How to change the email template and filter alert names to a specific receiver in Prometheus

Table of contents

Issue

Prometheus was installed by the HPE Professional Services (PS) team. The customer wanted to change the email template and filter alert name to a specific receiver. When an Airflow job fails, they want the alert to be sent to a specific individual or group with a custom email template.

Environment

Ezmeral Unified Analytics Version: 1.3

Cause

The customer's goal was to ensure that specific alerts, such as Airflow job failures, are sent to designated recipients using a customized email template. They attempted to modify the email template in Prometheus, but unfortunately their changes did not seem to be taking effect. Despite updating the configuration, they continue to receive the default email template, and the alerts are only being sent to the original recipient rather than to the intended recipients.

Resolution

This guide explains how to implement custom email templates for Prometheus alerts using Alertmanager. We will walk through creating custom templates, storing them in a ConfigMap, modifying Alertmanager configurations, and routing specific alerts to designated receivers.

Part 1: Preparing the Templates and ConfigMap:

1. **Create Template Files:** Start by creating two separate template files that will define the subject line and the HTML content of the email alerts.

   - **custom_mail_subject.tmpl**: This file will hold the subject of the alert emails.
     ```
     [Alerting] {{ .CommonLabels.alertname }}
     ```

   - **custom_mail_html.tmpl**: This file will define the body of the email using HTML.
     ```
     <html>
     <body>
       <h3>{{ .CommonLabels.alertname }}</h3>
       <p>{{ .CommonAnnotations.description }}</p>
     </body>
     </html>
     ```

2. **Create a ConfigMap:** Next, create a ConfigMap in the Prometheus namespace to store these template files. The ConfigMap will make the templates available to Alertmanager.

   ```
   kubectl create configmap alertmanager-templates -n prometheus \
     --from-file=custom_mail_subject.tmpl \
     --from-file=custom_mail_html.tmpl
   ```

Part 2: Modifying the Alertmanager Custom Resource (CR):

1. **Edit Alertmanager Custom Resource (CR):** Modify the Alertmanager CR to mount the ConfigMap containing the templates. This will ensure that Alertmanager can access the templates stored in the ConfigMap.

   - Run the following command to edit the Alertmanager CR:
     ```
     kubectl edit alertmanager prometheus-kube-prometheus-alertmanager -n prometheus -o ya
     ```

   - Add the following under the `spec` section to mount the ConfigMap:
     ```
     spec:
       volumes:

         - name: alertmanager-templates
           configMap:
     ```

```
      name: alertmanager-templates
  containers:
    - name: alertmanager
      volumeMounts:
        - name: alertmanager-templates
          mountPath: /etc/alertmanager/templates
```

Part 3: Updating the Alertmanager Configuration (Secret):

1. **Extract Existing Configuration:** To update the Alertmanager configuration, you need to extract the existing `alertmanager.yaml` from the Secret:

   - Get the Secret containing the Alertmanager configuration:
     `kubectl get secret alertmanager-prometheus-kube-prometheus-alertmanager -n prometheus`
   - Decode the Base64-encoded `alertmanager.yaml` file:
     `echo '<base64_config_data>' | base64 -d > alertmanager.yaml`

2. **Modify `alertmanager.yaml`:** Now, you will modify the `alertmanager.yaml` file to reference your templates and configure the alert routing.

   - **Add template references:** In the configuration, add a reference to the templates:

     ```
     templates: - '/etc/alertmanager/templates/*.tmpl'
     ```

   - **Update the `email_configs` section:** Define how and to whom emails should be sent, using the custom templates:
     receivers:

     ```
     - name: 'email_receiver'
       email_configs:
       - to: 'lalit1@hpe.com'
         html: '{{ template "custom_mail_html.tmpl" . }}'
         headers:
           subject: '{{ template "custom_mail_subject.tmpl" . }}'
     ```

   - **Add additional receivers and routes:** Configure specific alert receivers based on alert types (e.g., for Kubeflow or Airflow):

     ```
     global:
       resolve_timeout: 5m
       smtp_from: <email_from>
       smtp_smarthost: <smtp_host>
       smtp_require_tls: <true/false>

     route:
       group_by: ['alertname']
       group_wait: 30s
       group_interval: 5m
       repeat_interval: 4h
       receiver: 'kubeflow_receiver'

       routes:
         - match:
             alertname: 'Kubeflow job failing'
           receiver: 'kubeflow_receiver'
         - match:
             alertname: 'Airflow job failing'
           receiver: 'airflow_receiver'

     receivers:
     - name: 'kubeflow_receiver'
       email_configs:
       - to: 'kubeflow_admin@hpe.com'
         html: '{{ template "kubeflow_html.tmpl" . }}'
         headers:
           subject: '{{ template "kubeflow_html.tmpl" . }}'
     - name: 'airflow_receiver'
       email_configs:
       - to: 'admin_airflow@hpe.com'
         html: '{{ template "airflow_html.tmpl" . }}'
         headers:
           subject: '{{ template "airflow_subject.tmpl" . }}'
     ```

3. **Encode and Update Secret:**

   - After making the necessary changes to `alertmanager.yaml`, encode the file back to Base64:
     `cat alertmanager.yaml | base64 -w0`
   - Replace the existing Base64 data in the Secret:
     `kubectl edit secret alertmanager-prometheus-kube-prometheus-alertmanager -n prometheus`

Part 4: Restarting the Pods:

To apply the changes, restart the Prometheus Operator and Alertmanager pods:

1. **Restart the Alertmanager pod:**

   ```
   kubectl delete pod -n prometheus -l alertmanager=prometheus-kube-prometheus-alertmanager
   ```

2. **Restart the Prometheus Operator pod:**

   ```
   kubectl delete pod -n prometheus -l app=kube-prometheus-stack-operator
   ```

By following above steps, can introduce custom templates into your Prometheus/Alertmanager setup and filter specific alerts to be sent to the appropriate receivers.