

Your organization has implemented Google Workspace (Formerly G Suite) for email, enterprise collaboration, and cloud communications. The next step to take full advantage of your Google Workspace engagement is to move your applications to the Google Cloud Platform. The Google Cloud Platform provides better managed services for your applications and reduces the underlying infrastructure. GCP is competitively priced and especially for smaller component deployments like yours, GCP's user friendly interface is more easily managed by less tech-savvy system developers.

The Objective

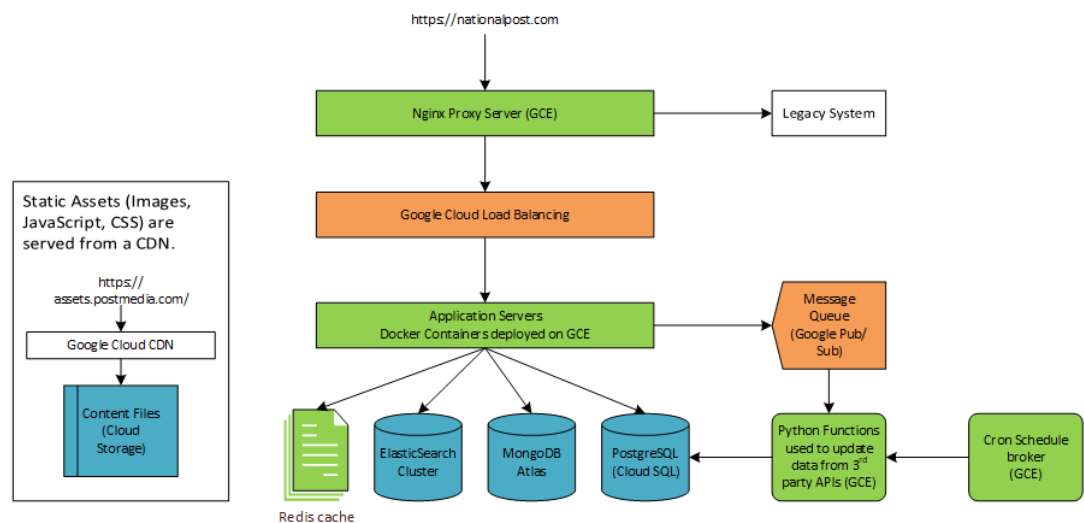
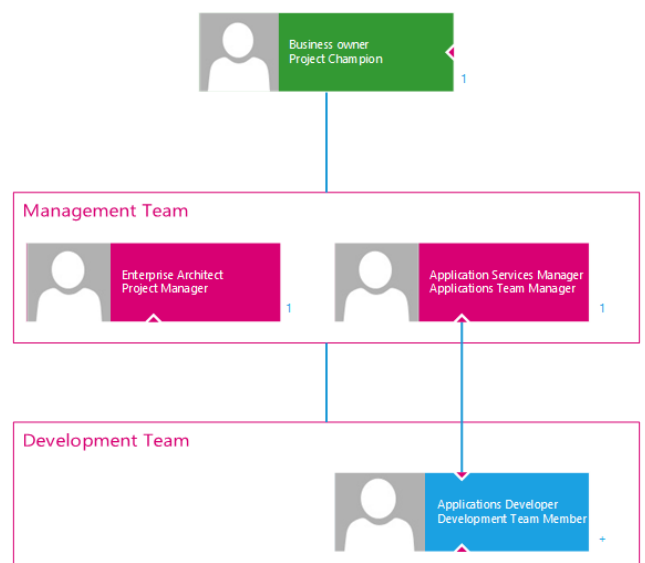


Figure 2. The resulting architecture will run all applications on Google Cloud Platform and maintain or improve the performance established in *baselines*.

Migrating services from AWS to GCP will take 3 months. Upfront costs related to activities performed on the VMs for transferring data between the platforms and running tests amounts to about \$4,692.97.

The Project Team



Responsibilities

The project sponsor acts as the project champion within the leadership team. They define strategic objectives and the vision for the project. They define the governance framework and best practices within the business. They secure approval for the project, project funding, and project resources. They are responsible for securing the go ahead at each phase of the project. They review milestone reports and ultimately the acceptance of the project deliverables.

Members of the project management team work together to ensure the success of the project. Their tasks involve resource planning, milestone planning, and other activities for managing project resources and project activities.

The architect plans and completes all aspects of the migration. Their core responsibilities include defining necessary refactoring required to make the migration successful, designing strategies for data migration, defining cloud-solution requirements, and determining migration priorities and production switchover mechanisms. They understand the business case and the vision for the project. They define the required resources time and financial commitment. They lead the project team by assigning tasks to the capable resources through the phases of the project. The architect is responsible for managing scope creep and controlling cost using cost controlling strategies. The architect is responsible for the quality delivered and should use the tools at their disposal to evaluate and optimize the solution. They review progress, report milestones achieved and adjust timelines.

The functional manager manages and owns the resources from their business unit that are gathered for the project. They share the responsibility of directing the work of their resources. They manage resource availability. Their goal is to avoid burnout, overworking, conflicts with the business, and issues that arise from employee turnover.

The development team is responsible for carrying out technical activities related to the project. Such activities include installing and configuring databases, systems, and applications.

The development team member has the required skills to fulfill technical tasks related to the project. Such skills include MongoDB, PostgreSQL, Python, Nginx server, GCP implementation, amongst others.

Authority relationships

The project sponsor is the highest-level person involved in the project within the organization. They are responsible for all the business' concerns relating to the project.

The architect reports directly to the project sponsor on important matters relating to the project. The architect communicates statuses to the business owner and coordinates with business teams around conflicts due to business commitments.

The architect and the functional manager make up the management team for managing project resources and scheduling.

The functional manager is the other member of the management team responsible for managing resources and scheduling for the project.

The development team members report to the management team regarding scheduling and business conflicts. The development team members report to the architect for project tasks and activities.

Stakeholders

1. Users spread across multiple availability zones and regions.
2. The business impacted by the reduced cost and increased performance.
3. Applications teams perform fewer tasks managing the infrastructure and take full advantage of modern tools for monitoring performance.

Project management methodology

The project will be funded by the organization's senior leadership. The project will be managed by the architect. Internal resources from development teams will be assigned to the project to provide technical expertise in fulfilling technical tasks. Execution phases of the project (development, staging, and production) will follow an agile methodology with the project development team working with the architect to complete action items. The architect will maintain periodic meetings with the business to review progress and gain acceptance for deliverables.

Work Breakdown

- **Scope**
 - Conduct needs analysis
 - Identify stakeholders, needs, influence on the project
 - Determine project scope
 - Develop a proposal
 - Define preliminary resources
 - Secure project sponsorship
 - Secure core resources
- **Concept**
 - Identify security and compliance requirements
 - Create governance workbook
 - Draft preliminary architecture specifications
 - Define Migration Approach
 - Perform cost benefit analysis
 - Recommend an option and obtain approvals
 - Scope all deliverables, objectives, quality, cost
 - Establish organization structure
 - Define progress reporting - status reports, burndown, interdependency, schedule/plan
 - Develop a project charter
- **Planning**
 - Project Plan
 - Delivery and mitigation plan
 - Obtain approvals to proceed (concept, timeline, budget)
 - Secure required resources
- **Hold Kick-Off**
- **Set up recurring business meetings**
- **Acquire access to current applications, data stores, and services**
- **Design**
 - Set up a Google Cloud account, organization, and infrastructure project

- Create cloud source repositories
- Review preliminary architecture specifications
- Establish cloud KPIs
- Establish pre-migration performance baseline in Blue Medora dashboard
- Obtain approval to proceed
- Development
 - Review functional specifications
 - Assign development staff
 - Provision Dev environment
 - Clone and prepare EC2 instances
 - Configure network, access, and space requirements
 - Copy over application servers and workers
 - Set up Redis
 - Configure internal http load balancing from proxy server to application servers
 - Configure pub/sub queue
 - Develop system performance monitoring, issue logging and reporting
 - Unit Testing
 - Integration Testing
 - Obtain approvals to proceed
- Staging
 - Provision staging environment
 - import dev resources to staging environment
 - Copy over MongoDB instance
 - Configure the Postgres cloud SQL database
 - Reindex Elasticsearch cluster
 - Transfer the S3 bucket
 - Integration Testing
 - End-to-end testing
 - Pilot
 - Obtain approvals to proceed
- Production
 - Provision production environment
 - Run test suites in production for sanity check
 - Documentation
 - Deployment
- Post Implementation Review
- Handover to business and maintenance team
- Project end

As seen in our work breakdown structure, the architect will perform a review of the concept as well as every stage of the project with the project team to incorporate other viewpoints for better quality. The three stages of the deployment: development for testing application instances in the new environment, staging for running a pilot, and production for changeover from the old platform are designed to reduce the cost of unforeseen issues, as well as reduce running both environments in parallel.

Scope control

The scope of this project has been defined as migrating the architecture and components from AWS to GCP. Any changes to the applications must be related to meeting the performance standards set in *baselines* and enabling the applications and services to run on the new platform. There is no effort to improve the applications or the architecture solution.

Cost

The project team members are already a part of the organization and there will be no need for additional resources. The additional costs incurred by the business are related to running the development environment, and running the staging environment for the pilot, and running the production environment in parallel before the changeover date. Upfront costs related to activities performed on the VMs for transferring data between the platforms and running tests amounts to about \$4,692.97.

Cost Benefit Analysis

Pricing model	AWS	GCP
Cloud CDN	\$30,510	\$24,548.20
File Storage	\$1,627.60	\$451.00
Cloud DNS	\$412.00	\$404.80
Proxy	\$45.34	\$100.76
Load balancer	\$426.03	\$427.85
Redis Cache	\$45.34	\$51.84
Application Servers	\$45.34	\$51.84
Elasticsearch Cluster	\$346.10	\$580.73
Mongo DB	\$346.10	\$580.73
PostgreSQL	\$9,200.44	\$9,699.60
Message Queue	\$912	\$2,047.60
3 rd party API workers	\$45.34	\$51.84
Cron schedule broker	\$45.34	\$51.84
Total estimated monthly cost	\$44,006.97	\$39,048.63

Cost control

To control project costs, the cloud services must be turned off while there are no related project activities being performed.

Moving at least one of these services to Google whilst the others remain in AWS would cause latency as requests would need to traverse the VPN or public internet, therefore we want to minimise the amount of time we are running with the application split across the two providers.

Our database is one of the biggest costs we have in our infrastructure, so we want to make sure that we are not running large replicas of this database side by side for too long.

Risk management

Deploying the application servers first would be cheap, simplify the deployment and would be easily reproducible. It would also allow us to verify the deployment and switchover when the time came, giving us a viable rollback mechanism. We have avoided downtime by running the application on both platforms during the changeover, and with replicated databases to avoid data loss.

Likelihood	Severity			
		Low	Medium	High
	High			
	Medium	Data loss when electing new master database		
	Low			Unidentified errors causing application crash

We are setting up replication from the master in AWS and then electing the master in GCP after all the data is transferred. When the election is taking place, there can be data lost during that less than one second window. To mitigate the risk of data loss when electing new master database, we will perform the changeover during the lowest activity window so that there will be minimal to no data loss.

We are using the Linux package provided by Migrate for google cloud to reconfigure our machines to work in the google cloud environment. There are also other reconfigurations that will be done to connect the new services. When changes are made to applications, there are often unforeseen issues that arise. To mitigate unidentified errors causing application crash, we will use extensive unit, integration, and end-to-end testing of all applications, integrations, and scenarios. We will also run a pilot of the working environment, as well as run using DNS weighted routing to run the two environments in parallel for a one-week review period.

Quality control

AWS CloudWatch is used to collect metrics for monitoring the performance of our services in AWS. The following KPIs will set a baseline for the performance of our applications:

Category	KPIs
User experience	Page load time
	Lag
	Response time
Application performance	Error rates
	Throughput
	Availability
	Apdex (Application Performance Index)
Infrastructure	CPU usage %
	Disk performance
	Memory usage
	Network throughput

The current performance of our applications in AWS will determine if the future performance in GCP is acceptable. We will collect performance metrics for an hour each during peak, normal, and quiet periods. This will also inform our selection of a changeover window. For GCP Blue Medora has dashboards with the same metrics.

Status reports

Bi-weekly status reports detail: the changes to the plan and reasons for those changes, the percentage completion of activities within the current phase, the status of project resources, and other key updates.

Milestone reports inform acceptance and go ahead for the next phase of the project.

Documentation

A production runbook details the architecture's components and their relationships, the activities that must be performed while maintaining the production environment, the process for making changes to the production environment, some issues that were encountered and issues that may be expected by the maintenance team.

Assumptions

Our first key assumption is that the organisation already has the skills needed for performing the migration within the IT teams. We assume that there is no question about running a hybrid cloud architecture where we leave components that are cheaper to run in AWS on that platform. We assume that there is no need to refactor the architecture to take advantage of serverless computing or managed services in GCP for the scope of this project. The following assumption were made to compare3 costs:

Cloud CDN: 50TB/month in Asia/Pacific, Europe, North America, Oceania, and China/Japan.

Cloud Storage: 50GB standard storage, 10million Class A operations, 1000 million Class B operations monthly

Cloud DNS: 24 managed zones, 1billion queries

EC2 instances: 1 per application (Nginx, app servers, MongoDB, Elasticsearch cluster, cron-based workers, cron schedule broker). Regular, General purpose, standard 4 CPUs, 16GB RAM, us-east-1, running 24 hrs/day 7days/wk.

Load-balancer: us-east-1, autoscaling, 50TB/month

PostgreSQL: 3 instances, 4 cores, 16GB RAM, 8TB SSD, 1TB backup, 24 hrs/day 7days/wk.

Message Queue: 50TB/month

Schedule and milestones

Activity	Date of completion
Sponsorship and resources approved	Fri May 07
Concept approved	Mon May 17
Planning complete	Mon May 24
Kick-Off	Mon May 31
Design approved	Wed Jun 02
Development complete	Fri Jun 18
Go decision from pilot	Thu Jul 08
Deployed to production	Wed Jul 21
Post implementation review	Fri Jul 23
Kickoff maintenance teamwork	Tue Jul 27
Project end	Tue Jul 27

Acceptance

The project sponsor must review and approve the project scope and resource requirements.

The project sponsor must accept the proposed solution and implementation approach.

The project sponsor must accept the defined project plan.

The project sponsor must accept the designed architecture as it will exist and run within the google cloud platform.

The project sponsor must review and accept the results of the pilot.

Final acceptance criteria: the final solution must meet the set baselines collected for the current solution in AWS. The final solution must not require manual intervention from the maintenance team to run in normal circumstances.

After all these activities, when the final deliverables have been accepted as they meet the final acceptance criteria and the final solution has been handed over to the maintenance team, the end of the project has been reached.