iCode

Внедрение ERP-систем

СПОСОБЫ УСТАНОВКИ

Сравнительный анализ способов установки Odoo

СПОСОБЫ УСТАНОВКИ

1. Установка **в виде службы** на сервер (**Ubuntu**)
2. Оркестрация с помощью **Nomad**

Установка в виде службы на сервер (Ubuntu)

ubuntu odoo

ПРЕИМУЩЕСТВА

Прямая интеграция с системой: Приложение работает непосредственно на хостовой ОС, что может упростить настройку и интеграцию с другими компонентами системы

Меньшее потребление ресурсов: Без дополнительного слоя абстракции, как в случае с Docker, приложение может потреблять меньше ресурсов

Простота настройки: Для пользователей, знакомых с настройкой и управлением Linux сервисами, установка и настройка Odoo как сервиса может быть более привычной и понятной

НЕДОСТАТКИ

Масштабируемость: При увеличении нагрузки вы ограничиваетесь стратегией увеличения ресурсов хостовой VM

Зависимости: Сложность в управлении зависимостями и версиями

Сложность переноса: Перенос на другой сервер или в другую среду может потребовать дополнительных усилий по настройке окружения и зависимостей

Повторяемость: Могут возникнуть сложности с воспроизведением точно такой же среды развертывания на другом сервере или у других разработчиков

iCode

Установка в виде службы на сервер (Ubuntu)

ИНФРАСТРУКТУРНЫЕ СХЕМЫ ПРИМЕНЕНИЯ UBUNTU

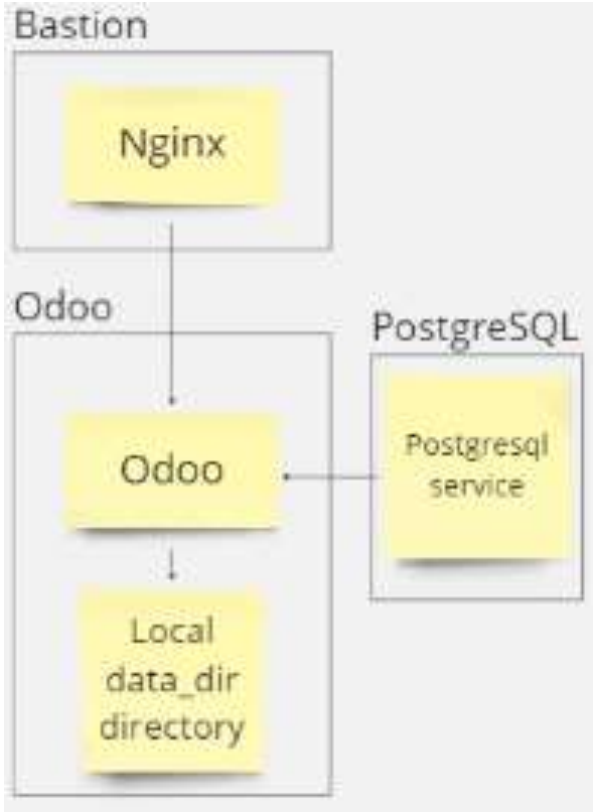
На данной схеме реализована **установка на 3 ВМ**:

1. Бастион - на ней располагается Nginx для маршрутизации HTTP запросов к приложению

2. ВМ на которой с помощью сервиса запускается **Odoo**, data_dir приложения хранится на этой же ВМ

3. Отдельно стоящая ВМ в которой установлен **PostgreSQL** в виде сервиса

Вариации схемы: Все компоненты схемы могут быть расположены на одной ВМ и развернуты в виде сервисов (Nginx, Odoo, PostgreSQL)



Установка в виде службы на сервер (Ubuntu)

ИНФРАСТРУКТУРНЫЕ СХЕМЫ ПРИМЕНЕНИЯ UBUNTU

На данной схеме реализована установка Odoo **в нескольких экземплярах**:

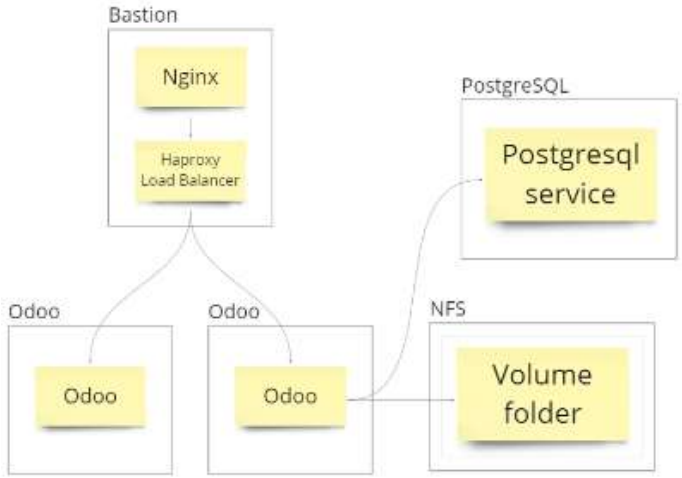
1. Бастион - на ней располагается Nginx для маршрутизации HTTP запросов к приложению

2. HAProxy выступает в виде балансировщика

3. ВМ на которой с помощью сервиса запускается Odoo, для хранения data_dir выполняется монитированием сетевого диска

4. Отдельно стоящая ВМ на которой установлен **PostgreSQL** в виде сервиса

5. NFS ВМ на которую выносится все файлы с которыми работает Odoo, это необходимо для доступа data_dir для различных экземпляров Odoo



Данная схема решает вопрос с динамическим масштабированием приложения.
Масштабирование происходит путем **добавления новой ВМ** на которую разворачивается новая инсталляция Odoo

HashiCorp

Оркестрация с помощью Nomad

ПРЕИМУЩЕСТВА

Масштабируемость и управление ресурсами: Nomad позволяет легко масштабировать Odoo и управлять выделением ресурсов для каждого экземпляра

Возможность использования **внутренних балансировщиков** для распределения нагрузки между экземплярами Odoo

Расширенные возможности **мониторинга потребления ресурсов Odoo**

Версионирование деплойментов

Распределенная система работы кластера (при падении одного из компонентов кластера, его работу перехватывает другой экземпляр, тем самым обеспечивая отказоустойчивость).

В одном кластере могут располагаться dev, stage, test, prod окружения с разделением прав доступа для разработчиков

Быстрое и простое добавление ресурсов и подключение новых компонентов кластера для вертикального и горизонтального масштабирования.

Гибкость и настройка: Nomad предоставляет богатые возможности конфигурации и планирования задач

Возможности **мультирегионального развертывания** приложения

Возможность развертывания любого другого приложения рядом в выделенных неймспейсах

НЕДОСТАТКИ

Настройка: Настройка и развертывание системы Nomad может потребовать больше усилий и навыков по сравнению с другими методами

Мониторинг: В идеальном кейсе необходимо добавить систему мониторинга для отслеживания работы кластера

Сложность: Для небольших и простых развертываний Odoo, использование Nomad может быть избыточным

iCode

HashiCorp

Оркестрация с помощью Nomad

ОПИСАНИЕ ОСНОВНЫХ КОМПОНЕНТОВ КЛАСТЕРА NOMAD

Кластер состоит из следующих **компонентов**:

Nomad –высокодоступный распределенный кластер с поддержкой центра обработки данных и планировщик приложений, предназначенный для поддержки современного центра обработки данных с поддержкой долговременных служб, пакетных заданий и многого другого. Nomad прост в эксплуатации и масштабировании, а также имеет встроенную интеграцию с Consul и Vault.

Consul – приложение для обнаружения сервисов (service discovery) на основе DNS и проверки их доступности. Используется для обеспечения связи между компонентами микросервисной инфраструктуры, позволяя создавать отказоустойчивую и масштабируемую систему с возможностью балансировки нагрузки.

Масштабирование кластера Nomad происходит **добавлением ресурсов** на клиентские компоненты кластера либо **увеличением кол-ва** клиентских **компонентов** кластера.

Дополнительные компоненты:

Boundary –бесплатная система безопасности с открытым исходным кодом на основе идентификационных данных. Ролевая и логическая авторизация сервисов. Использует технологию единого входа для управления подключения и отключения. Интегрируется с существующими инструментами и API

Vault – инструмент с открытым исходным кодом, который обеспечивает безопасное хранение и доступ к различным секретам, таким как пароли, сертификаты, токены

iCode

Оркестрация с помощью Nomad

ОПИСАНИЕ ОСНОВНЫХ КОМПОНЕНТОВ КЛАСТЕРА NOMAD

На данной схеме реализована установка Odoo в **нескольких экземплярах**:

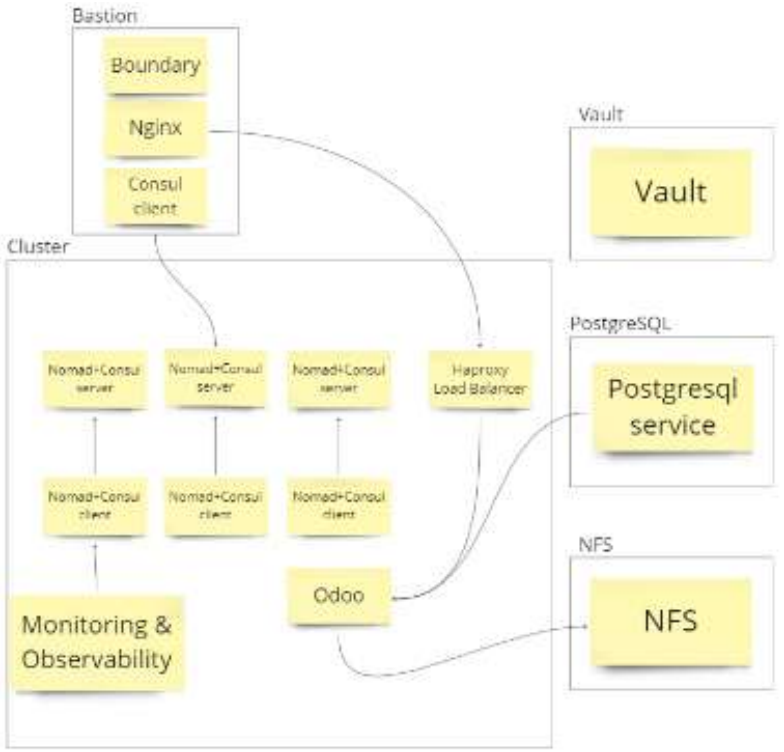
Bastion: на ней располагается Nginx для маршрутизации HTTP запросов к приложению, Consul клиент для определения сервисов внутри кластера путем DNS форвардинга. Также есть возможность установки Boundary на этой или на отдельно стоящей ВМ, обеспечивая дополнительную аутефикацию разработчикам для доступа к UI кластера, либо дополнительную возможность для предоставления SSH доступа к поинтам Boundary (любая ВМ которая указана в конфигурации)

HAProxy: выступает в виде балансировщика и располагается внутри кластера. Также он видит приложение Odoo благодаря Consul

Nomad+Consul server VM: управляющие ВМ кластера

Nomad+Consul client VM: клиентские ноды кластера, на которых располагается приложение

Separate VM with PostgreSQL: Отдельно стоящая ВМ в которой установлен PostgreSQL в виде сервиса



NFS VM: на которую выносится все файлы с которыми работает Odoo, это необходимо для доступа data_dir для различных экземпляров Odoo

СПОСОБЫ УСТАНОВКИ

РЕКОМЕНДУЕМЫЕ ПРОДУКТЫ ДЛЯ УЛУЧШЕНИЯ МОНИТОРИНГА

OpenSearch обеспечивает высокопроизводительный поиск в реальном времени и возможности аналитики на больших объемах данных. Это позволяет клиенту быстро анализировать и получать ценные инсайты из своих данных, что может улучшить принятие решений и повысить эффективность бизнеса

Prometheus и **Victoria Metrics** специализируются на мониторинге и оповещении в реальном времени, что позволяет оперативно реагировать на потенциальные проблемы в работе системы или приложений до того, как они окажут влияние на бизнес

Grafana поможет визуализировать метрики полученные от Prometheus, настроить оповещения по определенным триггерам когда требуется ручное вмешательство для устранения различных грядущих или наступивших инцидентов

iCode

СПОСОБЫ УСТАНОВКИ

ПРОЦЕСС ОБНОВЛЕНИЯ И ДОСТАВКИ

В нашей работе мы используем **GitLab CI/CD**

Общие элементы
Независимо от способа развертывания, pipeline в GitLab CI включает следующие этапы:

- ✓ **Тестирование**: Запуск автоматических тестов для проверки качества кода и функциональности
- ✓ **Сборка**: Подготовка всех необходимых артефактов для развертывания
- ✓ **Развертывание**: Автоматическое развертывание на тестовой/предпродукционной среде для дальнейшего тестирования и проверки
- ✓ **Продакшн**: Выпуск в продакшн после всех проверок и утверждений

iCode

Установка в виде службы на сервер (Ubuntu)

ОТКАЗОУСТОЙЧИВОСТЬ (UBUNTU)

Отказоустойчивость:

- ✓ Нативная установка обычно менее отказоустойчива, поскольку все компоненты системы (база данных, приложение) работают на одном узле. Отказ одного компонента может повлиять на доступность всей системы
- ✓ Репликация данных и балансировка нагрузки должны настраиваться вручную

Обновление:

- ✓ Обновления требуют внимательного планирования и могут потребовать временного простоя, особенно если требуются изменения схемы базы данных
- ✓ Возможность автоматизации процесса обновления ограничена и часто требует ручного вмешательства

Обслуживание:

- ✓ Требуется регулярное ручное вмешательство для резервного копирования, мониторинга и обновления системы

HashiCorp

Оркестрация с помощью Nomad

ОТКАЗОУСТОЙЧИВОСТЬ (NOMAND)

Отказоустойчивость:
- ✓ Nomad предлагает высокую отказоустойчивость и масштабируемость на уровне кластера благодаря автоматическому планированию, миграции задач и восстановлению после сбоев
- ✓ Встроенные механизмы обнаружения сервисов и интеграция с Consul упрощают создание отказоустойчивых распределенных систем

Обновление:
- ✓ Поддержка стратегий обновления с минимальным простоем, включая голубые/зеленые развертывания и развертывания с подтверждением
- ✓ Автоматическое управление версиями и зависимостями задач

Обслуживание:
- ✓ Централизованное управление и мониторинг состояния кластера и задач
- ✓ Упрощение процесса масштабирования и обслуживания благодаря декларативному описанию задач и автоматическому планированию ресурсов

iCode

РЕКОМЕНДАЦИИ ПО ТИПАМ ОТКАЗОУСТОЙЧИВОСТИ

Выбор между способами установки зависит от требований к отказоустойчивости, масштабируемости и сложности управления

- ✓ Для **крупных**, **высоконагруженных систем**, требующих высокой доступности и масштабируемости, использование **Nomad** предоставит лучшие возможности по отказоустойчивости и обслуживанию

- ✓ **Установка в виде службы на сервер (Ubuntu):**
- Оптимальный выбор **для малых и средних проектов**, где простота управления и быстрая настройка имеют первостепенное значение
- Предпочтительна в ситуациях, где **контейнеризация или использование оркестраторов не представляется возможным или целесообразным** из-за специфических требований окружения

СПОСОБЫ УСТАНОВКИ

КАКИЕ ИНСТРУМЕНТЫ И ПРОДУКТЫ МЫ ИСПОЛЬЗУЕМ

iCode

Благодарим
за Ваше внимание!