**[1]**

**Добрый день, тема моего грейда:  
Практическая часть:**

**[2]**

NFS (Network File System) — это протокол распределенной файловой системы, позволяющий пользователю клиентского компьютера получать доступ к файлам по сети аналогично тому, как осуществляется доступ к локальным хранилищам. Разработанный компанией Sun Microsystems в 1984 году, стал стандартом для совместного использования файлов в системах Unix и Linux.

Компоненты NFS

* **NFS-сервер:** машина или устройство, которое предоставляет доступ к своим файлам и каталогам другим системам по сети. Он запускает программное обеспечение сервера NFS и управляет контролем доступа и блокировкой файлов (в более поздних версиях).
* **Клиент NFS:** машина или устройство, которое получает доступ к общим файлам и каталогам с сервера NFS. Он монтирует общие ресурсы в свою локальную файловую систему, делая их видимыми как локальные файлы.
* **Удаленный вызов процедур (RPC):** протокол без сохранения состояния и установления соединения, который позволяет программе на одном компьютере запускать программу на другом компьютере. Он используется NFS для связи между клиентами и серверами.
* **Протокол NFS:** определяет формат и процедуры для операций NFS, таких как чтение, запись файлов и управление метаданными. Он работает поверх RPC.

[3]

Принцип работы NFS  
NFS работает по модели клиент-сервер. На сервере размещаются файлы и каталоги, к которым предоставляется общий доступ, а клиент получает доступ к этим общим ресурсам, как если бы они хранились локально. Кросс-платформенная совместимость NFS гарантирует, что пользователи смогут получать доступ к файлам из разных операционных систем, что делает его универсальным – NFS будет работать в любых ОС, поддерживающих файловую систему. На слайде так же указаны основные версии NFS, показывающие этапы его развития

[4]

Процесс работы:

1. Сервер экспортирует каталоги и запускает службы NFS.
2. Клиент отправляет серверу запрос на монтирование.
3. Сервер обрабатывает запрос и аутентифицирует клиента.
4. После аутентификации Сервер предоставляет доступ, а клиент монтирует общий каталог.
5. Клиент запрашивает доступ к файлу для операций чтения/записи.
6. Сервер обрабатывает эти запросы и передает данные.
7. Клиент взаимодействует с файлом так, как если бы он находился в локальном хранилище.
8. По завершении работы клиент размонтирует файловую систему.
9. Сервер на протяжении своей работы регистрирует действия пользовтелей для мониторинга и устранения неполадок.

[5]

Безопасность в NFS версий 2 и 3 обеспечивалась на уровне доверия к IP адресам юзеров. В более поздней 4й версии появилась аутентификация через различные протоколы, например – Церберус

Состояния файлов в NFS старых версий не отслеживались, что могло приводить к ошибкам при одновременной работе с файлами – о подобной проблеме в Django нам рассказывал Сергей в прошлый раз, здесь похожая ситуация. В более новых версиях NFS научился управлять сессиями и блокировать файлы.

[6]

Преимущества и недостатки:

[7]

На этом с NFS все, перейдем к S3

S3 буквально – сервис простого хранения

S3 (Simple Storage Service) — протокол передачи данных, разработанный компанией Amazon. Он же — объектное хранилище. Его уникальность заключается в хранении огромного объема данных в исходном формате без какой либо иерархии и разбивки на отдельные каталоги. У хранилища S3 нет ограничений по масштабированию, это виртуальная СХД, доступ к которой возможен из любой точки, где есть интернет. Пользователи настраивают хранилище под свои требования и задают приоритеты объектам.

[8]

Основные компоненты, относящиеся к хранимым данным:

Бакеты –

Объекты –

Метаданные -

[9]

S3 Имеет доступ через API, что делает удобным взаимодействие с хранилищем.

Что такое объектное хранилище?

это способ хранить данные и гибко получать к ним доступ как к объектам.

в объектных хранилищах нет такого понятия, как структура каталогов. Все объекты находятся в одном условном «каталоге» – bucket. Структурирование данных предлагается делать на уровне приложения. Основное преимущество хранения данных в объектах – это возможность абстрагирования системы от технических деталей. Нас уже не интересует, какая файловая (или тем более операционная) система хранит наши данные. Мы не привязываемся к данным какими-то конкретными способами их представления, которые нам обеспечивает платформа.

[10]

Преимущества S3:

* S3 имеет Возможность быстро обратиться к файлу по его идентификатору, получить его метаданные. При этом в бэкенде у хранилища может быть всё, что угодно, API запросов будет везде одинаковый.
* Данные могут храниться в отдельных контейнерах со своими настройками доступа, хранения, производительности и т.д. То есть их удобно структурировать и разделять информацию по различным типам.
* Все объекты хранилища располагаются в плоском адресном пространстве, без иерархии, которая присутствует в обычной файловой системе. Это упрощает доступ и работу с файлами. Нет проблем именами файлов.
* для провайдера предоставляющего сервис S3 есть возможность Удобной тарификации в зависимости от активности клиентов. Это позволяет клиентам платить только за потребляемые ресурсы: объём, трафик, количество запросов.

[11]

Из недостатков почему-то везде указывается отсутствие файловой системы, которое мне хочется записать в достоинства, но в принципе ясно, по какой причине это считается минусом – например, при миграции данных могут возникать некоторые проблемы с организацией хранения данных. В случае нашего проекта я сомневаюсь, что такие проблемы возникнут, но будем все равно иметь это ввиду.

[12]

К слову о миграции данных. На слайде изображен приблизительный алгоритм миграции данных в условиях работающего приложения.

[13]

Взаимодействие Django с S3 осуществляется с помощью библиотеки django-storages, для ручного взаимодействия с S3 можно использовать boto3

Минимальные настройки проекта, позволяющие работать связать Django-приложение с S3 представлены на скрине.

[14]

Переходя к практической части, сразу оговорюсь:

В рамках тестового стенда для наглядности был выбран сервис MiniO - объектное хранилище с открытым кодом, совместимое с S3 API.