

Table of Contents

Deployment SCENARIO: "Mixed Mode".....	1
Before Starting.....	1
Service Installation.....	1
Service Configuration.....	1
CONFIGURE WN with Torque/Maui (SL5, follow same rules for SL6 -> check documentation).....	1
Configuration ON Torque server.....	4
Service Testing.....	7
Notes & Service Troubleshooting.....	7

Deployment SCENARIO: "Mixed Mode"

Before Starting

1. HOSTNAME SL5: emitestbed34.cnaf.infn.it + 2 IP for virtual machines emitestbed35.cnaf.infn.it, emitestbed36.cnaf.infn.it
2. HOSTNAME SL6: cert-06.cnaf.infn.it + 2 IP for virtual machines emitestbed19.cnaf.infn.it, emitestbed20.cnaf.infn.it
3. OS: SL5 / SL6 X86_64 Installed
4. No Host certificate required
5. No Network Bridge configured
6. Hardware must support virtualization (please run `grep --color vmx /proc/cpuinfo`)

Service Installation

1. Repositories (see EMI basic configuration): egi-trustanchors.repo + emi-2-rc-sl5.repo + epel.repo
 1. `$> yum clean all`
 2. `$> yum makecache`
 3. `$> yum install ca-policy-egi-core`
 4. `$> yum install lcg-CA`
 5. `$> yum install yum-protectbase.noarch`
2. INSTALLING WN + TORQUE
 1. `$> yum install emi-wn emi-torque-client`
 2. `$> yum install emi-release`
 3. `$> yum install kvm-qemu-img`
 4. `$> yum install kmod-kvm`
 5. `$> yum install libvirt`
 6. `$> yum install python-virtinst`
 7. `$> yum install pyOpenSSL`
3. INSTALLING WNODES
 1. `$> yum install wnodes*`

Service Configuration

CONFIGURE WN with Torque/Maui (SL5, follow same rules for SL6 -> check documentation)

1. Install WN with torque following deployment logbook here WN deployment logbook, excluding GLEXEC, MPI
 1. `$> cp /etc/munge/munge.key from the CE`
 2. `$> chown munge /etc/munge/munge.key`
 3. `$> /etc/init.d/munge start`
2. Wnodes specific configuration
 1. `$> -> /etc/wnodes/nameserver/mac_list.ini`

```
[DEFAULT_VLAN]
network_type = OPEN
bait_host =
vm_host = emitestbed35.cnaf.infn.it^00:16:3E:MACADDRESS;emitestbed36.cnaf.infn.
```

1. `$> grep -v "#" /etc/wnodes/nameserver/wnodes_hv_config.ini`

[HV_CONF]

```
HV_PORT=8222
BAIT_PORT=8111
LOG_FILE_NAME=wnodes_hv.log
MAX_LOG_FILE_SIZE=100000
MAX_COUNT_LOG_FILE=5
LOCAL_REPO_DIR=/usr/local/wnodes/repo
BAIT_IMG_TAG=wnodes_sl5_bait
BAIT_VM_RAM=800
HOST_GROUP_EMITESTBED=emitestbed*
ENABLED_VLAN_GROUP_EMITESTBED=DEFAULT_VLAN
SSH_KEY_FILE=/root/.ssh/hv_id_rsa
USE_LVM=NO
VOLUME_GROUP=vg0
SERVICE_NIC_IP=10.1.1.2
SERVICE_NIC_IP_MASK=255.255.255.0
DNS_RANGE=10.1.1.3,10.1.1.30
DNS_LEASE_TIME=5m
ENABLE_MIXED_MODE=yes
```

1. 1. `$> grep -v "#" /etc/wnodes/nameserver/wnodes_bait_config.ini`

```
[BAIT_CONF]
min_vm_cpu = 1
default_vm_bandwidth = 50
min_vm_mem = 1500
reservation_length = 1200
max_vm_bandwidth = 100
log_file_name = wnodes_bait.log
status_retry_count = 3
max_vm_storage = 15
batch_system_type = PBS
max_log_file_size = 100000
enabled_vlan_group_emitestbed = DEFAULT_VLAN
lsf_profile = /etc/profile.d/lsf.sh
max_vm_mem = 2000
min_vm_bandwidth = 10
max_vm_cpu = 1
enable_mixed_mode = yes
type = BATCH;BATCH_REAL
default_vm_img = wnodes-emi-images
scheduling_interval = 60
hv_port = 8222
use_lvm = NO
default_vm_storage = 10
min_vm_storage = 10
max_count_log_file = 5
host_group_emitestbed = emitestbed*
bait_port = 8111
default_job_type = BATCH
default_vm_cpu = 1
px_failed_return_status = 3
vm_unreach_timeout = 600
default_vm_mem = 2000
```

1. 1. `$> grep -v "#" /etc/wnodes/manager/wnodes.ini`

```
[NAMESERVER]
NS_HOST = emitestbed34.cnaf.infn.it
NS_PORT = 8219
```

1. 1. `$> service wnodes_nameserver start`
2. `$> wnodes_manager -a wnodes-emi-images http
torquemada.cr.cnaf.infn.it/wnodes/wnodes_sl5_wn_emi x86_64 raw
/dev/mapper/VolGroup00-LogVol00`

3. `$> [root@emitestbed34 ~]# wnodes_manager -l`

```
tag                loca  path                                                                 arch  form  dev
wnodes-emi-images  http  torquemada.cr.cnaf.infn.it/wnodes/wnodes_sl5_wn_emi  x86_64  raw  /dev/
```

1. 1. `$> grep -v "#" /etc/wnodes/hypervisor/wnodes.ini`

```
[NAMESERVER]
NS_HOST = emitestbed34.cnaf.infn.it
NS_PORT = 8219
```

1. 1. `$> grep -v "#" /etc/wnodes/bait/wnodes.ini`

```
[NAMESERVER]
NS_HOST = emitestbed34.cnaf.infn.it
NS_PORT = 8219
```

1. 1. `$> mkdir -p /usr/local/wnodes/repo --> workaround`
2. `$> service libvirtd start`
3. `$> service wnodes_hypervisor start --> this will start the process wnodes_bait`
4. `$> SOME CHECKS`

```
[root@emitestbed34 ~]# wnodes_manager -t all
emitestbed34 :
[root@emitestbed34 ~]# wnodes_manager -s emitestbed34
Bait                : emitestbed34;
Bait status         : ['OPEN', 'Everything is OK, the BAIT process can start', 1347887272.8612399, 0,
```

No active jobs

```
[root@emitestbed34 ~]# wnodes_manager -S emitestbed34
[root@emitestbed34 ~]#
```

1. 1. `$> chmod 500 /usr/bin/wnodes/site_specific/wnodes_preexec`
2. `$> wget patch_wnodes_preexec.txt applying patch`
3. `$> cp /etc/wnodes/site_specific/wnodes_preexec.conf.tpl /etc/wnodes/site_specific/wnodes_preexec.conf`
4. `$> [root@emitestbed34 ~]# grep -v "#" /etc/wnodes/site_specific/wnodes_preexec.conf`
-----> NOTE : this file must be the same on the template virtual image

```
[general]
TMPFILE=/tmp/my_bait
LOCAL_DOMAIN=cnaf.infn.it
NS_HOST=emitestbed34.cnaf.infn.it
NS_PORT=8219
BAIT_PORT=8111
FAIL_RETURN_STATUS = 3
```

```
[default]
TYPE=BATCH
IMG=wnodes-emi-images
NETWORK_TYPE=OPEN
CPU=1
MEM=1900
STORAGE=30
ENABLEVIRTIO=YES
BANDWIDTH=10
PX_SCRIPT=/usr/bin/wnodes/site_specific/wnodes_preexec
```

```
['dongiovanni']
TYPE=BATCH
IMG=wnodes-emi-images
```

```
NETWORK_TYPE=OPEN
CPU=1
MEM=2500
STORAGE=30
ENABLEVIRTIO=YES
BANDWIDTH=10
PX_SCRIPT=
```

1. 1. \$> cat /var/torque/mom_priv/prologue

```
#!/bin/bash
while [ ! -f /usr/bin/wnodes/site_specific/wnodes_preexec ]; do sleep 3 ; done
sleep 10
/usr/bin/wnodes/site_specific/wnodes_preexec -f /etc/wnodes/site_specific/wnodes_preexec.conf --j
```

1. 1. \$> chmod 500 /var/torque/mom_priv/prologue
2. Wnodes specific configuration: WN image

Configuration ON Torque server

```
[root@emi-demo13 ~]# cat /etc/cron.d/fix_wnodes_job
*/5 * * * root /usr/bin/fix_jobs_mauai.sh
[root@emi-demo13 ~]# cat /usr/bin/fix_jobs_mauai.sh
#!/bin/bash
for i in diagnose -q | grep -P -i "Hold|Def" | awk {print $2}
do
releasehold $i
done
for i in showq | grep BatchHold | awk {print $1}
do
releasehold $i
done
```

```
[root@emi-demo13 ~]# cat siteinfo/wnodes_queue_command
create queue emiwnodes
set queue qwnodes queue_type = Execution
set queue qwnodes Priority = 1000000
set queue qwnodes max_running = 80
set queue qwnodes resources_max.cput = 100:00:00
set queue qwnodes resources_max.walltime = 100:00:00
set queue qwnodes resources_default.neednodes = cloudf
set queue qwnodes enabled = True
set queue qwnodes started = True
```

```
[root@emi-demo13 ~]# cat siteinfo/wnodes_queue_commandsl6
create queue qwnodessl6
set queue qwnodessl6 queue_type = Execution
set queue qwnodessl6 Priority = 1000000
set queue qwnodessl6 max_running = 80
set queue qwnodessl6 resources_max.cput = 100:00:00
set queue qwnodessl6 resources_max.walltime = 100:00:00
set queue qwnodessl6 resources_default.neednodes = cloudfsl6
set queue qwnodessl6 enabled = True
set queue qwnodessl6 started = True
```

```
[root@emi-demo13 ~]# qmgr < /root/siteinfo/wnodes_queue_command
Max open servers: 9
create queue qwnodes
set queue qwnodes queue_type = Execution
set queue qwnodes Priority = 1000000
set queue qwnodes max_running = 80
set queue qwnodes resources_max.cput = 100:00:00
set queue qwnodes resources_max.walltime = 100:00:00
set queue qwnodes resources_default.neednodes = cloudf
set queue qwnodes enabled = True
```

WnODesLogBook < EMI < TWiki

```
set queue qwnodes started = True
```

```
[root@emi-demo13 ~]# qmgr < /root/siteinfo/wnodes_queue_commandsl6  
.....output
```

```
.....
```

```
[root@emi-demo13 ~]# set qwnodes cloudtf resources_default.neednodes = cloudtf  
[root@emi-demo13 ~]# qmgr -c "set server managers += root@emitestbed35.cnaf.infn.it"  
[root@emi-demo13 ~]# qmgr -c "set server managers += root@emitestbed36.cnaf.infn.it"  
[root@emi-demo13 ~]# qmgr -c "set server managers += root@emitestbed34.cnaf.infn.it"  
[root@emi-demo13 ~]# qmgr -c "set queue demo resources_default.neednodes = lcgpro"
```

```
FOR SL6
```

```
[root@emi-demo13 ~]#set qwnodessl6 cloudtfls16 resources_default.neednodes = cloudtfls16  
[root@emi-demo13 ~]# qmgr -c "set server managers += root@cert-06.cnaf.infn.it"  
[root@emi-demo13 ~]# qmgr -c "set server managers += root@emitestbed19.cnaf.infn.it"  
[root@emi-demo13 ~]# qmgr -c "set server managers += root@emitestbed20.cnaf.infn.it"
```

```
[root@emi-demo13 ~]# cat /var/torque/server_priv/nodes  
emitestbed23.cnaf.infn.it np=2 lcgpro  
emi-demo09.cnaf.infn.it np=2 lcgpro  
emitestbed34.cnaf.infn.it np=3 cloudtf bait  
emitestbed35.cnaf.infn.it qwnodes  
emitestbed36.cnaf.infn.it qwnodes  
cert-06.cnaf.infn.it np=3 cloudtfls16 bait  
emitestbed19.cnaf.infn.it qwnodessl6  
emitestbed20.cnaf.infn.it qwnodessl6
```

```
[root@emi-demo13 ~]# cat /var/spool/maui/maui.cfg  
# MAUI configuration example  
SERVERHOST          emi-demo13.cnaf.infn.it  
ADMIN1              root  
ADMIN3              edginfo rgma edguser ldap  
ADMINHOSTS          emi-demo13.cnaf.infn.it  
RMCFG[base]         TYPE=PBS  
SERVERPORT          40559  
SERVERMODE          NORMAL
```

```
# Set PBS server polling interval. If you have short # queues or/and jobs it is worth to set a sh
```

```
RMPOLLINTERVAL      00:00:10
```

```
# a max. 10 MByte log file in a logical location
```

```
LOGFILE              /var/log/maui.log  
LOGFILEMAXSIZE       10000000  
LOGLEVEL             1
```

```
# Set the delay to 1 minute before Maui tries to run a job again, # in case it failed to run the  
# The default value is 1 hour.
```

```
DEFERTIME            00:01:00
```

```
# Necessary for MPI grid jobs
```

```
ENABLEMULTIREQJOBS  TRUE  
NODECFG[emitestbed34.cnaf.infn.it] PARTITION=virtual  
CLASSCFG[qwnodes]  PLIST=virtual PDEF=virtual
```

```
[root@emi-demo13 ~]# /etc/init.d/maui restart
```

```
Shutting down MAUI Scheduler: [ OK ]
```

```
Starting MAUI Scheduler: [ OK ]
```

```
[root@emi-demo13 ~]#
```

```
CHECKING NODES ->
```

```
[root@emi-demo13 ~]# pbsnodes -a  
emitestbed23.cnaf.infn.it
```

WnODesLogBook < EMI < TWiki

```
state = free
np = 2
properties = lcgpro
ntype = cluster
status = rectime=1347958286,varattr=,jobs=,state=free,netload=14304694806,gres=,loadave=0.00
gpus = 0

emi-demo09.cnaf.infn.it
state = free
np = 2
properties = lcgpro
ntype = cluster
status = rectime=1347958285,varattr=,jobs=,state=free,netload=20164290477,gres=,loadave=0.04
gpus = 0

emitestbed34.cnaf.infn.it
state = free
np = 3
properties = cloudf,bait
ntype = cluster
status = rectime=1347958296,varattr=,jobs=,state=free,netload=23750277315,gres=,loadave=0.01
gpus = 0

emitestbed35.cnaf.infn.it
state = down,offline
np = 1
properties = qwnodes
ntype = cluster
status = rectime=1347884870,varattr=,jobs=,state=free,netload=392665007,gres=,loadave=1.08,n
gpus = 0

emitestbed36.cnaf.infn.it
state = down,offline
np = 1
properties = qwnodes
ntype = cluster
gpus = 0

cert-06.cnaf.infn.it
state = free
np = 3
properties = cloudfsl6,bait
ntype = cluster
status = rectime=1347958307,varattr=,jobs=,state=free,netload=3443115449,gres=,loadave=0.00,
gpus = 0

emitestbed19.cnaf.infn.it
state = down,offline
np = 1
properties = qwnodessl6
ntype = cluster
status = rectime=1347632818,varattr=,jobs=,state=free,netload=2338705,gres=,loadave=0.00,ncp
gpus = 0

emitestbed20.cnaf.infn.it
state = offline
np = 1
properties = qwnodessl6
ntype = cluster
status = rectime=1347958318,varattr=,jobs=,state=free,netload=408458961,gres=,loadave=0.00,n
gpus = 0

[root@emi-demo13 ~]#
```

Service Testing

+++--- On WN hosting Wnodes server

1. Check daemons:

+++--- On Torque server

1. Enter a pool account user: `$>su - tst01`
2. Submit a test job
 1. `$> qsub -q qwnodes test.sh ->` (where test is a bash executable with commands like `/bin/hostname` inside)
 2. `$> qstat -a`

check that the output has a virtual host -> hostname

1. Submit a grid test job

```
glite-ce-job-submit -d -r emi-demo13.cnaf.infn.it:8443/cream-pbs-qwnodes -a test.jdl
2012-09-18 11:43:49,108 INFO - *****
2012-09-18 11:43:49,109 INFO - CREAM User Interface version 1.2.0 - Starting at Tue Sep 18 11:43:

2012-09-18 11:43:49,109 DEBUG - Using certificate proxy file [/tmp/x509up_u500]
2012-09-18 11:43:49,137 INFO - VO from certificate=[testers.eu-emi.eu]
2012-09-18 11:43:49,139 WARN - No configuration file suitable for loading. Using built-in configu
2012-09-18 11:43:49,140 INFO - Logfile is [/tmp/glite_cream_cli_logs/glite-ce-job-submit_CREAM_do
2012-09-18 11:43:49,141 DEBUG - Processing file [/home/dongiovanni/Test.sh]...
2012-09-18 11:43:49,142 DEBUG - Inserting mangled InputSandbox in JDL: [{"/home/dongiovanni/Test.
2012-09-18 11:43:49,149 INFO - Registering to [http://emi-demo13.cnaf.infn.it:8443/ce-cream/servi
2012-09-18 11:43:49,150 INFO - certUtil::generateUniqueID() - Generated DelegationID: [0cde6d23c1
2012-09-18 11:43:50,706 INFO - JobID=[https://emi-demo13.cnaf.infn.it:8443/CREAM575922133]
2012-09-18 11:43:50,707 INFO - UploadURL=[gsiftp://emi-demo13.cnaf.infn.it/var/cream_sandbox/test
2012-09-18 11:43:50,710 INFO - Sending file [gsiftp://emi-demo13.cnaf.infn.it/var/cream_sandbox/t
2012-09-18 11:43:51,247 INFO - Now invoking JobStart for JobID [https://emi-demo13.cnaf.infn.it:8
https://emi-demo13.cnaf.infn.it:8443/CREAM575922133
[dongiovanni@emitestbed08 ~]$ glite-ce-job-status https://emi-demo13.cnaf.infn.it:8443/CREAM57592

***** JobID=[https://emi-demo13.cnaf.infn.it:8443/CREAM575922133]
Status = [IDLE]

[dongiovanni@emitestbed08 ~]$ glite-ce-job-status https://emi-demo13.cnaf.infn.it:8443/CREAM57592

***** JobID=[https://emi-demo13.cnaf.infn.it:8443/CREAM575922133]
Status = [RUNNING]
```

Notes & Service Troubleshooting

1. IMAGE CONFIGURATION:

1. to use Wnodes through a grid job you need to configure an emiWN in the image. *. as wnlst > just put the hostname
2. problems with virtual machine start can occur when rpm version of wnodes on the image are not up-to-date or configuration files are different from the bait host. Please remember to check this congruency. A better approach would be to have those files shared through a shared file system.
3. notice also that fetch curl operation at vm machine boot can take a lot. -> shared file system is suggested for curl handling too.

-- DaniloDongiovanni - 04-May-2012

This topic: EMI > WnODesLogBook

Topic revision: r8 - 2012-12-06 - FabioCapanniniExternal



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback