Yes, you can perform these steps using the OpenShift Console. Here’s how you can manually deploy PostgreSQL and Nexus Repository Manager through the OpenShift Console:

### Steps to Deploy PostgreSQL:

1.Create a Namespace for Nexus\*\*:

1.Go to the OpenShift Console\*\*.

2.Navigate to\*\* `Home > Projects`.

3.Click\*\* `Create Project`.

4.Enter\*\* the name (e.g., `nexus`) andclick\*\* `Create`.

2.Create a PersistentVolumeClaim for PostgreSQL\*\*:

1.Go to\*\* `Workloads > PersistentVolumeClaims`.

2.Select\*\* the `nexus` namespace.

3.Click\*\* `Create PersistentVolumeClaim`.

4.Fill in the details\*\*:

-Name\*\*: `postgres-pvc`

-Access Modes\*\*: `ReadWriteOnce`

-Size\*\*: `10Gi`

5.Click\*\* `Create`.

3.Create a Deployment for PostgreSQL\*\*:

1.Go to\*\* `Workloads > Deployments`.

2.Select\*\* the `nexus` namespace.

3.Click\*\* `Create Deployment`.

4.Fill in the details\*\*:

-Name\*\*: `nexus-postgresql`

-Image\*\*: `postgres:13`

5.Under `Environment Variables`\*\*, add:

-POSTGRES\_DB\*\*: `nexusdb`

-POSTGRES\_USER\*\*: `nexus`

-POSTGRES\_PASSWORD\*\*: `nexuspassword`

6.Under `Volumes`\*\*, add:

-Volume Name\*\*: `postgres-storage`

-Volume Type\*\*: `PersistentVolumeClaim`

-Claim Name\*\*: `postgres-pvc`

7.Under `Mounts`\*\*, add:

-Mount Path\*\*: `/var/lib/postgresql/data`

-Volume Name\*\*: `postgres-storage`

8.Click\*\* `Create`.

4.Create a Service for PostgreSQL\*\*:

1.Go to\*\* `Networking > Services`.

2.Select\*\* the `nexus` namespace.

3.Click\*\* `Create Service`.

4.Fill in the details\*\*:

-Name\*\*: `nexus-postgresql`

-Selector\*\*: `app=nexus-postgresql`

-Port Name\*\*: `postgres`

-Port\*\*: `5432`

-Target Port\*\*: `5432`

5.Click\*\* `Create`.

### Steps to Deploy Nexus Repository Manager:

5.Create a PersistentVolumeClaim for Nexus\*\*:

1.Go to\*\* `Workloads > PersistentVolumeClaims`.

2.Select\*\* the `nexus` namespace.

3.Click\*\* `Create PersistentVolumeClaim`.

4.Fill in the details\*\*:

-Name\*\*: `nexus-pvc`

-Access Modes\*\*: `ReadWriteOnce`

-Size\*\*: `10Gi`

5.Click\*\* `Create`.

6.Create a Deployment for Nexus Repository Manager\*\*:

1.Go to\*\* `Workloads > Deployments`.

2.Select\*\* the `nexus` namespace.

3.Click\*\* `Create Deployment`.

4.Fill in the details\*\*:

-Name\*\*: `nexus`

-Image\*\*: `sonatype/nexus3:latest`

5.Under `Environment Variables`\*\*, add:

-NEXUS\_DB\_HOST\*\*: `nexus-postgresql`

-NEXUS\_DB\_PORT\*\*: `5432`

-NEXUS\_DB\_NAME\*\*: `nexusdb`

-NEXUS\_DB\_USER\*\*: `nexus`

-NEXUS\_DB\_PASSWORD\*\*: `nexuspassword`

6.Under `Volumes`\*\*, add:

-Volume Name\*\*: `nexus-storage`

-Volume Type\*\*: `PersistentVolumeClaim`

-Claim Name\*\*: `nexus-pvc`

7.Under `Mounts`\*\*, add:

-Mount Path\*\*: `/nexus-data`

-Volume Name\*\*: `nexus-storage`

8.Click\*\* `Create`.

7.Create a Service for Nexus Repository Manager\*\*:

1.Go to\*\* `Networking > Services`.

2.Select\*\* the `nexus` namespace.

3.Click\*\* `Create Service`.

4.Fill in the details\*\*:

-Name\*\*: `nexus`

-Selector\*\*: `app=nexus`

-Port Name\*\*: `http`

-Port\*\*: `8081`

-Target Port\*\*: `8081`

5.Click\*\* `Create`.

8.Create a Route for Nexus Repository Manager\*\*:

1.Go to\*\* `Networking > Routes`.

2.Select\*\* the `nexus` namespace.

3.Click\*\* `Create Route`.

4.Fill in the details\*\*:

-Name\*\*: `nexus`

-Service\*\*: `nexus`

-Target Port\*\*: `8081`

5.Click\*\* `Create`.

### Verify the Deployment:

1.Check the status of the PostgreSQL and Nexus pods\*\*:

- Go toWorkloads > Pods\*\* and select the `nexus` namespace.

- Ensure that both PostgreSQL and Nexus pods are running and healthy.

2.Access Nexus Repository Manager\*\*:

- Go toNetworking > Routes\*\* and select the `nexus` namespace.

- Click on the URL for the Nexus route.

- Log in using the default credentials (`admin` / `admin123`) or any custom credentials you set during setup.

By following these steps using the OpenShift Console, you can manually deploy PostgreSQL and Nexus Repository Manager without using a PostgreSQL operator. If you encounter any issues or need further assistance, feel free to ask!