

# Інструкція до інсталяції та конфігурування системи AtOM. Access to Memory



Львівський національний університет імені Івана Франка

Львів 2024

# Зміст

<b>1</b>	<b>Загальна інформація</b>	<b>1</b>
1.1	Технічний огляд . . . . .	1
<b>2</b>	<b>Технічні вимоги</b>	<b>2</b>
2.1	Мінімальні вимоги . . . . .	2
2.2	Рекомендовані вимоги . . . . .	2
2.3	Обов’язкові залежності програмного забезпечення . . . . .	2
2.4	Інші залежності програмного забезпечення . . . . .	3
<b>3</b>	<b>Встановлення системи AtoM</b>	<b>4</b>
3.1	Linux . . . . .	4
3.1.1	Встановлення залежностей . . . . .	4
3.1.2	Завантаження AtoM . . . . .	5
3.1.3	Створення бази даних . . . . .	6
3.1.4	Інсталювання . . . . .	6
3.2	Windows . . . . .	8
<b>4</b>	<b>Конфігурація</b>	<b>8</b>
4.1	Маніпуляція налаштуваннями з веб-сторінки . . . . .	8
4.2	Конфігураційні файли . . . . .	8
<b>5</b>	<b>Постскрипtum</b>	<b>8</b>

# 1 Загальна інформація

**AtoM** визначається як "Access to Memory". Це є веб-програма з відкритим вихідним для архівного опису та доступу на основі стандартів у багатомовному середовищі з **мультирепозитарною системою**, тобто такою, яка використовується мережею архівних установ або інших типів сховищ.

## 1.1 Технічний огляд

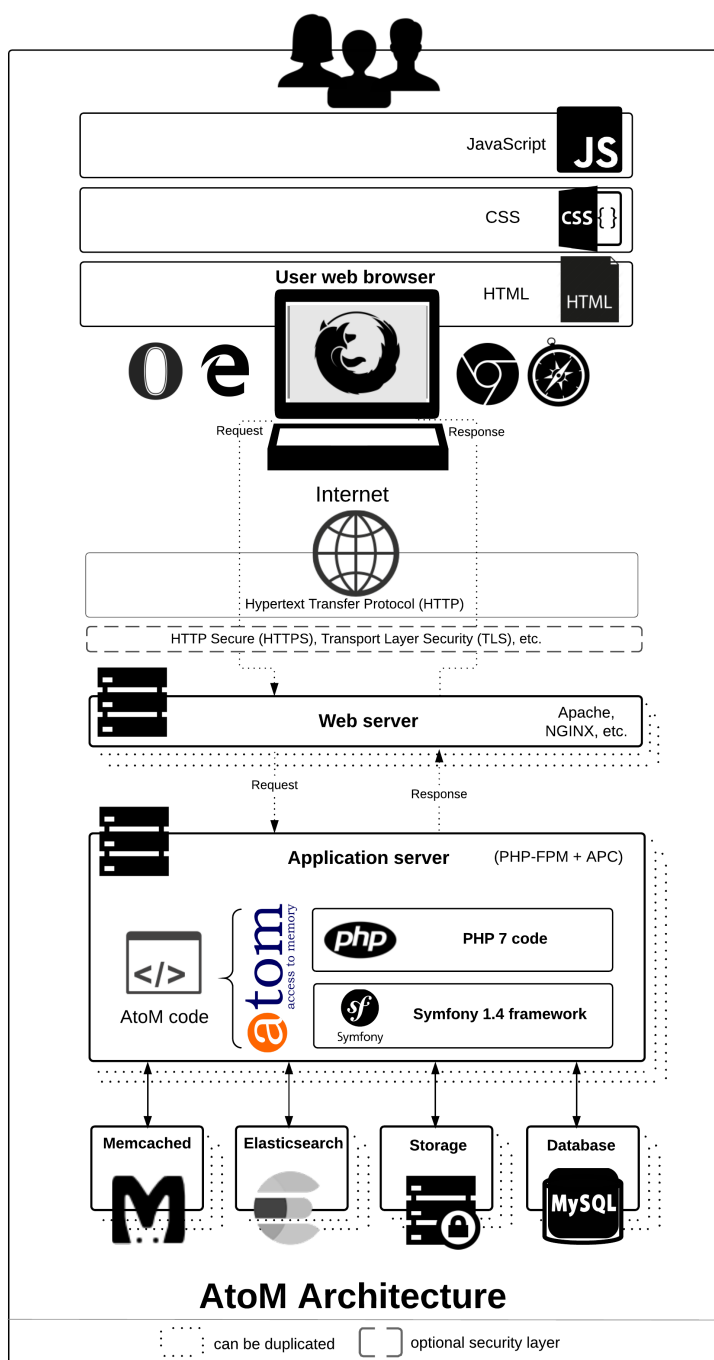


Рис 1. Архітектура AtoM.

**AtoM складається з:**

- **HTML-сторінок**, які надаються браузеру з веб-сервера. Основні два сервера, які використовуються та були протестовані розробниками є **Nginx** та **Apache**.
- **Бази даних**. **MySQL (8.0)** використовується в розробці, але AtoM використовує рівень абстракції бази даних і тому потенційно сумісний з **Postgres**, **SQLite**, **SQLServer**, **Oracle**.
- **PHP-клієнт**, який керує запитами та відповідями між веб-клієнтами, логікою програми та вмістом програми, що зберігається в базі даних.
- **Symphony-Framework**, який організовує складові частини за допомогою об'єктної орієнтації та передових практик веб-дизайну.
- **Elasticsearch**. Розподілений пошуковий сервер на базі Apache Lucene, який діє як пошукова та аналітична система програми. Elasticsearch не інтегровано безпосередньо в код AtoM як бібліотеку, а як службу, розгорнуту в тій самій мережі, з якою AtoM взаємодіє через повний API REST.

## 2 Технічні вимоги

Наступна інформація призначена для надання вихідної точки для налаштування вашої системи. Вона надає специфікації для розгортання «все-в-одному», зі всіма службами (тобто nginx, Persona server, ES, memcached), встановленими в одній машині.

### 2.1 Мінімальні вимоги

- Процесор: 2 vCPU @ 2.3 ГГц
- Оперативна пам'ять: 7 ГБ
- Місце на диску (приблизно): мінімум 50 ГБ для базової частини AtoM, додатковий простір необхідний для підтримки значної кількості цифрових об'єктів.

### 2.2 Рекомендовані вимоги

- Процесор: 4 vCPU @ 2.3 ГГц
- Оперативна пам'ять: 16 ГБ
- Місце на диску (приблизно): мінімум 50 ГБ для базової частини AtoM, додатковий простір необхідний для підтримки значної кількості цифрових об'єктів.

### 2.3 Обов'язкові залежності програмного забезпечення

- Веб-сервер, як-от Apache або Nginx; Artefactual.
- Elasticsearch версії 5.x. Elasticsearch 6.0 або новіша версія не підтримується, оскільки вони припинили підтримку деяких API, які все ще використовуються в AtoM.
- Java 8 (потрібна для Elasticsearch)
- MySQL 8.0
- PHP 7.4

- Сервер задач Gearman

Наступні PHP розширення є обов'язковими:

- cURL
- JSON
- APC (також потрібен apcu-bc)
- PDO і PDO-MySQL
- XSL

## 2.4 Інші залежності програмного забезпечення

**ImageMagick** — це програмний пакет для створення, редагування, компонування або перетворення растрових зображень. Він може читати та записувати зображення в різних форматах (понад 100), включаючи DPX, EXR, GIF, JPEG, JPEG-2000, PDF, PhotoCD, PNG, Postscript, SVG і TIFF.

**ImageMagick** використовується в AtoM для створення деривативів зображень (референсного та мініатюри) з головного цифрового об'єкта, включаючи створення деривативів з багатосторінкових TIFF. ImageMagick і Ghostscript потрібні для створення одно- та багатосторінкових PDF деривативів зображень.

**Ghostscript** — це пакет програмного забезпечення на основі інтерпретатора для мов опису сторінок PostScript компанії Adobe Systems та Portable Document Format (PDF). Основними його цілями є растризація або відтворення таких мов опису сторінок для відображення або друку документів та перетворення між файлами PostScript і PDF.

**Ghostscript** використовується в AtoM разом із ImageMagick для створення одно- та багатосторінкових PDF деривативів зображень.

**FFmpeg** — це повне, кросплатформне рішення для запису, конвертування та трансляції аудіо та відео. Він включає libavcodec — провідну бібліотеку аудіо/відео кодеків.

**FFmpeg** використовується в AtoM для створення відео деривативів, включаючи створення Flash відео деривативу для перегляду у браузері.

**pdftotext** — це утиліта командного рядка з відкритим кодом для перетворення PDF файлів у звичайні текстові файли, тобто витяг даних з файлів у форматі PDF. Вона безкоштовно доступна та включена за замовчуванням у багатьох дистрибутивах Linux, а також доступна для Windows як частина Xpdf.

**Apache FOP** (Formatting Objects Processor) — це програма форматування для друку, керована об'єктами форматування XSL (XSL-FO) та незалежний від формату виведення формувач. Це програма на Java, яка зчитує дерево об'єктів форматування (FO) і відображає результат на заданий вихід.

**Apache FOP** використовується в AtoM для створення PDF-довідників.

## 3 Встановлення системи AtoM

### 3.1 Linux

#### 3.1.1 Встановлення залежностей

AtoM 2.8 вимагає MySQL версії 8.0 або вище, оскільки він використовує вирази загальних таблиць.

```
1 sudo apt update
2 sudo apt install mysql-server
```

#### Порада

Стандартна установка MySQL не є повністю захищеною, але вона містить скрипт безпеки, який можна виконати для покращення налаштувань за замовчуванням:

```
1 sudo mysql_secure_installation
```

При виконанні безпечної інсталяції можна налаштувати обов'язкову складність паролів.

Нарешті, налаштуємо режими роботи MySQL. Сервер MySQL може працювати в різних SQL-режимах, що впливає на синтаксис SQL, який підтримує MySQL, і на перевірку даних, яку він виконує.

Вставте наступні значення в новий файл за адресою `/etc/mysql/conf.d/mysqld.cnf` і збережіть:

```
1 [mysqld]
2 sql_mode=ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION
3 optimizer_switch='block_nested_loop=off'
```

Можна використати редактор nano

```
1 sudo nano /etc/mysql/conf.d/mysqld.cnf
```

Для збереження у nano натисніть `Ctrl+X` і оберіть опцію Yes, після чого натисніть Enter.

Тепер перезапустимо MySQL:

```
1 sudo systemctl restart mysql
```

Встановлюємо сервер для пошуку - Elasticsearch

Сервер пошуку на основі Apache Lucene, розроблений на Java, який забезпечує AtoM численними додатковими функціями, продуктивністю та масштабованістю. Це, ймовірно, найбільша зміна, введена в AtoM 2.x, і ми задоволені результатами.

Ubuntu не надає пакета, але ви можете завантажити його безпосередньо з **сайту Elasticsearch**, якщо у вас виникають проблеми зі встановленням описаним далі способом.

Переконайтеся, що встановлено Java. У цьому прикладі ми будемо використовувати OpenJDK, але також підійде Oracle JVM.

```
1 sudo apt install openjdk-11-jre-headless apt-transport-https software-properties-common
```

Після успішної установки Java продовжуйте встановлення Elasticsearch. Завантажте та встановіть відкритий ключ підпису, який використовується в їхньому репозиторії:

```
1 wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
```

## Важливо

Не пропустіть тире (-) в кінці цієї команди!

Тепер додайте їхній репозиторій:

```
1 echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-5.x.list
```

Готові до встановлення. Виконайте:

```
1 sudo apt update
2 sudo apt install elasticsearch
```

Запустіть сервіс та налаштуйте його на автозапуск при завантаженні системи:

```
1 sudo systemctl enable elasticsearch
2 sudo systemctl start elasticsearch
```

Встановлюємо PHP та German-job-server

Ubuntu 20.04 постачається з PHP 7.4, який значно швидший за попередні версії. Наступна команда встановить PHP разом із рештою PHP-розширень, **необхідних для AtoM**:

```
1 sudo apt install php-common php7.4-common php7.4-cli php7.4-curl php7.4-json php7.4-ldap php7.4-mysql php7.4-opcache php7.4-readline php7.4-xml php7.4-mbstring php7.4-xsl php7.4-zip php-apcu php-apcu-bc
```

Якщо ви використовуєте Memcached як кешуючий механізм, вам також потрібно встановити php-memcache:

```
1 sudo apt install php-memcache
```

Сервер завдань Gearman необхідний для AtoM, починаючи з версії 2.2.

```
1 sudo apt install gearman-job-server
```

### 3.1.2 Завантаження AtoM

Після встановлення всіх залежностей ми готові завантажити та встановити AtoM. Найбезпечніший спосіб — це завантажити AtoM у вигляді архіву (tarball). Досвідчені користувачі можуть скористатися публічним репозиторієм для завантаження коду.

Всі наступні інструкції припускають, що ми встановлюємо AtoM у каталозі `/usr/share/nginx` та використовуємо AtoM версії 2.8.0.

```
1 wget https://storage.accesstomemory.org/releases/atom-2.8.0.tar.gz
2 sudo mkdir /usr/share/nginx/atom
3 sudo tar xzf atom-2.8.0.tar.gz -C /usr/share/nginx/atom --strip 1
```

**Примітка:** Архів може бути недоступний, якщо версія ще в розробці. В такому випадку скористайтеся альтернативним методом завантаження, описаним нижче.

Встановіть git:

```
1 sudo apt install git
```

Завантажте код AtoM з гілки stable/2.8.x:

```
1 sudo mkdir -p /usr/share/nginx/atom
2 sudo git clone -b stable/2.8.x http://github.com/artefactual/atom.git /usr/share/nginx/atom
```

Якщо вам потрібно завантажити лише певну кількість ревізій, скористайтеся командою:

```
1 git clone -b stable/2.8.x --depth 1 http://github.com/artefactual/atom.git /usr/share/nginx/atom
```

Ми використовуємо **Composer** для встановлення та управління сторонніми бібліотеками PHP. Для встановлення Composer виконайте інструкції на <https://getcomposer.org/d>. Після встановлення Composer виконайте команду для встановлення бібліотек:

```
1 cd /usr/share/nginx/atom
2 sudo ~/composer.phar install --no-dev
```

Або, якщо Composer встановлений глобально:

```
1 sudo composer install --no-dev
```

Для розробницького середовища, де потрібні dev-бібліотеки, використовуйте:

```
1 sudo ~/composer.phar install
```

Після завантаження коду необхідно скомпілювати файли теми.

```
1 sudo npm install
2 sudo npm run build
```

```
1 sudo apt install npm make
2 sudo npm install -g "less@<4.0.0" n
3 sudo n stable
4 sudo npm install
5 sudo npm run build
6 sudo make -C /usr/share/nginx/atom/plugins/arDominionPlugin
7 sudo make -C /usr/share/nginx/atom/plugins/arArchivesCanadaPlugin
8 sudo rm -rf node_modules
```

### 3.1.3 Створення бази даних

Припускаючи, що ви використовуєте MySQL на `localhost`, створіть базу даних за допомогою наступної команди, використовуючи пароль, який ви створили раніше:

```
1 sudo mysql -h localhost -u root -p -e "CREATE DATABASE atom CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci;"
```

**Примітка:** Якщо ви не вказуєте пароль MySQL root після параметра `-p`, вас буде запрошено ввести його після запуску команди. Якщо ви вказуєте пароль, не залишайте пробіл після `-p`; наприклад, `-pPASSWORD`. Пам'ятайте, що введення пароля в командному рядку менш безпечно, оскільки він може бути видимим для інших у файлі `.bash_history`.

Зауважте, що база даних названа `atom`. Ви можете змінити її назву за потреби.

Якщо сервер MySQL не знаходиться на тому ж сервері, що й ваш веб-сервер, замініть `localhost` на адресу вашого сервера MySQL.

**Попередження:** Переконайтеся, що ви використовуєте порожню базу даних! Не використовуйте стару базу даних, якщо вона не порожня. Ви завжди можете видалити її за допомогою команди `DROP DATABASE` і створити знову.

Також рекомендується створити окремого користувача MySQL для AtoM для підвищення безпеки. Нижче наведені команди для створення користувача з іменем `atom` з паролем `12345` та призначення йому необхідних прав доступу:

```
1 sudo mysql -h localhost -u root -p -e "CREATE USER 'atom'@'localhost' IDENTIFIED BY '12345';"
2 sudo mysql -h localhost -u root -p -e "GRANT ALL PRIVILEGES ON atom.* TO 'atom'@'localhost';"
```



**Примітка:** Права `INDEX`, `CREATE` та `ALTER` необхідні під час процесу встановлення або оновлення AtoM. Вони можуть бути видалені з користувача після завершення установки, або ви можете обмежити права користувача в файлі `config.php`.

### 3.1.4 Інсталювання

Тепер ви готові запустити інсталятор. Це проста команда, яка налаштовує AtoM відповідно до вашого середовища, додає необхідні таблиці та початкові дані до бази даних і створює індекс Elasticsearch.

```
1 cd /usr/share/nginx/atom
2 php symfony tools:install
```

Інсталятор запитає деталі конфігурації, такі як місцезнаходження сервера бази даних, ім'я користувача, пароль тощо. Приклад конфігурації:

- Database host: localhost
- Database port: 3306
- Database name: atom
- Database user: atom
- Database password: 12345
- Search host: localhost
- Search port: 9200
- Search index: atom

Надайте права на каталог AtoM:

```
1 sudo chown -R www-data:www-data /usr/share/nginx/atom
2 sudo chmod -R 755 /usr/share/nginx/atom
```

### Налаштування Worker для Gearman

Gearman використовується для підтримки асинхронних завдань. Створіть файл сервісу:

```
1 sudo nano /usr/lib/systemd/system/atom-worker.service
```

Вставте наступне:

```
1 [Unit]
2 Description=AtoM worker
3 After=network.target
4
5 [Service]
6 Type=simple
7 User=www-data
8 Group=www-data
9 WorkingDirectory=/usr/share/nginx/atom
10 ExecStart=/usr/bin/php7.4 -d memory_limit=-1 -d error_reporting="E_ALL" symfony jobs:worker
11 Restart=on-failure
12 RestartSec=30
13
14 [Install]
15 WantedBy=multi-user.target
```

Перезавантажте, активуйте і запустіть worker:

```
1 sudo systemctl daemon-reload
2 sudo systemctl enable atom-worker
3 sudo systemctl start atom-worker
```

## Налаштування PHP-FPM для AtoM

Встановіть PHP-FPM та додайте конфігураційний пул:

```
1 sudo apt install php7.4-fpm
2 sudo nano /etc/php/7.4/fpm/pool.d/atom.conf
```

Вставте наступне налаштування:

```
1 [atom]
2 user = www-data
3 group = www-data
4 listen = /run/php/php7.4-fpm.atom.sock
5 listen.owner = www-data
6 listen.group = www-data
7 listen.mode = 0600
8 pm = dynamic
9 pm.max_children = 30
10 pm.start_servers = 10
11 pm.min_spare_servers = 10
12 pm.max_spare_servers = 10
13 pm.max_requests = 200
```

Перезапустіть PHP-FPM:

```
1 sudo systemctl enable php7.4-fpm
2 sudo systemctl start php7.4-fpm
```

## Налаштування Nginx для AtoM

Встановіть Nginx і створіть конфігурацію для AtoM:

```
1 sudo apt install nginx
2 sudo nano /etc/nginx/sites-available/atom
```

Додайте наступне налаштування:

```
1 server {
2     listen 80;
3     server_name localhost;
4
5     root /usr/share/nginx/atom;
6     index index.php;
7
8     location / {
9         try_files $uri $uri/ /index.php?$query_string;
10    }
11
12    location ~ \.php$ {
13        include snippets/fastcgi-php.conf;
14        fastcgi_pass unix:/run/php/php7.4-fpm.atom.sock;
15    }
16
17    location ~ /\.ht {
18        deny all;
19    }
20 }
```

Активуйте конфігурацію та перезапустіть Nginx:

```
1 sudo ln -s /etc/nginx/sites-available/atom /etc/nginx/sites-enabled/
2 sudo systemctl restart nginx
```

Тепер відкрийте браузер і перейдіть за адресою `http://localhost`, щоб перевірити, чи працює AtoM.

### 3.2 Windows

Кожен елемент програмного забезпечення, **необхідний для AtoM**, сумісний з Windows. Однак слід знати, що процес може бути зовсім не простим, якщо ви не знайомі з серверними середовищами під платформою Windows.

## 4 Конфігурація

Система **AtoM** надає багато можливостей щодо конфігурації. У цьому розділі буде описано як модерацію системи зі сторони web-сторінки, використовуючи роль адміністратора, так і ручну зміну налаштувань у внутрішніх файлах самого проекту.

### 4.1 Маніпуляція налаштуваннями з веб-сторінки

### 4.2 Конфігураційні файли

## 5 Постскрипtum

Маємо дякувати за прочитання