# Quickstart: Deploy a Neo4j instance to a local Kubernetes installation via Docker Desktop for Mac

*A quick start guide for deploying a Neo4j instance to a local Kubernetes installation (via Docker Desktop for Mac OS) using Neo4j Helm charts.*

## 1. Prerequisites

* Verify that you have [configured the Neo4j Helm charts and created a release name for your Neo4j instance](https://neo4j.com/docs/operations-manual/current/kubernetes/helm-charts-setup/).
* Verify that you have installed Docker Desktop for Mac. For more information, see [Docker official documentation](https://docs.docker.com/docker-for-mac/install/).
* Enable the Docker Desktop Kubernetes engine. For more information, see [Docker official documentation](https://docs.docker.com/desktop/kubernetes/).
* Verify that you have sufficient CPU and RAM for your Neo4j deployment. For more information, see [System requirements](https://neo4j.com/docs/operations-manual/current/installation/requirements/).
* Verify that you do not have a running instance of Neo4j (e.g., via Neo4j Desktop or Neo4j Browser) to avoid port clashes.
* Verify that you have a valid license if you want to install Neo4j Enterprise Edition. For more information, see <https://neo4j.com/licensing/> or write to licensing@neo4j.com.

## 2. Create a Helm deployment values file

Create a new file *my-neo4j.values.yaml* with the following content:

Yaml

Copy to Clipboard

neo4j:

resources:

cpu: "1"

memory: "2Gi"

*# Uncomment to set the initial password*

*#password: "my-initial-password"*

*# Uncomment to use enterprise edition*

*#edition: "enterprise"*

*#acceptLicenseAgreement: "yes"*

volumes:

data:

mode: defaultStorageClass

defaultStorageClass:

requests:

storage: 2Gi

For details of all Neo4j Helm chart configuration options, see [Configure and install Neo4j using a customized Helm chart](https://neo4j.com/docs/operations-manual/current/kubernetes/configuration/).

| By default, the helm chart installs Neo4j Community Edition. If you want to install Neo4j Enterprise Edition, uncomment the configuration parameters edition: "enterprise" and acceptLicenseAgreement: "yes" in *my-neo4j.values.yaml*. |
| --- |

## 3. Create a Neo4j instance using dynamically provisioned storage

1. Ensure your Helm Chart repositories are up to date:
2. Shell
3. Copy to Clipboard
4. helm repo update
5. Install Neo4j using the deployment values file created in [Create a Helm deployment values file](https://neo4j.com/docs/operations-manual/current/kubernetes/quickstart-docker-desktop/#dd-create-values-file):
6. Shell
7. Copy to Clipboard
8. helm install my-neo4j-release neo4j/neo4j-standalone -f my-neo4j.values.yaml

NAME: my-neo4j-release

LAST DEPLOYED: Thu Jun 10 10:43:01 2021

NAMESPACE: default

STATUS: deployed

REVISION: 1

TEST SUITE: None

NOTES:

Thank you for installing neo4j.

Your release "my-neo4j-release" has been installed .

To view the progress of the rollout try:

$ kubectl rollout status *--watch --timeout=600s statefulset/my-neo4j-release*

Once rollout is complete you can log in to Neo4j at "neo4j://my-neo4j-release.default.svc.cluster.local:7687". Try:

$ kubectl run *--rm -it --image "neo4j:4.3.2" cypher-shell \*

*-- cypher-shell -a "neo4j://my-neo4j-release.default.svc.cluster.local:7687" -u neo4j*

1. Graphs are everywhere!
2. The command creates a Neo4j StatefulSet that relies on the default Kubernetes StorageClass to dynamically create a persistent volume. Generally speaking, when using Docker Desktop this volume will not survive a Kubernetes restart.
3. Run the kubectl rollout command provided in the output of helm install to watch the Neo4j’s rollout until it is complete.
4. Shell
5. Copy to Clipboard
6. kubectl rollout status --watch --timeout=600s statefulset/my-neo4j-release

Since you have not passed a password for the neo4j user, the Neo4j Helm chart has set an automatically generated one. You can find it in the Helm install output. Please make a note of it.

## 4. Verify the installation

1. Check that statefulset is OK. Initially it will not be ready but just check there is something there.
2. Shell
3. Copy to Clipboard
4. kubectl get statefulsets

NAME READY AGE

1. <release-name> 1/1 5m11s
2. Check that the PVC is OK (the STATUS must be Bound):
3. Shell
4. Copy to Clipboard
5. kubectl get pvc

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

1. data-<release-name>-0 Bound <release-name>-pv 10Gi RWO manual 8m36s
2. Check that the pod is READY:
3. Shell
4. Copy to Clipboard
5. kubectl get pods

NAME READY STATUS RESTARTS AGE

1. <release-name>-0 1/1 Running 0 5m53s
2. Check that the pod logs look OK:
3. Shell
4. Copy to Clipboard
5. kubectl exec <pod-name> -- tail -n50 /logs/neo4j.log

Changed password for user 'neo4j'.

Directories in use:

home: /var/lib/neo4j

config: /config/

logs: /data/logs

plugins: /var/lib/neo4j/plugins

import: /var/lib/neo4j/import

data: /var/lib/neo4j/data

certificates: /var/lib/neo4j/certificates

run: /var/lib/neo4j/run

Starting Neo4j.

2021-06-02 17:38:27.791+0000 INFO Command expansion is explicitly enabled for configuration

2021-06-02 17:38:27.819+0000 INFO Starting...

2021-06-02 17:38:31.195+0000 INFO ======== Neo4j 4.3.2 ========

2021-06-02 17:38:34.168+0000 INFO Initializing system graph model for component 'security-users' with version -1 and status UNINITIALIZED

2021-06-02 17:38:34.188+0000 INFO Setting up initial user from `auth.ini` file: neo4j

2021-06-02 17:38:34.190+0000 INFO Creating new user 'neo4j' (passwordChangeRequired=false, suspended=false)

2021-06-02 17:38:34.205+0000 INFO Setting version for 'security-users' to 2

2021-06-02 17:38:34.214+0000 INFO After initialization of system graph model component 'security-users' have version 2 and status CURRENT

2021-06-02 17:38:34.223+0000 INFO Performing postInitialization step for component 'security-users' with version 2 and status CURRENT

2021-06-02 17:38:34.561+0000 INFO Bolt enabled on 0.0.0.0:7687.

2021-06-02 17:38:36.910+0000 INFO Remote interface available at http://localhost:7474/

1. 2021-06-02 17:38:36.912+0000 INFO Started.
2. Check that the services look OK:
3. Shell
4. Copy to Clipboard
5. kubectl get services

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 3d1h

my-neo4j-release ClusterIP 10.103.103.142 <none> 7687/TCP,7474/TCP,7473/TCP 2d8h

my-neo4j-release-admin ClusterIP 10.99.11.122 <none> 6362/TCP,7687/TCP,7474/TCP,7473/TCP 2d8h

1. my-neo4j-release-neo4j LoadBalancer 10.110.138.165 localhost 7474:31237/TCP,7473:32026/TCP,7687:32169/TCP 2d3h
2. Use port forwarding to get access to the browser:
3. Shell
4. Copy to Clipboard
5. kubectl port-forward svc/<release-name> tcp-bolt tcp-http tcp-https
6. In a web browser, open the Neo4j Browser at [*http://localhost:7474*](http://localhost:7474/).
7. Use the automatically generated password (as printed in the output of the helm install command) or the one you have set up with the helm install command.

## 5. Uninstall Neo4j and clean up your Docker Desktop

### **5.1. Uninstall Neo4j Helm deployment**

1. Uninstall Neo4j Helm deployment.
2. Shell
3. Copy to Clipboard
4. helm uninstall <release-name>
5. release "<release-name>" uninstalled
6. Check the name of the PersistentVolumeClaim (pvc):
7. Shell
8. Copy to Clipboard
9. kubectl get pvc

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

1. data-<release-name>-0 Bound <release-name>-pv 1Ti RWO manual 43h

| If you re-create Neo4j with the same settings, it will pick up the PVC again, and all the data is still on it.  When you use dynamically provisioned volumes and delete the PersistentVolume, the underlying data may or may not be removed, depending on the Docker Desktop version and configuration. |
| --- |

### **5.2. Fully remove all the data and resources**

To fully remove all the data and resources, delete the PersistentVolumeClaim in Kubernetes.

| The dynamically provisioned volumes are automatically removed when the PersistentVolumeClaim is deleted. |
| --- |

Shell

Copy to Clipboard

kubectl delete pvc <pvc-name>

Shell

Copy to Clipboard

persistentvolumeclaim "data-<release-name>-0" deleted