POSTGRESQL ON AWS:

TIPS & TRICKS
(AND HORROR STORIES)

ALEXANDER KUKUSHKIN

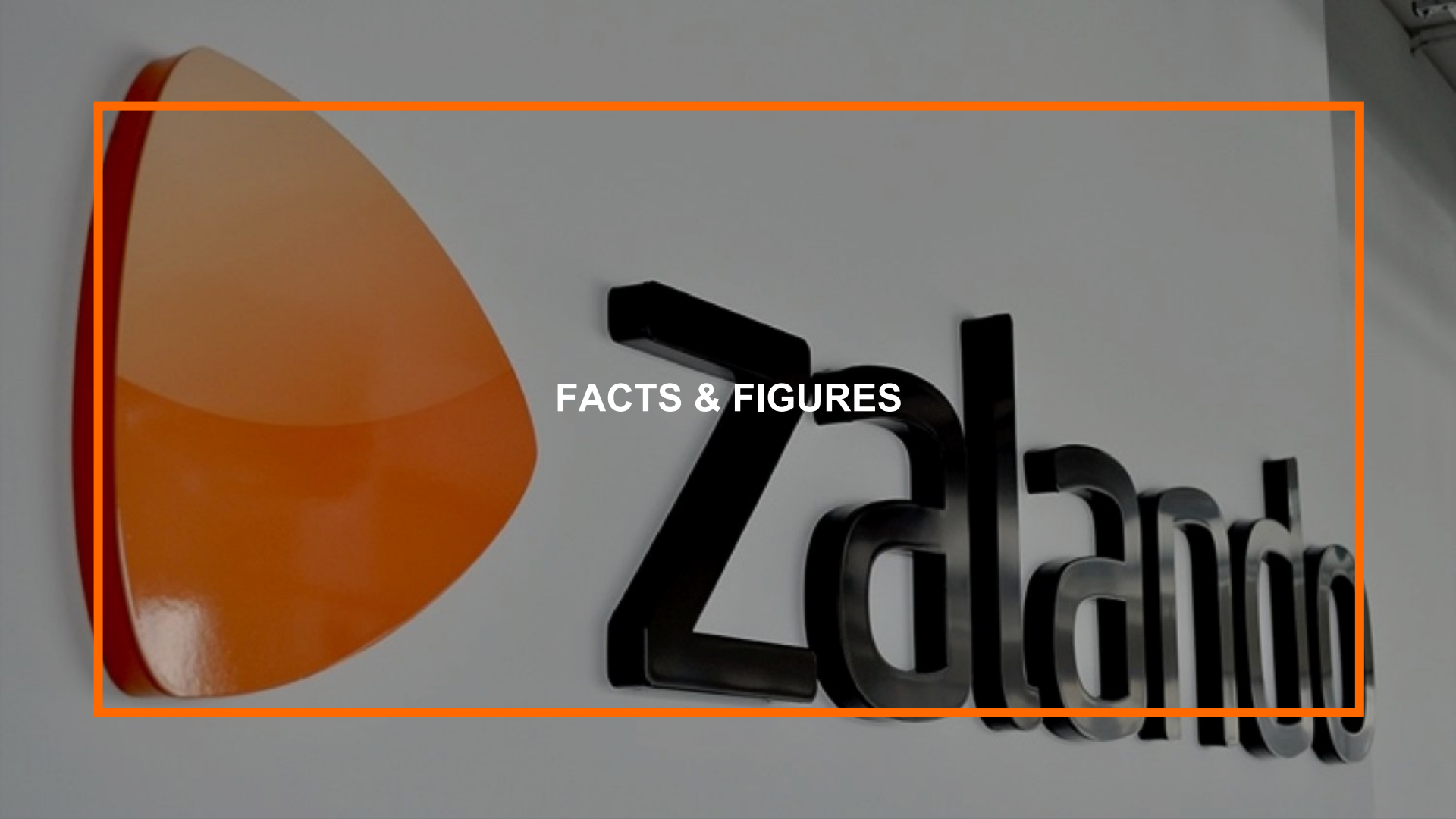
07-07-2017

ABOUT ME

Alexander Kukushkin

Database Engineer @ZalandoTech
Email: alexander.kukushkin@zalando.de
Twitter: @cyberdemn

zalando


FACTS & FIGURES

ZALANDO AT A GLANCE

~3.6 billion EURO
revenue 2016

~200 million
visits per month

~20 million
active customers

~200,000
product choices

~2,000
brands

15
countries

>12,000
employees in Europe

50%
return rate across all categories

zalando

ZALANDO TECHNOLOGY

BERLIN

DORTMUND

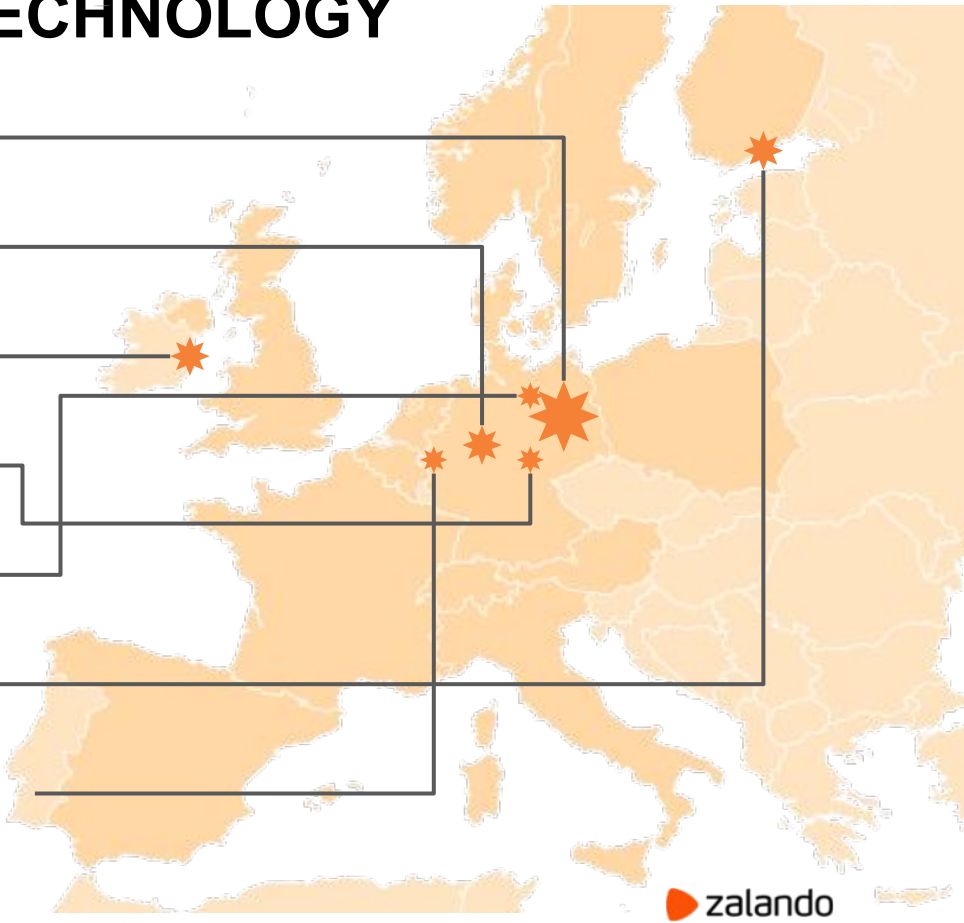
DUBLIN

ERFURT

HAMBURG

HELSINKI

MÖNCHENGLADBACH



zalando

ZALANDO TECHNOLOGY

- > 150 databases in DC
- > 170 databases on AWS
- > 1700 tech employees
- We are hiring!

WHY WE USE AWS

- Fast hardware (service) provisioning
- Easy scale up/down/in/out
- Pay only for resources you need

GLOSSARY

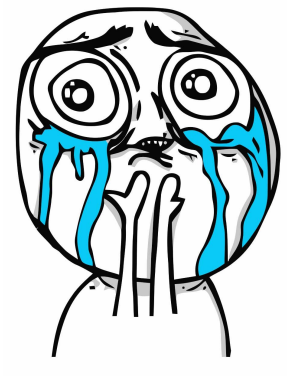
- RDS - Relational Database Service
- EC2 - Elastic Compute Cloud
- EBS - Elastic Block Store
- S3 - Simple Storage Service
- ELB - Elastic Load Balancing
- AZ - Availability Zone
- ASG - Auto Scaling Group

POSTGRESQL HA ON AWS

- Shared storage - **EBS,** attach it to a different **EC2** Instance
  - Works within one **AZ**
  - Replicas can’t execute queries
- Storage mirroring - **Multi-AZ RDS** (**EBS** is replicated to another **AZ**)
  - Works between **AZ**
  - Replicas can’t execute queries
- Streaming replication - you can start additional **RDS** replicas
  - Works between **AZ**
  - Replicas can execute queries
  - Automatic Failover problem (if not **Multi-AZ** deployment)

RDS SUMMARY

- No superuser access
- No replication connection
- No custom extensions



- Either you run **Multi-AZ RDS** deployment (replica can’t receive queries)
- Or **Master** + **Replica** on **RDS** and don’t have automatic failover

EC2 VS. RDS: PRICING*

Instance Type	RDS Multi-AZ	2 x EC2 + ELB**	3 x EC2 + ELB
(db.)t2.micro	$36.0/month	$41.8/month	$51.8/month
(db.)t2.small	$64.8/month	$60.5/month	$79.9/month
(db.)t2.large	$241.9/month	$177.1/month	$254.9/month
(db.)m4.large	$312.5/month	$194.4/month	$280.8/month
(db.)m4.xlarge	$626.4/month	$367.2/month	$540.0/month

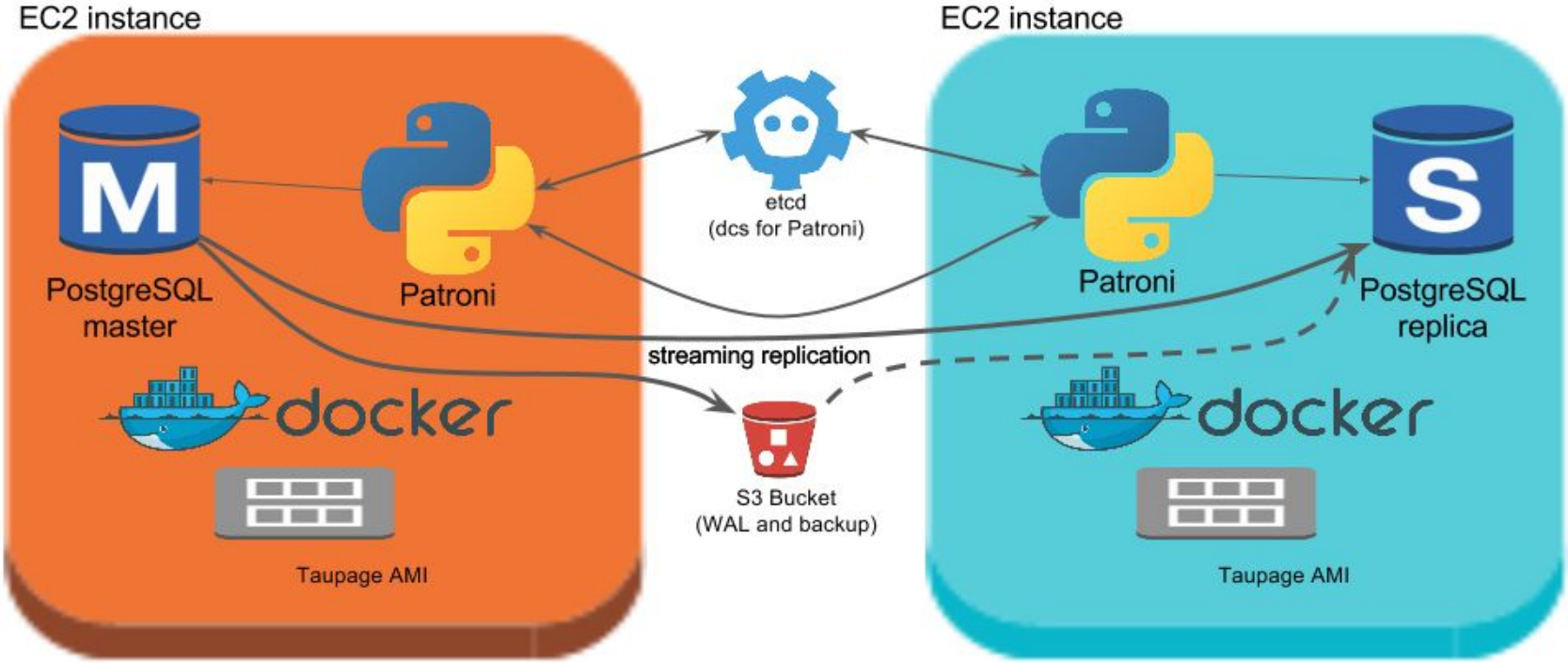
* storage price is not included

** ELB - ~$21/month

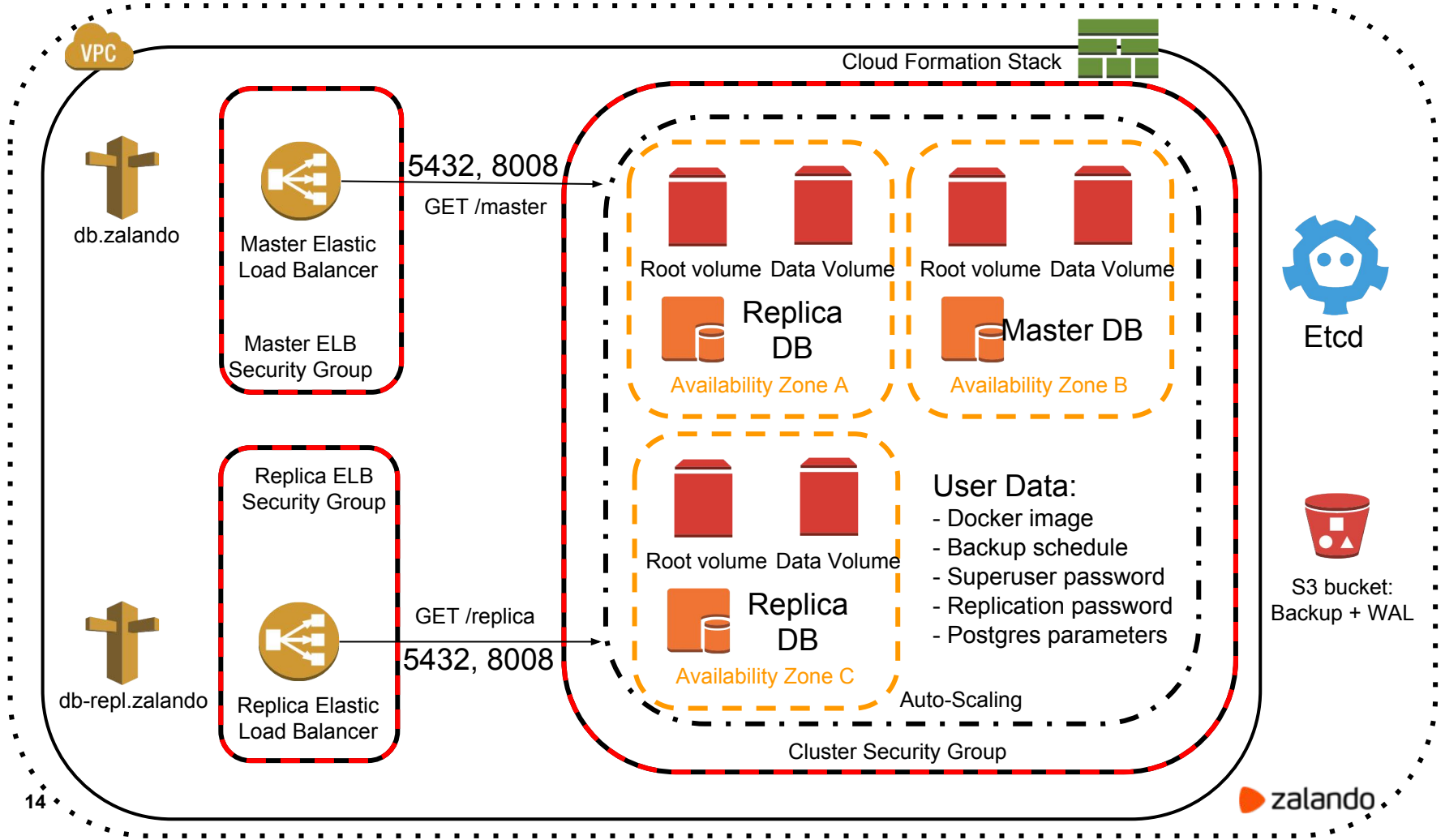
SPILO AND PATRONI

- Patroni - High-Availability and automatic failover
- Spilo - Docker package of Patroni and WAL-E for AWS or Kubernetes
- Use CloudFormation stacks and ASG for deployments
- One Docker container per EC2 Instance
- ELB for traffic routing

AWS DEPLOYMENT



zalando


VPC
Cloud Formation Stack
db.zalando
Master Elastic
Load Balancer
Master ELB
Security Group
5432, 8008
GET /master
Root volume
Data Volume
Replica
DB
Availability Zone A
Root volume
Data Volume
Master DB
Availability Zone B
Etcd
Replica ELB
Security Group
Root volume
Data Volume
Replica
DB
Availability Zone C
User Data:
- Docker image
- Backup schedule
- Superuser password
- Replication password
- Postgres parameters
S3 bucket:
Backup + WAL
db-repl.zalando
GET /replica
5432, 8008
Replica Elastic
Load Balancer
Auto-Scaling
Cluster Security Group

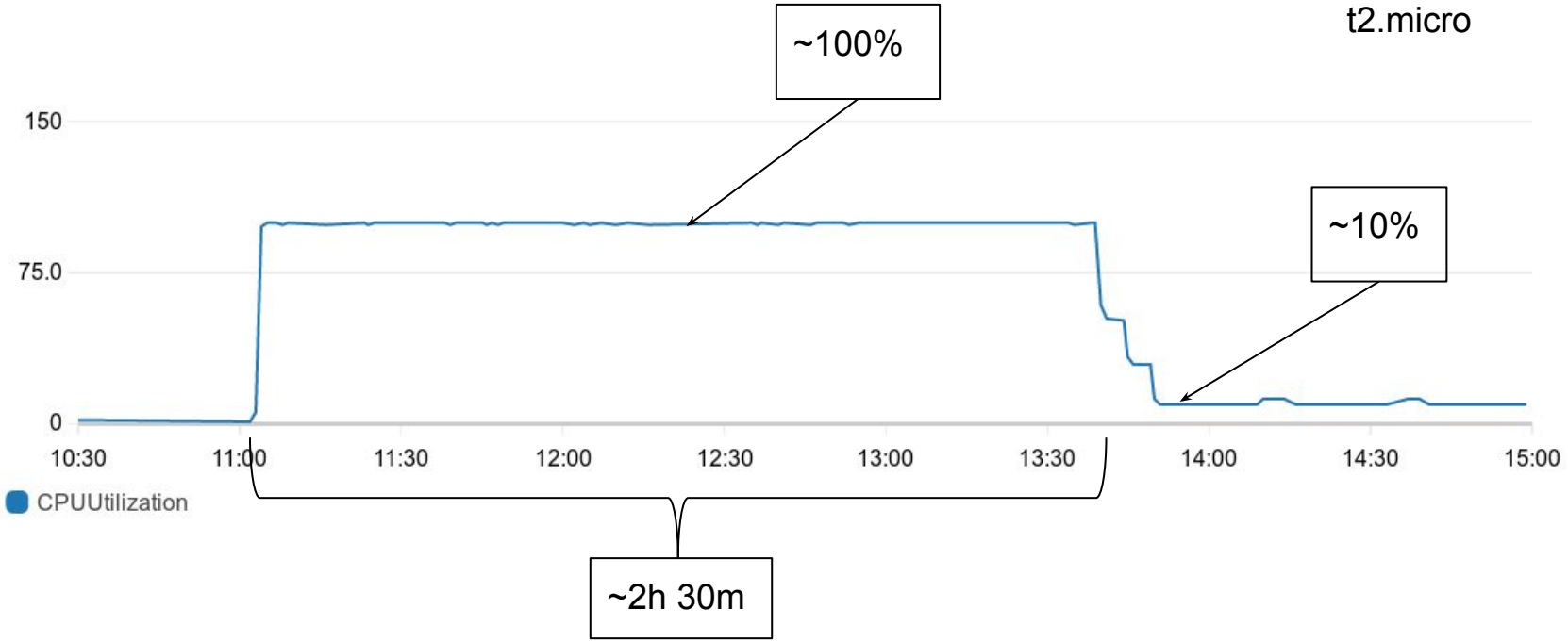
zalando

EC2 INSTANCE FEATURES

- Performance: Fixed vs. Burstable (T2 Instances)
- Storage: Instance Store (Ephemeral) vs. EBS
- EBS-optimized Instances
- Enhanced Networking

BURSTABLE PERFORMANCE OR WHY IS MY DATABASE VERY SLOW?

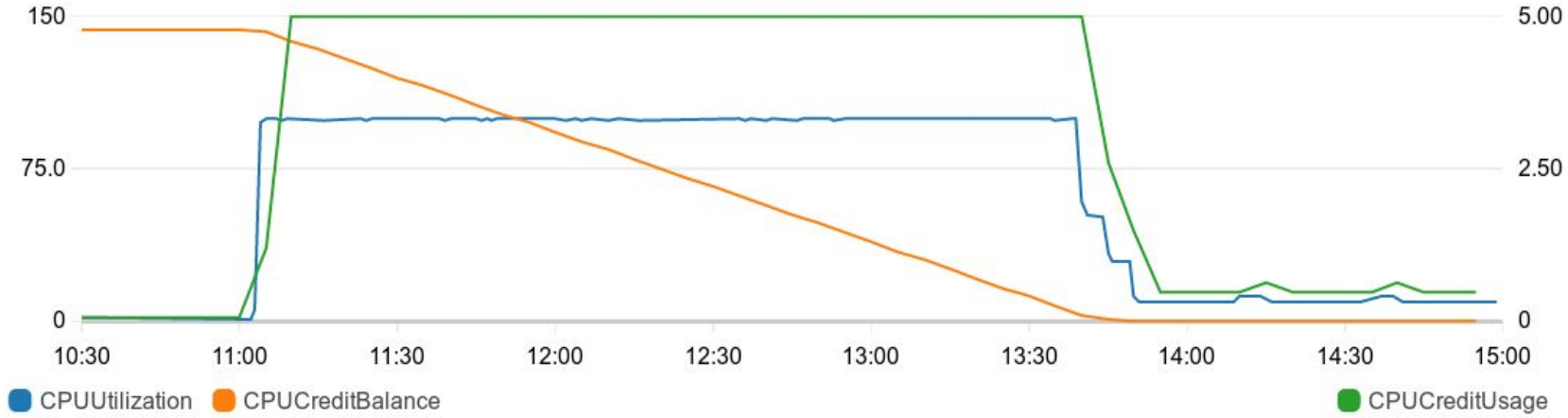
BURSTABLE PERFORMANCE



zalando

BURSTABLE PERFORMANCE

t2.micro



zalando

T2 INSTANCES

- Designed to provide moderate baseline performance and the capability to **burst** to significantly higher performance as required by your workload.
- If your account is less than 12 months old, you can use a t2.micro instance for **free** within certain usage limits.

Instance Type	Initial CPU credit	Credits earned per hour	vCPUs	Base performance (CPU utilization)	Max CPU credit balance
t2.micro	30	6	1	10%	144
t2.small	30	12	1	20%	288
t2.medium	60	24	2	40%	578
t2.large	60	36	2	60%	864

CPU CREDITS

- One CPU credit is equal to one vCPU running at 100% utilization for one minute.
- When a T2 instance uses fewer CPU resources than its base performance level allows (such as when it is idle), the unused CPU credits (or the difference between what was earned and what was spent) are stored in the credit balance for up to 24 hours, building CPU credits for bursting.
- When a T2 instance requires more CPU resources than its base performance level allows, it uses credits from the CPU credit balance to burst up to 100% utilization.

HORROR STORY

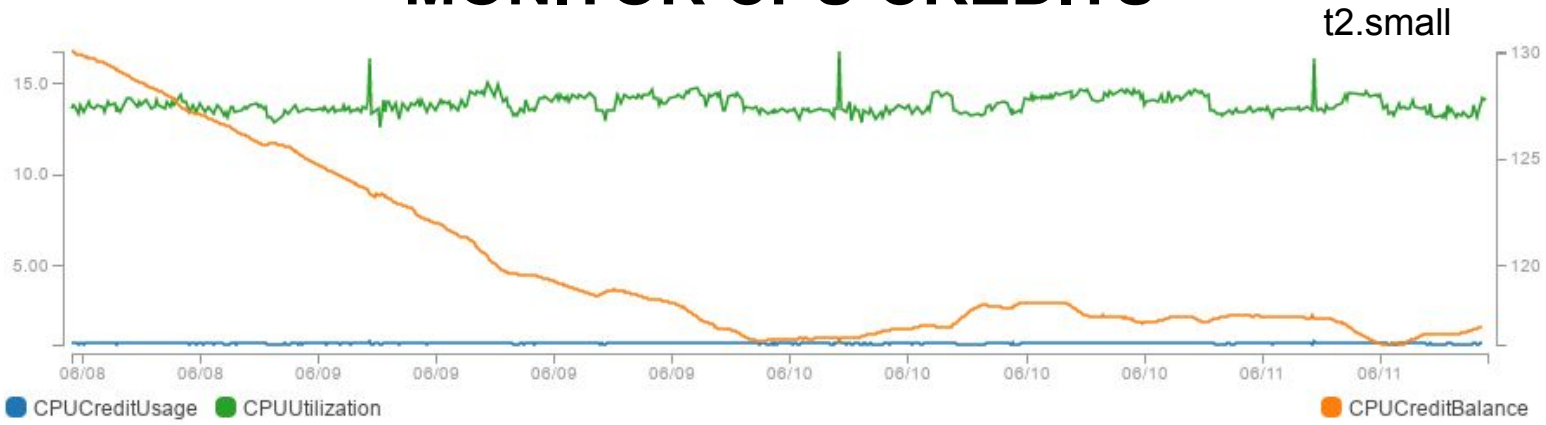
- WAL-E wal-fetch/wal-prefetch is terribly slow
  - Prefetch spawns 8 worker processes (by default)

    and can burn all CPU Credits

How we solved it:

use “-p 0” to disable prefetch

MONITOR CPU CREDITS



- **CPUCreditUsage** metric indicates the number of CPU credits used during the measurement period
- **CPUCreditBalance** metric indicates the number of unused CPU credits a T2 instance has earned

RESERVED INSTANCES

- 1 year or 3 year contracts
- Significant discount compared to On-Demand instance pricing
- Capacity reservation
- Customers using both Reserved and On-Demand instances will have

  Reserved Instance rates applied first to minimize costs

RESERVED INSTANCES

m4.large, Standard 1-Year Term					
Payment Option	**Upfront**	**Monthly**	**Effective Hourly**	**Savings**	**On-Demand Hourly**
No Upfront	$0	$59.79	$0.082	32%	$0.12
Partial Upfront	$342	$28.47	$0.078	35%	
All Upfront	$670	$0	$0.076	36%	
m4.large, Standard 3-Year Term					
No Upfront	$0	$44.17	$0.061	50%	$0.12
Partial Upfront	$736	$20.44	$0.056	53%	
All Upfront	$1383	$0	$0.053	56%	

* Frankfurt region

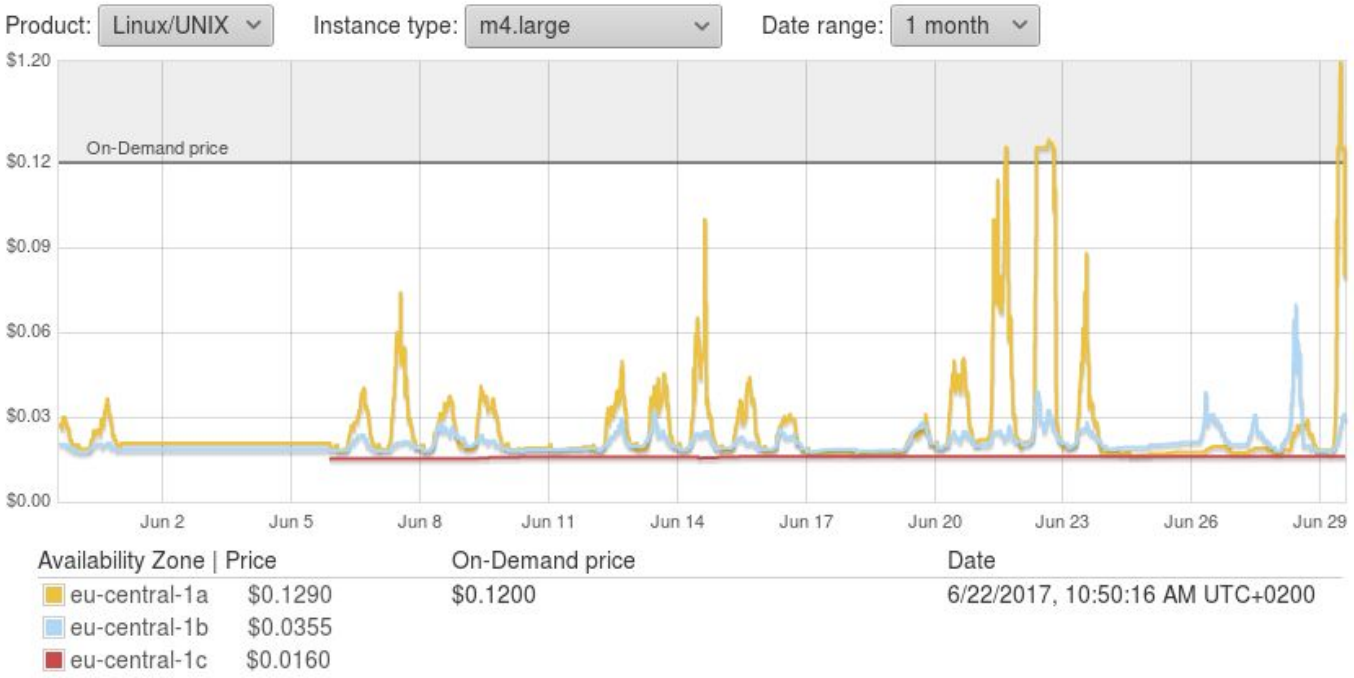
SPOT INSTANCES

- Bidding on “AWS overcapacity”
- Variable price point, save up to 90% vs. on-demand

Risks

Unavailability or loss of instance if outbid!!!

SPOT MARKET



zalando

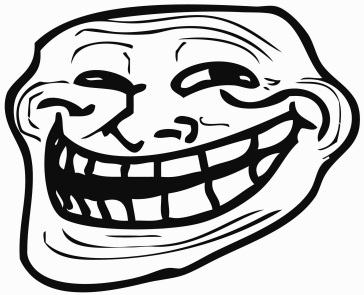
USE CASES

Everything that can fail or be unavailable for short duration

- Async processing
- Reporting
- CI
- Staging systems
- Testing of backups

HORROR STORY

- There were only 2 AZ in eu-central-1 until June 2017
- ASG is trying to do its best to distribute resources across different AZ, but when **Spot Price** is high, it might happen that two EC2 Instances will run in the same AZ
- When price will go down, ASG will rebalance resources, i.e. spawn a new instance in another AZ and terminate one of the running instances
- With 50% probability it will kill the “master”



How we solved it:

AutoScalingGroup -> “TerminationPolicies”: [“NewestInstance”, “Default”]

ELASTIC BLOCK STORE

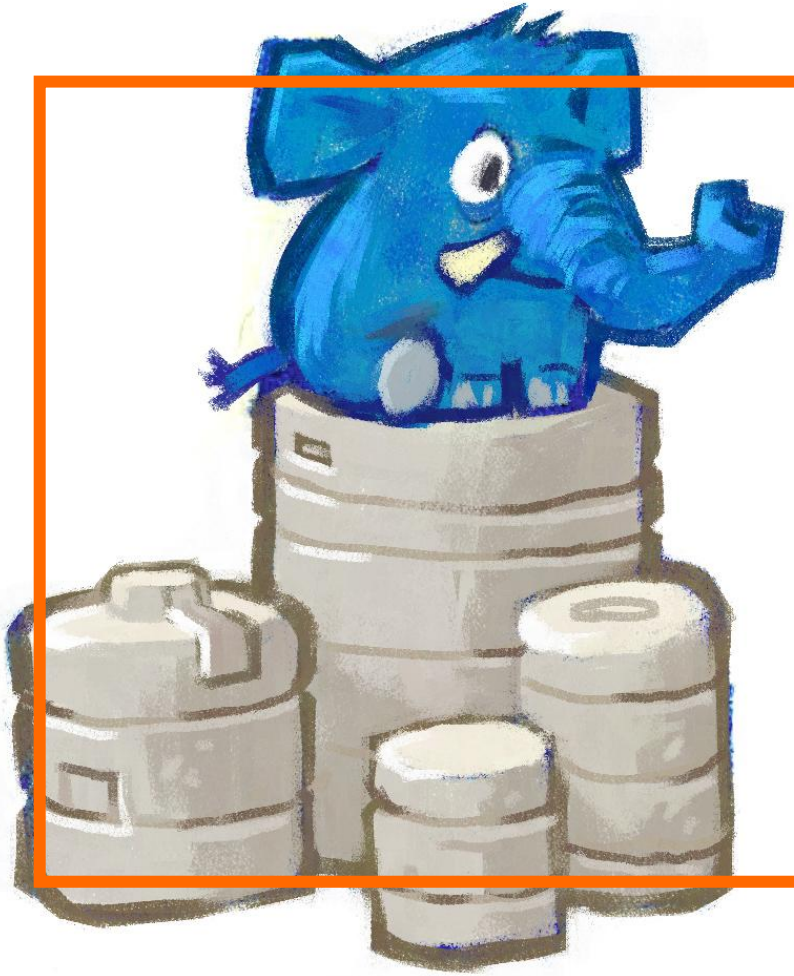
Name	io1	gp2	st1	sc1
Size	4GB - 16TB	1GB - 16TB	500GB - 16TB	500GB - 16TB
Max IOPS/Volume	20000	10000	500	250
Max Throughput/Volume	320MB/s	160MB/s	500MB/s	250MB/s
Price	$0.149/GB-month $0.078/provisioned IOPS	$0.119/GB-month	$0.054/GB-month	$0.03/GB-month

ELASTIC BLOCK STORE

- EBS volumes are created in a specific AZ, and can then be attached to any instances in that AZ
- IOPS and throughput depends on Volume Size
- ST1 and SC1 - HDD, good throughput, but low IOPS
- GP2 and IO1 - SSD, usually good choice for databases

IO1 VS. GP2

- IO1 gives better IOPS guarantees
- But you have to pay for it – the same size costs 25% more. Plus you have to pay for provisioned IOPS
- For RDS minimum size of IO1 volume is 100 GB + 1000 provisioned IOPS. It will cost you $13.80 + $110 per month
- You can get 1000 IOPS with 334 GB GP2 volume for $42/month



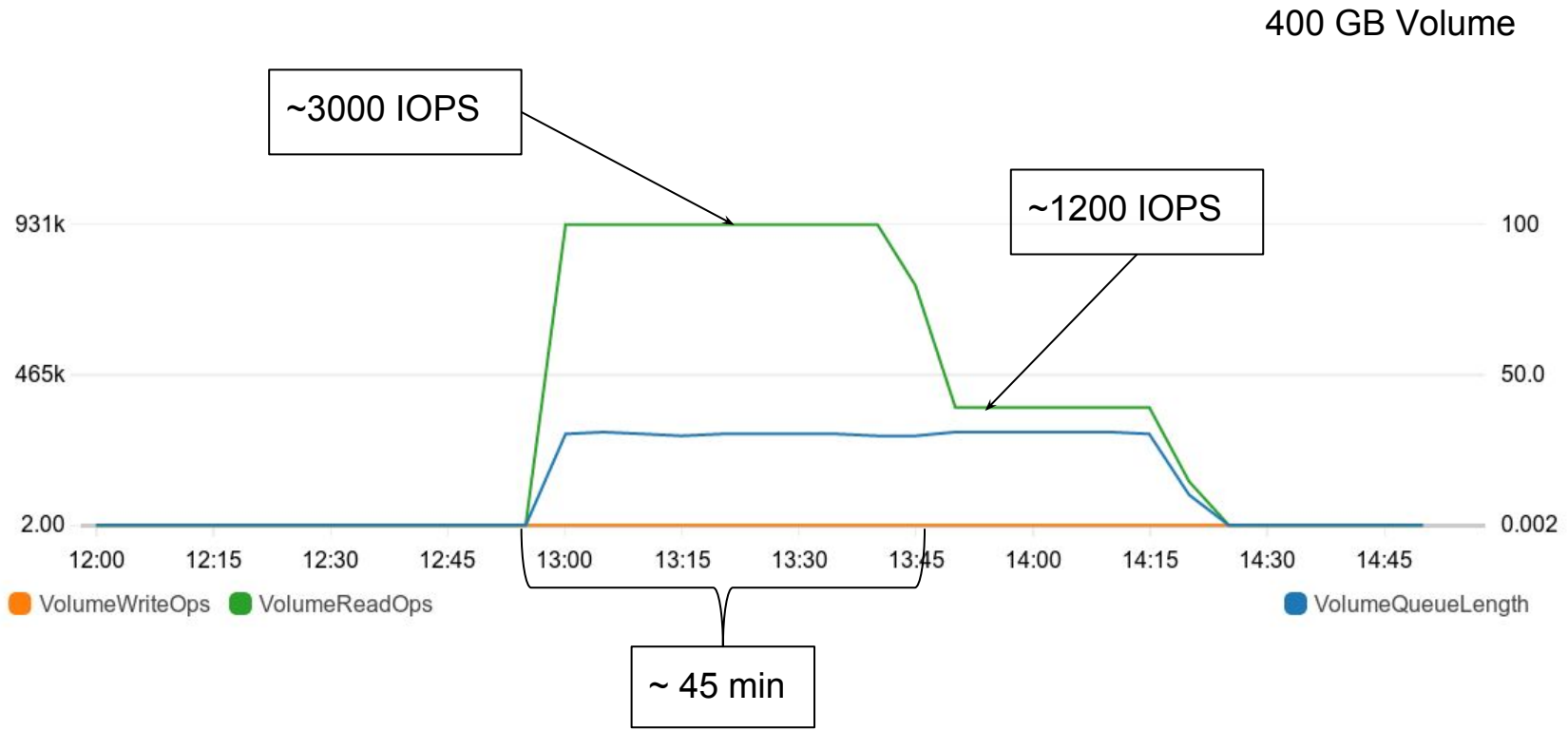
SEQUEL:

BURSTABLE PERFORMANCE

OR WHY IS MY DATABASE

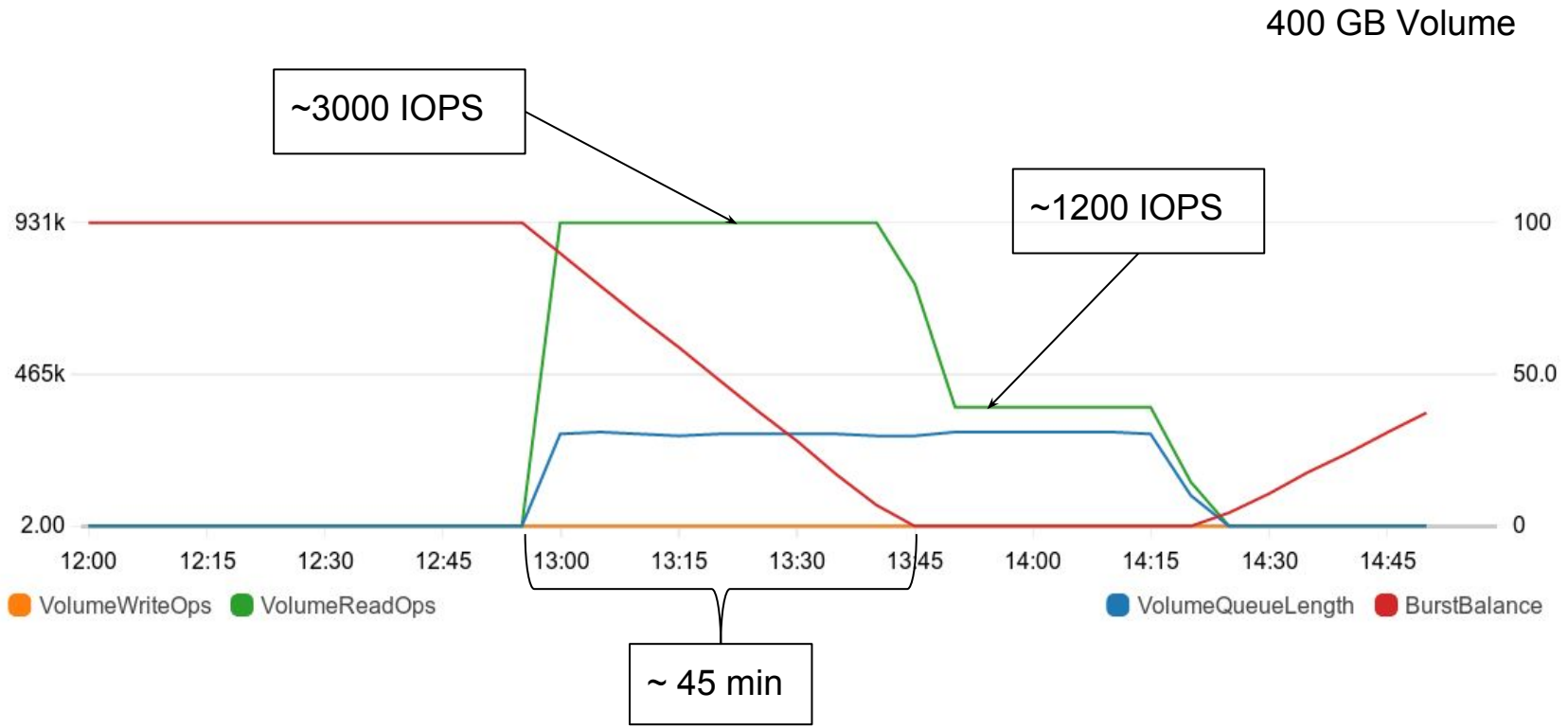
SLOW AGAIN?

BURSTABLE PERFORMANCE



zalando

BURSTABLE PERFORMANCE

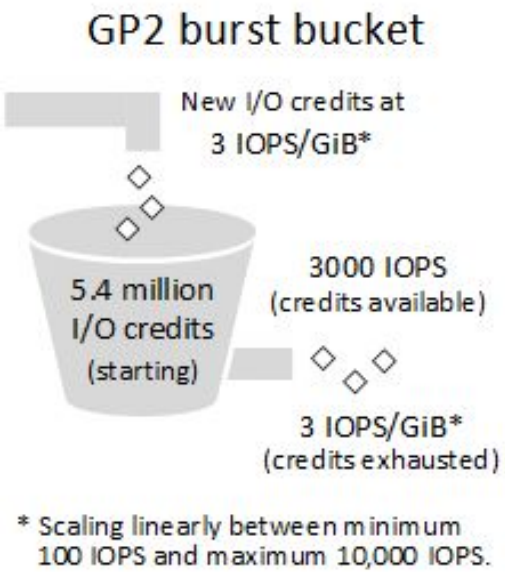


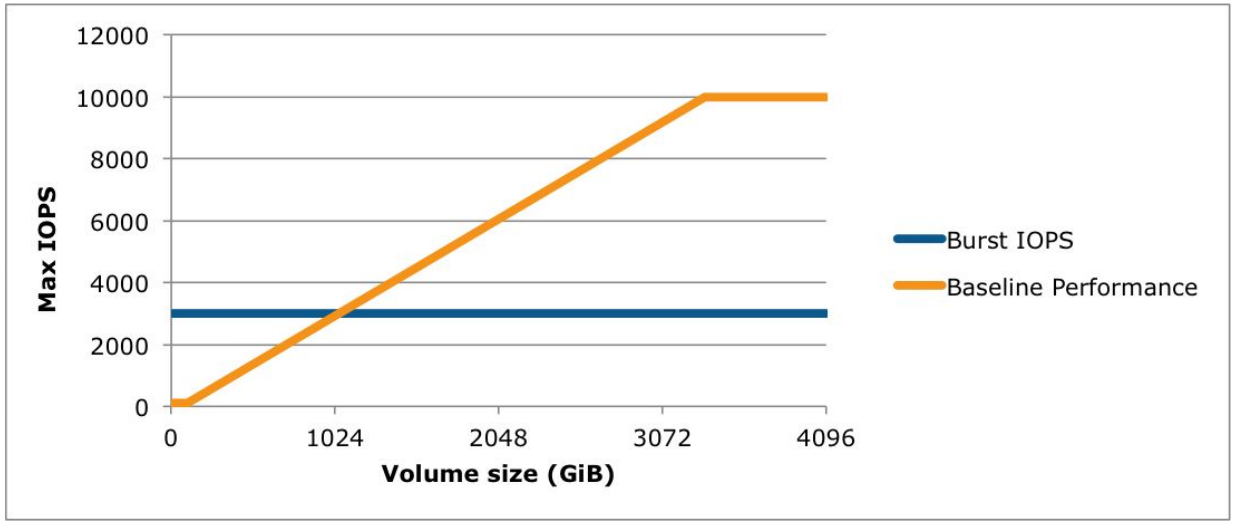
zalando

I/O CREDITS AND BURST PERFORMANCE

- The performance of gp2 volumes is tied to volume size (3 IOPS/GiB)
- The maximum and initial I/O credit balance for a volume is 5.4 million
- When your volume requires more than the baseline performance I/O level, it draws on I/O credits in the credit balance to burst to the required performance level, up to a maximum of 3,000 IOPS
- When your volume uses fewer I/O credits than it earns in a second, unused I/O credits are added to the I/O credit balance

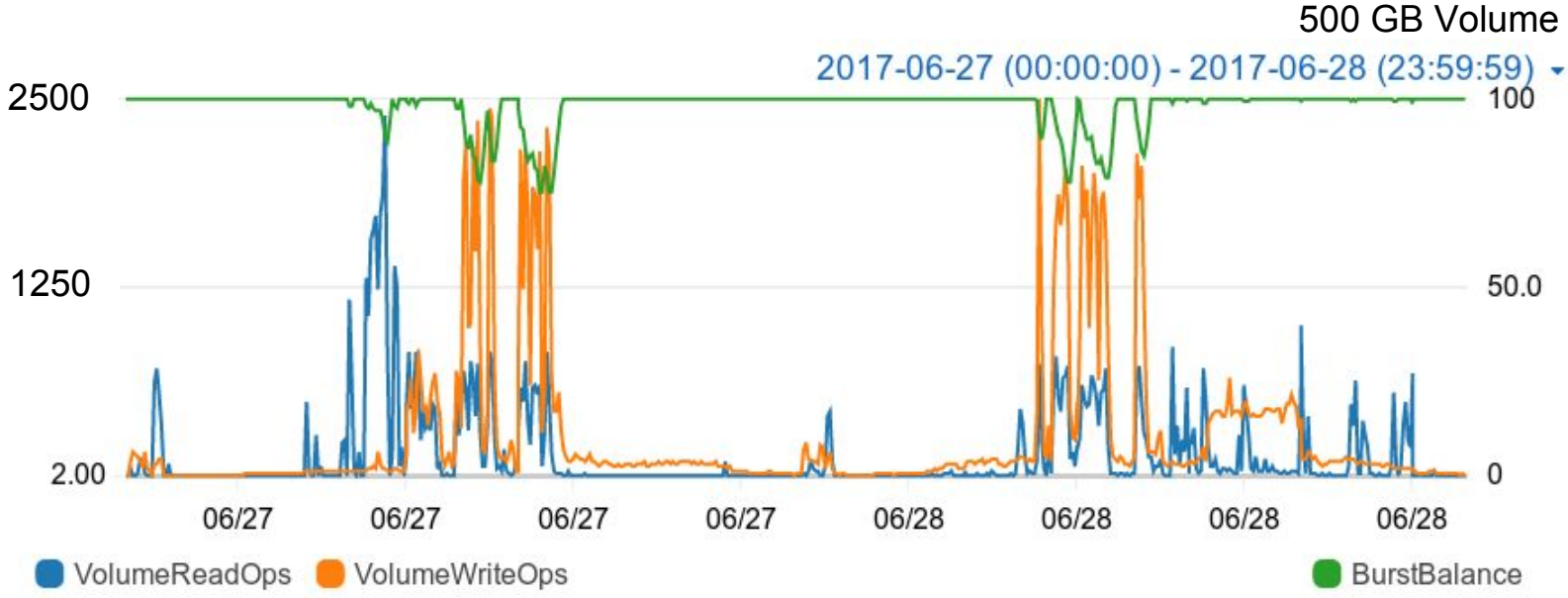
GP2 VOLUMES EXPLAINED





zalando

MONITOR I/O WITH CLOUDWATCH



- Especially **BurstBalance** if GP2 volume is smaller than 1TB

zalando

EBS-OPTIMIZED INSTANCES

- Dedicated bandwidth to Amazon EBS, with options between 500 Mbps and 12,000 Mbps, depending on the instance type you use

- Provides the best performance for your EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance

EBS-OPTIMIZED INSTANCES

InstanceType	vCPU	Memory	Max IOPS	Throughput (Mb/s)	Price (per month)
m4.large	2	8 GB	3600	56.25	$87.6
r4.large	2	15 GB	3000	54	$116.8
m4.xlarge	4	16 GB	6000	93.75	$175.2
r4.xlarge	4	30 GB	6000	109	$233.6
m4.4xlarge	16	64 GB	16000	250	$700.7
r4.4xlarge	16	120 GB	18750	437	$934.4

EBS TIPS

- GP2 is MUCH cheaper than IO1
- To get more than 10000 IOPS or 160MB/s with GP2 - build a RAID-0 from multiple volumes
- Choose an EC2 Instance with enough bandwidth

HORROR STORY

- AWS didn’t provide a means to monitor EBS Burst Balance until November 2016
- RDS still doesn’t provide information about GP2 Volume Burst Balance

INSTANCE STORE VOLUMES

- Pros:
  - Located on disks that are physically attached to the host computer
  - Amazing throughput and latencies compared to EBS
- Cons:
  - Provides only **temporary** block-level storage
  - Data in the instance store is lost under the following circumstances:
    - The underlying disk drive fails
    - The instance stops or terminates

PRICE COMPARISON

InstanceType	m4.xlarge	r4.xlarge	i3.xlarge	m4.2xlarge	r4.2xlarge	i3.2xlarge
vCPU	4	4	4	8	8	8
Memory	16 GB	30 GB	30 GB	32 GB	60 GB	60 GB
Max IOPS	6000	6000	6000	8000	12000	12000
Throughput (Mb/s)	93.75	109	100	125	218	200
Instance Storage (NVMe)	-	-	950 GB	-	-	1900 GB
Price (per month)	$175.2	$233.6	$271.56	$350.4	$467.2	$543.12
Price with 950 GB EBS	$288.25	$346.65		$463.45	$580.25	
Price with 1900 GB EBS	$401.3	$459.7		$576.5	$693.3	

INSTANCE STORE SUMMARY

- For high-intensive OLTP, i3 instances are a rescue:
  - r4.2xlarge + 3.3TB EBS ($863.75/month) wasn’t able to keep up
  - The switch to i3.2xlarge solved all problems and saved 37% of costs

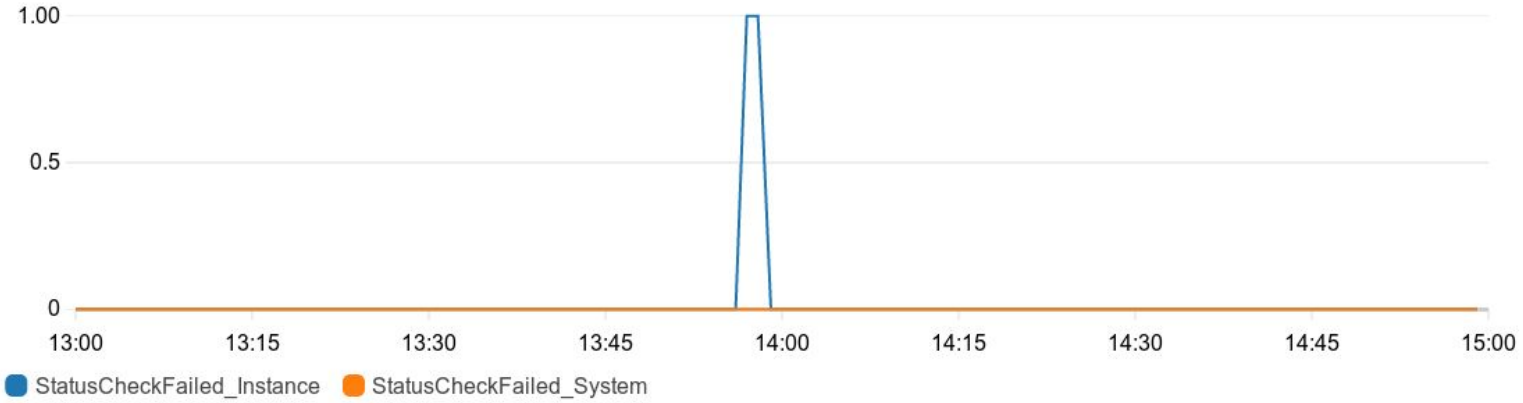
No horror stories with i3 instances involved so far ;)

ENHANCED NETWORKING

- Enhanced networking uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on supported instance types
- SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualized network interfaces
- Enhanced networking provides higher bandwidth, higher packet per second (PPS) performance, and consistently lower inter-instance latencies

HORROR STORY

- Heavy loaded M4 instances periodically losing network
- Heavy loaded R4 instances were simply dying (terminated by AWS)
- The only visible indicator of a problem was “StatusCheckFailed” CloudWatch metric



zalando

HORROR STORY

- It took us several days of investigating:
  - Thank you AWS support for pointing to an outdated **ixgbevf** driver
  - The rest we figured out on our own
- Our AMI is built on Ubuntu 14.04:
  - Ubuntu 14.04 has an outdated driver **ixgbevf**
  - Ubuntu 14.04 doesn’t have an **ena** driver

ENABLING ENHANCED NETWORKING

1. AMI must contain drivers (linux kernel module)
    a. **Ixgbevf** for C3, C4, D2, I2, R3, and M4 (excluding m4.16xlarge)
    b. **ena** for F1, I3, P2, R4, X1, and m4.16xlarge
2. You have to explicitly enable enhanced networking for an AMI or a specific instance. If you use AMI from AWS it's already enabled.

INTEL 82599 VF

- Verify that the *ixgbevf* module is installed

```
[ec2-user ~]$ modinfo ixgbevf
filename:       /lib/modules/3.10.48-55.140.amzn1.x86_64/kernel/drivers/amazon/ixgbevf/ixgbevf.ko
version:        2.14.2
```

- Check that support is enabled on Instance level

```
$ aws ec2 describe-instance-attribute --instance-id instance_id --attribute sriovNetSupport
```

- Or support is enabled on AMI level

```
$ aws ec2 describe-image-attribute --image-id ami_id --attribute sriovNetSupport
```

If the attribute isn't set, SriovNetSupport is empty; otherwise, it is set as follows:

```
"SriovNetSupport": {
    "Value": "simple"
},
```

ENA

- Verify that the *ena* module is installed

```
[ec2-user ~]$ modinfo ena
filename:       /lib/modules/4.4.11-23.53.amzn1.x86_64/kernel/drivers/amazon/net/ena/ena.ko
version:        0.6.6
```

- Check that support is enabled on Instance level

```
$ aws ec2 describe-instances --instance-id instance_id --query 'Reservations[].Instances[].EnaSupport'
```

- Or support is enabled on AMI level

```
$ aws ec2 describe-images --image-id ami_id --query 'Images[].EnaSupport'
```

If the attribute is set, the response is **true**

VERIFY THAT THE MODULE IS IN USE

```
[ec2-user ~]$ ethtool -i eth0
driver: vif
version:
firmware-version:
bus-info: vif-0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

```
[ec2-user ~]$ ethtool -i eth0
driver: ixgbevf
version: 2.14.2
firmware-version: N/A
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: no
supports-register-dump: yes
supports-priv-flags: no
```

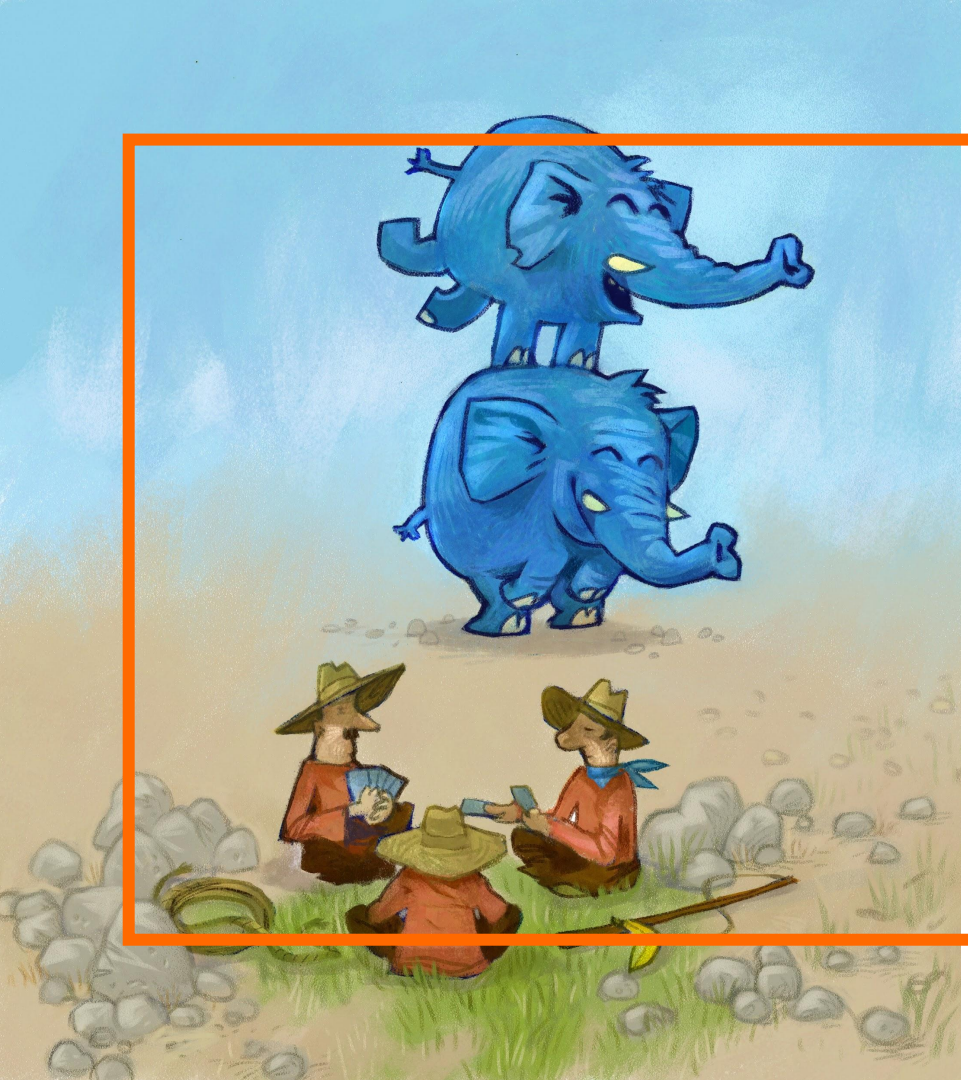
```
[ec2-user ~]$ ethtool -i eth0
driver: ena
version: 0.6.6
firmware-version:
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

SUMMARY

- Start with small Instances and Volumes, it’s easy to scale up later
- Some of the resources (**CPUCreditUsage**, **CPUCreditBalance**, **BurstBalance**) it’s possible to monitor only with CloudWatch
- Always do backups and test them

USEFUL LINKS

- Easy Amazon EC2 Instance Comparison - EC2instances.info
- Easy Amazon RDS Instance Comparison - RDSInstances.info
- Simple monthly calculator - calculator.s3.amazonaws.com/index.html
- Patroni - github.com/zalando/patroni
- Spilo - github.com/zalando/spilo



QUESTIONS?