The dns-controller Kubernetes deployment has not updated the Kubernetes cluster's - AWS

Asked 5 years, 3 months ago Modified 5 months ago Viewed 2k times

9

I'm Trying to have a kubernetes cluster on aws and It's keep on failing while validation. using following command to update the cluster `kops update cluster cluster.foo.com --yes` and post running this `kops validate cluster`

```
Using cluster from kubectl context: cluster.foo.com

Validating cluster cluster.api.com

INSTANCE GROUPS
NAME            ROLE    MACHINETYPE MIN MAX SUBNETS
master-eu-west-2a   Master  t2.medium   1   1   eu-west-2a
nodes           Node    t2.medium   2   2   eu-west-2a

NODE STATUS
NAME    ROLE    READY

VALIDATION ERRORS
KIND    NAME        MESSAGE
dns apiserver   Validation Failed

The dns-controller Kubernetes deployment has not updated the Kubernetes cluster's API
DNS entry to the correct IP address.  The API DNS IP address is the placeholder address
that kops creates: 203.0.113.123.  Please wait about 5-10 minutes for a master to
start, dns-controller to launch, and DNS to propagate.  The protokube container and
dns-controller deployment logs may contain more diagnostic information.  Etcd and the
API DNS entries must be updated for a kops Kubernetes cluster to start.

Validation Failed
```

Please help in finding the root cause.

1. I tried deleting and recreating multiple time but that did not helped me.
2. Also tried manually placing the master public and private IP to route 53 but it break everything.

kubernetes kubectl kops

Share Improve this question Follow

asked Feb 4, 2019 at 18:49

bashIt
1,056 1 11 27

Could you provide logs from kubernetes cluster? – Nick Rak Feb 6, 2019 at 10:44

Could you please share the steps you've taken to create the kops cluster and kops config (without sensitive information)? If you've followed some guide/manual, please provide the link. – VAS Feb 15, 2019 at 17:49

I'm having the same problem with kOps 1.25 – Trevor Sullivan Sep 28, 2022 at 2:48

4 Answers

Sorted by: Highest score (default)

2

Since EC2 uses elastic IP address for public IP, each time you reboot master node it will receive a new public IP. It happens that KOPS does not pick up the new IP for the Kube API. For example, if your cluster name was `kube.mydomain.com`, the API DNS would be: `api.kube.mydomain.com` as you can see from Route53.

You'd see timeout error when you try to reach your cluster:

```
 $ kops rolling-update cluster
Using cluster from kubectl context: kube.mydomain.com

Unable to reach the kubernetes API.
Use --cloudonly to do a rolling-update without confirming progress with the k8s API


error listing nodes in cluster: Get "https://api.kube.mydomain.com/api/v1/nodes": dial
tcp 3.8.157.44:443: i/o timeout
$
```

To fix this: Each time your EC2 master node receives a new public IP, you must manually update the public IP against DNS of `api.kube.mydomain.com` in Route53.

Also ensure that the master's private IP is updated against the DNS of `api.internal.kube.mydomain.com`. Otherwise, the nodes will got to network-unavailable state.

Share Improve this answer Follow edited Feb 13, 2021 at 22:25 answered Feb 13, 2021 at 18:15

Parthiban Sekar
39 5

Updating the control panel address handled automatically by dns-controller. There is absolutely no reason to do any manual updates if everything is working correctly. – Ole Markus With Feb 14, 2021 at 6:24

0

As my experience if you have difference in version of kops and kubectl and kubernetes plane version then Kops will never update the Route53 enteries you must need to have the same version for all in my case

```
[root@ip-20-0-0-66 kuberneteswithkops]# kops version
Version 1.15.0 (git-9992b4055)
[root@ip-20-0-0-66 kuberneteswithkops]# kubectl version
Client Version: version.Info{Major:"1", Minor:"15", GitVersion:"v1.15.3",
GitCommit:"2d3c76f9091b6bec110a5e63777c332469e0cba2", GitTreeState:"clean",
BuildDate:"2019-08-19T11:13:54Z", GoVersion:"go1.12.9", Compiler:"gc",
Platform:"linux/amd64"}`
```

Share Improve this answer Follow

answered Mar 22, 2020 at 12:32

Mansur Ul Hasan
3,242 30 26

0

This happened to me when I applied custom instance_policies to my instance groups.

The reason why is Kops controller doesn't have the permission to change it your Route 53 `kops-controller.internal.` dns entry in your zone.

To fix this, apply this change to your master IAM role.

```
{
  "Version": "2012-10-17",
  "Statement":       [
       {
            "Action": [
                "route53:ChangeResourceRecordSets",
                "route53:ListResourceRecordSets",
                "route53:GetHostedZone"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:route53:::hostedzone/${hostedzone}"
            ]
        },
        {
            "Action": [
                "route53:GetChange"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:route53:::change/*"
            ]
        },
        {
            "Action": [
                "route53:ListHostedZones"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
]
```

Share Improve this answer Follow

answered Feb 25, 2022 at 2:08

jmcgrath207
1,547 2 22 35

0

I had this issue and for me the problem was that I was trying to use `t2.micro` for both the nodes and the control-plane (master node).

The issue was resolved by switching to `t3.small` for the nodes and `t3.medium` for the control-plane.

Share Improve this answer Follow

answered Nov 26, 2023 at 16:22

M3RS

7,194 6 37 51