school of continuing studies | YORK UNIVERSITÉ UNIVERSITY

DRaaS Reflection Paper

Professor - Claude Sam-Foh

Date – 10th August, 2025

CSCL1060 Business Continuity in the Cloud

Group 6

261 Xuanyu Zhu

I did a comparison of disaster recovery on AWS, Azure, and Google Cloud. My conclusion is to make Google Cloud (GCP) the first choice for most new web, API, and data workloads. GCP has a strong global network, simple traffic failover, realistic backup and restore, and comprehensible cost controls. These are the items that matter most during an outage when the team must act fast. Azure and AWS are both possible, but I only bring them in for edge cases. By default, I'd architect a GCP-first DR.

For Windows and Active Directory, GCP solves the must-have requirements without fuss. Google offers a Managed Service for Microsoft AD, first-class Windows Server images, and integration so Cloud SQL for SQL Server can use Windows Authentication. This translates to practicality in that I can domain-join instances, keep Kerberos/NTLM flows, and enforce group policies as usual. I don't lose core identity functionality just because I've chosen GCP. Azure is naturally close to Microsoft, but for DR planning, GCP's Microsoft AD and SQL integration are already good enough to unblock a GCP-first approach.

In global reach, raw region counts don't tell the whole story. What minimizes recovery time is how fast traffic can divert when a region fails. GCP has an edge here. Google's worldwide HTTP(S) Load Balancer uses a single anycast IP address for all regions. When one region fails, traffic can divert to a healthy region behind the same IP address. There is no manual DNS shuffle and no need to wait for DNS TTL to expire. When I do need DNS-level control, Cloud DNS offers health-checked failover. This design eliminates steps from the runbook, makes drills simpler, and generally reduces RTO because the network gets out of the way.

For compliance, GCP covers the major standards that the majority of teams need for DR, including ISO families, SOC reports, PCI, and FedRAMP. For real projects, this portfolio is usually sufficient to pass an audit and respond to internal control maps. If a niche or regional attestation is the only deciding factor, I would compare catalogs among providers. For most DR programs, however, GCP's compliance coverage is not a deal-breaker and enables the project to move forward.

For recovery speed and ease, GCP gives me building blocks that fit together nicely. Cloud Backup and DR (through the Actifio acquisition) unifies protection and recovery. It delivers agentless snapshots for Compute Engine and for vSphere-based VMs, changed-block tracking to reduce data movement, and application-aware recovery. It is also supported by Google Cloud VMware Engine. That is, I can easily protect common workloads and restore them without writing heavy custom glue. When I pair this with Compute Engine, Cloud Storage, Cloud SQL, and global load balancing, I get a DR pattern that's not only flexible, but easy to test. I can automate the runbook using Terraform for infrastructure and Cloud Scheduler and Cloud Functions (or Cloud Run) for scheduled checks, integrity testing, and failover exercises.

GCP is also strong at the data layer that tends to drive RPO and RTO. Cloud SQL provides high availability and cross-region replicas for quicker cutovers. Spanner provides multi-region, strongly consistent databases when I need global write availability. BigQuery natively supports snapshotting and time-travel style recovery for analytics data. Filestore and NetApp volumes support snapshots for file workloads. Cloud Storage offers Object Versioning and retention controls for immutable backups. These services all reduce the amount of custom code I need, and also make it easier to illustrate recovery steps for auditors.

Price is another reason I use GCP for DR. Sustained-Use Discounts are automatically applied to instances with long run times, which works well for warm standby configurations. Committed-Use Discounts lower ongoing costs for pilot-light databases and small application tiers. Spot VMs give big discounts for non-critical batch rebuilds and for DR testing. Cloud Storage lifecycle policies can transfer older snapshots to colder classes without manual intervention. Per-second billing and simple, global load balancing pricing simplify the math. These dials enable me to spin up a low-cost standby that I can still test each month without incurring a fortune.

Operations are simpler on GCP because the pieces are consistent and globally scoped. One global load balancer in front of multiple regions gives a consistent external footprint. Cloud Logging and Cloud Monitoring give me one place to view latency, error rates, and failover health checks. Identity and Access Management is the same for all services, and Secret Manager gives me centralized secrets with audit logging. If I need policy guardrails, Organization Policies and VPC-SC (service perimeter) enable me to segregate backup data and reduce blast radius. None of these are flashy capabilities, but they all reduce day-two work and allow the team to move faster in a real incident.

My recommendation is a GCP-first DR plan. Backup workloads with Cloud Backup and DR, deploy to two or more regions, and put traffic behind the global load balancer so failover is immediate from the user's point of view. Use Cloud DNS health checks if you require DNS-level control. Reduce expenses through Sustained-Use and Committed-Use Discounts, and use Spot VMs for rebuilds and exercises. Incorporate Azure only where a workload is tightly coupled with Microsoft technologies that truly need Azure-native services. Incorporate AWS only where an exceedingly rapid, prescriptive lift-and-shift is the quickest stopgap. For the majority of workloads, GCP alone is the best combination of resilience, speed, and cost control.

In hindsight on the Q&A, I should have framed every question starting with GCP. In the Windows/AD question, I should have started by mentioning that GCP offers Managed Microsoft AD and Windows Authentication support for SQL Server and that these features cover basic enterprise needs. In the region's question, I should have led with GCP's single anycast IP and automatic cross-region failover because that is what actually minimizes RTO. For compliance, I should have stated plainly that the big frameworks most programs need are in GCP's catalog. For "proof" of simpler setup and faster recovery, I should have referred to Cloud Backup and DR's agentless deployment and application-aware restores, then explained how the global load balancer reduces manual steps for failover. In this position, the audience knows why GCP is my default go-to for DR, with the other clouds saved for when a workload truly needs them.