PERCONA

TRAINING

Monitoring

http://www.percona.com/training/

Monitoring

- Existing Tools and Services
- DIY high level
- Alerting in Practice

Existing Tools and Services

Open Source

- Percona Monitoring and Management (PMM)
  - https://pmmdemo.percona.com/
- Nagios
- Prometheus
- Grafana
- AlertManager

Commercial

- Datadog
- NewRelic
- MongoDB Cloud Manager
- VividCortex

Alerting in Practice

- Based on our experience from MongoDB Managed Services
- Based on Nagios
  - Checks we use
  - SEV 1/2 Checks
  - SEV 3/4 Checks

Alerting in Practice

pmp-check-mongo.py Nagios Plugin (discountinued)

- Check connect time against certain timeout
- Current vs available connections
- Number of queued operations
- Check for replication lag
- Total indexes count (more indexes affect writes)
- Timing custom query test
- Check `oplog` retention period
- Check if a `PRIMARY` exists
- Imbalanced number of chunks on a sharded collection

Alerting in Practice

- SEV 1/2 Checks
  - Failed Mongo Connect
  - No Primary
  - Replication Lag > 200s
  - Election Occurred
  - OpLog Time < backup interval
  - Queues in Mongo > 30 consistently
- Sev 3/4 Checks
  - Query Time, Connections, Indexes
  - Balance,Table Scans

Alerting in Practice

- Prometheus AlertManager
  - PMM can integrate with external AlertManager instance
  - https://www.percona.com/blog/2020/02/21/percona-monitoring-and-management-meet-prometheus-alertmanger/

```
  - alert: MongodbInstanceNotAvailable
expr: mongodb_up < 1
for: {{ pmm_alertmanager_rule_periods.med_interval }}{{ pmm_alertmanager_rule_periods.interval_unit }}
labels:
severity: critical
service: mongodb
runbook: {{ pmm_alerting_runbook_path }}#MongoDBInstanceNotAvailable
annotations:
{% raw %}
summary: "CRITICAL: The MongoDB service is unreachable (instance {{ $labels.node_name }})"
description: "The mongod service appears to be down, or unreachable. Please investigate."
{% endraw %}
```

Alerting in Practice

- PMM Integrated alerting
  - No external components required
  - Template based
  - Currently email, Pagerduty and Slack are supported
  - Available as technical preview in PMM 2.13

Exercises - PMM

1. Install PMM client

```
yum install pmm2-client
```

2. Create the PMM role

```
db.createRole({
   "role":"explainRole",
   "privileges":[
      {
         "resource":{
            "db":"",
            "collection":""
         },
         "actions":[
            "listIndexes",
            "listCollections",
            "dbStats",
            "dbHash",
            "collStats",
            "find"
         ]
      }
   ],
   "roles":[]
})
```

Exercises - PMM

3. Create the PMM user

```
db.getSiblingDB("admin").createUser({
   "user":"mongodb_exporter",
   "pwd":"percona",
   "roles":[
      {
         "role":"explainRole",
         "db":"admin"
      },
      {
         "role":"clusterMonitor",
         "db":"admin"
      },
      {
         "role":"read",
         "db":"local"
      }
   ]
})
```

Exercises - PMM

4. Point the pmm agent to the server

```
pmm-admin config --server-url=https://server_user:server_password@server_ip:443 --
```

5. Configure the mongo exporter

```
pmm-admin add mongodb --username=mongodb_exporter --password=percona --host=127.0.
```

6. Connect to the PMM server and verify the metrics are being uploaded

PMM Demo

- Demo site always available for you to experiment
- https://pmmdemo.percona.com/

PERCONA
TRAINING

Questions