MySQL In the Cloud

Migration, Best Practices, High Availability, Scaling

Peter Zaitsev

CEO

Percona University, Budapest

May 11th, 2017

PERCONA

Let me start….

With some Questions!

PERCONA

Question One

How Many of you are running MySQL In the Cloud ?

PERCONA

Question Two

Are you running it in Public Cloud ? Private Cloud ? Both ?

Question Three ?

Are you using DBaaS such as Amazon RDS or Google CloudSQL or running your own ?

Question Four

Are you using Containers ? (Such as Docker)

PERCONA

Lets Cover some Basics

What is “Cloud”

Dynamic Programmable Infrastructure

Public and Private

Public	Private
• Infrastructure Shared with other Users • Amazon AWS typical example	• Infrastructure Private for company • OpenStack installation typical example

PERCONA

All of those XaaS

IaaS (Infrastructure as a service)

- Works in Infrastructure level: “Compute”, “Storage”, “Network”
- Examples: AWS EC2, S3, EBS

DBaaS (Database as a service)

- Provides Database Service (Instances or Clusters) to use
- Examples: Amazon RDS, Google Spanner

PaaS (Platform as a service)

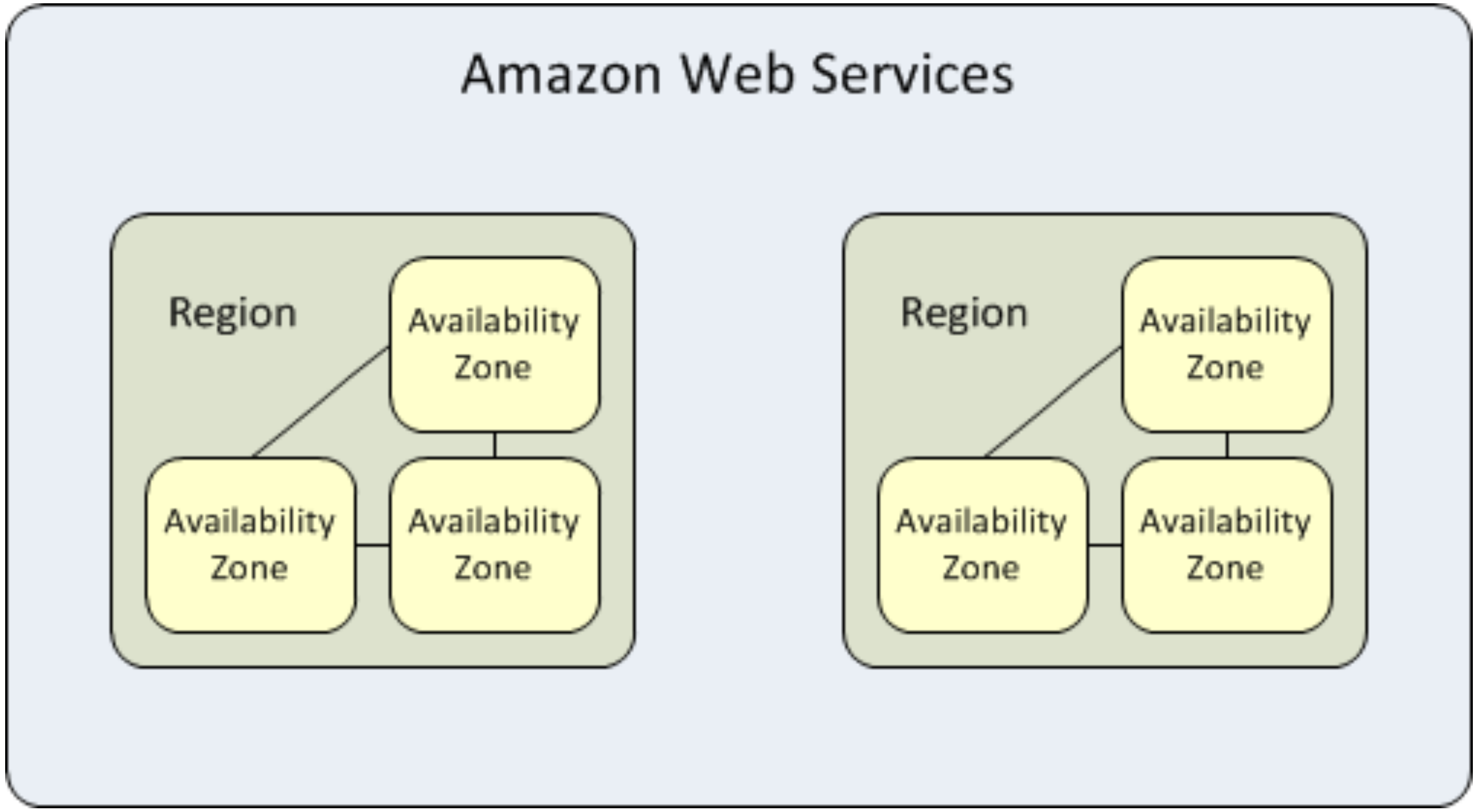
- Provides full platform for your application development
- Examples: Heroku, Amazon Elastic Beanstalk, OpenShift

PERCONA

Regions and Availability Zones

Region	Availability Zone
• Specifies Geographic Region • Hierarchy - North America – West – California • High Latency between Regions • Complete Isolation	• Is located in the region • Reasonably isolated from each other • Medium Latency between AZ

Making it Visual



Source: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

Top Cloud Providers

amazon web services

Google Cloud Platform

Microsoft Azure

PERCONA

Technologies to be aware of

openstack®

kubernetes

OPENSHIFT

docker

Decisions to Make

Should you move to the cloud ?

This is decision you rarely have

Programmable infrastructure is the future

Virtualization overhead is going down

Some clouds providers support Bare Metal

PERCONA

Public ? Private ? Hybrid

Public Cloud	Private Cloud	Hybrid Cloud
• Agility • Scalability • Costs • Small and Medium Businesses	• Control • Costs • Legacy Integration • Some Enterprise Companies	• Infrastructure using Both • Can get benefits of both • At the cost of extra complexity

PERCONA

Single vendor vs Multi Vendor

Single Vendor	Multi Vendor
• Use all features vendor has to offer • Danger of Vendor Lock In	• Have to use “lowest common denominator” • Avoid Vendor Lock In

DBaaS

DBaaS (ie Amazon RDS)	IaaS (ie EC2+EBS+S3)
• Easier • Takes off some operational pains • Less Flexible • More Expensive • More Lock-In	• Harder to roll your own • Operations on your own (or your partner) • More Flexible • Less Expensive • Less Lock-In

PERCONA

Open Source in the Cloud

Open Source Compatible is not same as Open Source

PERCONA

Migration

Keep it Simple

Do not try doing upgrade at the same time as migration

Exactly same minor version is optimal

Same major version - must

PERCONA

Moving to IaaS Cloud

General Practices as in Datacenter Migrations apply

Easy to use Binary Backups

Slave_compressed_protocol or compression in VPN

Support utilities may need to be modified for EBS/S3

PERCONA

Moving to DBaaS

Need to use database dump to copy

- Mysqldump
- Mysqlpump
- Mydumper

Can set external slave (Amazon RDS)

- CALL mysql.rds_set_external_master

Monitoring Backup may need revision

- Do not have direct access to physical box
- Do not have root user

New With Amazon RDS Aurora

Can use Percona Xtrabackup’s Backup to seed the cluster

http://amzn.to/2pk6lq7

PERCONA

Moving from DBaaS

Logical Database dump as well

Replication supported for Migration only

Configure Binary Log Retention
mysql.rds_set_configuration

Best Practices

Being Cost Efficient

- Know your cloud vendor pricing policies
- Look beyond “compute” pricing
- Best Price/Performance configuration in the cloud is likely to be different
- AWS: Reserve Instances
- AWS: Spot Instances

PERCONA

Guarantee versus Burst

Guaranteed	Burst
• Performance resource is "guaranteed" to have in worst case scenario • This is what you can plan for	• Performance resource can provide • Typically not guaranteed • Typically limited in length to prevent abuse

Network

Understand Application-Database Network Latency

Same AZ Optimal; Same Region Must have

10Gb Network

Understand network “jitter”

Latency is critical for most applications

Bandwidth can be important for dumps and batch job

CPU

Same whenever you’re in the cloud or not

MySQL uses single thread for single query

Multi-Core gives good scalability for “Web” workloads

Memory

Use mainly as a cache

Very important for Performance

PERCONA

Storage

Instance Local Storage	Cloud Block Storage	File/Object Storage
• May or may not be available • Not Highly available • May be inexpensive and high Performance	• Reliable • Remote • Separately Prices • EBS on AWS	• Store Files/Objects • No interactive block level access • S3 on AWS

Things to Note

You can't get any combination

EBS Performance depends on the instance size

Provisioned IOPs for Optimal Performance

Glacier storage for old backups

PERCONA

Operating System

Consider Cloud Optimized Linux Versions

At very least use Recent Linux Versions

“Cloud Only” Linux might be inconvenient for development

MySQL Version and Configuration

Use Recent Version

Do not count on good Defaults

PERCONA

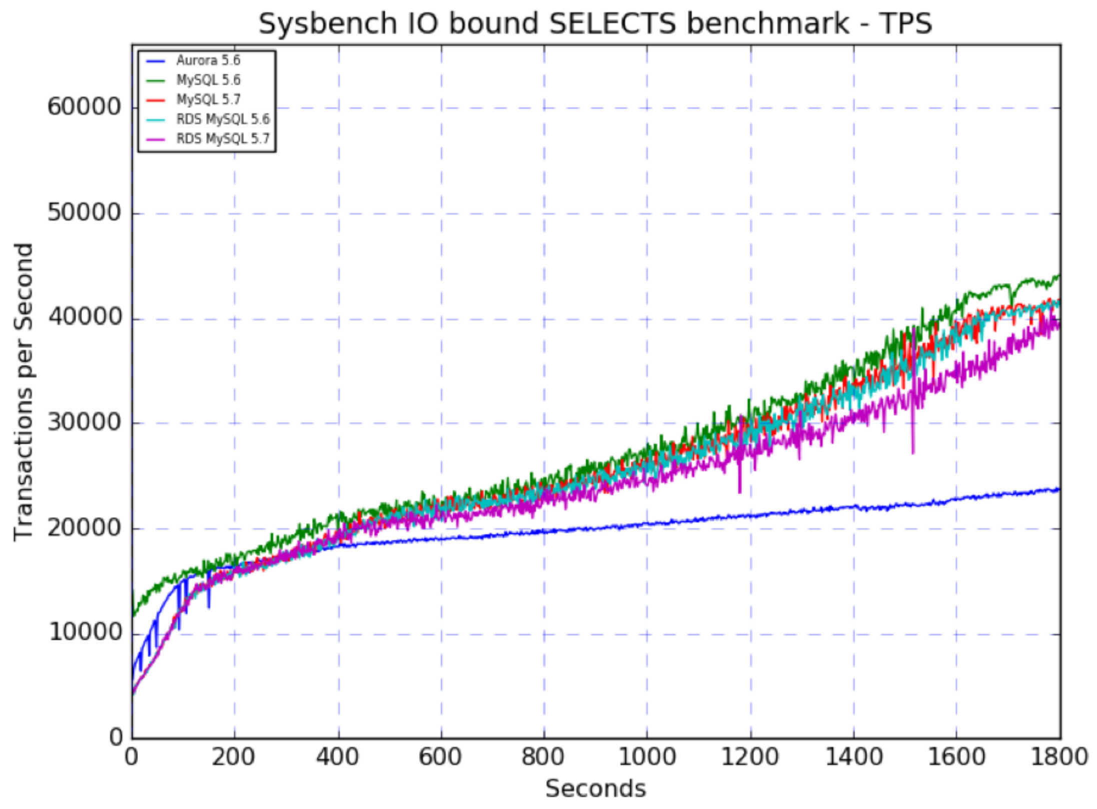
DBaaS

There is essentially same systems underneath!

Most of same practices Apply

PERCONA

DBaaS is not always faster



Source: https://twindb.com/rds-vs-aurora-vs-ec2-benchmark/

High Availability

Your Choices

Roll your own

Use DBaaS

Things to Consider

You have less control or visibility into the infrastructure

Things as IP take –over might not work

Load Balancers

Cloud Load Balancer (Elastic Load Balancer at AWS)

HAProxy

ProxySQL

Maintaining copies of Data

MySQL Replication

MySQL Group Replication

Percona XtraDB Replication (PXC) and Galera

PERCONA

Why Percona XtraDB Cluster in the Cloud

Read/Write to any node works great with simple load balancers

Automatic Provisioning and Auto Scaling

Can run with local instance storage

Can deploy across multiple AZ

Scaling

Scalability in the Cloud

“Better”	“Worse”
• Due to cloud optimized options like Amazon Aurora	• Due to restricted hardware choices

PERCONA

Scaling How

Scale Up	Scale Out
• Vertical Scaling • Scale with the Harware Size – CPUs, Memory, Storage	• Horizontal Scaling • Scale by adding nodes

PERCONA

Bad reputation of Scaling Up… but

Reasonable commodity MySQL Server Can handle

- 3-5TB database size
- 100K+ queries/sec
- 5M+ rows read/sec
- 100K rows modified/sec

PERCONA

Scaling What ?

Reads

Writes

Data Size

Scaling Reads

Replication

Caching

Moving some load from MySQL

Scaling Writes

- New MySQL Versions
- Parallel Replication
- TokuDB
- Functional Partitioning
- Sharding

New in Sharding

ProxySQL

Vitess

PERCONA

Scaling Data Size

Functional Partitioning and Sharding

Data Archiving

TokuDB for Compression

Often Operations drive this needs not App Performance

Database Performance Matters