Руководство по инсталляции

2023

Содержание

[1.Введение. 3](#_Toc151915103)

[2.Требования к аппаратному и программному обеспечению. 4](#_Toc151915104)

[3.Инсталляция Программного обеспечения. 5](#_Toc151915105)

# 1.Введение.

Облако должно обеспечить:

Централизованное хранение данных, распределенное по темам. Файловый обменник предоставляет централизованное место для хранения всех проектных данных, чертежей, спецификаций, расчетов и других документов, связанных с проектированием здания. Хранение данных распределено по ячейкам, которые закреплены по необходимым темам проектной работы. Таким образом все файлы по проекту находятся в одном месте, а не разбросаны по разным устройствам и папкам, а также файлы отсортированы по темам для того, чтобы файлы были не перемешаны и доступ к ним был получен сразу.

Загрузку, архивирование и резервное копирование файлов. Файлы можно загружать в любом формате, для того чтобы вся документация, чертежи и схемы в любом формате могли находится в программе. Доступ к истории загрузки необходим для наблюдения прогресса, а также для работы над предыдущими версиями. Архив и резервные копии обеспечат сохранность данных в случае сбоев.

Открытый доступ для всех сторон. Компании проектирования необходимо сотрудничать с различными командами и специалистами, включая архитекторов, инженеров, дизайнеров и клиентов. Открытый доступ упрощает работу, позволяя всем сторонам видеть последние версии файлов и обсуждать их в режиме реального времени.

Авторизацию пользователей. Обменник позволяет определять, кто работает над конкретными задачами, и следить за ходом реализации проекта, это позволяет следить за сроками выполнения работы.

Документ "Руководство по инсталляции"(далее Руководство) к комплекту эксплуатационных документов программного обеспечения Облачного хранилища для строительной компании с функцией архивации.

Данное руководство предназначено для администраторов, осуществляющих установку и первоначальную настройку системы.

Руководство содержит следующую информацию:

требования к системе, на которую устанавливается ПО;

процесс настройки операционной системы перед установкой ПО.

# 2.Требования к аппаратному и программному обеспечению.

Персональные компьютеры (ПК), предназначенные для установки ПО, должны соответствовать следующим требованиям:  
Ubuntu 20.04-22.04  
4 ядра от процессора intel xeon E5-2660v3  
Память 4 Гб  
Ssd накопитель - на 2 ТБ.

Софт содержит следующие требования:  
Java 17  
Maven  
Certbot  
Nginx  
Postgresql  
git  
wget

# 3.Инсталляция Программного обеспечения.

Для того чтобы установить ПО, выполните следующие действия:

1. Покупка домена: либо Заказчик покупает его за деньги, либо пользуется бесплатной версией.
2. Дальше находим бесплатную виртуальную машину, либо также покупаем платную версию.
3. На виртуальную машину ставим Ubuntu 20 (операционная система Linux).
4. Берем ip-адрес нашей виртуальной машины и добавляем в а-запись домена. Чтобы домен заработал нужно подождать около суток.
5. Ставим на виртуальную машину Java, Web-Сервер Ngixn, Postgres для БД, Certbot для Ngixn.

1. sudo apt install openjdk-17-jre - эта команда устанавливает Java Runtime Environment (JRE) для Java версии 17. sudo дает команде права суперпользователя, а apt (Advanced Package Tool) является системой управления пакетами в Debian и Ubuntu, которая устанавливает, обновляет и удаляет программы.

2. sudo apt install nginx - устанавливает веб-сервер Nginx.

3. sudo apt -y install postgresql - устанавливает систему управления базами данных PostgreSQL. Флаг -y автоматически подтверждает все запросы о необходимости установки, что позволяет установке пройти без взаимодействия с пользователем.

4. sudo apt install python3-certbot-nginx - устанавливает Certbot для Nginx, автоматический клиент, который позволяет получить SSL-сертификаты от Let's Encrypt и настраивает их для использования с Nginx. В данном случае, Certbot установится для Python 3.

1. Обновляем индекс пакетов.

5. sudo apt update - обновляет индекс пакетов. Выполнив эту команду, система узнает о последних доступных версиях пакетов и их зависимостях.

1. Вносим таблицы в БД.

6. sudo -i -u postgres - переключает пользователя из текущего в пользователя postgres с помощью интерактивного режима. Пользователь postgres обычно используется для администрирования PostgreSQL.

7. psql -U postgres - запускает PostgreSQL командный интерфейс (psql) под пользователем postgres. Это позволяет выполнять SQL команды напрямую.

8. createdb EtuProject - создает новую базу данных с именем EtuProject в PostgreSQL.

9. \password postgres - устанавливает или изменяет пароль для пользователя postgres в PostgreSQL.

10. \q - выходит из интерактивного режима psql.

11. psql -d EtuProject -подключается к ранее созданной базе данных EtuProject для выполнения SQL команд.

12. Далее следует серия команд SQL для создания таблиц users, files, comments, cards в базе данных EtuProject. Эти команды определяют схему таблиц и их структуру, включая первичные ключи, типы данных и другие ограничения.

CREATE TABLE IF NOT EXISTS users(id bigint NOT NULL DEFAULT nextval('users\_id\_seq'::regclass),mail character varying(30) COLLATE pg\_catalog."default" NOT NULL,password character varying(32) COLLATE pg\_catalog."default" NOT NULL,"time" bigint NOT NULL DEFAULT nextval('users\_time\_seq'::regclass),deleted boolean NOT NULL DEFAULT false,salt character varying(10) COLLATE pg\_catalog."default",usertype smallint NOT NULL DEFAULT 0,iname character varying(15) COLLATE pg\_catalog."default" NOT NULL DEFAULT '',fname character varying(15) COLLATE pg\_catalog."default" NOT NULL DEFAULT '',oname character varying(15) COLLATE pg\_catalog."default" NOT NULL DEFAULT '',idmanager bigint NOT NULL DEFAULT 0,CONSTRAINT users\_pkey PRIMARY KEY (id));

CREATE TABLE IF NOT EXISTS files(id bigint NOT NULL GENERATED BY DEFAULT AS IDENTITY (INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 999999999999999 CACHE 1),realname character varying(30) NOT NULL,genname character varying(35) NOT NULL,"time" bigint NOT NULL,type character varying(5) NOT NULL,idcard bigint NOT NULL,size bigint NOT NULL,deleted boolean NOT NULL,idown bigint NOT NULL,CONSTRAINT files\_pkey PRIMARY KEY (id));

CREATE TABLE IF NOT EXISTS comments(id bigint NOT NULL GENERATED BY DEFAULT AS IDENTITY (INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 9999999999999 CACHE 1),idown bigint NOT NULL,idcard bigint NOT NULL,body character varying(500) NOT NULL,deleted boolean NOT NULL DEFAULT false,"time" bigint NOT NULL,CONSTRAINT comments\_pkey PRIMARY KEY (id));

CREATE TABLE IF NOT EXISTS cards( id bigint NOT NULL GENERATED BY DEFAULT AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 999999999999 CACHE 1 ), name character varying(30) NOT NULL, "time" bigint NOT NULL, idobject bigint NOT NULL, deleted boolean NOT NULL DEFAULT false, idown bigint NOT NULL DEFAULT 0, description character varying(100) NOT NULL, status character varying NOT NULL, idexecutor bigint NOT NULL DEFAULT 0, CONSTRAINT cards\_pkey PRIMARY KEY (id));

13. \q (повторно)

Выходит из интерактивного режима psql.

14. sudo -i -u root - переключение из текущего пользователя на пользователя root в интерактивном режиме. Пользователь root имеет полные привилегии администратора в системе Linux.

1. Клонируем с GitHub проект.

15. git clone <https://github.com/FlexedDkim/EtuProject.git> - клонирует репозиторий с исходным кодом проекта EtuProject с GitHub.

16. cd EtuProject - переход в директорию, куда был клонирован репозиторий EtuProject.

1. С помощью Maven собираем проект и с помощью Spring Boot запускаем проект.

С помощью команд:

17. mvn install - выполняет цель Maven install, которая компилирует исходный код проекта и упаковывает его в формат, готовый к использованию (например, JAR или WAR), а также устанавливает его в локальный репозиторий.

18.mvn sprin-boot:run - запускает весь проект окончательно.

1. После запуска переходим в браузер, заходим на сайт и видим страницу с Авторизацией. Переходим на страницу Регестрации.
2. Регистрируем Обычного пользователя, повышаем его до Администратора и пользуемся функционалом нашего ПО.