

# Booting Weblogic

Cato Aune

Jon Petter Hjulstad

SYSCO AS



Tech 14 – Super Sunday, December 7th, 2014

# Agenda

- About us and our company
- Why this presentation ?
- Involved components
- Howto
- What are the options?
- Our recommendations
- Sample script – a walkthrough
- Q&A

## Information about us

- Jon Petter Hjulstad, Dept Manager Middleware, Sysco
- Cato Aune, Senior Consultant, Sysco
- Middleware consultants – Oslo, Norway
  - Colleagues in Lima, Peru
- Focusing on BPM, SOA, WLS, EM, OVM
- Blog: <http://sysco.no/blogg/>



## Information about SYSCO

- IT company established 2004
- Continuous growth, over 100 employees
- Operations, development, consulting in technology and economics
  - Competence in database technology, middleware
  - Special focus in the energy sector
- Engineered Systems Partner of the Year Norway 2014
- 6 Locations in Norway, 1 in Peru



BKK



aibel®



# Booting Oracle WebLogic

- WebLogic - advanced and flexible
  - Makes it a bit complex
  - Many choices that has to be made
- No out-of-the-box start scripts
- Many resources on the Net
  - Some good
  - Some that might not fit your requirements
  - Some not so optimal



# Why automatic/scripted boot

- No user intervention
  - No one has to be present (physical or “virtual”)
  - Less error prone
  - Do it the same way every time
- Makes it easier to start / stop single instances for the ops staff
- Want services to be restarted automatically if needed
  
- Use what is available in WLS



# Prereqs

- WebLogic installed, domain created
- Node Manager installed and configured
  - nmEnroll
  - nmGenBootStartupProps
- For demo purposes
  - Not using SSL (SecureListener=false in nodemanager.properties)
  - Little error handling



# Sharing

- Feel free to use the scripts “as is” or as a basis for your own enhancements to fit your requirements
- All scripts, some more background information and suggestions for enhancement are in our blog <http://sysco.no/blogg>





# Components

- Node Manager
- WebLogic Scripting Tool (WLST)
- Shell scripts

# Node Manager

Node Manager is a WebLogic Server utility that enables you to

- Start
- Shut down
- Restart

Administration Server and Managed Server instances



# Node Manager

Before WebLogic 12.1.2

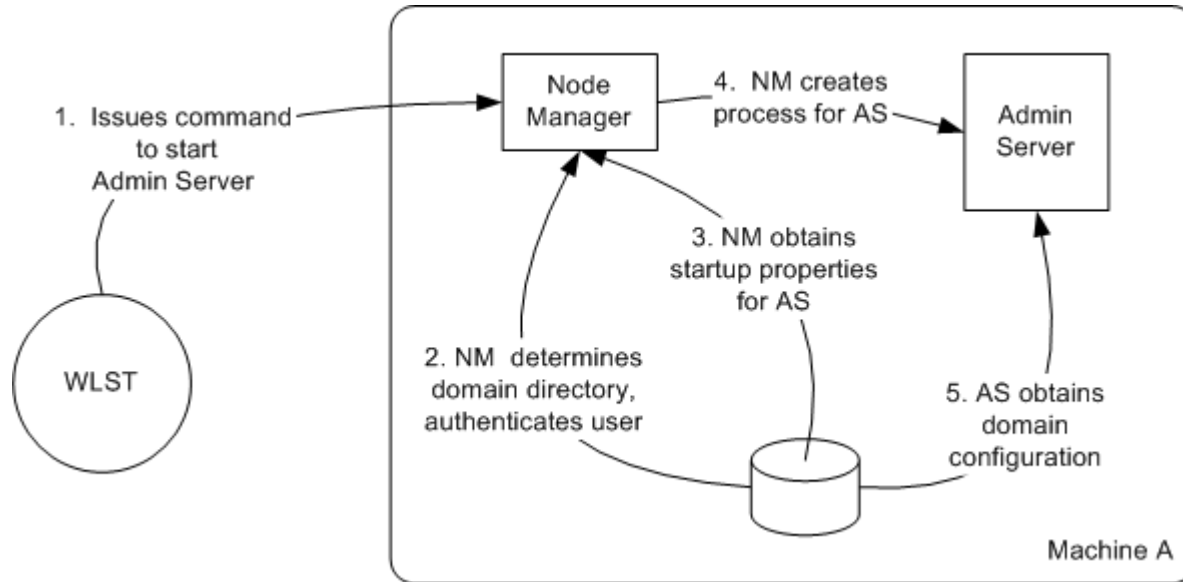
- One Node Manager per server
- Central Node Manager config

From WebLogic 12.1.2

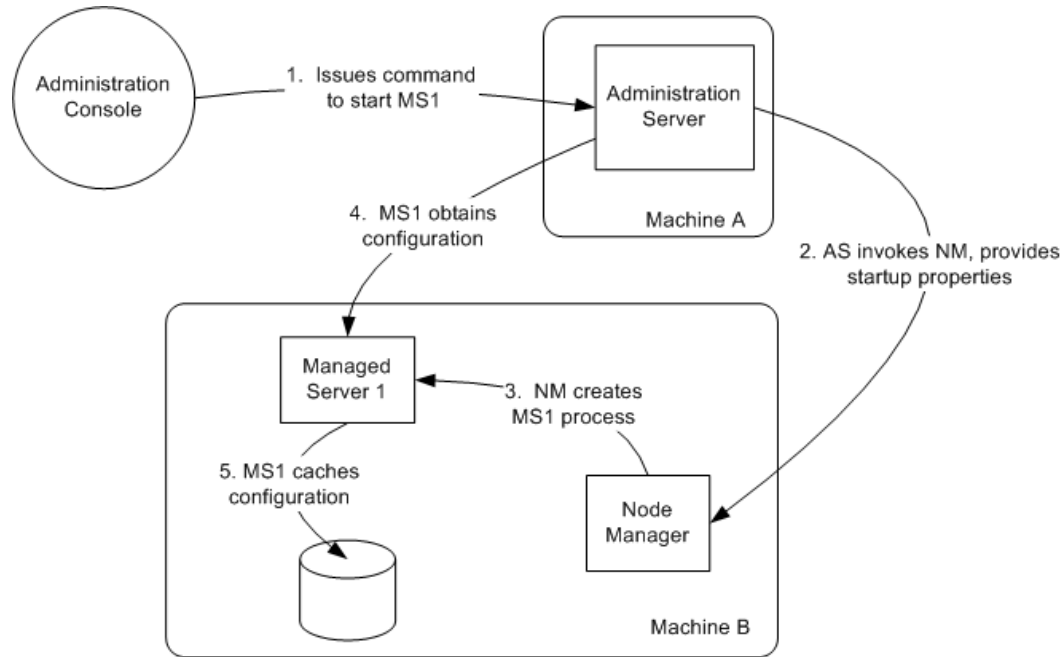
- One Node Manager per domain (default)
- Node Manager config within domain home



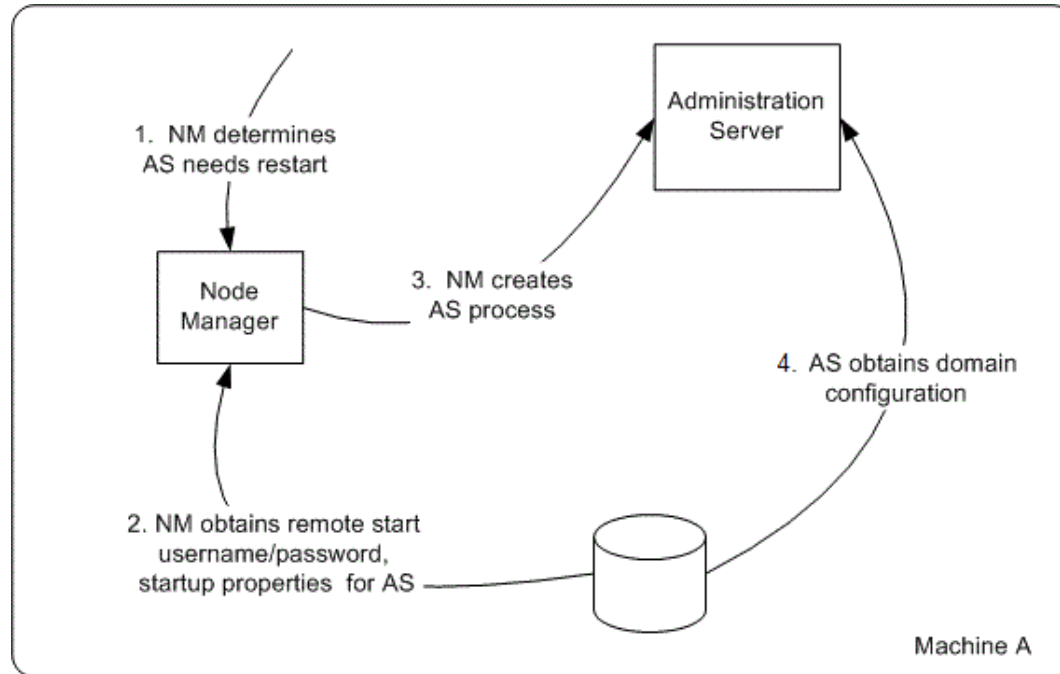
# Starting an Administration Server



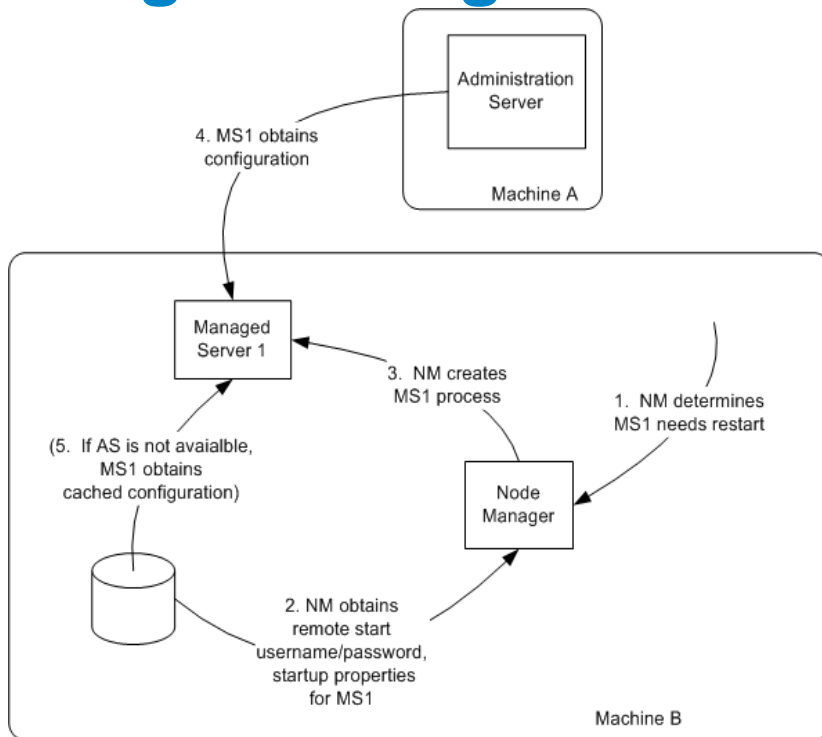
# Starting a Managed Server



# Restarting an Administration Server



# Restarting a Managed Server



# WebLogic Scripting Tool (WLST)

- WLST is based on Jython (Python)
- Can do pretty much everything with WLST
- Be sure to have correct path (source setDomainEnv.sh)
- Recording option in Admin Console, might be used as starting point for automation





# Shell scripts called during boot

- Start Node Manager
  - init.d / xinit.d scripts for Linux
  - Script to create a Windows service
- Start WebLogic
  - Custom bash/cmd script for starting the AdminServer and managed servers

# Different methods

Start script

Using WLST and Admin Server

Using WLST and Node Manager

Choose one method and stick with it



# Start scripts

Generated when a domain is created

```
<domain home>/startWebLogic.sh
```

```
<domain home>/bin/startManagedWebLogic.sh
```

Works well, but make sure to use nohup and put the process in the background

```
$ nohup startWeblogic.sh &
```



# Using WLST and Admin Server

- Possible to start AdminServer
  - Not recommended with Fusion Middleware suite products
- Connect to AdminServer to start managed servers
  - `connect(userConfigFile=userFile, userKeyFile=keyFile, url=adminUrl)`
  - `start(...)`



# Using WLST and Admin Server

- Requires
  - Running AdminServer
  - Running Node Manager
  - AdminServer communicates with Node Manager
- Node Manager sets the
  - JAVA\_VENDOR, JAVA\_HOME, JAVA\_OPTIONS
  - SECURITY\_POLICY, CLASSPATH, ADMIN\_URL



# Using WLST and Node Manager

- Connect to Node Manager
  - nmConnect
- Start AdminServer and managed servers
  - nmStart
- Does not set the variables with information from AdminServer.  
Possible to provide this information manually along with nmStart



# Starting from NodeManager

```
nmConnect(userConfigFile=nmUserFile,  
          userKeyFile=nmKeyFile, host=nmHost,  
          port=nmPort, domainName=domain,  
          domainDir=domainPath, nmType=nmType)
```

```
nmStart('AdminServer')  
nmStart('ms1')
```

# Recommendations

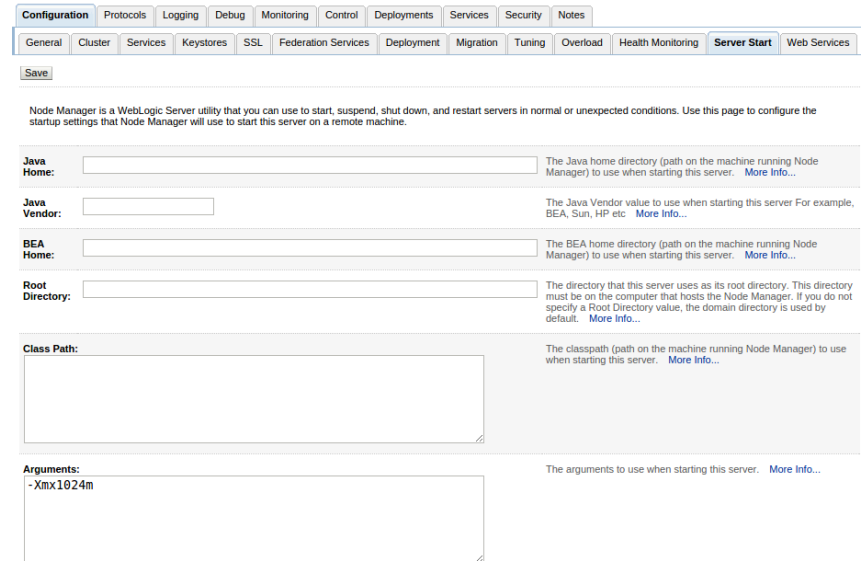
- It is recommended to always use Node Manager to start AdminServer and managed servers
- It is recommended to let Node Manager use start script (StartScriptEnabled=true)
- It is recommended to start from AdminServer to give server start arguments and SSL arguments to Node Manager





# Custom WebLogic config

- Where to put custom config depends on how WebLogic is started
  - Start via AdminServer
    - Custom config could be in config.xml and managed from Admin Console
  - Start via Node Manager
    - Custom config in config.xml will not be used



The screenshot shows the 'Server Start' configuration page in the WebLogic Administration Console. The 'Server Start' tab is selected, and the 'Save' button is visible. The page contains several configuration fields with descriptive text and links to 'More Info...':

- Java Home:** A text input field. Description: "The Java home directory (path on the machine running Node Manager) to use when starting this server. More Info..."
- Java Vendor:** A text input field. Description: "The Java Vendor value to use when starting this server For example, BEA, Sun, HP etc More Info..."
- BEA Home:** A text input field. Description: "The BEA home directory (path on the machine running Node Manager) to use when starting this server. More Info..."
- Root Directory:** A text input field. Description: "The directory that this server uses as its root directory. This directory must be on the computer that hosts the Node Manager. If you do not specify a Root Directory value, the domain directory is used by default. More Info..."
- Class Path:** A large text area. Description: "The classpath (path on the machine running Node Manager) to use when starting this server. More Info..."
- Arguments:** A text area containing the text "-Xmx1024m". Description: "The arguments to use when starting this server. More Info..."

# Custom WebLogic config

- Do not put custom config in `setDomainEnv.sh`, `startWebLogic.sh` or `startManagedWebLogic.sh`
  - it will be overwritten
- Config could be put in `setUserOverrides.sh`
  - must be in `DOMAIN_HOME/bin`



# Custom WebLogic config

- Make sure to use the generated startscript
  - StartScriptEnabled=true
- If you want to use your own (StartScriptName=myStartupScript.sh)
  - make sure to call the other script files in DOMAIN\_HOME/bin like setDomainEnv.sh, setUserOverrides.sh and/or use JAVA\_OPTIONS if it is set from AdminServer

# Our approach

- Enable start script in Node Manager  
(StartScriptEnabled=true in nodemanager.properties)
- Connect to Node Manager and start AdminServer
- Connect to AdminServer and start managed servers



## Put it together - wls.py

```
import sys
def startAdmin():
    print 'Starting AdminServer'
    nmConnect(userConfigFile=nmUserFile,
              userKeyFile=nmKeyFile, host=nmHost,
              port=nmPort, domainName=domain,
              domainDir=domainPath, nmType=nmType)
    nmStart('AdminServer')
    nmDisconnect()
    return
```



## wls.py – Part II

```
def stopAdmin():  
    print 'Stopping AdminServer'  
    connect(userConfigFile=wlsUserFile,  
            userKeyFile=wlsKeyFile, url=adminUrl)  
    shutdown('AdminServer', force='true')  
    return
```



## wls.py – Part III

```
def startManaged(managed):  
    print 'Starting ', managed  
    connect(userConfigFile=wlsUserFile,  
            userKeyFile=wlsKeyFile, url=adminUrl)  
    start(managed)  
    disconnect()  
    return
```



## wls.py – Part IV

```
def stopManaged(managed):  
    print 'Stopping ', managed  
    connect(userConfigFile=wlsUserFile,  
            userKeyFile=wlsKeyFile, url=adminUrl)  
    shutdown(managed, force='true')  
    disconnect()  
    return
```





## wls.py – Part V

```
if ((len(sys.argv) < 2) | (len(sys.argv) > 3)):
    print 'Wrong number of arguments'
elif (sys.argv[1] == 'startadmin'):
    startAdmin()
elif (sys.argv[1] == 'stopadmin'):
    stopAdmin()
elif (sys.argv[1] == 'start'):
    startManaged(sys.argv[2])
elif (sys.argv[1] == 'stop'):
    stopManaged(sys.argv[2])
```



## startall.sh

```
wlst.sh -loadProperties config.properties -  
skipWLSModuleScanning wls.py startadmin
```

```
wlst.sh -loadProperties config.properties -  
skipWLSModuleScanning wls.py start ms1
```



# stopall.sh

```
wlst.sh -loadProperties config.properties -  
skipWLSModuleScanning wls.py stop ms1
```

```
wlst.sh -loadProperties config.properties -  
skipWLSModuleScanning wls.py stopadmin
```



# config.properties

```
adminUrl=t3://wls12c.dev.sysco.no:7001
nmHost=wls12c.dev.sysco.no
nmPort=5556
nmUserFile=/u01/app/oracle/config/nmUserFile
nmKeyFile=/u01/app/oracle/config/nmKeyFile
nmType=plain
wlsUserFile=/u01/app/oracle/config/wlsUserFile
wlsKeyFile=/u01/app/oracle/config/wlsKeyFile
domain=mydomain
domainPath=/u01/app/oracle/u_p/domains/mydomain
```



# Encrypt credentials in 11g

Deprecated in 12c but still works

For Node Manager:

```
$ java weblogic.Admin
```

```
-username nodemanager
```

```
-userconfigfile /u01/app/oracle/config/nmUserFile -userkeyfile  
/u01/app/oracle/config/nmKeyFile STOREUSERCONFIG
```

# Encrypt credentials in 11g

Enter the password for user nodemanager :

Creating the key file can reduce the security of your system if it is not kept in a secured location after it is created. Do you want to create the key file? y or n           **y**

# Encrypt credentials in 11g

For WebLogic:

```
$ java weblogic.Admin
```

```
-username weblogic
```

```
-userconfigfile /u01/app/oracle/config/wlsUserFile -userkeyfile  
/u01/app/oracle/config/wlsKeyFile STOREUSERCONFIG
```



# Encrypt credentials in 12c

```
wls:/offline> nmConnect(  
    'nodemanager','welcome1','localhost',5556,'mydomain',  
    '/u01/app/oracle/user_projects/domains/mydomain',  
    'plain')
```

Currently connected to Node Manager to monitor the domain mydomain.





## Encrypt credentials in 12c - NM

```
wls:/mydomain/serverConfig> storeUserConfig(  
    '/u01/app/oracle/config/nmUserFile',  
    '/u01/app/oracle/config/nmKeyFile',  
    'true')
```

Creating the key file can reduce the security of your system if it is not kept in a secured location after it is created. Do you want to create the key file? y or n:y

## Encrypt credentials in 12c - WLS

```
wls:/mydomain/serverConfig> storeUserConfig(  
    '/u01/app/oracle/config/wlsUserFile',  
    '/u01/app/oracle/config/wlsKeyFile',  
    'false')
```

Creating the key file can reduce the security of your system if it is not kept in a secured location after it is created. Do you want to create the key file? y or n:y

# OS boot scripts - Linux

/etc/init.d/nodemanager (depends on network)

Script is available in the Oracle documentation,  
but you have to do some modifications

[http://docs.oracle.com/middleware/1213/wls/NODEM/java\\_nodemgr.htm#BABJIDFD](http://docs.oracle.com/middleware/1213/wls/NODEM/java_nodemgr.htm#BABJIDFD)

/etc/init.d/weblogic (depends on nodemanager)

```
# Required-Start: $nodemanager
```

```
PROGRAM_START="$BOOT_HOME/startall.sh"
```

```
PROGRAM_STOP="$BOOT_HOME/stopall.sh"
```



# Q&A



# Thanks for attending!

- Feel free to contact us!
- <https://twitter.com/jphjulstad>
- <https://twitter.com/catoaune>
- Resources:
- <http://docs.oracle.com/middleware/1213/wls/index.html>
- <http://sysco.no/blogg>



# OS boot scripts - Windows

Starting Node Manager as a Windows service is supported out-of-the-box.

Follow the instructions in the documentation (or on the next slides)

NB!

-Xrs JVM property for each Managed Server that will be under Node Manager control.



# Windows - Node Manager

1. Log in to the machine with Administrator privileges.
2. Open a DOS command prompt window.
3. Change to the *DOMAIN\_HOME*\bin directory.



# Windows - Node Manager

4. Enter the following command:

```
installNodeMgrSvc.cmd
```

5. After a few seconds, the following message is displayed:

Oracle WebLogic <*domain-name*> NodeManager installed.

