

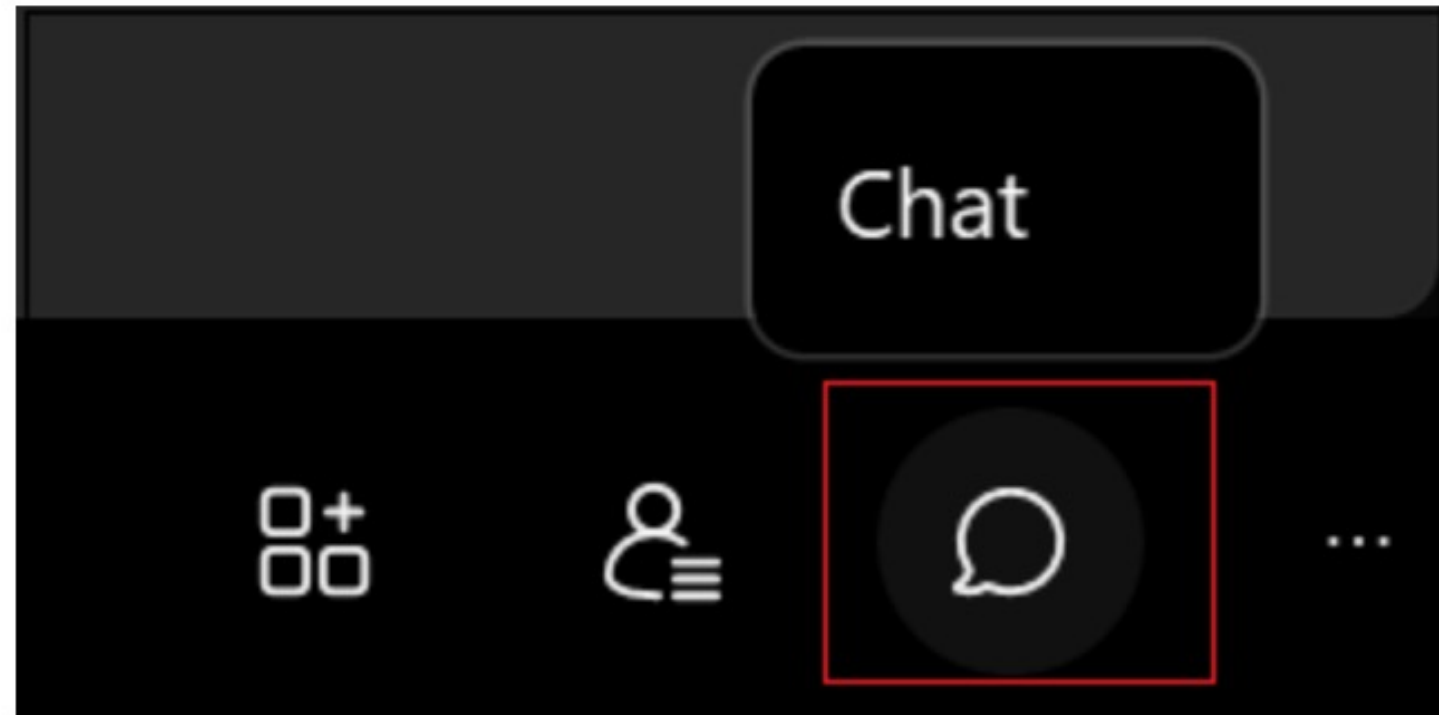
Holiday Readiness 2024

2023
Peak
Retrospect

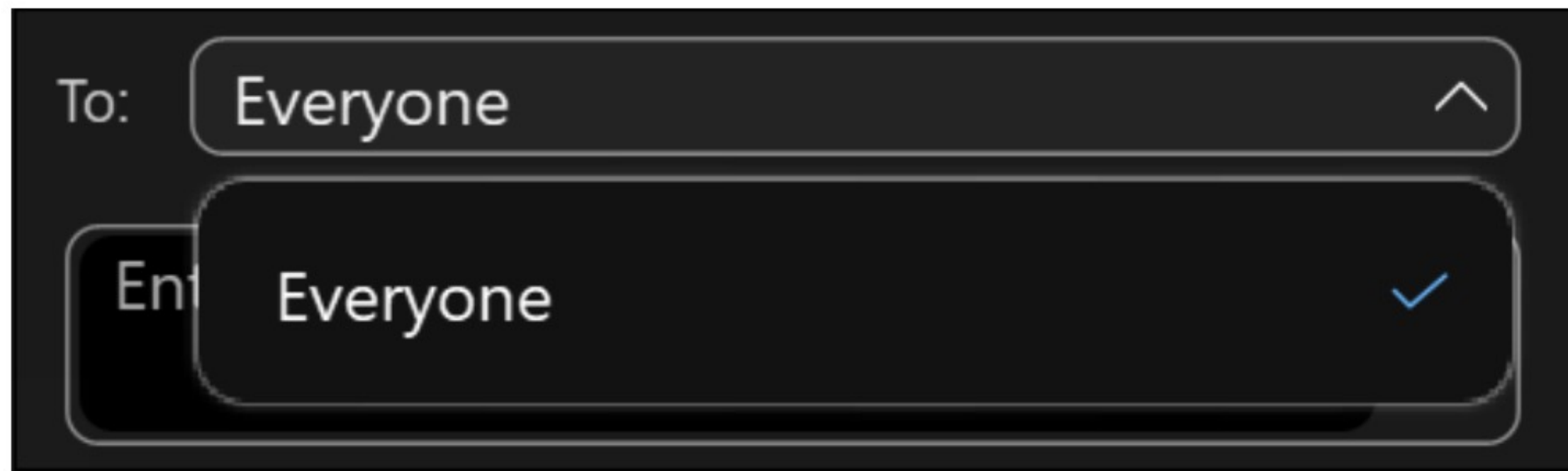


Have a Question(s)?

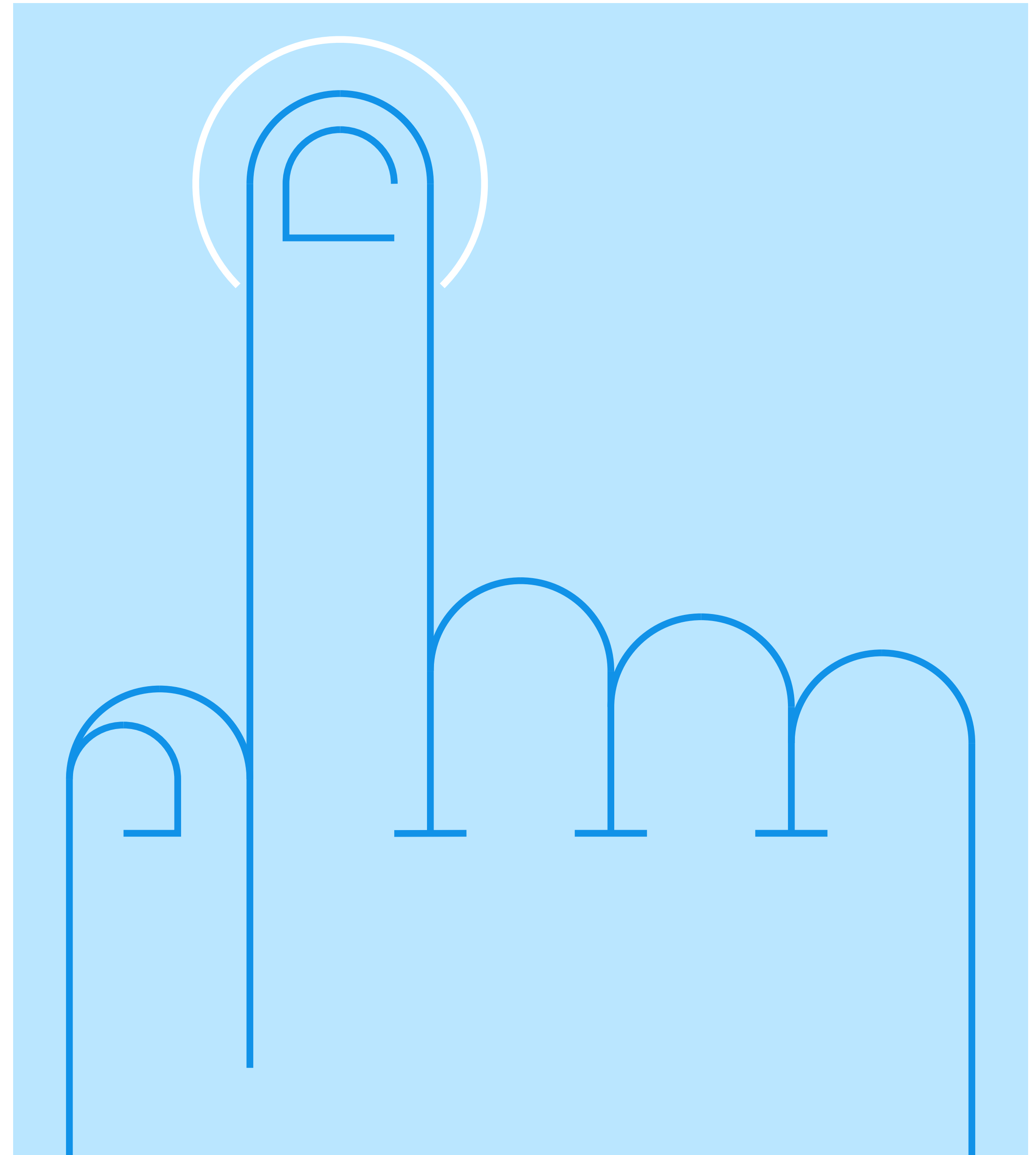
- 1 Open the Chat panel from the link in the lower right of the meeting window:



- 2 In the **To** drop-down list, select the recipient of the message.



- 3 Enter your message in the chat text box, then press **Enter** on your keyboard.



Your Holiday Readiness Team

... and today's speakers



Mike Callaghan
Program Director –
WW Supply Chain Support



Shoeb Bihari
Technical Lead / SRE Advisor –
Order Management Support



Senthil Ponnusamy
Technical Lead / SRE Advisor –
Order Management Support



Chiranjeevi Dasegowda
Technical Support Analyst
Order Management Support



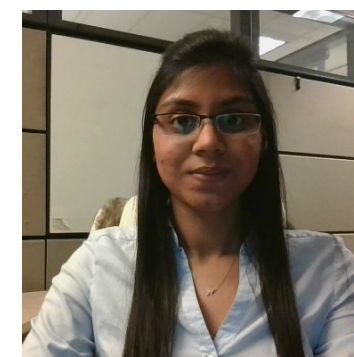
Jitendra Buge
Technical lead
Order Management Support



Paresh Vinaykya
Executive Technical Account
Manager – Expertise Connect



Jelena Markovic
Technical Support Analyst
Order Management Support



Damini Tacouri
Technical Support Analyst
Order Management Support

Agenda



Our Journey to peak success

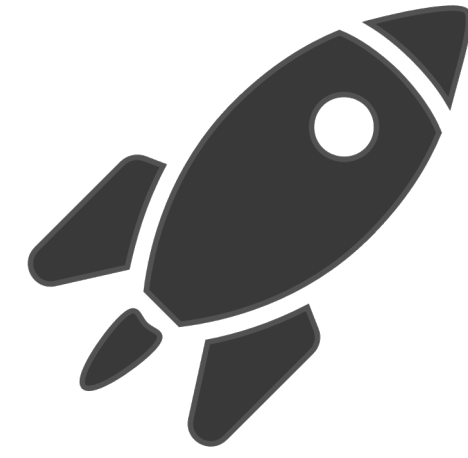
Key Metrics

Enhancements

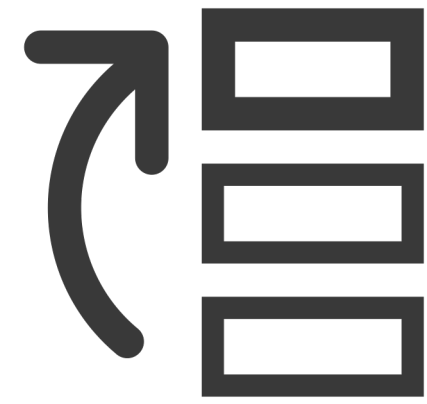
Common Issues → [Recommendations](#)

- Case Studies
- Payment Integration
- Handling Node/DG Updates
- Effective Server Consolidation
- Common JMS Errors
- Certified Containers

What are *your* plans in 2024 for IBM Order Management?



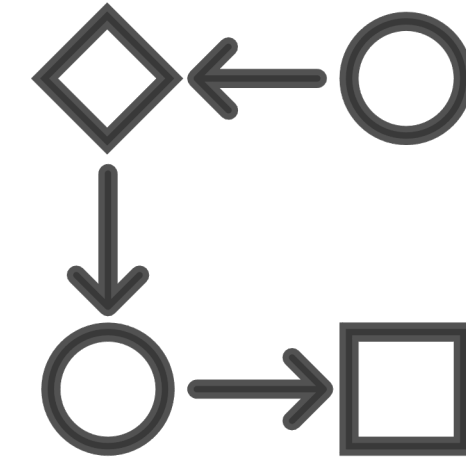
First Go-live



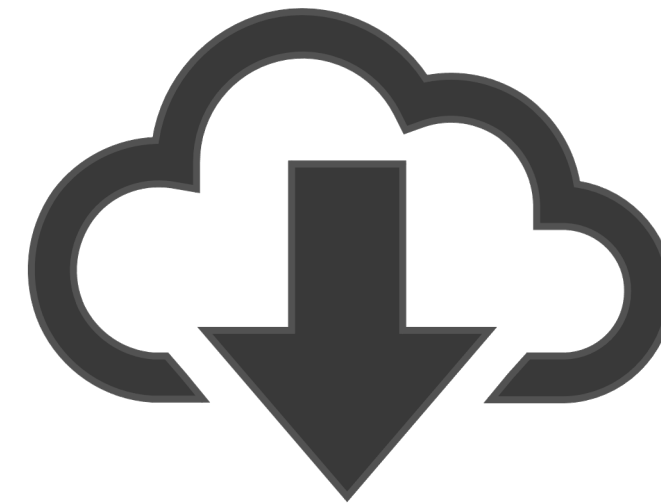
Stack upgrades



Container deployment



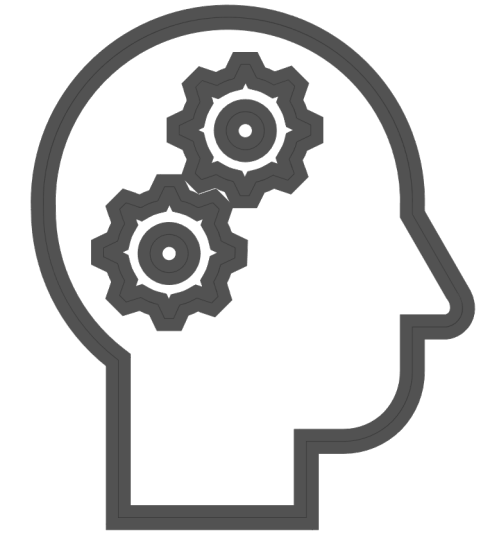
OMoC migration



On-Prem to Cloud



Expansion



Sterling Intelligent Promising



Call Center, Store, Order Hub



Holiday Peak Season

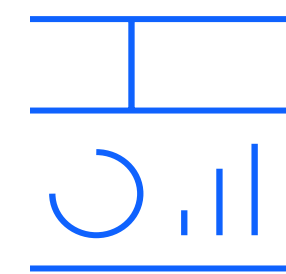
IBM OMS Holiday Readiness

Our Mission Statement



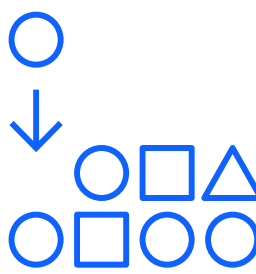
- ✓ 2023-03 | [Journey to Peak Success](#)
- ✓ 2023-05 | [Payment Integration](#)
- ✓ 2023-08 | [Best Practices](#)
- ✓ 2023-09 | [Panel Discussion](#)
- ✓ 2023-10 | [Peak Day Preparedness](#)

In case you missed it...
ibm.biz/IBM-OMS-HolidayReadiness



Stable Platform

Continuous improvement of platform and monitoring, with focus on performance, stability, reliability



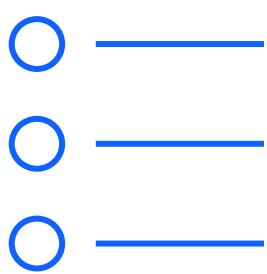
Best Practices

Establish, expand and apply a robust collection of proven self-help best practices focused on peak season success



Proactive Engagement

Early and regular identification, communication, and mitigation of potential risks

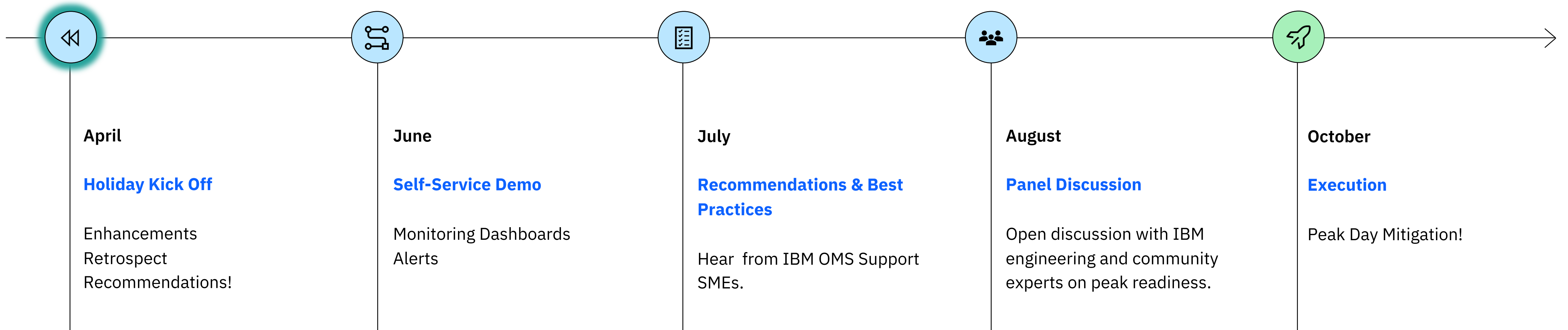


Prescriptive Guidance

Deeper partnership with specific clients in need of direct analysis and prescriptive guidance via Advanced Support and Expertise Connect

Journey to Peak Success

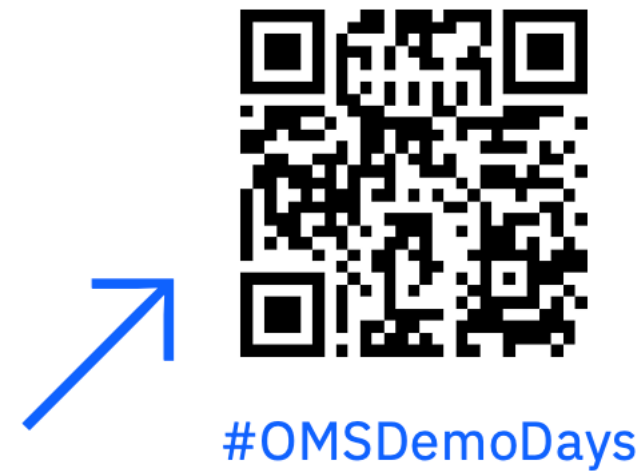
The IBM OMS Support team are continuously expanding our technical best practices based on the observations and learnings over our supported launches and peak events!



What's New?

24.1.3.1-10.0.2403.1

10.0.2403.1 | Operator: V1.0.14



The IBM Order Management

- » [Continuous improvement](#) into our core platform to promote performance, stability, resiliency, self-service, security

OMoC Platform

- IBM® Db2® 11.5.9.0
- WebSphere® Liberty 23.0.0.12
- IBM SDK Java™ 8.0.8.15
- IBM MQ 9.3.0.15
- Struts 2.5.33
- OrderHub Angular v15
- Third-party library upgrades

Certified Containers

- Horizontal Pod Autoscalers (HPAs)
- Deployment scaling policy
 - `apps.oms.ibm.com/autoscaled`
- Enhanced logging for SIP Certified Containers

Self-Service Tool

- Enhanced Monitoring Dashboards
 - Filtering Capability
- Reset Transaction Database Process
- Support for Call Center & Order Hub Deployment

Sterling Intelligent Promising (SIP)

- `suppressZeroQty` for Availability Snapshot
- OMS Integration Adaptor has been enhanced to support
 - Automatic batching & consolidation

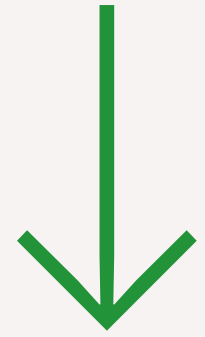
OMS Core

- Promising interrupt properties
- Inventory Purge Enhancement
 - `TableCode`
 - `IsOrgMigratedToIV`

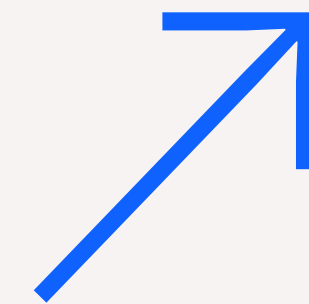
Order Service

- New UE `com.yantra.ycp.japi.ue.YCPGetExternalAuthenticationTokenUE` to handle custom the JSON Web Token (JWT) generation and validation.

Top Issues



can be easily
avoided



Follow the Guide

Case Study

Promising API calls with large number lines

Slowness & Excessive Timeouts for Realtime calls

- Inventory lookup timeouts
- Delay in order processing
- Reprocessing overhead
- Unable to meet NFRs for targeted Order Types (BOPIS, Curb-Side Pickup, etc.)

[Handle orders with large number of order lines with OOB integration adapter →](#)

[Use IBM Order Management for complex sourcing →](#)

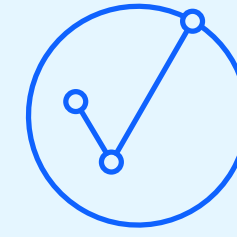


Challenge

- Systemwide slowness; various APIs showing high response time
- Excessive database contention due slow transactions
- High transaction processing backlog (schedule, release, hold resolution, etc.)
- Frequent pod restarts due to Out-Of-Memory event; health-check failures

Root Cause

- `findInventory` transaction resulting in inventory availability call with more than 5000 item-node; saturating JVM resource while processing the transaction
- large number of `YFSPromiseProdLineChoice` objects causing OOM

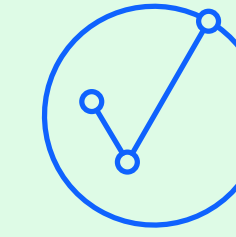


Mitigation

- Restarted JVM to mitigate
- Increased pod Memory/CPU limits

Solution

- Refactored API input to adhere to best practices; use OOB integration adaptor
- Refactored sourcing rule to limit the # of nodes passed in a sourcing sequence.
- Applied solver properties to limit choice generation.
 - `yfs.solver.MaxChoiceFailures`
- Applied solver properties to terminate long running transactions gracefully:
 - `yfs.solver.WarningOrExitOnInterrupt`
 - `yfs.solver.InerruptAfterMinutes`



Recommendation

- Use optimal inventory/promising API; or invoking SIP APIs
- Use OOB integration adapter when calling SIP – IV service; adaptor has been enhanced to handle large lines.
- Configure multiple DG in sourcing sequence with maximum of 25 to 30 ship nodes.
- Performance test expected volume to validate the SaaS extension and configuration
- Implement circuit-breakers to prevent cascading impact.

Case Study

Peak workload testing and API input validation

Slowness & Excessive Timeouts for Realtime calls

- API payload validation
- Consider concurrent Store & Call Center users
- Validate frontend customization for open ended API calls or searches.

[API Performance Best Practices →](#)



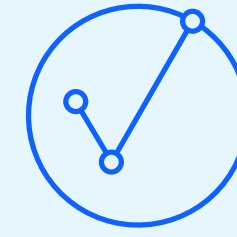
Challenge

- High resource utilization for Application workload and Database
- Higher database response time; slow SQL query execution
- Store, Call Center screens and API calls are slow.

Root Cause

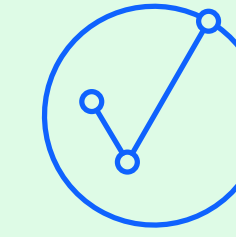
- Use of `getOrderList` and `getShipmentList` API calls with VERBOSE output template and without filterable attributes
- Eg,

```
SELECT YFS_ORDER_HEADER.* FROM YFS_ORDER_HEADER YFS_ORDER_HEADER WHERE ( 1 = 1 ) ) ORDER BY ORDER_HEADER_KEY
```



Solution

- Applied index on custom table.
- Reduced the frequency of calls coming in from Store home screen auto refresh customization.
- Tuned `getOrderList` and `getShipmentList` API input to avoid blank queries.
- Added UI customization input validation to avoid API calls without proper filtering attributes.



Recommendation

- Run the performance tests with the expected peak API workload
- Have a test case closer to the number of Store and Call Centre users during peak and associated user actions
- Look for edge cases; avoid the use of list APIs with VERBOSE output template without proper filtering attribute-value.
- Optimize output template; only retrieve the data needed to be displayed on the UI
- Avoid redundant calls from home screen; auto refresh with heavy API calls to get count.
- Use the SST monitoring dashboards to review resource utilization and performance metrics.

Case Study

Pay attention to costly queries

High Database CPU with cascading impact

- YFS_PERSON_INFO
- YFS_INVENTORY_NODE_CONTROL
- YFS_CALENDAR_SHIFT
- YFS_CALENDAR_SHIFT_EFFPERD
- EXTN_MY_CUSTOM_TABLE

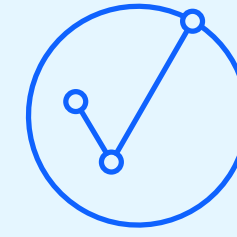


Challenge

- High resource utilization for Application workload and Database
- High Database CPU
- Store, Call Center screens and API calls are slow
- Impact to promising API calls timing out

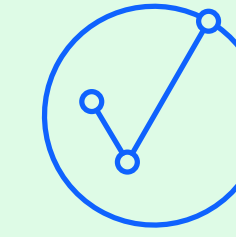
Root Cause

- High CPU caused by YFS_PERSON_INFO query; OOB index may not work for all situations.
- Frequent query execution against YFS_CALENDAR_SHIFT and YFS_CALENDAR_SHIFT_EFFPERD resulting in higher cumulative CPU



Solution

- Cleaned up stale data from YFS_INVENTORY_NODE_CONTROL, YFS_CALENDAR* table
- Applied index on YFS_PERSON_INFO table according to BillTo and ShipTo Address information
- Tuned YFS_Calendar_ShiftDBCacheHome, YFS_Calendar_Shift_Eff_PerdDBCacheHome cache size according to the table size
- Added UI customization input validation to avoid API calls without proper filtering attributes for customer lookup flows



Recommendation

- Review/Apply YFS_PERSON_INFO composite index
- Ensure configuration data reflects production usage
- Purge stale entries from configuration tables. If > 10K, then adjust entity cache accordingly
- Use the SST monitoring dashboards to review slow entities and queries
- Enable entity cache for custom configuration table; implement/enable purge

[Review/Apply YFS_PERSON_INFO index →](#)

[Purge YFS_PERSON_INFO →](#)

[Purge YFS_INVENTORY_NODE_CONTROL →](#)

On Premise

Payment Integration

– Prior Webcast:



[Payments Deep Dive session](#)

- Details on new serviceability enhancements and walk-through of the new **Payment Audit** feature.
- Example happy path scenario with the **Dynamic Charge Transaction Request Distribution** feature.
- Recommendations, best practices, common payment configurations, Do's & Don'ts based on lessons learned from common issues seen during prior peak seasons.

Automatic Hold

– Implement the Automatic order hold so that if there is a looping condition detected due to payment mismatch, order can be put on hold via change order. [Read more →](#)

- `yfs.payment.infiniteLoop.paymentHoldType`
- `yfs.payment.infiniteLoop.allowViewingOfOrder`

Excessive Charge Transaction Records

- Review orders having many charge transaction records.
- Having excessive YCT records shows underlying issue.
- Place orders having excessive YCT on hold, to prevent further processing.
- `SELECT ORDER_HEADER_KEY, COUNT(*) FROM OMDB.YFS_CHARGE_TRANSACTION GROUP BY ORDER_HEADER_KEY HAVING COUNT(*) > 100 ORDER BY ORDER_HEADER_KEY DESC WITH UR;`

APIs

- Review *javadocs* before implementing `processOrderPayments`, and use `RequestCollection`, `ExecuteCollection`, `RequestCollection`.
- Do not call `processOrderPayments` as part of long transaction boundary.
- This API is intended for In-person scenarios e.g., carry lines.

Monitor Backlog

- Query `YFS_ORDER_HEADER` table to get payment collection backlog, refer to `getJobs` query.
- Query `YFS_CHARGE_TRANSACTION` table to get payment execution backlog, refer to `getJobs` query
- Queries indicate how many orders are eligible to be picked and processed by the agents.
- Redundant processing of problematic orders can lead to bottlenecks.

UserExit

- Tax related UE output should include all necessary taxes to avoid wiping out previous existing taxes.
- Correct authorization IDs should be stamped along with corresponding expiration dates.
- Handle all the exceptions from the collection UE. Otherwise, charge and authorization transactions will get stuck in the 'invoked' user exit status.

Payment Collection Failure

Ensure the following parameter is set to ensure `PAYMENT_COLLECTION` agent does not fail with `java.lang.IllegalArgumentException`:
Comparison method violates its general contract!

[Read more →](#)

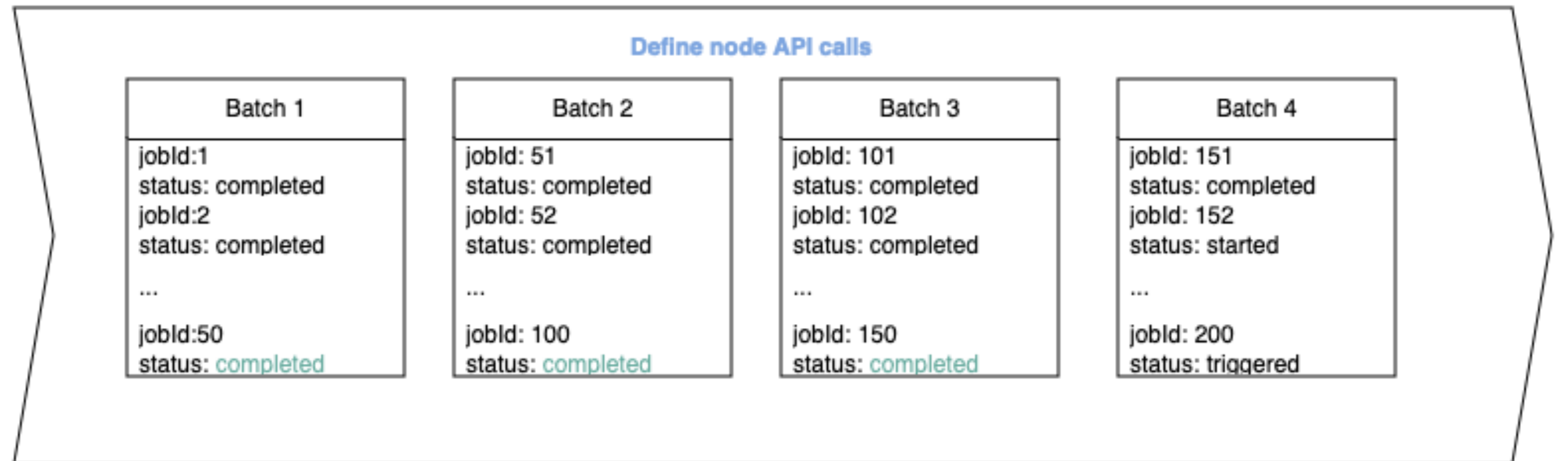
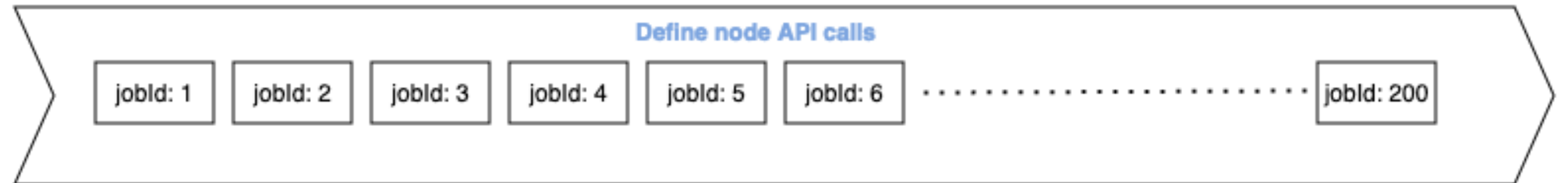
Sterling Intelligent Promising - Inventory Service

Massive bulk node updates and/or more frequent node updates and changes in DGs definition can lead to [degraded response](#)



[Update nodes in smaller batches](#); whether its planed or unplanned **ALWAYS** perform update actions in batches rather than updating all nodes at once.

1



Update nodes in batches

- Use **jobid** to track the status of the last in a batch node update.
- Once this job completes, update the next batch of nodes.

Sterling Intelligent Promising - Inventory Service

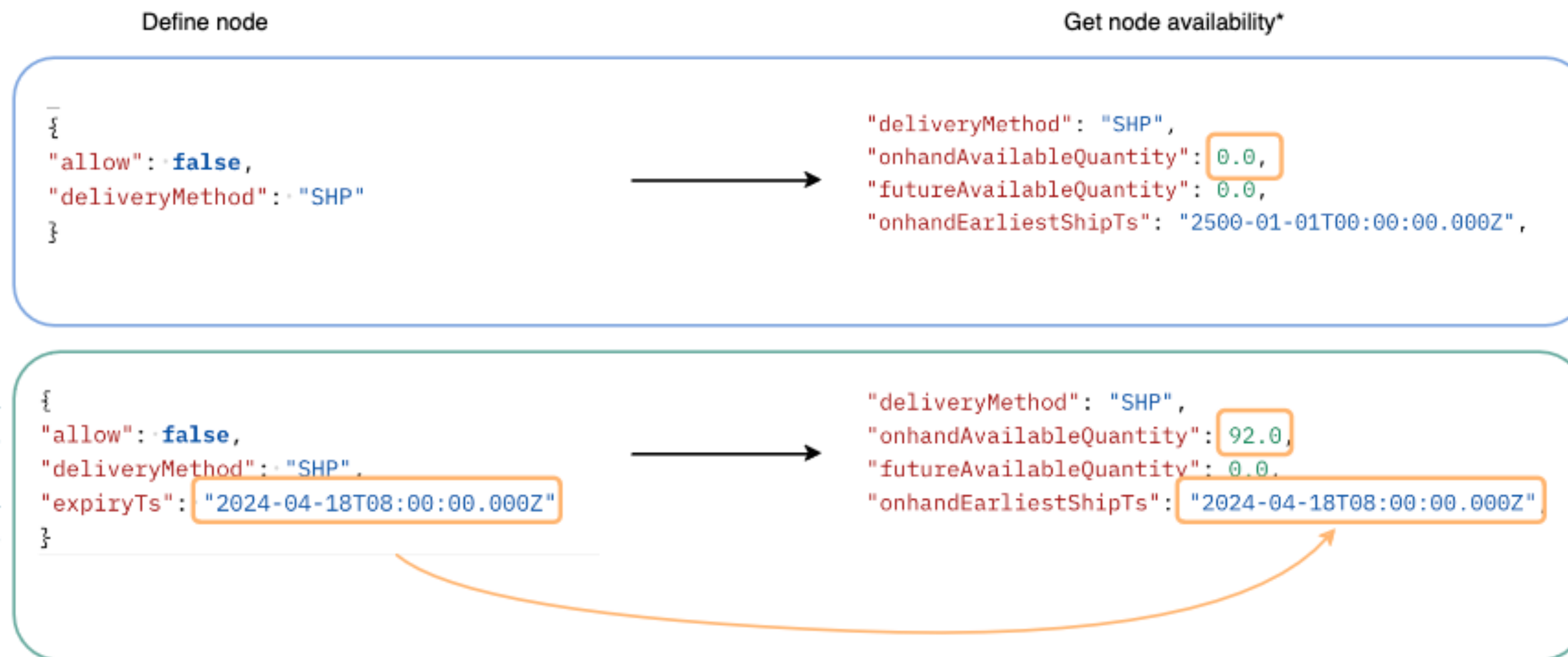
Massive bulk node updates and/or more frequent node updates and changes in DGs definition can lead to degraded response



Use expiryTs when turning nodes off; Turning nodes off using expiryTs attribute is less expensive than turning nodes off and back on.

If a fulfillment override is set to expire, IV does not update the availability of all items at a node to 0, instead availability remains as it is; however, onhandEarliestShipTs will equal to the expiryTs for frontend.

2



Availability APIs output with and without expiryTs use for node fulfillment overrides

- Use expiryTs when turning nodes off*
- expiryTs → onhandEarliestShipTs

Note: Events are not generated upon reaching expiryTs; refactor frontend logic to consider onhandEarliestShipTs

Best Practices

Sterling Intelligent Promising - Inventory Service

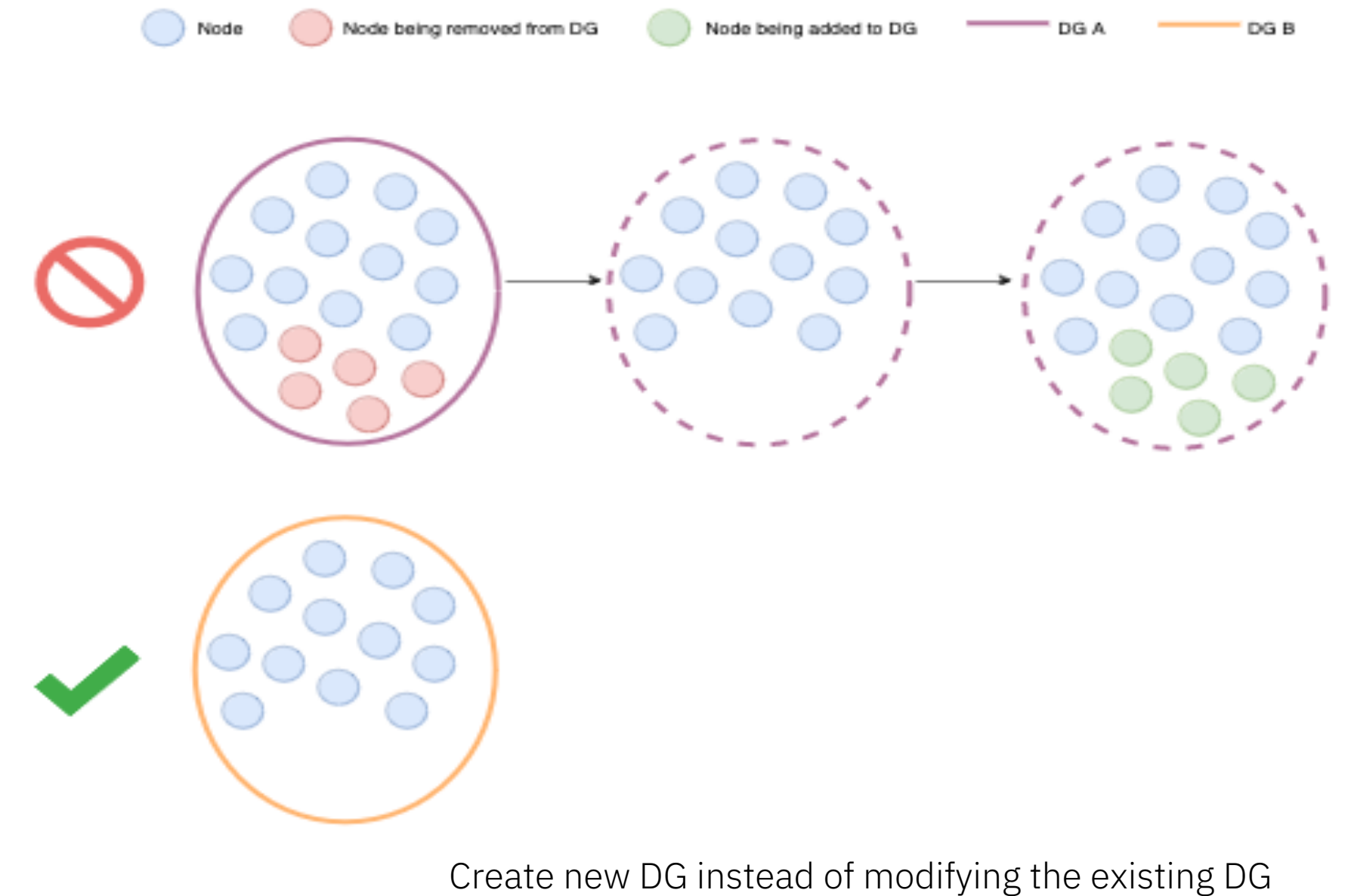
Massive bulk node updates and/or more frequent node updates and changes in DGs definition can lead to **degraded response**



When updating DGs, update each DG **only once**; avoid redundant availability recomputes by combining updates to the DG.

Create **New DG**, if there are major changes.

3



Common JMS Errors

These common error may surprise you during high volume:

- MQRC_BACKED_OUT

- MQRC_RESOURCE_PROBLEM

- Address excessively long running transactions
- High number of uncommitted transactions due to incorrect batch size, ack settings.

- Confluent > Kafka Connector > IBM MQ Connection

- » `batch.size`
- » `max.pending.messages`
- » `receiver.threads`

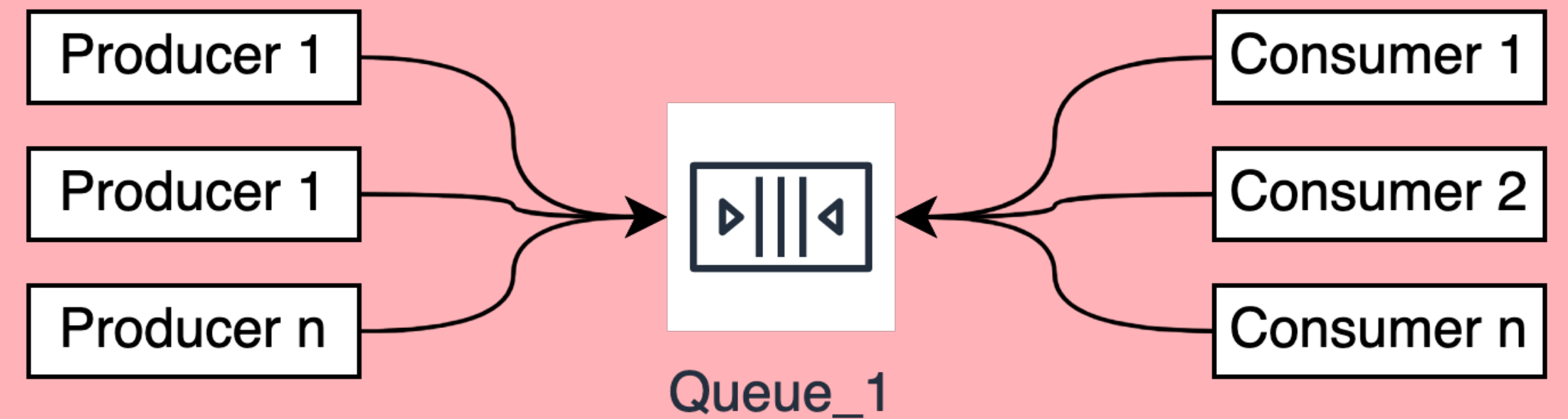
- JMS Performance properties

- » `yfs.yfs.jms.session.disable.pooling=N`
- » `yfs.jms.sender.anonymous.reuse=true`
- » `yfs.yfs.jms.sender.multiThreaded=true`
- » `yfs.agent.bulk.sender.enabled=Y`
- » `yfs.agent.bulk.sender.batch.size=5000` (increase as needed)

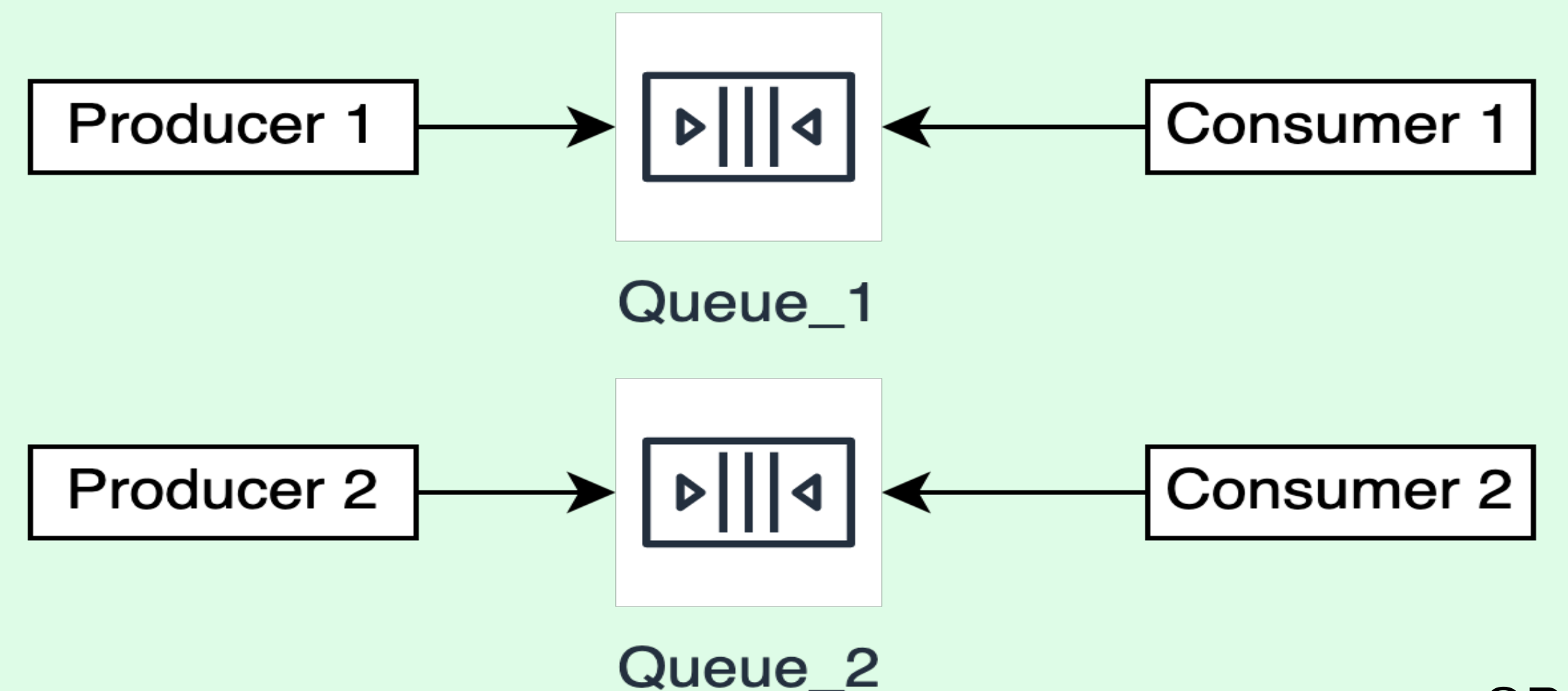
- MQRC_CONNECTION_BROKEN

- Abnormal (idle) connection termination
- Enable Retries (3x, 100ms)

Having single queue for all integration can result in [severe bottleneck](#).



[Seamlessly scale](#) with dedicated queues for each integration



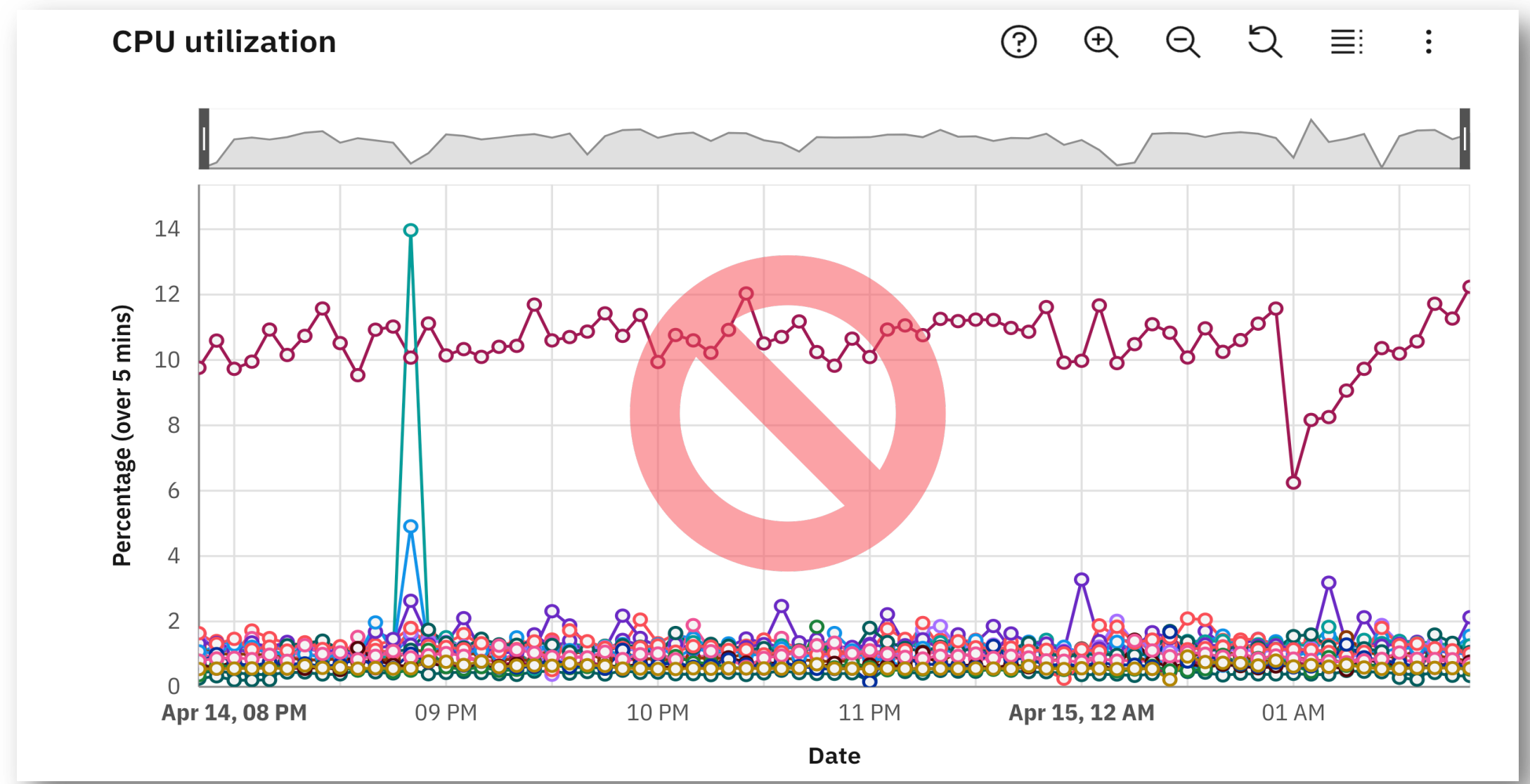
Effective Server Consolidation

CPU and Memory usage for optimal consolidation.

CPU allocation 90.00%



Memory allocation 69.00%



1

Reduce # of Instances according to the workload; scale up using SST Server scheduling feature

Stop the servers that are not required or are **obsolete**.

Stop any failed servers; potentially misconfigured

2

Use Self-Service **server scheduling** feature

Schedule the processes such that it uses the CPU and memory resources optimally.

Choose **correct performance profile**; start with Balanced

3

Consolidate by **grouping** the server based on the **functionality**

Consolidate based on the **scaling requirement**

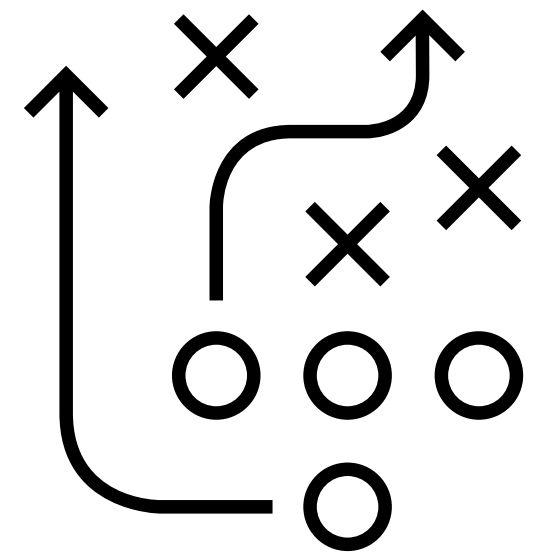
Consolidate based on the **workload pattern**;

4

Arbitrary groups with no more than 5 integration service or agent criteria; priority groups

Best Practices

IBM OMS & SIP Certified Containers



Key Resource:

- [OMS Operator Best Practices Blog →](#)
- [SIP Operator Best Practices Blog →](#)
- [OIDC Support Features →](#)
- [Development Toolkit →](#)
- [OMS Operator Best Practices Doc →](#)
- [Release Schedule →](#)
- [Upgrade →](#)
- [Customization →](#)

Helm → Operator

Migrate to Operator as [Helm has been deprecated](#) as of June 2022.

- Operators are essential for deploying new features and enhancements.
- Comprehensive migrations steps are available [here →](#)

Align Operator and Product Version

Always update both operator and product version.

- Avoid unforeseen compatibility issues; see [Clarity Report →](#)
- Follow vendor documentation to install prerequisites such as standalone Kafka, Cassandra, Elasticsearch, etc.
- [Support policy and delivery mode →](#)

Download & Upgrade to Latest DTK

The Self-Service Tool (SST) updates the [DTK](#) monthly

- Use integrated Development Toolkit (DTK)
 - Reconcile customization
- [Get Onboarded to SST →](#)

Operator CRD and Annotation Management

Keep the Operator CRD and annotations updated to reflect any new additions or changes.

- Regularly update the Operator Custom Resource Definitions (CRDs), specifically [OMEnvironment OR SIPEnvironment YAML](#), to match the latest CRD structure.
- Monitor and manage annotations carefully to ensure operational integrity and performance. [Read more →](#)

Use Persistent Volume

Safeguard from data loss

Implement daily synchronizations of [Persistent Volumes \(PV\)](#) to ensure data protection and retention.

Handle updates seamlessly

Maintains synchronization between Elasticsearch and Cassandra; ensures consistency across databases, preventing discrepancies.

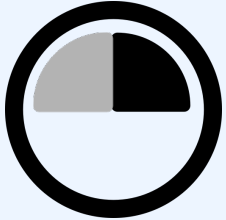
Operator Upgrade Strategy

Automatic vs. Manual Subscription

- [Automatic](#) subscription; good for [lower environments](#).
- [Manual](#) provide more control; suitable for [production](#)
- [Subscription →](#)

How to Succeed

Plan



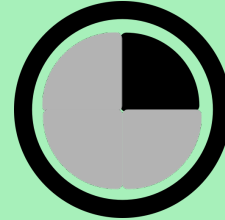
- ✓ Retrospective
- ✓ Latest product levels
- ✓ Detailed projections
- ✓ Catch prior webcasts
- ✓ Engage help as needed

Prepare



- ✓ Align to IBM schedule
- ✓ Representative testing
- ✓ Proactive housekeeping
- ✓ Clean up the noise
- ✓ Track risks

Execute



- ✓ Clear runbooks, RACI
- ✓ Quickly detect issues
- ✓ Throttle as necessary
- ✓ Quick mitigation

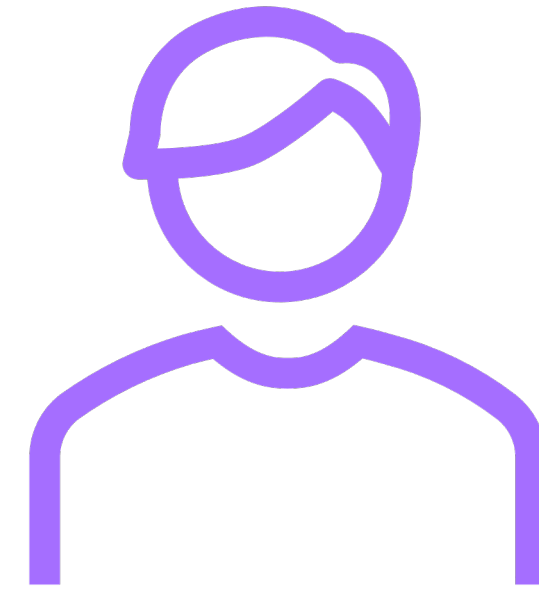


IBM Support Offering – Advanced Support

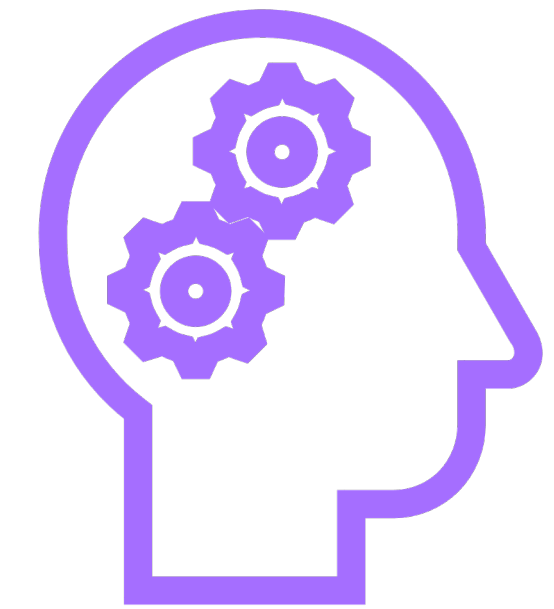
*NEW in
2Q23!*

An enhanced support experience on top of your active IBM support subscription, providing prioritized case handling and shorter response time objectives

www.ibm.com/support/pages/ibm-advanced-support-offering



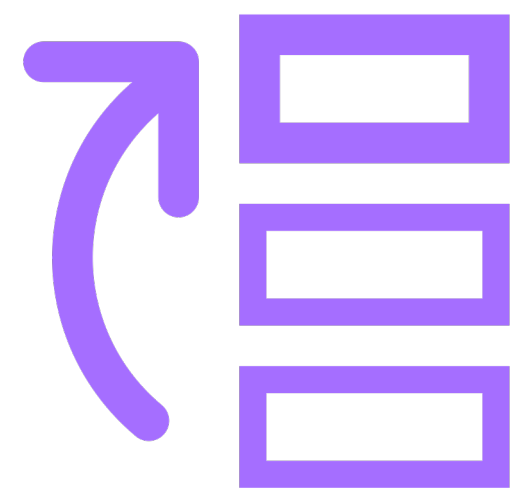
Named IBM Advanced Support Focal (ASF)



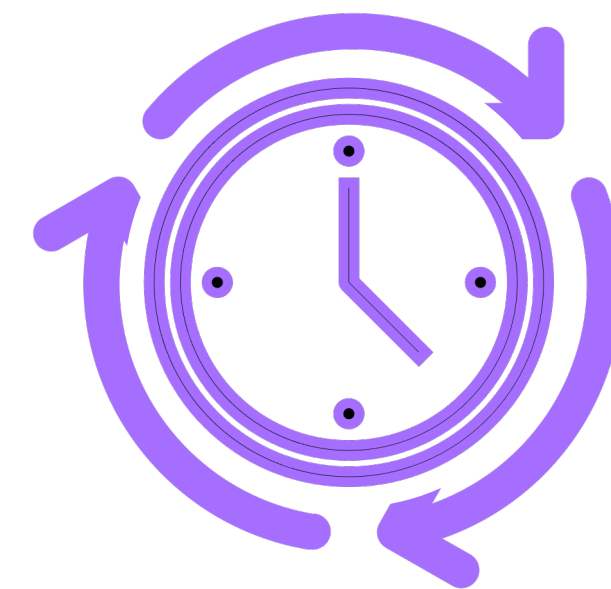
Priority access to Senior Technical Support Squad



Enhanced initial, ongoing response SLOs



Higher ongoing case prioritization

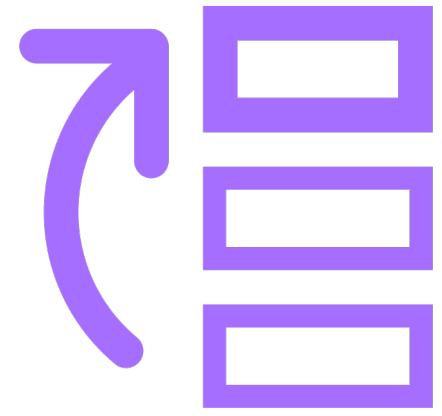


24x7 coverage for priority Sev-2



Manage, escalate backlog via cadence, reporting

Next Steps



IBM Advanced Support

www.ibm.com/support/pages/ibm-advanced-support-offering

Contact your IBM Client Success Manager, Account representative, or Mike Callaghan(mcallagh@ca.ibm.com)



Sterling OMS Support 101

www.ibm.com/community/101/sterling/oms/



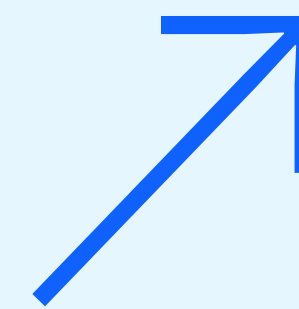
Technical Best Practices

Start with the new [Performance Guide](#)



Are you ready?

Technical Best Practices



Follow the Guide

Thank you

This content was provided for informational purposes only. The opinions and insights discussed are those of the presenter and guests and do not necessarily represent those of the IBM Corporation.

Nothing contained in these materials, or the products discussed is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers, or altering the terms and conditions of any agreement you have with IBM.

The information presented is not intended to imply that any actions taken by you will result in any specific result or benefit and should not be relied on in making a purchasing decision. IBM does not warrant that any systems, products or services are immune from, or will make your enterprise immune from, the malicious or illegal contact of any party.

All product plans, directions and intent are subject to change or withdrawal without notice. References to IBM products, programs or services do not imply that they will be available in all countries in which IBM operates. IBM, the IBM logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or services names may be trademarks or services marks of others.

© 2024 International Business Machines Corporation
ibm.com/trademark.