HMC Unified Resource Manager Web Services API and User Interface Hints and Tips

SHARE in Atlanta – Session 10847
March 2012

Joe Gdaniec
IBM Corporation - HMC/SE Development
gdaniec@us.ibm.com

File Date: 2012-03-13

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

APPN*
CICS*
DB2*
DB2 Connect
e-business logo*
Enterprise Storage Server*
ESCON*
FICON
FICON Express
GDPS*
Geographically Dispersed Parallel Sysplex
HiperSockets
HyperSwap
IBM
IBM eServer
IBM ^*
IBM logo*
IMS
Infoprint*
Language Environment*
MQSeries*
Multiprise*
NetView*
On demand business logo
OS/2*
OS/390*
Parallel Sysplex*
POWER
PR/SM
Processor Resource/Systems Manager
pSeries*
RACF*
Resource Link
RMF
S/390*
S/390 Parallel Enterprise Server
Sysplex Timer*
TotalStorage*
VM/ESA*
VSE/ESA
VTAM*
WebSphere*
z/Architecture
z/OS*
z/VM*
zSeries*
zSeries Entry License Charge

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

Red Hat, the Red Hat "Shadow Man" logo, and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc., in the United States and other countries.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

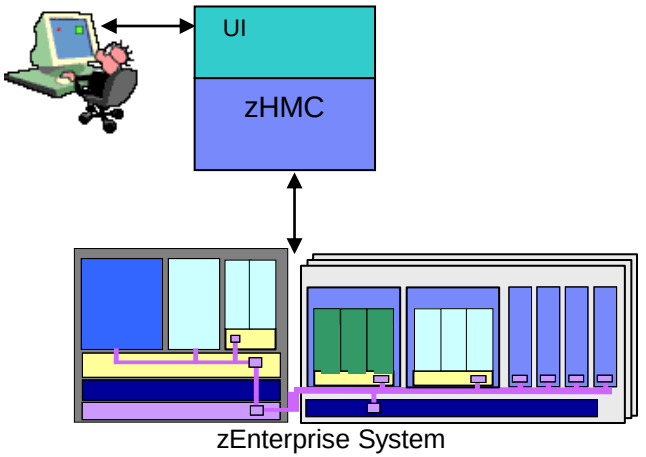
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Please see http://www.ibm.com/legal/copytrade.shtml for copyright and trademark information.

Agenda

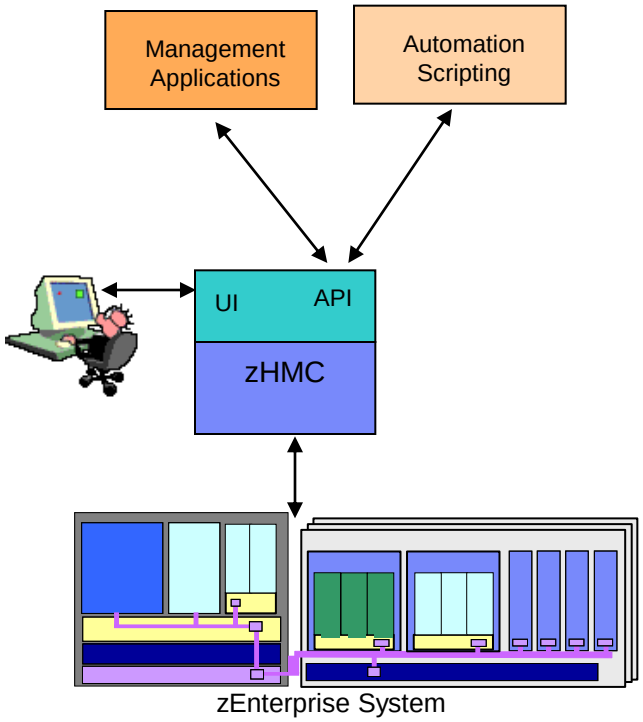
Unified Resource Manager GA1 (HMC 2.11.0) Recap

- New virtualization and ensemble management function in HMC, SE, zBX blades
- Establishes new managed resource types, controls and policy
  - zBX and blades
  - Virtual Servers and Hypervisors
  - Virtual networks on the IEDN
  - Workloads
  - Performance policies on Workloads
- HMC takes on role of the management appliance and access point for zManager
- Unified Resource Manager function surfaced through HMC UI, but this is just a first step



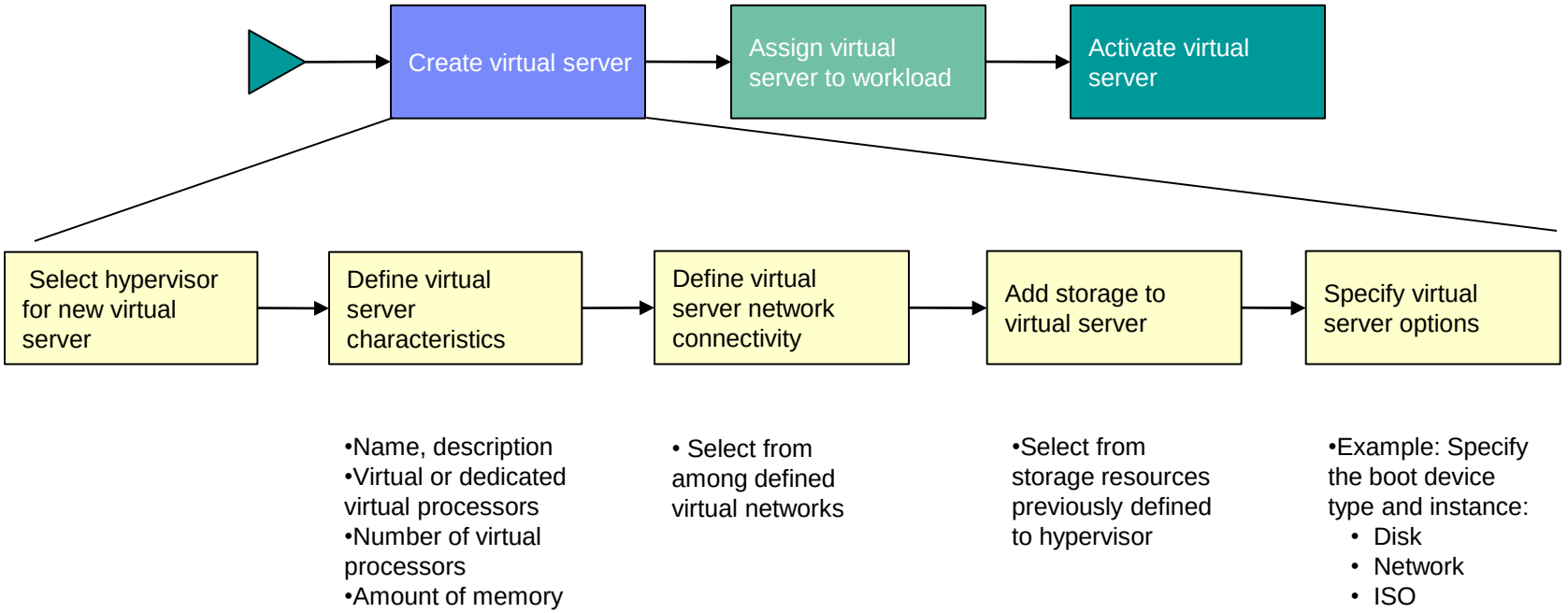
GA2 (HMC 2.11.1) Focus: Enable Management Tools

- GA2 release now allows API access to Unified Resource Manager function
- New HMC Web Services API allows programmatic access to the same underlying Unified Resource Manager function as is accessed via the HMC UI
  - Same resource types, instances and policy
  - API functions corresponding to views and tasks in the UI
    - Listing resource instances
    - Creating, changing, deleting resource instances
    - Operational control of resource instances
    - Etc.
- Goal: Enable management of Unified Resource Manager from external (to HMC) tools
- Priority scenarios: Discovery, Monitoring and Provisioning use cases



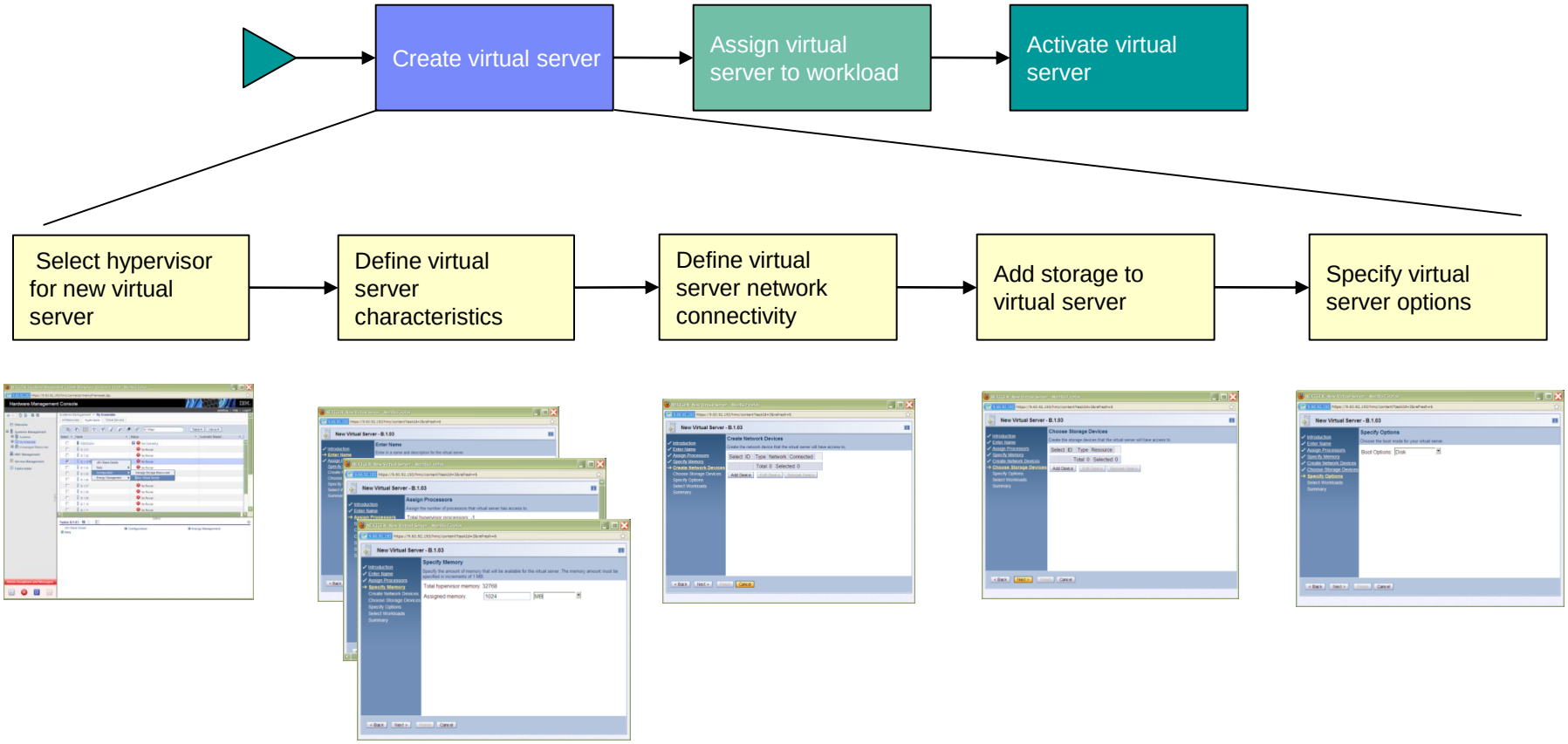
HMC WS API and UI Provide Same Level of Function

- Example: Creating a Virtual Server on an IBM Blade
- Regardless of the interface used, this is accomplished through a series of steps:



HMC WS API and UI Provide Same Level of Function (con't)

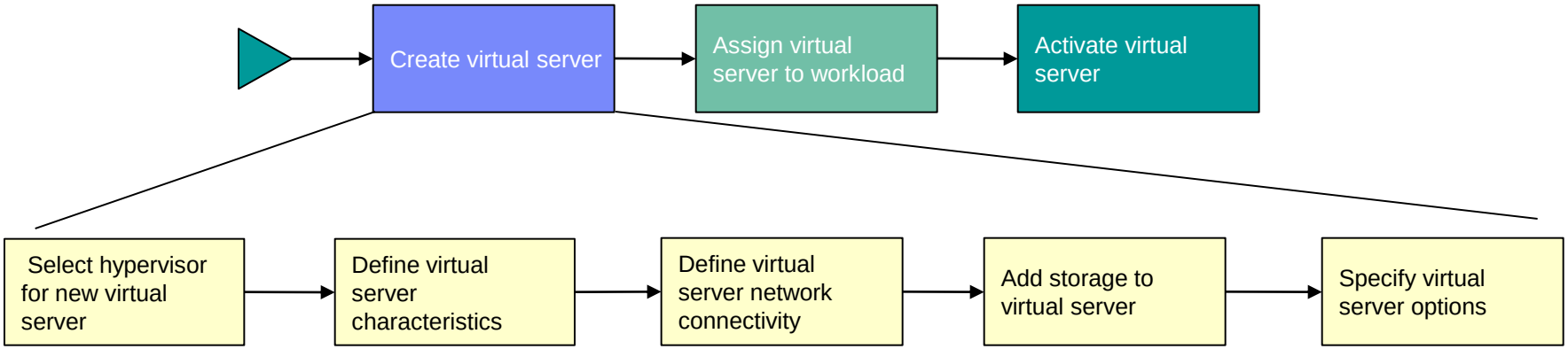
- HMC UI: Steps are accomplished using panels in a wizard-style task



HMC WS API and UI Provide Same Level of Function (con't)

- WS API: Steps are accomplished by calling management primitives of the API

Note: Function names listed below are conceptual, not the actual API syntax



- Call List Hypervisors function to obtain a list of hypervisors
- <Invoking application selects desired hypervisor>

- Call Create Virt Server function specifying selected hypervisor as target and basic VS parameters to get base VS created

- Call List Virt Networks function to obtain current virtual networks
- <Select network>
- Call Add Virt Adapter function specifying new VS as target and virtual network parameters

- Call List Stg Resourcces function to obtain list of available volumes
- <Select volume>
Call Add Virt Disk function specifying new VS as target and selected storage resource

- <Select boot device>
- Call Update Virt Server function to set boot device

Summary of API Functional Scope

- Core (Traditional) Entities:
  - List, Get/Set properties
  - Start/Stop/Restart/etc.
  - For CPC, Image, LPAR, Groups, Capacity records, and Console (HMC itself)
- Ensemble
  - List, Get/Set properties
  - Add/Remove CPC members
- Workloads
  - List. Create, Delete
  - Get/Set properties
  - Add/Remove virtual-server members
  - Create/Delete performance policies
  - Get performance report data
- Virtual Networks:
  - List, Create, Delete
  - Get/Set Properties
  - Recovery actions
- Virtualization Hosts (X, P, zVM)
  - List, Get/Set properties
  - Start/Stop
- Virtual Servers (X, P, zVM):
  - List. Create, Delete
  - Get/Set properties
  - Start/Stop
- Storage:
  - Define, List
  - Assign to virtualization hosts
  - Assign to storage groups
- zBX Infrastructure (BladeCenters, Blades):
  - List, Get/Set Properties
  - Get/Set energy management modes
- Service Oriented Functions
  - Metrics retrieval
  - Inventory

HMC Web Services API Characteristics

- HMC Web Services API is a new API implementation in the HMC
  - Includes existing SNMP/CIM function plus new Unified Resource Manager capabilities
  - This new API is the focus for future evolution
  - Existing SNMP and CIM APIs remaining in place with their existing capabilities, may be extended on a case-by-case basis

- Design based on current industry design practices
  - Requests and responses structured as web services based on REST design patterns
  - Data is represented in Javascript Object Notation (JSON)
  - Status and property change notifications delivered via JMS
  - HMC provides an embedded JMS broker configured to support API specific use

- HTTP over TCP/IP Sockets is underlying network transport, SSL for connection security

- Fully documented and supported for customer / ISV use:
  - SC27-2616-00 *System z Hardware Management Console Web Services API*
  - Available on IBM ResourceLink: http://www.ibm.com/servers/resourcelink

    Then navigate: Library / zEnterprise 196

What is a REST-oriented Web Service?

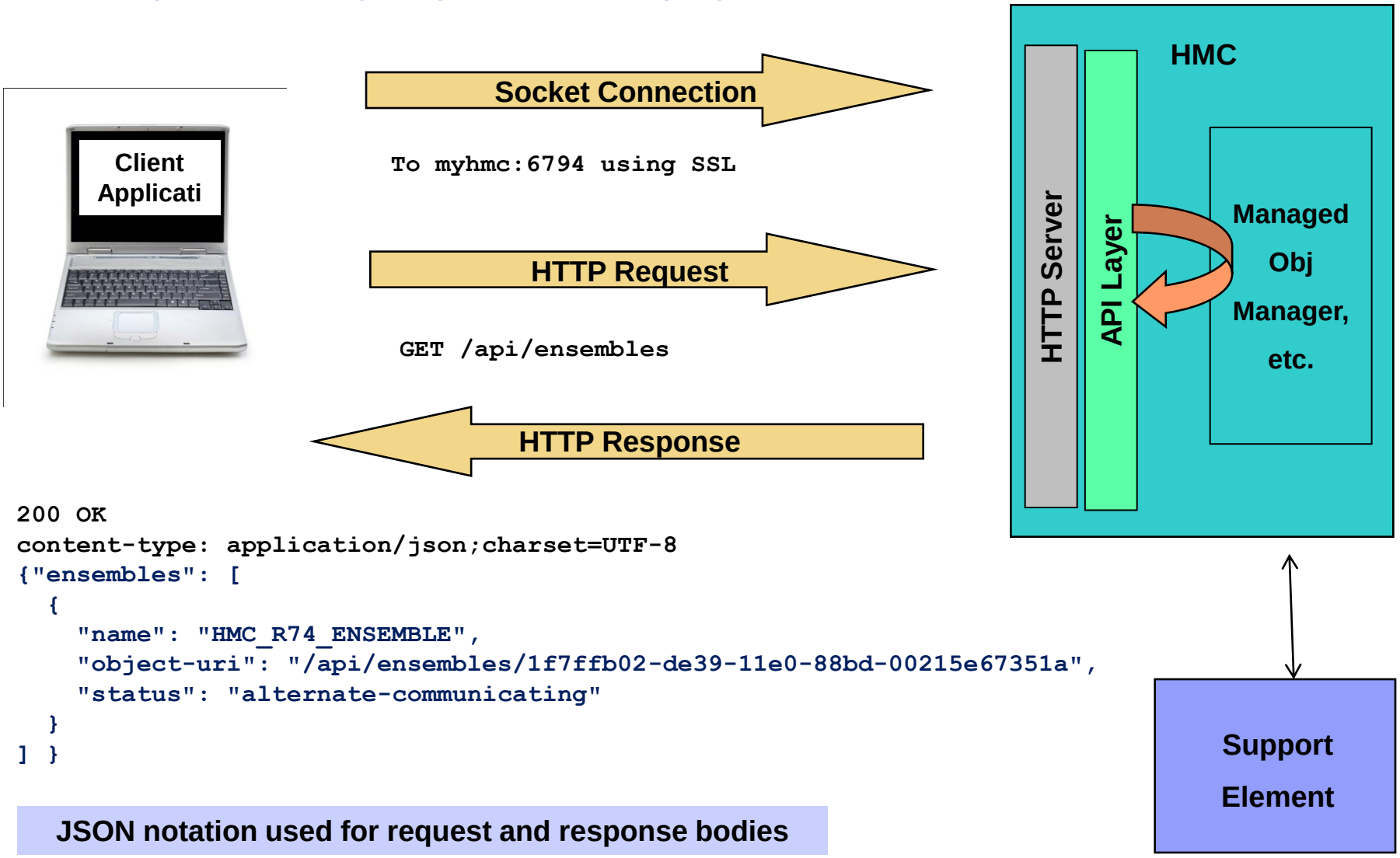
- *REST* = **Re**presentational **S**tate **T**ransfer
  - A style of software interface design
  - Simplifies client – server interactions
  - Introduced in 2000 by Roy Fielding (phD dissertation)
  - Used widely in today's world wide web services
  - Based on HTTP protocol

- Fundamentals
  - All actions are against a specific resource
  - The resource instance is identified in the HTTP URI for the web service call
  - Type of operation on that resource is specified by using HTTP "method"

– Standard HTTP methods apply across all resources
  - GET – Collect information about a resource
  - POST – Create a new resource, perform other type of operation
  - PUT – Complete update of a resource (all properties)
  - DELETE – Delete a resource

- Generic Examples:
  - To get list of virtual servers: **`GET /api/virtual-servers`**
  - To get information about a virtual server: **`GET /api/virtual-servers/1234`**

API Request Flow (Simplified Example)

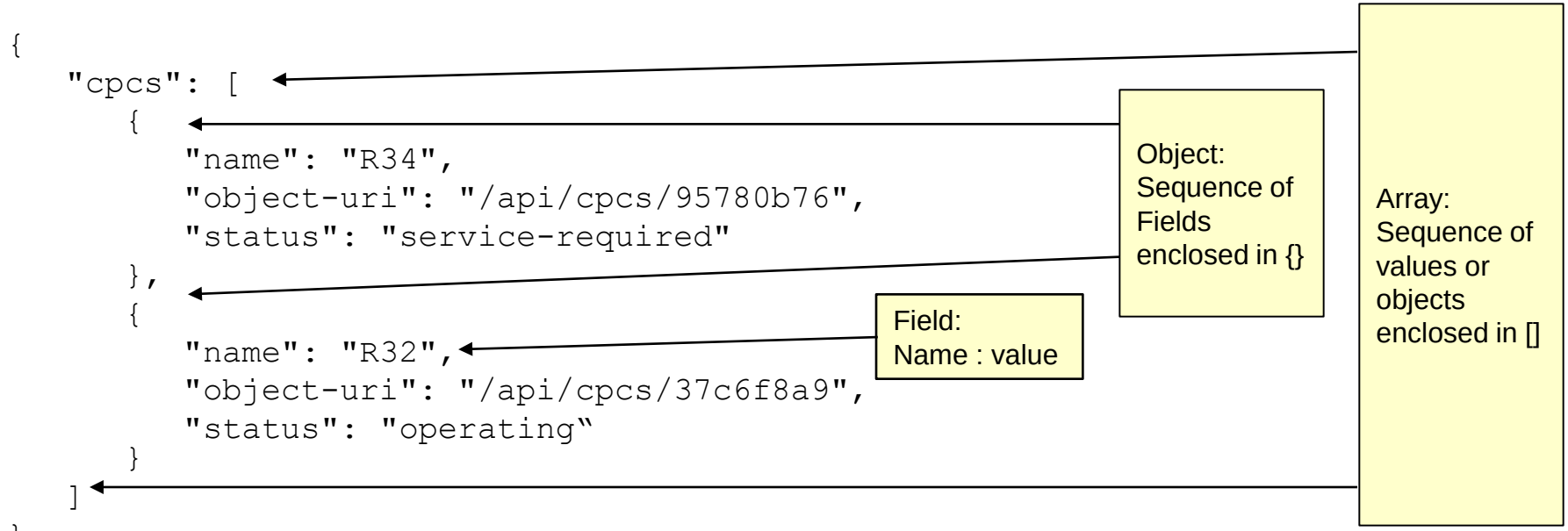



```
200 OK
content-type: application/json;charset=UTF-8
{"ensembles": [
  {
    "name": "HMC_R74_ENSEMBLE",
    "object-uri": "/api/ensembles/1f7ffb02-de39-11e0-88bd-00215e67351a",
    "status": "alternate-communicating"
  }
] }
```

JSON notation used for request and response bodies

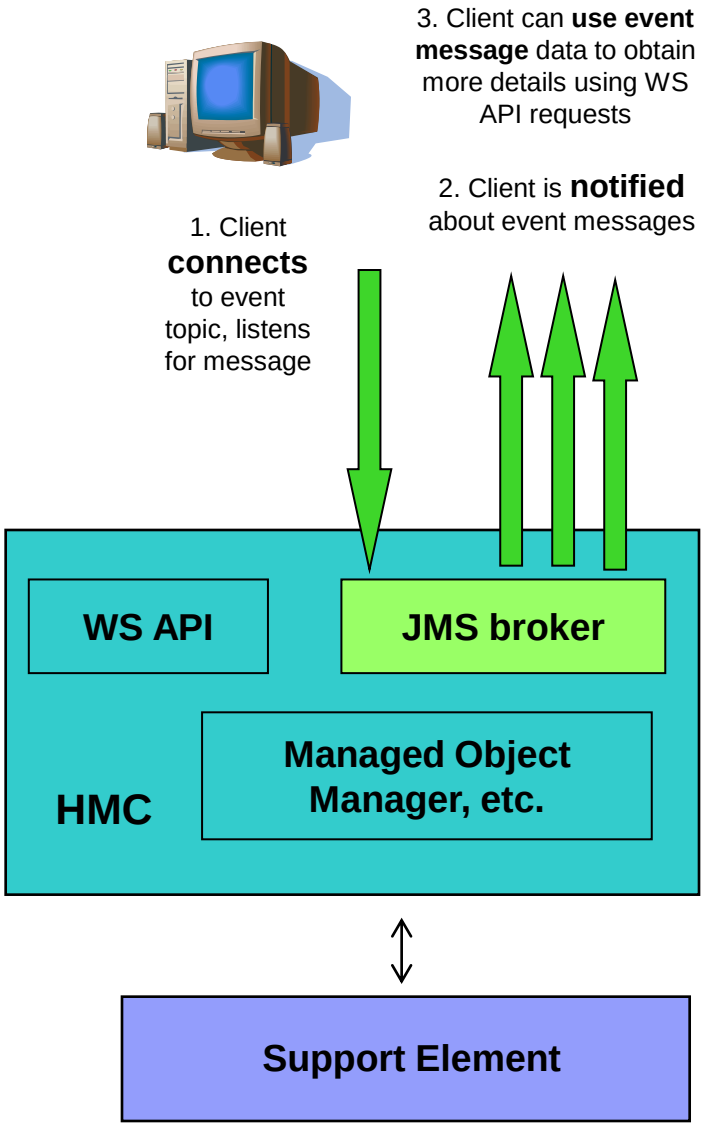
Javascript Object Notation (JSON)

- Lightweight data interchange format for use between applications
- Much simpler than XML, but still expressive enough
- Used by Google, Yahoo, Web 2.0 applications etc.
- Syntax and tutorials available at www.json.org
- JSON parsers widely available (eg. At json.org)
- Becoming the standard notation used with REST-style APIs



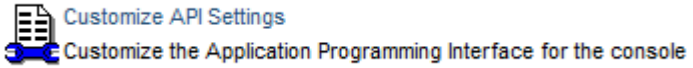
What is Asynchronous Notification?

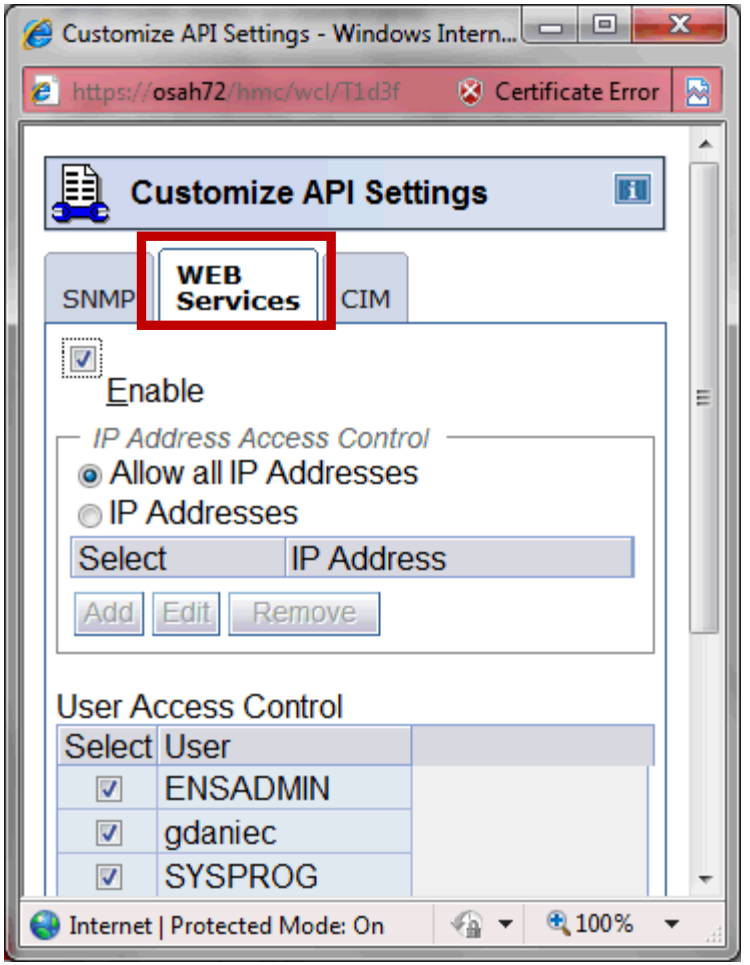
- Asynchronous Notification
  - WS APIs require remote client to repeatedly pull (poll) from HMC server to keep data current
  - Asynchronous notification prevents need for constant polling from remote client.
  - Allows HMC to push a notification to a remote client about events or state changes on the resource being managed by the HMC/SE e.g. a server is being deactivated
- JMS: Java Message Service
  - Used for asynchronous notification
  - JMS is an industry standard for messaging, based on J2EE
  - HMC embeds Apache ActiveMQ as its JMS provider
  - Non-Java clients can connect to broker as well, using STOMP protocol



Web Services API Enablement

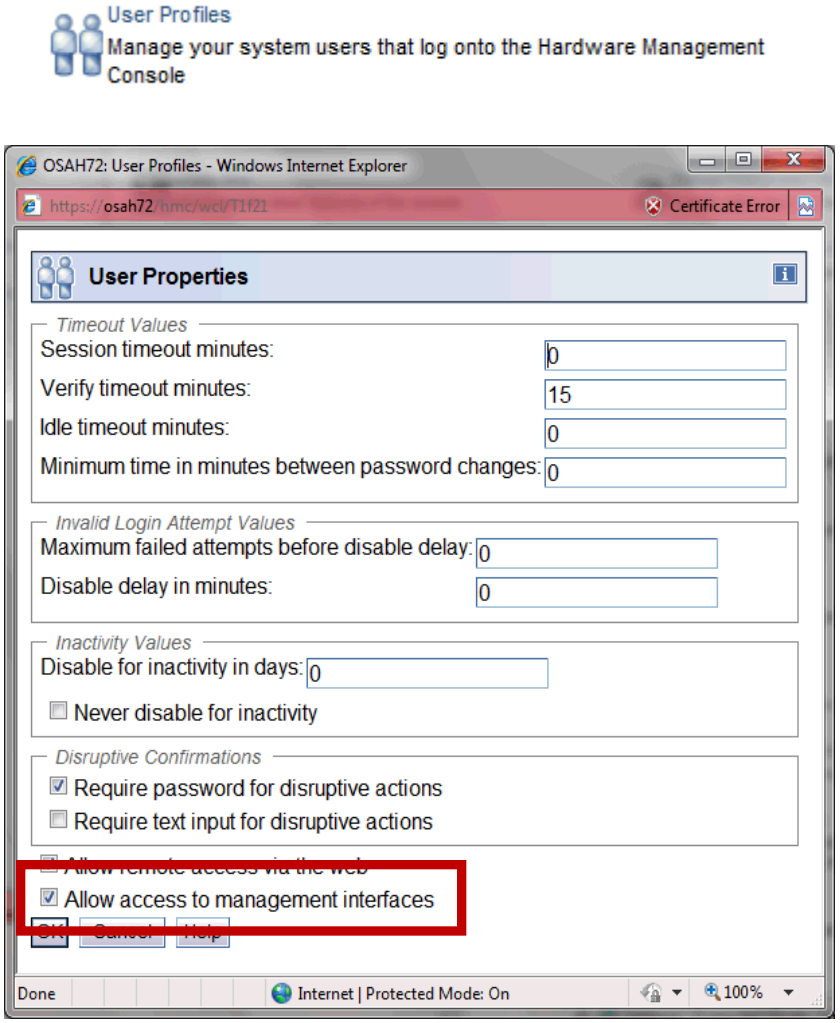
- WS API is Disabled by default
- Overall On / Off switch and other configuration via a new tab in the existing Customize API Settings task
- API enablement is done separately from enabling remove browser access to HMC
- Installation can also optionally control the IP addresses from which API connections can be made
- When enabled HMC listens for API connections on a different TCP/IP port than is used for remote browser access





Web Services API Access Control

- Connection to API requires authentication under an HMC application login identity
  - All connections to the API specify an HMC user name and password
  - HMC local or LDAP validation of user name and password supposed, same as UI
- New User Profile option controls whether an HMC user can use the API or not
- Individual requests are authorized using the HMC's authorization controls
  - Requests always performed under an HMC user context
  - Authorized under the task and resource roles authorized for that user
  - Existing HMC User Profiles or new User Templates used to configure roles and permissions



Client Programming Considerations

- Web Services API design is client platform and client programming language neutral
  - This is a key reason behind the choice of a HTTP/Web Services style
  - Client platform can be Windows, AIX, Linux, zLinux, Mac, or z/OS or…
  - Clients can be written in programming languages like C/C++ or Java, or scripting languages like Python, Perl, etc.
  - No need for install HMC-specific client-side libraries to use APIs
- In choosing a client language, look for the following either as built-ins or available via add-on libraries:
  - Support for HTTP
  - Support for SSL
  - Support for creating and parsing JSON documents
  - (Optionally) Support for JMS connections to ApacheMQ using either OpenWire or STOMP protocols (if asynchronous notification capabilities are to be used)
- Python is a very good choice because support for all of the above is readily available

Getting Started with the API: Some Script Snippets (using Python)

- API is session-oriented: All requests are made in the context of an API session
- Basic pattern for an API client:
  1. Establish SSL socket connection with HMC
  2. Logon to open an API session
  3. Make requests using that API session
  4. Logoff to close the API session
- Python code snippets illustrating these steps follows…

1. Establish an SSL socket connection with the HMC:

```
# Connect to HMC at address <host> with 300 second request timeout
conn = httplib.HTTPSConnection(host, 6794, timeout=300)
conn.connect()
```

Getting Started with the API: Some Script Snippets (using Python)…

2. Log on to the HMC to open an API session:

```
# Log on to HMC as <userid> with password <password>
logon_req = {"userid": userid, "password": password}
req_body = json.dumps(logon_req)
req_hdrs = {"Content-Type": "application/json"}
conn.request("POST", "/api/sessions", req_body, req_hdrs)

response = conn.getresponse()
if response.status != 204:
   # If the response provides a body, always read it.
   resp_body = response.read()
if response.status != 200:
   # Handle failure (eg. wrong psasword)
   raise Exception("Request failed (status: %d)" % response.status)

# Retrieve session id from response for later use
logon_resp = json.loads(resp_body)
session_id = logon_resp["api-session"]
```

Getting Started with the API: Some Script Snippets (using Python)…

3. Make requests using the API session:

```
# Issue request for HMC's properties
# Use the session id for the session we just created
req_hdrs = {"X-API-Session": session_id}
conn.request("GET", "/api/console", None, req_hdrs)

response = conn.getresponse()
if response.status != 204:
   resp_body = response.read()
if response.status != 200:
   raise Exception("Request failed (status: %d)" % response.status)

# Convert result JSON into Python objects for processing
console_props = json.loads(resp_body)
print "HMC name is %s." % console_props["name"]
```

Getting Started with the API: Some Script Snippets (using Python)…

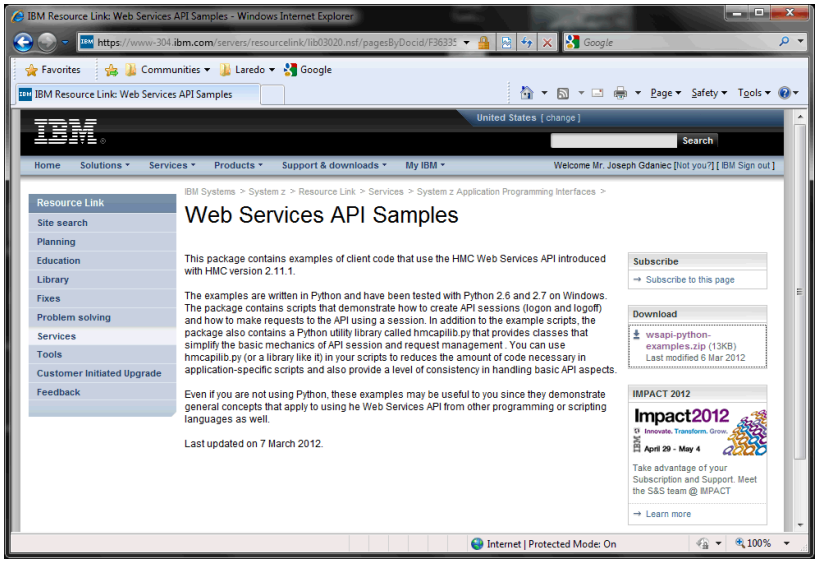
4. Log off from the HMC to close the API session:

```
# Log off from HMC to free session resources
req_hdrs = {"X-API-Session": session_id}
conn.request("DELETE", "/api/sessions/this-session", None, req_hdrs)

response = conn.getresponse()
if response.status != 204:
   resp_body = response.read()
if response.status != 204:
   raise Exception("Request failed (status: %d)" % response.status)

# On success, no response to process from Logoff
```

Getting Started with the API: Samples

- Python sample code is available on ResourceLink:
  http://www.ibm.com/servers/resourcelink

  Then navigate:
  Services / API / Web Services API Samples

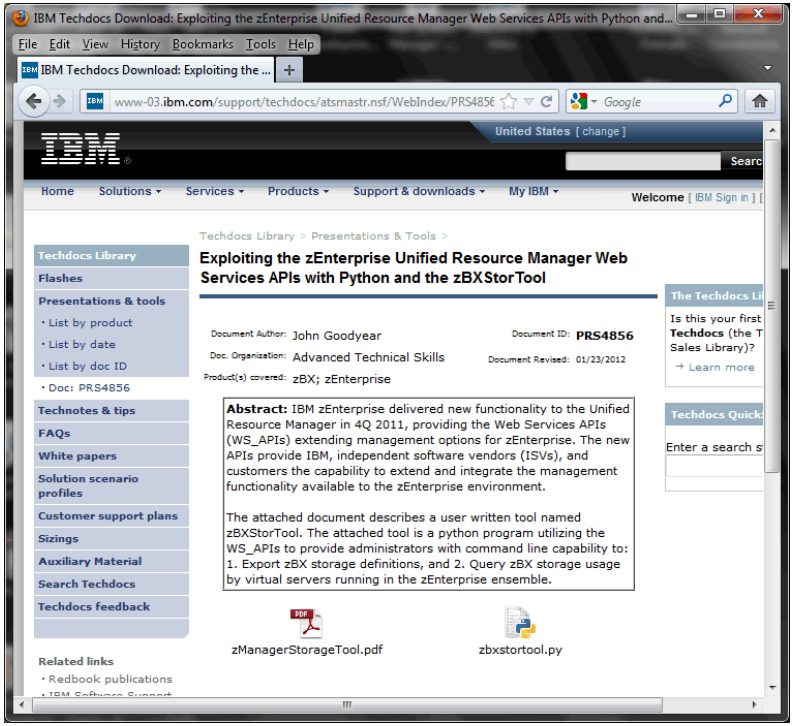


- Package provides simple logon/logoff test script and a script that demonstrates how to create and delete a virtual server using the WS API

- Samples are based on a sample Python utility library (hmcapilib.py) that demonstrates best practices in using the API
  - Handles repetitive aspects of making API requests: logon, logoff, converting to/from JSON, setting HTTP headers, etc.
  - Includes error checking and capturing of error status/reason on errors

Usage Example: zBXStorTool

- Developed by John Goodyear of the IBM Washington Systems Center
- Provides functions that simplify storage administration for zEnterprise zBX:
  - Export storage definitions for entire ensemble or filtered by hypervisor
  - Show relationship between virtual servers and the storage resources they use
- Python script and whiltepaper with client programming hints and tips



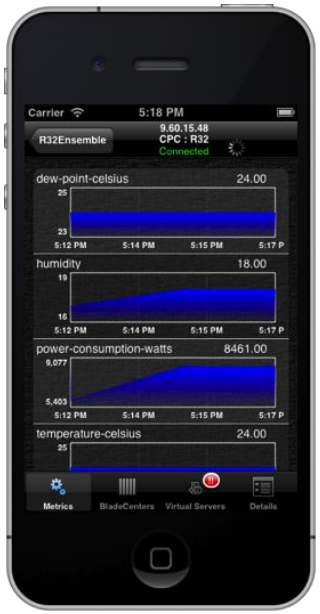
- Provides a more comprehensive example of WS API usage
- Available from the Techdocs Library as document # PRS4856 :

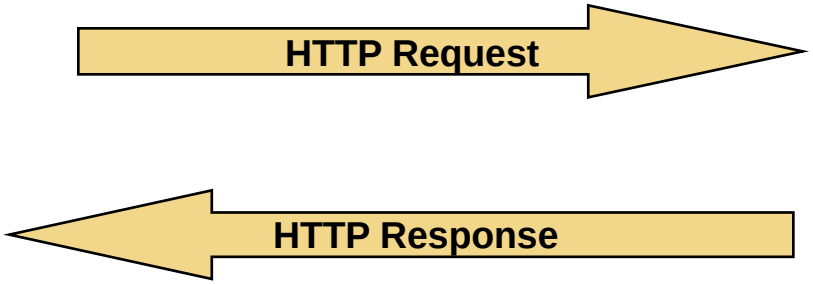
  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS4856

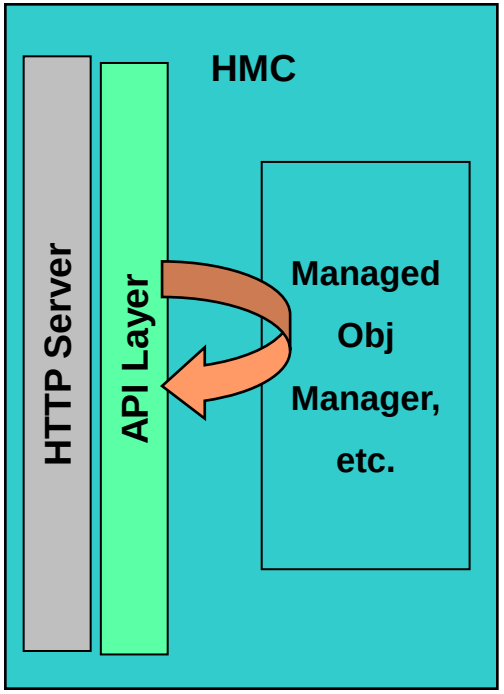
UsageExample: System z Mobile Application Proof of Concept

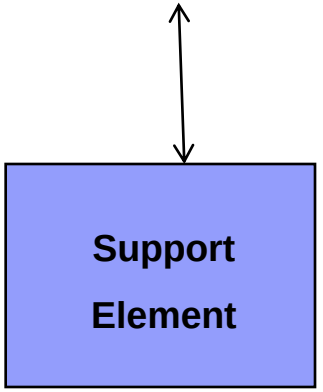
- Proof of Concept Mobile Application for monitoring and controlling a zEnterprise system from a mobile device
- Allows person on the IT floor to grab customizable subset of information about the machine

Mobile Application Proof of Concept – Under the Hood



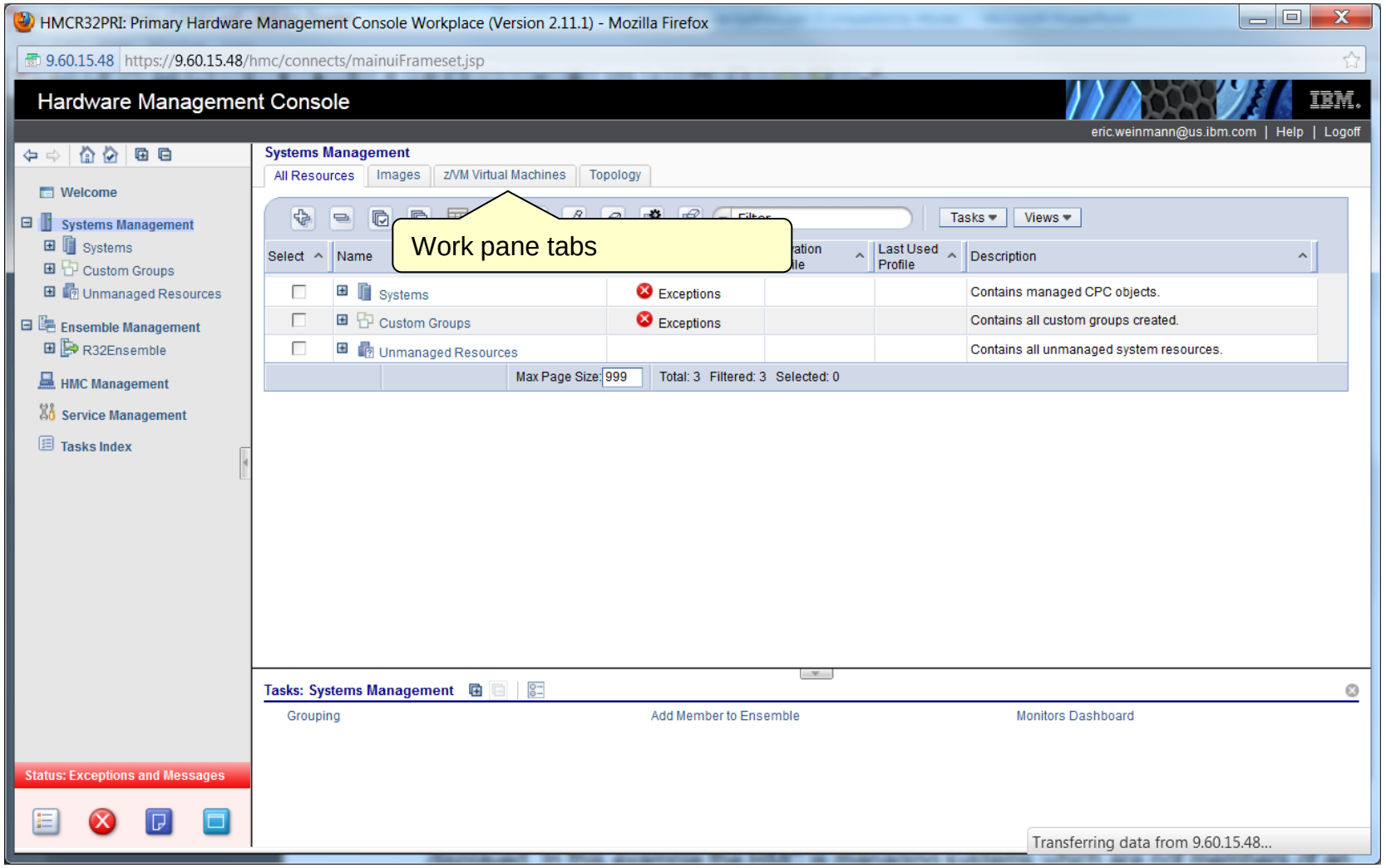




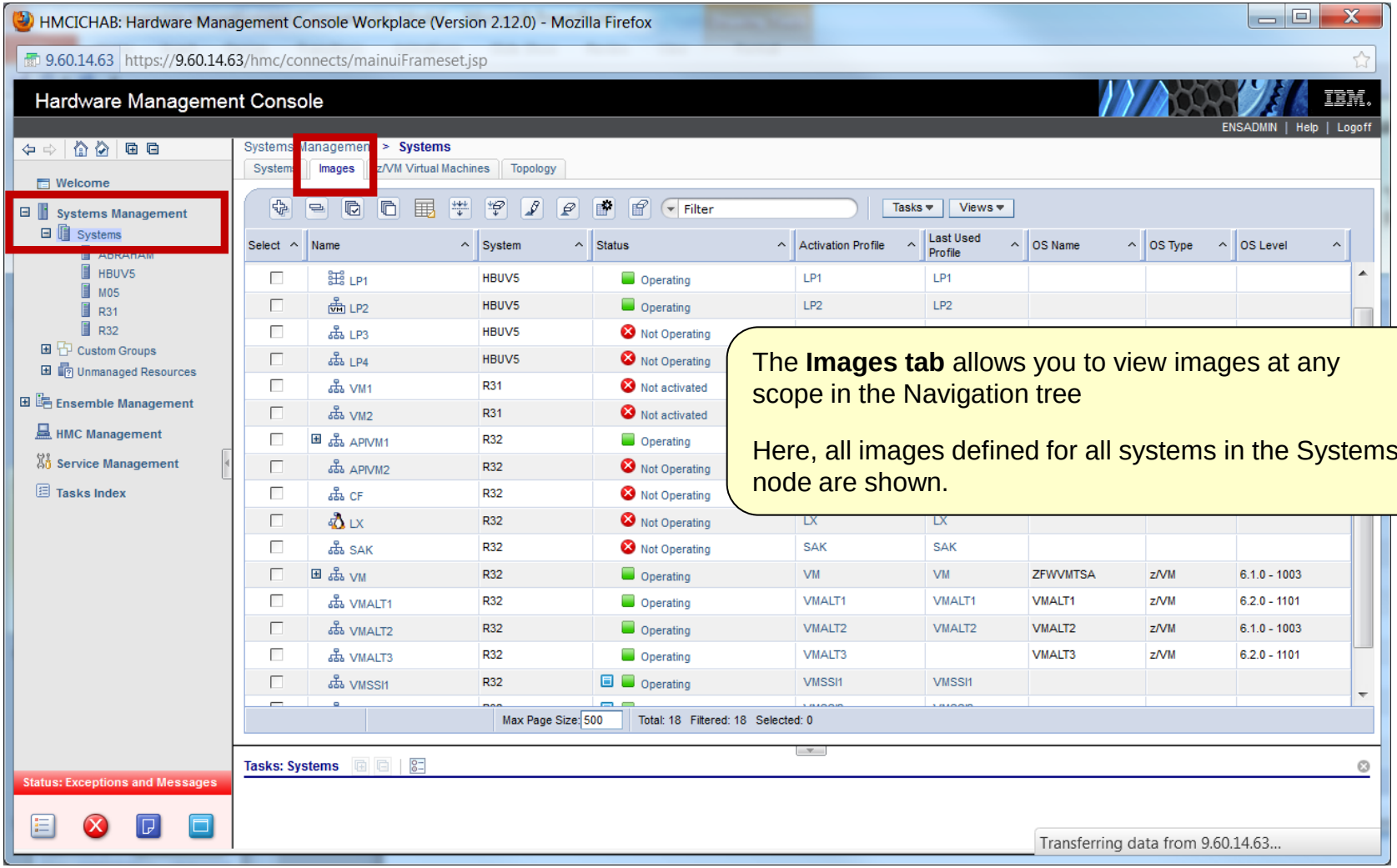


- Mobile device communicates directly with HMC using the new Web Services API. No intermediate management server needed.
- RESTful orientation of the API makes this easy to do using standard application capabilities on these types of devices
- Authentication and authorization enforcement by the API keeps the environment secure

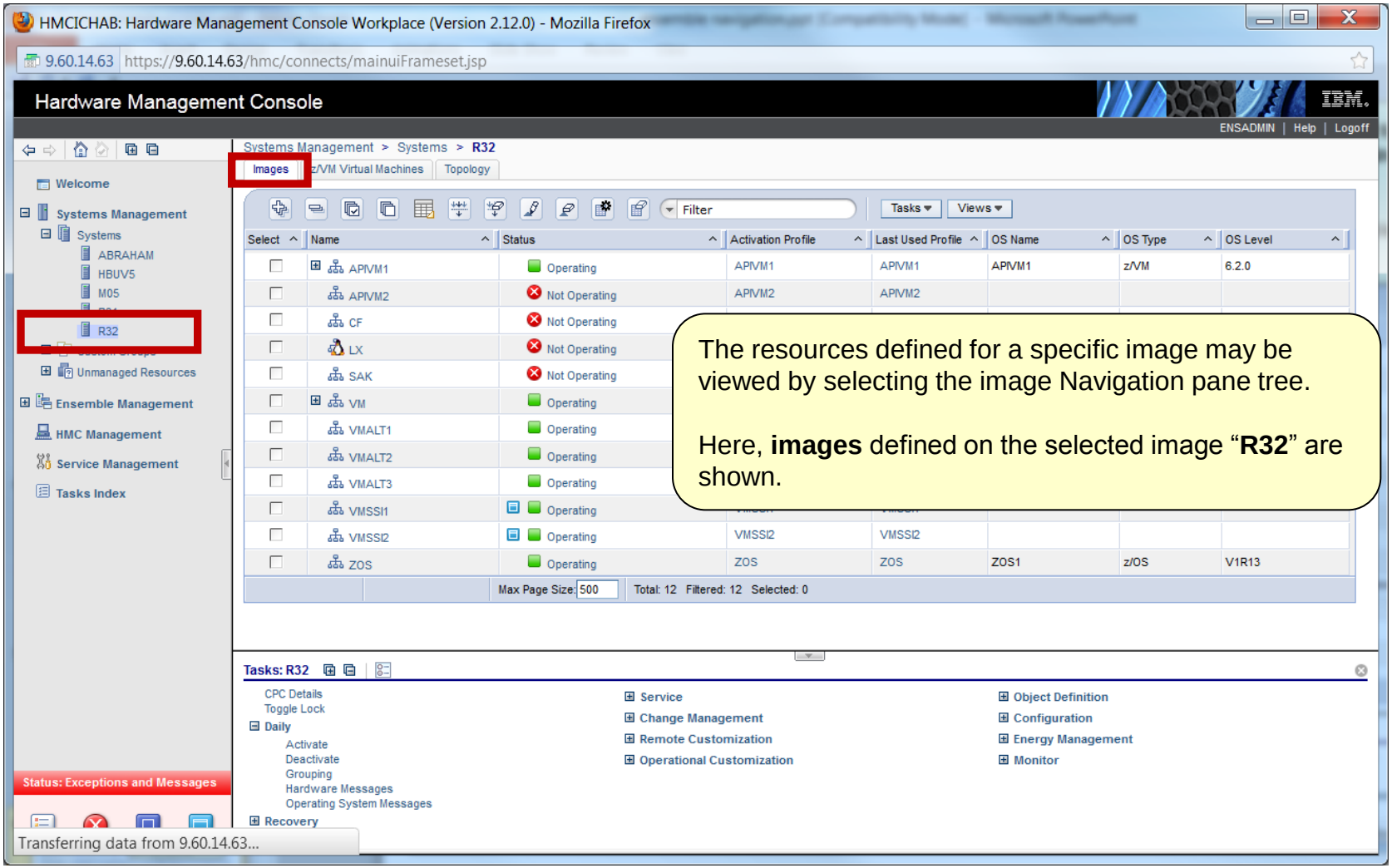
Tree Style Resource Navigation Enhancements



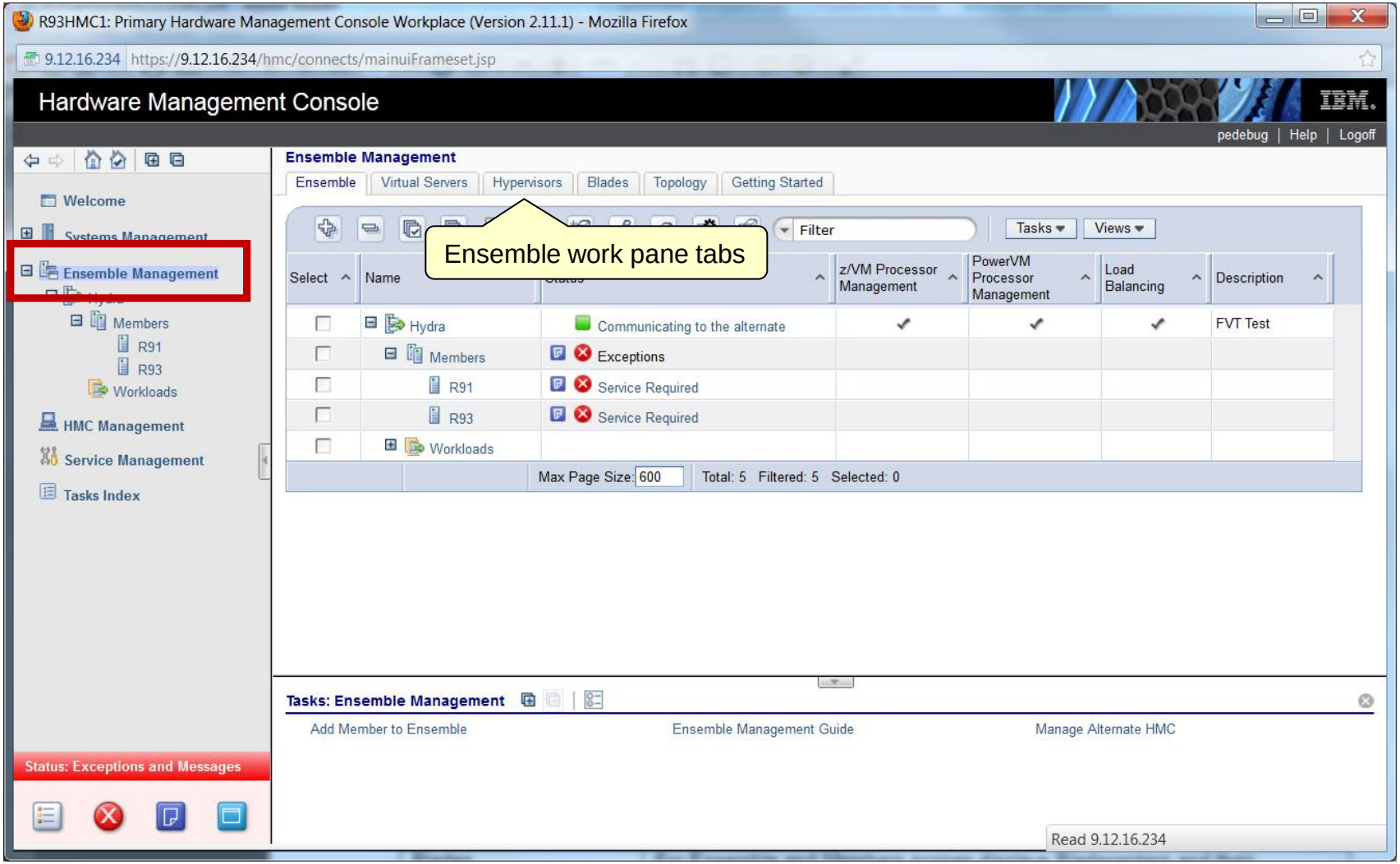
Systems Tabs - Images



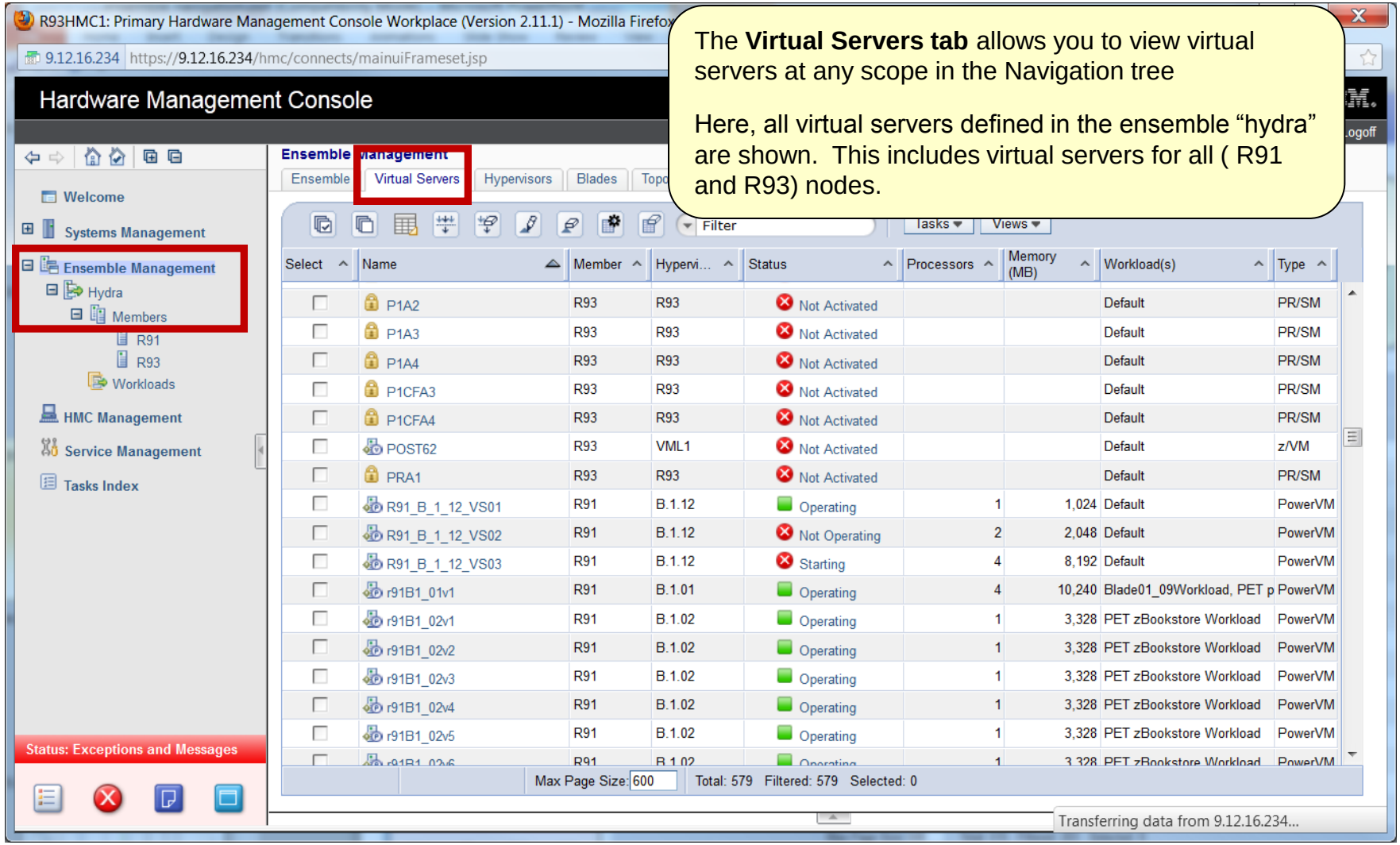
Systems Tabs - Images



Ensemble Tabs



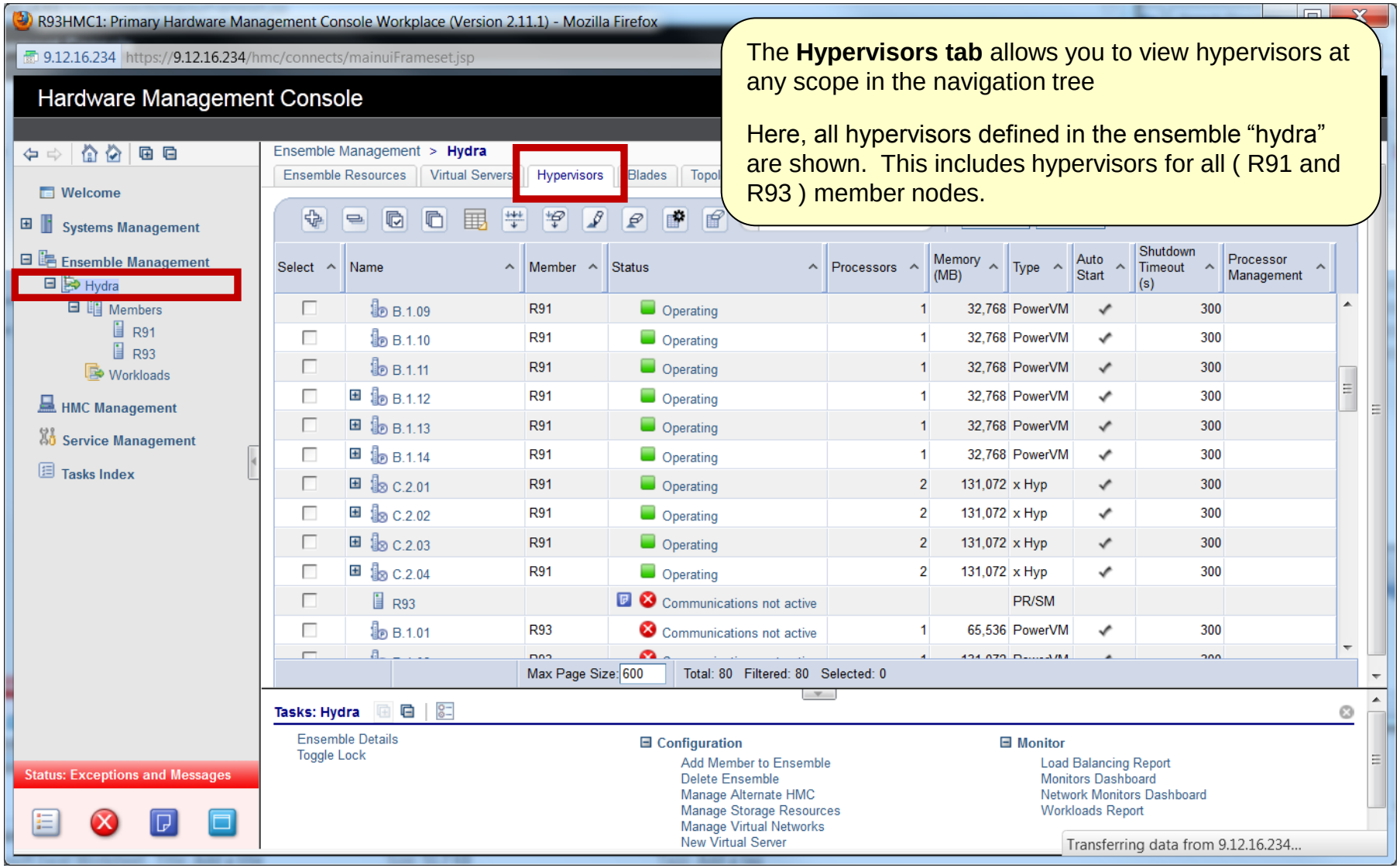
Ensemble Tabs – Virtual Servers



The **Virtual Servers tab** allows you to view virtual servers at any scope in the Navigation tree

Here, all virtual servers defined in the ensemble “hydra” are shown. This includes virtual servers for all (R91 and R93) nodes.

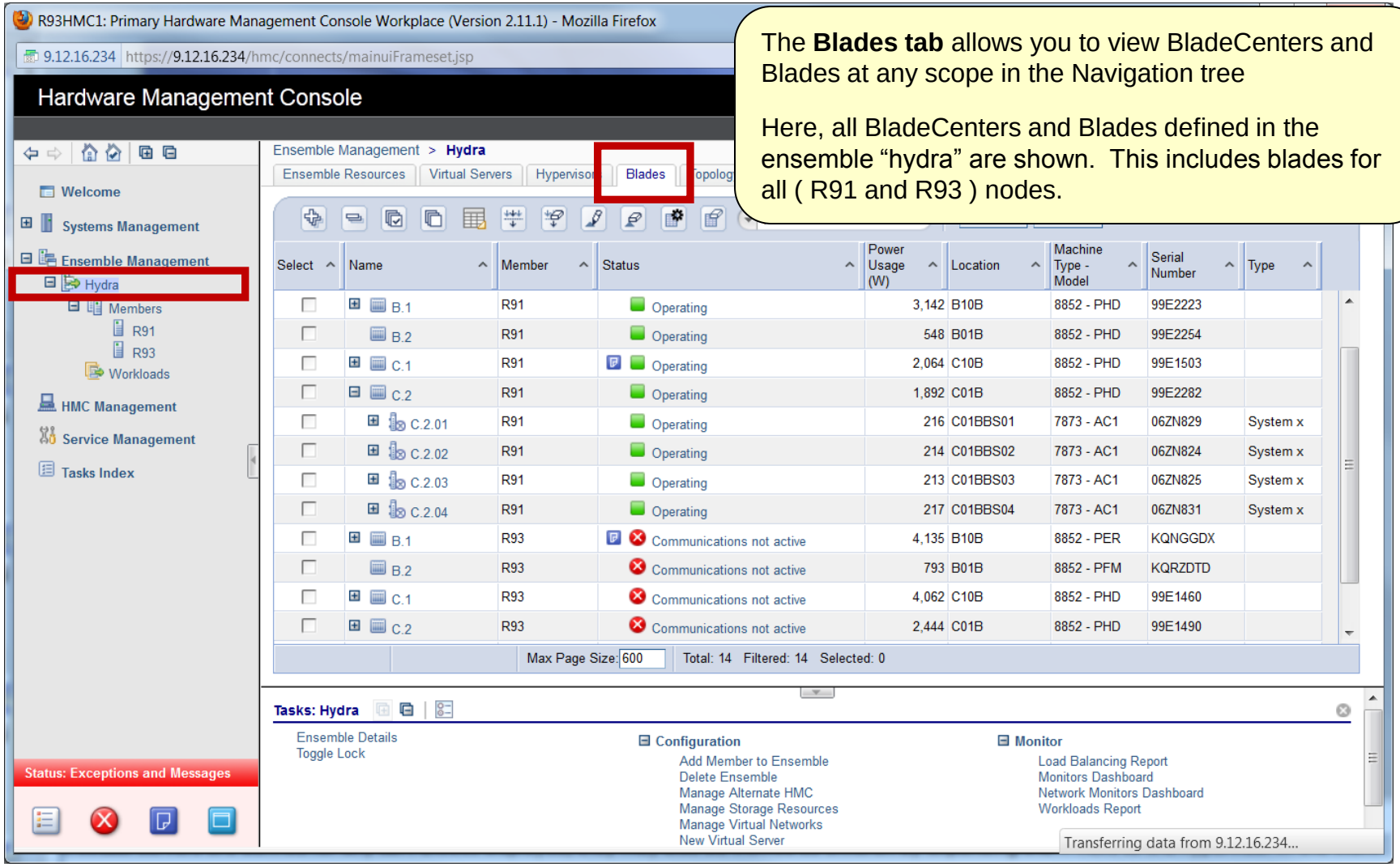
Ensemble Tabs – Hypervisors



The **Hypervisors tab** allows you to view hypervisors at any scope in the navigation tree

Here, all hypervisors defined in the ensemble “hydra” are shown. This includes hypervisors for all (R91 and R93) member nodes.

Ensemble Tabs – Blades



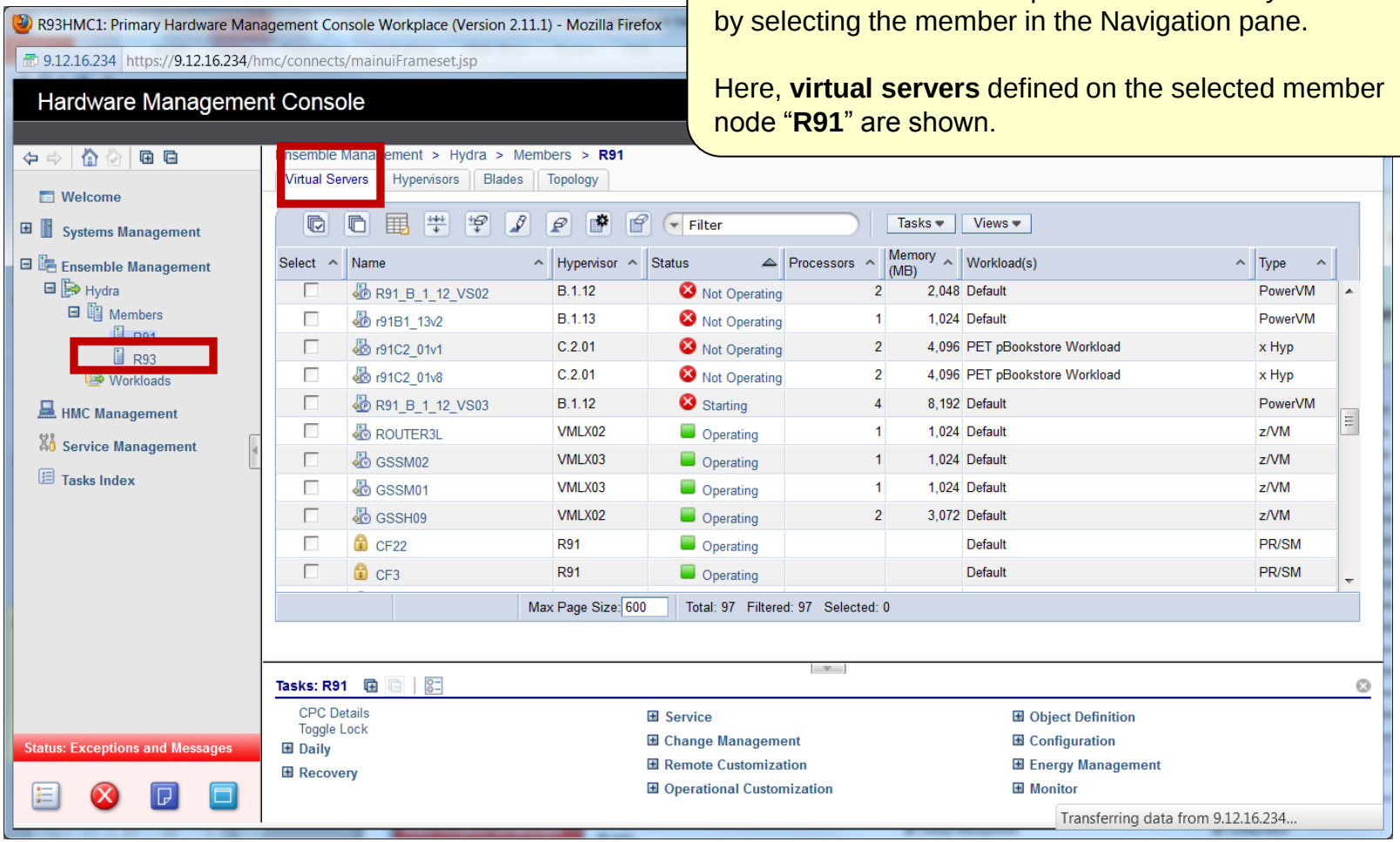
The **Blades tab** allows you to view BladeCenters and Blades at any scope in the Navigation tree

Here, all BladeCenters and Blades defined in the ensemble “hydra” are shown. This includes blades for all (R91 and R93) nodes.

Member Node Tabs

Resources defined for a specific member may be viewed by selecting the member in the Navigation pane.

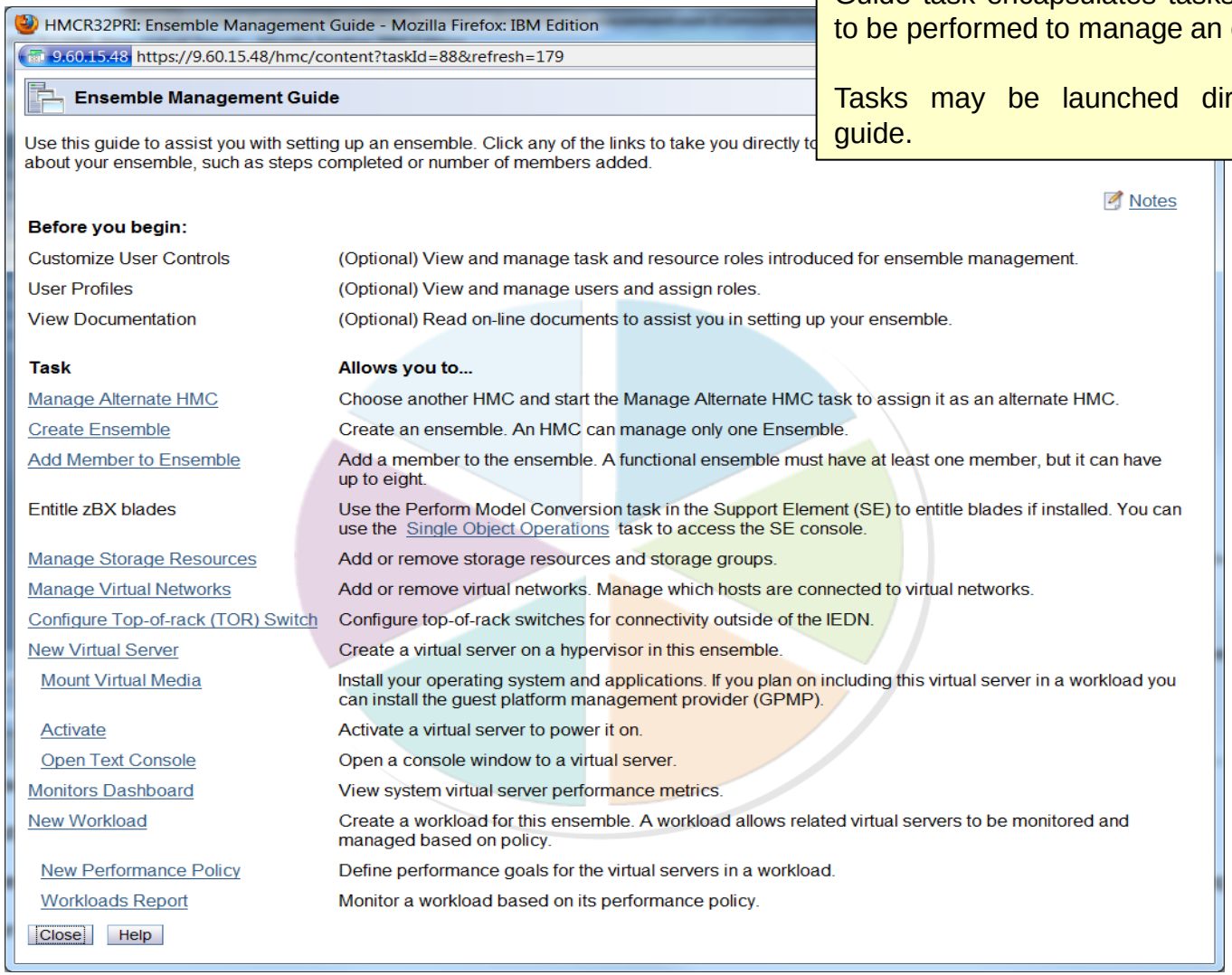
Here, **virtual servers** defined on the selected member node "**R91**" are shown.



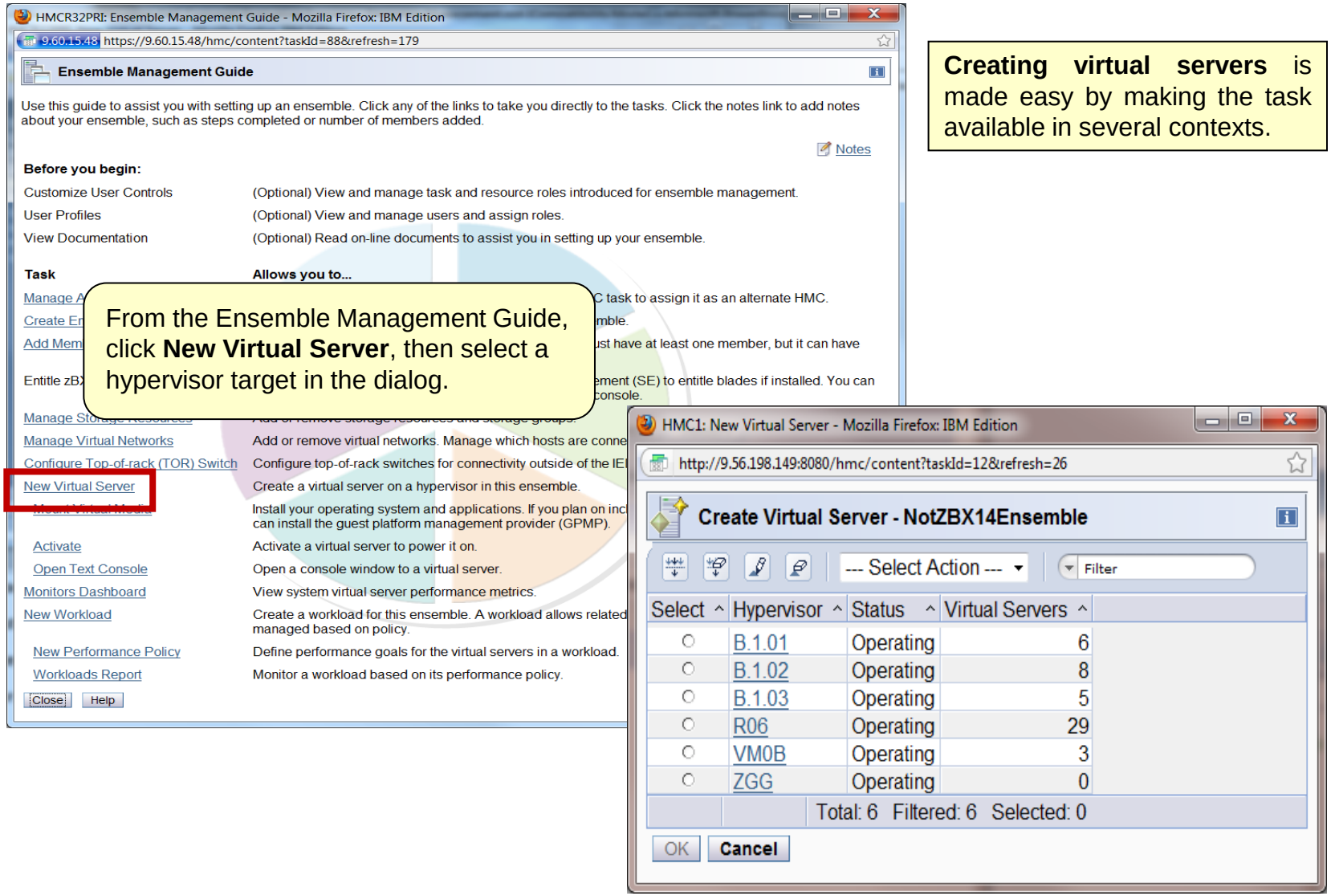
Ensemble Guide Task

Guide task encapsulates tasks that may need to be performed to manage an ensemble.

Tasks may be launched directly from the guide.

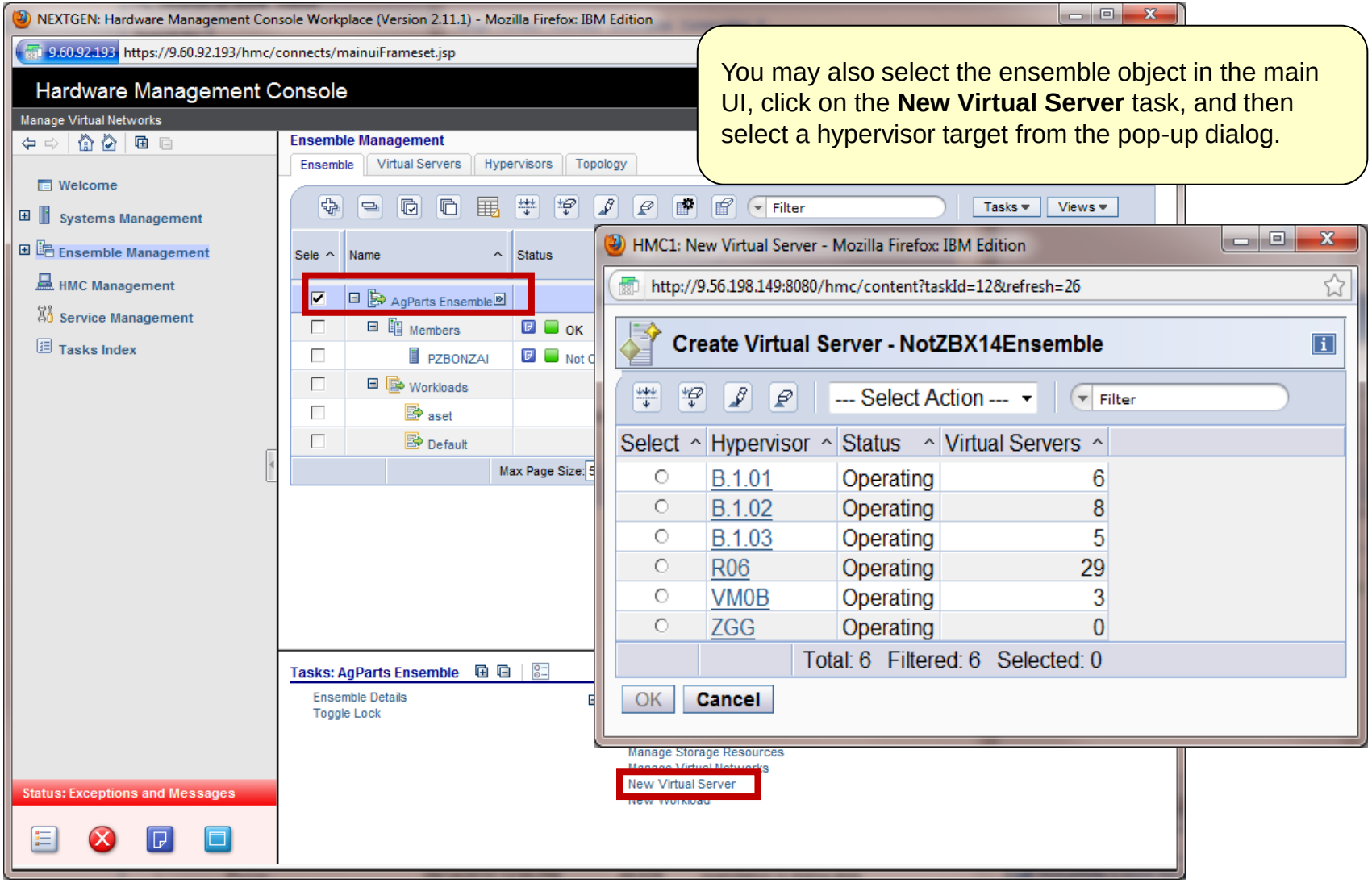


Create Virtual Server – Ensemble Management Guide



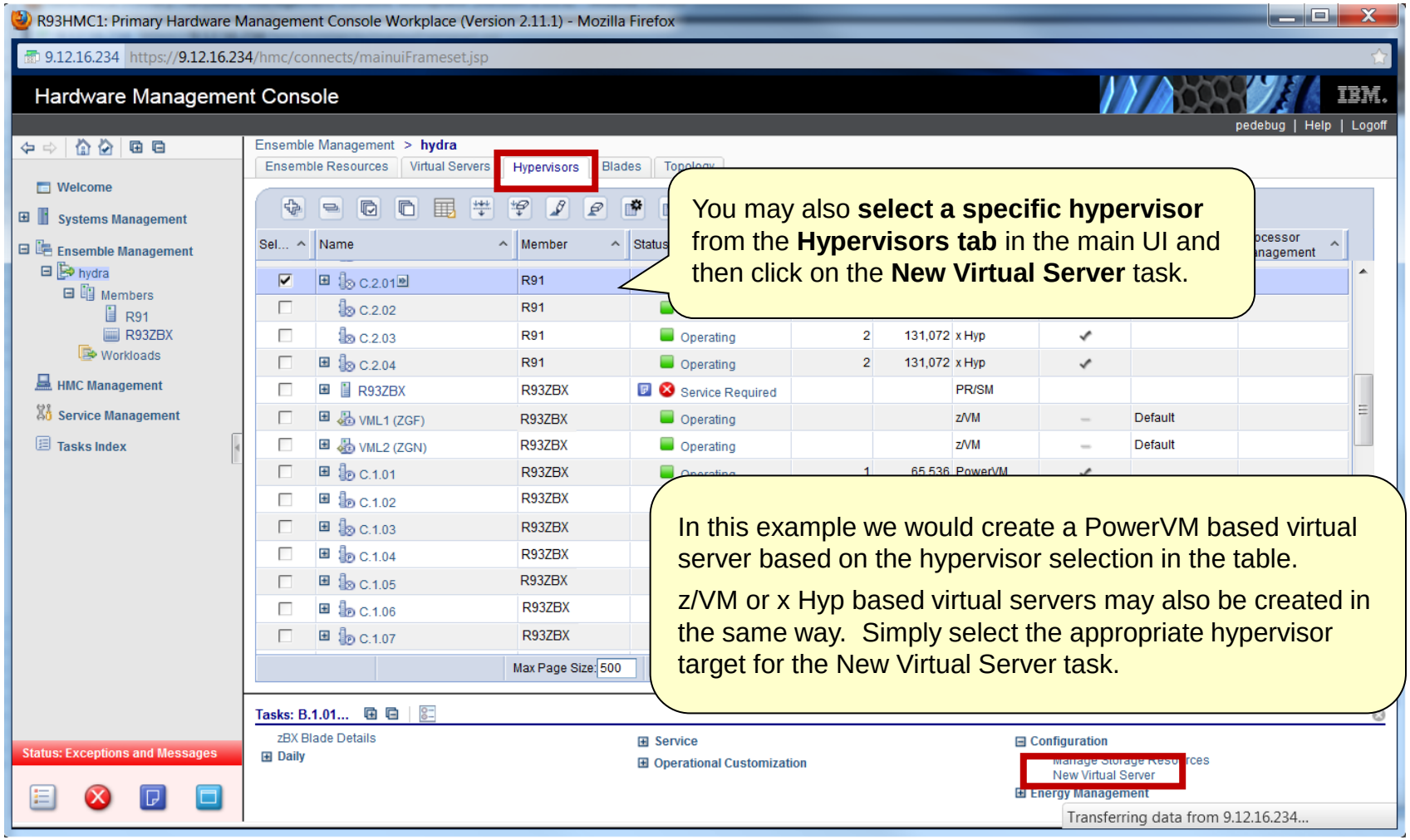
Creating virtual servers is made easy by making the task available in several contexts.

Create Virtual Server – Ensemble Target

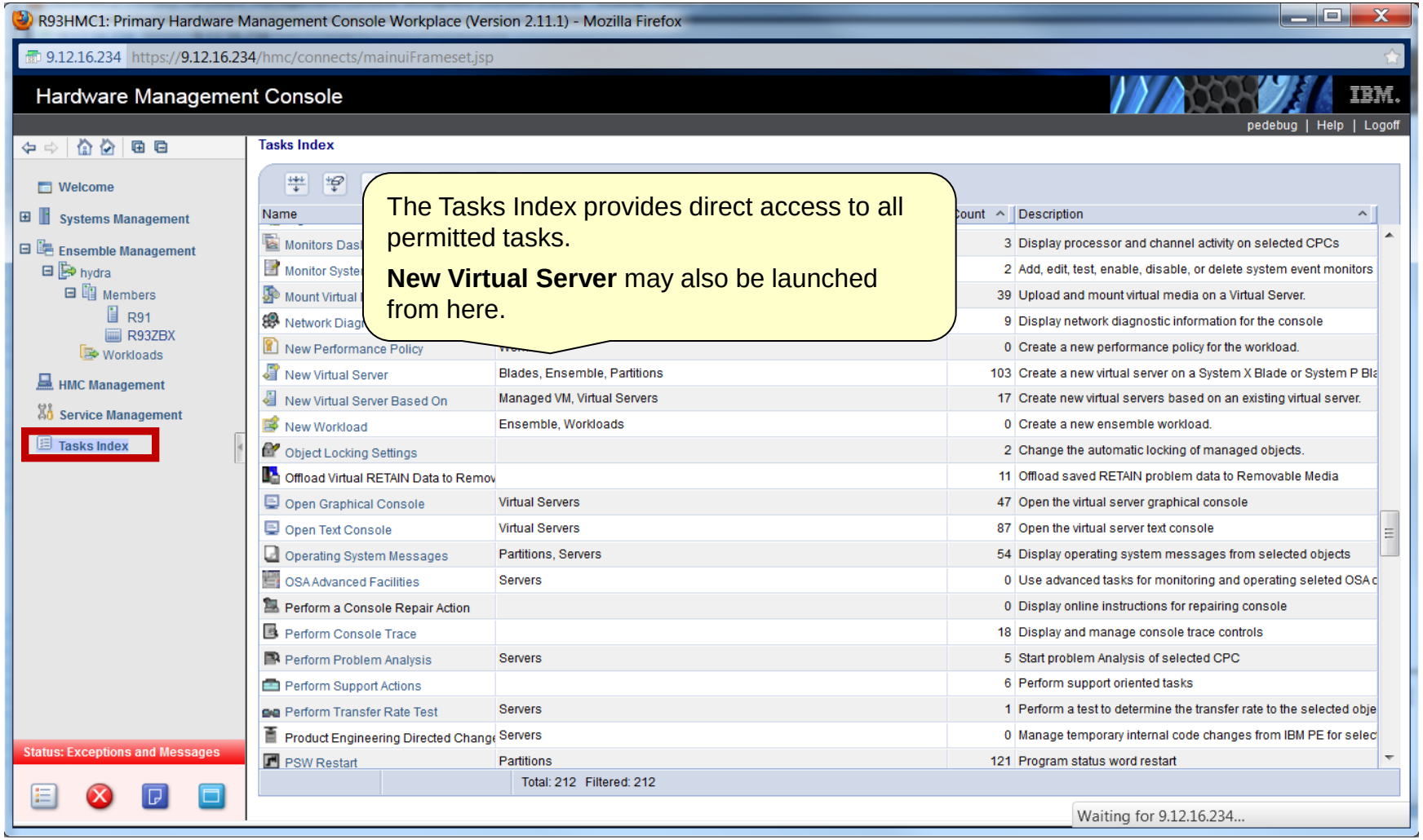


You may also select the ensemble object in the main UI, click on the **New Virtual Server** task, and then select a hypervisor target from the pop-up dialog.

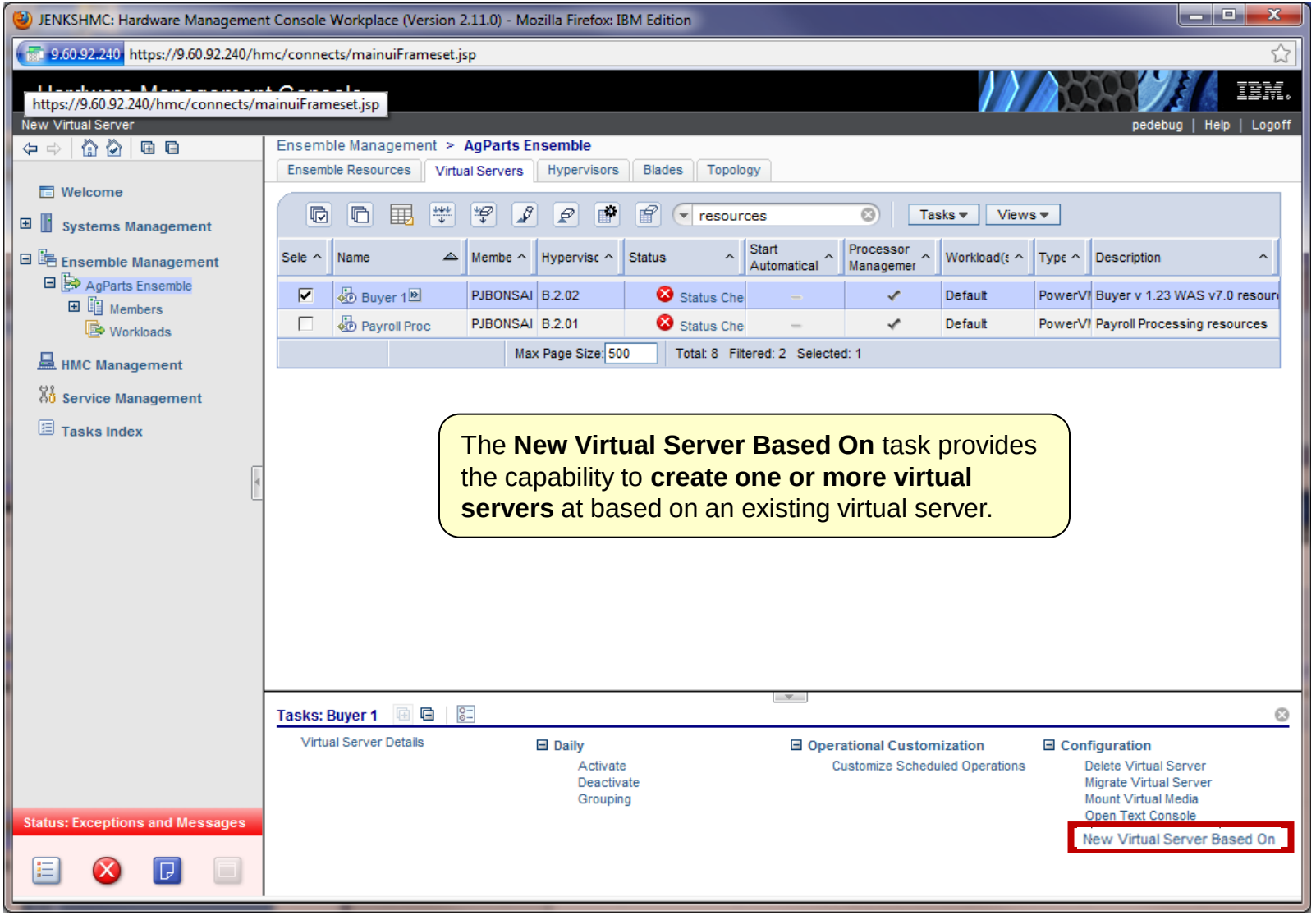
Create Virtual Server – Hypervisor Target



Create Virtual Server – Tasks Index

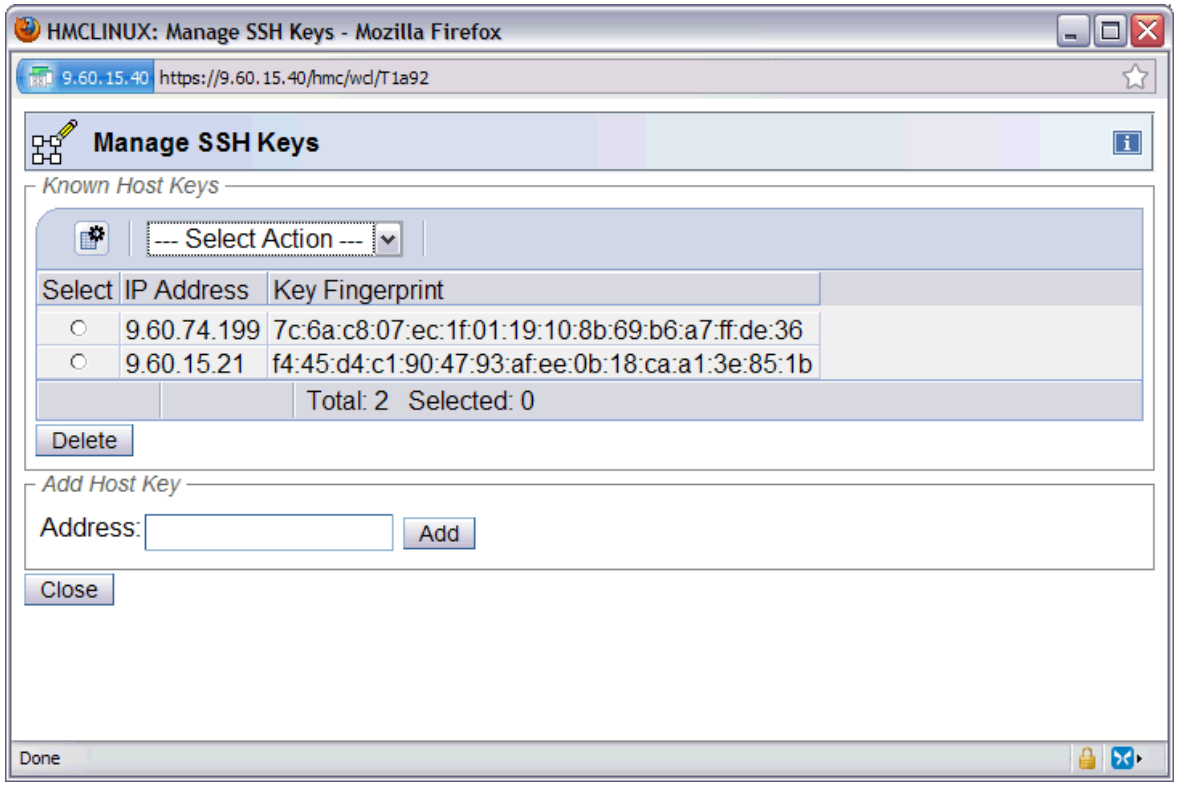


Create Virtual Server - Based On Another Virtual Server



Secure FTP Support

- Allow secure FTP connection from a HMC/SE FTP client to a customer FTP server
- SSH File Transfer Protocol which is an extension of the Secure Shell protocol
- Manage SSH Keys task allows the customer import public keys associated with a host address – added to both HMC and SE.

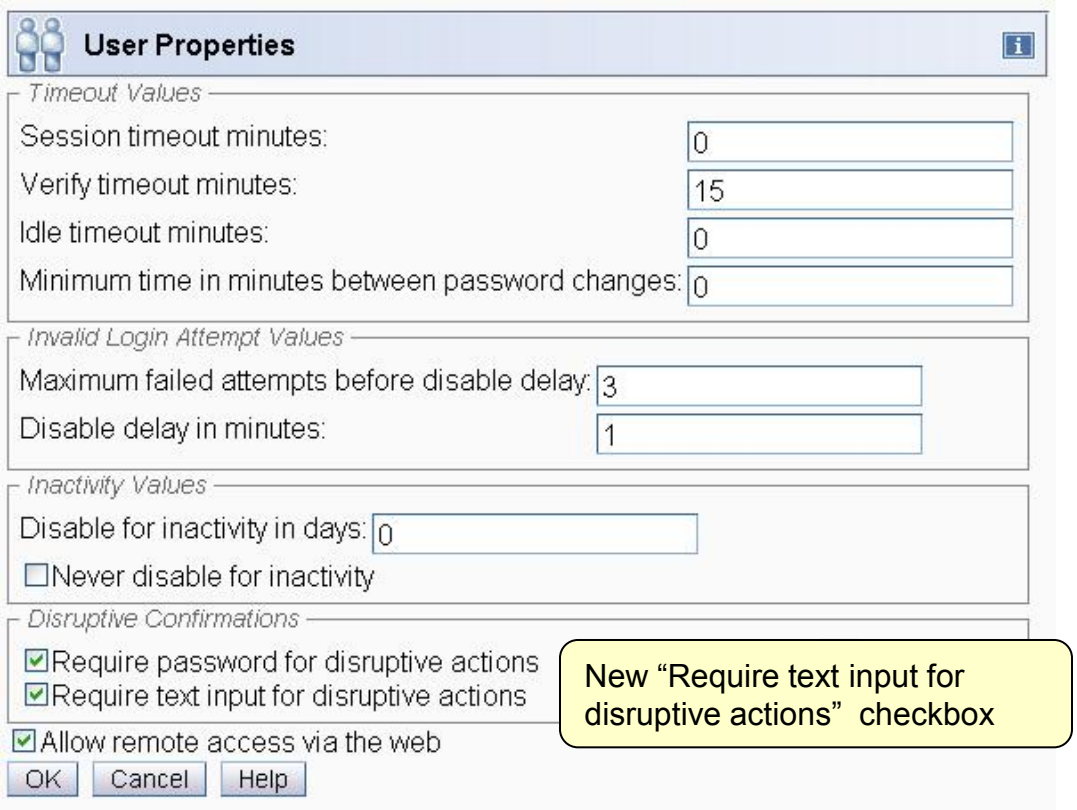


Secure FTP support - Tasks

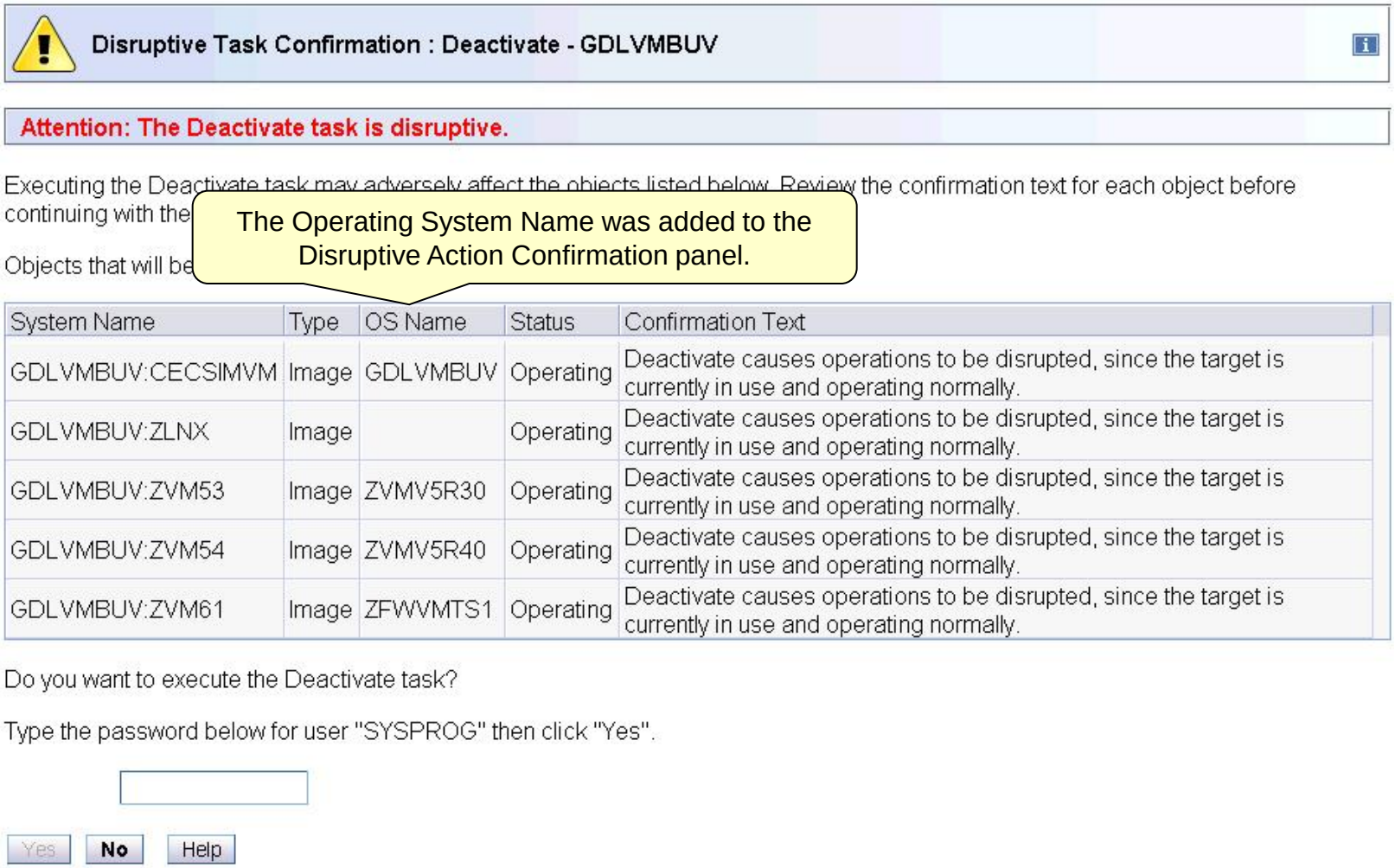
- Tasks utilizing FTP now provide a selection for the Secure Host connection.
  - Input/Output (I/O) Configuration -> Import/Export Source File ->FTP Location
  - Customize Scheduled Operations (Audit and Log Management only)
  - Retrieve Internal Code -> Retrieve code changes from FTP site to the selected objects
  - Change Console Internal Code -> Retrieve Internal Code Changes ->Retrieve code changes from FTP site to the HMC
  - Advanced Facilities->Card Specific Advanced Facilities->Manual Configuration Options->Import/Export source file by FTP (For OSA-ICC PCHIDS only – Channel Type=OSC)

Disruptive Action Confirmation

- User Profiles task provides option to requires text input for disruptive actions.
- The required text is either the OS Name or System Name



Disruptive Action Confirmation - Deactivate



Disruptive Task Confirmation : Deactivate - GDLVMBUV

Attention: The Deactivate task is disruptive.

Executing the Deactivate task may adversely affect the objects listed below. Review the confirmation text for each object before continuing with the

The Operating System Name was added to the Disruptive Action Confirmation panel.

Objects that will be

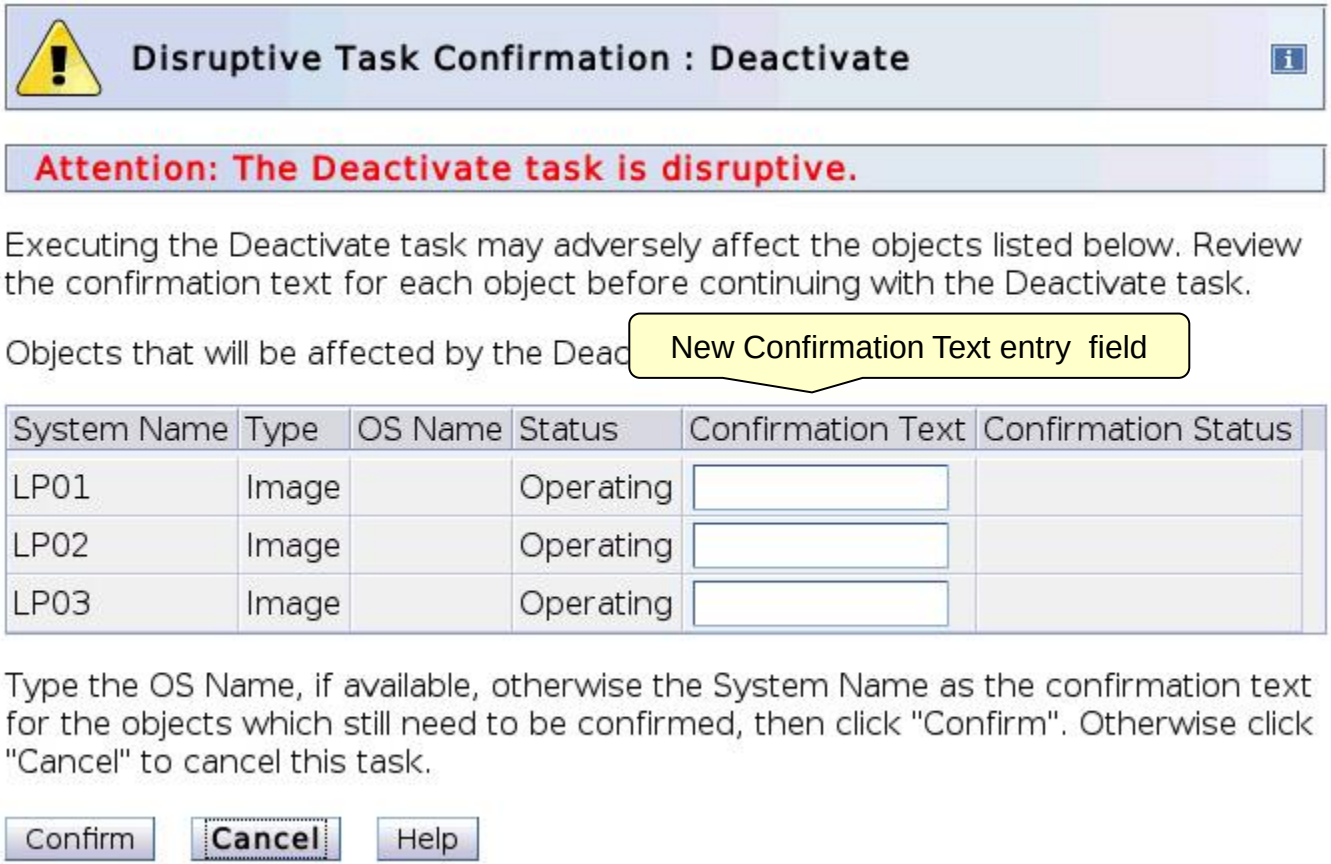
System Name	Type	OS Name	Status	Confirmation Text
GDLVMBUV:CECSIMVM	Image	GDLVMBUV	Operating	Deactivate causes operations to be disrupted, since the target is currently in use and operating normally.
GDLVMBUV:ZLNX	Image		Operating	Deactivate causes operations to be disrupted, since the target is currently in use and operating normally.
GDLVMBUV:ZVM53	Image	ZVMV5R30	Operating	Deactivate causes operations to be disrupted, since the target is currently in use and operating normally.
GDLVMBUV:ZVM54	Image	ZVMV5R40	Operating	Deactivate causes operations to be disrupted, since the target is currently in use and operating normally.
GDLVMBUV:ZVM61	Image	ZFWVMTS1	Operating	Deactivate causes operations to be disrupted, since the target is currently in use and operating normally.

Do you want to execute the Deactivate task?

Type the password below for user "SYSPROG" then click "Yes".

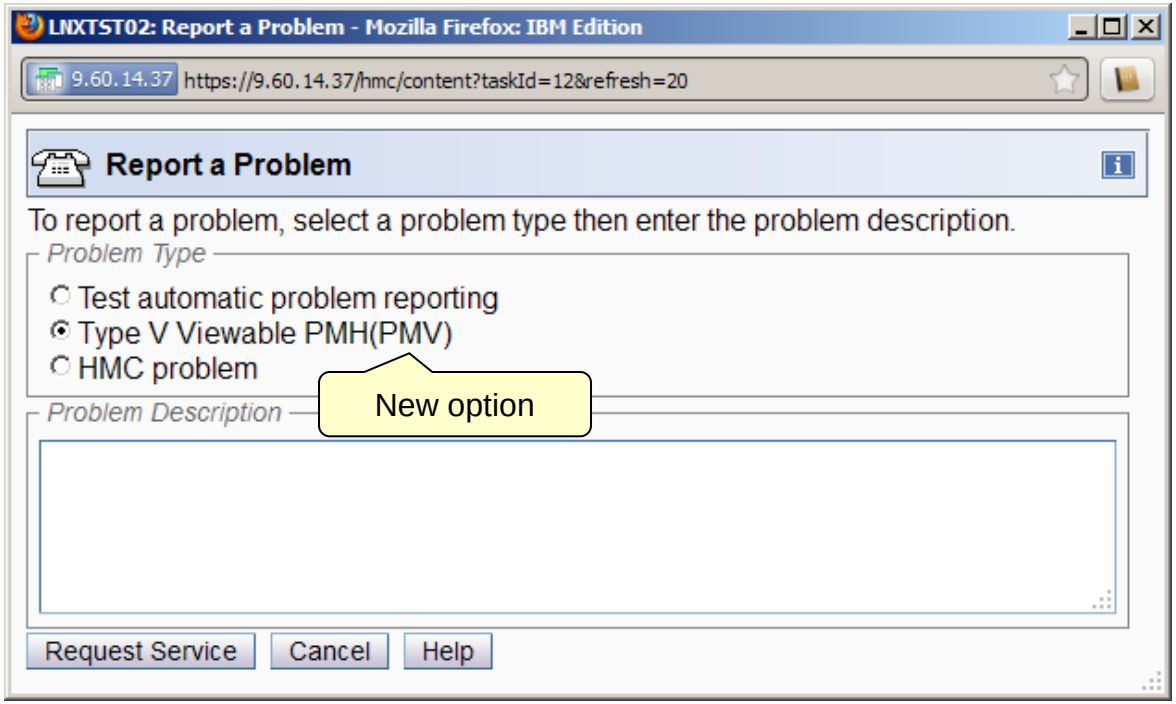
Yes | No | Help

Disruptive Action Confirmation – Deactivate Text Entry



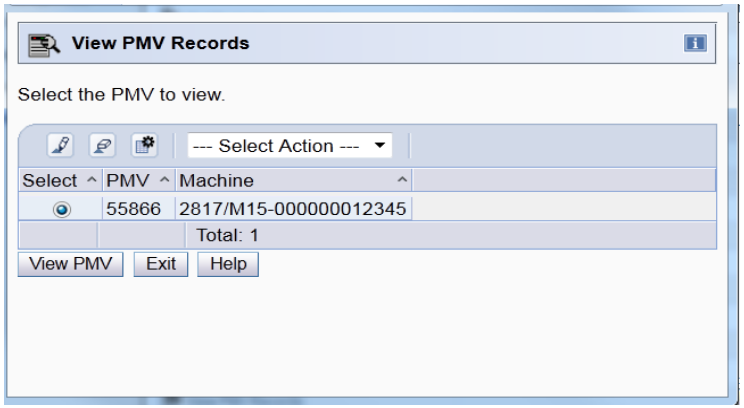
Problem Management Viewable (PMV) Records

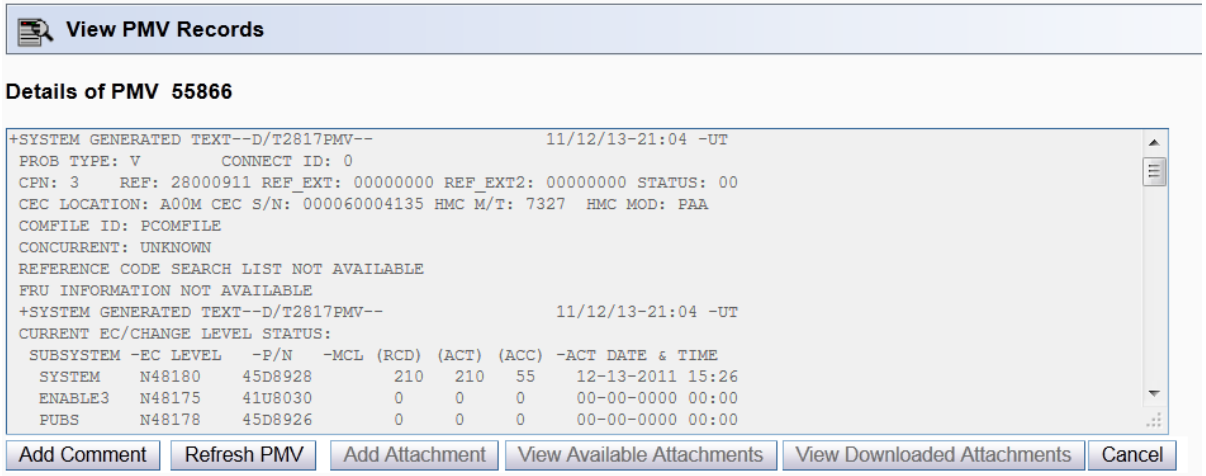
- Create, View, or Edit Problem Management Viewable (PMV) records issued to the IBM Service Support System (Retain) for the Hardware Management Console or selected servers (CPCs).
- Typically used to report issues to the IBM Service Support System for errors that are not automatically recorded by the console.



Problem Management Viewable (PMV) Records – View, Edit, Comment

- View, edit, or add comments to PMV records to interact with IBM service



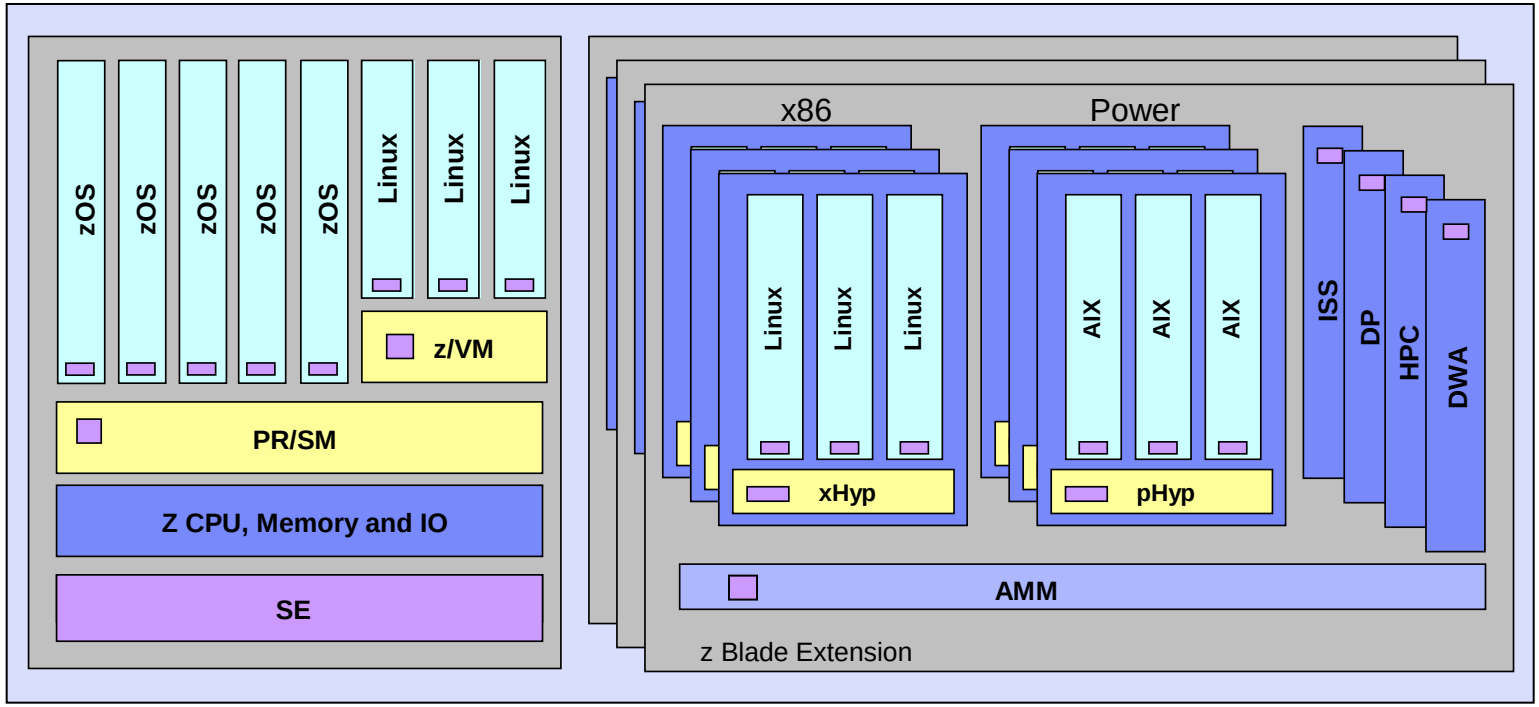


Information and Education Resources – Resource Link

- IBM Resource Link (http://www.ibm.com/servers/resourcelink)
  - System z HW publication library - click on "Library“
    https://www.ibm.com/servers/resourcelink/hom03010.nsf/pages/library?OpenDocument&login
  - System z HW education & Hardware Management Console, click on "Education“
    https://www.ibm.com/servers/resourcelink/hom03010.nsf/pages/education?OpenDocument&login
  - System z HW Customer Initiated Upgrade, click on "Customer initiated upgrade“
    https://www.ibm.com/servers/resourcelink/hom03010.nsf/pages/customerInitatedUpgrade?OpenDocument&login

Backup

Unified Resource Manager GA1 Recap



- Unified Resource Manager GA1 put in place the building blocks:
  - Consistent hardware management of Z, P and X resources
  - Integrated POWER and System X hypervisors managed as System Z components
  - Fully encapsulated management network to allow secure and reliable control
  - Firmware managed data network to provide interconnect at application level
  - Virtualization management across Z, P and X hypervisors
  - Policy based workload optimization managed by firmware

HMC System Support

- The new HMC Version 2.11.1 will support the systems/SE (Support Element) versions shown in the table.
- The 2.11.1 HMC will support up to two 10/100/1000 Mb Ethernet LANs (1 Gb LAN support)
  - Optional HMC External Switch available as 1 Gb
  - Internal z196 switch for HMC to SE LAN connection has 1 Gb ports

Machine Family	Machine Type	Firmware Driver	SE Version
z114	2818	93	2.11.1
z196	2817	93, 86	2.11.1, 2.11.0
z10 BC	2098	79	2.10.2
z10 EC	2097	79	2.10.2
z9 BC	2096	67	2.9.2
z9 EC	2094	67	2.9.2
z890	2086	55	1.8.2
z990	2084	55	1.8.2
z800	2066	3G	1.7.3
z900	2064	3G	1.7.3
9672 G6	9672/9674	26	1.6.2
9672 G5	9672/9674	26	1.6.2