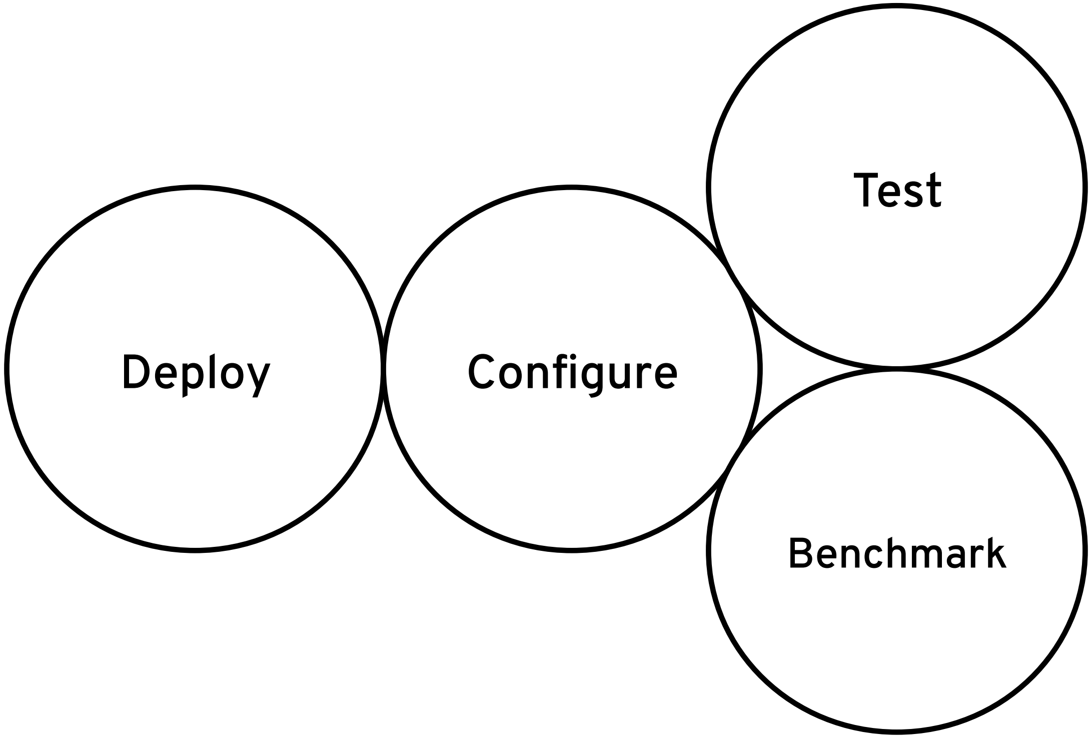
Testing and Benchmarking



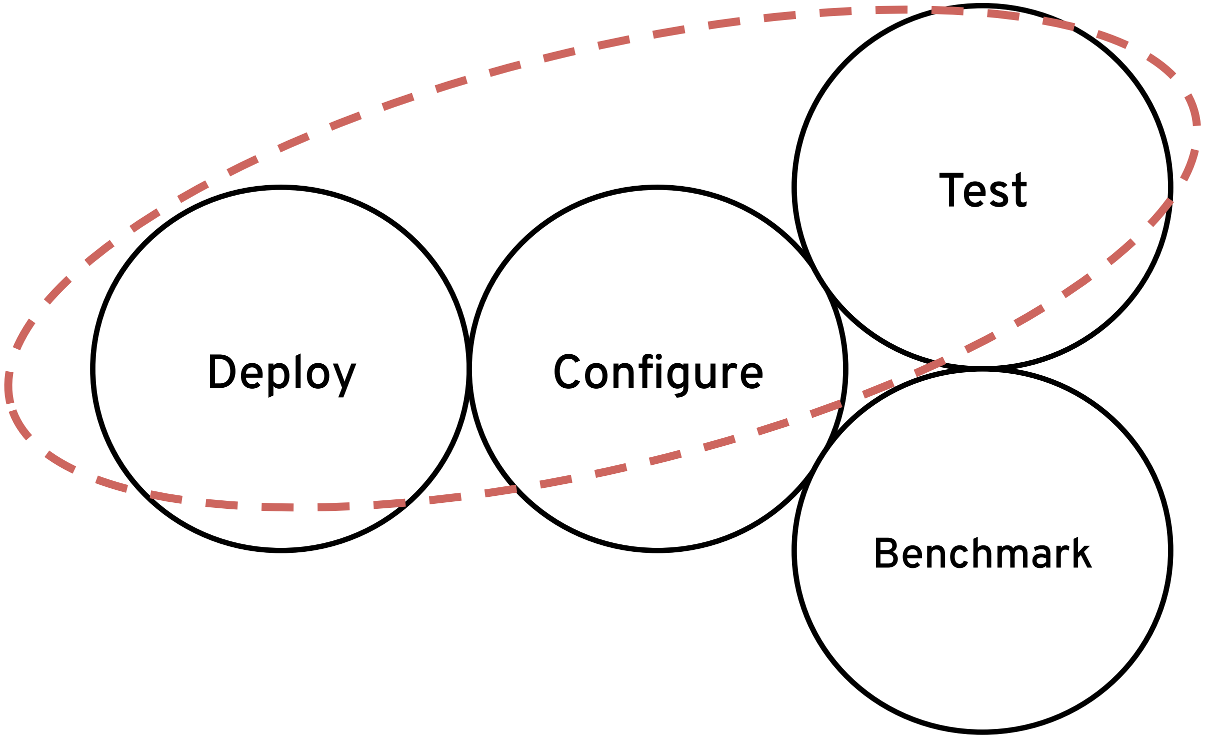
Deploy

Configure

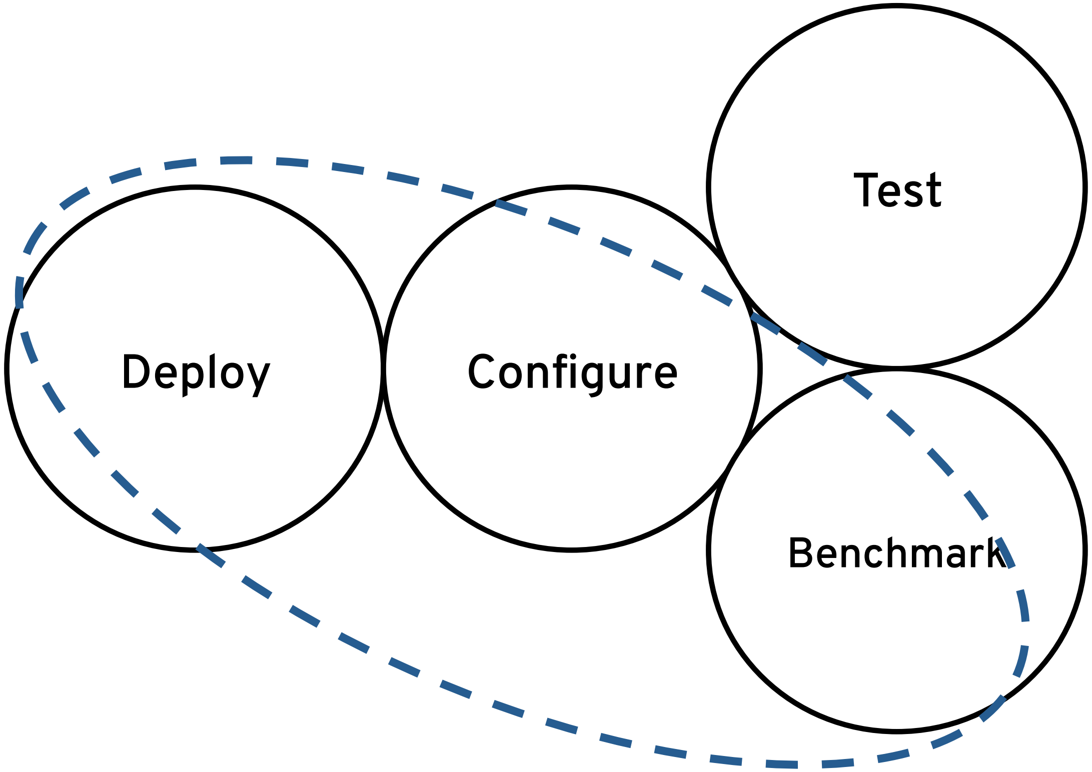
Test

Benchmark

Integration Tests



Benchmarking



Non-Functional (NF) Tests

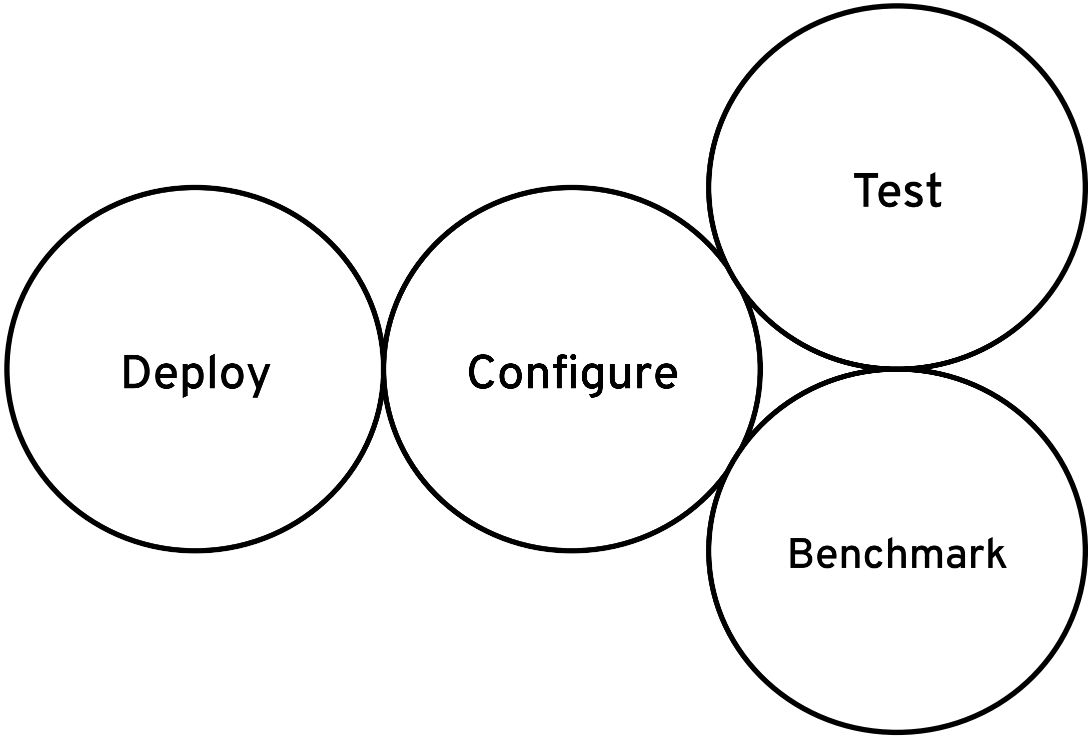
- Wikipedia:

  *“a **non-functional requirement** is one that specifies criteria that can be used to judge the qualities of a system, rather than specific behaviors”*

- For example:
  - scalability, performance, reliability, etc.
- Difficulty: qualities are subjective
  - Need to figure how to quantify them so we can objectively define validation criteria

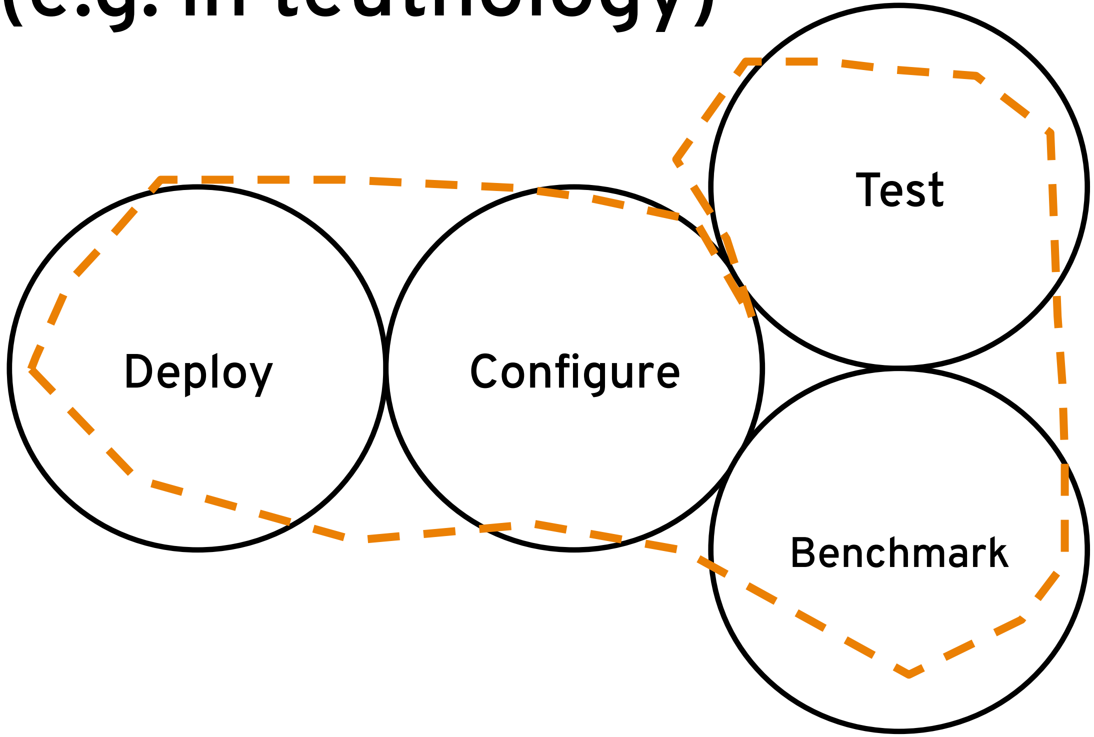
One Possible Approach

- Gather performance metrics
- Test assertions over measurement data



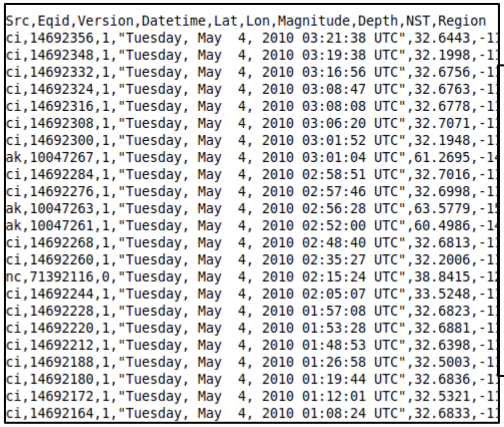
One Possible Approach

- Define tests over benchmark output data
- Validate these tests in the same way that QA is done (e.g. in teuthology)



Challenges

1. Hardware non-determinism
   – Docker (cgroups' blkio/net_cls subsystems)
2. Need a way to specify tests
   – validation language (aver project @ UCSC)



```
Src,Eqid,Version,Datetime,Lat,Lon,Magnitude,Depth,NST,Region
ci,14692356,1,"Tuesday, May  4, 2010 03:21:38 UTC",32.6443,-11
ci,14692348,1,"Tuesday, May  4, 2010 03:19:38 UTC",32.1998,-11
ci,14692332,1,"Tuesday, May  4, 2010 03:16:56 UTC",32.6756,-11
ci,14692324,1,"Tuesday, May  4, 2010 03:08:47 UTC",32.6763,-11
ci,14692316,1,"Tuesday, May  4, 2010 03:08:08 UTC",32.6778,-11
ci,14692308,1,"Tuesday, May  4, 2010 03:06:20 UTC",32.7071,-11
ci,14692300,1,"Tuesday, May  4, 2010 03:01:52 UTC",32.1948,-11
ak,10047267,1,"Tuesday, May  4, 2010 03:01:04 UTC",61.2695,-14
ci,14692284,1,"Tuesday, May  4, 2010 02:58:51 UTC",32.7016,-11
ci,14692276,1,"Tuesday, May  4, 2010 02:57:46 UTC",32.6998,-11
ak,10047263,1,"Tuesday, May  4, 2010 02:56:28 UTC",63.5779,-15
ak,10047261,1,"Tuesday, May  4, 2010 02:52:00 UTC",60.4986,-14
ci,14692268,1,"Tuesday, May  4, 2010 02:48:40 UTC",32.6813,-11
ci,14692260,1,"Tuesday, May  4, 2010 02:35:27 UTC",32.2006,-11
nc,71392116,0,"Tuesday, May  4, 2010 02:15:24 UTC",38.8415,-12
ci,14692244,1,"Tuesday, May  4, 2010 02:05:07 UTC",33.5248,-11
ci,14692228,1,"Tuesday, May  4, 2010 01:57:08 UTC",32.6823,-11
ci,14692220,1,"Tuesday, May  4, 2010 01:53:28 UTC",32.6881,-11
ci,14692212,1,"Tuesday, May  4, 2010 01:48:53 UTC",32.6398,-11
ci,14692188,1,"Tuesday, May  4, 2010 01:26:58 UTC",32.5003,-11
ci,14692180,1,"Tuesday, May  4, 2010 01:19:44 UTC",32.6836,-11
ci,14692172,1,"Tuesday, May  4, 2010 01:12:01 UTC",32.5321,-11
ci,14692164,1,"Tuesday, May  4, 2010 01:08:24 UTC",32.6833,-11
```

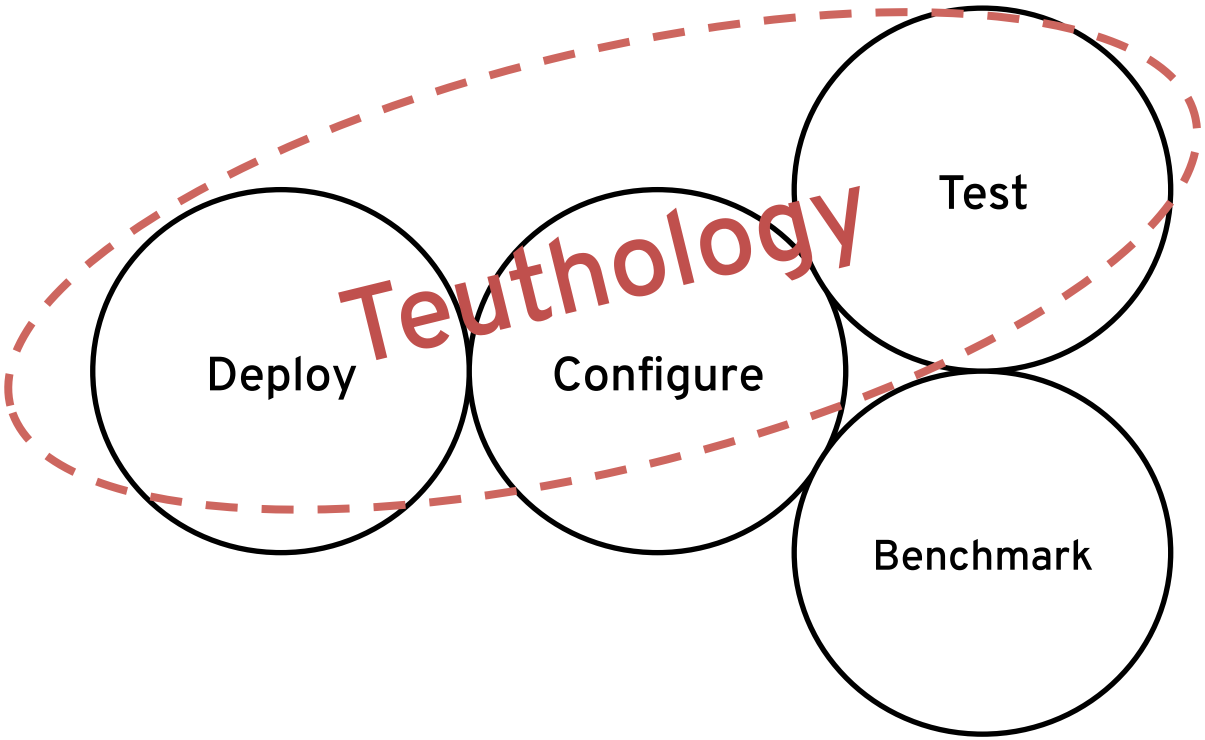
- Log file
- CSV
- RDBMS
- TSDB
- …

```
for
  cluster_size = *
expect
  ceph >= (raw * 0.9)
when
  network not saturated
```

Implementing NF Tests for Ceph

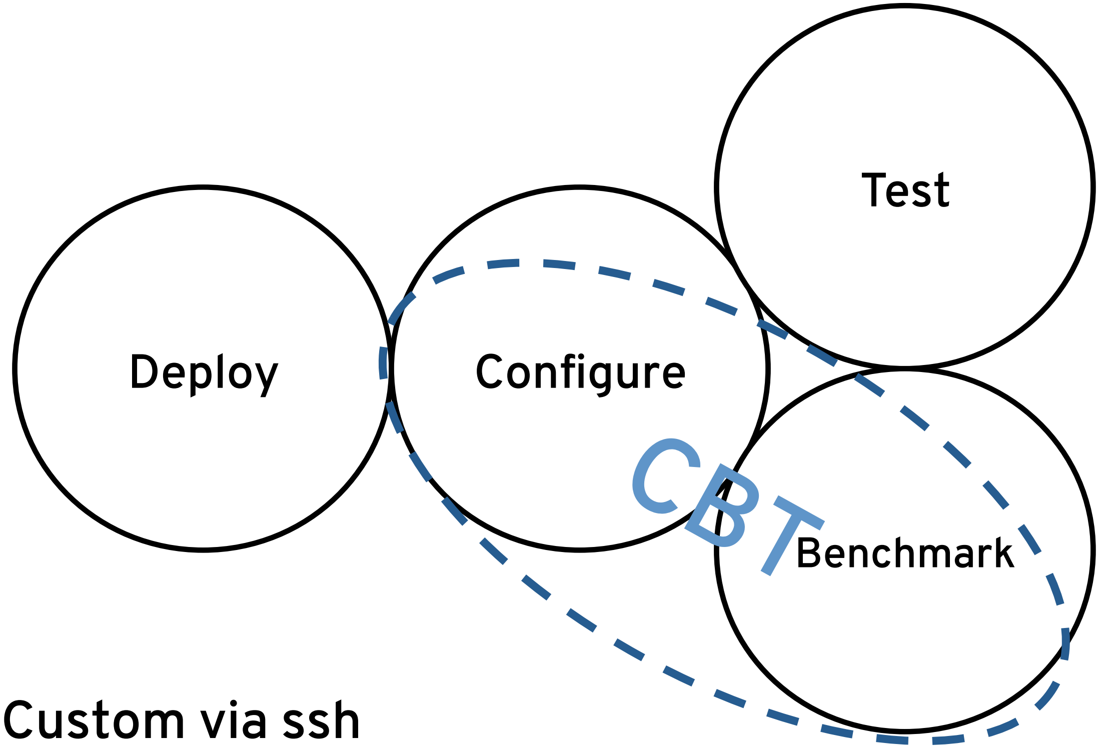
1. Deploy Ceph on Docker
   - configure cgroups dynamically
2. Run Ceph benchmarks
   - radosbench initially
3. Validate assertions over output
   - hook aver

Alternative 1



+ Deploy/Configure (via Chef/Ansible)
+ Tests
- No Benchmarking
- No Docker

Alternative 2



+ Configure – Custom via ssh
+ Benchmark – Pluggable (radosbench, cosbench, etc.)
+ Small codebase
- No Docker
- No Deployment
- No Testing

Our Plan (So Far)

1. Add docker task to teuthology
   - Leverage maestro-ng
2. Using the docker task, deploy Ceph
   - Adding the ability to configure cgroups
3. Modify ceph-qa-suite/radosbench task
   - Need to provide access to the bench output
4. Create *aver* task in teuthology (aver wrapper)
   - Check validations against benchmark output data
5. Specify validation statements
   - scalability, availability, performance, etc.