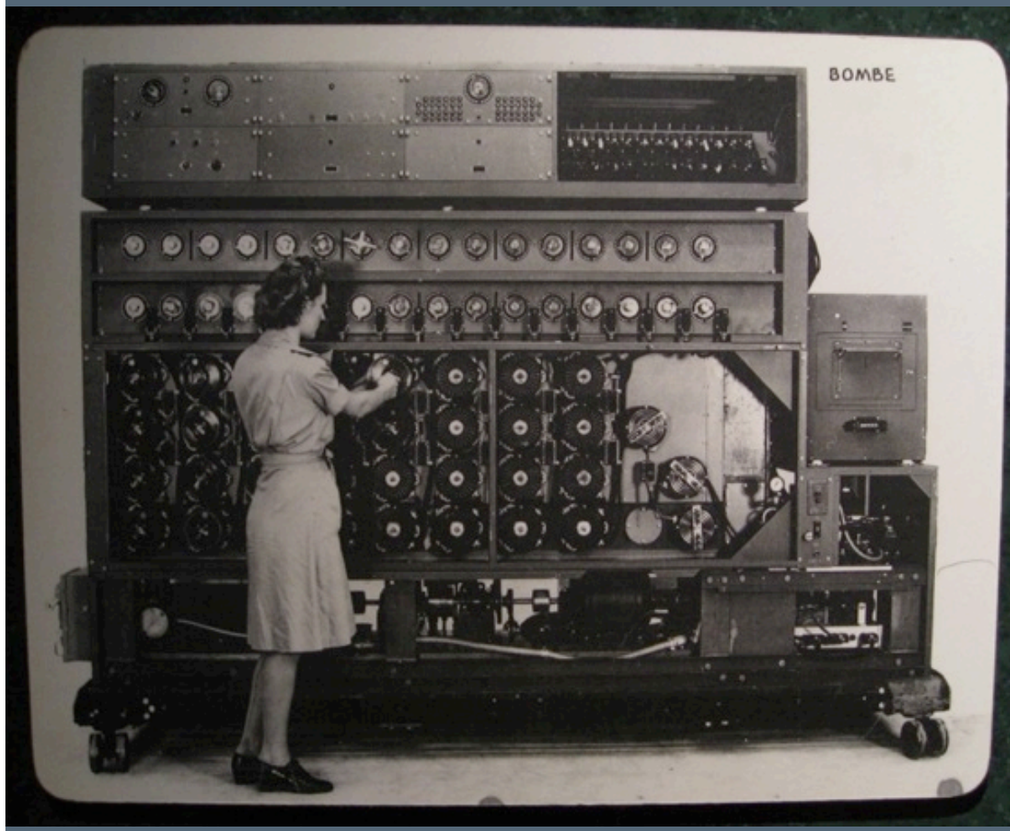
Infrastructure as Code

Infrastructure as Code

- Programmatically provision and configure components
- Treat like any other code base
- Reconstruct business from code repository, data backup, and compute resources



Infrastructure as Code

BUILD

- Develop reusable Cookbooks
- Expose tunable settings
- Test locally to reduce risk, and ensure compliance

DEPLOY

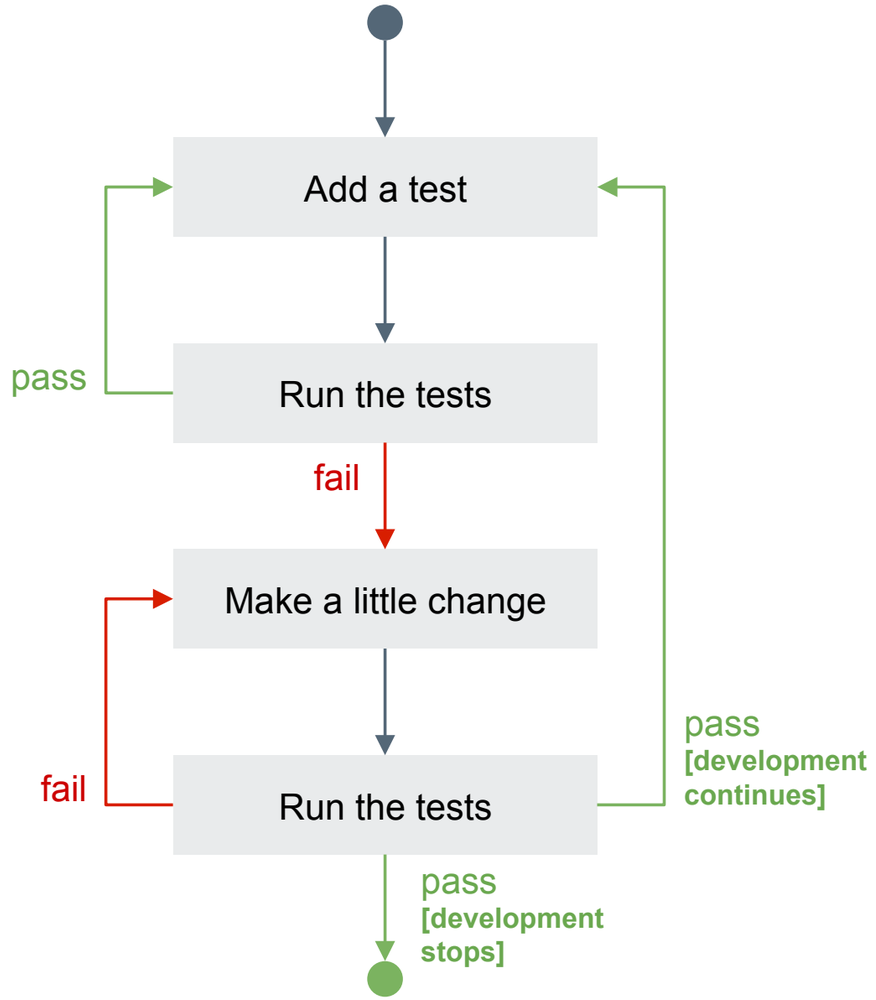
- Commit to Source Code
- Automated Testing through Continuous Integration
- Automatically promote across environments

MANAGE

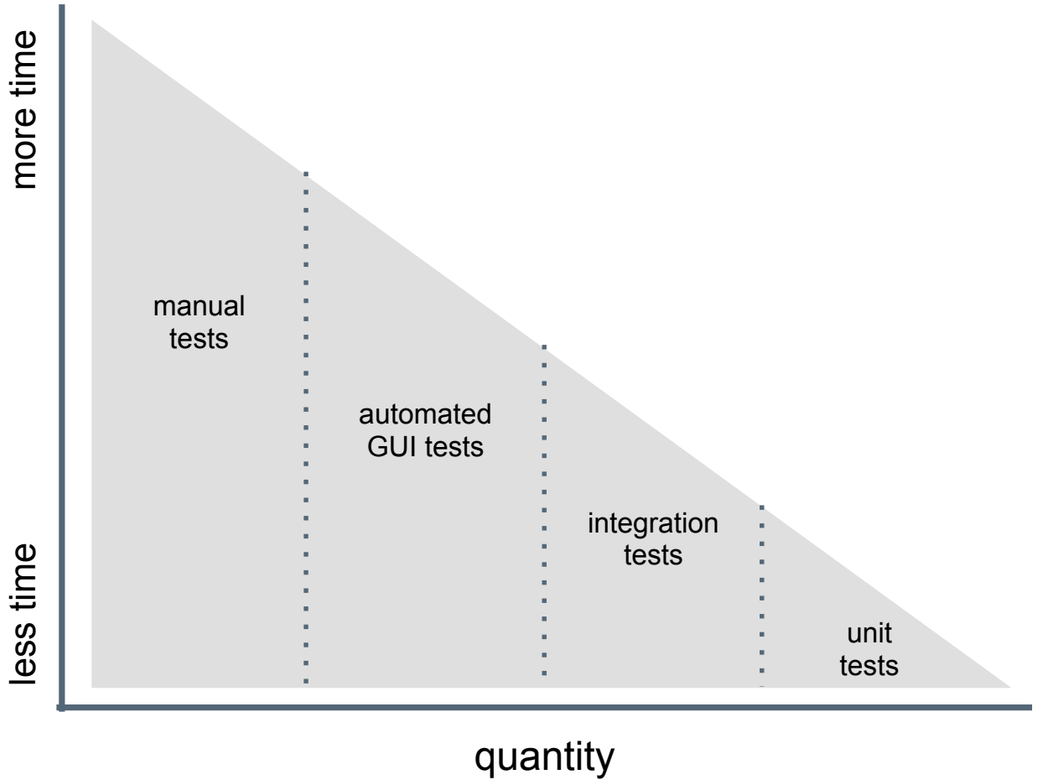
- Easily deploy new configurations in a matter of minutes.
- Continuously verify and repair misconfigured systems

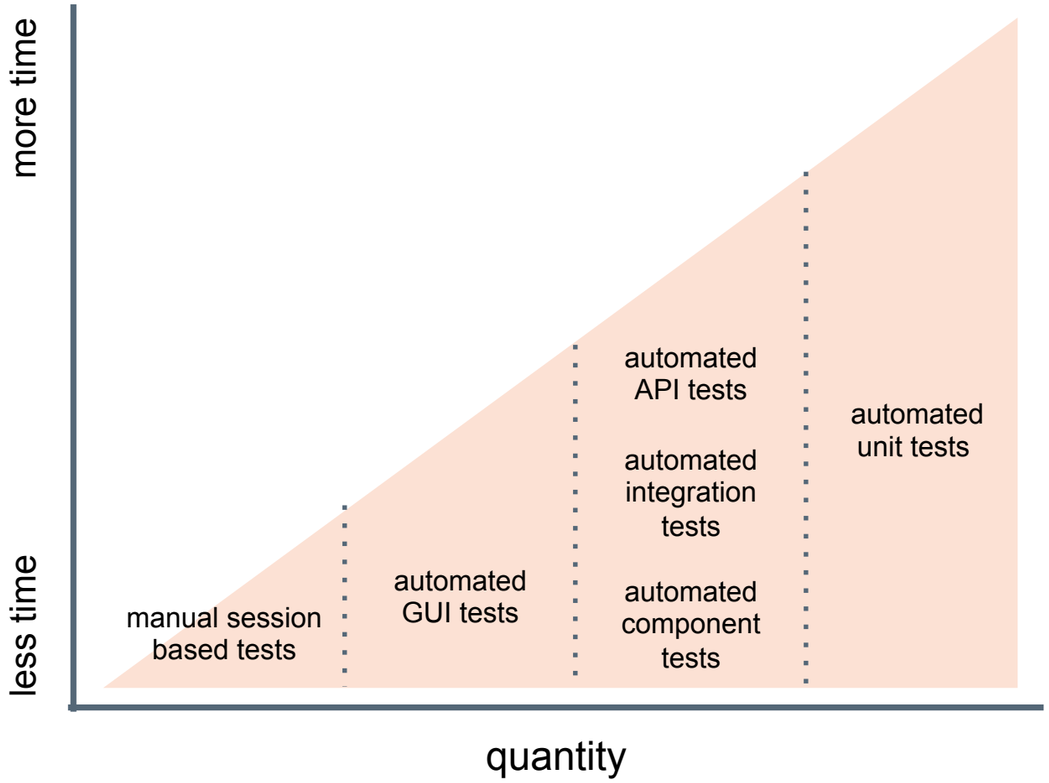
Test-driven Development

- Write a unit test, watch it fail
- Write some code
- Write and run more unit tests
- Run some integration/acceptance tests
- Code review
- Delivery pipeline to production
- Lowered chance of production failure



Software Testing and Why it Matters





Testing **builds safety** through feedback loops

Inexpensive experiments to provide validation

Reduces risk

Optimize Testing: Do more of the inexpensive testing first!

Remember…

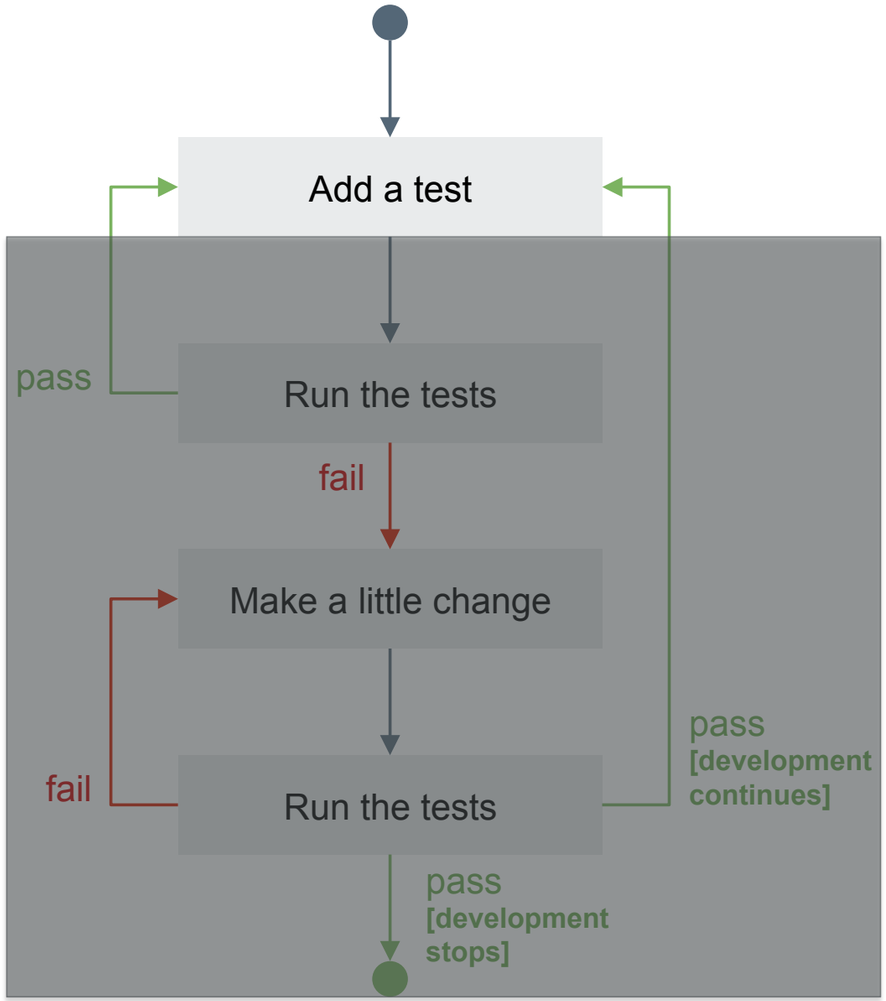
Infrastructure policies need testing

- ↳ Linting
- ↳ Static Analysis
- ↳ Unit Testing
- ↳ Integration Testing
- ↳ Compliance Testing

“Infrastructure as Code” should be tested like ANY other codebase.

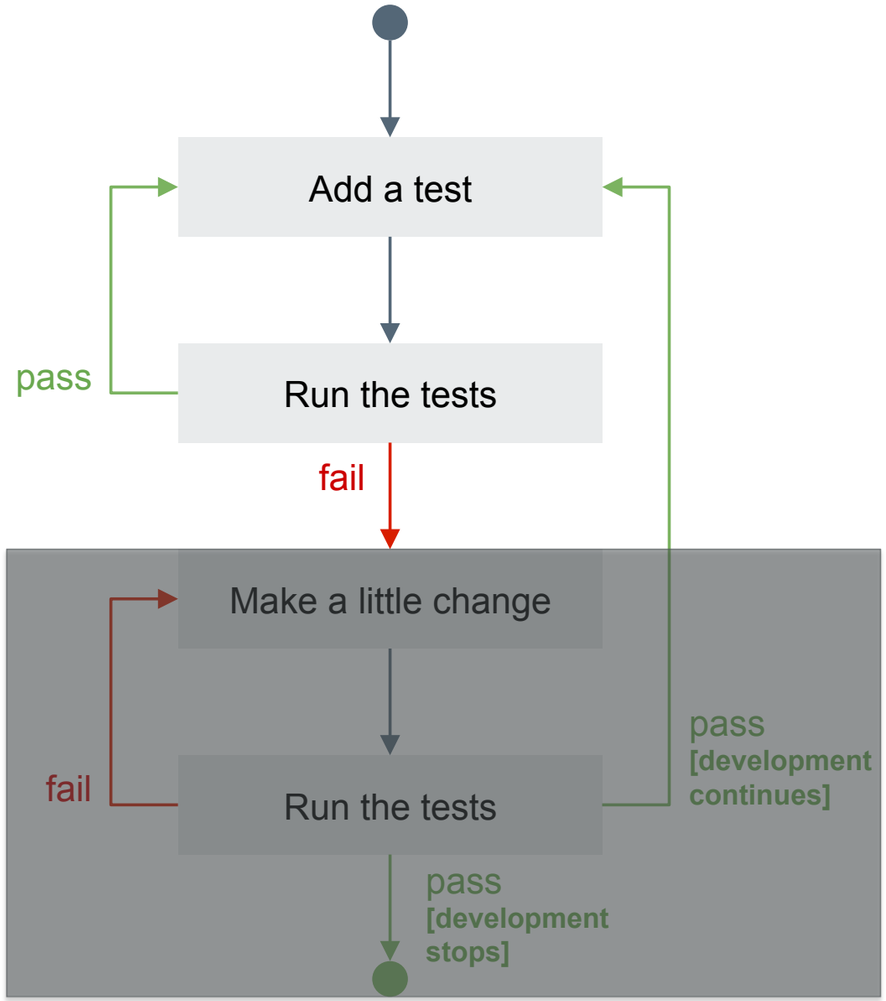
Integration Testing – Add tests

```
describe package 'httpd' do
  it { should be_installed }
end

describe service 'httpd' do
  it { should be_running }
  it { should be_enabled }
end

describe port(80) do
  it { should be_listening }
end
```



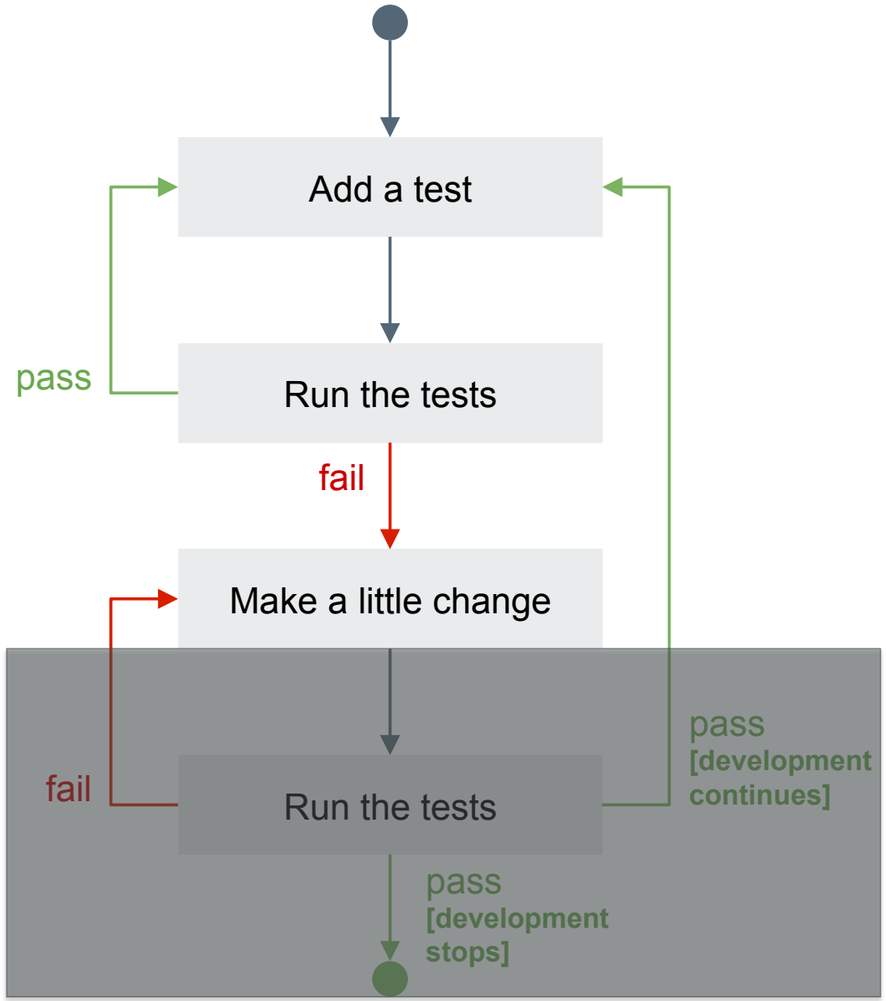
Integration Testing - Run the tests

```
-----> Verifying <default-centos-72>...
       Loaded

Target:  ssh://vagrant@127.0.0.1:2222


  System Package
     ø  httpd should be installed
     expected that `System Package httpd` is installed
  Service httpd
     ø  should be running
     expected that `Service httpd` is running
     ø  should be enabled
     expected that `Service httpd` is enabled
  Port 80
     ø  should be listening
     expected `Port 80.listening?` to return true, got false

Test Summary: 0 successful, 4 failures, 0 skipped
```



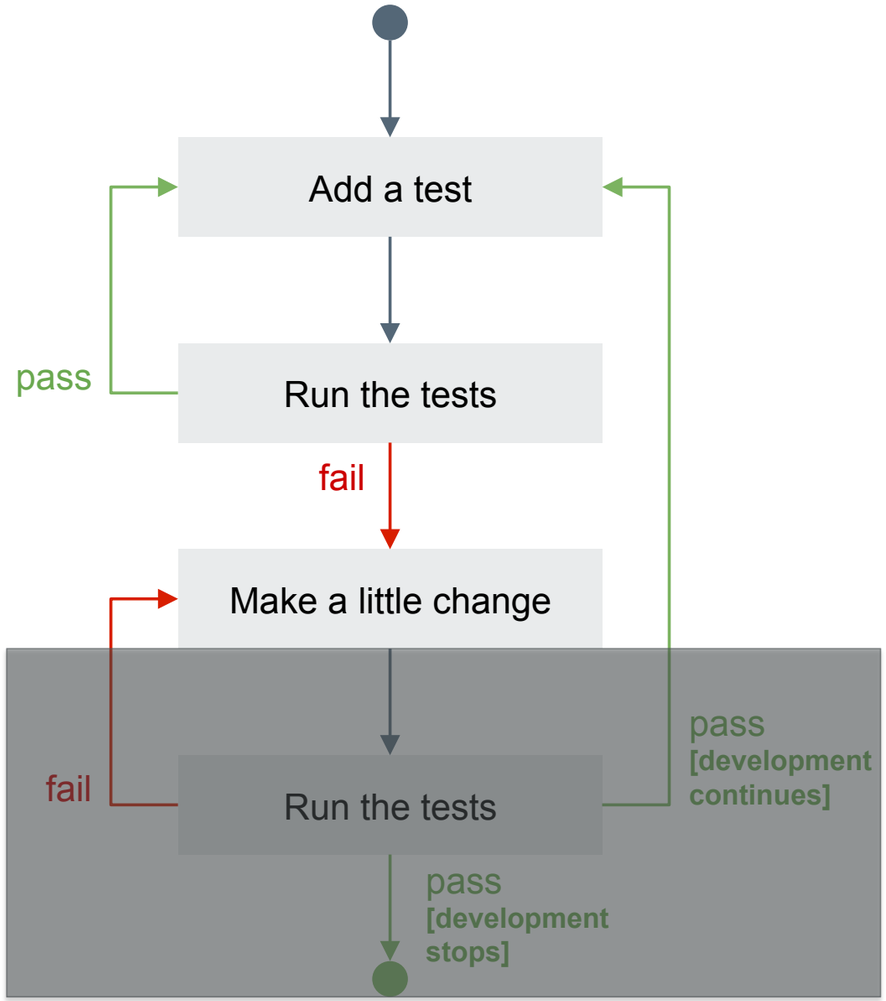
Integration Testing – Make a change

```
package 'httpd' do
  action :install
end

service 'httpd' do
  action [ :start, :enable ]
end
```



Integration Testing – Apply the change

```
       Synchronizing Cookbooks:
         - httpd (0.1.0)
       Installing Cookbook Gems:
       Compiling Cookbooks...
       Converging 2 resources
       Recipe: httpd::default
         * yum_package[httpd] action install
           - install version 2.4.6-45.el7.centos of package httpd
         * service[httpd] action start
           - start service service[httpd]
         * service[httpd] action enable
           - enable service service[httpd]

       Running handlers:
       Running handlers complete
       Chef Client finished, 3/3 resources updated in 10 seconds
       Finished converging <default-centos-72> (0m15.73s).
-----> Kitchen is finished. (0m18.13s)
```



Integration Testing – Run the tests

```
-----> Verifying <default-centos-72>...
       Loaded

Target:  ssh://vagrant@127.0.0.1:2222


  System Package
     ✔  httpd should be installed
  Service httpd
     ✔  should be running
     ✔  should be enabled
  Port 80
     ✔  should be listening

Test Summary: 4 successful, 0 failures, 0 skipped
       Finished verifying <default-centos-72> (0m0.91s).
```