

Описание ПО Модуль мониторинга СХД.

Программное обеспечение Модуль мониторинга СХД предназначено для предоставления средств наблюдения за состоянием систем хранения данных под управлением ПО BAUM STORAGE AI.

1. Описание работы системы

1.1 Программная архитектура.

В системе используется система микрофронтендов, основанная на React+webpack. Программное обеспечение представляет из себя реализацию компонентной базы для предоставления интерфейса для взаимодействия пользователя с элементами системы в веб-браузере. Для реализации взаимодействия с REST-API СХД BAUM STORAGE AI используется механизм кэширующего прокси со своей реализацией REST-интерфейса для решения проблемы “политики одного источника” при взаимодействии с СХД. Компонентная база приложения разделена на бизнес-логику и неуправляемые UI компоненты, а также общие компоненты, предоставляющие базовую логику обработки типовых ситуаций.

Механизм работы с приложением строится следующим образом:

1. Пользователь запускает приложение путем старта Docker-контейнера на операционной системе, находящейся в единой сети с СХД BAUM STORAGE AI, которые необходимо мониторить.
2. После старта приложения пользователю необходимо открыть веб-браузер и открыть страницу с IP-адресом или именем хоста той системы, где запущен Docker-контейнер.
3. При первом посещении страницы Приложения пользователь будет видеть пустое окно с логотипом и кнопкой Настройки в верхнем правом углу экрана.
4. Нажав на нее, пользователь может добавить новый кластер для мониторинга.
5. Для того чтобы добавить новый кластер - необходимо ввести авторизационные данные для подключения к СХД BAUM STORAGE AI, после чего нажать кнопку “Добавить”

6. Программный модуль, получив команду на добавление нового кластера - отправляет запрос к кэширующему прокси-серверу, куда передает авторизационные данные
7. В свою очередь кэширующий прокси-сервер забирает из заголовка запроса информацию о целевых системах, отправляет запрос к REST API СХД BAUM STORAGE AI
8. После добавления нового кластера, информация о добавленном кластере сохраняется в localStorage веб-приложения, а также, в локальной БД кэширующего прокси-сервера, для ускорения взаимодействия с СХД

1.2 Событийная модель

В Модуле мониторинга СХД используется классическая модель взаимодействия со сторонним внешним API путем коммуникации через REST API интерфейсы.

Добавленные в Приложение кластера BAUM STORAGE AI, сохраняются в хранилище браузерной информации, далее при необходимости получить данные по кластеру - происходит опрос нужного хоста из хранилища.

Каждый блок опросной части:

- Показатели нагрузки на систему
- Показатели производительности системы в разбивке по чтению и записи
- Индикацию о состоянии контроллеров кластера
- Индикацию состояния аппаратного обеспечения
- Индикацию состояния дисков
- Индикацию состояния журнального раздела
- Список событий

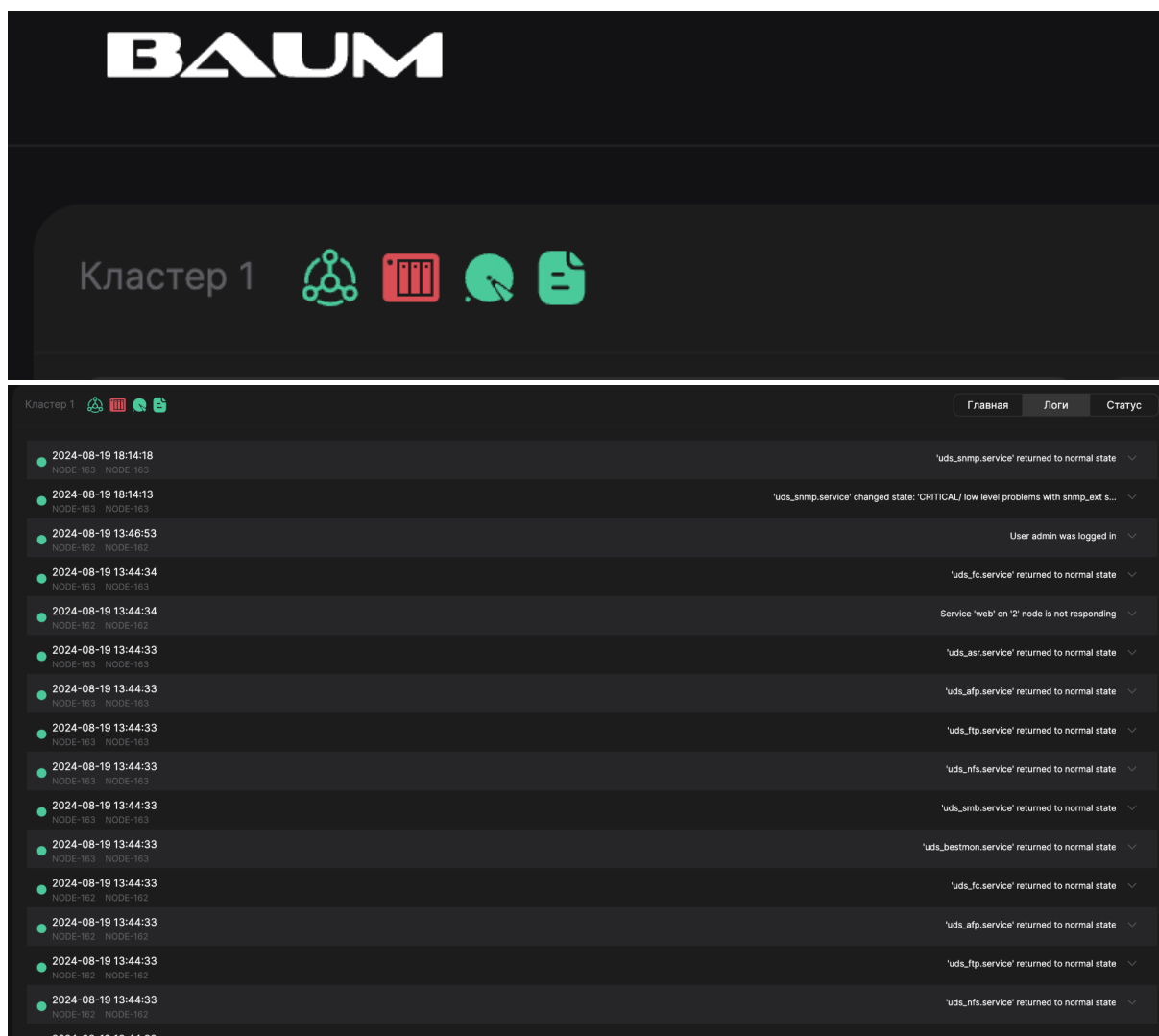
Получает информацию из потока обновлений, который запускает запросы к указанному кластеру и деформируя полученные объекты, заносит данные в Redux-Store, хранилище мета-информации веб-приложения.

В Приложении реализован механизм кэширования, при котором если пришедшие данные не отличаются от предыдущих полученных данных - обновление хранилища не будет вызвано, и также, не будет вызвана перерисовка компонентов.

1.3 Метрики приложения

Модуль мониторинга СХД предоставляет следующие параметры для мониторинга жизнедеятельности кластера:

- Показатели нагрузки на систему
- Показатели производительности системы в разбивке по чтению и записи
- Индикацию о состоянии контроллеров кластера
- Индикацию состояния аппаратного обеспечения
- Индикацию состояния дисков
- Индикацию состояния журнального раздела
- Список событий



Все метрики полностью консистентны с показателями СХД, также, помимо показателей количества и списка проблем можно получить детализированную информацию по каждому разделу оборудования в разделе “Статус”

Все снимаемые метрики отображаются в разделе, соответствующему полученному блоку данных, обновление блоков - централизованное. Метрики по оборудованию снимаются одного REST-маршрута, событий лог - с другого, состояние контроллеров кластера - с третьего.

1.4 Компоненты системы

1) Точка монтирования компонентов - App

Начальный компонент, который включает в себя базовую разметку приложения, навигационную структуру, и разметку верхнего меню. Включает в себя базовую стилистическую структуру и определяет сбросы уровня корневого html

2) Слой верхнего меню

Компонент навигации, который подгружается вместе с точкой монтирования, независимо от него, содержит логотип и кнопку “Настройки”

3) Компонент ошибок и уведомления ApplicationErrorPlate

Отвечает за вывод ошибок, является прямым потомком App, имеет структуру независимого DOM-дерева - Portals.

4) Компонент ошибок и уведомления WarningPlate

Отвечает за вывод уведомлений, является прямым потомком App, имеет структуру независимого DOM-дерева - Portals.

5) Компонент Кластера - ClusterForm

Отвечает добавление списка кластеров в приложение.

6) Компонент Настроек

Отвечает за управление списком кластеров, является прямым потомком компонента ClusterForm

7) Компонент Dashboard

Является основным компонентом системы - отвечает за вывод и отображение информации по метрикам системы.

8) Слой Redux-Store и срезов -

Хранилище метаданных, реализация кэширования и функций - обработчиков, которые реализуют операции управления хранилищем, а также функции-обработчики для взаимодействия с REST-API

9) Слой UI-компонентов

Простые компоненты, не реализующие бизнес-логику, предоставляющие возможность вывода элементов управления.