

Документация администратора

Astra Monitoring

Оглавление

1	Обзор платформы	3
1.1	Описание	3
1.2	Архитектура продукта и описание компонентов	3
1.2.1	Клиентская часть	3
1.2.2	Серверная часть	4
1.2.3	Пользовательский интерфейс	4
2	Подготовка к установке платформы Астра Мониторинг	5
2.1	Системные требования	5
2.2	Программные требования	6
2.3	Сетевые настройки	6
2.4	Требования к учётной записи	7
3	Установка и обновление платформы Астра Мониторинг	9
3.1	Установка и обновление Астра Мониторинг с использованием Docker Compose	9
3.2	Установка и обновление Астра Мониторинг в кластер Kubernetes (Helm Chart)	10
3.3	Установка базы данных ClickHouse	17
3.4	Установка и обновление клиентской части для мониторинга продуктов ПАО Группа Астра	17
3.4.1	Установка Vector	18
3.4.2	Установка node-exporter	19
3.4.3	Установка systemd-exporter	19
3.4.4	Установка клиентской части для мониторинга ALD Pro / FreeIPA	20
3.4.5	Установка клиентской части для мониторинга ПК СВ Брест	21
4	Настройка платформы Астра Мониторинг	24
4.1	Добавление объектов мониторинга	24
4.2	Настройка триггеров для создания событий	25
4.3	Настройка оповещений	26
5	Резервное копирование баз данных	28
5.1	Резервное копирование базы данных ClickHouse	28
5.2	Резервное копирование базы данных PostgreSQL	28
6	Мониторинг продуктов ПАО Группа Астра	30
6.1	Мониторинг ОС Astra Linux	30
6.2	Мониторинг FreeIPA	30
6.3	Мониторинг ALD Pro	36
6.4	Мониторинг ПК СВ Брест	40
6.5	Мониторинг операционной системы Astra Linux	61
6.6	Настроенные триггеры для создания событий по логам	63
7	Пользовательский интерфейс Астра Мониторинг	65
7.1	Пользовательское окно "События"	65
7.1.1	Общая информация	65
7.1.2	Основная таблица с событиями	65
7.1.3	Правая боковая панель	66
7.1.4	Дополнительные элементы	66
7.2	Пользовательское окно "Объекты"	67
7.2.1	Описание функционала окна "Объекты"	67
7.2.2	Дополнительные возможности окна "Объекты"	69

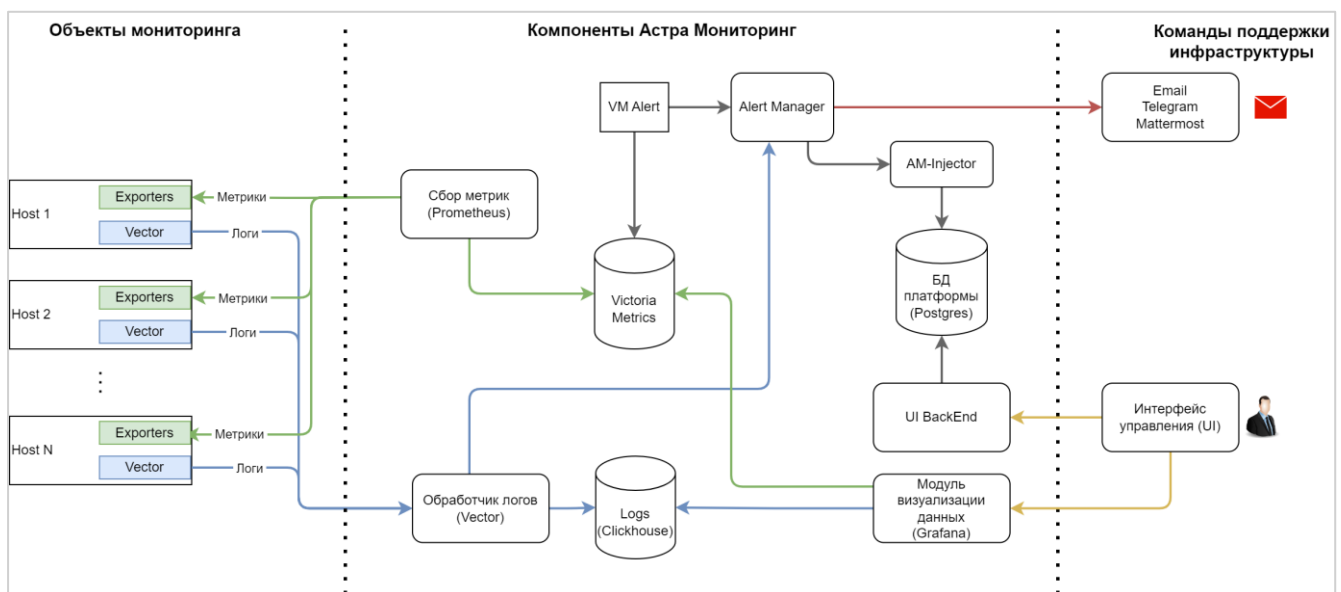
1 Обзор платформы

1.1 Описание

Астра Мониторинг - это программная платформа, предназначенная для мониторинга продуктов ПАО Группа Астра, а также физической, виртуальной инфраструктуры, сервисов, приложений.

Платформа предназначена для сбора метрик, сбора и анализа логов, формирования событий по предустановленным порогам, уведомления о событиях через соответствующие каналы.

1.2 Архитектура продукта и описание компонентов



Можно выделить следующие основные элементы: клиентская часть, серверная часть, пользовательский интерфейс.

1.2.1 Клиентская часть

Компоненты:

- Набор "Экспортеров". Обеспечивают сборку числовых характеристик наблюдаемой системы (загрузка CPU, ОЗУ, загрузку сетевых интерфейсов и т.д.).
- Программный продукт Vector. Обеспечивает обработку и отправку логов на платформу Астра Мониторинг.

Клиентская часть состоит из хостов - объектов мониторинга, на которые установлен набор различных экспортеров, в зависимости от роли, которую выполняют эти хосты.

Например, на сервер контроллера домена могут быть установлены `node-exporter`, `systemd-exporter`, `freeipa-exporter` и `vector` для сбора логов, а на хост виртуализации Брест - `node-exporter`, `systemd-exporter`, `vector` и `libvirt-exporter`. Установленные экспортеры собирают доступные им метрики, в зависимости от назначения экспортера, и публикуют их по http, используя специфичный для конкретного экспортера порт, например, `http://<host_address>:9100/metrics` для `freeipa-exporter`.

1.2.2 Серверная часть

Компоненты:

- Prometheus. Метрики собираются со всех совместимых систем и сервисов.
- Vectod.dev. Логи принимаются и обрабатываются продуктом Vector, который осуществляет запись полученных данных в базу данных на основе Clickhouse. Архитектура передачи данных "Vector - Vector" позволяет осуществлять доставку логов на сложные конфигурации дочерних подразделений.
- Victoria Metrics. Метрики с объектов наблюдения сохраняются в базу данных Victoria Metrics.
- Clickhouse. Данные хранятся в СУБД и за счёт сжатия занимают меньше места, чем сырые данные. Логи разбиты на категории по уровню критичности события и все записи с низким уровнем критичности (info, debug, trace) сохраняются в базу данных Info Logs, а записи с высоким уровнем критичности (warning, critical, error) помещаются в базу данных Critical Logs.
- PostgreSQL. База данных, которая отвечает за хранение объектов наблюдения, информации о событиях объектов мониторинга.
- AlertManager. Сохраняет информацию о событии в БД PostgreSQL через AM-Injector. Реализует отправку уведомления о событии в соответствующие каналы (Email, Telegram, Mattermost).

Серверная часть состоит из указанных выше компонентов и предназначена для сбора, хранения и обработки данных от объектов мониторинга.

Например, Prometheus собирает метрики путем обращения к опубликованным на объектах мониторинга экспортерам и передает их далее на хранение в базу Victoria Metrics. Собранные в базе Victoria Metrics данные регулярно анализируются компонентом vmalet в соответствии с заданными правилами триггеров. При обнаружении соответствия данных какому-либо правилу, vmalet отправляет сообщение в alertmanager с заданной в правиле информацией о хосте, имени правила, его кратком и полном описании, критичности и о прочих сопутствующих тегах. В alertmanager события группируются, обрабатываются и далее отправляются по заданным каналам оповещениями командам поддержки, а также в базу данных платформы через am-injector и далее в интерфейс управления.

Сбор и обработка логов производится в несколько ином порядке - установленный на объекте мониторинга vector собирает логи в соответствии с конфигурацией и отправляет их в серверную часть компонента vector. Серверный обработчик логов vector производит дополнительную обработку поступающих логов, выделение и добавление ключевых полей, а также анализ поступающих данных на соответствие заданным правилам, например, ищет записи об ошибках выполнения каких-либо операций. При обнаружении подобных ошибок может быть отправлено сообщение в alertmanager и далее по указанной выше схеме. Собранные и обработанные логи отправляются на хранение в базу данных Clickhouse.

1.2.3 Пользовательский интерфейс:

Компоненты:

- Модуль визуализации метрик и логов построен на базе программного продукта Grafana, представляет из себя набор представлений данных и интерфейс анализа логов;
- Интерфейс управления - Admin UI. Предназначен для добавления объектов мониторинга, а также для просмотра информации о событиях по объектам мониторинга.

Пользовательский интерфейс позволяет визуализировать собранные данные, отобразить метрики в виде индикаторов и графиков, увидеть информацию об обнаруженных на объектах мониторинга проблемах, добавлять объекты мониторинга в систему или удалять их и т.д.

2 Подготовка к установке платформы Астра Мониторинг

Платформа Астра Мониторинг может быть развернута в нескольких вариантах:

Размер установки	Способ распространения	Ограничения
Small	1 сервер, Docker compose/Однонодовый кластер Kubernetes	Нет отказоустойчивости на уровне приложения Низкая производительность из-за совмещения компонент
Medium	2 сервера, Docker compose / Однонодовый кластер Kubernetes + Сервер БД	Нет отказоустойчивости на уровне приложения Производительность средняя из-за вынесения роли БД на отдельный сервер
Large	3+ нодовый кластер Kubernetes	Отказоустойчивость на уровне кластера Производительность выше среднего

2.1 Системные требования

Платформа Астра Мониторинг может быть установлена в кластер Kubernetes или используя Docker Compose сборку.

Требуемые ресурсы для запуска в кластере Kubernetes: vCPU 4900m, memory - 4574Mi

	Количество	Requests		Limits	
		memory	CPU	memory	CPU
Admin-UI	2	20	10	100	100
AlertManager	1	30	10	100	200
ClickHouse	2	500	500	800	1500
Grafana	1	150	50	300	150
PostgreSQL	1	100	100	200	200
Prometheus	1	500	100	1024	300
Vector	1	70	50	150	200
Vmalert	1	50	20	200	150
Vmetrics	1	300	100	800	500
Итого		2240Mi	1450m	4574Mi	4900m

Минимальные требования для запуска с использованием Docker Compose: 4 vCPU, 8GB RAM, 200GB SSD.

2.2 Программные требования

При развертывании в кластере Kubernetes должны быть выполнены следующие условия:

1. Kubernetes cluster - версия не ниже v1.23.5 (протестирована работа на версиях v1.23.5 - локальная установка и v1.23.6 - Yandex Managed Kubernetes).
2. Установленный Ingress Nginx для публикации ресурсов.
3. Настроенный Persistent Storage (для Clickhouse инстансов - SSD хранение, для остального можно использовать HDD в качестве хранения). Если используется Yandex Managed Kubernetes, то можно использоваться `ус-network-hdd``ус-network-ssd``ус-network-nvme` для дисков БД (Clickhouse, PostgreSQL) и `csi-s3` для бэкапов Clickhouse (т.е. требуется ReadWriteMany для CronJob для бэкапа).
4. Установленный Helm v3.
5. Установленный Reloader для перезапуска приложения при изменении конфигурации компонентов. Этот компонент не является обязательным и не идет в составе платформы Астра Мониторинг, однако наличие данного компонента облегчает настройку.
6. Выделенный Namespace для запуска приложения.

При развертывании как Docker Compose должны быть выполнены следующие условия:

1. Версия докера начиная с 24.0.6
2. убедиться, что необходимые порты не заняты другими приложениями на сервере:

Компонент	Порт
vector	9102/TCP
alertmanager	9093/TCP
grafana	3000/TCP
prometheus	9090/TCP
victoria metrics	8428/TCP
vmalert	8880/TCP
admin-ui	80/TCP
admin-back	8000/TCP

2.3 Сетевые настройки

- Сетевая связность до хранилища Docker образов и Helm charts (<https://registry.astralinux.ru/> <https://dl.astralinux.ru/>)
- Сетевая связность сервера мониторинга и объектов мониторинга
- Открытые на сетевых экранах и доступные для сервера системы мониторинга следующие сетевые потоки (табл)

Источник (source)	Назначение (destination)	Порт	Протокол	Экспортер	Комментарий
Система мониторинга (сервер сбора данных)	registry.astralinux.ru	443	HTTPS		Доступ до опубликованных Docker образов
	dl.astralinux.ru	443	HTTPS		Доступ до опубликованных Helm чартов и экспортеров

Источник (source)	Назначение (destination)	Порт	Протокол	Экспортер	Комментарий
	im.astralinux.ru	443	HTTPS		Доступ к корпоративному Mattermost (опционально)
	Все объекты мониторинга	9100	TCP	node-exporter	Сбор системных метрик
	Контроллеры FreeIPA/ALD-Pro	9888	TCP	freeipa-exporter	Сбор метрик FreeIPA/ALD-Pro
	Все объекты мониторинга	9558	TCP	systemd-exporter	Сбор метрик сервисов systemd
	Ноды Бреста/OpenNebula	9177	TCP	libvirt-exporter	Сбор метрик системы виртуализации
	Хост для мониторинга Бреста/OpenNebula	9621	TCP	one-exporter	Сбор метрик через API Бреста/OpenNebula. Может быть одна из нод Бреста/OpenNebula либо выделенный сервер
Все объекты мониторинга	Система мониторинга (сервер сбора данных)	30607 (для Kubernetes) 9102 (для Docker)	TCP		Точка доступа для сбора логов с объектов мониторинга

В зависимости от особенностей среды мониторинга или требований экспортеров, порты могут отличаться от указанных.

Для мониторинга ПК СВ Брест необходимо обеспечить доступность API Брест (как общего кластерного URL, так и URL менеджмент-серверов) для установленного one-exporter.

2.4 Требования к учётной записи

Для запуска системы в Kubernetes необходим пользователь с полным доступом до нужного namespace. Для использования утилит kubectl и helm необходимо иметь рабочую конфигурацию .kube/config для подключения к требуемому кластеру Kubernetes.

Для запуска системы в Docker Compose необходим пользователь состоящий в группе `docker`.

На объектах мониторинга нужны следующие права, в зависимости от типа экспортера:

- freeipa-exporter:
 - для установки и запуска сервиса необходимы root права
 - для получения данных необходима учетная запись с пользовательскими правами на контроллере домена
 - для получения данных по репликации данная учетная запись должна иметь роль с привилегией на чтение (Read Replication Agreements)
- one-exporter:
 - для установки и запуска сервиса необходимы root права
 - для получения данных необходима локальная учетная запись в системе Брест (Драйвер авторизации: соге, нельзя для этого использовать доменную учётную запись - ограничение Брест)
 - для проверки доступности веб-консоли Бреста необходимо создать обычного пользователя на контроллере домена.
- systemd-exporter:
 - для установки и запуска сервиса необходимы root права

4. libvirt-exporter:
 - для установки и запуска сервиса необходимы root права
5. node-exporter:
 - для установки и запуска сервиса необходимы root права

3 Установка и обновление платформы Астра Мониторинг

3.1 Установка и обновление Астра Мониторинг с использованием Docker Compose

Перед выполнением установки необходимо убедиться, что требования из пункта [2.2 Программные требования](#) выполнены.

1. Скачать и распаковать архив с подготовленным окружением для Docker Compose

```
> curl -sLo astra-monitoring.tgz
  https://dl.astralinux.ru/am/generic/compose/astra-monitoring-
  latest.tgz
> tar zxvf astra-monitoring.tgz
> cd astra-monitoring/
> ls -l --group-directories-first
admin-ui/
alertmanager/
clickhouse/
grafana/
postgresql/
prometheus/
vector/
victoria-metrics/
vmalert/
docker-compose.yml
```

2. Запустить приложение:

```

> docker compose up -d
[+] Running 14/14
 ✓ Network astra-monitoring_default Created
   0.1s
 ✓ Container am-injector Started
   0.3s
 ✓ Container clickhouse Started
   0.2s
 ✓ Container vmetrics Started
   0.2s
 ✓ Container postgresql Started
   0.1s
 ✓ Container alertmanager Started
   0.3s
 ✓ Container grafana Started
   0.3s
 ✓ Container vmaalert Started
   0.3s
 ✓ Container admin-back Started
   0.4s
 ✓ Container prometheus Started
   0.3s
 ✓ Container admin-ui Started
   0.3s
 ✓ Container pg-updater Started
   0.2s
 ✓ Container vector Started
   0.2s
 ✓ Container ch-updater Started
   0.2s

# проверить, что все компоненты запущены
> docker compose ps -a

```

Обновление платформы Астра Мониторинг в случае использования Docker Compose выполняется через скачивание новой версии `astra-monitoring.tgz` с дальнейшей распаковкой его в ту же директорию и повторного запуска команды `docker compose up -d`.

3.2 Установка и обновление Астра Мониторинг в кластер Kubernetes (Helm Chart)

Перед выполнением установки необходимо убедиться, что требования из пункта 2.2 выполнены.

1. Добавить helm репозиторий `astra-monitoring`.

```

helm repo add astra-monitoring https://dl.astralinux.ru/am/helm
helm repo update

```

2. Скачать последнюю опубликованную версию Helm Chart и распаковать его.

```

helm pull astra-monitoring/astra-icl-monitoring
tar zxvf astra-icl-monitoring-<version>.tgz
cd astra-icl-monitoring/

```

Работа с Helm Chart как с архивом обусловлено тем, что для конфигурирование отдельных компонент происходит с изменением файлов в Helm Chart.

3. Заполнить `values.yaml` и примените чарт.

```

environment: prod

# Надо ли создавать секреты, по умолчанию - Да
createGrafanaSecret: true
createClickhouseSecret: true
createPostgreSQLSecret: true

# Основной тип стораджа
storageClass: yc-network-hdd
# Тип стораджа для бекапов (требуется Read-Write-Many)
storageClassBackup: csi-s3

ingress:
# DNS-суффикс для публикации приложений
# Публикация будет выглядеть так: http(s)://grafana-
  <release_name><dnsSuffix>/
  dnsSuffix: .example.loc
  className: nginx
# Конфигурация https для Ingress
  tlsEnabled: False
  secretName: ingress-tls

# Описание конфигурации Clickhouse
clickhouse:
# Название и путь до образа
  image: clickhouse/clickhouse-server
# Версия образа
  tag: 23.8.2.7
# Размер хранилища под логи
  storageSize: 50Gi

# Включить запуск задачи по обновлению структуры бд
  db_update_enabled: "true"
# Название и путь до образа
  db_update_image: registry.astralinux.ru/am/ch-cluster-update
# Версия образа
  db_update_tag: 0.01

# Включить регулярный бекап бд
  db_backup_enabled: "true"
# Название и путь до образа
  db_backup_image: registry.astralinux.ru/am/ch-backup-db
# Версия образа
  db_backup_tag: 0.21
# Количество хранимых бекапов
  db_backup_retention_days: 5
# Таймаут для подключения к экземпляру бд
  db_management_timeout: 5
# Порт для подключения к экземпляру бд
  db_management_port: 9000

# Количество шардов Clickhouse
  shards: 3

# Описание конфигурации Clickhouse Keeper
clickhousekeeper:
# Название и путь до образа
  image: clickhouse/clickhouse-keeper
# Версия образа
  tag: latest

# Описание конфигурации Clickhouse Proxy
clickhouseproxy:
# Название и путь до образа
  image: contentsquareplatform/chproxy
# Версия образа

```

```

tag: v1.24.0-amd64

# Описание конфигурации PostgreSQL
postgresql:
  # Название и путь до образа
  image: postgres
  # Версия образа
  tag: 15.4-alpine3.18
  # Размер хранилища под конфигурацию системы мониторинга
  storageSize: 10Gi

  # Включить запуск задачи по обновлению структуры бд
  db_update_enabled: "true"
  # Название и путь до образа
  db_update_image: registry.astralinux.ru/am/pg-update-db
  # Версия образа
  db_update_tag: "0.18"

  # Включить регулярный бекап бд
  db_backup_enabled: "true"
  # Таймаут для подключения к экземпляру бд
  db_management_timeout: 5

# Параметры запуска WebUI
admin_ui:
  # По умолчанию Ingress хост динамически создаётся по шаблону
  #   admin-ui-<Release.Name><Values.ingress.dnsSuffix>
  # задание опции ingressPrefix меняет поведение, Ingress хост
  # будет создаваться по шаблону:
  #   <Values.admin_ui.ingressPrefix><Values.ingress.dnsSuffix>
  # ingressPrefix: admin-ui
  # user: admin
  front:
    image: registry.astralinux.ru/am/admin-console/admin-console-ui
    tag: latest
  back:
    image: registry.astralinux.ru/am/admin-console/admin-console-api
    tag: latest
  am_injector:
    image: registry.astralinux.ru/am/am-injector
    tag: latest
    port: 5007

alert_manager:
  # По умолчанию Ingress хост динамически создаётся по шаблону
  #   alertmanager-<Release.Name><Values.ingress.dnsSuffix>
  # задание опции ingressPrefix меняет поведение, Ingress хост
  # будет создаваться по шаблону:
  #   <Values.alert_manager.ingressPrefix><Values.ingress.dnsSuffix>
  # ingressPrefix: alertmanager
  mattermost:
    api_url: https://link-to-api-mattermost/
    channel: <Alerts channel>
    send_resolved: True

# Параметры запуска Grafana
grafana:
  # По умолчанию Ingress хост динамически создаётся по шаблону
  #   grafana-<Release.Name><Values.ingress.dnsSuffix>
  # задание опции ingressPrefix меняет поведение, Ingress хост
  # будет создаваться по шаблону:
  #   <Values.grafana.ingressPrefix><Values.ingress.dnsSuffix>
  # ingressPrefix: grafana
  # Размер хранилища данных для Grafana
  storageSize: 5Gi
  # Название и путь до образа

```

```

image: grafana/grafana-oss
# Версия образа
tag: 9.5.2
# Уникальный ID источника данных Prometheus
prometheus_uid: 'gOyMaiEVz'
# Уникальный ID для источника данных ClickHouse
clickhouse_uid: 'P53F4937A2B5B9367'

# Дополнительные переменные окружения для запуска
extraEnv:
- name: GF_USERS_ALLOW_SIGN_UP
  value: 'false'
- name: GF_USERS_DEFAULT_THEME
  value: 'dark'
- name: GF_EXPLORE_ENABLED
  value: 'true'
- name: GF_ALERTING_ENABLED
  value: 'false'
- name: GF_UNIFIED_ALERTING_ENABLED
  value: 'true'
- name: GF_INSTALL_PLUGINS
  value: 'grafana-clickhouse-datasource 2.1.1'

# Путь и версия Grafana Plugin
plugin:
  version: 'v0.0.12'
  path: 'https://artifactory.astralinux.ru:443/artifactory/aicl-
generic/grafana-plugin/$GRAFANA_PLUGIN_VERSION/grafana-plugin.tar.gz'

ldap:
  host: dc01.example.loc
  port: 389
  bind:
    # DN для аутентификации в домене
    dn: uid=sa-grafana,cn=users,cn=accounts,dc=example,dc=loc
    # Пароль для аутентификации
    password: password
    # Базовый DN для поиска пользователей
    search_dns: cn=accounts,dc=example,dc=loc
    viewer_group_dn: '*'

# Конфигурация Vector'a на стороне системы мониторинга
vector:
  image: timberio/vector
  # tag: latest-alpine
  tag: 0.33.0-alpine
  #
  # Номер порта, который будет опубликован на нодах k8s
  # см. ограничения https://kubernetes.io/docs/concepts/services-
networking/service/#type-nodeport
  # port: 30607
  #
  service:
    type: NodePort
    ports:
      - port: 9102
        targetPort: 9102
        name: vector
        nodePort: 30607
      - port: 9090
        targetPort: 9090
        name: promremote

# Конфигурация Victoria Metrics
victoria_metrics:
  # По умолчанию Ingress хост динамически создаётся по шаблону

```

```

# vmetrics-<Release.Name><Values.ingress.dnsSuffix>
# задание опции ingressPrefix меняет поведение, Ingress хост
# будет создаваться по шаблону:
# <Values.victoria_metrics.ingressPrefix><Values.ingress.dnsSuffix>
# ingressPrefix: victoria_metrics

# Конфигурация Prometheus
prometheus:
# По умолчанию Ingress хост динамически создаётся по шаблону
# prometheus-<Release.Name><Values.ingress.dnsSuffix>
# задание опции ingressPrefix меняет поведение, Ingress хост
# будет создаваться по шаблону:
# <Values.prometheus.ingressPrefix><Values.ingress.dnsSuffix>
# ingressPrefix: prometheus

# Название, путь и версия образа
image: prom/prometheus
tag: latest
# Размер хранилища данных для Prometheus
storageSize: 5Gi
tsdb:
# Длительность хранения метрик
retention: 120h
# URL для автообнаружения конфигурации (discovery)
sdPath: '/api/endpoints?format=prometheus'
# Описание таргетов для сбора данных Prometheus
targets: []
# Примеры задания объектов мониторинга
# FreeIPA:
# - target: ['10.10.10.5:9888']
# labels:
# hostname: dc01.example.loc
# group: freeipa-example
# - target: ['10.10.10.10:9888']
# labels:
# hostname: dc02.example.loc
# group: freeipa-example
# Linux:
# - target: ['10.10.10.5:9100']
# labels:
# hostname: dc01.example.loc
# group: freeipa-example
# - target: ['10.10.10.10:9100']
# labels:
# hostname: dc02.example.loc
# group: freeipa-example
# - target: ['10.10.11.2:9100']
# labels:
# hostname: node1.brest.example.loc
# group: opennebula
# - target: ['10.10.11.3:9100']
# labels:
# hostname: node2.brest.example.loc
# group: opennebula
# - target: ['10.10.11.4:9100']
# labels:
# hostname: node3.brest.example.loc
# group: opennebula
# - target: ['10.10.11.1:9100']
# labels:
# hostname: brest.example.loc
# group: brest
# OpenNebula:
# - target: ['10.10.11.1:9621']
# labels:
# hostname: brest.example.loc

```

```
#       group: brest
# PostgreSQL:
#   - target: ['10.10.11.1:9187']
#     labels:
#       hostname: brest.example.loc
#       group: brest
# Libvirt:
#   - target: ['10.10.11.2:9177']
#     labels:
#       hostname: node1.brest.example.loc
#       group: opennebula
#   - target: ['10.10.11.3:9177']
#     labels:
#       hostname: node2.brest.example.loc
#       group: opennebula
#   - target: ['10.10.11.4:9177']
#     labels:
#       hostname: node3.brest.example.loc
#       group: opennebula
```

values.yaml

```
helm upgrade <name> ./ -n <k8s_namespace> --install --set
grafana.password="<grafana_password>" --set
clickhouse.password="<clickhouse_password>" --set
postgresql.password="<postgresql_password>" --set
adminui.password="<admin_ui_password>" --values ./values.yaml
```

4. Проверить, что все компоненты запущены.


```
> helm list -n <k8s_namespace>
NAME                                NAMESPACE          REVISION
UPDATED                             STATUS              CHART
APP VERSION
mon-dev                             astra-monitoring   1
2023-11-02 07:46:44.834388279 +0000 UTC deployed  astra-
icl-monitoring-0.0.12              0.0.12

> kubectl get pods -n <k8s_namespace>
NAME                                READY   STATUS    RESTARTS   AGE
mon-dev-admin-ui-6dfdbb5974-k29k2   2/2     Running   0           6d1h
mon-dev-alertmanager-7dd48549f9-6b84b 1/1     Running   0           20d
mon-dev-am-injector-686bd6bffb-xszpq  1/1     Running   0           20d
mon-dev-clickhouse-1-f4cfd5cbf-t7r8k  1/1     Running   1 (6h51m ago) 6h53m
mon-dev-clickhouse-2-6b66c87957-497vj  1/1     Running   0           6h53m
mon-dev-clickhouse-3-7c988f79cc-w4vck  1/1     Running   1 (6h52m ago) 6h53m
mon-dev-clickhouse-keeper-db9b7cf66-8nkjk 1/1     Running   0           42d
mon-dev-clickhouse-proxy-5785b6c659-wnrph 1/1     Running   0           41d
mon-dev-clickhouse-update-job-92j62    0/1     Completed 0           6h53m
mon-dev-grafana-74bd4b4d97-6qpch      1/1     Running   0           6h52m
mon-dev-postgresql-686b945c4d-qx6xt    1/1     Running   0           34d
mon-dev-postgresql-update-job-ppflv    0/1     Completed 0           6h51m
mon-dev-prometheus-85859b7cf8-t7w9f   1/1     Running   0           7d5h
mon-dev-vector-6f8c54675b-9cdmm       1/1     Running   0           7d5h
mon-dev-vmalert-6c89c886d7-2z8vb      1/1     Running   0           43d
mon-dev-vmetrics-69bf697cbc-q5kbr     1/1     Running   0           43d
```

Обновление выполняется повторением шага 2 и 3, разархивирование должно происходить в тот же каталог, откуда выполнялась установка.

3.3 Установка базы данных ClickHouse

Установка системы в кластер Kubernetes (3.2) или с использованием Docker Compose (3.1) включает в себя разворачивание актуальной версии базы данных. Возможность установки и использования внешней базы данных Clickhouse будет доработана в следующей версии системы.

3.4 Установка и обновление клиентской части для мониторинга продуктов ПАО Группа Астра

Здесь и далее установка пакетов описана для Astra Linux 1.7.

3.4.1 Установка Vector

Установка одинакова для мониторинга любого из продуктов ПАО Группа Астра.

Допускается использовать более новую версию Vector, но не старше версии 0.33 (изменение формата конфигурационных файлов).

```
# Установить DEB-пакет
> curl -sLo /tmp/vector_0.28.2-1_amd64.deb
'https://packages.timber.io/vector/0.28.2/vector_0.28.2-1_amd64.deb'
> sudo dpkg -i /tmp/vector_0.28.2-1_amd64.deb
> rm -f /tmp/vector_0.28.2-1_amd64.deb

# Внести изменения в файл /lib/systemd/system/vector.service для того, чтобы:
# Vector запускался с root правами. Необходимо закомментировать строчки User и Group
# в разделе Service.
# Vector писал логи в файл (для отделения логов Vector от логов основных систем).
> sudo vi /lib/systemd/system/vector.service

#/lib/systemd/system/vector.service
...
[Service]
#User=vector
#Group=vector
StandardOutput=file:/var/log/astra-monitoring/vector.log
ExecStartPre=/usr/bin/vector validate
...

# Создать правила для ротации файлов логов Vector.
> sudo vi /etc/logrotate.d/astra-monitoring

# /etc/logrotate.d/astra-monitoring
/var/log/astra-monitoring/*.log {
    daily
    rotate 3
}

# Измените файл, чтобы указать для Vector директорию, которая будет содержать правила
# обработки.
> sudo vi /etc/default/vector

# /etc/default/vector
VECTOR_CONFIG_TOML=/etc/vector/conf.d/*.toml
VECTOR_LOG=warn

# Создайте (если директория создана, то предварительно удалите все файлы toml)
# директорию /etc/vector/conf.d/ и скопируйте предоставленные конфигурационные файлы
# (vector.toml и все *.toml) по пути /etc/vector/conf.d/.
> sudo mkdir /etc/vector/conf.d/

# Создайте директорию /var/log/astra-monitoring для хранения логов Vector.
> sudo mkdir /var/log/astra-monitoring

# Внести изменение в конфигурационный файл вектора, указав принимающий логи сервер и
# порт
> sudo vi /etc/vector/conf.d/vector.toml

#/etc/vector/conf.d/vector.toml
...
[sinks.vector_cached]
type = "vector"
inputs = [ "aldpro_*", "vector_internal_logs_transformed" ]
```

```
address = "<IP_address>:<port>"

# Выполните команду reload для systemd.
> sudo systemctl daemon-reload

# Перезапустите сервис vector:
> sudo systemctl restart vector.service
```

3.4.2 Установка node-exporter

Node-exporter устанавливается на все хосты, независимо от мониторинга продуктов ПАО Группа Астра.

Пакет node-exporter может быть установлен из репозитория:

```
> sudo apt install prometheus-node-exporter
```

При размещении сервера с установленным node-exporter за сетевым экраном, NAT или другим подобным компонентом, необходимо опубликовать порт 9100/TCP и убедиться в его доступности для системы мониторинга. Если серверов несколько, то порты могут быть опубликованы, например, как 19100, 29100 и т.п.

3.4.3 Установка systemd-exporter

Systemd-exporter устанавливается на все хосты, независимо от мониторинга продуктов ПАО Группа Астра.

```
# Скачиваем опубликованный архив
> curl -sLo systemd-exporter.tgz https://dl.astralinux.ru/am/generic/systemd-
exporter/latest/systemd-exporter.tgz
> tar zxvf systemd-exporter.tgz
> cd ./systemd-exporter/

# Копируем исполняемый файл в каталог
> sudo cp ./systemd-exporter /usr/local/bin/
> sudo chmod 755 /usr/local/bin/systemd_exporter
> sudo chown root:root /usr/local/bin/systemd_exporter

# Копируем конфигурацию сервиса в системный каталог
> sudo cp ./systemd-exporter.service /usr/lib/systemd/system/systemd-exporter.service

# Редактируем параметры сервиса (описание параметров далее по тексту)
> vi ./systemd-exporter.conf
> sudo cp ./systemd-exporter.conf /etc/systemd-exporter.conf
```

Изменить конфигурационный файл с названием /etc/systemd-exporter.yml со следующим содержанием, изменив в нем параметры:

```
Services:
- includes: .*\.service # выбор всех сервисов
General:
  Interval: 60 # интервал опроса 60 секунд
```

Пример конфигурационного файла:

```

Services:
- includes: (dirsrv|gssproxy|httpd|apache2|certmonger|ipa-dns|ipa-otpd|ipa-
custodia|krb5-admin|krb5-kdc|pki-tomcat|sssd|bind9) .*
- excludes: .*\.target
labels:
- product: ald-pro
  subsystem: freeipa
  system: astra
General:
Interval: 60

```

3.4.4 Установка клиентской части для мониторинга ALD Pro / FreeIPA

Клиентская часть мониторинга ALD Pro / FreeIPA состоит из нескольких компонентов.

Установка freeipa-exporter

Для установки необходимо скачать и распаковать архив с настройками и экспортером:

```

# Скачиваем опубликованный архив
> curl -sLo freeipa-exporter.tgz https://dl.astralinux.ru/am/generic/freeipa-
exporter/latest/freeipa-exporter.tgz
> tar zxvf freeipa-exporter.tgz
> cd ./freeipa-exporter/

# Копируем исполняемый файл в каталог
> sudo cp ./freeipa-exporter /usr/local/bin/
> sudo chmod 755 /usr/local/bin/freeipa-exporter
> sudo chown root:root /usr/local/bin/freeipa-exporter

# Копируем конфигурацию сервиса в системный каталог
> sudo cp ./freeipa-exporter.service /usr/lib/systemd/system/freeipa-exporter.service

# Редактируем параметры сервиса (описание параметров далее по тексту)
> vi ./freeipa-exporter.conf
> sudo cp ./freeipa-exporter.conf /etc/freeipa-exporter.conf

# Обновляем конфигурацию и запускаем сервис
> sudo systemctl enable freeipa-exporter.service
> sudo systemctl start freeipa-exporter.service
> sudo systemctl status freeipa-exporter.service

```

Редактирование файла параметров сервиса:

```

USER_STR:
- Username: <user name>
  User_passwd: <user password>
  Users_dn: <user DN string>

```

/etc/ipahealthcheck-exporter.conf

В FreeIPA необходимо создать пользователя (пункт "2.4 Требования к учётной записи") и подставить его имя вместо <user name> (например, testuser) . Пароль зашифровать base64 и заменить в поле <user passwd>. В User_dn поместить правильный dn для пользователей (например, cn=users,cn=accounts,dc=example,dc=loc).

```

DNScheck:
- DNSserver: 77.88.8.8 (адрес DNS сервера для отправки запросов.
  Может быть localhost для проверок локального DNS)
  hostname: artifactory.astralinux.ru (FQDN имя сервера, для которого необходимо
  получить ip-адрес)
  hostip: 51.250.55.40 (ip-адрес, который мы ожидаем получить в
  результате DNS-запроса. Может быть localhost для проверок локального DNS, в этом
  случае сравнивается со всеми ip-адресами сервера)
  type: external (тип проверки - external для внешних DNS-
  серверов и internal для локальных)

- DNSServer: localhost
  hostname: artifactory.astralinux.ru
  hostip: 51.250.55.40
  type: internal

- DNSServer: localhost
  hostname: dc01.example.loc
  hostip: localhost
  type: internal

```

/etc/ipahealthcheck-exporter.conf

Для проверок DNS необходимо указать требуемые параметры в разделе DNScheck. Проверок может быть несколько для каждого из типов external/internal

Конфигурация сборщика логов

Приложение Vector должно быть установлено на всех необходимых серверах:

```

# Скачиваем опубликованный архив с конфигурацией Vector для ALD Pro
> curl -sLo freeipa-vector.tgz https://dl.astralinux.ru/am/generic/freeipa-
  exporter/latest/freeipa-vector.tgz
> tar zxvf freeipa-vector.tgz
> cd ./freeipa-vector/

# Копируем конфигурационные файлы в каталог Vector
> sudo cp *.toml /etc/vector/conf.d/

# Обновляем конфигурацию и запускаем сервис
> sudo systemctl restart vector.service

```

3.4.5 Установка клиентской части для мониторинга ПК СВ Брест

Клиентская часть мониторинга ПК СВ Брест состоит из нескольких компонентов.

Установка one-exporter

Установка one-exporter выполняется на **одном** сервере, у которого есть сетевой доступ к API ПК СВ Брест и который будет доступен для системы мониторинга напрямую, либо через публикацию порта экспортера. Для установки также требуется создать сервисную учетную запись в Брест с использованием драйвера авторизации "core".

Для установки необходимо скачать и распаковать архив с настройками и экспортером:

```

# Скачиваем опубликованный архив
> curl -sLo one-exporter.tgz https://dl.astralinux.ru/am/generic/one-
exporter/latest/one-exporter.tgz
> tar zxvf one-exporter.tgz
> cd ./one-exporter/

# Копируем исполняемый файл в каталог
> sudo cp ./one-exporter /usr/local/bin/
> sudo chmod 755 /usr/local/bin/one-exporter
> sudo chown root:root /usr/local/bin/one-exporter

# Копируем конфигурацию сервиса в системный каталог
> sudo cp ./one-exporter.service /usr/lib/systemd/system/one-exporter.service

# Редактируем параметры сервиса (описание параметров далее по тексту)
> vi ./one-exporter.conf
> sudo cp ./one-exporter.conf /etc/one-exporter.conf

```

Изменить конфигурационный файл с названием `/etc/one-exporter.yml` со следующим содержимым, изменив в нем параметры, отмеченные `<.>` на свои:

```

# credentials to access OpenNebula
user: <one_api_user>
password: <one_api_password>

# OpenNebula frontend endpoint
# an empty endpoint will default to http://localhost:2633/RPC2
# endpoint:

endpoint: <one_api_url>

# frequency to retrieve metrics in seconds. defaults to 60.
# interval: 60

# host and port to run the exporter on
host: 0.0.0.0
port: 9621

# exporter uri to publish on. defaults to /metrics
# path: /metrics

```

где `<one_api_user>` - имя сервисной учетной записи API Брест, созданной ранее (пункт "2.4 Требования к учётной записи"),

`<one_api_password>` - пароль учетной записи, закодированный в base64,

`<one_api_url>` - адрес API Брест.

Затем разрешить запуск сервиса после перезагрузки и запустить сервис:

```

sudo systemctl enable one-exporter.service
sudo systemctl start one-exporter.service
sudo systemctl status one-exporter.service

```

При размещении сервера с установленным `one-exporter` за сетевым экраном, NAT или другим подобным компонентом, необходимо опубликовать порт 9621/TCP и убедиться в его доступности для системы мониторинга.

Установка `libvirt-exporter`

Установка `libvirt-exporter` выполняется на всех серверах-гипервизорах Брест с работающим компонентом `libvirt`.

Для установки необходимо скачать и распаковать архив с настройками и экспортером:

```

# Скачиваем опубликованный архив
> curl -sLo libvirt-exporter.tgz https://dl.astralinux.ru/am/generic/libvirt/libvirt-
exporter.tgz
> tar zxvf libvirt-exporter.tgz
> cd ./libvirt-exporter/

# Копируем исполняемый файл в каталог
> sudo cp ./libvirt_exporter_improved /usr/local/bin/
> sudo chmod 755 /usr/local/bin/libvirt_exporter_improved
> sudo chown root:root /usr/local/bin/libvirt_exporter_improved

# Копируем конфигурацию сервиса в системный каталог
> sudo cp ./libvirt-exporter.service /usr/lib/systemd/system/libvirt-exporter.service

# Разрешаем и включаем сервис
> sudo systemctl enable libvirt-exporter.service
> sudo systemctl start libvirt-exporter.service
> sudo systemctl status libvirt-exporter.service

```

При размещении сервера с установленным libvirt-exporter за сетевым экраном, NAT или другим подобным компонентом, необходимо опубликовать порт 9177/TCP и убедиться в его доступности для системы мониторинга. Если серверов несколько, то порты могут быть опубликованы, например, как 19177, 29177 и т.п.

Конфигурация сборщика логов

Приложение Vector должно быть установлено на всех необходимых серверах (см. 3.4.1)

```

# Скачиваем опубликованный архив с конфигурацией Vector для Бреста
> curl -sLo one-vector.tgz https://dl.astralinux.ru/am/generic/one-
exporter/latest/one-vector.tgz
> tar zxvf one-vector.tgz
> cd ./one-vector/

# Копируем конфигурационные файлы в каталог Vector
> sudo cp *.toml /etc/vector/conf.d/

# Обновляем конфигурацию и запускаем сервис
> sudo systemctl restart vector.service

```

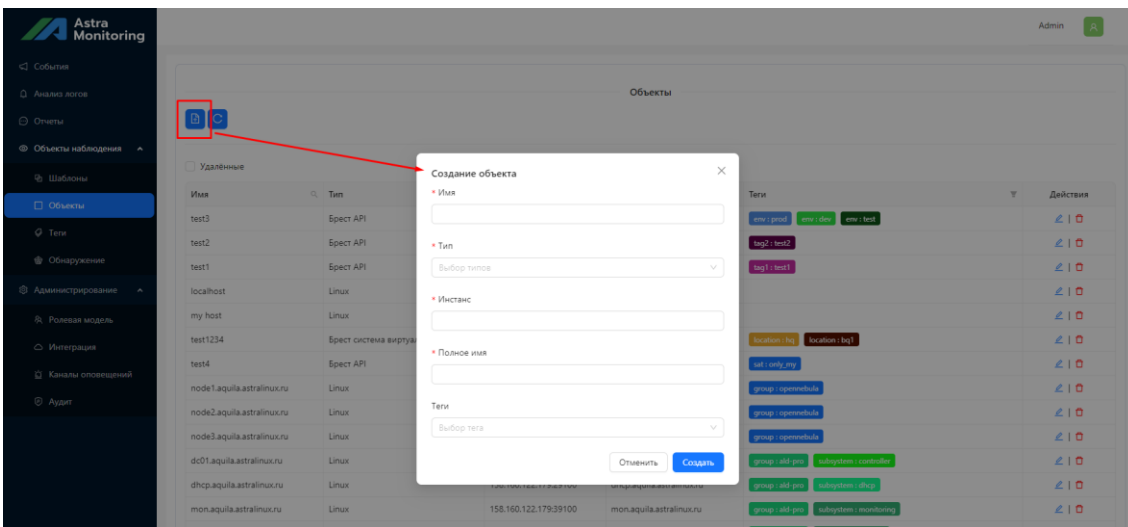
4 Настройка платформы Астра Мониторинг

4.1 Добавление объектов мониторинга

Добавление объекта на платформу мониторинга не влечет за собой автоматическую установку экспортеров.

Необходимо заранее убедиться, что все требуемые экспортеры уже установлены внутри объекта, запущены и работают, обеспечивая сбор и выдачу соответствующих метрик, а также доступны для системы мониторинга.

Добавление объектов мониторинга производится в пользовательском web-интерфейсе, раздел Объекты наблюдения → Объекты → кнопка добавления нового объекта



В открывшейся форме необходимо ввести:

Параметр	Описание
1 Имя	Имя объекта мониторинга, идентифицирующее объект в веб-интерфейсе системы
2 Тип	Тип объекта - некий ожидаемый набор метрик, характерный для заданного продукта, например, системы виртуализации, контроллера домена или системных метрик Linux. Необходим для применения к объекту различных политик, специфичных для этого типа
3 Инстанс	Адрес объекта (exporter) в формате prometheus instance , с номером порта, для получения метрик системой мониторинга
4 Полное имя	Имя объекта, которым будут маркироваться получаемые метрики и которое будет в дальнейшем использоваться при работе с панелями метрик и т.д. В общем случае, соответствует hostname или fqdn хоста
5 Теги	Это набор меток, необходимых для дополнительной идентификации объекта, например, имя группы, подсистемы или компонента, к которым относится данный объект мониторинга. Этими тегами будут промаркированы полученные от объекта метрики. Тегов может быть несколько. Некоторые теги, такие, как group или subsystem (ALD Pro), используются для корректной подстановки объекта в специфичных для этого продукта панелях метрик.

После ввода всех необходимых параметров необходимо нажать кнопку "Создать", в результате чего новый созданный объект появится в общем списке объектов мониторинга.

Из итогового списка объектов мониторинга затем формируется список targets в формате json, включающий в себя полное имя (hostname), адрес инстанса (instance) и теги (labels) каждого объекта мониторинга. Этот список доступен по специальному адресу <admin-ui-url>/api/endpoints?format=prometheus. Prometheus обращается к указанному адресу, считывает список объектов и, посредством механизма [HTTP Service Discovery](#), добавляет их в свою конфигурацию, после чего начинает опрос и получение метрик от объектов мониторинга.

Управление объектами мониторинга в рамках подхода IaC (Infrastructure-as-Code) не поддерживается на данном этапе и будет рассмотрено в следующих релизах

4.2 Настройка триггеров для создания событий

Правила триггеров для создания событий описаны в конфигурационных файлах:

- в директории `./alert-rules/` в случае установки в кластер Kubernetes (см. Установка и обновление Астра Мониторинг в кластер Kubernetes).
- в директории `./vmalert/config/` в случае установки в виде Docker Compose (см. Установка и обновление Астра Мониторинг с использованием Docker Compose).

Файлы логически разделены по продуктам, например, `ald-pro.yaml` или `brest.yaml`.

Правила описываются в формате YAML следующим образом:

```
groups:
  - name: ald-pro
    interval: 30s
    concurrency: 2
    rules:
      - alert: IPATestUserConnection
        expr: |
          ipa_user_connect{} == 0
        for: 5m
        labels:
          severity: critical
          team: ipa_support
        annotations:
          summary: "Server {{ $labels.hostname }} has an issue with test user authentication"
          description: "Test user {{ $labels.user_check }} can't connect to domain on server {{ $labels.hostname }} ({{ $labels.instance }})"

      - alert: IPAUserConnectionTime
        expr: |
          avg_over_time(ipa_time_for_connect[5m]) > 2 *
          avg_over_time(ipa_time_for_connect[5m] offset 5m)
        labels:
          severity: warning
          team: ipa_support
        annotations:
          summary: "Server {{ $labels.hostname }} ({{ $labels.instance }}) has highly increased test user connection time ({{ humanize $value }} ms.)"
          description: "Test user {{ $labels.user_check }} has an issue with very long test user connection time ({{ humanize $value }} ms.) on server {{ $labels.hostname }} ({{ $labels.instance }})"
```

где:

- `name` - имя группы правил
- `interval` - интервал обработки правил группы
- `alert` - имя триггера

- expr - условие для срабатывания триггера в формате [PromQL](#)
- for - минимальная длительность состояния перед срабатыванием триггера
- labels - дополнительные лейблы для триггера:
 - severity - уровень критичности
 - team - маршрут для отправки уведомлений о срабатывании триггера (например, команда поддержки системы/продукта)
- annotations - дополнительные информационные сообщения для получателей сообщений:
 - summary - краткое описание проблемы в заголовке сообщения
 - description - общее описание проблемы в тексте сообщения

Для добавления нового правила к уже существующей группе, необходимо определить имя триггера, условия для его срабатывания, критичность и получателя сообщений о срабатывании триггера. Далее добавить описание триггера в формате YAML в список **rules** требуемой группы правил, например, группы ald-pro в файле `./alert-rules/ald-pro.yaml` (`./vmalert/config/ald-pro.yaml`)

Для создания нового набора правил необходимо создать новый файл в директории `./alert-rules/` (`./vmalert/config/`) с требуемой группой (например, файл `rupost.yaml` с группой правил `rupost`), описать в группе требуемый набор правил и сохранить файл.

После окончания добавления или редактирования правил, необходимо обновить компоненты системы с помощью инструментов `docker-compose` / `helm-chart` в соответствии с описанием в разделах Установка и обновление Астра Мониторинг с использованием `Docker Compose` и Установка и обновление Астра Мониторинг в кластер `Kubernetes`.

4.3 Настройка оповещений

Правила оповещений о событиях описаны в конфигурационном файле:

- в директории `./templates/alertmanager-cm.yaml` в случае установки в кластер `Kubernetes` (см. Установка и обновление Астра Мониторинг в кластер `Kubernetes`).
- в директории `./alertmanager/config/alertmanager.yml` в случае установки в виде `Docker Compose` (см. Установка и обновление Астра Мониторинг с использованием `Docker Compose`).

Настройки описываются в формате YAML, для примера с интеграцией с `Mattermost` (необходимо подставить `<webhookid>` и `<channel-name>`)

```

global:
templates:
- '/etc/alertmanager-templates/*.tmpl'
route:
  receiver: alert-mmmost
  group_by: ['alertname', 'hostname', 'severity', 'metric']
  group_wait: 3m
  group_interval: 5m
  repeat_interval: 1d
  routes:
  - receiver: alert-mmmost
    matchers:
    - team = ipa_support

receivers:
- name: alert-mmmost
  slack_configs:
  - api_url: https://im.astralinux.ru/hooks/<webhookid>
    channel: '<channel-name>'
    send_resolved: true
    title: '{{ template "slack.title" . }}'
    color: '{{ template "slack.color" . }}'
    text: '{{ template "slack.text" . }}'
  webhook_configs:
  - url: 'http://am-injector:5007/receive/alertmanager'

inhibit_rules:
- source_matchers: [severity="critical"]
  target_matchers: [severity="warning"]
  # Apply inhibition if the alertname is the same.
  # CAUTION:
  #   If all label names listed in `equal` are missing
  #   from both the source and target alerts,
  #   the inhibition rule will apply!
  equal: [alertname, hostname, instance]

```

Более подробно о возможностях настройки подсистемы оповещения можно найти в официальной документации <https://prometheus.io/docs/alerting/latest/configuration/>

Конфигурация `webhook_configs` должна быть сохранена для наполнения базы событий и отображение актуальной информации в Admin UI.

После внесения изменений, необходимо обновить компоненты системы с помощью инструментов `docker-compose` / `helm-chart` в соответствии с описанием в разделах Установка и обновление Астра Мониторинг с использованием Docker Compose и Установка и обновление Астра Мониторинг в кластер Kubernetes.

5 Резервное копирование баз данны

5.1 Резервное копирование базы данных ClickHouse

Автоматическое резервное копирование реализовано только при установке в кластер Kubernetes.

В платформе Астра Мониторинг при установке в кластер Kubernetes предусмотрено создание автоматических резервных копий базы данных Clickhouse (CronJob).

По умолчанию, резервные копии создаются каждый день в 2:00 UTC. Каждые 5 дней создаётся полная резервная копия, в остальные дни создаётся инкрементальная копия.

Резервная копия сохраняется в PVC Kubernetes "pvc-{{ \$.Release.Name }}-clickhouse-backup".

Параметры работы сервиса по созданию резервных копий задаются в файле values.yaml (Установка и обновление Астра Мониторинг в кластер Kubernetes):

```
...
# Описание конфигурации Clickhouse
clickhouse:
  ...
  # Включить регулярный бекап бд
  db_backup_enabled: "true"
  # Название и путь до образа
  db_backup_image: registry.astralinux.ru/am/ch-backup-db
  # Версия образа
  db_backup_tag: 0.21
  # Количество хранимых бекапов
  db_backup_retention_days: 5
  # Таймаут для подключения к экземпляру бд
  db_management_timeout: 5
  # Порт для подключения к экземпляру бд
  db_management_port: 9000
...
```

5.2 Резервное копирование базы данных PostgreSQL

Автоматическое резервное копирование реализовано только при установке в кластер Kubernetes.

В платформе Астра Мониторинг при установке в кластер Kubernetes предусмотрено создание автоматических резервных копий базы данных PostgreSQL (CronJob).

По умолчанию, резервные копии создаются каждый день в 2:00 UTC, при этом создаётся полная резервная копия (с использованием pg_basebackup) и хранятся 3 дня.

Резервная копия сохраняется в PVC Kubernetes "pvc-{{ \$.Release.Name }}-postgresql-backup".

Параметры работы сервиса по созданию резервных копий задаются в файле values.yaml (Установка и обновление Астра Мониторинг в кластер Kubernetes):

```
...
# Описание конфигурации PostgreSQL
postgresql:
  ...
  # Включить регулярный бекап бд
  db_backup_enabled: "true"
  # Таймаут для подключения к инстансу бд
  db_management_timeout: 5
...
```

6 Мониторинг продуктов ПАО Группа Астра

6.1 Мониторинг ОС Astra Linux

Для мониторинга системных показателей используется компонент node-exporter.

Экспортер так же используется на панелях с метриками продуктов, для отображения системных показателей (например, ЦПУ, память) рядом с продуктовыми для удобства анализа.

В стандартную панель с метриками node-exporter добавлена переменная group, для фильтрации объектов наблюдения по группам.

6.2 Мониторинг FreeIPA

№	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
1	Проверка возможности входа в ldap тестового пользователя	Метрика отрабатывает локальную попытку входа пользователя в контроллер домена.	FreeIPA state	User connection test	Пререквизиты: <ol style="list-style-type: none">Используется ipahealthcheck_exporter (github.com/ccin2p3/go-freeipa/freeipa).Создан технический пользователь в FreeipaПользователь прописан в ipahealthcheck-exporter.confПароль для пользователя должен быть зашифрован Base64 <p>Специальный технический пользователь локально подключается к ldap и происходит попытка получить статус этого пользователя. Если ldap возвращает ошибку, то метрика переходит в NOK и возвращает 0. Если попытка успешная то возвращает 1.</p>	ipa_user_connect
			FreeIPA state with history data	Test user connection		
2	Кол-во неуспешных попыток авторизации в ldap	Метрика подсчитывает кол-во неуспешных попыток аутентификации на контроллере домена.	FreeIPA state with history data"	Failed user login attempts	Метрика собирается из логов. Пререквизиты: <ol style="list-style-type: none">Используется сервис vectorВ /etc/vector/conf.d/freeipa.toml прописан путь к логу /var/log/dirsrv/*/accessТам же настроен парсинг этого лога:	

№	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
					<pre>(.*(?P<valid_cred>Invalid credentials))</pre> <p>После неуспешной попытки входа в логе появляется запись "Invalid credentials". Vector обрабатывает запись и пишет ее в БД с необходимыми labels.</p>	
3	Время (длительность) прохождения аутентификации тестовым пользователем	Метрика подсчитывает время, необходимое для проведения аутентификации пользователя	FreeIPA state	User connection time	<p>Пререквизиты:</p> <ol style="list-style-type: none"> 1. Используется ipahealthcheck_exporter (github.com/ccin2p3/go-freeipa/freeipa). 2. Создан технический пользователь в FreeIPA 3. Пользователь прописан в ipahealthcheck-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 <p>Специальный технический пользователь локально подключается к ldap и происходит попытка получить статус этого пользователя. Замеряется время прохождения этого процесса. Если происходит ошибка, то возвращается -1.</p>	ipa_time_for_connection
			FreeIPA state with history data	User connection time		
4	Статус репликации на контроллере домена	Метрика проверяет статус репликации между контроллерами	FreeIPA state	Replication	<p>Пререквизиты:</p> <ol style="list-style-type: none"> 1. Используется ipahealthcheck_exporter. 2. Пользователь должен иметь роль с привилегией на чтение Read Replication Agreements <p>Специальный технический пользователь локально подключается к ldap, делает запрос на статус репликации. Ответ анализируется на наличие настроенных репликаций к другим хостам, их статус и последнее время репликации между ними.</p>	ipa_ldap_replication_status
			FreeIPA state with history data"	Summary replication for each controller		ipa_last_ldap_replication_time
				Replication status		
				Last replication time (time from now)		
5	Кол-во заблокированных пользователей	Метрика подсчитывает кол-во выключенных пользователей	FreeIPA state	Disabled users	<p>Развернуть</p> <p>Пререквизиты:</p> <ol style="list-style-type: none"> 1. Используется ipahealthcheck_exporter (gopkg.in/ldap.v2). 	ipa_disabled_users
			FreeIPA state with history data	Amount of disabled users		

№	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
					<ol style="list-style-type: none"> 2. Создан технический пользователь в FreeIPA 3. Пользователь прописан в ipahealthcheck-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 <p>Специальный технический пользователь локально подключается к ldap и собирает данные обо всех пользователях, у которых параметр "nsAccountLock" = TRUE.</p>	
6	Проверка соответствия заданному dns имени - ip адресу при запросе через локальные dns настройки	Метрика проверяет работу DNS службы, выполняя преобразование заданного имени в ip-адрес (DNS запрос) в соответствии с локальными настройками DNS.	FreeIPA state	Internal DNS test	<p>Развернуть</p> <p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется ipahealthcheck_exporter. 2. В ipahealthcheck-exporter.conf прописаны необходимые параметры для DNS запроса: имя (fqdn), ожидаемый ip-адрес (может быть localhost), тип проверки (internal) <p>Экспортер делает запрос к внутреннему DNS-серверу для выяснения ip-адреса, соответствующего имени из запроса. Если ip адрес совпадает с ожидаемым, проверка пройдена.</p>	dns_internal_check
			FreeIPA state with history data	DNS check for expected IP address using local dns settings		
7	Проверка соответствия заданному dns имени - ip адресу при запросе через локальные dns настройки	Метрика проверяет работу DNS службы, выполняя преобразование заданного имени в ip-адрес (DNS запрос) с обращением к заданному внешнему серверу DNS	FreeIPA state	External DNS test	<p>Развернуть</p> <p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется ipahealthcheck_exporter. 2. В ipahealthcheck-exporter.conf прописаны необходимые параметры для DNS запроса: имя (fqdn), ожидаемый ip-адрес, адрес внешнего DNS-сервера, тип проверки (external) <p>Экспортер делает запрос к указанному в настройках внешнему DNS-серверу для выяснения ip-адреса, соответствующего имени из запроса. Если ip адрес совпадает с ожидаемым, проверка пройдена.</p>	dns_external_check
			FreeIPA state with history data	DNS check for expected IP address using external dns server		

№	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
8	Проверка доступности dns сервера контроллера домена путём выполнения локального dns запроса	Метрика проверяет доступность DNS сервера, указанного в локальных настройках	FreeIPA state with history data	DNS resolve check using local dns settings	<p>Развернуть</p> <p>Пререквизиты:</p> <ol style="list-style-type: none"> Используется ipahealthcheck_exporter. В ipahealthcheck-exporter.conf прописаны необходимые параметры для DNS запроса: имя (fqdn), ожидаемый ip-адрес (может быть localhost), тип проверки (internal) <p>Экспортер делает запрос к DNS-серверу указанному в локальных настройках для получения ответа на dns запрос. Ожидается ответ от dns сервера.</p>	dns_internal_resolve
9	Проверка доступности global dns путём выполнения dns запроса	Метрика проверяет доступность external DNS сервера	FreeIPA state with history data	DNS resolve check using query to external dns server	<p>Пререквизиты:</p> <ol style="list-style-type: none"> Используется ipahealthcheck_exporter. В ipahealthcheck-exporter.conf прописаны необходимые параметры для DNS запроса: имя (fqdn), ожидаемый ip-адрес, адрес внешнего DNS-сервера, тип проверки (external) <p>Экспортер делает запрос к внешнему DNS-серверу для получения ответа на dns запрос. Ожидается ответ от dns сервера.</p>	dns_external_resolve
10	Статус ntp (ошибок синхронизации с вышестоящими ntp серверами)	Метрика проверяет ошибки синхронизации контроллера с вышестоящими серверами времени.	FreeIPA state	NTP	<p>Пререквизиты:</p> <ol style="list-style-type: none"> Используется локальная служба chrony <p>Проверяется вывод команды <code>chronyc tracking</code>. Проверка осуществляется по полю: "Leap status". Статус "Normal" считается ОК и возвращается 0. Статус "Insert second" считается NOK и возвращается 1. Статус "Delete second" считается NOK и возвращается 2. Статус "Not synchronized" считается NOK и возвращается 3. Любой другой статус или ошибка получения данных возвращается 4.</p>	ipa_ntp_check
			FreeIPA state with history data	NTP Status		
11		Метрика проверяет	FreeIPA state"	Services state	Пререквизиты:	systemd_unit_state_id

№	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
	Статус служб Freeipa	состояние служб контроллера домена	FreeIPA state with history data	FreeIPA summary services state on all controllers FreeIPA services on \$hostname	<ol style="list-style-type: none"> Используется systemd_exporter. Список сервисов фильтруется в самом запросе. 	
12	Кол-во записей (изменений) в базу LDAP за единицу времени		FreeIPA state with history data	Changes in LDAP	<p>Метрика собирается из логов.</p> <p>Пререквизиты:</p> <ol style="list-style-type: none"> Используется сервис vector В /etc/vector/conf.d/freeipa.toml прописан путь к логу /var/log/dirsrv/*/access Там же настроен разбор этого лога: <pre>(.*conn=(?P<connection_id>[0-9]+) op=(?P<operation_id>[0-9]+) (?P<ldap_operation>[A-Z]+) (. * \$))</pre> <p>В логе собирается информация по типам операций ldap_operation. Все операции собираются, им присваивается labels и vector пишет все в БД. Метрика строится по следующим типам операций: ADD, DEL, MOD.</p>	
13	Кол-во логинов в web ui от пользователей с админ правами за единицу времени	Метрика собирает кол-во успешных входов пользователей уровня Администратор через веб консоль Freeipa	FreeIPA state	Admin UI logins	<p>Метрика собирается из логов.</p> <p>Пререквизиты:</p> <ol style="list-style-type: none"> Используется сервис vector В /etc/vector/conf.d/freeipa.toml прописан путь к логу /var/log/auth.log Там же настроен разбор этого лога: <pre>^(. *), (?P<username>[a-zA-Z0-9-_. /]+)@ [a-zA-Z0-9-_. _]+</pre> <p>Из лога собираются вход любого пользователя с меткой: "TGS_REQ". На уровне grafana делается выборка пользователей с именем, которое содержит "adm".</p>	

№	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
14	Кол-во событий уровня error во всех логах системы за единицу времени	Метрика собирает кол-во ошибок уровня error из предоставленных логов	FreeIPA state with history data	Errors in log files	<p>Метрика собирается из логов.</p> <p>Пререквизиты:</p> <ol style="list-style-type: none"> 1. Используется сервис vector 2. В <code>/etc/vector/conf.d/freeipa.toml</code> прописан путь к логам 3. Работает по всем логам, которые собираются на данном хосту <p>Все события уровня ERROR собираются, помечаются label "Severity" для дальнейшего анализа на стороне grafana.</p>	
15	Кол-во Search запросов на контроллере домена	Метрика собирает кол-во Search запросов к контроллеру	FreeIPA state with history data	Search requests in Ldap	<p>Метрика собирается из логов.</p> <p>Пререквизиты:</p> <ol style="list-style-type: none"> 1. Используется сервис vector 2. В <code>/etc/vector/conf.d/freeipa.toml</code> прописан путь к логу <code>/var/log/dirsrv/*/access</code> 3. Там же настроен разбор этого лога: <pre>(.*conn=(?P<connection_id>[0-9]+) op=(?P<operation_id>[0-9]+) (?P<ldap_operation>[A-Z]+) (. * \$))</pre> <p>В логе собирается информация по типам операций ldap_operation. Все операции собираются, им присваивается labels и vector пишет все в БД. Метрика строится по следующим типам операций: SRCH.</p>	
16	Нагрузка на процессор на контроллерах	Метрика показывает загрузку процессора на хостах контроллера домена	FreeIPA state with history data	Total CPU usage, % CPU	<p>Пререквизиты:</p> <ol style="list-style-type: none"> 1. Используется prometheus node exporter. <p>Стандартный способ сбора метрики с хостов основанный на стандартном node exporter. Метрика образовывается путем вычитания текущего времени использования ЦПУ и минуту назад.</p>	node_cpu_seconds_total
17	Использование памяти	Метрика показывает использование памяти на	FreeIPA state with history data	Total memory usage, %	<p>Пререквизиты:</p> <ol style="list-style-type: none"> 1. Используется prometheus node exporter. 	node_memory_MemTotal_bytes

№	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
		хостах контроллера домена		Memory Stack	Стандартный способ сбора метрики с хостов основанный на стандартном node exporter. Метрика образовывается путем вычитания из максимального количества ОЗУ всей использованной памяти.	node_memory_Buffers_bytes node_memory_MemFree_bytes node_memory_Cached_bytes

6.3 Мониторинг ALD Pro

	Наименование	Описание	Название подсистемы ALD-Pro	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
34	Проверка синхронизации времени	Метрика проверяет ошибки синхронизации контроллера с вышестоящими серверами времени.	Синхронизация времени	ALD-Pro time synchronization	NTP Status	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется локальная служба chrony <p>Проверяется вывод команды <code>chronyc tracking</code>. Проверка осуществляется по полю: <code>"Leap status"</code>. Статус <code>"Normal"</code> считается ОК и возвращается 0. Статус <code>"Insert second"</code> считается NOK и возвращается 1. Статус <code>"Delete second"</code> считается NOK и возвращается 2. Статус <code>"Not synchronized"</code> считается NOK и возвращается 3. Любой другой статус или ошибка получения данных - возвращается 4.</p>	ipa_ntp_check
28	Проверка работы ДНС через локальный запрос	Метрика проверяет работу DNS службы, выполняя преобразование заданного имени в ip-адрес (DNS запрос) в соответствии с локальными настройками DNS.	Разрешение имен	ALD-Pro name resolution	Internal DNS test	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется <code>ipahealthcheck_exporter</code>. В <code>ipahealthcheck-exporter.conf</code> прописаны необходимые параметры для DNS запроса: имя (<code>fqdn</code>), ожидаемый ip-адрес (может быть <code>localhost</code>), тип проверки (<code>internal</code>) <p>Экспортер делает запрос к внутреннему DNS-серверу для выяснения ip-адреса,</p>	dns_internal_check

	Наименование	Описание	Название подсистемы ALD-Pro	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
						соответствующего имени из запроса. Если ip адрес совпадает с ожидаемым, проверка пройдена.	
29	Проверка работы ДНС через запрос к внешнему ДНС	Метрика проверяет работу DNS службы, выполняя преобразование заданного имени в ip-адрес (DNS запрос) с обращением к заданному внешнему серверу DNS	Разрешение имен	ALD-Pro name resolution	External DNS test	<p>Пререквизиты:</p> <ol style="list-style-type: none"> 1. Используется ipahealthcheck_exporter. 2. В ipahealthcheck-exporter.conf прописаны необходимые параметры для DNS запроса: имя (fqdn), ожидаемый ip-адрес, адрес внешнего DNS-сервера, тип проверки (external) <p>Экспортер делает запрос к указанному в настройках внешнему DNS-серверу для выяснения ip-адреса, соответствующего имени из запроса. Если ip адрес совпадает с ожидаемым, проверка пройдена.</p>	dns_external_check
21	Доступность WEB портала ALD-Pro	Метрика показывает доступность web портала ALD-Pro	Портал управления	ALD-Pro management portal	Connect to WEB portal	<p>Пререквизиты:</p> <ol style="list-style-type: none"> 1. Используется ipahealthcheck_exporter <p>Измеряется доступность https странички портала</p>	ald_portal_check
9	Срок службы LDAP сертификата	Метрика показывает время, оставшееся до истечения срока действия LDAP сертификата ALD-Pro	Служба каталогов	ALD-Pro directory service	LDAP certificate expiration time	<p>Пререквизиты:</p> <ol style="list-style-type: none"> 1. Используется ipahealthcheck-exporter <p>Делается запрос openssl для получения времени истечения сертификата</p>	ald_ldap_cert_exp_check
10	Срок службы WEB сертификата портала ALD-Pro	Метрика показывает время, оставшееся до истечения срока действия WEB сертификата портала ALD-Pro	Служба каталогов	ALD-Pro directory service	WEB certificate expiration time	<p>Пререквизиты:</p> <ol style="list-style-type: none"> 1. Используется ipahealthcheck-exporter <p>Делается запрос openssl для получения времени истечения сертификата</p>	ald_web_cert_exp_check

	Наименование	Описание	Название подсистемы ALD-Pro	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
11	Статус службы krb5kdc	Метрика показывает состояние сервиса krb5-kdc	Служба каталогов	ALD-Pro directory service	Status of krb5kdc service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
12	Статус службы ipa-custodia	Метрика показывает состояние сервиса ipa-custodia	Служба каталогов	ALD-Pro directory service	Status of ipa-custodia service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
13	Статус службы apache2	Метрика показывает состояние сервиса httpd/apache2	Служба каталогов	ALD-Pro directory service	Status of apache2 service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
14	Статус службы dirsrv	Метрика показывает состояние сервиса dirsrv	Служба каталогов	ALD-Pro directory service	Status of directory service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
15	Статус службы bind9	Метрика показывает состояние сервиса bind9	Служба каталогов	ALD-Pro directory service	Status of named service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
16	Статус службы krb5-admin	Метрика показывает состояние сервиса krb5-admin	Служба каталогов	ALD-Pro directory service	Status of kadmind service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
17	Статус службы ipa-dnskeysyncd	Метрика показывает состояние сервиса ipa-dnskeysyncd	Служба каталогов	ALD-Pro directory service	Status of ipa-dnskeysyncd service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
18	Статус службы ipa-otpd.socket	Метрика показывает состояние сервиса ipa-otpd.socket	Служба каталогов	ALD-Pro directory service	Status of ipa-otpd service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
19	Статус службы isc-dhcp-server	Метрика показывает состояние сервиса isc-dhcp-server	Динамическая настройка узлов	ALD-Pro dynamic node configuration	Status of isc-dhcp-server service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
20	Статус службы smbд	Метрика показывает состояние сервиса smbд	Общий доступ к файлам	ALD-Pro file sharing	Status of samba service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
21	Статус службы apache2	Метрика показывает состояние сервиса httpd/apache2	Установка ОС по сети	ALD-Pro installing the OS over the network	Status of apache2 service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id

	Наименование	Описание	Название подсистемы ALD-Pro	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
22	Статус службы rabbitmq-server	Метрика показывает состояние сервиса rabbitmq-server	Установка ОС по сети	ALD-Pro installing the OS over the network	Status of rabbitmq-server service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
23	Статус службы postgresql	Метрика показывает состояние сервиса postgresql	Установка ОС по сети	ALD-Pro installing the OS over the network	Status of postgresql service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
24	Статус службы salt-master	Метрика показывает состояние сервиса salt-master	Портал управления	ALD-Pro management portal	Status of salt-master service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
25	Статус службы apache2	Метрика показывает состояние сервиса httpd/apache2	Портал управления	ALD-Pro management portal	Status of apache2 service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
26	Статус службы celery	Метрика показывает состояние сервиса celery	Портал управления	ALD-Pro management portal	Status of celery service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
27	Статус службы rabbitmq-server	Метрика показывает состояние сервиса rabbitmq-server	Портал управления	ALD-Pro management portal	Status of rabbitmq-server service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
28	Статус службы aldpro-mp-services	Метрика показывает состояние сервиса aldpro-mp-services	Портал управления	ALD-Pro management portal	Status of aldpro-mp-services service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
23	Использование памяти службами ALD-Pro	Метрика показывает использование памяти определенными службами	Портал управления	ALD-Pro management portal	Использование памяти процессам и служб	Пререквизиты: 1. Используется systemd exporter	systemd_unit_memory_usage
24	Статус службы zabbix-agent	Метрика показывает состояние сервиса zabbix-agent	Мониторинг	ALD-Pro monitoring	Status of zabbix-agent service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
25	Статус службы zabbix-server	Метрика показывает состояние сервиса zabbix-server	Мониторинг	ALD-Pro monitoring	Status of zabbix-server service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id
26	Статус службы apache2	Метрика показывает состояние	Мониторинг	ALD-Pro monitoring	Status of apache2 service	Пререквизиты: 1. Используется systemd exporter	systemd_unit_state_id

	Наименование	Описание	Название подсистемы ALD-Pro	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
		сервиса apache2					
27	Статус службы postgresql	Метрика показывает состояние сервиса postgresql	Мониторинг	ALD-Pro monitoring	Status of postgresql service	Прerequisites: 1. Используется systemd exporter	systemd_unit_state_id
30	Статус службы cups	Метрика показывает состояние сервиса cups	Печать	ALD-Pro printing system	Status of cups service	Прerequisites: 1. Используется systemd exporter	systemd_unit_state_id
31	Статус службы apache2	Метрика показывает состояние сервиса apache2	Репозитории программного обеспечения	ALD-Pro software repositories	Status of apache2 service	Прerequisites: 1. Используется systemd exporter	systemd_unit_state_id
32	Статус службы postgresql	Метрика показывает состояние сервиса postgresql	Репозитории программного обеспечения	ALD-Pro software repositories	Status of postgresql service	Прerequisites: 1. Используется systemd exporter	systemd_unit_state_id
33	Статус службы rabbitmq-server	Метрика показывает состояние сервиса rabbitmq-server	Репозитории программного обеспечения	ALD-Pro software repositories	Status of rabbitmq-server service	Прerequisites: 1. Используется systemd exporter	systemd_unit_state_id
35	Статус службы chrony	Метрика показывает состояние сервиса chrony	Синхронизация времени	ALD-Pro time synchronization	Status of chrony service	Прerequisites: 1. Используется systemd exporter	systemd_unit_state_id
22	Использование ЦПУ службами ALD-Pro	Метрика показывает использование ЦПУ определенными службами	Портал управления	ALD-Pro management portal	Использование ЦП процессам и служб, %	Прerequisites: 1. Используется systemd exporter 2. Используется node exporter	systemd_unit_cpu_usage node_cpu_seconds_total

6.4 Мониторинг ПК СВ Брест

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
1	Количество запущенных ВМ на выбранном кластере	Метрика показывает сколько ВМ находится в запущенном (running) состоянии в выбранном кластере	Brest Cluster info	Running VMs	Прerequisites: 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь	one_cluster_runningvms

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
					3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста Через API делается запрос всех запущенных VM на инстансе Бреста.	
2	Всего ЦПУ в кластере	Метрика показывает сколько всего ЦПУ может быть использовано в выбранном кластере	Brest Cluster info	CPUs total	Пререквизиты: 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста Через API делается запрос сколько всего ЦПУ на инстансе Бреста.	one_cluster_totalcpu
3	Количество использованных ЦПУ в кластере	Метрика показывает сколько всего ЦПУ используется в данный момент в выбранном кластере	Brest Cluster info	CPUs used	Пререквизиты: 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста Через API делается запрос сколько всего ЦПУ используется на инстансе Бреста.	one_cluster_cpuusage

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
4	Всего памяти в кластере	Метрика показывает сколько всего Memory в выбранном кластере	Brest Cluster info	Memory total	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос сколько всего Memory на инстансе Бреста.</p>	one_cluster_totalmem
5	Количество используемой памяти в кластере	Метрика показывает сколько всего Memory используется в выбранном кластере	Brest Cluster info	Memory used	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос сколько всего Memory используется на инстансе Бреста.</p>	one_cluster_memusage
6	Суммарный статус всех сервисов на хостах выбранного кластера	Метрика показывает суммарный статус всех заранее заданных сервисов по нодам выбранного кластера	Brest Cluster info	Services on nodes	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется systemd exporter 	systemd_unit_state_id
7	Суммарный статус всех сервисов на fronts выбранного кластера	Метрика показывает суммарный статус всех заранее заданных сервисов по фронтам выбранного кластера	Brest Cluster info	Services on fronts	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется systemd exporter 	systemd_unit_state_id

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
8	Количество использованного места в DataStore	Метрика показывает общий размер DataStore выбранного кластера и сколько места использовано.	Brest Cluster info	Datastores size	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос общего количества места на DataStore и сколько использовано на инстансе Бреста.</p>	one_ds_usedmb one_ds_totalmb
9	Распределение DataStore по кластерам	Метрика показывает распределение DataStore по выбранным кластерам	Brest Cluster info	Datastore uses on cluster	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос какие DataStore представлены на кластерах инстанса Бреста.</p>	one_ds_available
10	Количество используемого ЦПУ, %	Метрика показывает отношение использованных ЦПУ к всего имеющихся ЦПУ на выбранном кластере	Brest Cluster info	CPUs usage, %	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 	one_cluster_cpuusage one_cluster_totalcpu

Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД	
				<p>4. Пароль для пользователя должен быть зашифрован Base64</p> <p>5. В конфиг внесен адрес API Бреста</p> <p>На основе данных API вычисляется отношение использованного ЦПУ к всего имеющемуся.</p>		
11	Количество используемой памяти в кластере, %	Метрика показывает отношение занятой памяти к всего имеющейся памяти на выбранном кластере	Brest Cluster info	Memory usage, %	Прerequisites: <ul style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>На основе данных API вычисляется отношение использованной памяти к всего имеющейся памяти.</p>	one_cluster_memusage one_cluster_totalmem
12	Статус front хостов в выбранной зоне	Метрика показывает какой фронт выполняет роль leader, а какой роль follower.	Brest Management servers	RAFT status	Прerequisites: <ul style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос статуса фронтов на инстансе Бреста.</p>	one_zone_raft

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
13	Статус проверки доступности API для данной зоны	Метрика показывает доступность API для данного инстанса Бреста	Brest Management servers	API connection	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>При первом запросе к API анализируется ответ и делается вывод о доступности или не доступности API.</p>	one_api_connect
14	Статус проверки доступности WEB консоли управления Брестом	Метрика показывает доступность веб консоли управления Брестом	Brest Management servers	WEB console connection	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесены специальные переменные для проверки (webuser, webpasswd, webconsole) <p>С хоста, где установлен One-exporter делается попытка обратиться к URL веб консоли управления Бреста</p>	one_web_connect
15	Время отклика WEB консоли управления Брестом	Метрика показывает время отклика при замере доступности веб консоли управления Брестом	Brest Management servers	WEB console time connection	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 	one_web_connect_duration

Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
				<p>3. Пользователь прописан в one-exporter.conf</p> <p>4. Пароль для пользователя должен быть зашифрован Base64</p> <p>5. В конфиг внесены специальные переменные для проверки (webuser, webpasswd, webconsole)</p> <p>С хоста, где установлен One-exporter делается попытка обратиться к URL веб консоли управления Бреста и время ответа замеряется</p>	
16 Статус сервисов на выбранном front	Метрика показывает статусы определенных сервисов на выбранном фронте	Brest Management servers	Services on front	<p>Прerequisites:</p> <p>1. Используется systemd exporter</p>	systemd_unit_state_id
17 Статус front хоста	Метрика показывает статус выбранного фронта	Brest Management servers	Host state	<p>Прerequisites:</p> <p>1. Используется node exporter</p>	node_exporter_build_info
18 Использование ЦПУ на выбранном front, %	Метрика показывает распределение нагрузки на ЦПУ по процессорам на фронте	Brest Management servers	CPUs usage, %	<p>Прerequisites:</p> <p>1. Используется node exporter</p>	node_cpu_seconds_total
19 Использование памяти на выбранном front	Метрика показывает распределение памяти на фронте	Brest Management servers	Memory utilization	<p>Прerequisites:</p> <p>1. Используется node exporter</p>	<p>node_memory_MemTotal_bytes</p> <p>node_memory_MemFree_bytes</p> <p>node_memory_Cached_bytes</p> <p>node_memory_Buffers_bytes</p> <p>node_memory_SReclaimable_bytes</p> <p>node_memory_SwapTotal_bytes</p> <p>node_memory_SwapFree_bytes</p>
20 Поиск по логам выбранного хоста	Метрика позволяет осуществлять поиск по ключевым словам в логах или просто просматривать логи	Brest Management servers	Поиск по логам	<p>Метрика собирается из логов.</p> <p>Прerequisites:</p> <p>1. Используется сервис vector</p> <p>2. В /etc/vector/conf.d/ лежит файл с описанием нужных логов</p>	Таблица logs в БД ClickHouse

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
					3. Там же настроен парсинг этого лога по необходимым параметрам	
21	Список zombie-VM	Метрика показывает список zombie-VM по хостам виртуализации в выбранном кластере	Brest VMs info	List of zombie VMs	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос на наличие зомби VM на хостах виртуализации на инстансе Бреста.</p>	one_host_zombie
22	Список ВМок со статусами.	Метрика показывает все текущие статусы ВМок в данном кластере	Brest VMs info	List of VMs with status	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос статусов всех VM на хостах виртуализации на инстансе Бреста.</p>	one_vm_state
23	Информация о VM	Метрика показывает название выбранной виртуальной машины, кто ее владелец и о группе в Брест	Brest VMs info	VM Info	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 	one_vm_state one_vm_totalcpu

Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД	
				3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста Через API делается запрос об имени VM, информации о владельце, группе		
24	Использование ЦПУ конкретной VM	Метрика показывает использование ЦПУ выбранной виртуальной машиной	Brest VMs info	CPU usage (API)	Пререквизиты: <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста Через API делается запрос о использовании ЦПУ конкретной VM	one_vm_cpusage one_vm_totalcpu
25	Использование памяти конкретной VM	Метрика показывает использование памяти выбранной виртуальной машиной	Brest VMs info	Memory usage (API)	Пререквизиты: <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста Через API делается запрос о использовании памяти конкретной VM	one_vm_memusage one_vm_totalmem

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
26	Скорость чтения/записи в секунду	Метрика показывает скорость работы с диском выбранной VM	Brest VMs info	Disk write/read	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о скорости чтения/записи на диск конкретной VM</p>	one_vm_disk_read_bytes one_vm_disk_write_bytes
27	Среднее число операций чтения/записи в секунду	Метрика показывает число операций с диском выбранной VM	Brest VMs info	Disk IOPS	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о числе операций с диском конкретной VM</p>	one_vm_disk_read_iops one_vm_disk_write_iops
28	Сетевой трафик	Метрика показывает объем сетевого трафика у выбранной VM	Brest VMs info	Network usage	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен 	one_vm_network_rx one_vm_network_tx

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
					<p>быть зашифрован Base64</p> <p>5. В конфиг внесен адрес API Бреста</p> <p>Через API делается запрос о сетевом трафике у конкретной VM</p>	
29	Скорость трафика через сетевые интерфейсы VM	Метрика показывает скорость передачи данных на выбранной VM	Brest VMs info	Network speed	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о сетевом трафике у конкретной VM</p>	one_vm_network_rx one_vm_network_tx
30	Информация по хосту виртуализации	Метрика выводит данные об IP адресе, имени хоста виртуализации и времени последней загрузки хоста.	Brest Virtualization servers		<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется node exporter <p>На основе метрики из node exporter собираются данные об IP адресе, имени хоста и времени его последней загрузки</p>	node_boot_time_seconds
31	Статус хоста	Метрика показывает текущий статус хоста виртуализации	Brest Virtualization servers	Host state	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста 	one_host_state

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
					Через API делается запрос о текущем статусе хоста виртуализации	
32	Статус сервисов на выбранном хосте виртуализации	Метрика показывает статус всех заранее заданных сервисов по выбранному хосту виртуализации	Brest Virtualization servers	Services on host	Прerequisites: 1. Используется systemd exporter	systemd_unit_state_id
33	Утилизация ЦПУ на хосте	Метрика показывает распределение ЦПУ между процессорами на хосте виртуализации	Brest Virtualization servers	CPU utilization	Прerequisites: 1. Используется node exporter На основе метрики из node exporter собираются данные о использовании ЦПУ на хосте виртуализации	node_cpu_seconds_total
34	Утилизация ЦПУ на каждую VM	Метрика показывает распределение ЦПУ между VM на хосте виртуализации	Brest Virtualization servers	CPU utilization by VMs	Прerequisites: 1. Используется libvirt exporter На основе метрики из экспортера собираются данные о использовании ЦПУ для каждой VM на хосте виртуализации	libvirt_domain_info_cpu_time_seconds_total
35	Утилизация памяти	Метрика показывает распределение памяти на хосте виртуализации	Brest Virtualization servers	Memory Basic	Прerequisites: 1. Используется node exporter На основе метрики из node exporter собираются данные о использовании памяти на хосте виртуализации	node_memory_MemTotal_bytes node_memory_MemFree_bytes node_memory_Cached_bytes node_memory_Buffers_bytes node_memory_SReclaimable_bytes node_memory_SwapTotal_bytes node_memory_SwapFree_bytes
36	Утилизация памяти по VM	Метрика показывает распределение памяти между VM на хосте виртуализации	Brest Virtualization servers	Memory balloon by VMs	Прerequisites: 1. Используется node exporter 2. Используется libvirt exporter На основе метрик собираются данные о использовании памяти по всем VM на хосте виртуализации	node_memory_MemTotal_bytes libvirt_domain_memory_stats_actual_balloon
37	Среднее число операций чтения в секунду	Метрика показывает среднее число операций чтения с диска на хосте виртуализации	Brest Virtualization servers	Disk Read IOPS	Прerequisites: 1. Используется node exporter На основе метрики из node exporter собираются данные о	node_disk_reads_completed_total

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
					числе операций чтения с диска на хосте виртуализации	
38	Количество запросов операций чтения в секунду от VM	Метрика показывает среднее число операций чтения с диска от каждой VM на хосте виртуализации	Brest Virtualization servers	Read requests by VMs	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется libvirt exporter <p>На основе метрики из экспортера собираются данные об операциях чтения с диска для каждой VM на хосте виртуализации</p>	libvirt_domain_block_stats_read_requests_total
39	Среднее число операций записи в секунду	Метрика показывает среднее число операций записи на диск на хосте виртуализации	Brest Virtualization servers	Disk Write IOPS	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о числе операций записи на диск на хосте виртуализации</p>	node_disk_writes_completed_total
40	Количество запросов операций записи в секунду от VM	Метрика показывает среднее число операций записи на диск от каждой VM на хосте виртуализации	Brest Virtualization servers	Write requests by VMs	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется libvirt exporter <p>На основе метрики из экспортера собираются данные об операциях записи на диск для каждой VM на хосте виртуализации</p>	libvirt_domain_block_stats_write_requests_total
41	Сетевой трафик	Метрика показывает скорость передачи сетевого трафика на хосте виртуализации	Brest Virtualization servers	Network traffic, b/s	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о скорости передачи сетевого трафика на хосте виртуализации</p>	node_network_receive_bytes_total node_network_transmit_bytes_total
42	Сетевой трафик VM	Метрика показывает скорость передачи сетевого трафика по каждой VM на хосте виртуализации	Brest Virtualization servers	Network traffic by VMs, b/s	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется libvirt exporter <p>На основе метрики из экспортера собираются данные о сетевом трафике от каждой VM на хосте виртуализации</p>	libvirt_domain_interface_stats_receive_bytes_total libvirt_domain_interface_stats_transmit_bytes_total
43	Количество пакетов в секунду	Метрика показывает скорость передачи сетевых пакетов на хосте виртуализации	Brest Virtualization servers	Network traffic, packets/s	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о количестве переданных пакетов в секунду на хосте виртуализации</p>	node_network_receive_packets_total node_network_transmit_packets_total

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
44	Количество пакетов в секунду VM	Метрика показывает скорость передачи сетевых пакетов по каждой VM на хосте виртуализации	Brest Virtualization servers	Network traffic by VMs, packets/s	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется libvirt exporter <p>На основе метрики из экспортера собираются данные о количестве переданных пакетов в секунду от каждой VM на хосте виртуализации</p>	libvirt_domain_interface_stats_receive_packets_total libvirt_domain_interface_stats_transmit_packets_total
45	Поиск по логам выбранного хоста	Метрика позволяет осуществлять поиск по ключевым словам в логах или просто просматривать логи	Brest Virtualization servers	Поиск по логам	<p>Метрика собирается из логов.</p> <p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется сервис vector 2. В /etc/vector/conf.d/ лежит файл с описанием нужных логов 3. Там же настроен парсинг этого лога по необходимым параметрам 	Таблица logs в БД ClickHouse
46	Статус front хостов в выбранной зоне	Метрика показывает какой фронт leader лидером, а какой follower.	Brest summary	RAFT status	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос статуса фронтов на инстансе Бреста.</p>	one_zone_raft
47	Количество смены статуса лидера на fronts	Метрика показывает сколько раз менялся лидер у фронтов.	Brest summary	RAFT switches	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 	one_zone_raft

Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД	
				<p>4. Пароль для пользователя должен быть зашифрован Base64</p> <p>5. В конфиг внесен адрес API Бреста</p> <p>Через API делается запрос сколько раз менялся лидер фронтов на инстансе Бреста.</p>		
48	Количество активных хостов для зоны	Метрика показывает количество активных хостов виртуализации для выбранной зоны с разбивкой по кластерам.	Brest summary	Hosts	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о количестве активных хостов для зоны.</p>	one_cluster_activehosts
49	Количество запущенных VM для зоны	Метрика показывает количество запущенных VM для зоны с разбивкой по кластерам.	Brest summary	Running VMs	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о количестве запущенных VM для зоны.</p>	one_cluster_runningvms
50	Количество запущенных VM на всех кластерах	Метрика показывает количество запущенных VM для	Brest summary	Running	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 	one_cluster_runningvms

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
		зоны для всех кластеров.			<ol style="list-style-type: none"> 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о количестве запущенных VM для зоны.</p>	
51	Количество выключенных VM на всех кластерах	Метрика показывает количество выключенных VM для зоны для всех кластеров.	Brest summary	Powered off	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о количестве выключенных VM для зоны.</p>	one_vms_states_count
52	Количество VM в статусе failed на всех кластерах	Метрика показывает количество VM в статусе failed для зоны для всех кластеров.	Brest summary	Failed	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста 	one_vms_states_count

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
					Через API делается запрос о количестве VM в статусе failed для зоны.	
53	Количество VM в статусе pending на всех кластерах	Метрика показывает количество VM в статусе pending для зоны для всех кластеров.	Brest summary	Pending	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о количестве VM в статусе pending для зоны.</p>	one_vms_states_count
54	Количество VM в статусе hold на всех кластерах	Метрика показывает количество VM в статусе hold для зоны для всех кластеров.	Brest summary	Hold	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о количестве VM в статусе hold для зоны.</p>	one_vms_states_count
55	Количество VM в статусе init на всех кластерах	Метрика показывает количество VM в статусе init для зоны для всех кластеров.	Brest summary	Init	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 	one_vms_states_count

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
					<p>4. Пароль для пользователя должен быть зашифрован Base64</p> <p>5. В конфиг внесен адрес API Бреста</p> <p>Через API делается запрос о количестве VM в статусе init для зоны.</p>	
56	Количество VM в статусе cloning на всех кластерах	Метрика показывает количество VM в статусе cloning для зоны для всех кластеров.	Brest summary	Cloning	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о количестве VM в статусе cloning для зоны.</p>	one_vms_states_count
57	Количество VM в статусе clone fail на всех кластерах	Метрика показывает количество VM в статусе clone fail для зоны для всех кластеров.	Brest summary	Clone fail	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о количестве VM в статусе clone fail для зоны.</p>	one_vms_states_count
58	Количество ЦПУ по кластерам	Метрика показывает общее количество ЦПУ для зоны с	Brest summary	CPU total	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 	one_cluster_totalcpu

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
		разбивкой по кластерам.			<ol style="list-style-type: none"> Создан технический пользователь в Брест как core пользователь Пользователь прописан в one-exporter.conf Пароль для пользователя должен быть зашифрован Base64 В конфиг внесен адрес API Бреста <p>Через API делается запрос о общем количестве ЦПУ для зоны.</p>	
59	Количество использованных ЦПУ по кластерам	Метрика показывает количество использованных ЦПУ для зоны с разбивкой по кластерам.	Brest summary	CPU used	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется one exporter Создан технический пользователь в Брест как core пользователь Пользователь прописан в one-exporter.conf Пароль для пользователя должен быть зашифрован Base64 В конфиг внесен адрес API Бреста <p>Через API делается запрос о количестве использованных ЦПУ для зоны.</p>	one_cluster_cpuusage
60	Количество памяти по кластерам	Метрика показывает общее количество памяти для зоны с разбивкой по кластерам.	Brest summary	Memory total	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется one exporter Создан технический пользователь в Брест как core пользователь Пользователь прописан в one-exporter.conf Пароль для пользователя должен быть зашифрован Base64 В конфиг внесен адрес API Бреста 	one_cluster_totalmem

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
					Через API делается запрос о общем количестве памяти для зоны.	
61	Количество используемой памяти по кластерам	Метрика показывает количество использованной памяти для зоны с разбивкой по кластерам.	Brest summary	Memory used	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о количестве использованной памяти для зоны.</p>	one_cluster_memusage
62	Amount of used CPUs	Метрика показывает отношение использованных ЦПУ к общему количеству ЦПУ для зоны с разбивкой по кластерам.	Brest summary	CPUs usage, %	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о количестве использованных ЦПУ для зоны.</p>	one_cluster_cpuusage one_cluster_totalcpu
63	Отношение количества использованной памяти к общему количеству памяти	Метрика показывает отношение использованной памяти к общему количеству памяти для зоны с разбивкой по кластерам.	Brest summary	Memory usage, %	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 	one_cluster_memusage one_cluster_totalmem

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
					<p>4. Пароль для пользователя должен быть зашифрован Base64</p> <p>5. В конфиг внесен адрес API Бреста</p> <p>Через API делается запрос о количестве использованной памяти для зоны.</p>	
64	Размер используемых DataStore	Метрика показывает размер и количество занятого места на DataStore для зоны.	Brest summary	Datastores size	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о размерах датасторов для зоны.</p>	one_ds_usedmb
65	Разбивка DataStore по кластерам	Метрика показывает распределение DataStore по кластерам для зоны.	Brest summary	Datastore uses on cluster	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется one exporter 2. Создан технический пользователь в Брест как core пользователь 3. Пользователь прописан в one-exporter.conf 4. Пароль для пользователя должен быть зашифрован Base64 5. В конфиг внесен адрес API Бреста <p>Через API делается запрос о использовании датасторов на кластерах для зоны.</p>	one_ds_available
66	Утилизация ЦПУ по хостам в группе	Метрика показывает использование ЦПУ по хостам внутри группы	Brest summary	CPU utilization by hosts in group	<p>Прerequisites:</p> <ol style="list-style-type: none"> 1. Используется node exporter 	node_cpu_seconds_total

	Наименование	Описание	Название панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
					На основе метрики из node exporter собираются данные о распределении ЦПУ по хостам внутри группы	
67	Использование памяти на хостах виртуализации в группе	Метрика показывает использование памяти по хостам внутри группы	Brest summary	Memory Used	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о распределении памяти по хостам внутри группы</p>	node_memory_MemTotal_bytes node_memory_MemFree_bytes node_memory_Cached_bytes node_memory_Buffers_bytes node_memory_SReclaimable_bytes
68	Скорость входящего сетевого трафика	Метрика показывает скорость входящего сетевого трафика по хостам внутри группы	Brest summary	Network Receive Traffic	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о входящем сетевом трафике по хостам внутри группы</p>	node_network_receive_bytes_total
69	Скорость исходящего сетевого трафика	Метрика показывает скорость исходящего сетевого трафика по хостам внутри группы	Brest summary	Network Transmit Traffic	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о исходящем сетевом трафике по хостам внутри группы</p>	node_network_transmit_bytes_total
70	Использованное дисковое пространство по хостам	Метрика показывает использованное дисковое пространство на хостах внутри группы	Brest summary	Disk Space Used Basic	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о использовании дискового пространства на хостах внутри группы</p>	node_filesystem_size_bytes node_filesystem_available_bytes

6.5 Мониторинг операционной системы Astra Linux

	Наименование	Описание	Наименование панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
1	Утилизация ЦПУ	Метрика показывает использование ЦПУ на объекте наблюдения		CPU Busy CPU Basic	<p>Прerequisites:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter</p>	node_cpu_seconds_total

Наименование	Описание	Наименование панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
				собираются данные о загрузке ЦПУ	
2 Утилизация ЦПУ за период	Метрика показывает использование ЦПУ на объекте наблюдения: средние значения за 5 и 15 минут	Node Exporter Full	Sys Load (5 m avg) Sys Load (15 m avg)	<p>Пререквизиты:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о загрузке ЦПУ</p>	node_load5 node_load15
3 Утилизация памяти	Метрика показывает использования памяти на объекте наблюдения	Node Exporter Full	Memory Basic Memory Stack RAM Used SWAP Used Memory Active \ Inactive (detail) Memory Committed	<p>Пререквизиты:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о использовании памяти на объекте наблюдения</p>	node_memory_MemTotal_bytes node_memory_MemFree_bytes node_memory_Cached_bytes node_memory_Buffers_bytes node_memory_SReclaimable_bytes node_memory_SwapTotal_bytes node_memory_SwapFree_bytes node_memory_Inactive_bytes node_memory_CommitLimit_bytes node_memory_Committed_AS_bytes node_memory_HardwareCorrupted_bytes node_memory_PageTables_bytes node_memory_SwapCached_bytes
4 Сетевой трафик	Метрика показывает скорость передачи сетевого трафика	Node Exporter Full	Network traffic Network traffic basic	<p>Пререквизиты:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о скорости передачи</p>	node_network_transmit_bytes_total node_network_receive_bytes_total

Наименование	Описание	Наименование панели метрик	Название метрики / графика	Технология получения метрики	Название метрики в БД
				сетевого трафика на хосте виртуализации	
5 Среднее число операций чтения в секунду	Метрика показывает среднее число операций чтения с диска	Node Exporter Full	Disk IOPS	<p>Пререквизиты:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о числе операций чтения с диска на объекте наблюдения</p>	node_disk_reads_completed_total
6 Среднее число операций записи в секунду	Метрика показывает среднее число операций записи на диск	Node Exporter Full	Disk IOPS	<p>Пререквизиты:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о числе операций записи на диск на объекте наблюдения</p>	node_disk_writes_completed_total
7 Среднее число операций в секунду	Метрика показывает среднее число операций записи и чтения на диск	Node Exporter Full	Disk IOPS	<p>Пререквизиты:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о числе операций записи и чтения на диск на объекте наблюдения</p>	node_disk_reads_completed_total
8 Использованное дисковое пространство по хостам	Метрика показывает использованное дисковое пространство	Brest summary	Disk Space Used Basic Disk Space Used	<p>Пререквизиты:</p> <ol style="list-style-type: none"> Используется node exporter <p>На основе метрики из node exporter собираются данные о использовании дискового пространства на объекте наблюдения</p>	node_filesystem_size_bytes node_filesystem_avail_bytes

6.6 Настроенные триггеры для создания событий по логам

Ниже описаны триггеры для создания событий по логам, которые в данный момент настроены в Astra Monitoring.

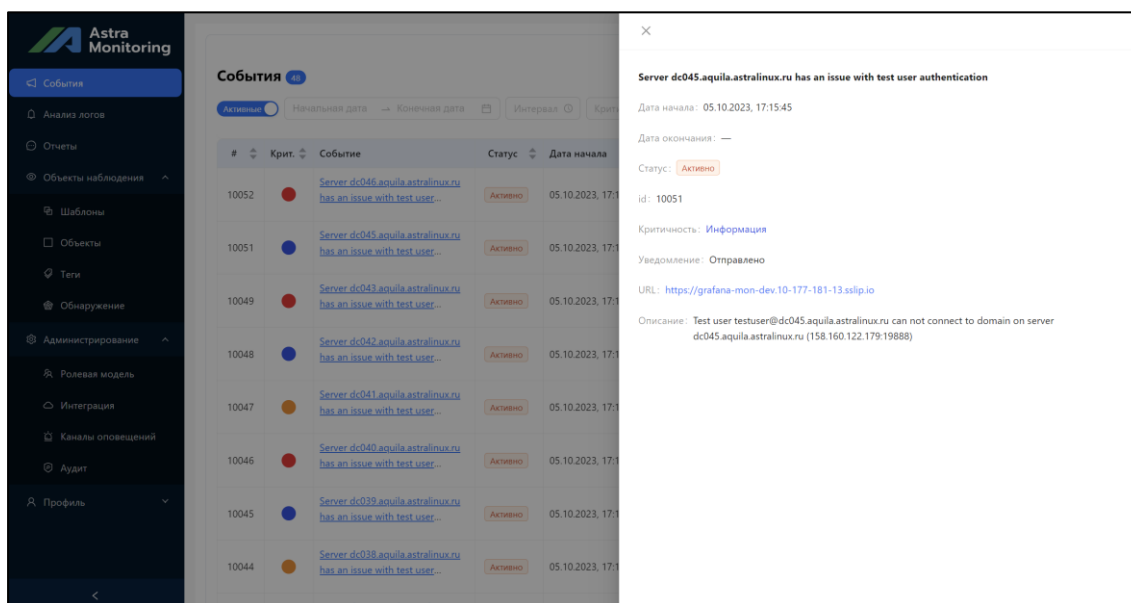
Продукт	Условие (триггер)	Описание	Лог	Технические детали
ALD Pro	<p>Вызывается критическое уведомление на сообщение в логе 'Not listening for new connections - too many fds open' .</p> <p>Критическое уведомление снимается, если в логе появляется сообщение 'Listening for new connections again'.</p>	<p>Лог сигнализирует о проблеме с количеством открытых файловых дескрипторов (fds) в системе. Эта ошибка обычно говорит о том, что был достигнут лимит операционной системы на количество файловых дескрипторов, которые может открыть процесс. В результате новые соединения не будут прослушиваться, что может вызвать сбой в доступности сервиса.</p>	<p>/var/log/dirsrv/slapd- <DOMAIN>/error</p>	<p>Имя события - 'Not listening for new connections'</p> <p>Severity = 'critical'</p> <p>Для срабатывания триггера используется регулярное выражение: r'^{(?P<timestamp>\d{2})[a-zA-Z]{3}\d{4}:\d{2}:\d{2}\.\d{9}\+\d{4}}\] - ERR - .*? - Not listening for new connections - too many fds open'</p> <p>Для гашения: r'^{(?P<timestamp>\d{2})[a-zA-Z]{3}\d{4}:\d{2}:\d{2}\.\d{9}\+\d{4}}\] - ERR - .*? - Listening for new connections again'</p>

7 Пользовательский интерфейс Астра Мониторинг

7.1 Пользовательское окно "События"

7.1.1 Общая информация

В окне "События" размещена таблица, предназначенная для отображения списка событий и связанных с ними данных. В таблице присутствуют интерактивные элементы для фильтрации и сортировки записей. Над таблицей размещены элементы, дополняющие возможности поиска и отображающие активные фильтры. При одинарном щелчке мыши (ЛКМ) на выбранном событии открывается правая боковая панель, содержащая подробную информацию о событии.



В правом нижнем углу под таблицей расположена функция навигации для перехода между страницами.



7.1.2 Основная таблица с событиями

Таблица содержит следующий данные по каждому событию:

	Объект интерфейса	Значение
1	#	Уникальный идентификатор события
2	Крит.	Уровень критичности события, выделенный цветом (активные события обозначаются ярким цветом, завершённые - тусклым):

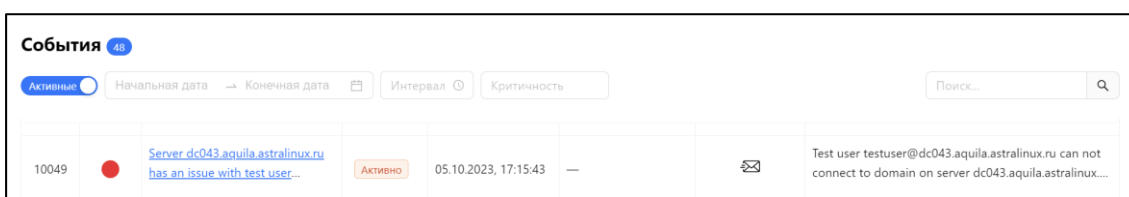
Объект интерфейса	Значение
	<ul style="list-style-type: none"> • Critical - красный круг • Warning - желтый круг • Info- синий круг • Неизвестное значение - серый знак вопроса
3 Событие	Название объекта мониторинга с контекстной гиперссылкой на соответствующую панель метрик в Grafana
4 Статус	Состояние события, указывающее, активно оно или уже завершено
5 Дата начала	Дата и время начала события в формате в соответствии с настройками браузера
6 Дата окончания	Дата и время окончания события в формате в соответствии с настройками браузера. Поле может быть пустым, если событие еще активно
7 Уведомление	Статус уведомления о событии, обозначенный иконкой конверта, если уведомление отправлено или прочерком, если уведомление не отправлено
8 Описание события	Краткое описание события

7.1.3 Правая боковая панель

При одинарном щелчке мыши (ЛКМ) на выбранном событии активируется боковая панель, размещаемая справа. Панель содержит полную информацию о данном событии. Чтобы скрыть боковую панель, пользователь может либо нажать на иконку "крестик" в верхнем левом углу, либо выполнить щелчок ЛКМ в области, не принадлежащей данной боковой панели.

7.1.4 Дополнительные элементы

Над таблицей находится несколько интерактивных элементов, которые дополняют функциональность и возможности при работе с окном "События" (реализуют возможность поиска и фильтрации в соответствии с выбранными критериями):



Описание элементов поиска и фильтрации:

Объект интерфейса	Значение
1 Переключатель статуса	Этот переключатель позволяет фильтровать и отображать только активные события в таблице (параметр Статус). По умолчанию он включен для облегчения навигации и обеспечения оптимальной производительности при просмотре списка событий.
2 Поиск по заданной дате начала события	Этот поисковый инструмент позволяет быстро находить события, которые были созданы в определенном диапазоне дат (параметр Дата начала). Пользователь может указать начальную и конечную даты для поиска. В этом случае отображаются только события, попадающие в

Объект интерфейса	Значение
	заданный интервал. Поиск по дате исключает поиск по динамическому временному интервалу.
3 Поиск по заданному временному интервалу	Этот поисковый инструмент позволяет быстро находить события, которые были созданы в определенном временном диапазоне. Поиск по динамическому временному интервалу исключает поиск по дате начала события.
4 Поиск по названию объекта мониторинга (Monitoring object) и описанию (Description)	С помощью этой функции можно осуществлять поиск событий на основе названия объектов мониторинга и их описания. Пользователь может ввести ключевые слова или фразы, характеризующие объекты или их описание. В этом случае отобразятся только события, соответствующие заданным поисковым критериям.
5 Активные фильтры по критичности	В этом поле отображаются активные фильтры по критичности

7.2 Пользовательское окно "Объекты"

7.2.1 Описание функционала окна "Объекты"

В окне "Объекты" пользователь имеет возможно добавлять, изменять, удалять объекты мониторинга. Все объекты представлены в табличном виде.

Имя	Тип	Инстанс	Полное имя	Теги	Действия
test3	Брест Фронт	test3	test3	env: prod env: dev env: test	↗ ✖
test2	Брест Фронт	test2	test2	tag: test2	↗ ✖
test1	Брест Фронт	test1	test1	tag: test1	↗ ✖
localhost	Astra Linux	localhost:9100	localhost		↗ ✖
my host	Astra Linux	tasma.stp.local:9100	my.host.local		↗ ✖
test1234	Брест хост	test1234	test1234	location: bg location: bg1	↗ ✖
test4	Брест Фронт	test4	test4	sat: only my	↗ ✖
node1.aquila.astralinux.ru	Astra Linux	10.177.123.104:19100	node1.aquila.astralinux.ru	group: opennebula	↗ ✖
node2.aquila.astralinux.ru	Astra Linux	10.177.123.104:29100	node2.aquila.astralinux.ru	group: opennebula	↗ ✖
node3.aquila.astralinux.ru	Astra Linux	10.177.123.104:39100	node3.aquila.astralinux.ru	group: opennebula	↗ ✖
dc01.aquila.astralinux.ru	Astra Linux	158.160.122.179:19100	dc01.aquila.astralinux.ru	group: add-pro subsystem: controller	↗ ✖
dhcp.aquila.astralinux.ru	Astra Linux	158.160.122.179:29100	dhcp.aquila.astralinux.ru	group: add-pro subsystem: dhcp	↗ ✖
mon.aquila.astralinux.ru	Astra Linux	158.160.122.179:39100	mon.aquila.astralinux.ru	group: add-pro subsystem: monitoring	↗ ✖

Таблица содержит следующий данные по каждому объекту мониторинга:

Объект интерфейса	Значение
1 Имя	Имя объекта мониторинга в текстовом виде.
2 Тип	Ожидаемый набор метрик, характерный для заданного продукта, например, системы виртуализации, контроллера домена или системных метрик Linux. Необходим для применения к объекту различных политик, специфичных для этого типа. Предопределены следующие типы:

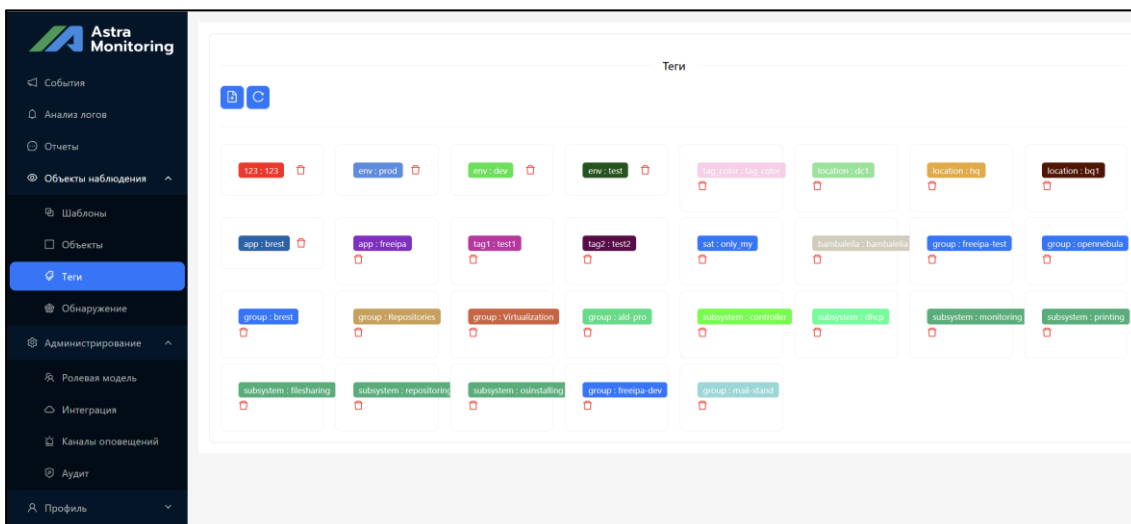
Объект интерфейса	Значение
	<ul style="list-style-type: none"> • Брест система виртуализации - получение метрик libvirt. • Брест API - получение метрик opennebula. • Linux - получение метрик ОС (CPU, Mem, Network and etc). • FreeIPA - получение метрик и логов Freeipa. • ALD-Pro - получение метрик и логов ALD Pro • Брест БД - получение метрик PostgreSQL Брест. • Postfix - получение метрик о состоянии подсистемы RuPost Postfix • Postfix БД - получение метрик PostgreSQL Postfix (RuPost) .
3 Инстанс	Адрес объекта (exporter) в формате prometheus instance , с номером порта, для получения метрик системой мониторинга
4 Полное имя	Имя объекта, которым будут маркироваться получаемые метрики и которое будет в дальнейшем использоваться при работе с панелями метрик и т.д. В общем случае, соответствует hostname или fqdn хоста
5 Теги	Набор меток (labels), необходимых для дополнительной идентификации объекта.

⚠ Поля "Инстанс" и "Тип" мониторинга определяют хост и порт для подключения, а так же набор метрик и логов для сбора.

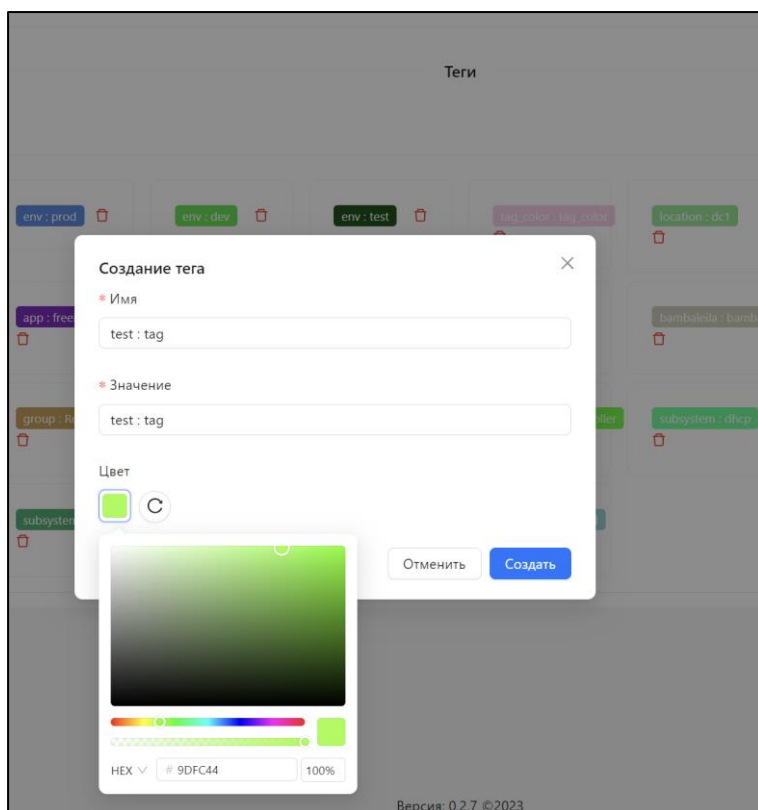
Для каждой метрики необходим уникальный идентификатор. Идентификатор для метрики конкретного объекта наблюдения формируется из следующих полей:

- Инстанс
- Набор тегов
- Название метрики

Теги представляют собой пользовательские метки объектов наблюдения. Данные теги могут быть произвольными. Для создания и удаления тегов предусмотрен отдельный интерфейс в окне "Теги":



Для каждого тега при создании можно указать произвольное название и выбрать цвет:



7.2.2 Дополнительные возможности окна "Объекты"

Для поля "Имя" предусмотрена возможность контекстного поиска.

Для полей "Теги" и "Тип" предусмотрена возможность фильтрации в соответствии с имеющимися значениями в столбцах.

При включении "Удаленные" в таблице дополнительно будут отображены ранее удаленные объекты наблюдения:

Имя	Тип	Инстанс	Полное имя	Теги	Действия
gca-yc-ruca-sbrm02.astralinux.ru (Удалён)	Linux	10.177.180.174:9100	gca-yc-ruca-sbrm02.astralinux.ru	group : Repositories	🔄 🗑️
gca-yc-ruca-sdstr01.astralinux.ru (Удалён)	Linux	10.177.180.168:9100	gca-yc-ruca-sdstr01.astralinux.ru	group : Repositories	🔄 🗑️
gca-yc-ruca-sedge01.astralinux.ru (Удалён)	Linux	10.177.180.167:9100	gca-yc-ruca-sedge01.astralinux.ru	group : Repositories	🔄 🗑️
gca-yc-ruca-brm01.astralinux.ru (Удалён)	Linux	10.177.181.36:9100	gca-yc-ruca-brm01.astralinux.ru	group : Repositories	🔄 🗑️

Предусмотрен постраничный вывод объектов мониторинг и возможность выбора количества отображаемых объектов на странице:

Объекты



Удалённые

Имя	Тип	Инстанс	Полное имя	Теги	Действия
rupost.db01.astralinux.ru	Linux	158.160.117.150:39558	rupost.db01.astralinux.ru	group: mail-stand	↗ 🗑
rupost.mail01.astralinux.ru	Linux	158.160.117.150:19100	rupost.mail01.astralinux.ru	group: mail-stand	↗ 🗑
rupost.mail02.astralinux.ru	Linux	158.160.117.150:29100	rupost.mail02.astralinux.ru	group: mail-stand	↗ 🗑
rupost.db01.astralinux.ru	Linux	158.160.117.150:39100	rupost.db01.astralinux.ru	group: mail-stand	↗ 🗑
gka-yc-ruca-srv-mail01	Linux	10.177.180.199:9100	gka-yc-ruca-srv-mail01	group: mail-stand	↗ 🗑
gka-yc-ruca-srv-mail01	Linux	10.177.180.199:9558	gka-yc-ruca-srv-mail01	group: mail-stand	↗ 🗑
gka-yc-ruca-srv-mail01	Linux	10.177.180.199:9187	gka-yc-ruca-srv-mail01	group: mail-stand	↗ 🗑
gka-yc-ruca-srv-mail01	Linux	10.177.180.199:9154	gka-yc-ruca-srv-mail01	group: mail-stand	↗ 🗑
rupost.mail01.astralinux.ru	Linux	158.160.117.150:19777	rupost.mail01.astralinux.ru	group: mail-stand	↗ 🗑
rupost.mail02.astralinux.ru	Linux	158.160.117.150:29777	rupost.mail02.astralinux.ru	group: mail-stand	↗ 🗑
rupost.nfs01.astralinux.ru	Linux	158.160.117.150:49100	rupost.nfs01.astralinux.ru	group: mail-stand	↗ 🗑
test_type	ALD-Pro	test_type	test_type	env: prod env: dev	↗ 🗑

- 10 / page
- 20 / page
- 50 / page
- 100 / page