



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



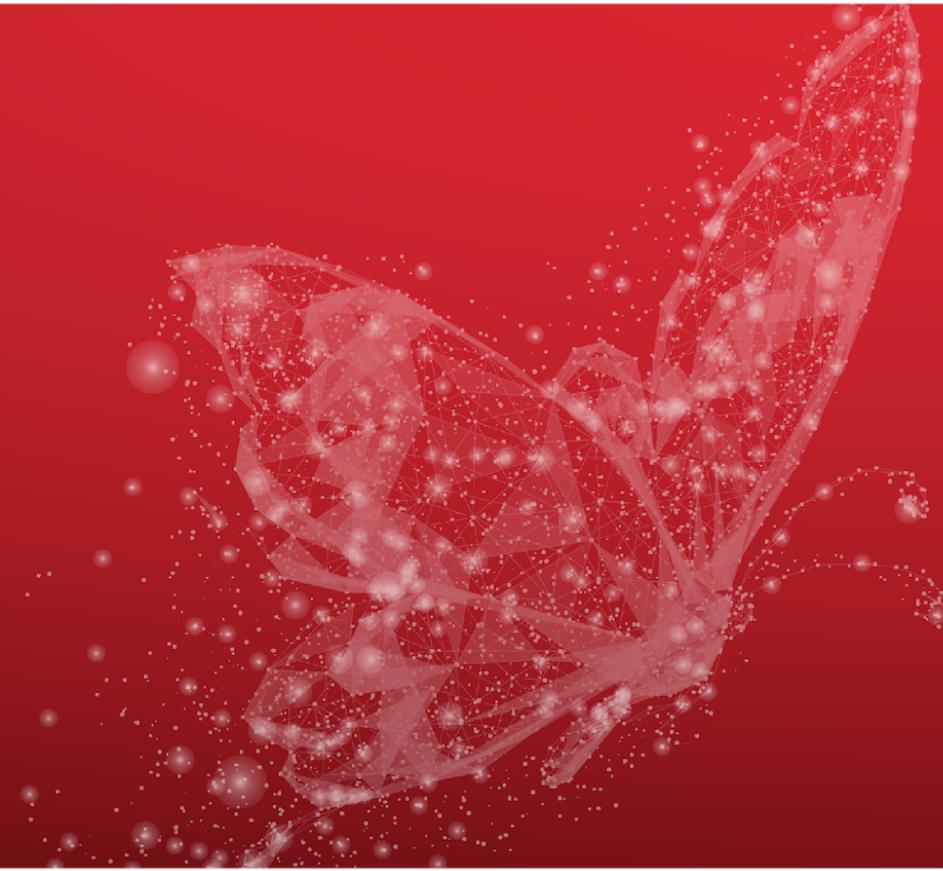
Training Flow



- 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

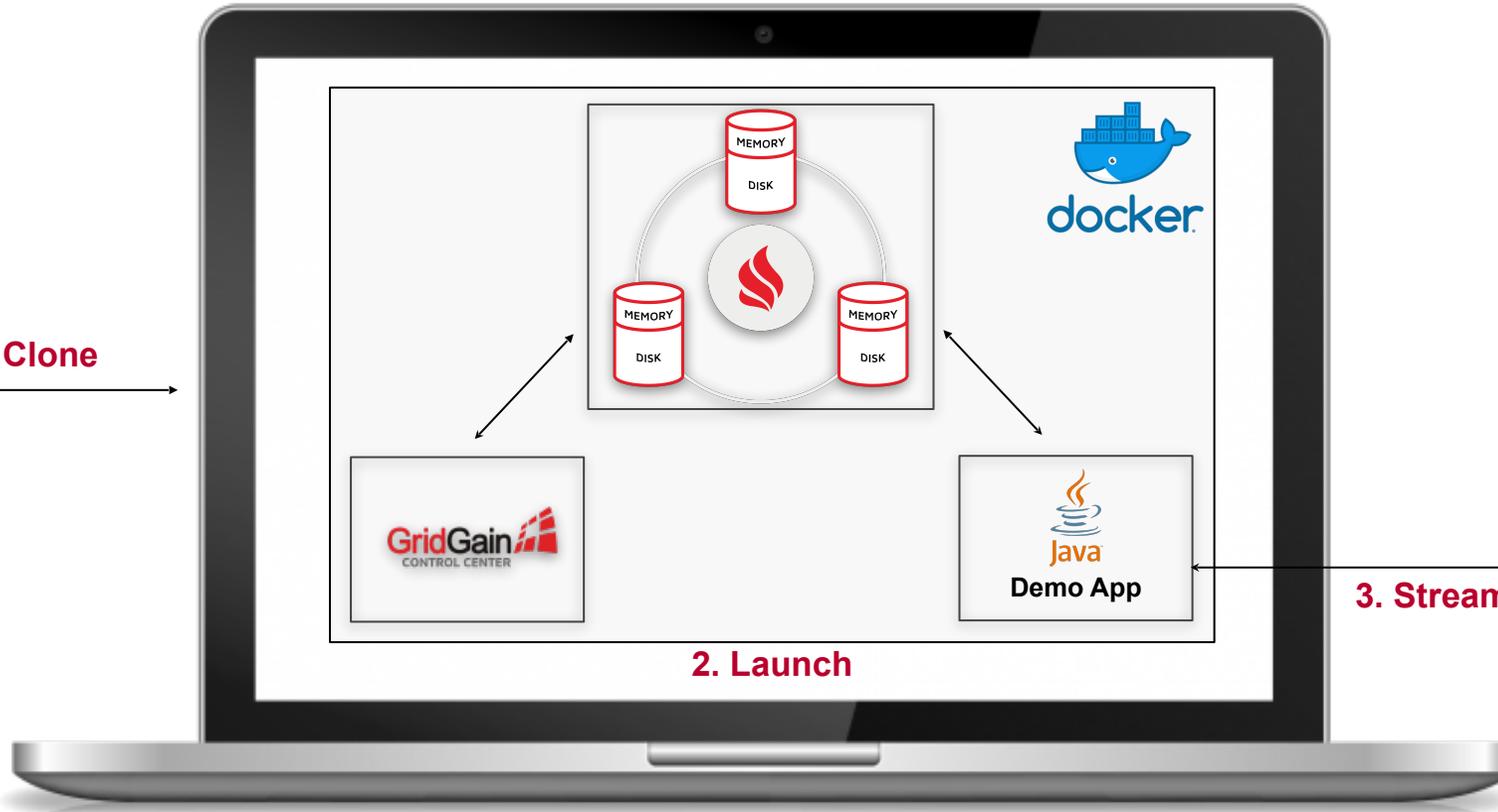
What You're Going to Build



Demo Setup



1. Clone



2. Launch

PubNub

Market Orders Stream

3. Stream

Your Laptop

GridGain Control Center

Take Full Control of Your Apache Ignite Clusters



The screenshot displays the GridGain Control Center interface for a cluster named 'trusting_feistel'. The dashboard includes several monitoring widgets:

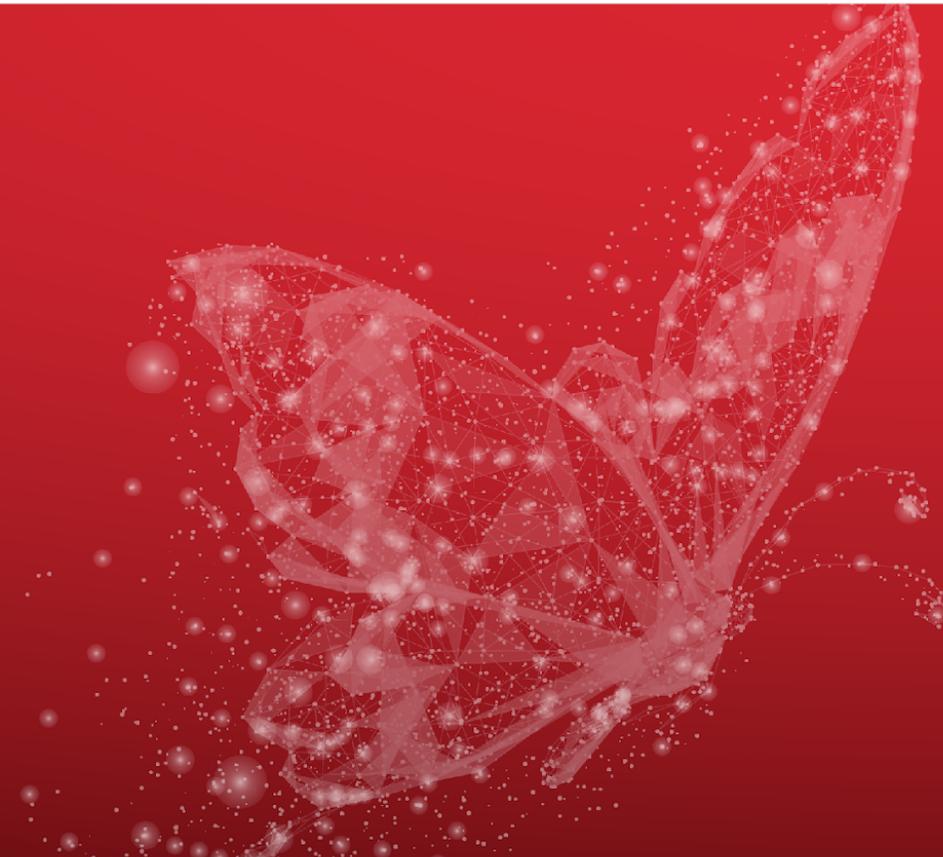
- Nodes:** A table listing three nodes with their consistency IDs, online status, types, versions, and IP addresses.
- Thread Time:** A line chart showing the current thread CPU time for five nodes over a 24-hour period.
- Heap Usage:** A color-coded bar chart showing the percentage of heap memory used across three nodes.
- Memory:** A table showing the heap used memory for three nodes.
- Heap Size:** A line chart showing the heap size for five nodes over a 24-hour period.
- Rebalance:** A modal window showing a 100% progress indicator for a rebalance operation.

Order	Consistency ID	Online	Type	Version	IP
1	b9a42525...	✓	Coord...	8.7.14	192.168.8...
2	36ae0c82...	✓	Server	8.7.14	192.168.8...
3	b992d4a4...	✓	Server	8.7.14	192.168.8...

Consistency ID	Heap Used
E88F3FDA	62.96MB
B9A42525	69.66MB
B992D4A4	49.69MB

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.yaml 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/
./management.sh --uri http://localhost:8443,http://control-center-frontend:8443
```

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



The screenshot shows the GridGain Control Center interface. At the top, it says "GridGain CONTROL CENTER" and "Trial: Not for Production Usage". The user "exciting_jepsen" is logged in. The main area is divided into "QUERIES LIST", "RUNNING QUERIES", and "QUERY STATISTICS". The "QUERIES LIST" tab is active, showing a tree view for "exciting_jepsen" with sub-items "Schemas", "Caches", and "Nodes". The "Query" editor shows the SQL query: `1 SELECT * FROM Trade ORDER BY order_date DESC LIMIT 10;`. Below the query, the results are displayed in a table:

ID	BUYER_ID	SYMBOL	ORDER_QUANTITY	BID_PRICE	TRADE_TYPE	ORDER_DATE
1720	4	Linen Cloth	506	251.260716188347...	day	1598627788000
1723	2	Google	731	166.446180805165...	limit	1598627788000
1722	1	Bespin Gas	228	110.818547731279...	day	1598627788000
1721	1	Apple	600	194.143961895261...	market	1598627788000
1719	5	Elerium	445	173.871947283836...	fill or kill	1598627787000

Task #2

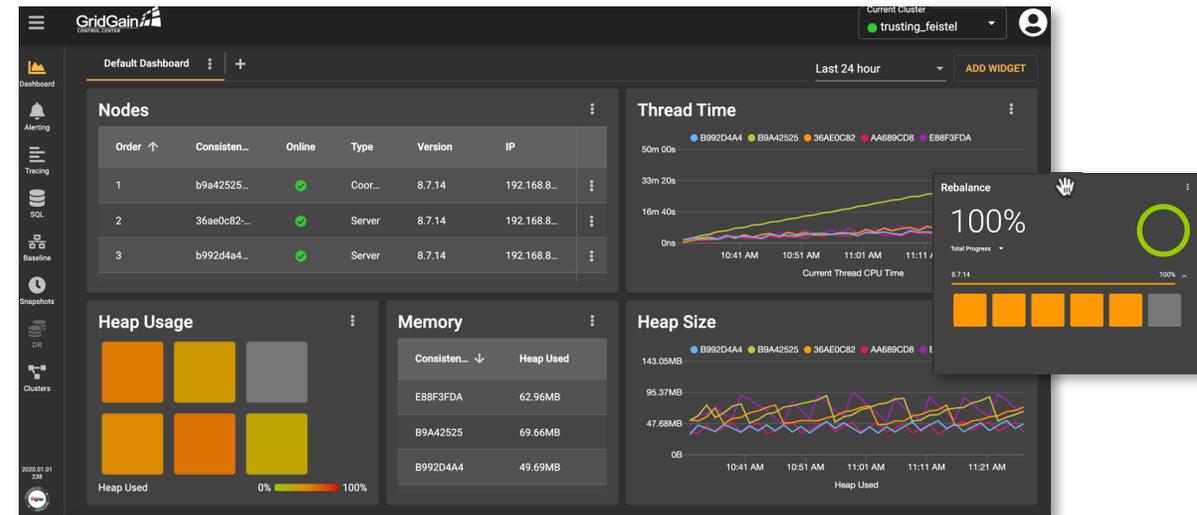
Configuring Storage Usage Metrics



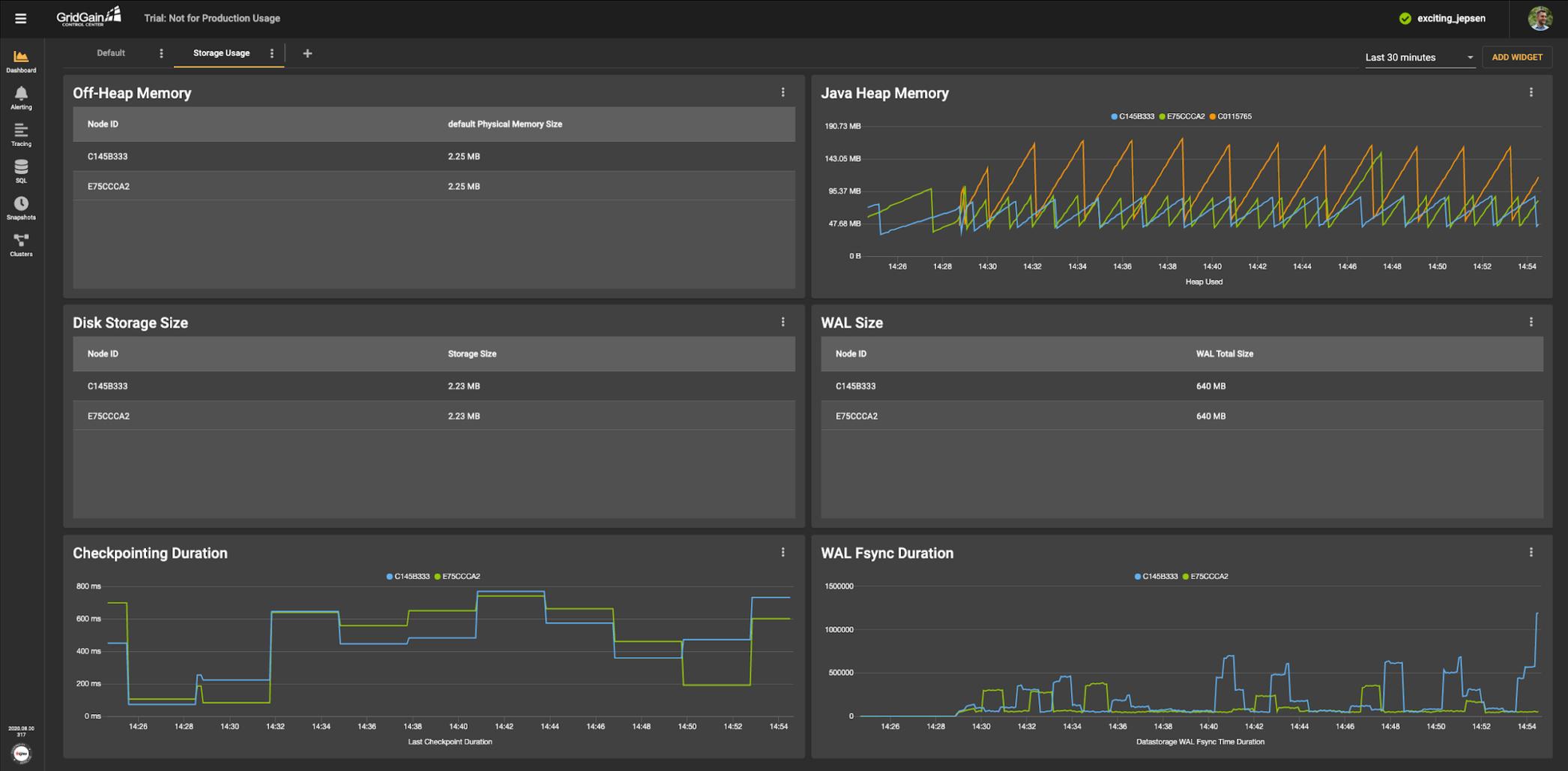
Customizable, Intuitive Monitoring Dashboards



- 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

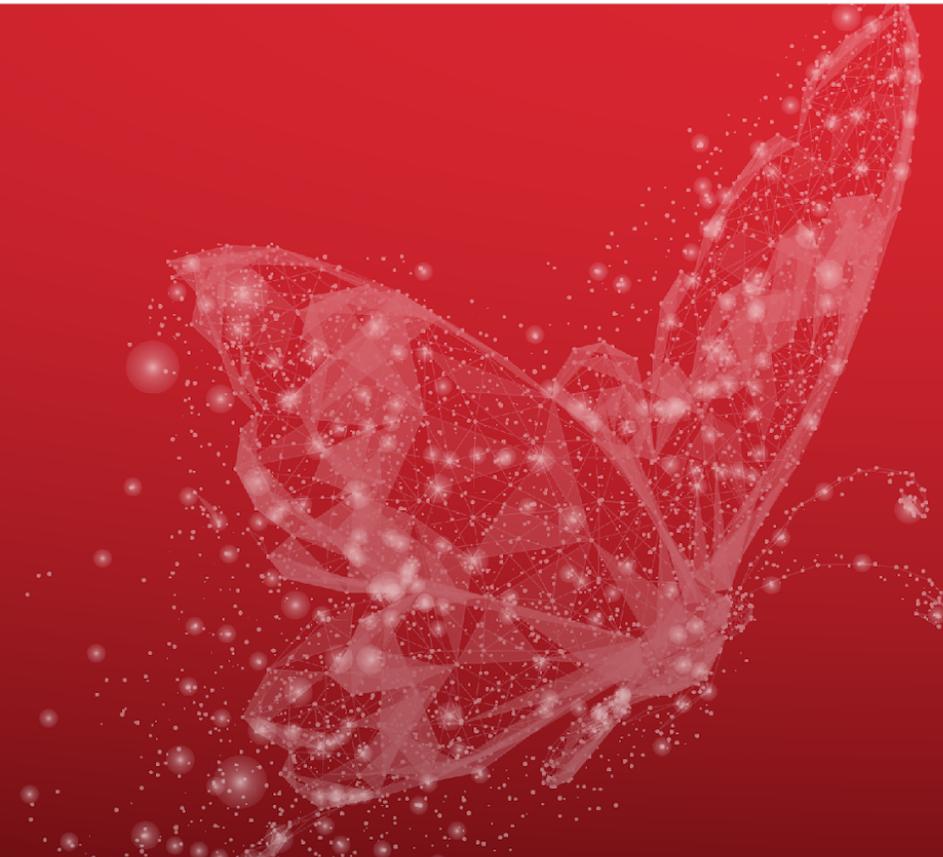
Step 6: Monitor the checkpointing duration

- **Last Checkpoint Duration** metric

Step 7: Watch the WAL sync time

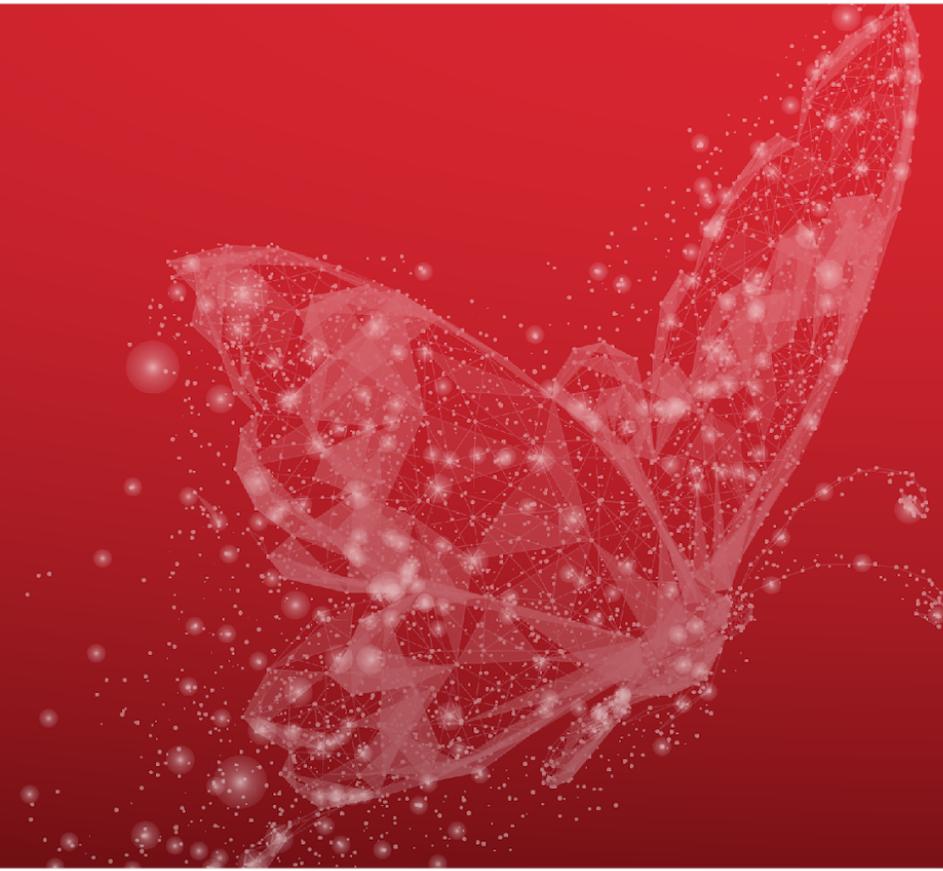
- **Datastorage WAL Fsync Time Duration** metric

5 Minutes Break



Task #3

Configuring Alerts



User-Defined Production Alerts



CONFIGURATIONS		EVENTS		NOTIFICATION CHANNELS	
Tag	Type	Condition	Count	Status	
Node Count	Cluster metric	Server Nodes Count < 10 for 0s	0	OK	
Memory	Node metric	Heap Used > 95.37MB for 0s	0	OK	
CPU	Node metric	CPU Load > 2% for 1s	0	OK	

CONFIGURATIONS		EVENTS		NOTIFICATION CHANNELS	
Created On ↓	State	Message		Alert	
5/15/20, 10:41 AM	Alerting	Node "Heap Used" metric was > 10000000		Memory	Memory
5/15/20, 10:41 AM	Alerting	Node "Heap Used" metric was > 10000000		Memory	Memory
5/15/20, 10:41 AM	Alerting	Node "Heap Used" metric was > 10000000		Memory	Memory
5/15/20, 10:41 AM	Alerting	Node "Heap Used" metric was > 10000000		Memory	Memory
5/15/20, 10:41 AM	Alerting	Node "Heap Used" metric was > 10000000		Memory	Memory
5/15/20, 10:41 AM	Alerting	Cluster "Server Nodes Count" metric was < 10		Node Count	Node Count

- 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



The screenshot shows the GridGain Alerting configuration page. The interface includes a top navigation bar with the GridGain logo, a trial notice, and a user profile. A left sidebar contains navigation icons for Dashboard, Alerting, Tracing, SQL, Snapshots, and Clusters. The main content area has tabs for CONFIGURATIONS, EVENTS, and NOTIFICATION CHANNELS. An 'ADD ALERT' button is located in the top right of the main area. Below the tabs is a table listing configured alerts.

Name	Type	Condition	Count	Status	Notifications	Enabled
Off-Heap Memory Usage Threshold	Node metric	default Physical Memory Size > 476.84 MB for 10 s	0	OK	My Email Server	<input checked="" type="checkbox"/>
Cluster Nodes Threshold	Cluster metric	Server Nodes Count < 2 for 0 s	0	OK	My Email Server	<input checked="" type="checkbox"/>

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

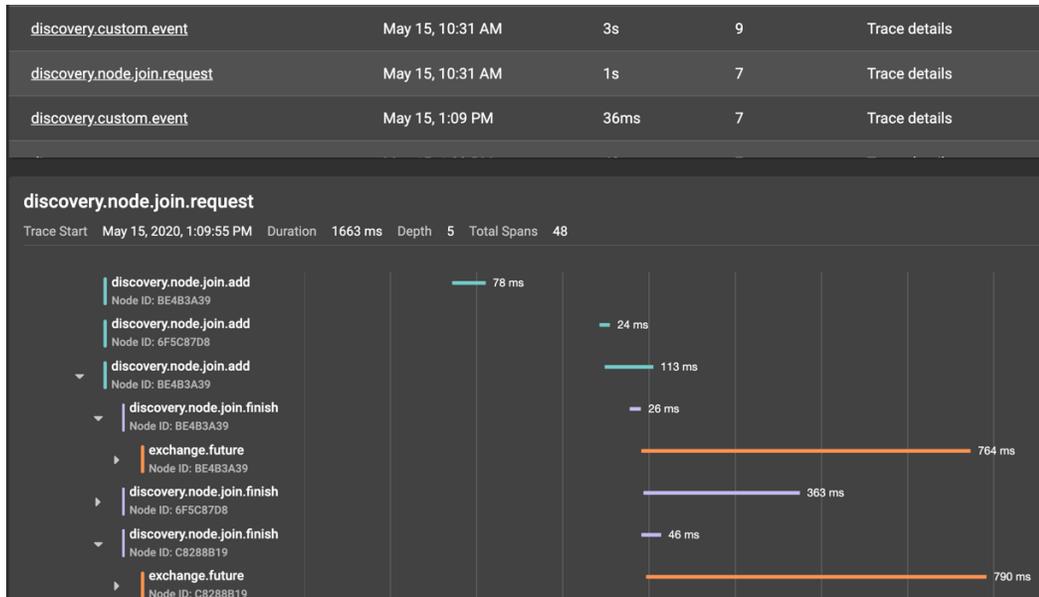
- Triggered when the **number of nodes is ≤ 2**

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



Dashboard

Alerting

Tracing

SQL

Snapshots

Clusters

Name	Start Time ↓	Duration	Total Spans	Details
transaction	Aug 27, 13:01:31.3...	20 ms	11	
transaction	Aug 27, 13:01:31.0...	19 ms	11	
transaction	Aug 27, 13:01:30.8...	26 ms	15	
transaction	Aug 27, 13:01:30.5...	12 ms	16	
transaction	Aug 27, 13:01:30.3...	10 ms	16	
transaction	Aug 27, 13:01:30.0...	10 ms	16	
transaction	Aug 27, 13:01:29.8...	20 ms	16	
transaction	Aug 27, 13:01:29.5...	13 ms	16	

transaction

Trace Start August 27, 2020 at 13:01:31.331 GMT-7 Duration 20 ms Depth 6 Total Spans 16

1 transactions.commit Node ID: B882C0A0 15 ms

2 tx.near.process.prepare.requ Node ID: D6CEAFA2 1 ms

3 tx.dht.process.prepare Node ID: 7410F9DA 6 ms

transactions.dht.finish Node ID: D6CEAFA2 0 ms

tx.near.process.prepare

Filters

Span Name

Roots only

Event Node ID

Node Consistent ID

Start Time

From

To

Hands-on

Enabling Tracing for Transactions



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

```
docker exec -it docker_ignite-server-node_1 bash

cd bin/
```

Step 2: Enable the tracing of Ignite transactions:

```
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:

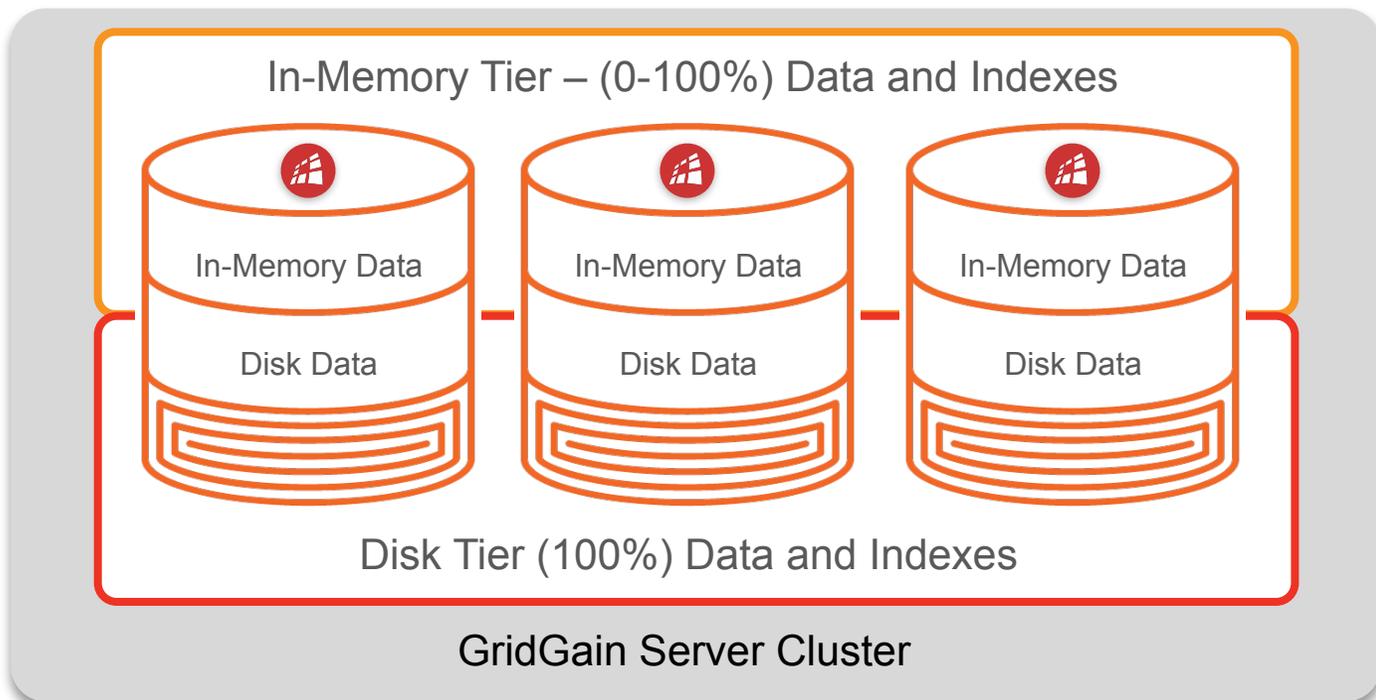
```
JVM_OPTS="-DIGNITE_ENABLE_EXPERIMENTAL_COMMAND=true" ./control.sh --tracing-configuration set --scope TX --sampling-rate 0
```

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

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

Start Time	Snapshot ID	Command	Status
Jan 14, 14:05 PM	1581480021019	MOVE	In Progress: 75%
Jan 14, 14:06 PM	1581480021018	DELETE	OK
Jan 14, 14:07 PM	1581480021017	MOVE	OK
Jan 14, 14:08 PM	1581480021016	CHECK	OK
Jan 14, 14:09 PM	1581480021015	MOVE	OK
Jan 14, 14:10 PM	1581480021014	COPY	OK
Jan 14, 14:11 PM	1581480021013	CHECK	OK
Jan 14, 14:12 PM	1581480021012	MOVE	Corrupted
Jan 14, 14:13 PM	1581480021011	COPY	Failed

What You're Doing Next

Corrupt the Cluster and Restore It From a Snapshot



The screenshot shows the 'SNAPSHOTS' management page. The table contains the following data:

Start Time	Type	ID	Mode	Status	Caches
Aug 28, 9:11	FULL	1598631062178	MANUAL	OK	3

The context menu options are: Restore From Snapshot, Show Related Snapshots, Check, Move, Copy, and Remove. The 'Restore From Snapshot' option is highlighted with a red box and an arrow.

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

Hands-on

Corrupt the Cluster



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

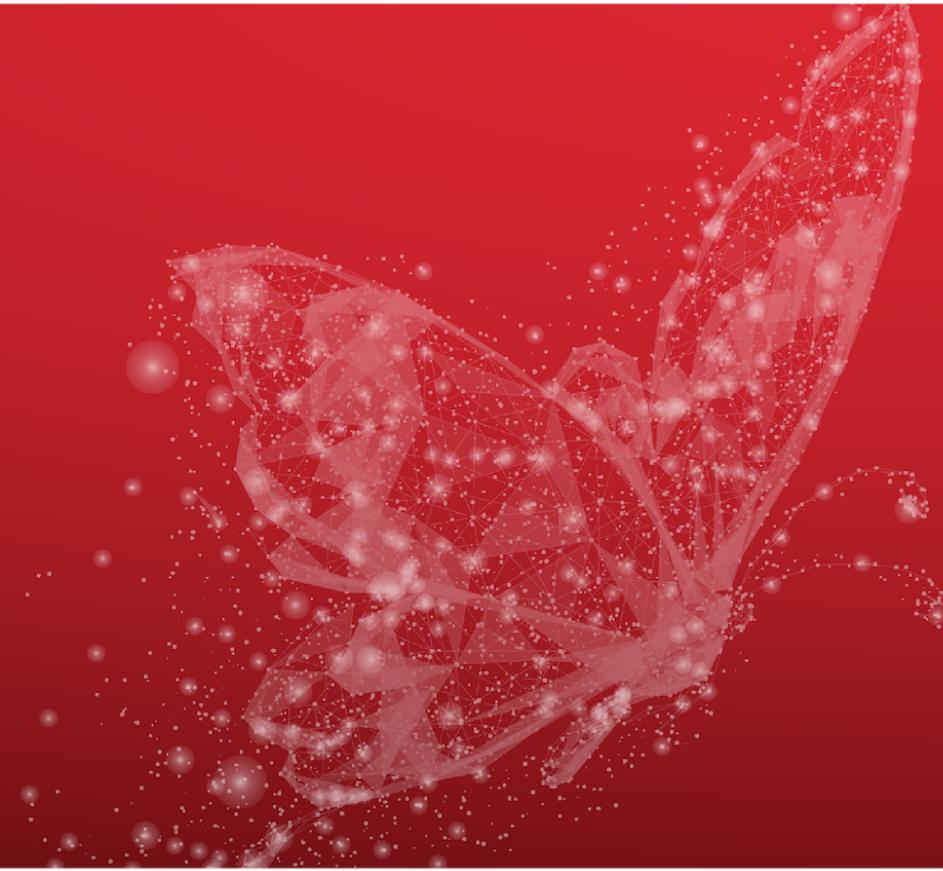


Step 5: With Control Center's Snapshots screen, restore the corrupted table:

Step 6: Check that the lost data is restored:

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

Summary



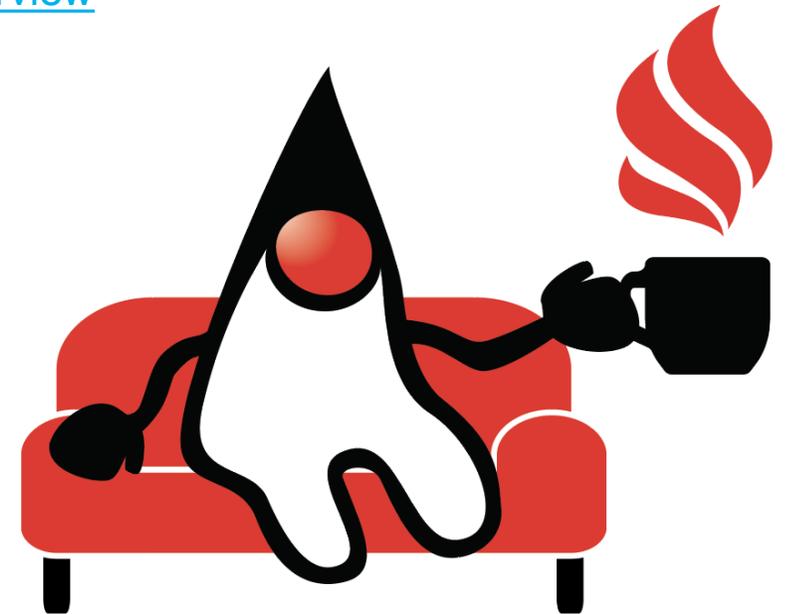
Stop the Setup and Release Resources



```
docker-compose -f docker/ignite-cluster.yaml down  
  
docker-compose -f docker/control-center.yaml down  
  
docker-compose -f docker/ignite-streaming-app.yaml 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/](https://www.meetup.com/Apache-Ignite-Virtual-Meetup/)

