

The logo for the Red Hat Summit. It features the word "SUMMIT" in large, white, sans-serif capital letters on a red rectangular background. To the right of "SUMMIT" is a white chevron shape pointing right. To the right of the chevron is the text "JBoss WORLD" in white, stacked in two lines.

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT

**LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.**

www.theredhatsummit.com

JBoss in the Trenches

Ronak Mallik, Architect
Amentra, A Red Hat Company

Brian Carothers, Vice President
Amentra, A Red Hat Company

June 25, 2010

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Welcome to JBoss in the Trenches!

In this session, you'll learn about the best ways to configure and manage JBoss.

The Best?

Configure and Manage JBoss?

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Disclaimer!

- Ivory Tower Approach
- Assumes large, well-planned, enterprise environments with an infinite amount of time, resources, and money.



Wait a second... Then why am I here?

Even if you're not the biggest shop on the block, you should stick around because:

- I will show you what makes the biggest shops so successful.
- Guidelines and ideals you should work toward



Wait a second... These recommendations are obvious!

Yes, this may be true.

BUT

Rarely do we see all of these recommended architectures and processes actually implemented and used.



Why we do the things we do...

- Automation
- Scalability
- Roll-back
- Security
- Performance
- Redundancy
- Monitoring



JBoss Versions

What version are you running?

```
[TomcatDeployer] deploy, ctxPath=/jmx-console, warUrl=.../deploy/jmx-console.war/  
[Http11Protocol] Starting Coyote HTTP/1.1 on http-0.0.0.0-8080  
[AjpProtocol] Starting Coyote AJP/1.3 on ajp-0.0.0.0-8009  
[Server] JBoss (MX MicroKernel) [4.3.0.GA_CP07] (build: SVNTag=JBAPP_4_3_0_GA_CP07 date=200911251949) Started in 39s:47ms
```

JBOSS_HOME/lib/jboss-system.jar/META-INF/MANIFEST.MF:

```
Implementation-Vendor-Id: http://www.jboss.org/  
Implementation-Version: 4.3.0.GA_CP07 (build: SVNTag=JBAPP_4_3_0_GA_CP07 date=200911251949)  
Implementation-Vendor: JBoss, Inc.
```

Enterprise Application Platform
vs.
Community Versions of JBoss

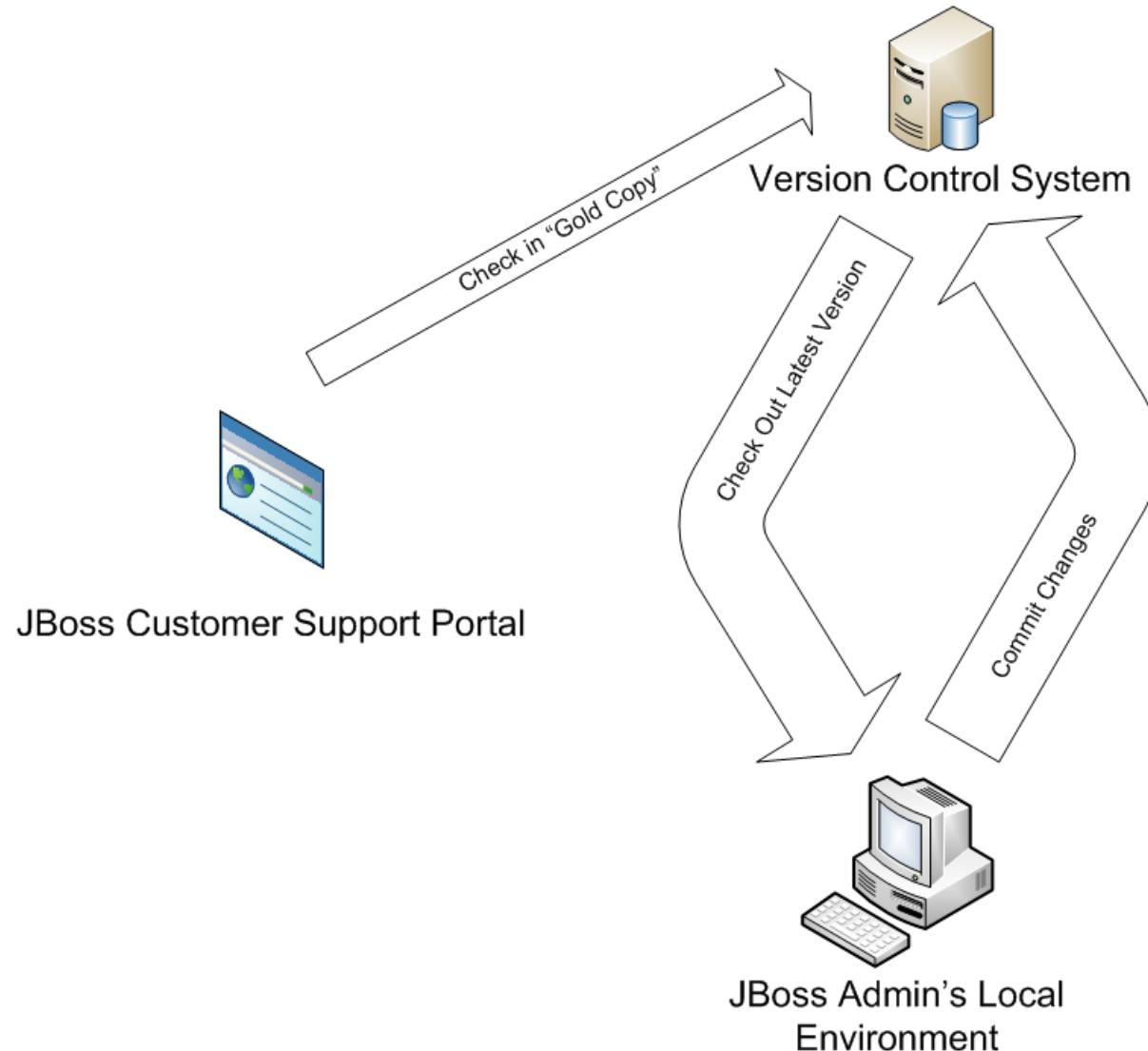
SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



JBoss Versioning



JBoss Versioning

- Store a “Gold Version” (freshly downloaded, MD5 sum verified) in a version control system (VCS)
- All configuration changes and updated .jars committed to VCS
- Automated push to environments

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



JBoss Cumulative Patches (CP)

- Best to be on latest and greatest, but check release notes
- Best way to apply the CP:
 - Capture all environment specific configurations in a document (simple as diff with Gold Copy)
 - Downloaded latest full EAP + CP, then apply environment specific changes
- You can also:
 - Apply patch manually
 - Utilize JBoss Operations Network to replace your .jar files.



Acronym Soup

Don't forget to make
use of your VCS when
applying your CP!

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



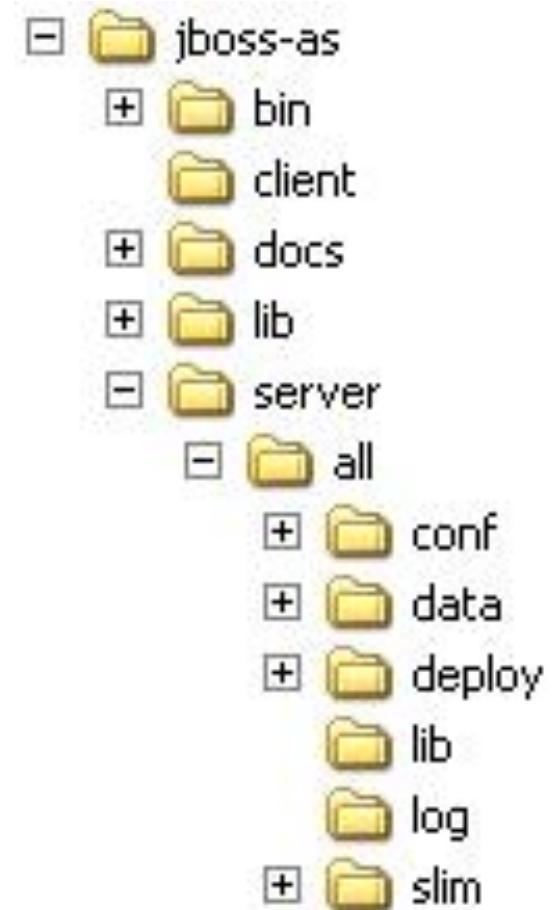
JBoss Slimming

- Lighter is better! Get rid of unused services!
- Benefits:
 - Performance
 - Using less memory + resources
 - Faster startup time
 - Security
 - Unused ports should be removed



JBoss Slimming Process

- Create another folder outside of deploy called “slim” and place unused .jars and service.xml files there.
- What to actually remove?
 - Great guides on the JBoss.org wiki for both 4.3.X and 5.X, however, this is very application dependent.
- Remember, make use of the VCS for this process!



Questions?
Comments?
Suggestions?



Running Multiple JBoss Instances on One Server

- Sometimes, no luxury of running one JBoss instance per box...
 - **Usually, less is better**
 - IO on file system
 - CPU Slicing
 - File locking
- If you must, you can either:
 - Bind to different IPs on the box (**recommended**)
 - Bind to same IP, use the Service Binding Manager to manage ports



Application Environments



Development



QA



Integration



Staging

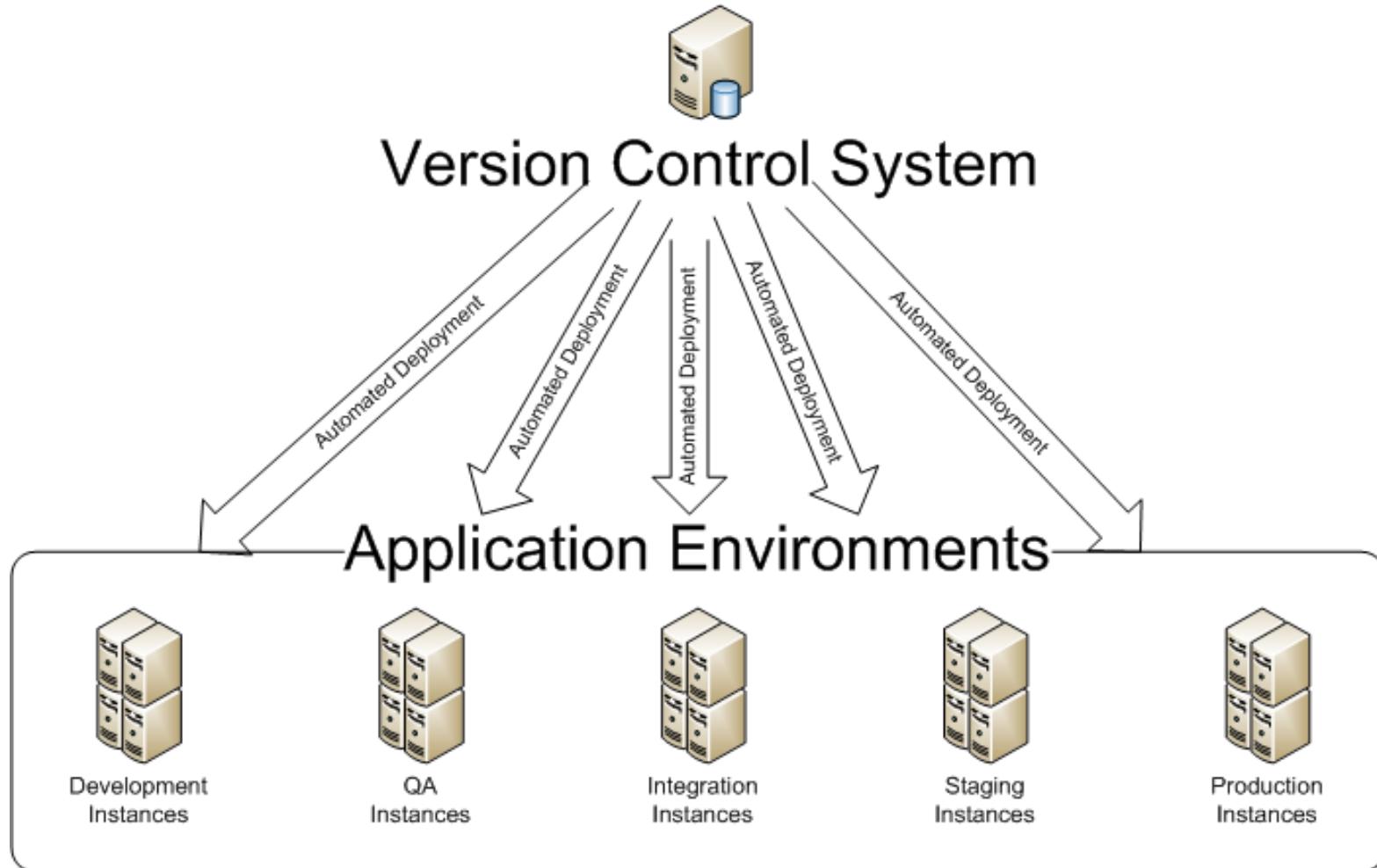


Production

- An example of the different application environments you may make use of
- Ideally, they should not share any resources
 - e.g. front end load-balancers, web servers, databases, other integration systems
- Your staging (or performance test environment) should mimic your production environment as much as possible.



Automated Deployment to Environments



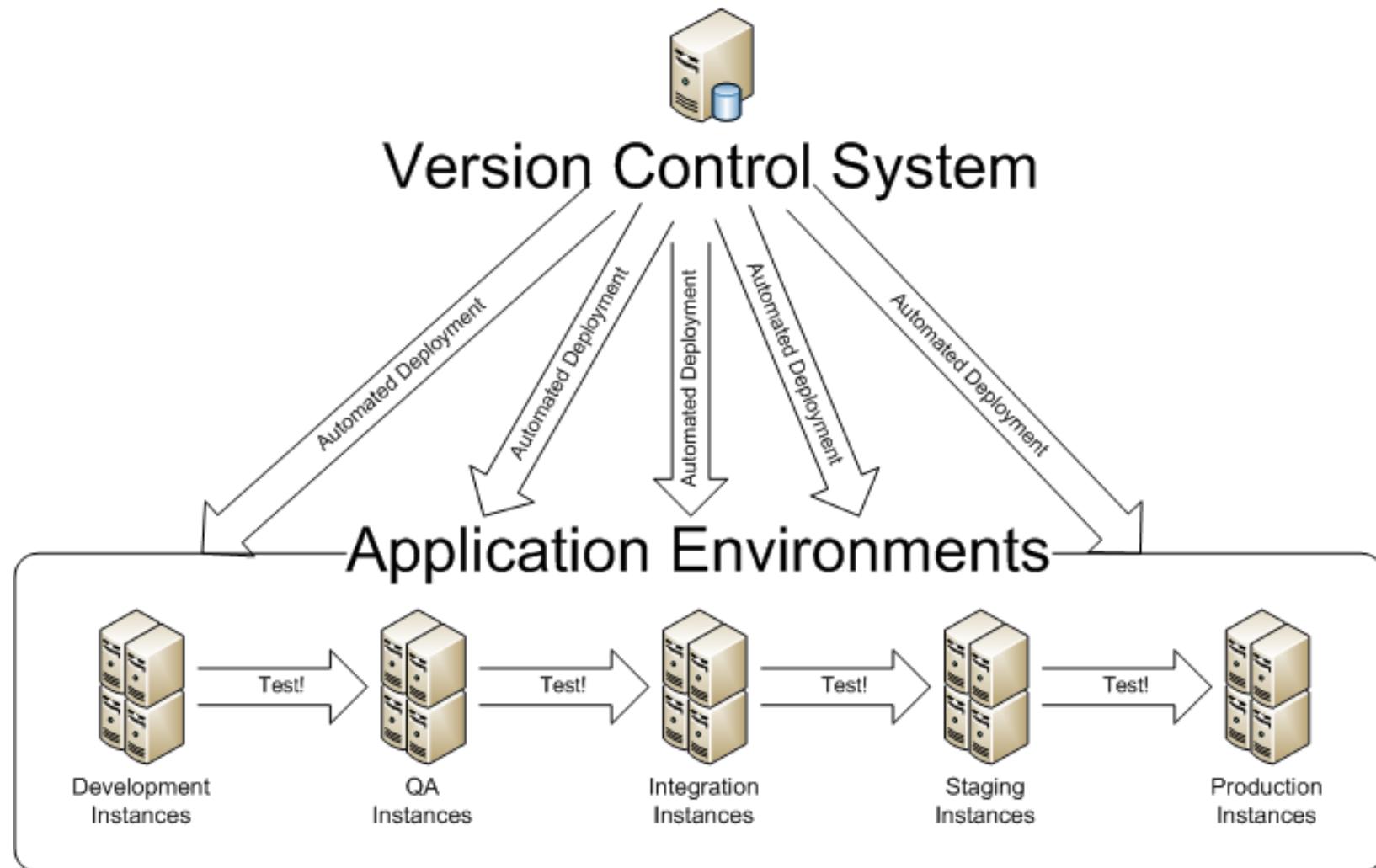
Automated Deployment to Environments

- A script or tool will check out the latest JBoss configuration file from your VCS, apply any environment specific changes, and deploy this to a given environment
- There should never be changes made manually to any environments*
- Tool should provide an audit trail
 - To see what is deployed where!

* Except the “lowest” development environment



Movement Through Environments



Movement Through Environments

- For both JBoss configurations and application code:
- There should never be changes made manually to any environments*
 - Criticality of using auto-deploy from VCS
- Test in each environment before promotion!

* Except the “lowest” development environment



Logging

- Should be different for your different application environments
- As regions get higher and higher (closer to production) logging levels should be less and less verbose.
- In your log4j configuration, define robust log rolling policies. Use past history in prod-like environments (staging) to determine log space requirements. Sync them to backup servers as needed.



Messaging

- JBoss Messaging (Versions 1.4.0 SP3 CP01 and 1.4.6 for JBoss EAP 4.3.X and 5.0.X respectively) implements the JMS 1.1 spec, and then some.
- Notable features:
 - Strict message ordering. *
 - Fully clustered queues and topic with intelligent load distribution.
 - High availability and seamless fail-over

* With given conditions / performance reduction



Security

- Secure your JMX invokers.
- Secure your Admin consoles and remove them, if not used, in the production environment.
- Eliminate clear text password in files:
 - Datasource Passwords
 - LoginModule Passwords
 - Tomcat Connector Passwords etc
- Slimming
 - remove unused services and ports
- Create a specific “JBoss” user on server, which owns JBoss folders and files. run.sh must be run by this user.



Questions?
Comments?
Suggestions?



Performance Testing

- The key here is piece of mind. Have a plan, and stick to it. Enterprise monitoring must be in place.
- First, establish your target performance profile
- Establish a baseline.
- BE SCIENTIFIC
- Do NOT forget endurance testing
- Don't forget failover scenario testing



Performance Tuning Tips

- Slimming
- In production, turn off hot deployment
- Tune your GC collection strategies
 - Depends on the application idiosyncrasies (AI)
- -Xms = -Xmx
- jboss.system:service=ThreadPool
- Tradeoff between having massive heap and fewer nodes or smaller heaps and more nodes in cluster depends on AI and network performance
- Tune database connection pool size after AI analysis.
- Enable EJB3 Entity / Hibernate caching to reduce load on DB
- OS tuning



Performance Tuning Tips

- Pre-compile JSP's
- Tune the web connectors
 - HTTP, HTTPS, AJP
 - maxThreads
 - Depends on number of instances, instances per box, cores, etc
 - Goes hand in hand with Apache tuning
- For example, worker MPM, need to tune: ThreadLimit, ServerLimit, StartServers, MinSpareThreads , MaxSpareThreads, MaxClients, ThreadsPerChild, MaxRequestsPerChild.
- For specific thread calculations and configs, see Sam Mendenhall's fantastic app @ <http://lbconfig.appspot.com/>
- Enable sticky sessions for consistent requests for a session



High Availability

- Overall architecture for HA includes front end hardware load balancing strategies, webserver tier, app server tier, DB tier, and any other integration points.
- You must consider all possible failover scenarios and the desired course of action to ensure your desired level of HA.
- JBoss Clustering
 - Doesn't usually improve performance, in fact, it's usually the opposite (unless no state/cache is used...)
- Different approaches to clustering
 - HA-singleton
 - Session sharing and tomcat clustering for web applications
 - Stateless session bean clustering
 - Stateful EJB3 entity beans



Monitoring

- Necessary in any load testing environment and obviously your production environment
- Plethora of tools exist.
- Regardless of which tool you use, you need to be able to monitor the following metrics (in real-time and historically)
 - OS level health including CPU usage, swap space, etc
 - Active threads on application server
 - Front end http/ajp web connector threads
 - Database connection pools
 - JVM Metrics such as memory usage and GCs



Management

- Alerts, Alarms, and Automated Actions (oh my)
- Must define overall game plan for operations management
- Need a documented process flow for dealing with issues in given environments
- User roles/permissions for given management tool or access to servers.



Conclusion

These are principles that your team should work toward.

Due to available resources, space, and timelines, it is understandable that your team won't be able to implement all of the previous recommendations.

The more time and effort you put in to have a standardized, consistent, methodology and environment the more efficient, agile, reliable your environment will be.



Questions?
Comments?
Suggestions?



Thank you!

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#summitjbw

READ THE BLOG

<http://summitblog.redhat.com/>

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT

