Demo Project: Configure Monitoring for a Third-Party Application

Project Description

In this project, we will monitor **Redis** using **Prometheus Exporter**. The steps include:

1. Deploy **Redis service** in the Kubernetes cluster.
2. Deploy **Redis Exporter** using Helm Chart.
3. Configure **Alert Rules** (e.g., Redis is down or too many connections).
4. Import **Grafana Dashboard** for Redis visualization.

References:

- **Redis Exporter Github:** https://github.com/oliver006/redis_exporter
- **Prometheus Redis Exporter Helm Chart**: https://github.com/prometheus-community/helm-charts/blob/main/charts/prometheus-redis-exporter

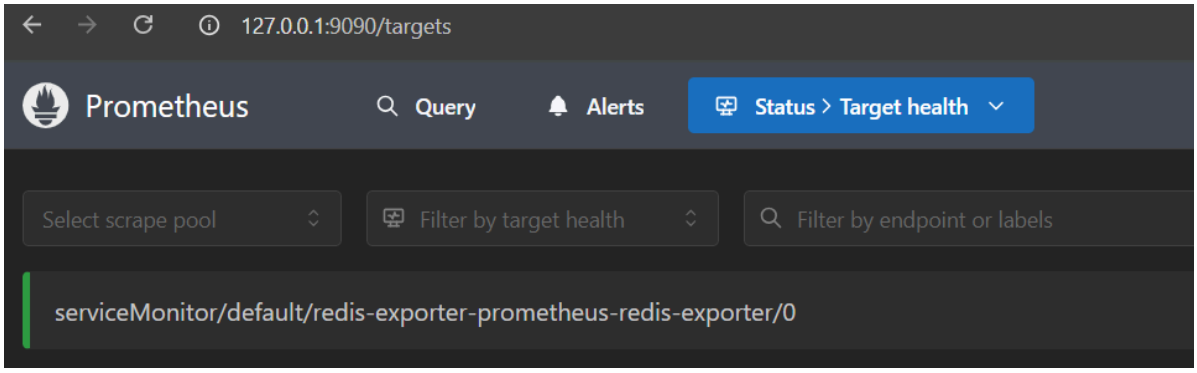
---

Step 1: Deploy Redis Exporter using Helm

1. **Create a values file** `redis-values.ymal`

```
serviceMonitor:
  enabled: true
  labels:
    release: monitoring
redisAddress: redis://redis-cart:6379
```

2. **Add Helm Repository and Install Redis Exporter:**
   - `helm repo add prometheus-community https://prometheus-community.github.io/helm-charts`
   - `helm install redis-exporter prometheus-community/prometheus-redis-exporter -f redis-values.yaml`
3. **Confirm Deployment:**
   - Check if Redis Exporter is installed: `helm ls`
   - Ensure Redis Exporter Pod is runningedis exporter pod is shown: `kubectl get pod`
   - Confirm ServiceMonitor is created: `kubectl get servicemonitor`
4. **Verify the Redis Exporter service** is listed in **Prometheus UI:**



Step 2: Create Alert Rules for Redis

Reference Alert Rules:

Use the Awesome Prometheus Alerts (https://samber.github.io/awesome-prometheus-alerts/rules.html) for pre-confured templates.

1. **Create** `redis-rules.yaml` :

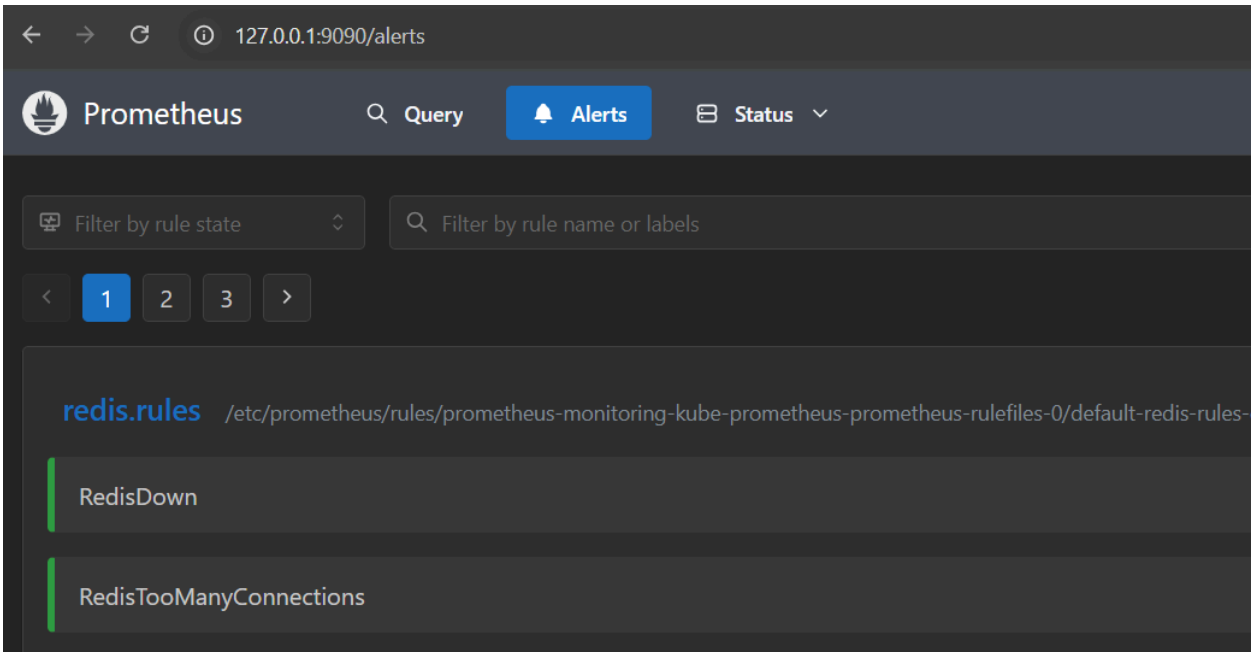
```
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  name: redis-rules
  labels:
    app: kube-prometheus-stack
    release: monitoring
spec:
  groups:
  - name: redis.rules
    rules:
    - alert: RedisDown
      expr: redis_up == 0
      for: 0m
      labels:
        severity: critical
      annotations:
        summary: Redis down (instance {{ $labels.instance }})
        description: "Redis instance is down\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
    - alert: RedisTooManyConnections
      expr: redis_connected_clients / redis_config_maxclients * 100 > 90
      for: 2m
      labels:
        severity: warning
      annotations:
        summary: Redis too many connections (instance {{ $labels.instance }})
        description: "Redis has {{ $value }} connections\n  LABELS = {{ $labels }}"
```

2. **Apply the Alert Rules:**

   - `kubectl apply -f redis-rules.yaml`

- Confirm the rule is applied: `kubectl get prometheusrule`

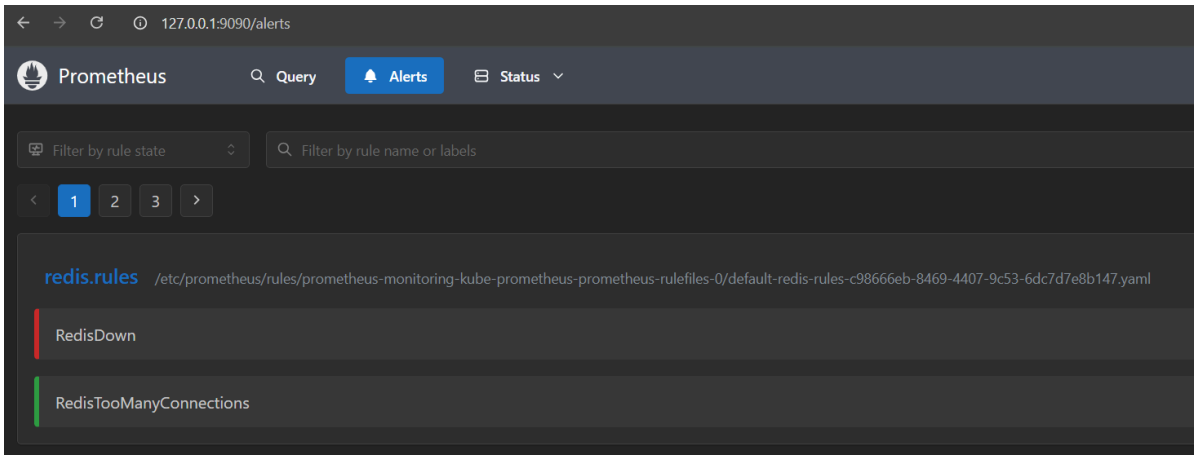
3. **Verify Alerts in Prometheus UI:**



Step 3: Trigger and Validate "Redis Down" Alert

1. **Simulate Redis Downtime:**
   - List all pods: `kubectl get pod`
   - Set replicas to “0”: `kubectl edit deployment redis-cart`
2. **Observe Alerts in Prometheus UI:**

   Check if the **Redis Down** alert is triggered.



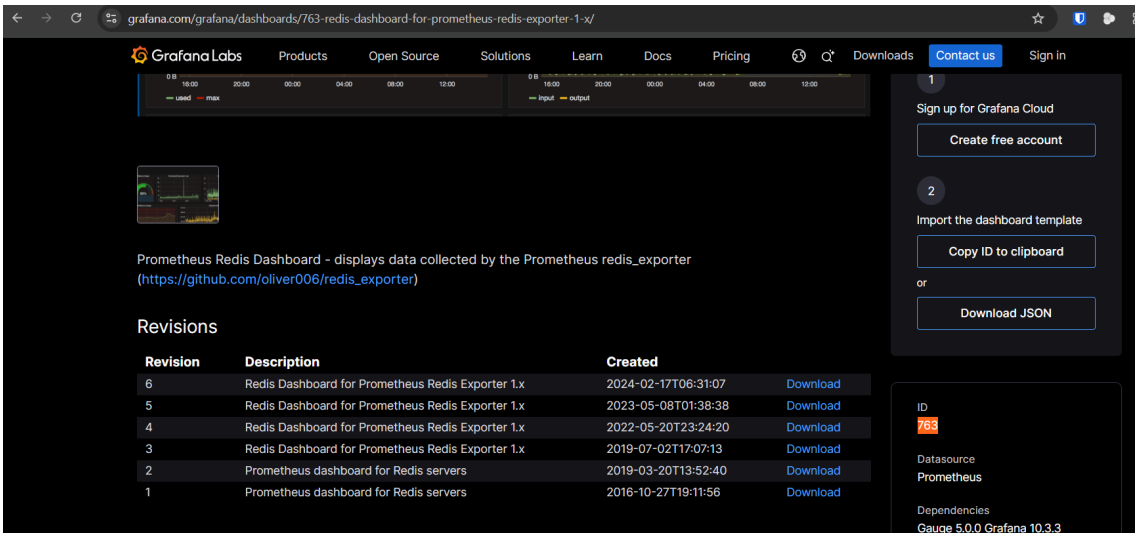
3. **Restore Redis Service:**
   - Set replicas back to "1": `kubectl edit deployment redis-cart`

Step 4: Import Redis Dashboard in Grafana

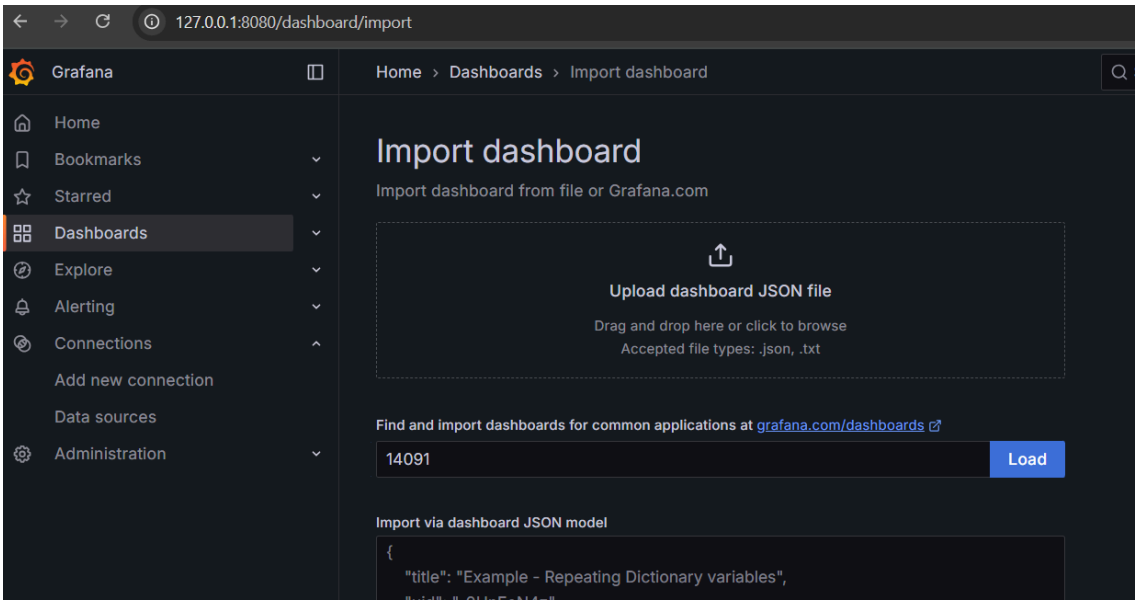
1. **Find Redis Dashboard:**

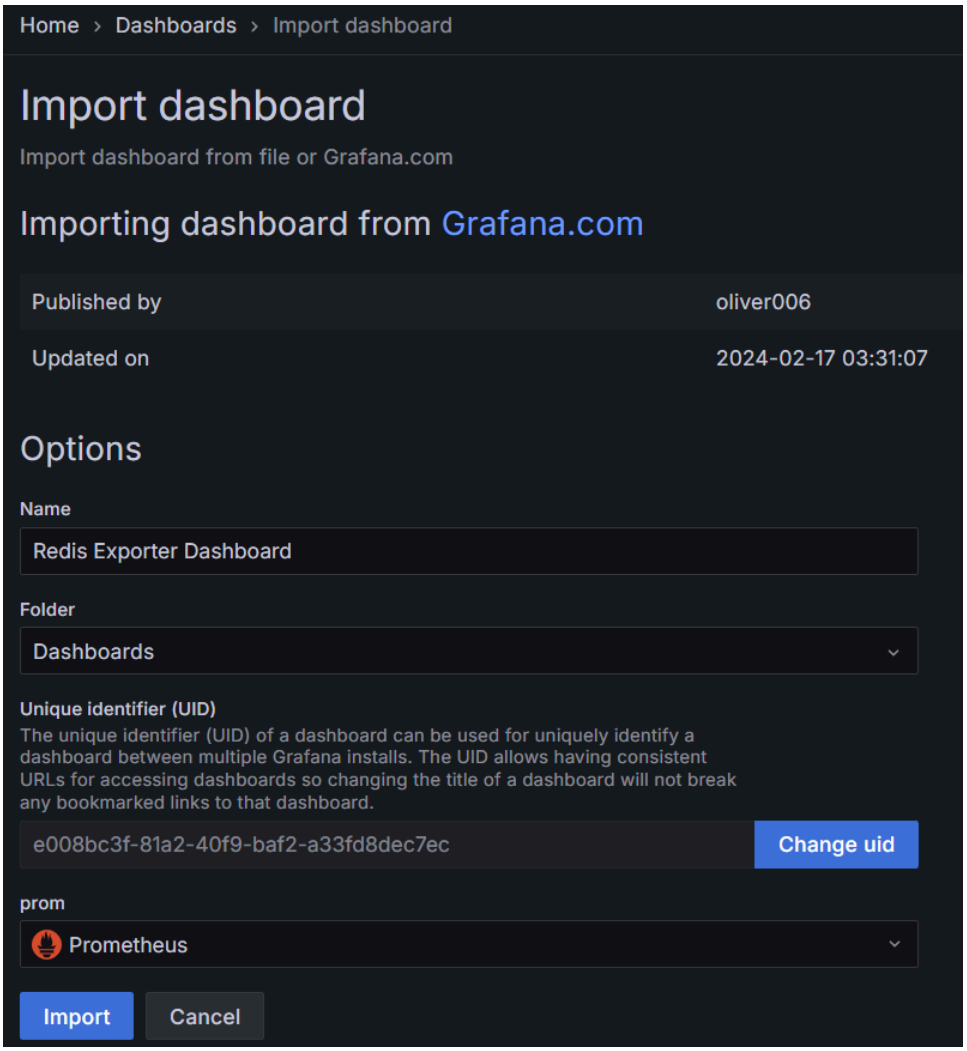
   Google **"Prometheus Redis Exporter Grafana Dashboard"**.
   Example: Grafana Redis Dashboard:
   https://grafana.com/grafana/dashboards/763-redis-dashboard-for-prometheus-redis-exporter-1-x/

2. **Import Grafana Dashboard**
   - Copy the **Dashboard ID** from Grafana Labs.

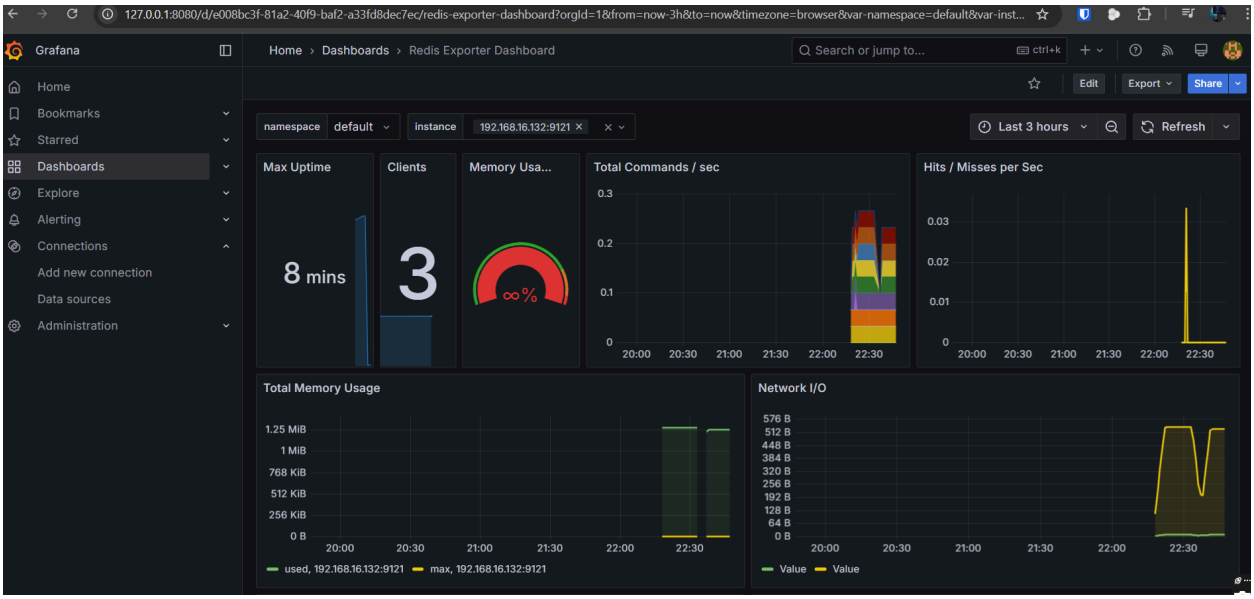


3. In **Grafana UI:**
   - Click **Import Dashboard**.
   - Paste the **Dashboard ID**.
   - Name it **"Redis Exporter Dashboard"**.
   - Click **Import**.





4. **Verify Redis metrics are displayed in Grafana:**


127.0.0.1:8080/d/e008bc3f-81a2-40f9-baf2-a33fd8dec7ec/redis-exporter-dashboard?orgId=1&from=now-3h&to=now&timezone=browser&var-namespace=default&var-inst...
Grafana
Home
Bookmarks
Starred
Dashboards
Explore
Alerting
Connections
Add new connection
Data sources
Administration
Home › Dashboards › Redis Exporter Dashboard
Search or jump to...
ctrl+k
Edit
Export
Share
namespace
default
instance
192.168.16.132:9121
Last 3 hours
Refresh
Max Uptime
8 mins
Clients
3
Memory Usa...
∞%
Total Commands / sec
0.3
0.2
0.1
0
20:00
20:30
21:00
21:30
22:00
22:30
Hits / Misses per Sec
0.03
0.02
0.01
0
Total Memory Usage
1.25 MiB
1 MiB
768 KiB
512 KiB
256 KiB
0 B
used, 192.168.16.132:9121
max, 192.168.16.132:9121
Network I/O
576 B
512 B
448 B
384 B
320 B
256 B
192 B
128 B
64 B
0 B
Value
Value