

Предложите решение для обеспечения развертывания, запуска и управления приложениями. Решение может состоять из одного или нескольких программных продуктов и должно описывать способы и принципы их взаимодействия.

Решение должно соответствовать следующим требованиям:

- **Поддержка контейнеров;**
- **Обеспечивать обнаружение сервисов и маршрутизацию запросов;**
- **Обеспечивать возможность горизонтального масштабирования;**
- **Обеспечивать возможность автоматического масштабирования;**
- **Обеспечивать явное разделение ресурсов доступных извне и внутри системы;**
- **Обеспечивать возможность конфигурировать приложения с помощью переменных среды, в том числе с возможностью безопасного хранения чувствительных данных таких как пароли, ключи доступа, ключи шифрования и т.п.**

Обоснуйте свой выбор.

Ответ.

Для выполнения перечисленных выше задач я бы предложил использовать Kubernetes. Предоставляемые им службы, такие как управление конфигурацией, обнаружение служб, балансировка нагрузки, сбор индексов и агрегирование журналов, реализованы на разных языках. Это позволяет нескольким группам в организации использовать платформу (включая разработчиков Java) для нескольких целей: разработка приложений, среда тестирования, создание среды (запуск системы управления ресурсами, создание служб, репозитория) и многое другое. Предлагаемое мной решение стало практически стандартом индустрии для оркестрации контейнеров. Оно лежит в основе многих облачных платформ контейнеризации и большинство компаний использует именно его, соответственно, и специалистов хорошо знающих и работающих с этим продуктом найти будет проще. С его помощью можно несколькими командами обновить нужную часть приложения, или даже все, откатить обновление назад, выделить ресурсы там, где их не хватает, и сократить мощности там, где их много.

В Kubernetes поддерживается автомасштабирование на основе наблюдаемого использования ЦП, памяти и ряда других пользовательских метрик. Kubernetes может автоматически масштабировать ИТ-систему в зависимости от потребностей приложения: входящего трафика и обрабатываемой нагрузки. Это значит, что приложение почти мгновенно получит нужные ресурсы в период пиковой нагрузки, а также не будет тратить

ресурсы в менее загруженное время. Есть обеспечение автомасштабирования подов (Horizontal Pod Autoscaler, HPA), которое автоматически изменяет количество подов в зависимости от значений выбранных показателей. Вертикальное автомасштабирование подов (Vertical Pod Autoscaler, VPA), которое автоматически изменяет объем ресурсов, выделяемых существующим подам.

Kubernetes может служить приложением для безопасной обработки конфиденциальной информации, связанной с работой приложений — паролей, OAuth-токенов, SSH-ключей. Обеспечение информационной безопасности происходит за счет механизма сетевых политик и ролевой авторизации пользователей по ключу. В зависимости от способа применения, данные и настройки можно обновлять без создания контейнера заново. Kubernetes – отличный вариант для Big Data проектов, когда требуется запускать множество контейнерных кластеров, работать с высоконагруженными сетями, а также с облачными PaaS/IaaS-решениями.

Кроме этого у Kubernetes несколько преимуществ для разработчиков. - В продакшн среде используются ровно те же контейнеры, которые тестировались в dev-среде. Так что если приложение работает в тесте, то, скорее всего, оно будет работать и в рабочей среде. При поставке кода в контейнере все зависимости уже в нем содержатся. Инфраструктура для приложений декларативно описана, так формируются «неизменяемые» конфигурации: в Kubernetes передается информация (схема), как решение должно работать, и он ее выполняет. Также Kubernetes следит, чтобы состояние кластера соответствовало желаемому, при необходимости перезапускает нужные компоненты.

Тем не менее Kubernetes может быть оказаться избыточен там где невелико число управляемых контейнеров.