

# Autoscaling mit AWS

# Horizontale / vertikale Skalierung

- vertikal: mehr CPU/RAM
  - irgendwann ist Schluss
  - u.a. Overhead durch Kernel Context Switche
- horizontal
  - viele, potenziell schwächere Maschinen
  - Anwendung muss damit umgehen können

# Herausforderungen bei horizontaler Skalierung (Applikationslayer)

- Persistente Daten dürfen nicht mehr lokal gespeichert werden
- Anwendung entweder stateless oder Speicherung der Session in Datenbank
- Datei-Upload nicht mehr lokal
  - S3
  - EFS
  - GlusterFS
  - ...
- Tasks soll(t)en über Queues verarbeitet werden und nicht lokal

# Horizontale Skalierung (PostgreSQL)

- Von Haus aus Replication von Master zu Slave möglich
- Skalierung erfolgt über Middleware
  - master/master durch PgCluster möglich - bisher noch nicht eingesetzt
- pgPool
  - Master-Slave Replication, Queries können ge-loadbalanct werden
  - Mehrere pgPooler teilen sich eine virtuelle IP
  - Je nach Anforderung komplexe Konfiguration vorhanden (Load Balancing, Failover etc.)

# Autoscaling mit AWS

- Autoscaling nur horizontal möglich
- Load Balancer (LB)
  - Verteilt eingehende Anfragen auf mehrere Backendsysteme
- Launch Configuration (LC)
  - Definiert AMI, Sicherheitsgruppen, Start-Script, Keypair usw.
- Autoscaling Group (ASG)
  - Nutzt eine LC
  - Minimale/maximale Anzahl Instanzen innerhalb der ASG
  - Zugeordnete Load Balancer

# Launch Configuration

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Spot Requests

Reserved Instances

Dedicated Hosts

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

LOAD BALANCING

Load Balancers

Target Groups

AUTO SCALING

Launch Configurations

Auto Scaling Groups

SYSTEMS MANAGER SERVICES

Run Command

State Manager

Automations

Patch Baselines

Create launch configuration

Create Auto Scaling group

Actions

Filter:

< 1 to 1 of 1 Launch Configurations >

Name	AMI ID	Instance Type	Spot Price	Creation Time
my-lc	ami-af0fc0c0	t2.micro		April 3, 2017 8:15:12 PM UTC+2

Launch Configuration: my-lc

Details

AMI ID

ami-af0fc0c0

IAM Instance Profile

Key Name

asg\_keypair

EBS Optimized

false

Spot Price

RAM Disk ID

User data

-

Instance Type

t2.micro

Kernel ID

Monitoring

false

Security Groups

sg-925efaf9

Creation Time

Mon Apr 03 20:15:12 GMT+200 2017

Block Devices

/dev/xvda

IP Address Type

Only assign a public IP address to instances launched in the default VPC and subnet. (default)

Copy launch configuration

# Autoscaling Group

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Spot Requests

Reserved Instances

Dedicated Hosts

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

LOAD BALANCING

Load Balancers

Target Groups

AUTO SCALING

Launch

Configurations

Auto Scaling Groups

SYSTEMS MANAGER SERVICES

Run Command

State Manager

Automations

Patch Baselines

Create Auto Scaling group

Actions

Filter:

<< < 1 to 1 of 1 Auto Scaling Groups > >>

<input type="checkbox"/>	Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
<input checked="" type="checkbox"/>	my-asg	my-lc	1	1	1	1	eu-central-1b	300	300

Auto Scaling Group: my-asg

Details

Activity History

Scaling Policies

Instances

Monitoring

Notifications

Tags

Scheduled Actions

Launch Configuration

my-lc

Load Balancers

Target Groups

Desired

1

Min

1

Max

1

Health Check Type

EC2

Health Check Grace Period

300

Termination Policies

Default

Creation Time

Mon Apr 03 20:14:35 GMT+200 2017

Availability Zone(s)

eu-central-1b

Subnet(s)

subnet-9607e9ec(172.31.16.0/20) | Default in eu-central-1b

Default Cooldown

300

Placement Group

Suspended Processes

Enabled Metrics

Instance Protection

Cancel

Save

# Wie wird skaliert?

- Scaling Policies
  - Wenn Wert X über Zeitraum Y mache Z
- Scheduled Actions
  - Zu Zeitpunkt X mache Z



# Scaling Policies

The screenshot displays the AWS Management Console interface. On the left is a navigation sidebar with categories like EC2 Dashboard, INSTANCES, AMIs, ELASTIC BLOCK STORE, NETWORK & SECURITY, LOAD BALANCING, AUTO SCALING, and SYSTEMS MANAGER SERVICES. The 'Auto Scaling Groups' link is highlighted.

The main content area shows the 'Create Auto Scaling group' button and a filter bar. A 'Create Alarm' modal dialog is open in the center. It contains the following fields:

- Send a notification to:** No SNS topics found...
- Whenever:** Average of CPU Utilization
- Is:** >= 50 Percent
- For at least:** 1 consecutive period(s) of 5 Minutes
- Name of alarm:** awssec2-my-asg-High-CPU-Utilization

To the right of the form is a line graph titled 'CPU Utilization Percent' showing a peak for 'my-asg' at 14:00. Below the graph are 'Cancel' and 'Create Alarm' buttons.

Below the modal, the 'Create Scaling policy' form is visible. It includes:

- Name:** my-scaling-policy
- Execute policy when:** No alarm selected (with a 'Create new alarm' link)
- Take the action:** Add 5 instances
- Instances need:** 100 seconds to warm up after each step

Buttons for 'Cancel' and 'Create' are at the top right of the form. A link 'Create a simple scaling policy' is at the bottom left.

# Scheduled Action

The screenshot displays the AWS Management Console interface. On the left is a navigation sidebar with categories like EC2 Dashboard, INSTANCES, IMAGES, ELASTIC BLOCK STORE, NETWORK & SECURITY, LOAD BALANCING, and AUTO SCALING. The 'Auto Scaling Groups' link is highlighted. The main content area shows the 'Auto Scaling Group: my-asg' page with tabs for Details, Activity History, Scaling Policies, Instances, Monitoring, Notifications, Tags, and Scheduled Actions. The 'Scheduled Actions' tab is active, showing a table with no entries and a 'No scheduled actions for this Auto Scaling group' message. A 'Create Scheduled Action' modal dialog is open in the center. The dialog has a title bar with a close button. Inside, the 'Name' field is 'my-scheduled-action' and the 'Auto Scaling Group' is 'my-asg'. A blue box prompts the user to 'Provide at least one of Min, Max and Desired Capacity'. Below this, the 'Min' field is 4, 'Max' is 10, and 'Desired Capacity' is 6. The 'Recurrence' dropdown is set to 'Cron' with the value '0 23 \* \* FRI' and an example '0 23 \* \* MON-FRI'. The 'Start Time' field is '00 : 00 UTC' with a note to specify the start time in UTC. The 'End Time' field has a 'Set End Time' link. At the bottom right of the dialog are 'Cancel' and 'Create' buttons.

**Create Scheduled Action**

Name: my-scheduled-action

Auto Scaling Group: my-asg

Provide at least one of Min, Max and Desired Capacity

Min: 4

Max: 10

Desired Capacity: 6

Recurrence: Cron 0 23 \* \* FRI Example: 0 23 \* \* MON-FRI

Start Time: 00 : 00 UTC Specify the start time in UTC  
The first time this scheduled action will run

End Time: [Set End Time](#)

[Cancel](#) [Create](#)

Auto Scaling Group: my-asg

Details Activity History Scaling Policies Instances Monitoring Notifications Tags **Scheduled Actions**

Create Scheduled Action Actions

Filter: Filter scheduled actions...

Name	Start Time	End Time	Recurrence	Desired Capacity	Min	Max
No scheduled actions for this Auto Scaling group						

# Wie bekomme ich meine App skaliert?

- Es gibt nicht "den einen" Weg - hängt vom bisherigen Projekt ab
- CodeDeploy
- Beanstalk
- Prebaked AMI und AWS API
- Configuration Management mit Chef oder Hosted Chef, Puppet oder Ansible
- Deployment mit Spinnaker
- ...

# Fragen?

- Entweder per Twitter an @schakko
- oder per E-Mail an [christopher.klein@neos-it.de](mailto:christopher.klein@neos-it.de)