Apache Ignite Management and Monitoring Solution With GridGain Control Center

Denis Magda
October/2020
Your Trainer: Denis Magda

➔ Distributed in-memory system
  ◆ Apache Ignite Committer and PMC Member
  ◆ Head of DevRel at GridGain

➔ Java engineering and architecture
  ◆ Java engineering at Oracle
  ◆ Technology evangelism at Sun Microsystems
Training Flow

- Introduction, what you’re going to build (5 mins)
- **Task #1**: starting the demo setup (20 mins)
- **Task #2**: configuring storage usage metrics (20 mins)
- Break (5 mins)
- **Task #3**: configuring alerts (20 mins)
- **Task #4**: tracing operations performance (20 mins)
- **Task #5**: restoring the cluster from a snapshot (20 mins)
Introduction
What You’re Going to Build
Demo Setup

1. Clone Demo App
2. Launch Market Orders Stream
3. Stream Demo App

Your Laptop

GitHub
Training Project

PubNub
Market Orders Stream

Java
Demo App
Task #1
Starting the Demo Setup
Hands-on
Prerequisites

Step 0: Get what you need.
• Chrome or Safari
• Docker 19 or later
• Docker Compose 1.25.5 or later
• Java Developer Kit 8 or later
• Apache Maven 3.3 or later
Hands-on
Download the Project and Start an Ignite Cluster

Step 1: Download and unzip the training project:
https://github.com/GridGain-Demos/ignite-streaming-monitoring-demo.git

Step 2: Start an Ignite cluster:

docker-compose -f docker/ignite-cluster.yml up -d --scale ignite-server-node=2
Hands-on Deploy GridGain Control Center

Step 3: Deploy Control Center in Docker:

docker-compose -f docker/control-center.yaml up -d

Step 4: Instruct the Ignite cluster to work with your Control Center deployment:

docker exec -it docker_ignite-server-node_1 bash
cd bin/
Hands-on
Finish the Setup of GridGain Control Center

Step 5: Open Control Center and Create Your Account
http://localhost:8443

Step 6: Search for a cluster token to tether the cluster with Control Center:

docker container logs docker_ignite-server-node_1
Hands-on
Start the Market Orders Application

Step 7: Build the application with Maven:

```
mvn clean package
```

Step 8: Change the application `execTime` to 90 minutes
see `docker\ignite-streaming-app.yaml`

Step 9: Run the application in Docker

```
docker build -f docker/StreamingAppDockerfile -t ignite-streaming-app .
docker-compose -f docker/ignite-streaming-app.yaml up -d
```
Check the Setup is Working

```
1  SELECT * FROM Trade ORDER BY order_date DESC LIMIT 10;
```
Task #2
Configuring Storage Usage Metrics
Customizable, Intuitive Monitoring Dashboards

- Easy to Use with Visual Cues
  - Visualize cluster status and manage behavior

- Highly Flexible
  - Drag and drop panes to view metrics of interest
  - Tracks over 200 cluster metrics
    - Open Census compliant
What You’re Doing Next
Storage Usage Dashboard
Hands-on
Create Dashboard With Memory Usage Metrics

Step 1: Create a “Storage Usage” dashboard

Step 2: Add a widget reporting the off-heap memory usage
• Physical Memory Size metric

Step 3: Add a widget collecting Java Heap usage stats
• Heap Used metric
Hands-on
Add Disk Usage Metrics

Step 4: Monitor the storage-usage size
• Storage Size metric

Step 5: Track the WAL size
• WAL Total Size metric

Step 6: Monitor the checkpointing duration
• Last Checkpoint Duration metric

Step 7: Watch the WAL sync time
• Datastorage WAL Fsync Time Duration metric
5 Minutes Break
Task #3
Configuring Alerts
User-Defined Production Alerts

• Quickly Identify and Resolve Issues with Configurable Alerts
  – Create custom active alerts on any metric
  – Monitor cluster, node, and cache events
  – Configure flexible notifications
    • Email and SMS
What You’re Doing Next
Configure Several Alerts
Hands-on
Create Memory-Usage and Node-Count Alerts

Step 1: Create a custom notification channel

Step 2: Add a memory-usage alert
• Triggered when the Physical Memory Size metric exceeds 500MB

Step 3: Set up a node-count alert
• Triggered when the number of nodes is <= 2
Task #4
Tracing Operations Performance
Active Tracing & Root Cause Analysis

- Accelerate Development Time and Reduce Production Downtime
  - Isolate and identify the root cause of any performance issues

- Easily Track API Call Execution
  - Identify all response times for API calls
  - Single view of stack traces and logs from all nodes
  - Follow execution across the cluster
  - Combine with application traces
  - Open Census compliant
### What You’re Doing Next

#### Analyze Tracing Samples of Running Transactions

<table>
<thead>
<tr>
<th>Name</th>
<th>Start Time</th>
<th>Duration</th>
<th>Total Spans</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>transaction</td>
<td>Aug 27, 13:01:31</td>
<td>20 ms</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>transaction</td>
<td>Aug 27, 13:01:31</td>
<td>19 ms</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>transaction</td>
<td>Aug 27, 13:01:30</td>
<td>26 ms</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>transaction</td>
<td>Aug 27, 13:01:30</td>
<td>12 ms</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>transaction</td>
<td>Aug 27, 13:01:30</td>
<td>10 ms</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>transaction</td>
<td>Aug 27, 13:01:29</td>
<td>10 ms</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>transaction</td>
<td>Aug 27, 13:01:29</td>
<td>20 ms</td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>

![Timeline Diagram]

- **Date:** August 27, 2020
- **Duration:** 20 ms
- **Depth:** 6
- **Total Spans:** 16

**Transactions:**
- `transactions.commit`
- `transactions.near.prepare`
- `transactions.dht.prepare`
- `transactions.dht.finalize`
- `transactions.dht.process.prep`
Hands-on
Enabling Tracing for Transactions

Step 1: Connect to the container of the first cluster node:

```bash
docker exec -it docker_ignite-server-node_1 bash
cd bin/
```

Step 2: Enable the tracing of Ignite transactions:

```bash
JVM_OPTS="-DIGNITE_ENABLE_EXPERIMENTAL_COMMAND=true" ./control.sh --tracing-configuration set --scope TX --sampling-rate 0.3
```
**Hands-on**

**Analyze Transactions Traces**

**Step 3:** With Control Center’s Tracing screen, observe distinct steps of distributed transactions and execution time of each step

**Step 4:** Disable the tracing for transactions:

```
JVM_OPTS="-DIGNITE_ENABLE_EXPERIMENTAL_COMMAND=true" ./control.sh --tracing-configuration set --scope TX --sampling-rate 0
```
Task #5
Restoring the Cluster From a Snapshot
Centralized Backup and Recovery Management

- Full and incremental snapshots
- Continuous archiving (WAL)
- Network backups
- Point-in-time Recovery
- Heterogeneous Recovery

In-Memory Tier – (0-100%) Data and Indexes

Disk Tier (100%) Data and Indexes

GridGain Server Cluster

External Store (Backup)
Disaster Recovery & Backup Management

- Easy Configuration of Backups
  - Fully managed backups and comprehensive data recovery tools
  - GridGain Ultimate Edition only

- Comprehensive Backup Monitoring and Management
  - Create full and incremental backups
  - Validate backup integrity
  - Recover state from a specific point in time
  - Automate backup creation lifecycle
What You’re Doing Next
Corrupt the Cluster and Restore It From a Snapshot
Hands-on
Create a Full Cluster Snapshot

Step 1: Pause the application (only for the demo purpose)

```
docker-compose -f docker/ignite-streaming-app.yaml stop
```

Step 2: With Control Center’s Snapshots screen, create a full cluster snapshot
**Step 3:** With Control Center’s SQL screen, remember the number of settled trades before the corruption:

```
SELECT count(*) FROM Trade;
```

**Step 4:** Corrupt the Trades table:

```
DELETE FROM Trade;
```
Hands-on
Restore the Cluster With the Snapshot

Step 5: With Control Center’s Snapshots screen, restore the corrupted table:

Step 6: Check that the lost data is restored:

```
SELECT count(*) FROM Trade;
```
Summary
Stop the Setup and Release Resources

```
docker-compose -f docker/ignite-cluster.yml down

docker-compose -f docker/control-center.yml down

docker-compose -f docker/ignite-streaming-app.yml down
```
Learn More

• Bookmark the written version of the training
  – https://www.gridgain.com/docs/tutorials/management-monitoring/overview

• Refer to GridGain Control Center documentation
  – https://www.gridgain.com/docs/control-center/latest/overview

• Check the “Ignite in Production” playlist
  – Find the GridGain channel on YouTube
Stay connected with Apache Ignite users & experts

meetup.com/Apache-Ignite-Virtual-Meetup/