Инструкция по применению Астра Мониторинг.

Мониторинг операционной системы

Astra Monitoring

Оглавление

Программный платформа "Астра Мониторинг" представляет собой решение для отслеживания состояния всей IT-инфраструктуры, включающей в себя как виртуальные, так и физические слои, а также приложения и сервисы, в т.ч. продуктовые решения ПАО Группа Астра. Платформа позволяет:

- осуществлять сбор данных и логов с объектов наблюдения;
- создавать события в соответствии с настроенными порогами;
- оповещать о критических событиях;
- предоставлять данные по инцидентам в виде отчетов с разным уровнем агрегации.

В общем случае платформа мониторинга представляет из себя серверные компоненты, разворачиваемые на одном или нескольких сервера. И клиентские компоненты, разворачиваемые на объектах наблюдения (сервер, ПК).

1 Назначение инструкции

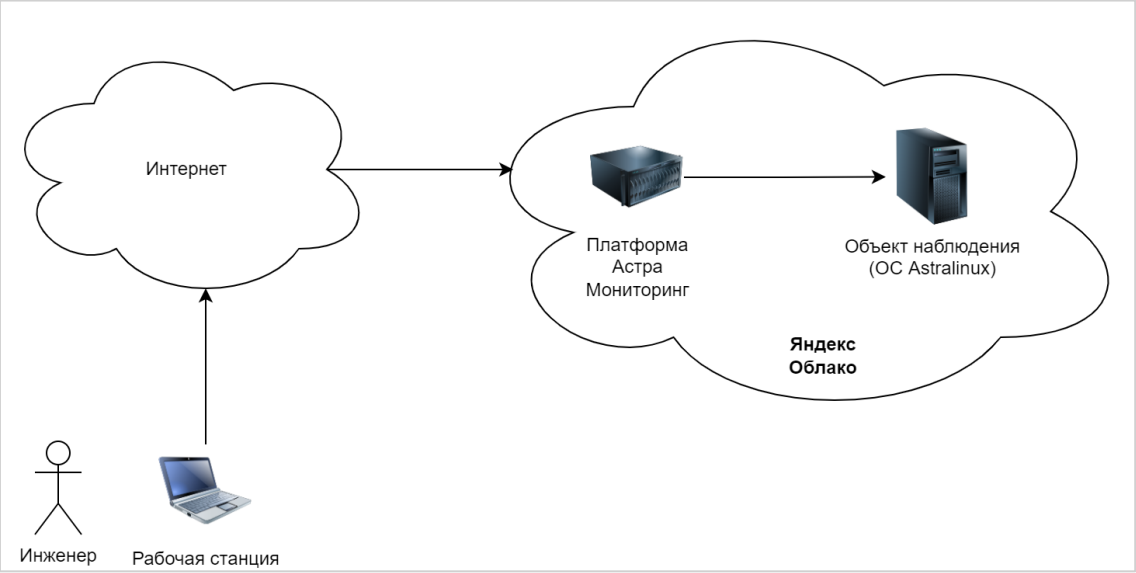
Астра Мониторинг позволяет получать метрики, данные журналов и события по текущему статусу функционирования продуктов ПАО Группа Астра.
Данная инструкция иллюстрирует применение Астра Мониторинг в качестве мониторинга операционной системы Astra Linux.
Остальной функционал платформы Астра Мониторинг в виде сбора логов и получения метрик продуктов ПАО Группа Астра не предусмотрен тестовым стендом, и их мониторинг выходит за рамки данной инструкции.

2 Описание сценария

В исходном состоянии рабочее окружение состоит из следующих компонентов:

- Облачное окружение в виде Яндекс Облако (Yandex Cloud). Состоит из следующих компонентов:
    - Платформа Астра Мониторинг. Развернута на узле, который представляет собой виртуальную машину. Платформа осуществляет сбор метрик с объектов наблюдения, формирует события в соответствии с предустановленными порогами, осуществляет оповещение по новым событиям через протокол SMTP на электронную почту;
    - Узел управления объектами наблюдения и отображения событий. Представлен в виде UI интерфейса. Развернут на узле платформы Астра Мониторинг
    - Объект наблюдения, которые представляет собой виртуальную машину с предустановленной ОС Astralinux 1.7.5.
- Рабочая станция инженера, выполняющего все перечисленные далее операции.

Цель сценария - сбор метрик с узла наблюдения (метрики ОС Astralinux), формирование события по предустановленным порогам, создание оповещений по событиям.



Процедура состоит из следующих стадий:

1. Подготовка рабочего окружения, включая рабочую станцию, доступ к облаку (к платформе Астра Мониторинг и к объекту наблюдения. Доступ к UI интерфейсу Астра Мониторинг.
2. Добавление объекта наблюдения на платформу Астра Мониторинга.
3. Фиксация пороговых значений метрик для формирования события.
4. Фиксация параметров оповещения (учетные данные для протокола SMTP).
5. Эмуляция действий на объекте наблюдений для изменения значений метрик.
6. Формирование события в соответствии с заданными пороговыми значениями. Отображение события в UI интерфейсе Астра Мониторинг.

7. Формирование оповещения в соответствии с заданными почтовыми учетными данными пользователя.

Перед выполнением этой процедуры инженер получает учетные данные для доступа к платформе Астра Мониторинг, UI интерфейсу платформы и объекту наблюдения.

3 Реквизиты для начала работы

Для подготовки и выполнения последующих действий понадобятся следующие учетные данные:

Узел	Метод доступа	Адрес	Учетная запись	Пароль	Комментарий
Платформа Астра Мониторинг	SSH	Публичный: XXX.XXX.XXX.XXX Внутренний: YYY.YYY.YYY.YYY			Для доступа по SSH необходимо использовать публичный адрес.
Объект мониторинга (ОС Astralinux)	SSH	Публичный: XXX.XXX.XXX.XXX Внутренний: YYY.YYY.YYY.YYY	SSH-ключ	SSH-ключ	Для доступа по SSH необходимо использовать публичный адрес. Внутренний адрес объекта мониторинга будет использоваться платформой для сбора метрик
Почтовый ящик	-	https://mail.yandex.ru/	astramonitoringnotify@yandex.ru	Пароль	Почтовый ящик используется в сценарии для просмотра уведомлений о событиях в системе.
Платформа Астра Мониторинг. UI	http	https:// XXX.XXX.XXX.XXX	Имя учетной записи	Пароль	HTTP доступ в UI Астра Мониторинг. Для простоты настройки стенда сертификат для подключения по HTTPS не используется.
Графический интерфейс Grafana	http	http:// XXX.XXX.XXX.XXX:3000/	Имя учетной записи	Пароль	Для просмотра метрик и их исторических значений используется компонент Grafana. Этот компонент развернут на узле "Платформа Астра Мониторинг".

Для доступа по SSH к стенду необходимо использовать SSH-ключ для подключения. Для подключения к узлам XXX.XXX.XXX.XXX, YYY.YYY.YYY.YYY необходимо соответствующий ключ

4 Подготовка рабочего окружения

4.1 Подготовка рабочей станции

Для конечной проверки работоспособности развернутой инфраструктуры необходимо следующее типовое программное обеспечение:

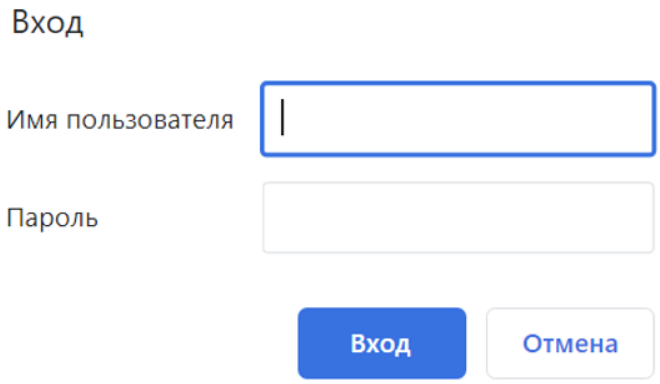
- Web-браузер Mozilla FireFox или Google Chrome – текущая стабильная версия;
- Клиент SSH.

4.2 Подготовка платформы Астра Мониторинг

Узел с платформой Астра Мониторинг содержит все необходимое программное обеспечение и ключи авторизации, обеспечивающие работоспособность платформы.

В адресной строке браузера перейдите по ссылке: http://XXX.XXX.XXX.XXX.

В открывшемся окне введите имя учетной записи и пароль. Нажмите кнопку "Вход".



4.3 Подготовка объекта наблюдения

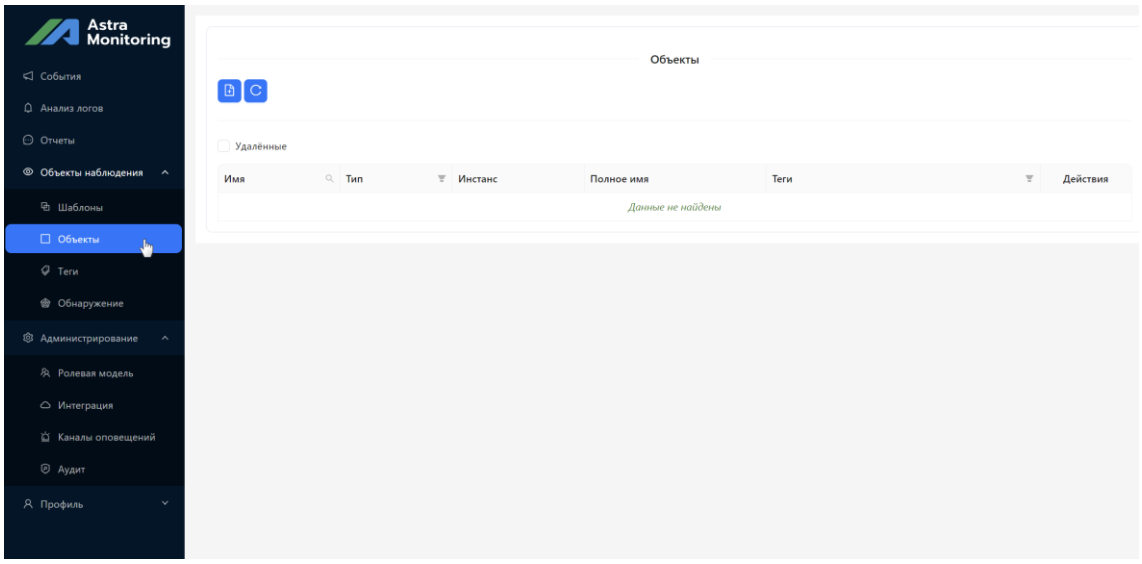
Узел объекта наблюдения содержит все необходимые компоненты для мониторинга метрик ОС AstraLinux. А также содержит пакет "stress-ng" для загрузки CPU в соответствии с установленными порогами для генерации события.

5 Выполнение сценария с использованием рабочего окружения

5.1 Добавление объекта наблюдения на платформу Астра Мониторинга

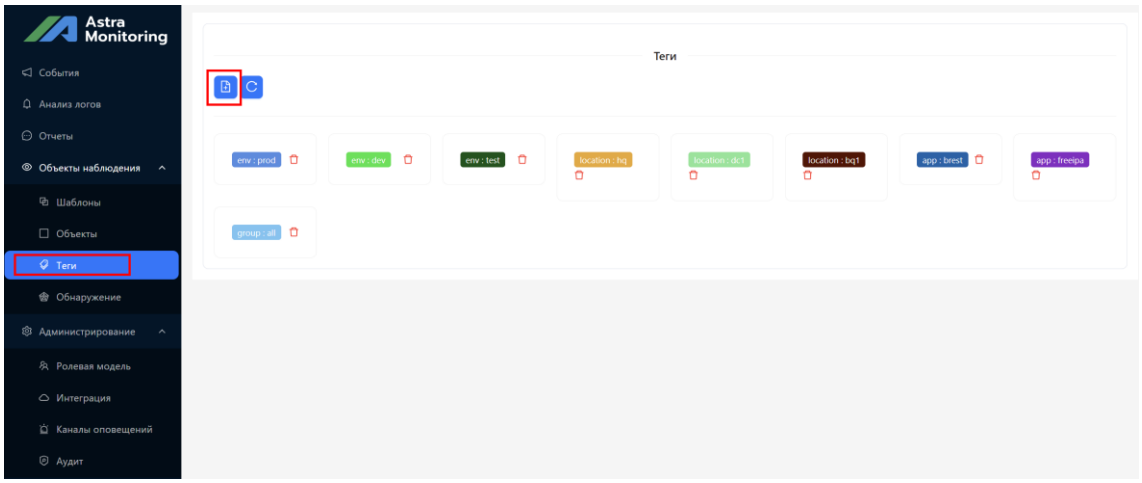
После успешного входа в UI Астра мониторинг для добавления объекта необходимо перейти на вкладку "Объекты наблюдения" → "Объекты".

В окне "Объекты" перечислены все объекты мониторинга, которые добавляются на платформу Астра Мониторинг для сбора метрик и формирования событий.

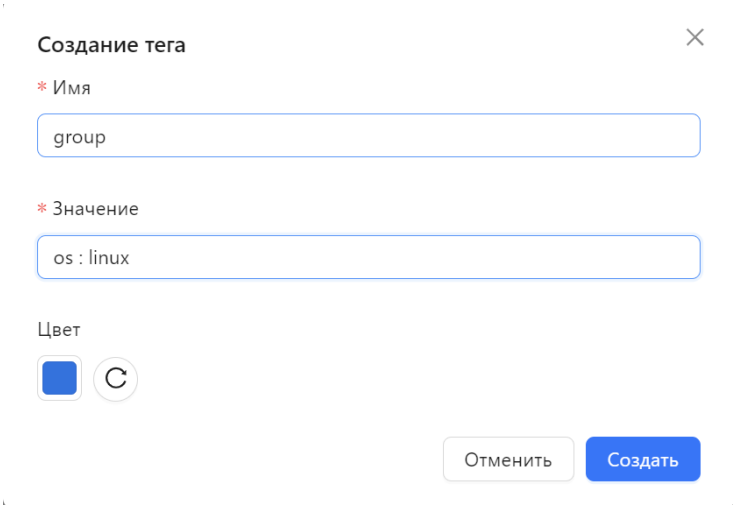


Можем видеть, что наблюдаемые объекты отсутствуют.
У каждого объекта наблюдений предусмотрен набор обязательных полей: "Имя", "Тип", "Полное имя", "Инстанс", "Теги". Набор этих полей определяет ключ каждой из получаемых метрик для последующей идентификации каждой отдельно взятой метрики для каждого объекта мониторинга.

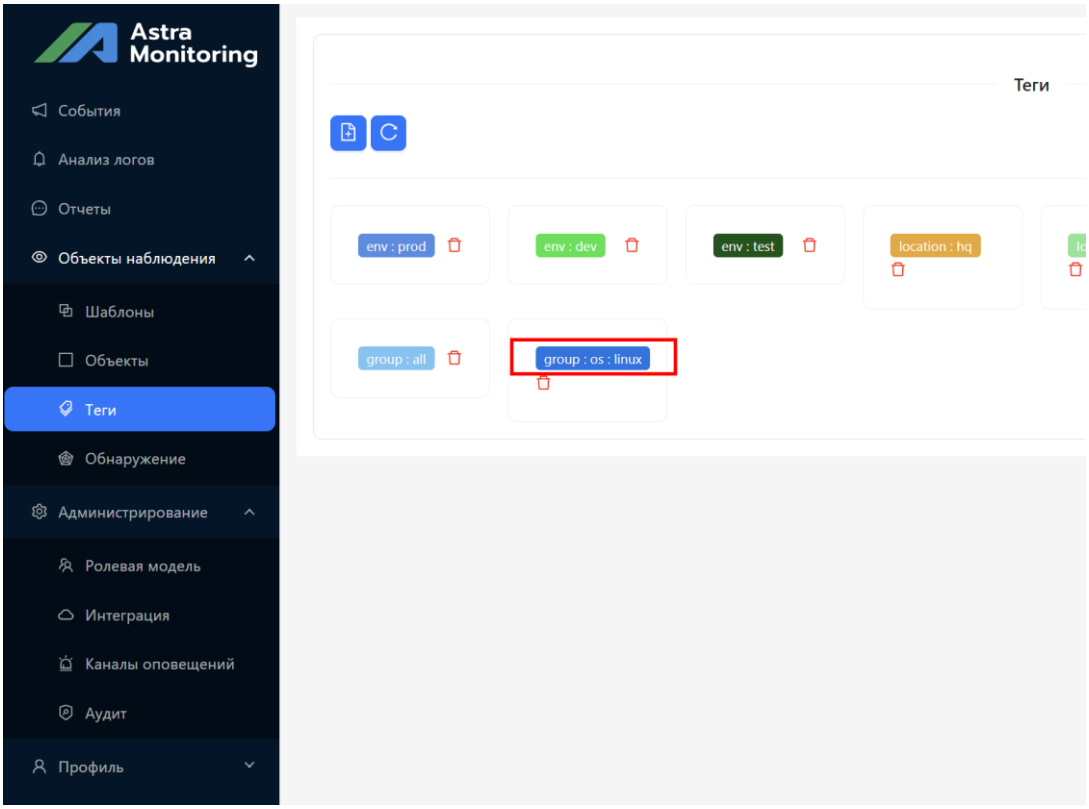
Создадим новый "Тег". Для этого перейдем во вкладку "Объекты наблюдения"→ "Теги" и добавим новый "Тег".



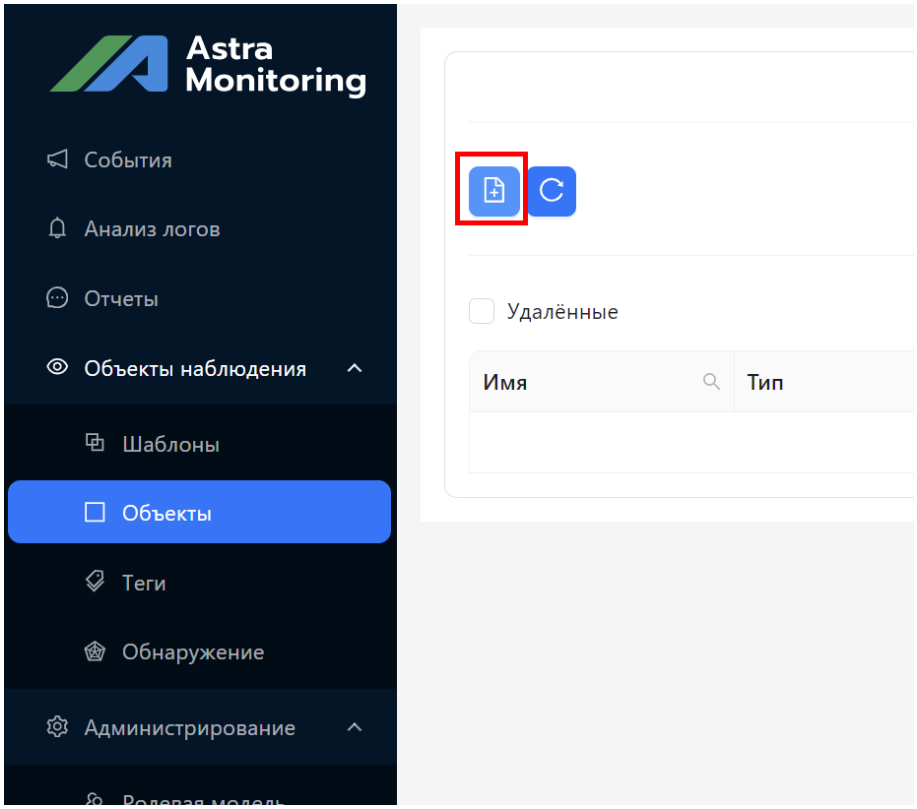
Новому тегу дадим имя "group", значение - "os : linux":



После создания нового тега мы увидим его присутствие на странице "Объекты наблюдения"→ "Теги":



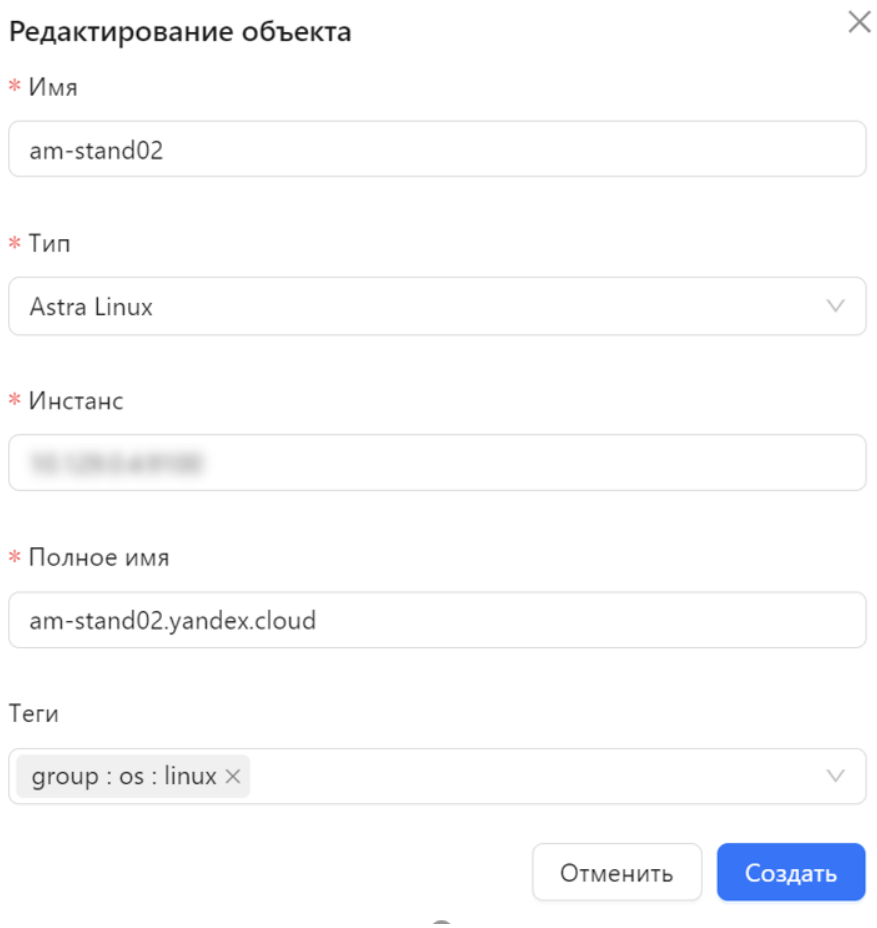
Добавим объект мониторинга (виртуальную машину с ОС Astralinux). Для этого необходимо нажать на соответствующую кнопку:



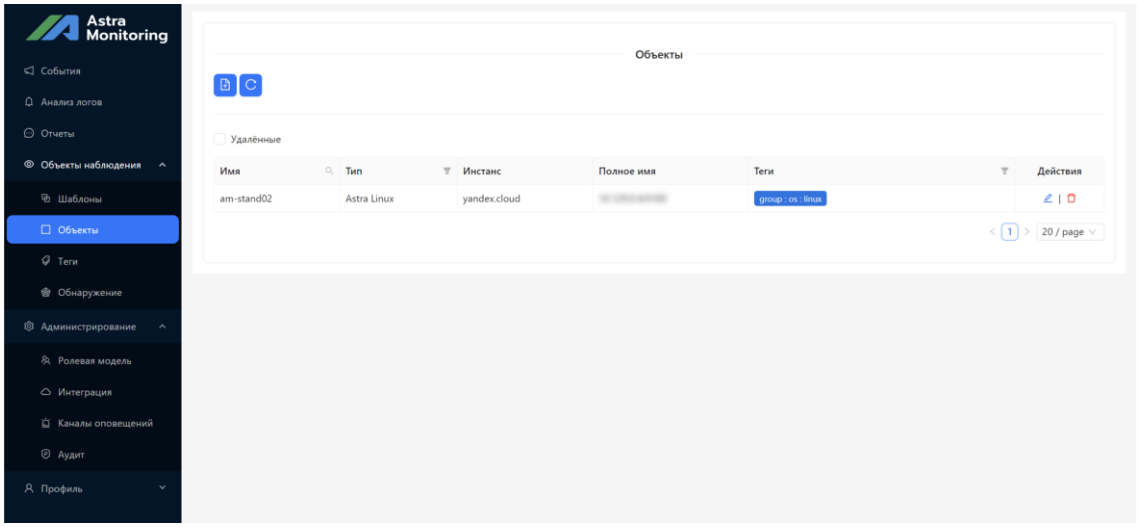
Во сплывающем окне введите следующие параметры:

- Поле "Имя". Значение "am-stand02".
- Поле "Тип". Значение "Astra Linux" необходимо выбрать из списка.
- Поле "Инстанс". Здесь укажем IP адрес объекта наблюдения и порт. Будем использовать внутренний адрес объекта во внутренней подсети Yandex.Cloud. Укажем значение "YYY.YYY.YYY.YYY:9100". Порт 9100 это порт на объекте наблюдения, который будет регистрировать обращения платформы Астра Мониторинг и передавать на платформу метрики ОС Astralinux.
- Поле "Полное имя". Значение произвольное, например, "am-stand02.yandex.cloud"
- Поле "Тег". Выбираем ранее созданный тег "group : os : linux".

Значения полей также представлены на снимке экрана:



После создания объект мониторинга появится в нашем списке объектов наблюдения:



Как только объект мониторинга был добавлен, платформа Астра Мониторинг начала сбор метрик с указанного объекта.

5.2 Фиксация пороговых значений метрик для формирования события

Проверим пороговые значения для CPU перед тем, как перейти к эмуляции загрузки CPU на объекте наблюдения.

Подключимся к серверу платформы Астра Мониторинг через SSH (используем внешний IP адрес для подключения: XXX.XXX.XXX.XXXl).

После подключения выполним команду:

```
nano /home/user/astra-monitoring/vmalert/config/cpu.alert.yaml
```

В консоли отобразится содержимое файла cpu.alert.yaml:

```
groups:
- name: example
  interval: 30s
  concurrency: 2
  rules:
  - alert: CPU_Utilization
    expr: avg by (group, hostname, instance)
(irate(node_cpu_seconds_total{mode="idle"}[2m]) * 100) < 70
    #for: 1m
    labels:
      severity: warning
      team: ipa_support
      metric: cpu
    annotations:
      summary: "High CPU utilization on {{ $labels.hostname }} ({{
humanize $value }}% idle)"
      description: "{{ $labels.hostname }} ({{ $labels.instance }}) has
high CPU utilization for more than 2 minutes"
  - alert: CPU_Utilization
    expr: avg by (group, hostname, instance)
(irate(node_cpu_seconds_total{mode="idle"}[2m]) * 100) < 40
    #for: 1m
    labels:
      severity: critical
      team: ipa_support
      metric: cpu
    annotations:
      summary: "High CPU utilization on {{ $labels.hostname }} ({{
humanize $value }}% idle)"
      description: "{{ $labels.hostname }} ({{ $labels.instance }}) has
high CPU utilization for more than 2 minutes"
```

В конфигурационном файле для формирования события присутствует два порога: 40% и 70% простоя CPU.

Первое событие имеет критичность "warning". Событие формируется при простое CPU в течении двух минут < 70%. Другими словами, при загрузке CPU >= 30% формируется событие "warning".

Второе событие имеет критичность "critical". Событие формируется при простое CPU в течении двух минут < 40%. Другими словами, при загрузке CPU >= 60% формируется событие "critical".

По ходу выполнения сценария с помощью пакета "stress-ng" создадим нагрузку на процессор больше 60%.

5.3 Фиксация параметров оповещения (учетные данные для протокола SMTP)

При формировании события будет отправлено оповещение на почтовый ящик.

Рассмотрим блок "recevivers"

```
receivers:
- name: alert-mmost
  email_configs:
  - to: astramonitoringnotify@yandex.ru
    send_resolved: true
```

Письмо с содержанием описания события будет отправлено на astramonitoringnotify@yandex.ru.

5.4 Эмуляция действий на объекте наблюдений для изменения значений метрик

Осуществим вход через SSH на объект наблюдения XXX.XXX.XXX.XXX.

Используя пакет "stress-ng" создадим нагрузку на процессор (CPU) в районе в размере 75%. Необходимо запустить следующую команду:

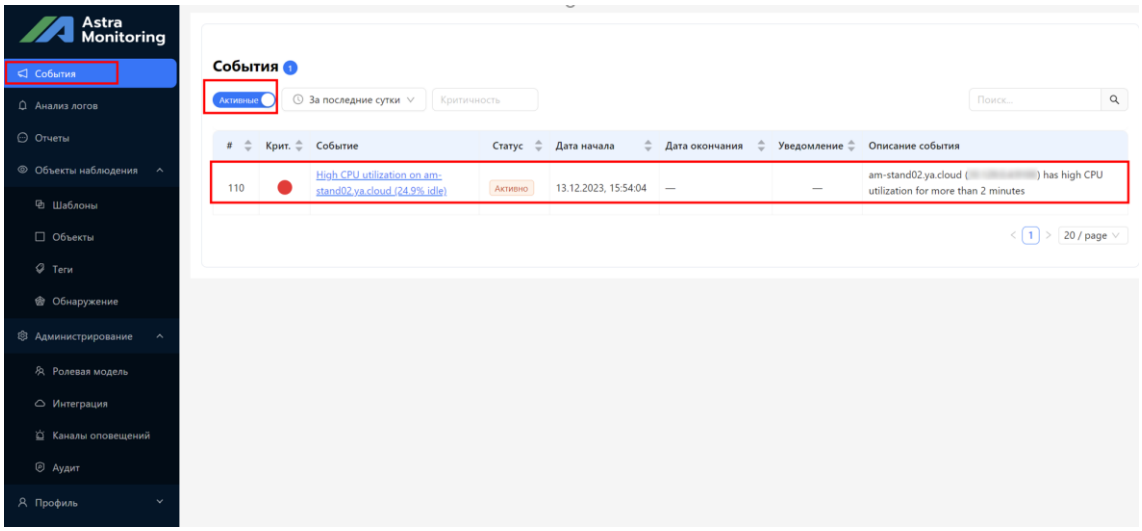
```
stress-ng -c 0 -l 75
```

Для контроля загрузки CPU можно использовать пакет "top".

5.5 Формирование события в соответствии с заданными пороговыми значениями. Отображение события в UI интерфейсе Астра Мониторинг

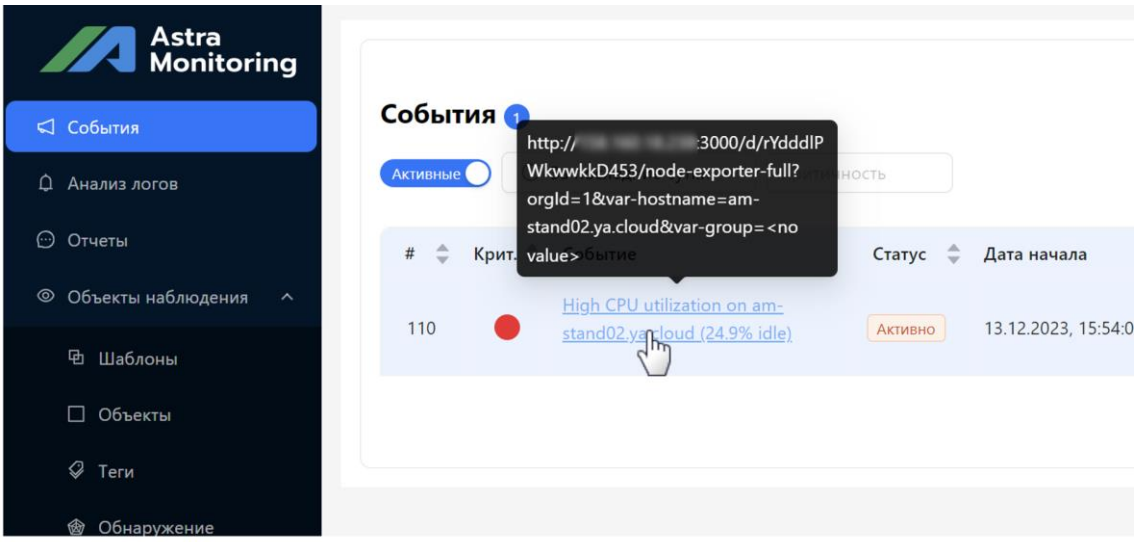
После выполнения действий п.5.4 необходимо перейти в графический интерфейс, выбрав вкладку "События".

Событие о загрузке CPU появится в интерфейсе события при выбранном переключателе "Активные" через TODO_УКАЗАТЬ_ВРЕМЯ:

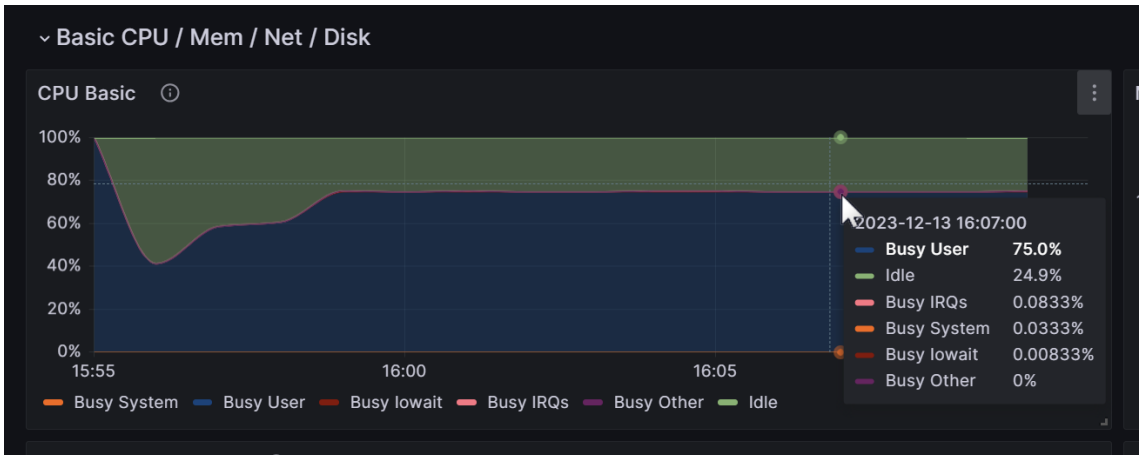


Событие показывает, что CPU высоко загружен (24.9% в состоянии "Idle", что соответствует 75.1% загрузки CPU).

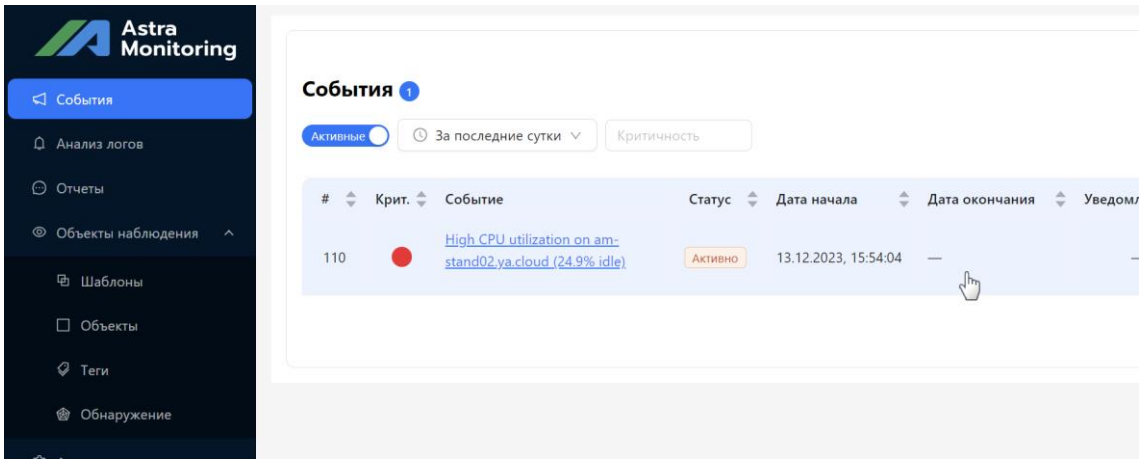
При щелчке ЛКМ по гиперссылке можно осуществить переход к панели с метриками объекта наблюдения.



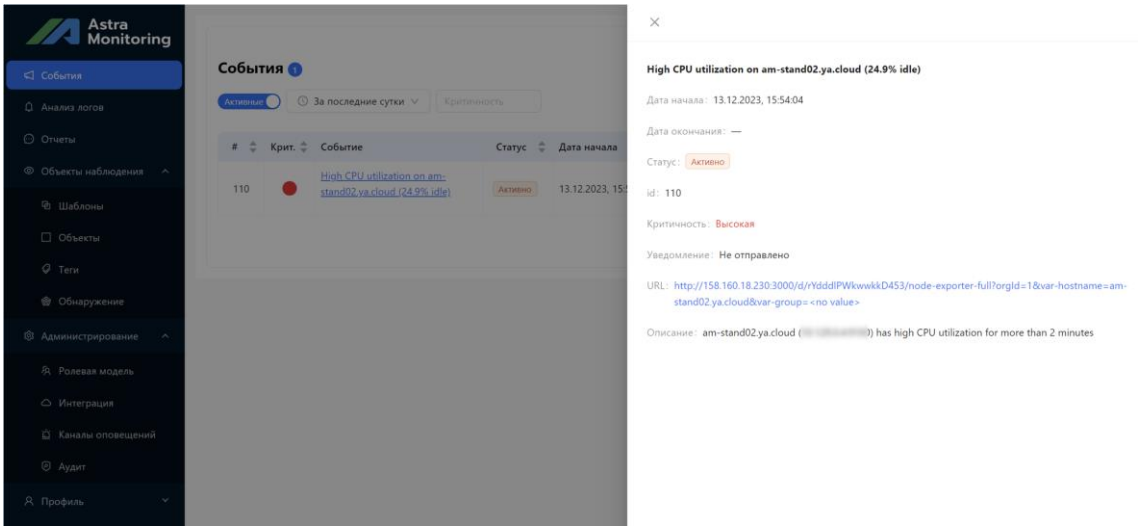
На панели с метриками в разделе "Basic CPU / Mem / Net / Disk" на графике "CPU Basic" можно видеть соответствующий рост загрузки CPU:



Дополнительную информацию о событии можно посмотреть, осуществив ЛКМ мыши на строке события:

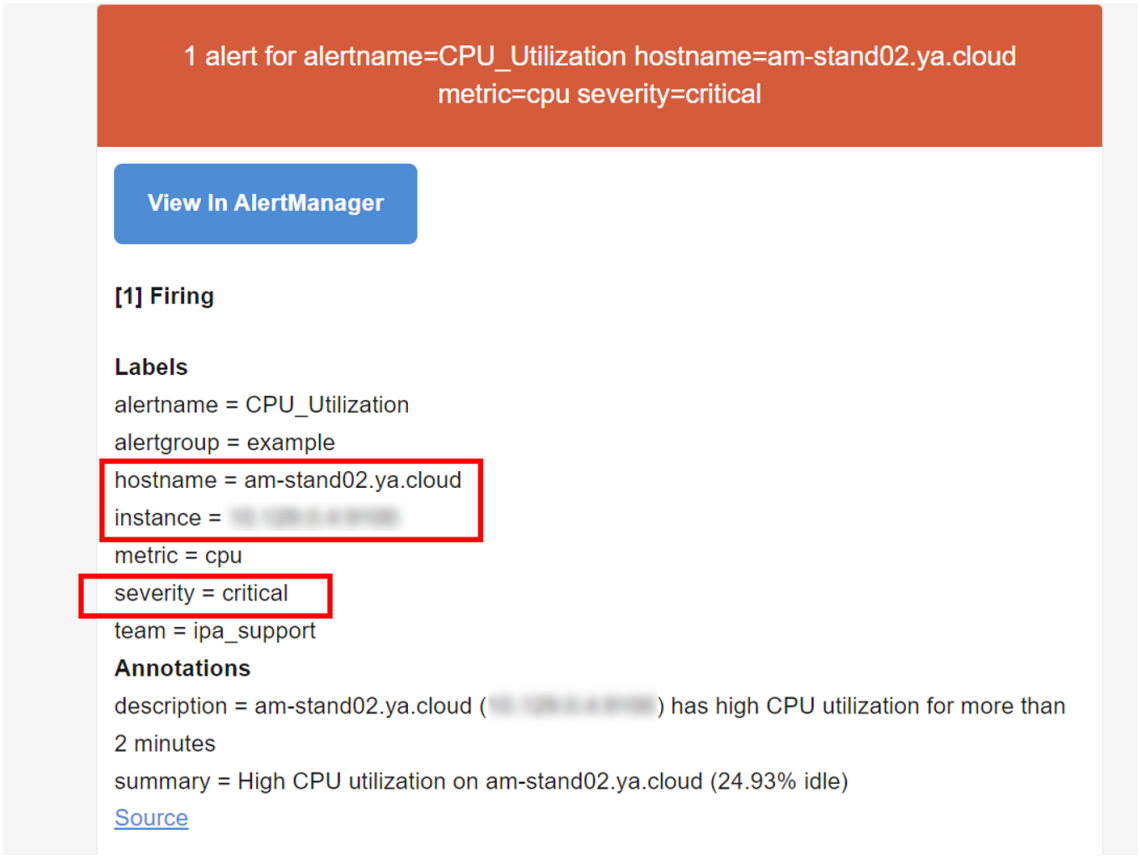


Дополнительная информация о событии отобразится в виде дополнительного окна в правой части WEB-интерфейса:



5.6 Формирование оповещения в соответствии с заданными почтовыми учетными данными пользователя

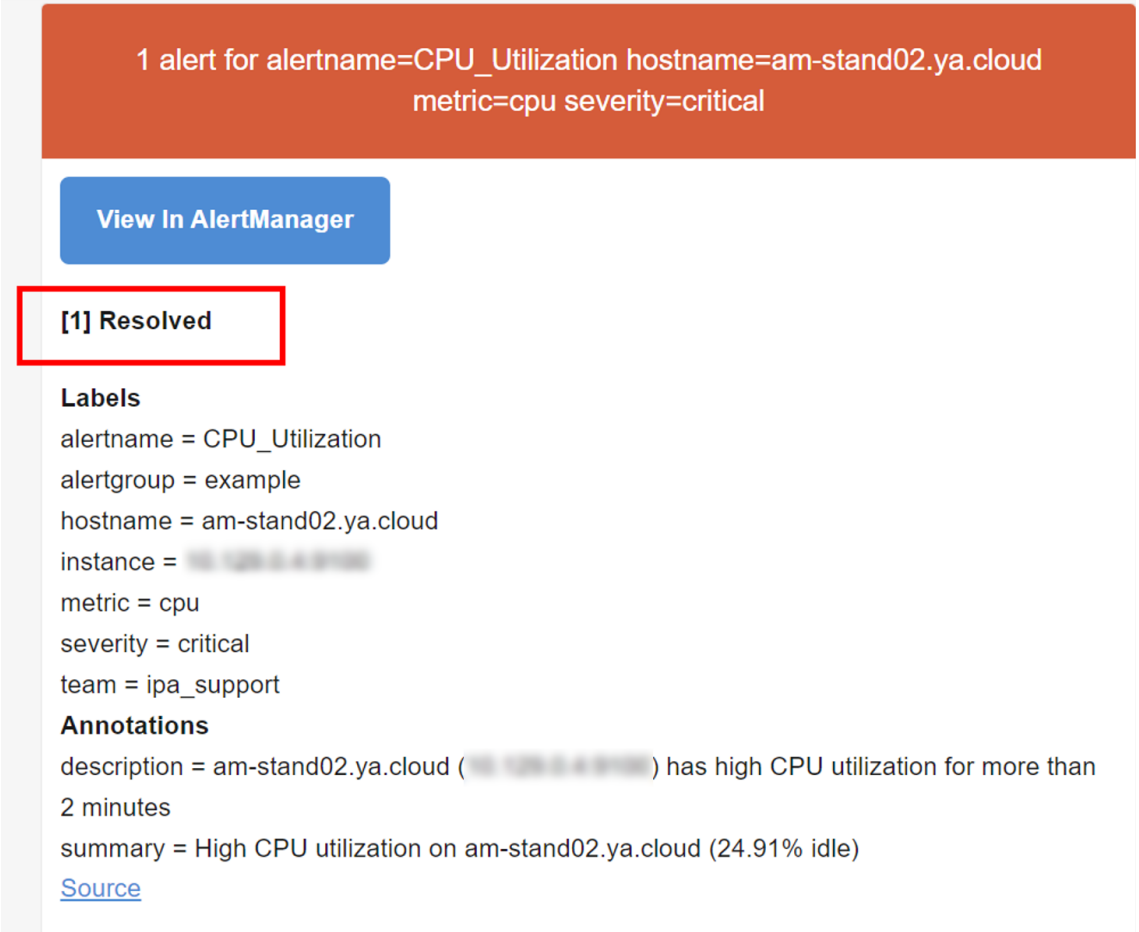
На почтовый ящик astramonitoringnotify@yandex.ru направлено письмо с описанием события о загрузке CPU:



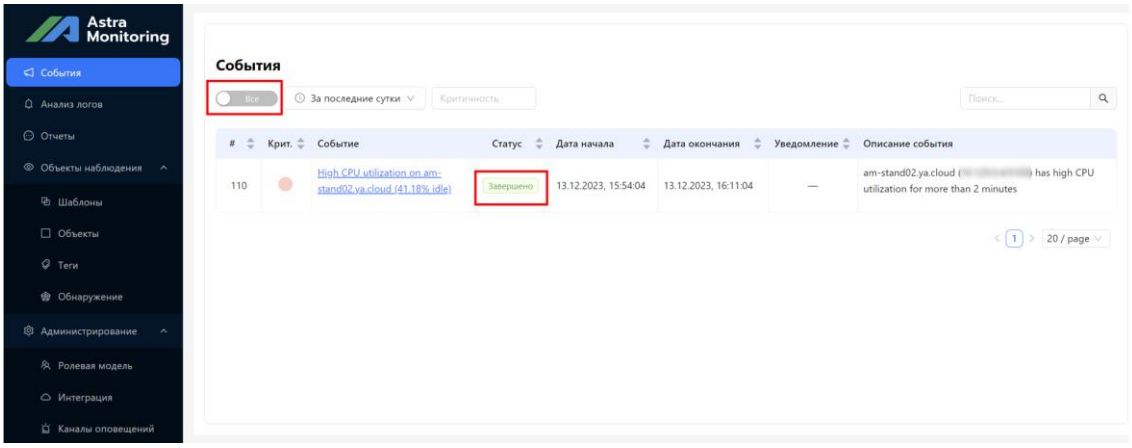
5.7 Сброс события

В пункте "5.4 Эмуляция действий на объекте наблюдений для изменения значений метрик" была запущена команды для создания нагрузки. Прервем ее выполнение, нажав "Ctrl + C".

Спустя некоторое время (приблизительно 5 минут)) на почтовый ящик получим нотификацию об успешном разрешении события:



При этом, в пользовательском интерфейса во вкладке "События" созданное ранее событие будет отображено со статусом "Завершено" (для отображения событий со статусом "Завершено" необходимо в графическом элементе выбрать отображать все события с помощью соответствующего переключателя).


Astra Monitoring
События
Анализ логов
Отчеты
Объекты наблюдения
Шаблоны
Объекты
Теги
Обнаружение
Администрирование
Ролевая модель
Интеграция
Каналы оповещений
События
Все
За последние сутки
Критичность
Поиск...
Крит. Событие Статус Дата начала Дата окончания Уведомление Описание события
110
High CPU utilization on am-stand02.ya.cloud (41.18% idle)
Завершено
13.12.2023, 15:54:04
13.12.2023, 16:11:04
—
am-stand02.ya.cloud (has high CPU utilization for more than 2 minutes
1
20 / page

6 Заключение

По окончании выполнения инструкции вы получили краткое представление о назначении платформы Астра Мониторинг. Полученный опыт по использованию платформы для сбора метрик с демонстрационного стенда (а также эмуляций событий и отправка уведомлений) поможет применять решение Астра Мониторинг для сбора информации с объектов наблюдения под управлением ОС Astralinux, формировании событий и оповещений для незамедлительной реакции на изменения в IT-инфраструктуре.

7 Справочная информация

В случае возникновения вопросов, касающихся выполнения сценария и доступа к узлам платформы Астра Мониторинг, а также доступа к объекту наблюдения, необходимо оформить письменное обращение на следующие почтовые адреса:

- ptrubenkov@astralinux.ru. Трубенков Павел Леонидович, руководитель продукта.