INTRO TO THE HANDS-ON

- A public cloud platform: Google Cloud Platform
- A remote development environment : GitHub Codespaces

SDD - Data Engineering
https://supaerodatascience.github.io/deep-learning/

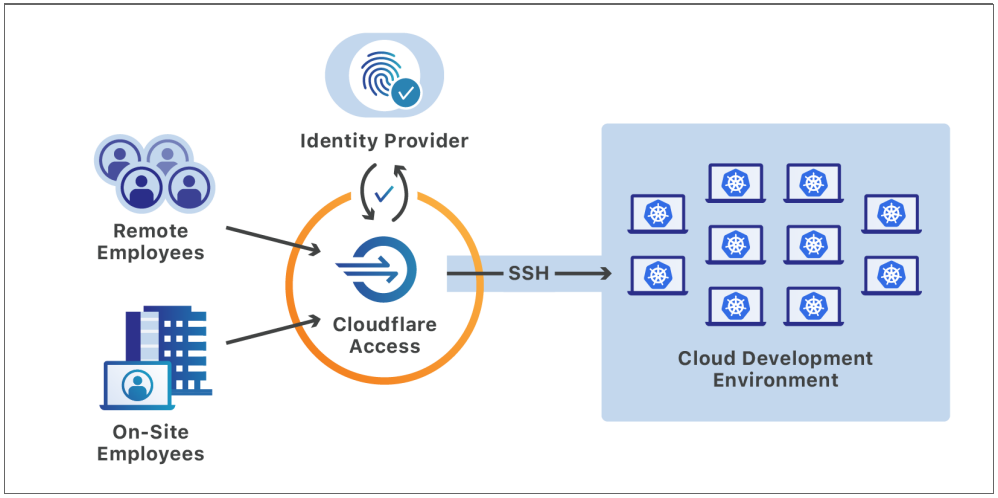
REMOTE DEVELOPMENT

WHY "REMOTE DEVELOPMENT" ?

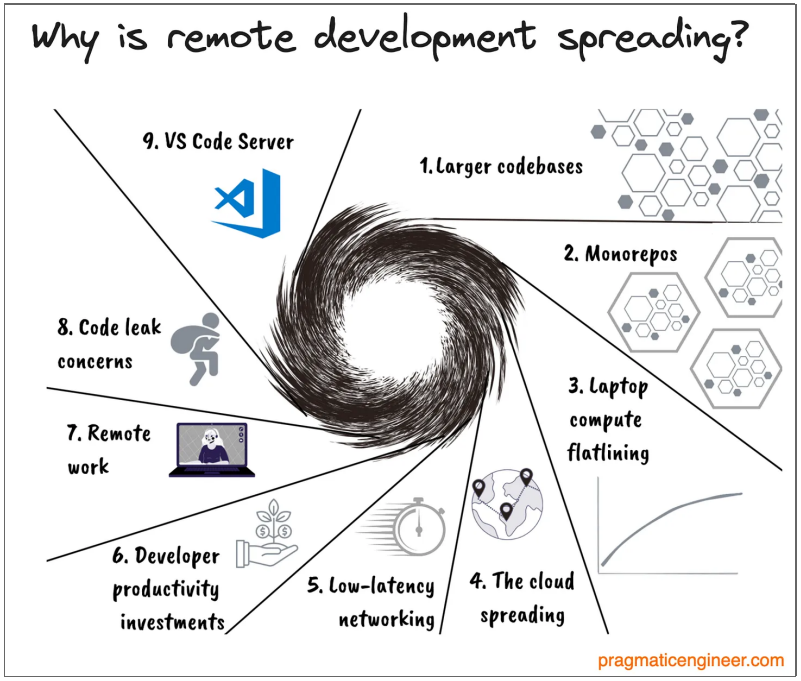
AI / Data Science = Data + Compute + Software

- Your job consist in handling huge volume of data
- Your job requires high computational resources
- You're working as a team with centralized computing platform
- You're working remotely

KEY USE CASE

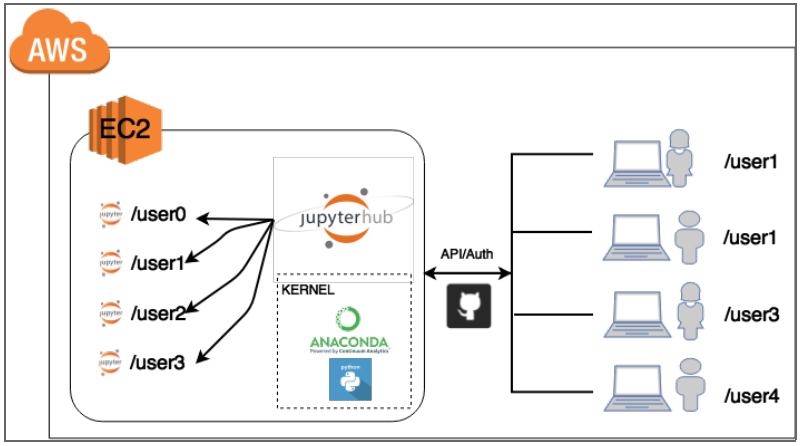

Identity Provider
Remote Employees
On-Site Employees
Cloudflare Access
SSH
Cloud Development Environment

KEY USE CASE



https://newsletter.pragmaticengineer.com/p/cloud-development-environments-why-now

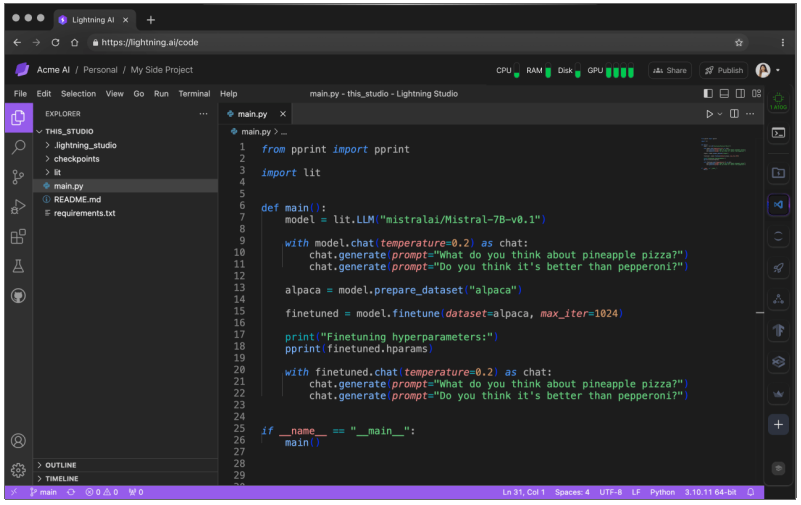
YOUR FUTURE DAILY ROUTINE



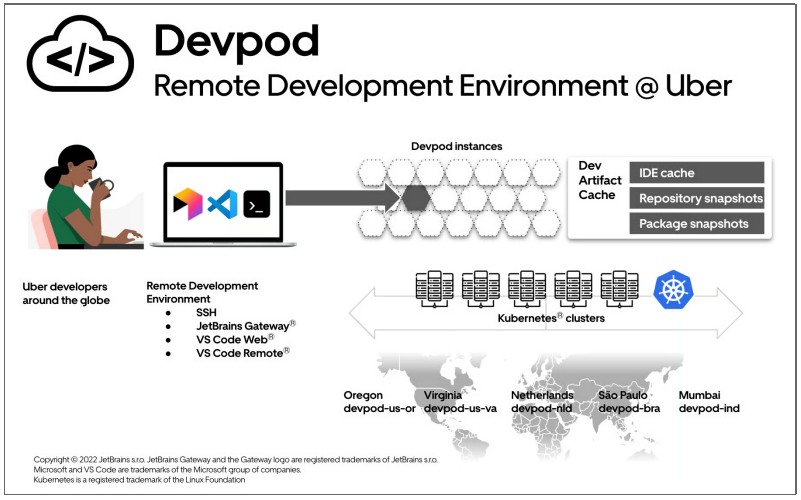
Another example (more managed) **Google Colab**

YOUR FUTURE DAILY ROUTINE

An AI-centric example, **https://lightning.ai/studios**



YOUR FUTURE DAILY ROUTINE



Uber Blog describing their way of working

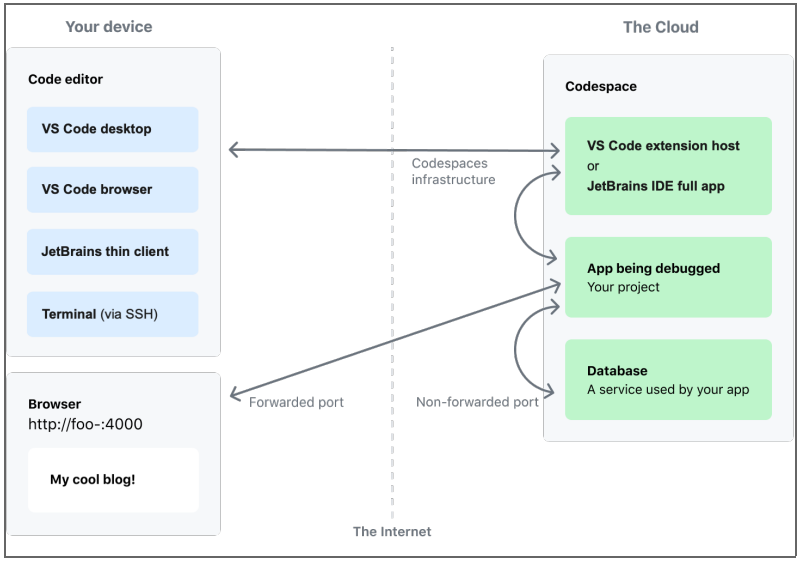
PROBLEMATICS

- How to transfer code ?
- How to interact with the machines ?
- How to get access to the data ?


Codespaces
mona-github-github-g58jpq2w5w7.github.dev
EXPLORER
GITHUB
.devcontainer
build-devcontainer.sh
devcontainer.json
on-create-command.sh
post-attach-command.sh
setup-devcontainer.sh
ssrh
welcome_message.txt
.github
.vscode
app
api
assets
components
controllers
helpers
jobs
mailers
models
emotion.rb
devcontainer.json
attr_reader :label
attr_reader :pronounceable_label
Public: Get the Emoji that this reaction's content represents.
#
Returns an Emoji.
attr_reader :emoji_character
def initialize(content:, label: nil, pronounceable_label: nil, emoji_character: nil)
@content = content
@label = label || @content
@pronounceable_label = pronounceable_label || @label
@emoji_character = emoji_character || Emoji.find_by_alias(@content)
@platform_enum = @pronounceable_label.gsub(" ", "_").upcase
freeze
end
Emotion.create(content: "+1", pronounceable_label: "thumbs up")
Emotion.create(content: "-1", pronounceable_label: "thumbs down")
CPUs up to
32 cores
Memory up to
64 GB
main
Current branch
css-changes
tabbed-nav
DEBUG CONSOLE

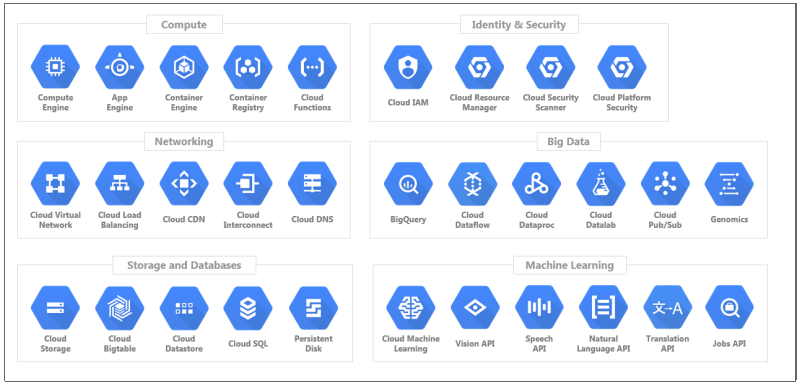
GITHUB CODESPACES

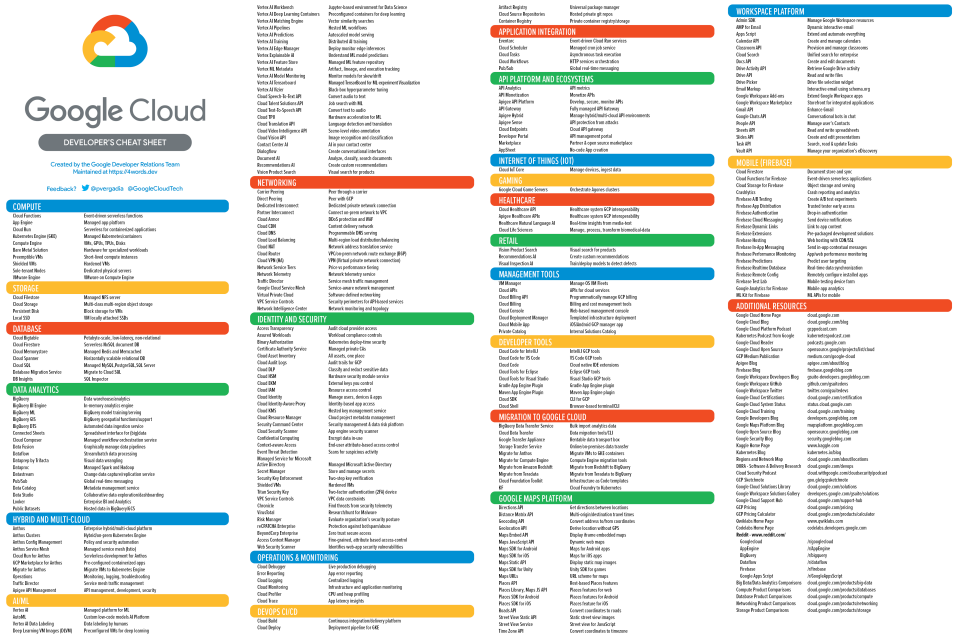
- **Github Codespaces** : A managed development environment by Microsoft Azure
- A virtual machine and a **containerized development environment**
- A lot of built-in bonuses including "in-browser" connection & TCP port forwarding with reverse proxy



GOOGLE CLOUD PLATFORM

- One of the main cloud provider
- Behind AWS in SaaS (serverless...)
- More "readable" product line (for a Cloud Provider...)
- Very good "virtual machine" management
  - per second billing
  - fine-grained resource allocation

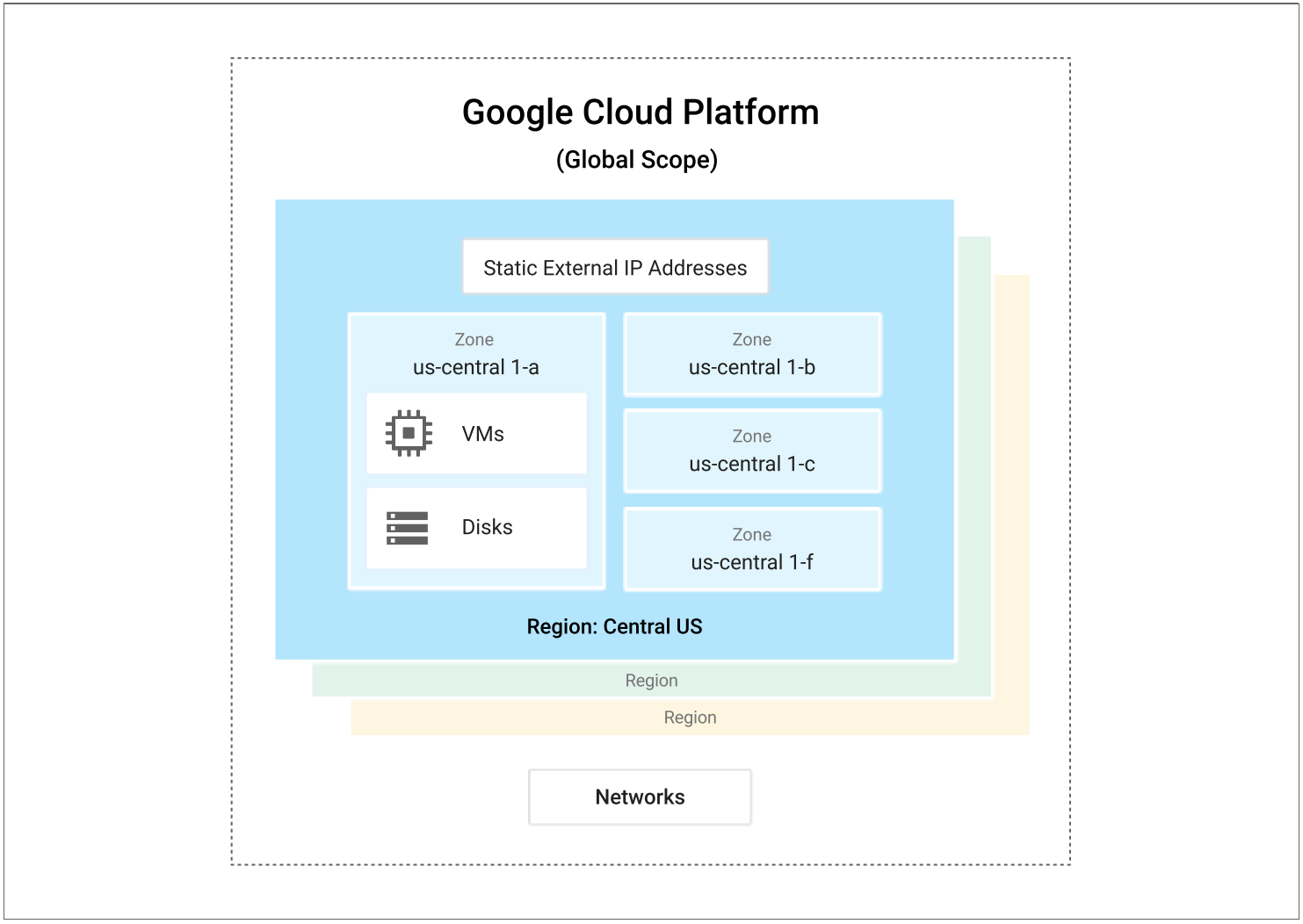

Compute
Compute Engine
App Engine
Container Engine
Container Registry
Cloud Functions
Identity & Security
Cloud IAM
Cloud Resource Manager
Cloud Security Scanner
Cloud Platform Security
Networking
Cloud Virtual Network
Cloud Load Balancing
Cloud CDN
Cloud Interconnect
Cloud DNS
Big Data
BigQuery
Cloud Dataflow
Cloud Dataproc
Cloud Datalab
Cloud Pub/Sub
Genomics
Storage and Databases
Cloud Storage
Cloud Bigtable
Cloud Datastore
Cloud SQL
Persistent Disk
Machine Learning
Cloud Machine Learning
Vision API
Speech API
Natural Language API
Translation API
Jobs API



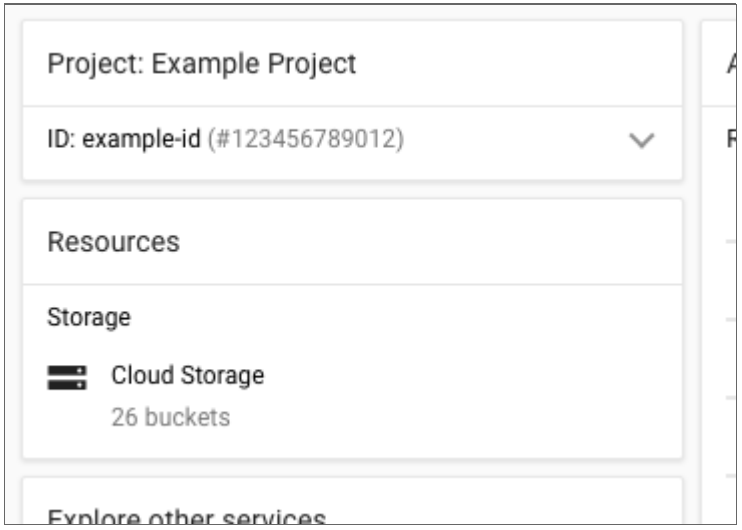
GCP Services in 4 words or less

CONCEPTS

ZONES AND REGIONS

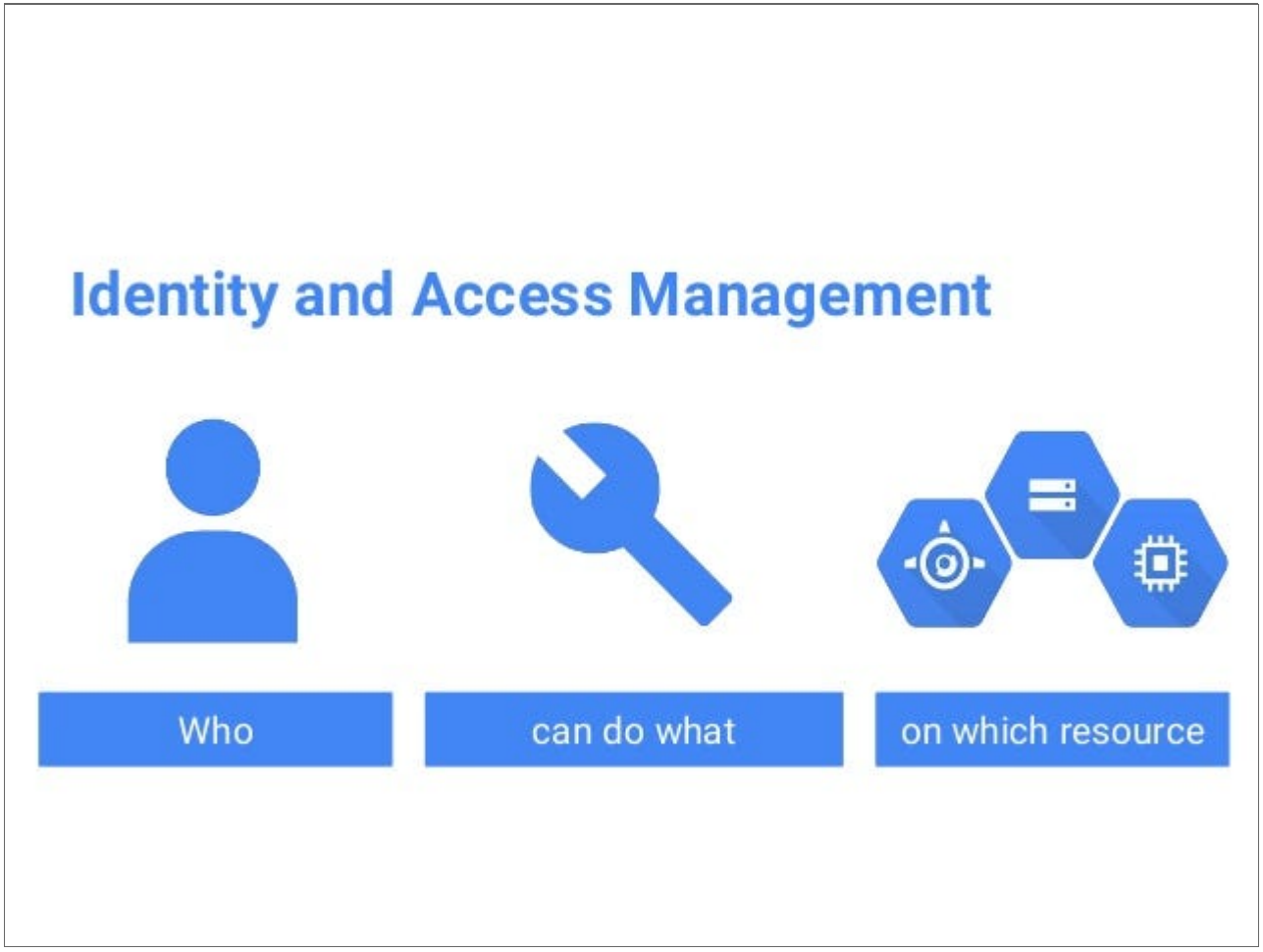


PROJECTS



- Access (Enabling API/Services)
- Ressources (Quota by project)
- Networking
- Billing

CONCEPTS: IDENTITY AND ACCESS MANAGEMENT (IAM)



MAIN PRODUCTS WE ARE GOING TO BE LOOKING AT

- Google Compute Engine (virtual machine solutions)
- Google Cloud Storage (storage solutions)

GOOGLE COMPUTE ENGINE (GCE)

- The VM solution for GCP
- Images: Boot disks for VM instances example: `ubuntu-1804`
- Machine Types: Ressources available to your instance example: `n1-standard-8` (8 vCPU, 30 Gb RAM)
- Storage Options: "Attached disk" that can persist once the instance is destroyed... can be HDD, SDD...
- Preemptible: "Spot instances" on AWS", cheap but can be killed any minute by GCP

GOOGLE CLOUD STORAGE (GCS)

- Cheaper storage than persistent disks
- Can be shared between multiple instances / zones
- Higher latency than local disk
- Data is stored in "buckets" **whose name are globally unique**

GCS STORAGE CLASSES

Class	Use Case	Price
Standard	Frequent access	~$0.02/GB
Nearline	Monthly access	~$0.01/GB
Coldline	Quarterly access	~$0.004/GB
Archive	Yearly access	~$0.0012/GB

Choose based on access frequency!

GCS WITH GSUTIL CLI

```
# List buckets
gsutil ls

# Create bucket (name must be globally
unique!)
gsutil mb gs://my-unique-bucket-name

# Upload file
gsutil cp local_file.csv gs://bucket/path/

# Download file
gsutil cp gs://bucket/path/file.csv ./

# Recursive copy (for folders)
gsutil -m cp -r gs://bucket/data/ ./
local_data/
```

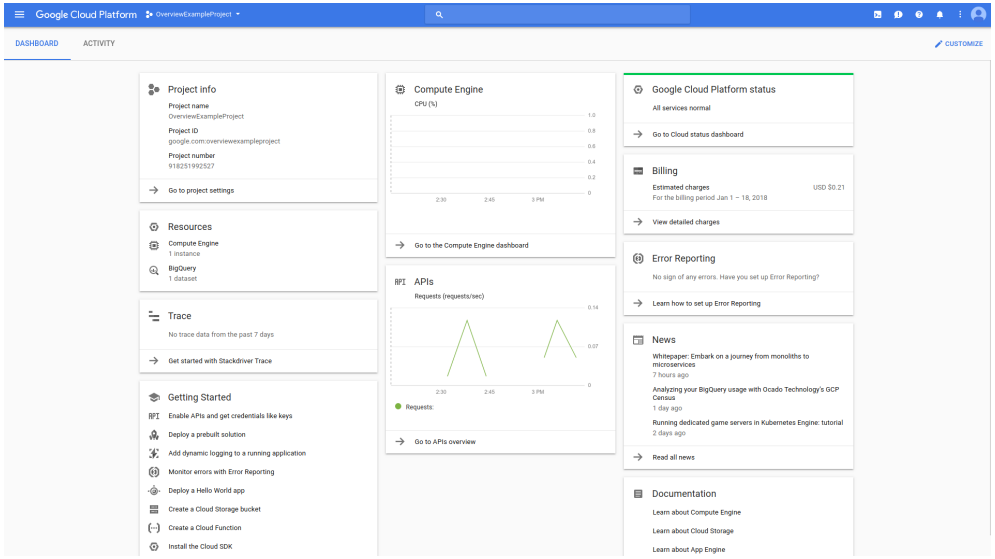
GCS ACCESS FROM VMS

VMs need **access scopes** to use GCS:

```
# When creating VM, add storage scope
gcloud compute instances create my-vm \
    --scopes=storage-rw  # read-write access
to GCS
```

Or use a service account with appropriate IAM roles

INTERACTING WITH GCP: THE CONSOLE



https://console.cloud.google.com

INTERACTING WITH GCP: SDK & CLOUD SHELL

- Using the gcloud CLI: **https://cloud.google.com/sdk/install**
- Using Google Cloud Shell: A small VM instance you can connect to with your browser (that we won't use)

SELF-PACED HANDS-ON

OBJECTIVES

- ✅ Discover GitHub Codespace
- ✅ Create your GCP account, configure your credentials
- Create your first VMs and connect to it via SSH
  - Learn to use Tmux for detachable SSH sessions
- Interact with Google Cloud Storage
- End to End example :
  - Write code in your codespace
  - Transfer it to your GCE instance
  - Run the code, it produces a model
  - Transfer the model weights to google cloud storage
  - Get it back from your codespace
- Bonus content
  - Managed SQL
  - Infrastructure as code

SSH TUNNEL, PORT FORWARDING

*In computer networking, **a port is a communication endpoint**. At the software level, within an operating system, a port is a logical construct that identifies a specific process or a type of network service. **A port is identified for each transport protocol and address combination** by a 16-bit unsigned number, known as the port number. The most common transport protocols that use port numbers are the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).*

Wikipedia

EXAMPLES OF PROTOCOLS & USUAL PORTS

Examples

- SSH on port 22
- HTTP on port 80
- HTTPS on port 443

http apps can serve content over specific ports

Example

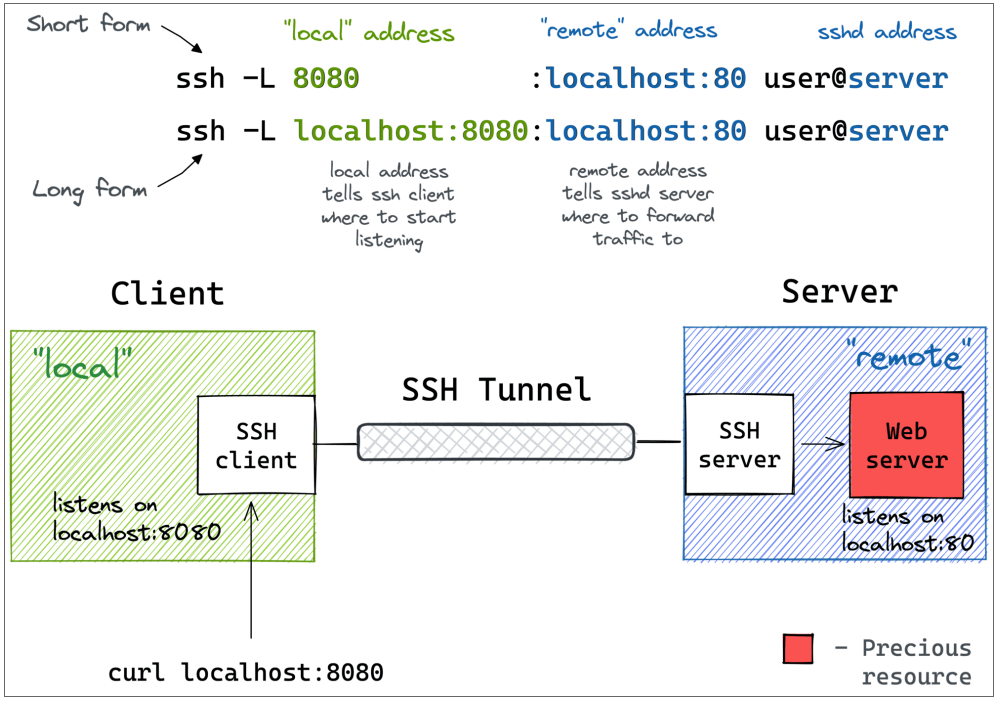
- Jupyter default is 8888 (that's why you open **http://localhost:8888**)

SSH TUNNELS

We usually connect to web app using `http://{ip}:{port}`

😨 but what if the machine is not available from the public internet / local network ?

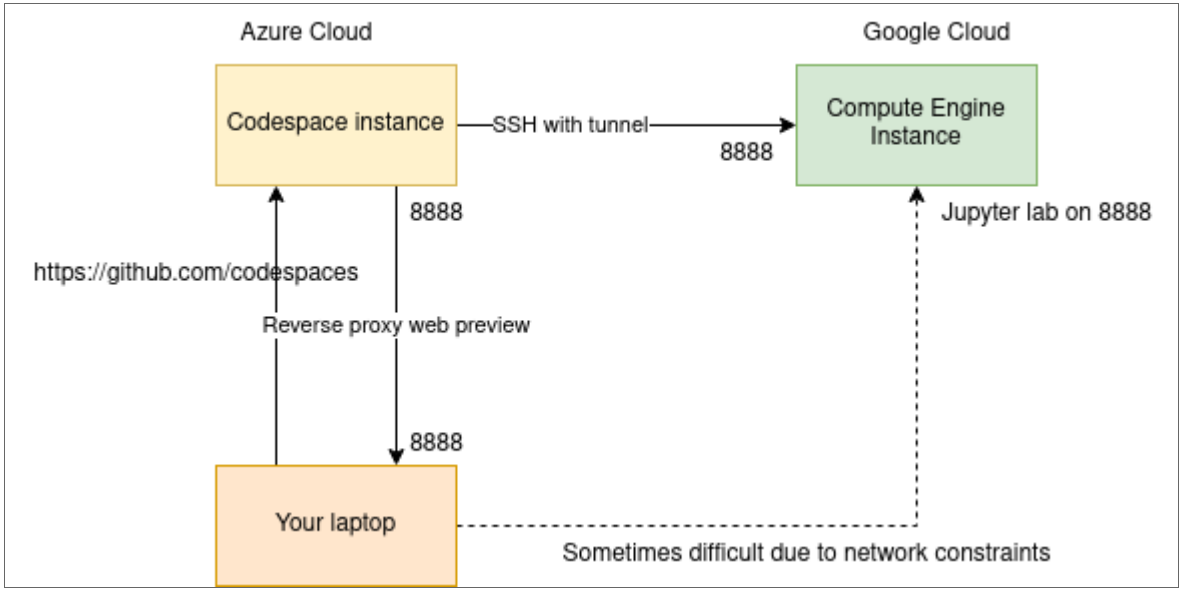
➡️ Enter SSH with port forwarding



Visual guide

TUNNELS OF TUNNELS

- Some of you did Local Machine -> (browser) -> Codespace -> (ssh) -> VM -> Jupyterlab on port 8888
- With port transfers !
- What happens when you go to `http://(url-generated-by-codespaces):8888` in this case ?



Speaker notes