**Multi-tenancy on cloud sql**

1st slide :

introduction to the company CHDS : Clement Huge (25+) / Naginder Singh (25+) / Amit Singh / Sandeep Kumar / Mick OBrien / Antonio Cardosio / Bhuvnesh Gora …   
 Certifications on all hyperscales : AWS, Azure, Google Cloud  
 Additional certification : ISOSEC 27001 certifified auditor  
 Areas of expertice : data engineering, data management, system designs, architecture, compliance, auditing isosec27001, PCI-DSS, GDPR  
 Consulting partners : freelance.com, webapper, go2ria, Transputec, parrallo…  
 ISV partnerships : Cozyroc, RegData

Industry verticals :   
 Banking, Payment, Medical, Service, Manufacturing, Retail, Luxe, Telco, Saas, AdTech, IT

2nd / 3rd slide :

Cloud SQL in numbers  
 Cloud SQL main functionalities :   
 Relational Db engines : Mysql, PgSQL, MSSQL  
 Elasticity for compute, memory, storage (autoscaling)  
 Interoperability with Google cloud services and other managed services  
 Integrated security and networking : IAM, firewalling, cloud sql proxy, public and private ips, SSL certificate, Data at rest  
 HADR : High availability with end point management, cross-regional read replica  
 Automated provisioning including terraform compatiblity  
 Automated backup and recovery : automatic failover, cloning, backup  
 Automated maintenance for OS, minor versions and in-place upgrade

4th slide :

What is tenancy ? What is not tenancy ?

Scope of the session : relational database only !   
 other nosql models can answer tenancy

5th Slide :

Tenancy topologies ?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Isolation** | **Seggregation** | **Compliance** | **SLA** | **COST** | **Flexibility** |
| Rows/Partitions | Very weak | Weakest | Cumbersome | Cheapest | Lowest |
| Table | Weak | Weak | Difficult | Cheap | Low |
| Database | Neutral | Neutral | Fair | Neutral | Neutral |
| Instance | Good | Good | Good | Expensive | Good |
| Project | Very good | Very good | Easy | Expensive | High |
| Organization | Excellent | Excellent | Easiest | Expensive | Highest |

Multi-tenancy hybrid topology : we will do a focus on it.

7th slide : How does cloud SQL help you to accomplish multi-tenancy ?

We will cover : setup and maintenance, compliance and client seggregation requirements, SLAs, TCO/Cost consolidation, security

8th slide : Setup

Setup via console, gcloud, api rest, terraform apply

* Allow organization to label database instance properly, setup the right server settings, the right sizing
* Does not allow organization yet to implement from terraform SQL scripts (workaround ?)
* Specific concerns to consider for multi-tenancy at this stage :   
  security (authentication), memory management and compute, collation, timezone, region and zone settings   
  HADR consideration : HA ; Read replicas based on tenant tier  
  Backup consideration : backup and PITR based on tenant tier  
  tenant as label ? tenant tier as label ? client locality as label, SLA as Label, Saas version as label ?
* Demo video ? Show how terraform can be applied to create a cloud sql instance

9th slide : Maintenance

* Managed maintenance on OS and minor versions
* Choose deny-maintenance based on timezone and criticity
* Self-managed maintenance possible to catch up (due to deny period)
* Self-managed index management, data corruption within database

Multi-tenancy enable management of different maintenance windows based on client locality and seggregation. Beware of multi-tenancy within same database !

10th slide : Compliance  
 - GDPR : clients under GDPR should be under European region / Assured workload

* Sovereignty : consider specific GCO cloud partnership : S3NS in France, T-Systems in Germany…
* Client seggregation : due dilligence done by client, different application login, different connection….
* Reduce footprint of compliance by isolating workloads and tenants/ regulations

11th slide : SLAs

* Tenant tiering : provide smaller instance size for trial clients, automate import/export tenant databases, provide increased reliability
* Provide different Support level per criticity of instances

12th slide : TCO

* Tenancy tiering : dedicate instance for big tenants vs consolidate multiple tenants per database, standard vs entreprise MSSQL editions, Cloud SQL vs Alloydb for PGSQL, define different RPO/RTO per tier
* Consolidate tenants per database / instance thanks to cloud elasticity (increase ressource as needed or add new instances)
* Consolidate with bigger instances than onPrem and reduce overal cost of hardware and licence by multualizing cost and reducing overal TCO.
* Consider SMT off for large instances after such consolidation work to reduce even more licence cost.

13th slide : focus on hybrid tenancy

multiple instances with multiple databases with multiple tenants per table partitions.

Avantages :

Tenancy tiering is possible : all tenants do not require same functionalities, same SLAs, same compliance, same seggregation…

Disadvantages :

Tenancy maintenance could be much harder as you share tables on multiple tenants.   
 Index keys are composite and harder to implement  
 Lock escalation can be an issue and data distribution can skew exécution plans. PSOP can help prevent it but only to some extent.

Resharding tenants are harder