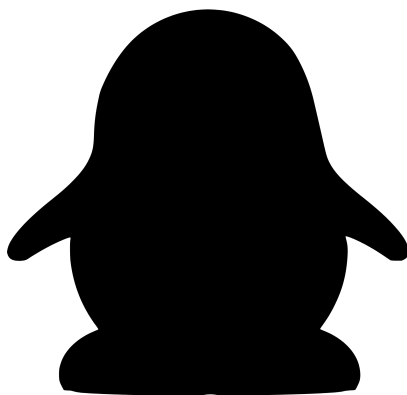




inovex

inovex classes

Linux



Session 5

Februar 2019

Agenda

17:00 - Recap Session 4

17:15 - Linux auf dem Desktop

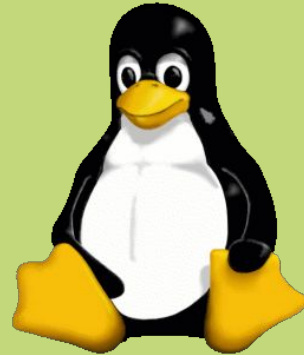
17:45 - Git auf der Kommandozeile

18:30 - LVM

18:50 - irgendwas mit \$cloud

19:05 - Gesamt-Recap + Feedbackrunde

Recap Session 4



Recap

Overview

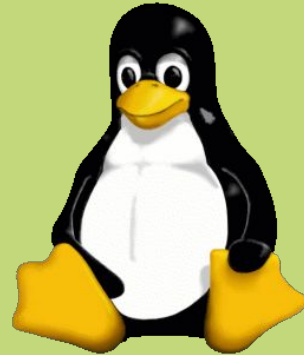
- › Boot Process
- › Disks
- › Partitions
- › File Systems
- › Mountpoints

Recap Session 3

Aktualisierte Aufgabenstellung:

<https://github.com/inovex/linux-class>

Desktop Linux



Desktop Linux

Agenda

- Grundlage X Window System (1984-87) auch X11 / X
 - Software zur Fensterdarstellung und Netzwerk-Protokoll
 - Client / Server Architektur
 - freie Forks: XFree86, Xorg
- Various Desktops bundled with Ubuntu
 - Gnome, KDE, XFCE, ...
- Install “additionally” Desktop Software (eg. at Ubuntu Server)
- Showtime!
 - TWM (1987), Ratpoison (2000), i3, etc. etc. Window-Manager
 - ubuntu, kubuntu, xubuntu Desktop Environments

Desktop Linux

Various Desktops bundled with Ubuntu

Gnome / Ubuntu (official default - again! [R.I.P. Unity])

<http://releases.ubuntu.com/18.04.2/ubuntu-18.04.2-desktop-amd64.iso>

KDE / Kubuntu

<http://cdimage.ubuntu.com/kubuntu/releases/18.04/release/kubuntu-18.04.2-desktop-amd64.iso>

XFCE / Xubuntu

<http://ftp.uni-kl.de/pub/linux/ubuntu-dvd/xubuntu/releases/18.10/release/xubuntu-18.10-desktop-amd64.iso>

Want More?

<https://www.ubuntu.com/download/flavours>

Desktop Linux

Install “additionally” Desktop Software

Gnome:

```
sudo apt-get install --no-install-recommends ubuntu-desktop
```

KDE:

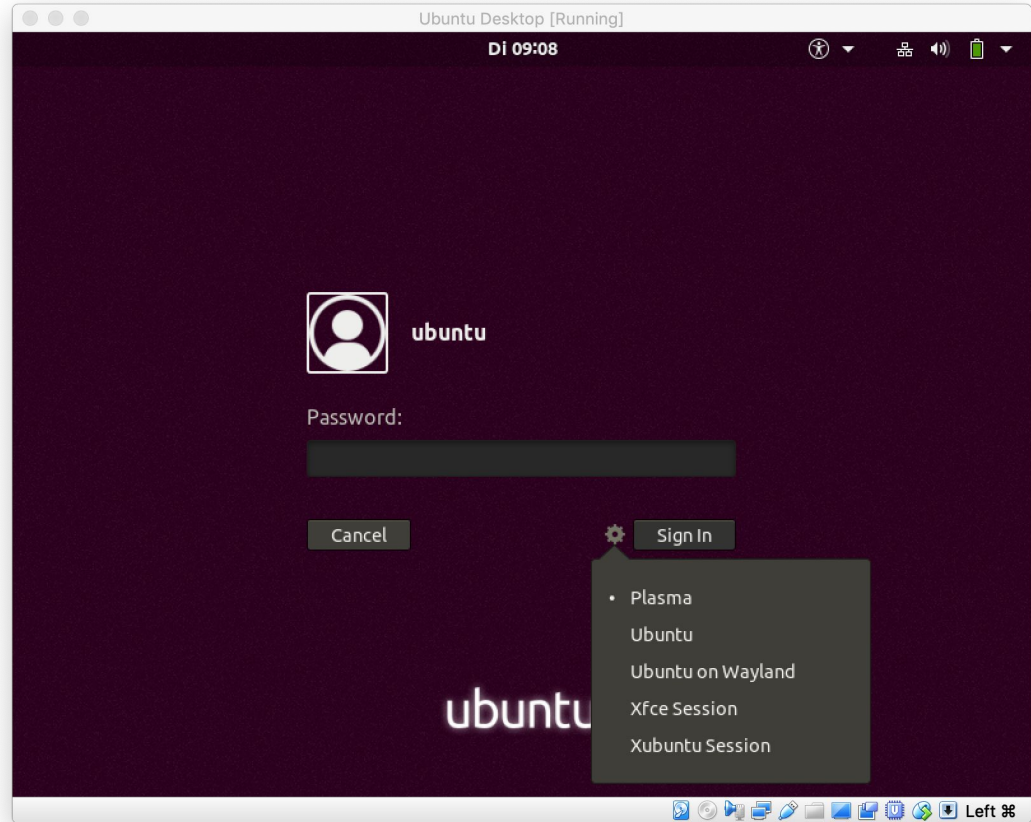
```
sudo apt-get install --no-install-recommends kubuntu-desktop
```

XFCE:

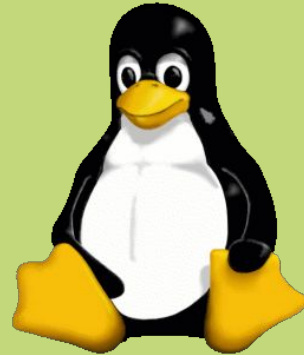
```
sudo apt-get install --no-install-recommends xubuntu-desktop
```

Desktop Linux

Showtime!



Git



Git

VCS?

Manage changes over time

Data protection

Collaboration

```
>> https://git-scm.com
```

Git

What is it?

Distributed version control

Offline capable

Collaboration

```
>> https://git-scm.com
```

Git

Version control key concepts

Make changes

Share changes

Retrieve changes

Git

~~Share Changes~~ Stage, Commit & Push

git add

git commit

git push

In case of fire



1. git commit



2. git push



3. leave building

Git

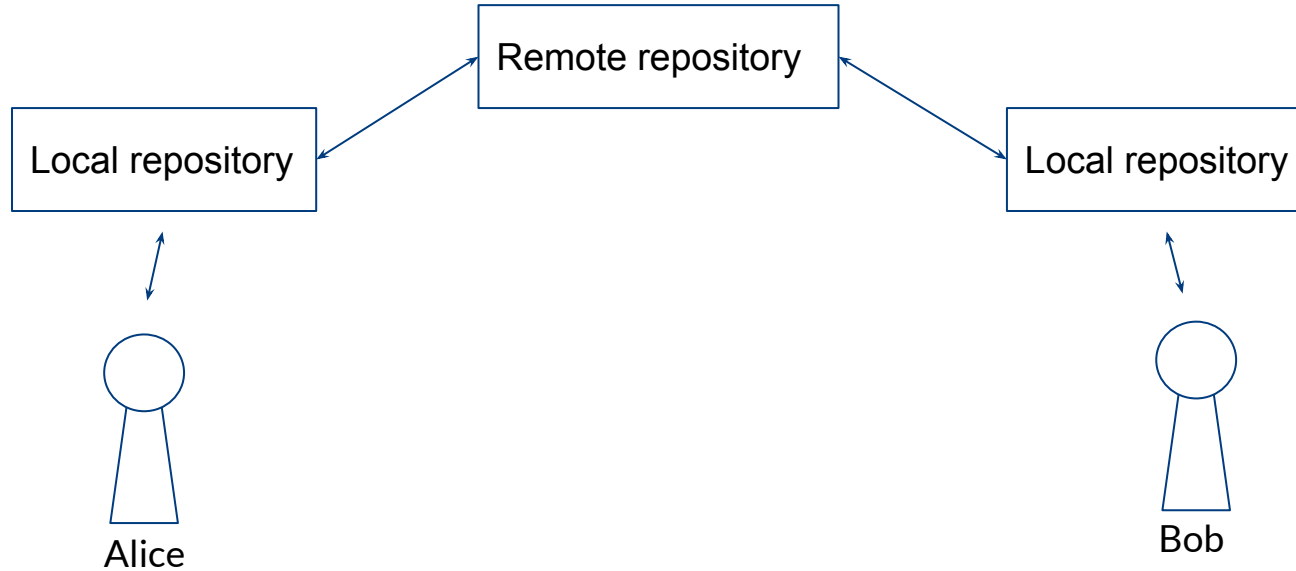
~~Retrieve Changes~~ Pull

git pull

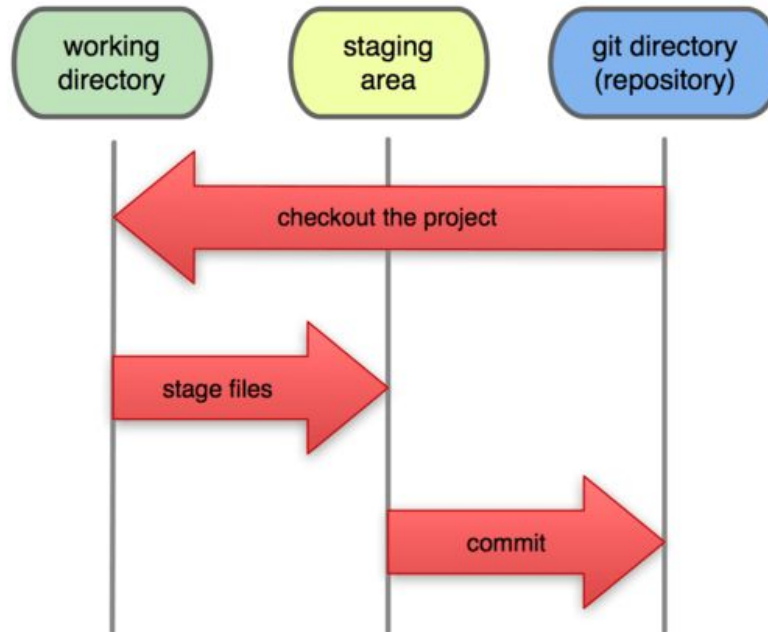
merge conflicts?

Git

Overview



Git



Git

Collaboration

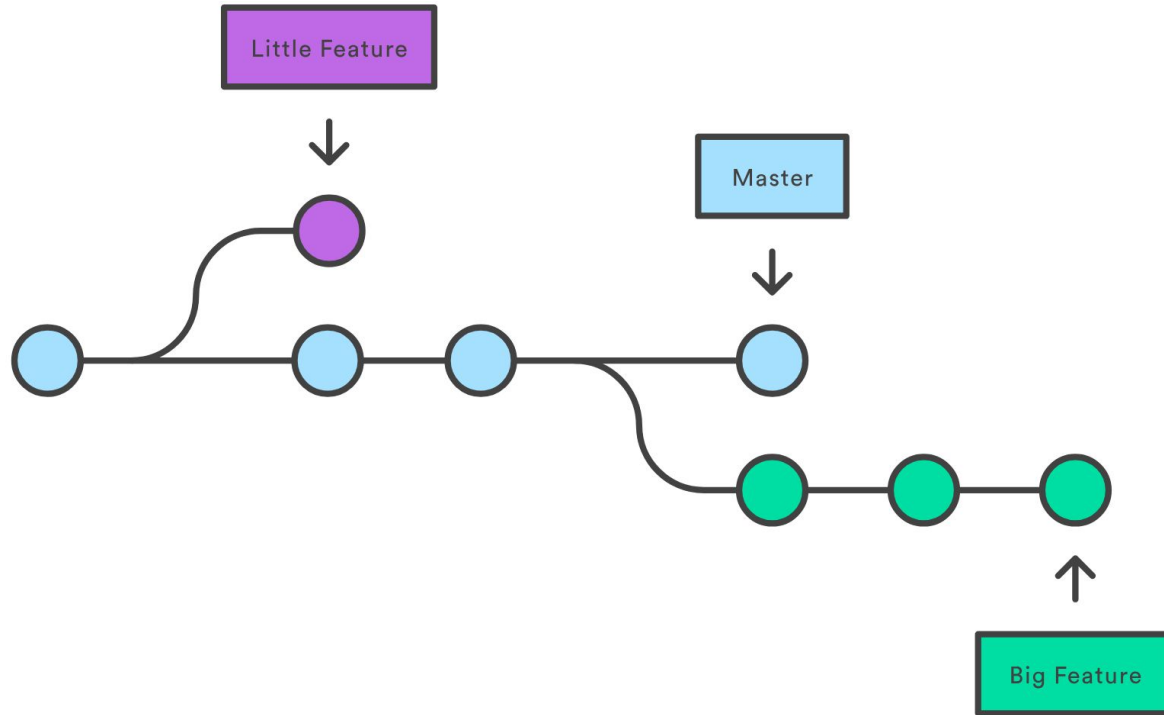
branching

`git checkout -b <branchname>`

merging

Git

Branching



Git

Demo

Git

Tasks

See: <https://github.com/inovex/linux-class>

LVM



Logical Volume Manager

“Zusatzschicht zwischen Datenspeicher und Dateisystem”

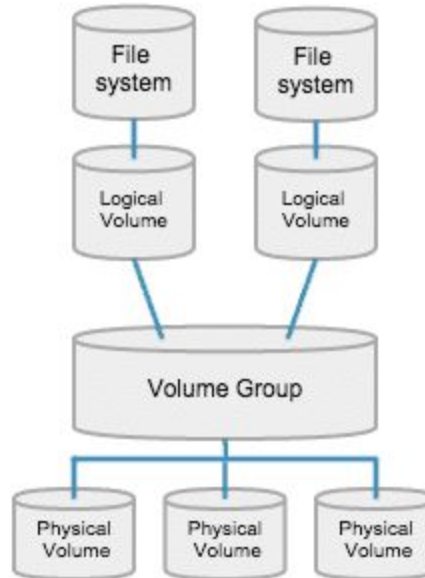
LVM - Vorteile

- Zusammenfassen mehrerer Datenspeicher zu einem “Device”
- Änderungen der Größe zur Laufzeit
- Reallokierung von Daten “online”
- Disk Striping -> Read-Performance
- Disk Mirroring -> Ausfallsicherheit
- online Snapshots

LVM - Nachteile

- erhöhte Komplexität

LVM - Aufbau



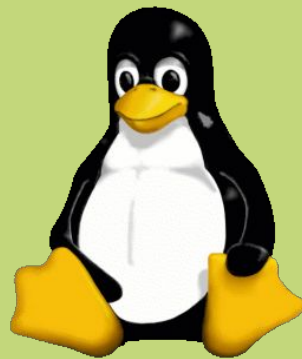
Quelle: thegeekdiary.com

LVM - Beispiel



Quelle: blog.frands.net

\$cloud / terraform



irgendwas mit \$cloud

Motivation

- Server aufsetzen? repetitive Tätigkeit
- Herausforderung nach dem dritten? Eher gering
- Folge: Fehler sind wahrscheinlich
- Konsequenz: Automation

```
15 ifconfig
16 ip a
17 vi /etc/sysconfig/network-scripts/ifcfg-ens192
18 ip a
19 dhclient renew
20 dhclient --help
21 man dhclient
22 dhclient
23 ip a
24 init 6
25 subscription-manager --list
26 subscription-manager list
27 subscription-manager register
28 export http_proxy="10.0.0.1"
29 subscription-manager register
30 vi /etc/rhsm/rhsm.conf
31 subscription-manager register
32 vi /etc/rhsm/rhsm.conf
33 ip a
34 route
35 vi /etc/sysconfig/network
36 ip a
37 ifconfig ens192
38 route
39 systemctl restart network
40 route
41 vi /etc/sysconfig/network
42 ping 8.8.8.8
43 subscription-manager register
44 route
45 curl google.de
46 tail -f /var/log/rhsm/rhsm.log
47 ping subscription.rhsm.redhat.com
```

irgendwas mit \$cloud

Welches Tool?

- Myriade an Tools
 - Eigene Bash-Skript
 - Sammlung an vorgefertigten Bash-Skripten
 - Tools für die Automation in jeder Tiefe eines Applications-Stacks
 - Tools in jeder erdenklichen Sprache,
 - viele unternehmensinterne Tools,
 - viele auch schon wieder ausgestorben...
- im Cloud Kontext quasi immer vertreten: Terraform

irgendetwas mit \$cloud

Hashicorp Terraform



Terraform

“Write, Plan, and Create Infrastructure as Code”

irgendetwas mit \$cloud

Hashicorp Terraform



Terraform

<demo>

\$ terraform init

\$ terraform plan

\$ terraform apply

\$ terraform destroy

Abschlussrunde



Recap 1-5

Was haben wir gemacht?

- Was ist Linux überhaupt? Und wieso lohnt es sich das zu kennen?
- Remote-Arbeit über SSH
- Software-Verwaltung in Linux Betriebssystemen (Paket-Management)
- Bash + Bash Scripting
- Virtualisierung + Docker
- Disk, Partitionen, Filesysteme und Mounts
- Linux Bootvorgang
- Linux auf dem Desktop
- Git per Kommandozeile
- Erweiterung von Disks, Partitionen und Filesysteme um LVM
- Wie arbeitet man eigentlich in Cloud Umgebungen?

Feedback

Wie haben wir es gemacht?

- Was fandet ihr gut?
- Was fandet ihr schlecht?
- Würdet ihr den Kurs weiterempfehlen?
- Fehlte euch Inhalt?
- Interesse an weiteren Kursen?
- Interesse an Linux?

Vielen Dank

.. uns hat es Spaß
gemacht :-)



inovex