ООО «АИДАТех»
119021, Г.Москва, ул Льва Толстого, д. 2/22 стр. 6
ИНН 9704261020
КПП 770401001

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «KAGECORE ML PLATFORM»

Описание функциональных характеристик ПО

Листов 31

Содержание

Обозначения и сокращения 3
Термины и определения 6
1 Назначение Системы 7
2 Основные функции Системы 9
3 Компоненты Системы 10
3.1 Основные компоненты Системы 10
3.1.1 KageCore ML Platform 11
3.1.1.1 Система управления виртуализацией 11
3.1.1.2 Система контейнеризации и оркестрации 14
3.1.2 KageCore ML Platform. Модуль витрины сервисов 16
3.1.3 KageCore ML Platform. Модуль тарификации 19
3.1.4 KageCore ML Platform. Модуль пользовательского мониторинга 20

Обозначения и сокращения

В настоящем документе применяют следующие сокращения и обозначения:

API	-	Application Programming Interface, программный интерфейс взаимодействия
CPU	-	Central Processing Unit, процессор
CLI	-	Command Line Interface, интерфейс командной строки
CSV	-	Comma-Separated Values, текстовый формат для представления табличных данных, где значения в каждой строке разделены запятыми
FTP	-	File Transfer Protocol, протокол передачи файлов по сети
HDFS	-	Hadoop Distributed File System, распределенная файловая система Hadoop для хранения файлов больших размеров с возможностью потокового доступа к информации, поблочно распределённой по узлам вычислительного кластера, который может состоять из произвольного аппаратного обеспечения
HTML	-	HyperText Markup Language, язык гипертекстовой разметки
HTTP	-	HyperText Transfer Protocol, протокол передачи гипертекста
HTTPS	-	HyperText Transfer Protocol Secure, расширение протокола HTTP для поддержки шифрования
GID	-	Group Identifier, идентификатор группы
GPU	-	Graphics Processing Unit, графический процессор
IAM	-	Identity and Access Management, совокупность технологий, операций, методов и политик для управления доступом пользователей к инфраструктуре
IP	-	Internet Protocol, межсетевой протокол
IPMI	-	Intelligent Platform Management Interface, интеллектуальный интерфейс управления платформой, предназначенный для автономного мониторинга и управления функциями, встроенными непосредственно в аппаратное и микропрограммное обеспечения серверных платформ
JSON	-	JavaScript Object Notation, текстовый формат обмена данными, основанный на JavaScript
MAAS	-	Metal as a Service, система управления физическими серверами, позволяющая автоматизировать их развертывание и управление

ML	-	Machine Learning, машинное обучение
ONNX	-	Open Neural Network Exchange, открытая библиотека программного обеспечения для построения нейронных сетей глубокого обучения
LDAP	-	Lightweight Directory Access Protocol, легковесный протокол доступа к каталогам
LUN	-	Logical Unit Number, логический (виртуальный) том
MTU	-	Maximum Transmission Unit, максимальная единица передачи
NFS	-	Network File System, протокол сетевого доступа к файловым системам
OSI	-	The Open Systems Interconnection model, сетевая модель стека (магазина) сетевых протоколов OSI/ISO
PDF	-	Portable Document Format, формат файлов, разработанный компанией Adobe Systems для представления документов в электронном виде, сохраняющий форматирование и внешний вид документа, независимо от используемой платформы и программного обеспечения
RAM	-	Random Access Memory, оперативная память
RBAC	-	Role-Based Access Control, управление доступом на основе ролей
SDK	-	Software Development Kit, набор инструментов для разработки программного обеспечения, объединённый в одном ПО, обычно содержит комплект необходимых библиотек, компилятор, отладчик, а также интегрированную среду разработки
SSH	-	Secure Shell, безопасная оболочка
SSL	-	Secure Sockets Layer, уровень защищенных сокетов
TLS	-	Transport Layer Security, протокол защиты транспортного уровня
UI	-	User Interface, пользовательский интерфейс
URL	-	Uniform Resource Locator, унифицированный указатель ресурса
USB	-	Universal Serial Bus, последовательный интерфейс для подключения периферийных устройств к вычислительной технике
VNC	-	Virtual Network Computing, протокол удаленного доступа к рабочему столу компьютера
БД	-	база данных

ВМ	-	виртуальная машина
ИТ	-	информационные технологии
ОС	-	операционная система
СКО	-	Система контейнеризации и оркестрации
ПО	-	программное обеспечение
СУВ	-	Система управления виртуализацией
СУВР	-	Система управления вычислительными ресурсами
СУБД	-	система управления базами данных
СХД	-	система хранения данных
ФСТЭК	-	Федеральная служба по техническому и экспортному контролю
ЦП	-	центральный процессор

Термины и определения

В настоящем документе применяют следующие термины с соответствующими определениями:

ALD Pro	-	набор сетевых служб сервера Astra Linux для создания службы каталога и организации централизованного управления ИТ-инфраструктурой. Продукт построен на хорошо известных компонентах с открытым исходным кодом, которые используют только открытые протоколы для обмена информацией
кластер	-	логическая группа хостов с общими доменами хранения и ЦП одного типа (Intel или AMD). Если модели ЦП хостов относятся к разным поколениям, то используются только те функции, которые присутствуют во всех моделях. Виртуальные машины динамически распределяются между хостами кластера и могут перемещаться между ними в соответствии с политиками, заданными в кластере, и настройками виртуальных машин. Кластер является самым высоким уровнем, на котором могут определяться политики электропитания и разделения нагрузки
контейнер	-	легковесные запускаемые образы, в состав которых входит некоторое ПО и его зависимости. Поскольку в контейнерах виртуализируется операционная система, вы можете запускать контейнеры одинаково в любом совместимом окружении

1 Назначение Системы

Платформа KageCore ML Platform, включая модули KageCore ML Platform. Модуль тарификации, KageCore ML Platform. Модуль витрины сервисов и KageCore ML Platform. Модуль пользовательского мониторинга (далее – ПО, Система), предназначена для предоставления высокопроизводительных вычислительных ресурсов и совокупности сервисов (IaaS/PaaS/SaaS) в интересах одной или нескольких организаций и проектов, позволяя эффективнее обучать и эксплуатировать модели искусственного интеллекта, оптимизировать производственные и научно-исследовательские процессы и тем самым способствовать ускоренному развитию технологий ИИ.

ПО предназначено для решения следующих задач:

– формирование пула аппаратных средств (серверы с CPU и GPU, высокоскоростные сети, системы хранения), доступного пользователям из единого веб-интерфейса на основе квотирования, ролевой модели и механизма биллинга;

– автоматизация процесса заказа ресурсов (vCPU, RAM, GPU, объём дисков) через портал самообслуживания и программный интерфейс (API);

– использование готовых шаблонов (маркетплейса) с преднастроенными библиотеками и фреймворками (Keras, PyTorch, ONNX, Scikit-learn, TensorFlow), а также средами разработки JupyterLab, VSCode;

– использование механизмов контейнеризации и виртуализации для гибкой оркестрации и оперативного масштабирования ML-заданий;

– предоставление инфраструктурных сервисов (виртуальные машины) и платформенных (среды разработки, БД, аналитические инструменты) в унифицированном виде;

– разграничение прав и ресурсов на уровне отдельных организаций, проектов и групп, что позволяет параллельно вести несколько сценариев инференса и обучения;

– внедрение роли и квот (RBAC), позволяющих ограничивать доступ и лимитировать объём ресурсов (CPU, GPU, память, хранилище);

– учёт и детальный биллинг (включая CPU, GPU, хранение данных), обеспечивающие прозрачность и справедливое распределение затрат между участниками;

– поддержка экспериментов и трассировки (логирование метрик, параметров, артефактов) с целью воспроизводимости и контроля качества обучаемых моделей;

– предоставление пользователям возможности оперативного выбора и автоконфигурации необходимых сервисов из унифицированного каталога (marketplace), содержащего преднастроенные модули, которые охватывают базовые и

прикладные сервисы, а также автоматизированного развертывания в контейнерной или виртуальной среде без ручной настройки;

– обеспечение администраторов и пользователей платформы инструментами наблюдения за компонентами платформы. Предоставление возможности сбора доступных метрик и их отображение, а также сбора анализа журналов приложений, операционных систем, при необходимости.

2 Основные функции Системы

ПО предназначено для выполнения следующих функций в формате самообслуживания:

– доступ пользователей к вычислительным ресурсам из веб-интерфейса;

– обеспечение возможности заказа виртуальных машин с GPU как ресурсов;

– обеспечение возможности заказа контейнеров с GPU (целых и частей GPU в эксклюзивном режиме, а также целых и частей GPU в режиме совместного использования) и средами разработки с установленными компонентами, необходимыми для разработки и обучения ML моделей;

– обеспечение доступа на основе ролей (RBAC), позволяющую назначать права пользователям в зависимости от их ролей;

– обеспечение интеграции с внешними каталогами пользователей (LDAP, ALD Pro, FreeIPA);

– ограничение объёмов вычислительных ресурсов, доступных для заказа в проекте/папке/организации.

Дополнительные функции:

– выделенный портал управления администратора;

– автоматизация развёртывания вычислительных узлов при масштабировании;

– инфраструктурный и пользовательский мониторинг для наблюдаемости систем KageCore ML Platform.

3 Компоненты Системы

3.1 Основные компоненты Системы

ПО KageCore ML Platform состоит из следующих основных составных частей:

– система управления виртуализацией (далее – СУВ);

– система контейнеризации и оркестрации (далее – СКО);

– система управление вычислительными ресурсами и обеспечивающая масштабирование уровня IaaS/PaaS/SaaS (далее – СУВР);

– инфраструктурный мониторинг;

– система машинного обучения.

В состав ПО также входят самостоятельные модули:

1) KageCore ML Platform. Модуль витрины сервисов.
2) KageCore ML Platform. Модуль тарификации.
3) KageCore ML Platform. Модуль пользовательского мониторинга.

Эксплуатация ИИ-моделей предполагает использование распределённой инфраструктуры на базе Docker и Kubernetes. Система пользовательского мониторинга обеспечивает:

– оперативное выявление сбоев — аварийное завершение контейнеров, зависание процессов;

– интеграцию с системами оповещения и автоматического реагирования — перезапуск контейнеров, масштабирование подов;

– поддержку аудита и отчётности — сбор метрик для анализа эффективности и подготовки регламентных отчётов.

Мониторинг является неотъемлемой частью MLOps процессов, обеспечивающей прозрачность и управляемость жизненного цикла ИИ-решений.

3.1.1 KageCore ML Platform

KageCore ML Platform — это унифицированная инфраструктурная платформа, предназначенная для предоставления ИТ-услуг по моделям IaaS в единой управляющей среде. Платформа и ее модули, обеспечивают полный цикл управления вычислительными ресурсами: от заказа и автоматизированного развёртывания до эксплуатации, мониторинга и тарификации. Она объединяет компоненты виртуализации, контейнеризации и оркестрации, машинного обучения и системного мониторинга, обеспечивая пользователям гибкий и стандартизированный доступ к современным цифровым технологиям. Архитектура платформы построена с учётом требований к масштабируемости, отказоустойчивости, безопасности и импортозамещению, поддерживает работу как в гибридных, так и в полностью изолированных (без доступа в Интернет) контурах. Платформа совместима с российскими операционными системами, включая Astra Linux, RedOS и их сертифицированные ФСТЭК версии, что делает её пригодной для использования в регулируемых отраслях и государственных структурах. Управление осуществляется через единый веб-интерфейс, API и командную строку, что обеспечивает удобство как для администраторов, так и для конечных пользователей. Интеграция с каталогами пользователей (ALD Pro, LDAP), средствами DNS/NTP, системами хранения и мониторинга позволяет разворачивать KageCore ML Platform как автономное решение или в составе существующей ИТ-инфраструктуры. Основу платформы составляют система управления виртуализацией (СУВ), система контейнеризации и оркестрации (СКО) и система управления вычислительными ресурсами (СУВР), которые работают в тесной связке, обеспечивая прозрачное предоставление ресурсов в формате «as a service».

3.1.1.1 Система управления виртуализацией

СУВ предназначена для размещения пользовательских заказов развёрнутых в виде ВМ.

Перечень управляющих компонентов программной инфраструктуры Системы:

– СУВР.

– Каталог пользователей ALD Pro (в случае отсутствия каталога пользователей у заказчика).

– Средства автоматического развёртывания и предоставления сервисов DNS/NTP (в случае отсутствия указанных сервисов у заказчика).

– Инфраструктурная часть модуля пользовательского мониторинга.

СУВ обеспечивает размещение вычислительных ресурсов, заказанных пользователем в СУВР, в рамках услуги IaaS/PaaS/SaaS в виде ВМ для выполнения как расчётных задач с использованием CPU и GPU-ускорителей, так и для использования готовых моделей как сервис (инференс).

Система управления виртуализацией (СУВ) обеспечивает виртуализацию аппаратных ресурсов и создание изолированных виртуальных машин на базе физических серверов архитектуры x86. Она рассчитана на крупные инфраструктуры и поддерживает работу не менее 250 хостов виртуализации в рамках одного кластера, а также централизованное управление не менее чем 400 хостами из единого интерфейса.

Развёртывание возможно в разных режимах: СУВ устанавливается как на выделенный физический сервер, так и в виде виртуальной машины, в том числе на гипервизорах управляемого кластера. При этом система допускает установку без подключения к сети Интернет. Управление обеспечивается через единый веб-интерфейс, консоль администратора и интерфейс командной строки, а также через административный и пользовательский веб-портал. Для удобства администраторов доступна локализация интерфейса на русском языке.

Работа с виртуальными машинами организована с учётом требований к производительности и отказоустойчивости. СУВ поддерживает создание шаблонов ВМ, быстрое развёртывание новых экземпляров из библиотеки образов, а также клонирование работающих виртуальных машин без необходимости их выключения. Ресурсы можно масштабировать на лету: система позволяет динамически добавлять RAM, vCPU и дисковое пространство без остановки виртуальной машины. Для сохранения состояния поддерживается создание мгновенных снимков с возможностью возврата. Виртуальные машины могут запускаться автоматически при старте сервера, восстанавливаться в случае аварийного состояния, а при выходе из строя физического узла система обеспечивает автоматический перезапуск ВМ на других серверах кластера. Важнейшей функцией является живая миграция: виртуальные машины можно перемещать между серверами без выключения операционной системы, а балансировка нагрузок выполняется автоматически, также в «горячем» режиме.

СУВ гибко работает с системами хранения данных. Она функционирует как с физическими СХД, так и в составе гиперконвергентной среды. Поддерживается тонкое выделение дискового пространства, что позволяет использовать хранилища эффективнее. Администратор может перевести домен хранения в режим обслуживания без остановки приостановленных ВМ, а после удаления хранилищ система автоматически очищает всю связанную с ними информацию на хостах. Для обеспечения высокой доступности предусмотрена поддержка метро-кластеров: инфраструктура может строиться сразу на нескольких площадках, используя как классические СХД с необходимым функционалом, так и программно-определяемые хранилища.

Сетевая подсистема СУВ включает полный набор функций корпоративного уровня. Система позволяет создавать виртуальные сети на базе распределённого коммутатора с управлением через графический интерфейс, обеспечивать распределённую коммутацию пакетов уровня L2 независимо от физической топологии и местоположения серверов. Администратор может управлять IP-адресацией подключённых ВМ, настраивать MTU на распределённом коммутаторе, а также задавать маршрутизацию трафика: как между изолированными сетями без выхода за пределы виртуального коммутатора, так и между изолированными сетями и физической сетью. Поддерживается трансляция сетевых адресов (SNAT), редактирование таблицы маршрутизации и назначение статических маршрутов. При необходимости виртуальные машины можно подключать к существующему широковещательному домену с сохранением централизованного управления IP-адресами.

СУВ предусматривает развитые механизмы безопасности и управления доступом. Поддерживается ролевое разграничение прав с интеграцией со службой каталогов, а управление локальными пользователями возможно через веб-интерфейс. Система позволяет задавать максимальное количество сессий для пользователей из внешних доменов. В критических случаях администратор может подключиться к гостевой операционной системе через сервер виртуализации даже при отсутствии сетевого адаптера у ВМ.

Дополнительно СУВ поддерживает проброс USB-устройств, локальных дисков и LUN на виртуальные машины, а также технологию NVIDIA vGPU с возможностью использования NVIDIA Unified Memory для виртуальных GPU. Интеграция с внешними системами мониторинга осуществляется за счёт передачи им данных о состоянии системы.

Для повышения надёжности СУВ поддерживает ручное и автоматическое резервное копирование конфигурации сервера управления, в том числе по расписанию, с выбором места хранения резервной копии.

Работа администратора упрощается за счёт широких возможностей визуализации и отчётности. Система позволяет экспортировать списки виртуальных машин, хостов виртуализации, доменов хранения, пулов и событий в формате CSV, формировать отчёты о состоянии, утилизации и ошибках виртуальной инфраструктуры за заданный период времени в форматах PDF и HTML. Кроме того, СУВ строит диаграмму виртуальной инфраструктуры в реальном времени, на которой отображаются кластеры, серверы, хранилища, их текущее состояние и взаимосвязи.

Для подключения к виртуальным машинам система поддерживает консольный доступ по протоколам SPICE, VNC и HTML5.

Таким образом, СУВ объединяет масштабируемость до 400 хостов под единым управлением, гибкость в управлении вычислительными и сетевыми ресурсами, развитые механизмы обеспечения высокой доступности и отказоустойчивости, а также удобные средства администрирования, визуализации и интеграции, что делает её надёжной платформой для построения корпоративной виртуальной инфраструктуры.

3.1.1.2 Система контейнеризации и оркестрации

Система контейнеризации и оркестрации (СКО) предназначена для размещение заказов (CaaS/PaaS/SaaS) произведенных пользователев в СУВР, и создания среды исполнения контейнеров, в которой пользователи могут запускать расчётные задачи машинного обучения и разворачивать конечные продукты в режиме инференса. Платформа обеспечивает гибкое выделение ресурсов, включая CPU, RAM, GPU и дисковое пространство, что позволяет адаптировать среду под конкретные задачи.

Ресурсы для заказов создаются из заранее преднастроенных образов, содержащих основные фреймворки и библиотеки Python: Keras, ONNX, PyTorch, Scikit-learn и TensorFlow. Для соответствия требованиям импортозамещения и сертификации СКО поддерживает, размещение компонентов СКО в российских операционных системах, включая Astra Linux и РЕД ОС, в том числе сертифицированную ФСТЭК версию РЕД ОС. Платформа может устанавливаться как на bare-metal инфраструктуру, так и в закрытых контурах без доступа к сети Интернет.

СКО оснащена развитой системой централизованного управления квотированием ресурсов, что особенно важно при решении задач искусственного интеллекта, требующих значительных вычислительных мощностей — GPU, CPU, памяти и хранилища. Квотирование позволяет справедливо распределять ресурсы между проектами, задавать многоуровневую политику управления (на уровне организации, папки или проекта), а также формировать основу для учёта и бюджетирования ИТ-ресурсов. Администратор может задавать лимиты (квоты) на объём ресурсов и количество объектов для конкретного пространства имён (namespace).

Для обеспечения прозрачности работы и стабильности платформа оснащена встроенными средствами мониторинга и логирования. СКО предоставляет возможность собирать метрики и логи контейнерной среды, экспортировать их во внешние хранилища, а также визуализировать данные в удобном интерфейсе. Дополнительно реализован встроенный функционал сканирования образов контейнеров на наличие уязвимостей, что повышает уровень безопасности. Для защиты от перегрузки платформа поддерживает встроенные механизмы ограничения нагрузки.

Для управление объектами контейнерной инфраструктуры, администраторы имеют возможность создавать, удалять и редактировать поды (pods), а также просматривать логи и метрики утилизации вычислительных ресурсов по каждому из них. Аналогичный функционал доступен и для пространств имён, при этом предоставляется доступ к метрикам использования ресурсов для каждого пространства.

Таким образом, СКО представляет собой гибкую и безопасную платформу для контейнеризации и оркестрации, способную работать в изолированных средах и на российских операционных системах. Она сочетает поддержку преднастроенных образов для задач машинного обучения, развитую систему квотирования и мониторинга, средства визуализации и встроенные механизмы защиты, что делает её эффективным инструментом для организации CaaS/PaaS/SaaS-сервисов корпоративного уровня.

3.1.2 KageCore ML Platform. Модуль витрины сервисов

Модуль «Витрина сервисов» является центральным элементом платформы KageCore ML Platform и представляет собой единый интерфейс для заказа, управления и контроля использования ИТ-услуг по моделям PaaS и SaaS, а также расширяя возможности платформы в части IaaS. Он необходим для автоматического размещения и управления жизненным циклом пользовательских заказов в СКО или СУВ, обеспечивая пользователям стандартизированный и удобный доступ к широкому спектру вычислительных ресурсов и платформенных сервисов. Через витрину пользователи могут формировать заказы, управлять ими, вносить изменения в ранее созданные заявки, а также отслеживать использование ресурсов в режиме реального времени, что особенно важно для эффективного управления проектами и оптимизации затрат. Интерфейс поддерживает UI-схемы для упрощённого заказа, версионирование продуктов и настройку сложных процессов в виде графов, что делает взаимодействие с платформой интуитивно понятным и эффективным как для разработчиков, так и для администраторов.

На уровне IaaS расширение возможностей происходит за счет предоставления пользователю функционала над созданными заказами, а именно расширять заказанные вычислительные ресурсы, планировать отложенные действия. Лицензии на операционные системы и коммерческое ПО предоставляются заказчиком. Такая гибкость позволяет планово провидить обновления, выводить ВМ для планового технического обслуживания, а также выполнять действия в едином пользовательском интерфейсе, без необходимости перемещаться в интерфейс СКО или СУВР. Поддержка российских операционных систем, таких как Astra Linux и РЕД ОС, обеспечивает соответствие требованиям информационной безопасности и импортозамещения, что критично для государственных и регулируемых отраслей.

На уровне PaaS витрина предоставляет доступ к готовым окружениям разработки, включая JupyterLab и VS Code, с предустановленными фреймворками и библиотеками Python, такими как Keras, PyTorch, ONNX, Scikit-learn и TensorFlow. Эти технологии применяются для решения задач машинного обучения и построения цифровых решений: Scikit-learn используется для классических задач ML, таких как классификация, кластеризация и регрессия; PyTorch и TensorFlow — для разработки и обучения современных нейронных сетей, включая модели глубокого обучения в компьютерном зрении, обработке естественного языка и генеративных моделях; ONNX обеспечивает переносимость обученных моделей между различными фреймворками и средами, что упрощает их интеграцию в производственные процессы. Благодаря предустановленным окружениям пользователи могут быстро начать работу, избежав ручной настройки зависимостей, что снижает риски несовместимости версий, повышает воспроизводимость экспериментов и сокращает время вывода ИИ-решений в промышленную эксплуатацию.

Помимо сред разработки, в состав PaaS входят разнообразные платформенные сервисы, предназначенные для построения современных распределённых приложений и аналитических решений. Сервисы управления кодом и файлами, такие как GitFlic и FileBrowser, используются для совместной разработки, хранения и управления исходным кодом и данными. Брокеры сообщений — Apache Kafka и RabbitMQ — применяются для построения отказоустойчивых систем обмена данными, потоковой обработки событий и интеграции микросервисов. Системы управления базами данных включают как классические реляционные СУБД (MySQL, PostgreSQL), так и специализированные решения: ClickHouse — для аналитики в реальном времени, Elasticsearch — для поиска и лог-анализа, Redis — для кэширования, MongoDB — для хранения полуструктурированных данных, а векторные базы Milvus и ChromaDB — для работы с эмбеддингами в задачах семантического поиска, рекомендательных систем и RAG-архитектур. Программные серверы приложений, такие как Nginx и WildFly, позволяют разворачивать веб-сервисы и enterprise-приложения.

Среда разработки предоставляет графический веб-интерфейс и поддерживает выполнение кода на Python 2/3, Ruby, Perl, R и Bash, что делает её универсальной платформой для анализа данных, научных вычислений, статистического моделирования и разработки скриптов. Возможность выполнять код по блокам, визуализировать результаты в формате текста, изображений, аудио и видео, а также работать с файлами (.py, .R, .md) позволяет создавать интерактивные аналитические отчёты, исследовательские блокноты и обучающие материалы. Поддержка Markdown, JSON, CSV, Vega и VegaLite с предварительным просмотром, множественный курсор, работа с несколькими проектами и интеграция терминала с ядром блокнота делают рабочее пространство гибким и удобным для командной разработки.

Среда управления жизненным циклом моделей, построенная на базе MLflow, играет ключевую роль в обеспечении воспроизводимости, аудита и прозрачности процесса разработки ИИ. Она предоставляет веб-интерфейс и API с поддержкой Python, Java, R и REST, обеспечивая регистрацию экспериментов с фиксацией параметров, метрик, артефактов и метаданных. Это позволяет отслеживать, какие гиперпараметры, данные и код использовались при обучении модели, что критично для анализа результатов, сравнения подходов и воспроизведения лучших решений. Визуализация истории экспериментов, группировка запусков и тегирование помогают организовать работу над проектами, особенно в условиях командной разработки. Централизованное хранилище моделей поддерживает анализ, выполнение инференса и упаковку кода для повторных экспериментов, что упрощает переход от прототипирования к промышленному внедрению. Метаданные хранятся в СУБД, а артефакты — в S3-совместимых хранилищах, на FTP/SFTP, NFS или HDFS, что обеспечивает масштабируемость и отказоустойчивость. Система активно используется для управления полным циклом ML — от экспериментов до деплоя, интеграции с MLOps и обеспечения соответствия регуляторным требованиям.

На уровне SaaS витрина сервисов предоставляет доступ к готовым прикладным решениям, которые можно использовать без необходимости развертывания и настройки инфраструктуры. К числу таких сервисов относятся, например, платформы для автоматизированного построения прогнозных моделей, сервисы обработки естественного языка (NLP) для анализа текстов, классификации запросов и генерации ответов, а также решения для автоматизации бизнес-процессов на основе ИИ. Такие SaaS-продукты применяются для быстрого внедрения цифровых решений в бизнес-подразделениях: отделы аналитики используют их для оперативной обработки данных, HR — для анализа резюме и оценки кандидатов, колл-центры — для автоматизации обработки обращений, а маркетинг — для сегментации клиентов и персонализации предложений. Благодаря готовым интерфейсам и интеграциям, пользователи могут подключаться к сервисам по API или через веб-панель, не обладая глубокими техническими знаниями, что способствует демократизации доступа к технологиям искусственного интеллекта.

Таким образом, витрина сервисов выступает как универсальная подсистема, объединяющая возможности IaaS, PaaS и SaaS, и обеспечивает полный цикл работы с цифровыми и ИИ-решениями — от заказа ресурсов и разработки до эксплуатации, мониторинга и тарификации. Она создаёт единую среду для эффективной и безопасной реализации современных технологических проектов, поддерживая как фундаментальные ИТ-потребности, так и передовые задачи в области искусственного интеллекта, аналитики и цифровой трансформации, обеспечивая при этом прозрачность, контроль и масштабируемость.

3.1.3 KageCore ML Platform. Модуль тарификации

Модуль тарификации предоставляет администраторам изолированный веб-интерфейс для управления стоимостью ресурсов и сервисов. Поддерживаются тарифные классы для процессорных ядер (vCPU), памяти (RAM), GPU, дискового пространства, лицензий на операционные системы и промежуточное ПО. Администраторы могут создавать собственные тарифные классы, назначать тарифы для каждого сервиса и формировать индивидуальные тарифные планы для разных организаций и продуктов платформы, в соответствии с ролевой моделью. Возможность пополнения счетов пользовательских папок позволяет гибко управлять бюджетами, распределять финансовые ресурсы между подразделениями и контролировать расходы, что особенно актуально в условиях внутреннего IT-аутсорсинга и shared-service-моделей.

3.1.4 KageCore ML Platform. Модуль пользовательского мониторинга

Модуль пользовательского мониторинга KageCore ML Platform обеспечивает комплексный процесс наблюдения за состоянием ИТ-инфраструктуры системы, включая сбор, обработку, анализ и визуализацию данных, а также своевременное оповещение об инцидентах. Он предназначен для поддержания высокой доступности, производительности и безопасности ресурсов, а также для интеграции с централизованными системами мониторинга и логирования, что делает его ключевым компонентом операционной устойчивости платформы.

Модуль поддерживает обработку разнообразных типов данных, что позволяет унифицировать сбор информации из гетерогенной инфраструктуры. В качестве источников журналов данных принимаются логи в форматах Syslog, соответствующие стандартам RFC 3164 и RFC 5424, — традиционно используемые в Unix- и Linux-системах для передачи системных событий. Также поддерживаются структурированные логи в формате JSON, что обеспечивает удобную парсинговую обработку и интеграцию с современными приложениями и микросервисами. Для задач кибербезопасности реализована поддержка формата CEF (Common Event Format), стандартизированного решения для обмена событиями безопасности между системами SIEM. Кроме того, модуль обрабатывает логи веб-серверов в Apache Combined и Common Log Format, а также принимает неструктурированные текстовые логи (Plain Text) с временной меткой, обеспечивая гибкость при работе с устаревшими или специализированными приложениями.

Сбор метрик осуществляется в формате Prometheus, что обеспечивает совместимость с широким спектром современных технологий, включая Kubernetes и облачные сервисы. Поддерживаются все основные типы метрик: Counter — монотонно возрастающие счётчики, используемые для учёта событий, таких как количество запросов или ошибок; Gauge — метрики с произвольными значениями, применяемые для отслеживания текущих состояний, например, уровня потребления CPU, RAM или числа активных подключений; Histogram — для анализа распределения значений, например, времени обработки запросов; а также Summary — метрики, позволяющие вычислять квантили и агрегированные значения, что критично для анализа производительности.

Источниками данных выступают как системные демоны, так и специализированные агенты. В качестве источников логов используются syslog, auditd, journald, а также любые форматы, поддерживаемые векторным агентом (vector agent), что обеспечивает гибкость и масштабируемость сбора. Метрики поступают через экспортеры, совместимые с форматом Prometheus, включая node_exporter, kube-state-metrics и пользовательские решения. Сбор данных реализован по двум основным методам: poll (опрос источников по расписанию) и push (передача данных от источника по событию), что позволяет адаптировать подход к особенностям каждого сервиса.

Для обеспечения интеграции с внешними системами мониторинга и аналитики модуль предоставляет возможность экспорта данных: логи могут быть переданы в сторонние системы через протоколы, совместимые с vector sync-s, что обеспечивает надёжную доставку и буферизацию; метрики экспортируются в формате Prometheus, что позволяет интегрировать их с любыми системами, поддерживающими этот стандарт.

Управление сроком хранения данных реализовано на гибкой основе, что позволяет организовывать как кратковременное, так и долгосрочное архивирование. Для хранения метрик используются системы VictoriaMetrics и/или Prometheus, обеспечивающие высокую производительность при работе с временными рядами. Журналы данных сохраняются в Loki, оптимизированной для масштабируемого хранения логов, а также в Opensearch — для полнотекстового поиска и анализа. Такой подход позволяет балансировать между требованиями к производительности, стоимости хранения и длительности ретеншена.

Перед обработкой и хранением входящие данные могут быть обогащены или трансформированы: модуль поддерживает добавление меток (labels), фильтрацию по условиям, модификацию содержимого и преобразование форматов — например, из текстового в JSON. Это позволяет нормализовать данные из разных источников, добавлять контекст (например, теги проекта, среды или зоны), а также очищать или маскировать чувствительную информацию, повышая качество анализа и соответствие требованиям безопасности.

Визуализация данных осуществляется через настраиваемые панели и дашборды, которые можно создавать, редактировать и экспортировать на основе шаблонов. Поддерживаются временные графики (Time Series) для анализа динамики метрик, числовые индикаторы (Stat Panel) с отображением изменений, табличные представления (Table) для детального анализа, а также панели отображения логов (Logs Panel) с возможностью фильтрации по уровням, источникам и ключевым словам. Для анализа распределений используются тепловые карты (Heatmap), особенно эффективные при работе с histogram-метриками. Дополнительно доступны информационные панели (Text Panel) с поддержкой Markdown, индикаторы с пороговыми значениями (Gauge) и список активных оповещений (Alert List), что позволяет консолидировать всю критически важную информацию на единой панели.

Оповещение об инцидентах реализовано на основе анализа поступающих данных: модуль позволяет настраивать алерты по пороговым значениям на графиках, а также по сложным условиям с использованием языка PromQL. При превышении заданных критериев система формирует уведомления и отправляет их во внешние каналы связи, например: электронную почту, мессенджер Telegram. Администраторы могут добавлять, удалять или временно отключать оповещения, что обеспечивает гибкость при обслуживании системы и реагировании на инциденты.

Таким образом, модуль пользовательского мониторинга KageCore ML Platform обеспечивает полный цикл работы с данными наблюдаемости — от сбора и обработки до визуализации, анализа и оповещения, обеспечивая прозрачность, контроль и оперативное реагирование на изменения в инфраструкту.