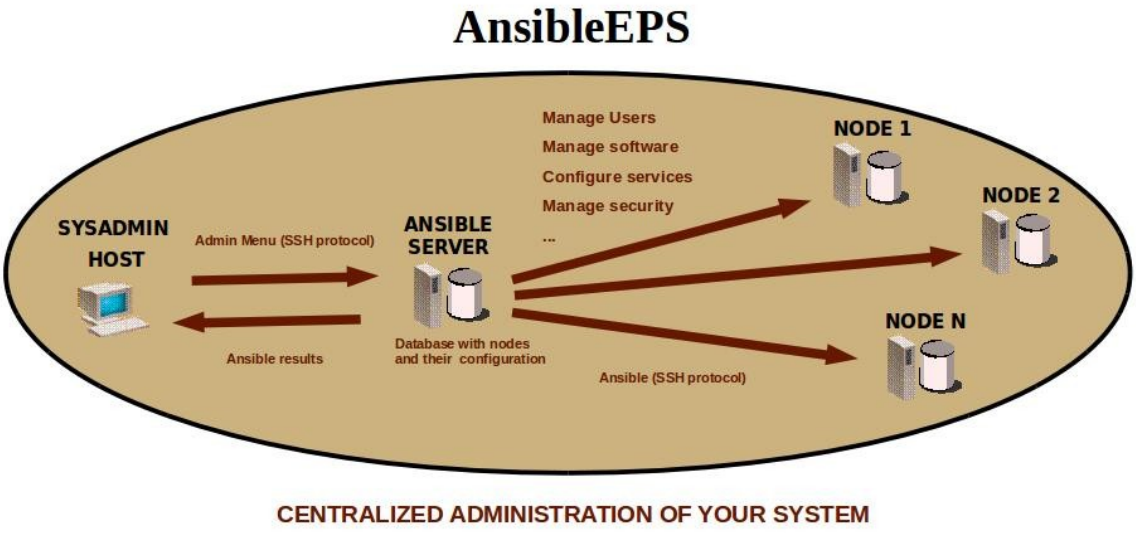
Ansible EPS

What's ansibleEPS?

AnsibleEPS is a **collection of ansible playbooks** (little ansible programs) to manage Linux/Unix hosts of your network.

Management permits in a centralized way: **software management** (install, update, delete), **services** (start, stop, configure), **users** (add, modify, delete), **security**, and much more.



What can I do with AnsibleEPS?

Especifically, ansibleEPS does the following tasks:

Common management for all hosts (installation, configuration and checking)	● Locales ● Groups and users ● Repositories ● Concurrency ● Services to disable ● /etc/resolv.conf file ● Nscd service ● Securetty ● Global profile variables ● Nvi editor ● Utilities (extra software) ● Munin-node client ● Nrpe client ● Ldap Authentication with PAM ● Syslog service ● Sshd service ● Ntp service ● Bacula client ● Ossec client
Hosts file (/etc/hosts) management for all hosts (configuration and checking)	Three level configuration ● Global, to apply on all hosts ● Group, to apply on all hosts belonging to group selected ● Host, to apply on host selected

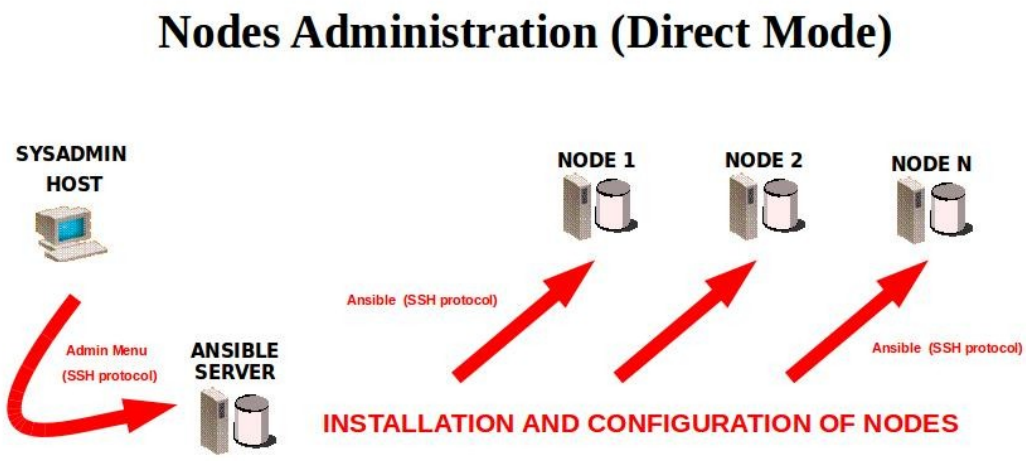
Sudo file (/etc/sudoers) management for all hosts (installation, configuration and checking)	Three level configuration ● Global, to apply on all hosts ● Group, to apply on all hosts belonging to group selected ● Host, to apply on host selected
TCP Wrappers (/etc/hosts.allow & /etc/hosts.deny files) management for all hosts (configuration and checking)	Three level configuration ● Global, to apply on all hosts ● Group, to apply on all hosts belonging to group selected ● Host, to apply on host selected
PAM Access (/etc/security/access.conf file) management for all hosts (installation, configuration and checking)	Three level configuration ● Global, to apply on all hosts ● Group, to apply on all hosts belonging to group selected ● Host, to apply on host selected
NRPE management for all hosts (installation, configuration and checking)	Three level configuration ● Global, to apply on all hosts ● Group, to apply on all hosts belonging to group selected ● Host, to apply on host selected
IPTables management for all hosts (configuration and checking)	Three level configuration ● Global, to apply on all hosts ● Group, to apply on all hosts belonging to group selected ● Host, to apply on host selected
Crontab management for all hosts (configuration and checking)	Three level configuration ● Global, to apply on all hosts ● Group, to apply on all hosts belonging to group selected ● Host, to apply on host selected
Proxmox servers management (configuration and checking)	Configuration of Proxmox servers
My.cnf file management for mysql servers (configuration and checking)	Three level configuration ● Global, to apply on all hosts ● Group, to apply on all hosts belonging to group selected ● Host, to apply on host selected
Apache includes management for apache servers (configuration and checking)	Three level configuration ● Global, to apply on all hosts ● Group, to apply on all hosts belonging to group selected ● Host, to apply on host selected
Windows NRPE management for Windows hosts (configuration and checking)	Three level configuration ● Global, to apply on all hosts ● Group, to apply on all hosts belonging to group selected ● Host, to apply on host selected
Bacula servers management (configuration and checking)	Configuration of Bacula servers

DHCP servers management (configuration and checking)	Configuration of DHCP servers
Munin servers management (configuration and checking)	Configuration of Munin servers
Nagios servers management (configuration and checking)	Configuration of Nagios servers
Email addresses list of QMail server	Addresss list generation of Qmail server

How does it work?

Management can be done in **two ways** (we can use both):

(A) **Direct Mode**, using the **Admin Menu**, and selecting preferred option and arguments as: host to be managed, group of hosts to be managed, or any other argument required.



```
############## ADMIN MENU EPS #################
##                                            ##
##  0. Help                                   ##
##--------------------------------------------##
##  1. Common module (nodes config)           ##
##  2. Hosts File module (nodes config)       ##
##  3. Sudoers module (nodes config)          ##
##  4. TCP Wrappers module (nodes config)     ##
##  5. PAM Access module (nodes config)       ##
##  6. NRPE module (nodes config)             ##
##  7. IPTables module (nodes config)         ##
##  8. Crontab module (nodes config)          ##
##  9. Proxmox module (Proxmox nodes config)  ##
## 10. My.cnf file (Mysql nodes config)       ##
## 11. Apache includes (Apache nodes config)  ##
## 12. All modules (nodes config)             ##
##--------------------------------------------##
## w1. Win NRPE module (Windows nodes config) ##
##--------------------------------------------##
##  b. Bacula Servers (bacula config)         ##
##  d. DHCP Servers (DHCP config)             ##
##  m. Munin Servers (munin config)           ##
##  n. Nagios Servers (nagios config)         ##
##  i. Email addresses List (getting data)    ##
##--------------------------------------------##
##  a. Add Node(s)                            ##
##  s. Stop/Start/Restart System              ##
##  l. Check System Logs (Errors and Changes) ##
##  c. Clean System Logs                      ##
##  x. View executions list                   ##
##  r. Log Running Executions (Crt+C to Exit) ##
##  u. Update (servers update)                ##
##  v. Inventory List                         ##
##  q. Quit Menu                              ##
##                                            ##
################################################

Select option:
```

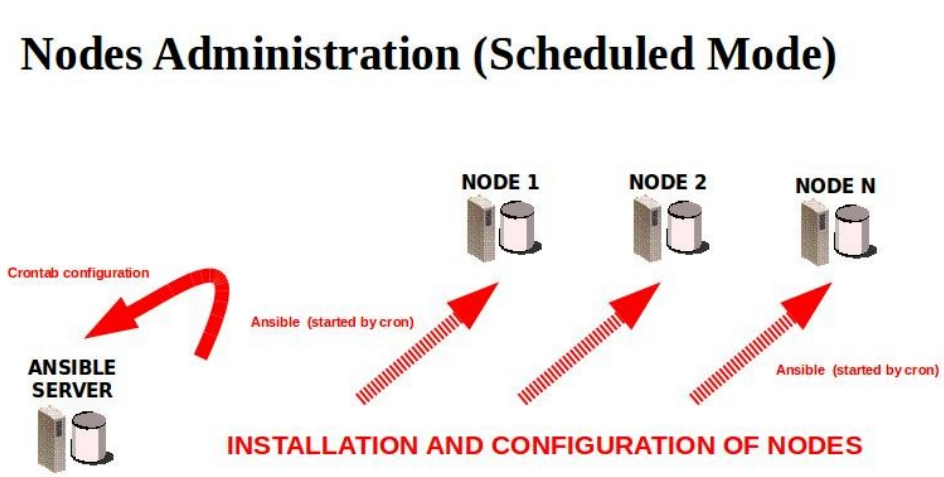
This way, we can manage anything for a host, a group or all hosts, just selecting option and arguments

(B) **Scheduled Mode**, we can select time (usually at night) to execute all managements for all hosts. This way we'll be sure that next morning every feature of every hosts will be correctly configured.

All changes made and errors produced will be saved in log files to be watched next morning.

Installation & Configuration

We have to install and prepare system to be ready for management. There are some step to do:



Installation

(1) Copy tgz package **'ansibleEPS.tgz'** and installation script **'install.py'** (located in **install** directory) in a directory on a CentOS 6 host (this will be our 'Ansible Server').

(2) Execute **'install.py'** script to install 'AnsibleEPS'.

Configuration

(1) First of all, we have to **configure a host as node**. What's a node?

A node is a host directly SSH accessed by Ansible server (host where Ansible was installed) with a predefined user and no password.

By default, user predefined connecting to nodes is 'ansible', but we can change it modifying 'ansible_ssh_user' variable in '/etc/ansibleEPS/group_vars/all/all' main configuration file.

To configure a host as node, simply select **'add node'** option in **Admin menu** '/etc/ansibleEPS/menu.py'. Select 'user to connect' and hostname or IP. The script will try to connect by SSH to host as 'root' (we have to type password), then it will create 'user to connect' and 'authorized_keys' file with 'Ansible server' public key, also it will install 'sudo' package and configure 'sudoers' file permitting execute everything to 'user to connect'.

- Special case for **Windows hosts**:

A Windows hosts needs at least **PowerShell 3.0** and **.NET 4.0** (or newer), and a **WinRM** listener installed.

This scripts install and configure the WinRM listener (run in PowerShell):

```
$url = "https://raw.githubusercontent.com/ansible/ansible/devel/examples/scripts/ConfigureRemotingForAnsible.ps1"
$file = "$env:temp\ConfigureRemotingForAnsible.ps1"
(New-Object -TypeName System.Net.WebClient).DownloadFile($url, $file)
powershell.exe -ExecutionPolicy ByPass -File $file
```

Finally, we have to config 'Windows connection', creating an **administrator user** ('ansible' for example), and configure '**/etc/ansibleEPS/group_vars/windows**' file:

```
# Windows connection
ansible_user: ansible
ansible_password: XXXXXXX
ansible_port: 5986
ansible_connection: winrm
ansible_winrm_server_cert_validation: ignore
```

(2) Second step: **add nodes to inventory**

Inventory file is **'/etc/ansibleEPS/eps'**. Nodes are introduced to inventory inside groups.

Ansible Documentation at: http://docs.ansible.com/ansible/intro_inventory.html#inventory

(3) **Modify Admin variables**. Every data we need to manage nodes are located at:

- **'/etc/ansibleEPS/group_vars/all'** directory, with global data to manage nodes
- **'/etc/ansibleEPS/group_vars'** directory, with group-level data to manage nodes
- **'/etc/ansibleEPS/host_vars'** directory, with host-level data to manage nodes

Example: In 'all' configuration file exists a variable 'nameserver' with IP's of DNS servers for all nodes. It can be different for a group of hosts (a group called 'remote', defined at inventory with nodes inside), then DNS server has change, so we have to create a file '/etc/ansibleEPS/group_vars/remote' and define 'nameserver' variable inside with different values. The same way, DNS servers can be different for a specific node (called 'special'), so we have to create a file '/etc/ansibleEPS/host_vars/special' and define 'nameserver' variable inside with other values.

(4) **Execute playbooks** from Admin menu '/etc/ansibleEPS/menu.py'

Does it work for all Operating System?

Most playbooks need to install specific software on nodes. By default playbooks work with 'yum' and 'apt' package managers. To add new package managers, its necessary to modify playbooks code.

System is ready to work with the following Operating System (all of them with 'yum' or 'apt' package manager):

- CentOS 6
- CentOS 7
- Debian 4
- Debian 6
- Debian 7
- Debian 8
- Ubuntu 14
- Ubuntu 16
- Ubuntu 18
- Windows (PowerShell 3.0+, .NET 4.0+)

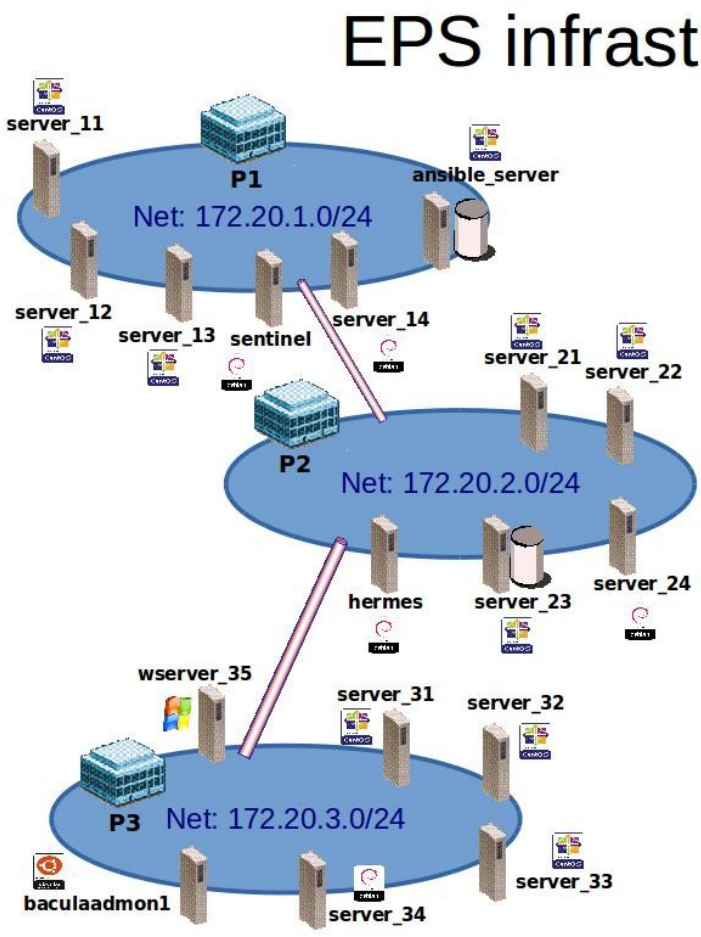
There's a specific file for every Operating System and version in '/etc/ansibleEPS/group_vars' directory. They include variables and specific values. For example: specific users, repositories, software, configuration files, etc.

Adding a new Operating System and Version (based on 'yum' or 'apt' package manager) is very simple: just create a new file in '/etc/ansibleEPS/group_vars/' directory with name 'OperatingSystem-Version' and fill it with variables and specific values (easier copying other similar file and modifying values).

Example: To prepare system for RedHat 7 nodes, we can copy 'CentOS-6' file (the most similar) to 'RedHat-7', verifying values and changing if necessary.

Basic infrastructure example

AnsibleEPS has a basic inventory and variables created as a example of use. This is the insfrastructure schema:



Service	Host (IP)
DHCP	server_12 (172.20.1.12) server_22 (172.20.2.22) server_32 (172.20.3.32)
DNS	server_11 (172.20.1.11) server_21 (172.20.2.21) server_31 (172.20.3.31)
wordpress	server_11 (172.20.1.11) server_21 (172.20.2.21)
LDAP	server_13 (172.20.1.13) server_23 (172.20.2.23) server_33 (172.20.3.33)
mysql	server_23 (172.20.2.23)
proxmox	server_14 (172.20.1.14) server_24 (172.20.2.24) server_34 (172.20.3.34)
bacula	baculadmon1 (172.20.3.30)
qmail	hermes (172.20.2.20)
nagios	sentinel (172.20.1.15) server_12 (172.20.1.12)
munin	server_22 (172.20.2.22)
ansible	ansible_server (172.20.1.10)
ossec	server_12 (172.20.1.12)
apache	hermes (172.20.2.20) sentinel (172.20.1.15) server_12 (172.20.1.12) server_22 (172.20.2.22)
windows	wserver_35 (172.20.3.35)