



# Medici Administration

Luigi Marini



National Center for Supercomputing Applications  
University of Illinois at Urbana-Champaign

# Installation Methods

- Ubuntu package
- Virtual machine
- Manual install from build
- Manual install from source

# Ubuntu Package

- Add public key for Medici repository
  - `wget -q -O - http://opensource.ncsa.illinois.edu/ubuntu/public.key | apt-key add -`
- Add Medici Debian repository to list
  - Add to `/etc/apt/sources.list.d/ncsa.list` the following line:
  - `deb http://opensource.ncsa.illinois.edu/ubuntu/ stable main`
  - (or development for development release)
- Update and install
  - `apt-get update`
  - `apt-get install medici-webapp medici-extractor`
- Connect to `http://localhost:8080/medici`

# Ubuntu Package

- Change tomcat to port 80
  - `sed -i -e 's/8080/80/g' /etc/tomcat6/server.xml`
  - `sed -i -e 's/#*AUTHBIND=.*/*AUTHBIND=yes/g' /etc/default/tomcat6`
- Run Medici as root context
  - `mv /etc/tomcat6/Catalina/localhost/ROOT.xml /etc/tomcat6/Catalina/localhost/ROOT.bak`
  - `mv /etc/tomcat6/Catalina/localhost/medici.xml /etc/tomcat6/Catalina/localhost/ROOT.xml`
- Connect to `http://localhost`

# Virtual Machine

- Instructions available at:
  - <ftp://ftp.ncsa.illinois.edu/outgoing/tmclaren/Medici%20VM%20Download.pdf>
- VM creation process is manual
- New 1.2 VM coming...
- Can always update using debian packages
  - Just change `/etc/apt/sources.list.d/ncsa.list` entry from development to stable and run
  - `apt-get dist-upgrade medici-webapp medici-extractor`

# Manual Install from Build

- See wiki!
- <https://opensource.ncsa.illinois.edu/confluence/display/M MDB/Clean+Install+of+Medici+1.2>

# Manual install from source

- See development discussion

# Optimizations

- Extractor
  - Increase number of threads in server.properties
    - workers.0=1
    - workers.1=1
    - workers.2=1
    - workers.3=1
  - Memory
    - Increase jvm memory in ExtractionServer.ini
    - -Xmx1G
- Tomcat
  - Increase memory
    - Add `-Xmx1024m` to `JAVA_OPTS` in `catalina.sh`



# File Storage

- File Storage
  - Uses the file system by default
    - Set hfc.path in server.properties of webapp
  - Have used both in past:
    - Network File System (NFS)
    - Lustre File System

# Roles and Permissions

- Access control in Medici is handles using a role-based access control scheme (RBAC)
- Users have Roles
- Roles have Permissions
- New Roles can be created
- Permissions are fixed by the app
  - for example “Upload Data”
- See <https://localhost:9980/#administration>

# Batch upload using REST service

- Can use any language/script you like that knows how to speak HTTP
- See example shell script available here:
  - <https://opensource.ncsa.illinois.edu/confluence/display/MMDB/File+Upload#FileUpload-ScriptUpload>
  - Uses curl

**For more information please visit**

<http://medici.ncsa.illinois.edu>

email us at

[medici@ncsa.illinois.edu](mailto:medici@ncsa.illinois.edu)

join the discussion at

[medici-users@ncsa.illinois.edu](mailto:medici-users@ncsa.illinois.edu)