Terraform: Infrastructure as Code

Immutable Infrastructure

Evolution of Management IaC

- Manual(documentation?)
- Scripts
- Scripts of Scripts:
  - sshkit, fabric2/3, etc...
- CMS (Configuration Management Software)
- Infrastructure as code
- Immutable Infrastructure

Immutable vs. mutable

Mutable infrastructure is infrastructure that can be modified or updated after it is originally provisioned

Immutable infrastructure is infrastructure that cannot be modified once originally provisioned

Mutability trade-offs

FE	BE	Nginx	PostgreSQL
1	1	1.6	12
1	1	1.8	12
1	1	1.9	12
1	1	1.6	13

- You have system that just works fine in 99% of the uptime ...
- ... and you spend 80% of *yours* time to fix the remaining 1%

Immutable Infrastructure - Why ?

- Advantages
    - Predictable server state
    - Predictable deployments
    - Less toil work
    - No configuration drift or snowflake servers
    - Consistent staging environments and easy horizontal scaling
    - Simple rollback and recovery processes

Immutable Infrastructure - How ?

- Pets vs Cattle
- Snowflakes vs Phoenixes
- Servers-as-a-Cloud:
  - Isolated instances
  - Fast provisioning
  - From custom images
  - Well API-automated creation and destruction
- Automated CI/CD
- SOA -> IaaS, PaaS
- Stateless approach
- Persistent Data Layer
- DevOps Culture