

# **CONTINUOUS DELIVERY OF CONTINUOUS DELIVERY**

Gerd Aschemann

# ENTWICKLERTAG KARLSRUHE 2016

- Gerd Aschemann
- **gerd@aschemann.net**
- **http://aschemann.net**
- twitter:  
**@GerdAschemann**

# FRAGESTUNDE

- Wer nutzt Continuous Integration?
- Wer betreibt Continuous Delivery?
- Wer setzt auf *Infrastructure as Code* (IaC)?
- Wer baut seine CIP/CDP per IaC?
- Wem sagt das alles gar nichts?

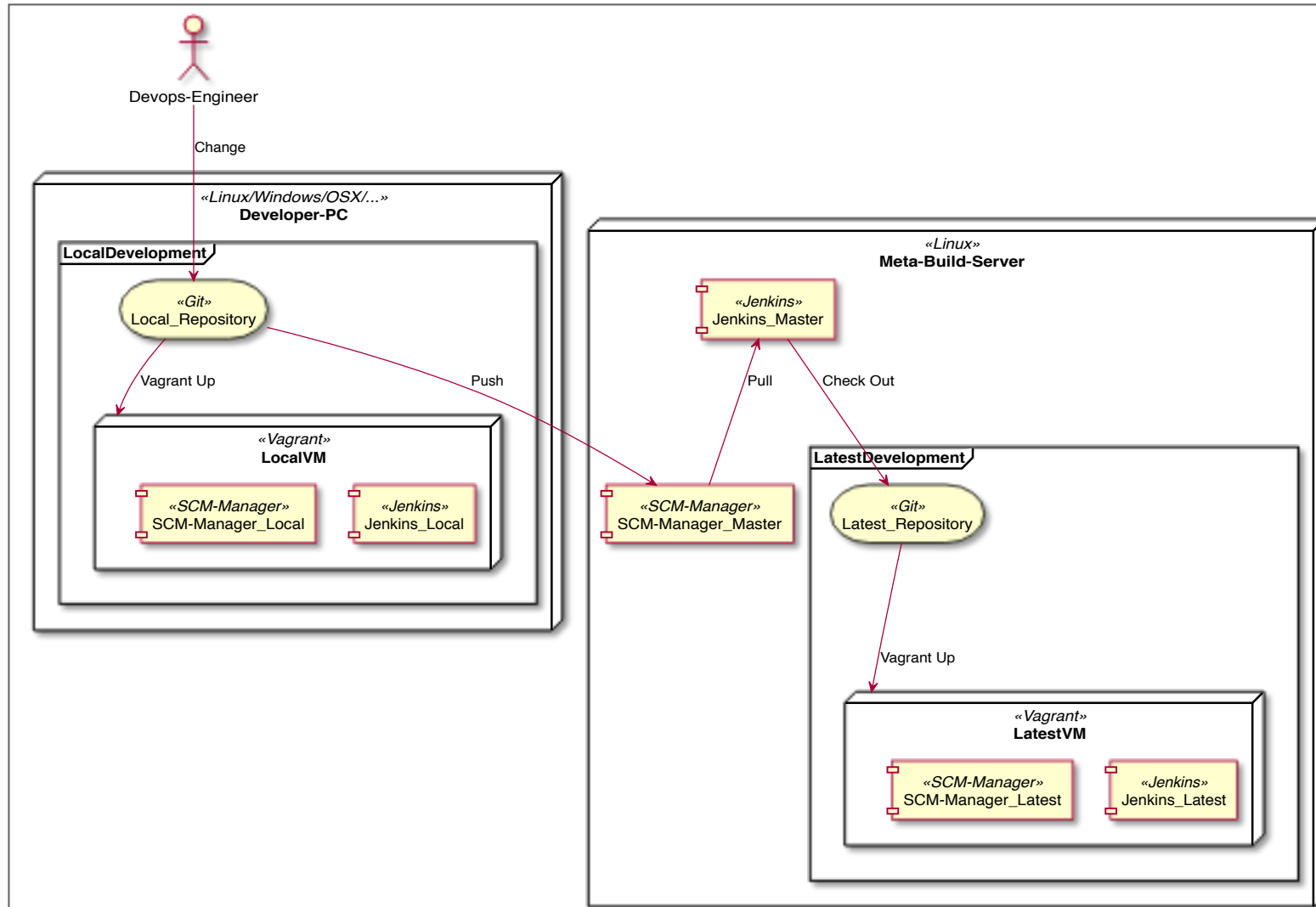
# EIN DRAMA IN FÜNF AUFZÜGEN

## Vagrant-Setup!

- Demonstration von CD4CD
  - Vagrant
  - Puppet (+ Shell + Perl)
  - Jenkins (+ Maven)
  - Docker/Nexus
- Was/Wie/Warum oder Warum nicht?

→ **Platform as Code (PaC)!**

# ZIEL-ARCHITEKTUR (VEREINFACHT)



## STAGE 0: BASE

- `modules/ (base/)`
  - `scripts/init.sh`
  - `puppet/init.pp`
- `composites`
  - `scripts/run.sh`
  - `lists/minimal`
- `Vagrantfile`

## **BASE / SCRIPTS / INIT . SH**

- Install puppet
- Install puppet modules
  - stdlib
  - etckeeper

## **BASE/PUPPET/INIT.PP**

- Purge packages (NFS, ...)
- Install packages (git, perl-test, ...)
- Setup hiera
- Setup etckeeper!



# COMPOSITES

- `lists/minimal`

# VAGRANTFILE

- Environment
  - Boxname
  - Memory
  - IP/Port Ranges
- Caching!
- Provision: `composite` (minimal)

# STAGE 1: JENKINS

- Neu: `modules/jenkins-native`
  - `scripts/init.sh`: Wie gehabt
  - `puppet/init.pp` !!!
- `composite/lists:jenkins-only`
- NEU: `scripts/test.pl`
  - ⚠ Das ist ein Unit-Test!
  - ❗ Ist das ein Unit-Test?

## STAGE 2: JENKINS INTEGRATION TEST

- Neu: `modules/it-jenkins-minimal`
- `scripts, composites`: Wie gehabt
- `puppet/init.sh`
  - Seed Job
  - HelloWorld
- Integrations-Test: Wurden die Jobs ausgeführt?

## STAGE 3: DOCKER

- Neu: `modules/data`
- Neu: `modules/docker`
- Manuelle/Inkrementelle Ausführung führt zu Fehler?!
- Keine Änderung in `Vagrantfile/composites`

## STAGE 4: NEXUS

- Neu: `modules/nexus`
- Keine Änderung in `Vagrantfile/composites`
- ❗ Puppet Docker-Module erzeugt OS-Service (Nexus)

# STAGE 5: JENKINS + NEXUS INTEGRATION TEST

- `Neu: modules/jenkins-settings-nexus-localhost`
- `Neu: modules/it-jenkins-nexus-minimal`
- `Neu: composites/lists/it-jenkins-nexus-minimal`

# USE CASES

- Kontinuierliche Fortentwicklung der CDP
- Automatisierter regelmäßiger (Neu) Aufbau der CDP
- Testen neuer Komponenten(versionen)
- Testen neuer Prozesse
- Schulung/Einarbeitung



# ENVIRONMENTS / CD STAGES

- Development/Continuous Integration (Automatisierte Tests)
- User Acceptance Test (Exploratives Testen)
- (Performance/Exception/... Tests)
- Produktion

# OFFENE FRAGE(N)

- Was ist mit dem Zustand (in Produktion)?
  - Administrativer Zustand → **PaC**
  - Operativer Zustand → *it depends*
    - Repositories
    - Alte Instanz(en) vorhalten?
    - Backup?

# FAZIT

- CD-Plattform als PaC realisiert
- Nutzung von Virtualisierung und Containern
- Modularisierung und Aggregation
- Puppet ist oft *hakelig* (Ansible auch)
- Ausblick:
  - Feedback-Loops/Monitoring fehlt
  - Unabhängigkeit von Puppet/Ansible/Salt/Chef???

# LINKS

- Beispiele:
  - <https://github.com/devopssquare/demo-etka16>
  - <https://github.com/devopssquare/helloworld>
  - <https://github.com/devopssquare/helloworld-seed>
- Slides: <http://aschemann.net/gerd/publications/cdofcd-etka2016.pdf>

**DANKE!**