

# AtoM Camp: Installing AtoM with Ansible

**Document author:** David Juhasz

**Date and time:** Tuesday, 09 May 2017, 1pm - 2pm

**Instructor:** Steve Breker

## 1. Quick overview of Ansible

- Automation and orchestration - like Chef, Puppet
- FOSS, Project now managed and copyrighted by RedHat
- Written in python and PowerShell
- Agentless structure - no minimal software required on installation nodes (python is required)
- Idempotent (unless you do it wrong)
- Uses SSH for secure connection and communication with remotes
- Config uses [YAML](#) and [Jinja templates](#).

## 2. Install ansible

- Centos/Redhat:

```
$ sudo yum install ansible
```

- Ubuntu/Debian:

```
$ sudo apt-get install software-properties-common
$ sudo apt-add-repository ppa:ansible/ansible
$ sudo apt-get update
$ sudo apt-get install ansible
```

- MacOSX: Install with pip

```
$ sudo pip install ansible
```

- Windows: not supported (for control machine)

## 3. Clone <https://github.com/djjuhasz/deploy-pub/tree/dev/add-ufw-role>

```
$ git clone -b dev/atomcamp-deploy \
git@github.com:djjuhasz/deploy-pub.git
$ cd deploy-pub/playbooks/atom-xenial/
```

## 4. Overview of ansible directory structure and file types:

- No hard and fast rules as far as I can tell. There are many variations on the “recommended” directory structure shown below.

```

production          # inventory file for production servers
staging             # inventory file for staging environment

group_vars/
  group1            # here we assign variables to particular
  group2            # groups

host_vars/
  hostname1         # if systems need specific variables, put
  hostname2         # them here

library/            # if any custom modules, put them here
(optional)
filter_plugins/     # if any custom filter plugins, put them here
(optional)

site.yml            # master playbook
webservers.yml      # playbook for webserver tier
dbservers.yml       # playbook for dbserver tier

roles/
  common/           # this hierarchy represents a "role"
    tasks/          #
      main.yml       # <-- tasks file can include smaller files
                     # if warranted
    handlers/        #
      main.yml       # <-- handlers file
    templates/       # <-- files for use with the template
                     # resource
      ntp.conf.j2    # <----- templates end in .j2
    files/           #
      bar.txt        # <-- files for use with the copy resource
      foo.sh         # <-- script files for use with the script
                     # resource
    vars/            #
      main.yml       # <-- variables associated with this role
    defaults/        #
      main.yml       # <-- default lower priority variables for
                     # this role
    meta/            #
      main.yml       # <-- role dependencies
    library/         # roles can also include custom modules
    lookup_plugins/  # or other types of plugins, like lookup in
                     # this case

```

- **playbook** - top level of ansible script hierarchy - a collection of roles that are run again a set of hosts.
- **roles** - second level script files used to define a set of steps for deploying for a particular "role" (e.g. atom, elasticsearch, mysql).

- Roles can be defined locally, or can be linked and downloaded via a “requirements.yml” file.
  - Roles can all be deployed to one host or split among multiple hosts.
  - Include tasks, handlers, variable files and more.
- **tasks** - third level of script file hierarchy. Smaller units of code, that define the tasks required for a deployment (e.g. copy config file, restart php, install imagemagick package)
- **blocks** - a single command within a task file, using multiple lines to split up parameters, calls, options, etc.
- **handlers** - tasks that are run once no matter how many times they are called (triggered by a “notify” block)
- **vars** - variables for roles and tasks
- **defaults** - defaults for variables and config settings; “vars” overwrite these defaults.
- **templates** - config file templates for installed apps. Written in “Jinja” templating language (.j2)
- **meta** - metadata about playbook (author, license, etc.)
- 5. Not included in “recommend” structure, but used in Artefactual playbooks
  - **hosts.yml** - inventory file. A list of host systems that can be targets for deployment. Groups of hosts (e.g. “webserver”) can be used to target playbooks to a number of hosts simultaneously (e.g. updating nginx for all webserver hosts).
  - **requirements.yml** - manifest file of roles required for a playbook.
  - **LICENSE**
  - **README.md**
- 6. Download remote roles
  - Many are from <https://github.com/artefactual-labs>

```
$ ansible-galaxy install -f -p roles/ -r requirements.yml
```

- 7. Edit local config variables

- hosts.yml

```
[atom]
192.168.1.1
```

- vars-singlenode.yml

- 8. Deploy!

```
$ ansible-playbook singlenode.yml \
  --inventory-file="./hosts.yml" \
  --user="root" \
  --extra-vars="atom_flush_data=yes" \
  --verbose
```

## 9. Ansible ecosystem

- Galaxy - community roles
- Vault - At rest encryption of sensitive data
- Tower - Paid GUI management and reporting tool

## 10. Questions and troubleshooting