OPENUEM

An Open-Source Unified Endpoint Manager
that is self-hosted and lets you manage your IT
assets thanks to its agents

# Getting Started with OpenUEM

## Docker Deployment Guide for Evaluation

15 minutes

Beginner

Evaluation Setup

# Deployment Overview

## What This Guide Accomplishes

- Deploy a fully functional OpenUEM server on your local machine
- Complete deployment in approximately 15 minutes
- Suitable for evaluation, testing, and development purposes
- Configure certificate-based authentication for secure access

## Evaluation vs. Production

| Aspect | This Guide |
|---|---|
| Server Name | localhost |
| DNS Required | No |
| Credentials | Defaults (safe for testing) |
| Remote Agents | Not supported |
| Use Case | Evaluation only |

⚠ **For production deployments with remote agents, DNS configuration, and enhanced security, refer to the complete Docker installation guide.**

# Prerequisites

## Required Software

| Component | Version |
|---|---|
| Docker | 20.10+ |
| Compose | 2.0+ |
| Git | 2.0+ |

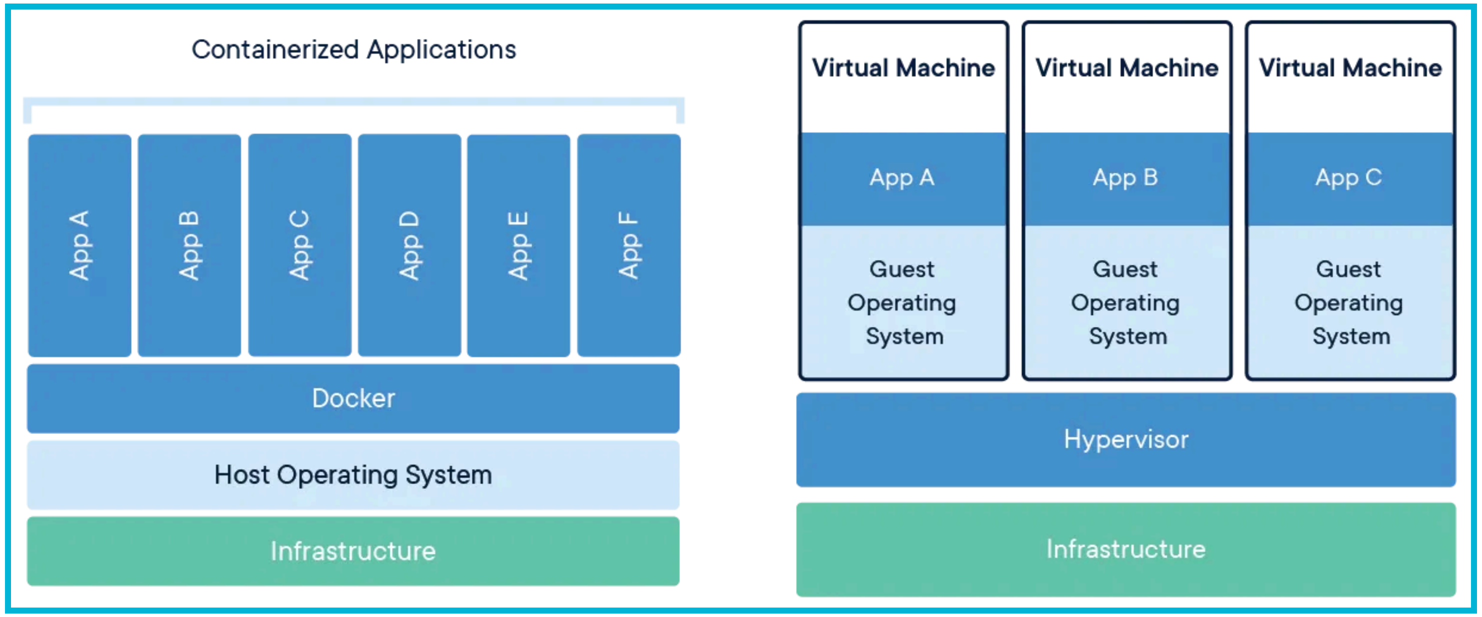
### System Requirements

RAM: 4GB min, 8GB rec.

Disk: 2GB for images

Network: Internet required

## Why Docker?



Docker containers share the host OS kernel, making them lightweight and fast to start compared to traditional virtual machines. This enables efficient deployment of OpenUEM components.

### Verify Prerequisites

Check Docker:

```
docker --version
```

Check Compose:

```
docker compose version
```

Check Git:

```
git --version
```

# Architecture Components

## Deployed Services

**PostgreSQL Database**
Stores device inventory, user data, and configuration

**NATS Server**
Message broker for agent communication

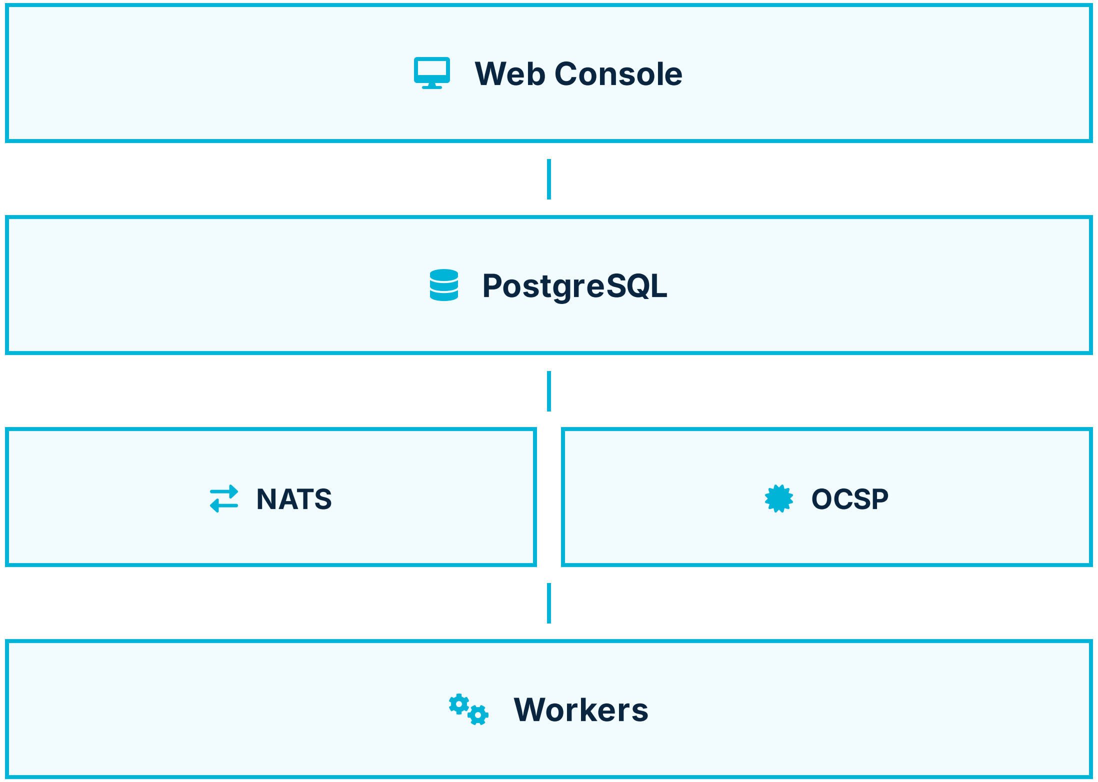
**Web Console**
Web-based administrative interface

**OCSP Responder**
Certificate validation service

**Certificate Manager**
Automated certificate lifecycle management

**Worker Services**
Agents Worker, Notification Worker

## Component Interaction



Web Console

PostgreSQL

NATS

OCSP

Workers

All components communicate over internal Docker network

Components are configured automatically through environment variables and Docker Compose profiles

# 1 Clone the Repository

Clone the OpenUEM Docker configuration repository to your local system. This repository contains the Docker Compose configuration, environment templates, and initialization scripts.

### Commands

```
git clone https://github.com/open-uem/openuem-docker
cd openuem-docker
```

## What's Included

**docker-compose.yml**
Container orchestration configuration

**.env-example**
Environment variable template

**README.md**
Repository documentation

## ✓ Verification

**Confirm successful clone:**

```
ls -la
```

You should see the key files listed above in the directory.

# 2 Environment Configuration

Create the environment configuration file and set the server name for local evaluation.

## Create .env File

Copy the template:

```
cp .env-example .env
```

## Key Configuration

Set in .env file:

```
SERVER_NAME=localhost
```

## Why localhost?

**Using localhost restricts access to your local machine only, making it ideal for quick evaluation without DNS configuration.**

## Other Variables

All other environment variables will use their **default values**, which are safe and appropriate for testing and evaluation purposes.

**For production deployments with custom domains and security hardening, refer to the complete Docker installation guide for all 18+ configurable variables.**

# 3 Initialize Database & Certificates

Execute the initialization profile to create the PostgreSQL database and generate the required SSL/TLS certificates.

### Initialization Command

```
docker compose --profile init up -d --build
```

## What Happens During Initialization

- Downloads PostgreSQL container image (~500 MB)
- Initializes OpenUEM database schema
- Creates server certificates for HTTPS
- Creates Docker volume for database persistence
- Generates Certificate Authority (CA) certificate
- Generates administrator user certificate

# 3 Certificates Created

## Directory Structure

```
certificates/
├── ca/
│   └── ca.cer
├── console/
│   ├── console.cer
│   └── console.key
├── users/
│   └── admin.pfx
└── agents/
    ├── agent.cer
    └── agent.key
```

## Certificate Files

**ca.cer**

Certificate Authority root certificate for browser trust

**console.cer + console.key**

Server certificates for HTTPS web console access

**admin.pfx**

Administrator user certificate for authentication

**agent.cer + agent.key**

Agent enrollment certificates for device management

**Verification**

Wait until you see the **certificates/** directory created before proceeding to the next step.

**Estimated Time: 3-5 minutes**

Initial execution requires downloading container images (~500 MB). Subsequent executions will be significantly faster as images are cached locally.

# 4 Start OpenUEM Services

Launch all OpenUEM application components using the **openuem** profile. This starts the core services needed for the management platform.

### Command

```
docker compose --profile openuem up -d --build
```

## Services Started

# 4 Verify Services

Confirm that all OpenUEM services started successfully by checking the container status.

### Verification Command

```
docker compose ps
```

### Expected Output - All Services Running

```
■ openuem-console-1                        Up
■ openuem-nats-server                      Up
■ openuem-ocsp-responder-1                 Up
■ openuem-agents-worker-1                  Up
■ openuem-cert-manager-worker-1            Up
■ openuem-notification-worker-1            Up
■ openuem-db-1                             Up (healthy)
```

**Troubleshooting**

If any container shows a status other than "Up", check the logs with: `docker compose logs [container-name]`

# Certificate-Based Authentication

OpenUEM uses **certificate-based authentication** instead of traditional username/password credentials to provide enhanced security through cryptographic identity verification.

## Why Certificates Instead of Passwords?

### Enhanced Security

Digital certificates are cryptographically signed and significantly harder to compromise than passwords

### Strong Authentication

Cryptographic proof of identity prevents credential stuffing and brute-force attacks

### Passwordless Access

Eliminates risks of weak passwords, password reuse, and phishing attacks

### Audit Trail

Non-repudiable proof of identity for compliance and auditing requirements

✓ **Certificate-based authentication provides enterprise-grade security without the vulnerabilities of traditional password systems**

# Certificates Required

## CA Certificate

### CA Certificate

`ca.cer`

Installed in **Trusted Root** store. Allows your browser to trust OpenUEM's self-signed certificates.

**Purpose:** Establishes trust chain for all OpenUEM certificates

## User Certificate

### User Certificate

`admin.pfx`

Installed in **Personal** store. Provides your administrative identity for authentication.

**Purpose:** Cryptographic proof of administrator identity

**Default Certificate Password**

When importing the user certificate (admin.pfx), use password: `changeit`

✓ **Certificate-based authentication provides enterprise-grade security without the vulnerabilities of traditional password systems**

# Certificate Import Process

Before accessing the console, import **two certificates** into your browser: the CA certificate and the user certificate.

## 1. CA Certificate

**File:** `certificates/ca/ca.cer`

**Purpose:** Allows your browser to trust OpenUEM's self-signed certificates

**Import to:** Trusted Root Certification Authorities store

## 2. User Certificate

**File:** `certificates/users/admin.pfx`

**Purpose:** Provides your administrative identity for authentication

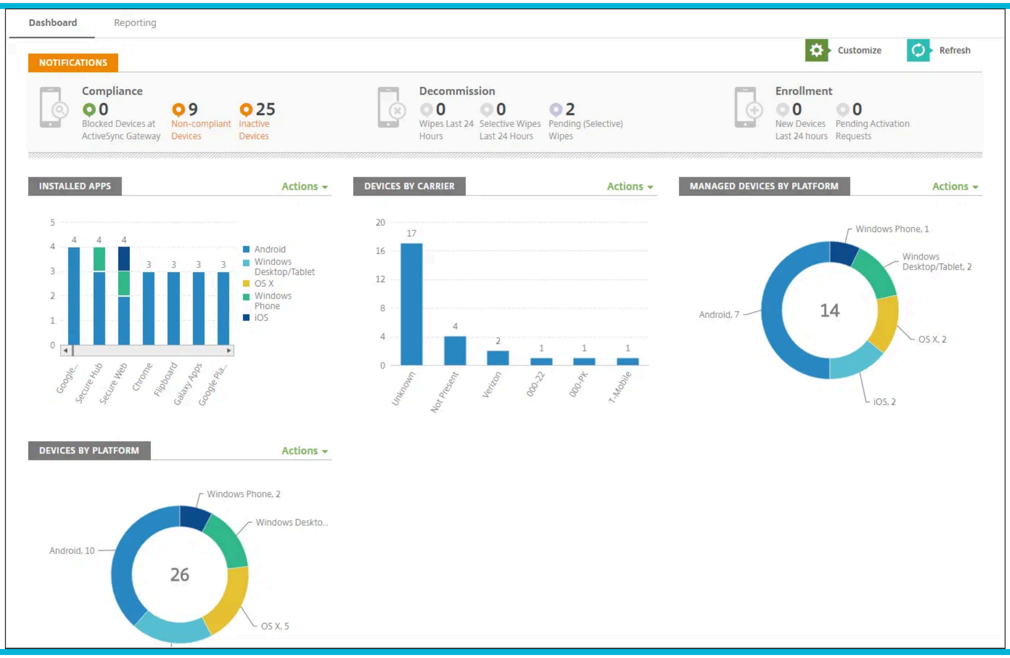
**Import to:** Personal certificate store

### Browser Support

Chrome   Firefox   Safari   Edge

Each browser has specific import procedures. Detailed step-by-step instructions are provided in the full guide.

# Access to Console



After successful certificate import, you'll have access to the OpenUEM administrative console with full device management capabilities.

### Default Certificate Password

When importing the user certificate (admin.pfx), use password: `changeit`

✓ **Certificate import complete - you're ready to manage your endpoints with OpenUEM**

# Access the Console

## Console URL

Open your browser and navigate to:

```
https://localhost:1323
```

! You must use https:// (not http://)

## First Login

### 1. Certificate Selection Prompt

Your browser will prompt you to select a client certificate for authentication

### 2. Select Admin Certificate

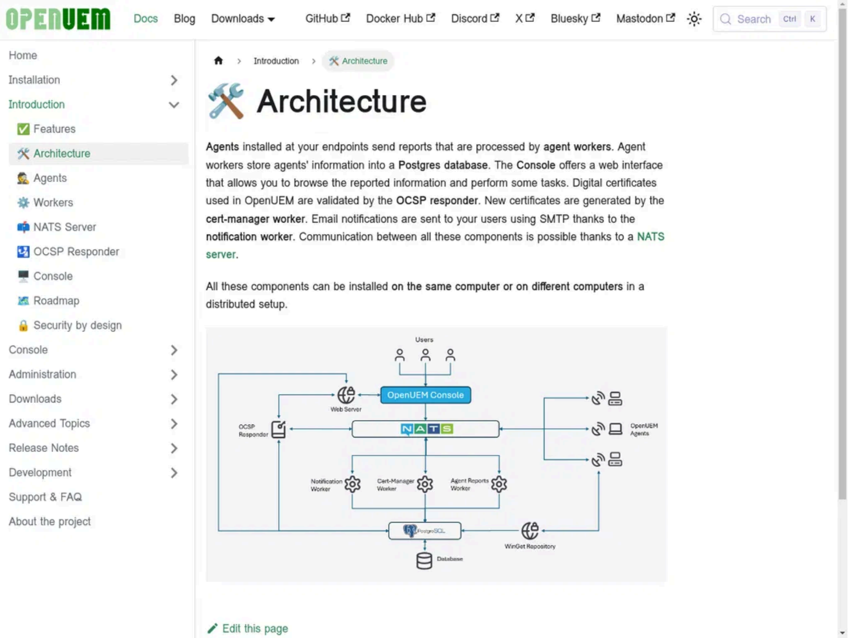
Choose the **admin** certificate you imported earlier

### 3. Automatic Authentication

You will be automatically logged into the OpenUEM console

# OpenUEM Dashboard Overview

## OpenUEM Architecture



### Architecture

**Agents** installed at your endpoints send reports that are processed by **agent workers**. Agent workers store agents' information into a **Postgres database**. The **Console** offers a web interface that allows you to browse the reported information and perform some tasks. Digital certificates used in OpenUEM are validated by the **OCSP responder**. New certificates are generated by the **cert-manager worker**. Email notifications are sent to your users using SMTP thanks to the **notification worker**. Communication between all these components is possible thanks to a NATS server.

All these components can be installed **on the same computer or on different computers** in a distributed setup.

Complete system architecture showing all deployed components and their interactions

## Dashboard Sections

- Devices
- Users
- Policies
- Reports
- Settings

# Next Steps

Now that OpenUEM is running, here are the recommended actions to get the most out of your deployment:

1

## Install Your First Agent

Deploy the OpenUEM agent on a test endpoint to begin collecting inventory data and testing remote management capabilities. Refer to the **Agent Installation Guide** for platform-specific instructions.

2

## Explore OpenUEM Features

Familiarize yourself with OpenUEM's capabilities by reviewing the **Introduction** and **Features** documentation. Discover device management, policy enforcement, and reporting tools.

# Next Steps (continued)

3

## Plan Production Deployment

When ready for production, review the **complete Docker installation guide** for DNS configuration, custom database credentials, reverse proxy setup, and security hardening best practices.

4

## Join the Community

Connect with other OpenUEM users and developers on **Discord** and **GitHub**. Get support, share experiences, and contribute to the project's development.

✓ **You're all set! Start managing your endpoints with OpenUEM**

# Key Takeaways

### 15-Minute Deployment

Quick evaluation deployment with Docker using localhost configuration. Perfect for testing and familiarizing yourself with OpenUEM capabilities.

### Certificate-Based Security

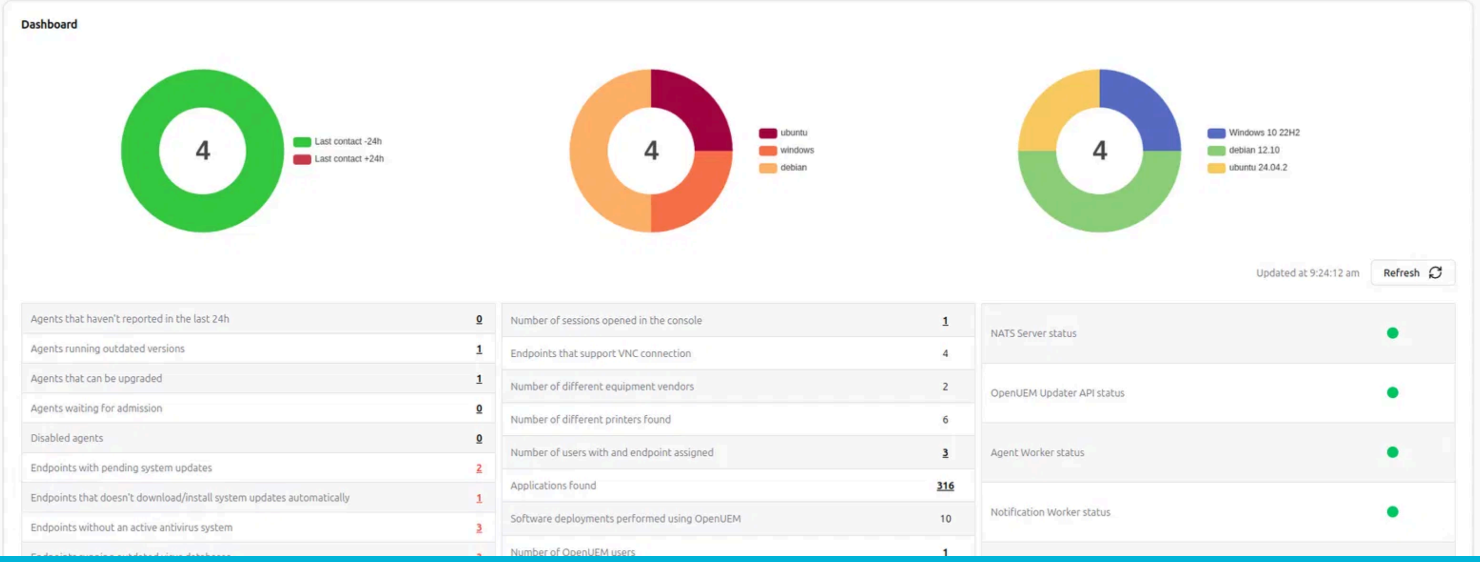
Enterprise-grade authentication without password vulnerabilities. Cryptographic proof of identity for enhanced security and compliance.

### Production Deployment

For production environments, follow the complete Docker installation guide with DNS configuration, custom credentials, and security hardening.

# Resources & Community

## OpenUEM Console



Dashboard

Updated at 9:24:12 am

| | |
|---|---|
| Agents that haven't reported in the last 24h | 0 |
| Agents running outdated versions | 1 |
| Agents that can be upgraded | 1 |
| Agents waiting for admission | 0 |
| Disabled agents | 0 |
| Endpoints with pending system updates | 2 |
| Endpoints that doesn't download/install system updates automatically | 1 |
| Endpoints without an active antivirus system | 3 |

| | |
|---|---|
| Number of sessions opened in the console | 1 |
| Endpoints that support VNC connection | 4 |
| Number of different equipment vendors | 2 |
| Number of different printers found | 6 |
| Number of users with and endpoint assigned | 3 |
| Applications found | 316 |
| Software deployments performed using OpenUEM | 10 |
| Number of OpenUEM users | 1 |

| | |
|---|---|
| NATS Server status | ● |
| OpenUEM Updater API status | ● |
| Agent Worker status | ● |
| Notification Worker status | ● |

Full-featured administrative console for unified endpoint management

## Resources & Community

### Documentation

openuem.eu/docs

### GitHub

github.com/openuem

### Community

Join us on Discord