



**HCL Connections Product  
Documentation**

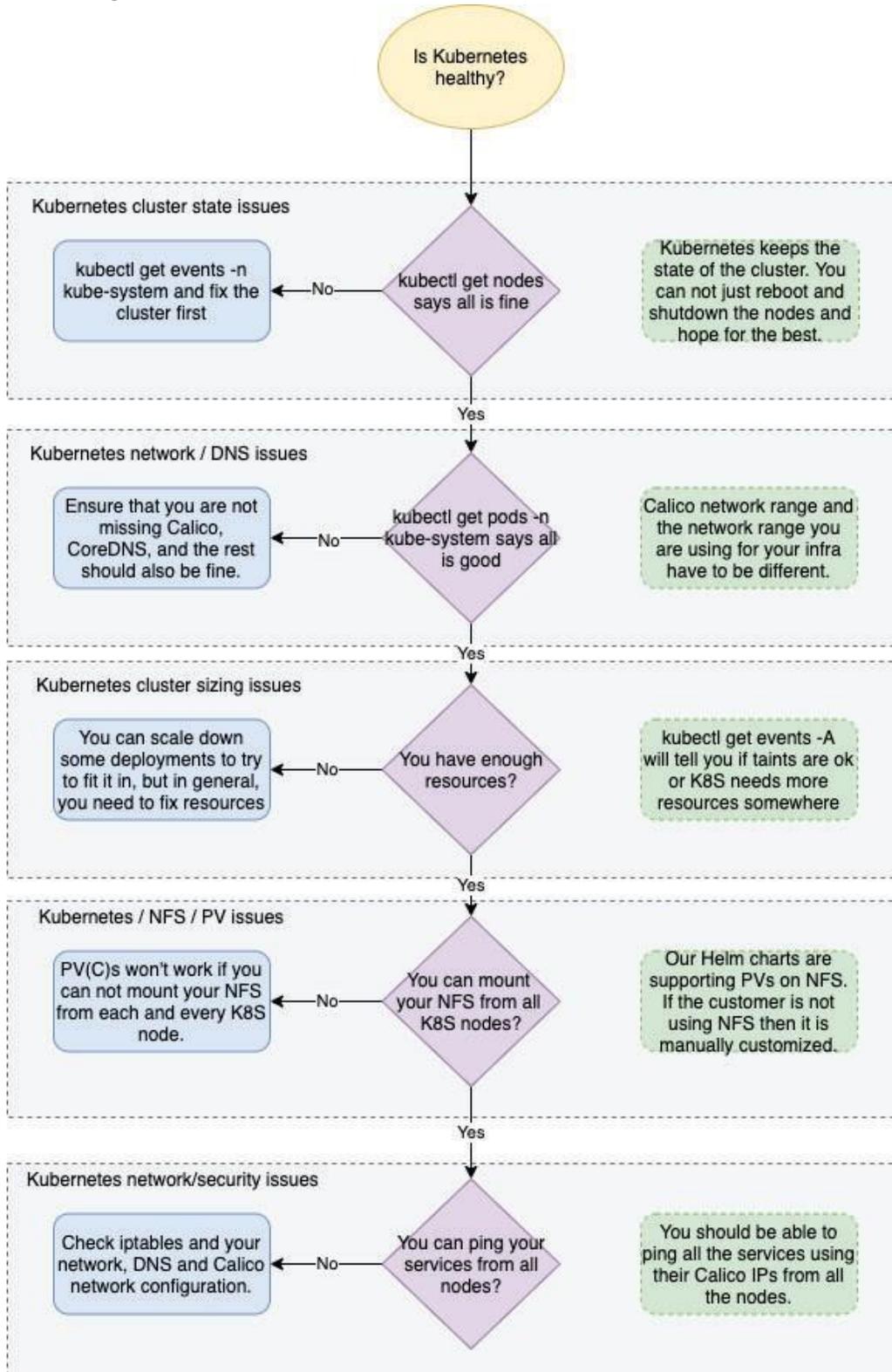
**Troubleshooting Component Pack**



# Troubleshooting Component Pack for HCL Connections

## Troubleshooting Order

### Troubleshooting Kubernetes Cluster



### *Kubernetes cluster state issues*

Checking the nodes and events log is your very best starting point. All the nodes that Kubernetes knew about should be, if possible, in a Ready state. If cluster was high available, and quorum cannot be established again, this needs to be resolved first.

If lot of pods are in a Terminating state for a long time, this also needs to be cleaned up, and check if Docker itself is properly up and running everywhere, as it can indicate that Docker crashed somewhere.

Of course, ensure that you have enough disk space on all the nodes. No more disk space can make lot of crazy issues.

### *Kubernetes network / DNS issues*

The rule of thumb here is to:

- Ensure that `kubectl get pods -n kube-system` show you clean state. But you probably wouldn't read this if it does.
- CoreDNS pods will not start before Calico controller and node pods are up and running. Note that you will have one Calico node pod for each cluster member.
- Calico will fail on so many crazy ways if the Calico network (by default 192.168.0.0/16) is overlapping with your Kubernetes network (meaning your nodes are running on the same network). The most prominent issue is DNS resolution failing all around.
- You should not try to do magic with customizing subnetting for your Calico network. You should be able to ping from each Kubernetes host each service by its Calico IP address internally.
- You should be able to resolve both internal DNS and external DNS, from your hosts as from Kubernetes nodes internally. Ensure that no firewall is blocking anything related to this. Also ensure that from both Kubernetes hosts **and** from inside the pods you are getting proper resolution of the site name (for example, if you are troubleshooting Kubernetes used for `preview.hclconnections.net`, you need to be able to resolve properly `preview.hclconnections.net` from Kubernetes nodes and from inside the pods).
- You should be able to reach out your site using HTTPS from internal network.

### *Kubernetes cluster sizing issues*

There are two ways to check this:

- `kubectl get events -A` and see if Kubernetes is complaining that it cannot schedule something because something is missing.
  - Kubernetes will complain specifically about missing CPU and memory. What will be a bit more cryptical is disk pressure. You need to ensure that you have enough disk space on nodes and that it is performant enough.
- Seeing that stuff is not being scheduled for a long time but then out of nowhere it gets scheduled. Also seeing that some things are not scheduled at all is quite clear indicator.
  - Go to the first step, and figure out what is missing.

The rule of thumb on fixing the above issues is not to run more than one replica on one worker. That means don't run three replicas of each pod on a single worker.

If you are still having the issues, then you need to calculate better the performance of your Kubernetes infrastructure.

### *Kubernetes NFS/PV issues*

The most prominent symptoms for those are:

- Deployments like Mongo and ElasticSearch will never come fully up.
- Error messages in those deployments saying that cannot mount something somewhere.

To ensure it works as expected:

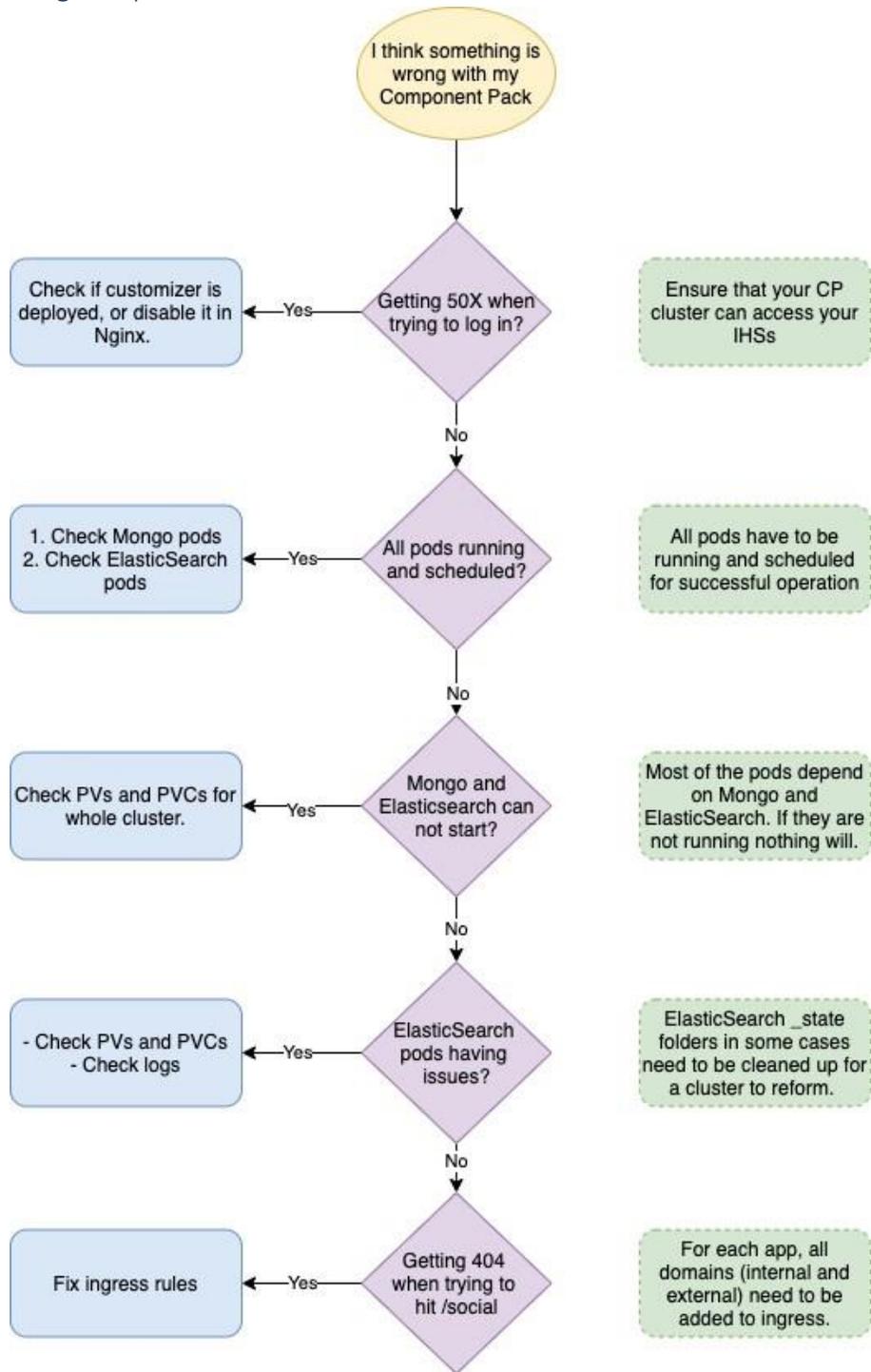
- Ensure that from each Kubernetes node you can mount your data from the NFS server you are using. Simply do manual mounts and fix if something is not working. Once it is fixed, pods will mount it automatically.
- Ensure that NFS itself is accepting requests (on a network level) from your Kubernetes cluster. You would discover this in the previous step as well.
- Check PVs with `kubectrl get pv -n connections`
  - For each PV that it lists, describe it, and see where does it expect it, and check if those paths exist exactly like that on you NFS server. It can happen that folders are not created or folder names are wrong, which is the same thing.
  - All states should be fine.
- Check PVCs with `kubectrl get pvc -n connections`
  - For each PVC check corresponding PV.
- Ensure that the ownership and permission of the folders are correctly set according to the Connections administrator documentation.

### *Kubernetes network/security issues*

From the network security point of view, please ensure that:

- All Kubernetes nodes can talk to each other and to the load balancer they are using in front of the masters. If this is not possible question is how Kubernetes installation was actually done.
- On each node, check iptables. Calico is using iptables to manage its own states. If you are changing/updating Calico rules, flush iptables once done, so Calico can recreate new state.
- Ensure all the ports used by ingresses are properly open and set.
- Ensure that Kubernetes cluster can talk on a local level with Connections and DB servers.

## Troubleshooting Component Pack installation



### *Component Pack deployed, I'm trying to log in, but getting 50x*

Check if Customizer is deployed and properly configured. The easiest way to narrow this down to Customizer is to go to Nginx and disable sending requests to Customizer. If after that log in works fine, continue troubleshooting Customizer. If you still have the same issue, leave the Customizer disabled and troubleshoot the cluster.

*After installation not all pods are running and scheduled*

Check pods logs, check event logs, but also do the steps from troubleshooting Kubernetes cluster. If Kubernetes cluster is healthy, and you don't have any issues with PVs and PVCs everything should work fine.

Specifically, pay attention to Mongo pod. If Mongo is not up and running from whatever reason, there is not point proceeding until it is fixed, since Mongo will also be needed to log in properly to OrientMe.

ElasticSearch is another key component to check here, because without it some core features would not work, including OrientMe.

Bunch of services are depending on Mongo itself. Fixing Mongo will fix them as well. Same goes with ElasticSearch.

*All looks fine but not Mongo and ElasticSearch*

It is 99.999% PV(C) issue. Troubleshoot it carefully as described above.

*All is finally good, but posting is not working in OrientMe, and ElasticSearch logs are showing some crazy stuff*

ElasticSearch is keeping metrics and search data. Losing search data is fine, it will be regenerated, but metrics will not. If you can allow yourself to get rid of the data, clean up PVs for ES and delete all ES pods (which will recreate them). This should fix the issue.

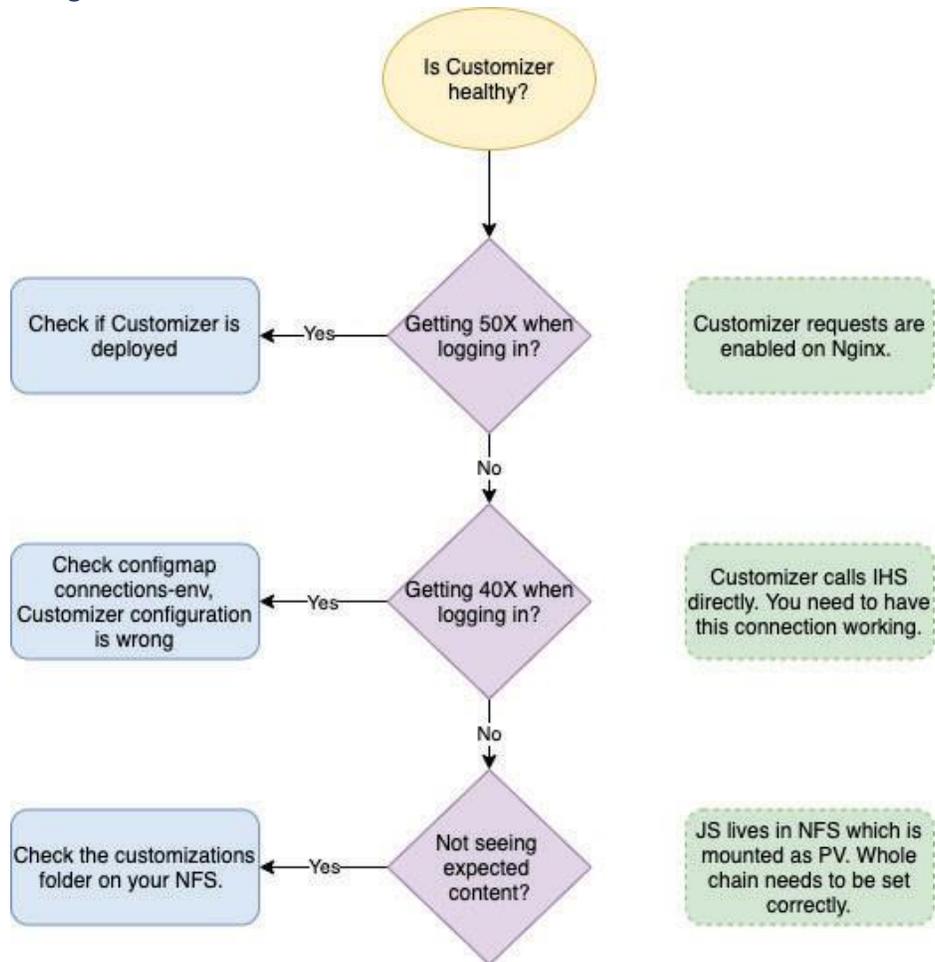
If you still want to keep the data, you need to fix the cluster state. In all ES data PVs find the folder called `_state` and remove it. Then remove all ES pods and let them recreate. This will force ES to recreate the cluster but keep the data.

*Getting 404 when trying to hit /social, but otherwise Customizer is enabled and works fine*

Check ingress rules with **`kubectrl get ingress -n connections`**

You should have both frontend and internal domains enabled there for all the ingresses (for preview.hclconnections.net frontend would be \*.hclconnections.net, and for the cluster running on the node called for example kubernetes.internal.somedomain.somewhere.com the domain would be \*.internal.somedomain.somewhere.com; both need to be enabled, and 404 mostly means that you have only internal enabled but not external, as internal gets enabled by default).

## Troubleshooting Customizer



*Customizer is enabled, and I am getting 50x when trying to log in*

See *Component Pack deployed*, I am trying to log in but getting 50x section for this.

*Logging in takes forever and ends up with 40X HTTP response*

Check out your connections-env configmap.

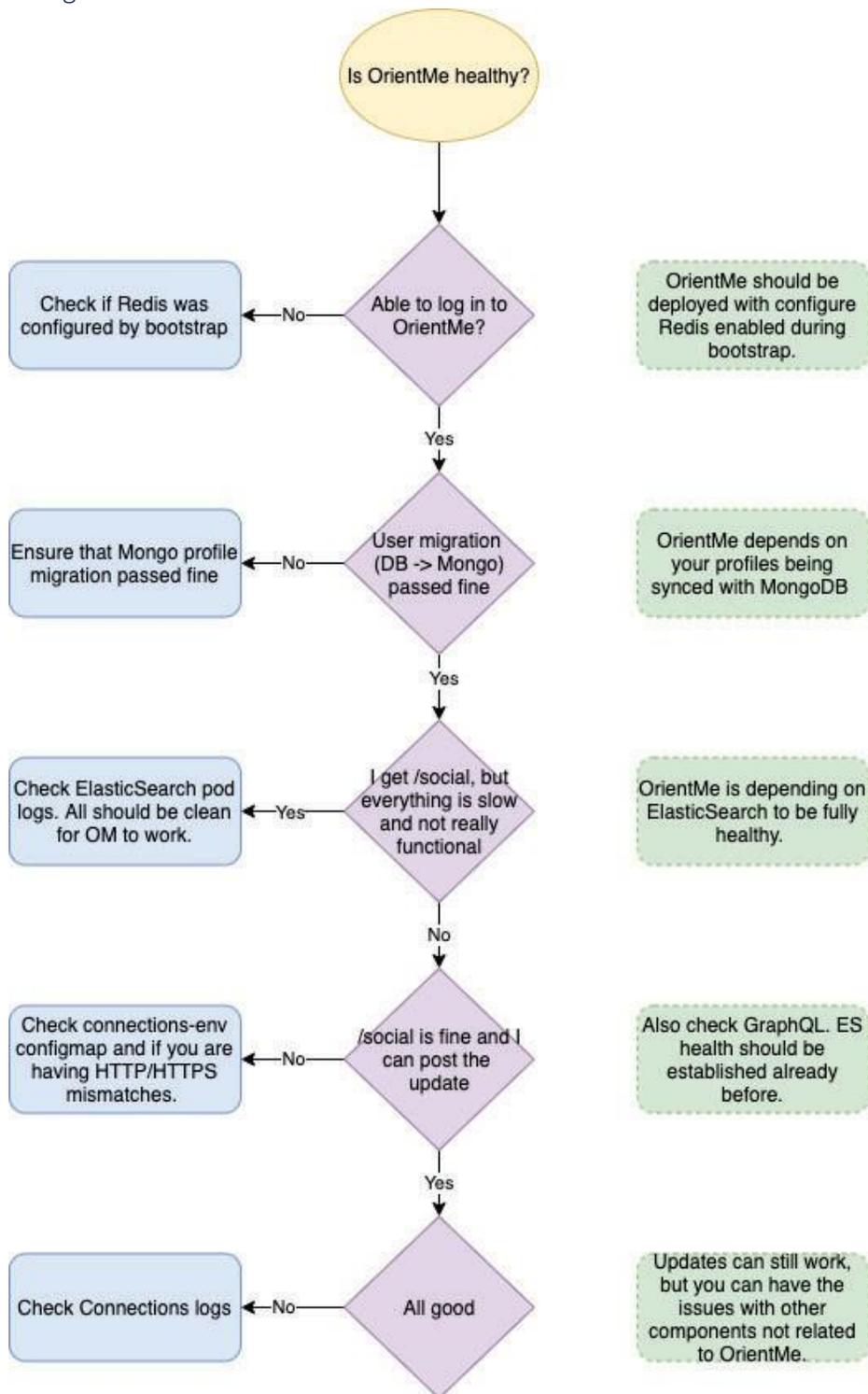
Customizer should be pointed to the internal name of IHS server exclusively (will not work with external, as 40X mostly means that you are in the loop which you would get if pointing to external name). Also, you should ensure it is pointing to IHS using https and port 443, and that it is properly enabled on anotehr side and HTTPS enforced.

*Not seeing expected content*

Customizer is also using PVs and reading stuff from there.

Start with troubleshooting PV(C)s for Customizer, and check if your content is in its folder on NFS.

## Troubleshooting OrientMe



### *Getting some strange error from IHS when hitting /social*

This means that either OrientMe wasn't deployed at all, or if it was, bootstrap was run `env.skip_configure_redis=true` (it should be false to enable everything for OrientMe).

You can check the logs of the bootstrap run, and the bottom of it would tell you if it is enabled for OrientMe or not.

The easiest way to fix this is to run bootstrap again with a proper parameter and reinstall Infrastructure and OrientMe after that.

*Getting that cryptic "go talk to your administrator" error when I try to visit /social*

This means that you need to sync your profiles from whatever relational DB you are using to Mongo.

*OrientMe seems very slow and not really functional, cannot post the update*

Check Connections logs, and check Elasticsearch. Elasticsearch logs should be all good and ES clustered. Connections should be also properly up and running.

Check also connections-env configmap for the parameters used by OrientMe. Check if there is HTTP/HTTPS mismatch.

Inside the configmap, please note that OrientMe, unlike Customizer, should point to the dynamicHost url, not internal IHS FQDN/load balancer.

*OrientMe looks to be working fine, but I am seeing some messages popping up*

Check Connections logs. This is probably not related to OrientMe but to some related app(s).