Red Hat
Ansible Automation
Platform

ANSIBLE AUTOMATES

Practical Ansible Testing with Molecule

Jacob Hunt
Principal Technical Account Manager

Red Hat

I...

- Work at Red Hat
- Live in Idaho
- Am A Technical Account Manager

My journey with Ansible...

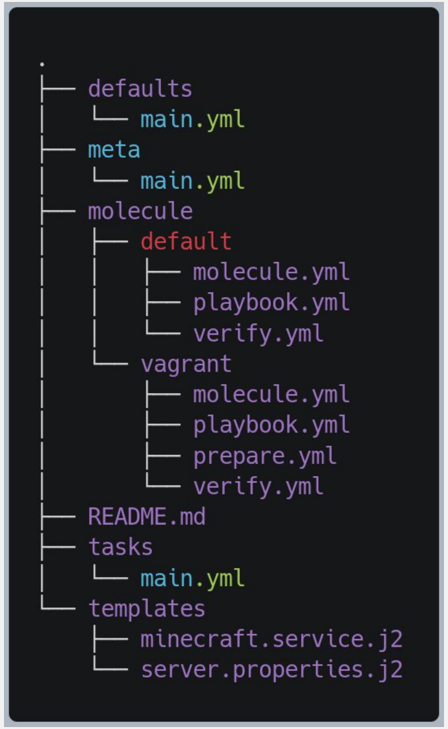
What’s Molecule?

- Testing tool for Ansible
- Spearheaded by John Dewey (@retroh)
- Donated to Red Hat by Cisco last year
- Allows you to easily spin up local `infrastructure` to test your roles/playbooks

What Problem(s) does Molecule Solve?

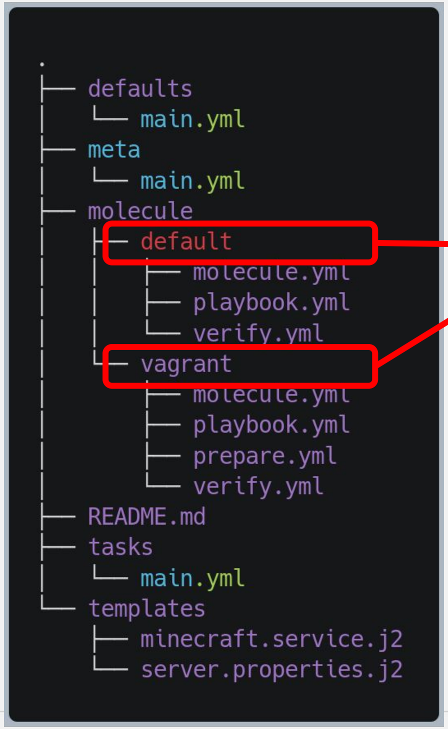
- Standardized testing infrastructure configuration
- Fast, iterative development loops
- All-in-one, batteries included validation for syntax, style, idempotence, correctness
- Trivially add CI for your Ansible, on any platform that lets you run a container

Project structure



```
.
├── defaults
│   └── main.yml
├── meta
│   └── main.yml
├── molecule
│   ├── default
│   │   ├── molecule.yml
│   │   ├── playbook.yml
│   │   └── verify.yml
│   └── vagrant
│       ├── molecule.yml
│       ├── playbook.yml
│       ├── prepare.yml
│       └── verify.yml
├── README.md
├── tasks
│   └── main.yml
└── templates
    ├── minecraft.service.j2
    └── server.properties.j2
```

Project structure

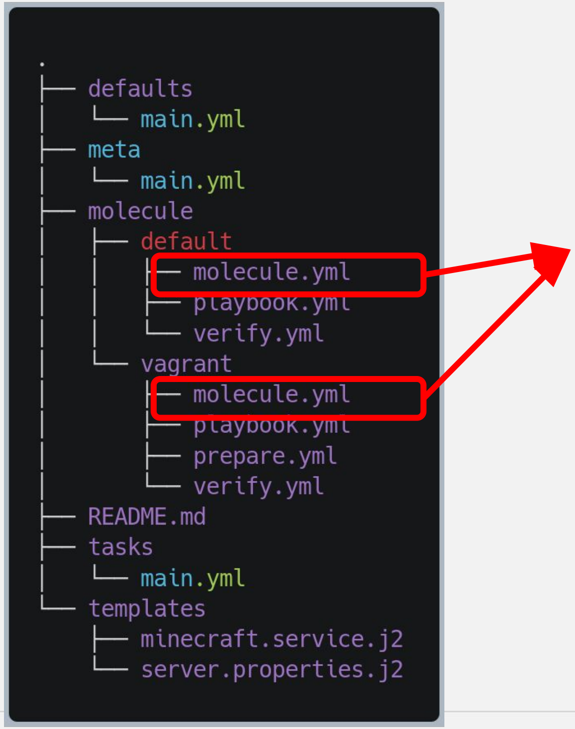


```
.
├── defaults
│   └── main.yml
├── meta
│   └── main.yml
├── molecule
│   ├── default
│   │   ├── molecule.yml
│   │   ├── playbook.yml
│   │   └── verify.yml
│   └── vagrant
│       ├── molecule.yml
│       ├── playbook.yml
│       ├── prepare.yml
│       └── verify.yml
├── README.md
├── tasks
│   └── main.yml
└── templates
    ├── minecraft.service.j2
    └── server.properties.j2
```

Molecule Scenarios

- Contains everything necessary to test the role in a certain way

Project structure

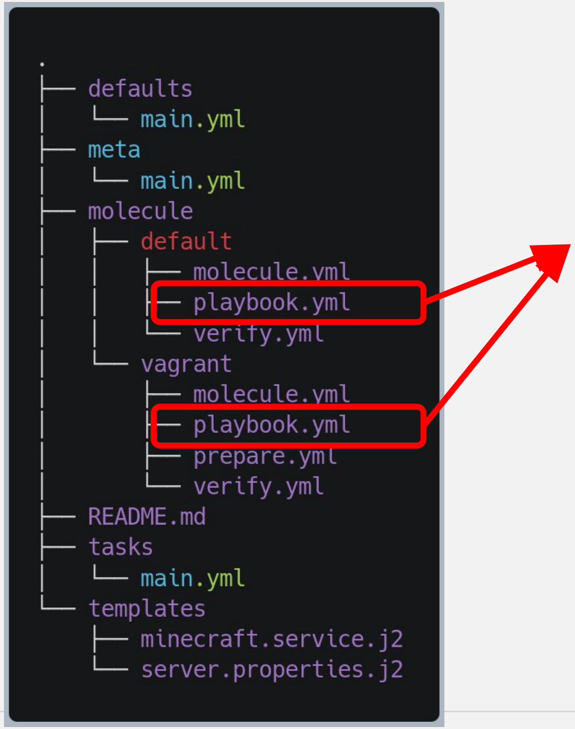


```
.
├── defaults
│   └── main.yml
├── meta
│   └── main.yml
├── molecule
│   ├── default
│   │   ├── molecule.yml
│   │   ├── playbook.yml
│   │   └── verify.yml
│   └── vagrant
│       ├── molecule.yml
│       ├── playbook.yml
│       ├── prepare.yml
│       └── verify.yml
├── README.md
├── tasks
│   └── main.yml
└── templates
    ├── minecraft.service.j2
    └── server.properties.j2
```

Molecule configuration file

- Contains the actual specific configuration for how to spin up infrastructure, and what to run against that infrastructure

Project structure

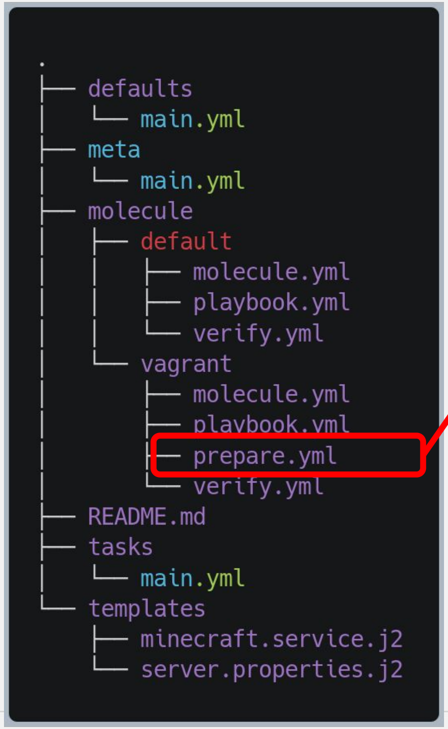


```
.
├── defaults
│   └── main.yml
├── meta
│   └── main.yml
├── molecule
│   ├── default
│   │   ├── molecule.yml
│   │   ├── playbook.yml
│   │   └── verify.yml
│   └── vagrant
│       ├── molecule.yml
│       ├── playbook.yml
│       ├── prepare.yml
│       └── verify.yml
├── README.md
├── tasks
│   └── main.yml
└── templates
    ├── minecraft.service.j2
    └── server.properties.j2
```

The playbook for running your role against the configured hosts

- Defaults to just a role import, but fully configurable

Project structure

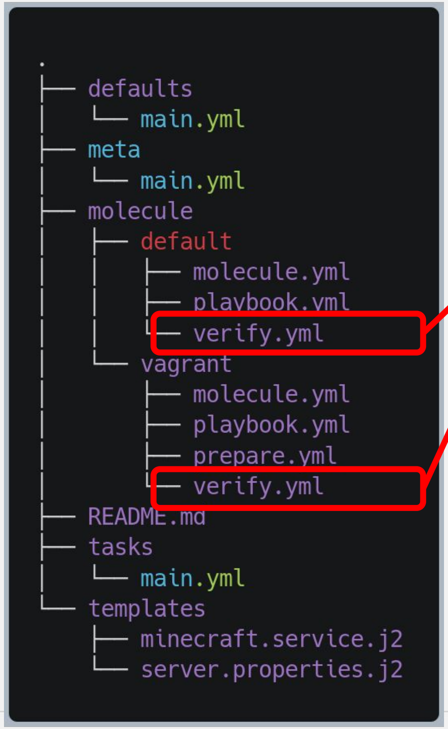


```
.
├── defaults
│   └── main.yml
├── meta
│   └── main.yml
├── molecule
│   ├── default
│   │   ├── molecule.yml
│   │   ├── playbook.yml
│   │   └── verify.yml
│   └── vagrant
│       ├── molecule.yml
│       ├── playbook.yml
│       ├── prepare.yml
│       └── verify.yml
├── README.md
├── tasks
│   └── main.yml
└── templates
    ├── minecraft.service.j2
    └── server.properties.j2
```

Runs before playbook.yml, for any one-time pre-configuration

- Optional

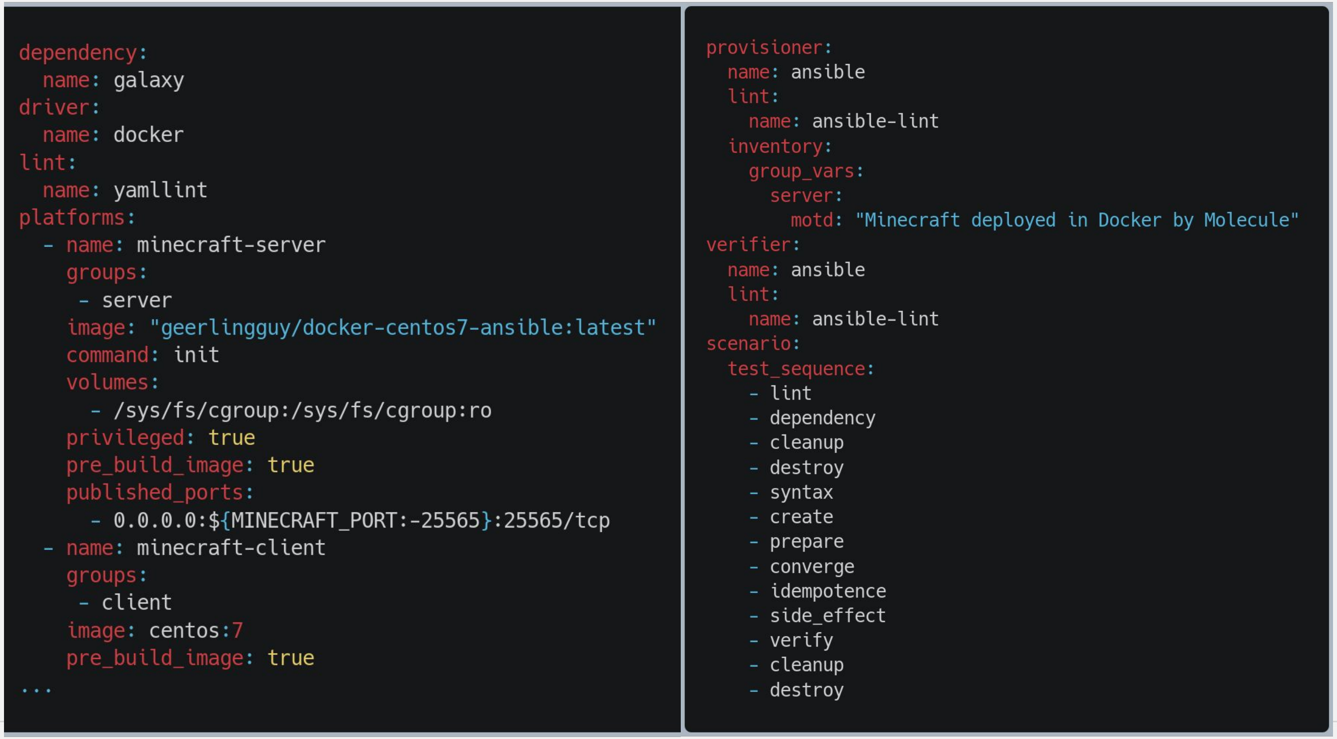
Project structure



```
.
├── defaults
│   └── main.yml
├── meta
│   └── main.yml
├── molecule
│   ├── default
│   │   ├── molecule.yml
│   │   ├── playbook.yml
│   │   └── verify.yml
│   └── vagrant
│       ├── molecule.yml
│       ├── playbook.yml
│       ├── prepare.yml
│       └── verify.yml
├── README.md
├── tasks
│   └── main.yml
└── templates
    ├── minecraft.service.j2
    └── server.properties.j2
```

The playbook for running your tests after playbook.yml has completed

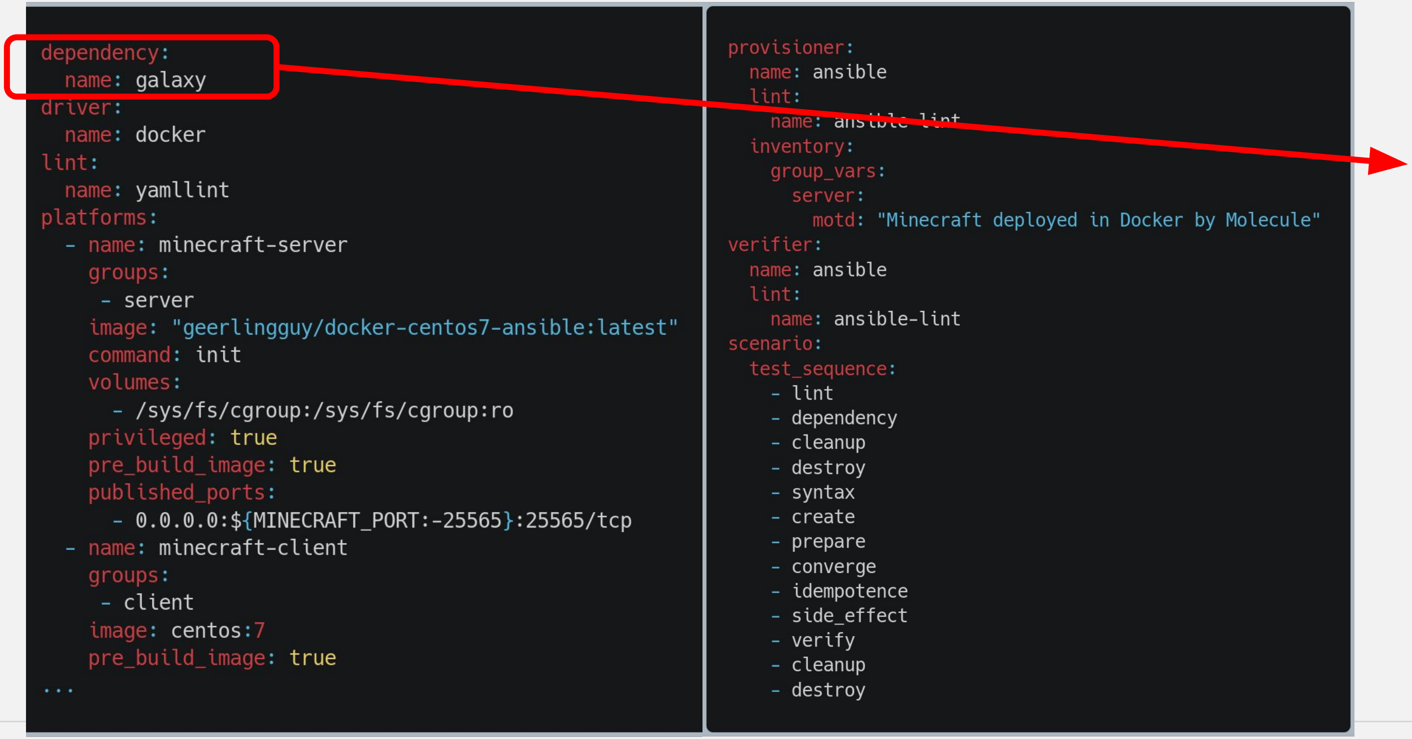
Anatomy of a molecule.yml



```yaml
dependency:
  name: galaxy
driver:
  name: docker
lint:
  name: yamllint
platforms:
  - name: minecraft-server
    groups:
     - server
    image: "geerlingguy/docker-centos7-ansible:latest"
    command: init
    volumes:
      - /sys/fs/cgroup:/sys/fs/cgroup:ro
    privileged: true
    pre_build_image: true
    published_ports:
      - 0.0.0.0:${MINECRAFT_PORT:-25565}:25565/tcp
  - name: minecraft-client
    groups:
     - client
    image: centos:7
    pre_build_image: true
...
```

```yaml
provisioner:
  name: ansible
  lint:
    name: ansible-lint
  inventory:
    group_vars:
      server:
        motd: "Minecraft deployed in Docker by Molecule"
verifier:
  name: ansible
  lint:
    name: ansible-lint
scenario:
  test_sequence:
    - lint
    - dependency
    - cleanup
    - destroy
    - syntax
    - create
    - prepare
    - converge
    - idempotence
    - side_effect
    - verify
    - cleanup
    - destroy
```

Anatomy of a molecule.yml



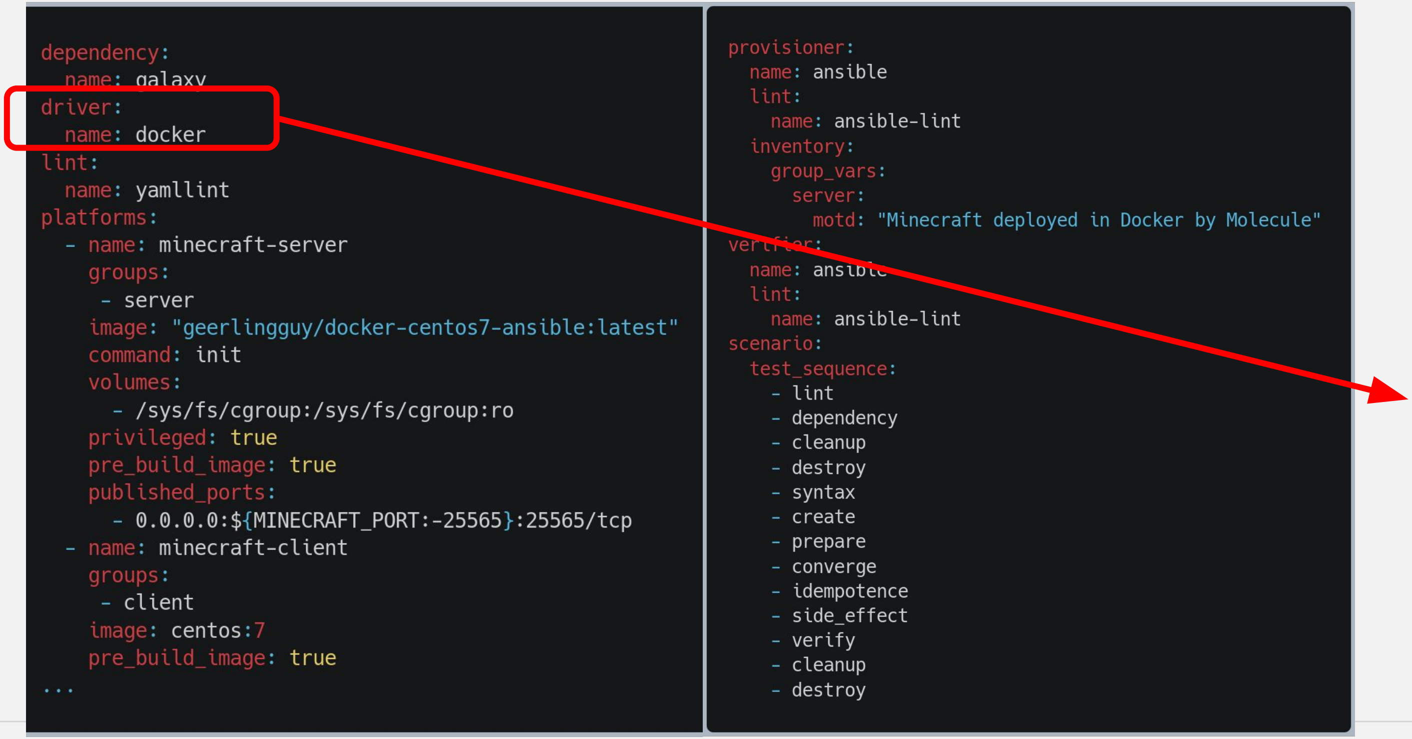
```yaml
dependency:
  name: galaxy
driver:
  name: docker
lint:
  name: yamllint
platforms:
  - name: minecraft-server
    groups:
     - server
    image: "geerlingguy/docker-centos7-ansible:latest"
    command: init
    volumes:
      - /sys/fs/cgroup:/sys/fs/cgroup:ro
    privileged: true
    pre_build_image: true
    published_ports:
      - 0.0.0.0:${MINECRAFT_PORT:-25565}:25565/tcp
  - name: minecraft-client
    groups:
     - client
    image: centos:7
    pre_build_image: true
...
```

```yaml
provisioner:
  name: ansible
  lint:
    name: ansible-lint
  inventory:
    group_vars:
      server:
        motd: "Minecraft deployed in Docker by Molecule"
verifier:
  name: ansible
  lint:
    name: ansible-lint
scenario:
  test_sequence:
    - lint
    - dependency
    - cleanup
    - destroy
    - syntax
    - create
    - prepare
    - converge
    - idempotence
    - side_effect
    - verify
    - cleanup
    - destroy
```

Install dependencies

- Galaxy
- Gilt
- Shell

Anatomy of a molecule.yml



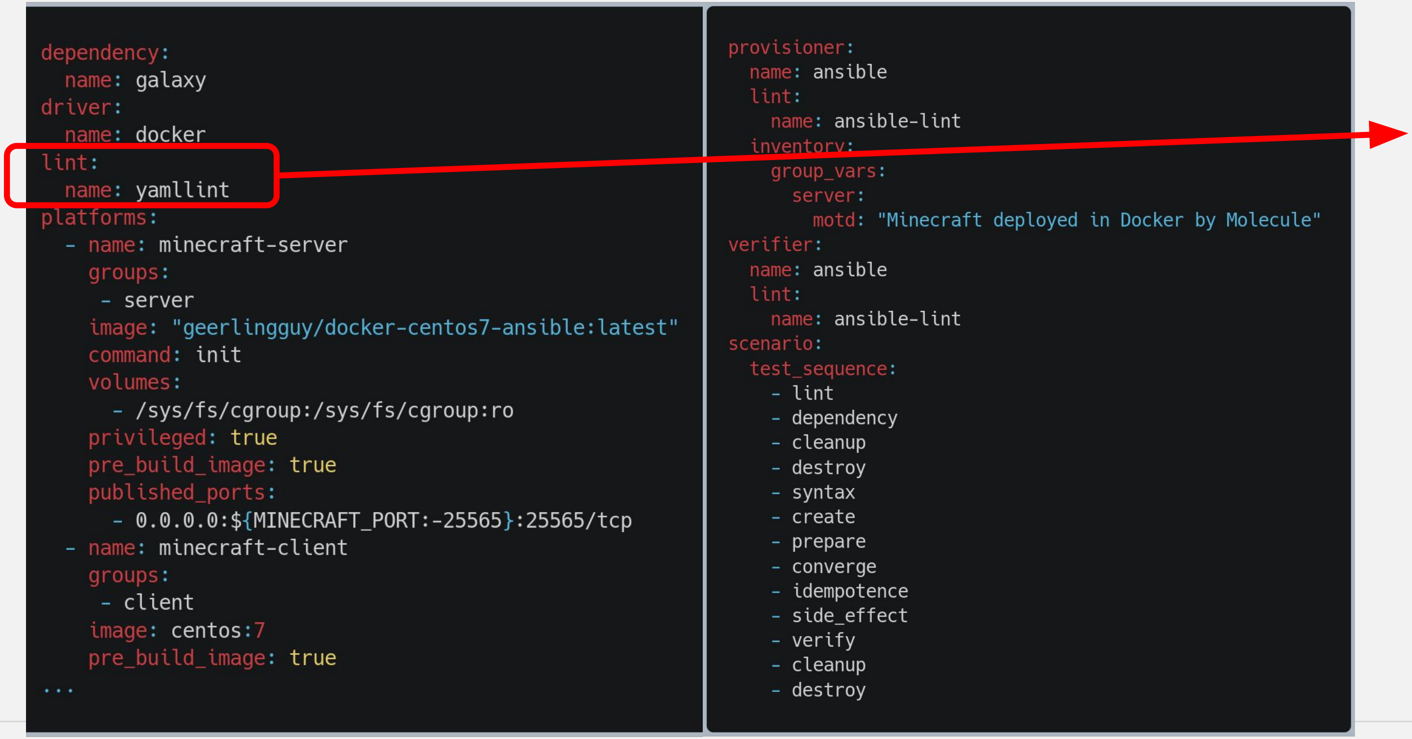
```yaml
dependency:
  name: galaxy
driver:
  name: docker
lint:
  name: yamllint
platforms:
  - name: minecraft-server
    groups:
     - server
    image: "geerlingguy/docker-centos7-ansible:latest"
    command: init
    volumes:
      - /sys/fs/cgroup:/sys/fs/cgroup:ro
    privileged: true
    pre_build_image: true
    published_ports:
      - 0.0.0.0:${MINECRAFT_PORT:-25565}:25565/tcp
  - name: minecraft-client
    groups:
     - client
    image: centos:7
    pre_build_image: true
...
```

```yaml
provisioner:
  name: ansible
  lint:
    name: ansible-lint
  inventory:
    group_vars:
      server:
        motd: "Minecraft deployed in Docker by Molecule"
verifier:
  name: ansible
  lint:
    name: ansible-lint
scenario:
  test_sequence:
    - lint
    - dependency
    - cleanup
    - destroy
    - syntax
    - create
    - prepare
    - converge
    - idempotence
    - side_effect
    - verify
    - cleanup
    - destroy
```

Specifies the backend molecule will use to provide instances

- Azure
- Delegated
- DigitalOcean
- Docker
- EC2
- GCE
- Hetzner Cloud
- Linode
- LXC
- LXD
- Openstack
- Podman
- Vagrant

Anatomy of a molecule.yml



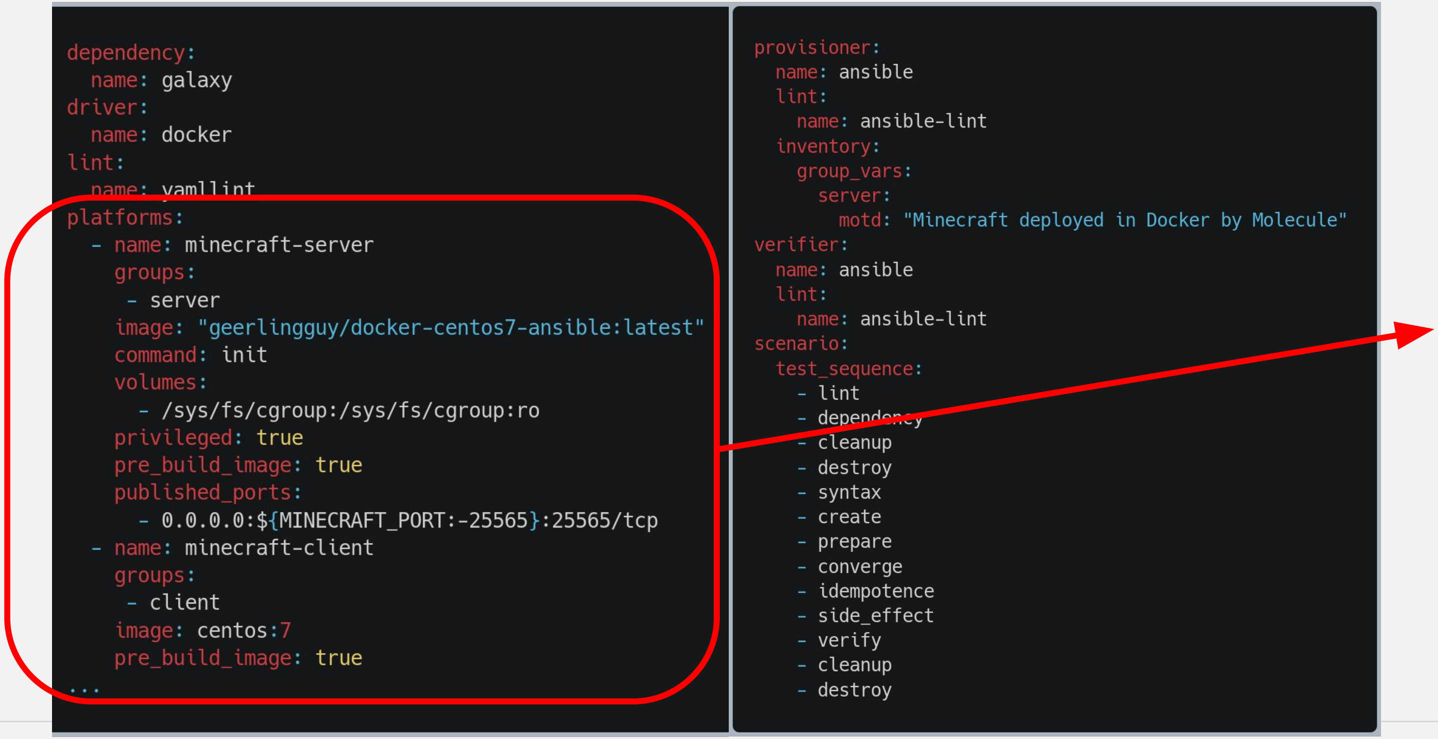
```yaml
dependency:
  name: galaxy
driver:
  name: docker
lint:
  name: yamllint
platforms:
  - name: minecraft-server
    groups:
     - server
    image: "geerlingguy/docker-centos7-ansible:latest"
    command: init
    volumes:
      - /sys/fs/cgroup:/sys/fs/cgroup:ro
    privileged: true
    pre_build_image: true
    published_ports:
      - 0.0.0.0:${MINECRAFT_PORT:-25565}:25565/tcp
  - name: minecraft-client
    groups:
     - client
    image: centos:7
    pre_build_image: true
...
```

```yaml
provisioner:
  name: ansible
  lint:
    name: ansible-lint
  inventory:
    group_vars:
      server:
        motd: "Minecraft deployed in Docker by Molecule"
verifier:
  name: ansible
  lint:
    name: ansible-lint
scenario:
  test_sequence:
    - lint
    - dependency
    - cleanup
    - destroy
    - syntax
    - create
    - prepare
    - converge
    - idempotence
    - side_effect
    - verify
    - cleanup
    - destroy
```

Configure linting

- Allows override of default linting rules or addition of new ones

Anatomy of a molecule.yml



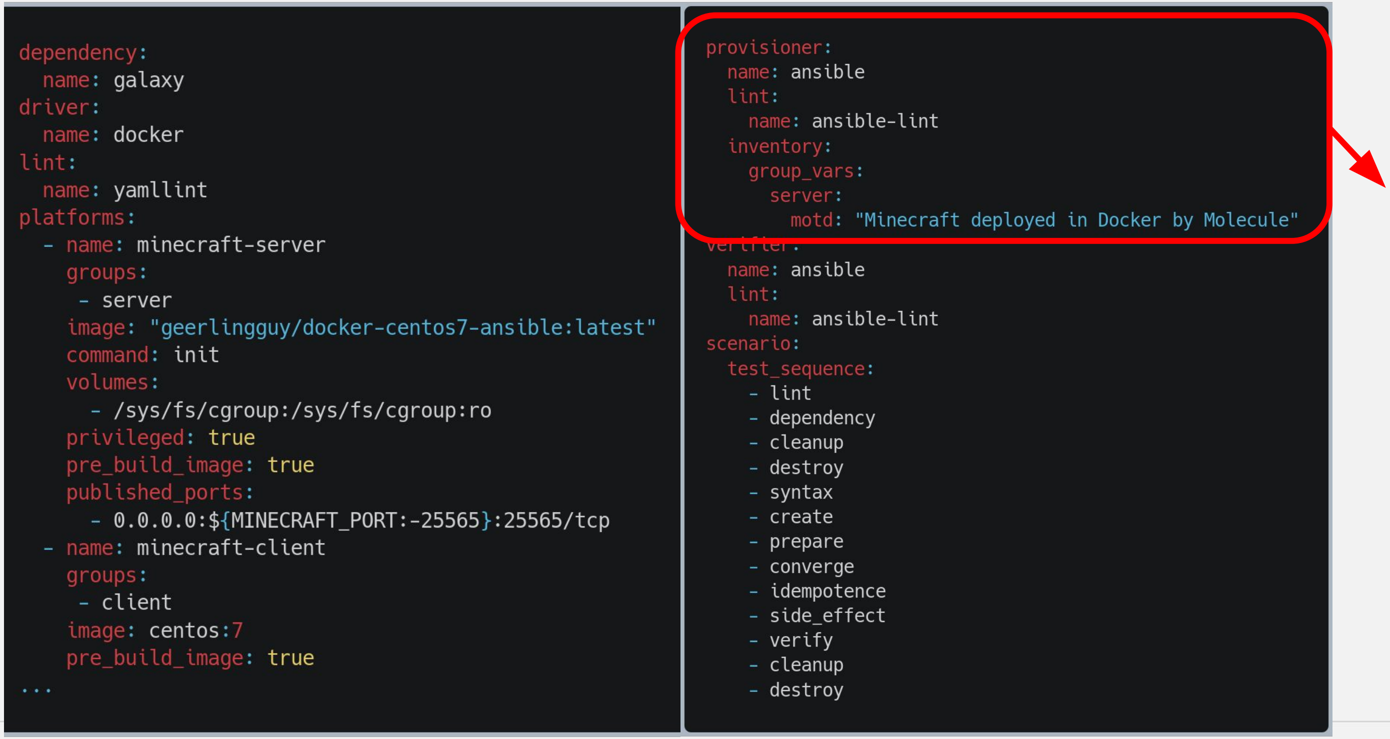
```
dependency:
  name: galaxy
driver:
  name: docker
lint:
  name: yamllint
platforms:
  - name: minecraft-server
    groups:
     - server
    image: "geerlingguy/docker-centos7-ansible:latest"
    command: init
    volumes:
      - /sys/fs/cgroup:/sys/fs/cgroup:ro
    privileged: true
    pre_build_image: true
    published_ports:
      - 0.0.0.0:${MINECRAFT_PORT:-25565}:25565/tcp
  - name: minecraft-client
    groups:
     - client
    image: centos:7
    pre_build_image: true
...
```

```
provisioner:
  name: ansible
  lint:
    name: ansible-lint
  inventory:
    group_vars:
      server:
        motd: "Minecraft deployed in Docker by Molecule"
verifier:
  name: ansible
  lint:
    name: ansible-lint
scenario:
  test_sequence:
    - lint
    - dependency
    - cleanup
    - destroy
    - syntax
    - create
    - prepare
    - converge
    - idempotence
    - side_effect
    - verify
    - cleanup
    - destroy
```

Driver specific platform configuration

- Defines instances for molecule to manage
- Uses Ansible modules to handle interactions
- Populates your hosts

Anatomy of a molecule.yml



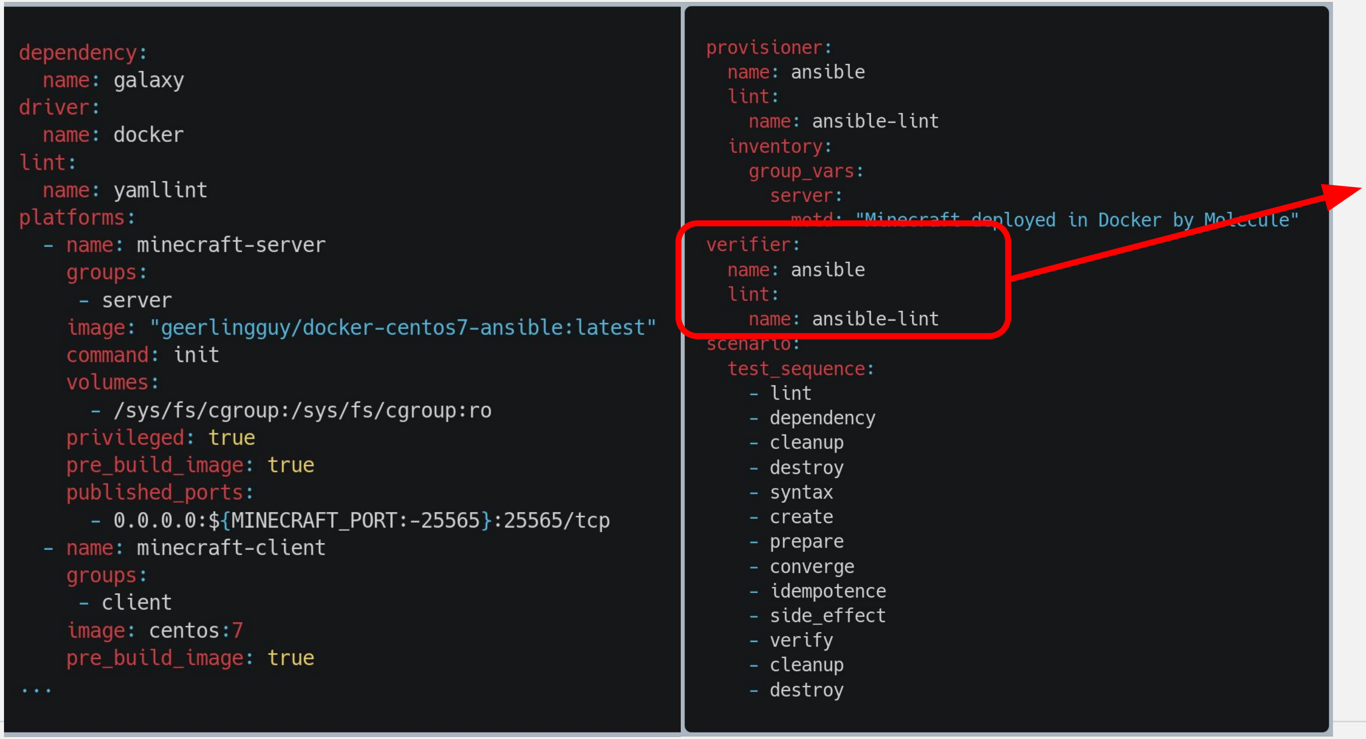
```yaml
dependency:
  name: galaxy
driver:
  name: docker
lint:
  name: yamllint
platforms:
  - name: minecraft-server
    groups:
     - server
    image: "geerlingguy/docker-centos7-ansible:latest"
    command: init
    volumes:
      - /sys/fs/cgroup:/sys/fs/cgroup:ro
    privileged: true
    pre_build_image: true
    published_ports:
      - 0.0.0.0:${MINECRAFT_PORT:-25565}:25565/tcp
  - name: minecraft-client
    groups:
     - client
    image: centos:7
    pre_build_image: true
...
```

```yaml
provisioner:
  name: ansible
  lint:
    name: ansible-lint
  inventory:
    group_vars:
      server:
        motd: "Minecraft deployed in Docker by Molecule"
verifier:
  name: ansible
  lint:
    name: ansible-lint
scenario:
  test_sequence:
    - lint
    - dependency
    - cleanup
    - destroy
    - syntax
    - create
    - prepare
    - converge
    - idempotence
    - side_effect
    - verify
    - cleanup
    - destroy
```

Configure Ansible playbook runs

- Override playbooks
- Add group/host vars
- Set any options to be passed to ansible-playbook

Anatomy of a molecule.yml



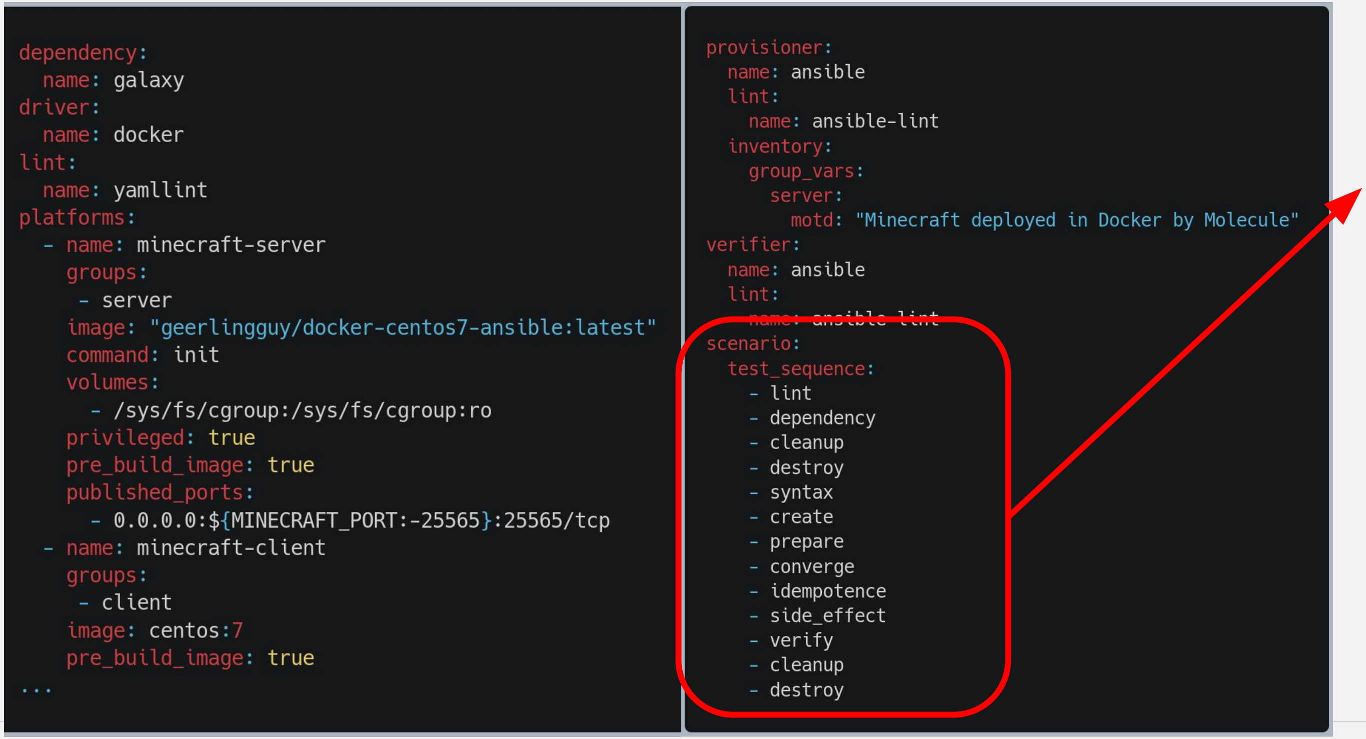
```
dependency:
  name: galaxy
driver:
  name: docker
lint:
  name: yamllint
platforms:
  - name: minecraft-server
    groups:
     - server
    image: "geerlingguy/docker-centos7-ansible:latest"
    command: init
    volumes:
      - /sys/fs/cgroup:/sys/fs/cgroup:ro
    privileged: true
    pre_build_image: true
    published_ports:
      - 0.0.0.0:${MINECRAFT_PORT:-25565}:25565/tcp
  - name: minecraft-client
    groups:
     - client
    image: centos:7
    pre_build_image: true
...
```

```
provisioner:
  name: ansible
  lint:
    name: ansible-lint
  inventory:
    group_vars:
      server:
        motd: "Minecraft deployed in Docker by Molecule"
verifier:
  name: ansible
  lint:
    name: ansible-lint
scenario:
  test_sequence:
    - lint
    - dependency
    - cleanup
    - destroy
    - syntax
    - create
    - prepare
    - converge
    - idempotence
    - side_effect
    - verify
    - cleanup
    - destroy
```

Configure what to run for tests

- Ansible (I added this!)
- Goss
- Inspec
- Testinfra

Anatomy of a molecule.yml



```
dependency:
  name: galaxy
driver:
  name: docker
lint:
  name: yamllint
platforms:
  - name: minecraft-server
    groups:
     - server
    image: "geerlingguy/docker-centos7-ansible:latest"
    command: init
    volumes:
      - /sys/fs/cgroup:/sys/fs/cgroup:ro
    privileged: true
    pre_build_image: true
    published_ports:
      - 0.0.0.0:${MINECRAFT_PORT:-25565}:25565/tcp
  - name: minecraft-client
    groups:
     - client
    image: centos:7
    pre_build_image: true
...
```

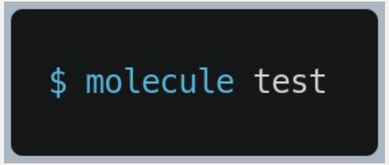
```
provisioner:
  name: ansible
  lint:
    name: ansible-lint
  inventory:
    group_vars:
      server:
        motd: "Minecraft deployed in Docker by Molecule"
verifier:
  name: ansible
  lint:
    name: ansible-lint
scenario:
  test_sequence:
    - lint
    - dependency
    - cleanup
    - destroy
    - syntax
    - create
    - prepare
    - converge
    - idempotence
    - side_effect
    - verify
    - cleanup
    - destroy
```

Configure how the scenario runs

- Allows addition, removal or reordering of steps on a per-command basis

Testing Loop

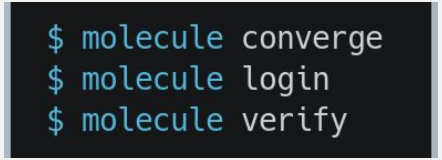
For CI/testing:



```
$ molecule test
```

lint / dependency / cleanup / destroy / syntax / create / prepare / converge / idempotence / side_effect / verify / cleanup / destroy

For development:



```
$ molecule converge
$ molecule login
$ molecule verify
```

dependency / create / prepare / converge
verify

Demo time (...kinda)

https://github.com/fabianvf/practical-testing-with-molecule

ANSIBLE AUTOMATES

THANK YOU

youtube.com/AnsibleAutomation

facebook.com/ansibleautomation

linkedin.com/company/Red-Hat

twitter.com/ansible