

# Can't connect to local registry 127.0.0.1:32000

## Issue

- Getting a connection refused when trying to upload images to the local registry.

```
"msg": "non-zero return code",  
"rc": 1,  
"start": "2022-05-26 10:37:08.441849",  
"stderr": "Error: Get \"https://localhost:32000/v2/\": dial tcp [::1]:32000: connect: connection refused; Get  
\"http://localhost:32000/v2/\": dial tcp [::1]:32000: connect: connection refused",  
"stderr_lines": [  
  "Error: Get \"https://localhost:32000/v2/\": dial tcp [::1]:32000: connect: connection refused; Get  
  \"http://localhost:32000/v2/\": dial tcp [::1]:32000: connect: connection refused"  
],
```

## Environment

- Red Hat Enterprise Linux 8.6
- Element On-Premise Installer 2022-05.04

## Resolution

First, let's begin by removing any bits of the old image that may be in containerd. We need the name of the image to do this and looking at this error:

```
"cdkbot/registry-amd64:2.6": rpc error: code = Unknown desc = failed to pull and unpack image
```

```
"docker.io/cdkbot/registry-amd64:2.6": failed to extract layer
```

the image is named `docker.io/cdkbot/registry-amd64:2.6` . So we will now run:

```
microk8s.ctr rm docker.io/cdkbot/registry-amd64:2.6
```

Unmount the offending volume from the `kubectl describe pod` setup:

```
sudo umount /var/snap/microk8s/common/var/lib/containerd/tmpmounts/containerd-mount490181863
```

If this succeeds, then you can issue:

```
microk8s.ctr pull docker.io/cdkbot/registry-amd64:2.6
```

and if this succeeds, you can then run:

```
kubectl delete pod -n container-registry registry
```

and watch the registry come back up.

If you cannot get the mounted volume to unmount, you may need to reboot to completely clear the issue.

## Root Cause

The root cause is that the registry container will not start:

```
$ kubectl get pods -A
NAMESPACE          NAME                                READY  STATUS      RESTARTS  AGE
kube-system        hostpath-provisioner-566686b959-jl2b4  1/1    Running     1         69m
...
container-registry registry-9b57d9df8-kmks4             0/1    ImagePullBackOff  0         44m
```

To figure out why this won't start, we need to run `kubectl describe pod -n container-registry registry` :

```
$ kubectl describe pod -n container-registry registry
Name:          registry-9b57d9df8-k7v2r
Namespace:    container-registry
Priority:      0
Node:         mynode/192.168.122.1
Start Time:   Thu, 26 May 2022 11:33:04 -0700
Labels:       app=registry
```

pod-template-hash=9b57dea58

```
...
Normal BackOff      5m41s (x4 over 7m36s) kubelet      Back-off pulling image "cdkbot/registry-
amd64:2.6"
Warning Failed       5m41s (x4 over 7m36s) kubelet      Error: ImagePullBackOff
Warning Failed      2m58s          kubelet      Failed to pull image "cdkbot/registry-amd64:2.6": rpc
error: code = Unknown desc = failed to pull and unpack image "docker.io/cdkbot/registry-amd64:2.6": failed to
extract layer sha256:8aa4fcad5eeb286fe9696898d988dc85503c6392d1a2bd9023911fb0d6d27081: failed to
unmount /var/snap/microk8s/common/var/lib/containerd/tmpmounts/containerd-mount490181863: failed to
unmount target /var/snap/microk8s/common/var/lib/containerd/tmpmounts/containerd-mount490181863: device
or resource busy: unknown
```

Looking at the above, we can see that `/var/snap/microk8s/common/var/lib/containerd/tmpmounts/containerd-mount490181863` is busy and failing to unmount, thus causing our problem.

We've also noticed in this case that bits of an old image download can be left in containerd and we've updated the resolution to handle this as well.

---

Revision #3

Created 26 May 2022 18:53:26 by Karl Abbott

Updated 6 November 2024 12:49:24 by Kieran Mitchell Lane