

Frontend - Take Home Test

Question:

Design and Implement a web site for Oodle observability alerting feature.

Workflows in Alerting feature you need to design for:

- Create an Alert rule
- View/Edit an Alert rule
- Listing all Alert rules
- Listing all Alerts (Look at [Given section](#) point 2 on how alerts are automatically created)
- Select and view Alert details.

Note: Alert notification is not in scope for this question. Simply listing the alerts is sufficient.

Prelude:

1. Customers emit various metrics to oodle.
 - a. An example is customer emitting server latency metric for a server called api-server. The latency value of the metric is in milliseconds. This is emitted in the following format: `{ "metric name" : "server_latency_ms", "labels": [{"service_name": "api-server"}], "samples": [{"timestamp": "1704250373", "value": 1000}] }`
 - i. The server_latency_ms metric for service_name - "api-server" at time 1704250373 has the value 1000 (1 second)
 - b. Similarly different types of metrics are emitted by the customer and stored in oodle's database.
 - c. These metrics can be queried by the customer from the UI.
2. Apart from looking at the metrics in the UI, an important component in Observability is Alerting.
 - a. Alerting is the ability for the customer to configure certain rules so that they get notified when the rule condition evaluates to be true. Eg: Alert the customer when the server_latency_ms metric value is greater than 5 seconds for 10 minutes.
 - b. The customer configures alert rules in oodle. Oodle periodically evaluates the alert rules and if the rule condition is satisfied, it creates an alert that will be sent to the customer through some notification channel (slack, pager duty etc)
 - c. You can read the following doc on Alerting by Grafana to better understand "Alerting":

- i. <https://grafana.com/docs/grafana/latest/alerting/>
 - ii. <https://grafana.com/docs/grafana/latest/alerting/fundamentals/alert-rules/>
3. Alert Rule: Create an alert rule to notify an alert when the server latency metric from service api-server is greater than 5 seconds for more than 10 mins. The latency metrics are emitted in milliseconds. The rule should be evaluated every 10 mins.
 - a. Alert Rule "name": "Server latency alert"
 - b. Alert Rule "expression": "server_latency_metrics{service='api-server'} > 5000"
 - c. Alert Rule "for": 1257893400000000000 (the value for "for" is represented in nanoseconds. This value represents 10 mins in nanoseconds)
 - d. Alert Rule "interval": 600 (the value for "interval" is represented in seconds. This value represents 10 mins)
4. Alert: The oodle alerting will evaluate rules periodically based on the "interval" field. For the above alert rule. Every 10 mins the expression "server_latency_metrics{service='api-server'} > 5000" will be evaluated. If the expression is true for the time defined by the "for" field. An alert is created. The alert would look like:
 - a. {


```

          "Name": "Server latency alert",
          "alertRuleUUID": "uuid of the above alert_rule", // The alert was created for this rule.
          "expression": "server_latency_metrics{service='api-server'}",
          "Status": "Firing" // It is in pending state if the alert expression is true but it has not been true for 10 mins yet
          }
```

Outcome:

Present a working demo for:

1. Create around 10 alert rules.
2. List all the alert rules.
3. Show all alerts along with their status.
4. Show the details about an alert.
5. Update an alert rule.

Tip: Look at how alert rules are created for various observability products for inspiration. Eg: Grafana, Splunk (signalfx), Datagog, Chronosphere.

Given:

1. We have implemented a simple backend server with a dockerfile for you to run locally. Integrate with this backend server for building the website.

- a. Prerequisite for running backend server:
 - i. Download docker engine: <https://docs.docker.com/engine/install/>
 - ii. **Download the server:**
https://drive.google.com/file/d/1tcoMFadLeFC2ttY9HwHXy03o5bRb_z6k/view?usp=sharing
- b. Steps to run the backend server.

Unset

```
# navigate to the alerts_backend_mock folder that you downloaded.
cd> alerts_backend_mock
> docker build -t alerts_backend_mock .

# Verify if you can see the docker image for alerts_backend_mock
> docker image ls

> docker run -it --rm -p 8080:8080 alerts_backend_mock
2024/01/03 18:01:58 Server started
```

2. For each alert rule you create, the backend server will automatically create an alert. The alert rule is evaluated to true using a probabilistic function. Each alert rule will have at most 1 alert. You can refer to `alerts_backend_mock/go/alert_rules.go` :
`EvaluateAlertRules` function if you are interested in understanding the probabilistic function.
3. **Example curl commands for the backend server:**
<https://drive.google.com/file/d/1QHbnBhptAkyorC11g7epxduooKJrLrr4/view?usp=sharing>
4. Swagger API spec for the backend server: You can copy paste the above spec to <https://editor.swagger.io/> to render the spec.

Unset

```
swagger: '2.0'

info:
  version: 0.0.1
  title: Oodle Alerts API
  description: API of the Oodle Alert management.

consumes:
  - "application/json"
produces:
  - "application/json"
```

```
paths:
  /alerts:
    get:
      tags:
        - alerts
      description: Get the list of alerts. Alerts are instances created from an alerting
rule after evaluating the alert-rule expression.
      responses:
        '200':
          description: Get alerts
          schema:
            '$ref': '#/definitions/alerts'
        '400':
          $ref: '#/responses/BadRequest'
        '500':
          $ref: '#/responses/InternalServerError'
  /alert/{uuid}:
    get:
      tags:
        - alerts
      description: Get alert by uuid
      parameters:
        - in: path
          name: uuid
          type: string
          format: uuid
          description: uuid of the alert to get
          required: true
      responses:
        '200':
          description: The alert with uuid
          schema:
            '$ref': '#/definitions/alert'
        '400':
          $ref: '#/responses/BadRequest'
        '500':
          $ref: '#/responses/InternalServerError'

  /alert-rules:
    get:
      tags:
        - alert-rules
      description: Get the list of created alert rules
      responses:
```

```
'200':
  description: Get alert rules response
  schema:
    '$ref': '#/definitions/alertRules'
'400':
  $ref: '#/responses/BadRequest'
'500':
  $ref: '#/responses/InternalServerError'

/alert-rule/{uuid}:
get:
  tags:
    - alert-rules
  description: Get alert rule by uuid
  parameters:
    - in: path
      name: uuid
      type: string
      format: uuid
      description: uuid of the alert rule to get
      required: true
  responses:
    '200':
      description: The alert rule with uuid
      schema:
        '$ref': '#/definitions/alertRule'
    '400':
      $ref: '#/responses/BadRequest'
    '500':
      $ref: '#/responses/InternalServerError'
put:
  tags:
    - alert-rules
  description: Update alert rule by uuid
  parameters:
    - in: path
      name: uuid
      type: string
      format: uuid
      description: uuid of the alert rule to update
      required: true
    - in: body
      name: alertRule
      description: The alert rule to create
```

```

      required: true
      schema:
        $ref: '#/definitions/alertRule'
    responses:
      '200':
        description: The updated alert rule
        schema:
          '$ref': '#/definitions/alertRule'
      '500':
        $ref: '#/responses/InternalServerError'
      '400':
        $ref: '#/responses/BadRequest'

/alert-rule:
  post:
    tags:
      - alert-rules
    description: Create a new Alert Rule
    parameters:
      - in: body
        name: alertRule
        description: The alert rule to create
        required: true
        schema:
          $ref: '#/definitions/postableAlertRule'
    responses:
      '200':
        description: The created alert rule
        schema:
          '$ref': '#/definitions/alertRule'
      '400':
        $ref: '#/responses/BadRequest'
      '500':
        $ref: '#/responses/InternalServerError'

/clear:
  post:
    tags:
      - admin
    description: Clears all the data stored in-memory
    responses:
      '200':
        description: success response
      '500':

```

```
$ref: '#/responses/InternalServerError'
```

```
responses:
```

```
  BadRequest:
```

```
    description: Bad request
```

```
    schema:
```

```
      type: string
```

```
  InternalServerError:
```

```
    description: Internal server error
```

```
    schema:
```

```
      type: string
```

```
definitions:
```

```
  alerts:
```

```
    type: array
```

```
    items:
```

```
      $ref: '#/definitions/alert'
```

```
  alert:
```

```
    type: object
```

```
    properties:
```

```
      id:
```

```
        type: string
```

```
        format: uuid
```

```
        description: uuid to identify the alert
```

```
      name:
```

```
        type: string
```

```
        description: name of the alert derived from the alert rule
```

```
      expression:
```

```
        description: The alert-rule expression evaluated to create the alert
```

```
      startsAt:
```

```
        type: string
```

```
        format: date-time
```

```
        description: The time the alert started at
```

```
      updatedAt:
```

```
        type: string
```

```
        format: date-time
```

```
        description: The time the alert was last updated at
```

```
      endsAt:
```

```
        type: string
```

```
        format: date-time
```

```

    description: The time the alert ends at
  status:
    type: string
    enum: ["normal", "pending", "alerting"]
    description: Normal - The state of an alert that is neither firing nor pending.
Pending      - The state of an alert that has been active for less than the configured
threshold duration. Alerting - The state of an alert that has been active for longer
than the configured threshold duration.
  labels:
    $ref: '#/definitions/labels'
  annotations:
    $ref: '#/definitions/annotations'
  alertRuleUUID:
    type: string
    format: uuid

alertRules:
  type: array
  items:
    $ref: '#/definitions/alertRule'

postableAlertRule:
  type: object
  properties:
    name:
      type: string
    description:
      type: string
      description: description about the alert rule.
    expression:
      description: The expression evaluated for the alert rule. Alerting rules allow you
to define alert conditions based on Prometheus expression language expressions.
Eg:request_latency_seconds:mean5m{job="myjob"} > 0.5
      for:
        type: integer
        format: int64
        description: represents the elapsed time between two instants as an int64
nanosecond count - max time is 290 years. "for" clause causes oodle to wait for a
certain duration between first encountering a new expression output vector element
and counting an alert as firing for this element. default value of 1257893400000000000
(10 mins) is used.
      interval:
        type: integer

```



```

    format: duration
    description: an interval in seconds which determines how often the alert rule is
evaluated. default value of 600 (10 mins) is used.
    labels:
      $ref: '#/definitions/labels'
    annotations:
      $ref: '#/definitions/annotations'
  required:
    - labels
    - annotations
    - expression
    - description
    - name

  alertRule:
    description: An alert rule is a set of criteria that determine when an alert should
fire. It consists of an expression which needs to be evaluated to true, an interval
which determines how often the alert rule is evaluated, and a duration over which the
expression should be true for an alert to fire. Alert rules are evaluated over their
interval, and each alert rule can have zero, one, or any number of alerts firing at a
time. The state of the alert rule is determined by its most "severe" alert, which can be
one of Normal, Pending, or Firing.
    allOf:
      - type: object
    properties:
      id:
        type: string
        format: uuid
        description: uuid to identify the alert rule
      status:
        type: string
        enum: ['normal', 'pending', 'firing']
        description: Normal    None of the time series returned by the evaluation engine
is in a Pending or Firing state. Pending    At least one time series returned by the
evaluation engine is Pending. Firing    At least one time series returned by the
evaluation engine is Firing.
      - $ref: '#/definitions/postableAlertRule'
    required:
      - id
      - status

  labels:
    type: array

```

```

    items:
      $ref: '#/definitions/label'

  label:
    type: object
    description: The labels clause allows specifying a set of additional labels to be
    attached to the alert. Any existing conflicting labels will be overwritten
    properties:
      name:
        type: string
        description: Name of the label
      value:
        type: string
        description: Value for the label

  annotations:
    type: array
    items:
      $ref: '#/definitions/annotation'

  annotation:
    type: object
    description: The annotations clause specifies a set of informational labels that can
    be used to store longer additional information such as alert descriptions or runbook
    links
    properties:
      name:
        type: string
        description: Name of the annotation
      value:
        type: string
        description: Value for the annotation

```

Notes:

1. Use the /clear API to reset the data if required.

Open questions

Add any open questions below or add them as comments to the doc