Deckhouse Development Platform Documentation

Generated: July 09, 2026

Documentation

Administration guide

Overview

This document is the Administrator's Guide for the Deckhouse Development Platform and is part of the operational documentation for the Deckhouse Development Platform.

Installation

Deckhouse Development Platform can be installed in two ways: with external PostgreSQL and Redis instances (connecting to databases already deployed outside the cluster) or with internal instances (deploying PostgreSQL and Redis inside the cluster). External instances are recommended for production; internal instances are suitable for testing and pilot use. Both options are described below.

Installation with internal instances

To install Deckhouse Development Platform, enable the `development-platform` module in your Kubernetes cluster running on Deckhouse Kubernetes Platform. You can use ModuleConfig with minimal settings:

```
apiVersion: deckhouse.io/v1alpha1
kind: ModuleConfig
metadata:
  name: development-platform
spec:
  enabled: true
  version: 1
  settings:
    rbac:
      superAdminEmail: admin@deckhouse.io
# Super administrator email with full access to platform configuration. Can be changed
at any time.
    security:
      secretKey: "16charssecretkey" # Secret key for encrypting private data. If
changed, API access tokens will need to be regenerated and users will need to re-enter
their credentials.
```

After installation, the Deckhouse Development Platform web UI will be available at `https://ddp.<your domain>`.

When you do not specify `postgres` and `redis` sections, the platform deploys internal PostgreSQL and Redis instances inside the cluster. This scenario is not recommended for production and is suitable only for testing and pilot use; for production, use external resources.

Configuring internal instances (optional)

If you use internal instances, you can explicitly set `mode: internal` and specify images from a private Docker registry:

```
apiVersion: deckhouse.io/v1alpha1
kind: ModuleConfig
metadata:
  name: development-platform
spec:
  enabled: true
  version: 1
  settings:
    rbac:
      superAdminEmail: admin@deckhouse.io
    security:
      secretKey: "16charssecretkey"
    postgres:
      mode: internal
      image: registry.example.com/postgres:16.3  # PostgreSQL image from private
registry
    redis:
      mode: internal
      image: registry.example.com/redis:7.4.0    # Redis image from private registry.
    additionalImagePullSecrets:
      - "custom-registry-secret"                 # (optional) additional secrets for
private registry access.
```

Installation with external instances

This installation option is recommended for production: the platform connects to PostgreSQL and Redis deployed outside the cluster, which improves resilience and simplifies backup and scaling of databases.

Connecting external PostgreSQL

To use an external PostgreSQL instance, specify connection parameters in the `postgres` section:

```
apiVersion: deckhouse.io/v1alpha1
kind: ModuleConfig
metadata:
  name: development-platform
spec:
  enabled: true
  version: 1
  settings:
    rbac:
      superAdminEmail: admin@deckhouse.io
    security:
      secretKey: "16charssecretkey"
    postgres:
      mode: external
      host: postgres.example.com  # PostgreSQL server hostname or IP address.
      port: 5432                  # PostgreSQL port (default 5432).
      database: ddp               # Database name.
      username: ddp_user          # Connection username.
      password: secure_password   # Connection password.
```

pg_trgm extension

The platform requires the PostgreSQL `pg_trgm` extension. If you use an external PostgreSQL instance, enable it before starting DDP: connect to the database as a user with permission to create extensions and run:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm;
```

If you deploy the built-in PostgreSQL instance, the extension is created automatically.

Connecting external Redis

To use an external Redis instance, specify connection parameters in the `redis` section:

```
apiVersion: deckhouse.io/v1alpha1
kind: ModuleConfig
metadata:
  name: development-platform
spec:
  enabled: true
  version: 1
  settings:
    rbac:
      superAdminEmail: admin@deckhouse.io
    security:
      secretKey: "16charssecretkey"
    redis:
      mode: external
      host: redis.example.com       # Redis server hostname or IP address.
      port: 6379                    # Redis port (default 6379).
      database: "0"                 # Redis database index (default "0").
      password: redis_password      # Connection password (optional; leave empty if
Redis has no password).
```

Full example with external instances

Example configuration with external PostgreSQL and Redis:

```
apiVersion: deckhouse.io/v1alpha1
kind: ModuleConfig
metadata:
  name: development-platform
spec:
  enabled: true
  version: 1
  settings:
    rbac:
      superAdminEmail: admin@deckhouse.io
    security:
      secretKey: "16charssecretkey"
    postgres:
      mode: external
      host: postgres.production.example.com
      port: 5432
      database: ddp
      username: ddp_user
      password: secure_postgres_password
    redis:
      mode: external
      host: redis.production.example.com
      port: 6379
      database: "0"
      password: secure_redis_password
```

Architecture

Security

User guide

Overview

This document is the User’s Guide for the Deckhouse Development Platform and is part of the operational documentation for the Deckhouse Development Platform.

Interface

The Deckhouse Development Platform interface consists of the following sections:

- “Home”: the platform’s landing page. You can place one or more dashboards with widgets here.
- “Catalog”: a service catalog for viewing resources, entities, and relationships, and for running actions and scenarios.
- “Self-Service”: a section for configuring data sources, actions, webhooks, automations, scenarios, dashboards, and widgets. Access to this section can be restricted by the RBAC model.
- “Administration”: a section for managing teams, users, access policies, and credentials. Access to this section can be restricted by the RBAC model.

Global search

The top navigation bar includes a “Search” field. It lets you quickly find entities across the platform.

Limitations

- Maximum query length: 255 characters.
- Search works only for entities. Resources, teams, and other platform objects are not included in search results.

Product information