific, hence subsections of an error are subsets of it.
> To solve the innermost one, also check the solution for its parent.

Contents

- Template deployments fail
  - ... with ERROR: Missing input parameters
  - ... with ERROR: InvalidTemplateDeployment
    - Policy violation
    - Quota exceeded
  - ... with code CreateVMSSAgentPoolFailed
    - Unable to establish connection from agents to Kubernetes...
- Not enough free IPs in the subnet 'aks01-subnet'
- Pipelines cannot connect to the cluster
  - i/o timeout
  - User does not have access to the resource in Azure
- Changing property 'agentPoolProfile.vmSize' is not allowed
- One or more resource is non compliant
  - The deployment of the GitOps Flux extension timed out
  - The deployment of the GitOps Flux extension fails
- Pods from AKS extensions are not running
  - Some or all Pods from AKS extensions do not tolerate taints

Template deployments fail

... with `ERROR: Missing input parameters`

Solution
Go through this again, then check your deployment's parameters and retry.

... with `ERROR: InvalidTemplateDeployment`

Solution
Check the reason ***why*** your deployment failed.

Policy violation

Root cause
One or more attributes in your template violate an existing policy.

Solution

1. Carefully read the error message: it should™ report the name (and sometimes the ID) of the policy that has been violated:

> Resource '6f6552e3-570e-5bd8-8acc-9de2646c8ce3' was disallowed by policy.
> Policy identifiers:

```
[{
  "policyAssignment": { … },
  "policyDefinition": {
    "name": "AAB Azure Kubernetes Service - SSH Key DENY v1",
    "id": "/providers/ … /policyDefinitions/aab-aks-ssh-key-deny-v1"
  }
  …
}]
```

2. Look for that policy definition in *Azure Policy* > *Definitions* and check its rules.
   The same via CLI:

```
$ az policy definition list --query "[?(@.displayName=='AAB Azure Kubernetes Service - SSH Key DENY v1
[
  {
    "if": {
      "allOf": [
        {
          "equals": "Microsoft.ContainerService/managedClusters",
          "field": "type"
        },
        {
          "exists": true,
          "field": "Microsoft.ContainerService/managedClusters/linuxProfile.ssh"
        }
      ]
    },
    "then": {
      "effect": "deny"
    }
  }
]
```

3. Adjust your template.
   You'll need the checks to **fail** (so that the resource will *not* be denied).

Quota exceeded

Solution
Increase the limit yourself.

From Quota's levels like regional vCPU cores or specific VM size vCPU are limited by default and too low for our use case :

> The quotas are initially set by default to that low number, not by FSCP, but that's how a subscription is created. In regard to increasing limits, there's no limit. It doesn't matter if the limit is set to 10, 100 or 1000 CPUs. Cost only comes from what is actually being used. It all depends on the team's design, usage etc. so it's hard to suggest how much the limit should be raised.
>
> The approval process is usually automatically approved on Microsoft's side, so there's shouldn't be any issues there. There might be an occasional question from MS Support in case the limit raised is done for soon to be deprecated resources where they will just do a double-check if maybe an "upgrade" would be desired, but those are rare cases. Any team that needs their quota's increased, can manage that themselves since they have the Support Contributor Role.

... with code `CreateVMSSAgentPoolFailed`

Unable to establish connection from agents to Kubernetes API server

Error message example


```
{
  "code": "DeploymentFailed",
  "message": "At least one resource deployment operation failed. Please list deployment operations for deta
  "details": [
    {
      "code": "CreateVMSSAgentPoolFailed",
      "message": "Unable to establish connection from agents to Kubernetes API server, [...]. Details: Code
      [...]
    }
  ]
}
```


Root cause
Your subnet is not allowed to connect to the cluster's control plane or to system pods.

Solution

1. Check your subnet's NSG allows for such communication flows, and customize it if not.
2. Check with the NET team or Microsoft's support if any network issue is preventing your nodes to communicate.

Not enough free IPs in the subnet 'aks01-subnet'

See also not enough IPs for later.

Root cause
Each Node in the cluster reserves one IP address for itself, plus X (by default 30) IP addresses to assign its pods.
This means adding or deleting Pods will **not** change the number of occupied IP addresses. Also, deleting Pods at random will not be enough, because they will be immediately recreated by the ReplicaSets managing them.

Solution
You will need to reduce the number of **Nodes** in your Subnet.
To do this, you will probably need to reduce the number of your workload's replicas first to free up space on the Nodes. This is especially true if your cluster has **cluster autoscaling** enabled, as it will immediately ask for more nodes to accomodate the number of Pods defined in your workload if needed.

Please read the Save resources (and money) on clusters page for suggestions.

Pipelines cannot connect to the cluster

i/o timeout

Error message example

> Error: E0707 14:59:59.185143 2720 memcache.go:265] couldn't get current server API group list: Get "https://cdd01a-itc93kj6.hcp.westeurope.azmk8s.io:443/api?timeout=32s ": dial tcp 10.186.105.4:443: i/o timeout

Root cause
The Subnet's NSG is missing the rule to allow Azure DevOps Pipeline agents to reach the cluster's Control Plane.

Solution
Add the missing rule to the NSG.

User does not have access to the resource in Azure

Error message example

> Error from server (Forbidden): namespaces is forbidden: User "14c6d11f-ae9e-40d8-8c57-aac4f078a007" cannot list resource "namespaces" in API group "" at the cluster scope: User does not have access to the resource in Azure. Update role assignment to allow access.

Root cause
The user, Service Principal, Managed Identity or Whatever you are using to connect to the cluster is missing a suitable role assignment and is hence lacking permissions.

Solution
Choose a suitable RBAC role and assign it to the user.

Changing property 'agentPoolProfile.vmSize' is not allowed

See Resize node pools in Azure Kubernetes Service for details.

Specifically, look for the following:

- What happens when you change the SKU of a node pool?
- What is present in a **system** node pool? Can a cluster function when the system node pool is deleted or replaced during an upgrade?
- Does MS allow changing SKU of a AKS cluster which has a single node pool - the system one?
- Can you have more than one system node pool?

One or more resource is non compliant

Root cause
One or more attributes of the resource in case violates a policy.

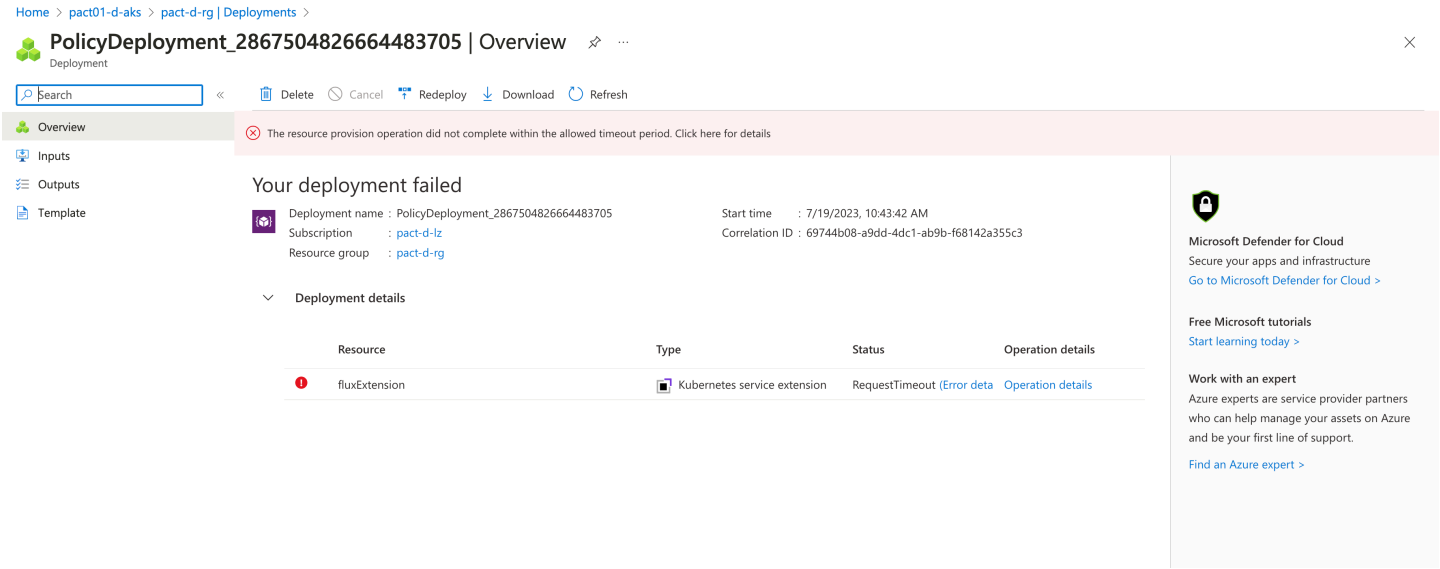
Solution

1. Follow the same steps as per [Template deployments fail with ERROR: InvalidTemplateDeployment for polilcy violation](#).
2. Check the effect of the policy:
   - If it checks for one or more attributes in your resource, just change the resource definition and apply.
   - If it is more complex, like the DINE policies for Flux or the Log Analytics Workspace, create a remediation task for the resource.
     See [this guide](#) and [this video](#) .

> Keep in mind a remediation will magically change your resources' attributes only for **DINE policies** or **policies with *Modify* effects**.
> [Read more.](#)

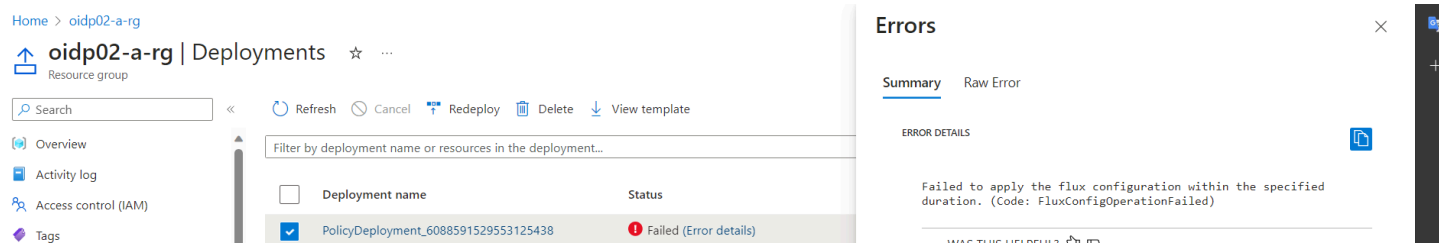
The deployment of the GitOps Flux extension timed out

Error message example



OR

The deployment of the GitOps Flux extension fails



Solution

1. Manually delete the Flux extension from your cluster.

```
# Using Azure CLI
# Delete the base flux configuration
az k8s-configuration flux delete -n baseline-configuration \
     --resource-group <resource-group> \
     --cluster-name <cluster name> \
     --cluster-type managedClusters \
     --subscription <subscription name> \
     --force --no-wait --yes
# Delete the flux extension
az k8s-extension delete -n fluxextension
     --resource-group <resource-group> \
     --cluster-name <cluster name> \
     --cluster-type managedClusters \
     --force --yes

# Using Azure Devops Pipeline
# Add the k8s extenstion on the agent first
- task: AzureCLI@2
    displayName: Install K8s Extention
    inputs:
      azureSubscription: "<RG Service connection>"
      scriptType: bash
      scriptLocation: inlineScript
      inlineScript: az extension add -n k8s-configuration && az extension add -n k8s-extension
```

1. Remediate the Flux DINE Policy.
   This will reinstate the extension and apply the mandatory components.
2. Confirm checking the `deny-all` global network policy is present:

```
kubectl get globalNetworkPolicies 'deny-all'
```

Pods from AKS extensions are not running

Thanks to **@Muhammad Faisal Tariq** and **@Marcel Verweij** for bringing this to our attention.

Some or all Pods from AKS extensions do not tolerate taints

Error message example

```
      KUBERNETES_PORT_443_TCP:       tcp://mndx-t-aks01-atuof260.hcp.westeurope.azmk8s.io:443
      KUBERNETES_SERVICE_HOST:       mndx-t-aks01-atuof260.hcp.westeurope.azmk8s.io
      POD_NAME:                      extension-agent-6f9bbfb45f-6cdpp (v1:metadata.name)
      AGENT_TYPE:                    ConfigAgent
      AGENT_NAME:                    ControllerManager
      KUBERNETES_PORT_443_TCP_ADDR:  mndx-t-aks01-atuof260.hcp.westeurope.azmk8s.io
      KUBERNETES_PORT:               tcp://mndx-t-aks01-atuof260.hcp.westeurope.azmk8s.io:443
    Mounts:
      /etc/acs/azure.json from acs-credential (rw)
      /fluent-bit/etc/ from fluentbit-clusterconfig (rw)
      /var/lib/docker/containers from varlibdockercontainers (ro)
      /var/log from varlog (ro)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-hv7pr (ro)
Conditions:
  Type           Status
  PodScheduled   False
Volumes:
  varlog:
    Type:          HostPath (bare host directory volume)
    Path:          /var/log
    HostPathType:
  varlibdockercontainers:
    Type:          HostPath (bare host directory volume)
    Path:          /var/lib/docker/containers
    HostPathType:
  fluentbit-clusterconfig:
    Type:      ConfigMap (a volume populated by a ConfigMap)
    Name:      extension-manager-fluentbit-config
    Optional:  false
  acs-credential:
    Type:          HostPath (bare host directory volume)
    Path:          /etc/kubernetes/azure.json
    HostPathType:  File
  kube-api-access-hv7pr:
    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
  ----     ------            ----                   ----               -------
  Warning  FailedScheduling  28m (x636 over 2d5h)   default-scheduler  0/4 nodes are available: 1 node(s) had untolerated taint {mendix: true}, 3 node(s) had untolerated taint {CriticalAddonsOnly: true}
. preemption: 0/4 nodes are available: 4 Preemption is not helpful for scheduling.
  Normal   NotTriggerScaleUp  5m58s (x3605 over 3d5h)  cluster-autoscaler  pod didn't trigger scale-up: 1 max node group size reached, 1 node(s) had untolerated taint {mendix: true}
  Normal   NotTriggerScaleUp  56s (x24019 over 3d5h)   cluster-autoscaler  pod didn't trigger scale-up: 1 node(s) had untolerated taint {mendix: true}, 1 max node group size reached
mfaisaltariq@V3NL4FDPC7 ~ %
```

Root cause
Pods coming from one or more AKS extensions do not tolerate some or all taints present on all Nodes. This is a known issue we sadly have next to no power upon.

Workaround
Dedicate a *relatively* small node pool with no such taints to the workload from these extensions.

Solution
Wait for MSFT to fix this issue.