

Installationsvägledning KLASSA 5 onprem

Att identifiera alla de informationsmängder som finns inom en verksamhet kan vara svårt. Men genom att utgå från t.ex. de informationstillgångar eller system som en organisation använder (eller planerar att införa) kan man strukturera upp informationsmängderna och lättare hantera eventuella risker och hot.

KLASSA är ett självskattningsverktyg som fungerar som en länk mellan verksamhetens behov av skydd och tekniken som kan säkerställa detta skydd. Skydden kan vara såväl tekniska som organisatoriska och benämns ofta säkerhetsåtgärder (exempelvis i NIS2) eller skyddsåtgärder (exempelvis i GDPR).

Med hjälp av KLASSA:s kravkataloger och automatiserade flöden underlättas arbetet med att värdera de informationsmängder, system eller motsvarande tillgångar som en verksamhet har. Därmed blir det också enklare att skapa handlingsplaner med de identifierade säkerhetsåtgärder som behöver vidtas för att hålla verksamheten trygg, samt säkerställa relevant regelefterlevnad.

Genom att använda KLASSA kan verksamheten också sammanställa behovet av säkerhet avseende sina informationsmängder, inför upphandlingar av nya system eller verktyg.

Innehåll

Installationsvägledning KLASSA 5	1
Innehåll	2
Installationsvägledning KLASSA 5 onprem	5
Förutsättningar	5
Installera Docker på Ubuntu	5
Steg 1: Ta bort eventuella gamla versioner	5
Steg 2: Lägg till Dockers officiella APT-repository	5
Steg 3: Installera Docker Engine	5
Steg 4: Verifiera installationen	6
Steg 5 (valfritt): Kör Docker utan sudo	6
Installera jq	6
Databaskonfiguration	6
Alternativ A: Lokal PostgreSQL i Docker	6
Alternativ B: Extern PostgreSQL-server	7
1. Skapa databas och användare	7
2. Tillåt nätverksåtkomst	7
Starta Klassa	7
1. Ladda Docker-imagen	7
2. Förbered loggkatalog	7
3. Starta Klassa	7
4. Verifiera	8
Skapa första administratörsanvändaren	8
Steg 1: Generera BCrypt-lösenordshash	8
Steg 2: Skapa användaren i databasen	8
Steg 3: Logga in	9
Spring-profiler	10
Miljövariabler — referens	10
Databas	10

Server	11
Allmänt.....	11
Fullständigt exempel — application.yml.....	12
SAML-autentisering	14
Krävda IdP-attribut	14
Hanterad SAML — Profil: auth-managed.....	15
Generera SP-certifikat.....	15
Starta Klassa med auth-managed.....	15
Miljövariabler för auth-managed	16
Certifikatvolym	16
Intern SAML (SKR) — Profil: auth-saml.....	17
BankID — Profil: auth-bank-id + auth-saml.....	17
Federerad SAML (ISF) — Profil: auth-federation + auth-saml	18
SP-metadata — Profil: auth-saml	19
E-postnotifieringar	20
Nginx reverse proxy.....	20
Installera nginx.....	20
Konfigurationsfil	20
Aktivera och starta	21
Felsökning	21
Visa loggar.....	21
Kontrollera containerstatus.....	21
Starta om Klassa	22
Stoppa Klassa.....	22
Driftsätt Klassa i Kubernetes.....	22
Förutsättningar	22
Steg 1: Ladda imagen på varje nod.....	22
Steg 2: Konfigurera manifesten	23
Steg 3: Skapa TLS-secret.....	23
Steg 4: Driftsätt.....	24

Steg 5: Verifiera	24
Felsökning Kubernetes	24

Installationsvägledning KLASSA 5 onprem

Denna installationsvägledning exemplifierar en lokal installation av KLASSA 5 med Ubuntu, PostgreSQL, Nginx och Kubernetes. KLASSA 5 onprem kan fungera under andra förutsättningar men det är inte testat.

Förutsättningar

Komponent	Krav
Operativsystem	Ubuntu 24.04 LTS eller senare
RAM	Minst 4 GB
Disk	Minst 10 GB ledigt
Nätverk	Åtkomst till port 8080 (eller valfri port)

Installera Docker på Ubuntu

Steg 1: Ta bort eventuella gamla versioner

```
sudo apt remove docker.io docker-compose docker-compose-v2  
docker-doc podman-docker containerd runc 2>/dev/null || true
```

Steg 2: Lägg till Dockers officiella APT-repository

```
sudo apt update  
sudo apt install -y ca-certificates curl  
  
sudo install -m 0755 -d /etc/apt/keyrings  
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o  
/etc/apt/keyrings/docker.asc  
sudo chmod a+r /etc/apt/keyrings/docker.asc  
  
sudo tee /etc/apt/sources.list.d/docker.sources <<EOF  
Types: deb  
URIs: https://download.docker.com/linux/ubuntu  
Suites: $(. /etc/os-release && echo "${UBUNTU_CODENAME:-  
$VERSION_CODENAME}")  
Components: stable  
Architectures: $(dpkg --print-architecture)  
Signed-By: /etc/apt/keyrings/docker.asc  
EOF  
  
sudo apt update
```

Steg 3: Installera Docker Engine

```
sudo apt install -y docker-ce docker-ce-cli containerd.io  
docker-buildx-plugin docker-compose-plugin
```

Steg 4: Verifiera installationen

```
sudo docker --version
```

Steg 5 (valfritt): Kör Docker utan sudo

```
sudo usermod -aG docker $USER  
newgrp docker
```

Logga ut och in igen för att ändringen ska gälla permanent.

Installera jq

jq används av deployment-scriptet för att läsa `deployment-manifest.json`. Behövs inte om du startar Klassa manuellt med `docker run`.

```
sudo apt install -y jq
```

Databaskonfiguration

Klassa 5 kräver en PostgreSQL-databas. Välj ett av alternativen nedan.

Alternativ A: Lokal PostgreSQL i Docker

Enklaste alternativet för test och utveckling. Klassa-containern körs med `--network host` och når databasen via `localhost`.

```
docker run -d \  
  --name klassadb \  
  --restart unless-stopped \  
  -e POSTGRES_DB=klassa5 \  
  -e POSTGRES_USER=klassa5 \  
  -e POSTGRES_PASSWORD=<DITT_LÖSENORD> \  
  -v klassapgdata:/var/lib/postgresql/data \  
  -p 5432:5432 \  
  postgres:15
```

Verifiera:

```
docker ps | grep klassadb
```

Alternativ B: Extern PostgreSQL-server

1. Skapa databas och användare

```
CREATE DATABASE klassa5;
CREATE USER klassa5 WITH ENCRYPTED PASSWORD '<DITT_LÖSENORD>';
GRANT ALL PRIVILEGES ON DATABASE klassa5 TO klassa5;

-- PostgreSQL 15+ kräver även:
\c klassa5
GRANT ALL ON SCHEMA public TO klassa5;
```

2. Tillåt nätverksåtkomst

Filerna finns vanligtvis i `/etc/postgresql/<version>/main/` (Ubuntu/Debian) eller `/var/lib/pgsql/data/` (Red Hat/CentOS).

postgresql.conf — ändra `listen_addresses`:

```
listen_addresses = '*'
```

pg_hba.conf — lägg till en rad i slutet (ersätt `<KLASSA_SERVER_IP>`):

```
host    klassa5      klassa5      <KLASSA_SERVER_IP>/32    scram-
sha-256
sudo systemctl restart postgresql
```

Starta Klassa

1. Ladda Docker-imagen

```
docker load -i klassa-<VERSION>.tar
```

Kontrollera att imagen laddades och notera taggen:

```
docker images | grep klassa
```

2. Förbered loggkatalog

```
sudo mkdir -p /opt/klassa/logs
sudo chmod 777 /opt/klassa/logs
```

3. Starta Klassa

Grundläggande start med användarnamn/lösenord-inloggning (ersätt `<TAG>` med taggen från steget ovan):

```
docker run -d \
  --name klassa-app \
  --restart unless-stopped \
  --network host \
  -e SPRING_PROFILES_ACTIVE=auth-password,production \
```

```
-e SERVER_SERVLET_CONTEXT_PATH=/online \  
-e KLASSA_DB_URL=localhost:5432/klassa5 \  
-e KLASSA_DB_USER=klassa5 \  
-e KLASSA_DB_PWD=<DITT_LÖSENORD> \  
-v /opt/klassa/logs:/workspace/logs \  
klassa:<TAG>
```

*Alternativ — konfigurera via YAML-fil i stället för miljövariabler:
skapa en `application.yml` och montera den i containern:*

```
``bash  
-v /opt/klassa/application.yml:/config/application.yml  
``
```

*Se avsnittet [Miljövariabler — referens](#miljövariabler--referens)
för alla tillgängliga inställningar i YAML-format.*

4. Verifiera

```
docker ps | grep klassa-app  
docker logs -f klassa-app
```

Öppna sedan en webbläsare och gå till:

```
http://<SERVER_IP>:8080/online/
```

Skapa första administratörsanvändaren

Klassa skapar ingen inloggningsbar standardanvändare vid uppstart. Första gången du kör med profilen `auth-password` måste du skapa en administratörsanvändare direkt i databasen.

Förutsättning: Klassa måste ha startats minst en gång — Flyway-migrationer måste ha körts och skapat databasschemat.

Steg 1: Generera BCrypt-lösenordshash

Installera `apache2-utils` om det saknas:

```
sudo apt-get install -y apache2-utils
```

Generera hash (ersätt `DittLösenord`):

```
htpasswd -nbB -C 10 dummy 'DittLösenord' | cut -d: -f2
```

Resultatet ser ut ungefär så här: `$2y$10$abc123...`

Steg 2: Skapa användaren i databasen

Anslut till databas-containern:

```
docker exec -it klassa-db psql -U klassa5 -d klassa5
```

Kör sedan följande SQL (ersätt e-post, namn och lösenordshash):

```
DO $$
DECLARE
  v_user_id UUID := gen_random_uuid();
  v_system UUID := '9c8e7f6d-4a3b-2c1e-9f8e-7d6c5b4a3e2f';
  v_org      UUID := '3f7a8c2d-9e1b-4f6a-a8c3-7d2e9f1b4a6c';
BEGIN
  INSERT INTO klassa."user"
    (id, is_super_admin, owner_organization_id, password,
     created_by, created_date, last_modified_by,
     last_modified_date)
  VALUES
    (v_user_id, true, v_org, '$2y$10$HASH_FRÅN_STEG_1',
     v_system, NOW(), v_system, NOW());

  INSERT INTO klassa.person
    (user_id, first_name, last_name, email,
     owner_organization_id,
     created_by, created_date, last_modified_by,
     last_modified_date)
  VALUES
    (v_user_id, 'Förnamn', 'Efternamn', 'admin@example.com',
     NULL, v_system, NOW(), v_system, NOW());

  INSERT INTO klassa.user_organization_roles (user_id,
     organization_roles)
  VALUES (v_user_id, 'ADMIN');
END $$;
```

`is_super_admin = true` ger användaren åtkomst till alla organisationer och Klassa-administratörsryn. Rollen kan tas bort i GUI:n efter att ytterligare administratörer skapats.

Steg 3: Logga in

Navigera till </online/login/password> och logga in med den e-postadress och det lösenord du angav ovan.

Spring-profiler

Aktiveras via miljövariabeln `SPRING_PROFILES_ACTIVE`. Kombinera med kommatecken.

Profil	Funktion
<code>auth-password</code>	Användarnamn/lösenord-inloggning
<code>auth-managed</code>	SAML-inloggning med IdP-konfiguration via GUI. SP-certifikat anges via miljövariabler.
<code>auth-saml</code>	Intern SAML-inloggning (SKR) med statisk IdP-konfiguration i YAML/miljövariabler.
<code>auth-bank-id</code>	BankID-inloggning via SAML. Kräver <code>auth-saml</code> .
<code>auth-federation</code>	Federerad SAML (Internetstiftelsen). Kräver <code>auth-saml</code> .
<code>notification-email</code>	Utgående e-postnotifieringar
<code>production</code>	Döljer "Testmiljö"-bannern i gränssnittet

Exempel:

```
SPRING_PROFILES_ACTIVE=auth-password,auth-managed,production
```

Miljövariabler — referens

Databas

Variabel	Standard	Beskrivning
<code>KLASSA_DB_URL</code>	<code>localhost:5432/postgres</code>	PostgreSQL <code>host:port/dbnamn</code>
<code>KLASSA_DB_USER</code>	<code>postgres</code>	Databas användare
<code>KLASSA_DB_PWD</code>	<code>postgres</code>	Databas lösenord

Server

Variabel	Standard	Beskrivning
KLASSA_APP_PORT	8080	HTTP-port
SERVER_SERVLET_CONTEXT_PATH	/online	Context path — måste vara `/online` annars genereras felaktiga redirect-URL:er
TOMCAT_ACCESSLOG_DIR	logs	Katalog för Tomcat- åtkomstloggar

Allmänt

Variabel	Standard	Beskrivning
KLASSA_AUTH_DEFAULT_METHOD	discovery	Vart icke-inloggade användare skickas. Värden: discovery (väljarvy), password, internal- saml, bank-id

Fullständigt exempel — application.yml

I stället för miljövariabler kan du montera en `application.yml` i containern på `/config/application.yml`:

```
spring:
  profiles:
    active: auth-password,auth-saml,auth-bank-id,auth-
    federation,notification-email,production

  datasource:
    url: jdbc:postgresql://db-host:5432/klassa5
    username: klassa5
    password: <LÖSENORD>

  mail:
    host: smtp.example.com
    port: 587
    username: user
    password: secret

  security:
    saml2:
      relyingparty:
        registration:
          saml:
            entity-id:
https://klassa.example.com/saml2/service-provider-metadata/saml
            signing:
              credentials:
                - private-key-location:
file:/opt/klassa/certificates/saml.key
                  certificate-location:
file:/opt/klassa/certificates/saml.crt
            assertingparty:
              metadata-uri: https://idp.example.com/metadata
            bank-id:
              entity-id:
https://klassa.example.com/saml2/service-provider-
metadata/bank-id
              signing:
                credentials:
                  - private-key-location:
file:/opt/klassa/certificates/bank-id.key
                      certificate-location:
file:/opt/klassa/certificates/bank-id.crt
                assertingparty:
                  metadata-uri: https://bankid-
idp.example.com/metadata
              internetstiftelsen:
                entity-id:
https://klassa.example.com/saml2/service-provider-
```

```
metadata/internetstiftelsen
  signing:
    credentials:
      - private-key-location:
file:/opt/klassa/certificates/internetstiftelsen.key
      certificate-location:
file:/opt/klassa/certificates/internetstiftelsen.crt

server:
  port: 8080
  tomcat:
    accesslog:
      directory: /workspace/logs

klassa:
  auth:
    default-method: discovery # discovery | password |
internal-saml | bank-id

  saml:
    internal:
      visible: true
      display-name: "Sveriges Kommuner och Regioner"
      path: saml
    bankid:
      visible: true
      display-name: "BankID"
      path: bank-id
    federated: {}
  sp:
    metadata:
      organization-name: <ER_ORGANISATION>
      organization-url: <ER_WEBBPLATS>
      administrative-email: <ADMIN_EPOST>
      technical-email: <TEKNISK_EPOST>
      support-email: <SUPPORT_EPOST>
      service-name: Klassa

  internetstiftelsens-federationer:
    metadata-url:
https://md.openfed.se/prod/md/metadata_set1_idp_01.xml
    signing-certificate-location:
file:/opt/klassa/certificates/internetstiftelsens-federationer-
metadata.crt
    template-registration-id: internetstiftelsen

  notification:
    from-address: noreply@klassa.se
    base-url: https://<DIN_DOMÄN>
```

SAML-autentisering

Krävda IdP-attribut

IdP:n måste skicka följande attribut i SAML-assertionen. Klassa använder dem för att identifiera användaren, fylla i profilen och tilldela roller och organisation.

Attributnamn	Friendly name	Källa (LDAP/AD)	Obligatorisk	Beskrivning
uid	uid	uid	Ja	Unikt användar-ID — används som Klassa-användarnamn
mail	mail	mail	Ja	E-postadress
gn	gn	givenName	Ja	Förnamn
sn	sn	sn	Ja	Efternamn
OrgID	OrgID	department	Ja	Organisations-ID — kopplar användaren till en organisation i Klassa
Roles	Roles	memberOf	Nej	Gruppmedlemskap — används för rolltilldelning i Klassa

OBS: Attributnamnen är skiftlägeskänsliga. Se till att IdP:n skickar dem exakt som ovan (t.ex. `OrgID`, inte `orgid`).

Rollvärde (SAML `Roles`-attribut)	Beskrivning
ADMIN	Fullständig administratör
ORGANISATION_ADMIN	Organisationsadministratör
LAW_ADMIN	Regelverks-/lagstiftningsadministratör
RISK_ADMIN	Riskadministratör

ASSET_ADMIN	Informationstillgångsadministratör
INFORMATION_ADMIN	Informationsadministratör
USER	Vanlig användare
AUDITOR	Granskare (läsrättigheter)
NO_ACCESS	Ingen åtkomst

Dessa attribut är under omarbetning till "KLASSA_[roll]"

Hanterad SAML — Profil: auth-managed

Med `auth-managed` konfigureras IdP:n i Klassas GUI efter uppstart. Du behöver bara tillhandahålla SP-signeringscertifikatet via miljövariabler.

Generera SP-certifikat

```
sudo mkdir -p /opt/klassa/managed
sudo openssl req -x509 -newkey rsa:2048 \
  -keyout /opt/klassa/managed/saml.key \
  -out /opt/klassa/managed/saml.crt \
  -days 3650 -nodes -subj "/CN=klassa"
sudo chmod 644 /opt/klassa/managed/saml.key
```

OBS: Certifikatfiler som monteras in i containern måste ha rättighet `644` (läsbar av alla) — containerns process-user är inte root och kan inte läsa `600`-filer.

Starta Klassa med auth-managed

```
docker run -d \
  --name klassa-app \
  --restart unless-stopped \
  --network host \
  -e SPRING_PROFILES_ACTIVE=auth-password,auth-
managed,production \
  -e SERVER_SERVLET_CONTEXT_PATH=/online \
  -e KLASSA_DB_URL=localhost:5432/klassa5 \
  -e KLASSA_DB_USER=klassa5 \
  -e KLASSA_DB_PWD=<DITT_LÖSENORD> \
  -e KLASSA_BASE_URL=https://<DIN_DOMÄN> \
  -e
KLASSA_SAML_MANAGED_KEY_LOCATION=file:/opt/klassa/managed/saml.
key \
  -e
KLASSA_SAML_MANAGED_CERT_LOCATION=file:/opt/klassa/managed/saml
```

```
.crt \
-v /opt/klassa/logs:/workspace/logs \
-v
/opt/klassa/managed/saml.key:/opt/klassa/managed/saml.key:ro \
-v
/opt/klassa/managed/saml.crt:/opt/klassa/managed/saml.crt:ro \
klassa:<TAG>
```

Miljövariabler för auth-managed

Variabel	Beskrivning
KLASSA_BASE_URL	Applikationens publika bas-URL — används i SP-metadata, ACS-URL och e-postnotifieringslänkar
KLASSA_SAML_MANAGED_KEY_LOCATION	Spring-resurssökväg till SP-signeringsnyckeln, t.ex. file:/opt/klassa/managed/saml.key
KLASSA_SAML_MANAGED_CERT_LOCATION	Spring-resurssökväg till SP-signeringscertifikatet, t.ex. file:/opt/klassa/managed/saml.crt

Certifikatvolym

SAML kräver SP-certifikat för signering. Standardsökvägarna utgår från att certifikaten är monterade på /opt/klassa/certificates/ i containern:

```
/opt/klassa/certificates/
saml.key / saml.crt ← Intern
SAML SP-nyckelpar
bank-id.key / bank-id.crt ← BankID
SP-nyckelpar
internetstiftelsen.key / internetstiftelsen.crt ← ISF
federerad SP-nyckelpar
internetstiftelsens-federationer-metadata.crt ← ISF
metadata-signeringscertifikat
```

Montera katalogen som en Docker-volym:

```
-v /opt/klassa/certificates:/opt/klassa/certificates:ro
```

Intern SAML (SKR) — Profil: auth-saml

Detta är en profil som har fått ett olyckligt namn. Den har inget att göra med SKR utan är en bilateral relation med en IdP med vissa egenskaper.

Variabel	Standard	Beskrivning
KLASSA_SAML_INTERNAL_ENTITY_ID	http://localhost:8080/saml2/service-provider-metadata/saml	SP Entity ID
KLASSA_SAML_INTERNAL_METADATA_URI	<i>(obligatorisk)</i>	IdP metadata-URL
KLASSA_SAML_KEY_LOCATION	file:/opt/klassa/certificates/saml.key	SP signeringsnyckel
KLASSA_SAML_CERT_LOCATION	file:/opt/klassa/certificates/saml.crt	SP signeringscertifikat
KLASSA_SAML_INTERNAL_VISIBLE	true	Visa i väljarvy
KLASSA_SAML_INTERNAL_DISPLAY_NAME	Sveriges Kommuner och Regioner	Etikett i väljarvy
KLASSA_SAML_INTERNAL_PATH	saml	Inloggnings-URL (/login/saml)

BankID — Profil: auth-bank-id + auth-saml

Detta är en profil som har fått ett olyckligt namn. Den har inget att göra med BankID utan är en bilateral relation med en IdP med vissa egenskaper.

Variabel	Standard	Beskrivning
KLASSA_SAML_BANKID_ENTITY_ID	http://localhost:8080/saml2/service-provider-metadata/bank-id	SP Entity ID
KLASSA_SAML_BANKID_METADATA_URI	<i>(obligatorisk)</i>	IdP metadata-URL
KLASSA_SAML_BANKID_KEY_LOCATION	file:/opt/klassa/certificates/bank-id.key	SP signeringsnyckel

KLASSA_SAML_BANKID_CERT_LOCATION	file:/opt/klassa/certificates/bank-id.crt	SP signeringscertifikat
KLASSA_SAML_BANKID_VISIBLE	true	Visa i väljarvy
KLASSA_SAML_BANKID_DISPLAY_NAME	BankID	Etikett i väljarvy
KLASSA_SAML_BANKID_PATH	bank-id	Inloggnings- URL (/login/bank-id)

Federerad SAML (ISF) — Profil: auth-federation + auth-saml

Detta är ren profil som läser ett aggregerat SAML-metadata. I detta fall är det Internetstiftelsens Svenska Federationer.

Variabel	Standard	Beskrivning
KLASSA_SAML_FEDERATED_ENTITY_ID	http://localhost:8080/saml2/service-provider-metadata/internetstiftelsen	SP Entity ID
KLASSA_SAML_FEDERATED_KEY_LOCATION	file:/opt/klassa/certificates/internetstiftelsen.key	SP signeringsnyckel
KLASSA_SAML_FEDERATED_CERT_LOCATION	file:/opt/klassa/certificates/internetstiftelsen.crt	SP signeringscertifikat
KLASSA_SAML_FEDERATED_METADATA_URLS	(tom)	Kommaseparerade IdP metadata-URL:er
KLASSA_SAML_FEDERATED_CERT_URLS	(tom)	Kommaseparerade metadata-signeringscertifikat-URL:er
KLASSA_SAML_FEDERATED_SIGNING_CERT_LOCATION	file:/opt/klassa/certificates/internetstiftelsen-federationer-metadata.crt	Metadata-signeringscertifikat (fil)

KLASSA_SAML_FEDERATED_TEMPLATE_REGISTRATION_ID	internetstiftelsen	Mall-registrerings-ID
--	--------------------	-----------------------

SP-metadata — Profil: auth-saml

Detta är det egna SAML-metadatat. Används exempelvis vid federationsregistrering — organisationsinformation som visas i IdP:ns väljarvy.

Variabel	Standard	Beskrivning
KLASSA_SAML_SP_ORGANIZATION_NAME	Adda AB	Organisationsnamn i metadata
KLASSA_SAML_SP_ORGANIZATION_URL	https://adda.se/	Organisations-URL
KLASSA_SAML_SP_ADMINISTRATIVE_EMAIL	adda@regent.se	Administrativ kontakt
KLASSA_SAML_SP_TECHNICAL_EMAIL	adda@regent.se	Teknisk kontakt
KLASSA_SAML_SP_SUPPORT_EMAIL	adda@regent.se	Supportkontakt
KLASSA_SAML_SP_SERVICE_NAME	Klassa	Tjänstenamn
KLASSA_SAML_SP_SERVICE_DESCRIPTION_SV	Ett verktyg som hjälper organisationer att systematiskt arbeta med informations säkerhet	Tjänstebeskrivning (SV)
KLASSA_SAML_SP_SERVICE_DESCRIPTION_EN	A tool that helps organizations systematically work with information security	Tjänstebeskrivning (EN)
KLASSA_SAML_SP_MDUI_DISPLAY_NAME	Klassa	MDUI visningsnamn
KLASSA_SAML_SP_MDUI_LOGO_URL	/online/img/logo_s.v.svg	MDUI logotyp-URL
KLASSA_SAML_SP_MDUI_LOGO_WIDTH	320	Logotypbredd (px)
KLASSA_SAML_SP_MDUI_LOGO_HEIGHT	132	Logotyphöjd (px)

E-postnotifieringar

Profil: notification-email

Variabel	Standard	Beskrivning
KLASSA_MAIL_HOST	localhost	SMTP-server
KLASSA_MAIL_PORT	1025	SMTP-port
KLASSA_MAIL_USERNAME	(tom)	SMTP-användarnamn
KLASSA_MAIL_PASSWORD	(tom)	SMTP-lösenord
KLASSA_MAIL_FROM	noreply@klassa.se	Avsändaradress
KLASSA_BASE_URL	http://localhost:8080	Applikationens publika bas-URL — används i notifieringslänkar och av auth-managed för SP-metadata/ACS-URL

Nginx reverse proxy

Om du kör Klassa bakom nginx för TLS-terminering i stället för att exponera port 8080 direkt.

Installera nginx

```
sudo apt install -y nginx
```

Konfigurationsfil

Skapa /etc/nginx/sites-available/klassa:

```
# HTTP → HTTPS-redirect
server {
    listen 80;
    server_name <DIN_DOMÄN>;
    return 301 https://$host$request_uri;
}

# HTTPS
server {
    listen 443 ssl;
    server_name <DIN_DOMÄN>;

    ssl_certificate /etc/nginx/certs/fullchain.pem;
```

```
ssl_certificate_key /etc/nginx/certs/privkey.pem;
ssl_protocols      TLSv1.2 TLSv1.3;
ssl_ciphers        HIGH:!aNULL:!MD5;

location /online/ {
    proxy_pass      http://localhost:8080/online/;
    proxy_http_version 1.1;
    proxy_set_header Host          $host;
    proxy_set_header X-Real-IP     $remote_addr;
    proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    # Krävs för att SAML-redirectar ska generera rätt
URL:er
    proxy_redirect  http://localhost:8080/
https://$host/;
}

# Omdirigera rooten till /online/
location = / {
    return 302 /online/;
}
}
```

OBS: `X-Forwarded-Proto: https` är viktigt för SAML — utan det genererar Klassa HTTP-baserade ACS-URL:er som IdP:n avvisar.

Aktivera och starta

```
sudo mkdir -p /etc/nginx/certs
sudo cp fullchain.pem /etc/nginx/certs/
sudo cp privkey.pem /etc/nginx/certs/
sudo chmod 600 /etc/nginx/certs/privkey.pem

sudo ln -sf /etc/nginx/sites-available/klassa /etc/nginx/sites-enabled/klassa
sudo rm -f /etc/nginx/sites-enabled/default
sudo nginx -t && sudo systemctl reload nginx
```

OBS: `default`-siten inaktiveras så att nginx inte svarar med en annan konfiguration på port 80/443.

Felsökning

Visa loggar

```
docker logs -f klassa-app
```

Kontrollera containerstatus

```
docker ps -a | grep klassa
```

Starta om Klassa

```
docker restart klassa-app
```

Stoppa Klassa

```
docker stop klassa-app
```

Driftsätt Klassa i Kubernetes

Denna sektion beskriver hur du driftsätter Klassa 5 i ett befintligt Kubernetes-kluster med hjälp av manifesten i mappen `kubernetes/`.

Förutsättningar

Komponent	Krav
Kubernetes	v1.25 eller senare
Ingress-controller	nginx ingress controller
Storage	StorageClass med dynamisk provisionering (t.ex. <code>local-path</code>)
kubectl	Konfigurerad med åtkomst till klustret (<code>~/.kube/config</code>)

Viktigt: Kontrollera att följande systempoddar körs innan du börjar:

```
``bash
kubectl get pods -n kube-system | grep -E 'coredns|local-path'
``
```

Om CoreDNS saknas: `sudo kubectl apply -f https://kubernetes.io/docs/concepts/services-networking/dns-pod-service/#coredns`

Om `local-path-provisioner` saknas: se [Felsökning Kubernetes](#felsökning-kubernetes) nedan.

Steg 1: Ladda imagen på varje nod

Kubernetes använder containerd direkt och hämtar inte imagen från Docker — den måste importeras på **varje nod** (master + workers).

Kopiera tar-filen till respektive nod:

```
scp klassa-<VERSION>.tar ubuntu@<MASTER_IP>:/tmp/
scp klassa-<VERSION>.tar ubuntu@<WORKER1_IP>:/tmp/
scp klassa-<VERSION>.tar ubuntu@<WORKER2_IP>:/tmp/
```

Importerar på **varje nod** (kör via SSH):

```
sudo ctr -n k8s.io images import /tmp/klassa-<VERSION>.tar
```

Verifiera att imagen finns och notera den exakta taggen:

```
sudo ctr -n k8s.io images ls | grep klassa
```

Uppdatera sedan image-fältet i `kubernetes/deployment.yaml` med den exakta taggen (t.ex. `docker.io/library/klassa:7678`).

Steg 2: Konfigurera manifesten

kubernetes/secret-db.yaml

```
stringData:  
  db-password: "<DITT_DATABASLÖSENORD>"
```

kubernetes/postgres.yaml

```
stringData:  
  POSTGRES_PASSWORD: "<DITT_DATABASLÖSENORD>"
```

Lösenordet måste vara identiskt i båda filerna.

kubernetes/deployment.yaml

Sätt rätt image och justera vid extern PostgreSQL-server:

```
image: <REGISTRY>/klassa:<TAG>  
env:  
  - name: KLASSA_DB_URL  
    value: "<POSTGRES_HOST>:5432/klassa5"  
  - name: KLASSA_DB_USER  
    value: "klassa5"  
  - name: KLASSA_NOTIFICATION_BASE_URL  
    value: "https://<DIN_DOMÄN>/online"
```

kubernetes/ingress.yaml

```
tls:  
  - hosts:  
    - <DIN_DOMÄN>  
    secretName: klassa-tls  
rules:  
  - host: <DIN_DOMÄN>
```

Steg 3: Skapa TLS-secret

```
kubectl create secret tls klassa-tls \  
  --cert=<sökväg-till-fullchain.pem> \  
  --key=<sökväg-till-private.key> \  
  -n klassa
```

Steg 4: Driftsätt

```
kubect1 apply -f kubernetes/namespace.yaml
kubect1 apply -f kubernetes/secret-db.yaml
kubect1 apply -f kubernetes/postgres.yaml
kubect1 apply -f kubernetes/configmap.yaml
kubect1 apply -f kubernetes/pvc-logs.yaml
kubect1 apply -f kubernetes/deployment.yaml
kubect1 apply -f kubernetes/service.yaml
kubect1 apply -f kubernetes/ingress.yaml
```

Steg 5: Verifiera

```
kubect1 get pods -n klassa -o wide
```

Förväntat resultat (efter ~2 minuter):

NAME	READY	STATUS	RESTARTS	AGE
klassa-xxxxxxxx-xxxxx	1/1	Running	0	2m
postgres-0	1/1	Running	0	2m

```
kubect1 logs -f deployment/klassa -n klassa
kubect1 get ingress -n klassa
```

Klassa är tillgängligt på https://<DIN_DOMÄN>/online/.

Felsökning Kubernetes

Poddar fastnar i Pending

```
kubect1 get pods -n local-path-storage
```

Om provisionern saknas:

```
kubect1 apply -f
https://raw.githubusercontent.com/rancher/local-path-
provisioner/v0.0.28/deploy/local-path-storage.yaml
```

DNS-fel (UnknownHostException: postgres)

```
kubect1 get pods -n kube-system | grep coredns
```

Om CoreDNS saknas:

```
sudo kubeadm init phase addon coredns
kubect1 rollout restart deployment/klassa -n klassa
```

404 Not Found från nginx

Kontrollera att du använder rätt URL — Klassa svarar på `/online/`:

```
https://<DIN_DOMÄN>/online/
```

Felsöka en kraschad pod

```
kubectl logs -n klassa deployment/klassa --previous  
kubectl describe pod -n klassa <POD_NAMN>
```

Starta om Klassa

```
kubectl rollout restart deployment/klassa -n klassa
```