The**hosts:** line sets which host group (from the Inventory file 'hosts') to apply the following roles to.

The**roles:** line, and subsequent role entries define the roles to apply to that hostgroup.   The roles currently defined in parallax can be seen in the above tree structure.

You can either put multiple named blocks in one site.yml file, or split them up, in the manner of 'example\_servers.yml' and 'repository\_server.yml'

Other stuff in**'site.yml':**

**'user:'** - This sets the name of the user to connect to the target as.  Sometimes shown as remote\_user in newer ansible configurations.

**'sudo:'**- This tells Ansible whether it should run sudo on the target when it connects.  You'll probably want to set this as "sudo: yes" most often, unless you plan to connect as root.  In which case, this (ಠ.ಠ) is for you.

**Roles**

**=====**

A role should encapsulate all the things that have to happen to make a thing work.  If that sounds vague, it's because it is.

The parallax example has a role called common, which installs and configures the things that I've found are useful as prerequisites for other things.  You should go through and decide which bits you want to put into your 'common' role, if you decide to have one.

Roles can have dependencies, which will require that another role be applied first.  This is good for things like handling the dependencies before you deploy code.

**Inside A Role**

**-------------**

Let's take a look at one of the pre-defined roles in Parallax:

├── redis

│ ├── files

│ ├── handlers

│ ├── meta

│ ├── tasks

│ └── templates

This, unsurprisingly is a quick role I threw together that'll install Redis from an Ubuntu PPA, and start the service.

In general, a role consists of the following subdirectories, "files", "handlers", "meta", "tasks" and "templates".

**files/** contains files that will be copied to the target with the copy: module.

**handlers/** contains YAML files which contain 'handlers' little bits of config that can be triggered with the notify: action inside a task. Usually just handlers/main.yml - See <http://docs.ansible.com/playbooks_intro.html#handlers-running-operations-on-change> for more information on what handlers are for.

**meta/** contains YAML files containing role dependencies.  Usually just meta/main.yml

**tasks/** contains YAML files containing a list of named steps which Ansible will execute in order on a target.  Usually tasks/main.yml

**templates/** contains Jinja2 template files, which can be used in a task with the template: module to interpolate variables in the template, then copy the template to a location on the target.  Files in this directory often end .j2 by convention.

**Example Role: Redis**

**-------------------**

**Path:** parallax/playbooks/example/roles/redis

**Structure:**

.

├── files

├── handlers

├── meta

├── tasks

│ └── main.yml

└── templates

All there is in this one, is a task file, unsurprisingly called 'main.yml' - Told you that name would crop up again.   
- Actually, there's a .empty file under files, handlers, meta, and templates.  This is just so that if you commit it to git, the empty directories won't vanish.

Let's have a look at the redis role's tasks:

$ cat tasks/main.yml

---

- name: Add the Redis PPA

apt\_repository: repo='ppa:rwky/redis' update\_cache=yes

- name: Install Redis from PPA

apt: pkg=redis-server state=installed

- name: Start Redis

service: name=redis state=started

Each named block has an action below it.  Each action refers to an Ansible Module. There's an index of all available modules and their documentation here: <http://docs.ansible.com/list_of_all_modules.html>

Basically explained:

**apt\_repository:** module configures a new apt repository for the system.  It can take a ppa name, or a URL for a repository.  update\_cache tells ansible to run apt-get update after it's added the new repository.

**apt:** module tells Ansible to run apt-get install $pkg using  whatever value has been defined for pkg.

**service:** tells Ansible to execute "sudo service $name start" on the target.

I recommend you have a trawl through the roles as configured in Parallax, and see if you can make sense of how they work.  If you open the Ansible Module Index, you'll be able to use that as a quick reference guide for the modules in the roles.