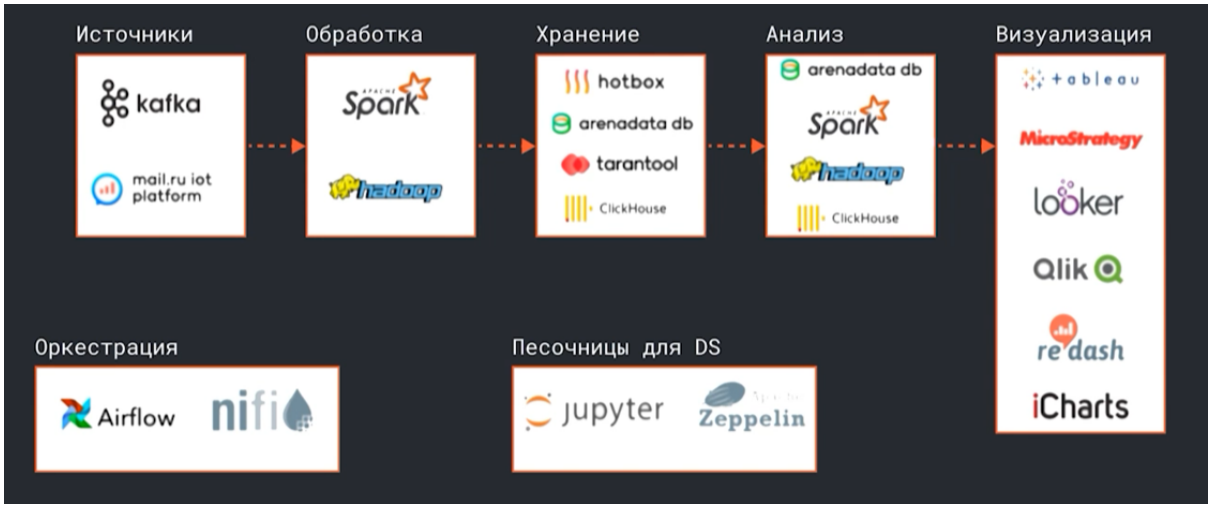
KARPOV.COURSES >>>
КОНСПЕКТ

> Конспект > 4 урок > Особенности построения Data решений в облаках для DE

> Оглавление

> Традиционный подход

При построении различных архитектур в облаках можно придерживаться традиционного подхода, который аналогичен построению своего решения on-prem у себя в датацентре. Множество компонентов: источников, обработки, хранения, анализа, визуализации и т.д., можно развернуть в облаке используя IaaS, т.е. устанавливаем на полученные сервера что и как хотим и далее самостоятельно администрируем.

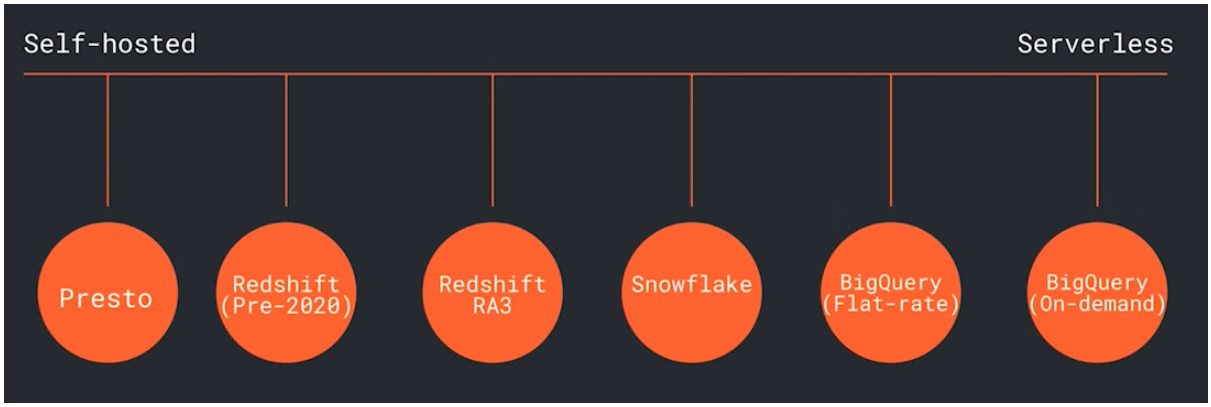


Чтобы получить максимальную выгоду от облаков, лучше использовать PaaS или SaaS решения. Далее рассмотрим некоторые PaaS или SaaS решения, которые встречаются при работе с данными в облаках.

> Self-hosted vs Serverless

Рассмотрим такие системы как Redshift, Snowflake, BigQuery и Presto.

Отличаются они между собой прежде всего уровнем работы с инфраструктурой: некоторые из этих систем можно самостоятельно установить, настроить и управлять (self-hosted подход), а у других практически нет возможности производить настройки системы (serverless подход).



Выбирая облачные решения, стоит учитывать, что будет сложно впоследствии переходить на on-prem решения. И чем более serverless или SaaS решение, тем более тяжёлым будет процесс отказа от него.

> Сравнение систем

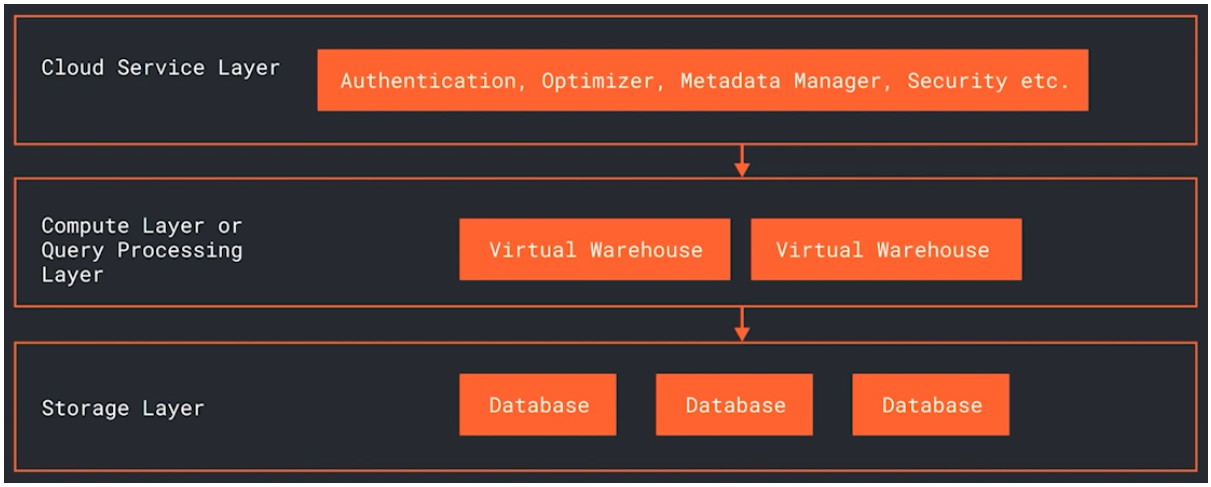
Если сравнивать эти системы по стоимости и производительности, то они довольно схожи.

Presto несколько медленнее, если сравнивать с конкурентами, но по стоимости исполнения запросов этот недостаток компенсируется. BigQuery тоже сравнительно медленнее и несколько дороже.

> Snowflake

Архитектура Snowflake состоит из 3 слоёв:

1. Cloud Service Layer отвечает за аутентификацию, авторизацию, вопросы связанные с безопасностью, хранением метаданных. Первичный слой, куда приходят запросы, далее отправляет запрос на более низкие слои.
2. Compute Layer or Query Processing Layer – слой, на котором происходят вычисления.
3. Storage Layer – базы данных, за этот слой отвечает S3.



В отличие от ClickHouse или GreenPlum, в которых каждая нода одновременно и storage и compute, а значит можем увеличивать только одновременно и то и другое, при архитектуре похожей на Snowflake можно независимо масштабировать storage или compute слои.

> Особенности Snowflake

- облачное хранилище данных по модели SaaS
- три слоя: Storage, Compute, Services
- у пользователей нет прямого доступа к данным в Storage: загружаем данные, используя API или другие инструменты, предоставляемые Snowflake
- управление структурой файлов, размером, сжатием Snowflake берёт на себя
- автоматическое разделение данных на микропартиции, что ускоряет время выполнения запросов
- можно запускать виртуальные кластеры под разные нагрузки и команды: разные пользователи могут обращаться к единому storage слою, используя разные кластера compute, т.е. разные виртуальные кластера. Compute слой создаётся отдельно под каждую задачу. Таким образом, устраняется конкуренция за ресурсы
- повторные запросы быстро работают за счёт кэширования
- полностью поддерживают ACID транзакции

> Преимущества Snowflake

- встроенная функция Timetravel: возможность обращаться по метке timestamp к состоянию таблицы в прошлом
- Auto-Suspend и Auto-Resume: автоматическое гашение кластеров в compute слое
- Dynamic Data Masking: при работе с персональными данными или данными, требующими осторожного обращения, для недопущения утечки данных можно использовать маскирование данных (например, некоторые поля заменяются другими данными)
- лёгкое и удобное масштабирование

> Дополнительные материалы по Snowflake

https://www.snowflake.com/product/architecture/

https://towardsdatascience.com/migrating-to-snowflake-like-a-boss-6163293f0bcb

https://towardsdatascience.com/why-you-need-to-know-snowflake-as-a-data-scientist-d4e5a87c2f3d

https://medium.com/hashmapinc/snowflakes-cloud-data-warehouse-what-i-learned-and-why-i-m-rethinking-the-data-warehouse-75a5daad271c

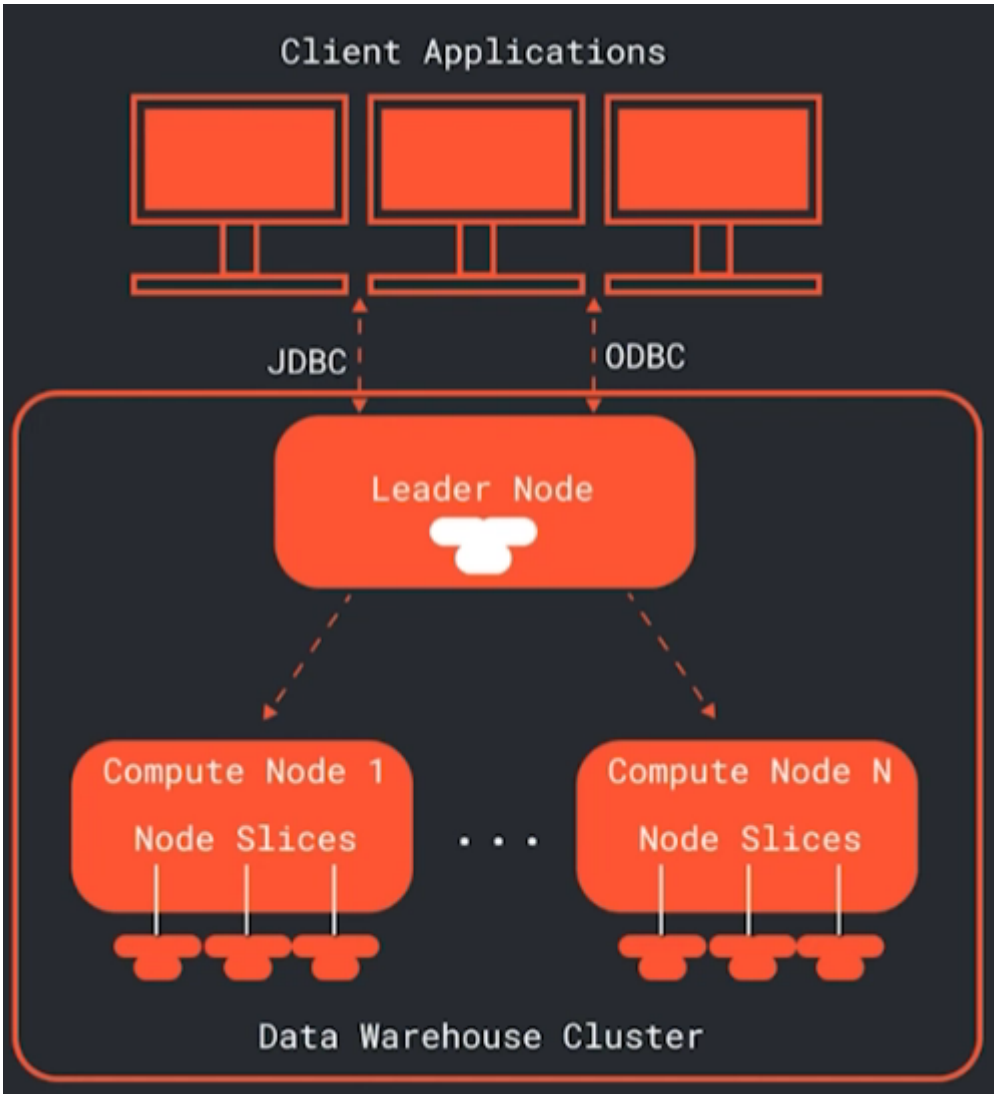
https://habr.com/ru/company/oleg-bunin/blog/514298/

https://habr.com/ru/company/manychat/blog/530054/

> AWS Redshift

Реализует однотенантную (single-tenant) модель, при которой инфраструктура принадлежит одному пользователю, т.е. хранилище и сервера полностью контролируются, а другие пользователи их не используют.

Архитектура AWS Redshift имеет более классический вид, похожа на GreenPlum: есть одна Leader Node и несколько Compute Node.



> Особенности AWS Redshift

- предоставляется по модели PaaS: придётся тюнить множество настроек (от них будет зависеть производительность), нужен администратор системы
- построен на базе Postgres
- классическая MPP архитектура, напоминающая Greenplum
- нет разделения на storage и compute слои
- Redshift Spectrum позволяет обращаться к данным в S3
- Redshift RA3 позволяет разделить storage и compute (с большими по сравнению со Snowflake затратами и усилиями)
- медленное и тяжёлое скалирование

> Дополнительные материалы по AWS Redshift

https://hevodata.com/blog/redshift-architecture/

https://aws.amazon.com/ru/redshift/features/ra3/

https://aws.amazon.com/ru/blogs/big-data/introducing-amazon-redshift-ra3-xlplus-nodes-with-managed-storage/

https://levelup.gitconnected.com/snowflake-vs-redshift-ra3-the-need-for-more-than-just-speed-52e954242715

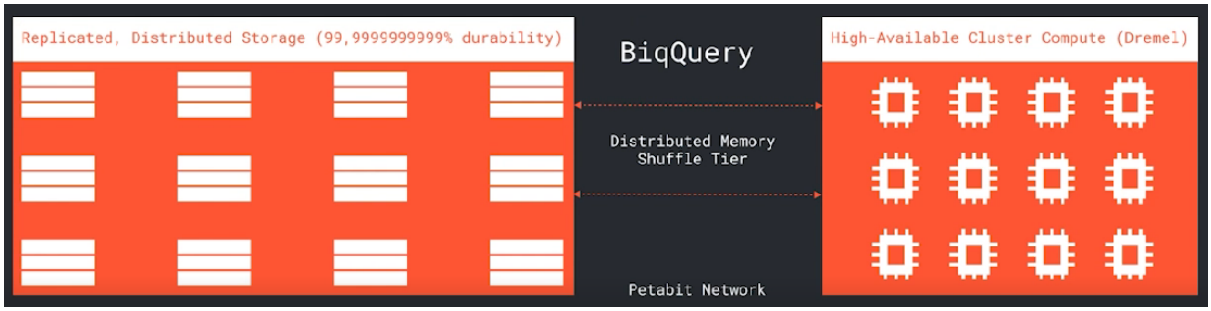
https://betterprogramming.pub/building-a-data-warehouse-on-amazon-redshift-49a620a5392f

---

> BigQuery

Реализует сложную мультитенантную (multi-tenant) модель, при которой единый сервис обслуживает разных пользователей системы.

В огромном хранилище данных хранятся ваши данные и данные других клиентов (storage слой), вычисления производятся в гигантском кластере (тысячи или даже десятки тысяч машин), в котором одновременно может производиться огромное количество запросов (compute слой). Слои compute и storage связаны между собой петабитной сетью.



> Особенности BigQuery

- предлагается по модели SaaS
- query-based pricing model: плата за каждый запрос, плата варьируется от объёма данных, которые были просканированы в ходе выполнения запроса
- стоимость storage рассчитывается для несжатых данных
- разделение storage и compute

- есть поддержка запросов к внешним источникам внутри экосистемы GCP (Google Cloud Platform)
- удобный ML внутри хранилища
- динамическое масштабирование «на лету» для индивидуальных запросов

> Дополнительные материалы по BigQuery

https://cloud.google.com/blog/products/data-analytics/new-blog-series-bigquery-explained-overview

https://panoply.io/data-warehouse-guide/bigquery-architecture/

https://servian.dev/snowflake-vs-bigquery-imo-c94bc6d5e4c9

https://www.xplenty.com/blog/snowflake-vs-bigquery/

> AWS Athena

Несмотря на то, что AWS Athena или Presto это сервисы запросов, и на них нельзя построить полноценное хранилище, они являются удобными системами для аналитических запросов, если данные распределены по разным системам и командам, и будут единой точкой обращения к этим данным.

> Особенности AWS Athena

- построена на базе Presto DB, представляет собой интерактивный сервис запросов: является open-source, можно установить у себя on-prem или установить в облаке внутри системы IaaS. Является по сути не хранилищем данных, а сервисом запросов, т.е. не хранит в себе данные
- интегрируется с AWS Glue (аналог Hive MetaStore) – бессерверная служба интеграции данных, упрощающая их поиск, подготовку и объединение из разных источников
- предлагается по модели SaaS, Serverless
- нет ACID, транзакций
- также использует S3 для хранения
- нет возможности вручную настраивать размер кластера под запрос

- партиционирование настраивается вручную: при правильном подходе можно получить более эффективное обращение к данным, т.к. будет сканироваться небольшой объём данных

> Дополнительные материалы по AWS Athena

https://aws.amazon.com/ru/athena/faqs/

https://www.cloudzero.com/blog/aws-athena

> Интересные материалы

https://poplindata.com/data-warehouses/2021-database-showdown-bigquery-vs-redshift-vs-snowflake/

https://a16z.com/2021/05/27/cost-of-cloud-paradox-market-cap-cloud-lifecycle-scale-growth-repatriation-optimization/

https://hevodata.com/learn/snowflake-vs-redshift-vs-bigquery/

https://medium.com/2359media/redshift-vs-bigquery-vs-snowflake-a-comparison-of-the-most-popular-data-warehouse-for-data-driven-cb1c10ac8555

> Подводя итог

На что нужно обращать внимание при выборе хранилища в облаке:

- по какой модели оно предоставляется: PaaS, SaaS или самостоятельно развернёте систему над IaaS
- какие настройки будут находиться в вашей зоне ответственности: для понимания необходимости наличия команды инженеров, администраторов.