# **4,11,18,25,32,39,46,53**

***4. Дайте характеристики моделям предоставления облачных сервисов***

Облачные услуги в основном предоставляются по трем моделям: программное обеспечение как услуга (*[SaaS](https://cloud.yandex.ru/blog/posts/2023/03/saas)*, *Software-as-a-Service*), платформа как услуга (*[PaaS](https://cloud.yandex.ru/blog/posts/2023/03/paas)*, *Platform-as-a-Service*) и инфраструктура как услуга (*[IaaS](https://cloud.yandex.ru/blog/posts/2022/01/iaas)*, *Infrastructure-as-a-Service*).

**Программное обеспечение как услуга** — Программные решения, не требующие установки на стороне пользователя. Программа размещается на серверах поставщика, а клиент получает к ней доступ через браузер или предустановленную интерфейсную оболочку. Примером таких решений служат различные почтовые службы, системы резервирования, виртуальные рабочие столы.

**Платформа как услуга** — готовые комплексные решения под определенные задачи. Пользователь использует аппаратные и программные ресурсы поставщика, настроенные специально для выполнения конкретных действий. Это могут быть системы машинного обучения и управления базами данных или среды разработки и тестирования приложений.

**Инфраструктура как услуга** — предоставление вычислительных мощностей в форме виртуальной инфраструктуры с заданной конфигурацией, требуемого объема и оснащенной системным программным обеспечением (например, хостинг). Модель включает в себя базовые элементы для построения облачной ИТ-системы. В рамках этой модели пользователи получают доступ к сетевым ресурсам, к виртуальным компьютерам или выделенному аппаратному обеспечению, а также к хранилищам данных. Модель «инфраструктура как услуга» обеспечивает наивысший уровень гибкости эксплуатации и управления ИТ-ресурсами

! Чем более высокоуровневая модель используется, тем меньший уровень ИТ компетенции требуется от компании. И наоборот – чем ниже уровень ИТ компетенции организации, тем более высокоуровневая модель требуется.

***Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание***

Изображение выглядит как текст, снимок экрана, Шрифт, Параллельный

Автоматически созданное описание

***Изображение выглядит как текст, снимок экрана, число

Автоматически созданное описание***

***11. В чём преимущество программно-определяемых ЦОДов (Software Defined Data Centers)?***

Программно-определяемый, или виртуальный ЦОД — это совокупность виртуальных элементов для хранения, обработки и передачи данных, абстрагированных от базового оборудования и управляемых через интернет и выделенные каналы.

**Перечислим основные плюсы виртуальных ЦОД**:

* **автоматизация и оптимизация**. Благодаря использованию программного обеспечения для управления инфраструктурой виртуальный ЦОД позволяет автоматизировать настройку сетевых коммутаторов, настройку виртуальных машин и т.д. Все управление происходит в удаленном формате;
* **отказ от капитальных затрат на покупку оборудования и снижение накладных расходов.**Виртуальный ЦОД позволяет отказаться от покупки оборудования и более эффективно использовать имеющиеся ресурсы, так как каждый компонент инфраструктуры может быть настроен и оптимизирован программным обеспечением.
* **гибкость и масштабируемость.** Программно-определяемые дата-центры позволяют быстро и гибко настраивать и масштабировать инфраструктуру ЦОДа в соответствии с потребностями бизнеса. Это позволяет ускорить процессы развертывания приложений и сервисов, а также повысить гибкость работы ЦОДа в целом;
* **сокращение времени на развертывание инфраструктуры.** Без необходимости приобретать оборудование для проекта и размещать его в физическом ЦОД. При этом время на развертывание сокращается месяцев до нескольких часов или минут;
* **эффективность использования ресурсов.**Программно-определяемые ЦОДы позволяют более эффективно использовать ресурсы серверов, хранилищ данных и сетевых устройств, что, в свою очередь, позволяет снизить затраты на оборудование и управление;
* **высокая доступность и отказоустойчивость**. Программно-определяемый центр обработки данных обеспечивает высокую доступность инфраструктуры ЦОДа благодаря гибкому управлению ресурсами и высокому уровню автоматизации. Это позволяет минимизировать время простоя и снижать риски проблем с доступностью приложений и сервисов. Отказоустойчивость достигается за счет использования технологий резервирования, балансировки нагрузки и т.д*.*

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

***18. Для каких задач организации с широкополосным доступом к своим сайтам используют сервис Load Balancing (в т.ч. его реализацию в «облаке» LBaaS)?***

LBaaS обеспечивает автоматическое управление и настройку серверов приложений и распределяет нагрузку между ними на основе определенных критериев, таких как количество запросов к серверу, текущая загрузка и пропускная способность.

Сервис Load Balancing используется для повышения доступности, надежности и производительности веб-приложений и сайтов, которые имеют большое количество посетителей и запросов. Некоторые из задач, для которых организации используют сервис Load Balancing, включают:

1. **Распределение нагрузки:** вычислительные ресурсы могут быть распределены между несколькими серверами для балансировки нагрузки на каждом из них. Это позволяет предотвратить перегрузку одного сервера и обеспечить более высокую доступность и производительность сайта.
2. **Высокая доступность:** если один из серверов недоступен, сервис Load Balancing может автоматически перенаправить трафик на другой сервер, чтобы обеспечить бесперебойную работу сайта. Это также может выполняться с использованием устройств, находящихся на разных физических местах, что увеличивает сопротивляемость системы к отказам.
3. **Обеспечение безопасности:** сервис Load Balancing может использоваться для обеспечения безопасности веб-приложений путем распределения нагрузки и снижения риска DDoS-атак
4. **Масштабирование:** сервис Load Balancing позволяет масштабировать инфраструктуру сайта, добавляя новые серверы по мере необходимости.
5. **Улучшение производительности:** сервис Load Balancing может улучшить производительность сайта, воспользовавшись вычислительной мощностью нескольких серверов.

***25.Опишите функциональные задачи ПО Apache Kafka.***

Kafka Apache — распределенная система обмена сообщениями между серверными приложениями в режиме реального времени. Благодаря высокой пропускной способности, масштабируемости и надежности применяется в компаниях, работающих с большими объемами данных. Написана на языках Java и Scala.

Основное назначение Apache Kafka — централизованный сбор, обработка, безопасное хранение и передача сообщений от разных сервисов. Обычно систему выбирают, когда нужно работать с большим количеством неструктурированных данных.

Где может применяться: системы аналитики, финансовые системы, социальные сети, онлайн-игры.

**Применение:**

• **Для связи микросервисов**. Kafka — связующее звено между отдельными функциональными модулями большой системы. Например, с ее помощью можно подписать микросервис на другие компоненты для регулярного получения обновлений.

• **Потоковая передача данных.** Высокая пропускная способность системы позволяет поддерживать непрерывные потоки информации. За счет грамотной маршрутизации «Кафка» не только надежно передает данные, но и позволяет производить с ними различные операции.

• **Ведение журнала событий.** Kafka сохраняет данные в строго организованную структуру, в которой всегда можно отследить, когда произошло то или иное событие. Информация хранится в течение заданного промежутка времени, что можно использовать для разгрузки базы данных или медленно работающих систем логирования.

• **Агрегация записей.** В Apache Kafka можно писать данные куда быстрее, чем в обычную базу данных. Это значит, что с помощью сообщений можно организовать сбор кучи метрик, считать от них, например, средние, и уже эти значения писать в БД.

**Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание**

***32.Объясните принципы маршрутизации микросервисов и особенности работы Istio (Service Mesh).***

Принципы маршрутизации микросервисов:

1. Каждый микросервис должен иметь уникальный идентификатор (ID), чтобы можно было обращаться к нему через API.

2. Маршрутизация должна быть основана на конфигурации, которая хранится централизованно, чтобы обеспечить единообразие.

3. Маршрутизация должна учитывать состояние микросервисов, чтобы запросы не были направлены на неработающие сервисы.

4. Необходима возможность добавления новых микросервисов без изменения конфигурации маршрутизации.

5. Управление версиями: Маршрутизатор должен уметь переключаться между различными версиями сервисов в зависимости от требований приложения.

6. Балансировка нагрузки: Маршрутизатор должен распределять трафик между несколькими экземплярами сервиса для обеспечения высокой доступности и производительности.

**Istio** является независимым Service Mesh с открытым исходным кодом, которая обеспечивает основы, необходимые для успешной работы архитектуры распределенных микросервисов.

**Service Mesh** – это выделенный уровень инфраструктуры для обработки связи между сервисами. Он отвечает за надежную доставку запросов через сложную топологию сервисов, которые составляют современное облачное собственное приложение.

1. Istio работает как service mesh (сеть сервисов), которая обеспечивает контроль соединений между микросервисами и обеспечивает безопасность (аутентификацию и авторизацию), балансировку нагрузки и маршрутизацию.

2. Мониторинг: Istio предоставляет возможность сбора и анализа данных о работе приложения, таких как логи и метрики.

3. Управление версиями: Istio позволяет управлять версиями микросервисов, что облегчает развертывание новых версий и откат к предыдущим версиям.

4. Расширяемость: Istio является расширяемым и может быть интегрирован с другими инструментами для управления контейнерами и оркестраторами.

5. Независимость от языка программирования: Istio не зависит от языка программирования, используемого для написания микросервисов, что облегчает интеграцию с различными технологиями.

***39. Раскройте особенности (преимущества и недостатки) серверной виртуализации на базе KVM***.

**KVM (Kernel-based Virtual Machine)** виртуализация — это технология виртуализации на уровне ядра Linux, которая позволяет создавать виртуальные машины на основе ресурсов физического сервера. Виртуальная машина – это программное приложение, исполняющее роль независимого компьютера в рамках другого физического компьютера.

***Преимущества серверной виртуализации на базе KVM:***

1. **Безопасность**.

Благодаря использованию отдельного ядра для каждой виртуальной машины KVM обеспечивает полную изоляцию между ними, предотвращая возможность распространения вредоносных программ и утечки данных.

1. **Производительность.**

KVM работает непосредственно с аппаратным обеспечением, что обеспечивает высокую производительность при выполнении задач в виртуальных машинах.

1. **Гибкость**.

KVM предоставляет предприятиям множество вариантов при инсталляции, поскольку работает с различными аппаратными установками.

1. **Масштабируемость.**

KVM позволяет легко масштабировать виртуальные машины в зависимости от потребностей пользователя, а также создавать и поддерживать большое количество виртуальных машин на одном хосте с различными характеристиками.

1. **Интеграция с другими технологиями.**

KVM может интегрироваться с другими технологиями, такими как OpenStack, что позволяет использовать его в различных сценариях виртуализации.

1. **Различные методы шифрования.**

KVM может использовать различные методы шифрования для обеспечения дополнительного уровня безопасности.

1. **Быстрое создание виртуальных машин.**

KVM позволяет быстро создавать и удалять виртуальные машины, что делает его удобным для тестирования и разработки.

1. **Типы хранилища.**

KVM позволяет использовать различные типы хранилища, включая локальные диски, сетевые хранилища и хранилища на базе SAN и NAS.

1. **Поддержка LIVE MIGRATION.**

KVM позволяет перемещать виртуальные машины между физическими серверами без прерывания их работы, что обеспечивает более высокую доступность и удобство управления.

***Недостатки серверной виртуализации на базе KVM:***

1. Для изменения ресурсов необходима **перезагрузка сервера**, а увеличение дискового пространства осуществляется только **вручную**
2. **Сложное администрирование**, так как большая часть настроек осуществляется вручную, в том числе и добавление **IP**-адресов.
3. Сложность использования для клиентов без опыта.

***46. Определите основные компоненты СХД и их характеристики.***

**СХД** (Система Хранения Данных— это физический сервер (или часть пространства сервера), который предназначен для хранения и обработки информации.

***Основные компоненты СХД***

1. **Хранилище данных (дисковый массив)** - это физический набор жестких дисков, объединенных в единую структуру хранения данных. Хранилище может иметь различные типы и размеры дисков, а также обеспечивать разные уровни доступности и производительности.

2. **Контроллер** - это устройство, которое управляет доступом к хранилищу данных и обеспечивает его работу в соответствии с настройками и требованиями пользователя. Он может выполнять функции кэширования, шифрования, сжатия данных и т.д.

3**. Интерфейсные карты** - это устройства, которые обеспечивают связь между хранилищем данных и компьютерами или сетью. Интерфейсы могут быть различными, например, Fibre Channel, iSCSI, SAS, SATA и т.д.

4. **Элементы питания (Power Supplies)** – это устройства, которые обеспечивают питание СХД и его компонентов. В большей части системы питания резервируются, так что в случае отказа одного элемента, работа системы не прерывается.

5. **Программное обеспечение (программное обеспечение системы хранения данных)** - это набор программных модулей, обеспечивающих управление, мониторинг и анализ состояния системы хранения данных. Это могут быть операционные системы, драйверы устройств, программы для резервного копирования и восстановления данных и т.д.

***Характеристики:***

1. **Производительность** - это скорость, с которой система хранения данных может обрабатывать данные. Она зависит от типа дисков, их скорости вращения, объема кэша, а также от качества контроллера.

2. **Ёмкость** - это объем данных, который может быть хранен на системе. Емкость зависит от количества дисков и их объема.

3. **Доступность** - это надежность работы системы и ее способность продолжать работу даже при отказе отдельных компонентов. Доступность может быть обеспечена за счет внедрения резервирования или зеркалирования данных.

4. **Масштабируемость** - это возможность расширения системы хранения данных по мере увеличения объема данных. Масштабирование может происходить путем добавления новых дисков, контроллеров, интерфейсных карт и т.д.

5. **Стоимость** - это стоимость приобретения и эксплуатации системы хранения данных. Стоимость зависит от многих факторов, например, от производителя, конфигурации, типа интерфейса и т.д.

***53. Перечислите основные киберугрозы и методы борьбы с ними для организации.***

**Основные киберугрозы для организации:**

1. Вредоносные программы - они могут сбросить, украсть или уничтожить данные, а также привести к отказу системы.

2. Фишинговые атаки - киберпреступники могут отправить электронную почту, представляющуюся легитимной, чтобы получить доступ к учетным записям или создать сайт-двойник для получения конфиденциальной информации и её незаконного использования.

3. DDoS-атаки - одновременная хакерская атака с большого числа компьютеров с целью доведения до отказа в работе

4. Атаки с использованием социальной инженерии - криминалы пытаются убедить пользователя раскрыть свои личные данные или кликнуть на злонамеренную ссылку.

5. Внутренние угрозы - уязвимость защиты от доступа изнутри может привести к утечке данных, краже, коррупции или другим формам мошенничества.

**Методы борьбы с киберугрозами для организации**:

1. Применение многофакторной аутентификации - использование многих факторов, таких как пароль и смарт-карта, для проверки личности пользователя.

2. Шифрование данных и сетевого трафика - данные могут быть зашифрованы, что делает их более безопасными.

3. Установка системы мониторинга угроз - системы мониторинга потоков сетевого трафика, поведения пользователей и других аномальных событий, могут помочь идентифицировать и предотвратить угрозы.

4. Своевременная установка обновлений (обновление программ и оборудования).

5. Обучение сотрудников - организации могут проводить регулярные навыки беседы на тему "кибербезопасности" и обучить сотрудников как реагировать на угрозы информационной безопасности.

6. Резервное копирование данных (копирование части или всей базы данных на определенный момент времени).

# **12,14,21,28,35,42,49**

**№12. Чем контейнеризация отличается от виртуализации?**

Контейнеризация — метод, с помощью которого программный код упаковывается в единый исполняемый файл вместе с библиотеками и зависимостями, чтобы обеспечить его корректный запуск. Такие файлы называют контейнерами. Контейнеры можно разворачивать в разных средах и там управлять их работой.

Виртуализация — это построение вычислительной среды, в которой на базе одних и тех же аппаратных ресурсов работает множество изолированных друг от друга виртуальных машин. Другими словами, это создание программных версий различных физических объектов: компьютеров, хранилищ данных, сетей, серверов и приложений.

Основное отличие контейнеризации от виртуализации заключается в том, что контейнер, это изолированная среда, использующая ядро хостовой машины для обработки данных, без ограничения процессорного времени и памяти. В то время как, у виртуальной машины полностью изолированы и системное ядро, и программные модули, и библиотеки. А по процессорному времени и памяти предоставляются строго ограниченные ресурсы.

**№14. Какую роль выполняет виртуализация сети в возможностях облачного провайдера.**

Главное преимущество виртуализации — это эффективное сокращение расходов на поддержку ИТ-инфраструктуры за счет экономии физических ресурсов, повышения уровня автоматизации процессов, адаптивности и масштабируемости бизнеса. Виртуализация помогает исключить расходы на покупку дополнительных серверов и их обслуживание, при этом максимизировать пользу имеющихся ресурсов.

**№21. Расскажите, за счёт каких облачных сервисов может быть значительно ускорен доступ к данным?**

**Google Drive / Google Doc**

**Microsoft One Drive**

**Amazon Web Services**

Эти сервисы предоставляют возможность хранения данных в облаке, что позволяет быстро получать доступ к данным из любого места и устройства с доступом в Интернет. Кроме того, облачные сервисы обычно имеют высокую производительность и масштабируемость, что позволяет обрабатывать большие объемы данных и обеспечивать быстрый доступ к ним.

*В этом билете точно расскажите про доступ к документам и данным из любой точки, где есть интернет и про возможность совместной работы в режиме онлайн с файлом.*

**№28. Какую роль выполняет сервисный портал в референсной архитектуре облачного решения?**

Референсная архитектура — набор рекомендаций того или иного вендора (какое оборудование взять, какое ПО на него поставить и т. д.). Этот подход менее гибок, но при соблюдении рекомендаций серьезно возрастает уверенность в том, что не будет узких мест и критических точек отказа.

Сервисный портал в референсной архитектуре облачного решения выполняет роль центра управления и мониторинга облачных сервисов. Он предоставляет пользователю удобный интерфейс для управления и мониторинга облачных сервисов, а также инструменты для мониторинга производительности и доступности сервисов. Сервисный портал также может предоставлять инструменты для управления доступом к облачным сервисам и контроля расходов на использование облачных ресурсов.

**№35. Опишите основные инфраструктурные компоненты для технологического слоя центра обработки данных.**

Технологический слой центра обработки данных включает в себя несколько основных инфраструктурных компонентов:

1. Серверы - физические или виртуальные машины, на которых размещаются приложения и сервисы, обрабатывающие данные.

2. Хранилища данных - системы, предназначенные для хранения и управления данными, которые используются приложениями и сервисами. В зависимости от требований к производительности и масштабируемости, хранилища могут быть реляционными или нереляционными.

3. Сетевая инфраструктура - сетевые коммутаторы и маршрутизаторы, обеспечивающие связность между серверами и хранилищами данных.

4. Кластеризация - технология, позволяющая объединять несколько серверов в кластер для обеспечения высокой доступности и масштабируемости приложений и сервисов.

5. Виртуализация - технология, позволяющая создавать виртуальные машины на физических серверах для оптимизации использования ресурсов и управления приложениями и сервисами.

6. Системы мониторинга и управления - инструменты для мониторинга производительности и доступности приложений и сервисов, а также для управления ресурсами и конфигурацией инфраструктуры.

**№42. Раскройте свойства виртуальной машины и контейнера. Отметьте преимущества и недостатки технологии контейнеризации.**

***Основные свойства виртуальной машины:***

- Изоляция - каждая VM работает в отдельном виртуальном окружении, что обеспечивает высокую степень изоляции и безопасности.

- Портативность - VM можно легко перемещать между физическими серверами и даже между облачными платформами.

- Гибкость - VM можно настраивать под различные требования, устанавливать нужное ПО и настраивать параметры аппаратной части.

***Основные свойства контейнера:***

- Легковесность - контейнеры используют общую операционную систему с хост-системой, что делает их более легковесными по сравнению с VM.

- Быстродействие - запуск и остановка контейнеров происходит быстрее, чем у VM.

- Портативность - контейнеры можно легко перемещать между физическими серверами и облачными платформами.

***Преимущества контейнеризации:***

- Легковесность и быстродействие.

- Более эффективное использование ресурсов сервера.

- Упрощение развертывания и масштабирования приложений.

- Повышение портативности и гибкости.

***Недостатки контейнеризации:***

- Ограниченные возможности изоляции и безопасности.

- Ограниченные возможности настройки аппаратной части контейнера.

- Необходимость использования общей операционной системы с хост-системой, что может привести к конфликтам и ошибкам.

**№49. Каким образом в СХД осуществляется логическая и физическая защита данных.**

Логическая защита данных в СХД осуществляется путем установки прав доступа к данным на уровне пользователей и групп пользователей. Это позволяет ограничить доступ к конфиденциальным данным только тем пользователям, которые имеют на это право. Кроме того, в СХД могут быть реализованы механизмы шифрования данных, что обеспечивает дополнительную защиту от несанкционированного доступа.

Физическая защита данных в СХД включает в себя использование различных мер защиты от физических угроз, таких как пожар, наводнение, кража и т. д. Для этого могут применяться следующие меры:

- Резервирование серверов и хранилищ данных для обеспечения отказоустойчивости и защиты от сбоев в работе оборудования.

- Размещение серверов и хранилищ данных в специально оборудованных помещениях с контролем температуры, влажности и других параметров окружающей среды.

- Использование систем контроля доступа и видеонаблюдения для защиты от кражи и несанкционированного доступа к оборудованию.

- Резервное копирование данных на внешние носители для защиты от потери данных в случае аварийной ситуации.

# **2, 9,19,22,29,36,43,50**

**№2. Чем монолитные приложения отличаются от микросервисных?**

**Монолитное приложение** представляет собой приложение, которое доставляется через единое развертывание (в виде файла WAR или приложения Node с одной точкой входа). Например, при создании интернет-магазина все отдельные модули (UI, бизнес-логика, дата-слой) реализуются внутри одного модуля в рамках одного кода. Такие приложения легки в развертывании, но их тяжело развивать и менять, если один компонент не работает, то всё приложение может остановиться.

В **микросервисной архитектуре** слабо связанные сервисы взаимодействуют друг с другом для выполнения задач, относящихся к их бизнес-возможностям. В данном случае каждый модуль нужно развертывать самостоятельно, например, в том же интернет-магазине модулям необходим свой сервис и база данных. Плюсы такого решения: их легче масштабировать, тестировать, микросервисы не зависят от технологии, которая используется в других сервисах – можно использовать лучшие технологии подгонки, разные языки программирования, отказоустойчивость выше. Соответственно, основной недостаток – проблемная реализация взаимодействия между сервисами.

**№9. На какие вопросы необходимо знать ответы организации при переходе от традиционной инфраструктуры к облачной. Приведите примеры ответов.**

1. Какие функции и приложения в настоящее время используются в организации, и как они могут быть перенесены в облако?

Пример ответа: Мы используем CRM-систему и электронную почту, которые могут быть перенесены в облако с помощью поставщика облачных услуг.

2. Какие изменения необходимо внести в сетевую инфраструктуру организации для подключения к облачным сервисам?

Пример ответа: Мы должны обновить нашу сетевую инфраструктуру и установить защищенное VPN-соединение для связи с облачным провайдером.

3. Какие меры безопасности будут приняты для защиты данных организации в облаке?

Пример ответа: Мы будем использовать многофакторную аутентификацию и шифрование данных, чтобы гарантировать безопасность наших данных в облаке.

4. Какие будут затраты на перенос инфраструктуры и приложений в облако?

Пример ответа: Мы ожидаем, что затраты на перенос инфраструктуры и приложений в облако составят около $50 000.

5. Какие преимущества и недостатки будут у нас при использовании облачных сервисов?

Пример ответа: Мы получим доступ к новым функциям и возможностям, однако, мы также будем зависеть от надежности и доступности облачных сервисов.

**№19. Чем FaaS отличается от PaaS или SaaS?**

FaaS (Function-as-a-Service), PaaS (Platform-as-a-Service), и SaaS (Software-as-a-Service) — это все модели облачных вычислений, но имеют различия в том, как они предоставляются и какие функции они обеспечивают.

- SaaS (Software-as-a-Service) — это модель облачных вычислений, в которой приложения и сервисы предоставляются конечным пользователям через интернет. Пользователи не управляют инфраструктурой, на которой работает приложение, но могут настраивать приложение в соответствии со своими потребностями.

- PaaS (Platform-as-a-Service) — это модель облачных вычислений, которая предоставляет пользователям платформу для создания, запуска и управления приложениями. Пользователи могут создавать собственные приложения, используя инструменты, предоставляемые PaaS-провайдером.

- FaaS (Function-as-a-Service) — это модель облачных вычислений, которая позволяет пользователям создавать и запускать функции приложений без необходимости управления инфраструктурой. FaaS-провайдер предоставляет инфраструктуру и автоматически масштабирует ресурсы, чтобы обеспечить выполнение функций.

Таким образом, основное отличие между FaaS, PaaS и SaaS заключается в том, что SaaS — это готовые приложения, PaaS — это платформа для создания и запуска приложений, а FaaS - это функции приложений, которые могут быть запущены в облачной среде без управления инфраструктурой.

**№22. Опишите функциональные задачи ПО Kubernetes и архитектуру решения.**

Kubernetes — это инструмент для автоматизации развертывания, масштабирования и управления контейнерами. Он является открытым исходным кодом и разработан компанией Google. Основными функциональными задачами ПО Kubernetes являются:

1. Оркестрация контейнеров: Kubernetes автоматически развертывает и управляет контейнерами в кластере.

2. Масштабирование приложений: Kubernetes позволяет масштабировать приложения горизонтально и вертикально.

3. Управление конфигурацией: Kubernetes позволяет управлять конфигурацией приложений и микросервисов.

4. Слежение и мониторинг: Kubernetes предоставляет функции мониторинга и слежения за работой приложений, микросервисов и контейнеров.

Архитектура:

Изображение выглядит как текст, снимок экрана, диаграмма, План

Автоматически созданное описание

**№29. Какими параметрами должен обладать провайдер инфраструктурных услуг для реализации государственных ИТ-проектов и размещения сервисов?**

1. Надежность и безопасность: Провайдер должен иметь высокий уровень безопасности, чтобы предотвратить утечку данных и обеспечить надежность облачной инфраструктуры.

2. Соответствие законодательству: Провайдер должен соответствовать требованиям законодательства и обеспечивать защиту государственных данных.

3. Доступность и отказоустойчивость: Провайдер должен обеспечить высокую доступность и отказоустойчивость облачной инфраструктуры, чтобы обеспечить непрерывность работы государственных сервисов.

4. Высокая производительность: Провайдер должен обеспечивать высокую производительность облачной инфраструктуры, чтобы обеспечить быстрый доступ к государственным сервисам.

5. Поддержка: Провайдер должен предоставлять высококачественную поддержку для государственных ИТ-проектов, включая техническую поддержку, консультации и помощь в развертывании и настройке сервисов.

6. Цена: Провайдер должен предлагать конкурентоспособную цену для государственных ИТ-проектов и сервисов, чтобы обеспечить экономическую эффективность использования облачной инфраструктуры.

Кроме того, провайдер должен иметь опыт работы с государственными организациями и предоставлять дополнительные сервисы, такие как мониторинг, аналитику и отчетность, чтобы обеспечить эффективное управление государственными ИТ-проектами и сервисами.

**№36. Каким образом виртуализация помогает масштабировать ресурсы центра обработки данных.**

1. Консолидация ресурсов: виртуализация позволяет объединить несколько физических серверов в единую систему виртуализации. Это позволяет эффективно использовать имеющиеся ресурсы и уменьшить количество серверов, что экономит место и энергию.

2. Разделение ресурсов: позволяет разделить ресурсы на виртуальные машины, каждая из которых может работать под управлением своей операционной системы. Это позволяет эффективно использовать ресурсы и управлять ими, давая пользователям возможность управлять своими рабочими нагрузками.

3. Динамическое распределение ресурсов: позволяет динамически перераспределять ресурсы между виртуальными машинами в зависимости от их текущих нужд и нагрузок на систему. Это позволяет эффективно использовать имеющиеся ресурсы и обеспечивать оптимальную производительность системы в целом.

4. Упрощение управления ресурсами: упрощает управление ресурсами, так как администраторы могут управлять всеми виртуальными машинами на одном физическом сервере, используя единую панель управления. Это уменьшает время и сложность управления ресурсами и помогает сократить время настройки и управления системой в целом.

5. Изоляция приложений: позволяет обеспечить изоляцию разных приложений и рабочих нагрузок на одном физическом сервере, что повышает безопасность и стабильность работы системы в целом.

**№43. Расскажите об особенностях процессоров МЦСТ «Эльбрус 2000». Какую роль архитектура играет в кибербезопасности?**

Процессоры МЦСТ «Эльбрус 2000» — это микропроцессоры с открытой архитектурой, разработанные в России. Они используются в высокопроизводительных вычислительных системах, в том числе в системах безопасности и защиты информации.

Основная особенность архитектуры процессоров Эльбрус 2000 - это использование параллельной архитектуры с большим числом ядер. Кроме того, они также используются специальные алгоритмы предварительного вычисления, которые эффективно ускоряют работу с данными. Это позволяет эффективно использовать их для решения сложных вычислительных задач, таких как обработка больших объемов данных или выполнение сложных алгоритмов.

Архитектура Эльбрус 2000 также имеет большое значение для кибербезопасности, так как она обеспечивает защиту от атак с использованием переполнения буфера памяти и других типов атак. Это достигается благодаря использованию специальных механизмов, которые контролируют доступ к памяти и могут обнаружить попытку злоумышленника изменить системные ресурсы.

Таким образом, архитектура процессоров Эльбрус 2000 предоставляет высокую производительность и эффективность вычислений, а также обеспечивает защиту от некоторых типов кибератак. Это делает их полезными и важными для использования в различных высокотехнологичных системах, в том числе в системах безопасности и защиты информации.

**№50. Опишите основные типы носителей данных. В чём их преимущества и недостатки?**

Основные типы носителей данных:

1. Жесткие диски – это основной тип носителя данных в компьютерах и серверах. Хранят большое количество информации, имеют высокую скорость чтения и записи, но могут выходить из строя.

2. Флеш-накопители – компактные и легкие, легко переносимы, но обычно имеют меньший объем хранения, чем жесткие диски.

3. CD и DVD диски – хранят информацию на оптическом носителе, обычно имеют меньший объем хранения, чем жесткие диски. Могут быть повреждены легче, чем жесткие диски.

4. Облачные сервисы – позволяют хранить информацию на удаленных серверах. Обеспечивают высокий уровень безопасности данных, но требуют постоянного доступа к Интернету.

Преимущества и недостатки:

1. Жесткие диски обеспечивают высокую скорость чтения и записи, большой объем хранения, но могут выходить из строя.

2. Флеш-накопители легко переносимы, компактные, но обычно имеют меньший объем хранения, чем жесткие диски.

3. CD и DVD диски имеют низкую стоимость, но могут быть повреждены легче, чем жесткие диски.

4. Облачные сервисы обеспечивают высокий уровень безопасности данных, но требуют постоянного доступа к Интернету и могут быть дороже, чем другие типы носителей данных.

**1 Объясните термин «облачные вычисления» и приведите примеры сервисов?**

Облачные вычисления — это технология, которая обеспечивает доступ к компьютерным ресурсам через интернет.

Облачные вычисления (англ. cloud computing) — технология распределённой обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю как Интернет-сервис.

Предоставление пользователю услуг как Интернет-сервис является ключевым. Однако под Интернет-сервисом не стоит понимать доступ к сервису только через Интернет, он может осуществляться также и через обычную локальную сеть с использованием веб-технологий.

СЕРВИСЫ: Gmail, Yahoo, Яндекс.Почту; социальные сети, такие как Facebook, «ВКонтакте»; различные сервисы хранения данных, такие как Dropbox, Яндекс.Диск.

**5,1,16,23,30,37,44, 51**

**5 Опишите известные вам типы облачных решений.**

основные типы облачных вычислений:

* Инфраструктура как услуга (IaaS)
* Платформа как услуга (PaaS)
* Программное обеспечение как услуга (SaaS)
* Все как услуга (XaaS)

Существуют три основные модели услуг облачных вычислений – Инфраструктура как услуга, Платформа как услуга и программное обеспечение как услуга.

* Инфраструктура как услуга (IaaS)

Инфраструктура как услуга (IaaS) - это модель обслуживания, которая формирует основу для развертывания вашей облачной технологии. С помощью поставщика IaaS вы получаете доступ по требованию через Интернет к основным ИТ-ресурсам, включая компьютеры (виртуальное или выделенное оборудование), сети и хранилища.

* Платформа как услуга (PaaS)

Платформа как услуга (PaaS) - это модель облачного сервиса, при которой вы получаете доступ к комбинированным аппаратным и программным средствам через поставщика услуг. PaaS чаще всего используется для разработки приложений.

* Программное обеспечение как услуга (SaaS)

Программное обеспечение как услуга (SaaS) - это модель облачного сервиса, которая предоставляет вам доступ к готовому программному продукту, запускаемому поставщиком услуг и управляемому им. Большинство решений SaaS, как правило, являются приложениями конечного пользователя.

**16. Дайте характеристики основным продуктам для оркестрации контейнеров.**

Оркестраторы контейнеров бывают разные. Одни довольно просты в управлении, но имеют ограниченный набор функций.

- Kubernetes

Эта портативная платформа с открытым исходным кодом считается отраслевым стандартом. Поддерживает и декларативную конфигурацию объектов, и автоматизацию. Можно автоматизировать масштабирование, развертывание с помощью шаблонов и управление рабочей нагрузкой и сервисами контейнеров. Поддерживает ОС: Windows, macOS, Linux.

Предлагает: мониторинг сервисов и распределение нагрузки, автомонтирование гибкой системы хранения, автоматизированные откаты и развертывания, автораспределение нагрузки

- OpenShift Container Platform

Платформа корпоративного уровня в формате PaaS для разработки, развертывания и управления не только контейнерными, но и классическими приложениями и их компонентами в режиме самообслуживания. Позволяет автоматизировать их в физических, виртуальных и общедоступных, частных и гибридных облачных средах. Поддержка ОС: Linux.

Предлагает: встроенные средства планирования, кластеризации и оркестрации; автомасштабирование; балансировку нагрузки

- Docker Swarm

Это «родная», базовая система кластеризации и оркестровки контейнеров Docker. Использует декларативную модель. Взаимодействие с кластером происходит через команды Docker, а управление действиями — через менеджер Swarm. Поддержка ОС: Windows, macOS, Linux.

Предлагает: децентрализованное проектирование, масштабирование, балансировку нагрузки, динамическое управление ролями, постепенное обновление, разные режимы работы сетей (bridge, overlay, host, none, macvlan)

- Nomad

Легкий в поддержке, гибкий, расширяемый оркестратор для развертывания и управления контейнерными и классическими приложениями в физической или облачной среде. Поддерживает ОС: Linux, Windows, macOS, а также Docker, Java, виртуальные машины. В основном Nomad управляет деплоями и перезапускает контейнеры при обнаружении ошибок. Этого часто оказывается достаточно для небольших независимых проектов.

Предлагает: простое управление, хорошую экосистему — встроенную интеграцию с другими продуктами (например, Consul, Terraform, Vault), автоматизацию переноса приложения из монолитной в микросервисную архитектуру

**23 Опишите функциональные задачи ПО OpenShift. В чём разница с Red Hat OpenShift Container Platform?**

Одной из самых востребованных на рынке универсальных платформ оркестрации контейнеров на основе Kubernetes является OpenShift от компании Red Hat. Red Hat присоединилась к разработке и быстро стала вторым по вкладу в развитие Kubernetes контрибьютором в сообществе.

Базовая версия Red Hat OpenShift в сочетании с многочисленными дополнениями закрывает все необходимые нужды компании, занимающейся контейнеризацией своих приложений. Здесь есть и централизованное управление, и глобальный регистр, и возможность создавать Windows-контейнеры, и многое другое.

OpenShift с равным успехом может работать на большинстве современных платформ:

· Облака Amazon Web Services, Microsoft Azure и Google Cloud

· Платформы виртуализации VMware vSphere, Red Hat Virtualization и Red Hat OpenStack

В основе OpenShift лежат «операторы» – функциональные сущности, содержащие в себе приложение, систему управления им, а также средства проверки, анализа и мониторинга. Это позволяет устранить необходимость выполнения большого количества рутинных задач. По сути, вся платформа OpenShift состоит из таких операторов.

Red Hat OpenShift преимущества:

* сократились сроки ввода новых продуктов в продакшн;
* заметно сократились затраты на разработку и тестирование.
* Появилась возможность управлять изменениями в режиме, что называется, реального времени;
* появился контроль работы приложений на уровне кода, благодаря чему служба ИБ теперь может осуществлять анализ и контроль рисков;
* у пользователей появилась возможность быстро влиять на функциональность и получать максимально гибкие и настроенные под их задачи продукты.

**30 Опишите один из методов интеграции микросервисных приложений. (????)**

Circuit Breaker — паттерн интеграции между сервисами с гарантией ответа на запрос. Предназначен для интеграции с внешними системами, ответ от которых может завершиться ошибкой и не является критичным для приложения. Основные характеристики: конечный автомат; имеет три состояния; работает в режиме прокси; гарантирует ответ на запрос; для каждого состояния своя метрика перехода в другое состояние; может кэшировать запросы.

Как работает: между Application и удаленным сервисом мы ставим Circuit Breaker. У Circuit Breaker есть три состояния: открытое, закрытое и полуоткрытое. Он всегда инициализируется в закрытом состоянии — первый запуск закрытый. Мы делаем вызов через Circuit Breaker. Circuit Breaker отправляет запрос в удаленный сервис, но перед этим сбрасывает счетчик — значок «умножить на ноль». Мы делаем вызов удаленного сервиса, получаем от него ответ и передаём ответ Application. При необходимости в промежутке кэшируем.

**37 Дайте характеристику основным свойствам облачного решения.**

Облачные решения — это технология распределенной обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю как интернет-сервис.

Это ресурс, включающий в себя инструменты и приложения, такие как хранилище данных, серверы, базы данных, сети, программное обеспечение. Облачные решения так названы потому, что информация, к которой осуществляется доступ, находится удаленно, в “облаке”, виртуальном пространстве. Пользователь получает доступ ко всем данным через Интернет: никакой привязки к месту, комфортная работа из любой точки мира , где есть возможность подключения к сети.

Преимущества:

* Экономия бюджета - Поставщики таких услуг используют модель оплаты по мере использования, поэтому компании, предлагающие облачные услуги, никогда не платят больше, чем они используют.
* Отсутствие территориальной привязки - Для пользователей облачные сервисы означают, что они могут получить доступ ко всему: от персональных файлов, до любых приложений и многому другому с любого устройства, имеющего подключение к Интернету.

**44 Каким образом производители процессоров переходят на новый технологический процесс? Раскройте физико-химические ограничения микроэлектроники на полупроводниках.**

Самые современные процессоры (из тех, что поступили в промышленное производство) сейчас имеют 7-нанометровый (7-нм) техпроцесс. Такими технологиями на данный момент достаточно хорошо овладела тайваньская компания TSMC, которая производит чипсеты по заказу крупнейших мировых производителей, таких, как Apple, Huawei и Qualcomm.

Процесс формирования микросхемы состоит из большого числа различных по своей физико-химической природе операций. Все операции необходимы, чтобы обеспечить создание прецизионных по составу и геометрическим размерам областей в кремнии, на основе которых формируются отдельные элементы, а также диэлектрических и металлических пленок, используемых для объединения отдельных элементов в микросхему. Все операции технологического процесса сильно связаны между собой.

Основными факторами, влияющими на результаты производственного процесса (выход годных микросхем, уровень их рабочих характеристик) являются суммарная плотность дефектов, приходящихся на единицу поверхности пластины (подложки) в процессе проведения всех технологических операций и количество фотолитографических операций (как самых ответственных).

Получение чистой поверхности полупроводниковых подложек связано с требованием недопустимости каких-либо загрязнений поверхности пластин, так как электрические параметры микросхем существенно зависят от степени совершенства поверхности, качества обработки и геометрической формы обработанных пластин и эти несовершенства неблагоприятно сказываются на последующих технологических процессах, создающих в подложке топологию будущей микросхемы. Несовершенства кристаллической решетки приповерхностного слоя пластины, образовавшегося после шлифовки и полировки с использованием абразивных микропорошков, все еще сохраняются (с близкой к идеальной поверхностью), поскольку все предыдущие технологические операции основываются на механической обработке поверхности пластин. Удаление с поверхности подложки остаточного механически нарушенного слоя необходимо для получения атомарно совершенной структуры поверхностного слоя, поэтому следующим технологическим процессом является химическая обработка пластин.

**51 Какие принципы управления данными в организации вы знаете?**

Управление данными (УД) — это организация их сбора, хранения, обработки, использования и архивации. Главная задача этого направления менеджмента — создать такую систему работы с данными, при которой у сотрудников компании и её партнёров есть оперативный доступ ко всем необходимым данным и возможность продуктивно их обрабатывать. Система УД должна быть эффективной, безопасной и экономичной.

Большая часть деятельности практически любой компании проходит онлайн — внутренняя документация, расчёты, общение с потенциальными и текущими клиентами. Оцифрованные бухгалтерские и складские документы, клиентские базы, показатели продаж — всё это данные.

Система УД нужна еще и потому, что у цифровых данных есть жизненный цикл. В зависимости от специфики сведений и их формата, этот цикл может включать разные стадии. Вот самые важные из них:

* создание или получение сведений;
* административные процессы: перемещение, преобразование, обеспечение доступа, хранение;
* работа над качеством: актуализация, изменение механизмов обработки, классификация;
* использование данных: оно может быть прямым (продажа самих данных или продуктов, построенных на них — прогнозных моделей, маркетинговых исследований, статистических выкладок) или косвенным — то есть, применением данных для принятия решений и других внутренних процессов;
* архивирование и ликвидация.

Эффективный менеджмент корпоративных данных строится на следующих принципах:

- Ценность. У данных есть уникальные свойства, особенности и определённая стоимость. И ценность можно и нужно оценивать в экономических терминах. Восприятие массива данных как актива помогает более осознанно оперировать информационными ресурсами.

- Профессионализм. Стоит помнить, что УД — это кросс-функциональный процесс, для которого требуются специалисты с разными знаниями и навыками. Один отдел вряд ли справится со всем менеджментом данных в компании: система должна включать профессионалов разных профилей и с широким кругом полномочий. Распространенная ошибка — оставлять стратегические потребности компании в данных на усмотрение ИТ-подразделений. Управление требует целостного взгляда на функционирование предприятия, поэтому значимые решения должны учитывать мнения служб продаж, маркетинга и руководства организации. ИТ-отдел обслуживает такие решения, а не принимает их.

- Достаточное количество и качество. Данные, на основании которых принимаются стратегические решения, в первую очередь должны быть достоверными. Кроме того, для получения репрезентативного результата необходимо обработать достаточное количество сведений (в том числе — метаданных).

- Непрерывность. УД — это управление жизненным циклом, оно нуждается в планировании, непрерывном контроле и постоянных улучшениях. При работе с данными важно учитывать вектор и стадии datachain, эволюцию информационных ресурсов, требования безопасности для каждого этапа на пути движения информации.

**Дайте характеристики и бизнес-выгоды популярным сервисам, доступным у российских облачных провайдеров**

1. Яндекс.Облако:

Характеристики - масштабируемость, высокая производительность, гибкость настройки, автоматическое масштабирование, поддержка различных операционных систем и программных продуктов.

Бизнес-выгоды - снижение затрат на инфраструктуру, ускорение процесса развертывания приложений, повышение доступности и надежности сервисов.

2. Selectel Cloud:

Характеристики - быстрое создание виртуальных машин, масштабирование ресурсов, высокая производительность, защита от DDoS-атак, гибкая настройка.

Бизнес-выгоды - снижение затрат на инфраструктуру, быстрое развертывание новых приложений, повышение доступности и надежности сервисов.

3. Cloud4Y:

Характеристики - масштабируемость, высокая производительность, автоматическое масштабирование, гибкость настройки, защита от DDoS-атак.

Бизнес-выгоды - снижение затрат на инфраструктуру, быстрое развертывание новых приложений, повышение доступности и надежности сервисов, защита от кибератак.

**Какие механизмы виртуализации применимы для программно-определяемых центров обработки данных?**

1. Виртуализация на уровне серверов - это механизм виртуализации, который позволяет запускать несколько виртуальных машин на одном физическом сервере. Это позволяет эффективно использовать ресурсы и управлять машинами.

2. Виртуализация на уровне хранения - это механизм виртуализации, который предоставляет централизованное хранилище данных для нескольких серверов. Это позволяет обеспечить высокую отказоустойчивость и управляемость данных.

3. Виртуализация на уровне сети - это механизм виртуализации, который обеспечивает гибкость и управляемость сетевой инфраструктуры. Он позволяет выделить виртуальные сетевые ресурсы и управлять ими централизованно.

4. Виртуализация на уровне приложений - это механизм виртуализации, который позволяет запускать приложения в контейнерах вместо виртуальных машин. Контейнеры обеспечивают изоляцию приложений и позволяют более эффективно использовать ресурсы.

5. Виртуализация на уровне рабочего стола - это механизм виртуализации, который позволяет запускать рабочие столы виртуально в централизованном режиме. Это позволяет упростить управление рабочими столами и повысить безопасность данных.

6. Виртуализация на уровне сервиса - это механизм виртуализации, который позволяет предоставлять услуги виртуально вместо физических ресурсов. Это позволяет упростить управление и оптимизировать использование ресурсов.

**Опишите задачи механизма Serverless. Приведите примеры локализованных услуг**

Механизм Serverless представляет собой платформу для разработки и запуска приложений, основанную на облачных сервисах. Задачи, решаемые механизмом Serverless, включают:

1. Обработка событий. С помощью Serverless можно автоматически обрабатывать события, например, обновление базы данных, приход новой электронной почты или подписка на RSS-канал.

2. Организация вычислительной инфраструктуры. Serverless предоставляет возможность создания различных сервисов, где каждый компонент отвечает за свою часть функциональности.

3. Анализ больших данных. Serverless может использоваться для анализа больших объемов данных, когда необходимо обработать их на больших кластерах с использованием нескольких ядер.

4. Создание мобильных приложений. Serverless может быть использован для создания и обслуживания мобильных приложений, используя функциональность хранения данных и обработки.

Примеры локализованных услуг, которые можно представить в виде Serverless-сервисов, включают:

1. Геолокационные сервисы. Различные приложения, использующие геолокацию могут быть развернуты на Serverless-платформе, чтобы ускорить и облегчить обслуживание пользователей.

2. Микросервисы. Serverless позволяет создавать микросервисы, используемые для обработки запросов различных систем. Это может быть полезно, когда требуется взаимодействие с другими системами и необходима быстрая реакция на запрос.

3. Видеостриминг. Serverless может быть использован для обработки потокового видео на локальном уровне без помощи облачных сервисов, что обеспечивает более быструю обработку и сокращает затраты на обработку.

4. Краудсорсинг. Serverless может использоваться для создания краудсорсинговых приложений, развернутых на локальном уровне. Например, пользователям, которые участвуют в краудсорсинговых проектах, может потребоваться быстрый доступ к обработке данных

**Дайте характеристики основным гипервизорам для серверной виртуализации.**

Основными гипервизорами для серверной виртуализации являются VMware, Hyper-V и KVM.

VMware - это один из наиболее популярных гипервизоров, который обеспечивает высокую производительность и надежность. Он поддерживает множество операционных систем, включая Windows, Linux и Unix, и позволяет создавать виртуальные машины с различными конфигурациями.

Hyper-V - это гипервизор от Microsoft, который поставляется вместе с операционной системой Windows Server. Он обеспечивает высокую производительность и поддерживает множество функций, таких как миграция виртуальных машин и горячее добавление ресурсов.

KVM - это гипервизор с открытым исходным кодом, который работает на базе Linux. Он обеспечивает высокую производительность и масштабируемость, а также поддерживает множество операционных систем, включая Windows, Linux и Unix. KVM также позволяет использовать различные типы хранилищ данных и сетевых протоколов.

**Кратко опишите основные инструменты DevOps.**

Основные инструменты DevOps включают в себя системы контроля версий, автоматизацию сборки и развертывания, инструменты для тестирования и мониторинга приложений, а также инструменты для управления конфигурацией и оркестрации контейнеров. Эти инструменты помогают автоматизировать и ускорить процессы разработки, тестирования и развертывания приложений, а также обеспечивают высокую доступность и надежность сервисов. Некоторые из наиболее популярных инструментов DevOps включают в себя Git, Jenkins, Docker, Kubernetes, Ansible и Prometheus.

**Какие типы серверных решений могут быть использованы в ЦОДе? За счёт каких технологий производители повышают энергоэффективность**

В ЦОДе могут использоваться различные типы серверных решений, включая вычислительные кластеры, гиперконвергентные инфраструктуры (HCI), серверы с высокой плотностью размещения, серверы с высокой производительностью и т.д.

Производители повышают энергоэффективность серверных решений за счет использования технологий, таких как:

1. Консолидация серверов - объединение нескольких физических серверов в одном виртуальном сервере позволяет сократить количество серверов, что снижает потребление электроэнергии.

2. Технологии управления энергопотреблением - производители используют различные технологии, такие как управление частотой процессора, управление питанием и т.д., чтобы снизить энергопотребление серверов.

3. Использование энергоэффективного оборудования - производители создают серверное оборудование с более эффективными компонентами, такими как блоки питания, вентиляторы и т.д.

4. Оптимизация охлаждения - производители используют различные технологии для оптимизации охлаждения серверов, такие как жидкостное охлаждение, управление вентиляторами и т.д., чтобы снизить потребление электроэнергии на охлаждение.

**Расскажите о трех основных классах СХД относительно специфики сетевой части**

1. СХД с прямым подключением (Direct Attached Storage, DAS) - в таких системах устройства хранения подключаются напрямую к серверам без использования сетевых технологий. Это наиболее простой и дешевый тип СХД, но он не обеспечивает возможности расширения и гибкости, которые предоставляют другие классы СХД.

2. СХД с подключением по Fibre Channel (FC SAN) - в таких системах устройства хранения подключаются к серверам посредством высокоскоростной сети Fibre Channel. Этот класс СХД обеспечивает высокую скорость передачи данных и возможности расширения, но требует дополнительных затрат на оборудование и настройку.

3. СХД с подключением по IP-сетям (iSCSI SAN) - в таких системах устройства хранения подключаются к серверам посредством сетей IP-протоколов. Этот класс СХД является наиболее гибким и экономичным, так как использует существующую сетевую инфраструктуру, но может иметь более низкую скорость передачи данных по сравнению с FC SAN.

**Как вы понимаете термин «нейронная сеть»? Каким образом данная технология может быть использована для корпоративных заказчиков (приведите несколько кейсов).**

Нейронная сеть - это алгоритм, который имитирует работу человеческого мозга. Он состоит из нейронов, которые обрабатывают информацию и передают ее дальше по сети.

Корпоративные заказчики могут использовать нейронные сети для автоматизации различных процессов и оптимизации бизнеса. Например:

1. Прогнозирование спроса на товары и услуги - нейронные сети могут анализировать данные о продажах и других факторах, таких как погода, сезонность и т.д., чтобы предсказать будущий спрос на товары и услуги.

2. Обработка больших объемов данных - нейронные сети могут использоваться для обработки больших объемов данных, таких как финансовые отчеты, медицинские записи и т.д., чтобы выявлять скрытые закономерности и тренды.

3. Распознавание образов и голоса - нейронные сети могут использоваться для распознавания образов и голоса, что может быть полезно для автоматической идентификации клиентов и улучшения качества обслуживания.

4. Управление рисками - нейронные сети могут использоваться для анализа рисков и принятия решений в области финансов и страхования.

5. Улучшение производительности - нейронные сети могут использоваться для оптимизации производственных процессов, улучшения качества продукции и сокращения времени на производство.

**15,7,8,26,33,40,47,54**

**Вопрос №7**

**Расскажите о методах импортозамещения популярных западных облачных платформ (Microsoft Azure, Google Cloud Platform, Amazon).**

Методы импортозамещения популярных западных облачных платформ, таких как Microsoft Azure, Google Cloud Platform и Amazon, могут включать в себя следующие подходы:

1. Развитие отечественных аналогов: компании могут разрабатывать свои собственные облачные платформы, которые будут конкурировать с западными аналогами. Например, в России есть несколько отечественных облачных платформ, таких как Яндекс.Облако, Selectel, VK Cloud Solutions, SberCloud.
2. Использование гибридной модели: компании могут использовать гибридную модель, которая сочетает в себе использование облачных решений от западных поставщиков и локального хранения данных на серверах в России.
3. Развитие экосистемы: компании могут развивать экосистему вокруг своих облачных платформ, чтобы привлечь больше клиентов и увеличить свою долю на рынке. Например, Яндекс.Облако разрабатывает собственные инструменты для анализа данных и машинного обучения, которые могут быть использованы клиентами для решения своих задач.

**Вопрос №8**

**Опишите архитектурную модель современной организации. Какие архитектурные слои могут быть реализованы по модели облачного доступа.**

Современная организация может иметь архитектурную модель, включающую в себя следующие слои:

1. Инфраструктурный слой, который включает в себя аппаратную и программную инфраструктуру, такую как серверы, хранилища данных, сетевые устройства и операционные системы.
2. Слой приложений, включающий в себя программное обеспечение, которое используется для обеспечения бизнес-процессов организации. Этот слой может включать в себя такие приложения, как ERP, CRM, управление документами, управление проектами и т.д.
3. Слой данных, включающий в себя данные, которые используются в бизнес-процессах. Эти данные могут быть хранены как локально, так и в облаке.
4. Слой безопасности, который обеспечивает защиту инфраструктуры, приложений и данных от внешних угроз.
5. Слой управления, который обеспечивает управление всей архитектурой организации.

Модель облачного доступа может быть реализована в слоях 1, 2 и 3. Например, инфраструктура может быть реализована в виде облачных вычислительных и хранилищ данных. Приложения могут использовать облачные API для получения доступа к функциональности и данных облачных сервисов. Данные, в свою очередь, могут храниться в облачных хранилищах данных для обеспечения более эффективного распределенного доступа и управления.

**Вопрос №15**

**Перечислите основные характеристики ЦОДов. Как эти характеристики влияют на качество предоставляемых услуг (SLA-service level agreement)?**

Основные характеристики ЦОДов:

1. Надёжность – ЦОД должен гарантировать стабильную и безопасную работу оборудования и иных систем, не допуская отключения по причине отказов, плановых работ и т.п.
2. Масштабируемость – возможность изменения объемов хранимых данных, мощности серверной фермы и других систем под изменяющегося клиента.
3. Пропускная способность – ЦОД должен обеспечивать высокую скорость передачи данных в Интернете, быстрый доступ к хранимым данным и т.п.
4. Безопасность – высокий уровень защиты данных, контроля и доступа к информации.
5. Энергоэффективность – ЦОД должен работать с наименьшим потреблением электроэнергии, что обеспечивает экономическую эффективность.
6. Гибкость – возможность быстро реагировать на изменения требований клиентов и рынка.

Качество предоставляемых услуг напрямую зависит от всех вышеперечисленных характеристик. Плохая надёжность ЦОДа может привести к невыполнению SLA. Пропускная способность ЦОДа и энергоэффективность влияют как на быстродействие, так и на экономичность. Безопасность – это основа, на которой основывается СЛА, поскольку безопасность – это приоритетная зона в любом ЦОД. В то же время гибкость ЦОДа позволяет быстро адаптироваться к новым задачам клиента и рынка, что способствует удовлетворению его требований.

**Вопрос №26**

**Какими особенностями обладает решение по виртуализации СХД VMware VSAN?**

Решение по виртуализации СХД VMware VSAN обладает следующими особенностями:

1. Гибкость: VMware VSAN позволяет выстраивать СХД с использованием обычных серверов, что делает решение очень гибким и адаптированным к любым потребностям.
2. Высокая производительность: благодаря распределенной архитектуре, VSAN обеспечивает высокую производительность и доступность данных, что позволяет обеспечивать работоспособность даже при выходе из строя одного или нескольких узлов.
3. Простота управления: виртуализация СХД под управлением VMware VSAN происходит на уровне гипервизора, что позволяет упростить управление всей инфраструктурой.
4. Низкая стоимость: VMware VSAN позволяет существенно снизить затраты на СХД, благодаря использованию обычных серверов и отказу от дорогостоящего сетевого оборудования.
5. Расширяемость: решение по виртуализации СХД VMware VSAN позволяет легко масштабировать инфраструктуру и добавлять новые узлы для увеличения объема хранилища и производительности.

**Вопрос №33**

**В чём заключаются принципы DevOps? Раскройте расширенный термин «DevSecOps».**

DevOps - это методология, которая объединяет разработку и операции в единую команду и цепочку процессов для повышения качества продукции и скорости её доставки. Принципы DevOps включают следующие элементы:

1. Автоматизация - процессы должны быть автоматизированы для исключения ошибок, ускорения развертывания и снижения затрат.
2. Культура - в команде должны доминировать открытость, сотрудничество и участие всех звеньев в процессе.
3. Интеграция - разработка, тестирование и Релиз должны происходить непрерывно, чтобы выявлялись ошибки и баги на ранних этапах.
4. Операции - персонал, отвечающий за эксплуатацию продукта, не должен выступать в качестве противоборствующей стороны при обнаружении ошибок.

DevSecOps - это расширенный вариант DevOps, включающий в себя аспекты безопасности в процессы разработки и эксплуатации. Главная цель DevSecOps заключается не только в обеспечении чистоты и безопасности продукта, но и в реактивной и превентивной защите от угроз. В этом случае процесс безопасности должен интегрироваться во все процессы DevOps и рассматриваться как важный этап в жизненном цикле продукта. DevSecOps включает в себя аспекты безопасности, такие как автоматическое выявление и устранение уязвимостей, использование защищенных приложений и настройка доступов к базам данных.

**Вопрос №40**

**Опишите основных характеристики основного микропроцессора для ПК и сервера (Central Processing Unit). В чём будет их отличия?**

Основными характеристиками основного микропроцессора для ПК и сервера являются:

1. Частота такта - скорость, с которой процессор выполняет команды.
2. Количество ядер - число независимых исполнительных единиц внутри процессора.
3. Объем кэш-памяти - место, где процессор хранит наиболее часто используемые данные для быстрого доступа.

Отличия между микропроцессорами для ПК и сервера заключаются в:

1. Частоте такта - процессор сервера имеет более низкую частоту такта, но большую вычислительную мощность за счет большего числа ядер и кэш-памяти.
2. Количество ядер - процессоры серверов имеют более высокое количество ядер, чем процессоры ПК, так как серверы обрабатывают большее количество запросов.
3. Объем кэш-памяти - у процессоров серверов обычно больший объем кэш-памяти для обработки большого объема данных.

Также процессоры для серверов обычно имеют более высокую надежность и поддерживают дополнительные функции, такие как виртуализация и управление энергопотреблением.

**Вопрос №47**

**Укажите основные транспортные протоколы в СХД и их различия.**

Основные транспортные протоколы в СХД (системах хранения данных) включают:

1. FC (Fibre Channel) – это протокол передачи данных, используемый для связи хранилищ данных с сервером. Он обеспечивает высокую скорость передачи данных (до 128 Гбит/с) и низкую задержку.
2. iSCSI (Internet SCSI) – это транспортный протокол, который позволяет использовать сетевую инфраструктуру для управления хранилищем данных. Он работает поверх TCP/IP и позволяет использовать стандартные сетевые технологии, такие как Gigabit Ethernet, для соединения серверов с хранилищем данных.
3. NFS (Network File System) – это протокол, который позволяет пользователям монтировать удаленные файловые системы на локальной машине. Он широко используется в среде Unix и Linux.

Различия между этими протоколами заключаются в способе передачи данных, скорости, надежности и возможности масштабирования. FC является наиболее быстрым протоколом, но также является наиболее дорогим. iSCSI позволяет использовать существующее сетевое оборудование, но может быть медленнее, чем FC. NFS наоборот, предназначен для файловых систем и не обеспечивает такой быстрой передачи данных, как FC и iSCSI.

**Вопрос №54**

**Выделите основные свойства сетевого маршрутизатора. На каком уровне модели OSI происходит работа с данными?**

Основные свойства сетевого маршрутизатора:

1. Используется для передачи данных между различными сетями.
2. Обладает несколькими интерфейсами, каждый из которых соединен с определенной сетью.
3. Осуществляет выбор наиболее эффективного пути для передачи данных.
4. Позволяет управлять трафиком в сети и обеспечивать безопасность.

Работа с данными происходит на сетевом уровне модели OSI (второй уровень). Маршрутизатор получает информацию о маршрутах из таблиц маршрутизации и принимает решение о том, какой путь использовать для передачи данных между сетями. Затем он перенаправляет данные через соответствующие интерфейсы.