

```
#Copyright(c) Galia A Lisovskaya aka sHaggY_caT
#Licensed by GNU GPL v2 or later
```

```
#Clases and defenitions for openvz containers management
```

```
#This class upload to HardwareNode new system configs, templates, and other files
```

```
class mod_class_ovzconfigs {
```

```
#Define base configs names for typical tarif planes
```

```
#define our system configs path
```

```
$vps_conf_path="/etc/sysconfig/vz-scripts/"
```

```
#upload openvz systctl.conf
```

```
  file {["/etc/systctl.conf":
    replace => true,
    owner   => "root",
    group   => "root",
    mode    => "0744",
    source  => "puppet:///mod_ovz/systctl.conf",
  }
```

```
#Upload openvz templates, if we find lock file. For update templates on hardware node, you may
create this lock file.
```

```
#exec { "Update ovz templates":
```

```

# command => "wget -O /vz/template/cache/centos-5-i386-default.tar.gz
http://install.local/ovz/templates/centos-5-i386-default.tar.gz && rm /vz/firsh_install.lock",
# path => "/usr/bin:/usr/sbin:/bin",
# onlyif => "test -f /vz/firsh_install.lock"
#}

#exec { "${templates}":

# command => "wget -O /vz/template/cache/${name}.tar.gz $template_url/${name}.tar.gz && rm
/vz/firsh_install.lock",
# path => "/usr/bin:/usr/sbin:/bin",
# onlyif => "test -f /vz/firsh_install.lock"
#}
#$config_exim_trusted_users = split(get_var('config_exim_trusted_users'), ',')

file {"vps756M":
  path => "${vps_conf_path}/ve-vps756M.conf-sample",
  source => "puppet:///mod_ovz/vps756M.conf",
  ensure => present,
  replace => true,
  #before => Exec["vcreate_${name}"],
# exec { "vzlist -ao veid,description | grep ${vps756M_name} | awk '{print $1}' | xargs vzctl
--applyconfig ":
# path => "/usr/bin:/usr/sbin:/bin:/sbin"
# }

    }

file {"vps128M":
  path => "${vps_conf_path}/ve-vps128M.conf-sample",
  source => "puppet:///mod_ovz/vps128M.conf",
  ensure => present,
  replace => true, }

file {"vps256M":
  path => "${vps_conf_path}/ve-vps256M.conf-sample",
  source => "puppet:///mod_ovz/vps256M.conf",
  ensure => present,
  replace => true, }

file {"vps512M":
  path => "${vps_conf_path}/ve-vps512M.conf-sample",
  source => "puppet:///mod_ovz/vps512M.conf",
  ensure => present,
  replace => true, }

}

```

```

define mod_class_ovzconfigs::mod_def_os_templates ($templates, $template_url){

    file {"${fqdn}_templates":
        path => "/vz/${fqdn}_templates.list",
        ensure => present,
        replace => true,
        source => $templates ? {
            "custom" => "puppet:///mod_ovz/${fqdn}_standart_templates.list",
            "standart" => "puppet:///mod_ovz/standart_templates.list", },
        before => Exec["${fqdn}_download_templates"],
    }

    exec {"${fqdn}_download_templates":
        command => "/usr/sbin/templates-download.sh ${fqdn} ${template_url}",
        path => ["/usr/bin", "/usr/sbin", "/bin", "/sbin"],
        subscribe => File["${fqdn}_templates"],
        require => File["${fqdn}_templates"],
        refreshonly => true, }

    }

define mod_class_ovzconfigs::mod_def_ovzbackup ($ensure = "off", $path_run_backup,
$hour_incr_cron,
$min_incr_cron, $day_incr_cron, $hour_zero_cron, $min_zero_cron, $day_zero_cron,
$type_of_backup = "local") {

case $ensure {

    off : {
        cron {"incremental_backup":
            ensure => absent, }

        cron {"zero_backup":
            ensure => absent,
        }
    }

    on : {
        cron {"incremental_backup":
            ensure => present,
            command => "${path_run_backup} ${type_of_backup} 1",
            user => root,
            hour => $hour_incr_cron,
            minute => $min_incr_cron,
            weekday => $day_incr_cron,
        }
    }
}

```

```

    cron {"zero_backup":
        ensure => present,
        command => "${path_run_backup} ${type_of_backup} 0",
        user => root,
        hour => $hour_zero_cron,
        minute => $min_zero_cron,
        weekday => $day_zero_cron,
    }
}
    }
}

```

```

define mod_class_ovzconfigs::mod_def_veadm ($status, $veid, $vetemplate =
"centos-5-i386-default", $vetarif,
$veip, $vedns = "10.0.5.51", $setboot) {

```

```

#Define base configs names for typical tarif planes

```

```

#define our system configs path
$veps_conf_path="/etc/sysconfig/vz-scripts/"

```

```

$vehostname = $name

```

```

case $status {

```

```

#Create new container
vecreate : {

```

```

    exec {"vecreate_${name}":
        command => "vzctl create ${veid} --ostemplate ${vetemplate} --config ${vetarif} --hostname
${vehostname} --ipadd ${veip} && vzctl set ${veid} --nameserver ${vedns} --onboot ${setboot}
--save && vzctl start ${veid}",
        path => "/usr/bin:/usr/sbin:/bin",
        unless => "test -d /vz/private/${veid}",
        require => File["${vetarif}"],
        before => Exec["ve_check_status_${name}"],
    }

```

```

    }
#Remove container
vedestroy : {

```

```

    exec {"vedestroy_${name}":
        command => "vzctl stop ${veid} --fast; vzdump --compress ${veid} && vzctl destroy ${veid}",
        path => "/usr/bin:/usr/sbin:/bin",

```

```

        onlyif => "vzlist ${veid}" }
    }
#Create not-standart container
vecustom : {

file {"vecus_${name}":
    path => "${vps_conf_path}/${veid}.conf",
    content => template("mod_ovz/${name}_vps.erb"),
    ensure => present,
    replace => true,
    before => Exec["vecrcus_${name}", "vecus-apply_${name}"],
    }

    exec {"vecrcus_${name}":
        command => "vzctl create ${veid} && vzctl start ${veid}",
        path => "/usr/bin:/usr/sbin:/bin",
        unless => "test -d /vz/private/${veid}",
        require => File["vecus_${name}"],
        }

#If vps off, stop it, if on, start it
exec { "vecus_check_status_${name}":
    command => "/usr/sbin/apply_ve_status.sh ${veid} ${setboot}",
    path => ["/usr/bin", "/usr/sbin"],
    onlyif => "test `usr/sbin/check_ve_status.sh ${veid} ${setboot}` = 0 "
    }

exec { "vecus-apply_${name}":
    command => "if [ -d /vz/private/${veid} ]; then vzctl restart ${veid}; fi",
    path => ["/usr/bin", "/usr/sbin"],
    onlyif => "test -d /vz/private/${veid}",
    subscribe => File["vecus_${name}"],
    require => File["vecus_${name}"],
    refreshonly => true
    }
}

}

if $status != "vecustom" {

#If tarif settings changed, aplly new tarif
exec { "veconfig-apply_${name}":
    command => "if [ -d /vz/private/${veid} ]; then vzctl set ${veid} --applyconfig ${vetarif} --save; fi",
    path => ["/usr/bin", "/usr/sbin"],
    onlyif => "test -d /vz/private/${veid}",
    subscribe => File["${vetarif}"],
    refreshonly => true

```

```

    }
#If VPS settings changed, apply new settings
exec { "ve_check_tarif_${name}":
    command => "vzctl set ${veid} --applyconfig ${vetarif} --save",
    path => ["/usr/bin", "/usr/sbin"],
    onlyif => "test `usr/sbin/check_tarif.sh ${veid} ${vetarif}` = 0 ",
    }
#If vps off, stop it, if on, start it
exec { "ve_check_status_${name}":
    command => "/usr/sbin/apply_ve_status.sh ${veid} ${setboot}",
    path => ["/usr/bin", "/usr/sbin", "/bin", "/sbin"],
    onlyif => "test `usr/sbin/check_ve_status.sh ${veid} ${setboot}` = 0 "
    }
#set new IP for container
exec {"ve_check_ip_${name}":
    command => "/usr/sbin/ve-apply_ip.sh ${veid} ${veip}",
    path => ["/usr/bin", "/usr/sbin", "/bin", "/sbin"],
    onlyif => "test `usr/sbin/ve-check_ip.sh ${veid} ${veip}` = 0 "
    }

#set new DNS for container
exec {"ve_check_dns_${name}":
    command => "/usr/sbin/ve-apply_dns.sh ${veid} ${vedns}",
    path => ["/usr/bin", "/usr/sbin", "/bin", "/sbin"],
    onlyif => "test `usr/sbin/ve-check_ip.sh ${veid} ${vedns}` = 0 "
    }
}
}

```

```

node 'ovz-test2.local' inherits ovzserver-custom {

```

```

    mod_class_ovzconfigs::mod_def_os_templates {"templates-ovz-test2.local":
    template_url => "http://install.local/ovz/templates/",
    templates => "standart",
    }

```

```

    mod_class_ovzconfigs::mod_def_ovzbackup {"backup-ovz-test2.local":

```

```
path_run_backup => "/usr/sbin/rc.HABackup.sh",
hour_incr_cron => "12",
min_incr_cron => "10",
day_incr_cron => "*",
hour_zero_cron => "4",
min_zero_cron => "0",
day_zero_cron => "5",
type_of_backup => "remote",
ensure => "on", }
```

mod_class_ovzconfigs::mod_def_veadm {"customer11.vps.local":

```
veid => "401",
setboot => "yes",
vetemplate => "ubuntu-9.04-x86",
vetarif => "vps512M",
veip => "10.0.5.41",
vedns => "10.0.5.51",
status => "vcreate",
}
```

mod_class_ovzconfigs::mod_def_veadm {"customer12.vps.local":

```
veid => "402",
setboot => "yes",
vetemplate => "centos-4-x86",
vetarif => "vps256M",
veip => "10.0.5.42",
vedns => "10.0.5.51",
status => "vcreate",
}
```

mod_class_ovzconfigs::mod_def_veadm {"customer13.vps.local":

```
veid => "403",
setboot => "yes",
vetemplate => "debian-5.0-x86",
vetarif => "vps128M",
veip => "10.0.5.43",
vedns => "10.0.5.51",
status => "vcreate",
}
```

mod_class_ovzconfigs::mod_def_veadm {"customer14.vps.local":

```
veid => "404",
setboot => "yes",
```

```
vetemplate => "suse-11.1-x86",  
vetarif => "vps756M",  
veip => "10.0.5.44",  
vedns => "10.0.5.51",  
status => "vcreate",  
}
```

```
mod_class_ovzconfigs::mod_def_veadm {"customer15.vps.local":
```

```
veid => "405",  
setboot => "no",  
vetemplate => "fedora-12-x86",  
vetarif => "vps256M",  
veip => "10.0.5.45",  
vedns => "10.0.5.51",  
status => "vcreate",  
}
```

```
mod_class_ovzconfigs::mod_def_veadm {"customer16.vps.local":
```

```
veid => "406",  
setboot => "yes",  
vetemplate => "centos-5-x86",  
vetarif => "vps512M",  
veip => "10.0.5.46",  
vedns => "10.0.5.51",  
status => "vcreate",  
}
```

```
mod_class_ovzconfigs::mod_def_veadm {"test-dns.local":
```

```
veid => "407",  
setboot => "yes",  
vetemplate => "centos-5-x86",  
vetarif => "vps512M",  
veip => "10.0.5.47",  
vedns => "10.0.5.51",  
status => "vcreate",  
}
```

```
mod_class_ovzconfigs::mod_def_veadm {"customer17.vps.local":
```

```
veid => "407",  
setboot => "yes",  
vetemplate => "centos-5-i386-default",  
vetarif => "vps256M",  
veip => "10.0.5.47",  
vedns => "10.0.5.51",
```

```
status => "vcreate",  
}
```

```
}
```