From: Kenneth S[redacted]
Sent: Tuesday, May 14, 2024 10:09 AM
To: Allen L[redacted] <Allen.L@mrcooper.com>; William W[redacted] <William.W@mrcooper.com>; Toby S[redacted] <Toby.S@mrcooper.com>; Larry F[redacted] <larry.fl@mrcooper.com>; Rengarajan G[redacted] <Rengarajan.G@mrcooper.com>
Cc: Matthew W[redacted] <Matthew.W@mrcooper.com>; Kelly R[redacted] <Kelly.R@mrcooper.com>
Subject: My work product in general and its relationship to my new manager's comments, prior technical lead Rengarajan G[redacted]
Importance: High

Good morning-

I'd like to be spending the time I'm composing this email on working through technical issues to help Mr. Cooper, but a comment made by Rengarajan in our Kubernetes upgrades meeting last Thursday puts me in the position where I must take that time to defend myself again.

I've attached the comment, and though- as usual- I cannot follow 100% of what he said, in this case, I received what he intended to communicate very clearly.

He is expressing emotional frustration at the fact that "genesys" and then he corrects himself "on prem Kubernetes cluster upgrades in general" have taken so long.

To understand the context, one must take a moment to look at the nature of ongoing verbal meetings in general. Typically, they are a hodgepodge of narratives that more-or-less have continuity from one week to the next, but sometimes aren't that closely related to either last week's narratives or the reality of events as they actually occur. In this case, these "change champions" meetings have existed to chart and follow the progress of Kubernetes upgrades in Google cloud/GKE , Azure/AKS, and our on-prem RKE upgrades.

I don’t recall at what point in the timeline the following narrative was put forth or in what form, but I recall at one point the narrative that Rasmi would focus on the GKE upgrades, that Leo would focus on the Azure/AKS upgrades, and that I would focus on the on-prem upgrades. So, the general theme of Rengarajan’s comment was that “Rasmi’s area is great and Leo’s area is great, but it’s Ken’s area that really dragging us down”.

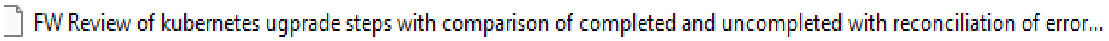
Since Rengarajan feels comfortable denigrating me with misleading and unhelpful commentary in a meeting, I think it’s fair that I examine what I *have* done in regard to the Kubernetes upgrades for *ALL* areas.

When the first upgrades were being attempted in AKS, Rengarajan and Larry F[redacted] were generally doing the driving, and I was present for every session. The pipelines for cluster configuration existed, and there were some cursory and informal notes left by Nikhil and others to guide steps, but there were many issues. The first upgrades were fraught with errors on the clusters after the pipelines were run. Larry and Rengarajan would put out these fires as the rest of us looked on, essentially, and those first couple of upgrades were a long night’s work for all involved.

I was the person from the very beginning who assembled notes, put together a documented set of steps that I refined with each session, and corrected errors in the pipelines so that fires would not have to be put out on the next runs. It’s fair to say that the standard pattern of these upgrades would be that Rengarajan and Larry would put out fires, and I was the one interested in correcting the problems at the source so that fires would not have to be put out.

I wrote and published the early procedures (which included the Traefik and cert-manager configurations) for **GKE, AKS, AND on-prem RKE**, and I performed and “drove” on quite a number of these GKE and AKS upgrades prior to the first on-prem attempts. It was MY work that took these upgrades from something that only Larry and Rengarajan could complete and turned them into routine procedures that many lower-level team-members could accomplish, and my procedures were used in just that way to parallelize the upgrades in GKE and AKS in the first (the difficult) round.

When the first on-prem cluster was attempted, there were additional errors related to the configuration pipelines that were unique to the on-prem clusters. The roles continued to be the same. In the attachment entitled:


FW Review of kubernetes ugprade steps with comparison of completed and uncompleted with reconciliation of error...

,

this relationship is seen clearly.

I am the one who is concerned with analyzing the steps of the upgrade, isolating the points in the pipeline code where things went wrong, and presenting these detailed observations to my technical leads, Rengarajan and Larry. Rengarajan, as always, ignored this communication. Larry offered some comments, which I further analyze. I solved two of the three issues myself, including one where I'm overriding/correcting both Larry and Kubernetes support. Larry did ultimately solve the other one, but it was because I- as was my intention- presented the issue in unambiguous, written terms in this email.

To review another such issue-

In the attachment entitled:

FW cluster config for on-prem clusters. specifically vanity domains.msg

The reader can follow that what I've done is go through the pipeline code line-by-line, understand exactly what it's going to do to modify certificates that exist, provide a detailed comparison to what the before and after will look like, and raise these concerns to my technical leads, Rengarajan and Larry. Rengarajan, as always, ignored this analysis. Larry's response was oddly unhelpful. I thought it would be clear that my purpose in this detailed review and subsequent flag-raising was to ask the question "Is this going to cause a problem". Larry's response generally led me to believe that it would not be a problem, that the pipeline will create a format that is "correct", and to not worry about it.

When it came time to do the upgrade, over a month later, however, this was in fact problem. Rengarajan was available to help troubleshoot, fortunately, as Larry was unavailable. The problem turned out to be exactly what I had predicted might be a problem way before it became a problem. The cert had to be put back into the form it was in prior. Once again, the technical leads seemed to prefer putting out fires to correcting issues at the source, though my efforts provided an opportunity to solve it beforehand.

Ultimately, however, as with GKE and AKS, I wrote the complete procedure and saw to it that the pipeline was corrected so that all of the non-genesys on-prem clusters were completed in relatively routine fashion, with configuration pipelines running cleanly.

On-prem genesys clusters are another challenge entirely. These were built independently by people who are no longer with the company, without documentation, and in some cases severely damaged by our security breach issues.

I was assigned to see what I can do to resolve the issues in the non-working genesyschop cluster. (I'm used to being assigned what everyone else has failed to resolve btw).

I determined that the portworx configuration was unsalvageable, so I uninstalled and installed the latest portworx storage cluster and provided volumes that got functionality for most processes back and running.

Around this time, the narrative in the meeting was that the on-prem genesys servers would not be upgraded, in favor of migrating them to a different environment. That narrative then changed and I was requested to attempt to upgrade chop, which I did. I did the same things we might have done in any of the other clusters, and as with many of those early-on, there were issues. In addition to the portworx storage cluster, there is a rook-ceph storage cluster, which was unhealthy after the storage breach, but is less healthy after the Kubernetes upgrade.

That is where we stand now, and that is the final context in which Rengarajan's comments were made.

I’ve attached just the two examples- though I could fill a hundred pages with such examples- and what I think a reasonable person would conclude is clear:

I’ve been exhibiting principle/lead level work since the day I walked in the door, but especially in relation to my time in devops compared to my background.

When I decided to try my hand at devops, I had only three prior jobs in the last 30 years: Unix admin, Oracle DBA, and Storage admin. My coding experience was shell scripting, period.

In short order, I obtained the associate certification for Azure and GKE, and was the first I know of to acquire the terraform certification.

I studied golang, javascript, typescript, and the pipeline languages so that I could attempt to follow what was going on with pipelines.

My first real accomplishment was a major one- I worked step-by-step through the pipelines that were written to move chef code into the chef-automate server through devops.

This was right at a time when Matt, Eli, Toby, and others were trying to achieve this same goal, so when it came up in our meeting- Matt was lamenting that someone should be responsible for showing us how this works, I was able to say “I know how it works”, and then proceeded to take the entire team line-by-line through that code and show them how to move cookbook code through the repo and into chef-automate.

Not long after that, there was a very high-profile Linux security bug that came out.

It turns out, I had been researching how to program compliance profiles and due to my principle/lead-level work, all of our Linux boxes had the fix- documented in our chef-automated program- for this bug on the *very next day*.

I then showed Eli and the rest of the team how to program a compliance profile.

I don’t doubt that Matt and Eli have done outstanding work in the chef and compliance profiles area after that, but they could not have done it if I had not led the way and broken down those barriers.

There are countless examples of me going above and beyond to make major impacts that have helped this company in numerous ways.

I have ALWAYS freely given away this information.

I’ve given presentations, as in the chef pipeline case. I’ve created tutorials for helm and Kubernetes. I created all of the procedures for Kubernetes upgrades. I’ve left documentation behind for just about everything I’ve touched, and the examples are too exhausting to assemble.

This is because my mindset is that my job and life will be easier the more I build up the people around me.

I have never felt threatened a second in my life because of someone else’s abilities.

On the other hand, the behavior I get from other people- in this case the Kubernetes leads but really just about everyone- is the opposite.

I’ve seen this behavior all of my working life, and it always makes me uncomfortable.

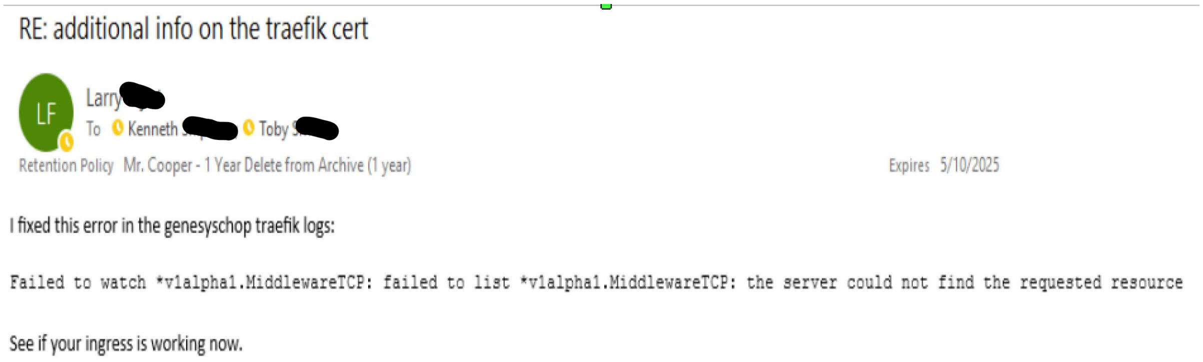
Neither Larry nor Rengarajan, generally, will share what they have done to fix an issue unless directly asked. I wasn’t following THEIR steps to perform Kubernetes upgrades, after all, we were following the ones I assembled.

There are also countless examples of this behavior, but Larry provided a timely one for me to use just last week:

After I performed the Kubernetes upgrade for genesyschop, there were application URLs that were not acquiring the proper certificate. I had done a lot of comparing to other environments and was admittedly stumped at that point, so I raised the issue to my technical leads.

I put the details into an email and sent it off to Larry- these are the responses:

In email:



RE: additional info on the traefik cert

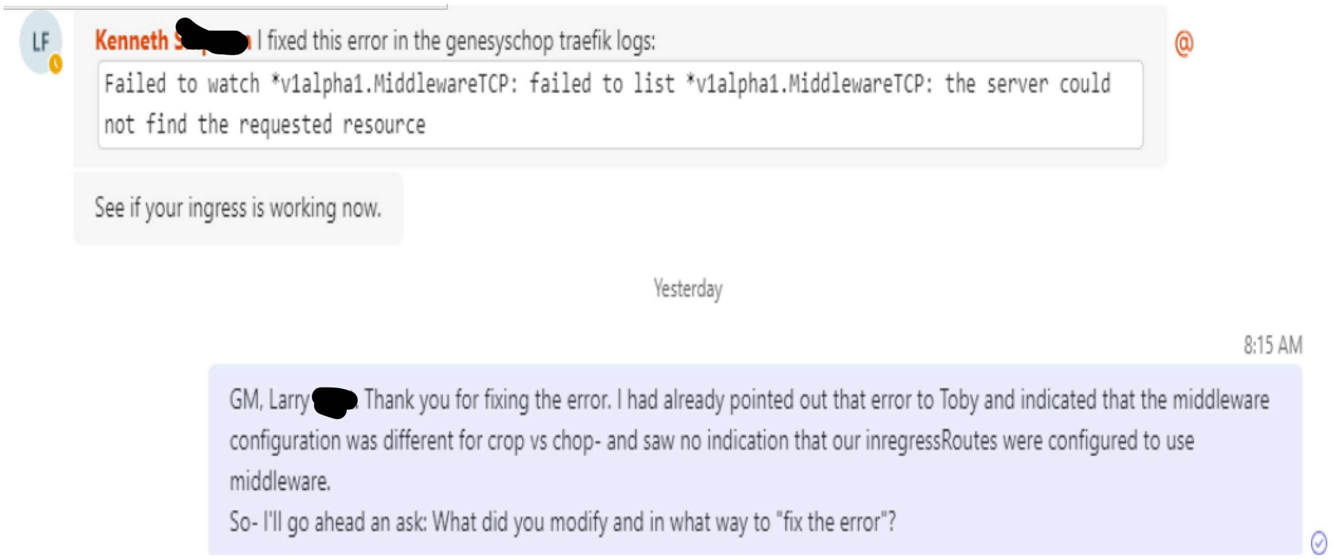
Larry [redacted]
To Kenneth [redacted] Toby [redacted]
Retention Policy Mr. Cooper - 1 Year Delete from Archive (1 year) Expires 5/10/2025

I fixed this error in the genesyschop traefik logs:

```
Failed to watch *v1alpha1.MiddlewareTCP: failed to list *v1alpha1.MiddlewareTCP: the server could not find the requested resource
```

See if your ingress is working now.

In teams:



Kenneth [redacted] I fixed this error in the genesyschop traefik logs:

```
Failed to watch *v1alpha1.MiddlewareTCP: failed to list *v1alpha1.MiddlewareTCP: the server could
not find the requested resource
```

See if your ingress is working now.

Yesterday

8:15 AM

GM, Larry [redacted] Thank you for fixing the error. I had already pointed out that error to Toby and indicated that the middleware configuration was different for crop vs chop- and saw no indication that our inregressRoutes were configured to use middleware.
So- I'll go ahead an ask: What did you modify and in what way to "fix the error"?

This is not an isolated incident- it's business as usual. Why would I have to ask that question?

Larry can be quite personable and helpful provided you address him in a verbal setting, but he seems very reluctant or incapable of providing any value in writing in my experience.

Rengarajan is much worse. I can honestly say that I don’t recall ever learning a single thing from Rengarajan.

The once or twice he did answer an email of mine, his responses generally indicated that he did not (or could not) read the content of my email.

To completely ignore my technical emails designed to correct pipelines and prevent potential outages before they happen, on the other hand, is unacceptable behavior in my opinion.

What kind of behavior is that?

How does that behavior help Mr Cooper?

In those emails, I am exercising my responsibility to the company that’s paying me money, and those emails exhibit what any reasonable person would conclude is conscientious effort and leadership.

What I got in return from Rengarajan is a complete disregard for my work and for my humanity in general. Rengarajan has completely ignored the reality of my valuable input, and generally that I exist, and substituted his own narrative about who I am and my work.

He was the technical lead over Kubernetes upgrades. I wasn’t stopping him from leading technically. I would be happy to follow any procedures he assembles for me, and if he’s not happy with the pace of progress of this mid-level engineer, I would think he has had the responsibility to mentor me into better performance with his technical leadership.

Just about every technical person at this company has benefited from my work and my documentation, but I've never seen a single thing from Rengarajan (or really mostly anyone else) that has helped me become better at devops, Kubernetes upgrades, or anything.

I'm done with providing outstanding work and results and having them be brushed under the rug by technical leads and management at Mr Cooper.

I'm the only one on that team who isn't a principle or lead, but I'm always expected to perform like a principle or lead, and I often outperform them.

I'll need a **significant** raise and title change before I listen to any verbal abuse from Rengarajan G[redacted]. I will expect that ALL of our communications will be in writing from now on. I can't understand much of what he says anyway.

Otherwise- considering the clearly documentable effort and impact my work has had on these upgrades, for ALL areas, it's fair for me to pull the "mid-level engineer" card for a change.

It's time for me to follow procedures that principles and leads write while I work at the pace of someone who stands to learn a lot from principles and leads.

What I intend to do now, with respect to Genesys Kubernetes upgrades, is to research the proper procedures for rolling upgrades in the presence of portworx, unless Rengarajan has already documented the proper steps and configuration, of course.

I'll be doing this at the pace of a mid-level engineer.

If you don't like it, Rengarajan, do it better or assign a higher-level engineer to show me how to do it.

If I don't get some relief from this management "style", I will be forced to conclude that my assignment to this person as my manager is retaliation and further harassment for my earlier claims of harassment.

Sincerely-

Kenneth S

Senior DevOps Engineer

Server Automation

8950 Cypress Waters Blvd.

EMAIL ATTACHMENT #1

From: Kenneth S[redacted]

Sent: Thursday, January 4, 2024 11:27 AM

To: Leonard C[redacted] <Leo.C[redacted]@mrcooper.com>; Shravya P[redacted] <Shravya.P[redacted]@mrcooper.com>

Subject: FW: Review of kubernetes ugprade steps with comparison of completed and uncompleted with reconciliation of errors

Good afternoon-

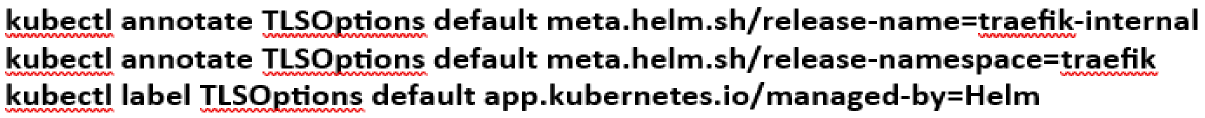
The notes below are mainly me jotting down my thoughts, raising questions about things that don't work or are incongruous with documentation, etc., but mostly are related to Nikhil's original set of steps that have been used to upgrade Azure/GKE, and most recently, on-prem clusters.

Feel free to add comments or additional observations.

From: Kenneth S[redacted]
Sent: Thursday, January 4, 2024 8:41 AM
To: Larry F[redacted]<larry.f@mrcooper.com>; Rengarajan G[redacted] <Rengarajan.G@mrcooper.com>; Toby S[redacted] <Toby.S@mrcooper.com>
Cc: David D[redacted] <david.d@mrcooper.com>
Subject: RE: Review of kubernetes ugprade steps with comparison of completed and uncompleted with reconciliation of errors

Another anomaly to mention is the TLSOptions label/annotations in the traefik stage.

This has been a relatively easy issue to resolve, by simply manually executing the following after the first failed run:



```
kubectl annotate TLSOptions default meta.helm.sh/release-name=traefik-internal
kubectl annotate TLSOptions default meta.helm.sh/release-namespace=traefik
kubectl label TLSOptions default app.kubernetes.io/managed-by=Helm
```

The config-traefik.yaml script runs these commands prior to the traefik install, however, in the following format:

```
                echo "Adding annotations to existing TLSOptions"

        kubectl label tlsoptions default app.kubernetes.io/managed-by=Helm --overwrite
```

```
kubectl annotate tlsoptions default meta.helm.sh/release-name=traefik-internal --overwrite

kubectl annotate tlsoptions default meta.helm.sh/release-namespace=traefik –overwrite
```

These steps fail both before and after the manual updates, though the stage completes successfully after the manual updates-

```
                                                  2023-12-10T02:12:04.0585299Z Adding
annotations to existing TLSOptions
2023-12-10T02:12:04.6637626Z Error from server (NotFound):
tlsoptions.traefik.containo.us "default" not found

2023-12-10T02:12:05.1927464Z Error from server (NotFound):
tlsoptions.traefik.containo.us "default" not found

2023-12-10T02:12:05.8651537Z Error from server (NotFound):
tlsoptions.traefik.containo.us "default" not found


Error: UPGRADE FAILED: Unable to continue with update: TLSOption "default" in
namespace "default" exists and cannot be imported into the current release (etc)
```

Since the “—overwrite” parameter shouldn’t be the factor that causes it to fail, I’m wondering if the case “tlso” vs “TLSO” is the issue.

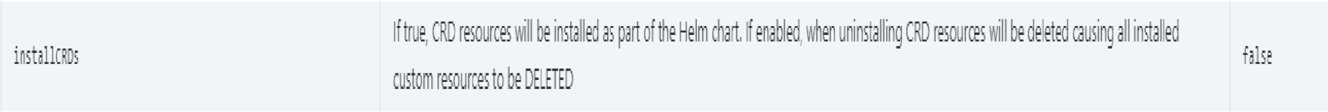
In any case, in my version- I’ve just swapped the version in the pipeline with the version that we manually type.

If there’s something of concern there, let me know.

From: Kenneth S[redacted]
Sent: Wednesday, January 3, 2024 1:59 PM
To: Larry F[redacted] <larry.f@mrcooper.com>; Rengarajan G[redacted] <Rengarajan.G@mrcooper.com>; Toby S[redacted] <Toby.S@mrcooper.com>
Cc: David D[redacted] <david.d@mrcooper.com>
Subject: RE: Review of kubernetes ugprade steps with comparison of completed and uncompleted with reconciliation of errors

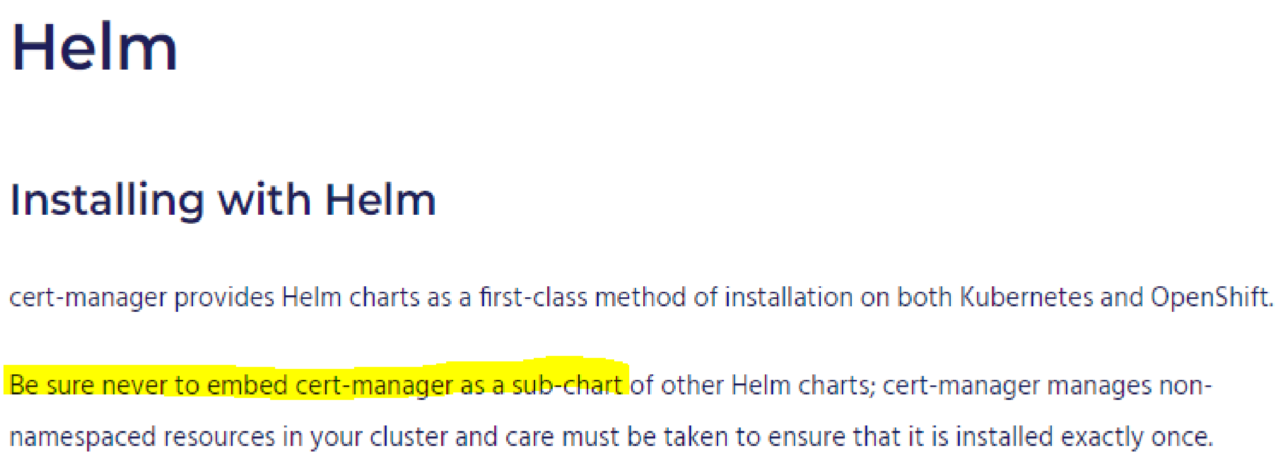
Some additional notes:

Cert-manager includes the “installCRDs” Parameter in 1.5.3 all the way up to the current version, so seems that helm still allows/does this:



`installCRDs`	If true, CRD resources will be installed as part of the Helm chart. If enabled, when uninstalling CRD resources will be deleted causing all installed custom resources to be DELETED	`false`

Another interesting note:



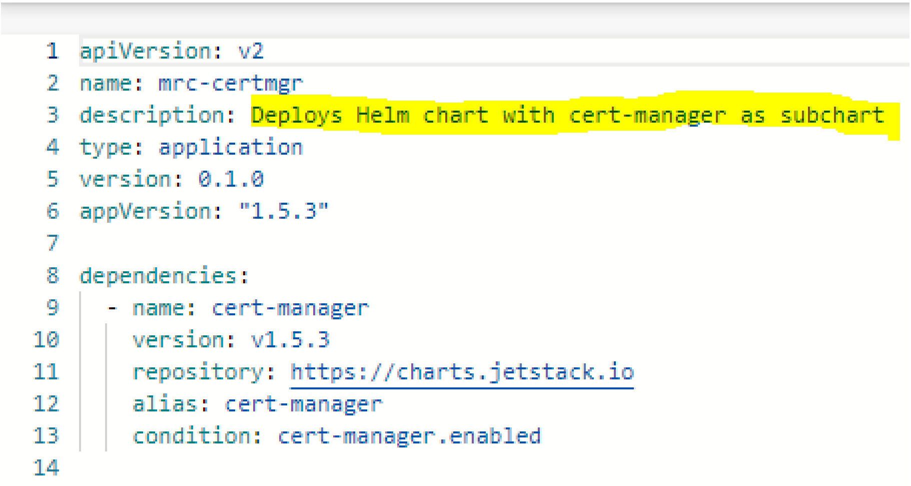
Helm

Installing with Helm

cert-manager provides Helm charts as a first-class method of installation on both Kubernetes and OpenShift.

Be sure never to embed cert-manager as a sub-chart of other Helm charts; cert-manager manages non-namespaced resources in your cluster and care must be taken to ensure that it is installed exactly once.

How we install it:



```yaml
apiVersion: v2
name: mrc-certmgr
description: Deploys Helm chart with cert-manager as subchart
type: application
version: 0.1.0
appVersion: "1.5.3"

dependencies:
  - name: cert-manager
    version: v1.5.3
    repository: https://charts.jetstack.io
    alias: cert-manager
    condition: cert-manager.enabled

```

Not sure if this violates the spirit as well as the letter of the law ATM, but it's worth noting.

From: Kenneth S[redacted] <Kenneth.S@mrcooper.com>
Sent: Wednesday, January 3, 2024 10:52 AM
To: Larry F[redacted] <larry.fl@mrcooper.com>; Rengarajan G[redacted] <Rengarajan.G@mrcooper.com>; Toby S[redacted] <Toby.S@mrcooper.com>

Cc: David D[redacted] <david.d@mrcooper.com>
Subject: RE: Review of kubernetes ugprade steps with comparison of completed and uncompleted with reconciliation of errors

Thanks for your comments and questions Larry. I think this type of written back-and-forth allows time to think about and dissect each point in a way that synchronous, verbal communication can't come close to doing, and of course we have a consolidated record that can be further scrutinized and reflected upon after the subconscious has a chance to do some work.

I've added my comments below.

From: Larry F[redacted] <larry.fl@mrcooper.com>
Sent: Wednesday, January 3, 2024 9:42 AM
To: Rengarajan G[redacted] <Rengarajan.G@mrcooper.com>; Kenneth S[redacted] <Kenneth.S@mrcooper.com>; Toby S[redacted] <Toby.S@mrcooper.com>
Cc: David D[redacted] <david.d@mrcooper.com>
Subject: RE: Review of kubernetes ugprade steps with comparison of completed and uncompleted with reconciliation of errors

Kenneth,

- Some clusters have more than one certificate so we should plan to set the labels and annotations on them all.

  The two we have scheduled for this weekend have one each, but point taken.

- Invalid ownership metadata can be fixed by setting the ownership annotations. Since these are already in your notes below, that should be easy to do.

  There is some evidence to suggest this error results from an attempt to install CRDs when they are already there. If one takes the error at its word, then it would follow that the condition complained about in the error would be valid, and it appears to not be,
  as "meta.helm.sh/release-name": must be set to "cert-manager" was already done in the first line of the manual annotations. I included a link to a solution provided by someone receiving the same error in a similar situation, and the solution was to uninstall CRDs, install the new ones, set installCRDs to false, and run the install. However, in that example, there were no

manual annotations added. This raises the question of whether the manual annotations themselves may be at issue. The error itself is misleading, so one is left to make the best guess as to its origin. It may be that the cert-manager install intends to make those annotations itself, since the manual install of the CRDs does not do this by design, and wasn't designed to handle the exception of finding them already there. It never complained about annotations on the rest of the CRDs, though, so the jury is still out on this error. I like to eliminate what's known to be wrong, potentially irrelevant or not, and go from there, however. In this case, manually installing CRDs and then telling Helm to install them is one bit of fat that can't hurt to be trimmed. This review does make me want to try the install without the manual annotations, however- something to consider over the next couple of days.

●Ryan at SUSE told us that cert manager no longer installs the CRDs via helm because they are too large. This means the install CRD parameter is probably irrelevant if it is using Helm.
I'm thinking "probably" is the key word here.

●Ryan also helped us when we had trouble with the mutating webhook after we deleted cert manager and the CRDS. Have you factored that into your analysis?
This was only an issue on c_______, and I don't recall much about that one, tbh. I'll get into my notes and see if I can refresh my memory.

Thanks,

Larry

From: Rengarajan G[redacted] <Rengarajan.G@mrcooper.com>
Sent: Tuesday, January 2, 2024 11:33 AM
To: Kenneth S[redacted] <Kenneth.S@mrcooper.com>; Toby S[redacted] <Toby.S@mrcooper.com>; Larry F[redacted] <larry.f@mrcooper.com>
Cc: David D[redacted]<david.d@mrcooper.com>
Subject: Re: Review of kubernetes ugprade steps with comparison of completed and uncompleted with reconciliation of errors

Mentioned steps looks like good to me. Some of the are one time activity we need to fix on adhoc as we run in to any potential issue(Specifically with cert manager). I am good with proceeding for planning for rest of the onprem cluster(may be 2) for this coming weekend.

Regards,

Rengarajan

From: Kenneth S[redacted] <Kenneth.S@mrcooper.com>
Date: Tuesday, January 2, 2024 at 10:28 AM
To: Toby S[redacted] <Toby.S@mrcooper.com>, Rengarajan G[redacted] <Rengarajan.G@mrcooper.com>, Larry F[redacted] <larry.f@mrcooper.com>
Cc: David D[redacted]<david.d@mrcooper.com>
Subject: Review of kubernetes ugprade steps with comparison of completed and uncompleted with reconciliation of errors

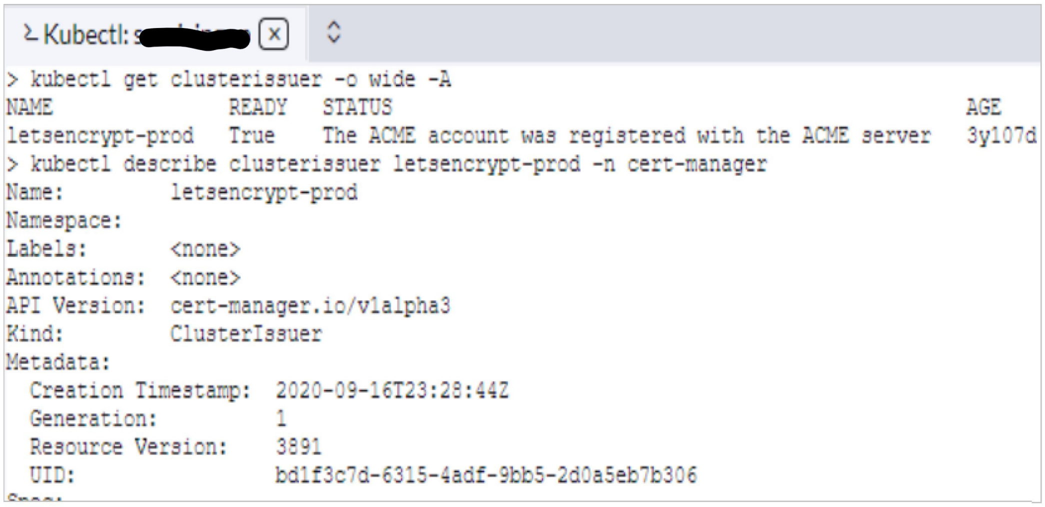
Good morning-

I've run through the steps we've been using in our environments (taken from Nikhil's steps and updated along the way). Just for comparison, I'm showing the current state of servicingop followed by post-upgrade r________ related to the steps in bold, with my comments in this color, indicating fixes for errors we've seen at the bottom.

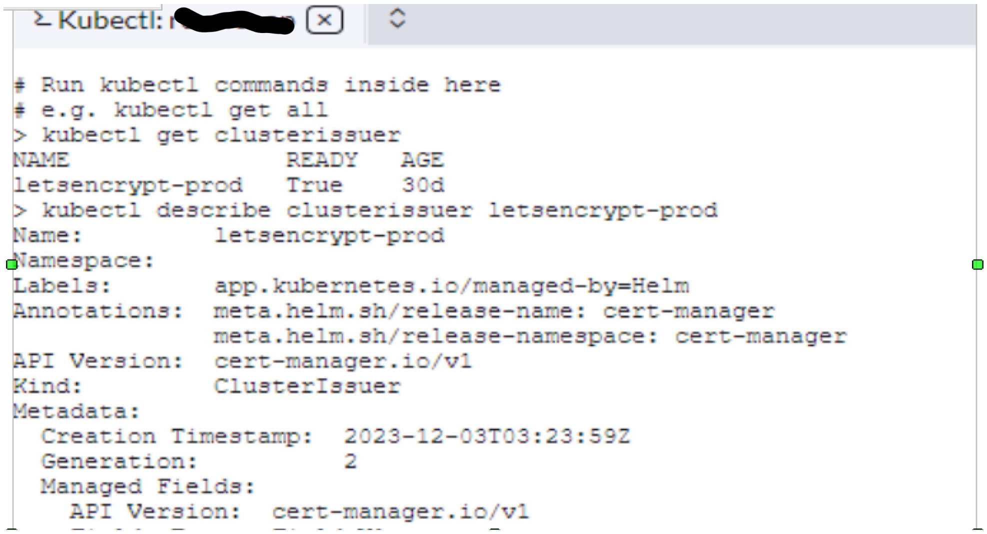
We can schedule another or more of onprem clusters when we all agree to pursue it. Hopefully this review will clear up the issues we've seen so far.

##annotate ClusterIssuer

```
kubectl label ClusterIssuer letsencrypt-prod app.kubernetes.io/managed-by=Helm --overwrite
kubectl annotate ClusterIssuer letsencrypt-prod meta.helm.sh/release-namespace=cert-manager --overwrite
kubectl annotate ClusterIssuer letsencrypt-prod meta.helm.sh/release-name=cert-manager --overwrite
```



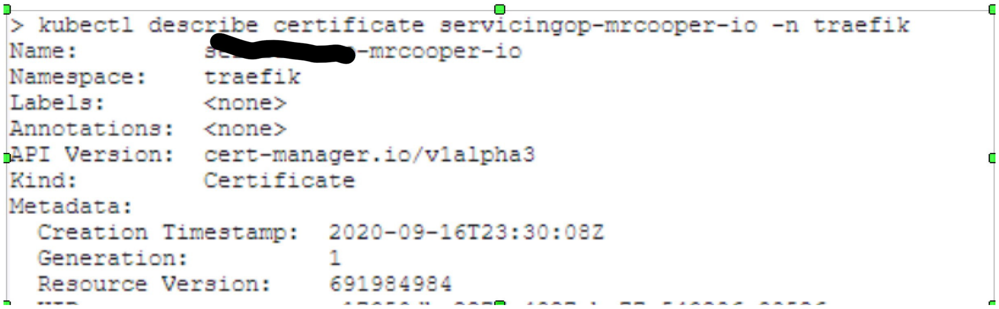
```
> kubectl get clusterissuer -o wide -A
NAME               READY   STATUS                                                AGE
letsencrypt-prod   True    The ACME account was registered with the ACME server   3y107d
> kubectl describe clusterissuer letsencrypt-prod -n cert-manager
Name:         letsencrypt-prod
Namespace:
Labels:       <none>
Annotations:  <none>
API Version:  cert-manager.io/v1alpha3
Kind:         ClusterIssuer
Metadata:
  Creation Timestamp:  2020-09-16T23:28:44Z
  Generation:          1
  Resource Version:    3891
  UID:                 bd1f3c7d-6315-4adf-9bb5-2d0a5eb7b306
```



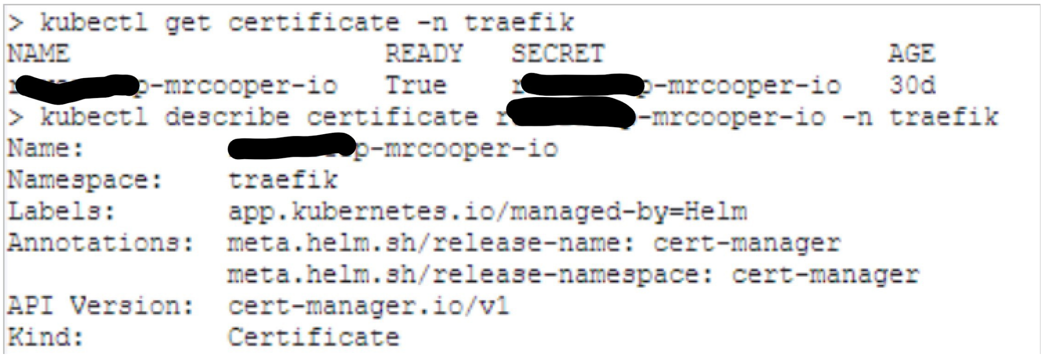
```
# Run kubectl commands inside here
# e.g. kubectl get all
> kubectl get clusterissuer
NAME               READY   AGE
letsencrypt-prod   True    30d
> kubectl describe clusterissuer letsencrypt-prod
Name:         letsencrypt-prod
Namespace:
Labels:       app.kubernetes.io/managed-by=Helm
Annotations:  meta.helm.sh/release-name: cert-manager
              meta.helm.sh/release-namespace: cert-manager
API Version:  cert-manager.io/v1
Kind:         ClusterIssuer
Metadata:
  Creation Timestamp:  2023-12-03T03:23:59Z
  Generation:          2
  Managed Fields:
    API Version:  cert-manager.io/v1
```

##Annotate Certificate (Change the name of certificate)

kubectl label Certificate "CERTNAME-HERE" -n traefik app.kubernetes.io/managed-by=Helm --overwrite
kubectl annotate Certificate "CERTNAME-HERE" -n traefik meta.helm.sh/release-name=cert-manager --overwrite
kubectl annotate Certificate "CERTNAME-HERE" -n traefik meta.helm.sh/release-namespace=cert-manager --overwrite



```
> kubectl describe certificate servicingop-mrcooper-io -n traefik
Name:         [redacted]-mrcooper-io
Namespace:    traefik
Labels:       <none>
Annotations:  <none>
API Version:  cert-manager.io/v1alpha3
Kind:         Certificate
Metadata:
  Creation Timestamp:  2020-09-16T23:30:08Z
  Generation:          1
  Resource Version:    691984984
```

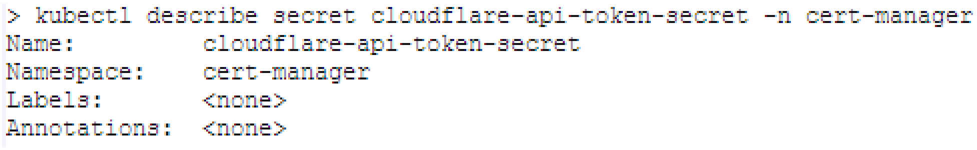


```
> kubectl get certificate -n traefik
NAME                    READY   SECRET                        AGE
[redacted]p-mrcooper-io    True    [redacted]p-mrcooper-io     30d
> kubectl describe certificate [redacted]-mrcooper-io -n traefik
Name:         [redacted]p-mrcooper-io
Namespace:    traefik
Labels:       app.kubernetes.io/managed-by=Helm
Annotations:  meta.helm.sh/release-name: cert-manager
              meta.helm.sh/release-namespace: cert-manager
API Version:  cert-manager.io/v1
Kind:         Certificate
```

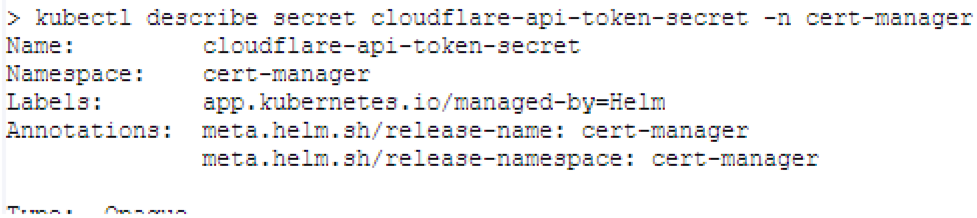
Annotate and Label Secret

kubectl label secret cloudflare-api-token-secret -n cert-manager app.kubernetes.io/managed-by=Helm --overwrite
kubectl annotate secret cloudflare-api-token-secret -n cert-manager meta.helm.sh/release-name=cert-manager --overwrite
kubectl annotate secret cloudflare-api-token-secret -n cert-manager meta.helm.sh/release-namespace=cert-manager --overwrite

Note these steps are also done by the pipeline, but after cert-manager has been deleted which throws a set of errors related to the missing namespace



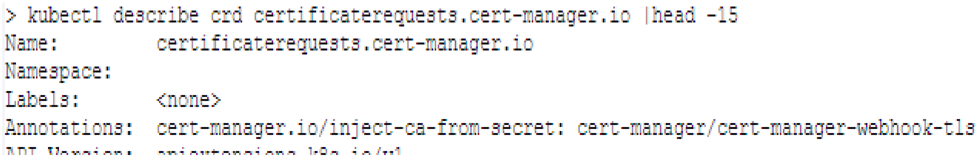
```
> kubectl describe secret cloudflare-api-token-secret -n cert-manager
Name:         cloudflare-api-token-secret
Namespace:    cert-manager
Labels:       <none>
Annotations:  <none>
```



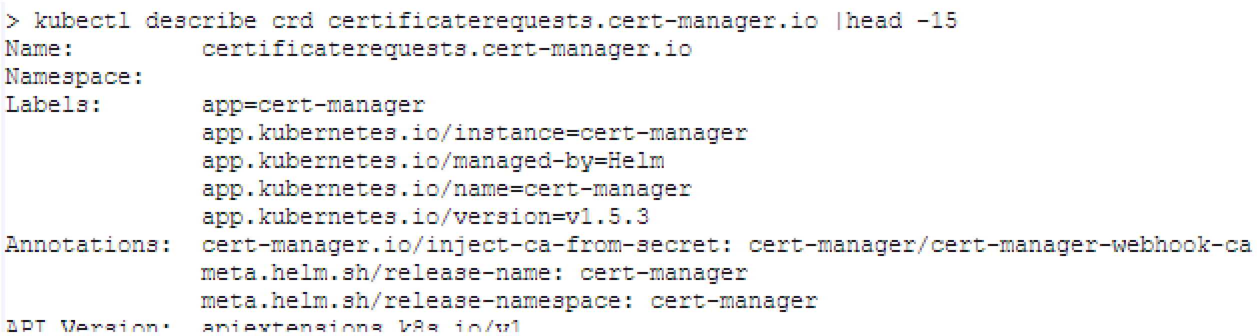
```
> kubectl describe secret cloudflare-api-token-secret -n cert-manager
Name:         cloudflare-api-token-secret
Namespace:    cert-manager
Labels:       app.kubernetes.io/managed-by=Helm
Annotations:  meta.helm.sh/release-name: cert-manager
              meta.helm.sh/release-namespace: cert-manager

Type:  Opaque
```

##Delete cert-manager fr

kubectl delete -f https://github.com/cert-manager/cert-manager/releases/download/v0.14.1/cert-manager.yaml



```
> kubectl describe crd certificaterequests.cert-manager.io |head -15
Name:         certificaterequests.cert-manager.io
Namespace:
Labels:       <none>
Annotations:  cert-manager.io/inject-ca-from-secret: cert-manager/cert-manager-webhook-tls
API Version:  apiextensions.k8s.io/v1
```



```
> kubectl describe crd certificaterequests.cert-manager.io |head -15
Name:         certificaterequests.cert-manager.io
Namespace:
Labels:       app=cert-manager
              app.kubernetes.io/instance=cert-manager
              app.kubernetes.io/managed-by=Helm
              app.kubernetes.io/name=cert-manager
              app.kubernetes.io/version=v1.5.3
Annotations:  cert-manager.io/inject-ca-from-secret: cert-manager/cert-manager-webhook-ca
              meta.helm.sh/release-name: cert-manager
              meta.helm.sh/release-namespace: cert-manager
API Version:  apiextensions.k8s.io/v1
```

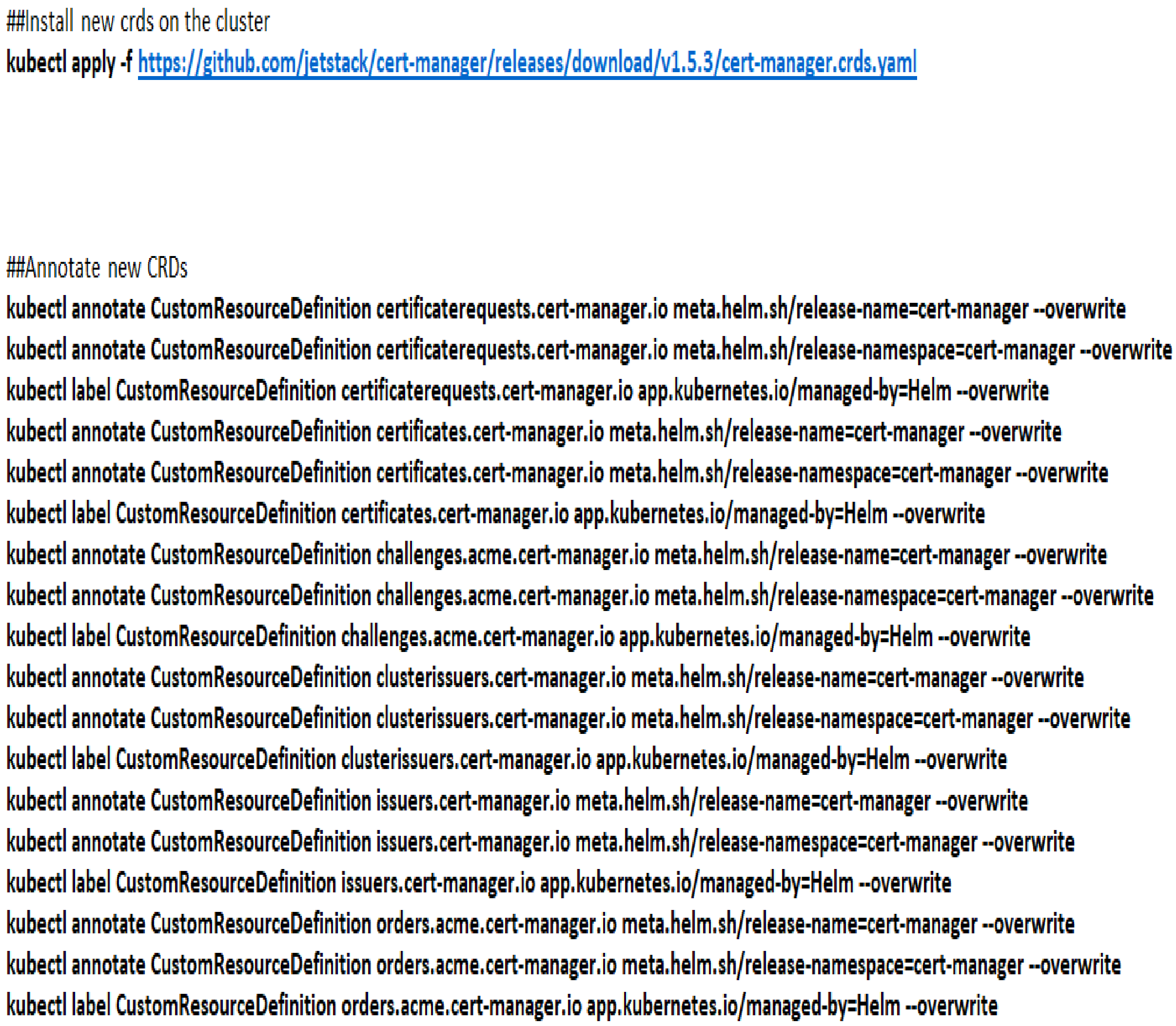
kubectl get crds |grep cert-manager

(run these if the output of the above command shows crds)

Delete CRDs for cert manager

```
kubectl delete crd certificaterequests.cert-manager.io
kubectl delete crd certificates.cert-manager.io
kubectl delete crd challenges.acme.cert-manager.io
kubectl delete crd clusterissuers.cert-manager.io
kubectl delete crd issuers.cert-manager.io
kubectl delete crd orders.acme.cert-manager.io
```

Typically, after the cert-manager deployment is removed, we’ve had no remaining crds



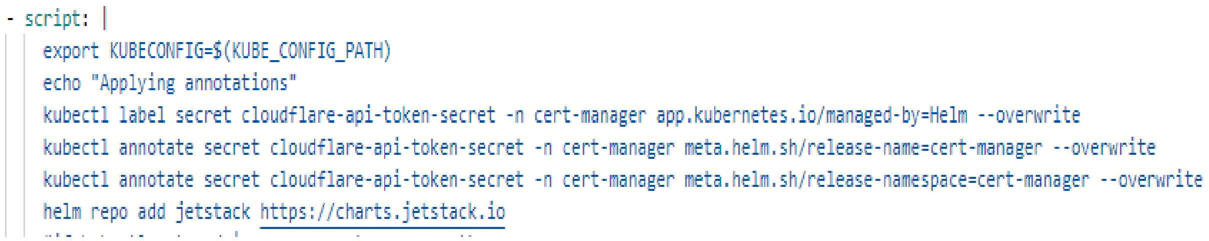
```
##Install new crds on the cluster
kubectl apply -f https://github.com/jetstack/cert-manager/releases/download/v1.5.3/cert-manager.crds.yaml
```

```
##Annotate new CRDs
kubectl annotate CustomResourceDefinition certificaterequests.cert-manager.io meta.helm.sh/release-name=cert-manager --overwrite
kubectl annotate CustomResourceDefinition certificaterequests.cert-manager.io meta.helm.sh/release-namespace=cert-manager --overwrite
kubectl label CustomResourceDefinition certificaterequests.cert-manager.io app.kubernetes.io/managed-by=Helm --overwrite
kubectl annotate CustomResourceDefinition certificates.cert-manager.io meta.helm.sh/release-name=cert-manager --overwrite
kubectl annotate CustomResourceDefinition certificates.cert-manager.io meta.helm.sh/release-namespace=cert-manager --overwrite
kubectl label CustomResourceDefinition certificates.cert-manager.io app.kubernetes.io/managed-by=Helm --overwrite
kubectl annotate CustomResourceDefinition challenges.acme.cert-manager.io meta.helm.sh/release-name=cert-manager --overwrite
kubectl annotate CustomResourceDefinition challenges.acme.cert-manager.io meta.helm.sh/release-namespace=cert-manager --overwrite
kubectl label CustomResourceDefinition challenges.acme.cert-manager.io app.kubernetes.io/managed-by=Helm --overwrite
kubectl annotate CustomResourceDefinition clusterissuers.cert-manager.io meta.helm.sh/release-name=cert-manager --overwrite
kubectl annotate CustomResourceDefinition clusterissuers.cert-manager.io meta.helm.sh/release-namespace=cert-manager --overwrite
kubectl label CustomResourceDefinition clusterissuers.cert-manager.io app.kubernetes.io/managed-by=Helm --overwrite
kubectl annotate CustomResourceDefinition issuers.cert-manager.io meta.helm.sh/release-name=cert-manager --overwrite
kubectl annotate CustomResourceDefinition issuers.cert-manager.io meta.helm.sh/release-namespace=cert-manager --overwrite
kubectl label CustomResourceDefinition issuers.cert-manager.io app.kubernetes.io/managed-by=Helm --overwrite
kubectl annotate CustomResourceDefinition orders.acme.cert-manager.io meta.helm.sh/release-name=cert-manager --overwrite
kubectl annotate CustomResourceDefinition orders.acme.cert-manager.io meta.helm.sh/release-namespace=cert-manager --overwrite
kubectl label CustomResourceDefinition orders.acme.cert-manager.io app.kubernetes.io/managed-by=Helm --overwrite
```

(These CRDS are not namespaced)

When running the config-certmanager.yaml stage called from the pipeline for r________, the following error occurred, followed by the piece of the script that generates the error. Presumably, the uninstall above deletes this namespace

```
2023-12-03T02:44:32.6515480Z Applying annotations

2023-12-03T02:44:33.6446895Z Error from server (NotFound): namespaces
"cert-manager" not found

2023-12-03T02:44:34.6429023Z Error from server (NotFound): namespaces
"cert-manager" not found

2023-12-03T02:44:36.0651924Z Error from server (NotFound): namespaces
"cert-manager" not found
```

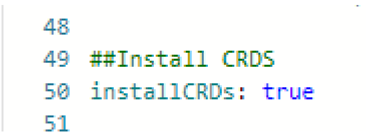


```
- script: |
    export KUBECONFIG=$(KUBE_CONFIG_PATH)
    echo "Applying annotations"
    kubectl label secret cloudflare-api-token-secret -n cert-manager app.kubernetes.io/managed-by=Helm --overwrite
    kubectl annotate secret cloudflare-api-token-secret -n cert-manager meta.helm.sh/release-name=cert-manager --overwrite
    kubectl annotate secret cloudflare-api-token-secret -n cert-manager meta.helm.sh/release-namespace=cert-manager --overwrite
    helm repo add jetstack https://charts.jetstack.io
```

This does not cause the stage to fail, but the following error does- and the reason for this is that we have manually installed CRDS and annotated them above, whereas our values file tells us to install CRDs on top. Theoretically, removing the redundant commands from the config-certmanager.yaml file above and changing the value for “installCRDs: true” to false in the values.yaml file will resolve both of these issues.

Error: Unable to continue with install: CustomResourceDefinition "certificaterequests.cert-manager.io" in namespace "" exists and cannot be imported into the current release: invalid ownership metadata; annotation validation error: missing key "meta.helm.sh/release-name": must be set to "cert-manager"

```
##[error]Bash exited with code '1'.
```



```

##Install CRDS
installCRDs: true

```

Notes from people with similar error and fix:

https://github.com/cert-manager/cert-manager/discussions/3483

Kenneth S[redacted]

Senior DevOps Engineer

Server Automation

8950 Cypress Waters Blvd.

Dallas, TX 75019

EMAIL ATTACHMENT 2

From: Larry F[redacted]<larry.f@mrcooper.com>
Sent: Tuesday, December 5, 2023 9:07 AM
To: Kenneth S[redacted] <Kenneth.S@mrcooper.com>; Rengarajan G[redacted] <Rengarajan.G@mrcooper.com>; Toby S[redacted] <Toby.S@mrcooper.com>
Cc: David D[redacted] <david.d@mrcooper.com>
Subject: RE: cluster config for on-prem clusters. specifically vanity domains

Ken,

You've found the right way and the wrong way to implement vanity domains. The right way being the cluster-config.yaml and the wrong way being manual updates to certificates.

You should ignore the discrepancies you see from people making manual updates and endeavor to implement the custom domains properly, the way you've discovered.

Larry

From: Kenneth S[redacted] <Kenneth.S@mrcooper.com>
Sent: Tuesday, December 5, 2023 9:04 AM
To: Larry F[redacted] <larry.f@mrcooper.com>; Rengarajan G[redacted] <Rengarajan.G@mrcooper.com>; Toby S[redacted] <Toby.S@mrcooper.com>
Cc: David D[redacted] <david.d@mrcooper.com>
Subject: RE: cluster config for on-prem clusters. specifically vanity domains

Slight correction for "secretName" below

From: Kenneth Shi[redacted] <Kenneth.S@mrcooper.com>
Sent: Tuesday, December 5, 2023 9:01 AM
To: Larry F[redacted]t <larry.f@mrcooper.com>; Rengarajan G[redacted] <Rengarajan.G@mrcooper.com>; Toby S[redacted] <Toby.S@mrcooper.com>
Cc: David D[redacted] <david.d@mrcooper.com>
Subject: RE: cluster config for on-prem clusters. specifically vanity domains

Circling back on this-

If I'm following the code correctly for vanity domains, whatever vanity domain names are passed to the "cluster-config.yaml" file are then passed to the "config-certmanager.yaml" file, which then runs a script that modifies the template "helmcharts/cert-manager/templates/certificate-vanity.yaml" and then applies it.

For example: the template starts like this for a vanity domain called “bozo.com” in a cluster named “mycluster”:

umodified template:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
 name: mrcooper-o
 namespace: traefik
spec:
 secretName: mrcooper-io
 issuerRef:
  name: letsencrypt-prod
  kind: ClusterIssuer
 dnsNames:
  - "*.dev."
  - "*.qa."
  - "*.load."
  - "*.uat."
  - "*.prod."
  - "*.$vanity"
```

This becomes modified to the following and applied (modifications highlighted):

```
apiVersion: cert-manager.io/v1
kind: Certificate
```

```
metadata:
 name: mycluster-bozo-com
 namespace: traefik
spec:
 secretName: mycluster-bozo-com
 issuerRef:
  name: letsencrypt-prod
  kind: ClusterIssuer
 dnsNames:
  - '*.dev.bozo.com'
  - '*.qa.bozo.co'
  - '*.uat.bozo.com'
  - '*.load.bozo.com'
  - '*.prod.bozo.com'
  - '*.bozo.com'
```

- `Bozo.com`

Based on your comment that two of the on-prem clusters use vanity domains: “g________”, and “i_______”, some questions arise:

1. The apiVersion in the template is `cert-manager.io/v1,` whereas, in the actual certs for infraop it is cert-manager.io/v1alpha3
2. It looks like i_____ has two vanity domains: mrcooper.in, and mrcooper.io.

a. For mrcooper.in, the current cert is much like the template, in that the dns names do not include reference to the cluster name, and the last entry is just the domain name by itself with no wildcard.

b. For mrcooper.io, the dnsNames include the cluster name, such as “’*.dev.i_______.mrcooper.io”, and there is no entry that specifies the dnsName without wildcard prefix.

3. For ge______, the cert “rook-admission-controller-cert” does not have dnsNames that reflect the vanity template in any way. However, that cluster also has a cert “g_________s include the cluster name. However, that cert has an apiVersion of “cert-manager.io/v1”
4. There are several other fields in the metadata area on current templates, but I’m assuming these are derived so I’m not that concerned about those.

Since our upgrade plan currently attempts to use the cluster_config.yaml to update traefik and cert-manager, and presumably the vanity domain-related certificates, we probably need to clear up these questions if possible before proceeding where vanity domains are in question.

Thanks,

Kenneth S

Senior DevOps Engineer

Server Automation

8950 Cypress Waters Blvd.

Dallas, TX 75019

From: Larry F[redacted] <larry.f@mrcooper.com>
Sent: Thursday, October 5, 2023 9:47 AM

To: Kenneth S[redacted] <Kenneth.S@mrcooper.com>; Rengarajan G[redacted] <Rengarajan.G@mrcooper.com>; Toby S[redacted] <Toby.S@mrcooper.com>
Cc: David D[redacted] <david.d@mrcooper.com>
Subject: RE: cluster config for on-prem clusters.

Kenneth,

None of our on-prem clusters use external ingress. You can review this by looking at Traefik and you'll only see traefik-internal.

Two of our on-prem clusters use vanity domains. G_______ has "rook-admission-controller-cert". Infraop has "in____-mrcooper-in". You can review this by looking at the cert manager certificates.

Larry

From: Kenneth S[redacted] <Kenneth.S@mrcooper.com>
Sent: Thursday, October 5, 2023 8:24 AM
To: Rengarajan G[redacted] <Rengarajan.G@mrcooper.com>; Larry F[redacted] <larry.f@mrcooper.com>; Toby S[redacted] <Toby.S@mrcooper.com>
Cc: David D[redacted]<david.d@mrcooper.com>
Subject: cluster config for on-prem clusters.

GM- as part of items to consider prior to the r_______ upgrade-

There is only one example of the cluster-config.yaml being ran for an on-prem cluster, which is sandboxop.

I've included its pipeline with a random other pipeline below for comparison.

The items of concern are the "external ingress" (which is false for sandboxop but true in all of the other examples I've reviewed)

the vanitydomains inclusion in sandboxop, which doesn't appear to be specified anywhere else.

I created a branch of repo Terraform.K8s called “re______cluster_config” to create the pipeline yaml for r______.

Let me know if there is any tribal knowledge that will help clear up what these settings should be for reverse on-prem.

Sandboxop-

```
resources:
  repositories:
    - repository: templates
      type: git
      name: "Infrastructure Automation/Terraform.Modules.Azure.K8s-Base"
      ref: refs/heads/module_split

trigger: none
extends:
  template: config-pipelines/cluster-config.yaml@templates
  parameters:
    clusterName: sandboxop
    cloudProvider: onPrem
    environment: lower
    externalIngress: false
    clusterSubDomain: sandboxop
    vanitydomains:
      - mrcooper.com
      - nationstarmtg.net
```

Iassist AKS-

```
resources:
  repositories:
    - repository: templates
      type: git
      name: "Infrastructure Automation/Terraform.Modules.Azure.K8s-Base"
      ref: refs/heads/module_split

trigger: none
extends:
  template: config-pipelines/cluster-config.yaml@templates
  parameters:
   clusterName: i_____-lower
   cloudProvider: azure
   externalIngress: true
   environment: lower
   clusterSubDomain: i_____
```

thanks,

Kenneth S[redacted]

Senior Infrastructure Engineer

Server Automation

8950 Cypress Waters Blvd.

Dallas, TX 75019