MAYFLOWER

Cloud2Go – Umsetzung der Cloud für Ihre Webanwendung

Daniel Hallmann I 28 October 2011

Agenda

- Einführung
- Anforderungen ermitteln
- Cloudtyp auswählen
- Aus Hardware wird Software
- Cloudkomponenten für ihre Anwendung
- Orientierungshilfen und Zahlen
- Fazit

“Cloud computing is like sex in high school: Everyone is talking about it, not many are actually doing it, and none are doing it well.”

2:25 PM Jun 9th from TweetDeck

jack_daniel
Jack Daniel

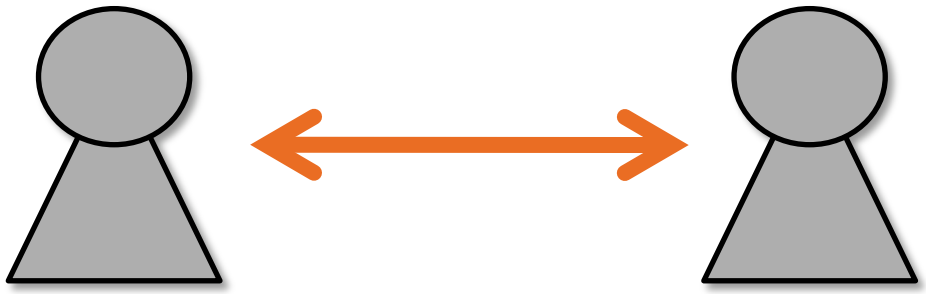
Agenda

- **Einführung**
- Anforderungen ermitteln
- Cloudtyp auswählen
- Aus Hardware wird Software
- Cloudkomponenten für ihre Anwendung
- Orientierungshilfen und Zahlen
- Fazit

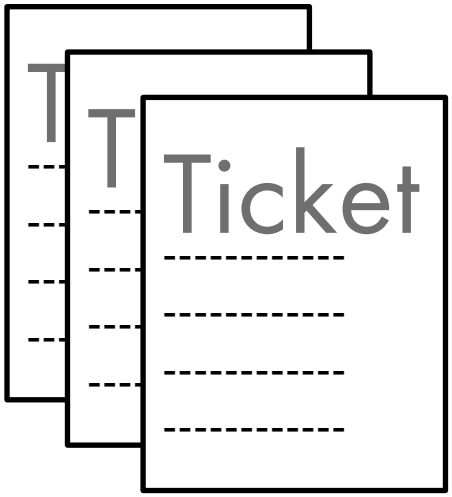
Ausgangssituation

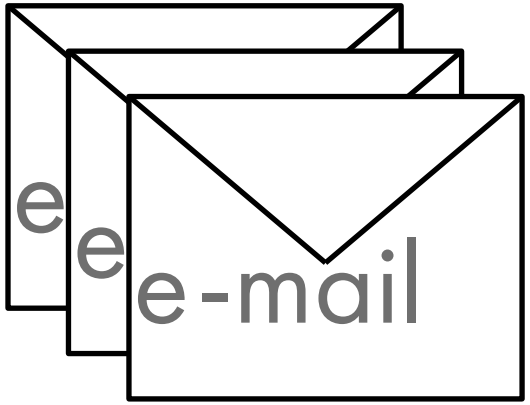
Zielsetzung

Person zu Person Kommunikation

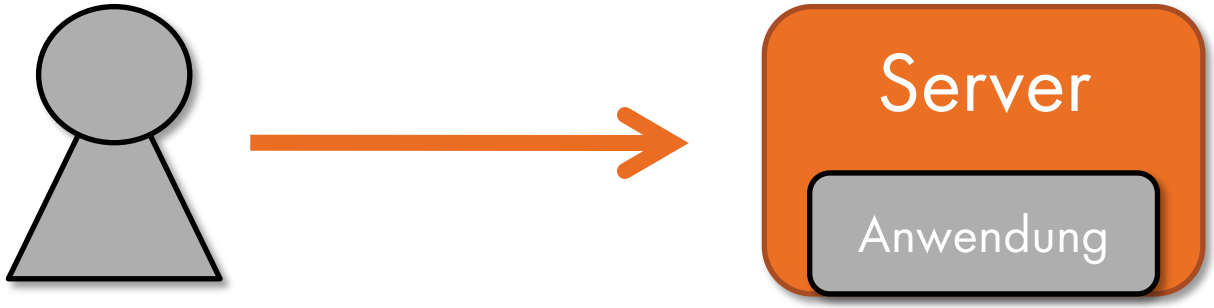


Emails / Tickets





Manuelle Arbeit



6 Wochen

Agenda

- Einführung
- **Anforderungen ermitteln**
- Cloudtyp auswählen
- Aus Hardware wird Software
- Cloudkomponenten für ihre Anwendung
- Orientierungshilfen und Zahlen
- Ausblick

2 Minuten

Ideen

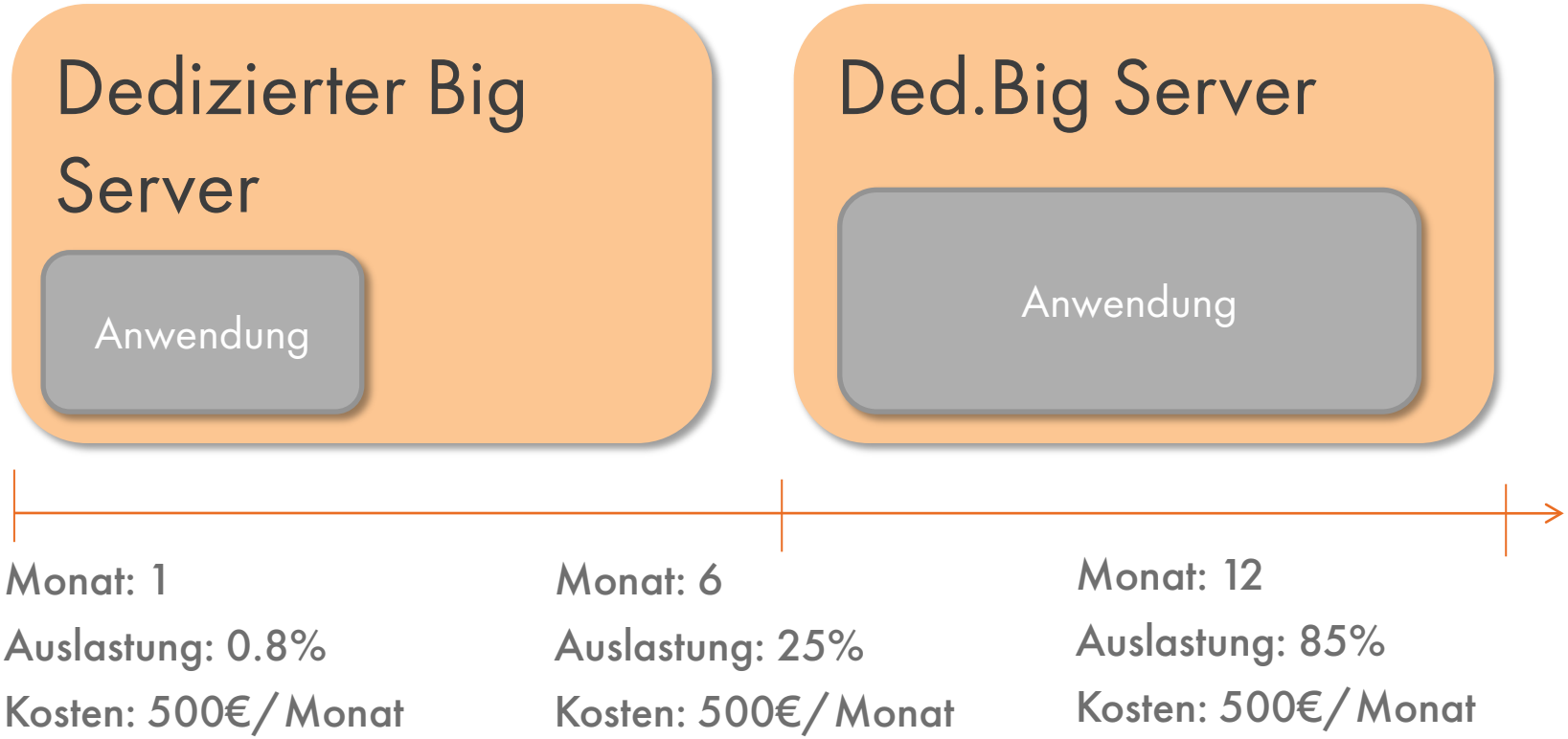
Wie hoch ist mein Ausfallrisiko (Lastspitzen)?

Best Case  Worst Case

10.000 aktive Nutzer/h

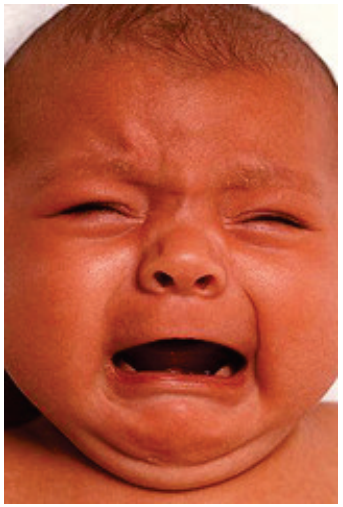
10 aktive Nutzer/h

Möchte ich eine maximale Ausfallsicherung zu Beginn?



Zeitlich

langfristig
Lastspitzen
abfangen



kurzfristig
Kennzahlen
ermitteln

durch

CLOUD

Agenda

- Einführung
- Anforderungen ermitteln
- **Cloudtyp auswählen**
- Aus Hardware wird Software
- Cloudkomponenten für ihre Anwendung
- Orientierungshilfen und Zahlen
- Fazit

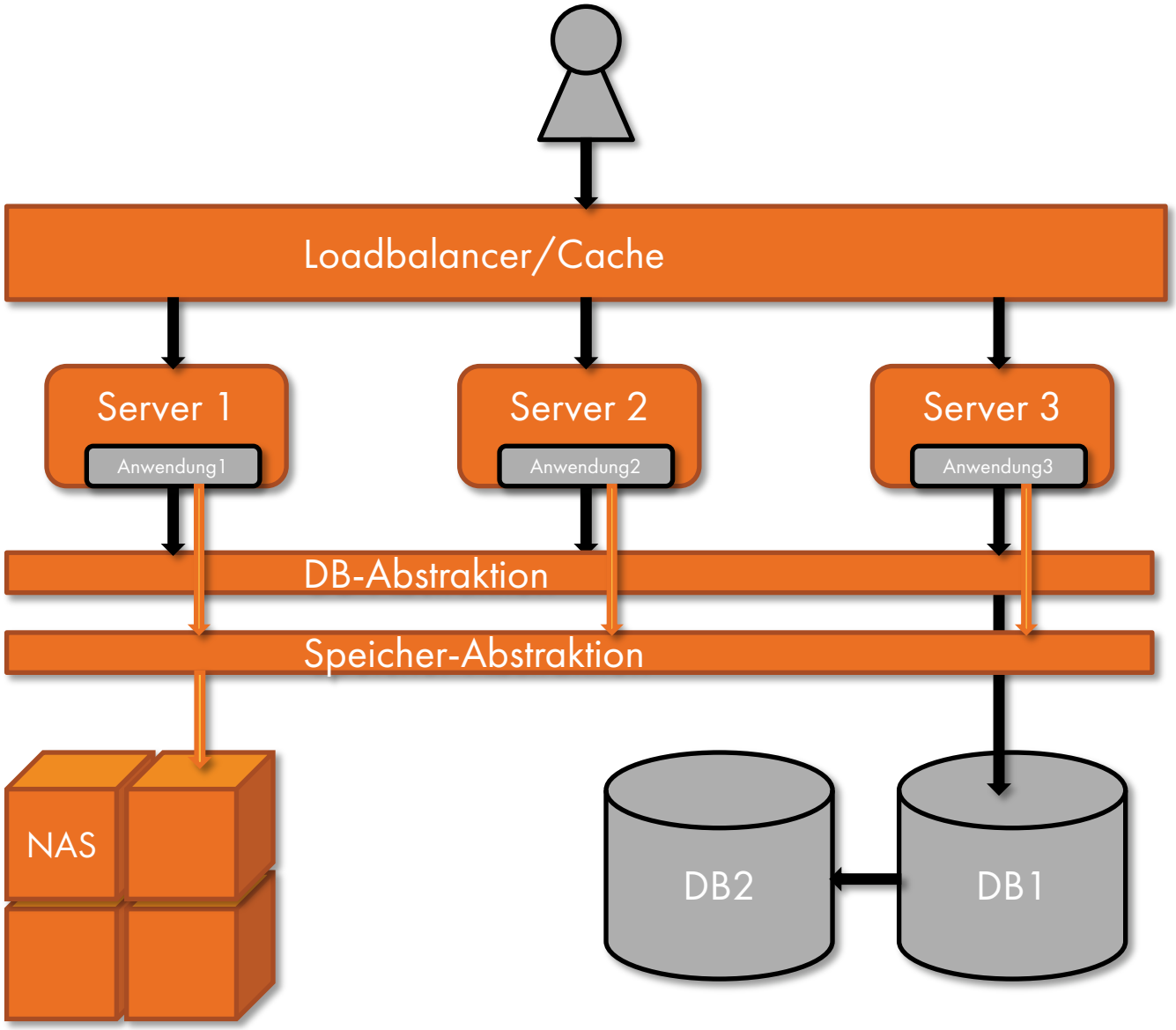
CLOUD?

PaaS
(Plattform as a Service)

PaaS

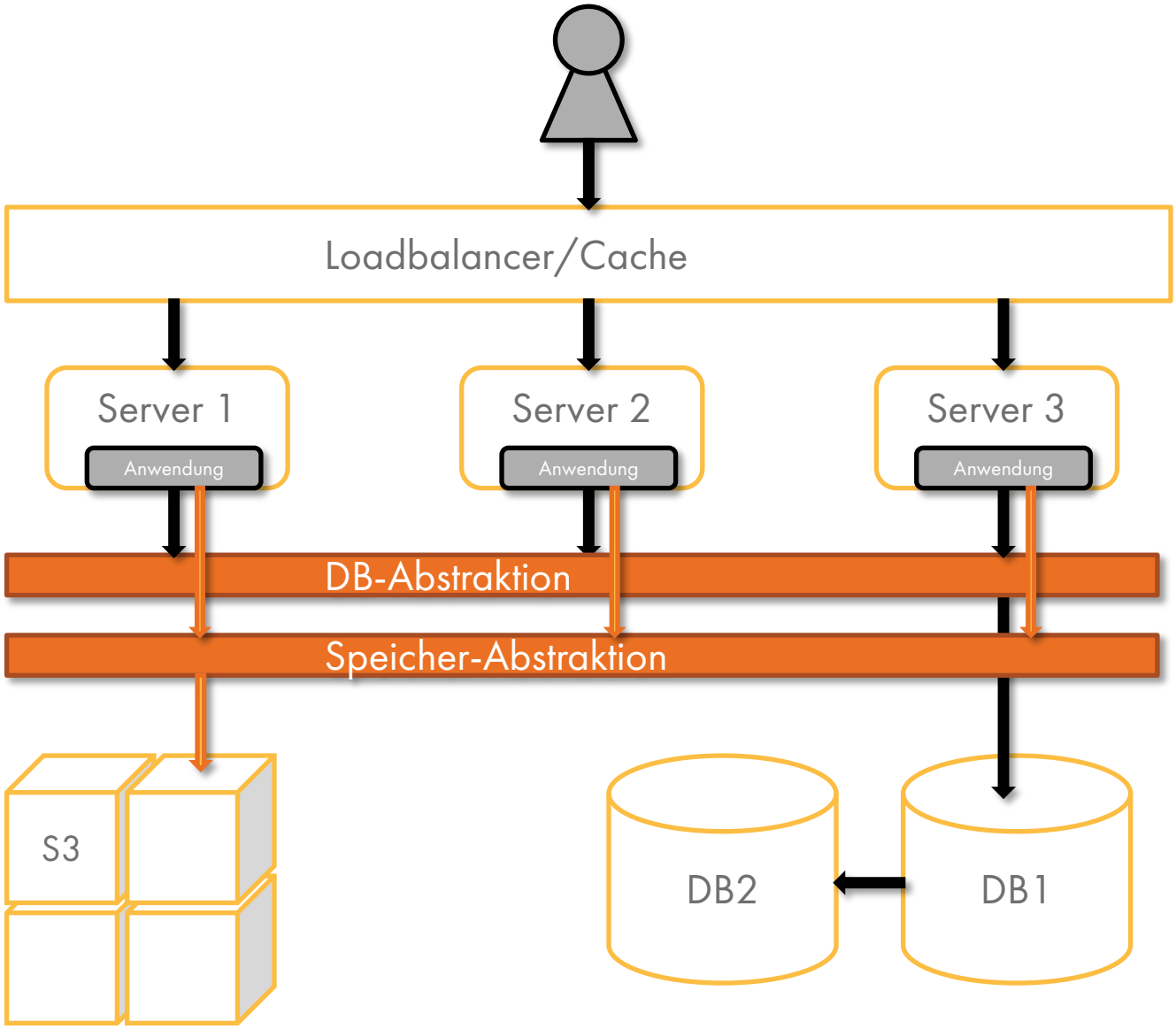
IaaS

(Infrastructure as a Service)


Loadbalancer/Cache
Server 1
Anwendung1
Server 2
Anwendung2
Server 3
Anwendung3
DB-Abstraktion
Speicher-Abstraktion
NAS
DB2
DB1

Agenda

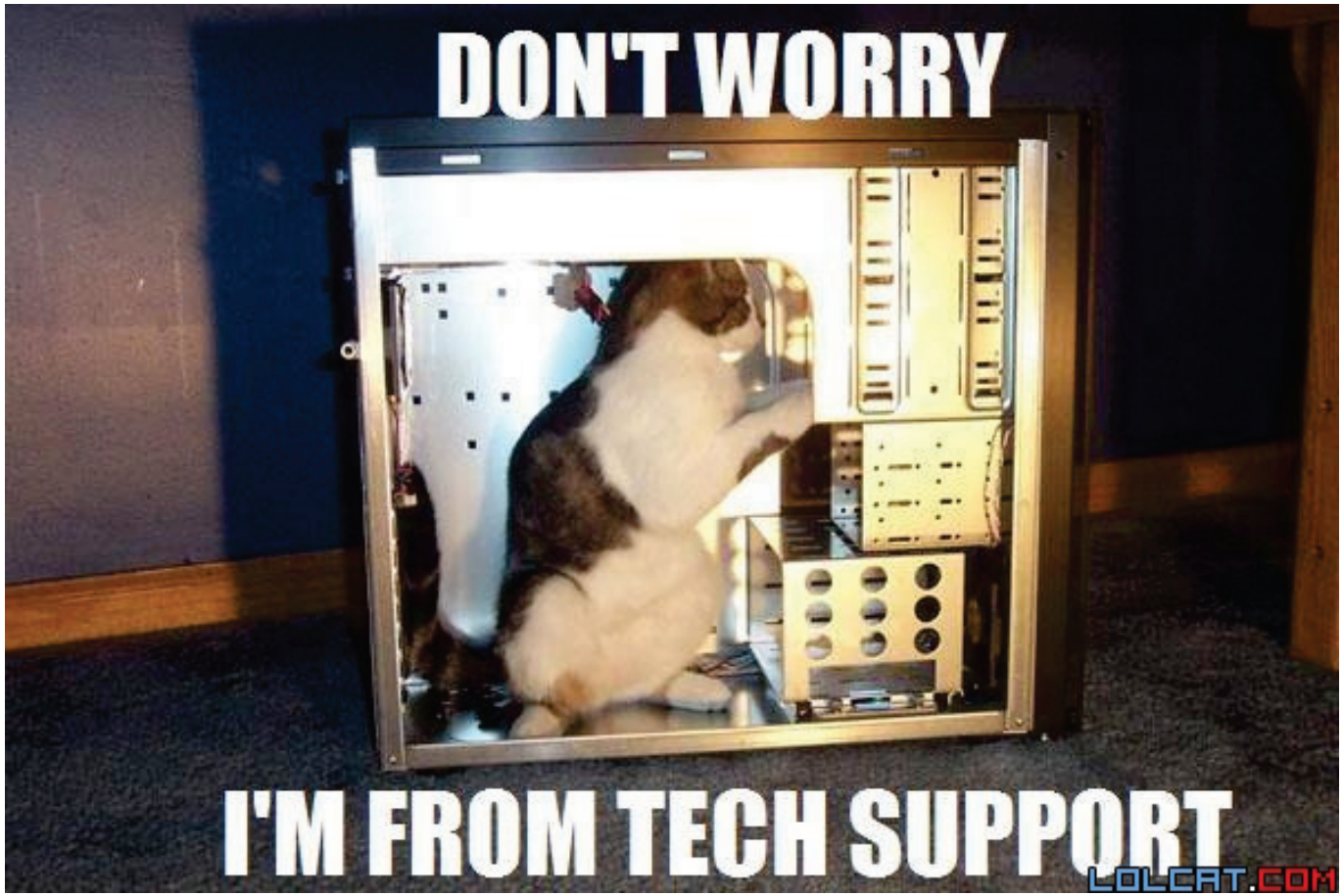
- Einführung
- Anforderungen ermitteln
- Cloudtyp auswählen
- **Aus Hardware wird Software**
- Cloudkomponenten für ihre Anwendung
- Orientierungshilfen und Zahlen
- Fazit


Loadbalancer/Cache
Server 1
Anwendung
Server 2
Anwendung
Server 3
Anwendung
DB-Abstraktion
Speicher-Abstraktion
S3
DB2
DB1

Hardware = Software

=

Immer noch Hardware (nur woanders)



Interfaces

Amazon SDK

Einige Funktionen von SDK:

AWS PHP-Bibliothek

Bauen Sie PHP-Anwendungen auf APIs auf, die die Programmierung für eine Web Service-Schnittstelle unkompliziert machen. Die Bibliothek stellt APIs bereit, die viele Routineaufgaben auf niedriger Ebene, z.B. Authentifizierung, verbergen.

Codebeispiele

Praktische Beispiele zur Verwendung der Bibliotheken für den Aufbau von Anwendungen.

PEAR Channel

Schnelle Installation und Aktualisierung des SDK mit unserem PHP Extension and Application Repository (PEAR) Kanal.

http://aws.amazon.com/sdkforphp/

Agenda

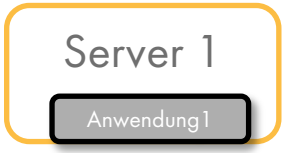
- Einführung
- Anforderungen ermitteln
- Cloudtyp auswählen
- Aus Hardware wird Software
- **Cloudkomponenten für ihre Anwendung**
- Orientierungshilfen und Zahlen
- Fazit

Komponenten

- **AMI** (Amazon Machine Images)
- **EC2** (Amazon Elastic Cloud)
- **RDS** (Amazon Relational Database Service)
- **SimpleDB**
- **S3** (Amazon Simple Storage Service)
- **ELB** (Amazon Elastic Load Balancing)
- **CloudWatch**

AMI´s (Virtual Machines)

- ToDos
  - OS-Image erstellen
  - Zentral hinterlegen
- Wins
  - Image wiederverwendbar
  - Einmaliger Administrationsaufwand



PHP Konfiguration config.php

```
<?php
define('AWS_KEY', 'your_access_key_id');
define('AWS_SECRET_KEY', 'your_secret_access_key');
```

PHP Beispiel EC2 starten

```
#!/usr/bin/php
<?php
require_once(config.php');
require_once('sdk.class.php');

$ec2 = new AmazonEC2();

$options = array('InstanceType' => "m1.micro");
$res = $ec2->run_instances("ami-48aa4921", $options);

if (!$res->isOK()) {
    exit("Could not launch instance: ". $res->body-
>Errors->Error >Message . "\n");
}
```

Login

daniel-hallmann@noon-2:.ec2 ssh -i dhkey.pem ec2-user@ec2-xxx-xxx-xxx-xxx.eu-west-1.compute.amazonaws.com

Konsole

```
Last login: Sun Oct  9 18:29:52 2011 from xxx.de

       __|  __|_  )  Amazon Linux AMI
       _|  (    /    Beta
      ___|\___|___|

See /usr/share/doc/system-release-2011.02 for latest release notes. :-)
[ec2-user@ip-xx-xx-xx-xx ~]$ ?
```

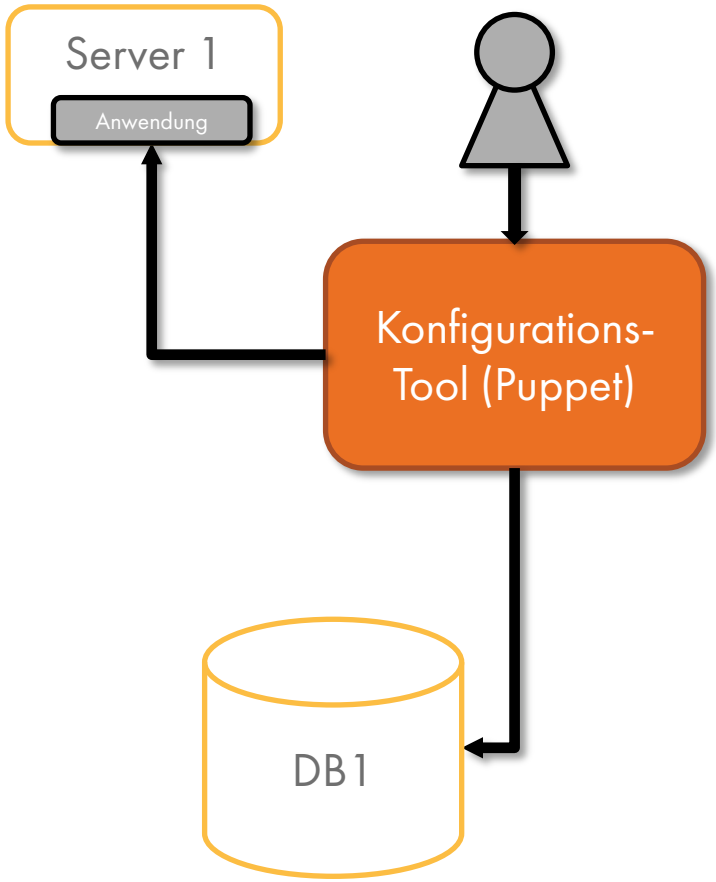
Konfiguration

ToDos

- Konfigurationsaufbau klären (Struktur)
- Einrichten Konf. Tools und Scripte

Wins

- Zentral konfigurierbar und ausrollbar
- Versionierbar

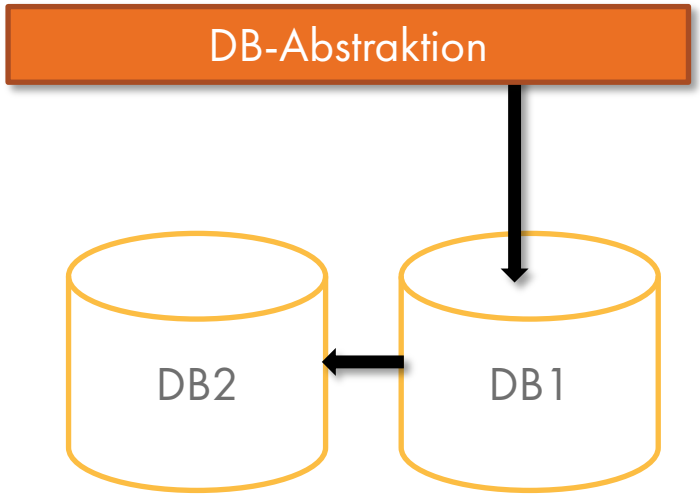


Puppet Konfigurationsscript

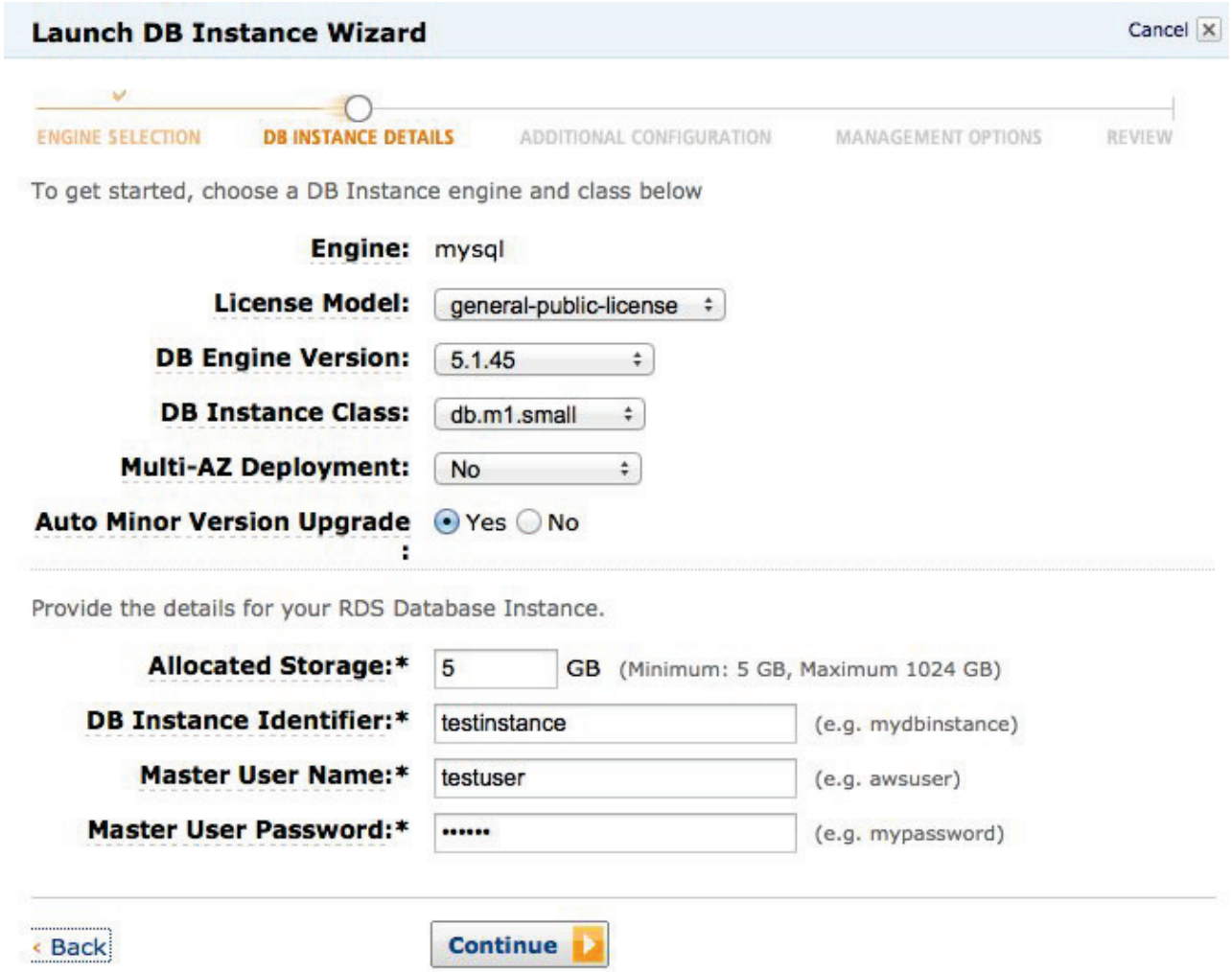
```
class web {
 package {"httpd":
  ensure => latest
 }

 file {"/etc/httpd/conf/httpd.conf":
  owner => root,
  group => root,
  mode => 664,
  source => puppet:///modules/apache/httpd.conf,
  require => Package["httpd"],
  notify => Service["httpd"]
 }

 service {"httpd":
  ensure => running,
  require => File["/etc/httpd/conf/httpd.conf"]
 }
}

node "web01.mydomain.de" {
 include web
}
```

RDS/SimpleDB

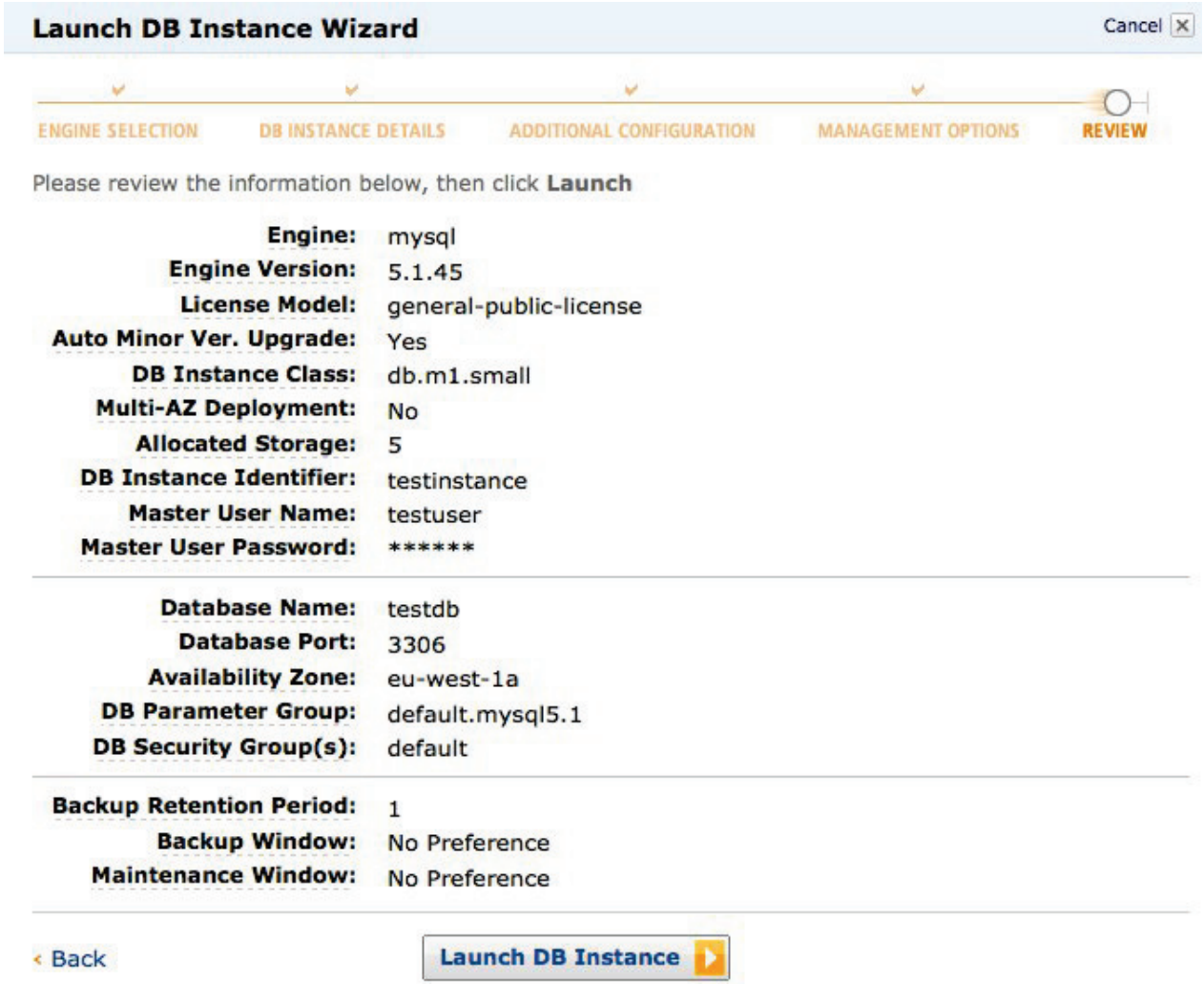
- ToDos
  - Datenbannkmodell klären
  - Einrichten, Daten migrieren
- Wins
  - Skalierbar
  - Einbindung über Service-Schnittstelle



RDS-Launch


Launch DB Instance Wizard
Cancel
ENGINE SELECTION
DB INSTANCE DETAILS
ADDITIONAL CONFIGURATION
MANAGEMENT OPTIONS
REVIEW
To get started, choose a DB Instance engine and class below
Engine: mysql
License Model: general-public-license
DB Engine Version: 5.1.45
DB Instance Class: db.m1.small
Multi-AZ Deployment: No
Auto Minor Version Upgrade: Yes No
Provide the details for your RDS Database Instance.
Allocated Storage:* 5 GB (Minimum: 5 GB, Maximum 1024 GB)
DB Instance Identifier:* testinstance (e.g. mydbinstance)
Master User Name:* testuser (e.g. awsuser)
Master User Password:* ••••• (e.g. mypassword)
Back
Continue

RDS-Launch


Launch DB Instance Wizard
Cancel
ENGINE SELECTION
DB INSTANCE DETAILS
ADDITIONAL CONFIGURATION
MANAGEMENT OPTIONS
REVIEW
Please review the information below, then click Launch
Engine: mysql
Engine Version: 5.1.45
License Model: general-public-license
Auto Minor Ver. Upgrade: Yes
DB Instance Class: db.m1.small
Multi-AZ Deployment: No
Allocated Storage: 5
DB Instance Identifier: testinstance
Master User Name: testuser
Master User Password: ******
Database Name: testdb
Database Port: 3306
Availability Zone: eu-west-1a
DB Parameter Group: default.mysql5.1
DB Security Group(s): default
Backup Retention Period: 1
Backup Window: No Preference
Maintenance Window: No Preference
Back
Launch DB Instance

PHP Beispiel create_rds.php

```
#!/usr/bin/php
<?php
require_once(config.php');
require_once('sdk.class.php');

$sdb = new AmazonRDS();

$res = $sdb-> create_db_instance('testinstance‘, 5,
‘db.m1.small‘, ‘MySQL‘, ‘testuser‘, ‘testpassword‘);

if (!$res->isOK())
{
    exit("Create db operation failed for domain
testdomain\n");
}
```

RDS-Login

daniel-hallmann@noon-2:.ec2 $mysql -u testuser -ptestpassword -h testinstance.ce4h8kirecso.eu-west-1.rds.amazonaws.com

MySQL-Konsole

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 27

Server version: 5.1.45-log MySQL Community Server (GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

MySQL-Konsole

```
mysql>show databases;
+--------------------------+
| Database                 |
+--------------------------+
| information_schema       |
| mysql                    |
+--------------------------+
2 rows in set (0.00 sec)
mysql>use mysql;
Database changed
mysql>show tables;
+--------------------------+
| Tables_in_mysql          |
+--------------------------+
| columns_priv             |
| ...                      |
+--------------------------+
```

PHP Beispiel create_domain.php

```
#!/usr/bin/php
<?php
require_once(config.php');
require_once('sdk.class.php');

$sdb = new AmazonSDB();

$res = $sdb->create_domain('testdomain');

if (!$res->isOK())
{
    exit("Create domain operation failed for domain
testdomain\n");
}
```

PHP Beispiel list_domains.php

```
#!/usr/bin/php
<?php
require_once(config.php');
require_once('sdk.class.php');

$sdb = new AmazonSDB();

$res = $sdb->list_domains();

if (!$res->isOK())
{
  exit("List domain operation failed\n");
}
foreach ($res->body->ListDomainsResult->DomainName as
$domainName)
{
  print($domainName . "\n");
}
```

PHP Beispiel list_domains.php

```
$php list_domains.php
testdomain
```

PHP Beispiel insert_items.php

```
#!/usr/bin/php
<?php
require_once(config.php');
require_once('sdk.class.php');

$values = array(array('fname' => 'Uwe', 'lname'     =>
'Brun'),
                                array('fname'  => 'Bernd',
'lname' => ,Gross',));

$sdb = new AmazonSDB();

foreach ($values as $row) {
    $attrs = array('Firstname' => $row['fname'],
'Lastname'  => $row['lname']);

    $res = $sdb->put_attributes('testdomain',
md5($row['fname']), $attrs, true);
}
```

PHP Beispiel query_items.php

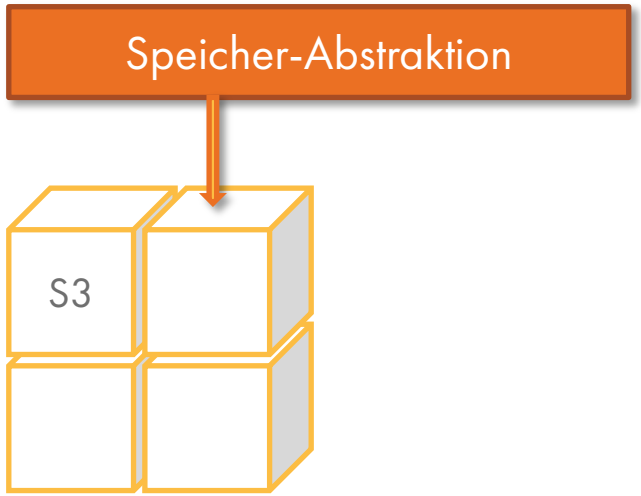
```
#!/usr/bin/php
<?php
require_once(config.php');
require_once('sdk.class.php');

$query = "select * from testdomain where Firstname like
'%Uwe%'";

$sdb    = new AmazonSDB();

$res = $sdb->select($query);

foreach ($res->body->SelectResult->Item as $item) {
  foreach ($item->Attribute as $attribute) {
    print($attribute->Name . ": " . $attribute->Value . ",
");
  }
}
```

PHP Beispiel

$php query_items.php
Firstname: Uwe, Lastname: Brun

Speicher

- ToDos
  - Speicheranforderung festlegen
  - Service-Schnittstellen integrieren
  - Datenmigration
- Wins
  - Skalierbar, unterstützt parallele Zugriffe
  - Redundante Speicherung



PHP Beispiel S3

```
#!/usr/bin/php
<?php
require_once(config.php');
require_once('sdk.class.php');

$s3 = new AmazonS3();

$res = $s3->create_bucket(“testbucket",
AmazonS3::REGION_EU_E1);

if (!$res->isOK()) {
    print("Bucket created\n");}
else {
    print("Error creating bucket\n");
}
```

PHP Beispiel list_buckets.php

```
#!/usr/bin/php
<?php
require_once(config.php');
require_once('sdk.class.php');

$s3 = new AmazonS3();

$buckets = $s3->get_bucket_list();

foreach ($buckets as $bucket) {
    print($bucket . "\n");
}
```

PHP Beispiel list_buckets.php

```
$php list_buckets.php
testbucket
```

PHP Beispiel create_item.php

```
#!/usr/bin/php
<?php
require_once(config.php');
require_once('sdk.class.php');

$s3  = new AmazonS3();

$res = $s3->create_object('testbucket',
       'testfile.txt',
       array(
         'body'               => 'testdata',
         'acl'                  => S3_ACL_PRIVATE,
         'contentType' => 'text/plain'
));
```

PHP Beispiel list_items.php

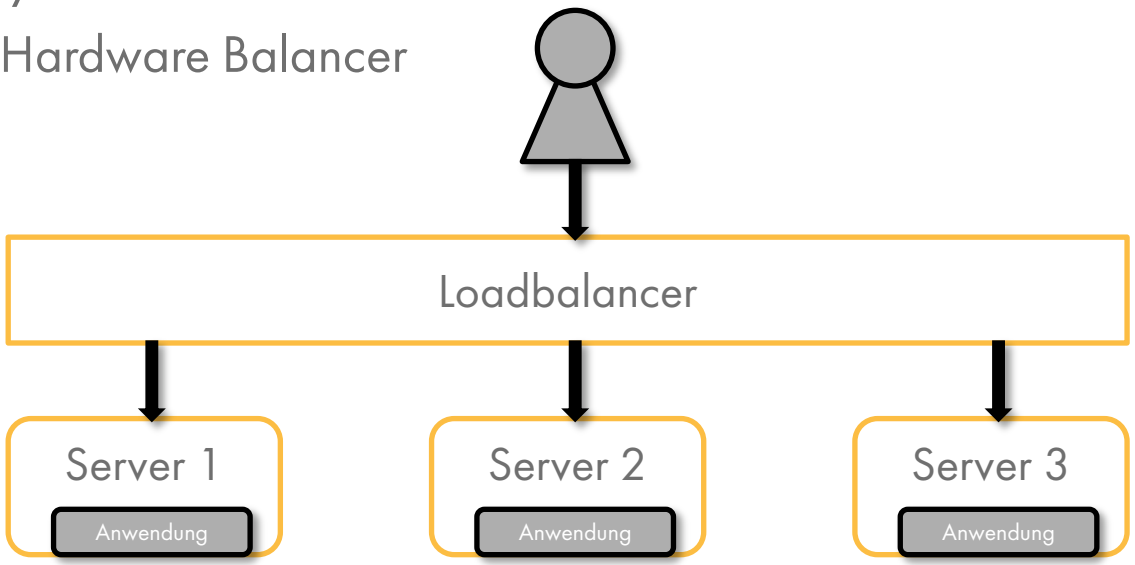
```
#!/usr/bin/php
<?php
require_once(config.php');
require_once('sdk.class.php');

$s3  = new AmazonS3();

$res = $s3->list_objects('testbucket');

$contents = $res->body->Contents;
foreach ($contents as $object)
{
  print $object->Key . "\n"; // Filename
}
```

PHP Beispiel list_items.php

$php list_items.php
testfile.txt

ELB

- ToDos
  - Service installieren/konfigurieren
- Wins
  - Skalierbar, Healthy Check
  - Ökonomischer als Hardware Balancer



Load Balancer Beispiel

$ elb-create-lb LoadBal --availability-zones eu-east-1a --listener "protocol=HTTP,lb-port=80,instance-port=80"

$elb-register-instances-with-lb LoadBal --instances i-0f414766
INSTANCE-ID i-0f414766

$elb-register-instances-with-lb LoadBal --instances i-c95650a0
INSTANCE-ID i-0f414766
INSTANCE-ID i-c95650a0

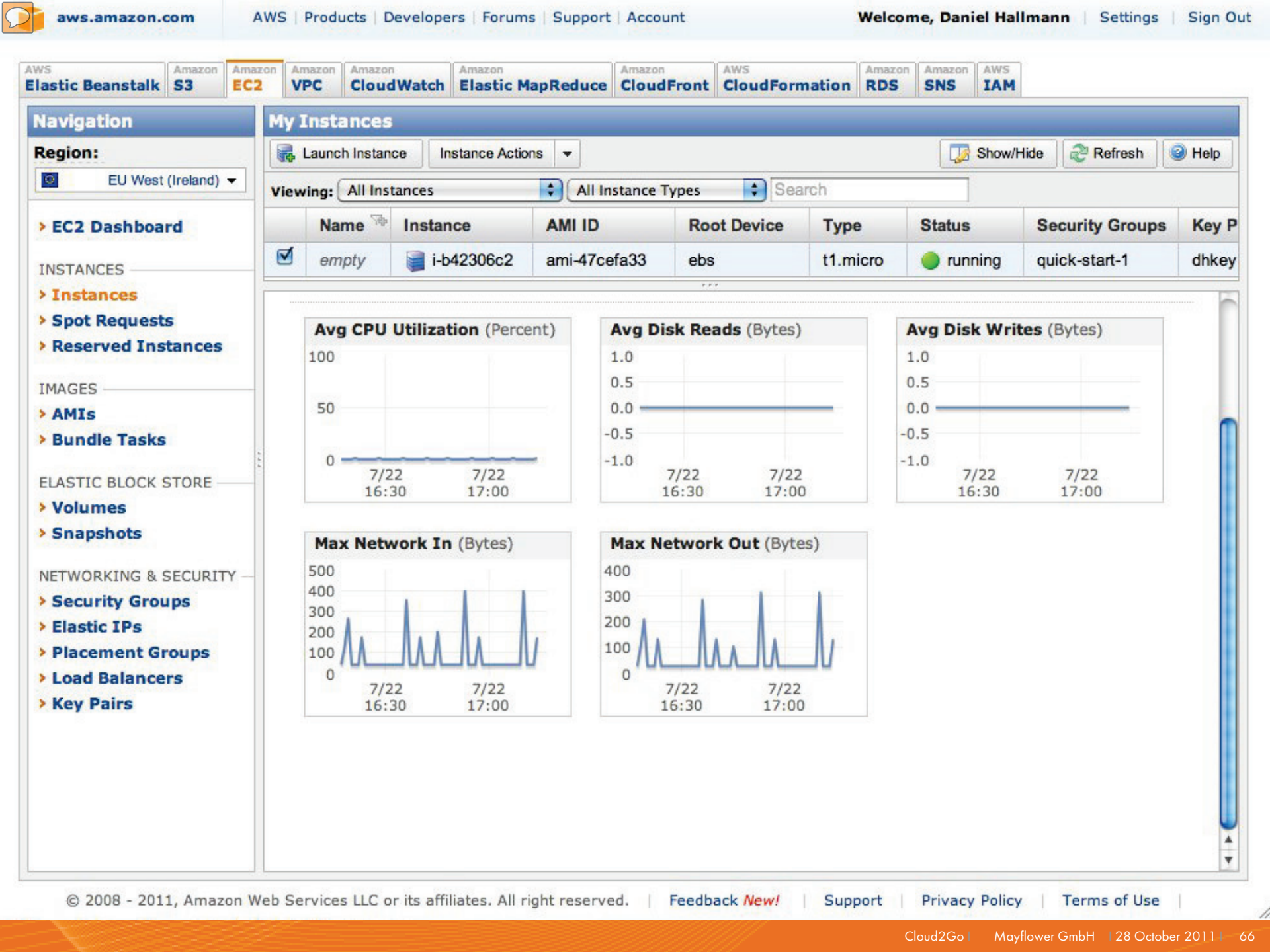
$(i-0f414766) /etc/init.d/nginx stop

$ elb-describe-instance-health LoadBal

INSTANCE-ID	INSTANCE-ID	STATE
INSTANCE-ID	i-c95650a0	InService
INSTANCE-ID	i-0f414766	OutOfService

PHP Beispiel get_statistics.php

```
#!/usr/bin/php
<?php
require_once(config.php');
require_once('sdk.class.php');

$elb  = new AmazonELB();

$res = $elb->create_load_balancer(‘LoadBal‘, array(80,
80, ‘HTTP‘), ‘eu-east-1‘);
$res2 = $elb->register_instances_with_load_balancer
(‘LoadBal‘, ‘i-0f414766‘);
$res3 = $elb->register_instances_with_load_balancer
(‘LoadBal‘, ‘i-c95650a0‘);

// $(i-0f414766) /etc/init.d/nginx stop

$res4 = $elb->describe_instance_health(‘LoadBal‘);
```


aws.amazon.com
AWS | Products | Developers | Forums | Support | Account
Welcome, Daniel Hallmann | Settings | Sign Out
AWS Elastic Beanstalk
Amazon S3
Amazon EC2
Amazon VPC
Amazon CloudWatch
Amazon Elastic MapReduce
Amazon CloudFront
AWS CloudFormation
Amazon RDS
Amazon SNS
AWS IAM
Navigation
Region:
EU West (Ireland)
EC2 Dashboard
INSTANCES
Instances
Spot Requests
Reserved Instances
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
NETWORKING & SECURITY
Security Groups
Elastic IPs
Placement Groups
Load Balancers
Key Pairs
My Instances
Launch Instance
Instance Actions
Show/Hide
Refresh
Help
Viewing: All Instances
All Instance Types
Search
Name
Instance
AMI ID
Root Device
Type
Status
Security Groups
Key P
empty
i-b42306c2
ami-47cefa33
ebs
t1.micro
running
quick-start-1
dhkey
Avg CPU Utilization (Percent)
100
50
0
7/22 16:30
7/22 17:00
Avg Disk Reads (Bytes)
1.0
0.5
0.0
-0.5
-1.0
Avg Disk Writes (Bytes)
Max Network In (Bytes)
500
400
300
200
100
0
Max Network Out (Bytes)
© 2008 - 2011, Amazon Web Services LLC or its affiliates. All right reserved.
Feedback New!
Support
Privacy Policy
Terms of Use

PHP Beispiel get_statistics.php

```
#!/usr/bin/php
<?php
require_once(config.php');
require_once('sdk.class.php');

$cW  = new AmazonCloudWatch();

$res = $cW->list_metrics();

$metrics = $res->body->ListMetricsResult->Metrics-
>member;

foreach ($metrics as $metric) {
    print "Namespace: " . $metric->Namespace . " : " .
            "Name: " . $metric->Dimensions->member->Name
. " : " .
            "Value: " . $metric->Dimensions->member-
>Value . "\n";
    }
```

PHP Beispiel get_statistics.php

```
$php get_statistics.php
Namespace: AWS/EC2 Name: ImageId      Value: ami-3c47a355
Namespace: AWS/EC2 Name: InstanceId   Value: i-aaba69c3
Namespace: AWS/EC2 Name: InstanceType Value: m1.small
...
Namespace: AWS/EC2 Name: CPUUtilization Value: 10
```

Agenda

- Einführung
- Anforderungen ermitteln
- Cloudtyp auswählen
- Aus Hardware wird Software
- Cloudkomponenten für ihre Anwendung
- **Orientierungshilfen und Zahlen**
- Fazit


Santander
V-Power
Santander
AMD
Santander

Unternehmen die Cloud nutzen

Nasdaq

NASA

Siemens

U.S. Department of State

Zahlen Amazon

Kunden in 190 Ländern

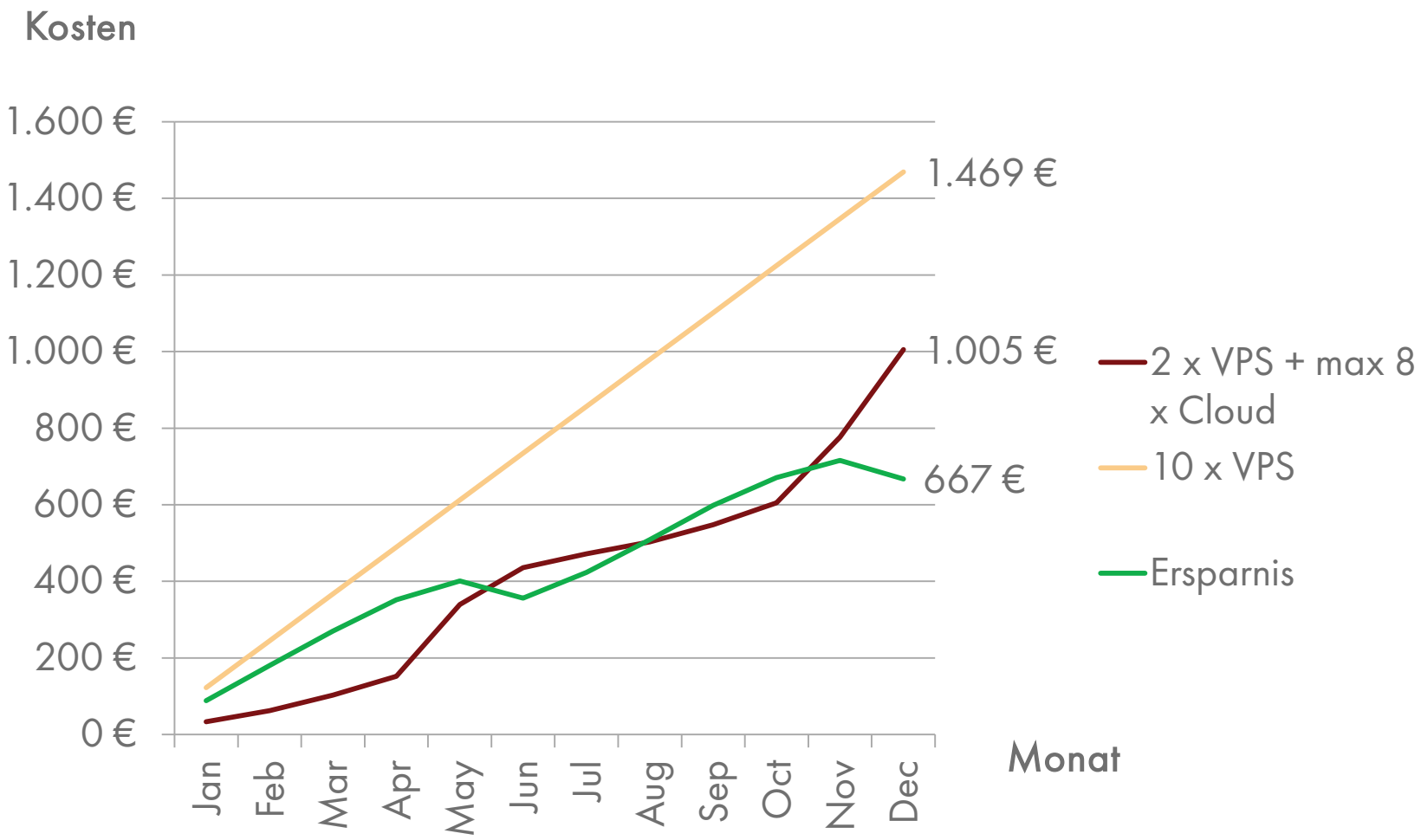
449.000.000.000 S3 Einträge
davon 71% im letzen Jahr

Top 500.000 Seiten davon
3674 mit Cloud Computing
Steigt stätig jählich um 50%

Geld

Bereich	VPS	Cloud
Server	Linux L 4.0 64bit 1GB 2Core	Micro 64bit 613MB Ram 2Core 0,014€/1h
Festplattenspeicher	50GB	0,099€/1GB - Monat 0,007€/10,000 - Req.
Monitoring	incl. (einfach)	0,010€/1h - Instanz
Datenbank	beliebig	0,078€/1h - Small
Nutzungsdauer	24/7	6/5
Preis/h	0,017€	0,102€
Preis/Monat	12,990€	12,240€

Vergleich VPS vs. Cloud (Einfache Beispielrechnung)



Agenda

- Einführung
- Anforderungen ermitteln
- Cloudtyp auswählen
- Aus Hardware wird Software
- Cloudkomponenten für ihre Anwendung
- Orientierungshilfen und Zahlen
- **Fazit**

Fazit

Flexibel

Modularer Aufbau

Webservices

Skalierung

Technologie-Trends nutzen

Links

- http://aws.amazon.com/ec2/
- http://puppetlabs.com/puppet/introduction/
- http://wiki.opscode.com/display/chef/Home
- http://varnish-cache.org/
- http://aws.amazon.com/sdkforphp/

Bilder

- Index: http://1.2.3.9/bmi/d36cz9buwru1tt.cloudfront.net/logo_aws.gif
- Schnecke: http://www.flickr.com/photos/horst-sievert/628491715
- Cloud: http://www.flickr.com/photos/kky/704056791
- Gazelle: http://www.flickr.com/photos/nationalzoo/4225498423
- Handschellen: http://www.flickr.com/photos/photokraft/3536539806
- Fernbedienung: http://www.flickr.com/photos/28478778@N05/5728485497
- Ferrari: http://www.flickr.com/photos/rodefeld/4826453453
- Chuck Norris: http://biegel.us/wp-content/uploads/2010/03/Chuck-Norris-714565.jpg
- Westerwelle: http://static.twoday.net/JanTobiParteivorsitzende/images/Guido-Westerwelle.jpg
- Kondom: http://www.flickr.com/photos/48329209@N03/4431560162
- Baby: http://www.flickr.com/photos/bbaunach/1055569383
- Mann/Frau: http://www.spassfieber.de/funpics/starke-frau-bierkiste.jpg
- LKW Überladen: http://www.spassfieber.de/funpics/heutransport-in-russland.jpg

Vielen Dank für Ihre Aufmerksamkeit!

Referent

Daniel Hallmann
daniel.hallmann@mayflower.de
+49 (89) 24 20 54 - 1135

Mayflower GmbH
Mannhardtstrasse 6
80538 München