1. Puppet Setup.

1.1 Setup puppet server

#yum install -y http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm
#yum install -y puppet-server

#mkdir -p /etc/puppet/environments/production/{modules,manifests}
#cd /etc/puppet/environments/production
#echo "modulepath = /etc/puppet/environments/production/modules" >
/etc/puppet/environments/production/environment.conf
echo "environment_timeout = 5s" >> /etc/puppet/environments/production/environment.conf

Open puppet.conf

[main]
+ certname = puppetmaster

+[master]
+ environmentpath = $confdir/environments
+ basemodulepath = $confdir/modules:/opt/puppet/share/modules

Stop Firewall
/etc/init.d/iptables stop

Start puppet server
service puppetmaster start

1.2 Setup puppet agent for wiki(centos) and wikitest(ubuntu)

Login to wiki to install puppet
#yum install -y http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm
#yum install -y puppet

Update puppetmaster info in puppet.conf

vi /etc/puppet/puppet.conf

[main]
+ server = puppetmaster

Run puppet command to request cert
puppet agent --verbose --no-daemonize –onetime

Login to wikitest to install puppet

wget https://apt.puppetlabs.com/puppetlabs-release-trusty.deb
sudo dpkg -i puppetlabs-release-trusty.deb
sudo apt-get update

sudo apt-get install puppet

vi /etc/puppet/puppet.conf

+[agent]
+ server = puppetmaster

puppet agent –enable
puppet agent --verbose --no-daemonize –onetime

Login to puppet server to sign node certs
puppet cert –list
puppet cert sign wiki
puppet cert sign wikitest

1.3 Testing node definations.

Login to puppetserver

vi /etc/puppet/environments/production/manifests/nodes.pp

```
node 'wiki'{
  file { '/info.txt':
    ensure => 'present',
    content => inline_template("This file was created by puppet at <%= Time.now %>\n"),
  }
}
```

```
node 'wikitest' {
  file { '/info.txt':
    ensure => 'present',
    content => inline_template("This file was created by puppet at <%= Time.now %>\n"),
  }
}
```

Login to both puppet agent, wiki & wikitest

puppet agent --verbose --no-daemonize –onetime
cat /info.txt

2. Manifests.

2.1 Managing Files

Login to puppet server
/etc/puppet/environments/production/manifests/nodes.pp

```
node 'wiki' {

  file { '/info.txt':
    ensure  => 'present',
    content => inline_template("Created by Puppet at <%= Time.now %>\n"),
  }
}

node 'wikitest' {

  file { '/info.txt':
    ensure  => 'present',
    content => inline_template("Created by Puppet at <%= Time.now %>\n"),
  }

}
```

Go to both the nodes and execute to apply changes
puppet agent --verbose --no-daemonize --onetime

2.2 Managing Packages & Services

```
node 'wiki' {

  file { '/info.txt':
    ensure  => 'present',
    content => inline_template("Created by Puppet at <%= Time.now %>\n"),
  }

  package { 'ntp':
    ensure => 'installed',
  }

  service { ntpd:
    ensure => 'running',
    enable => true,
  }

}

node 'wikitest' {

  file { '/info.txt':
    ensure  => 'present',
    content => inline_template("Created by Puppet at <%= Time.now %>\n"),
  }
```

```
  package { 'ntp':
    ensure => 'installed',
  }

  service { ntp:
    ensure => 'running',
    enable => true,
  }

}
```

2.3 Selectors

```
$ntpservice = $osfamily ? {
    'redhat' => 'ntpd',
    'debian' => 'ntp',
    default  => 'ntp',
  }

node 'wiki' {

  file { '/info.txt':
    ensure  => 'present',
    content => inline_template("Created by Puppet at <%= Time.now %>\n"),
  }

  package { 'ntp':
    ensure => 'installed',
  }

  service { $ntpservice:
    ensure => 'running',
    enable => true,
  }


}
node 'wikitest' {

  file { '/info.txt':
    ensure  => 'present',
    content => inline_template("Created by Puppet at <%= Time.now %>\n"),
  }

  package { 'ntp':
    ensure => 'installed',
  }

  service { $ntpservice:
    ensure => 'running',
    enable => true,
```

```
 }
}
```

2.4 Using classes

```
node 'wiki' {

  class { 'linux': }

}

node 'wikitest' {

  class { 'linux': }

}

class linux {

  $ntpservice = $osfamily ? {
    'redhat' => 'ntpd',
    'debian' => 'ntp',
    default  => 'ntp',
  }

  file { '/info.txt':
    ensure  => 'present',
    content => inline_template("Created by Puppet at <%= Time.now %>\n"),
  }

  package { 'ntp':
    ensure => 'installed',
  }

  service { $ntpservice:
    ensure => 'running',
    enable => true,
  }

}
```

2.5 Variables

```
node 'wiki' {

 class { 'linux': }

}

node 'wikitest' {

 class { 'linux': }

}

class linux {

 $admintools = ['git', 'nano', 'screen']

 package { $admintools:
  ensure => 'installed',
 }

 $ntpservice = $osfamily ? {
  'redhat' => 'ntpd',
  'debian' => 'ntp',
  default  => 'ntp',
 }

 file { '/info.txt':
  ensure  => 'present',
  content => inline_template("Created by Puppet at <%= Time.now %>\n"),
 }

 package { 'ntp':
  ensure => 'installed',
 }

 service { $ntpservice:
  ensure => 'running',
  enable => true,
 }

}
```

3. Modules

3.1 Creating modules

```
# cd /etc/puppet/environments/production/modules
# puppet module generate imran-mediawiki --environment production
# mv imran-mediawiki/ mediawiki
# cd mediawiki
# ls -ltr
# cd manifests
# vi init.pp
```

3.2 Customizing Modules

Login to puppetmaster

```
# cd /etc/puppet/environments/production/modules/mediawiki/manifests
# vi init.pp
```

```
class mediawiki {

  $phpmysql = $osfamily ? {
    'redhat' => 'php-mysql',
    'debian' => 'php5-mysql',
    default  => 'php-mysql',
  }

  package { $phpmysql:
    ensure => 'present',
  }
}
```

Open nodes.pp file and invoke mediawiki class

```
# cd /etc/puppet/environments/production/manifests
# vi nodes.pp

node 'wiki'{

  class {'linux': }
  class {'mediawiki':}

}

node 'wikitest' {

  class {'linux': }
  class {'mediawiki':}

}
.....Rest of the file.....
```

Run puppet agent from both the nodes to verify

```
# puppet agent --verbose --no-daemonize –onetime
```

3.3 Conditionals

Login to puppetmaster

cd /etc/puppet/environments/production/modules/mediawiki/manifests

vi init.pp

```
class mediawiki {

  $phpmysql = $osfamily ? {
    'redhat' => 'php-mysql',
    'debian' => 'php5-mysql',
    default  => 'php-mysql',
  }

  package { $phpmysql:
    ensure => 'present',
  }

  if $osfamily == 'redhat' {
    package {'php-xml':
      ensure => 'present',
    }
  }
}
```

Run puppet agent from both the nodes to verify

puppet agent --verbose --no-daemonize –onetime

4. Puppet Forge Modules.

4.1 Downloading from puppet forge
https://forge.puppet.com/

4.2 Apache module
Login to puppetmaster

```
# cd /etc/puppet/environments/production/modules
# puppet module install puppetlabs-apache --modulepath
/etc/puppet/environments/production/modules/

# ls -ltr
# cd mediawiki/manifests/
# vi init.pp

 class mediawiki {

  $phpmysql = $osfamily ? {
    'redhat' => 'php-mysql',
    'debian' => 'php5-mysql',
    default  => 'php-mysql',
  }

  package { $phpmysql:
    ensure => 'present',
  }

  if $osfamily == 'redhat' {
    package {'php-xml':
      ensure => 'present',
    }
  }


  class { '::apache':
    docroot    => '/var/www/html',
    mpm_module => 'prefork',
    subscribe  => Package[$phpmysql],
  }

  class {'::apache::mod::php': }
}
```

Run puppet agent from both the nodes to verify

```
# puppet agent --verbose --no-daemonize –onetime
# service httpd status (centos)
# service apache2 status (ubuntu)
```

4.3 VCSREPO MODULE

Login to puppetmaster

```
# cd /etc/puppet/environments/production/modules
# puppet module install puppetlabs-vcsrepo --modulepath
/etc/puppet/environments/production/modules/

# ls -ltr
# cd mediawiki/manifests/
# vi init.pp

class mediawiki {

 $phpmysql = $osfamily ? {
  'redhat' => 'php-mysql',
  'debian' => 'php5-mysql',
   default  => 'php-mysql',
 }

 package { $phpmysql:
  ensure => 'present',
 }

 if $osfamily == 'redhat' {
  package {'php-xml':
    ensure => 'present',
  }
 }


 class { '::apache':
  docroot    => '/var/www/html',
  mpm_module => 'prefork',
  subscribe  => Package[$phpmysql],
 }

 class {'::apache::mod::php': }

 vcsrepo { '/var/www/html':
  ensure    => 'present',
  provider  => 'git',
  source    => "https://github.com/wikimedia/mediawiki.git"
  revision  => 'REL1_23'
 }
}
```

Run puppet agent from both the nodes to verify

```
# puppet agent --verbose --no-daemonize –onetime
```

4.4 Resource Ordering

```
# vi init.pp
class mediawiki {

 $phpmysql = $osfamily ? {
   'redhat' => 'php-mysql',
   'debian' => 'php5-mysql',
   default  => 'php-mysql',
 }

 package { $phpmysql:
   ensure => 'present',
 }

 if $osfamily == 'redhat' {
   package {'php-xml':
     ensure => 'present',
   }
 }


 class { '::apache':
   docroot    => '/var/www/html',
   mpm_module => 'prefork',
   subscribe  => Package[$phpmysql],
 }

 class {'::apache::mod::php': }

 vcsrepo { '/var/www/html':
   ensure    => 'present',
   provider  => 'git',
   source    => "https://github.com/wikimedia/mediawiki.git",
   revision  => 'REL1_23',
 }

 file {'/var/www/html/index.html':
   ensure => 'absent',
 }

}
```

Run puppet agent from both the nodes to verify

```
# puppet agent --verbose --no-daemonize –onetime
```

This puppet run is gonna remove the index.html file after trying vcsrepo so ultimately it will fail. For this will use resource ordering.

Login to puppetmaster

```
# cd /etc/puppet/environments/production/modules//mediawiki/manifests
# vi init.pp
```

```
class mediawiki {

 $phpmysql = $osfamily ? {
  'redhat' => 'php-mysql',
  'debian' => 'php5-mysql',
  default  => 'php-mysql',
 }

 package { $phpmysql:
  ensure => 'present',
 }

 if $osfamily == 'redhat' {
  package {'php-xml':
   ensure => 'present',
  }
 }


 class { '::apache':
  docroot    => '/var/www/html',
  mpm_module => 'prefork',
  subscribe  => Package[$phpmysql],
 }

 class {'::apache::mod::php': }

 vcsrepo { '/var/www/html':
  ensure    => 'present',
  provider  => 'git',
  source    => "https://github.com/wikimedia/mediawiki.git",
  revision  => 'REL1_23',
 }

 file {'/var/www/html/index.html':
  ensure => 'absent',
 }

 File['/var/www/html/index.html'] -> Vcsrepo ['/var/www/html']

}
```

Run puppet agent from both the nodes to verify

```
# touch /var/www/html/index.html
# puppet agent --verbose --no-daemonize –onetime
```

4.5 MYSQL MODULE

Login to puppetmaster

```
# cd /etc/puppet/environments/production/modules
# puppet module install puppetlabs-mysql  --version 3.9.0 --modulepath
/etc/puppet/environments/production/modules/

# ls -ltr
# cd mediawiki/manifests/
# vi init.pp
class mediawiki {

$phpmysql = $osfamily ? {
  'redhat' => 'php-mysql',
  'debian' => 'php5-mysql',
  default  => 'php-mysql',
}

 package { $phpmysql:
   ensure => 'present',
 }

if $osfamily == 'redhat' {
  package {'php-xml':
    ensure => 'present',
  }
}

 class { '::apache':
   docroot    => '/var/www/html',
   mpm_module => 'prefork',
   subscribe  => Package[$phpmysql],
 }

 class {'::apache::mod::php': }

 vcsrepo { '/var/www/html':
   ensure    => 'present',
   provider  => 'git',
   source    => "https://github.com/wikimedia/mediawiki.git",
   revision  => 'REL1_23',
 }

 file {'/var/www/html/index.html':
   ensure => 'absent',
 }

 File['/var/www/html/index.html'] -> Vcsrepo ['/var/www/html']

 class {'::mysql::server':
   root_password => 'training',
```

```
 }
}
```

Run puppet agent from both the nodes to verify

```
# puppet agent --verbose --no-daemonize –onetime
```

4.6 FIREWALL MODULE

Login to puppetmaster

```
# cd /etc/puppet/environments/production/modules
# puppet module install puppetlabs-firewall --modulepath
/etc/puppet/environments/production/modules/

# ls -ltr
# cd mediawiki/manifests/
# vi init.pp
class mediawiki {

 $phpmysql = $osfamily ? {
  'redhat' => 'php-mysql',
  'debian' => 'php5-mysql',
  default  => 'php-mysql',
 }

 package { $phpmysql:
  ensure => 'present',
 }

 if $osfamily == 'redhat' {
  package {'php-xml':
   ensure => 'present',
  }
 }


 class { '::apache':
  docroot    => '/var/www/html',
  mpm_module => 'prefork',
  subscribe  => Package[$phpmysql],
 }

 class {'::apache::mod::php': }

 vcsrepo { '/var/www/html':
  ensure    => 'present',
  provider  => 'git',
  source    => "https://github.com/wikimedia/mediawiki.git",
  revision  => 'REL1_23',
 }
```

```
file {'/var/www/html/index.html':
  ensure => 'absent',
}

File['/var/www/html/index.html'] -> Vcsrepo ['/var/www/html']

class {'::mysql::server':
  root_password => 'training',
}

class { 'firewall': }

firewall { '000 allow http access':
  port   => '80',
  proto  => 'tcp',
  action => 'accept',

}

}
```

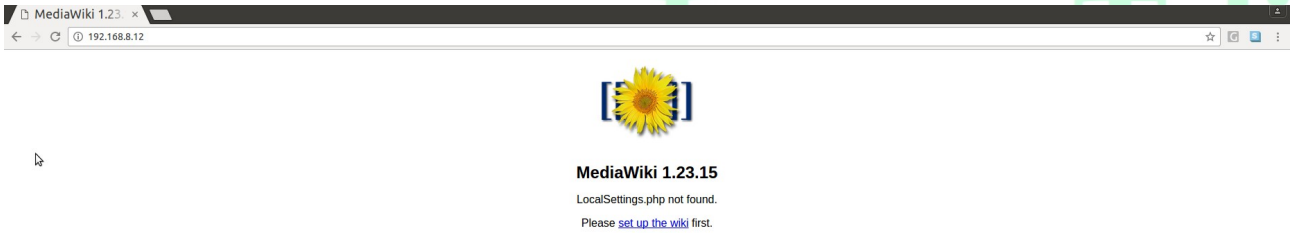
Run puppet agent from both the nodes to verify

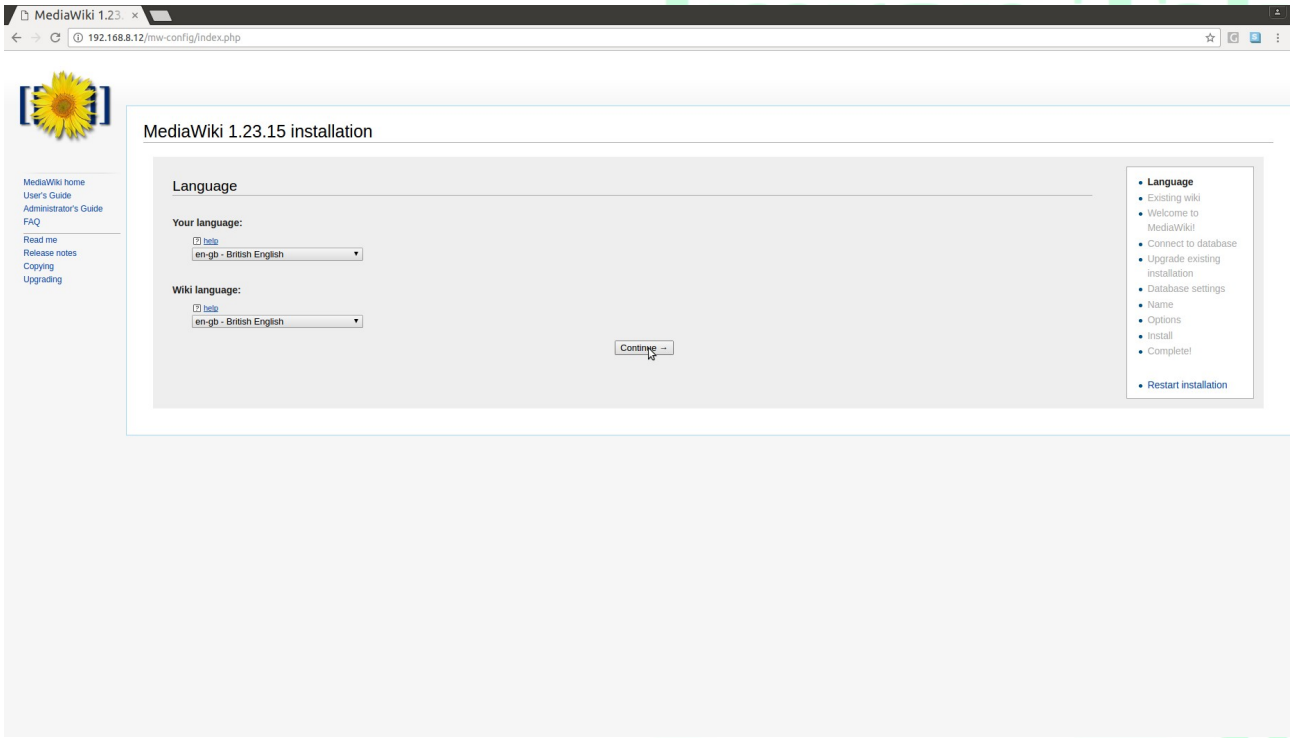
```
# puppet agent --verbose --no-daemonize –onetime
```

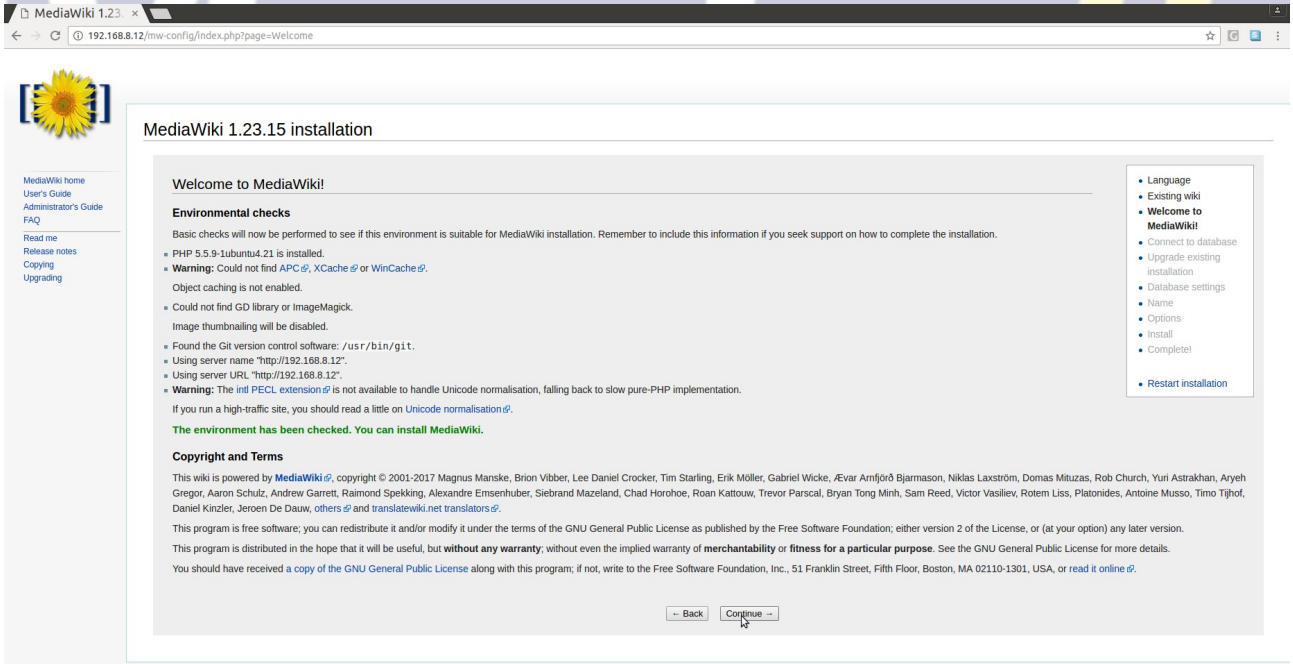
5. Setup MediaWiki Site.

Now since our mediawiki server is deployed to wiki and wikitest, lets go ahead check that out. Open up a browser and enter wiki server IP, you should see the mediawiki webpage as shown below.

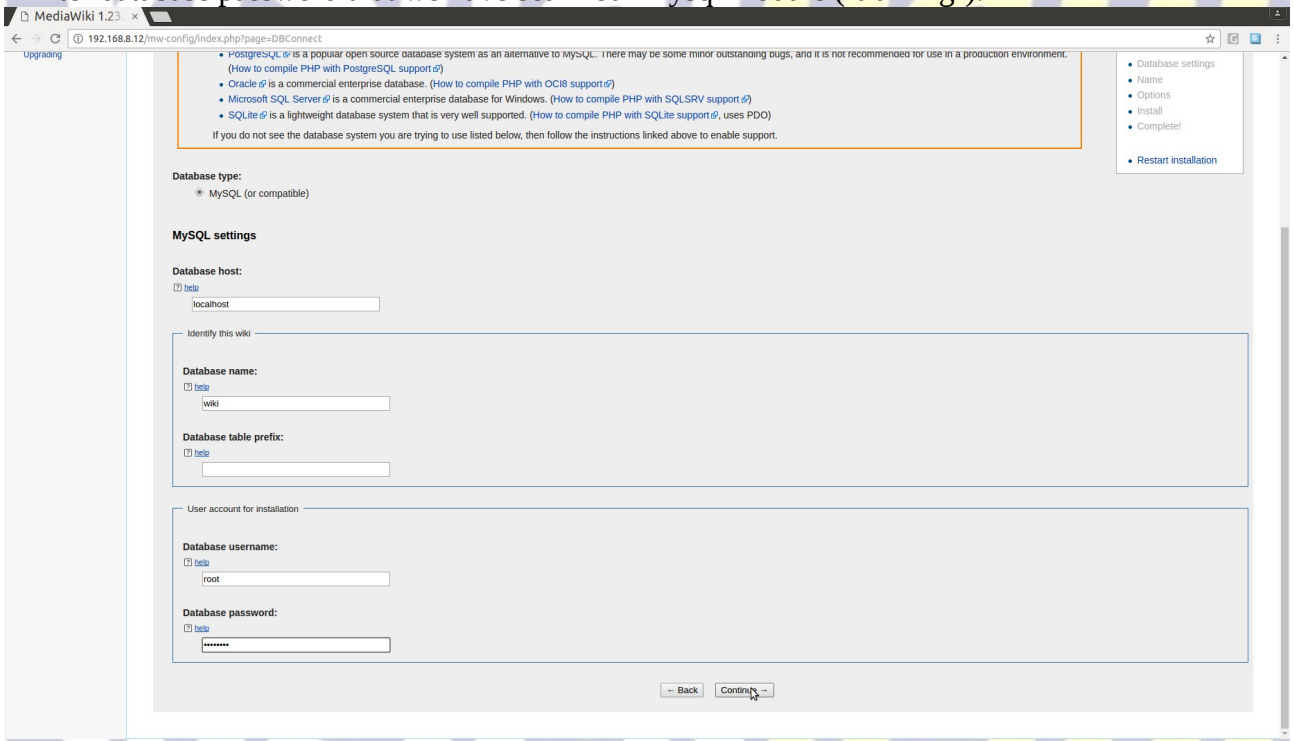
Follow the screenshots to setup mediawiki site.

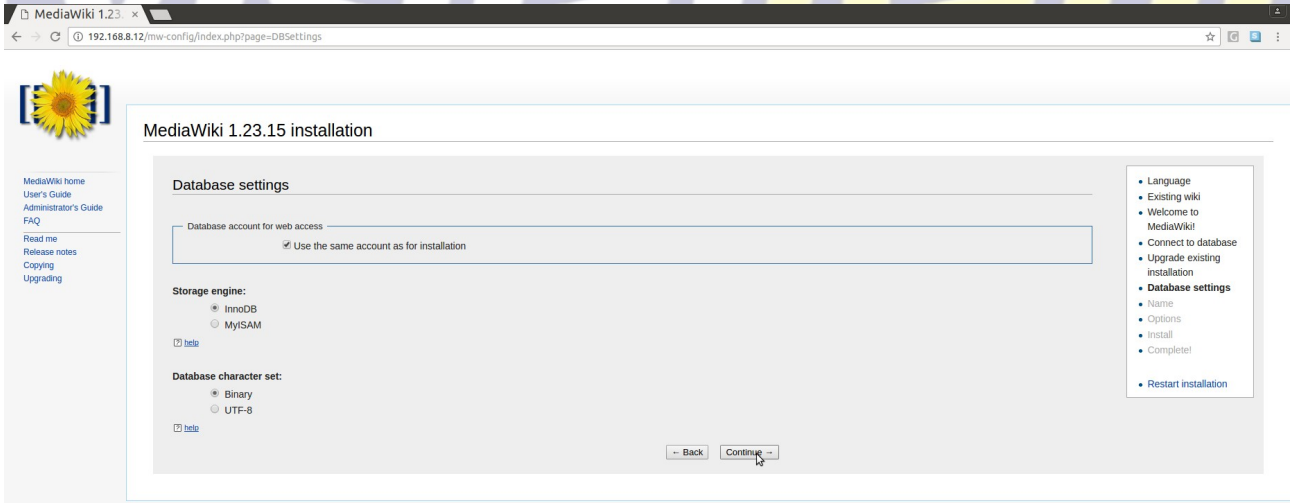




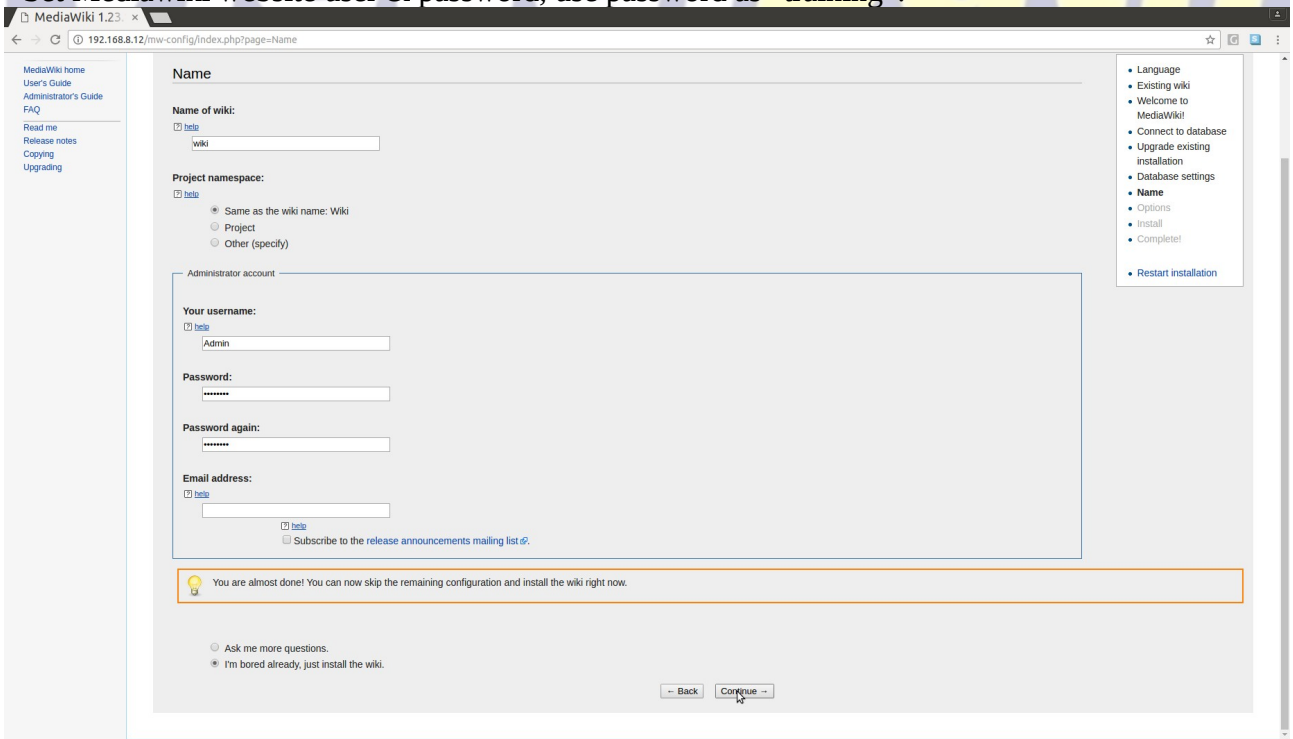


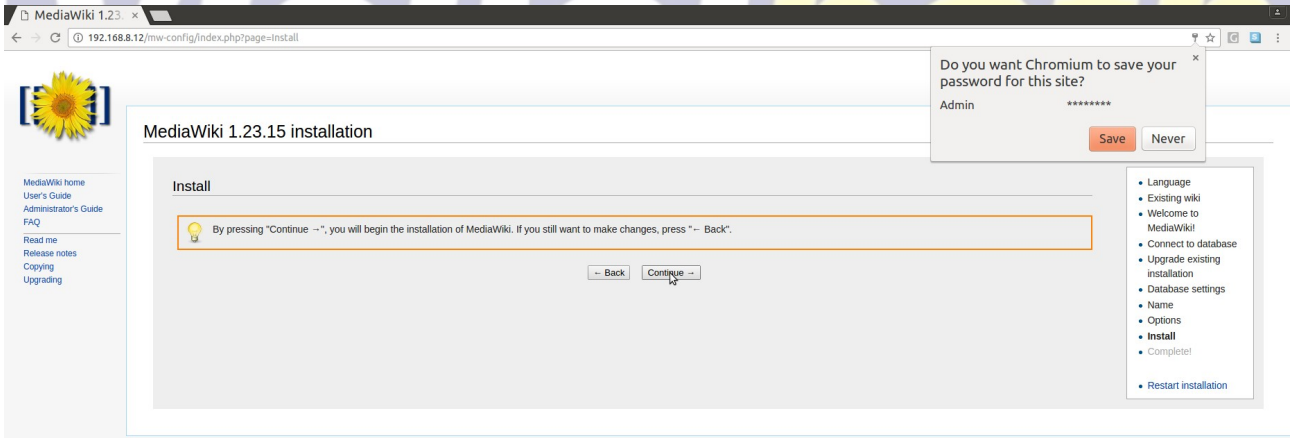
- Enter database password that we have set in our mysql module (‘training’).

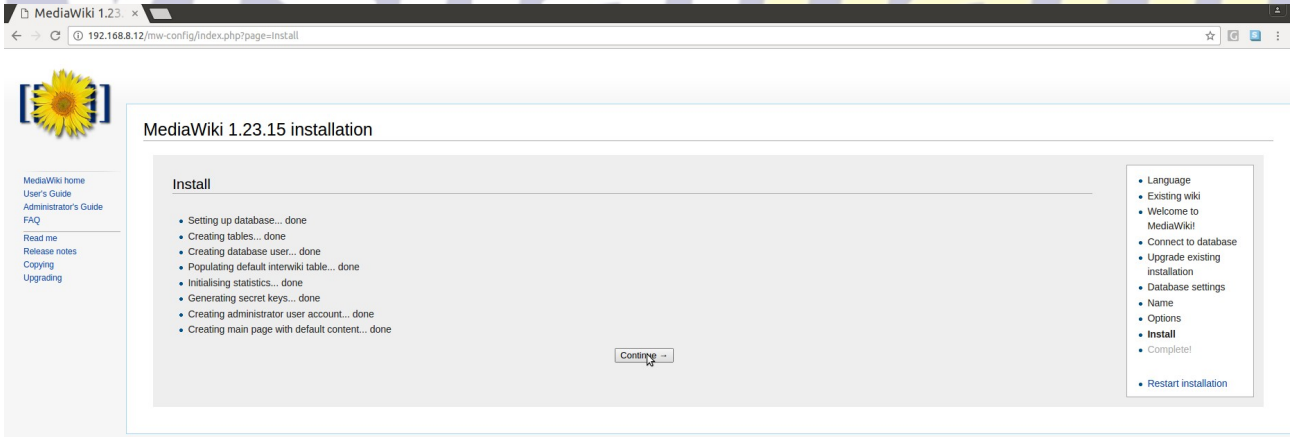




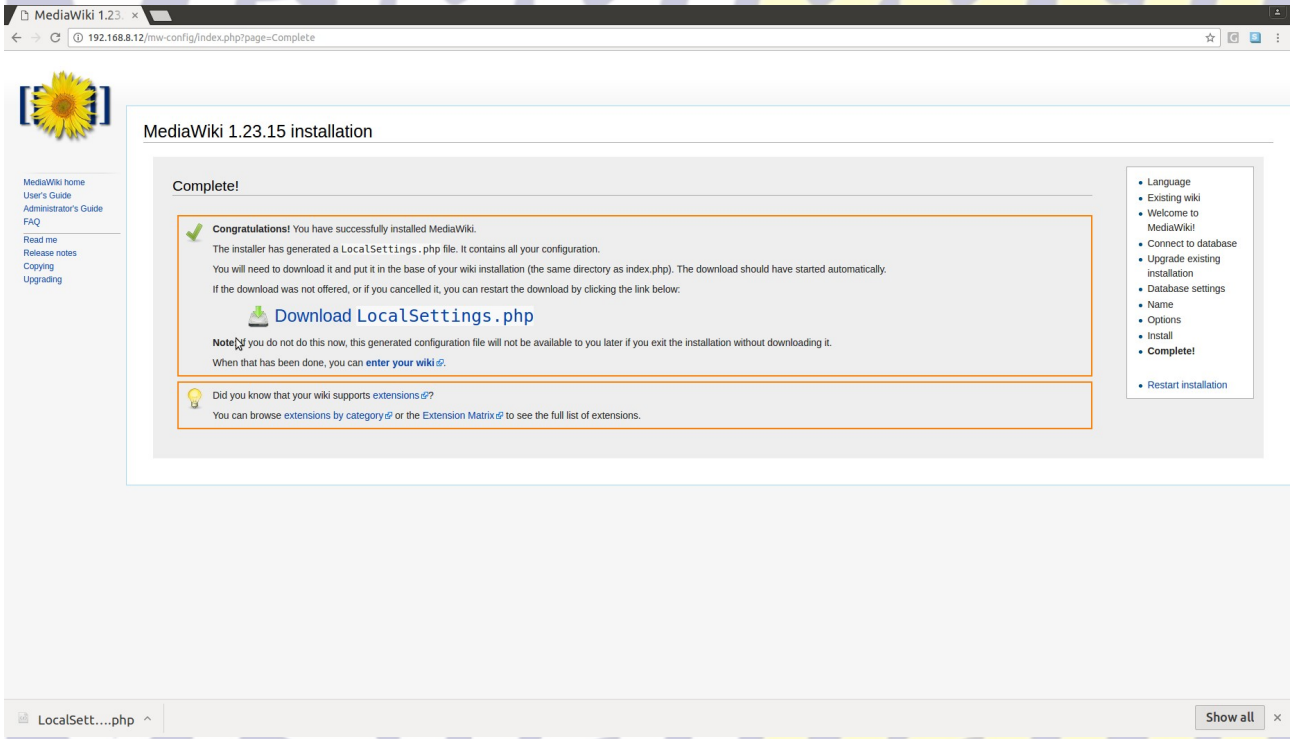
- Set Mediawiki website user & password, use password as “training”.







- After the setup, it will download a file named ‘LocalSettings.php’.



LocalSettings.php file has to be uploaded back to mediawiki site in /var/www/html/LocalSettings.php location. We can do that manually but we will use template to push this file through our puppet code.

6. Templates.

6.1 Setting Variables in Template

- Open localsettings.php file.
- Replace php variables **values** with **puppet variables.**

```
$wgSitename = "<%= wikisitename%>";
$wgMetaNamespace = "<%= wikimetanamespace%>";
$wgServer = "<%= wikiserver%>";
$wgDBserver = "<%= wikidbserver%>";
$wgDBname = "<%= wikidbname%>";
$wgDBuser = "<%= wikidbuser%>";
$wgDBpassword = "<%= wikidbpassword%>";
$wgUpgradeKey = "<%= wikiugradekey%>";
```

- Login to puppetmaster & create template file.

- Create template directory.

```
# mkdir  /etc/puppet/environments/production/modules/mediawiki/templates.
```

- Create LocalSettings.erb template file and paste content of LocalSettings.php we just edited.

```
# vi  /etc/puppet/environments/production/modules/mediawiki/templates/LocalSettings.erb
```

6.2 Declaring template.

- Add template definition in Mediawiki Module.

```
# vi /etc/puppet/environments/production/modules/mediawiki/manifests/init.pp
class { 'firewall': }

 firewall { '000 allow http access':
   port   => '80',
   proto  => 'tcp',
   action => 'accept',

 }

 file {'LocalSettings.php':
   path => '/var/www/html/LocalSettings.php',
   ensure => 'file',
   content => template('mediawiki/LocalSettings.erb'),
 }
```

- Define variables used in template file.

```
# vi /etc/puppet/environments/production/manifests/nodes.pp
```

- Assign variable values for wiki.

```
node 'wiki'{

 $wikisitename = 'wiki'
 $wikimetanamespace = 'Wiki'
```

```
  $wikiserver = "http://192.168.8.12"
  $wikidbserver = 'localhost'
  $wikidbname = 'wiki'
  $wikidbuser = 'root'
  $wikidbpassword = 'training'
  $wikiugradekey = 'puppet'

  class {'linux':}
  class {'mediawiki':}
}
```

- Pull the latest change from wiki.

```
# puppet agent --verbose –onetime --no-daemonize
```

- Go and check the mediawiki website from browser.

7. Hiera

vim /etc/puppet/environments/production/modules/mediawiki/manifests/init.pp

```
class mediawiki {

$wikimetanamespace = hiera('mediawiki::wikimetanamespace')
$wikisitename = hiera('mediawiki::wikisitename')
$wikiserver = hiera('mediawiki::wikiserver')
$wikidbserver = hiera('mediawiki::wikidbserver')
$wikidbname = hiera('mediawiki::wikidbname')
$wikidbuser = hiera('mediawiki::wikidbuser')
$wikidbpassword = hiera('mediawiki::wikidbpassword')
$wikiupgradekey = hiera('mediawiki::wikiupgradekey')

 $phpmysql = $osfamily ? {
   'redhat' => 'php-mysql',
   'debian' => 'php5-mysql',
   default  => 'php-mysql',
 }
```

- Remove variables from nodes.pp file.

- Create Hiera configuration file.

vi /etc/puppet/hiera.yaml

```
:backends:
  - yaml
:yaml:
  :datadir:
:hierarchy:
  - "%{clientcert}"
  - wikidefault
```

- Add variables for wiki node in /var/lib/hiera/wiki.yaml

vi /var/lib/hiera/wiki.yaml

```
---
mediawiki::wikisitename: wiki
mediawiki::wikisitenamespace: Wiki
mediawiki::wikiserver: http://192.168.1.11
mediawiki::wikidbname: wiki
```

- Add variables for wikitest node in /var/lib/hiera/wikitest.yaml

```
---
mediawiki::wikisitename: wikitest
mediawiki::wikisitenamespace: Wikitest
mediawiki::wikiserver: http://192.168.1.12
mediawiki::wikidbname: wikitest
```

- Add common variables in /var/lib/hiera/wikidefault.yaml

```
---
```

mediawiki::wikidbserver: localhost
mediawiki::wikidbuser: root
mediawiki::wikidbpassword: training
mediawiki::wikiupgradekey: puppet

- Execute puppet agent to test it.