Application Management Webinar

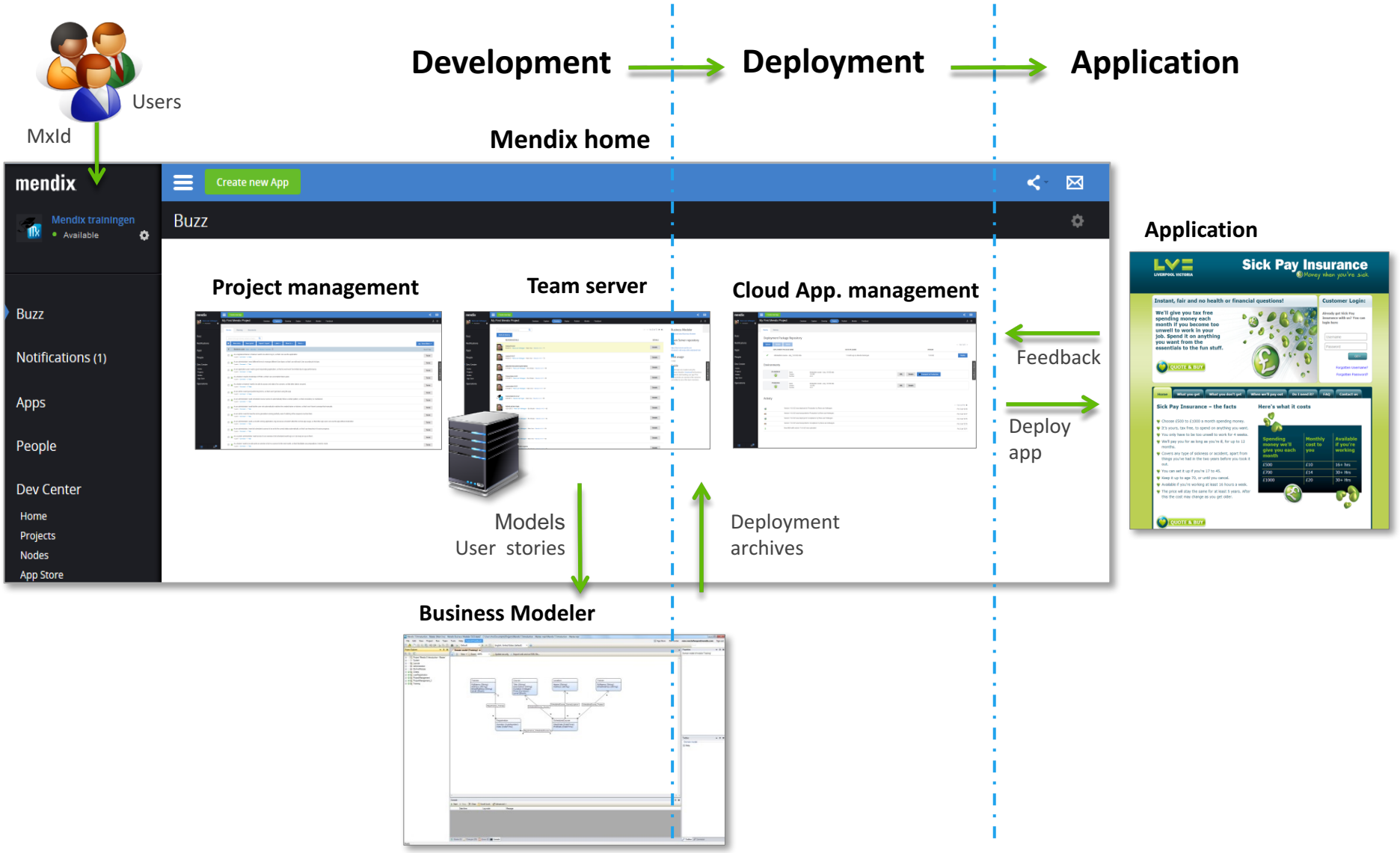
Daniela Field

Agenda

- Agile Deployment
- Project vs Node Security
- Deployment
- Cloud Administration
- Monitoring
- Logging
- Alerting

Cloud Overview

Cloud Overview – Project Management



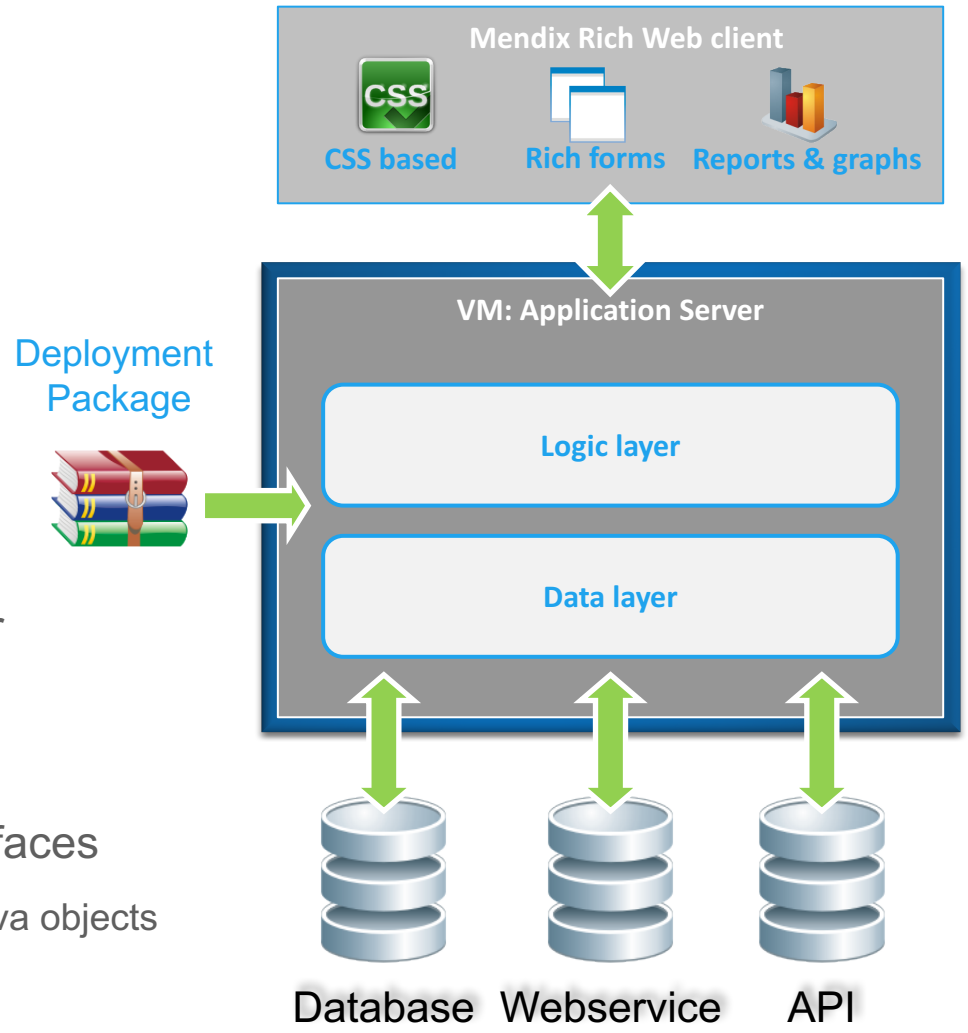
Cloud Overview – Inside the app environment

Web client (user interface)

- Runs in browser (AJAX)
- CSS based
- Plug-in structure

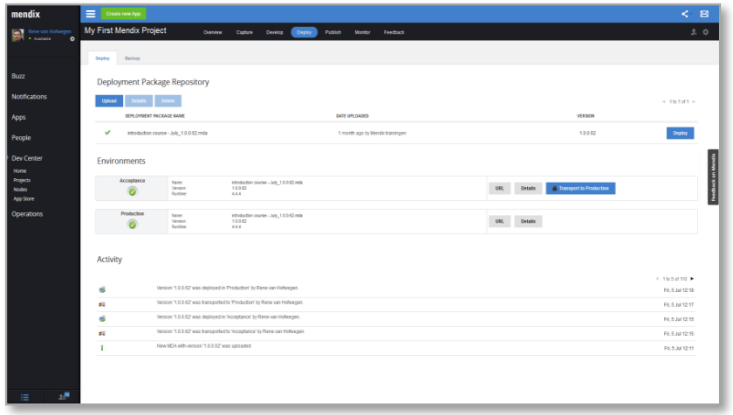
Application server

- Executes model
- Separation between logic & data layer
- Transforms data to domain model (Mx objects) from different sources
- Exposes logic & data to different interfaces
  - HTTP, web services, Java; JSON, XML, Java objects



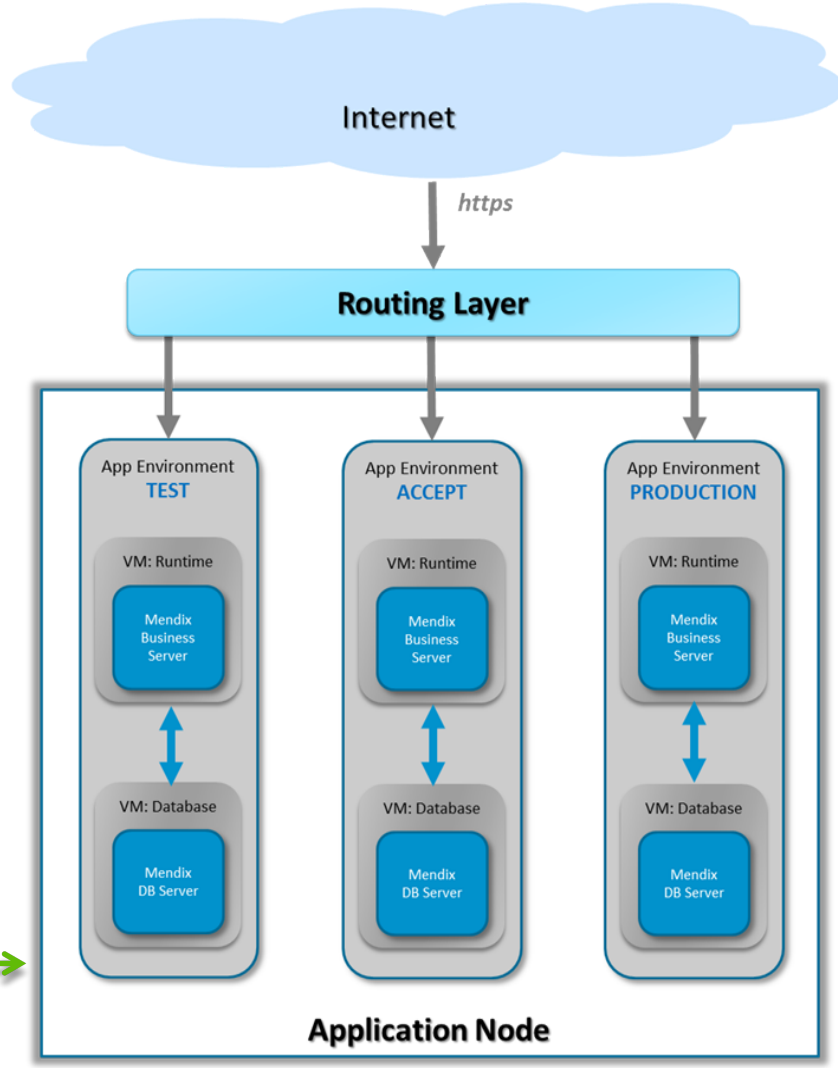
Cloud Overview – Cloud Setup

Mendix cloud Management



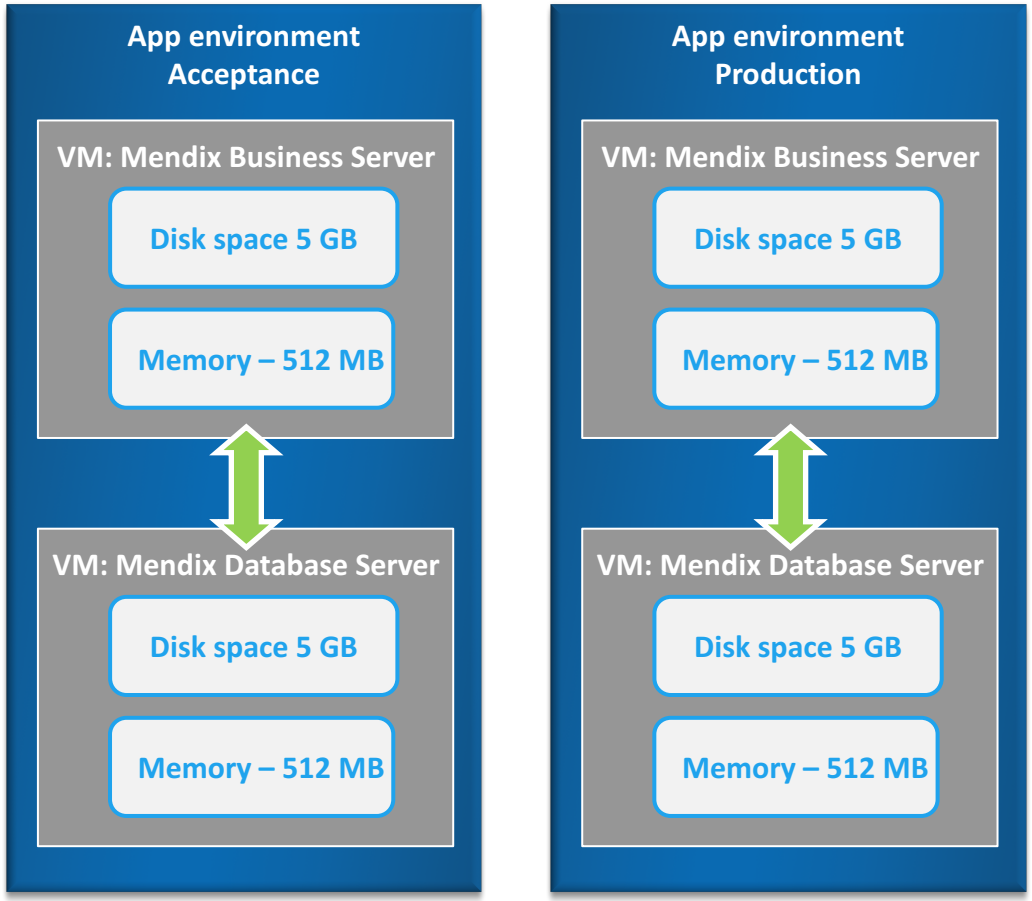
Management (start, stop, backup – restore)
Monitoring & initialization

Mendix Node Controller



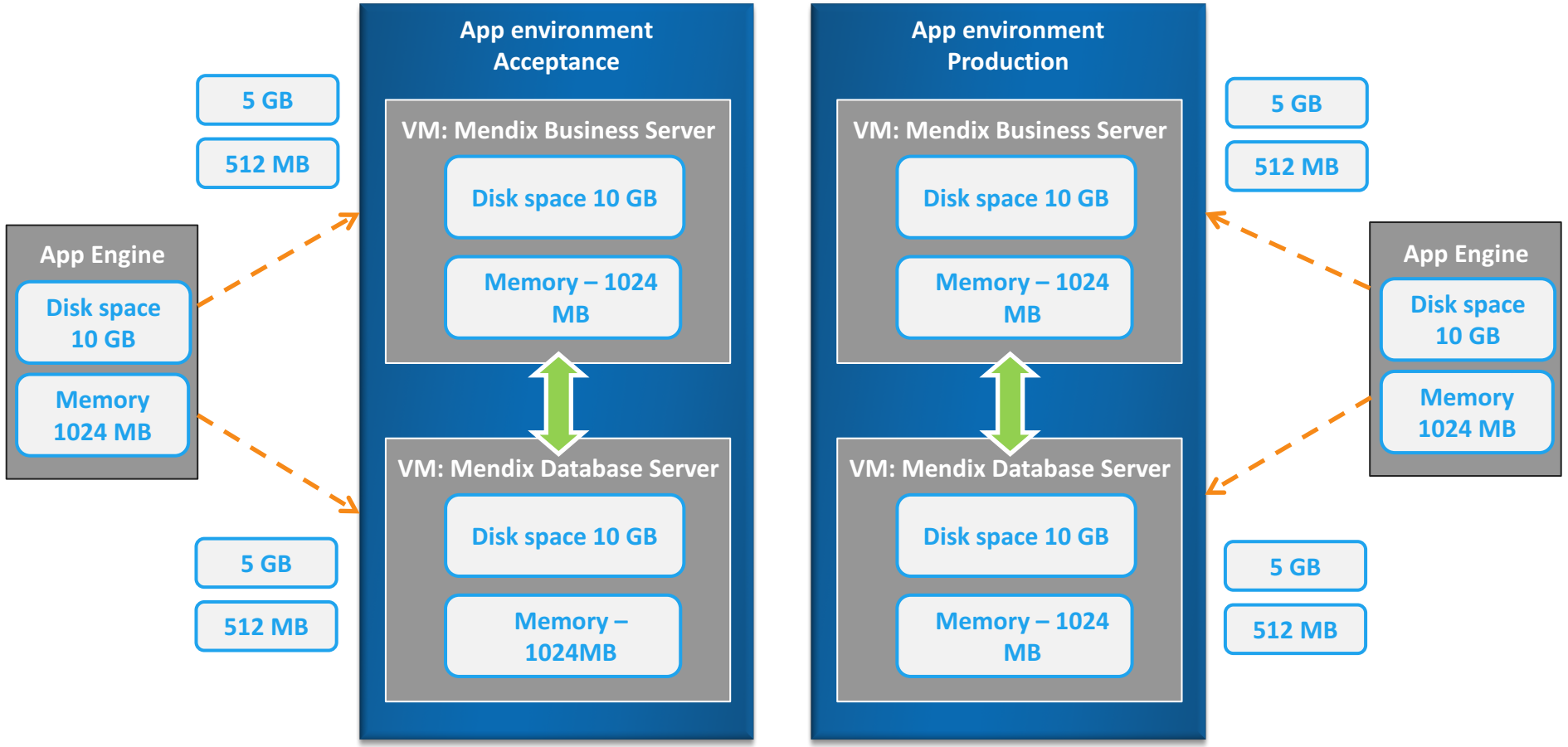
Cloud Overview – Environment sizes

Default Application Environments



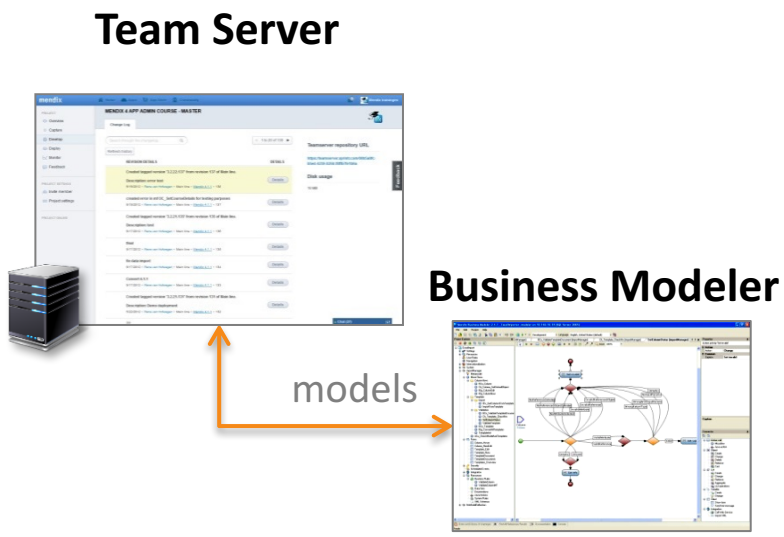
Cloud Overview – App Engines

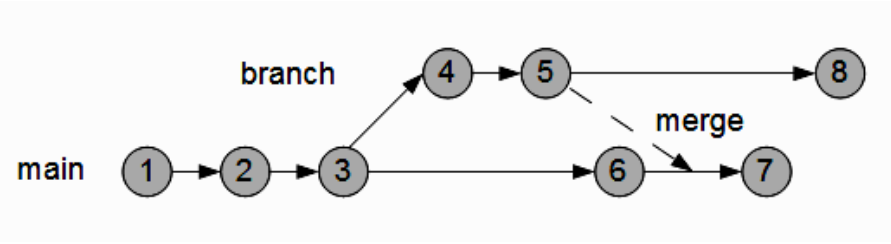
Adding an app engine, default sizing



Cloud Deployment – Versioning

- Revisions
  - Commit
  - Numbering
- Branches
  - Copy
  - Branch from Main line
    - Great feature
    - Integration external work
  - Branch from Tagged version
    - Fix in earlier release
- Merging



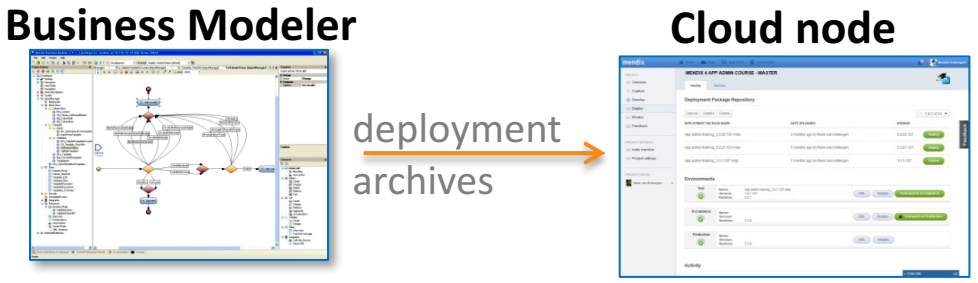


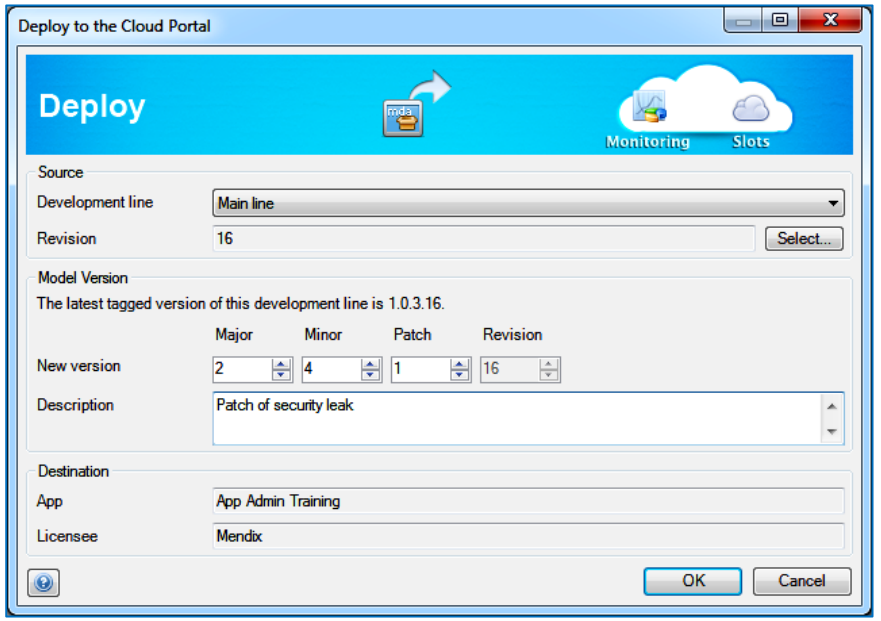
Cloud Deployment – Versioning

- Commit often
  - Conflict reduction/prevention
  - Insight of completed work
  - Determine correct version
  - Never commit errors
- Update Often
  - Conflict reduction/prevention
- Merge often
  - Direct after fix
  - After feature is completed
  - Changes are still known
  - Conflicts are limited
- Creation of branches
  - Fix in earlier release
  - (Big) feature (work > 1 day)
  - Integration external work

Cloud Deployment – Release types

- Tagged Version
- Release types
  - Patch
  - Minor
  - Major
- Version numbering
  - Major.Minor.Patch.Revision





Cloud Deployment – Release Management

Best practices for Release procedures

- Patch *-*-1
  - Urgent & normal fixes
  - No impact on domain model /data structure
  - No impact on existing situation/no scripting (conversion) needed
- Minor *-1-*
  - No or small impact on domain model /data structure
  - Small impact on existing situation
  - App admin should be informed about conversion scripts
- Major 1-*-*
  - Might have impact on domain model /data structure
  - Prepare conversion if needed
  - All possible scenarios, including domain model changes
  - App admin should be informed about conversion scripts

Cloud Deployment – Release Management

Release procedures – Why do we need them

- Database changes
- Changes in the startup Microflow
- New scheduled event
- Broken core functionality

Cloud Deployment – Start-up failure

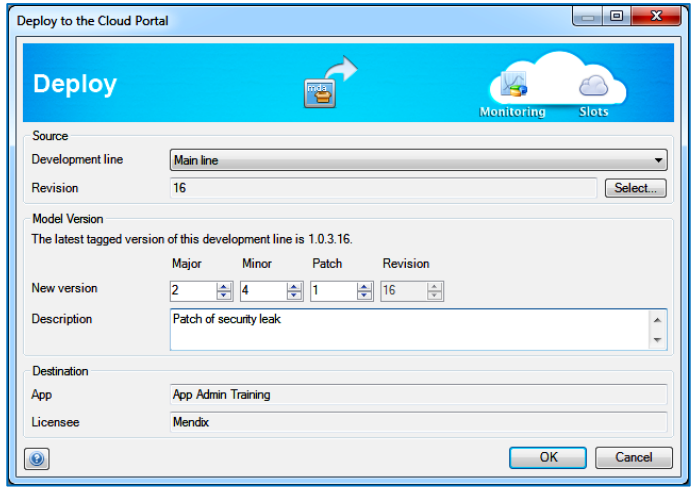
Start-up failure – Most common errors

- Startup Microflow causes an error
  - Manual start
  - Automatic restart
- Database synchronization errors
  - Data types are not compatible
  - Changes in the domain model too complicated (very rare)

Cloud Deployment – MDA

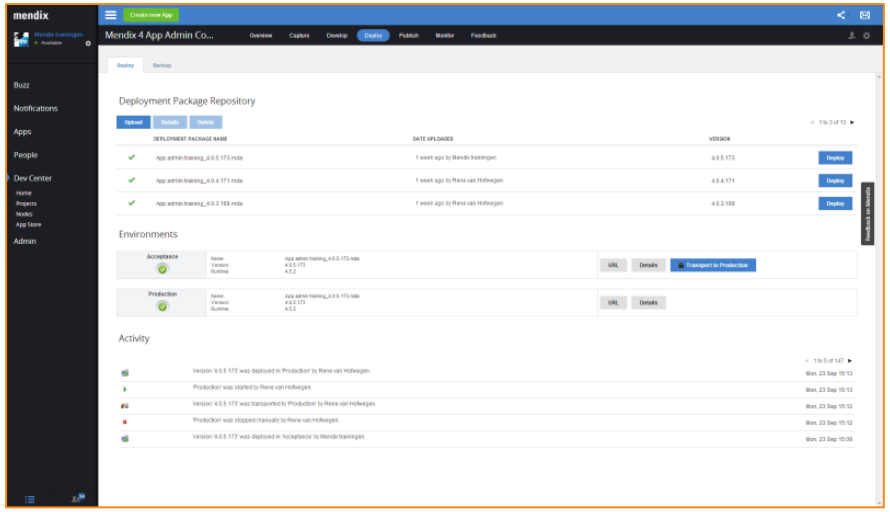
The modeler initiates the creation of the deployment archive (MDA)

- Based upon revision of the team server
- Adds a tag to the revision with version number
- The MDA file contains
  - The model
  - Java actions & additional libraries
  - Theme
  - Custom widgets
  - Meta information
    - App ID, App version, Mx version
- MDA creation
  - Mx 4 & earlier: MDA is created by the modeler
  - Mx 5 & up: MDA is created by the online build server
- The MDA is transported to the MDA Repository of the assigned cloud node

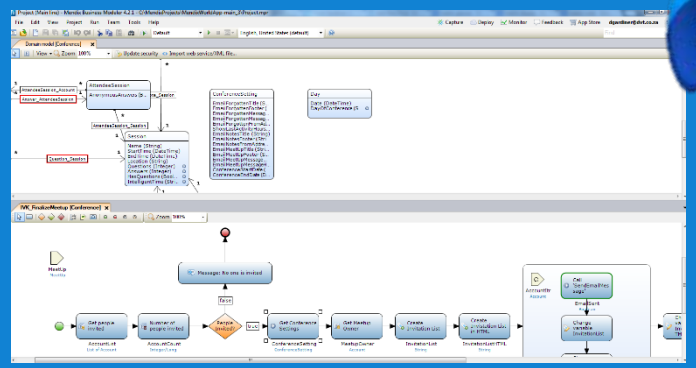


Cloud Deployment – DTAP & transports

- **DTAP** stands for
  - **D**evelopment => laptop / local => developer
  - **T**est (optional) => test environment => dev. team
  - **A**cceptance => test environment => customer
  - **P**roduction => the actual application => end-user
- Environments are identical
  - (Test,) Acceptance & Production



Cloud Deployment – Demo



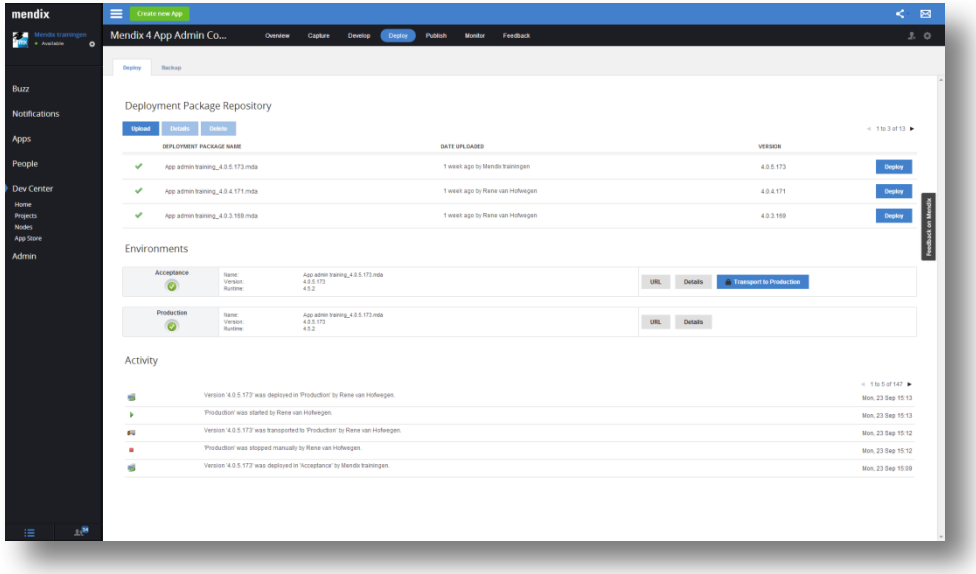
mx mendix

Administration

Mendix Cloud

Cloud Administration – Management

- Dashboard for instant overview
- Configuration of cloud nodes
- (T)AP management
- Transport & deployment of applications
- Security & auditing
- Backups
- Monitoring
- Alerting & logging



Configuration

Mendix Cloud

Cloud Configuration – Mendix Cloud Node

Cloud node

- Initiation by Mendix
- Based on license / order
- Mode (test - production) = app environment
- Resources (RAM, HD)
- License keys

Environment Settings

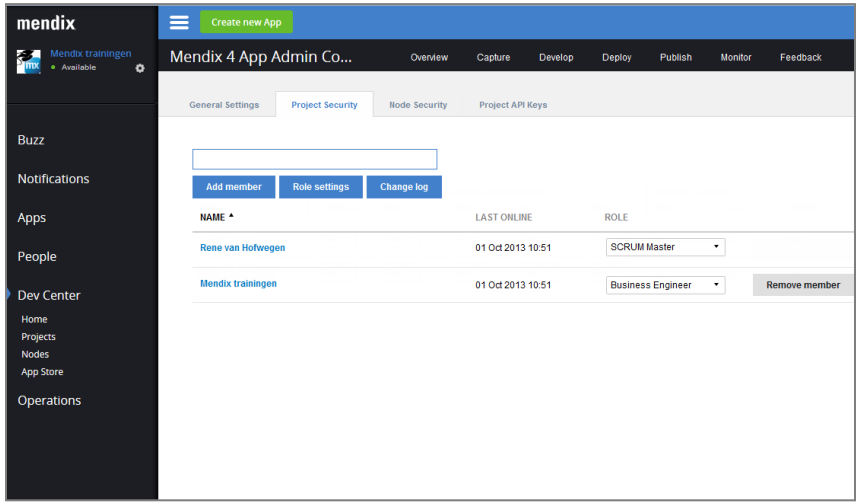
- Administrator password
- Model Options
- Network
- LogLevels
- Advanced

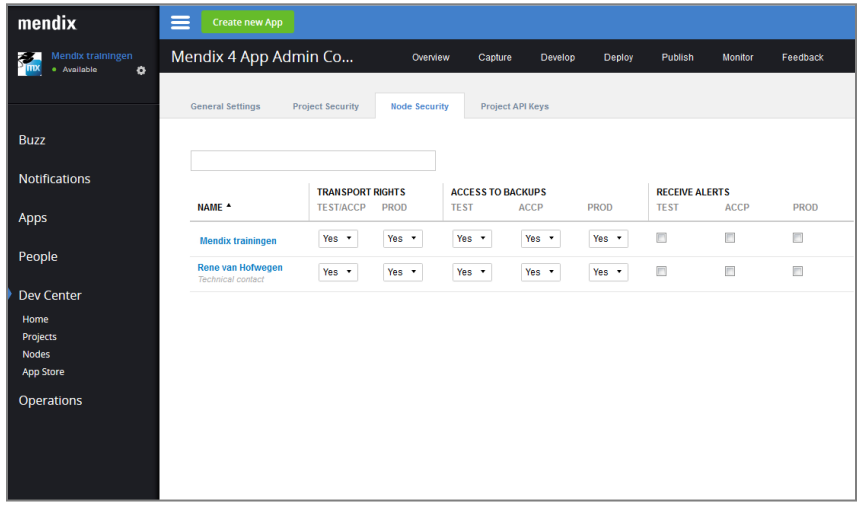
Security

Mendix Cloud

Cloud Security – Project vs Node

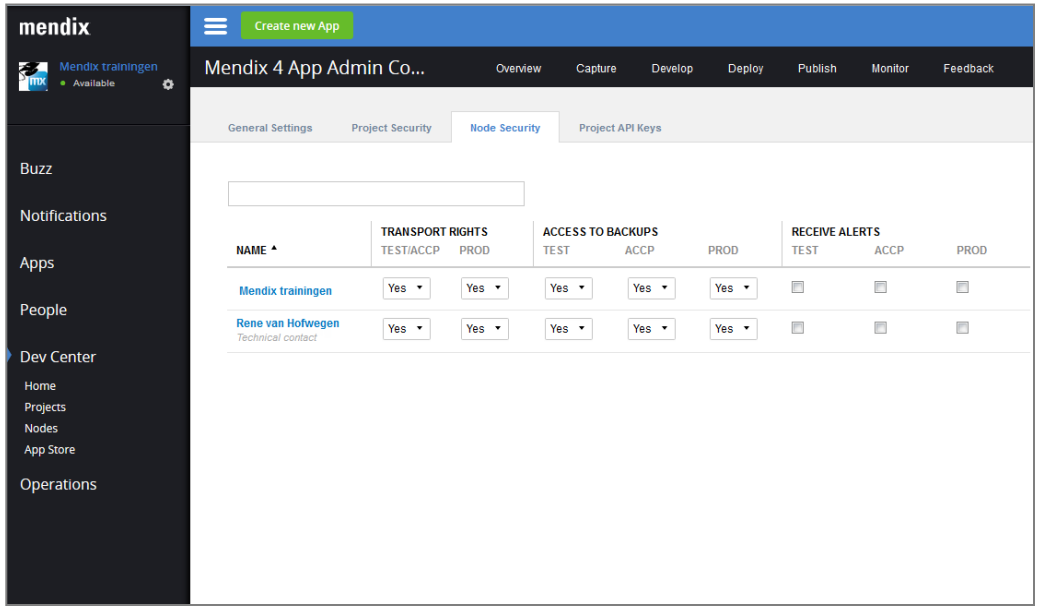
- Project security != Node security
  - Project security is project related access
    - Stories, documents, team server, feedback
  - Node security is cloud related access
    - Deployment, monitoring, back up





Cloud Security – App specific

- Each App has a technical contact
  - Specified in the (sales) contract with Mendix
  - Is the person to select the Mendix project
  - Is the only person who can set Node privileges
  - Cannot be deleted



Cloud Security – Logging & auditing

- Logging activities
  - Cloud app actions
  - Logged activities
    - Uploading MDA
    - Transportations
    - Starts & stops of applications
    - Creating & restoring backups
    - Changes in team members
  - For auditing purposes
  - Stored for 1 year

Cloud Security – Server

Java containment - custom Java actions

- No IO access , exceptions:
  - Provided by the MBS a.k.a. the Core.API
  - Write actions in temp directory permission
  - Read actions in temp & resources directory
- No system properties, etc

Production mode

- Strong administrator password is required
- Blocks accounts for 10 min. after 3 failed logins
- Logs all failed logins
- Production security must be set in the project
- Updates require
  - An additional authentication
  - Approval of customer for the acceptance environment

Use the emulate cloud security option

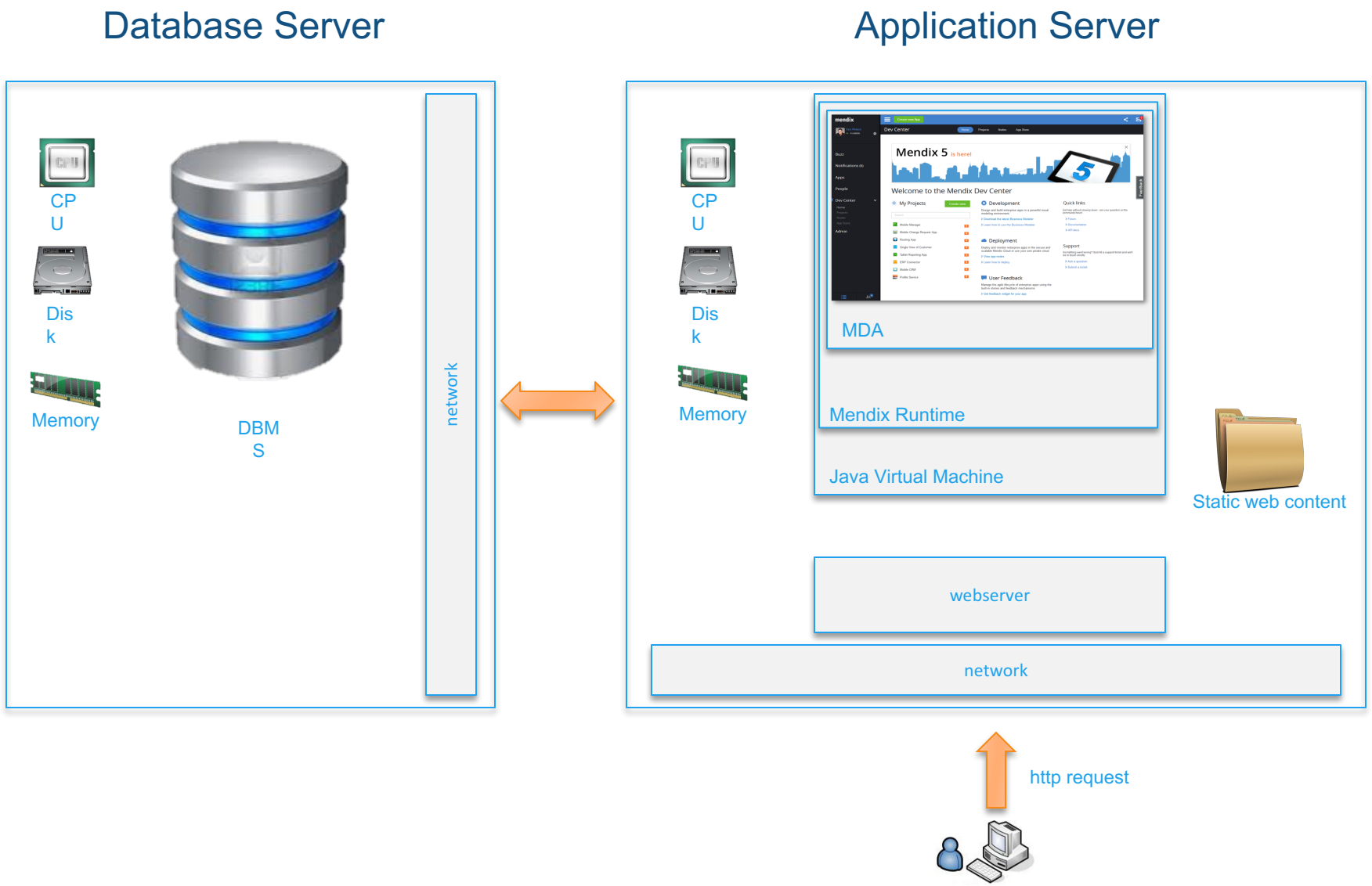
Monitoring

Mendix Cloud

Cloud Monitoring – Recommendations

- Monitoring per environment(server) of:
  - Disk
  - Memory
  - CPU
  - Webserver
  - Runtime
  - Database
- Alerting mechanism when
  - Critical errors occur
  - Failures might arise
- Storage of log
  - 2MB per file
  - Last 10 files

Cloud Monitoring – Overview



Cloud Monitoring – Statistics

- Mendix Runtime
  - Amount of handled external requests
  - Object Cache
  - User accounts and login sessions
  - JVM Object Heap
  - JVM Process Memory Usage
  - Threadpool for handling external requests
  - Total amount of threads in the JVM process
- Database Statistics
  - Database transactions and mutations
  - Database table vs. index size
- Application Node Statistics
  - Application node CPU usage
  - Application node disk IO/s
  - **Application node load**
  - **Application node operating system memory**
  - **Application node disk latency**

Cloud Monitoring – Statistics

- Application Node Statistics
  - Application node disk throughput
  - Application node disk usage (in bytes)
  - Application node disk usage in %
  - Application node disk utilization
- Database Node Statistics
  - Database node CPU usage
  - Database node disk IO/s
  - Database node load
  - Database node operating system memory
  - Database node disk latency
  - Database node disk throughput

Cloud Monitoring – Statistics

- Database Node Statistics
  - Database node disk throughput
  - Database node disk usage in %
  - Database node disk utilization
  - Amount of database connections

Backup management

Mendix Cloud

Cloud Backup – Automatic backup

- Only on production environments
- Every night (between 0:00 – 6:00)
- Full copy
  - Database
  - Files
- Meta data (stored in the database)
- Stored for
  - Younger than 2 weeks: all
  - Younger than 1 month: On the 1st, 8th,15th,22nd
  - Younger than 1 year: 1st of every month
  - Older than 1 year: 1st of January

Cloud Backup – Manual backup

- Available for every environment
- Thru cloud app management
- Application continues to run
- Full copy
  - Database
  - Files
- Meta data stored in the database
- Stored indefinitely
- Three backups per project!!

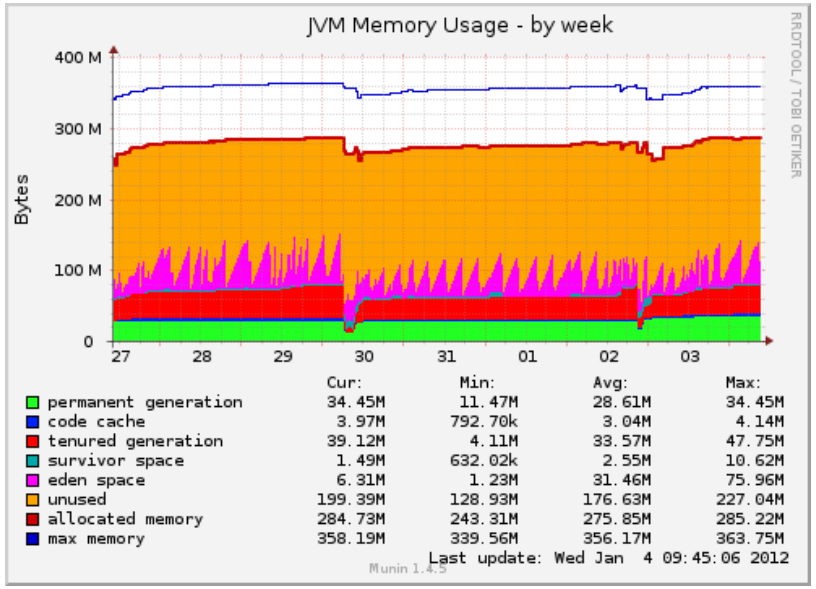
Cloud Backup – Restoring backup

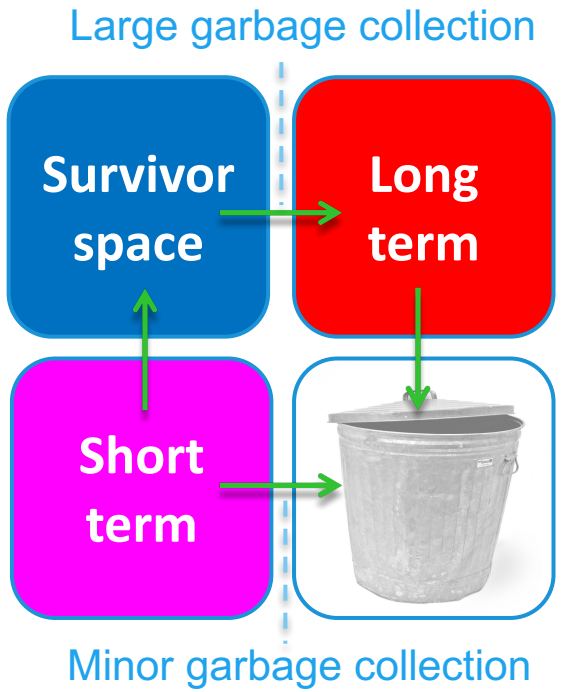
- Available for every environment
- Thru cloud app management
- Application has to be stopped
- Restore of:
  - Database backup
  - Files backup
  - Related MDA file

Analyzing Graphs

Cloud Monitoring – JVM Memory

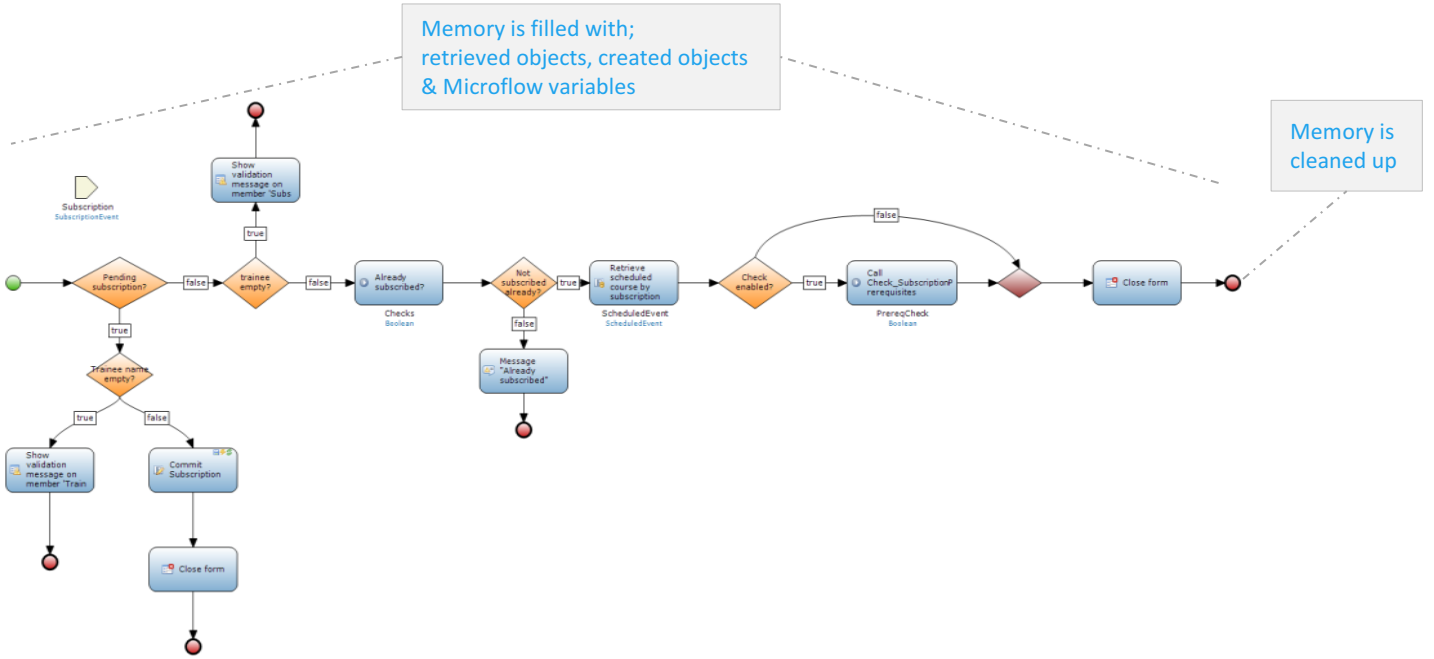
- Memory of Mendix Business server
- How does the Java heap space work?
- Thresholds (simplified)
  - 33% short term memory cleanup (minor GC)
  - 60% long term memory cleanup (major GC)





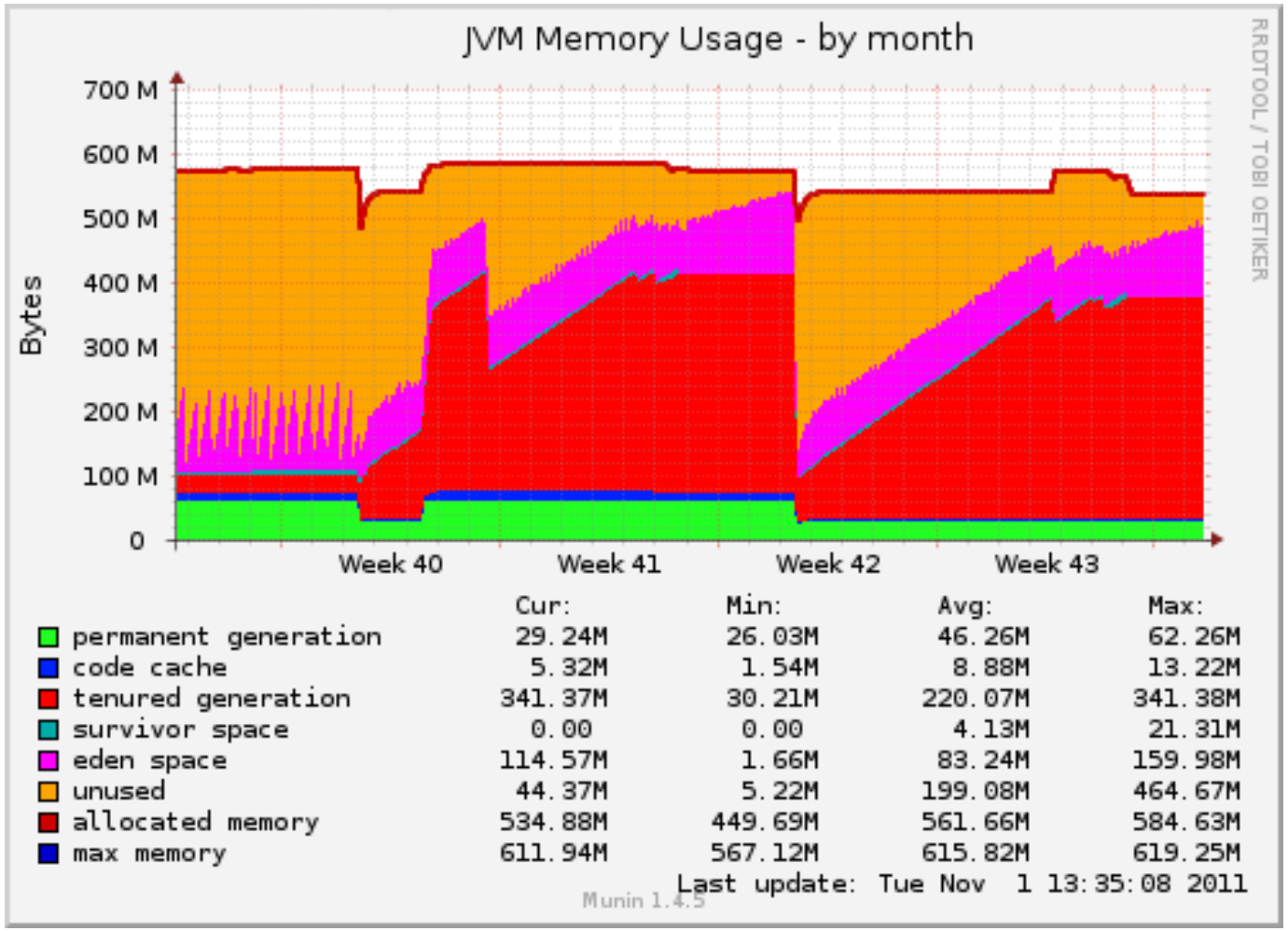
Cloud Monitoring – Memory Management

- Microflows will clean up after execution
- Used memory will be released for GC
- No memory leak occurs
- Custom Java (libraries) won't do this by default!



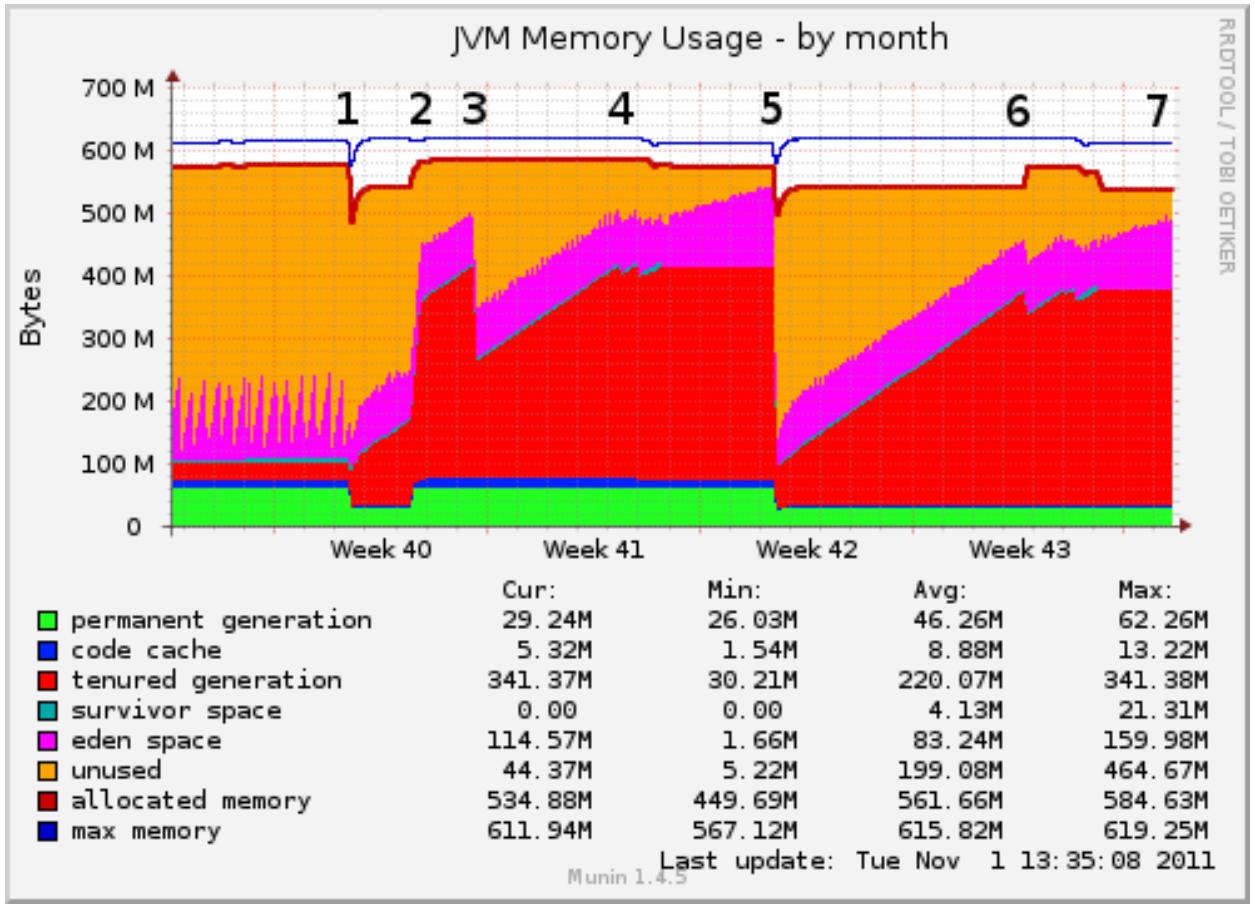
Analyzing graphs – Overview

- Analyzing = trend searching = looking for deviations
- Number of users should have an affect on
  - Requests
  - Database commands
  - Memory (<1 mb per concurrent user)
  - CPU
- Deviations (over time) can mean
  - ‘heavy’ microflows
  - Growing database
- The more (web service) users using the application, the more requests you can expect to the business server, which has an effect on the JVM Memory and CPU Usage.



Question: What happens within this graph & application?

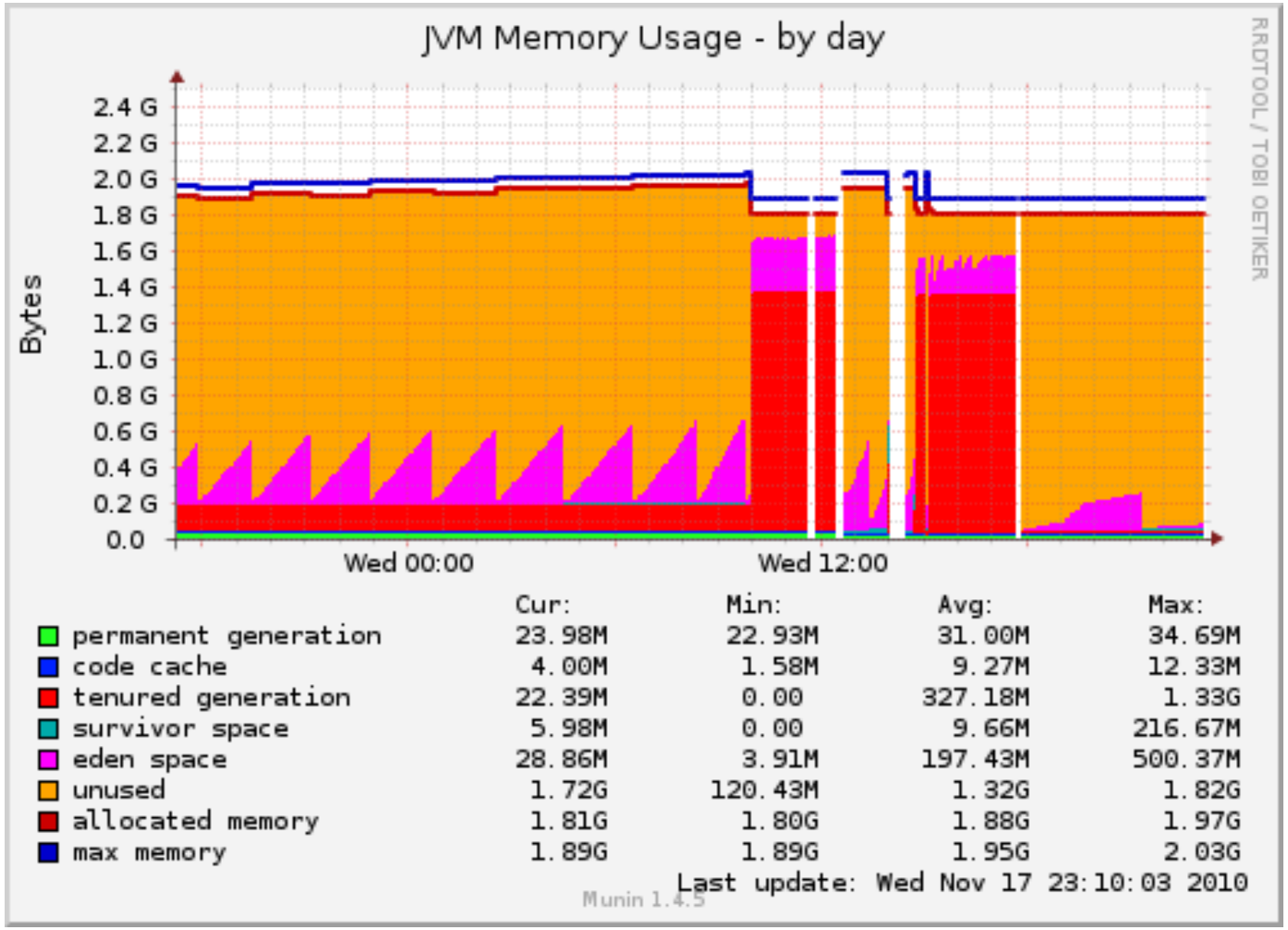
Analyzing graphs – Case 1



- **Question:** What happens within this graph & application?

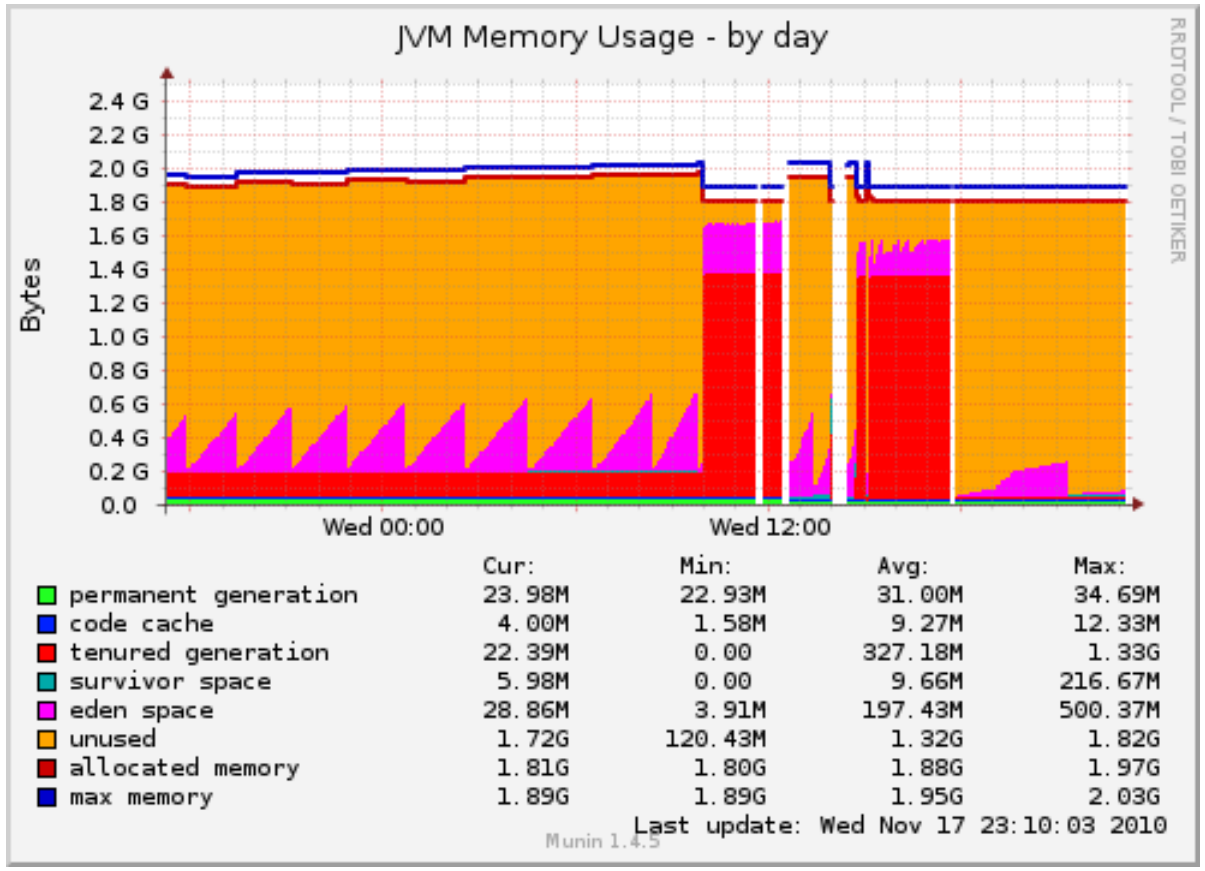
1. Objects are held captive, GC cannot free memory
2. Objects are held captive by new process
3. GC try to free memory from tenured generation
4. Tenured generation heap space keeps growing
5. The application crashes, given error: *java.lang.OutOfMemoryError: GC overhead limit exceeded*
6. Tenured generation is filled up again
7. Out of memory will happen again

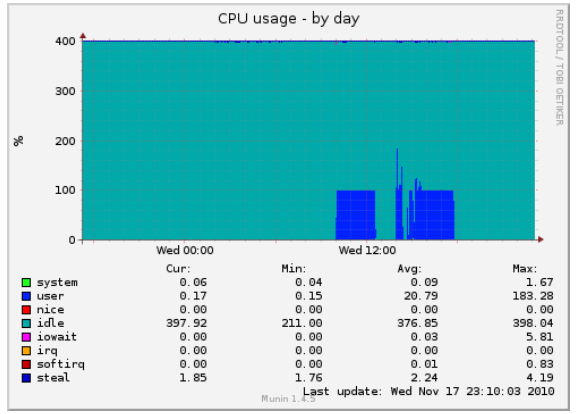
Analyzing graphs – Case 2

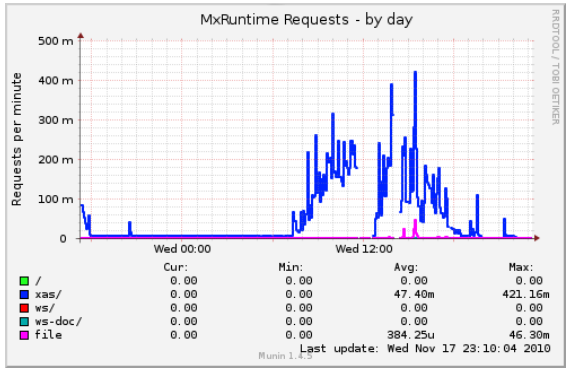


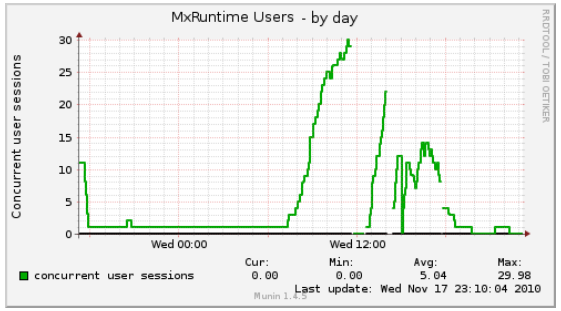
Question: What happens within this graph & application?

Analyzing graphs – Case 2





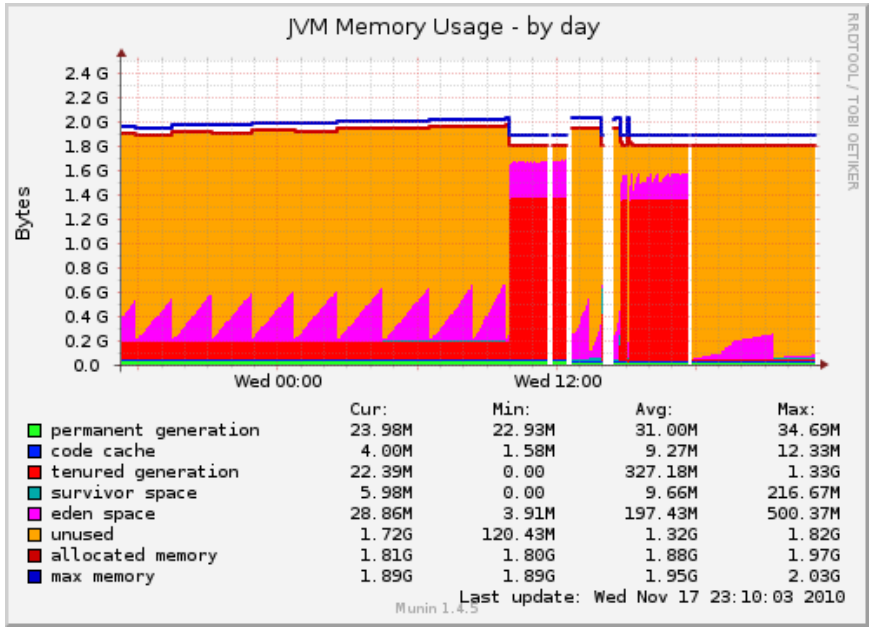


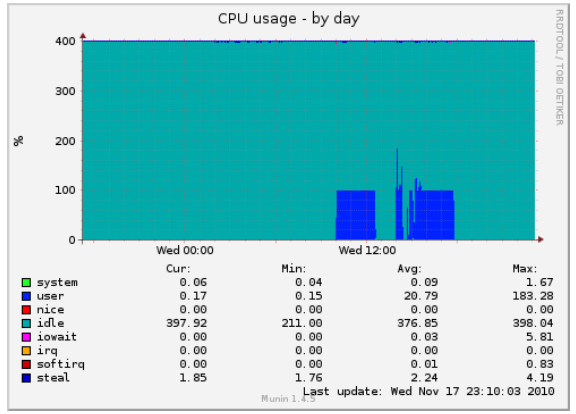


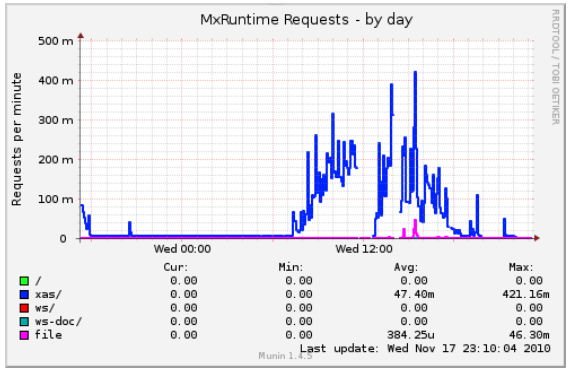
Question:

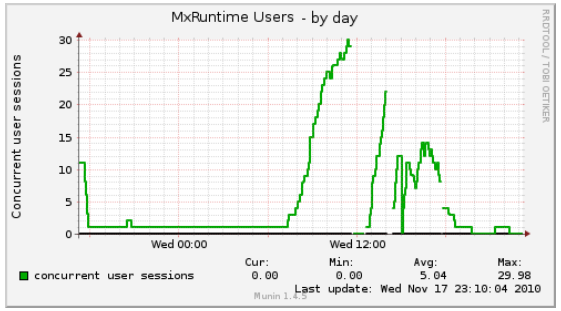
Which kind of information can be distillated out of the different graphs?

Analyzing graphs – Case 2









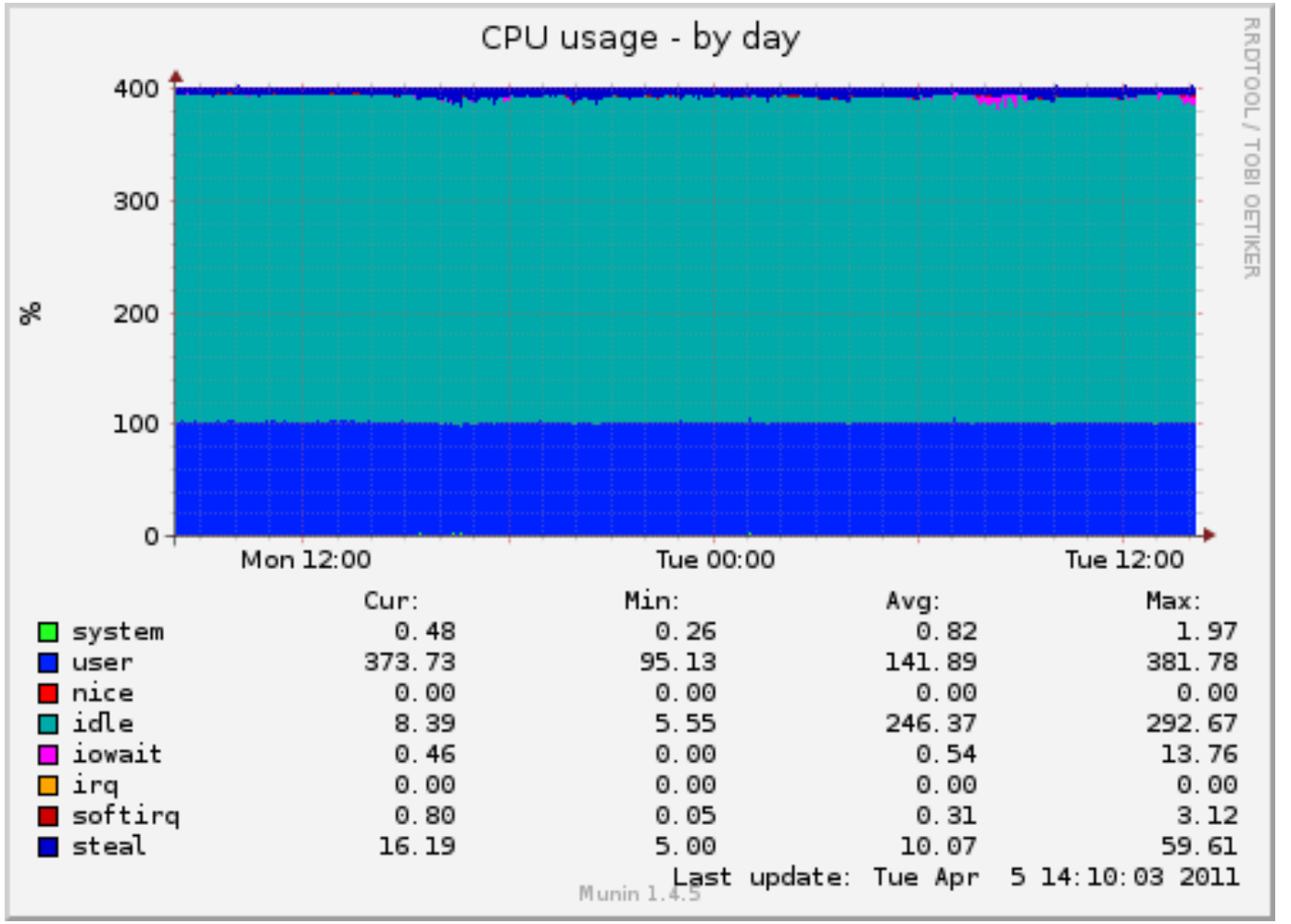
Question:

Which kind of information can be distillated out of the different graphs?

1. Due to unusual action creating a pile of objects, tenured generation grown extremely.
2. The application crashes, given error;

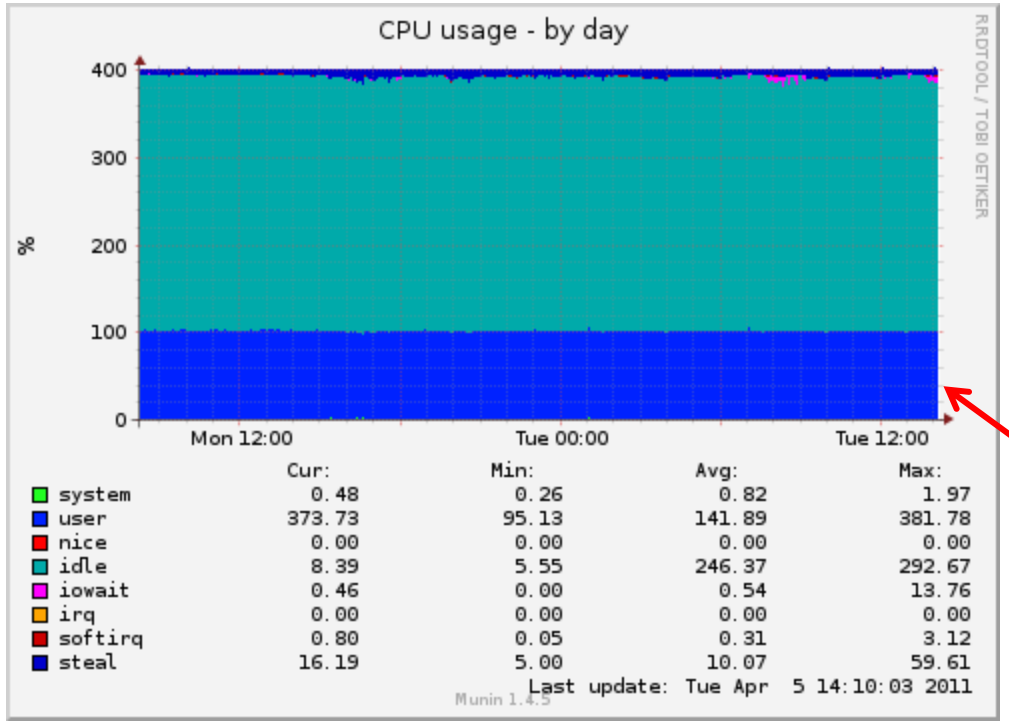
   java.lang.OutOfMemoryError: Java heap space

Analyzing graphs – Case 3



Question: What happens within this graph & application?

Analyzing graphs – Case 3



Constant CPU usage

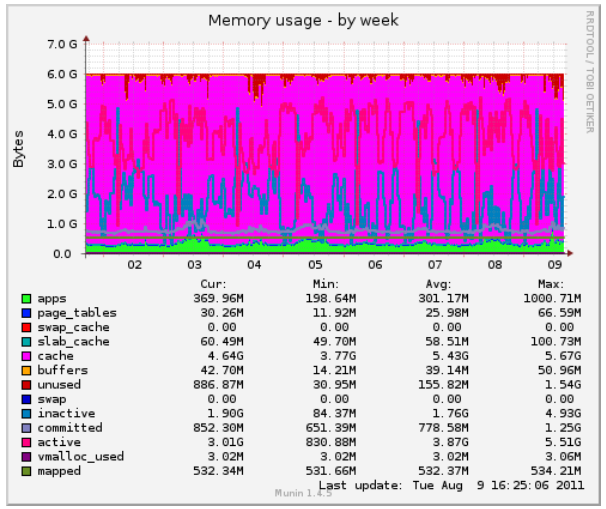
Question: What happens within this graph & application?

1. One microflow is running; Since only one core is used
2. It is a heavy Microflow; core is used at 100%
3. Might be a scheduled event, triggered multiple times and queuing

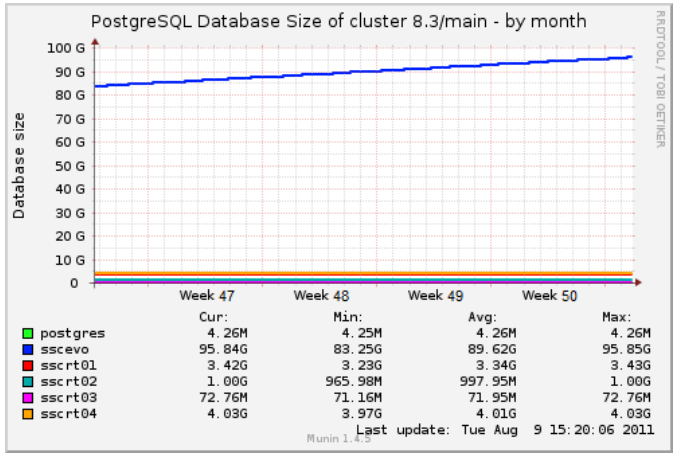
Analyzing graphs – Case 4

Data base is growing with 10 Gb in 4 weeks.
This indicates creation of a lot of objects.

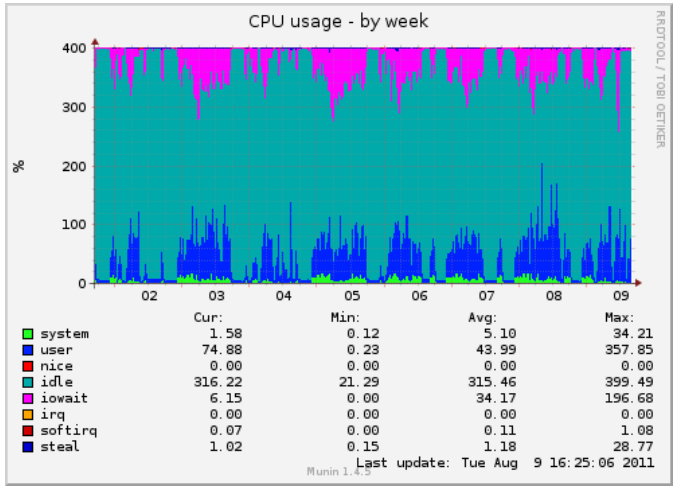
- Due to heavy application with useful data
- Generation of i.e. object for import, without proper cleaning



The database server will be used for temporary memory. This will have a negative influence on the database performance



Memory usage becomes a problem.
Data base contains more data than fits in the business server memory.



Analyzing graphs – Best practice

- Monitoring means a regular look at you monitoring graphs
- Search for trends
- Inform your PO and/or Development team when noting trends which MIGHT cause future problems (feedback item)
  - What might happen?
  - What is the consequence?
  - What might be a solution?

Logging

Logging – Overview

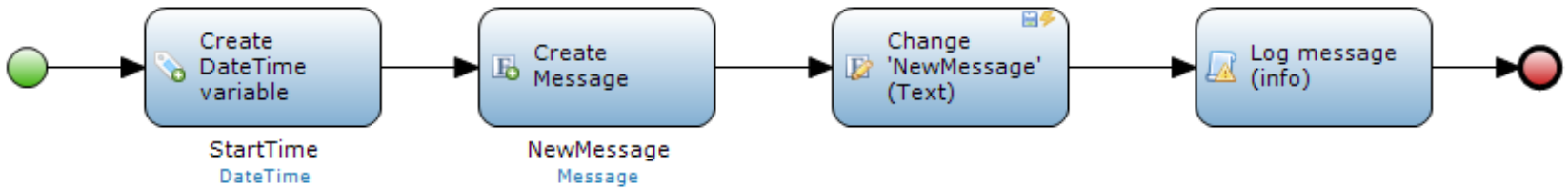
- Log of the MBS is stored in log files
- Always check the log file after a failed startup
- Not always necessary to find the actual root cause
- Recommended settings:
  - Archive Log files > 2 MB
  - Store the last 10 Log files

Logging – Common messages

- Errors in custom Java actions
- Java heap spaces
- Errors in microflows

Logging – Custom logging

- Log action in Microflow
- Log levels
- Usage
  - Log process critical fault with additional app/model information
  - Long running processes
    - Batches
    - Web services
    - Scheduled events
    -



Logging – Best practices

- Trace
  - Used for development long term
- Debug
  - Used for development short term
- Info
  - Logging of important business processes
- Warning
  - Logging of incident that need investigation
- Error
  - Logging of an incident which stops an critical use case
- Critical
  - Logging of incident which causes a sudden crash of the application

Logging – stacktraces examples 1/6

Occurrence

ERROR - M2EE: (1/1) java.lang.OutOfMemoryError: Java heap space

Possible cause

- Export to excel of an too large data set

Solution

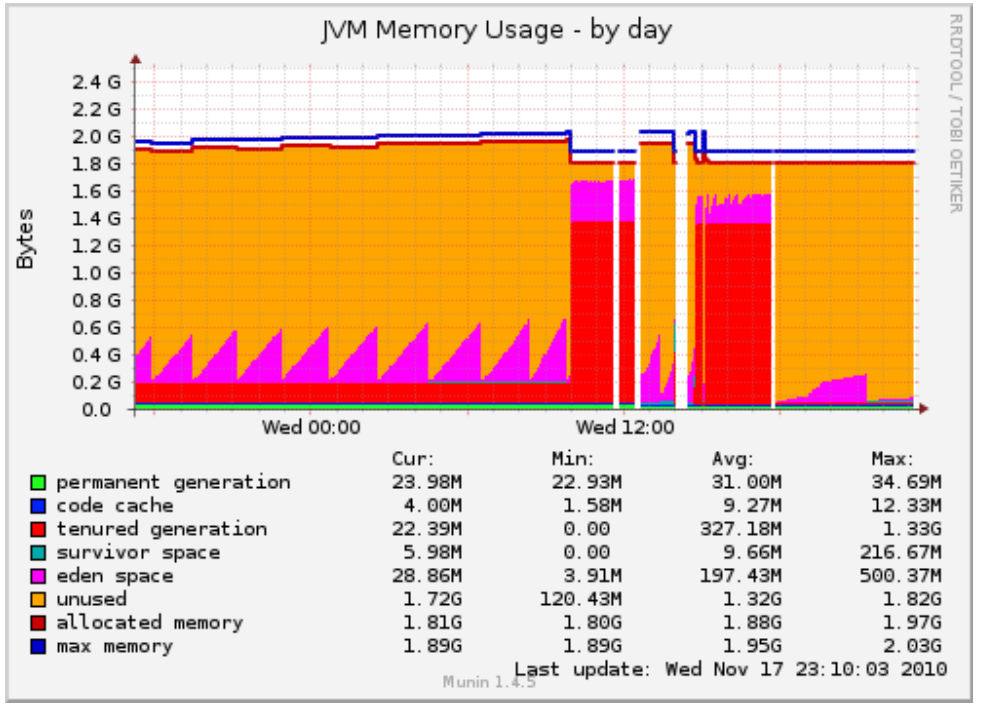
App admin

- Identify cause by searching for logs of excel exports
- Inform Developer about cause

Developer

- Change export to excel to a export to CSV

Reason: a data export to CSV is streaming, while export to excel isn't



Logging – stacktraces examples 2/6

Occurrence

ERROR - M2EE: (1/1) java.lang.OutOfMemoryError: Java heap space

Possible cause

- Processing a large set of data in a microflow

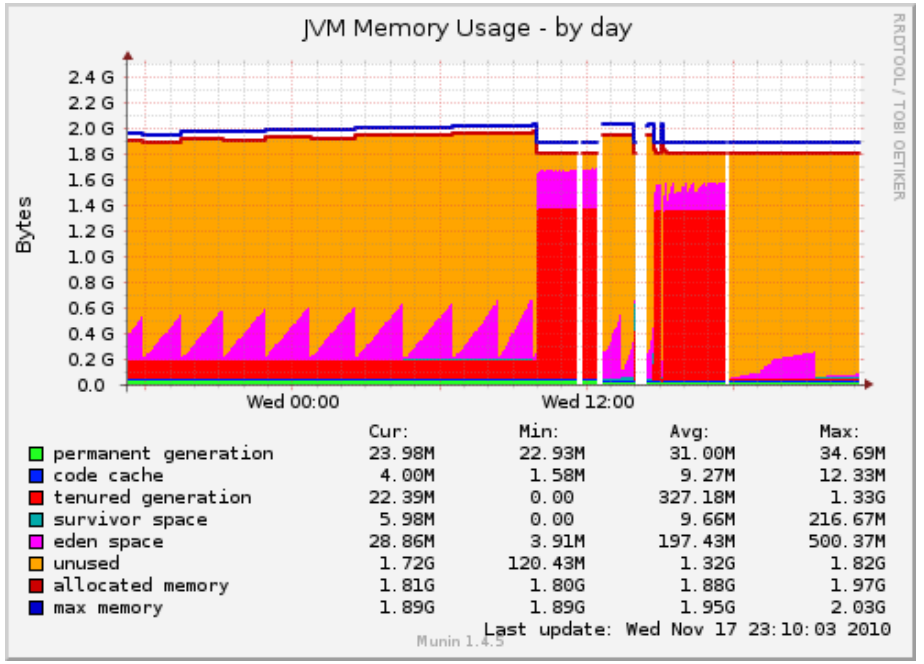
Solution

App admin

- Identify cause by searching for logs of large microflows
- Inform Developer with found logs
- Request Developer for adding logging activity in large microflows
- Request for more (temporary) memory

Developer

- Convert actions with batch operations
- Add logging activities on large microflows



Logging – stacktraces examples 3/6

Occurrence

ERROR - M2EE: (1/1) java.lang.OutOfMemoryError: Java heap space

Possible cause

- Combination of multiple synchronal heavy data related actions by multiple users

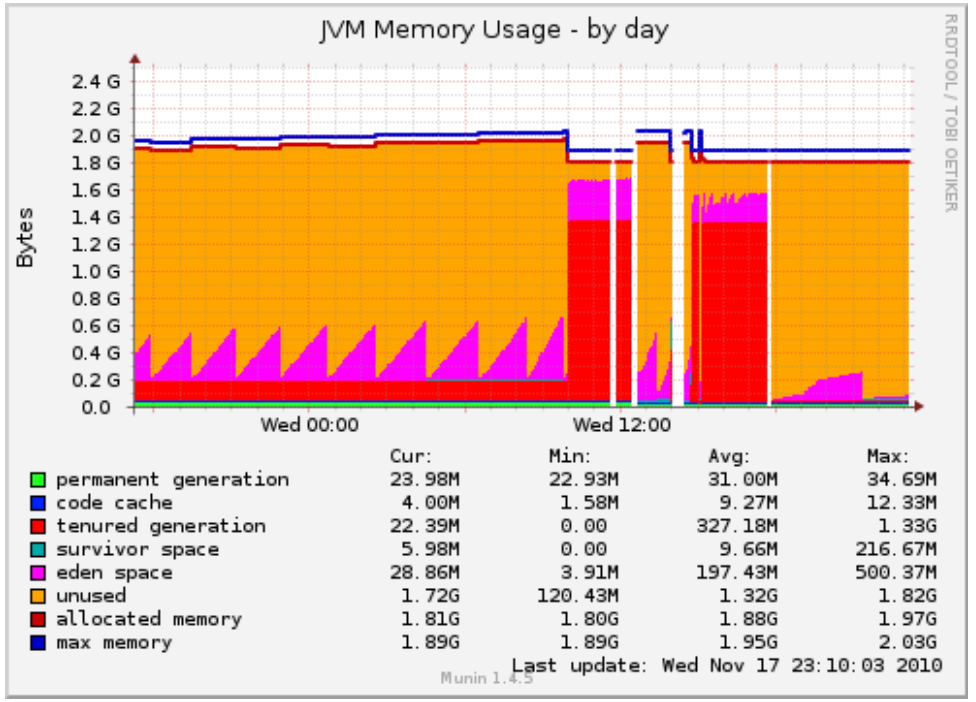
Solution

App admin

- Identify cause by searching for logs of large microflows
- Inform Developer with found logs
- Request Developer for adding logging activity in large microflows
- Request for more (temporary) memory

Developer

- Convert actions with batch operations
- Add logging activities on large microflows
- Implement planner module (+ Back end Business server)



Logging – stacktraces examples 4/6

Occurrence

ERROR - Connector: java.lang.OutOfMemoryError: GC overhead limit exceeded

Possible cause

- Garbage collector (Tenured Generation) is filled up due to Memory leak caused by custom Java action(s)
- OR external library which doesn't clean up the memory after finalization of action
- Occurrence rarely : Not a Mendix issue !

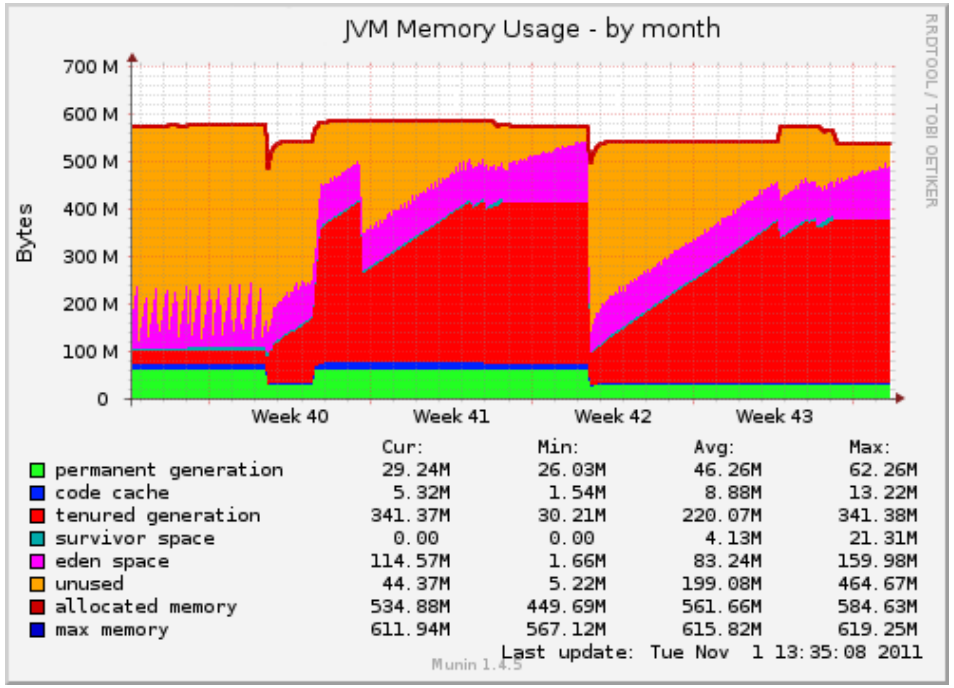
Solution

App admin

- Memory dump + analyze with eclipse (hard to do)
- Inform Developer about findings
- Temporary quick fix: if finding solution takes to much time; Scheduled restart

Developer

- Determine reason of generation of "Garbage"
- Adjust (Custom Java) so that they will clean up their own temporary objects



Logging – stacktraces examples 5/6

Occurrence

ERROR - Connector: java.lang.OutOfMemoryError: GC overhead limit exceeded

Possible cause

- Interval of a scheduled event is significant shorter then execution time of Microflow

Possible Result

New triggered scheduled events will be delayed

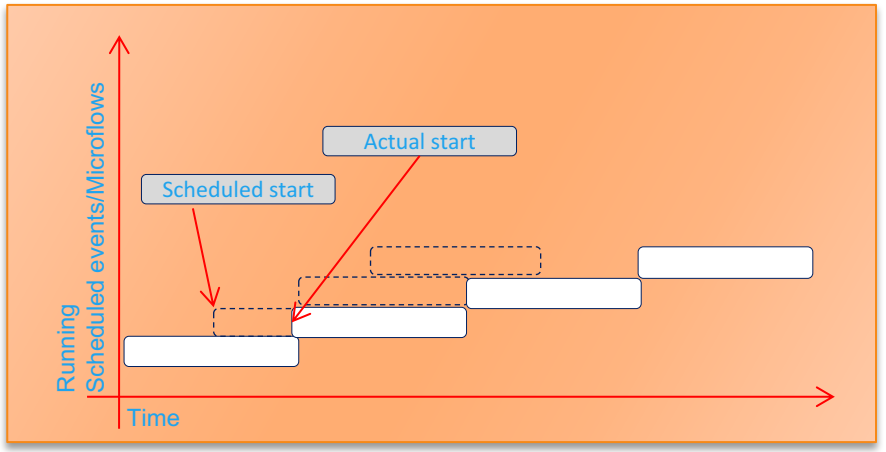
Solution

App admin

- Search log for scheduled event logs where scheduled events starts before previous has been ended.
- Inform developer

Developer

- Adjust execution cycle of scheduled events



Logging – stacktraces examples 6/6

Occurrence

com.mendix.core.CoreException: Exception occurred in action 'Microflow [Module.Microflow_Name]', all database changes executed by this action were rolled back

Cause

- A microflow caused an error
- Modeling development error

Solution

App admin

- Investigate log file asap
- Try to determine the (end user) scenario when the error occurred
- Inform developer about error with details/stacktrace

Developer

- Investigate related microflow
- Adjust behavior accordingly

Alerting

Alerting – Overview

- Status overview (Dashboard)
  - Overview app environment status
  - High-level, indicates presence of possible malfunction
- Email alerts

Alerting – Recommended settings

- Alerts (application status)
  - Warning / Critical Alerts
  - Application unexpected down
  - Critical messages in log
  - Failed health check
  - CPU load > 90-100%
- Store for 1 year

Alerting – Custom alerts

- By adding a ‘Log message’ with level ‘Critical’
- Add a health check Microflow to the Model
  - Empty string means every thing is OK
  - Returns a string with the error message

Alerting – Solving critical alerts

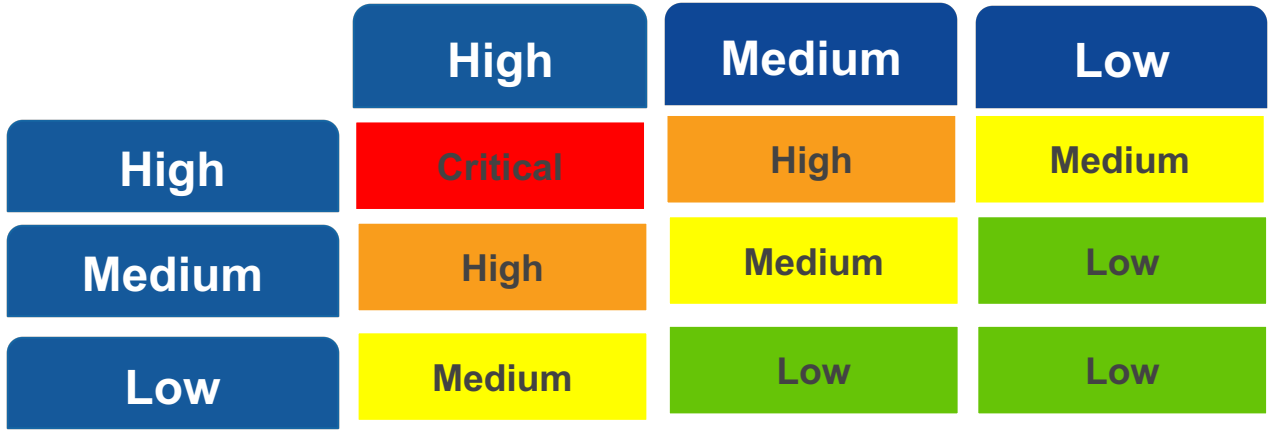
- Restart
- Search end of log file for stack trace
- Common critical causes
  - Java Heap Space
    - Excel exports (search for logs ??)
    - Batch operations in 1 microflow (often scheduled events)
    - Write log  for these operations (in the model)
- Temporarily increase memory

Mendix Support

Service Level specifics - priority

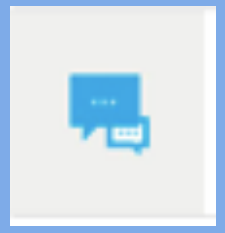
Impact:
- High: A high priority production issue with a high impact on the customer's business, impacting (almost) all users
- Medium: A production issue with intermediate impact on the customer's business, impacting a group of users
- Low: A trivial production issue with no impact on the customer's business.

Urgency:
- High: The operational functionality is severely disrupted.
- Medium: The operational functionality is limited disrupted
- Low: The operational functionality is hardly disrupted.



How to contact us

Community Forum
forum.mendix.com



Support Portal
support.mendix.com

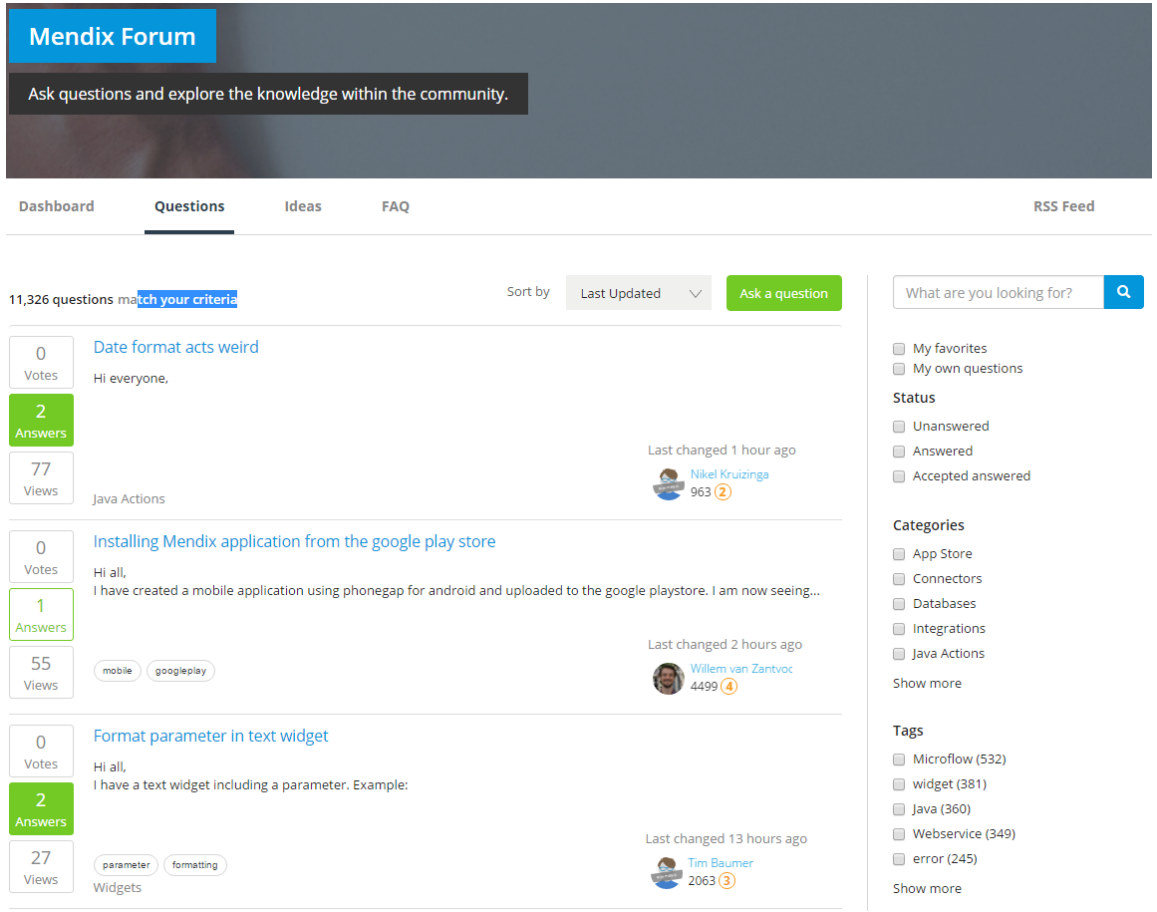
Email
Support@mendix.com

Phone
US 1 617 939-9638
UK +44 3308 280000
NL +31 10 478 8848
SA +27 11 550 2419

//developers.mendix.com/support/

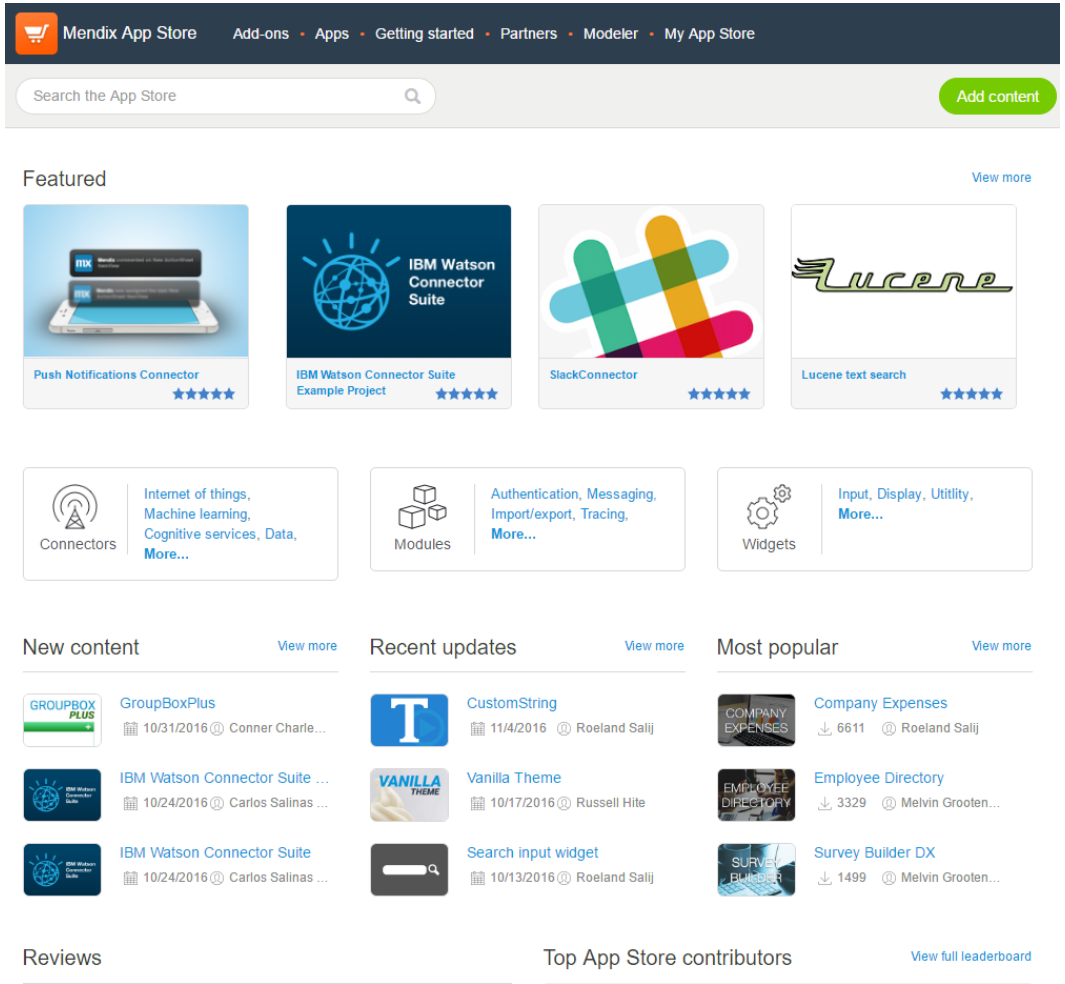
Important links : forum

- Forum : forum.mendix.com
- Connect with other developers : ask and answer questions
- Suggest new features and vote



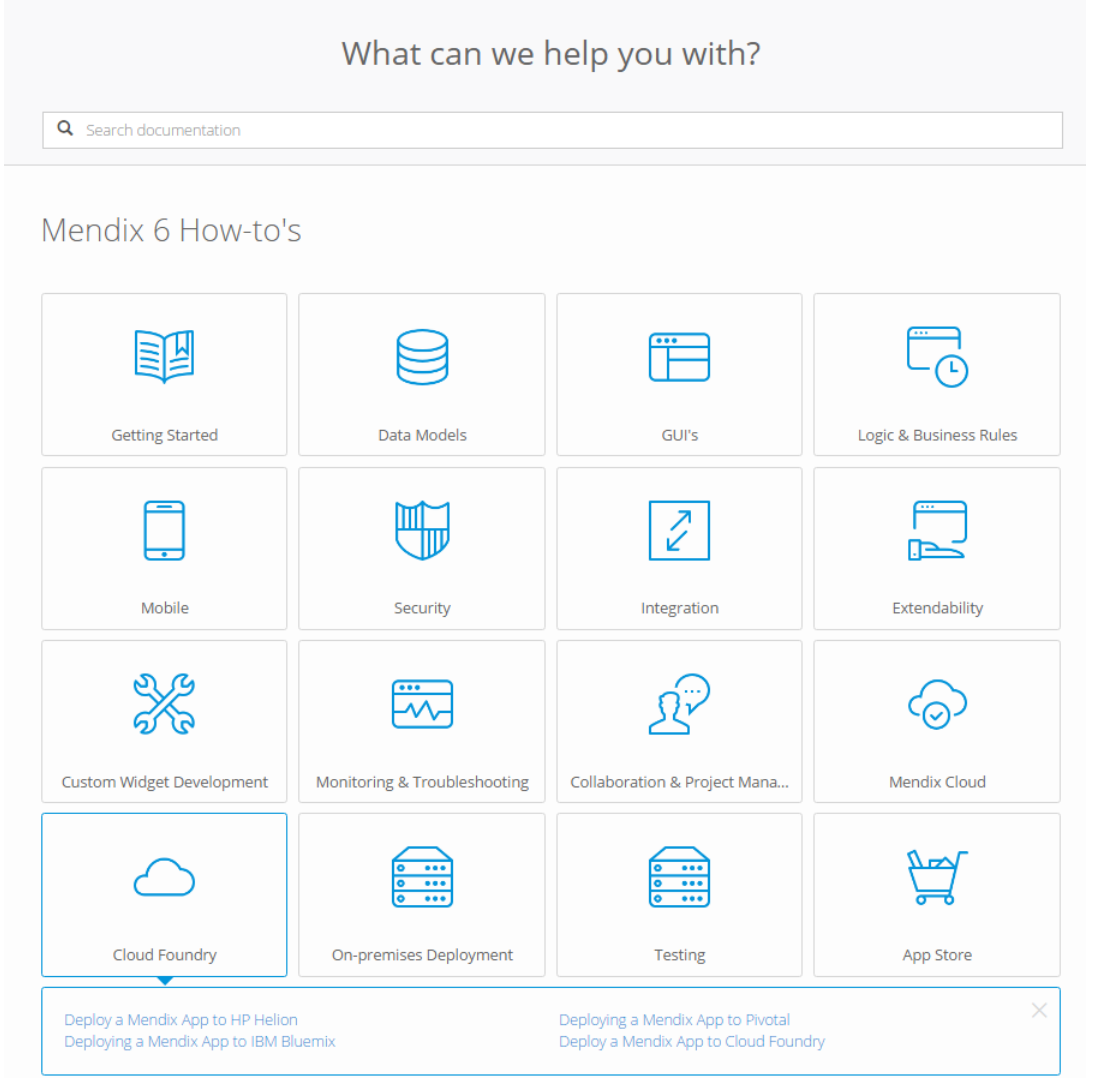
Important links : appstore

- Appstore : appstore.mendix.com
- Download the components to jumpstart your project, such as:
  - App services
  - Layouts
  - Themes
  - Modules
  - Widgets
- Support levels on Appstore Content
  - Platform
  - Extended Support
  - Community
  - https://world.mendix.com/display/howto50/App+Store+Content+Support:



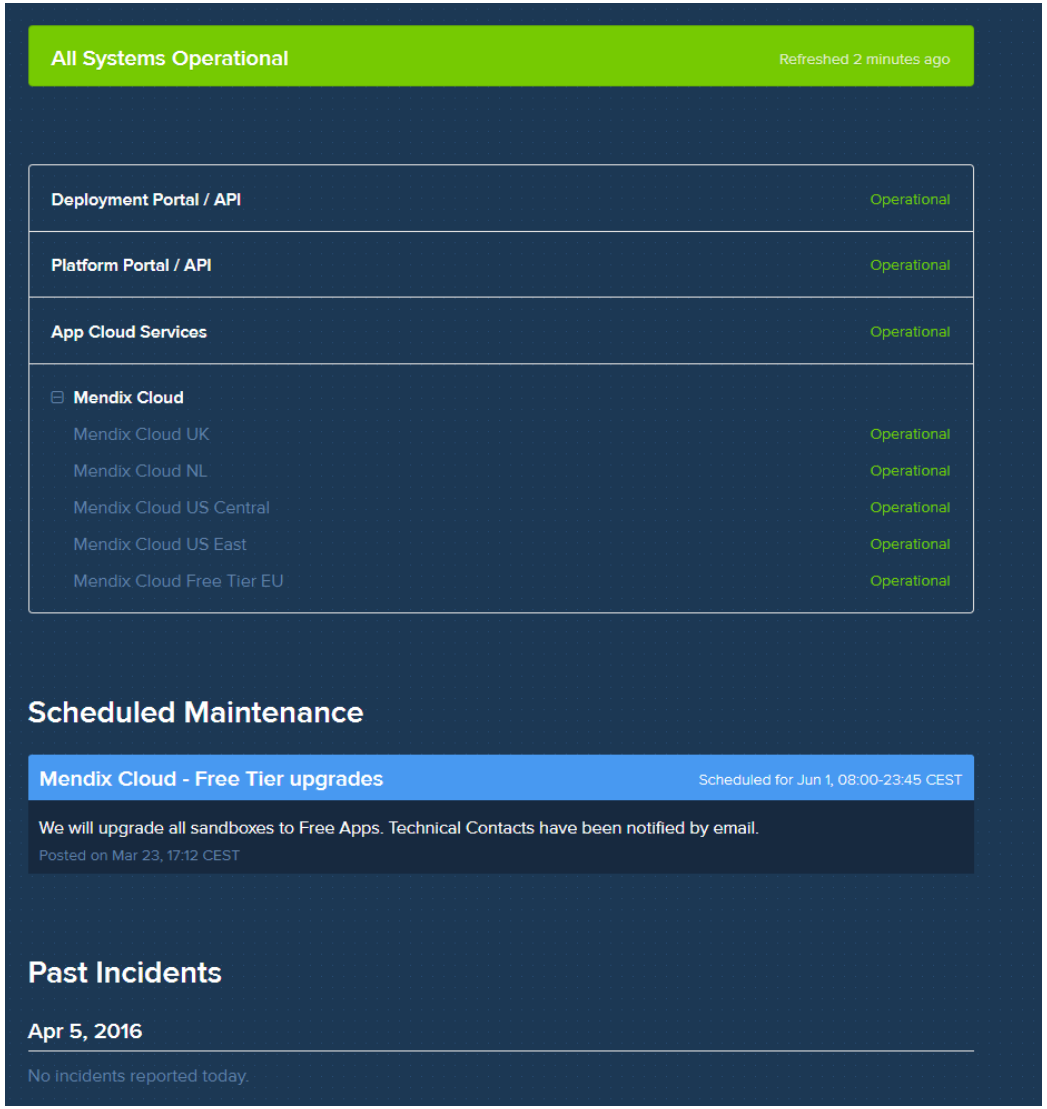
Important links : documentation

- Documentation : docs.mendix.com
- Contains the following :
  - Release Notes
  - API documentation
  - Reference Guide
  - Getting started / How-to’s
  - Blogs
  - Support documentation



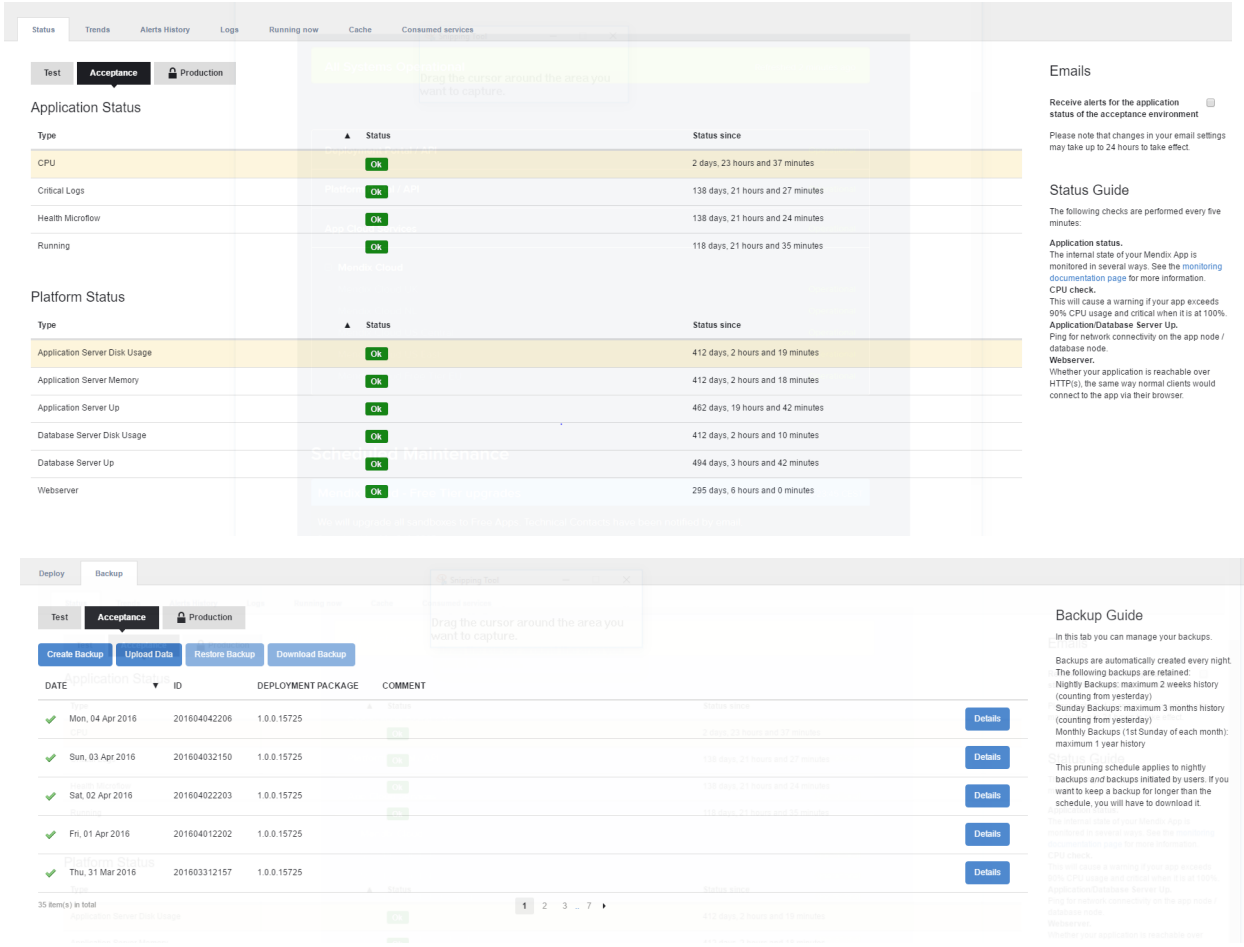
Important links : platform status

- Platform status : status.mendix.com
- Check status of Mendix in real time :
  - API (Deployment Portal / Platform Portal)
  - App Cloud Services
  - Mendix Cloud
- Incidents
- Maintenance
- Subscribe to updates



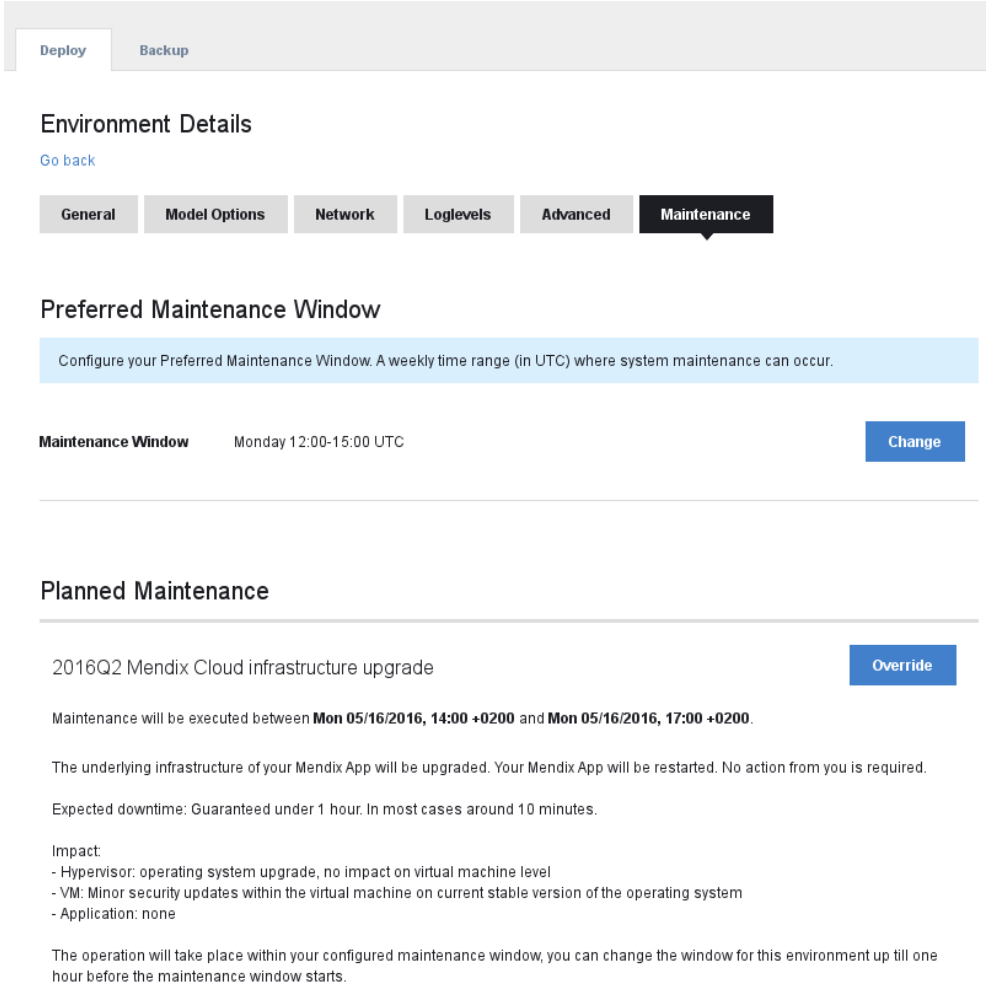
Important links : Cloud platform

- Cloud platform : cloud.home.mendix.com
- Customer self service management of apps by technical contact
- Deployments
- Backups
- Custom Domain URL
- Monitoring
  - Application status (all environments TAP)
  - Trends (graphs)
  - Alerts
  - Logs



Important links : Cloud platform - maintenance

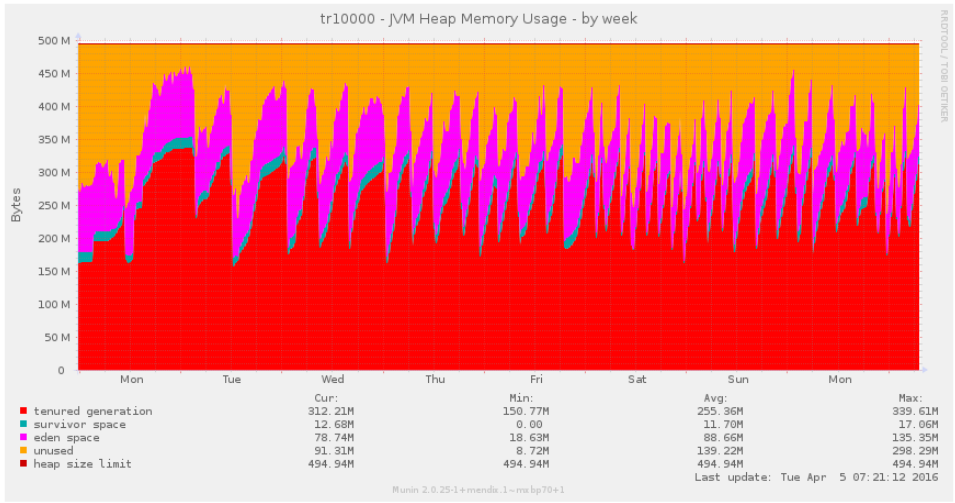
- Cloud platform : cloud.home.mendix.com
- Preferred maintenance window
- Email alert for planned maintenance
- Possibility to override preferred window



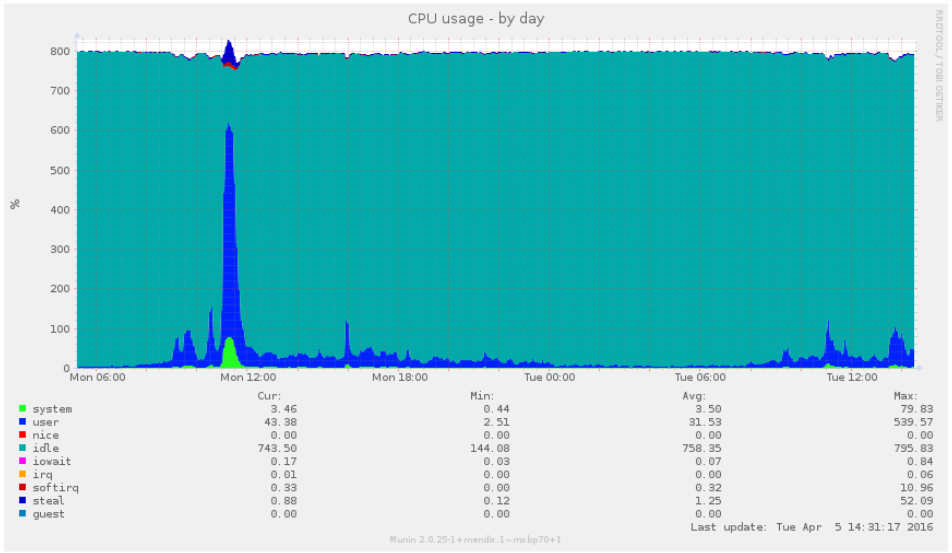
Important links : Cloud platform - trends

- Cloud platform : cloud.mendix.com
- Customer Self Service monitoring of trends
- Application Statistics
  - External request
  - Cache
  - Jetty threadpool
  - User accounts
  - JVM Heap
  - Database queries
- Database Statistics
  - Transactions
  - Table vs index size
- Application Node Statistics
  - CPU
  - Average load
  - Disk latency
  - Disk usage
  - Disk IO/s
  - OS memory
  - Disk throughput
  - Disk utilization
- Database Node Statistics
  - CPU
  - Average load
  - Disk latency
  - Disk usage
  - DB connections
  - Disk IO/s
  - OS memory
  - Disk throughput
  - Disk utilization

JVM Object Heap

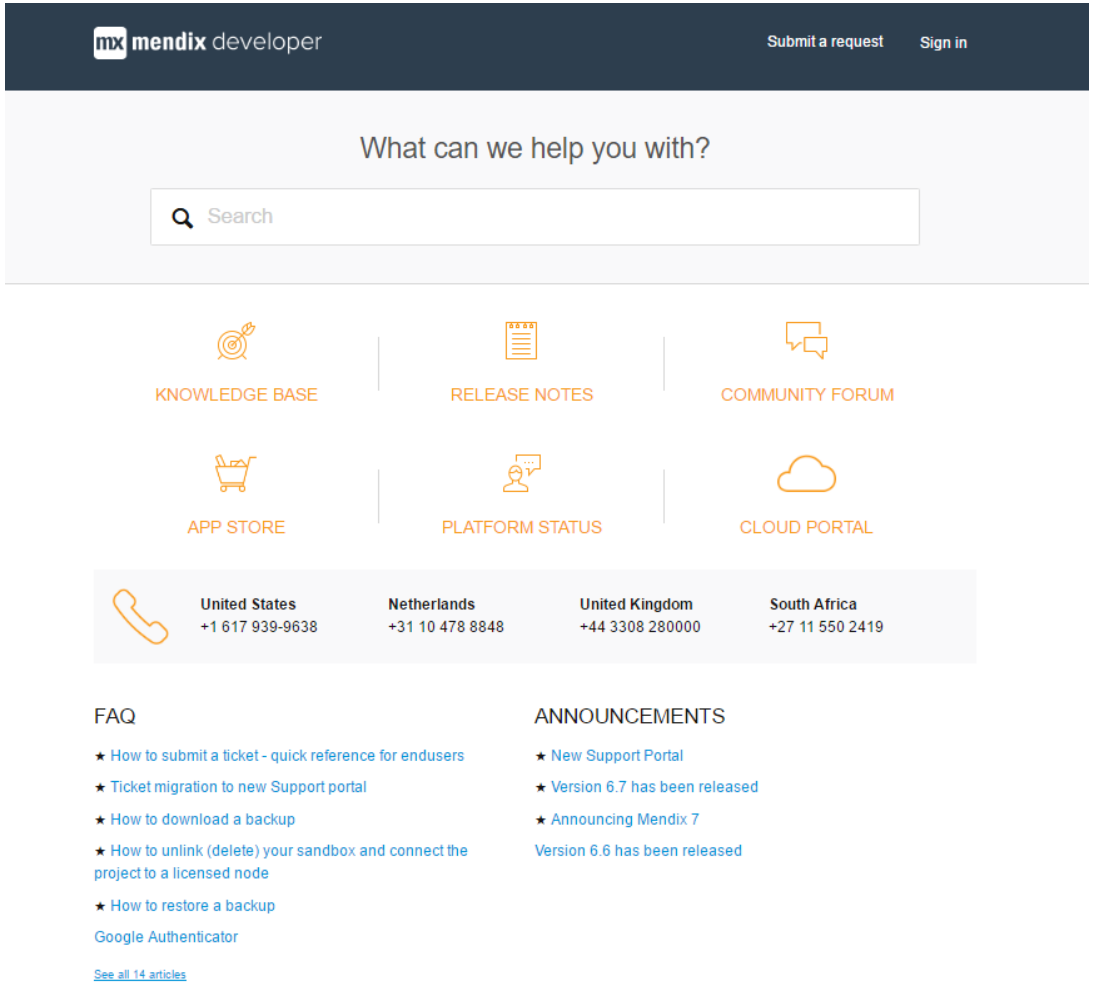


Application node CPU usage



Important links : support portal

- Support portal : support.mendix.com
- Contains the following :
  - Submit and browse tickets
  - View own tickets and tickets on company and affected app level
  - Incidents
  - Standard Changes
    - New App, Resizings, Reset Google authenticator etc.
  - Non standard changes
  - Questions



Q&A