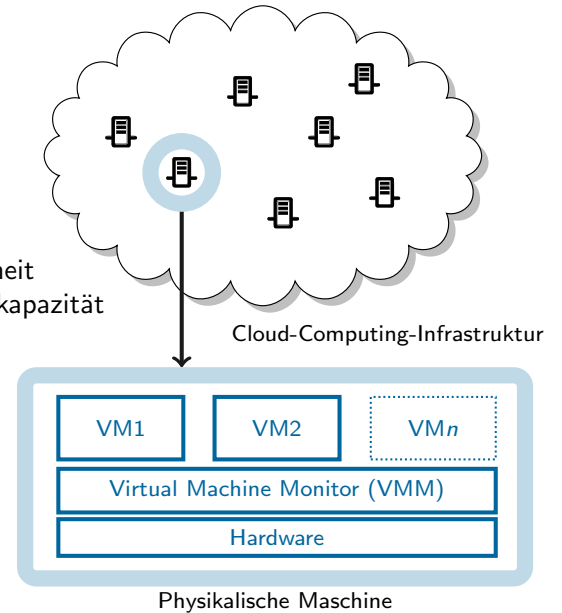


Cloud-Computing-Infrastruktur
Physikalische Infrastruktur
Software-Infrastruktur
Aufbereiten des Abbilds
Betrieb der virtuellen Maschine



- Standard-Hardware
 - „commodity hardware“
 - billig
 - **fehleranfällig**
- Charakteristik
 - hoher Grad der Verteiltheit
 - „unbegrenzte“ Speicherkapazität



Software-Infrastruktur am Beispiel von OpenStack

- **Glance:** Bereitstellung von Abbildern
 - Registry: Metadaten für Images
 - API unterstützt verschiedene Speichersysteme
- **Cinder:** Bereitstellung von Volumes
 - Volume-Service: Lokale Datenhaltung
 - Scheduler: Verteilung der Daten(-transfers) auf Rechner
- **Nova:** Verwaltung virtueller Maschinen
 - Compute: Steuerung von VMs (QEMU/Xen/...) auf Rechnern
 - Scheduler: Verteilung auf verfügbare Hardware
- **Neutron:** Netzwerkmanagement und virtuelle Router
 - Server: Steuerung und Zustandsverwaltung
 - Agents: Helfer für DHCP, Open vSwitch, Metadaten
- **Horizon (Dashboard):** Weboberfläche für Anwender



Software-Infrastruktur am Beispiel von OpenStack

- API-Dienst je Komponente für REST-Anwenderschnittstelle
 - Kommandozeilentools
- Kommunikation der Dienste intern via RabbitMQ
 - Gruppenkommunikation über Nachrichtenbus
 - Standardisiertes Protokoll: AMQP
- Speicher-Backend für Glance und Cinder: **Ceph**
 - Block-Storage oder Dateisystem verteilt auf Rechner-Cluster
 - Flexible Konfiguration von Replikationseigenschaften
 - Transaktionen über Paxos-Protokoll



Genereller Ablauf

- Speicherarten
 - Volume: Änderungen persistent, nur in einer Instanz
 - Image(Abbild): Änderungen flüchtig, Basis für viele Instanzen
- Abbild innerhalb von OpenStack erzeugen
 - Starten einer Grml-Instanz
 - Neues Volume anlegen und einhängen
 - Befüllen mit Daten (Betriebssystem)
 - Umwandeln in Image
- Web-Frontend
 - Dashboard: <http://i4cloud.informatik.uni-erlangen.de>
 - Zugangsdaten: siehe E-Mail mit Zugangsdaten
- Kommandozeile
 - OpenStack-Client-Programme: nova, cinder, glance, neutron
 - **Vor Verwendung:** openrc-Datei sourcen (vgl. Folie 2–29)



Grml-Instanz starten

- Name für Instanz festlegen
- Instanztyp i4.grml
 - Kein Swap/Ephemeral-Volume
- Automatisch Volume erzeugen
 - Ansonsten keine Zuweisung von Volumes
 - Volume beim Beenden löschen
- Kommandozeile:

```
$ nova image-list # --> grml id
$ nova network-list # --> internal net id
$ nova boot --flavor i4.grml \
  --nic net-id=<internal net id> \
  --block-device id=<grml id>,\
  source=image,dest=volume,size=1,\
  shutdown=remove,bootindex=0 \
  grml-instance
```

Achtung: Jeweils kein Leerzeichen nach dem Komma vor „source=im...“ und „shutdown=re...“ setzen.



Volume erzeugen/einhängen

Volumes & Snapshots

Name	Description	Size	Status	Type	Attached To	Availability Zone	Actions
26929a23-5e7e-4b90-a1e6-acf02d834501		1GB	In-Use	-	Attached to grml on vda	nova	Edit Volume More
retest		2GB	Available	-		nova	Edit Volume More
testv2		2GB	Available	-		nova	Extend Volume Edit Attachments Create Snapshot Delete Volume
testv		2GB	Available	-		nova	Delete Volume

- (1) Leeres Volume anlegen, benötigt Name und Größe
- (2) Volume der laufenden Instanz zuweisen
- Kommandozeile (Volume-Größe: 2 GB):

```
$ nova volume-create --display-name my-vol-name 2 # --> vol ID
$ nova volume-attach grml-instance <vol id>
```



Entwicklung des VM-Abbilds

Instance Details: grml-instance

- Konsole der laufenden Instanz im Dashboard öffnen
- Der Anleitung zur „Entwicklung eines VM-Abbilds“ aus der letzten Übung folgen
- Vor dem Verlassen der chroot-Umgebung die Java-Anwendung installieren



Installation der Java-Anwendung

- Automatisches Starten der Dienste
- Beim Systemstart führt `init(8)` die Init-Skripte aus
 - Kopieren der Projektdateien nach `/opt/mwcc-services`
 - Modifikation von `/etc/rc.local`

```
JAVA="/usr/bin/java"

cd /opt/mwcc-services
$JAVA -cp <classpath> mw.cache.MWCache <parameter> &
$JAVA -cp <classpath> mw.path.MWPathServer <parameter> &
$JAVA -cp <classpath> mw.MWRegistryAccess register <parameter>
```

- Hilfestellung und Debugging
 - Hinzufügen von `echo`-Anweisungen
 - Ausführen von `/etc/rc.local` in der `chroot`-Umgebung
 - Starten von `/etc/rc.local` mit `bash -x`



Shell Quoting

- Doppelte Anführungszeichen
 - Übergabe als einzelnes Argument
 - Expansion von Variablen möglich, „Escaping“ mittels Backslash (`\`)

```
$ argc() { echo "$0 has $# args"; }
$ argc Mein Home ist \"$HOME\"
Mein Home ist "/home/cloud" has 4 args
$ argc "Mein Home ist \"$HOME\""
Mein Home ist "/home/cloud" has 1 args
```

- Einfache Anführungszeichen (keine Expansion, kein Escaping):

```
$ argc 'Mein Home ist \"$HOME\"'
Mein Home ist \"$HOME\" has 1 args
```

- „Backticks“ (Gravis)
 - Führt Inhalt als Befehl aus
 - Ausgaben werden als Argument eingesetzt

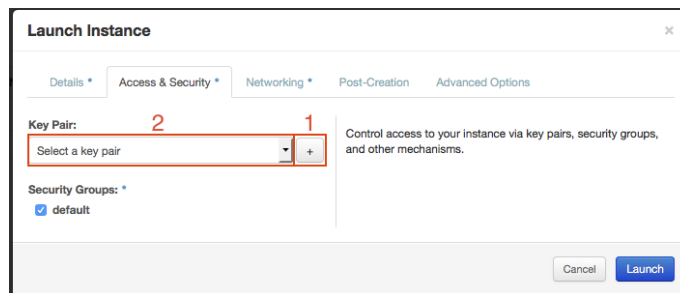
```
$ argc Es ist `date`
Es ist Fri Nov 11 11:11:00 CET 2011 has 8 args
$ argc "Es ist `date`"
Es ist Fri Nov 11 11:11:00 CET 2011 has 1 args
```



SSH-Schlüssel hinzufügen

Über die Weboberfläche

- Im Dialog zum Starten einer Instanz → „Access & Security“



- 1) **Öffentlichen Schlüssel** eines bereits vorhandenen Schlüsselpaars hinzufügen (z. B. `<gruppen_name>.pub`, vgl. Folie 2-27)
[Alternativ die Anweisungen im darauffolgenden Dialog befolgen, um ein neues Schlüsselpaar zu erzeugen.]
- 2) Den soeben neu angelegten Schlüsselpaareintrag aus Liste auswählen (danach die Instanz wie gehabt mit den gewünschten Eigenschaften starten)



SSH-Schlüssel hinzufügen

Über die Kommandozeile

- Umgebungsvariablen aus OpenStack-RC-Datei einbinden (vgl. Folie 2-29)

```
$ source /path/to/<gruppe>-openrc.sh
```

- Vorhandenen öffentlichen SSH-Schlüssel hinterlegen (einmalig)

```
$ nova keypair-add --pub-key <gruppen_name>.pub <gruppen_name>
```

- Privater SSH-Schlüssel in Datei `<gruppen_name>` (vgl. Folie 2-27)
- Öffentlicher Schlüssel in OpenStack als `<gruppen_name>` hinterlegt



Eigenes Abbild als VM starten

Launch Instance

Details * Access & Security * Networking *

Availability Zone: nova

Instance Name: * testrun

Flavor: * i4.tiny

Instance Count: * 1

Instance Boot Source: * Boot from image

Image Name: image-ce-test (2.0 GB)

- Instanztyp `i4.tiny`
→ Erzeugt Swap- u. Ephemeral-Disk
- Abbild direkt starten
- SSH-Schlüssel auswählen
- Kommandozeile: (Schlüsselübergabe mittels Parameter `--key_name`)

```
$ nova boot --flavor i4.tiny \  
  --nic net-id=<internal net id> \  
  --image <image name> \  
  --key_name <gruppen\_name> \  
  testrun
```



Öffentliche IP zuweisen

Project Compute

Access & Security

Security Groups Key Pairs Floating IPs API Access

Floating IPs

IP Address	Instance	Floating IP Pool	Actions
131.188.42.114	-	i4labnet	Associate ² More

Allocate IP To Project ¹ Release Floating IP

- (1) Öffentliche IP aus Pool allokieren, **nur einmalig nötig**
- (2) IP-Adresse an laufende Instanz zuweisen
- Kommandozeile:

```
$ nova floating-ip-create i4labnet  
$ nova floating-ip-associate my-vm-instance 131.188.42.115
```

- Abfrage innerhalb laufender VM: ('O' keine Null)

```
$ wget -qO - http://169.254.169.254/latest/meta-data/public-ipv4
```



Zugriffsregeln für Netzwerkverbindungen

Project Compute

Access & Security

Security Groups Key Pairs Floating IPs API Access

Security Groups

Name	Description	Actions
default	default	Manage Rules

- TCP-Ports müssen für öffentlichen Zugriff freigegeben werden
- Kommandozeile, z. B. für TCP-Port 22 (SSH):

```
$ nova secgroup-add-rule default tcp 22 22 0.0.0.0/0
```



Firewall-Zugriffsregeln

Add Rule

Rule: * Custom TCP Rule

Direction: Ingress

Open Port: * Port

Port:

Remote: * CIDR

CIDR: 0.0.0.0/0

Description: Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:
Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.
Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports.
Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Enter an integer value between 1 and 65535, an ICMP type and code in the spaces provided.

Cancel Add

Ingress = Eingehende Verbindungen, Egress = Ausgehende Verbindungen



■ Passwortloser Zugriff mit SSH

```
$ ssh-keygen -R <instanz_ip> # Alten Host-Key entfernen  
$ ssh -i <gruppen_name> cloud@<instanz_ip>
```

→ Schlüssel aus letzter Tafelübung, Instanz-IP aus vorheriger Zuweisung

→ Anderes VM-Image unter selber IP erfordert evtl. Zurücksetzen von Host-Key

■ Instanzen beenden: „Terminate“ auf der Weboberfläche, oder

```
$ nova list # id heraussuchen  
$ nova delete <instanz id>
```

■ Alte Abbilder/Volumes löschen: Weboberfläche, oder

```
$ nova volume-delete <volume id>  
$ nova image-delete <image id>
```



■ Modifikationen des VM-Abbilds über Grml-Instanz

- Installation weiterer Software-Pakete
- Anpassung der Startskripte
- Systemkonfiguration

■ Limitationen der Cloud-Umgebung des Lehrstuhls

- Ressourcen der drei Node-Controller sind **beschränkt**
- Beenden von nicht (mehr) benötigten Instanzen
- Jederzeit auf faire Verwendung achten

■ Infrastruktur

- Bitte sendet bei Problemen oder Ungereimtheiten schnellstmöglichst eine E-Mail an mw@i4.informatik.uni-erlangen.de

