LFG Weblogic Workflow

This document is to describe the process for utilizing the weblogic profiles to deploy the weblogic domains. The main goal for the profile::weblogic classes was to enforce some standards on deploying with the weblogic module. Profile classes are meant to pull some complexity out of deploying weblogic as well as give some sane defaults for the LFG's deployment of weblogic. Configuration of the weblogic service can be done several ways using the weblogic component module these profiles utilize the process of calling a class and passing parameters to it, building a hash to pass to a create_resource function in puppet and calling additional classes that provide those previous mentioned tasks to ensure the ordering process.

Key assumptions:
  Domain cluster consists of 1 x DomainAdmin, 2 x Nodemanagers, 2 x OHS hosts.
  One Domain per cluster.
  Weblogic 11g version

To utilize this code it requires a minimum of two facts:
  1) lfg_sysrole = This should come for the hostname of the host:
    wlsmgr = Domain admin
    wls   = Node manager
    ohs   = OHS host
    This facts is used to determine what the wls_role is, you can set this by creating a hiera file for the host and adding 'wls_role: <role name>' in it. This will then use that as the role rather then the lfg_sysrole determined method.
  2) wls_domain = This is the name of the domain to be deployed to the host. This could be dropped on the host at provision time. Create a file on the host '/etc/puppetlabs/facter/facts.d/wls_domain.txt' with 'wls_domain=<name>'. Example: wls_domain=Wls1036

  Once those facts are determined the class profile::weblogic will configure the machine based on the applicable profile, 'profile::weblogic::domainadmin', 'profile::weblogic::nodemanager', and 'profile::weblogic::ohs' to the specifications in the hieradata files. Example of a hieradata domain file is bellow:

```
 ---
node1_address:          "172.22.5.40"
node2_address:          "172.22.5.41"
domain_nodemanager_port:  '5556'
wls_version:            '1036'
adminserver_address:     &adminserver_address  "172.22.5.30"

logoutput:              &logoutput              true
log_dir:                &log_dir                '/opt/log/weblogic'
wls_os_user:            &wls_os_user            "webadmin"
wls_os_group:           &wls_os_group           "webadmns"
wls_weblogic_user:      &wls_weblogic_user      "weblogic"
wls_weblogic_home_dir:    &wls_weblogic_home_dir    "%{hiera('dft_wls_weblogic_home_dir')}"
wls_middleware_home_dir:  &wls_middleware_home_dir  "%{hiera('dft_wls_middleware_home_dir')}"
wls_domains_path:       &wls_domains_path       "%{hiera('dft_wls_domains_path')}"
domain_nodemanager_port:  &domain_nodemanager_port  "%{hiera('dft_domain_nodemanager_port')}"
Wls1036_domain_path:      "%{hiera('wls_domains_path')}/Wls1036"
```

```
domain_name:               &domain_name               "%{::wls_domain}"
domain_adminserver:        &domain_adminserver        "AdminServer"
domain_adminserver_address: &domain_adminserver_address "%{hiera('adminserver_address')}"
domain_wls_password:       &domain_wls_password       "%{hiera('dft_wls_weblogic_password')}"
domain_node1_address:      &domain_node1_address      "%{hiera('node1_address')}"
domain_node2_address:      &domain_node2_address      "%{hiera('node2_address')}"
domain_user_config_file:    "/home/webadmin/wls-Wls1036-WebLogicConfig.properties"
domain_user_key_file:       "/home/webadmin/wls-Wls1036-WebLogicKey.properties"

userconfig_instances:
 “%{hiera('wls_os_user')}" :
    log_output:           *logoutput
    user_config_dir:      "/home/%{hiera('wls_os_user')}"

authentication_provider_instances:
  'DefaultAuthenticator':
    ensure:             'present'
    control_flag:       'SUFFICIENT'
    order:              '0'

identity_asserter_instances:
  'DefaultIdentityAsserter':
    ensure:             'present'
    providerclassname:  'weblogic.security.providers.authentication.DefaultIdentityAsserter'
    activetypes:        'AuthenticatedUser'
    defaultmappertype:  'E'
    before:             Wls_domain[Wls1036]

wls_domain_instances:
  'Wls1036':
    ensure:                     'present'
    jpa_default_provider:       'org.eclipse.persistence.jpa.PersistenceProvider'
    jta_max_transactions:       '20000'
    jta_transaction_timeout:    '35'
    log_file_min_size:          '5000'
    log_filecount:              '10'
    log_filename:               "%{hiera('log_dir')}/Wls1036.log"
    log_number_of_files_limited: '1'
    log_rotate_logon_startup:   '1'
    log_rotationtype:           'bySize'
    security_crossdomain:       '0'

wls_adminserver_instances_domain:
  'AdminServer_Wls1036':
    ensure:                   'running'
    server_name:              *domain_adminserver
    domain_name:              *domain_name
    domain_path:              "%{hiera('Wls1036_domain_path')}"
    os_user:                  *wls_os_user
    weblogic_home_dir:        *wls_weblogic_home_dir
    weblogic_user:            *wls_weblogic_user
    weblogic_password:        *domain_wls_password
    jdk_home_dir:             '/usr/java/latest'
    nodemanager_address:      *domain_adminserver_address
    nodemanager_port:          *domain_nodemanager_port
    jsse_enabled:              true
    refreshonly:               true
    subscribe:                 Wls_domain[Wls1036]

user_instances:
  'testuser1':
    ensure:                 'present'
    password:               'weblogic1'
    authenticationprovider: 'DefaultAuthenticator'
    realm:                  'myrealm'
    description:            'my test user'

group_instances:
  'TestGroup':
    ensure:                 'present'
    authenticationprovider: 'DefaultAuthenticator'
    description:            'My TestGroup'
    realm:                  'myrealm'
    users:
      - 'testuser1'

machines_instances:
  'LocalMachine':
    ensure:         'present'
    listenaddress:  *domain_adminserver_address
    listenport:     '5556'
    machinetype:    'UnixMachine'
    nmtype:         'SSL'
  'nc2pxwls100':
    ensure:         'present'
    listenaddress:  *domain_node1_address
    listenport:     '5556'
    machinetype:    'UnixMachine'
    nmtype:         'SSL'
  'nc2pxwls101':
    ensure:         'present'
    listenaddress:  *domain_node2_address
    listenport:     '5556'
    machinetype:    'UnixMachine'
    nmtype:         'SSL'

server_vm_args_permsize:      &server_vm_args_permsize     '-XX:PermSize=256m'
server_vm_args_max_permsize:  &server_vm_args_max_permsize '-XX:MaxPermSize=256m'
server_vm_args_memory:        &server_vm_args_memory       '-Xms752m'
server_vm_args_max_memory:    &server_vm_args_max_memory   '-Xmx752m'

wls_java_arguments:
  - *server_vm_args_permsize
  - *server_vm_args_max_permsize
  - *server_vm_args_memory
  - *server_vm_args_max_memory
  - "-Dweblogic.Stdout=%{hiera('log_dir')}/${title}.out"
  - "-Dweblogic.Stderr=%{hiera('log_dir')}/${title}_err.out"
  - '-Dweblogic.security.SSL.enforceConstraints=off'
  - '-Dssl.debug=true'
  - '-Dweblogic.security.SSL.ignoreHostnameVerification=true'
  - '-Dweblogic.security.TrustKeyStore=DemoTrust'

server_instances:
  'AdminServer':
    ensure:                    'present'
    arguments:                 “%{hiera('wls_java_arguments')}”
    listenaddress:             *domain_adminserver_address
    listenport:                '7001'
    machine:                    'LocalMachine'
    logfilename:                "%{hiera('log_dir')}/AdminServer.log"
    log_datasource_filename:          "%{hiera('log_dir')}/AdminServer_datasource.log"
    log_file_min_size:            '2000'
    log_filecount:              '10'
    log_number_of_files_limited:       '1'
    log_rotate_logon_startup:        '1'
    log_rotationtype:             'bySize'
    log_http_filename:             "%{hiera('log_dir')}/AdminServer_access.log"
    log_http_format_type:           'extended'
    log_http_format:              'date time x-XForwardedFor s-ip cs-method cs-uri x-SOAPAction sc-status
bytes time-taken x-UserAgent'
    tunnelingenabled:              '0'
    max_message_size:               '10000000'
    sslenabled:                 '0'
    ssllistenport:              '7002'
    sslhostnameverificationignored:     '1'
    two_way_ssl:                 '0'
    client_certificate_enforced:     '0'
    jsseenabled:                '1'
  'wlsServer1':
    ensure:                    'present'
     arguments:                 “%{hiera('wls_java_arguments')}”
    listenaddress:              *domain_node1_address
    listenport:                '8001'
    logfilename:                "%{hiera('log_dir')}/wlsServer1.log"
    log_datasource_filename:          "%{hiera('log_dir')}/wlsServer1_datasource.log"
    log_http_filename:             "%{hiera('log_dir')}/wlsServer1_access.log"
    log_file_min_size:            '2000'
    log_filecount:              '10'
    log_number_of_files_limited:       '1'
    log_rotate_logon_startup:        '1'
    log_rotationtype:             'bySize'
    machine:                    'nc2pxwls100'
    sslenabled:                 '1'
    ssllistenport:              '8201'
    sslhostnameverificationignored:     '1'
    two_way_ssl:                 '0'
    client_certificate_enforced:     '0'
    jsseenabled:                '1'
    max_message_size:               '25000000'
    notify:                   Wls_adminserver[AdminServer_Wls1036]
  'wlsServer2':
    ensure:                    'present'
    arguments:                 “%{hiera('wls_java_arguments')}”
    listenport:                '8001'
    logfilename:                "%{hiera('log_dir')}/wlsServer2.log"
    log_datasource_filename:         "%{hiera('log_dir')}/wlsServer2_datasource.log"
    log_http_filename:            "%{hiera('log_dir')}/wlsServer2_access.log"
    log_file_min_size:           '2000'
    log_filecount:               '10'
    log_number_of_files_limited:      '1'
    log_rotate_logon_startup:        '1'
    log_rotationtype:            'bySize'
    machine:                    'nc2pxwls101'
    sslenabled:                  '1'
    ssllistenport:              '8201'
    sslhostnameverificationignored:   '1'
    listenaddress:               *domain_node2_address
    jsseenabled:                 '1'
    max_message_size:              '25000000'
    notify:                     Wls_adminserver[AdminServer_Wls1036]

wls_adminserver_instances_server:
  'AdminServer_Wls1036_2':
    ensure:               'running'
    server_name:             *domain_adminserver
    domain_name:               *domain_name
    domain_path:              "%{hiera('Wls1036_domain_path')}"
    os_user:               *wls_os_user
    weblogic_home_dir:          *wls_weblogic_home_dir
    weblogic_user:            *wls_weblogic_user
    weblogic_password:          *domain_wls_password
    jdk_home_dir:             '/usr/java/latest'
    nodemanager_address:          *domain_adminserver_address
    nodemanager_port:           *domain_nodemanager_port
    jsse_enabled:            true
    refreshonly:            true
    subscribe:             Wls_server[AdminServer]


server_channel_instances:
  'wlsServer1:Channel-Cluster':
    ensure:          'present'
    enabled:          '1'
    httpenabled:       '1'
    listenaddress:     *domain_node1_address
    listenport:       '8003'
    outboundenabled:  '0'
    protocol:         'cluster-broadcast'
    publicaddress:     *domain_node1_address
    tunnelingenabled: '0'
  'wlsServer2:Channel-Cluster':
    ensure:          'present'
    enabled:          '1'
    httpenabled:       '1'
    listenaddress:     *domain_node2_address
    listenport:       '8003'
    outboundenabled:  '0'
    protocol:         'cluster-broadcast'
    publicaddress:     *domain_node2_address
    tunnelingenabled: '0'
  'wlsServer1:HTTP':
    ensure:           'present'
    enabled:          '1'
    httpenabled:      '1'
    listenport:       '8004'
    publicport:       '8104'
    outboundenabled:  '0'
    protocol:         'http'
    tunnelingenabled: '0'
    max_message_size: '35000000'
  'wlsServer2:HTTP':
    ensure:           'present'
    enabled:          '1'
    httpenabled:      '1'
    listenport:       '8004'
    publicport:       '8104'
    outboundenabled:  '0'
    protocol:         'http'
    tunnelingenabled: '0'
    max_message_size: '35000000'

cluster_instances:
  'WebCluster':
    ensure:                    'present'
    migrationbasis:            'database'
    servers:
      - 'wlsServer1'
      - 'wlsServer2'
    messagingmode:             'unicast'
    unicastbroadcastchannel:   'Channel-Cluster'
    frontendhost:              *domain_adminserver_address
    frontendhttpport:          '2001'
    frontendhttpsport:         '2002'

datasource_instances:
    'hrDS':
      ensure:                     'present'
      drivername:                 'oracle.jdbc.xa.client.OracleXADataSource'
      extraproperties:
        - 'SendStreamAsBlob=true'
        - 'oracle.net.CONNECT_TIMEOUT=10001'
      globaltransactionsprotocol:  'TwoPhaseCommit'
      initialcapacity:            '2'
      jndinames:
        - 'jdbc/hrDS'
        - 'jdbc/hrDS2'
      maxcapacity:                '15'
      target:
        - 'wlsServer1'
        - 'wlsServer2'
      targettype:
        - 'Server'
        - 'Server'
      testtablename:              'SQL SELECT 1 FROM DUAL'
      url:                        "jdbc:oracle:thin:@dbagent2.alfa.local:1521/test.oracle.com"
      user:                       'hr'
      password:                   'hr'
      usexa:                    '1'
     'jmsDS':
      ensure:                   'present'
      drivername:               'com.mysql.jdbc.Driver'
      globaltransactionsprotocol:  'None'
      initialcapacity:          '0'
      jndinames:
        - 'jmsDS'
      maxcapacity:              '15'
      target:
        - 'wlsServer1'
        - 'wlsServer2'
      targettype:
        - 'Server'
        - 'Server'
      testtablename:            'SQL SELECT 1'
      url:                      'jdbc:mysql://10.10.10.10:3306/jms'
      user:                     'jms'
      password:                 'jms'
      usexa:                    '1'

  file_persistence_folders:
    "%{hiera('Wls1036_domain_path')}/persistence1":
      ensure:     directory
      recurse:    false
      replace:    false
      mode:       '0775'
      owner:      *wls_os_user
      group:      *wls_os_group
    "%{hiera('Wls1036_domain_path')}/persistence2":
      ensure:     directory
      recurse:    false
      replace:    false
      mode:       '0775'
      owner:      *wls_os_user
      group:      *wls_os_group
    "%{hiera('Wls1036_domain_path')}/persistence3":
      ensure:     directory
      recurse:    false
      replace:    false
      mode:       '0775'
      owner:      *wls_os_user
      group:      *wls_os_group
```

In the Puppet Enterprise console under Classification Tab a group should be created for each of the environments required. The group should have the class 'profile::weblogic' or a group for each weblogic host type can be created and assigned the applicable class.

The 'profile::weblogic::<type>' use a params class located at 'profile::weblogic::params' to build out a parameters list that can be utilized across different classes and resources. This profile module tries to utilize the same parameters as listed and used in the abrader-weblogic/biemond-orawls module documentation. LFG is using the abrader-weblogic module that was based on the biemond-orawls module. This module and code has been checked into the LFG git server.

Bellow is an example of the params.pp file in the weblogic profile.

```
class profile::weblogic::params {

  $wls_domain                = "${::wls_domain}"
  $param_lfg_sysrole           = "${::lfg_sysrole}"
  $jdk_home_dir              = hiera('profile::weblogic::jdk_home_dir', '/usr/java/latest')
  $download_dir              = hiera('download_dir', '/tmp/oracle')
  $fmw_plugins               = hiera_hash('profile::weblogic::base::addons', {})
  $opatch                 = hiera_hash('profile::weblogic::base::opatch', {})
  $wls_version               = hiera('wls_version', 1036)
  $wls_os_user               = hiera('wls_os_user', 'webadmin')
  $wls_os_group              = hiera('wls_os_group', 'webadmns')
  $wls_weblogic_user           = 'weblogic'
  $wls_weblogic_password         = 'weblogic1'
  $wls_oracle_base_home_dir       = '/opt/was/oracle'
  $wls_middleware_home_dir        = hiera('wls_middleware_home_dir', "$
{wls_oracle_base_home_dir}/middleware/${wls_version}")
  $wls_weblogic_home_dir         = hiera('wls_weblogic_home_dir', "$
{wls_middleware_home_dir}/wlserver")
  $wls_domains_path            = hiera('domains_dir', "$
{wls_middleware_home_dir}/user_projects/domains")
  $apps_dir               = hiera('wls_apps_dir', "$
{wls_middleware_home_dir}/user_projects/applications")
  $wls_log_output            = hiera('log_outpu', true)
  $wls_log_dir               = '/opt/log/weblogic'
  $adminserver_name            = hiera('adminserver_name', 'AdminServer')
  $development_mode             = hiera('wls_development_mode', true)
  $adminserver_address          = hiera('adminserver_address')
  $adminserver_port            = hiera('adminserver_port', '7001')
  $java_arguments              = hiera('wls_java_arguments', {})
  $jsse_enabled              = hiera('wls_jsse_enabled', true)
  $nodemanager_address           = $::ipaddress_eth1
  $nodemanager_port            = hiera('nodemanager_port', '5556')
  $weblogic_user             = hiera('wls_weblogic_user', 'weblogic')
  $webtier_enabled            = 'true'
  $custom_trust              = hiera('wls_custom_trust', false)
  $trust_keystore_file         = hiera('wls_trust_keystore_file'      , undef)
  $trust_keystore_passphrase       = hiera('wls_trust_keystore_passphrase' , undef)
  $custom_identity             = hiera('custom_identity', false)
  $custom_identity_keystore_filename    = "${source}/identity_${::hostname}.jks"
  $custom_identity_keystore_passphrase   = hiera('custom_identity_keystore_passphrase', 'welcome')
  $custom_identity_alias          = "${::hostname}"
  $custom_identity_privatekey_passphrase = hiera('custom_identity_privatekey_passphrase', 'welcome')
  $create_rcu               = hiera('create_rcu', false)
  $user_config_file            = hiera('domain_user_config_file', "/home/webadmin/wls-Wls1036-
WebLogicConfig.properties")
  $user_key_file             = hiera('domain_user_key_file', "/home/webadmin/wls-Wls1036-
WebLogicKey.properties")

  case $::osfamily {
    'RedHat': {
      $wls_filename       = hiera('wls_filename', 'wls1036_generic.jar')
      $source          = hiera('source', '/vagrant/oracle')
    }
    'AIX': {
```

```
   $wls_filename          = hiera('wls_filename', 'wls1036_generic.jar')
   $source                = hiera('source', '/vagrant/oracle')
  }
  default: {
   fail "Operating System family ${::osfamily} not supported"
  }
 }
 case $param_lfg_sysrole {
  'wlsmgr': {
   $wls_role = hiera('wls_role', 'domainadmin')
  }
  'wls': {
   $wls_role = hiera('wls_role', 'nodemanager')
  }
  'ohs': {
   $wls_role = hiera('wls_role', 'ohs')
  }
  default: {
   fail "lfg_sysrole is not set correctly, possible values are [wlsmgr, wls, ohs], $::lfg_sysrole"
  }
 }
}
```

The process of deploying a new weblogic domain in the LFG infrastructure would be as follows:

1. Create a domain yaml file for hiera. This will spell out the extra feature enabled and configurations of those feature in hash lists. To enable features not in the standard demo file please use the following documentation https://github.com/biemond/biemond-orawls and listed in the weblogic module resources. To get a list of weblogic resources you can run 'puppet resource –types | grep wls'. To explore any resource parameters you can run 'puppet describe <resource_name>'.

2. Deploy the hosts with LFG identified hostnaming convention.

3. Deploy and or verify the 'wls_domain' fact is correct and present.

4. Allow puppet to run on hosts 4-5 times, or run puppet on each host 4-5 times. This will need happen in a domainadmin, nodemanager, ohs host order.

5. Verify the hosts have been configured correctly.

Each of these steps are required to be completed in this order. This code will not function without the wls_role being identified, which is done with the fact lfg_sysrole, or wls_role in hiera, as well as wls_domain being set.