1. **/etc/ansible directory  
     
   [ansitest@basuritam22c ansible]$ ls -lrt**

**total 24**

**drwxr-xr-x. 2 root root 6 Dec 14 05:57 roles 🡨 default roles path**

**-rw-r--r--. 1 root root 1016 Dec 14 05:57 hosts 🡨 default file for hosts**

**-rw-r--r--. 1 root root 20277 Dec 14 05:57 ansible.cfg <- the default config file**

**[ansitest@basuritam22c ansible]$ pwd**

**/etc/ansible  
  
1.2) Configuration for password less communication**

1. **HOSTS FILE :**

**2.1) Grouping your hosts in the hosts file :**

**[local]**

**Localhost**

**Localhost.Localdomain**

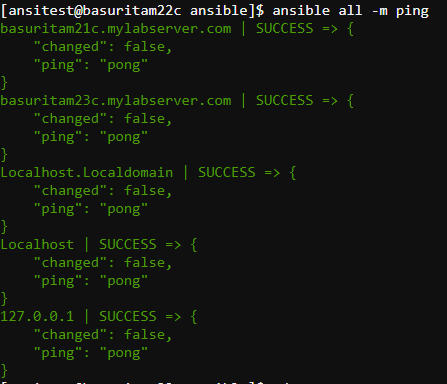
**127.0.0.1**

**[webhosts]**

**basuritam21c.mylabserver.com**

**[apphosts]**

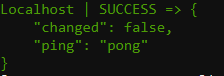
**basuritam23c.mylabserver.com**

**Now if I try to ping all hosts it will ping all entries.  
  
**

**So, it’s better to have one single entry for one server.**

**2.2) Overriding the HOSTS File  
  
remember we have 3 entries in [local] in my default hosts file at /etc/ansible/hosts.**

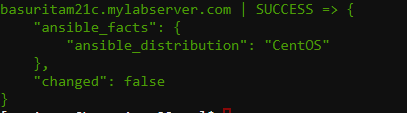
**2.2.1) via command line .  
  
pwd 🡪 /home/ansitest**$ ansible local -i /home/ansitest/exps/hosts -m ping

**-I <path\_to\_hosts\_file>  
  
2.2.2) From ansible.cfg**#inventory = /etc/ansible/hosts **old**  
  
inventory = /home/ansitest/exps/hosts  **new**   
 **and do 🡪** $ ansible local -m ping  
 **for both the cases O/P:**  
  


**3) Overriding the ansible.cfg file :  
  
Priority , where ansible looks for ansible.cfg  
  
A 🡪 export ANSIBLE\_CONFIG=/home/ansitest/exps/ansible.cfg   
 define ANSIBLE\_CONFIG var.  
  
B 🡪 Current Directory   
  
C 🡪 ~/.ansible.cfg ] remember .ansible.cfg at usre’s home dir  
  
D 🡪 /etc/ansible/ansible.cfg**

1. **PATH to your first PLAYBOOK.  
     
   3.1) Executing from command Line.  
   Some Examples.  
     
   3.1.1) List all your hosts from ansible command line  
   the python packages installed on hosts webhosts:**ansible webhosts -b -m shell -a 'yum list installed | grep python'

**-b, --become run operations with become (does not imply**

**password prompting)  
  
-m 🡪 Invoking module  
-a 🡪 passing arguments   
  
3.1.2) gather system facts of one host and save it to a file .**ansible webhosts -m setup --tree ~/webhosts\_facts  
  
**setup 🡪 module is used to gather\_facts**  
**Output will be json formatted data.**  
 **Filer some values .**ansible webhosts -m setup -a "filter=ansible\_distribution"  
  
****

**3.2) Command Line to playbook conversion :  
  
Install lynx package on webhosts   
  
command line 🡪** ansible webhosts -b -m yum -a "name=lynx state=installed update\_cache=true"

playbook1)

- **hosts: webhosts**

**tasks:**

**- name: Install lynx on webserver**

**yum: pkg=lynx state=installed update\_cache=true**

****

**GREEN 🡪 NO CHANGE  
YELLOW 🡪 CHANGE  
RED 🡪 FAILED**

**4) PLAYBOOK SECTION:  
4.1) TARGET SECTION :  
  
Edit the playbook and add the following to the target section we just created:**

**- Force use of SSH connections**

**- Always run the playbook as the user 'ansitest'**

**- Run this playbook as SUDO by default**

**- Do not use the setup module to gather facts from systems during execution  
  
  
target Section :**

**--- # Target Section**

**- hosts: apphosts**

**connection: ssh**

**user: ansitest**

**sudo: yes**

**gather\_facts: no**

**4.2) VARIABLE SECTION:  
  
3 types of variables  
  
a) Defined inside the playbook   
b) Defined outside of playbook in a separate file  
c) Assigned dynamically during playbook runtime   
  
4.2.1) Add 3 types of variables in your playbook .  
we have  
 ├── conf**

**├── copyright.yml**

**└── webdefaults.yml files containing var values**

**--- # Variable Section**

**- hosts: apphosts**

**connection: ssh**

**user: ansitest**

**sudo: yes**

**gather\_facts: no**

**vars:**

**playbook\_version: 0.1b**

**vars\_files:**

**- conf/copyright.yml**

**- conf/wbdefaults.yml**

**vars\_prompt:**

**- name: web\_domain**

**prompt: Web Domain-**

* **Conf   
  remember the ‘-‘  
    
  everything , referenced or sourced from outside the playbook should proceed with a  
  ‘-‘**

**4.3) TASK SECTION :**

**tasks:**

**- name: Install lynx on webserver**

**yum: pkg=lynx state=installed update\_cache=true**

**4.4) NOTIFY & HANDLERS:  
  
HANDLERS SECTION:**

Pr)   
**- Uses SSH**

**- Logs in to the remote system as 'ansitest' user**

**- Connects to one server or group from Step #2 above**

**- The playbook runs as 'sudo'**

**- Skip gathering remote facts**

**- Installs the 'Apache Web Server' using the appropriate package module**

**- Upon installation of the web server, notifies the appropriately titled handler to restart the service using the 'service' module.**

**--- # Handler Section**

**- hosts: webhosts**

**connection: ssh**

**user: ansitest**

**sudo: yes**

**gather\_facts: no**

**tasks:**

**- name: install httpd**

**yum: name=httpd state=installed**

**notify: Restart httpd**

**handlers:**

**- name: Restart httpd**

**action: service name=httpd state=restarted**



**Remember to give same name for notify and handlers section.**