Deploy Digital Channels using Docker Compose

2/5/2026

Contents

Learn how to deploy Digital Channels as a service in a non-production lab environment using Docker Compose. To deploy Digital Channels in a Kubernetes cluster for a production or non-production environment, see Deploy Digital Channels.

Related documentation:

-

Follow the steps on this page to deploy Digital Channels in a non-production lab environment using Docker Compose. With this deployment approach, you can quickly set up the Digital Channels components in one host running Docker. This deployment currently supports asynchronous chat and SMS.

> Important
>
> "Nexus" is the simplified name we use for the Digital Channels application and nodes, so you'll see that name referenced throughout this document.

Hardware requirements

- CPU: minimum 2 cores
- RAM: minimum of 16 GB

Software requirements

Use the latest stable version of:

- Docker
- Docker Compose

Your environment should have internet access to docker.io/library or you can manually load the following images to Docker:

- elasticsearch:7.3.0
- kibana:7.3.0
- filebeat:7.3.0
- postgres:11

- redis:5
- prometheus:latest
- grafana:latest

Configure Digital Channels

Download the Docker Compose file for Digital Channels and update the configuration parameters: docker-compose.yml

The following table lists the configurable parameters you can define for your installation in the **docker-compose.yml** file.

Name	Description	Default value
COMPOSE_PROJECT_NAME	The Docker Compose project name.	nexdc
NGINX_IMAGE	The address of the Nginx installation.	nginx
NGINX_TAG	The version of the Nginx installation.	latest
NGINX_PUBLIC_HTTP	Port mapping for the Nginx service.	80
NGINX_PUBLIC_HTTPS	Port mapping for the Nginx service.	443
REDIS_IMAGE	The address of the Redis installation.	redis
REDIS_TAG	The version of the Redis installation.	5
REDIS_PUBLIC_PORT	Port mapping for the Redis service.	6379
POSTGRES_IMAGE	The address of the PostgreSQL installation.	postgres
POSTGRES_TAG	The version of the PostgreSQL installation.	11
POSTGRES_USER	The PostgreSQL user Digital Channels can use to connect to the database.	nexus
POSTGRES_DB	The PostgreSQL database name.	nexus
POSTGRES_PASSWORD	The PostgreSQL user password.	nexus
POSTGRES_PUBLIC_PORT	Port mapping for the PostgreSQL service.	5432
NEXUS_IMAGE	The location of the Nexus image.	
NEXUS_TAG	The version of the Nexus image.	
NEXUS_PUBLIC_PORT	Port mapping for the Nexus service.	4004

Name	Description	Default value
ELASTICSEARCH_IMAGE	The address of the Elasticsearch installation.	docker.elastic.co/elasticsearch/elasticsearch
ELASTIC_RELEASE	The version of the Elasticsearch installation.	7.3.0
ELASTICSEARCH_PUBLIC_PORT	Port mapping for the Elasticsearch service.	9200
ELASTICSEARCH_CLUSTER_PORT	Port mapping for the Elasticsearch service.	9300
KIBANA_IMAGE	The address of the Kibana installation.	docker.elastic.co/kibana/kibana
KIBANA_PUBLIC_PORT	Port mapping for the Kibana service.	5601
FILEBEAT_IMAGE	The address of the Filebeat installation.	docker.elastic.co/beats/filebeat
PROMETHEUS_IMAGE	The address of the Prometheus installation.	prom/prometheus
PROMETHEUS_TAG	The version of the Prometheus installation.	latest
PROMETHEUS_PUBLIC_PORT	Port mapping for the Prometheus service.	9090
GRAFANA_IMAGE	The address of the Grafana installation.	grafana/grafana
GRAFANA_TAG	The version of the Grafana installation.	latest
GF_USERS_ALLOW_SIGN_UP	Allow user to sign up to Grafana.	false
GF_SECURITY_ADMIN_PASSWORD	Admin user password for Grafana.	secret
GRAFANA_PUBLIC_PORT	Port mapping for the Grafana service.	3000

If you must inject an environment variable in the Nexus service, specify them in the **nexus.env** file.

Install Digital Channels

Complete the following steps to install Digital Channels.

1. Unpack the installation package and navigate to the **target** folder.
2. Using OpenSSL, generate a self-signed certificate for SSL with a common name (CN) of either **nexus** or **localhost**.
   nexus example:

```
openssl req -x509 -newkey rsa:4096 -keyout loadbalancer/etc/nexus-dev.pem \
 -out loadbalancer/etc/nexus-dev.crt -days 3650 -nodes \
 -subj '/C=US/ST=California/L=Daly City/O=Genesys/OU=Lab/CN=nexus'
```

localhost example:

```
openssl req -x509 -newkey rsa:4096 -keyout loadbalancer/etc/nexus-dev.pem \
 -out loadbalancer/etc/nexus-dev.crt -days 3650 -nodes \
 -subj '/C=US/ST=California/L=Daly City/O=Genesys/OU=Lab/CN=localhost'
```

3. Log in to the nexus image repository.

   ```
   docker login [NEXUS_IMAGE_REPOSITORY]
   ```

4. Invoke docker-compose:

   ```
   docker-compose up -d // flag "-d" starts it as daemon
   ```

5. Wait until docker-compose is up and running. It might take up to 10 minutes.

Stop and remove containers

Use the following command to stop and remove containers:

```
docker-compose down
```

View a list of containers

Use the following command to view a list of installed containers:

```
docker-compose ps
```

Connect to databases

Connect to PostgreSQL using the following commands, making sure to replace the placeholders with your environment information:

- Postgres CLI

  ```
  docker run -it --rm --network _default postgres:11 psql -h postgres -U
  ```

- Web UI

  ```
  docker run --network _default -p 9009:8080 -e ADMINER_DEFAULT_SERVER=postgres adminer
  ```

In a web browser, go to http://localhost:9009 to confirm the connection.

Now, connect to Redis:

```
docker run -it --network _default --rm redis:5 redis-cli -h redis
```

Verify the services

Verify that Digital Channels and its services are working correctly with the following links:

Service name	Enpoints
Loadbalancer	[HTTP] http://127.0.0.1:80/nexus/v3/version [HTTPS] http://127.0.0.1/nexus/v3/version
Nexus-API	http://127.0.0.1:4004/health/detail/
Kibana	http://127.0.0.1:5601/kibana/api/status
Elasticsearch	http://127.0.0.1:9200/_cluster/health

Access the services

Access Digital Channels and its services with the links in the table below. Note that default ports are used in the example endpoints.

Service name	Endpoints	Credentials
Nexus API	[HTTP] http://127.0.0.1:80/ [HTTPS] http://127.0.0.1/	
Kibana	http://127.0.0.1/kibana http://127.0.0.1:5601/kibana	
Prometheus	http://127.0.0.1:9090/	
Grafana	http://127.0.0.1/grafana http://127.0.0.1:3000/grafana	Login: admin Password:

After logging in to Grafana, you can add your dashboards using any metrics exposed by Digital Channels.

More information

For more information about deploying and configuring Digital Channels, see Digital Channels Private Edition Guide.