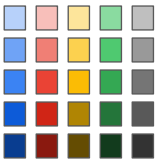
Google Cloud Platform

Managed Services

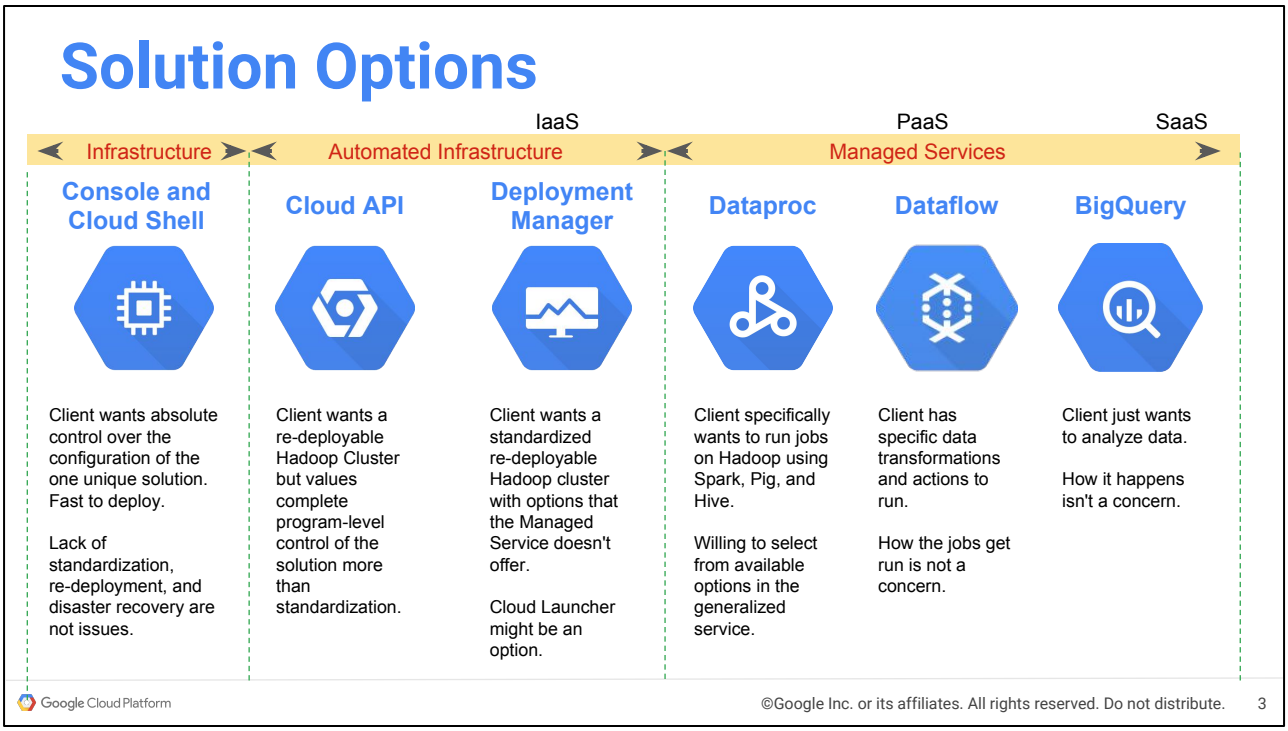
v 1.0

Managed Services



Agenda

1. Managed Services
2. Dataproc
3. Dataflow
4. BigQuery
5. Other Services
6. Review



Notice the progression from left to right also happens to be the order of subject as they were covered in this class.

The point of this slide is that there is a continuum of options available in GCP. And that you should consider all options in a particular context. The solutions to the right offer greater outsourcing of overhead, but with a loss of control. And the solutions to the left offer greater detail of control with increased responsibility and overhead.

The text below the icons describes a particular use case -- a Big Data application -- and what items of importance to the client might drive the solution choices more towards the left.

From right to left.
(1) A client wants to analyze data and doesn't care how the underlying systems work -- just results. BigQuery is a good option. This is SaaS or PaaS solution. The client doesn't have visibility to the jobs, just the results. But what if the client already knows HOW the analysis is to be performed?
(2) The client has specific data transformations to perform -- cares about how the work gets done -- but doesn't care which platform or how those jobs get run so long as it performs the work as described in a particular flow. Dataflow is a good option. This is a PaaS solution - the program and data is loaded onto the platform and it runs. The client doesn't have visibility into the resources, just the jobs. But what if the client want's to have visibility into the cluster and HOW the jobs are running?
(3) In this example the client has specific Spark or Pig or Hadoop MapReduce jobs to

run. Perhaps an investment was already made in developing this code. The jobs need to run on a plain Hadoop environment. And the client is willing to surrender customization for the benefit of lowered costs and outsourcing some of the overhead tasks (maintenance, software update) to Google. In this case the client gets a cluster, and can specify options such as number and size of disks, but not other options. Dataproc is a managed service that provides a cluster with limited customization options. But what if those limits are an issue for the client?
For example, what if the default blocking factor in HDFS of 128MB results in super-long job runs for the application, and a 512MB blocking factor would radically reduce run time?
(4) At this point the client crosses the threshold between PaaS and IaaS and enters the Infrastructure services you have been studying so far in this class. The client decides to take responsibility for and absorb the overhead for building, running, and maintaining a Hadoop cluster. As you have learned, there are three options: Deployment Manager, Cloud API automation, and directly building the solution using Console and Cloud Shell -- the "manual approach".
Based on your experience with Infrastructure and Automation, what reasons might you offer to the client to develop the solution in one of the three alternatives {Console and Cloud Shell | Cloud API | Deployment Manager}. Note that there are 3rd party options and tools you could use as well. These are just the native GCP options.

Data processing managed services

Dataproc

Spark, Hadoop, Hive, and Pig

- Fast
- Easy to use
- Low cost
- Rapid scaling of cluster
- Fully-managed service

Dataflow

Data processing service

- Fully-managed service
- Processing patterns include:
  - Batch processing
  - Stream processing
  - ETL

BigQuery

Data analytics service

- Interactive analysis
- Multi-petabyte datasets
- Unlimited storage
- Fully-managed service

Agenda

1. Managed Services
2. Dataproc
3. Dataflow
4. BigQuery
5. Other Services
6. Review

Dataproc

Low cost
- Fixed price
- Per-minute billing
- Custom VMs
- Pre-emptible VMs

Fast
- Custer provisioning
- Execution
- Server removal

Maintainable
- Image versioning management for Apache tools

Scale
- Manual scale cluster as needed

Support
- Spark
- PySpark
- Spark SQL
- MapReduce
- Hive
- Pig

Service integrations
- Cloud storage
- Big Query
- Big Table
- Stackdriver

Dataproc

Management options
- WebUI
- SSH
- REST API
- Cloud SDK

Features

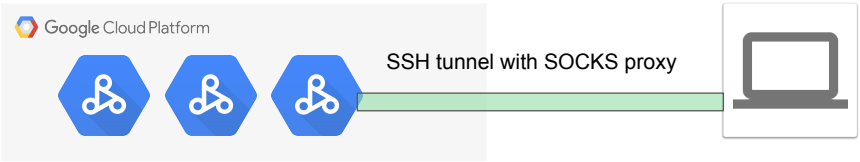
- Cluster start time
  - Elapsed time from cluster creation until it is ready < 90 secs
- Billing is per-minute used
- Preemptible VMs
  - Clusters can utilize preemptible VMs for cost reduction
- Job output is easy to locate without log files review
- Job cancellation does not require SSH to the cluster

For comparison: Typical cluster start time for non-GCP 180-360 seconds, hourly billing, non-preemptible VMs.

Faster data processing workflows because less time is spent waiting for clusters to provision and start executing applications.
Reduced costs for running Spark and Hadoop because you pay for what you actually use, not a cost which has been rounded up.
Lower total operating costs for Spark and Hadoop processing by leveraging the cost benefits of pre-emptibles.
Higher productivity because job output does not necessitate reviewing log files and canceling jobs does not require SSH.

Connecting to the cluster

- YARN (resource manager) web UI
  - `http://[master-host-name]:8088`
- HDFS (distributed file system) web UI
  - `http://[master-host-name]:50070`
- SSH
  - `gcloud compute ssh [master-host-name]`
- SOCKS



SOCKS passes browser data over SSH so you don't have to open additional firewall access

Note: You must use the `gcloud compute ssh` command to connect to the Master or to set up an SSH tunnel for use with SOCKS.
The Cloudshell SSH is incompatible and will not connect to the Master.

The SSH tunnel supports traffic proxying using the SOCKS protocol. This means that you can send network requests through your SSH tunnel in any browser that supports the SOCKS protocol. This method allows you to transfer all of your browser data over SSH, eliminating the need to open firewall ports to access the web interfaces.

Cluster actions

Startup

- Initialization actions in Github or Cloud Storage
- Customize Service Account for increased IAM control

Submit, monitor, and control jobs:

- Cloud console
- `gcloud tool`
- Cloud Dataproc API

Stackdriver monitoring

- Stackdriver agents enabled by default, can be disabled

Initialization actions are used with all new VMs in the cluster, gives you the opportunity to customize the cluster.
Instead of the default service account, you can use a custom service account which gives you the ability to assign roles and limit access to increase security.

Agenda

1. Managed Services
2. Dataproc
3. Dataflow
4. BigQuery
5. Other Services
6. Review

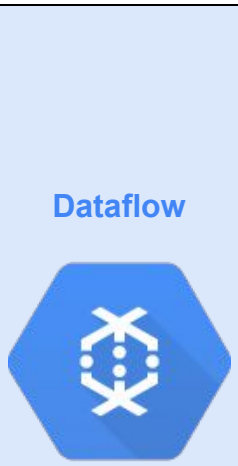
Dataflow

No-ops
- Fully-managed
- Automatic on-demand resource allocation
- Auto-sizing of workers (horizontal autoscaling)
- Freedom from operational tasks

Unified programming model
- Apache Beam SDK
- Separates data processing requirements from data source
- Apply solutions to both batch and stream data

Features
- Liquid sharing
- Autoscaling mid-job

Service integrations
- Cloud storage
- Cloud Pub/Sub
- Big Query
- Big Table
- Cloud Datastore



Can be extended with sinks to Kafka and HDFS.
Spark or Flink compatibility is possible.

Liquid sharing is dynamic work rebalancing.

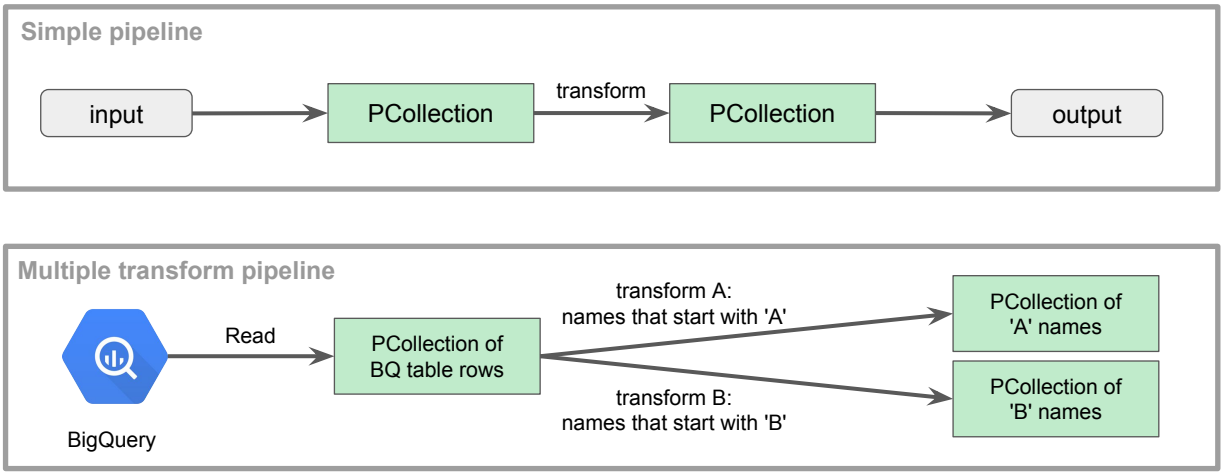
https://cloud.google.com/blog/big-data/2016/05/no-shard-left-behind-dynamic-work-rebalancing-in-google-cloud-dataflow

Dataflow data programming models

- Pipelines
  - A series of computations that accepts data and transforms it
  - Output can be to output sink or internal sink
  - Output and input can be the same for data format conversion
- PCollections
  - A specialized container of nearly unlimited size that represents a set of data in the pipeline
- Transforms
  - A data processing operation
- I/O Sources and Sinks

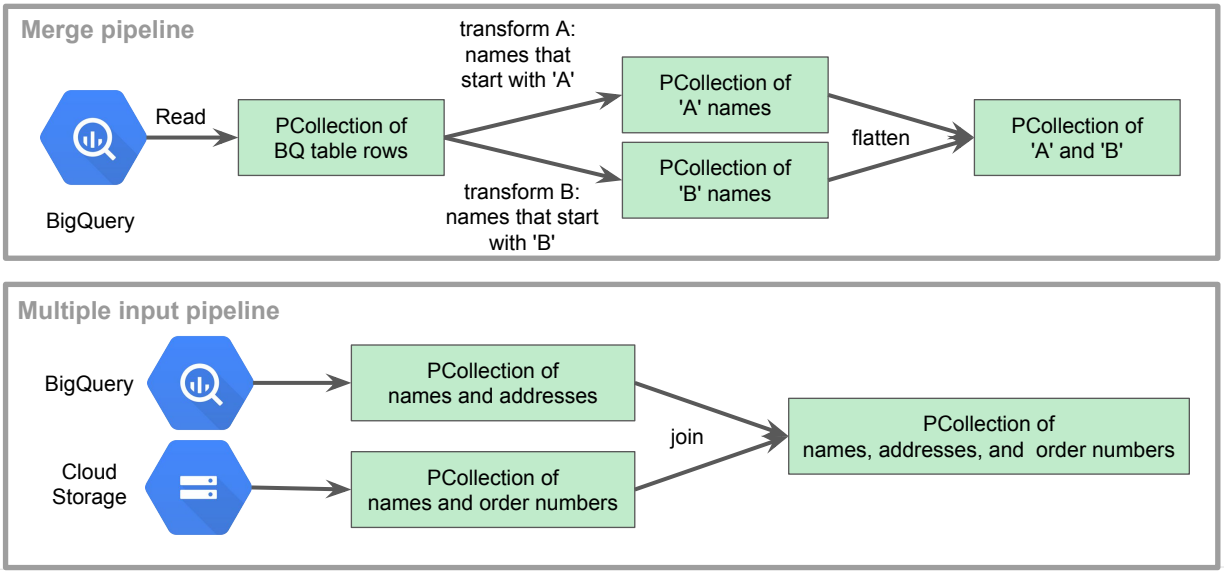
https://cloud.google.com/dataflow/model/programming-model

Pipeline examples (1)



These examples give you a sense of the processing capabilities of Dataflow. In the simple model pipeline, data is input from source into a PCollection, transformed, and output. The pipeline is a Directed Acyclic Graph (DAG). In the multiple transform pipeline, data read from BigQuery is filtered into two collections based on the initial character of the name.

Pipeline examples (2)



Data can be merged together, in the Merge example, using "flatten".
And data can also be input from multiple sources.

Agenda

1. Managed Services
2. Dataproc
3. Dataflow
4. BigQuery
5. Other Services
6. Review

Big Query

Analytics Data Warehouse
- Fully-managed
- Petabyte scale
- SQL interface
- Very fast

Example: Complex query over 100B Rows

- **Execution time: 21s**
  - 4 TB of data read
  - 100 Billion regular expressions run
  - 278 GB shuffled
  - 3600 HDD (@ 100MB/s)
- **Serial Execution time:**
  - 11.6 Hours to read 4 TB from disk (@ 100 MBps)
  - 27 hours to run 100 Billion regexps (@ 1 µsec each)
  - 37 minutes to shuffle 278 GB across the network (@ 1Gbps)
  - 8800 CPUs (@ 1 µsec each

Big Query

Management options
- WebUI
- SSH
- REST API
- Cloud SDK

Can be extended with sinks to Kafka and HDFS.
Spark or Flink compatibility is possible.

Liquid sharing is dynamic work rebalancing.

https://cloud.google.com/blog/big-data/2016/05/no-shard-left-behind-dynamic-work-rebalancing-in-google-cloud-dataflow

Query example

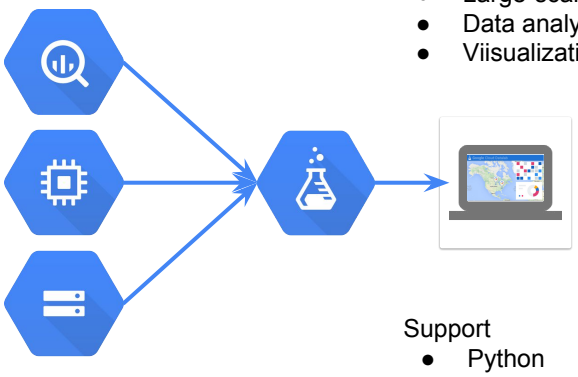
```
SELECT language, SUM(views) as views
FROM (
  SELECT title, language, MAX(views) as views
  FROM [bigquery-samples:wikipedia_benchmark.Wiki100B]
  WHERE REGEXP_MATCH(title, "G.*o.*")
  GROUP EACH BY title, language
)
GROUP EACH BY language
ORDER BY views desc
```

Agenda

1. Managed Services
2. Dataproc
3. Dataflow
4. BigQuery
5. Other Services
6. Review

Cloud Datalab

Interactive tool for data exploration
- Large-scale data
- Data analysis
- Viisualization



Support
- Python
- SQL
- JavaScript

Accepts input from
- BigQuery
- Compute Engine
- Cloud Storage

Cloud Datalab

Jupyter Notebook combines:
- Code
- Documentation
- Results
- Visualizations

Datalab is a data visualization service native to GCP.

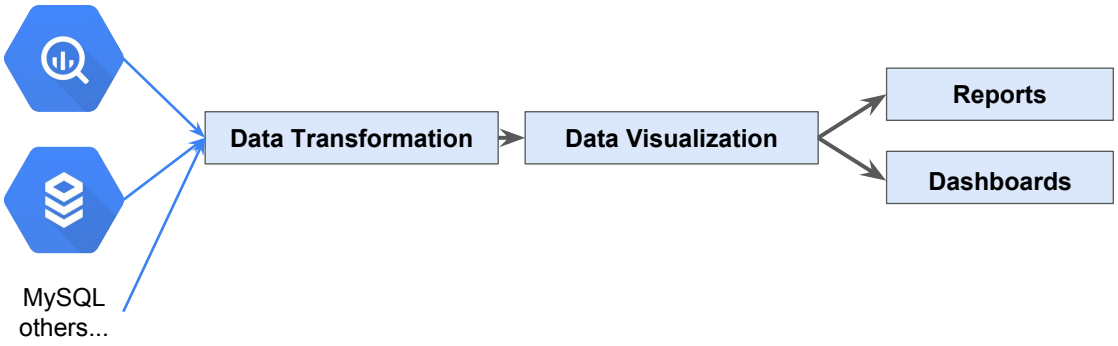
It is mentioned here so that you can consider it as part of a solution design if data visualization is needed.
There are many other 3rd party options available, such as Tableau.

JavaScript is support to enable BigQuery User Defined Functions.

Google Data Studio

Decision support report generator.
Useful for making data-backed architectural recommendations.

Data Connections



https://cloud.google.com/data-studio/

Agenda

1. Managed Services
2. Dataproc
3. Dataflow
4. BigQuery
5. Other Services
6. Review

More...

- Dataproc
  - https://cloud.google.com/dataproc/docs/
- Dataflow
  - https://cloud.google.com/dataflow/docs/
- BigQuery
  - https://cloud.google.com/bigquery/docs/
- Other Services
  - Datalab: https://cloud.google.com/datalab/
  - Datastudio: https://cloud.google.com/data-studio/

More to learn on these subjects. Here are some suggestions and links.


cloud.google.com