Deployment Guidance and Best Practice based upon Customer and Rational Field Experience

Steve Beard
IBM Rational Technical Lead for Deployment
stevenbeard@uk.ibm.com

Innovate2013
The IBM Technical Summit

Rational deployment challenge

- One of the questions that customers most frequently ask us is:

  How should we deploy an enterprise Rational environment that is: flexible, scalable, performant, available, resilient, secure and supportable etc.?

- Quite rightly, you balk at an answer that just points you to the system requirements and product documentation, or that gives you a myriad of options along with an ‘it depends’ answer

You want us to be more prescriptive and give you stronger guidance!

Rational deployment way forward

- Focus on a few recommended prescriptive instantiated deployment topologies (guidance and best practice)
- Focus our development and testing, particularly system and performance testing, on these recommended topologies (development)
- Reduce permutations that are of little or no business or technical value to our customers (development)
- Focus our Rational Field engagement and best practice on these recommended topologies (engagement)
- Provide a single repository for all additional deployment guidance and best practice for customer, partners and Rational staff (Deployment wiki and community)

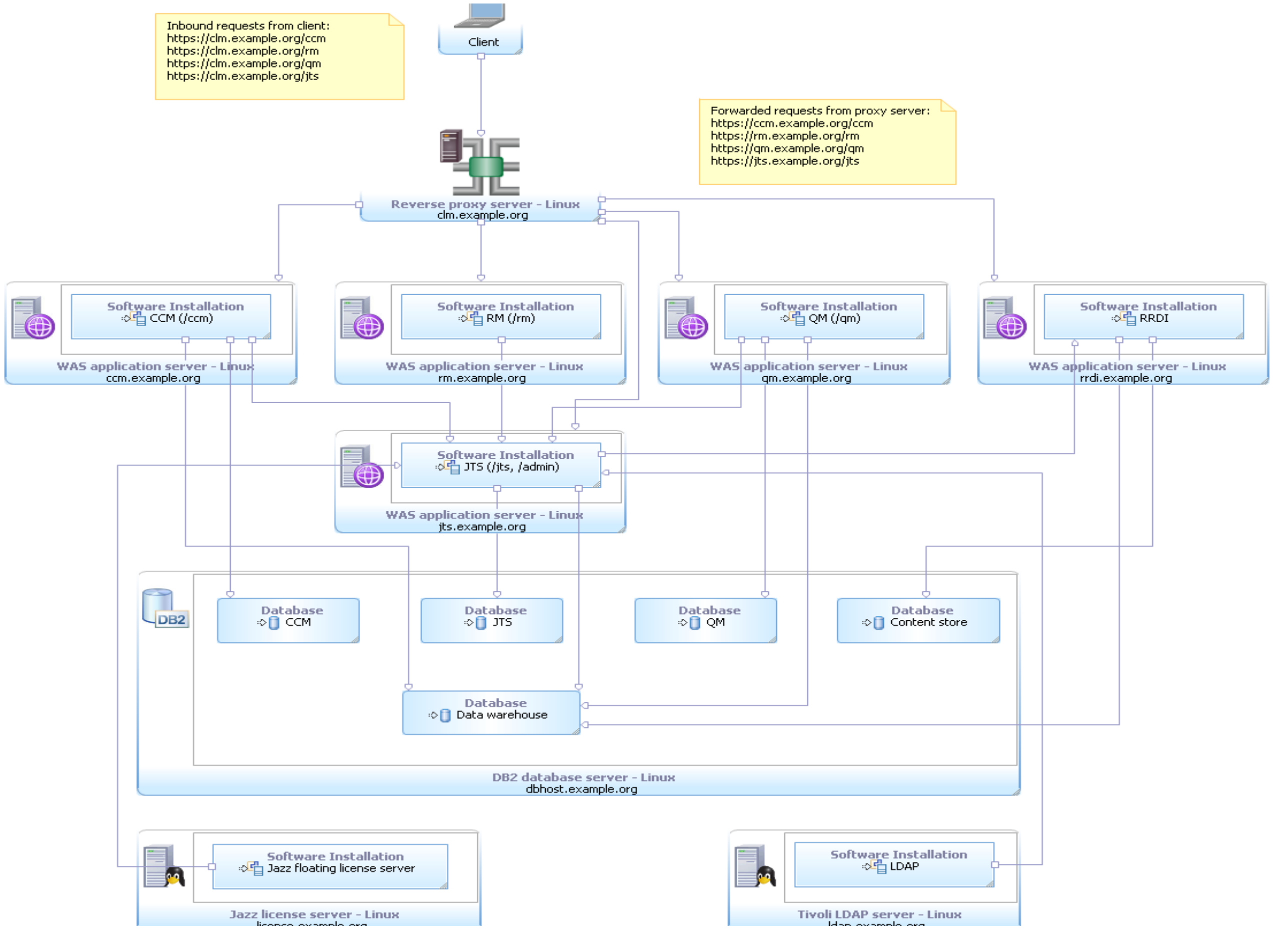
Vision: Improve our customer's time to value by providing more prescriptive guidance and simplifying the deployment of Rational solutions

Technical vs. organizational deployment

- Rational deployment splits into two distinct aspects:
  - **Technical deployment** of Rational products and solutions that includes: planning, designing, installing, configuring, integrating, upgrading and administering a Rational development environment that is: flexible, scalable, performant, available, resilient, secure and supportable
  - **Organizational deployment** which includes technical deployment, but also includes the business change, adoption, best practices, enablement and the whole Rational field engagement need to improve an organizations development capability

This presentation and the Deployment wiki on jazz.net focus on technical deployment, but need to be cognizant of the organizational requirements and constraints

Standard topologies and their instantiation [(CLM-E1) Enterprise - Distributed / Linux / DB2]


Inbound requests from client:
https://clm.example.org/ccm
https://clm.example.org/rm
https://clm.example.org/qm
https://clm.example.org/jts
Client
Forwarded requests from proxy server:
https://ccm.example.org/ccm
https://rm.example.org/rm
https://qm.example.org/qm
https://jts.example.org/jts
Reverse proxy server - Linux
clm.example.org
Software Installation
CCM (/ccm)
WAS application server - Linux
ccm.example.org
Software Installation
RM (/rm)
WAS application server - Linux
rm.example.org
Software Installation
QM (/qm)
WAS application server - Linux
qm.example.org
Software Installation
RRDI
WAS application server - Linux
rrdi.example.org
Software Installation
JTS (/jts, /admin)
WAS application server - Linux
jts.example.org
DB2
Database
CCM
Database
JTS
Database
QM
Database
Content store
Database
Data warehouse
DB2 database server - Linux
dbhost.example.org
Software Installation
Jazz floating license server
Jazz license server - Linux
Software Installation
LDAP
Tivoli LDAP server - Linux

What is your organizations instantiated deployment topology based upon?

- **OS edition/version and platform:**
- **Virtualized SW/version or physical:**
- **Application server edition/version:**
- **Database edition/version:**
- **Basic topology:**
- **Supporting services:**
- **Other aspects:**

The most common successful customer instantiated deployment topologies are based upon

- **OS edition/version and platform:** RHEL 5/6 or Windows 2003/8 on x86-64, some Power
- **Virtualized SW/version or physical:** VMware 5.X and some KVM, little physical
- **Application server edition/version**: WebSphere Application Server 7/8
- **Database edition/version:** Oracle RAC 11g R2 or DB2 HADR 9.7, a little Microsoft SQL Server
- **Basic topology:** Enterprise topology often with multiple RTC instances
- **Supporting services:**
  - IBM HTTP Server Reverse Proxy
  - LDAP or Active Directory
  - JTS and 3 node FlexLM for non-Jazz tools
  - OAuth for Single Sign-On (WAS)
- **Other aspects:**
  - Tier 1 storage, typically fiber attached SAN or Tier 2, typically NAS
  - 1GB or better local server network
  - Back-up based DR that is tested to another datacenter

Over 3 million permutation for deploying CLM alone, but we mainly see a few standard instantiated topologies

- Most common/recommended CLM topologies:
  - (CLM-E1) Enterprise - Distributed / Linux / DB2
    - Often with Oracle instead of DB2 and usually virtualized with VMware on x86-64
  - (CLM-E2) Enterprise - Distributed / all Microsoft
    - Usually virtualized with VMware on x86-64
- Focus on instantiated topologies that are based upon:
  - Those that are most successful to date for customers and internally within IBM
  - Those that are based upon the most commonly available platforms, OSs and middleware
  - Those that are based upon technologies that customers, partners and the IBM Rational Field have the most experience with
  - Those that are the focus of testing within IBM Rational

Deployment: Where to start?

- Plan and design you Rational deployment against your potential usage model, and non-functional requirements and constraints
- Build in as much scalability and flexibility into the design and deployment
- Monitor usage and performance etc.
- Adjust the configuration to meet your changing requirements, constraints and needs

Typical deployment workshop or review

Day 1

- **09:00 - 09:30:** Introductions, scope and agenda
- **09:30 - 10:00:** Present and discuss key consideration when designing a Rational development environment
- **10:00 - 11:30:** Understand general requirement, constraints and usage for environment
- **11:30 – 12:30:** Review existing development environment (Rational or 3rd party)
- **12:30 – 13:00:** Lunch
- **13:00 – 14:30:** Understand existing infrastructure services and preferred middleware/platforms (make key platform, OS and middleware decisions)
- **14:30 – 16:30:** Discuss deployment topology options based upon preferred technology and common services
- **16:30 – 17:00:** Review and refine Day 2 agenda, wrap-up Day 1

Day 2

- **9:00 - 10:00:** Review of Day 1
- **10:00 - 16:30:** Specific sessions to discuss and agree the approach to and design for:
  - Capacity, performance and monitoring
  - High-availability and disaster-recovery
  - Security
  - Audit and control
  - Compliance
  - Administration, configuring and tuning
- **16:30 – 17:00:** Overall wrap-up and discuss next steps
  - Who is going to document the detailed deployment design and by when!

Development a usage model for your Rational development environment

- Consider short, medium, and long-term
  - It is important to understand the potential capacity and scalability requirements from the outset so that you can design in flexibility to meet your changing needs
- At a minimum, this model should include:
  - Your development roles
  - Numbers of staff per role
  - Staff locations and numbers per role
  - Expected usage or process per role
- Further, try to estimate the amount of SCM data the environment will need to support
  - Both initially from migrating for previous SCM tools and the rate of growth of data over time

Properly consider and document your REAL development environment non-functional requirements

- **Capacity, performance and monitoring:** Consider common user operations as a bench mark of required performance as well as automated monitoring
- **High-availability:** What is the real availability you need to support your operational systems?
  - Almost no organizations need their development environments 100% available!
- **Disaster-recovery:** What is a reasonable recovery time and how much data can you afford to loose?
- **Security:** What corporate, industry, compliance or government security standards to you need to adhere to?
- **Audit and control:** What are you organization, industry or regulatory requirements?
- **Compliance:** What industry or regulatory standards to you have to comply with?

Re-use common managed IT infrastructure services

- These could include:
  - Central database service
  - Virtualization or cloud environment
  - Back-up, high-availability and disaster-recovery services
  - Authentication and licencing
  - Complete hosting or Platform-as-a-Service offering
- Make sure the Service-Level Agreement (SLA) meets your requirements
  - Typically they will be operational SLAs, so more than sufficient for a development environment
  - Usually higher quality-of-service than an individual team can support
- Separate infrastructure from tools and methods administration if possible

Consider using virtualization for flexibility and scalability

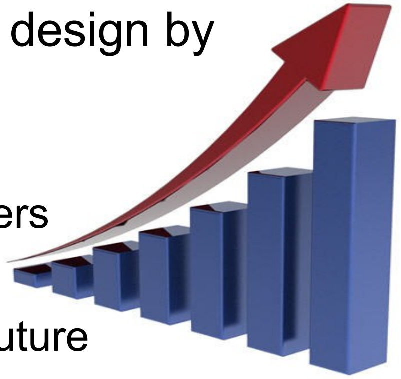
- Virtualize the application tier including your reverse proxy
  - Database tier flexibility and scalability is best severed by database clustering
- Well-managed virtualized servers permit modifying CPU, RAM and disc image sizes with much greater ease than physical servers
- A virtualized infrastructure can support certain levels of high-availability and disaster-recovery:
  - Monitor application and virtual server, and automatically restart if unresponsive
  - Cope with single hardware failure e.g. VMware HA
  - Failover between datacenters e.g. VMware vSphere/vMotion
- Principles of good virtualization:
  - Virtualization must be managed and monitored
  - Ensure that CPU, memory and network resources are dedicated and uncapped
  - Do not overcommit a virtualization environment

Start by making the major decisions about platform, OS and middleware

- Where possible, choose the middleware and editions that meet your ultimate requirements
  - Reduce the need for major migrations as you scale-up the environment
  - Reduce the amount of time you spend re-learning how to tune new middleware or editions
- Where possible, choose a platform, OS and middleware that you already have experience with
- Choose a deployment topology
  - Preferably from our recommended or alternative topologies documented in the Deployment wiki on Jazz.net

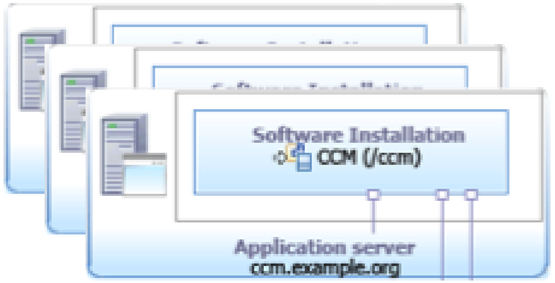
Estimate your potential usage BUT design an environment that could scale to support at least twice the usage

- Estimate your potential usage based upon potential usage model
- Use IBM Techline AND the local Rational Field Team to help you size you environment. IBM Techline provides:
  - IBM server sizing for IBM software and for selected software vendors
  - Solution design, configuration, validation and assistance
- Build as much flexibility, scalability and performance into you design by using:
  - Application server virtualization
  - Separate Jazz application onto separate virtual or physical servers
  - Database clustering e.g. Oracle RAC and DB2 HADR
  - A proxy server to allow you to refactor your environment in the future



Plan for multiple instances of specific application servers

- Plan for multiple application instances if you think you are going to scale to need them!
  - Split teams/projects across multiple instances along organizational boundaries or due to low dependency
  - Typically RTC and occasional RQM require multiple instances
- Each instance requires a separate context root e.g. two CCM instances could be ccm1 and ccm2
- Open Services for Lifecycle Collaboration (OSLC) WI linking and Distributed SCM support cross-instance working
- **Restriction:** Two instances of RRC cannot share the same JTS



Welcome to the Deployment wiki and community

Community technical leaders, senior editors, and sponsors

- Deployment wiki added to jazz.net site:
  - https://jazz.net/wiki/bin/view/Deployment/WebHome
- Repository for Rational deployment guidance and best practices for customers, partners, and IBM staff beyond the basic product and support documentation
- The scope of the wiki is the full range of Rational products and solutions
  - However, it focuses on advice for those that are most commonly deployed
- Read access to the Deployment wiki is unrestricted and without jazz.net login
  - Search engine indexable e.g. Google
  - Write access is controlled by experience groups

Have a look, provide feedback/suggestions, and become a contributor!

Final remarks, questions and discussion

- There is no such thing as a evaluation environment!
  - As soon as you give developers and teams access to an evaluation environment they will use it for productive work
- It is easier to scale-down and enterprise topology/environment than scale-up an evaluation topology
- Standardize on the same platform, OS, middleware, editions and versions for the whole of your Rational environment where possible
- Always include a reverse proxy in front of your application servers
- Treat development as a first-class business function and make sure your Rational environment will meet your business requirements and needs