##############################################

#### Set Variables

##############################################

# General

# Do not write trailing / on directories path, should be /home/oracle instead of /home/oracle/

mwHome='/opt/11.1.1.7/middleware'

javaVendor='Oracle'

javaHome='/opt/11.1.1.7/jrockit'

compDomain='mydom.com'

domainName='soa\_domain'

rootDomainsDir='/opt/11.1.1.7/admin/domains'

adminUrl='adminsoa-prod.mydom.com:7001'

wlUser='weblogic'

wlPwd='xxxx'

# LeasingDS

leasingPwd='xxxx'

leasingUrl='jdbc:oracle:thin:@dbserver1.mydom.com:1521/soainfra'

leasingUser='leasing'

leasingISGridLink='false'

#leasingOnsList='vip-as.mydom.com:6251,vip-as.mydom.com:6251'

# Keystores (created with soaScript.sh)

appKsPwd='soaKeystorePwd'

trustKsPwd='soaKeystorePwd'

prKeyPwd='soaKeystorePwd'

#AdminServer

adminKeyAlias='appIdentity001'

adminMemArgs='-Xms1536m -Xmx1536m'

#Soa

soaFrontEnd='soa.mydom.com'

soaCoherencePort='8089' # Caution: if there are other coherence clusters on the network with same port, choose a different port

soaHost1='soaprod-vip1'

soaHost2='soaprod-vip2'

soaMemArgs=' -Xms4096m -Xmx4096m'

# Bam

bamMemArgs=' -Xms2048m -Xmx2048m'

# WSM

wsmMemArgs=' -Xms1024m -Xmx1024m'

# Machines

machine1='testserver1'

machine2='testserver2'

################################################

####### Do not edit these following variables

soaurl1=soaHost1+'.'+compDomain

soaurl2=soaHost2+'.'+compDomain

leasingDsName='/SystemResources/Leasing'

soa1Args=soaMemArgs + ' -Dtangosol.coherence.wka1='+ soaurl1 + ' -Dtangosol.coherence.wka2=' + soaurl2 + ' -Dtangosol.coherence.localhost=' +soaurl1 + ' -Dtangosol.coherence.localport=' + soaCoherencePort + ' -Dtangosol.coherence.wka1.port=' + soaCoherencePort + ' -Dtangosol.coherence.wka2.port=' + soaCoherencePort

soa2Args=soaMemArgs + ' -Dtangosol.coherence.wka1='+ soaurl1 + ' -Dtangosol.coherence.wka2=' + soaurl2 + ' -Dtangosol.coherence.localhost=' +soaurl2 + ' -Dtangosol.coherence.localport=' + soaCoherencePort + ' -Dtangosol.coherence.wka1.port=' + soaCoherencePort + ' -Dtangosol.coherence.wka2.port=' + soaCoherencePort

machObjName1='com.bea:Name=' + machine1 + ',Type=UnixMachine'

machObjName2='com.bea:Name=' + machine2 + ',Type=UnixMachine'

securityPolicyFile=mwHome+'/wlserver\_10.3/server/lib/weblogic.policy'

rootDirectoryAserver=rootDomainsDir + '/'+domainName+ '/aserver/'+domainName

rootDirectoryMserver=rootDomainsDir + '/'+domainName+ '/mserver/'+domainName

soa1StageDirectory=rootDirectoryMserver + '/servers/WLS\_SOA1/stage'

soa2StageDirectory=rootDirectoryMserver + '/servers/WLS\_SOA2/stage'

bam1StageDirectory=rootDirectoryMserver + '/servers/WLS\_BAM1/stage'

bam2StageDirectory=rootDirectoryMserver + '/servers/WLS\_BAM2/stage'

wsm1StageDirectory=rootDirectoryMserver + '/servers/WLS\_WSM1/stage'

wsm2StageDirectory=rootDirectoryMserver + '/servers/WLS\_WSM2/stage'

uploadDirectory=rootDirectoryAserver + '/servers/AdminServer/upload'

rootDirectoryAserver=rootDomainsDir + '/'+domainName+ '/aserver/'+domainName

rootDirectoryMserver=rootDomainsDir + '/'+domainName+ '/mserver/'+domainName

soaTlogsDir=rootDomainsDir+'/'+domainName+'/aserver/SOA\_CLUSTER/tlogs'

bamTlogsDir=rootDomainsDir+'/'+domainName+'/aserver/BAM\_CLUSTER/tlogs'

wsmTlogsDir=rootDomainsDir+'/'+domainName+'/aserver/WSM\_CLUSTER/tlogs'

appKeystore=rootDomainsDir+'/'+domainName+'/aserver/certs/appIdentityKeystore.jks'

trustKeystore=rootDomainsDir+'/'+domainName+'/aserver/certs/appTrustKeyStore.jks'

soa1KeyAlias='appIdentity004'

soa2KeyAlias='appIdentity005'

bam1KeyAlias='appIdentity006'

bam2KeyAlias='appIdentity003'

wsm1KeyAlias='appIdentity002'

wsm2KeyAlias='appIdentity003'

########################################################

# Starts domain edition

########################################################

## Domain Connection

connect(wlUser,wlPwd,adminUrl)

edit()

startEdit()

## Domain edit

# Set weblogic plugin enabled

cd('/WebAppContainer/'+domainName)

cmo.setWeblogicPluginEnabled(true)

cmo.setXPoweredByHeaderLevel('SHORT')

# Create leasing datasource

cd('/')

cmo.createJDBCSystemResource('Leasing')

cd('/JDBCSystemResources/Leasing/JDBCResource/Leasing')

cmo.setName('Leasing')

cd('/JDBCSystemResources/Leasing/JDBCResource/Leasing/JDBCDataSourceParams/Leasing')

set('JNDINames',jarray.array([String('jdbc/Leasing')], String))

cd('/JDBCSystemResources/Leasing/JDBCResource/Leasing/JDBCDriverParams/Leasing')

cmo.setUrl(leasingUrl)

cmo.setDriverName('oracle.jdbc.OracleDriver')

set('Password',leasingPwd)

cd('/JDBCSystemResources/Leasing/JDBCResource/Leasing/JDBCConnectionPoolParams/Leasing')

cmo.setTestTableName('SQL SELECT 1 FROM DUAL')

cd('/JDBCSystemResources/Leasing/JDBCResource/Leasing/JDBCDriverParams/Leasing/Properties/Leasing')

cmo.createProperty('user')

cd('/JDBCSystemResources/Leasing/JDBCResource/Leasing/JDBCDriverParams/Leasing/Properties/Leasing/Properties/user')

cmo.setValue(leasingUser)

cd('/JDBCSystemResources/Leasing/JDBCResource/Leasing/JDBCDataSourceParams/Leasing')

cmo.setGlobalTransactionsProtocol('None')

cd('/JDBCSystemResources/Leasing/JDBCResource/Leasing/JDBCOracleParams/Leasing')

if leasingISGridLink == 'true':

 cmo.setFanEnabled(true)

 cmo.setOnsWalletFile('')

 cmo.unSet('OnsWalletPasswordEncrypted')

 cmo.setOnsNodeList(leasingOnsList)

cd('/SystemResources/Leasing')

set('Targets',jarray.array([ObjectName('com.bea:Name=BAM\_CLUSTER,Type=Cluster'), ObjectName('com.bea:Name=SOA\_CLUSTER,Type=Cluster')], ObjectName))

##Start servers edit

# edit AdminServer

cd ('/Servers/AdminServer/ServerStart/AdminServer')

cmo.setRootDirectory(rootDirectoryAserver)

cmo.setBeaHome(mwHome)

cmo.setJavaVendor(javaVendor)

cmo.setJavaHome(javaHome)

cmo.setSecurityPolicyFile(securityPolicyFile)

cmo.setArguments(adminMemArgs)

cmo.setClassPath('')

cd ('/Servers/AdminServer')

cmo.setKeyStores('CustomIdentityAndCustomTrust')

cmo.setCustomIdentityKeyStoreFileName(appKeystore)

cmo.setCustomIdentityKeyStorePassPhrase(appKsPwd)

cmo.setCustomIdentityKeyStoreType('jks')

cmo.setCustomTrustKeyStoreFileName(trustKeystore)

cmo.setCustomTrustKeyStorePassPhrase(trustKsPwd)

cmo.setCustomTrustKeyStoreType('jks')

cd ('/Servers/AdminServer/SSL/AdminServer')

cmo.setServerPrivateKeyAlias(adminKeyAlias)

cmo.setServerPrivateKeyPassPhrase(prKeyPwd)

# edit WLS\_SOA1

cd ('/Servers/WLS\_SOA1/ServerStart/WLS\_SOA1')

cmo.setRootDirectory(rootDirectoryMserver)

cmo.setBeaHome(mwHome)

cmo.setJavaVendor(javaVendor)

cmo.setJavaHome(javaHome)

cmo.setSecurityPolicyFile(securityPolicyFile)

cmo.setArguments(soa1Args)

cmo.setClassPath('')

cd ('/Servers/WLS\_SOA1')

cmo.setKeyStores('CustomIdentityAndCustomTrust')

cmo.setCustomIdentityKeyStoreFileName(appKeystore)

cmo.setCustomIdentityKeyStorePassPhrase(appKsPwd)

cmo.setCustomIdentityKeyStoreType('jks')

cmo.setCustomTrustKeyStoreFileName(trustKeystore)

cmo.setCustomTrustKeyStorePassPhrase(trustKsPwd)

cmo.setCustomTrustKeyStoreType('jks')

cmo.setReplicationGroup('WLS\_SOA1')

cmo.setPreferredSecondaryGroup('WLS\_SOA2')

cmo.setAutoMigrationEnabled(true)

set('CandidateMachines',jarray.array([ObjectName(machObjName2)], ObjectName))

cmo.setStagingDirectoryName(soa1StageDirectory)

cmo.setUploadDirectoryName(uploadDirectory)

cd ('/Servers/WLS\_SOA1/DefaultFileStore/WLS\_SOA1')

cmo.setDirectory(soaTlogsDir)

cd ('/Servers/WLS\_SOA1/SSL/WLS\_SOA1')

cmo.setServerPrivateKeyAlias(soa1KeyAlias)

cmo.setServerPrivateKeyPassPhrase(prKeyPwd)

# edit WLS\_SOA2

cd ('/Servers/WLS\_SOA2/ServerStart/WLS\_SOA2')

cmo.setRootDirectory(rootDirectoryMserver)

cmo.setBeaHome(mwHome)

cmo.setJavaVendor(javaVendor)

cmo.setJavaHome(javaHome)

cmo.setSecurityPolicyFile(securityPolicyFile)

cmo.setArguments(soa2Args)

cmo.setClassPath('')

cd ('/Servers/WLS\_SOA2')

cmo.setKeyStores('CustomIdentityAndCustomTrust')

cmo.setCustomIdentityKeyStoreFileName(appKeystore)

cmo.setCustomIdentityKeyStorePassPhrase(appKsPwd)

cmo.setCustomIdentityKeyStoreType('jks')

cmo.setCustomTrustKeyStoreFileName(trustKeystore)

cmo.setCustomTrustKeyStorePassPhrase(trustKsPwd)

cmo.setCustomTrustKeyStoreType('jks')

cmo.setReplicationGroup('WLS\_SOA2')

cmo.setPreferredSecondaryGroup('WLS\_SOA1')

cmo.setAutoMigrationEnabled(true)

set('CandidateMachines',jarray.array([ObjectName(machObjName1)], ObjectName))

cmo.setStagingDirectoryName(soa2StageDirectory)

cmo.setUploadDirectoryName(uploadDirectory)

cd ('/Servers/WLS\_SOA2/DefaultFileStore/WLS\_SOA2')

cmo.setDirectory(soaTlogsDir)

cd ('/Servers/WLS\_SOA2/SSL/WLS\_SOA2')

cmo.setServerPrivateKeyAlias(soa2KeyAlias)

cmo.setServerPrivateKeyPassPhrase(prKeyPwd)

# edit WLS\_BAM1

cd ('/Servers/WLS\_BAM1/ServerStart/WLS\_BAM1')

cmo.setRootDirectory(rootDirectoryMserver)

cmo.setBeaHome(mwHome)

cmo.setJavaVendor(javaVendor)

cmo.setJavaHome(javaHome)

cmo.setSecurityPolicyFile(securityPolicyFile)

cmo.setArguments(bamMemArgs)

cmo.setClassPath('')

cd ('/Servers/WLS\_BAM1')

cmo.setKeyStores('CustomIdentityAndCustomTrust')

cmo.setCustomIdentityKeyStoreFileName(appKeystore)

cmo.setCustomIdentityKeyStorePassPhrase(appKsPwd)

cmo.setCustomIdentityKeyStoreType('jks')

cmo.setCustomTrustKeyStoreFileName(trustKeystore)

cmo.setCustomTrustKeyStorePassPhrase(trustKsPwd)

cmo.setCustomTrustKeyStoreType('jks')

cmo.setReplicationGroup('WLS\_BAM1')

cmo.setPreferredSecondaryGroup('WLS\_BAM2')

cmo.setAutoMigrationEnabled(true)

cmo.setStagingDirectoryName(bam1StageDirectory)

cmo.setUploadDirectoryName(uploadDirectory)

cd ('/Servers/WLS\_BAM1/DefaultFileStore/WLS\_BAM1')

cmo.setDirectory(bamTlogsDir)

cd ('/Servers/WLS\_BAM1/SSL/WLS\_BAM1')

cmo.setServerPrivateKeyAlias(bam1KeyAlias)

cmo.setServerPrivateKeyPassPhrase(prKeyPwd)

# edit WLS\_BAM2

cd ('/Servers/WLS\_BAM2/ServerStart/WLS\_BAM2')

cmo.setRootDirectory(rootDirectoryMserver)

cmo.setBeaHome(mwHome)

cmo.setJavaVendor(javaVendor)

cmo.setJavaHome(javaHome)

cmo.setSecurityPolicyFile(securityPolicyFile)

cmo.setArguments(bamMemArgs)

cmo.setClassPath('')

cd ('/Servers/WLS\_BAM2')

cmo.setKeyStores('CustomIdentityAndCustomTrust')

cmo.setCustomIdentityKeyStoreFileName(appKeystore)

cmo.setCustomIdentityKeyStorePassPhrase(appKsPwd)

cmo.setCustomIdentityKeyStoreType('jks')

cmo.setCustomTrustKeyStoreFileName(trustKeystore)

cmo.setCustomTrustKeyStorePassPhrase(trustKsPwd)

cmo.setCustomTrustKeyStoreType('jks')

cmo.setReplicationGroup('WLS\_BAM2')

cmo.setPreferredSecondaryGroup('WLS\_BAM1')

cmo.setAutoMigrationEnabled(false)

set('CandidateMachines',jarray.array([ObjectName(machObjName1)], ObjectName))

cmo.setStagingDirectoryName(bam2StageDirectory)

cmo.setUploadDirectoryName(uploadDirectory)

cd ('/Servers/WLS\_BAM2/DefaultFileStore/WLS\_BAM2')

cmo.setDirectory(bamTlogsDir)

cd ('/Servers/WLS\_BAM2/SSL/WLS\_BAM2')

cmo.setServerPrivateKeyAlias(bam2KeyAlias)

cmo.setServerPrivateKeyPassPhrase(prKeyPwd)

# edit WLS\_WSM1

cd ('/Servers/WLS\_WSM1/ServerStart/WLS\_WSM1')

cmo.setRootDirectory(rootDirectoryMserver)

cmo.setBeaHome(mwHome)

cmo.setJavaVendor(javaVendor)

cmo.setJavaHome(javaHome)

cmo.setSecurityPolicyFile(securityPolicyFile)

cmo.setArguments(wsmMemArgs)

cmo.setClassPath('')

cd ('/Servers/WLS\_WSM1')

cmo.setKeyStores('CustomIdentityAndCustomTrust')

cmo.setCustomIdentityKeyStoreFileName(appKeystore)

cmo.setCustomIdentityKeyStorePassPhrase(appKsPwd)

cmo.setCustomIdentityKeyStoreType('jks')

cmo.setCustomTrustKeyStoreFileName(trustKeystore)

cmo.setCustomTrustKeyStorePassPhrase(trustKsPwd)

cmo.setCustomTrustKeyStoreType('jks')

cmo.setReplicationGroup('WLS\_WSM1')

cmo.setPreferredSecondaryGroup('WLS\_WSM2')

cmo.setStagingDirectoryName(wsm1StageDirectory)

cmo.setUploadDirectoryName(uploadDirectory)

cd ('/Servers/WLS\_WSM1/DefaultFileStore/WLS\_WSM1')

cmo.setDirectory(wsmTlogsDir)

cd ('/Servers/WLS\_WSM1/SSL/WLS\_WSM1')

cmo.setServerPrivateKeyAlias(wsm1KeyAlias)

cmo.setServerPrivateKeyPassPhrase(prKeyPwd)

# edit WLS\_WSM2

cd ('/Servers/WLS\_WSM2/ServerStart/WLS\_WSM2')

cmo.setRootDirectory(rootDirectoryMserver)

cmo.setBeaHome(mwHome)

cmo.setJavaVendor(javaVendor)

cmo.setJavaHome(javaHome)

cmo.setSecurityPolicyFile(securityPolicyFile)

cmo.setArguments(wsmMemArgs)

cmo.setClassPath('')

cd ('/Servers/WLS\_WSM2')

cmo.setKeyStores('CustomIdentityAndCustomTrust')

cmo.setCustomIdentityKeyStoreFileName(appKeystore)

cmo.setCustomIdentityKeyStorePassPhrase(appKsPwd)

cmo.setCustomIdentityKeyStoreType('jks')

cmo.setCustomTrustKeyStoreFileName(trustKeystore)

cmo.setCustomTrustKeyStorePassPhrase(trustKsPwd)

cmo.setCustomTrustKeyStoreType('jks')

cmo.setReplicationGroup('WLS\_WSM2')

cmo.setPreferredSecondaryGroup('WLS\_WSM1')

cmo.setStagingDirectoryName(wsm2StageDirectory)

cmo.setUploadDirectoryName(uploadDirectory)

cd ('/Servers/WLS\_WSM2/DefaultFileStore/WLS\_WSM2')

cmo.setDirectory(wsmTlogsDir)

cd ('/Servers/WLS\_WSM2/SSL/WLS\_WSM2')

cmo.setServerPrivateKeyAlias(wsm2KeyAlias)

cmo.setServerPrivateKeyPassPhrase(prKeyPwd)

# edit SOA\_CLUSTER

cd('/Clusters/SOA\_CLUSTER')

cmo.setFrontendHTTPPort(80)

cmo.setFrontendHTTPSPort(443)

cmo.setFrontendHost(soaFrontEnd)

set('CandidateMachinesForMigratableServers',jarray.array([ObjectName(machObjName1), ObjectName(machObjName2)], ObjectName))

cmo.setDataSourceForAutomaticMigration(getMBean(leasingDsName))

cd('/Clusters/SOA\_CLUSTER')

cmo.setWeblogicPluginEnabled(true)

# edit BAM\_CLUSTER

cd('/Clusters/BAM\_CLUSTER')

set('CandidateMachinesForMigratableServers',jarray.array([ObjectName(machObjName1), ObjectName(machObjName2)], ObjectName))

cmo.setDataSourceForAutomaticMigration(getMBean(leasingDsName))

cmo.setWeblogicPluginEnabled(true)

# Save changes

save()

activate()

disconnect()