СИСТЕМНЫЕ ТРЕБОВАНИЯ

Развертывание на GPU

8+ vCore
32+ Гб RAM
SSD 100 Гб свободного места
Требования к видеокарте:
24+ Гб VRAM
Compute Capability 8.0+
Linux с поддержкой Docker 22+ (совместимость проверена с RedHat, Debian, Ubuntu, CentOS)
NVIDIA Docker Toolkit (nvidia-docker2) или NVIDIA CUDA Toolkit

Развертывание на CPU (не рекомендуется)

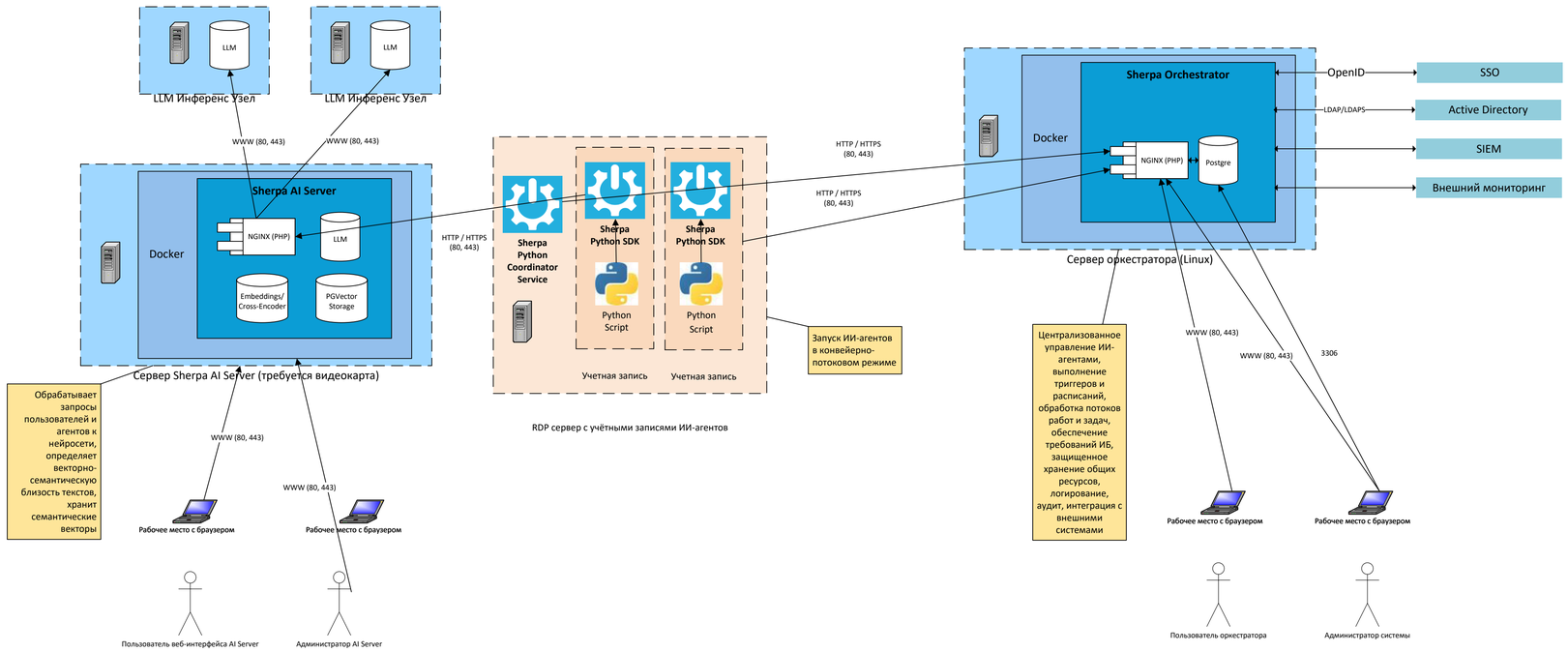
12+ vCore (рекомендуется 48+ vCore)
64+ Гб RAM
SSD 100 Гб свободного места
Требования к видеокарте:
24+ Гб VRAM
Compute Capability 8.0+
Linux с поддержкой Docker 22+ (совместимость проверена с RedHat, Debian, Ubuntu, CentOS)

Совместимые модели видеокарт

NVIDIA Data Center Products: NVIDIA A2, NVIDIA A16, NVIDIA A10, NVIDIA A30, NVIDIA A40, NVIDIA A100, NVIDIA L4, NVIDIA L40, NVIDIA H100.
NVIDIA RTX Desktop: RTX A4000, RTX A5000, RTX A6000, RTX 6000.
NVIDIA RTX Mobile: RTX A2000, RTX A3000, RTX A4000, RTX A5000.
GeForce Products: Geforce RTX 3060, Geforce RTX 3060 Ti, GeForce RTX 3070, GeForce RTX 3070 Ti, GeForce RTX 3080, GeForce RTX 3080 Ti, GeForce RTX 3090, GeForce RTX 3090 Ti, GeForce RTX 4070 Ti, GeForce RTX 4080, GeForce RTX 4090
GeForce Notebook Products: GeForce RTX 3050, GeForce RTX 3050 Ti, GeForce RTX 3060, GeForce RTX 3060 Ti, GeForce RTX 3070, GeForce RTX 3070 Ti, GeForce RTX 3080, GeForce RTX 3080 Ti, GeForce RTX 4050, GeForce RTX 4060, GeForce RTX 4070, GeForce RTX 4080, GeForce RTX 4090
Jetson Products: Jetson AGX Orin, Jetson Orin NX, Jetson Orin Nano

ТЕХНОЛОГИЧЕСКИЙ СТЕК (СПРАВОЧНО)

NGINX / PHP / Python/ C++ / Angular
PostgreSQL + PGVectors
Docker



ОПИСАНИЕ:
Sherpa AI Server предназначен для использованиея больших языковых моделей, векторно-сементического и гибридного хранения и поиска документов, а также организации интерфейсов взаимодействия пользователей с ИИ-агентами.

Диаграмма развертывания:
Все компоненты Sherpa AI Server устанавливаются локально в сети Заказчика, без связи с внешними серверами или службами SaaS. Развертывание Sherpa AI Server осуществляется с помощью Docker-контейнера. AI Server поддерживает работу с несколькими разными моделями и/или разными экземплярами одной модели, такие экземпляры располагаются в отдельных LLM Инференс Узлах. AI Server отвечает за маршрутизацию запросов между LLM Инференс Узлами.

Состав решения:
(1) Веб-интерфейс пользователя для непосредственного общения с большой языковой моделью
(2) Векторно-семантическое хранилище
(3) Инференс-сервер большой языковой модели
(4) Эмбеддинг-сервер большой языковой модели

Конфигурация сети:
Конфигурация портов и сетевые протоколы Sherpa Sherpa AI Server могут быть настроены для поддержки всех общих требований брандмауэра. Для взаимодействия с веб сервером используется https, опционально возможно http. При установке с помощью TLS клиент должен предоставить необходимые сертификаты, разместив их по пути /opt/app/config/certs/, переименовав их в aiserver.crt и aiserver.key.

Механизмы аутентификации
Аутентификация веб-пользователей в Sherpa AI Server производится с помощью логина и пароля.
Аутентификация внешних приложений, включая ИИ-агентов, производится с помощью API-токена.

ОПИСАНИЕ:
Sherpa Orchestrator и Sherpa Python Coordinator / Sherpa Python SDK обеспечивают автоматизацию бизнес-процессов с помощью Python-агентов, выполняемых на терминальных серверах или виртуальных машинах. Оркестратор по событиям или входящим данным из внешней среды (через API), по команде ИИ-агента или бизнес-пользователя или по расписанию запускает Python-сценарии (выполняемые в учётных записях терминальных серверов без участия пользователя). Python-сценарии с помощью Sherpa Python SDK обмениваются с оркестратором логами, задачами из очередей, централизованно хранимыми учетными данными и общими данными.

Диаграмма развертывания:
Все компоненты платформы устанавливается локально в сети Заказчика, без связи с внешними серверами или службами SaaS. Возможность и необходимость доступа компонентов платформы к внутренним и внешним системам определяется решаемой в рамках бизнес-процесса задачей. Развертывание Sherpa Python SDK производится с помощью exe-инсталлятора в технических пользовательских учетных записях. Установка Sherpa Coordinator Service производится в администраторской учетной записи терминального сервера с помощью exe-инсталлятора.
По умолчанию развертывание Sherpa Orchestrator осуществляется с помощью Docker-контейнера.

Последовательность шагов процесса:
(1) Разработчик сценария ИИ-агента с помощью любого подходящего Python IDE создаёт сценарий (скрипт) с использованием Sherpa Python SDK. Готовые сценарии передаются на технические учетные записи ИИ-агентов с помощью функции удаленной публикации в Sherpa Orchestrator. Версионирование сценариев также осуществляется с помощью Sherpa Orchestrator.
(2) Серверный компонент Sherpa Orchestrator поддерживает связь с агентами, запущенными на клиентских машинах, хранит конфигурации и версии сценариев, общие глобальные переменные и учетные данные, логи и скриншоты работы агентов, журналы аудита оркестратора, пользователей, роли и тенанты самого оркестратора, лицензии всех компонентов платформы и статистику исполнения сценариев.
(3) Пользователи и администраторы Sherpa Orchestrator получают доступ к ресурсам, настройкам и статистике с помощью веб-приложения оркестратора, доступного через веб-браузер. Sherpa Orchestrator включает в себя веб-сервер Nginx, интерпретатор PHP и реляционную базу данных (по умолчанию - MariaDB).
(4) По расписанию, вызову API, команде агента либо по другому поддерживаемому триггеру Sherpa Orchestrator даёт задание Sherpa Python Coordinator Service соответствующего терминального сервера создать RDP-подключение к локальной или удаленной учетной записи, выделенной для ИИ-агента, при этом на одном терминальном сервере может быть размещено и одновременно активно несколько таких учетных записей. Вход в учётную запись агента производится с предоставленными оркестратором логином и паролем. После входа в учетной записи запускается соответствующий экземпляр Sherpa Python SDK, подключается к оркестратору, получает задание на выполнение сценария и сам сценарий, хранящийся в оркестраторе. В процессе исполнения сценария Sherpa Python SDK может передавать оркестратору текущий статус, задачи, логи, значения глобальных переменных и учетных данных или получать от него задачи, значения глобальных переменных и учетных данных, команды для «мягкого» или «жесткого» завершения сценария. После завершения исполнения сценария Sherpa Python SDK выполняет logoff из своей учетной записи.

Конфигурация сети:
Конфигурация портов и сетевые протоколы могут быть настроены для поддержки всех общих требований брандмауэра. Конфигурация порта по умолчанию выглядит следующим образом:
*Sherpa Python SDK, Sherpa Coordinator исходящие на Sherpa Orchestrator: 80 или 443
*Во всех сетевых коммуникациях инициатива установки подключения и первоначального запроса принадлежит только клиентским компонентам, то есть Sherpa Python SDK и Sherpa Coordinator. Sherpa Orchestrator по своей инициативе не выполняет запросы к клиентам.
*Связь с базой данных: 3306 и 1433-настраивается
*Доступ пользователя к веб-интерфейсу Sherpa Orchestrator: 80 или 443
Для взаимодействия с веб сервером используется https, опционально возможно http.
Sherpa RPA поддерживает защищенную связь (с использованием протокола TLS 1.2) между Sherpa Python SDK, Sherpa Coordinator и Sherpa Orchestrator. При установке с помощью TLS клиент должен предоставить необходимые сертификаты, разместив их по пути /opt/app/config/certs/, переименовав их в orchestrator.crt и orchestrator.key.

Механизмы аутентификации
Аутентификация Sherpa Python SDK, Sherpa Coordinator в Orchestrator осуществляется с помощью Bearer Token, передаваемого в заголовке запросов. Bearer Token сопоставляется с уникальным GUID каждого экземпляра Sherpa Python SDK, Sherpa Coordinator. Для аутентификации пользователей веб-интерфейса Orchestrator применяется авторизация с помощью пары логин-пароль. При повторном входе используется сессионная кука, имеющая ограниченный срок жизни.

Логирование
Для логирования используется компонент Monolog. События аудита и системные ошибки сохраняются в выделенную таблицу базы данных.

СИСТЕМНЫЕ ТРЕБОВАНИЯ

Sherpa Orchestrator:

4 vCore
8 Гб RAM
SSD 100 Гб свободного места
Linux с поддержкой Docker 22+ (совместимость проверена с RedHat, Debian, Ubuntu, CentOS)

Sherpa Coordinator / Sherpa Python SDK:

2 vCore
4 Гб RAM
SSD / HDD 5 Гб свободного места
Windows 7 – Windows 11
или Windows Server 2012 – 2022
или Linux (Debian, Ubuntu, AstraLinux)
.NET Framework 4.8+
PowerShell 5.1+

ТЕХНОЛОГИЧЕСКИЙ СТЕК (СПРАВОЧНО)

Sherpa Orchestrator:

NGINX / PHP / Angular
MariaDB / PostgreSQL
Clickhouse (опция)
Docker

Sherpa Coordinator / Sherpa Python SDK:

C# (.NET Framework 4.8+)
Python 3.9+
PowerShell 5.1+