Google Cloud

# Introduction to Cloud Dataproc

## Data Engineering on Google Cloud Platform



**Notes:**

25 slides + 1 lab: 1 hour

# 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 |



**Notes:**

Examples of each?

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

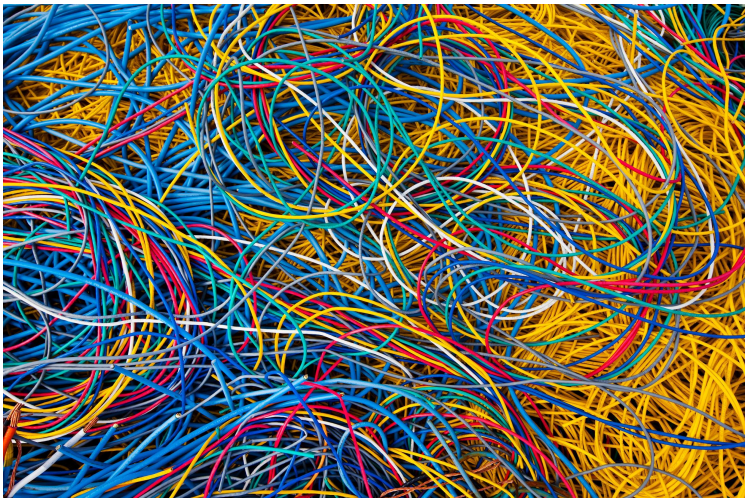
| | |
|---|---|
| Data you analyze today | Data you collect but don't analyze (WHY?) |
| Data you could collect but don't | Data from partners and 3rd parties |



**Notes:**

Ask them what kinds of data they have that they don't analyze. A common reason is that it is hard to analyze—it is unstructured.

Unstructured data accounts for 90% of enterprise data*





**Notes:**

Source: IDC as quoted in https://www.wired.com/insights/2014/07/rewiring-tackle-unstructured-data/

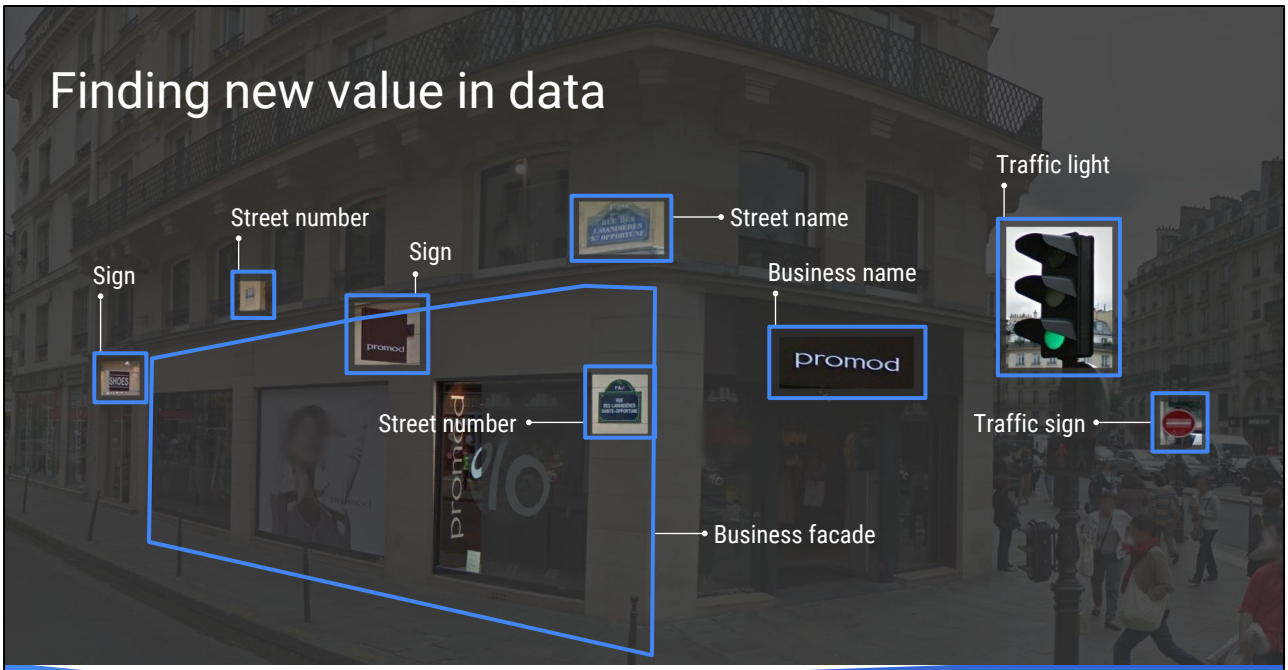



**Notes:**

At the time that Google sent cars out to create street view, they didn't know what will be possible to do later w/ the data. Now we're discovering new opportunities in the data, thanks to better/cheaper/larger processing systems.

Same is true for you. The fact that you're here today tells me that you'll have a lot more processing capabilities very soon.

# 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?



**Notes:**

Many of the human analysis involves humans creating ad-hoc queries on structured and unstructured data. They require human reasoning. The use cases on the left are perfect for human analysts because they are high-value and involve a lot of insight. They are one-offs.

# 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?



**Notes:**

The stuff on the right is not scalable unless we can use a computer to do it. They are very repeatable. While we could do it with machine learning, most often, the stuff on the right can be done by fancy counting.

The ones on the right are great candidates for big data processing, but these are structured.

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

| Human | Easy counting problems | Harder counting problems |
| --- | --- | --- |
| Real-time insight into supply chain operations. Which partner is causing issues? | Did error rates decrease after the bug fix was applied? | Are programmers checking in low-quality code? |
| Drive product decisions. How do people really use feature X? | Which stores are experiencing long delays in payment processing? | Which stores are experiencing lacking of parking space? |

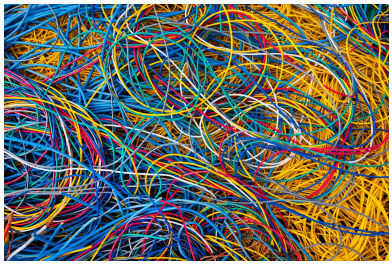


**Notes:**

Compare these with the ones on the previous slide. Structured vs. unstructured.

Low-quality could be determined by bad sentiment in code reviews and often through programmer's own negative comments in the code. They are checking in the code because they need to move on to their next project …. But there are also tools out there that will look for code-smells. Those tools can be run at scale on Dataproc ...

# Build on top of Google



Images
Audio
Video
Free-form text

ML API

Places
Labels
People
Events
...

Cloud ML Engine



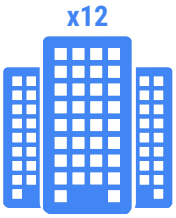
**Notes:**

https://pixabay.com/en/list-zettelbox-note-leaves-stack-1925395/ (cc0)

# 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



**Notes:**

But … we don't have that much data … do we really need to worry about anything more than in-memory spark?

# 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

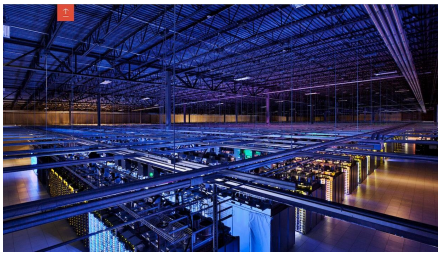


**Notes:**

Yes, you do. And you will :)

# How do you process large amounts of data?

Scaling up



Scaling out

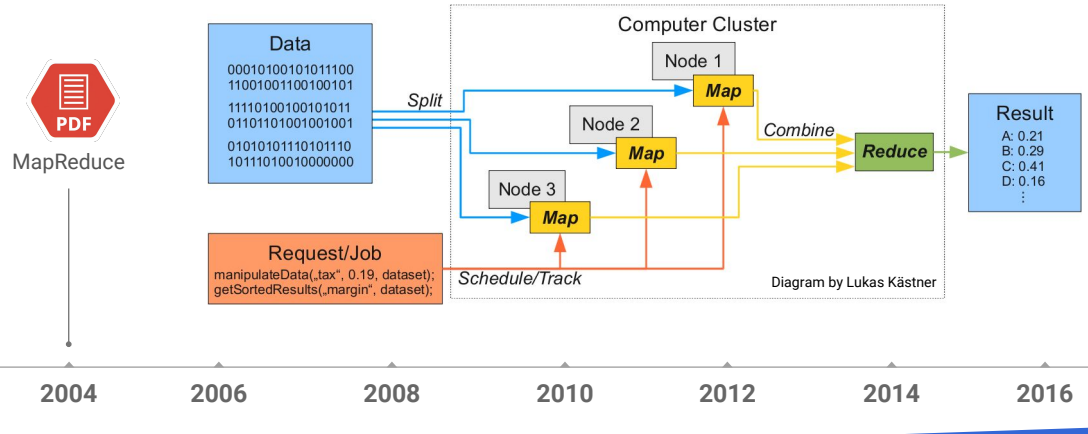


**Notes:**

Two options. The second one is infinitely scalable, but it is harder to program. Because it involves distributed training.

https://pixabay.com/en/server-technology-datacenter-37578/ (cc0)
https://www.google.com/about/datacenters/gallery

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

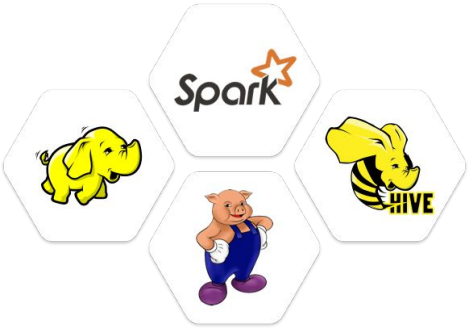





**Notes:**

Diagram source: https://www.flickr.com/photos/lkaestner/4861146813
cc-by-saLukas Kastner

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






**Notes:**

All these come out of the original MapReduce paradigm to process large datasets.

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



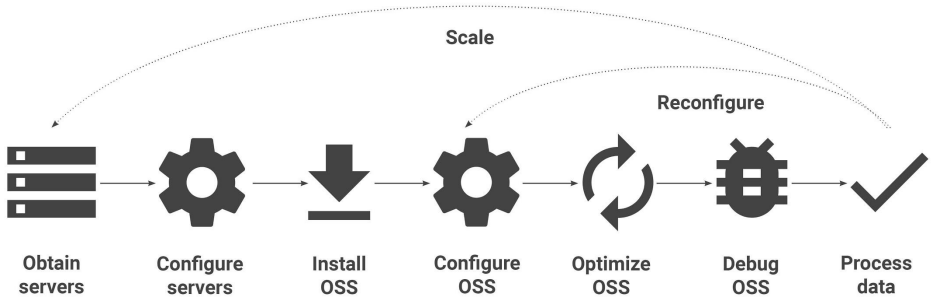

**Notes:**

You may know of Spark as an open source general large scale data processing tool like Apache Hadoop MapReduce, which it is. But a lot of layers have been built on top of that.

There is a full SQL implementation written on top of it, which provides a common DataFrame data model to Scala, Java, SQL, R, and Python.

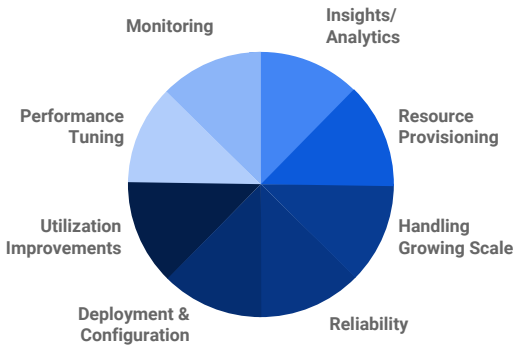
And on top that is the Spark MLlib Spark's Distributed Machine Learning library.

# Typical Spark and Hadoop deployments involve...



# Lots of time is spent on administration and operational issues





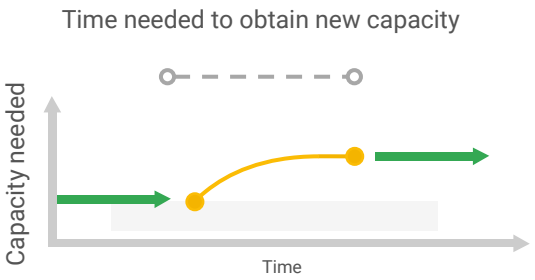



**Notes:**

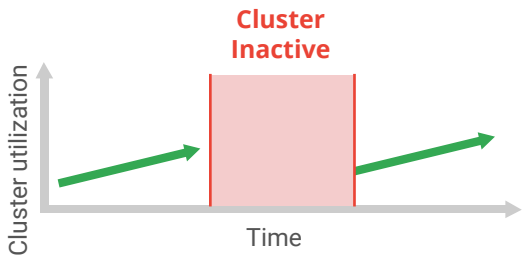
Why? … the following slides

# Scaling can take hours, days, or weeks





# You have to babysit utilization



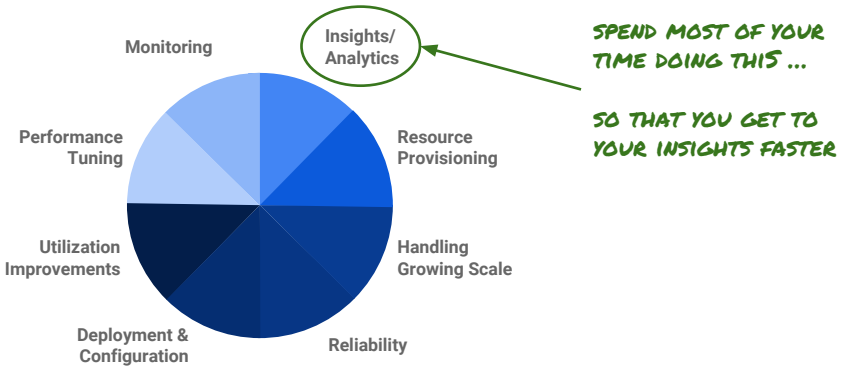

ENSURING CLUSTER IS ALWAYS BEING UTILIZED IS HARD



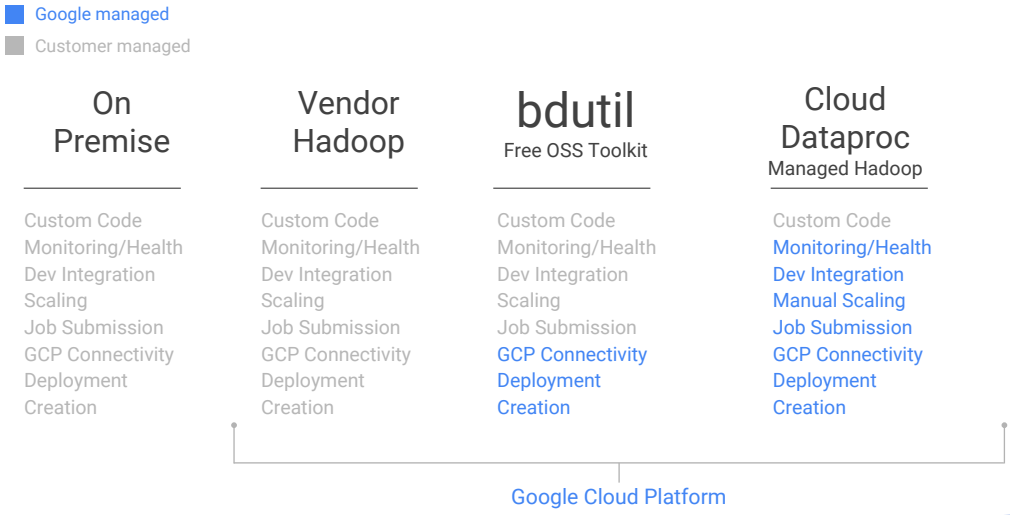
**Notes:**

Requires effort to pack clusters so the it does not have periods of inactivity and wasted resources.

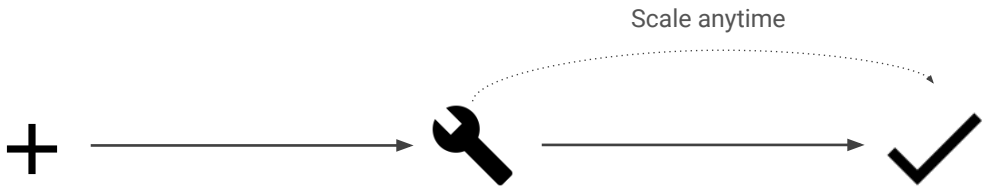
# But you want to focus on insights and analytics



# Dataproc eases Hadoop management



# Typical Dataproc deployments involve...




Create cluster

**(0 seconds)**

Configure cluster

**(20 seconds)**

Use cluster

**(90 seconds)**

# 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



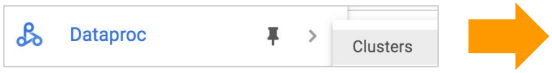
**Notes:**

We are going to look at #1 now. We will look at the others in later chapters.

# Agenda

Creating a Dataproc cluster + Lab

# Create a cluster from the web console



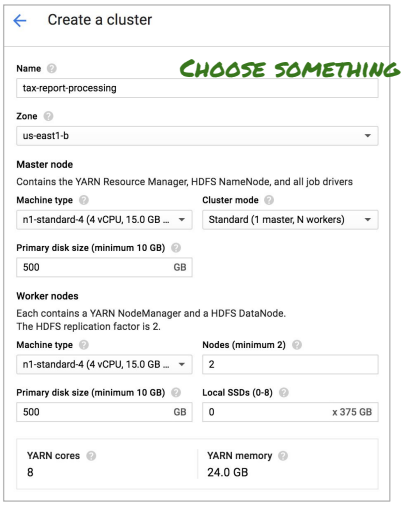




**Notes:**

From the left-hand side of the menu.

# The name needs to be unique within your project

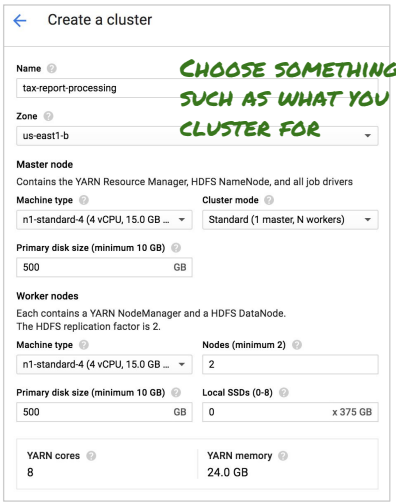





**Notes:**

“Tax-report-processing”. I’ll process tax reports on this cluster. Only that. One cluster per job. When the job is done, I’ll delete the cluster.

# One cluster per job



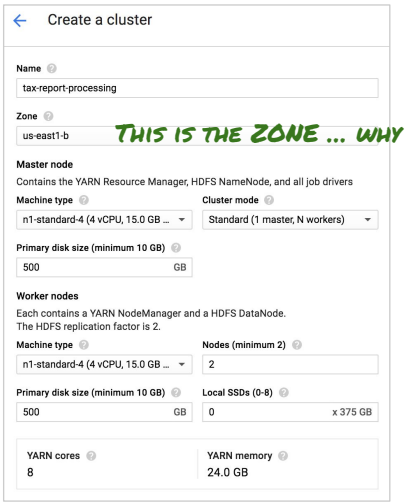

CHOOSE SOMETHING YOU WILL REMEMBER, SUCH AS WHAT YOU ARE GOING TO USE THE CLUSTER FOR



**Notes:**

When the job is done, I'll delete the cluster. This way, I get to maximize the use of the cluster. We won't put multiple jobs on the cluster because that will lead us down the path of figuring out how to manage resource usage and deal with idle times ...

# The zone is very, very important

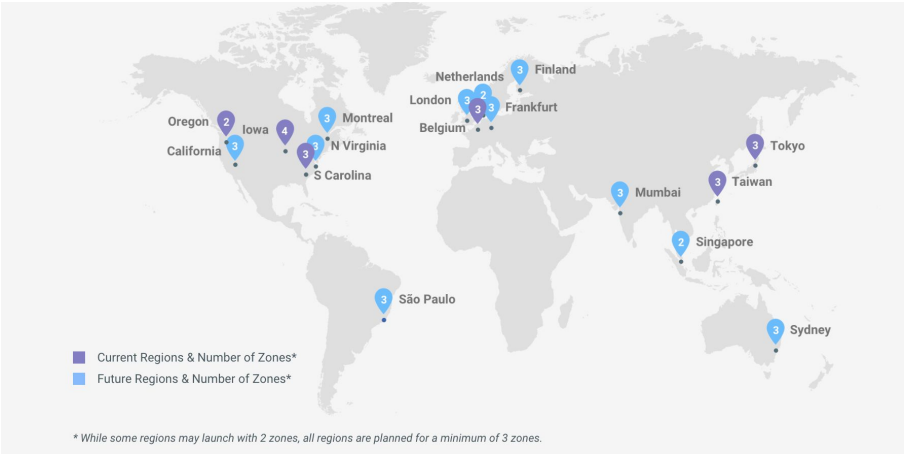





**Notes:**

Why is it important?

# The zone is where the compute nodes will live

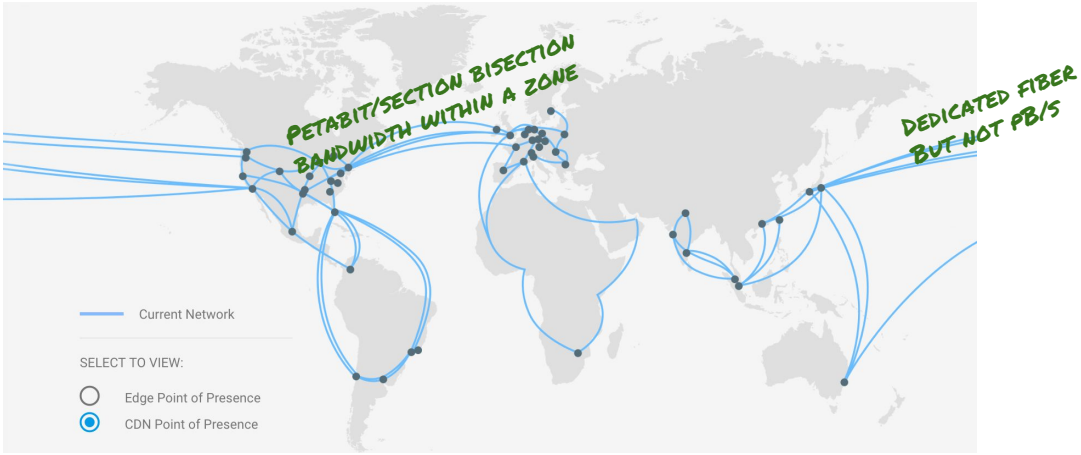





**Notes:**

Image from https://cloud.google.com/about/locations/#regions-tab as of March 2017

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



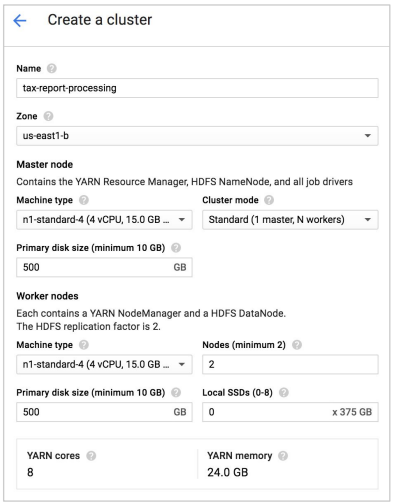



**Notes:**

Image from https://cloud.google.com/about/locations/#regions-tab as of March 2017

You have multiple zones within a region and the data in GCS is replicated across zones. So, you can pick any zone within the region where your data resides. On the other hand, if your data are only in asia and you decide to run the processing in us-central, you are going to face performance issues.

# Three cluster configurations possible




THE MASTER NODE MANAGES THE CLUSTER
CHOOSE BETWEEN:

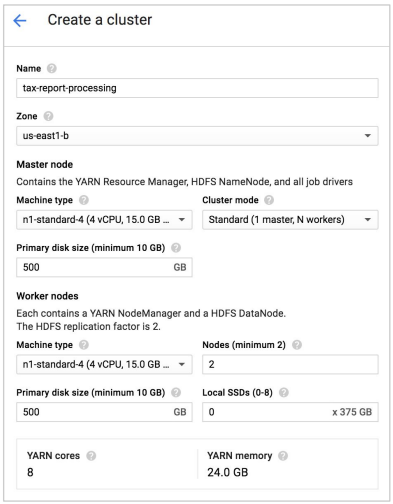
1. SINGLE NODE (FOR EXPERIMENTATION)
2. STANDARD (1 MASTER ONLY)
3. HIGH AVAILABILITY (3 MASTERS)



**Notes:**

Single node = no workers. Normally, you will just go with standard because we are going to create job-specific clusters. For long-running jobs and multiple jobs per cluster, you could go with high-availability which provides the ability to distribute management and also provides failover.

# HDFS file system available, but don't use it




MACHINE TYPE, NUMBER OF WORKERS

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



**Notes:**

You can use HDFS, but don't … you want to delete the cluster after your job is done.

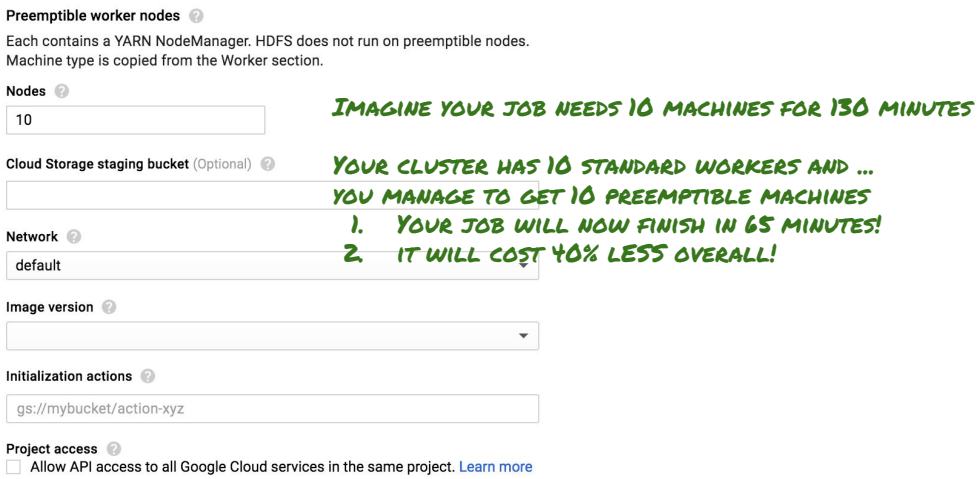
Even if you keep all your data in HDFS, you want to think carefully about the size of your disk.

The disk is used for temporary staging, and disk performance scales with size …

A good starting point: 500 GB per n1-standard-4 (this is the Dataproc default). It derives from ~3MB per cpu-second based on the Terasort benchmark.

Keep your data on GCS. We'll talk about why in Chapter 2.

# Preemptible workers can be a good deal



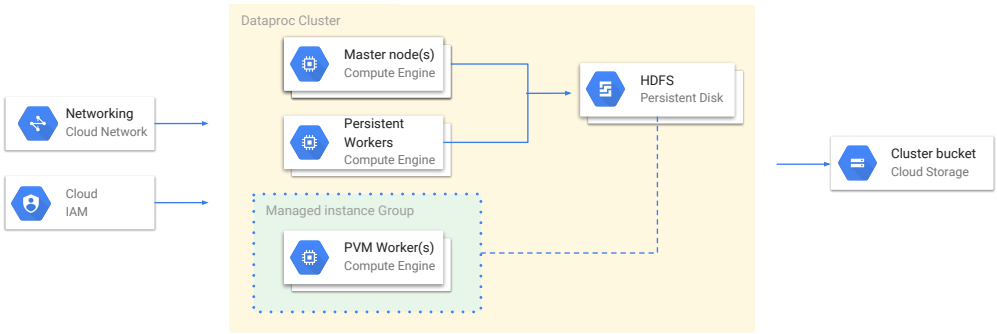



**Notes:**

Not guaranteed that you will be able to get preemptible machines

Each preemptible machine is 20% of the cost of the regular machine. So you have 10 machines at 100% of charge and 10 machines at 20% of charge.

# Dataproc manages joins/leaves of preemptible instances

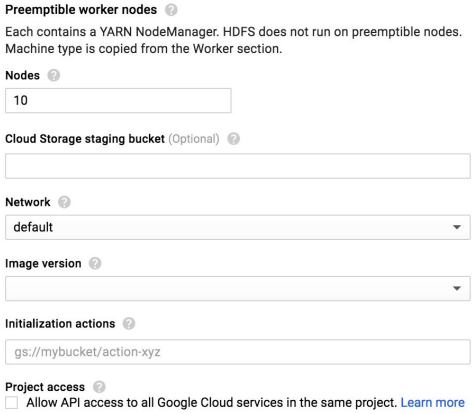





**Notes:**

There is nothing for you to manage. Dataproc will do the right things regarding staging data etc on HDFS. (The PVM itself won't have any HDFS nodes -- PVMs are best for compute-intensive jobs, but jobs running on it will have read access to HDFS)

# Can customize the Dataproc cluster




Can set up firewall rules etc.

Can also install custom software on the Dataproc workers and Master



**Notes:**

We'll do this when we install and run Datalab on the cluster

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

WEB CONSOLE

GCLOUD SDK
COMMAND LINE

```
gcloud dataproc clusters
        --master-machine
        --num-workers 2
        --worker-boot-di
```

REST API call

Google Cloud Platform

CUSTOM
SOFTWARE ...



**Notes:**

The web console makes a REST API call. That same REST API call can be made from the gcloud command-line client or even from your own software.

# 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
```



**Notes:**

This is very useful to script the creation, deletion, etc. of clusters.Adding firewall rules, etc.

Don’t memorize the command … you can get context-specific help.

# Lab - Leveraging Unstructured Data : Part 1

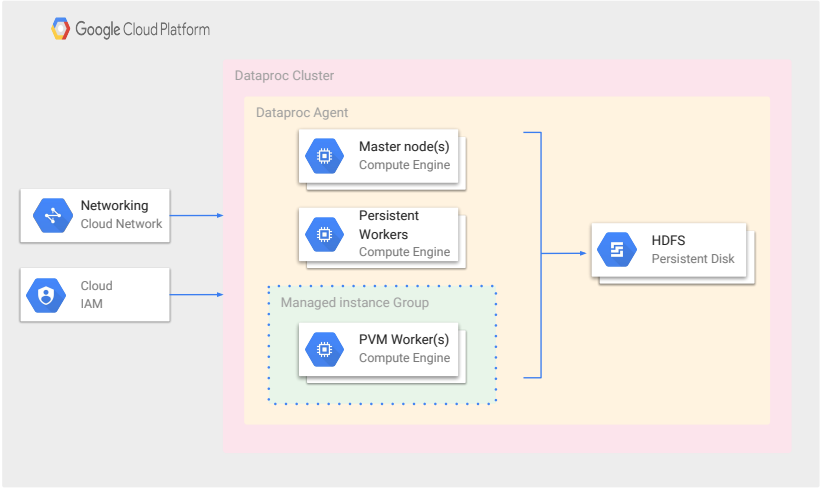
You will learn how to:

- 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

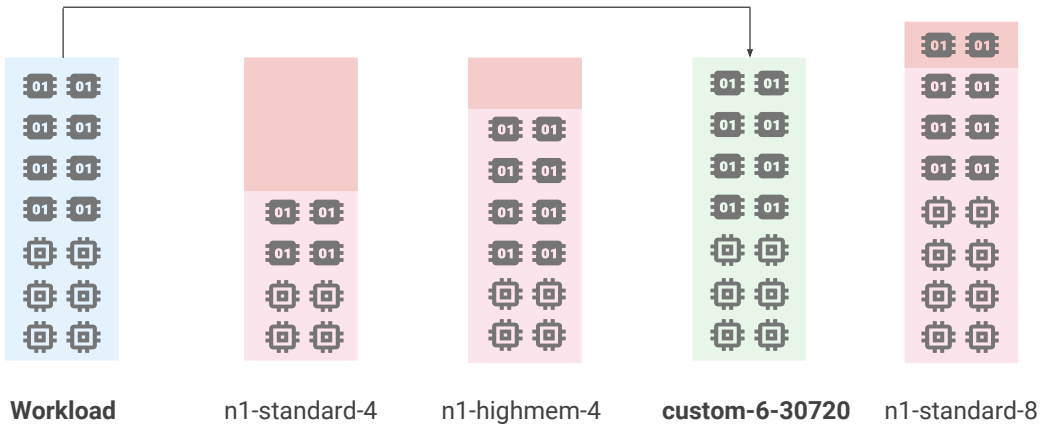





**Notes:**

Use Cloud IAM and networking to access the machines.

# Right-size your workload using machine types






**Notes:**

The idea is that n1-standard-4 is too small for the job.
N1-highmem-4 has the necessary memory, but lacks the cpu you need.
N1-standard-8 has everything, but you are overprovisioning cpus.
The custom then meets the need.

# Creating the custom machine type...

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

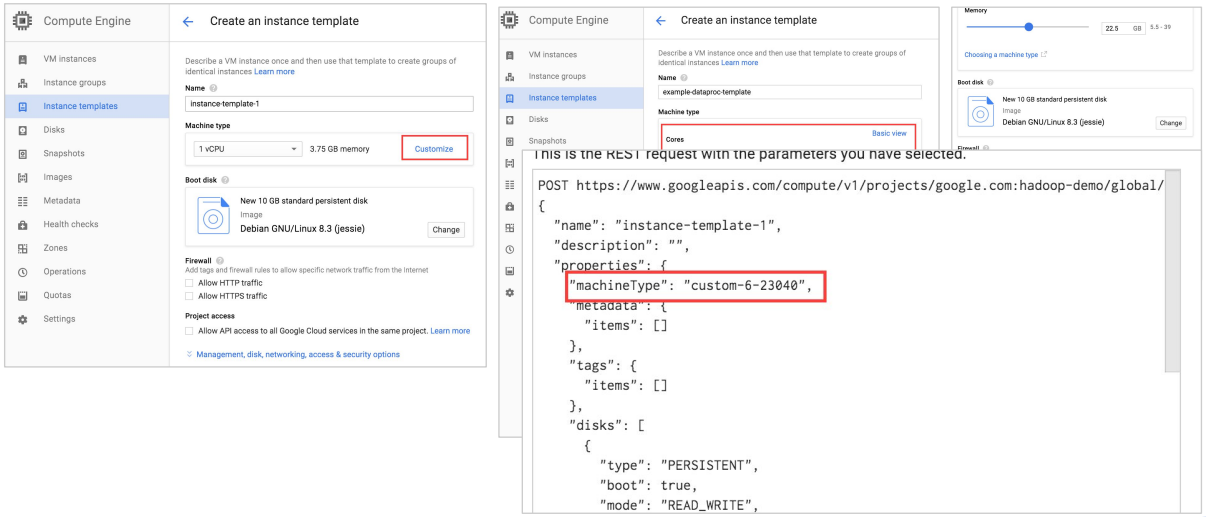
6 CPUs
30 GB * 1024 = 30720



**Notes:**

This is only possible using the gcloud SDK currently.You can not do it from the web console.

# You can find the name from the web console






**Notes:**

Look at the workflow in https://cloud.google.com/dataproc/docs/concepts/custom-machine-types and feel free to make it a demo

Essentially go through the process of creating a custom template, choose # of cpu & memory, get equivalent REST command, and look for custom machine type.Or you can do a little bit of math ;)


cloud.google.com
Image by Connie Zhou