

1. Методология DevOps и Agile. Их сходства и различия.

DevOps объединяет две разрозненные команды (разработку и эксплуатацию), чтобы обеспечить быстрые выпуски программного обеспечения. Agile ориентирован на сотрудничество небольших команд друг с другом для быстрого реагирования на изменчивые потребности пользователей.

Сходства и различия

В Agile разработка завершается сразу после развертывания. В DevOps операции, которые происходят постоянно, например, мониторинг и модификации ПО;

В Agile разные специалисты несут ответственность за разработку, тестирование и развертывание ПО. В DevOps за все эти процессы отвечают специально обученные инженеры;

2. Жизненный цикл DevOps.

- Непрерывное развитие;
- Непрерывная интеграция;
- Непрерывное тестирование;
- Непрерывное развертывание;
- Непрерывный мониторинг;
- Постоянная обратная связь.

3. Интерфейс API. Типы API. REST API

API — интерфейс прикладного программирования, набор четко сформулированных правил взаимодействия между различными программными модулями.

Типы API

Внутренние
Партнерские
Публичные

REST API — это прикладной программный интерфейс (API), который использует HTTP-запросы для получения, извлечения, размещения и удаления данных.

4. Принципы REST API

- единый интерфейс;
- разграничение клиента и сервера;
- нет сохранения состояния;
- кэширование всегда разрешено;
- многоуровневая система;
- код предоставляется по запросу.

5. Протоколы HTTP и HTTPS.

HTTP-Протокол передачи гипертекста по 80 порту

HTTPS-Протокол передачи гипертекста с шифрованием, использует ssl сертификаты по 443 порту

6. SSL Сертификаты. Типы SSL сертификатов

SSL сертификаты-криптографический протокол для безопасной связи

Самоподписанные. Сертификат подписывается самим сервером.
Подписанные доверенным центром сертификации

7. Docker. Компоненты модуля Docker.

Docker — это платформа контейнеризации, применяемая для разработки, доставки и выполнения контейнеров.

Клиент Docker — это приложение командной строки

Сервер Docker — это управляющая программа

Docker Hub — это реестр контейнеров Docker

8. Dockerfile. Инструкции Dockerfile

Dockerfile — это текстовый файл с инструкциями, используемыми для сборки и запуска образа Docker.

Инструкции Dockerfile

FROM — задаёт базовый (родительский) образ.

LABEL — описывает метаданные. Например — сведения о том, кто создал и поддерживает образ.

ENV — устанавливает постоянные переменные среды.

RUN — выполняет команду и создаёт слой образа. Используется для установки в контейнер пакетов.

COPY — копирует в контейнер файлы и папки.

ADD — копирует файлы и папки в контейнер, может распаковывать локальные .tar-файлы.

9. Git. Ключевые компоненты Git.

Git — это быстрая, гибкая, высокомасштабируемая, бесплатная система управления версиями с открытым кодом.

Каталог GIT: хранит базу данных объектов, в которой происходит клонирование в хранилище на другой удаленный компьютер.

Рабочее дерево: файлы извлекаются из базы данных и помещаются на диск для внесения изменений.

Промежуточная область: действует как индексная часть, которая направляет на следующий коммит.

10. Клонирование и разветвление.

Разветвления репозитория подразумевает создание копии репозитория в учетной записи GitHub.

Клонирование репозитория — создание копии репозитория и его журнала на локальном компьютере.

11. GitLab. Возможности GitLab

GitLab — это инструмент для хранения и управления репозиториями Git.

Он дает возможность выполнять совместную разработку силами нескольких команд, применять обновления кода и откатывать изменения, если это необходимо.

12. Основные команды Git.

git status-выводит состояние рабочего дерева

git add-вы указываете Git, что нужно начать отслеживание изменений в определенных файлах.

git commit-сохранение результатов работы

git log-просматриваем сведения о предыдущих фиксациях.

13. CI/CD. Этапы CI/CD

CI/CD (Continuous Integration, Continuous Delivery — непрерывная интеграция и доставка)
Непрерывное развертывание

Этапы
Сборка
Ручное тестирование.
Релиз
Развертывание
Поддержка и мониторинг.
Планирование

14. Обновление служб (Последовательное, Сине-зелёное, ранний выпуск).

В последовательном обновлении развертываются новые экземпляры службы, которые сразу же начинают получать запросы.

При "сине-зеленом" развертывании новая версия развертывается параллельно с предыдущей версией.

В раннем выпуске обновленная версия развертывается в небольшое количество клиентов.

15. Kubernetes. Архитектура Kubernetes

Kubernetes — это переносимая расширяемая платформа с открытым кодом для управления и оркестрации контейнерных рабочих нагрузок.

Задаются задачи-переходит в уровень управления-узел(место управления выполнения вычислительных рабочих нагрузок)-получаем результаты, так как было несколько узлов

16. Кластер Kubernetes

Кластер-состоит из набора машин, так называемых узлы, которые запускают контейнеризированные приложения. Кластер имеет как минимум один рабочий узел.

17. Kubernetes. Уровень управления Kubernetes

Kubernetes — это переносимая расширяемая платформа с открытым кодом для управления и оркестрации контейнерных рабочих нагрузок.

уровень управления в кластере Kubernetes, запускает набор сервисов, управляющих функциональностью оркестрации в Kubernetes.

18. Службы работающие на уровне управления Kubernetes.

Сервер API
Резервное хранилище
Планировщик
Диспетчер контроллеров
Диспетчер облачных контроллеров

19. Kubernetes. Узел Kubernetes.

Kubernetes — это переносимая расширяемая платформа с открытым кодом для управления и оркестрации контейнерных рабочих нагрузок.

узел(место управления выполнения вычислительных рабочих нагрузок)

20. Службы, выполняемые в узле Kubernetes.

kubelet-это агент, который выполняется на каждом узле в кластере
kube-proxy-отвечает за локальную сеть кластера и выполняется на каждом узле.

Среда выполнения контейнера-это базовое программное обеспечение, которое выполняет контейнеры в кластере Kubernetes.

21. Модули POD Kubernetes. Жизненный цикл модуля POD

Pod
Контроллер
Веб-сайт

Сначала вы отправляете файл YAML с манифестом модуля pod в кластер. После отправки и сохранения файла манифеста в кластере он определяет желаемое состояние модуля pod.

Планировщик планирует модуль pod на работоспособном узле, который имеет достаточное количество ресурсов для запуска модуля.

22. Варианты развёртывания модулей POD.

- Шаблоны модулей pod
- Контроллеры репликации
- Наборы реплик
- Развертывания

23. Службы Kubernetes. Хранилище Kubernetes.

Служба — это объект Kubernetes, который обеспечивает стабильное сетевое подключение для модулей pod. Служба Kubernetes используется для обеспечения обмена данными между узлами.

Время существования тома Kubernetes задается и соответствует времени существования модуля pod. Это сопоставление значений времени существования означает, что том будет существовать дольше, чем контейнеры, выполняющиеся в модуле pod. Если модуль pod удаляется, том также будет удален.

24. Основные сущности Kubernetes.

Node (master и slave)
Namespace
Pod
ReplicaSet
Deployment
StatefulSet
DaemonSet
Label и Selector
Service

25. Интерфейс API. Основные категории API

API — интерфейс прикладного программирования

Категории API

Для работы с документами.

Для работы с графическими данными, видео или аудио.

Для принятия данных с сервера.

Для считывания данных с устройств.

Для хранения информации у пользователя.

26. Способы вызова API.

Напрямую-Это способ, при котором пользователь целенаправленно работает с API и ее функционалом.

Косвенный-Каждый пользователь, открывая программу, работает с API.

27. Docker. Создание образов. Тег Образа

Docker — это платформа контейнеризации, применяемая для разработки, доставки и выполнения контейнеров.

Для создания образов Docker используется Unionfs. Unionfs — это файловая система, позволяющая объединять в стек несколько каталогов, называемых ветвями.

Тег образа- указывается версия в нем.

28. Конфигурация сети для контейнера Docker.

Конфигурация сети Docker по умолчанию обеспечивает изоляцию контейнеров в узле Docker.

3 предварительно настроенных сети:

- Мост
- Узел
- нет

29. Git. Ключевые компоненты Git.

Git — это быстрая, гибкая, высокомасштабируемая, бесплатная система управления версиями с открытым кодом.

Каталог GIT: хранит базу данных объектов, в которой происходит клонирование в хранилище на другой удаленный компьютер.

Рабочее дерево: файлы извлекаются из базы данных и помещаются на диск для внесения изменений.

Промежуточная область: действует как индексная часть, которая направляет на следующий коммит.

30. Основные команды Git.

git status- выводит состояние рабочего дерева
git add- вы указываете Git, что нужно начать отслеживание изменений в определенных файлах.
git commit-сохранение результатов работы
git log-можно просматривать сведения о предыдущих фиксациях.

31. HTTP Методы. Клиент серверная инфраструктура.

Методы HTTP состоят из запросов:

GET;
POST;
PUT;
DELETE.

Модуль Docker состоит из нескольких компонентов, реализующих клиент-серверную архитектуру, при которой клиент и сервер работают на одном узле.

32. Функции управляемые с HTTP.

Кеш
Ослабление ограничений источника
Аутентификация
Прокси и туннелирование.
Сессии

