VCD Data Solutions extension has components in these places:

- Data Solutions UI plugin in vCD portal
- RDE (Runtime Defined Entities) schemas and instances in vCD database
- DSO (Data Solutions operator) in guest TKGm

For troubleshooting, we need to collect data from both frontend and backend sides.

**Check Points**

RabbitMQ instances are deployed to TKGm clusters provisioned by CSE, so it’s important to understand TKGm health and resource status. Here are a few check points.

**Check cluster status**

![Kubernetes Cluster Status](image)

**Check cluster events**
Check all PODs in TKGm cluster namespaces

Mandatory namespaces:

- tkg-system (kapp-controller)
- tanzu-system (secretgen-controller)
- cert-manager
- rdeprojector-system
- vcd-ds-system

Optional namespaces:

- tanzu-system-dashboards
- tanzu-system-monitoring
- tanzu-system-ingress

Command Reference:

- `kubectl get pods <pod-name> -n <namespace>`
- `kubectl describe pods <pod-name> -n <namespace>`
- `kubectl logs -f -tail 100 pods <pod-name> -n <namespace>`

Frontend Troubleshooting

If you could enable the browser's inspection mode and capture the XHR messages send to the server, it would help much for the engineering team to identify the root cause.
**Backend Troubleshooting**

**Installation**
Follow the [guide](https://gitlab.eng.vmware.com/core-build/vcd-tds/-/blob/main/troubleshoot/preflight.yaml) to install the troubleshoot tools. If you're in an air-gapped environment, you need to copy the binaries to the host where kubectl is installed.

**Preflight**
To do the preflight check, run:

```
kubectl preflight /preflight.yaml
```

Support bundle
To generate the support bundle, run:

```
kubectl support-bundle ./support-bundle.yaml
```


CSE support bundle