

Модификаторы labels

Для модификации labels в Prometheus есть 2 варианта: использовать **relabel_configs** или **metric_relabel_configs**. **relabel_configs** производит все модификации до scraping, а **metric_relabel_configs** производит изменения после scraping, но перед сохранением в базу.

metric_relabel_configs обычно используется в двух случаях: когда надо удалить метрику или переименовать ее.

NB! metric_relabel_configs не применяется к автоматически генерируемым метрикам, таким как up.

Модификаторы

Далее приведен список действий, которые можно производить с метриками. Для metric_relabel_configs и relabel_configs они одинаковы.

1. Labels action

С помощью action можно задавать правила scraping. Возможны значения: keep или drop. Если используется keep, будет производиться scraping метрик только с этим labels. В случае, если drop, то метрики с этим labels не будут сохраняться.

Пример. В этом случае будет производиться scraping только метрик, которые имеют label _meta_openstack_tag равный db:

```
scrape_configs:
- job_name: 'openstack'
  openstack_sd_configs:
  - identity_endpoint: https://api.selvpc.ru/identity/v3
    port: 9100
    domain_name: '82113'
    username: '<user name>' # s00000
    project_name: '<Project name>' # project_s00000
    password: '<Password>'
    role: 'instance'
    region: '<Region>' # ru-3
    refresh_interval: 30s
relabel_configs:
- source_labels: [_meta_openstack_tag]
  regex: db
  action: keep
```

Пример. В этом случае будет производиться scraping метрик, у которых значение labels _meta_openstack_tag равно: front, либо db, либо back не будут сохранены.

```
scrape_configs:
- job_name: 'openstack'
  openstack_sd_configs:
  - identity_endpoint: https://api.selvpc.ru/identity/v3
    port: 9100
    domain_name: '82113'
    username: '<user name>' # s00000
    project_name: '<Project name>' # project_s00000
    password: '<Password>'
    role: 'instance'
    region: '<Region>' # ru-3
    refresh_interval: 30s
  relabel_configs:
  - source_labels: [_meta_openstack_tag]
    regex: front|db|back
    action: drop
```

NB! Важный момент: все метки, начинающиеся с _ , в конце обработки отбрасываются и не попадают в target labels.

NB! Для сокращения правил лучше использовать regExp.

2. Labels replace

Например, если есть несколько команд, работающих над проектом, они могут использовать разное именование одних и тех же меток. В этом случае можно привести все к единому виду:

```
relabel_configs:
- source_labels: [team]
  regex: '(.*)ing'
  replacement: '${1}'
  target_label: team
  action: replace
```

Обратите внимание: тут мы снова используем regex.

Пример. Если метрика имеет метку team со значением developing, то в результате преобразования значение team будет заменено на develop.

NB! Несмотря на то, что тут возможно использование `regExp`, их надо использовать с осторожностью, чтобы не заменить лишние метки.

3. Labels `labelmap`

Label map позволяет использовать имена source labels как имена для target labels. Именем для новой метки будет часть, удовлетворяющая регулярному выражению. Например, если есть source label `__meta_filepath`, то конфигурация:

```
relabel_configs:  
- regex: __meta_(filepath)  
  replacement: '${1}'  
  action: labelmap
```

добавит target label `filepath` со значением равным значению `__meta_filepath`.

4. Labels `list`

Некоторые SD не имеют значений `key:value`, вместо которых у них список. Вы можете за счет регулярного выражения выделить нужное значение. В этом примере, если в списке тэгов `consul` присутствует один из тэгов с именем: `prod|staging|dev`, то имя этого тэга сохранится как значение `label env`.

```
relabel_configs:  
- source_labels: [__meta_consul_tag]  
  regex: '.*(prod|staging|dev),.*'  
  target_label: env
```

6. Labels `labeldrop`

Все метки, удовлетворяющие регулярному выражению, будут удалены.

```
relabel_configs:  
- regex: sd  
  action: labeldrop
```

7. Labels `labelkeep`

Все метки, не удовлетворяющие регулярному выражению, будут удалены.

```
relabel_configs:  
- regex: sd  
  action: labelkeep
```

NB! Этот модификатор необходимо использовать с осторожностью, так как все служебные метки(относятся к source labels) тоже удаляются. В примере выше будут

удалены, в частности, такие метки, как: `__address__` и `__scheme__`, что делает невозможным проведение scraping с exporter.