Ansible tools to improve quality

Ansible (https://github.com/ansible/ansible) is easy to learn (https://www.ansible.com/blog/getting-started-writing-your-first-playbook), but difficult to master (http://yaml.org/). You'll save time if you invest a little time to setup your development environment.

There are quite a few tools that help improve the quality of Ansible (https://github.com/ansible/ansible) code. Some can be used when writing code, others on integration.

Writing Ansible code

Although vi is a great editor (http://www.viemu.com/a-why-vi-vim.html) and can be setup to work with Ansible (https://github.com/pearofducks/ansible-vim), when writing Ansible (https://github.com/ansible/ansible) code, multiple files are used. This makes an IDE useful.

Atom

1. atom (https://atom.io/) because it renders YAML nicely out of the box.
2. atom (https://atom.io/) in combination with language-ansible (https://atom.io/packages/language-ansible).
3. atom (https://atom.io/) in combination with autocomplete-ansible (https://github.com/h-hirokawa/atom-autocomplete-ansible).

Automatic tests for Ansible code

1. ansible-lint (https://github.com/ansible/ansible-lint).
2. ansible-lint (https://github.com/ansible/ansible-lint) in combination with galaxy-lint-rules (https://github.com/ansible/galaxy-lint-rules).
3. molecule (https://molecule.readthedocs.io/en/latest/) for testing Ansible code. Most people use it to spin up Docker or Vagrant instances and apply their code, but it can even be used to only lint: `molecule lint`.