



Introduction to Cloud Dataproc

Data Engineering on Google Cloud Platform

Agenda

Why unstructured data?

Why Cloud Dataproc?

Creating a Dataproc cluster + Lab

Custom machine types

Sources of data

Data you analyze today

Data you collect but don't analyze

Data you could collect but don't

Data from partners and 3rd parties

What are some reasons that you have data that you don't analyze?

Data you analyze today

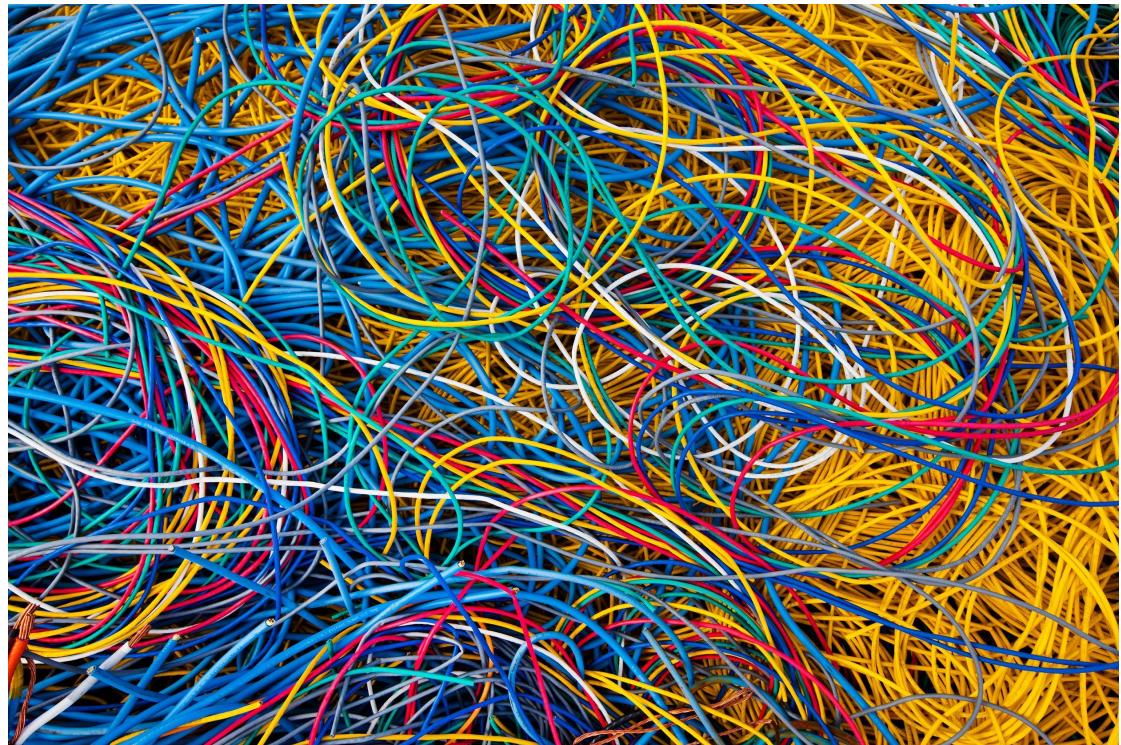
WHY?

Data you collect but don't analyze

Data you could collect but don't

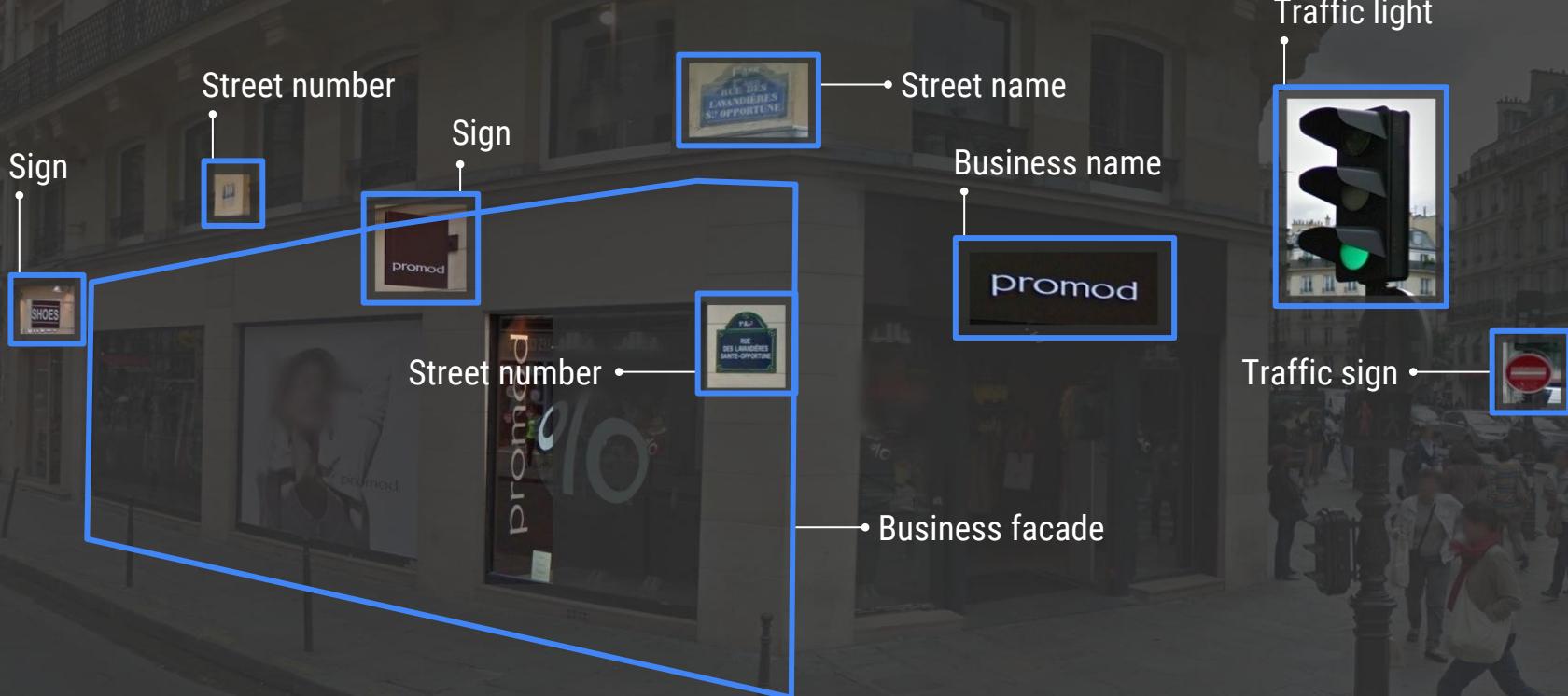
Data from partners and 3rd parties

Unstructured
data accounts
for 90% of
enterprise data*





Finding new value in data



Some Big Data applications involve a human

Human

Real-time insight into
supply chain operations.
Which partner is causing
issues?

Drive product decisions.
How do people really use
feature X?

Counting problems are perfect for big data analytical tools

Human

Real-time insight into supply chain operations.
Which partner is causing issues?

Drive product decisions.
How do people really use feature X?

Easy counting problems

Did error rates decrease after the bug fix was applied?

Which stores are experiencing long delays in payment processing?

These are also counting problems, but they are not as easy...

Human

Real-time insight into supply chain operations.
Which partner is causing issues?

Drive product decisions.
How do people really use feature X?

Easy counting problems

Did error rates decrease after the bug fix was applied?

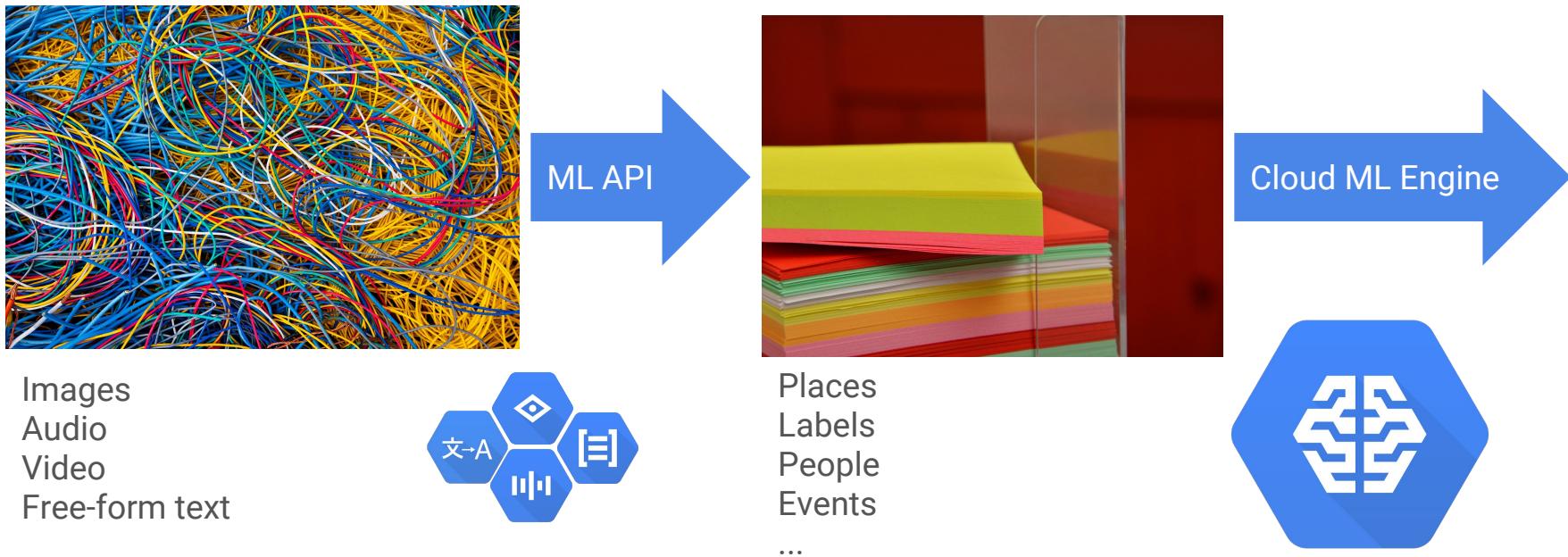
Which stores are experiencing long delays in payment processing?

Harder counting problems

Are programmers checking in low-quality code?

Which stores are experiencing lacking of parking space?

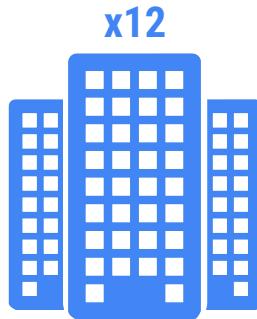
Build on top of Google



Agenda

Why Cloud Dataproc?

Will you ever have a PetaByte of data?



A stack of floppy disks
higher than twelve
empire state buildings



27 years to
download over 4G

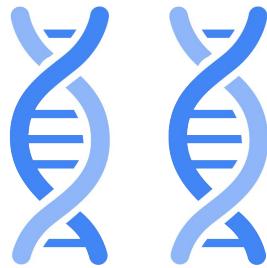


100 Libraries
of Congress



Every tweet ever
twittered...50 times

How *small* is a PetaByte?



2 micrograms of DNA

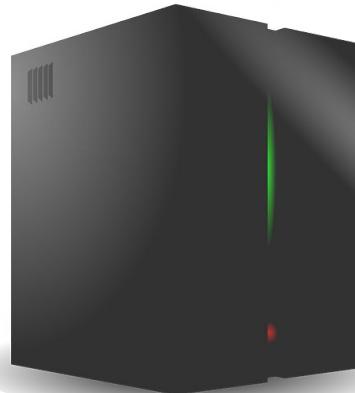


1 day's worth of video uploaded to
YouTube



200 servers logging at 50 entries per
second for 3 years

How do you process large amounts of data?

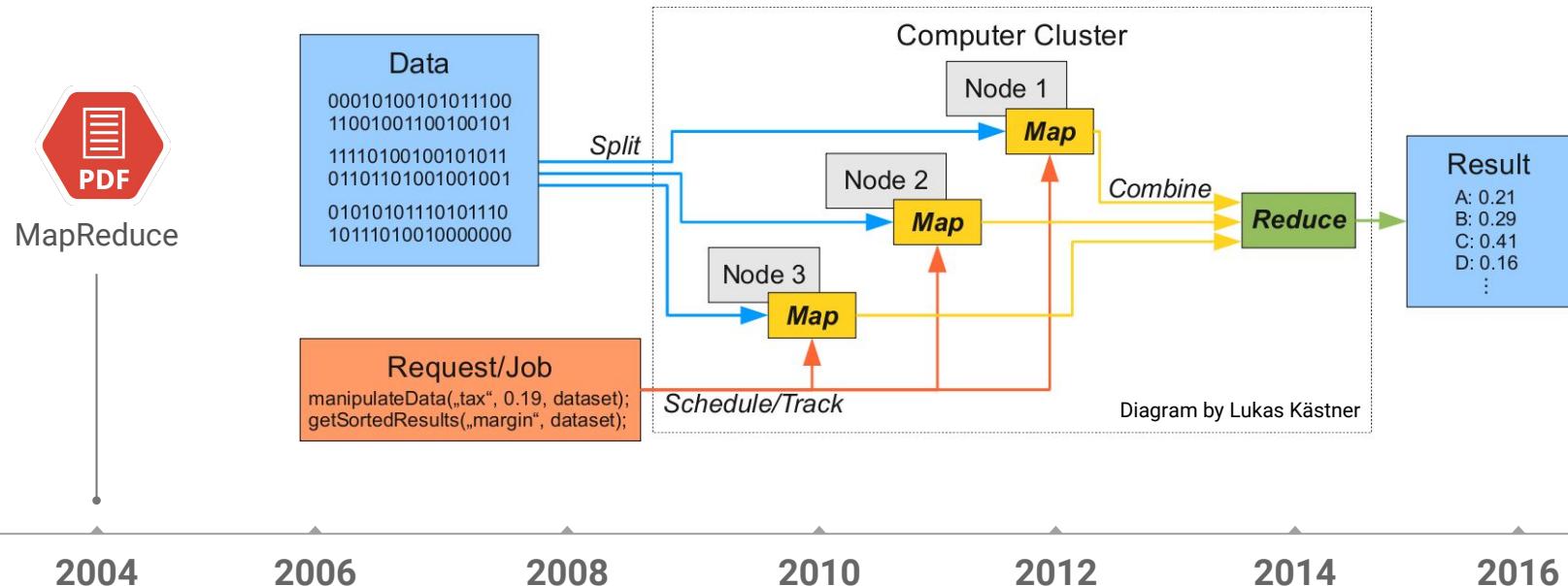


SCALING UP

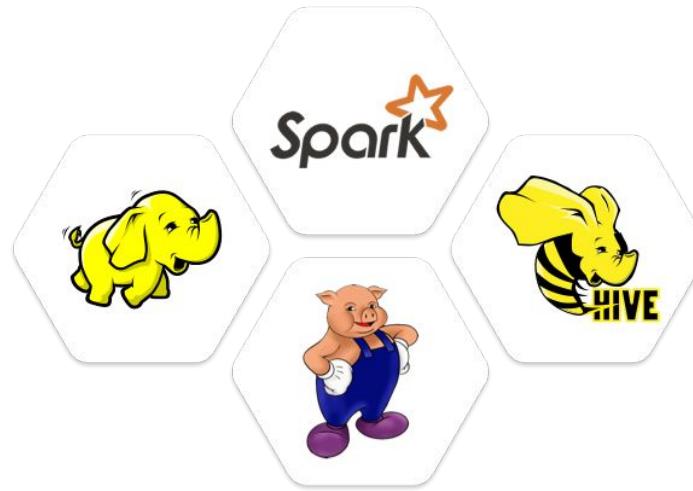


SCALING OUT

MapReduce approach splits Big Data so that each compute node processes data local to it



Many on-premise applications for Big Data are built using the open-source stack



Apache Spark is a popular, flexible, powerful way to process large datasets

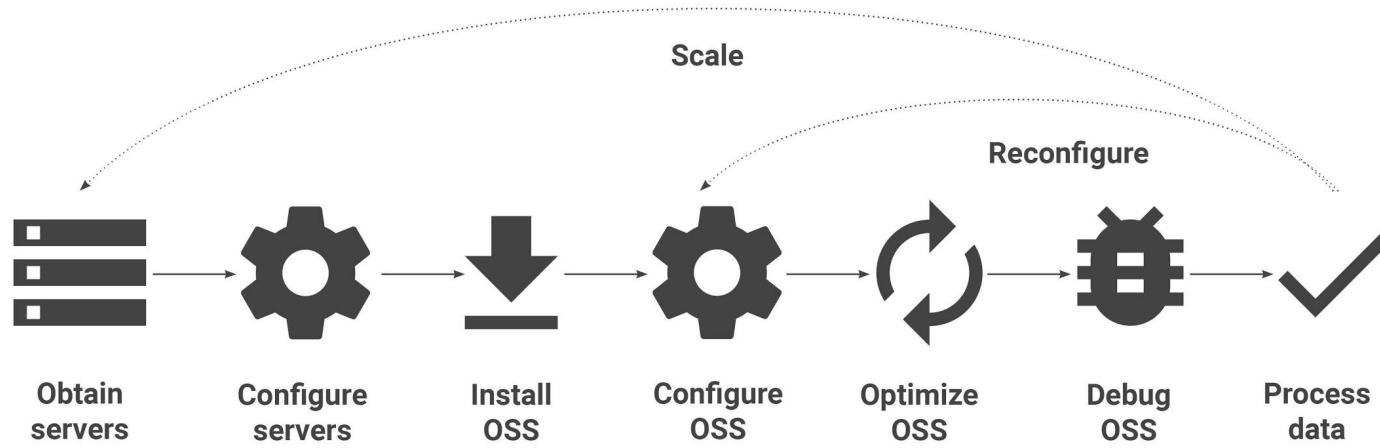


etc.

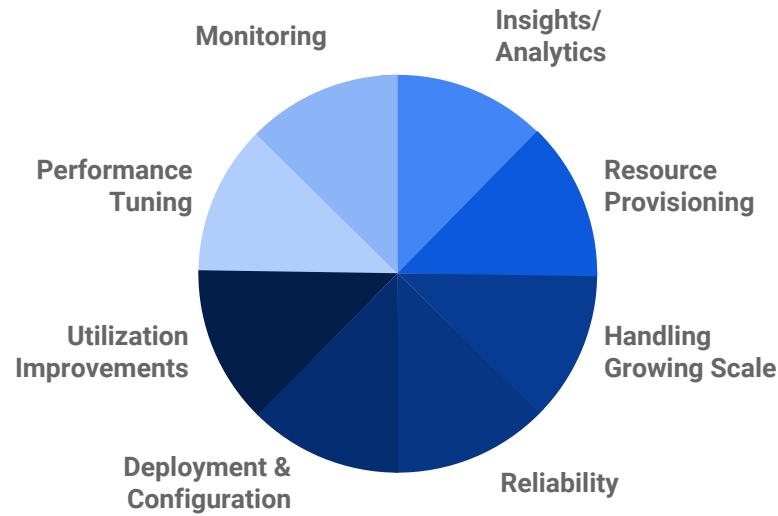


spark.apache.org

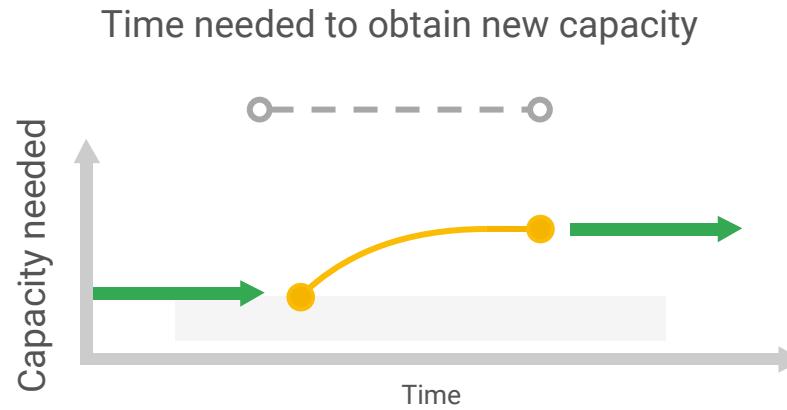
Typical Spark and Hadoop deployments involve...



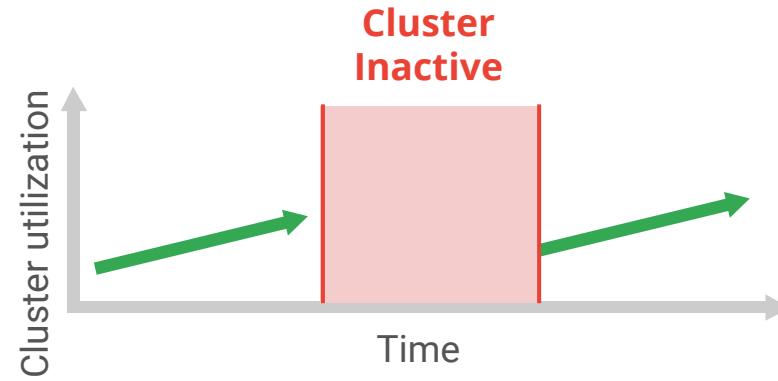
Lots of time is spent on administration and operational issues



Scaling can take hours, days, or weeks

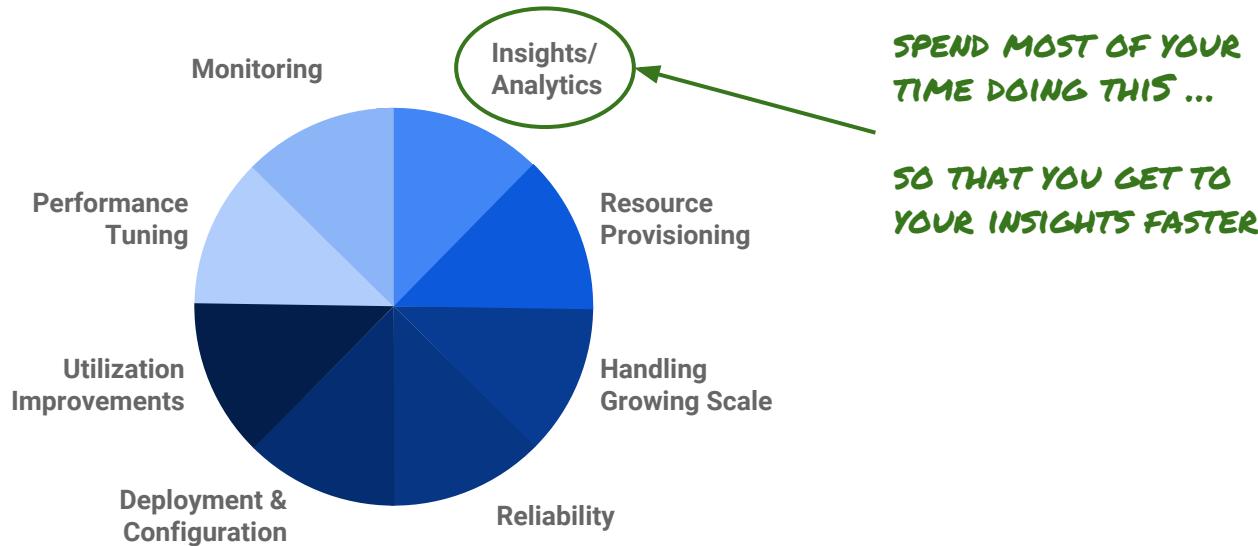


You have to babysit utilization



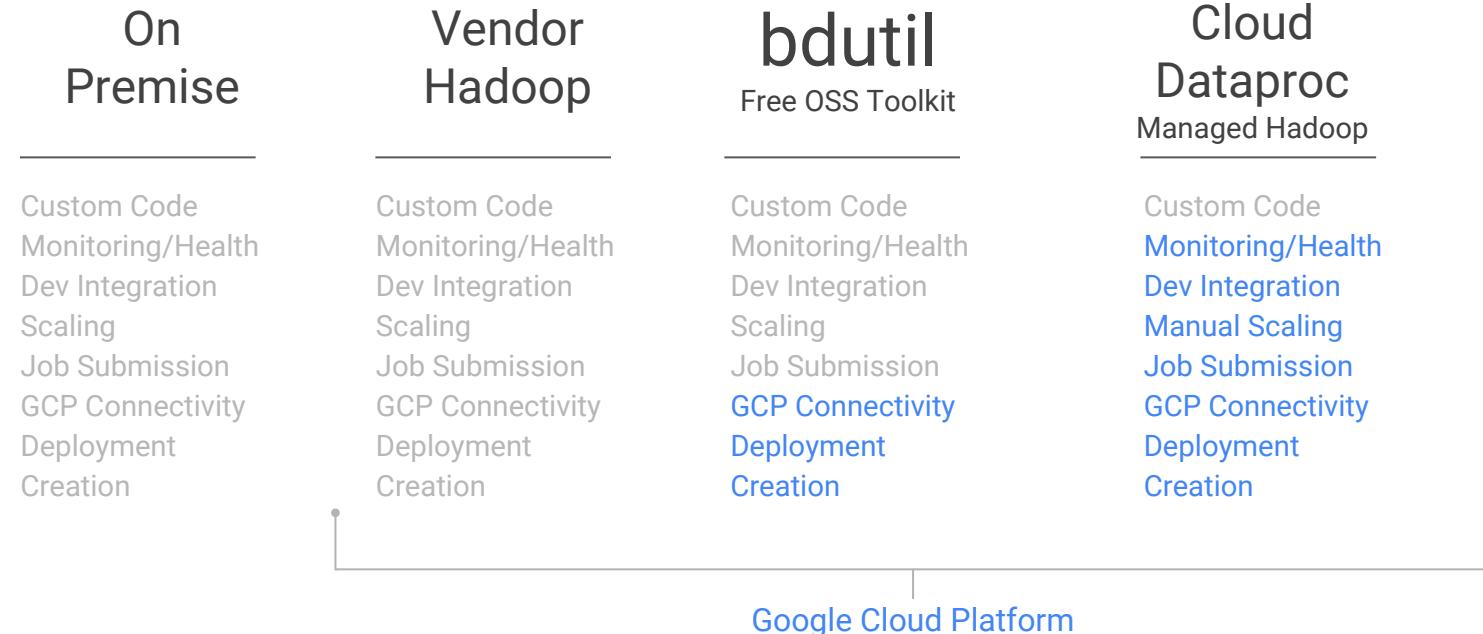
ENSURING CLUSTER IS ALWAYS
BEING UTILIZED IS HARD

But you want to focus on insights and analytics

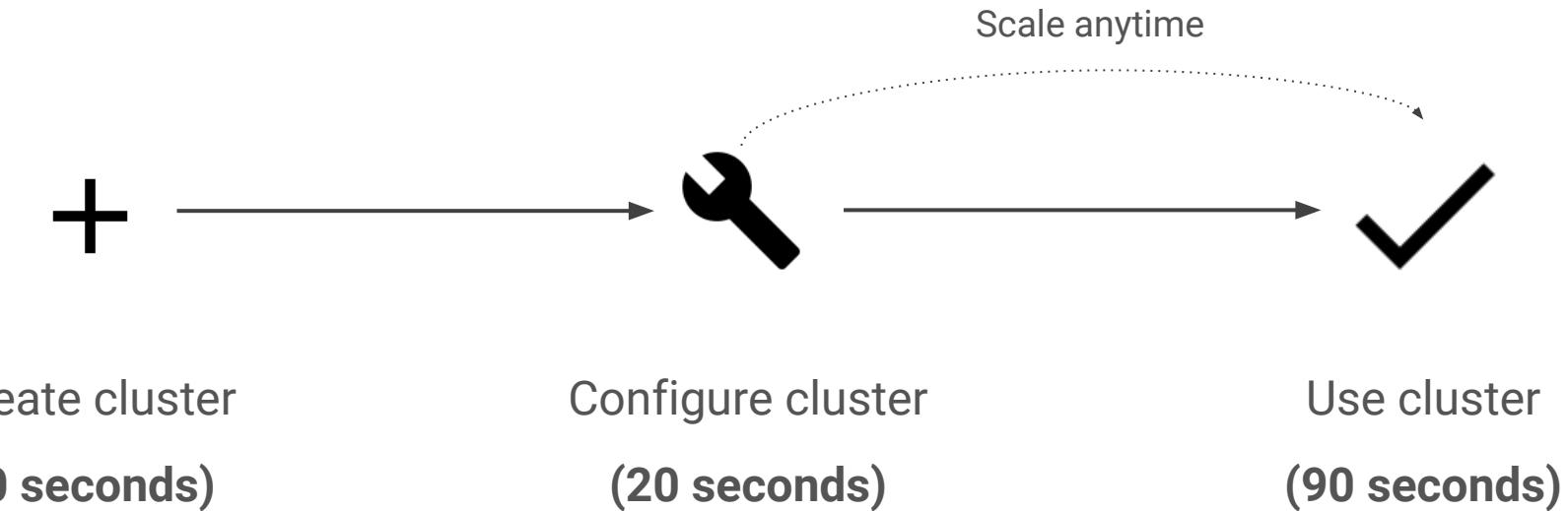


Dataproc eases Hadoop management

- █ Google managed
- Customer managed



Typical Dataproc deployments involve...



Cloud Dataproc provides compelling reasons to run open-source tools on GCP

- Stateless clusters in <90 seconds
- Supports Hadoop, Spark, Pig, Hive, etc.
- High-level APIs for job submission
- Connectors to Bigtable, BigQuery, Cloud Storage



Agenda

Creating a Dataproc cluster + Lab

Create a cluster from the web console

The image shows the Google Cloud Dataproc web interface. On the left, there's a navigation bar with a 'Dataproc' icon and the word 'Clusters'. A large orange arrow points from this screen to the right, where a detailed 'Create a cluster' dialog box is displayed.

Create a cluster

Name: tax-report-processing

Zone: us-east1-b

Master node
Contains the YARN Resource Manager, HDFS NameNode, and all job drivers

Machine type: n1-standard-4 (4 vCPU, 15.0 GB ...)

Cluster mode: Standard (1 master, N workers)

Primary disk size (minimum 10 GB): 500 GB

Worker nodes
Each contains a YARN NodeManager and a HDFS DataNode.
The HDFS replication factor is 2.

Machine type: n1-standard-4 (4 vCPU, 15.0 GB ...)

Nodes (minimum 2): 2

Primary disk size (minimum 10 GB): 500 GB

Local SSDs (0-8): 0 x 375 GB

YARN cores: 8

YARN memory: 24.0 GB

The name needs to be unique within your project

[← Create a cluster](#)

CHOOSE SOMETHING YOU WILL REMEMBER

Name [?](#)
tax-report-processing

Zone [?](#)
us-east1-b

Master node
Contains the YARN Resource Manager, HDFS NameNode, and all job drivers

Machine type [?](#) Cluster mode [?](#)
n1-standard-4 (4 vCPU, 15.0 GB ... ▾) Standard (1 master, N workers) ▾

Primary disk size (minimum 10 GB) [?](#)
500 GB

Worker nodes
Each contains a YARN NodeManager and a HDFS DataNode.
The HDFS replication factor is 2.

Machine type [?](#) Nodes (minimum 2) [?](#)
n1-standard-4 (4 vCPU, 15.0 GB ... ▾) 2

Primary disk size (minimum 10 GB) [?](#) Local SSDs (0-8) [?](#)
500 GB 0 x 375 GB

YARN cores [?](#) YARN memory [?](#)
8 24.0 GB

One cluster per job

[← Create a cluster](#)

**CHOOSE SOMETHING YOU WILL REMEMBER,
SUCH AS WHAT YOU ARE GOING TO USE THE
CLUSTER FOR**

Name [?](#)
tax-report-processing

Zone [?](#)
us-east1-b

Master node
Contains the YARN Resource Manager, HDFS NameNode, and all job drivers

Machine type [?](#) **Cluster mode** [?](#)

Primary disk size (minimum 10 GB) [?](#)
500 GB

Worker nodes
Each contains a YARN NodeManager and a HDFS DataNode.
The HDFS replication factor is 2.

Machine type [?](#) **Nodes (minimum 2)** [?](#)

Primary disk size (minimum 10 GB) [?](#) **Local SSDs (0-8)** [?](#)
500 GB 0 x 375 GB

YARN cores [?](#) **YARN memory** [?](#)
8 24.0 GB

The zone is very, very important

[← Create a cluster](#)

Name [?](#)
tax-report-processing

Zone [?](#)
us-east1-b **THIS IS THE ZONE ... WHY IS IT SO IMPORTANT?**

Master node
Contains the YARN Resource Manager, HDFS NameNode, and all job drivers

Machine type [?](#) Cluster mode [?](#)
n1-standard-4 (4 vCPU, 15.0 GB ...) Standard (1 master, N workers)

Primary disk size (minimum 10 GB) [?](#)
500 GB

Worker nodes
Each contains a YARN NodeManager and a HDFS DataNode.
The HDFS replication factor is 2.

Machine type [?](#) Nodes (minimum 2) [?](#)
n1-standard-4 (4 vCPU, 15.0 GB ...) 2

Primary disk size (minimum 10 GB) [?](#) Local SSDs (0-8) [?](#)
500 GB 0 x 375 GB

YARN cores [?](#) YARN memory [?](#)
8 24.0 GB

The zone is where the compute nodes will live



Match your data location with your compute location (same region)



Three cluster configurations possible

[← Create a cluster](#)

Name [?](#)
tax-report-processing

Zone [?](#)
us-east1-b

Master node
Contains the YARN Resource Manager, HDFS NameNode, and all job drivers

Machine type [?](#) Cluster mode [?](#)
n1-standard-4 (4 vCPU, 15.0 GB ... ▾) Standard (1 master, N workers) ▾

Primary disk size (minimum 10 GB) [?](#)
500 GB

Worker nodes
Each contains a YARN NodeManager and a HDFS DataNode.
The HDFS replication factor is 2.

Machine type [?](#) Nodes (minimum 2) [?](#)
n1-standard-4 (4 vCPU, 15.0 GB ... ▾) 2

Primary disk size (minimum 10 GB) [?](#) Local SSDs (0-8) [?](#)
500 GB 0 x 375 GB

YARN cores [?](#) YARN memory [?](#)
8 24.0 GB

THE MASTER NODE MANAGES THE CLUSTER
CHOOSE BETWEEN:

1. SINGLE NODE (FOR EXPERIMENTATION)
2. STANDARD (1 MASTER ONLY)
3. HIGH AVAILABILITY (3 MASTERS)

HDFS file system available, but don't use it

[← Create a cluster](#)

Name [?](#)
tax-report-processing

Zone [?](#)
us-east1-b

Master node
Contains the YARN Resource Manager, HDFS NameNode, and all job drivers

Machine type [?](#) Cluster mode [?](#)
n1-standard-4 (4 vCPU, 15.0 GB ... ▾) Standard (1 master, N workers) ▾

Primary disk size (minimum 10 GB) [?](#)
500 GB

Worker nodes
Each contains a YARN NodeManager and a HDFS DataNode.
The HDFS replication factor is 2.

Machine type [?](#) Nodes (minimum 2) [?](#)
n1-standard-4 (4 vCPU, 15.0 GB ... ▾) 2

Primary disk size (minimum 10 GB) [?](#) Local SSDs (0-8) [?](#)
500 GB 0 x 375 GB

YARN cores [?](#) YARN memory [?](#)
8 24.0 GB

MACHINE TYPE, NUMBER OF WORKERS

DISK PERFORMANCE SCALES WITH SIZE!!!
DON'T USE HDFS TO STORE INPUT/OUTPUT DATA

Preemptible workers can be a good deal

Preemptible worker nodes ?

Each contains a YARN NodeManager. HDFS does not run on preemptible nodes.
Machine type is copied from the Worker section.

Nodes ?

10

IMAGINE YOUR JOB NEEDS 10 MACHINES FOR 130 MINUTES

Cloud Storage staging bucket (Optional) ?

YOUR CLUSTER HAS 10 STANDARD WORKERS AND ...

YOU MANAGE TO GET 10 PREEMPTIBLE MACHINES

- 1. YOUR JOB WILL NOW FINISH IN 65 MINUTES!**
- 2. IT WILL COST 40% LESS OVERALL!**

Network ?

default

Image version ?

▼

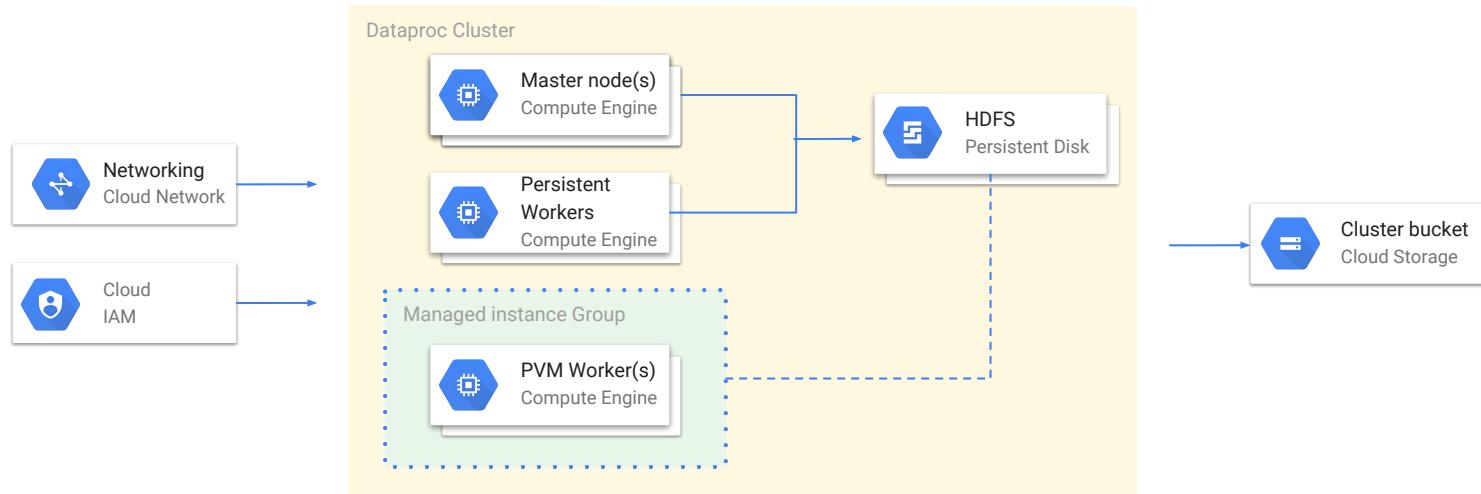
Initialization actions ?

gs://mybucket/action-xyz

Project access ?

Allow API access to all Google Cloud services in the same project. [Learn more](#)

Dataproc manages joins/leaves of preemptible instances



Can customize the Dataproc cluster

Preemptible worker nodes ?

Each contains a YARN NodeManager. HDFS does not run on preemptible nodes.
Machine type is copied from the Worker section.

Nodes ?

10

Cloud Storage staging bucket (Optional) ?

Network ?

default

Image version ?

Initialization actions ?

gs://mybucket/action-xyz

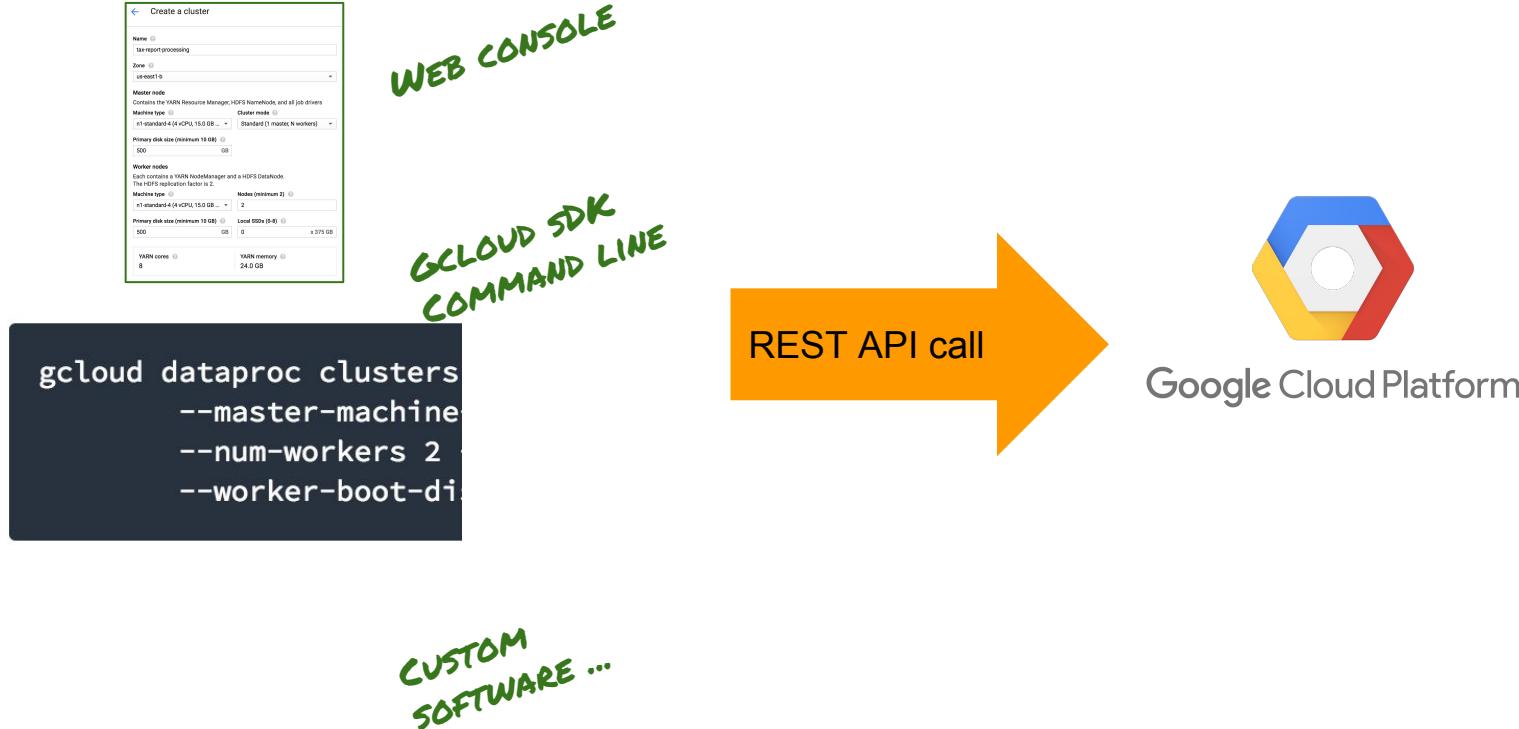
Project access ?

Allow API access to all Google Cloud services in the same project. [Learn more](#)

CAN SET UP FIREWALL RULES ETC.

CAN ALSO INSTALL CUSTOM SOFTWARE ON THE DATAPROC WORKERS AND MASTER

Most things you can do from the web console...



Creating a cluster using gcloud SDK

```
gcloud dataproc clusters create my-second-cluster --zone us-central1-a \
--master-machine-type n1-standard-1 --master-boot-disk-size 50 \
--num-workers 2 --worker-machine-type n1-standard-1 \
--worker-boot-disk-size 50
```

CONTEXT-SPECIFIC HELP

```
gcloud dataproc --help
gcloud dataproc clusters --help
gcloud dataproc clusters create --help
```

Lab - Leveraging Unstructured Data : Part 1

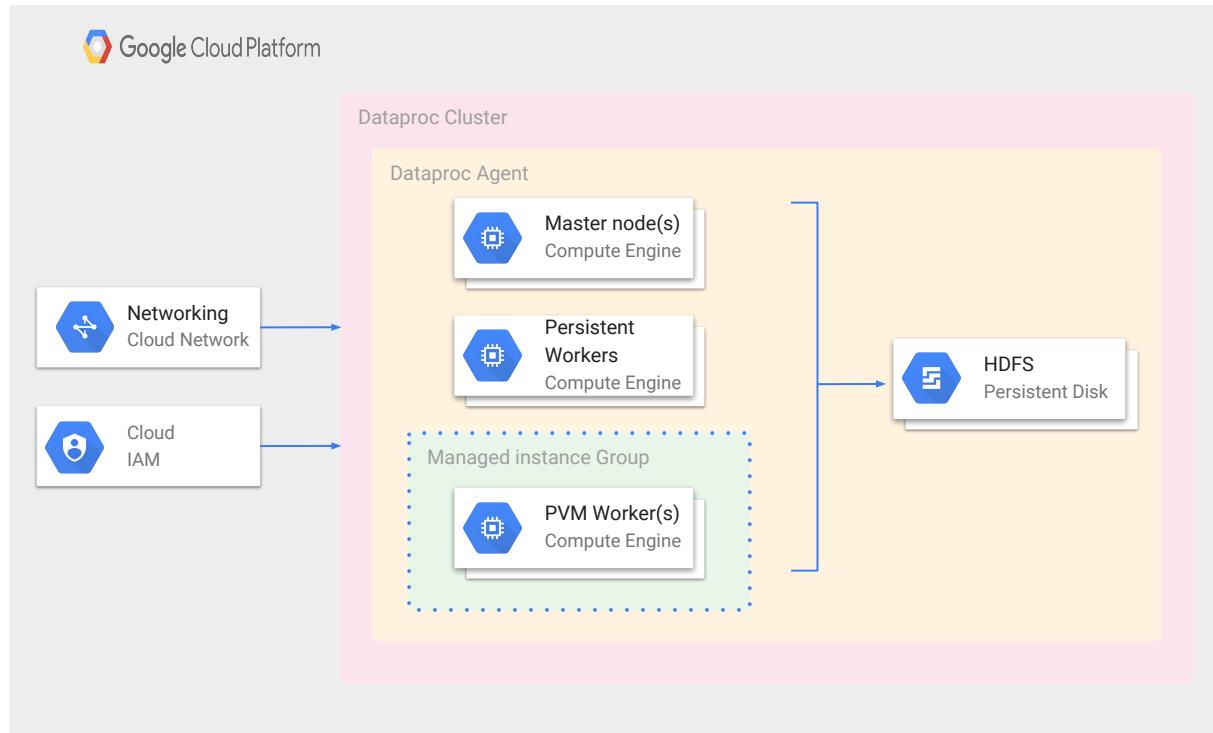
You will learn how to:

- Create a Dataproc cluster from the Web console
- SSH into the cluster
- Add a firewall rule that allows access to your cluster from the browser
- Create, manage and delete Dataproc clusters from the CLI

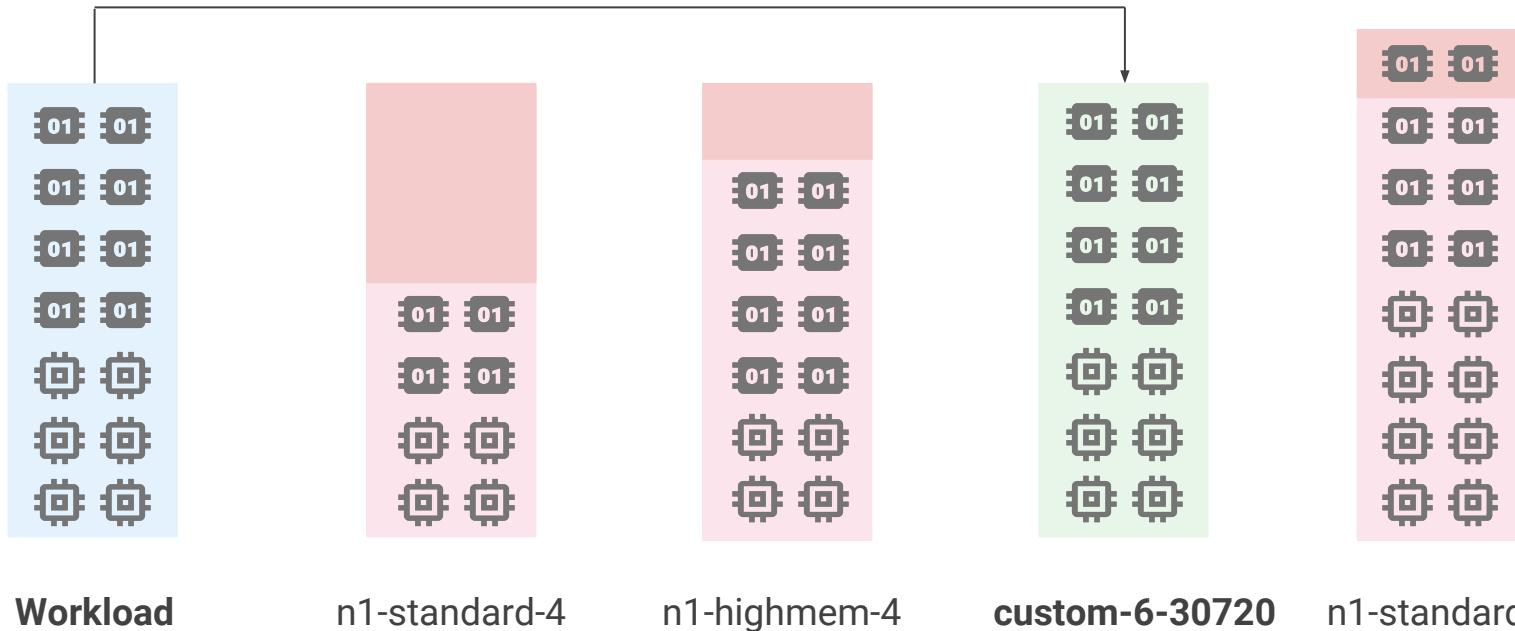
Agenda

Custom machine types

Cloud Dataproc hardware architecture



Right-size your workload using machine types



Creating the custom machine type...

```
gcloud dataproc clusters create test-cluster /  
  --worker-machine-type custom-6-30720 /  
  --master-machine-type custom-6-23040
```

*6 CPUs
30 GB * 1024 = 30720*

You can find the name from the web console

Compute Engine

[VM instances](#)

[Instance groups](#)

Instance templates

[Disks](#)

[Snapshots](#)

[Images](#)

[Metadata](#)

[Health checks](#)

[Zones](#)

[Operations](#)

[Quotas](#)

[Settings](#)

[Create an instance template](#)

Name: `instance-template-1`

Machine type: 1 vCPU, 3.75 GB memory [Customize](#)

Boot disk: New 10 GB standard persistent disk
Image: Debian GNU/Linux 8.3 (jessie) [Change](#)

Firewall: Allow HTTP traffic Allow HTTPS traffic

Project access: Allow API access to all Google Cloud services in the same project. [Learn more](#)

[Management, disk, networking, access & security options](#)

Compute Engine

[VM instances](#)

[Instance groups](#)

Instance templates

[Disks](#)

[Snapshots](#)

[Images](#)

[Metadata](#)

[Health checks](#)

[Zones](#)

[Operations](#)

[Quotas](#)

[Settings](#)

[Create an instance template](#)

Name: `example-dataproc-template`

Machine type: [Cores](#) [Basic view](#)

Memory: 22.5 GB (5.5 - 39)

Choosing a machine type [Learn more](#)

Boot disk: [New 10 GB standard persistent disk](#)
Image: Debian GNU/Linux 8.3 (jessie) [Change](#)

Firewall: [Add tags and firewall rules to allow specific network traffic from the Internet](#)

This is the REST request with the parameters you have selected.

```
POST https://www.googleapis.com/compute/v1/projects/google.com:hadoop-demo/global/instanceTemplates/example-dataproc-template
{
  "name": "instance-template-1",
  "description": "",
  "properties": {
    "machineType": "custom-6-23040",
    "metadata": {
      "items": []
    },
    "tags": {
      "items": []
    },
    "disks": [
      {
        "type": "PERSISTENT",
        "boot": true,
        "mode": "READ_WRITE",
        "autoDelete": false
      }
    ],
    "networkInterfaces": [
      {
        "name": "nic0",
        "network": "global/networks/default",
        "accessConfigs": [
          {
            "name": "External IP",
            "type": "ONE_TO_ONE_NAT"
          }
        ]
      }
    ],
    "labels": {
      "items": []
    }
  }
}
```

cloud.google.com

