# Solution M6: Puppet. Chef. Salt

The following is a one possible solution of the homework assignment

## Task 1

To solve this task, we can refer to this URL: <https://www.vagrantup.com/docs/provisioning/puppet_apply>

Let’s assume that we downloaded the project files and saved them in two sub-folders - **db** and **web**

Of course, we can download and extract the accompanying compressed archive file

Before, we create our **Vagrantfile**, we can create two Puppet manifests

One manifest named **web.pp** with the following content:

# Set a global package parameter

Package { ensure => 'installed' }

# List the required packages

$enhancers = [ 'httpd', 'php', 'php-mysqlnd' ]

# Install the required packages

package { $enhancers: }

service { httpd:

  ensure => running,

  enable => true,

}

file { '/var/www/html/index.php':

  ensure => present,

  source => "/vagrant/web/index.php",

}

file { '/var/www/html/docker.png':

  ensure => present,

  source => "/vagrant/web/docker.png",

}

class { 'firewall': }

firewall { '000 accept 80/tcp':

  action   => 'accept',

  dport    => 80,

  proto    => 'tcp',

}

host { 'web':

  ip           => '192.168.50.2',

  host\_aliases => 'dob-php',

}

host { 'db':

  ip           => '192.168.50.3',

  host\_aliases => 'dob-mysql',

}

# Allow Apache to connect over network

selboolean { 'Apache SELinux':

  name       => 'httpd\_can\_network\_connect',

  persistent => true,

  provider   => getsetsebool,

  value      => on,

}

And another one, named **db.pp** with the following content:

class { '::mysql::server':

  root\_password           => '12345',

  remove\_default\_accounts => true,

  restart                 => true,

  override\_options => {

    mysqld => { bind-address => '0.0.0.0'}

  },

}

mysql::db { 'docker\_info':

  user        => 'root',

  password    => '12345',

  dbname      => 'docker\_info',

  host        => '%',

  sql         => '/vagrant/db/init.sql',

  enforce\_sql => true,

}

class { 'firewall': }

firewall { '000 accept 3306/tcp':

  action   => 'accept',

  dport    => 3306,

  proto    => 'tcp',

}

host { 'web':

  ip           => '192.168.50.2',

  host\_aliases => 'dob-php',

}

host { 'db':

  ip           => '192.168.50.3',

  host\_aliases => 'dob-mysql',

}

Finally, our **Vagrantfile** may contain the following:

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

Vagrant.configure("2") do |config|

  $puppet = <<PUPPET

    sudo dnf install -y https://yum.puppet.com/puppet7-release-el-8.noarch.rpm

    sudo dnf install -y puppet

PUPPET

  $modules = <<MODULES

    puppet module install puppetlabs/mysql

    puppet module install puppetlabs-firewall

    sudo cp -vR ~/.puppetlabs/etc/code/modules/ /etc/puppetlabs/code/

MODULES

  config.vm.provider "virtualbox" do |vb|

    vb.memory = "2048"

  end

  config.vm.box = "shekeriev/centos-8-minimal"

  config.vm.synced\_folder ".", "/vagrant", type: "virtualbox"

  config.vm.provision "shell", inline: $puppet, privileged: false

  config.vm.define 'web' do |web|

    web.vm.hostname = 'web'

    web.vm.network 'private\_network', ip: '192.168.50.2'

    web.vm.provision "shell", inline: $modules, privileged: false

    web.vm.provision "puppet" do |puppet|

      puppet.manifests\_path = "manifests"

      puppet.manifest\_file = "web.pp"

      puppet.options = "--verbose --debug"

    end

  end

  config.vm.define 'db' do |db|

    db.vm.hostname = 'db'

    db.vm.network 'private\_network', ip: '192.168.50.3'

    db.vm.provision "shell", inline: $modules, privileged: false

    db.vm.provision "puppet" do |puppet|

      puppet.manifests\_path = "manifests"

      puppet.manifest\_file = "db.pp"

      puppet.options = "--verbose --debug"

    end

  end

end

As usual, it is matter of executing just the following command:

**vagrant up**

## Task 2

Our infrastructure will have one master node (CentOS), one agent node (CentOS) and a second agent node (Ubuntu)

We will prepare a simple **web.pp** manifest file with the following content:

if $facts['os']['family'] == 'RedHat' {

  $vpackage = 'httpd'

}

else {

  $vpackage = 'apache2'

}

package { $vpackage: }

service { $vpackage:

  ensure => running,

  enable => true,

}

file {'/var/www/html/index.html':

  ensure  => 'file',

  source => "/vagrant/web/index.html",

}

file {'/var/www/html/puppet-logo.png':

  ensure  => 'file',

  source => "/vagrant/web/puppet-logo.png",

}

The **Vagrantfile** will contain the following:

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

Vagrant.configure("2") do |config|

  $hosts = <<HOSTS

    echo '127.0.0.1 localhost' | sudo tee /etc/hosts

    echo '::1 localhost' | sudo tee -a /etc/hosts

    echo '192.168.50.2 master' | sudo tee -a /etc/hosts

    echo '192.168.50.3 node1' | sudo tee -a /etc/hosts

    echo '192.168.50.4 node2' | sudo tee -a /etc/hosts

HOSTS

  $masterrpm = <<MASTERRPM

    sudo dnf install -y https://yum.puppet.com/puppet7-release-el-8.noarch.rpm

    sudo dnf install -y puppetserver

    sudo sed -i 's@Defaults    secure\_path = /sbin:/bin:/usr/sbin:/usr/bin@Defaults    secure\_path = /sbin:/bin:/usr/sbin:/usr/bin:/opt/puppetlabs/bin@g' /etc/sudoers

    sudo puppet config set dns\_alt\_names master

    sudo puppet config set server master

    sudo puppet config set caserver master

    sudo puppet config set reportserver master

    sudo puppetserver ca setup

    sudo mkdir -p $(sudo puppet config print --section master confdir)

    echo node1 | sudo tee $(sudo puppet config print --section master confdir)/autosign.conf

    echo node2 | sudo tee -a $(sudo puppet config print --section master confdir)/autosign.conf

    sudo sed -i 's@-Xms2g -Xmx2g@-Xms512m -Xmx512m@g' /etc/sysconfig/puppetserver

    sudo systemctl enable --now puppetserver

    sudo cp /vagrant/web.pp /etc/puppetlabs/code/environments/production/manifests/web.pp

    sudo firewall-cmd --add-port=8140/tcp --permanent

    sudo firewall-cmd --reload

MASTERRPM

  $agentrpm = <<AGENTRPM

    sudo dnf install -y https://yum.puppet.com/puppet7-release-el-8.noarch.rpm

    sudo dnf install -y puppet-agent

    sudo sed -i 's@Defaults    secure\_path = /sbin:/bin:/usr/sbin:/usr/bin@Defaults    secure\_path = /sbin:/bin:/usr/sbin:/usr/bin:/opt/puppetlabs/bin@g' /etc/sudoers

    sudo puppet config set server master

    sudo puppet config set certname node1

    sudo systemctl enable --now puppet

    sudo puppet config set runinterval 30

    sudo firewall-cmd --add-port=80/tcp --permanent

    sudo firewall-cmd --reload

AGENTRPM

  $agentdeb = <<AGENTDEB

    wget https://apt.puppet.com/puppet7-release-focal.deb

    sudo dpkg -i puppet7-release-focal.deb

    sudo apt-get update

    sudo apt-get install -y puppet-agent

    sudo sed -i 's@/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin@/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin:/opt/puppetlabs/bin@g' /etc/sudoers

    sudo puppet config set server master

    sudo puppet config set certname node2

    sudo systemctl enable --now puppet

    sudo puppet config set runinterval 30

AGENTDEB

  config.vm.provider "virtualbox" do |vb|

    vb.memory = "2048"

  end

  config.vm.synced\_folder ".", "/vagrant", type: "virtualbox"

  config.vm.provision "shell", inline: $hosts, privileged: false

  config.vm.define 'master' do |master|

    master.vm.box = "shekeriev/centos-8-minimal"

    master.vm.hostname = 'master'

    master.vm.network 'private\_network', ip: '192.168.50.2'

    master.vm.provision "shell", inline: $masterrpm, privileged: false

  end

  config.vm.define 'node1' do |node1|

    node1.vm.box = "shekeriev/centos-8-minimal"

    node1.vm.hostname = 'node1'

    node1.vm.network 'private\_network', ip: '192.168.50.3'

    node1.vm.provision "shell", inline: $agentrpm, privileged: false

  end

  config.vm.define 'node2' do |node2|

    node2.vm.box = "shekeriev/ubuntu-20-04-server"

    node2.vm.hostname = 'node1'

    node2.vm.network 'private\_network', ip: '192.168.50.4'

    node2.vm.provision "shell", inline: $agentdeb, privileged: false

  end

end

As usual, it is matter of executing just the following command:

**vagrant up**