GridGain

Developing Apache Ignite Applications That Are Easy to Manage

Alexey Kukushkin

Professional Services Consultant

December 9, 2020

Motivation

- Operations typically exceed fifty percent of an IT system life cycle cost.
- Developing manageable applications, the developers can significantly reduce the total cost of ownership.
- Manageability is especially important for distributed applications considering their complexity and mission-critical use cases.

Agenda

- How to develop manageable applications with Apache Ignite
- Ignite’s monitoring and management features
- Demonstrate common operation use cases using popular monitoring systems

Developing for Monitoring & Management

Developing for Monitoring & Management

Developer’s goal:

- Smoothly integrate the application into enterprise monitoring infrastructure
- Minimize time and effort to recover from a failure

Developing for Monitoring & Management

It is the developer’s responsibility to:

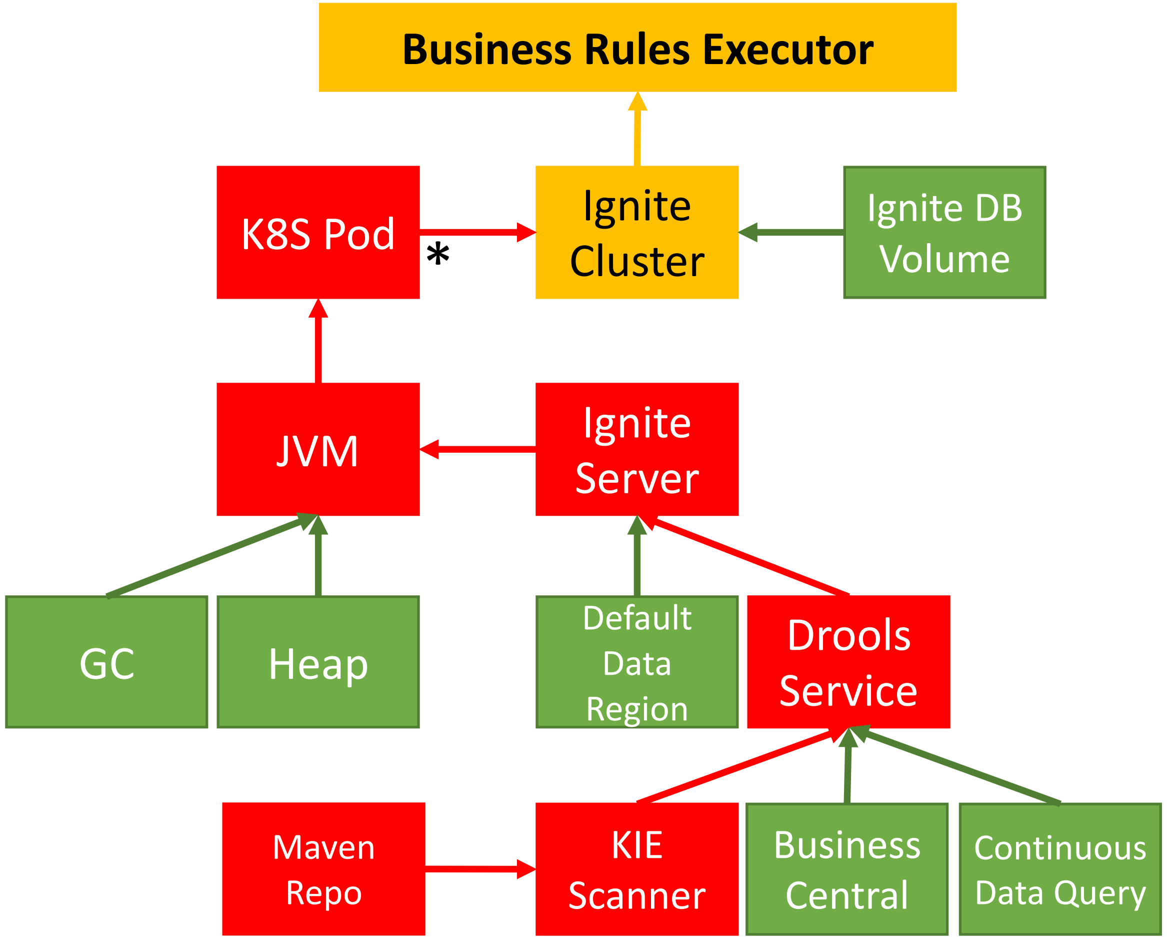
1. Develop Health Model
2. Instrument the code
3. Create and maintain Monitoring Spec for DevOps
4. Develop and add monitoring tests to CI/CD pipeline

Health Model

- Health Model describes:
  - Components that can be configured and restarted (reset)
  - Synthetic transactions
  - Health events
- DevOps are the Health Model’s target audience.
  - Developers create Monitoring Spec to communicate Health Model to DevOps
  - Developers test and update Monitoring Spec for every application and Ignite release

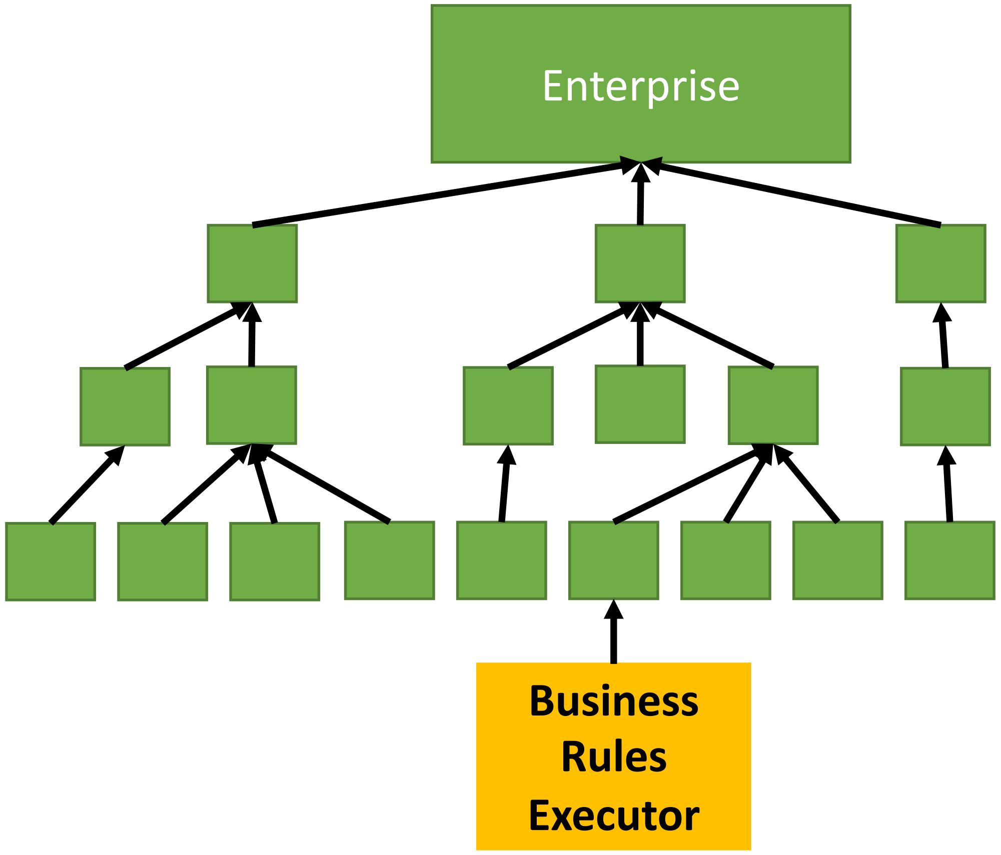
Application Health Model

- Health State
  - **OK (Green)**
  - **Warning (ORANGE): degradation**
  - **Critical (RED): not operational**
- Parent-Child
  - Cardinality
    - 1-to-1
    - 1-to-many
  - Health state propagation
    - Parent -> All Children
    - One of Children -> Parent
    - All Children -> Parent



Global Health Model

DevOps integrate the application into global health model



Ignite Health Model Components

- Physical or virtual machines, containers, Kubernetes pods
- Java and .NET VMs
  - Managed heap
  - GC
- Disk
  - Native persistence
  - Cluster snapshots
- Client and server nodes
- Cache Store
- Data regions

https://ignite.apache.org/docs/latest/

- Cluster topology
- Data Rebalancing
- Checkpointing
- Services
- Continuous Queries
- SQL & Scan Queries
- Transactions
- Compute Jobs
- Thread Pools
- Communication and Discovery Message Queues

Health Events

Events, Monitors, Alerts and Recovery

Monitoring & Management basics:

1. Application has logs and metrics that may trigger an **event**
2. Monitoring tool creates a **monitor** for every event
3. Occurrence of an event changes the monitor **state**
4. A change in the monitor state creates an **alert** and initiates automated **recovery** if provided

Health Events Are Actionable

- Describe only events that DevOps can act upon (Operationally Significant Events)
- Example health events:
  - Ignite cluster segmentation
  - JVM pause too long
  - Heap usage too high
  - Maven repo unavailable
- Example non-health events (do not include in Health Model)
  - Failed to parse SQL query
  - Invalid username or password

Health Events Are Specific

- Unambiguous recovery actions for DevOps
- Do not require additional investigation from DevOps
- Examples of specific events:
  - Failed to connect to maven repo
  - Malformed maven repo connection string
- Examples of non-specific event:
  - HTTP Exception
  - Malformed configuration

Event Collection

- Log File
  - Path
- JMX
  - Object name
  - Attribute name
- SQL
  - Ignite system view query
  - Application table query
- Command Line

Snapshot and Historical Event Collection

- Historical events
  - Append-only, multiple events in collection
  - Logs
  - Historical SQL tables
- Snapshot event:
  - single most-recent value
  - Most JMX metrics
  - Snapshot SQL tables
- Important for automated recovery (will review later)

Event Trigger Expression

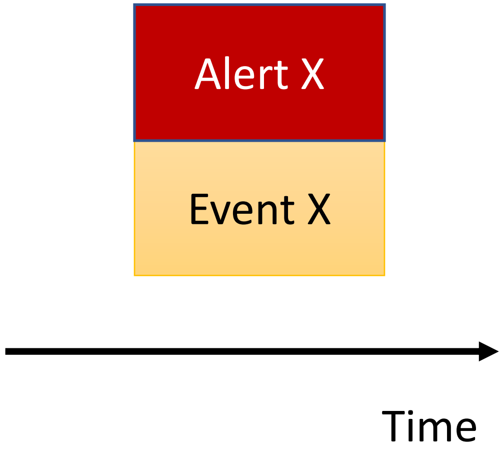
Event is created when the expression becomes “true”:

- Operators:
  - Text: contains, matches (regexp)
  - Relational: =, <, >, >=, <=
  - Logical: not, and, or
- Left operand: event source (implicit)
- Right operand: literal value
- Examples:
  - contains “[10530]”
  - > 95
  - (>= 10) or (< 20)

Alerts

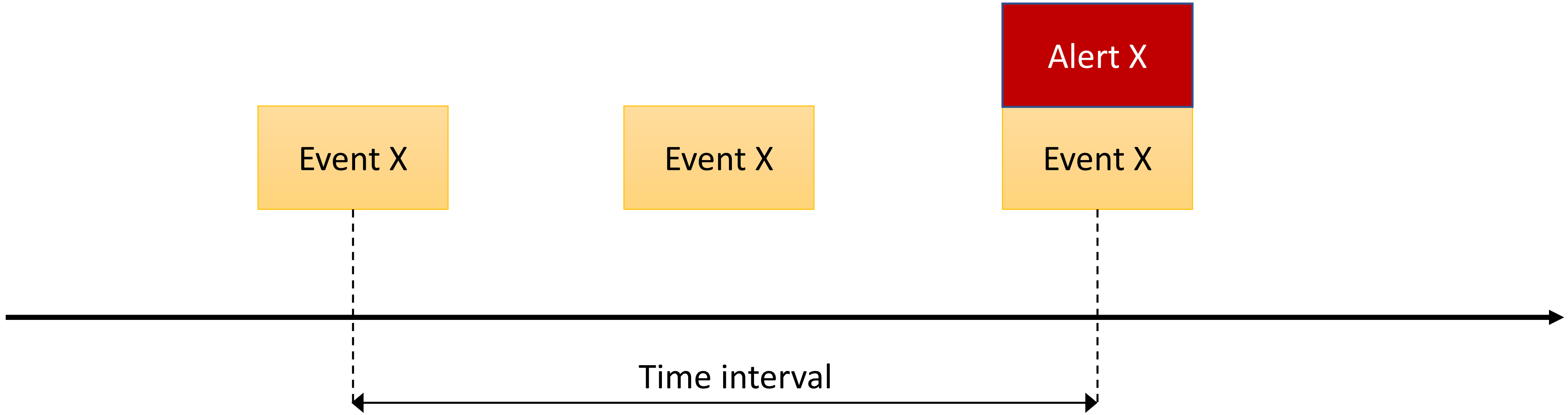
- Monitoring tool alerts on health events
  - Email, SMS, Phone
  - Ticket
- DevOps are responsible for configuring alerts
- Single event creates an alert on the first occurrence of the event

Single Event



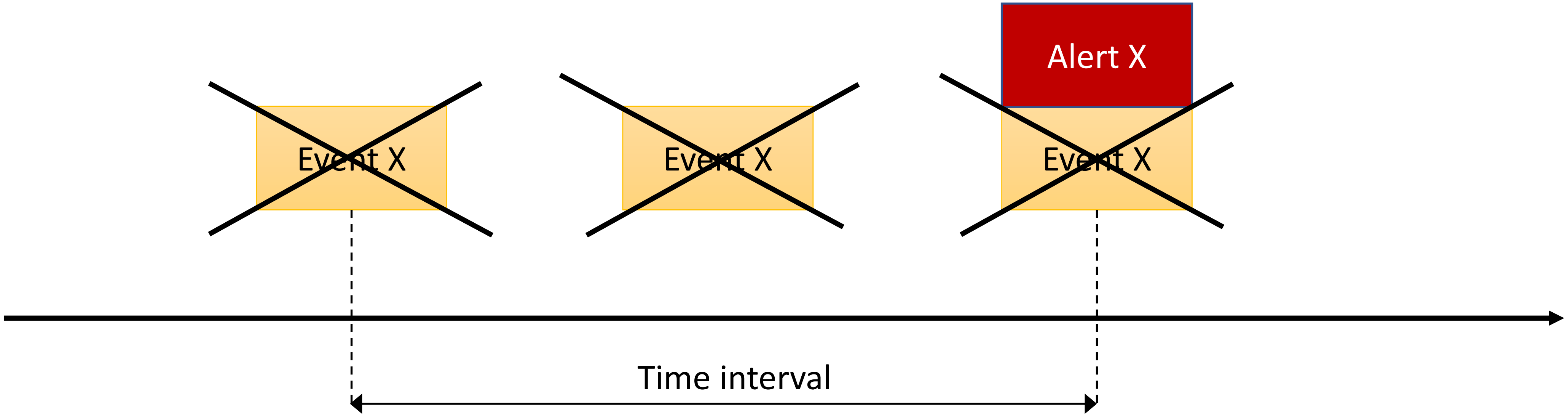
Alerts: Repeated Event

- Number of events
- Time interval



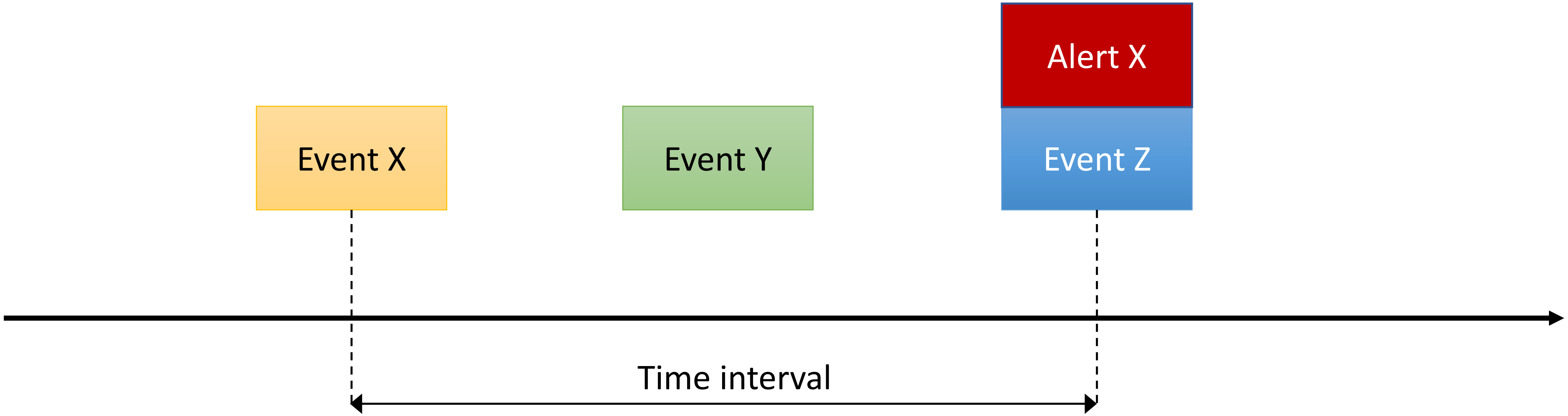
Alerts: Missing Event

- Number of events
- Time interval



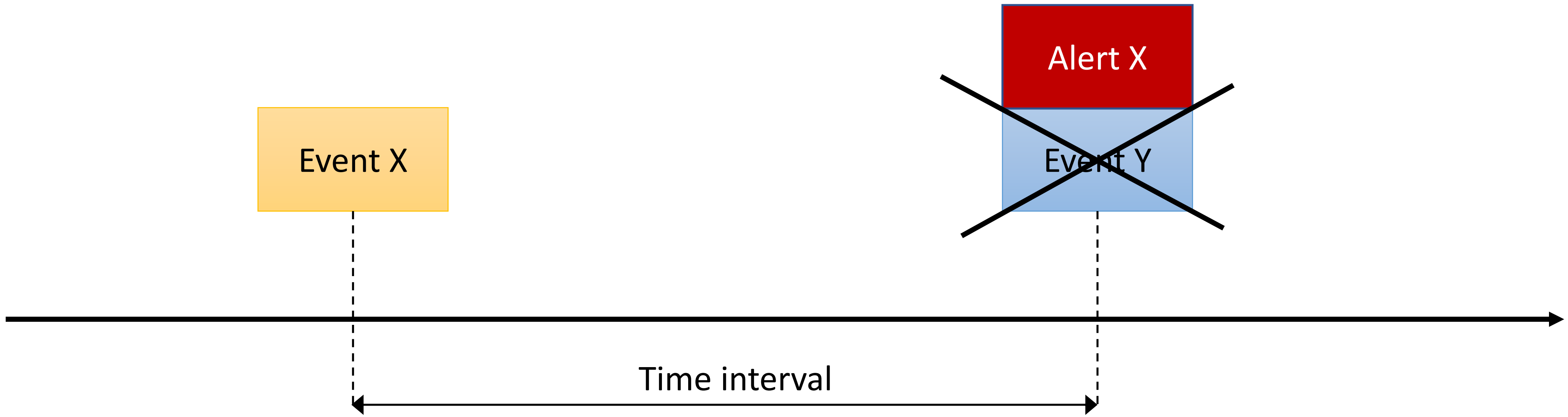
Alerts: Correlated Event

- Time interval
- Event Y, …, Event Z



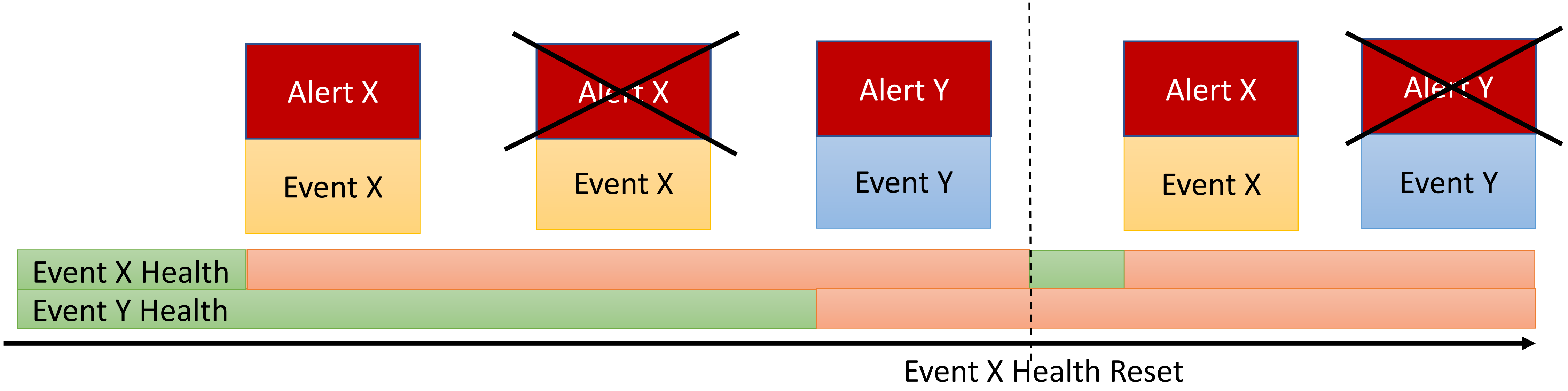
Alerts: Correlated Missing Event

- Time interval
- Missing Event



Alerts: Health State Reset

- No alerts for unhealthy events (in **warning** or **critical** state)
- Reset event health state to enable alerting on that event
- Health state is event-specific
- Component is unhealthy if at least one event state is unhealthy



Automated Health Reset

- **Expression**: reset health when the event’s expression becomes false
  - Not applicable to historical event collections (log files)
  - Example: JVM Heap > 4 GB
- **Event**: reset health on occurrence of another event
  - Example: contains “Connected to Audit cache store”
- **Timer**: reset health after specified timeout
  - When you are sure the problem fixes itself within a time interval

Manual Health Reset

- As a developer, do your best to automate your application recovery
  - Ignite Critical Failure Handler
  - Retry cycles in the application
  - Develop a mechanism to reset/restart application Health components that Ignite lacks (like Ignite services, continuous queries)

Event Schedule

- Avoid opening alerts for expected scheduled downtime
- Example schedules:
  - 24x7
  - M-F
  - M-F 8am-7pm

Event Knowledge

- Description
  - Why the event occurred
  - What is the impact
- Recovery actions
  - Specific
  - Avoid if-else
  - Easy to automate if possible

Log Monitoring

- Historical event collection
  - Automated health reset on false event expression not supported
  - Write another “healthy” event on recovery to automatically reset state
- Include a unique ID to the log message
  - Identifies specific event
  - ID never changes
  - Example log message: [10105] Failed to connect to BI Maven repo
- Be careful with regular expressions
  - A complex regex that is fast with on a small log can kill a production server with a huge log

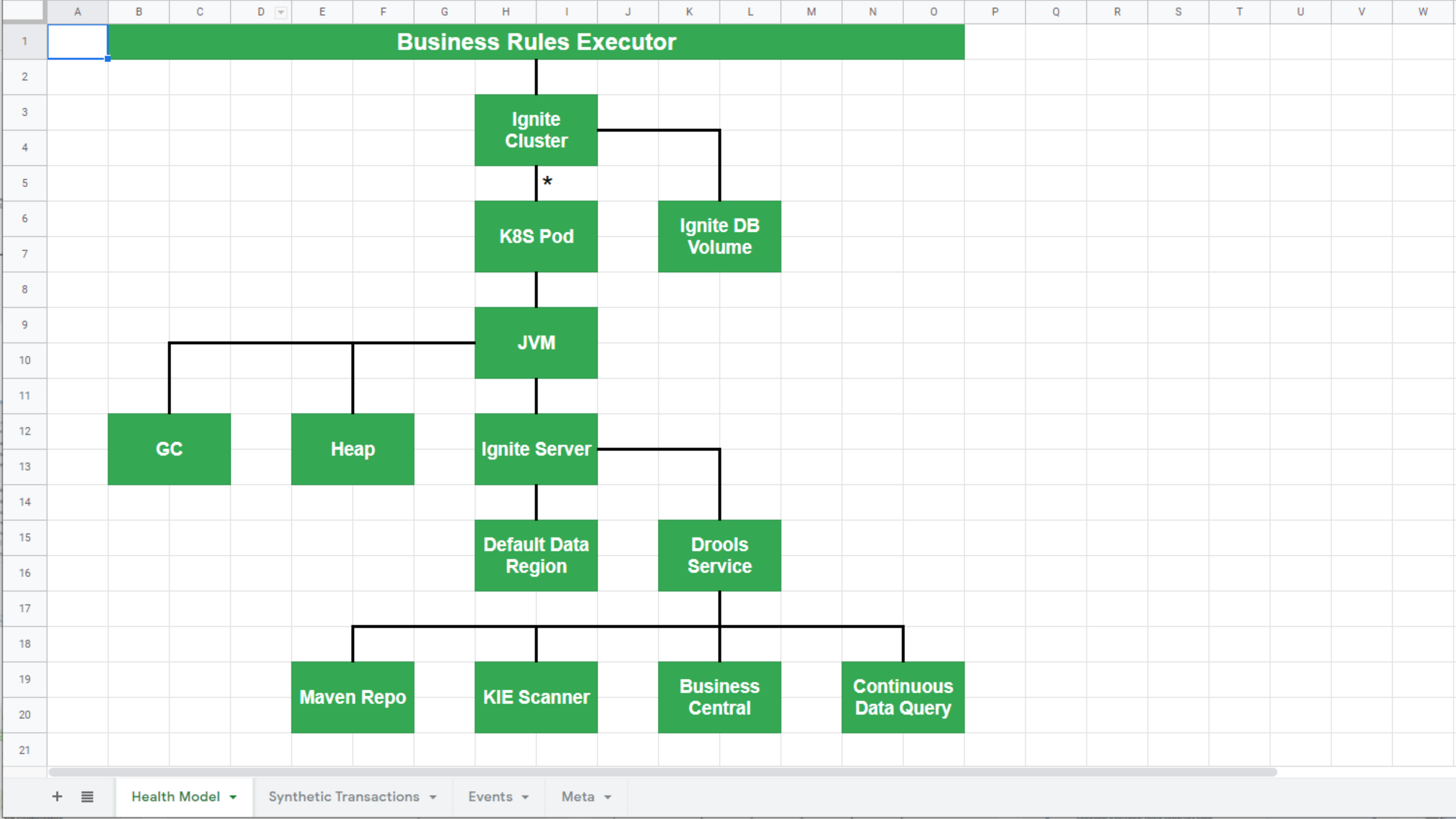
Synthetic Monitoring

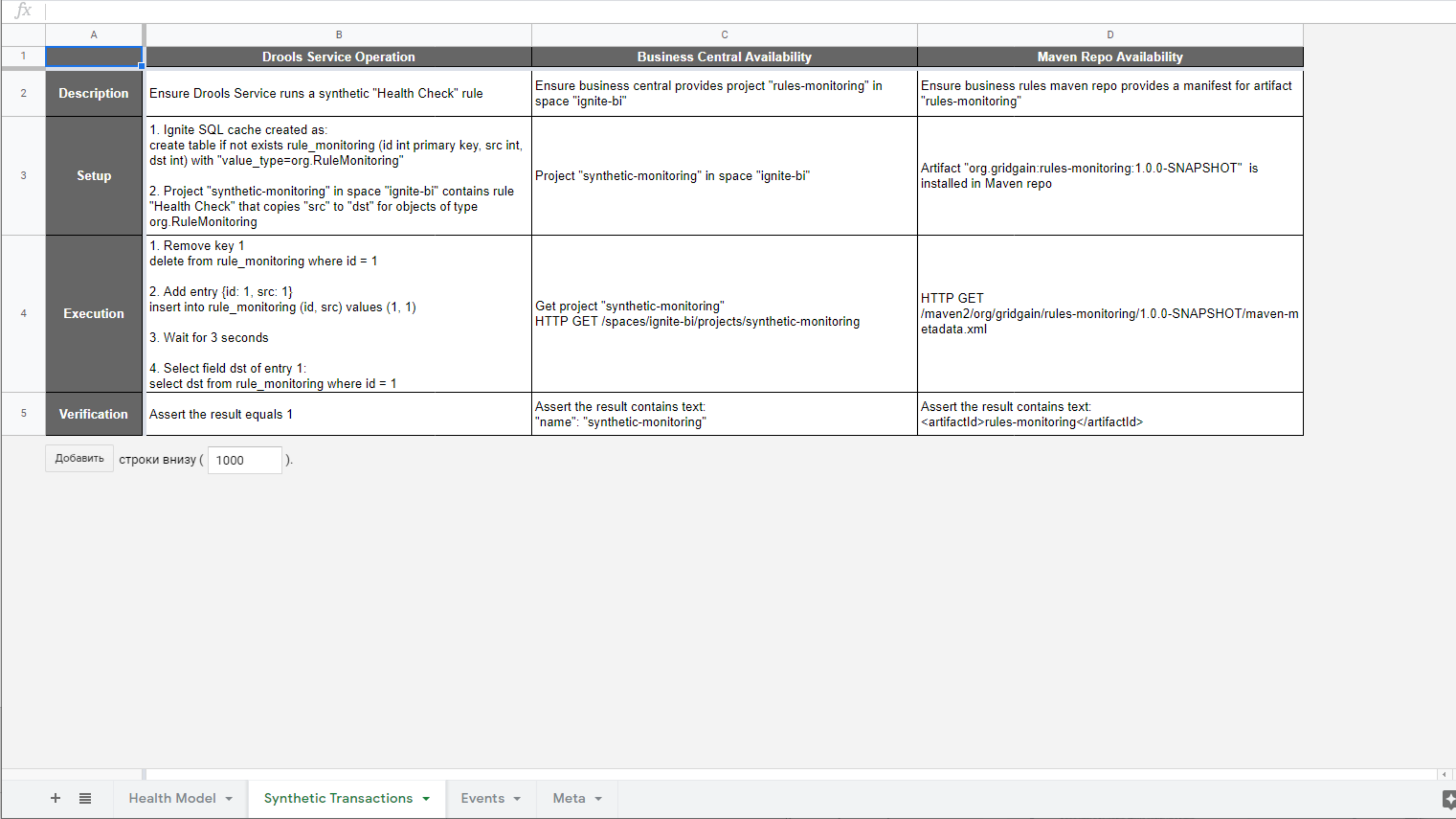
Synthetic Transactions:

- Proactively detect problems (before real users)
- Subset of automated tests
- Consider creating a Health Check operation
- Against live production
- Run Periodically

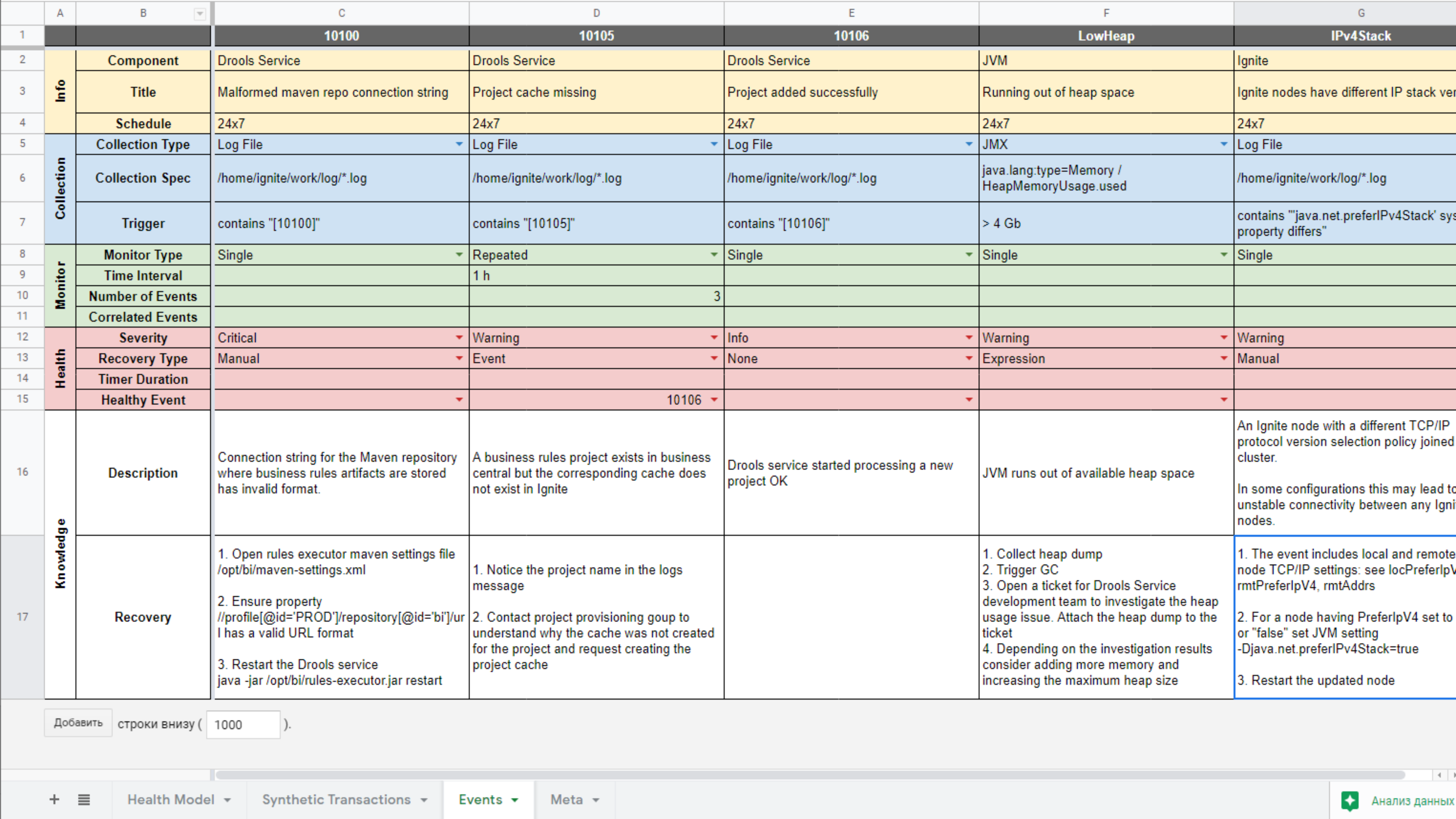
Monitoring Spec

- Developers create Monitoring Spec to communicate Health Model to DevOps
- Content
  - Health Model Components Hierarchy
  - Synthetic Monitoring
  - Health Events


A B C D E F G H I J K L M N O P Q R S T U V W
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
Business Rules Executor
Ignite Cluster
*
K8S Pod
Ignite DB Volume
JVM
GC
Heap
Ignite Server
Default Data Region
Drools Service
Maven Repo
KIE Scanner
Business Central
Continuous Data Query
+
Health Model
Synthetic Transactions
Events
Meta



	Drools Service Operation	Business Central Availability	Maven Repo Availability
Description	Ensure Drools Service runs a synthetic "Health Check" rule	Ensure business central provides project "rules-monitoring" in space "ignite-bi"	Ensure business rules maven repo provides a manifest for artifact "rules-monitoring"
Setup	1. Ignite SQL cache created as: create table if not exists rule_monitoring (id int primary key, src int, dst int) with "value_type=org.RuleMonitoring" 2. Project "synthetic-monitoring" in space "ignite-bi" contains rule "Health Check" that copies "src" to "dst" for objects of type org.RuleMonitoring	Project "synthetic-monitoring" in space "ignite-bi"	Artifact "org.gridgain:rules-monitoring:1.0.0-SNAPSHOT" is installed in Maven repo
Execution	1. Remove key 1 delete from rule_monitoring where id = 1 2. Add entry {id: 1, src: 1} insert into rule_monitoring (id, src) values (1, 1) 3. Wait for 3 seconds 4. Select field dst of entry 1: select dst from rule_monitoring where id = 1	Get project "synthetic-monitoring" HTTP GET /spaces/ignite-bi/projects/synthetic-monitoring	HTTP GET /maven2/org/gridgain/rules-monitoring/1.0.0-SNAPSHOT/maven-metadata.xml
Verification	Assert the result equals 1	Assert the result contains text: "name": "synthetic-monitoring"	Assert the result contains text: <artifactId>rules-monitoring</artifactId>



	A	B	C	D	E	F	G
1			10100	10105	10106	LowHeap	IPv4Stack
2	Info	Component	Drools Service	Drools Service	Drools Service	JVM	Ignite
3		Title	Malformed maven repo connection string	Project cache missing	Project added successfully	Running out of heap space	Ignite nodes have different IP stack ver
4		Schedule	24x7	24x7	24x7	24x7	24x7
5	Collection	Collection Type	Log File	Log File	Log File	JMX	Log File
6		Collection Spec	/home/ignite/work/log/*.log	/home/ignite/work/log/*.log	/home/ignite/work/log/*.log	java.lang:type=Memory / HeapMemoryUsage.used	/home/ignite/work/log/*.log
7		Trigger	contains "[10100]"	contains "[10105]"	contains "[10106]"	> 4 Gb	contains "'java.net.preferIPv4Stack' sys property differs"
8	Monitor	Monitor Type	Single	Repeated	Single	Single	Single
9		Time Interval		1 h			
10		Number of Events		3			
11		Correlated Events					
12	Health	Severity	Critical	Warning	Info	Warning	Warning
13		Recovery Type	Manual	Event	None	Expression	Manual
14		Timer Duration					
15		Healthy Event		10106			
16	Knowledge	Description	Connection string for the Maven repository where business rules artifacts are stored has invalid format.	A business rules project exists in business central but the corresponding cache does not exist in Ignite	Drools service started processing a new project OK	JVM runs out of available heap space	An Ignite node with a different TCP/IP protocol version selection policy joined cluster. In some configurations this may lead to unstable connectivity between any Igni nodes.
17		Recovery	1. Open rules executor maven settings file /opt/bi/maven-settings.xml 2. Ensure property //profile[@id='PROD']/repository[@id='bi']/url has a valid URL format 3. Restart the Drools service java -jar /opt/bi/rules-executor.jar restart	1. Notice the project name in the logs message 2. Contact project provisioning goup to understand why the cache was not created for the project and request creating the project cache		1. Collect heap dump 2. Trigger GC 3. Open a ticket for Drools Service development team to investigate the heap usage issue. Attach the heap dump to the ticket 4. Depending on the investigation results consider adding more memory and increasing the maximum heap size	1. The event includes local and remote node TCP/IP settings: see locPreferIpV rmtPreferIpV4, rmtAddrs 2. For a node having PreferIpV4 set to or "false" set JVM setting -Djava.net.preferIPv4Stack=true 3. Restart the updated node

Monitoring Tests

- Develop automated tests for abnormal scenarios
  - Include health event verification
- Test synthetic transactions
- Part of CI/CD pipeline

Monitoring & Management Tools

The tools fully supporting the described Health Model:

- Zabbix + Grafana
- Prometheus + ELK (Elasticsearch, Logstash, Kibana) + Grafana
- Microsoft SCOM / Microsoft Aquila (preview)

Apache Ignite Health Model

Apache Ignite Health Model

- The described Health Model is not specified for Ignite
- Include Ignite health model to your application health model
    - Include only events applicable to your application
- No event IDs in Apache Ignite log files
    - Use “contains” or “matches” to parse message text
    - Test on every Apache Ignite upgrade

Ignite Components Restart

- Management interface:
  - **Command line**: control.sh, ignitevisorcmd.sh
  - **SQL**: KILL (sqlline.sh is included in Ignite)
  - REST
  - JMX
  - API
- Restart (stop/start) is not available for many Ignite components
  - Continues queries cannot be cancelled and started
  - Services can be cancelled and started only in with Ignite API
- Consider adding management interface to your application

Ignite Node Configuration

Configuration:

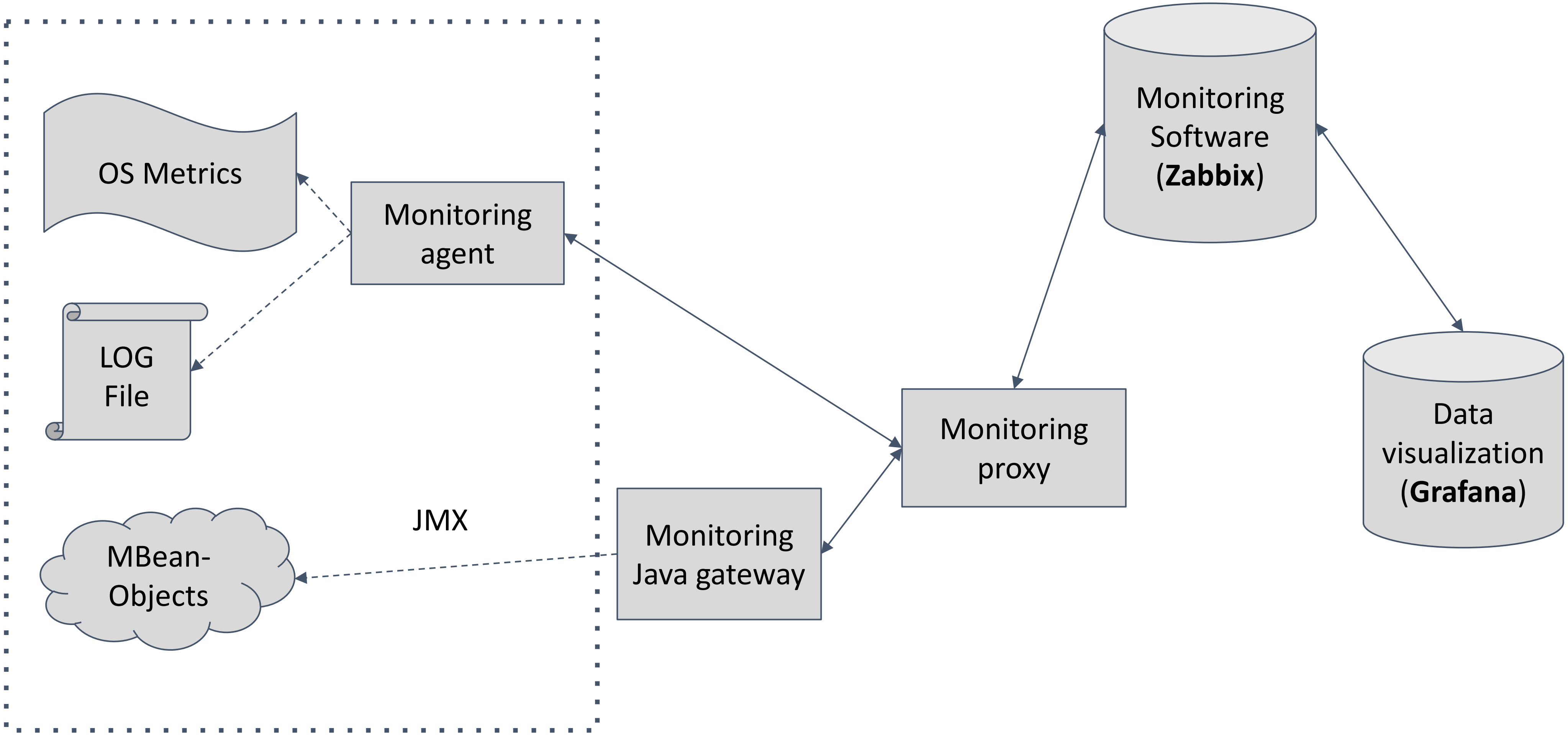
- Lifetime
  - Static
    - Spring XML (Java, .NET)
    - .NET App.config
    - Code (any platform)
  - Dynamic
    - Caches and indexes (code or SQL DDL)
      - Permanent if persistence is enabled
    - Cluster state, ID & Tag, statistics on/off
- Scope
  - Cluster-wide
    - Caches, services, binary configuration, etc.
  - Node-local
    - Work directories, data region, WAL, thread pools, etc.

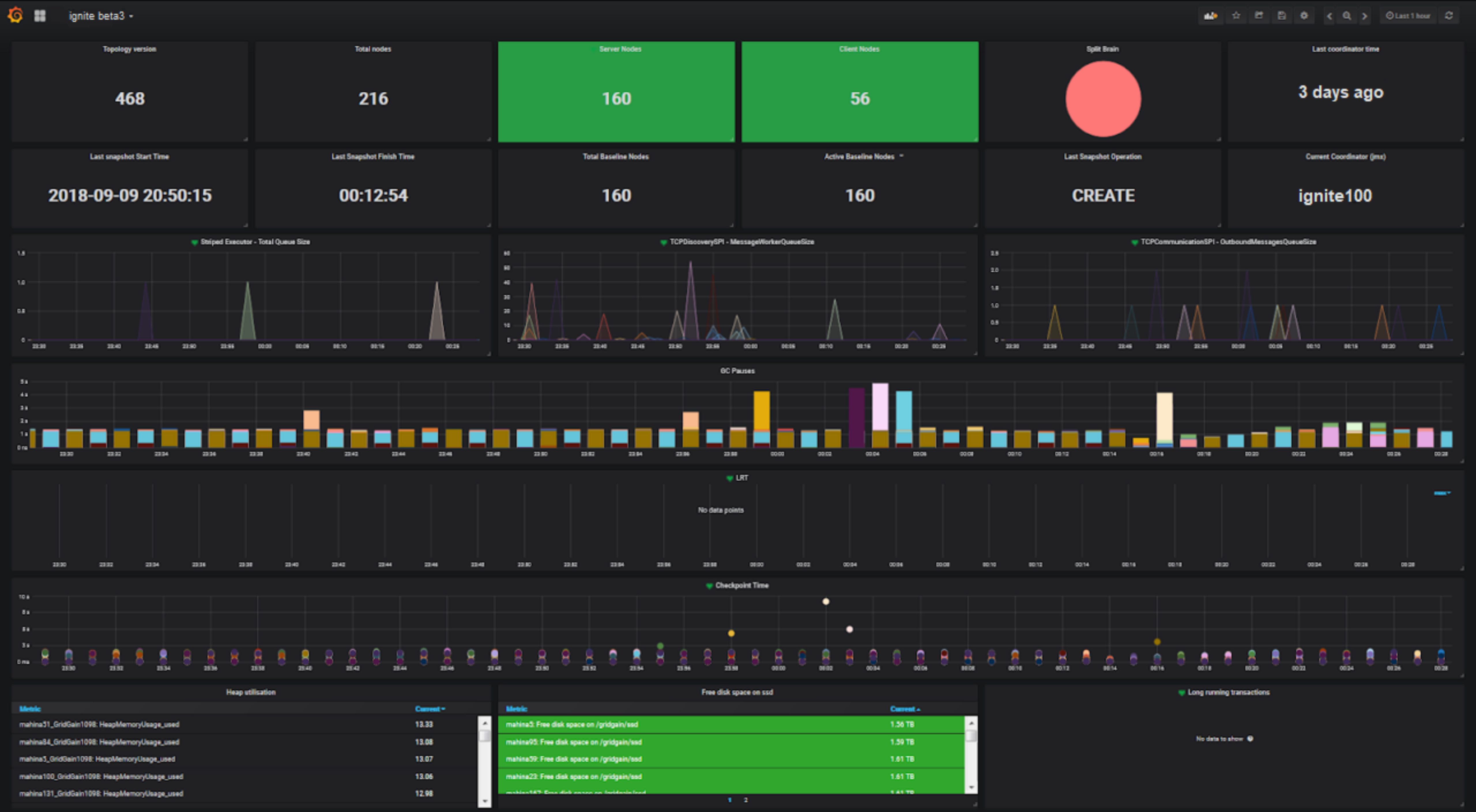
Ignite Node Restart

1. Deactivate if native persistence is enabled
   - Ignite REST API, control.sh, visor or Web Console
2. Restart
   - Kill/start process or delete Kubernetes pod

Demo: Monitoring with Zabbix & Grafana

Monitoring Architecture





Topology version
468

Total nodes
216

Server Nodes
160

Client Nodes
56

Split Brain

Last coordinator time
3 days ago

Last snapshot Start Time
2018-09-09 20:50:15

Last Snapshot Finish Time
00:12:54

Total Baseline Nodes
160

Active Baseline Nodes
160

Last Snapshot Operation
CREATE

Current Coordinator (jmx)
ignite100

Striped Executor - Total Queue Size
TCPDiscoverySPI - MessageWorkerQueueSize
TCPCommunicationSPI - OutboundMessagesQueueSize
GC Pauses
LRT
Checkpoint Time

No data points

Heap utilisation

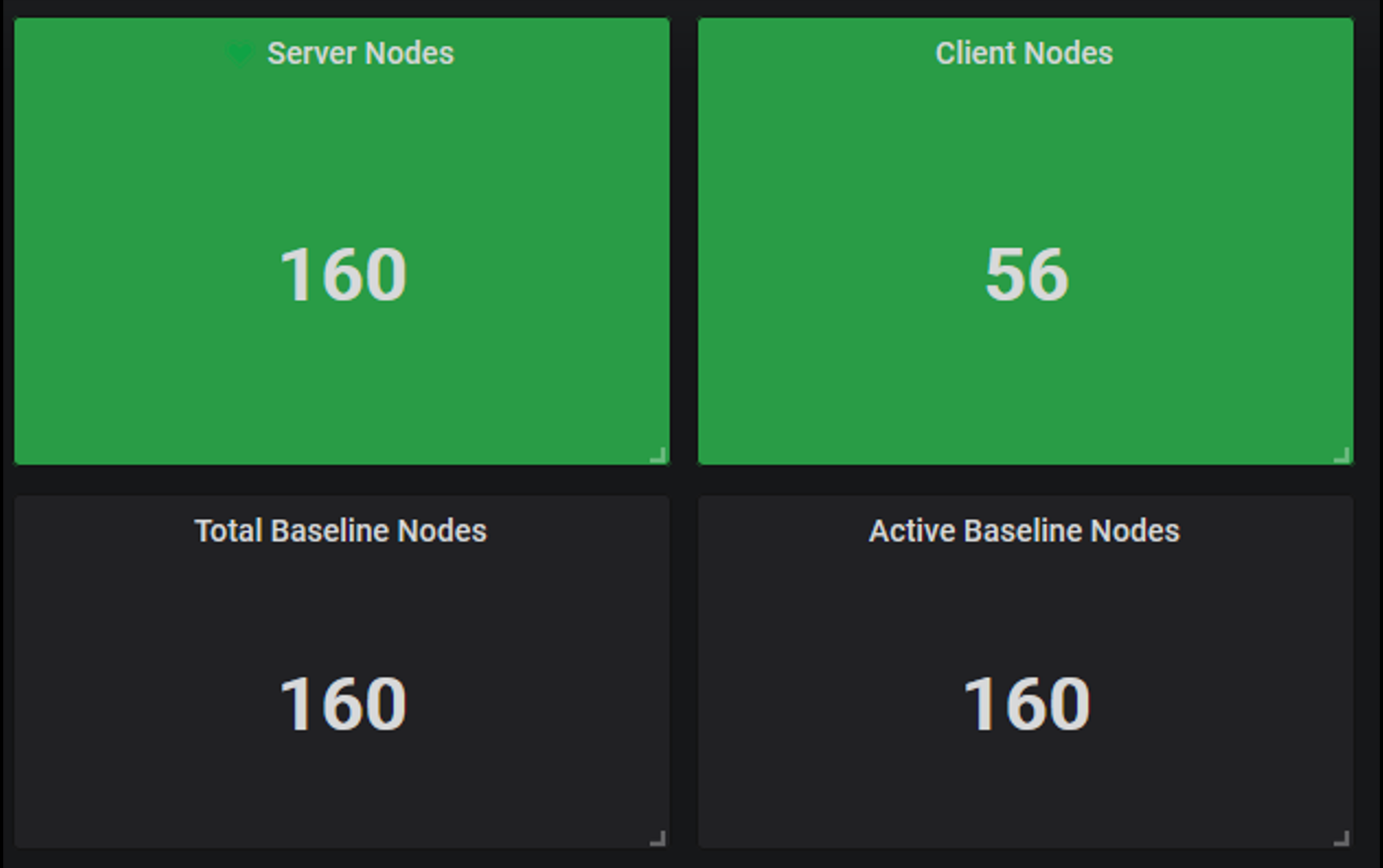
Metric	Current
mahina31_GridGain1098: HeapMemoryUsage_used	13.33
mahina84_GridGain1098: HeapMemoryUsage_used	13.08
mahina3_GridGain1098: HeapMemoryUsage_used	13.07
mahina100_GridGain1098: HeapMemoryUsage_used	13.06
mahina131_GridGain1098: HeapMemoryUsage_used	12.98

Free disk space on ssd

Metric	Current
mahina5: Free disk space on /gridgain/ssd	1.56 TB
mahina95: Free disk space on /gridgain/ssd	1.59 TB
mahina59: Free disk space on /gridgain/ssd	1.61 TB
mahina23: Free disk space on /gridgain/ssd	1.61 TB

Long running transactions

No data to show

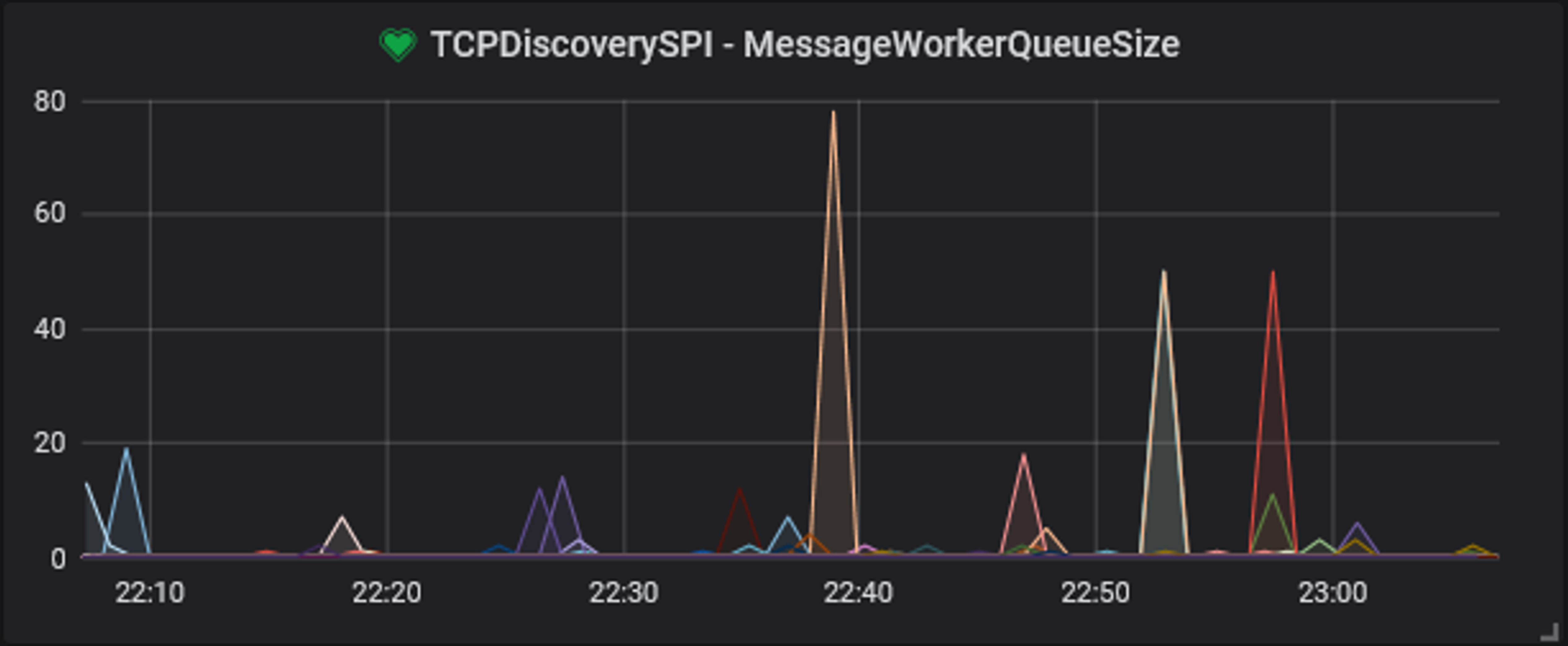


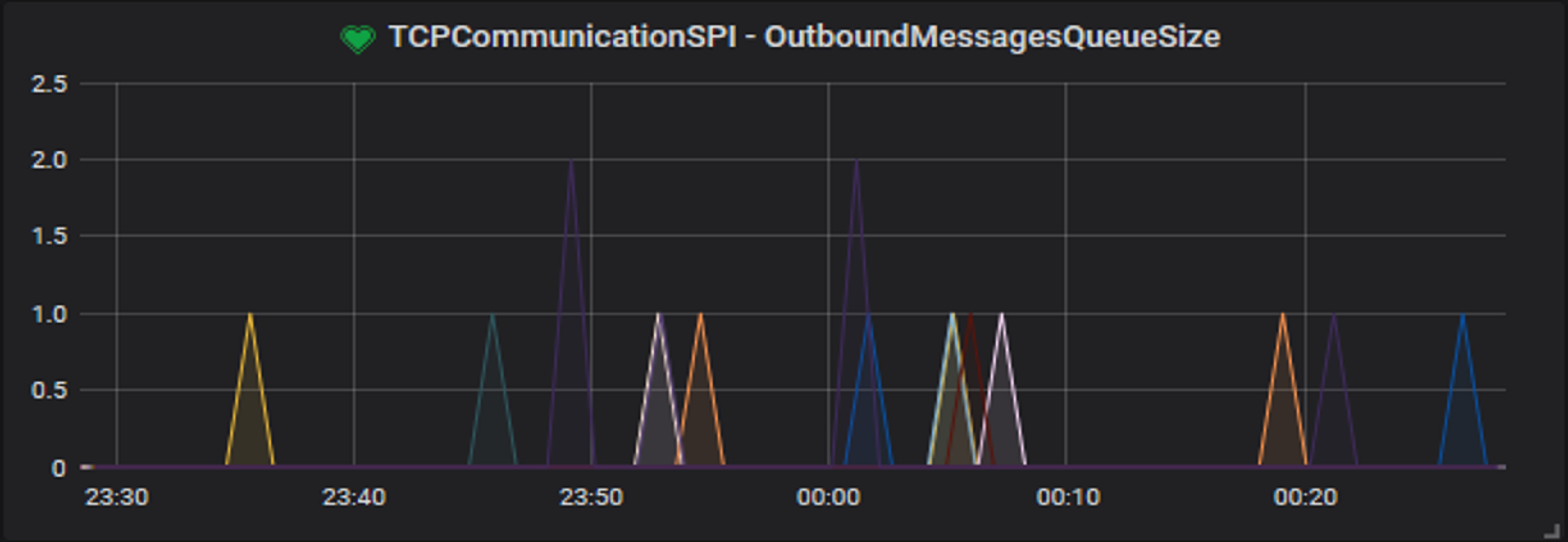
Server Nodes
160

Client Nodes
56

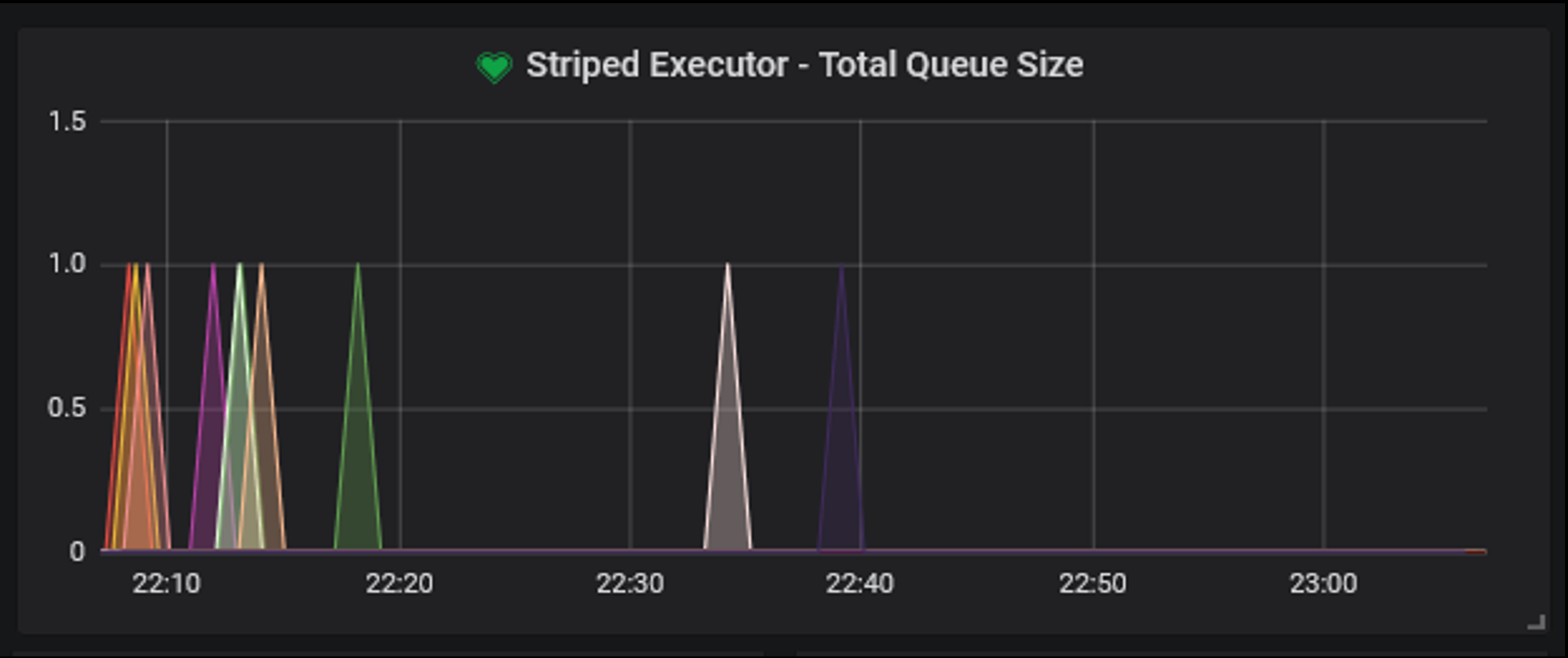
Total Baseline Nodes
160

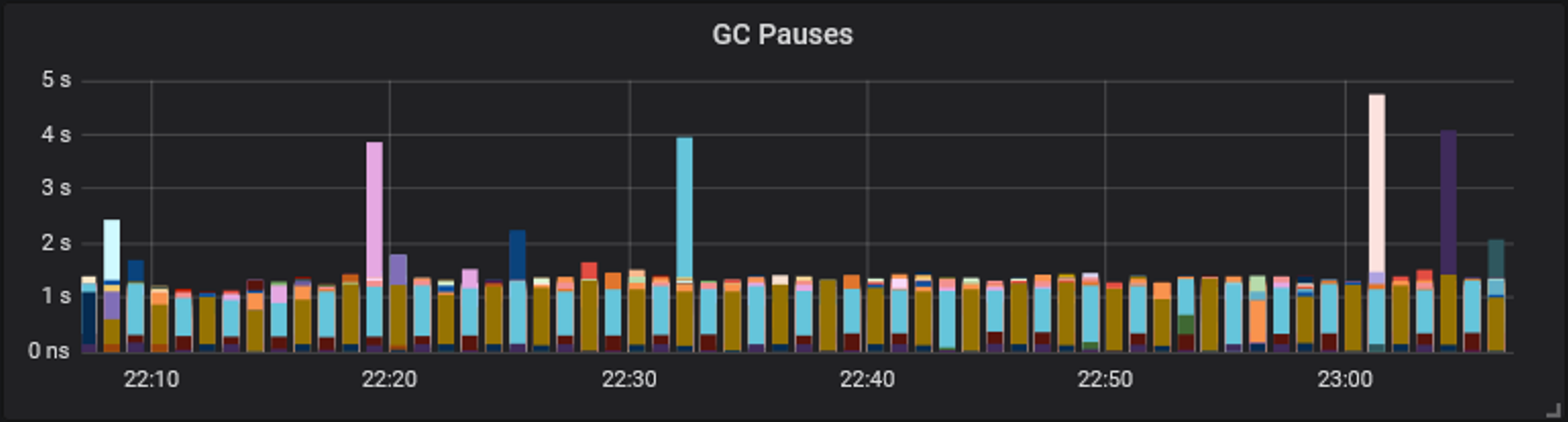
Active Baseline Nodes
160

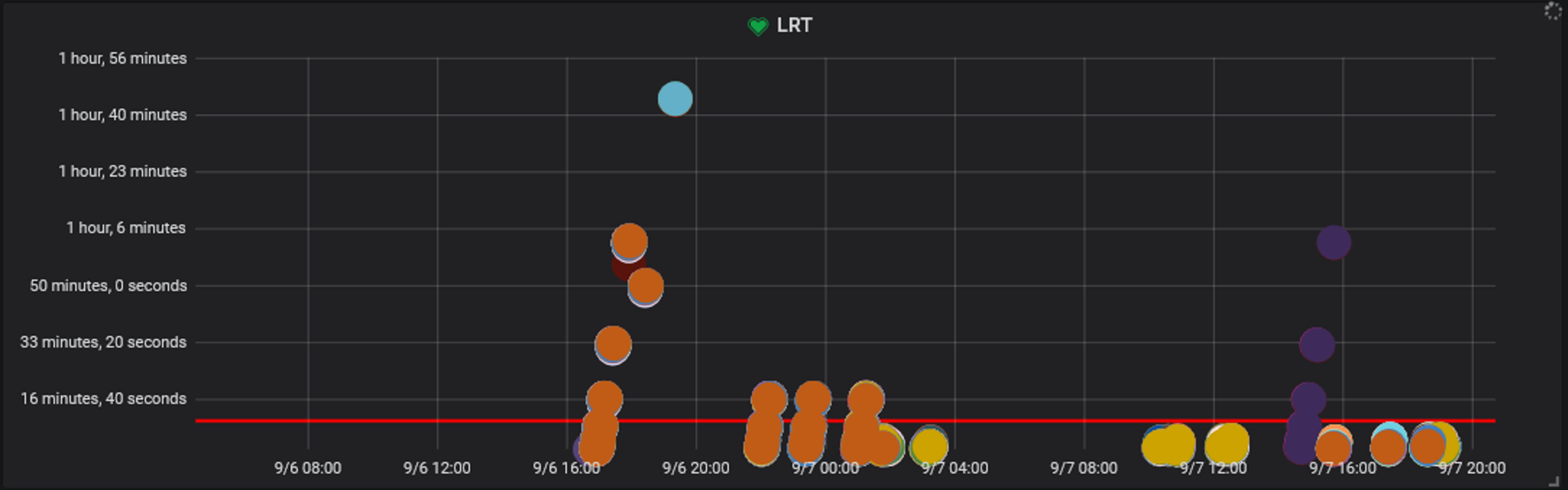

TCPDiscoverySPI - MessageWorkerQueueSize
80
60
40
20
0
22:10
22:20
22:30
22:40
22:50
23:00

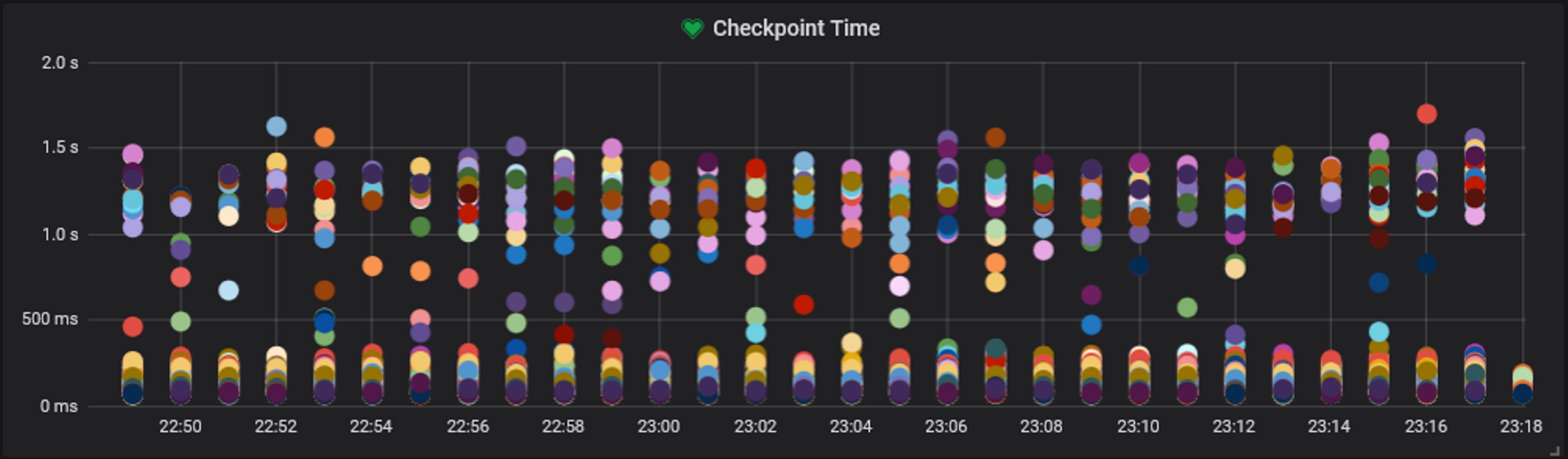


TCPCommunicationSPI - OutboundMessagesQueueSize
2.5
2.0
1.5
1.0
0.5
0
23:30
23:40
23:50
00:00
00:10
00:20

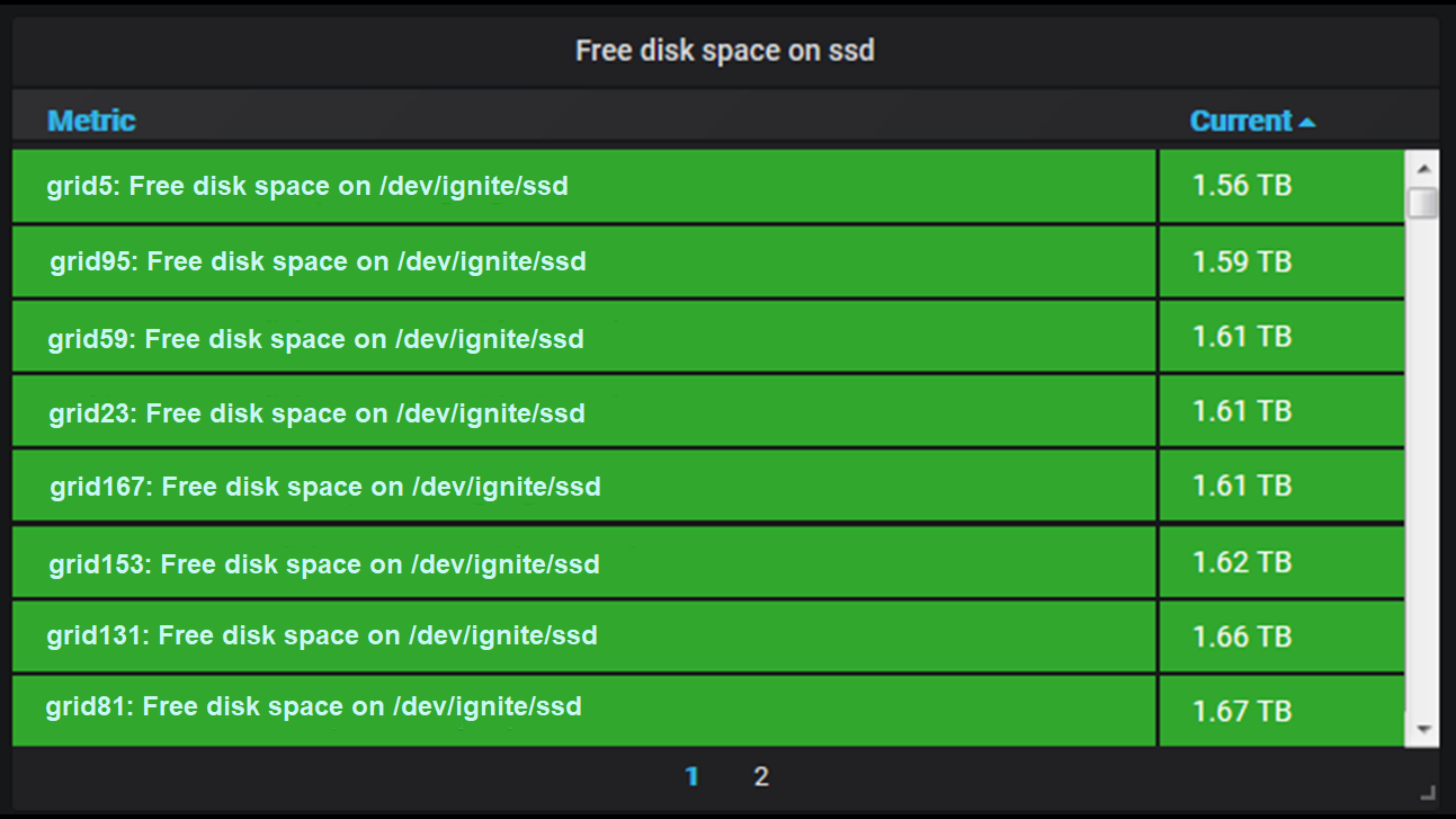

Striped Executor - Total Queue Size
1.5
1.0
0.5
0
22:10
22:20
22:30
22:40
22:50
23:00


GC Pauses
5 s
4 s
3 s
2 s
1 s
0 ns
22:10
22:20
22:30
22:40
22:50
23:00


LRT
1 hour, 56 minutes
1 hour, 40 minutes
1 hour, 23 minutes
1 hour, 6 minutes
50 minutes, 0 seconds
33 minutes, 20 seconds
16 minutes, 40 seconds
9/6 08:00
9/6 12:00
9/6 16:00
9/6 20:00
9/7 00:00
9/7 04:00
9/7 08:00
9/7 12:00
9/7 16:00
9/7 20:00

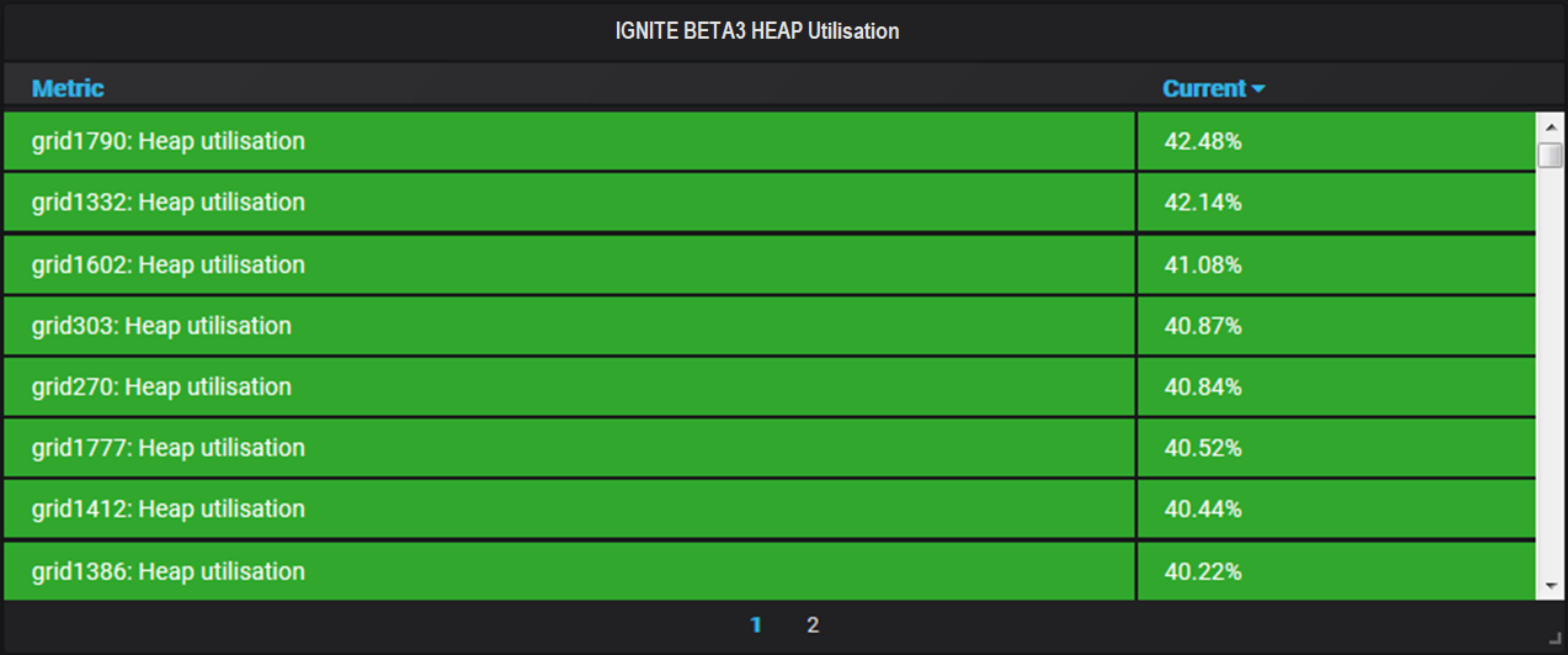


Checkpoint Time
2.0 s
1.5 s
1.0 s
500 ms
0 ms
22:50
22:52
22:54
22:56
22:58
23:00
23:02
23:04
23:06
23:08
23:10
23:12
23:14
23:16
23:18



Free disk space on ssd

Metric	Current ▲
grid5: Free disk space on /dev/ignite/ssd	1.56 TB
grid95: Free disk space on /dev/ignite/ssd	1.59 TB
grid59: Free disk space on /dev/ignite/ssd	1.61 TB
grid23: Free disk space on /dev/ignite/ssd	1.61 TB
grid167: Free disk space on /dev/ignite/ssd	1.61 TB
grid153: Free disk space on /dev/ignite/ssd	1.62 TB
grid131: Free disk space on /dev/ignite/ssd	1.66 TB
grid81: Free disk space on /dev/ignite/ssd	1.67 TB



IGNITE BETA3 HEAP Utilisation

Metric	Current
grid1790: Heap utilisation	42.48%
grid1332: Heap utilisation	42.14%
grid1602: Heap utilisation	41.08%
grid303: Heap utilisation	40.87%
grid270: Heap utilisation	40.84%
grid1777: Heap utilisation	40.52%
grid1412: Heap utilisation	40.44%
grid1386: Heap utilisation	40.22%

Summary

It is the developer’s responsibility to:

1. Develop Health Model
2. Instrument the code
3. Create and maintain Monitoring Spec for DevOps
4. Develop and add monitoring tests to CI/CD pipeline

Additional development cost is much smaller than the reduced total cost of ownership.