рубежка 1. вариант 7.

Что такое бессерверные вычисления?

Бессерверные вычисления – это модель облачных вычислений, при которой разработчики и пользователи не управляют серверной инфраструктурой напрямую. Судя по названию, многие могут подумать, что для вычисления не нужны никакие серверы, но это не так – серверы по-прежнему используются, но они полностью управляются облачным провайдером.

Например, у какой-то компании есть определённое приложение, и чтобы его запустит компании приходилось арендовать или приобретать серверы с фиксированными характеристиками. Это значит, что ей нужно учитывать возможные пики нагрузки, резервируя избыточные мощности с учётом возможного увеличения нагрузки. Даже если оставались какие-то неиспользованные ресурсы, за них все равно приходилось платить. А если вдруг происходило резкое увеличение нагрузки, то это вообще беда, могли возникать сбои из-за недостатка мощности.

Но потом появились бессерверные вычисления и в корни поменяли ситуацию. Теперь компании оплачивают только фактически использованные ресурсы, а распределением нагрузки занимается провайдер. Если на данный момент нагрузка не очень сильная, то ничего страшного! Мощности могут перераспределяться для других задач, что в свою очередь исключает лишние траты. А как говорится, сэкономленные деньги – заработанные деньги.

Как это работает?

Основная идея заключается в автоматическом масштабировании вычислительных мощностей. Когда нагрузка увеличивается, провайдер выделяет дополнительные ресурсы. При снижении активности избыточные мощности отключаются. Таким образом, разработчики сосредотачиваются исключительно на написании и развертывании кода, не задумываясь о том, как он будет выполняться, тк. настройкой серверов и их поддержкой занимается провайдер.

Поговорим о преимуществах и недостатках

Преимущества

1. Снижение затрат
   То есть мы платим только за используемые ресурсы, а не за их резервирование.
2. Гибкость
   Нам не нужно беспокоиться о настройке и поддержке серверной инфраструктуры.
3. Повышенная производительность
   Этот плюс вытекает из прошлого. Тк. мы можно не беспокоиться о настройке и поддержке серверной инфраструктуры, мы можем сосредоточиться на написании кода
4. Масштабируемость
   Автоматическое управление нагрузкой позволяет адаптироваться к изменяющимся требованиям.
5. Быстрое развёртывание
   Упрощение процесса развертывания приложений.

Недостатки

1. Отладка
2. "Холодный старт"

рубежка 2. вариант 7.

Безопасность одна из ключевых составляющих в принципе любой информационной системы, особенно в наше время. И конечно же, облако не стало исключением, ведь в нём данные и приложения становятся доступными. И дабы избежать различных неприятных ситуаций, существует множество компонентов безопасности, но вот некоторые из них:

1. Защита данных
   Защита данных включает в себя шифрование (использование собственных или облачных средств управления ключами, например AWS KMS или Azure Key Vault), резервное копирование и восстановление и классификацию данных (определение уровня конфиденциальности данных с соответствующей настройкой политик безопасности).
2. Управление доступом и идентификацией
   Этот компонент включает в себя многофакторную аутентификацию, ролевое управление доступом (RBAC – администраторы компании могут создавать конкретные роли на основе общих обязанностей сотрудника) и федеративное управление идентификацией (FIM – несколько компаний заключают между собой соглашение, которое позволяет участникам использовать одни и те же идентификационные данные для входа всех компаний, находящихся в группе).
3. Мониторинг и управление событиями безопасности
   Включает в себя системы обнаружения угроз, которые отслеживают активности. Также аллерты и автоматизацию (настройка оповещений о подозрительных действиях и автоматические реакции).
4. Сетевые меры безопасности
   Включает использование виртуальных частных сетей (VPC), настройку списков доступа (ACL), средства обнаружения и предотвращения попыток захватить конфиденциальную информация (WAF)
5. Управление уязвимостями
   Включает в себя регулярное обновление операционных систем и приложений в облаке, сканирование на уязвимость (например использование Nessus для регулярного анализа конфигураций и кода)

Касаемо обязательного использования. Я считаю, что всё вышеперечисленное обязательно должно быть настроено. Ведь никогда не знаешь откуда ждать подставы))