EBG Server Education
- LXCA User Interface

Jeff Van Heuklon | RAS architect − 4/15/15

Update picture with course product picture (Optional) or Delete

Overview for xHMC UI charts

- Architecture of the xHMC User Interface
  - Web Application
  - REST API Driven
  - Architecture Concepts
  - Dynamic and Static Data
  - Browsers and Tablets
- Content Overview
  - Initial Setup
  - Menu, Pods, Dashboard
  - Resource Uis
  - Jobs
- Possible Issues & Triage
  - Potential problems
  - Logging location
  - Browser debugging

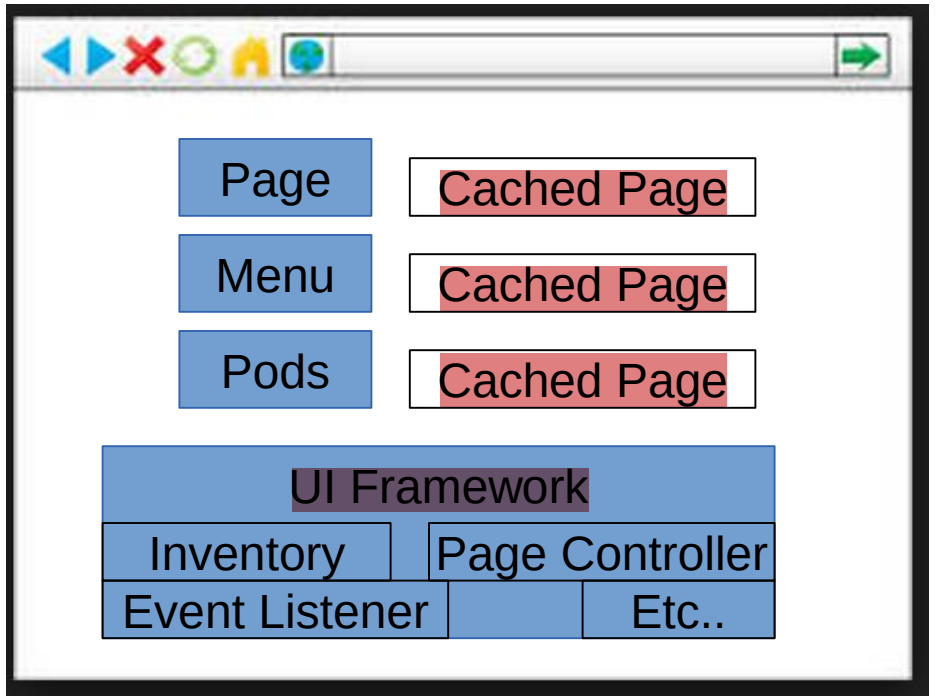
Architecture of xClarity Administrator User Interface

- Web Application
  - Typical web application.
  - Browser connects to server via ip or host name using https protocol
    - Http results in server not found
  - Client requests and loads our html, css and scripts that drive the Web App UI.
  - Each page has unique URL and is bookmarkable
  - Browser back & forward buttons generally work as expected to navigate to/from content pages of the app.
- REST API driven
  - All live content comes from REST API interfaces. (ie. List of servers, network setting values, etc...
    - This retrieval occurs after all the application and page scripts are loaded.
    - Usually will see a “loading...” icon or text on the screen while data may be getting fetched.
    - Note: All pages even on a busy system should load within a few seconds. Failure to load promptly could indicate a problem
  - The UI **usually** renders the data its given very faithfully. If data values looks wrong or are missing in the UI, it very often (but not always) means the data we're given from the REST API is also wrong.

Architecture of xClarity Administrator User Interface

- User Interace Underpinnings – aka GUI Framework
  - Architecturally, this is the invisible chunk of code running in the browser that is responsible for loading and managing active visible content.
    - Each content section (aka widget) behaves relatively independently, interacting only with framework.
  - Contains background code like event monitors, and client side inventory data storage and caching.

Xclarity Appliance

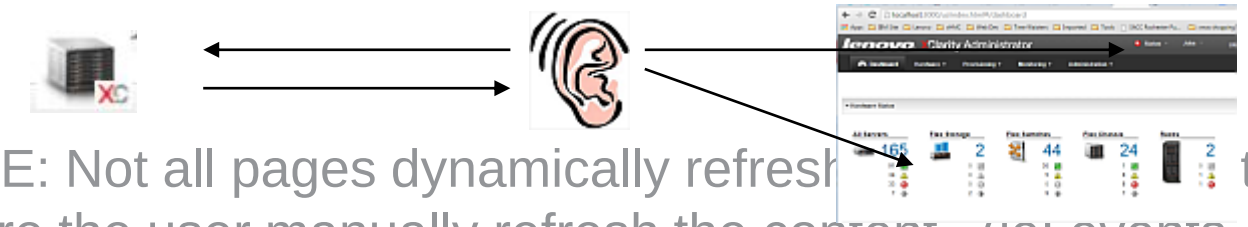


Architecture of xClarity Administrator User Interface

- User Interface Content
  - Common top level elements like the top menu, and stats / jobs pods, pages and dialogs
  - Each Content sections is loaded by the framework, then either destroyed or cached with the user is no longer viewing that “page”
    - Some pages are cached which will provide instant load when going back to that page.
    - Others are not, which will cause the page to load normally, re-rendering and fetch data anew.

Architecture of xClarity Administrator User Interface

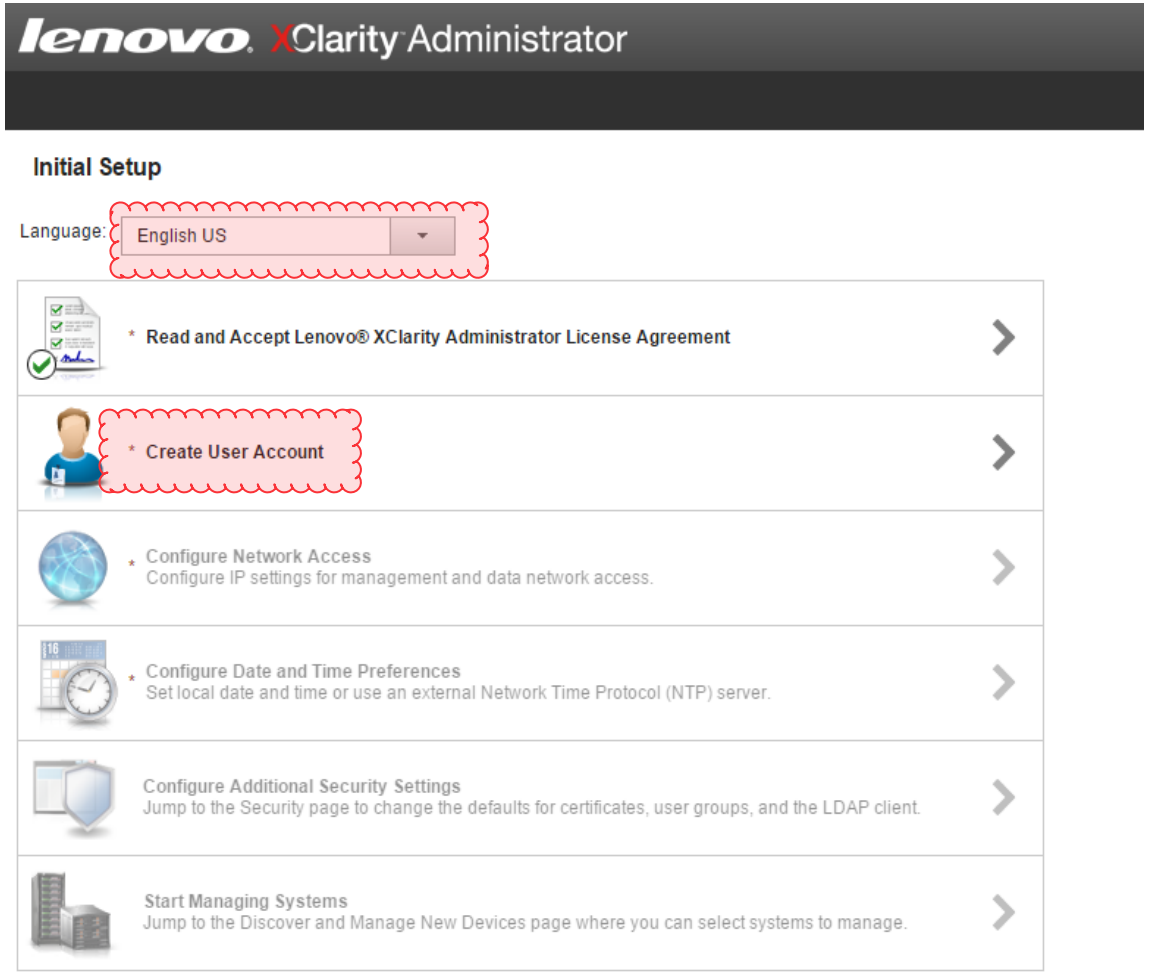
- Eventing notifications
  - The UI framework and server have a mechanism such that our web sessions "listen" for notifications to tell us if page content changes.
  - When a change is detected all listening pages get notified
  - If a widget on screen gets notified, it takes action and refreshes.



  - NOTE: Not all pages dynamically refresh in our implementations, some pages are forced to require the user manually refresh the content. (ie: events & jobs)
    - On these pages, you will see a refresh Icon:
- Browser & Tablets:
  - Support major browsers, firefox, ie, chrome (FF23+, IE9+, Chrome, Safari)
  - Using our UI through a touch interface (ie. Tablets and Touchscreens) is supported.

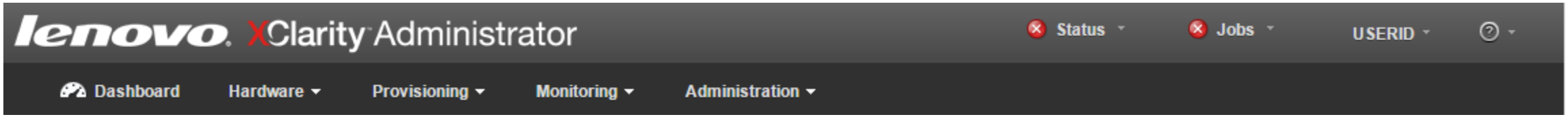
Content Overview: Initial Setup

- Login and Intial setup pages allow user to select
- language
- Content sections get unlocked as user completes
- steps
- NOTE: After create user step, the user is logged in
  - If the user walks away or restarts browser, they will be redirected to login again before proceding through the rest of the steps



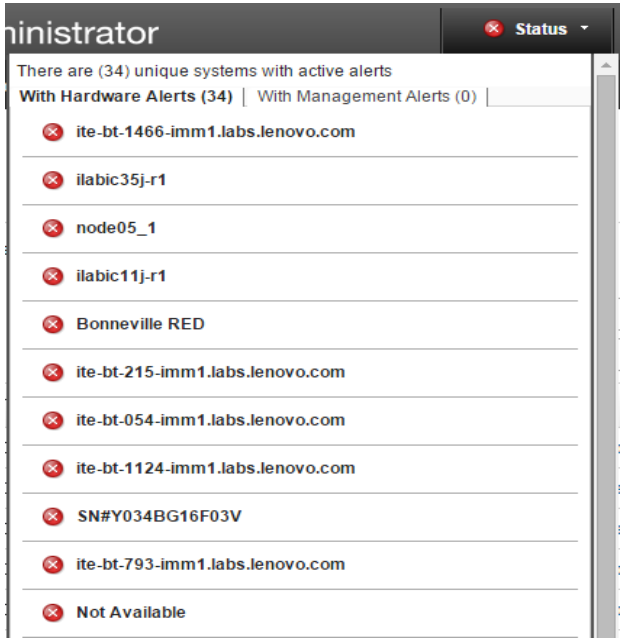
Content Overview: Menu & Pods

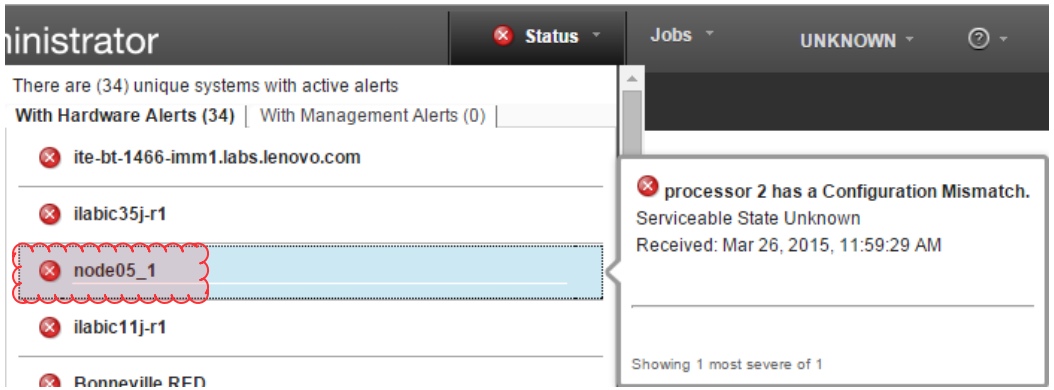
- Menu provides quick navigation to functions and important information
  - Note that system status and present job status is represented at a glance in the menu bar. Shown regardless of content panel shown.
  - User can quickly and easily expand the status list and active and recent jobs.



Content Overview: Menu → Status

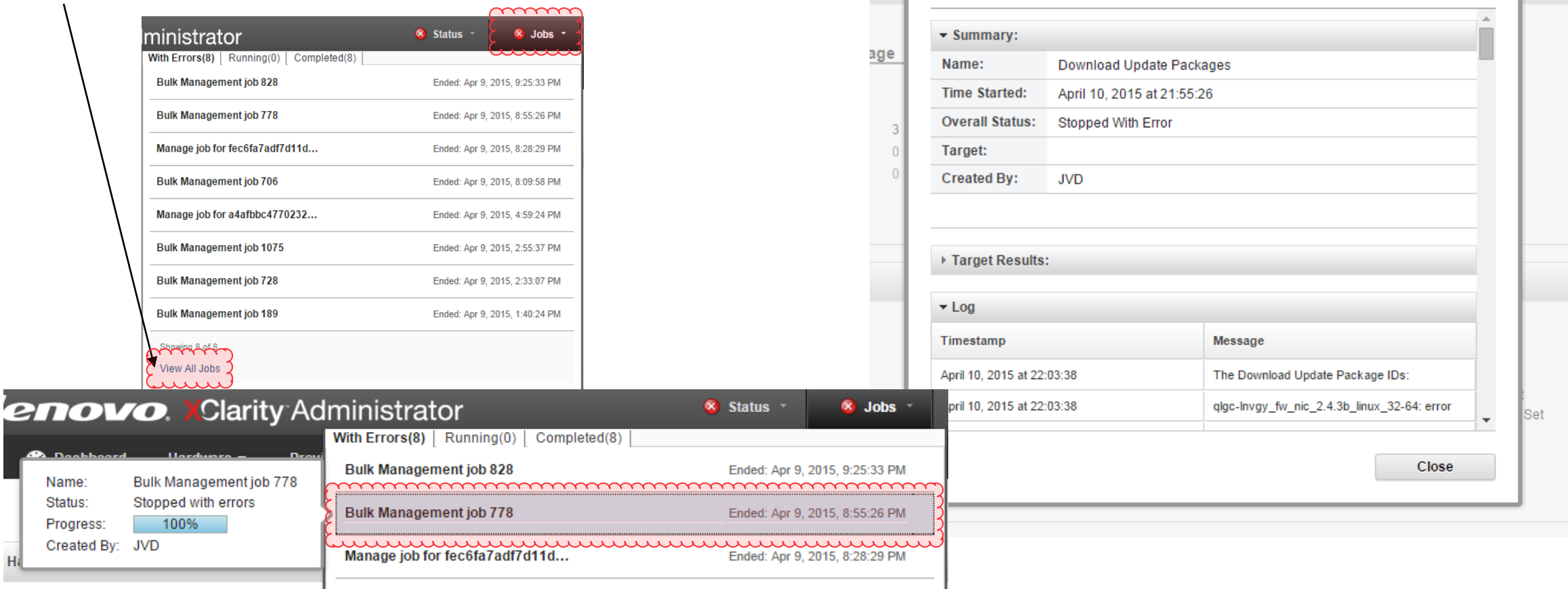
- Drill into to status to get at a glance problem information.





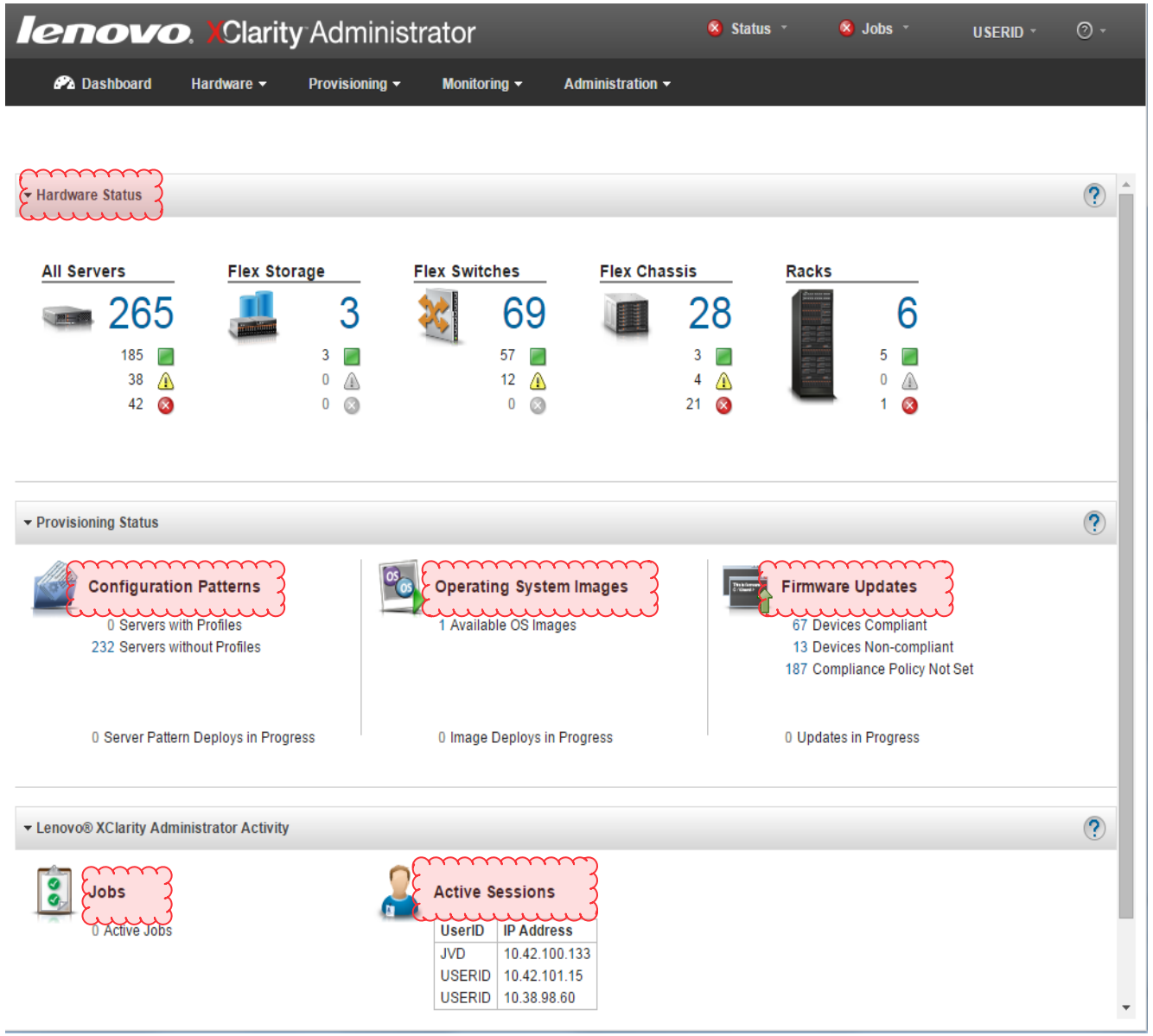
Content Overview: Menu → Jobs

- Expand jobs to see curent and recent job activity
  - Clicking a job will open Job details with complete job information
  - Also note quick link to all jobs from the pod



Content Overview: Dashboard

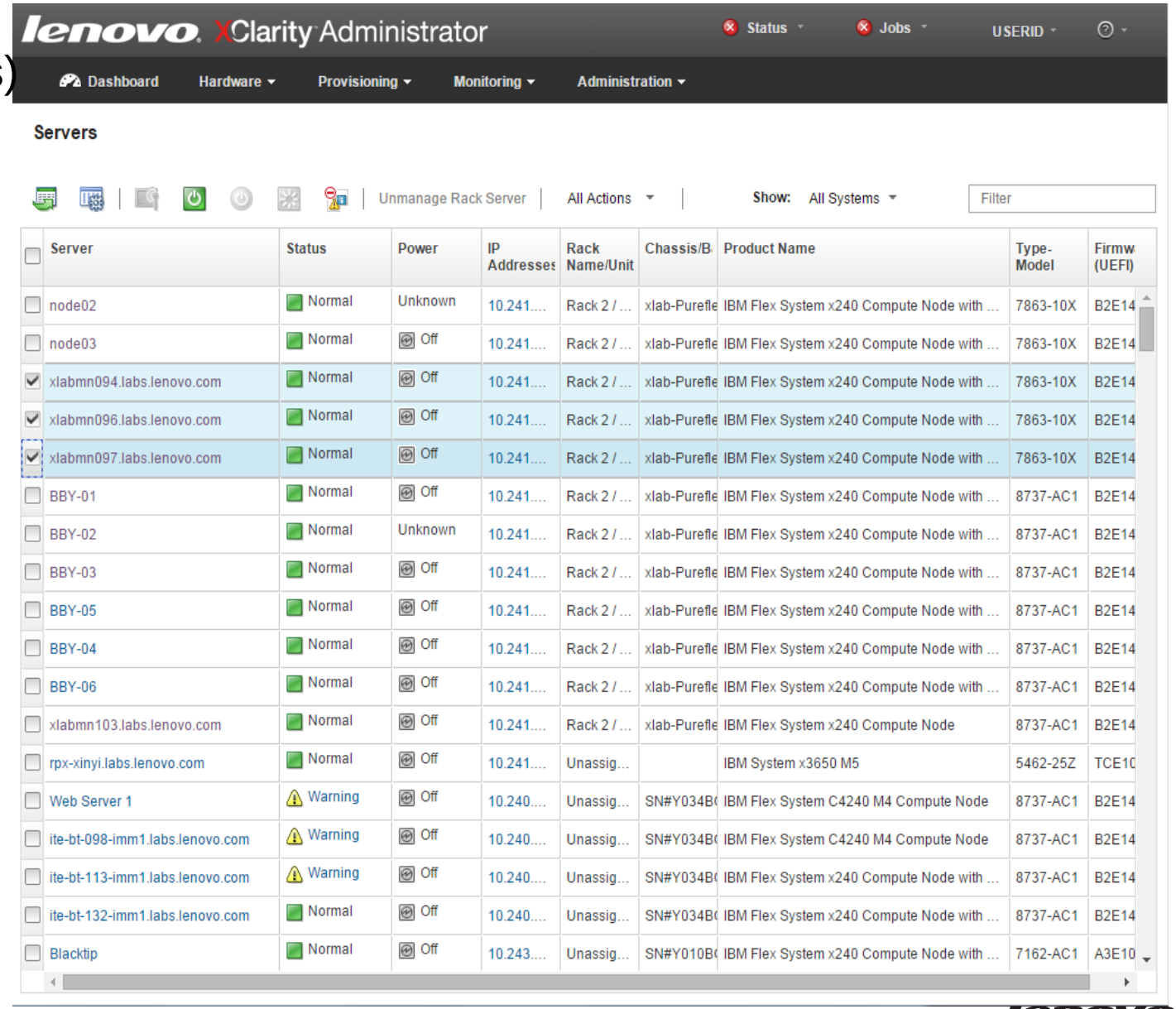
- Dashboard is the default xHMC page
  - Functions as a quick summary of all systems and activity
  - All sections on this page have quick links to dive quickly to associated content
    - Clicking server section to see all servers
- Contains Overview for:
  - All hardware
  - Config patterns
  - OS Deployment
  - Firmware updates
  - Jobs
  - Active sessions



Content Overview: Resource Views

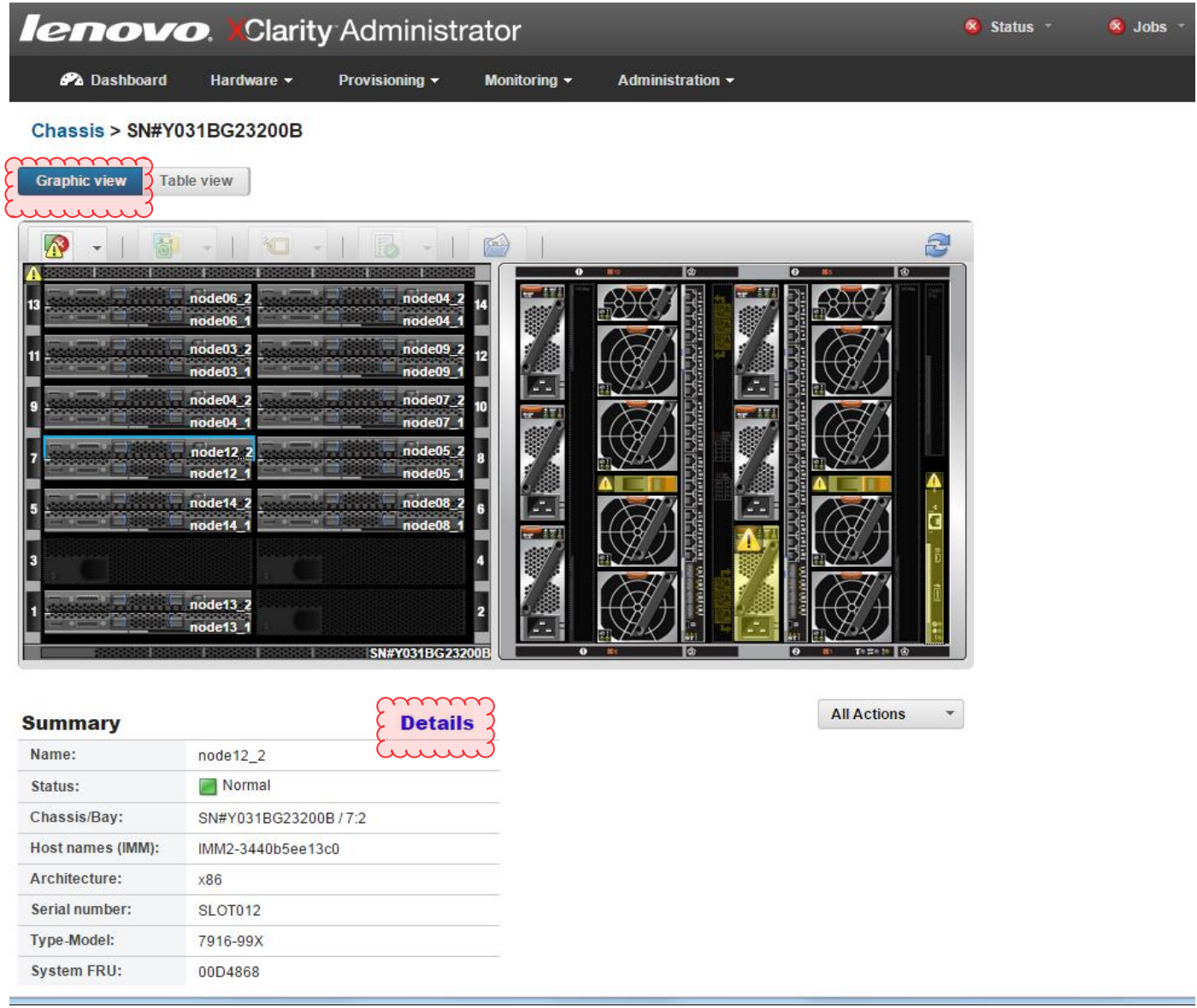
- Straightforward view (typical of other views)
  - Note actions change for IO modules, Chassis, and other hardware types
- Clicking name will take you to details
- Toolbar actions include (from left to right):
  - Export content to csv file
  - Customize visible columns
  - Launch remote control
  - Power on / off / restart
  - Exclude events (for selection)
  - Unmanage rack server (rack server only)
  - All actions (all the above and more)
  - Filter list by chassis or rack
  - Text matching filter (all table columns)
- Click the node name to dive into details

  - Jobs
  - Active sessions



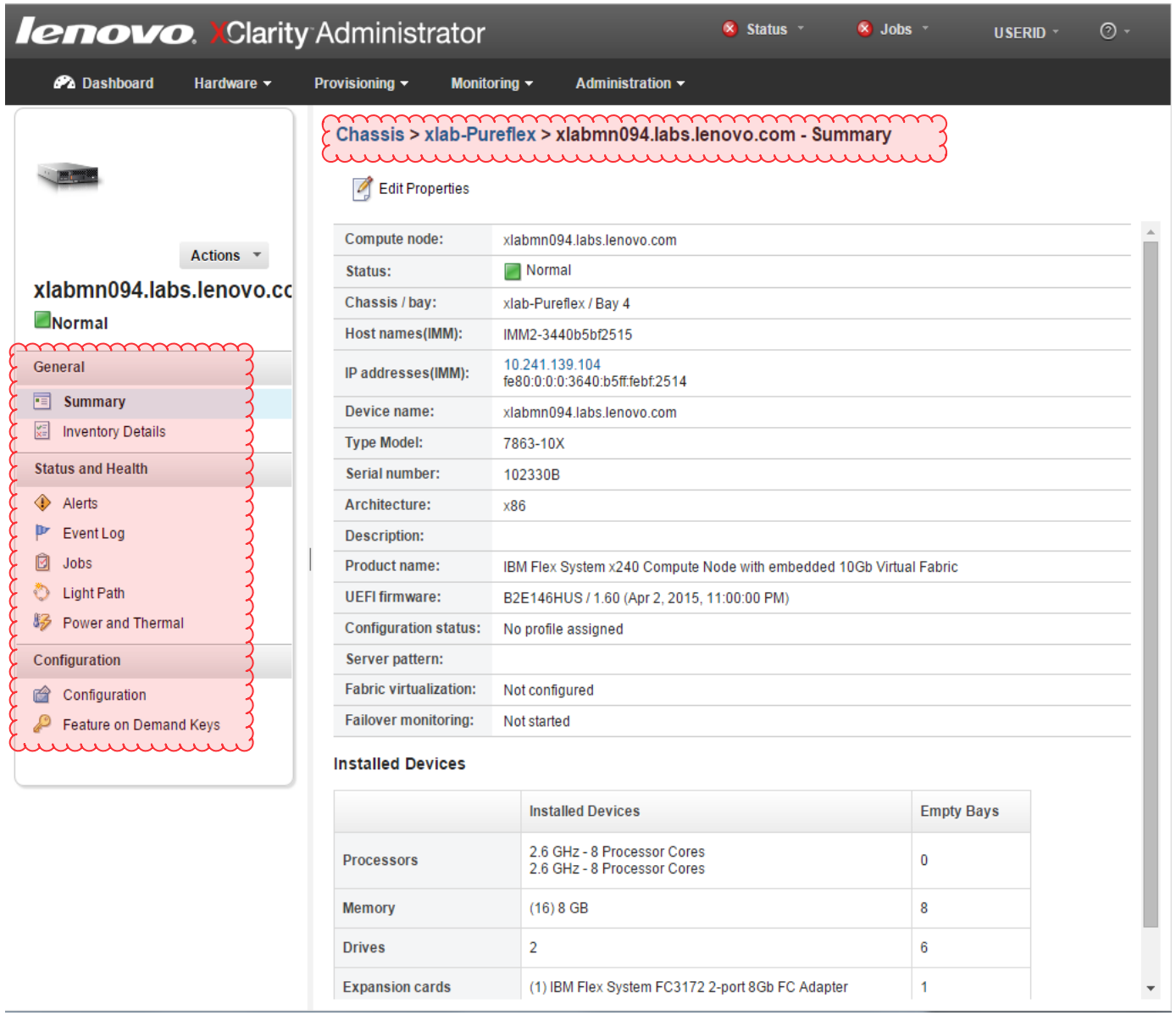
Content Overview: Resource Views → Graphical Chassis

- Reached by selecting from Chassis list view
- Graphical view toolbar allows different
- graphical overlays
  - Node status
  - LED states
  - Property values
  - Compliance status
  - Configuration patterns
  - Note: Tooltips change to reflect current overlay content
- Note quick links to details and available
- actions



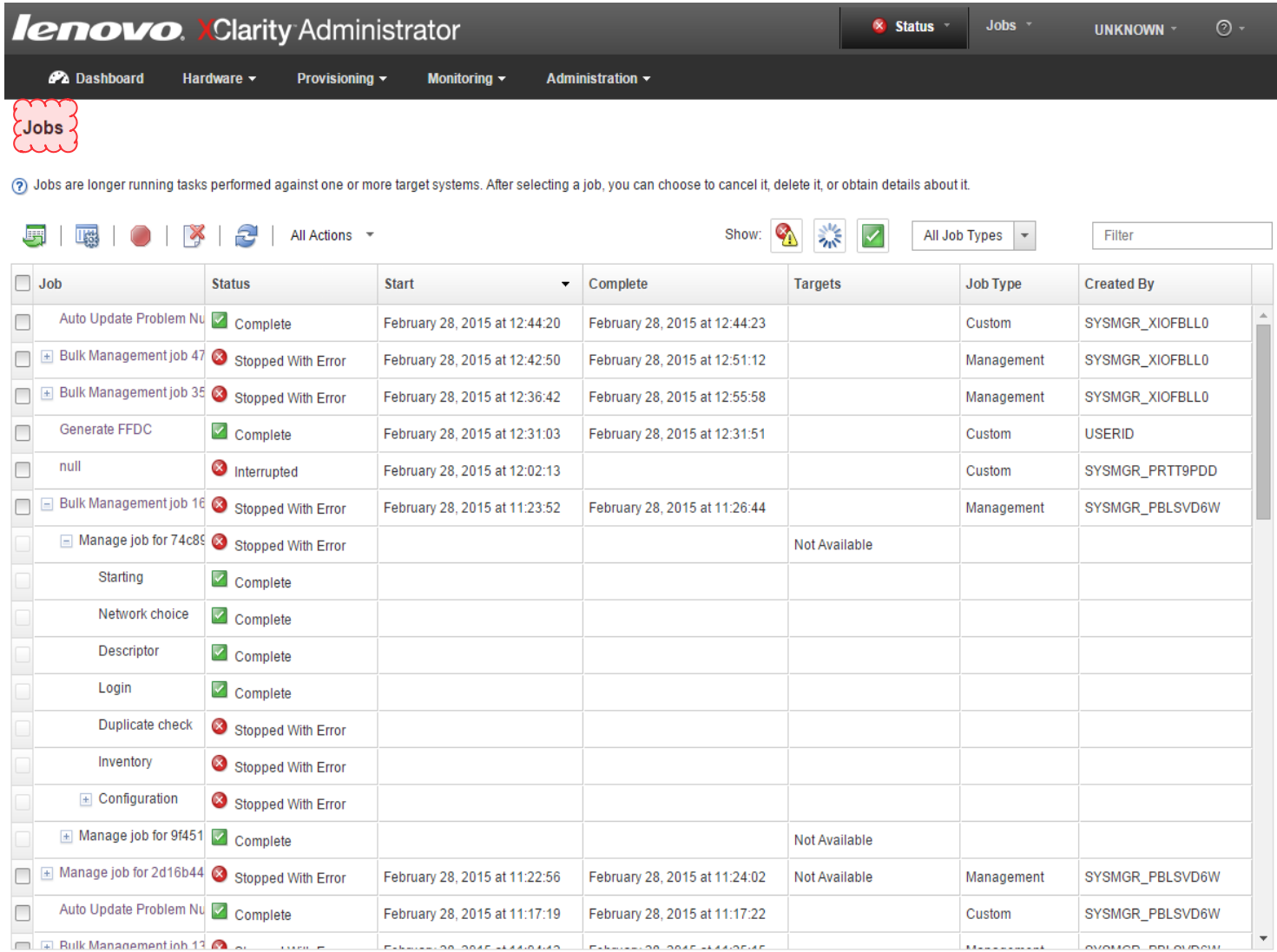
Content Overview: Resource Views → Details

- Contains full list of available node information
  - Where customers will likely start triage for hardware issues.
- Every detail xHMC collects on nodes can be
- found within
- Multiple content sections available
  - Summary: top level details
  - Inventory details: details on network, processors, memory, firmware, etc...
  - Alerts: affecting this node
  - Event log: affecting this node
  - Jobs: targetting this node
  - Lightpath: full LED details
  - Power and Thermal: details and historical graphs
  - Configuration (config patterns)
  - Feature on demand keys



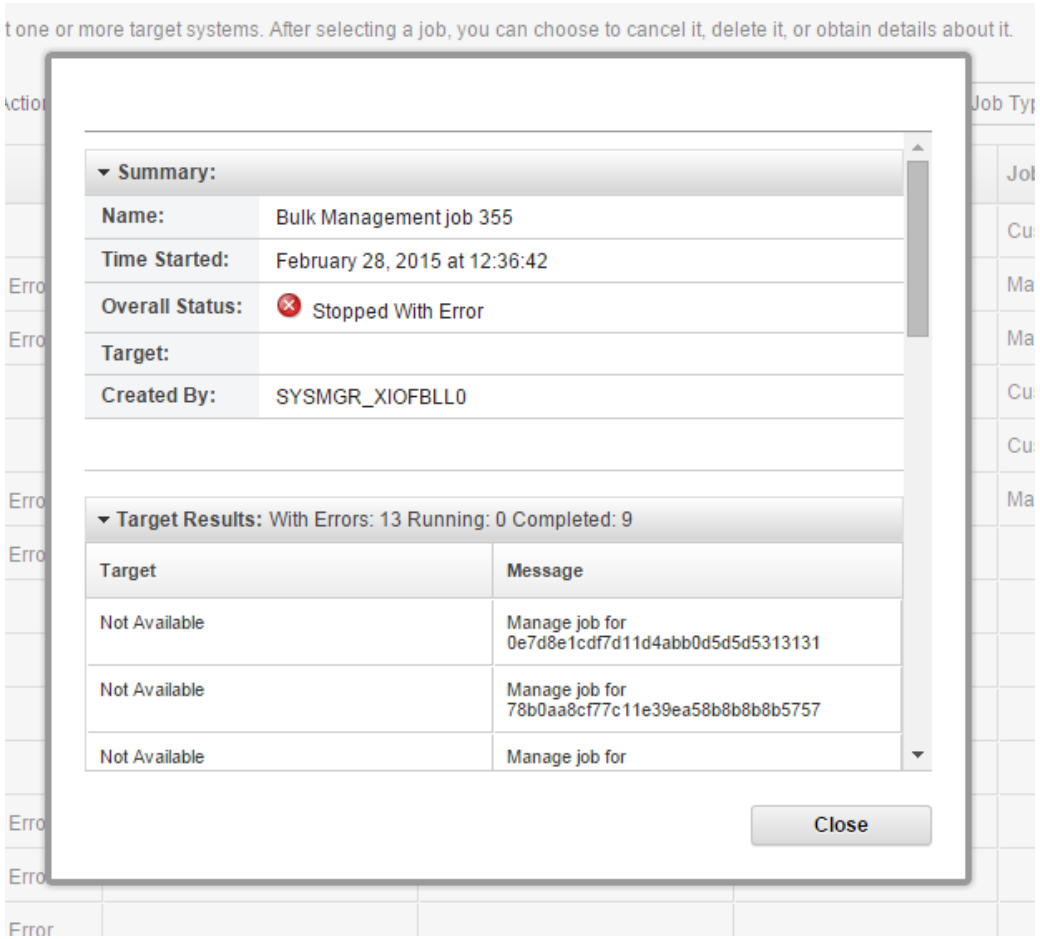
Content Overview: Jobs

- Menu → Monitoring → Jobs
- Most activities that you initiate are
- tracked as jobs
- All jobs do run even if the UI that initiated
- them is no longer up
- The jobs pod and this view can be
- referenced for status on active or recent
- jobs
- You can expand jobs for more details
- Note: This page is not auto-refreshed.



Content Overview: Jobs Details

- Clicking a job from the jobs view or pod will open its details
- Note something like the bulk management job will list its
- results per target.
- At the bottom of the dialog is the log for the job.
- This panel refreshes regularly to update with latest job
- progress



Possible Issues

- Unresponsive UI
  - Session ends in odd way may break the redirec to to login
  - Unknown issue causing UI to hang
  - SOLUTION: Refresh the browser page (F5, depending on the platform and browser)
  - SOLUTION +1: Clear browser cache and refresh the browser page
- BTW: For **any** odd UI issue, its always worth tyring to refresh the browser to see if that resolves issue
- And if that fails, clear cache and refresh browser
- If a page should fail to load or produce and error message when loading...
  - Sometimes navigating to node details will produce failure if the node no longer exists
    - Loaded from bookmark? Node just removed from inventory …?
  - Other failures would require logs and additional details
  - Some page load failures could be an exception caught when the page loads (again **shouldn't** occur... would need additional logs

Console FFDC data

Console log/FFDC data from user's browsers is periodically uploaded to the LXCA server

A couple times per minute, and at error time

This can be seen in the WebLog.txt file in the Appender folder of the FFDC .zip file

LDAP.txt	TEXT	201 KB	Fri May 23 14:45:...
WebLog.txt	TEXT	211 KB	Fri May 23 14:45:...
Servlets.txt	TEXT	219 KB	Fri May 23 14:45:...
DCS.txt	TEXT	156 KB	Fri May 23 14:45:...
Inventory.txt	TEXT	259 KB	Fri May 23 14:45:...
FlexCat.txt	TEXT	400 KB	Fri May 23 14:45:...
Services.txt	TEXT	159 KB	Fri May 23 14:45:...
Server.txt	TEXT	3 KB	Fri May 23 14:45:...

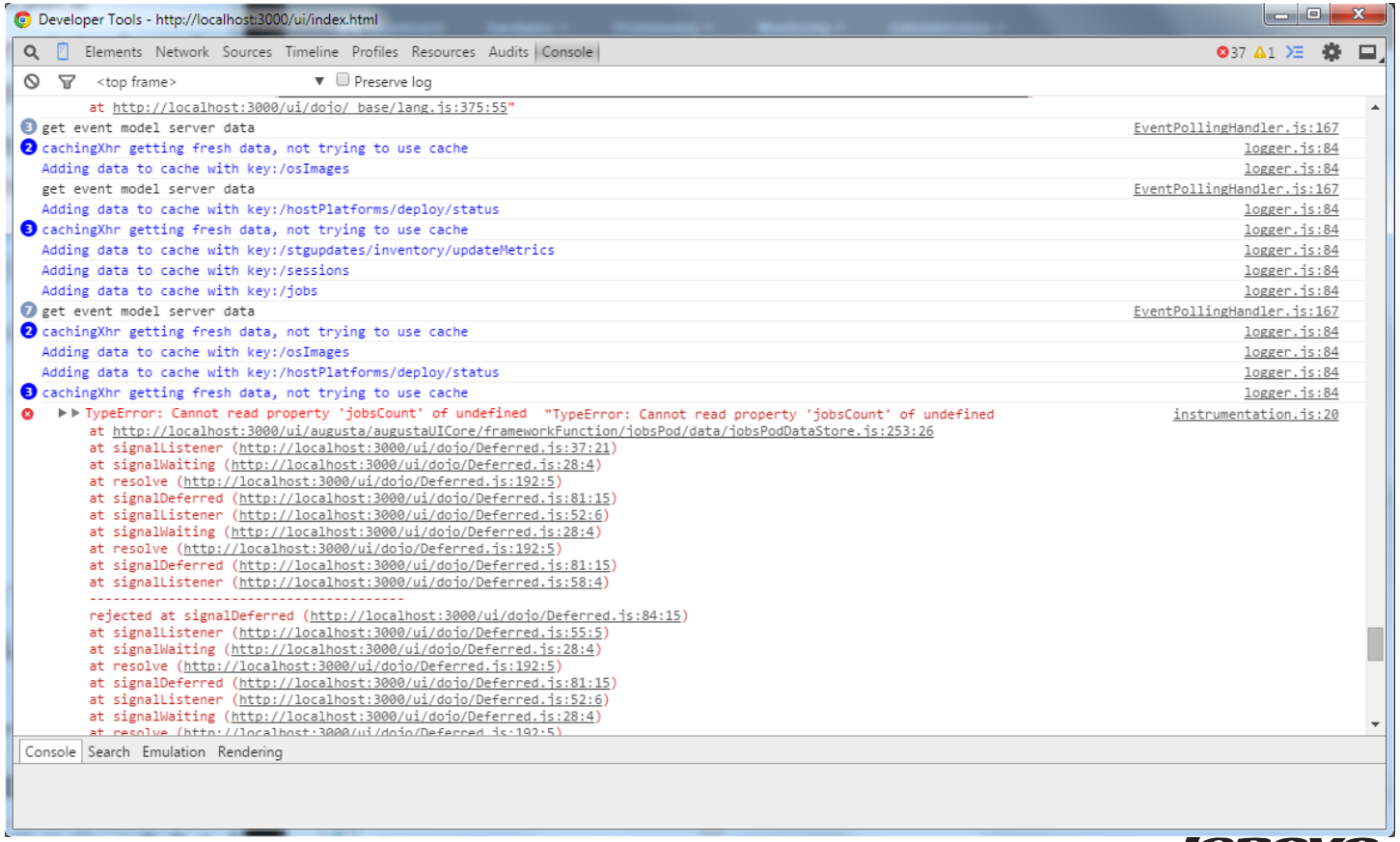
Log File	Log time	Thread caller	Log type	Log message
WebLog.txt	Thu May 22 18:32:11 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801531157 DataReader...
WebLog.txt	Thu May 22 18:32:11 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801531157 DataReader...
WebLog.txt	Thu May 22 18:32:11 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801531157 key found in...
WebLog.txt	Thu May 22 18:32:11 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801531157 key found in...
WebLog.txt	Thu May 22 18:32:11 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801531158 DataReader...
WebLog.txt	Thu May 22 18:32:11 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801531158 DataReader...
WebLog.txt	Thu May 22 18:32:11 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801531158 key not foun...
WebLog.txt	Thu May 22 18:32:11 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801531158 key not foun...
WebLog.txt	Thu May 22 18:32:11 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801531159 Adding data...
WebLog.txt	Thu May 22 18:32:11 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801531159 Adding data...
WebLog.txt	Thu May 22 18:32:12 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801532358 Entering: di...
WebLog.txt	Thu May 22 18:32:12 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801532358 Entering: di...
WebLog.txt	Thu May 22 18:32:12 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801532358 dijit._Contai...
WebLog.txt	Thu May 22 18:32:12 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801532358 dijit._Contai...
WebLog.txt	Thu May 22 18:32:12 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801532367 Entering: a...
WebLog.txt	Thu May 22 18:32:12 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801532367 Entering: a...
WebLog.txt	Thu May 22 18:32:12 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801532371 Exiting: aug...
WebLog.txt	Thu May 22 18:32:12 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801532371 Exiting: aug...
WebLog.txt	Thu May 22 18:32:12 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801532371 Exiting: dijit....
WebLog.txt	Thu May 22 18:32:12 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801532371 Exiting: dijit....
WebLog.txt	Thu May 22 18:32:12 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801532377 taskObject ...
WebLog.txt	Thu May 22 18:32:12 CDT 2014	[-1114695875@qtp-1136789200-6]	DEBUG	com.ibm.ofm.server.servlets.logging.LoggingConsoleServlet:handlePost 1400801532377 taskObject ...

Console FFDC data - continued

- It's worth noting the weblog file contains all logging input from all UI sessions. So there could be crosstalk.
- Also, as stated previously, there are cases where some logging or important information just doesn't make its way back to this log file
  - Network hiccups
  - Exceptions that kill the client script execution thread.

UI Triage: UI Logging (in browser)

- One unique feature of the user interface, is that every user (and you) has access to debugging tools build into most browsers
  - IE (F12 -> dev tools), Chrome (More tools -> dev tools), Firefox (developer → Web console)
- Our logging interface logs to BOTH to the browsers console log and back to the server weblog.txt
- Note: Due to the way client code works,
- sometimes the browser logging will
- contain more information.
  - Some UI failures can prevent logging from being sent to server weblog
  - Browser also has exception stacks
  - Line of code failing
  - Easier to read
  - So, are times when having access to browser log is more helpful
  - When reproducing issues in house, worth having dev tools open



UI Triage: More on browser tools

- When re-creating possible UI issues in house, suggested to utilize the dev tools available in the browser.
  - Look for exception in the console log (usually red text).
    - This usually indicates the UI code failed somehow.
- It's also possible to leverage browser network monitoring to capture REST API traffic to inspect data actually sent to the UI.
  - Particularly useful when determining if an issue is because of 'bad' data sent to the UI
- Screenshots are also very useful for triage for any UI type issue.


THANK YOU GRAZIE MERCI DANKE GRAZIAS 謝謝 СПАСИБО
GRACIAS OBRIGADO ありがとう DANK TAKK BEDANKT DAKUJEM