

# Alertmanager

Alertmanager — это компонент экосистемы Prometheus, который управляет срабатыванием и обработкой оповещений. Он играет ключевую роль в системе мониторинга на базе Prometheus, организуя оповещения о различных событиях и аномалиях в работе кластера.

## 1. Основные функции Alertmanager

---

- **Маршрутизация оповещений.** Вы можете настроить маршруты для оповещений в зависимости от их источников, приоритетов и других метаданных. Это позволяет направлять сообщения в разные команды или сервера в зависимости от типа проблемы.
- **Группировка оповещений.** Alertmanager агрегирует схожие оповещения в единую группу, что позволяет уменьшить избыточность и снизить уровень информационного шума. Такая агрегация упрощает обработку инцидентов и повышает эффективность управления системой оповещений.
- **Уведомления.** Alertmanager поддерживает интеграцию с различными системами оповещения, такими как электронная почта, Slack, PagerDuty, Telegram и многими другими, что делает его гибким для использования в различных рабочих процессах.
- **Подавление оповещений.** Alertmanager позволяет временно скрывать (или подавлять) определенные оповещения, которые находятся в процессе обработки.
- **Управление тишиной (Silence):** Функция позволяет временно "заглушать" определенные оповещения, чтобы избежать избыточных уведомлений в случае плановых работ или ожидаемых сбоев.

## 2. Особенности Alertmanager в Nova Container Platform

---

### 2.1. Оповещения в веб-консоли Nova Container Platform

В Nova Container Platform оповещения настроены на странице **Observer > Система оповещений**.

- В блоке **Оповещения** видны все текущие проблемы в кластере.
- В блоке **Управление оповещениями** видны все правила остановки оповещения. Также можно создать правило для заглушения оповещений.

- В блоке **Правила оповещения** видны все настроенные правила для оповещений, включая как предустановленные, так и созданные пользователем.



Alertmanager по умолчанию добавляет в настройку *AlertmanagerConfig* фильтр по имени Namespace, в котором создан ресурс *AlertmanagerConfig*.

Это значит, что вы сможете работать только с оповещениями из этого Namespace, например:

Если оповещение было создано через *AlertmanagerConfig* в Namespace `test`, в настройках блока `route` возможно работать только с фильтрами, связанными с этим Namespace.

Если вы укажете другое Namespace в блоке `route`, то *AlertmanagerConfig* перестанет работать, и уведомления не будут отправляться.

Чтобы фильтр не добавлялся по умолчанию и была возможность настраивать фильтры по нескольким Namespace в одном *AlertmanagerConfig*, выполните следующие шаги:

1. В веб-консоли Nova Container Platform перейдите на вкладку **Administration > CustomResourceDefinitions**, найдите ресурс *Kustomization*, перейдите на вкладку **Экземпляры**, найдите `nova-release-prometheus-post`.
2. На вкладке **YAML** добавьте патч:

```
spec:
  patches:
    - patch: |-
        - op: add
          path: /spec/alertmanagerConfigMatcherStrategy
          value:
            type: None
      target:
        kind: Alertmanager
        name: main
        namespace: nova-monitoring
```

YAML |

## 2.2. Оповещения в Alertmanager

Для Alertmanager в Nova Container Platform создан ресурс *Ingress* с именем `nova-alertmanager-main` в Namespace `nova-monitoring`.

При использовании ссылки из этого ресурса вы попадёте в веб-консоль Alertmanager, где доступны текущие оповещения, настройки Alertmanager и другие функции.

## 3. Примеры

### 3.1. Пример создания правила для оповещения

1. В веб-консоли Nova Container Platform перейдите на вкладку **Administration > CustomResourceDefinitions**, найдите ресурс *PrometheusRule*, перейдите на вкладку **Экземпляры**, нажмите на [ **Создать PrometheusRule** ].
2. В качестве примера используйте следующий манифест:

```
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  name: pod-alerts ①
  namespace: nova-monitoring ②
spec:
  groups:
    - interval: 10m ③
      name: pod-status ④
      rules:
        - alert: PodFailed ⑤
          annotations:
            description: >-
              Pod {{ $labels.pod }} in namespace {{ $labels.namespace }} has
              been in Failed state for more than 5 minutes.
            summary: >-
              Pod in {{ $labels.namespace }}/{{ $labels.pod }} is in Failed
              state
          expr: 'kube_pod_status_phase{phase="Failed"} > 0' ⑥
          for: 5m ⑦
          labels:
            severity: critical ⑧
        - alert: PodPending ⑤
          annotations:
            description: >-
              Pod {{ $labels.pod }} in namespace {{ $labels.namespace }} has
              been in Pending state for more than 5 minutes.
            summary: >-
              Pod in {{ $labels.namespace }}/{{ $labels.pod }} is in Pending
              state
          expr: 'kube_pod_status_phase{phase="Pending"} > 0' ⑥
          for: 5m ⑦
          labels:
            severity: critical ⑧
        - alert: PodUnknown ⑤
          annotations:
            description: >-
              Pod {{ $labels.pod }} in namespace {{ $labels.namespace }} has
              been in Unknown state for more than 5 minutes.
            summary: >-
              Pod in {{ $labels.namespace }}/{{ $labels.pod }} is in Unknown
              state
          expr: 'kube_pod_status_phase{phase="Unknown"} > 0' ⑥
          for: 5m ⑦
```

```
labels:
  severity: critical ⑧
```

1. Имя ресурса: может быть любым.
2. Namespace: обязательно должно быть тем, что в примере.
3. Интервал: определяет, как часто оцениваются правила в группе.
4. Имя группы оповещений.
5. Название оповещения.
6. PromQL выражение: определяет условие срабатывания данного оповещения.
7. Устанавливается период времени, в течение которого условие должно оставаться истинным, прежде чем сработает оповещение. В данном случае — 5 минут.
8. Метки назначаются оповещениям и позволяют управлять их маршрутизацией в Alertmanager.

## 3.2. Пример настройки оповещения

### 3.2.1. Пример настройки оповещения с фильтром по умолчанию

В данном примере будет настроено оповещение по электронной почте.

1. Создайте новое Namespace с именем `test`.
2. **(Опционально)** Если ваш SMTP-сервер требует авторизации для отправки писем, создайте секрет в Namespace `test`, содержащий пароль от почтового ящика отправки оповещений.
3. В веб-консоли Nova Container Platform перейдите на вкладку **Administration > CustomResourceDefinitions**, найдите ресурс `AlertmanagerConfig`, перейдите на вкладку **Экземпляры**, нажмите на [ **Создать AlertmanagerConfig** ].
4. Используйте следующий манифест:

YAML | 

```
apiVersion: monitoring.coreos.com/v1alpha1
kind: AlertmanagerConfig
metadata:
  name: test-alertmanager-config ①
  namespace: test ②
spec:
  receivers:
    - emailConfigs: ③
      - authIdentity: email-notification@test.local
        authPassword: ④
        key: pwd
        name: email
        authUsername: email-notification
```

```

    from: email-notification@test.local
    requireTLS: true
    smarthost: 'smtp.test.local:587'
    to: test-user@test.local
  name: email-receiver
route: ⑤
  continue: true
  groupBy:
    - alertname
  groupInterval: 5m
  groupWait: 30s
  receiver: email-receiver
  repeatInterval: 10m

```

1. Имя ресурса: может быть любым.
2. Имя Namespace, для которого будут работать оповещения.
3. Настройка оповещения по электронной почте. Также можно выбрать любой другой способ оповещения, представленный в подсказке на боковой панели.
4. Имя секрета, содержащего пароль от почтового ящика и его ключ.
5. Настройка маршрута. В текущей конфигурации будут приходить все оповещения из Namespace `test`.

### 3.2.2. Пример настройки оповещения без фильтра по умолчанию

В данном примере будет настроено оповещение в Telegram.

1. Создайте новое Namespace с именем `test`.
2. Создайте секрет в Namespace `test`, который будет содержать ID бота Telegram.
3. В веб-консоли Nova Container Platform перейдите на вкладку **Administration** > **CustomResourceDefinitions**, найдите ресурс *AlertmanagerConfig*, перейдите на вкладку **Экземпляры**, нажмите на [ **Создать AlertmanagerConfig** ].
4. Используйте следующий манифест:

YAML | 

```

apiVersion: monitoring.coreos.com/v1alpha1
kind: AlertmanagerConfig
metadata:
  name: test-alertmanager-config ①
  namespace: test ②
spec:
  receivers:
    - name: telegram-receiver ③
      telegramConfigs:
        - apiURL: 'https://api.telegram.org'
          botToken: ④

```

```
    key: bot_id
    name: telegram-bot
    chatID: -1234567890 ⑤
    sendResolved: true ⑥
route: ⑦
  continue: true
  groupBy:
    - alertname
  groupInterval: 5m
  groupWait: 30s
  matchers: ⑧
    - matchType: =~
      name: namespace
      value: ^(nova|kube).*
    - matchType: =~
      name: severity
      value: critical
  receiver: telegram-receiver
  repeatInterval: 10m
```

1. Имя ресурса: может быть любым.
2. Имя Namespace.
3. Настройка оповещения в Telegram. Также можно выбрать любой другой способ оповещения, представленный в подсказке на боковой панели.
4. Имя секрета, содержащего пароль ID Telegram-бота.
5. ID Telegram-чата.
6. Будут приходить оповещения о закрытии проблемы.
7. Настройка маршрута.
8. При составлении фильтров можно использовать данные с веб-консоли Alertmanager.

### 3.3. Пример расчета времени получения оповещения

Подставляем значение из примеров в формулу  $a + b + c + d = \text{time}$ .

Получается следующее:

$$30s + 10m + 5m + 30s = 16m$$

MATH | 

Т.е. оповещение будет получено примерно через 16 минут.

# Мониторинг сертификатов платформы

## 1. Обзор

Под `x509-certificate-exporter` в namespace `nova-monitoring` отвечает за мониторинг сертификатов платформы.

Под проверяет все секреты с типом `--secret-type=kubernetes.io/tls:tls.crt` и собирает информацию о дате окончания действия сертификата.

## 2. Grafana

Метрики с пода `x509-certificate-exporter` выведены в виде отдельного дашборда в Grafana с именем `Certificates Expiration`.

## 3. Оповещения

В Nova Container Platform Platform предварительно настроены оповещения для мониторинга сертификатов.

Весь список оповещений можно найти в разделе **Мониторинг > Правила оповещения**.

Например, оповещения по окончании срока действия сертификатов начнут появляться на главной странице веб-консоли платформы за 14 дней до окончания срока действия сертификата.

Если настроить [AlertManager](#), то оповещения будут приходить выбранным методом. Например, на электронную почту.

## 4. nova-ctl

В `nova-ctl` можно проверить срок действия сертификатов:

```
nova-ctl certs check-expiration
```

BASH | 

Обновить сертификаты:

```
nova-ctl certs renew
```

BASH | 

