Automated Testing for Infrastructure-as-a-code

Florian Winkler
B1 Systems GmbH

@pinguintiger,
@b1systems
Automated Testing for Infrastructure-as-a-code

- $ ls *
  - $ id; groups
  - $ last
  - $ rpm -qa
  - $ ps aux
  - $ git commit
  - $ exit 0
  - $ exit 1
  - $ ./run.sh
  - $ logout
• Florian Winkler
  – European
  – born in West-Berlin
  – likes penguins, cats and bears
  – likes good food and drinks
  – loves to cook
  – frequent festival visitor
• Florian Winkler
  – started with Basic on C64
  – Linux since 1999
  – professional since 2008
  – trainer since 2010
  – B1 Systems since 2014
    • Consultant
    • Trainer
    = SCI-, LF-approved
groups

• B1 Systems GmbH
  – founded in 2004
  – operating both nationally and internationally
  – about 100 employees
  – offices in Rockolding, Berlin, Cologne & Dresden
  – vendor-independent (hardware and software)
  – focus:
    • consulting
    • support
    • development
    • training
    • operations
    • solutions
DC Management like we are used to
- management
- developers
- QA
- L1, L2, L3 operations
- ticket-based workflows
- or even worse...
Things are changing ...
Just buzzwords (?)

- New ideas
  - $anything-as-a-service
  - agile
  - DevOps
  - hybrid
  - hyperconverged (sic!)

- Bingo!
Lets have a look at the tools ...
rpm -qa

• Source code management
  – git
  – svn
  – cvs
  – mercurial
rpm -qa

- Predefined installation
  - kickstart
  - AutoYAST
  - preseed
rpm -qa

• **Software and images**
  - Koji
  - Open Build Service
  - Kiwi
rpm -qa

• Installation tools
  – Cobbler
  – The Foreman
  – FAI
rpm -qa

- Configuration management
  - CFEngine
  - Puppet
  - Chef
  - Ansible
  - SaltStack
rpm -qa

- Enterprise tools
  - Red Hat Satellite
  - SUSE Manager
  - Spacewalk
rpm -qa

- Containers
  - LXC
  - Docker
  - BSD Jails
  - Solaris Zones
rpm -qa

- IaaS tools
  - docker-compose
  - Docker Swarm
  - Kubernetes
  - vagrant
  - terraform
rpm -qa

- Continuous integration/deployment
  - Jenkins
  - Travis
  - Gitlab CI
Wow, that‘s a lot ...
Considerations

- do NOT let the tools define your work
- define a goal
- check your workflows
- find the appropriate tools
- make these tools work for you
- automate
• DevOps
  – mindset
  – close collaboration
  – small teams
  – staging
  • Dev
  • Trial (optional)
  • Staging
  • Prod
  – automated tests
  – fast deployments
• git

  – central code repositories
  – full access control
  – Intelligent merging mechanisms
  – branches
ps aux

- Jenkins
  - central automation tool
  - feature rich
  - lots of plugins
  - different types of workers
  - variety of notifications
  - plugins for various ticket systems
But how does it work for me?
git commit

- SCM polling
  - code is committed
  - Jenkins polls the repo(s)
  - code is checked out if changed
  - defined build steps are executed
  - notification is sent out
  - successful build can trigger other project to be build
- code is committed
- GitHub hook triggers new build in Jenkins
- code is checked out
- defined build steps are executed
- notification is send out
- successful build can trigger other project to be build
git commit

• **External build trigger**
  – a special URL is called
  – triggers a new build in Jenkins
  – code is checked out
  – defined build steps are executed
  – notification is send out
  – successful build can trigger new projects to be built
• Ticket-based workflow
  – ticket is created/updated
  – triggers a new build in Jenkins
  – code is checked out
  – defined build steps are executed
  – ticket is updated/closed on success
  – successful build can trigger other project to be build
And that‘s all the magic?
exit 0

- **Automated workflow example (1/3)**
  - code is written/changed
  - code is committed to dev branch
  - ticket is automatically created
  - dev branch is checked out
  - tests are executed
    - i.e. syntax, linting
  - code is merged into trial branch
  - ticket is updated
exit 0

- Automated workflow example (2/3)
  - trial branch is checked out
  - tests are executed
    - i.e. integration tests
  - code is merged into staging branch
  - ticket is updated
• Automated workflow example (3/3)

  – staging branch is checked out
  – tests are executed
    – i.e. customer acceptance tests
  – code is merged into prod/master branch
  – ticket is closed
But things break ...
Something went wrong...

- $test is not successful
- responsible person/team is informed
- code is getting fixed and commited
- code is being checked out
- tests run again
Demo time?
• Demo (if we still have enough time)
Questions?
Stay in touch?

- https://www.b1-systems.de
  - winkler@b1-systems.de
  - info@b1-systems.de
  - training@b1-systems.de
  - @b1systems

- @pinguintiger (private, non-technical, mostly german)
Thank you!