



Turbine Upgrade Guide

CONTENTS

1. Pre-Upgrade Health Check	2
2. Post-Upgrade/Maintenance Checklist	1
3. Turbine 26.4.1 Upgrade Instructions	1
4. Turbine 26.3.3 Upgrade Instructions	1
5. Turbine 26.2.3 Upgrade Instructions	1
6. Turbine 26.1.4 Upgrade Instructions	1
7. Turbine 26.1.3 Upgrade Instructions	1
8. Turbine 26.1.2 Upgrade Instructions	1
9. Turbine 26.1.1 Upgrade Instructions	1
10. Turbine 26.1.0 Upgrade Instructions	1
11. Turbine 26.0.3 Upgrade Instructions	3
12. Turbine 25.5.1 Upgrade Instructions	1
13. Turbine 25.5.0 Upgrade Instructions	1
14. Turbine 25.4.3 Upgrade Instructions	1
15. Turbine 25.4.2 Upgrade Instructions	1
16. Turbine 25.4.1 Upgrade Instructions	1
17. Turbine 25.3.4 Upgrade Instructions	1
18. Turbine 25.3.3 Upgrade Instructions	1

1. Pre-Upgrade Health Check

Node Status

- Check that all Nodes are in Ready state

```
kubectl get nodes -owide
```

- If any nodes are in NotReady, SchedulingDisabled, etc then it would be recommended to review and fix them before upgrading.

Pod Status

- Check that all pods are in Running state and the pod values are matching 1/1, 3/3, etc.

```
kubectl get pods -A
```

- If any pods are not running i.e, 0/1, 2/3, CrashLoopBackoff, Error, etc then it would be advised to review the issue and fix it prior to upgrading

Cluster Status

- Check that Kubernetes is up and running and healthy.

```
kubectl cluster-info
```

- If Kubernetes indicates that it is not healthy then review and establish where the issue is prior to upgrading.

etcd health

- Turbine 26.0.x and earlier

Use this command when the cluster is on Turbine 26.0.x or any earlier release

```
for pod in $(kubectl get pods -l component=etcd -n kube-system \
-o jsonpath='{.items[*].metadata.name}')
do
    echo "### etcd pod : ${pod} ###"
    kubectl -n kube-system exec ${pod} -- /bin/sh \
    -c "ETCDCTL_API=3 etcdctl \
    --cert=/etc/kubernetes/pki/etcd/server.crt \
    --key=/etc/kubernetes/pki/etcd/server.key \
    --cacert=/etc/kubernetes/pki/etcd/ca.crt \
    endpoint health"
done
```

- Turbine 26.1.0 and later

Use this command when the cluster is on Turbine 26.1.0 or any later release.

That image is distroless and does not include /bin/sh. Call etcdctl directly instead.

```
for pod in $(kubectl get pods -l component=etcd -n kube-system \
-o jsonpath='{.items[*].metadata.name}')
do
    echo "### etcd pod : ${pod} ###"
```

11. Turbine 26.0.3 Upgrade Instructions

Use these instructions to upgrade your instance to Turbine 26.0.3. The instructions are organized into subsections of this document that will help you upgrade based on your installation type, whether you have an embedded or existing cluster installation.

End of Support for Red Hat Enterprise Linux 8 (RHEL 8)

RHEL 8 is no longer a supported operating system for Turbine. Beginning with Turbine 26.0.3, Kubernetes is updated to version 1.33, which does not support RHEL 8 (specifically, Linux kernel versions 4.18 and earlier). All customers running RHEL 8 must upgrade to RHEL 9 before upgrading to Turbine version 26.0.3 or greater.

You can use the following to determine your host's current RHEL version:

```
cat /etc/redhat-release
```

Mandatory Prerequisites to Upgrade MongoDB to Version 8.0

Before upgrading MongoDB to version 8.0, perform the following steps to check the current configuration and update it if needed.

1. Verify MongoDB Compatibility

1.1 Connect to MongoDB as an Admin (replace <namespace> with your actual namespace)

```
kubectl exec -it -n <namespace> mongo-0 -- mongosh -u Admin -p --  
authenticationDatabase admin --tls --tlsAllowInvalidCertificates admin
```

NOTE: You will be prompted to enter your MongoDB admin password.

1.2 Check the current MongoDB version (it should display 7.0.x)

```
use admin  
  
db.version()  
db.system.version.find()
```

1.3 Verify the Feature Compatibility Version

1.4 Update the Feature Compatibility Version (if required)

If the version does not include 7.0, update it:

```
db.adminCommand({ setFeatureCompatibilityVersion: '7.0' })
```

Upgrade Turbine on Embedded Cluster Installations

Use the following instructions to upgrade Turbine on an embedded cluster installation.

21. Turbine 25.1.9 Upgrade Instructions

Turbine Upgrade Instructions

Use these upgrade instructions to upgrade your instance of Turbine 25.1.9. The instructions are organized by subsections of this document that will help you upgrade by specific installation type, whether you have an embedded or existing cluster installation.

Known Issues

Kubernetes Version Preflight Check Is Misleading

- **Issue:** The Kubernetes version preflight check shows a misleading message. The supported range for existing Kubernetes cluster installations of Swimlane is 1.27 – 1.31.7.

Upgrade Turbine on Embedded Cluster Installations

Use the following instructions to upgrade this version of Swimlane on an embedded cluster installation.

Prerequisites:

- Turbine version 25.0.8
- MongoDB 7.0.19

Note: If you are already on version 25.1.2, no infrastructure upgrade is required. Proceed to step 4

Important! Swimlane recommends that you perform a MongoDB backup prior to starting this update. For more information, see [Backup and Restore on an Embedded Cluster with Snapshots](#).

Upgrade Instructions

1. SSH into any node in your deployment.
2. Scale down EKCO and kotsadm-rglite.

Note: EKCO and Kotsadm-rglite will automatically scale back up after the script in Step 3 has completed.

```
kubectl -n kurl scale deployment ekc-operator --replicas=0
```

```
kubectl -n default scale statefulsets kotsadm-rglite --replicas=0
```

3. Execute the following command to begin the Turbine Platform Installer component upgrade:

```
curl -sSL https://kurl.sh/turbine-turbine-25-1-9 | sudo bash -s ha
```

Note: If you have configured SELinux, firewall, or proxy add this to the end of the command above:

```
installer-spec-file=se.yaml
```

31. Turbine 25.0.4 Upgrade Instructions

Turbine Upgrade Instructions

Use these upgrade instructions to upgrade your instance of Turbine 25.x. The instructions are organized by subsections of this document that will help you upgrade by specific installation type, whether you have an embedded or existing cluster installation.

Upgrade Turbine on Embedded Cluster Installations

Use the following instructions to upgrade this version of Swimlane on an embedded cluster installation.

Prerequisites:

- Turbine version 24.3.2
- MongoDB 6.0.18

Important! Swimlane recommends that you perform a MongoDB backup prior to starting this update. For more information, see [Backup and Restore on an Embedded Cluster with Snapshots](#).

Upgrade Instructions

1. SSH into any node in your deployment.
2. Scale down EKCO and kotsadm-qlite.

Note: EKCO and Kotsadm-qlite will automatically scale back up after the script in Step 3 has completed.

```
kubectl -n kurl scale deployment ekc-operator --replicas=0  
  
kubectl -n default scale statefulsets kotsadm-qlite --replicas=0
```

3. Execute the following command to begin the Turbine Platform Installer component upgrade:

```
curl -sSL https://kurl.sh/turbine-turbine-25-0-4 | sudo bash -s ha
```

NOTE: If you have configured SELinux, add this to the end of the command above:

```
installer-spec-file=se.yaml
```

4. After the install script has completed, log in to the Turbine Platform Installer dashboard (<https://<SwimlaneDNS>:8800>).
5. Go to the Version History tab and click **Check For Updates**.
6. A pop-up window appears, click **Go to the updated version** and click Configure.
7. Click **Deploy** on the new 25.0.4 version with the highest sequence number at the top of the versions list.
8. Once the Application tab shows 'Ready' your upgrade is complete.

Upgrade Turbine on Existing Cluster Installations

41. Turbine 23.5.7 Upgrade Instructions

Use these upgrade instructions to upgrade your instance of Turbine 23.x. The instructions are organized by subsections of this document that will help you upgrade by specific installation type, whether you have an embedded or existing cluster installation.

Known Issue with Embedded/Existing Cluster/helm Installations:

We are aware of an issue where the user is 'unable to generate a URL while creating a webhook'. There is a workaround for that for now. This script which can be used on all deployment methods, for KOTS (existing or embedded clusters) and for helm deployments. Contact Support for additional details.

Prerequisites:

- Requires a fully functional jq software version 1.6 and all dependent packages.

Instructions:

1. Set the value for the NS variable in line 5 to the Kubernetes namespace where Turbine has been installed.

1. KOTS existing cluster and Helm deployments allow setting the namespace where the app is installed.
2. KOTS embedded cluster should be set to "default".

2. Copy the script to a computer where you have access to the Kubernetes cluster.

```
#!/bin/bash
set -e
# Set the Kubernetes namespace where Turbine is installed, this should be
"default" for a Turbine installation with an embedded cluster
# NS=default
NS=
[[ -z $NS ]] && echo "This script requires that you set the name of the
Kubernetes namespace where Turbine is installed in the NS variable." && exit 1

configmap=$(kubectl -n $NS get configmap tenant-services -o
jsonpath='{.data.turbine\-\init\-\tenant\-\job\.yaml}')
tenants=$(kubectl get ns -l "app.kubernetes.io/name=\"tenant-services\" -o json
| jq -r '.items[] | .metadata.labels.\"tenant.swimlane.io/account-id\" + \";\" +
.metadata.labels.\"tenant.swimlane.io/tenant-id\"')
IFS=$'\n'
for i in $tenants;do
accountid=$(echo $i | cut -d";" -f1)
tenantid=$(echo $i | cut -d";" -f2)
yaml=$(echo "$configmap" | sed -r -e "s/REPLACE_ACCOUNT_ID/${accountid}/g" -e
"s/REPLACE_TENANT_ID/${tenantid}/g")
echo "$yaml" | kubectl -n tenant$tenantid apply -f -
done
```

3. Run the script on the host with a command like "chmod +x script.sh && ./script.sh"
4. The script will show the output of the kubectl command which applies the job manifest for each tenant

```
$ chmod +x script.sh && ./script.sh
job batch/tenant-init-701d3c50-1000-4006-0445-023009560d10 created
```