



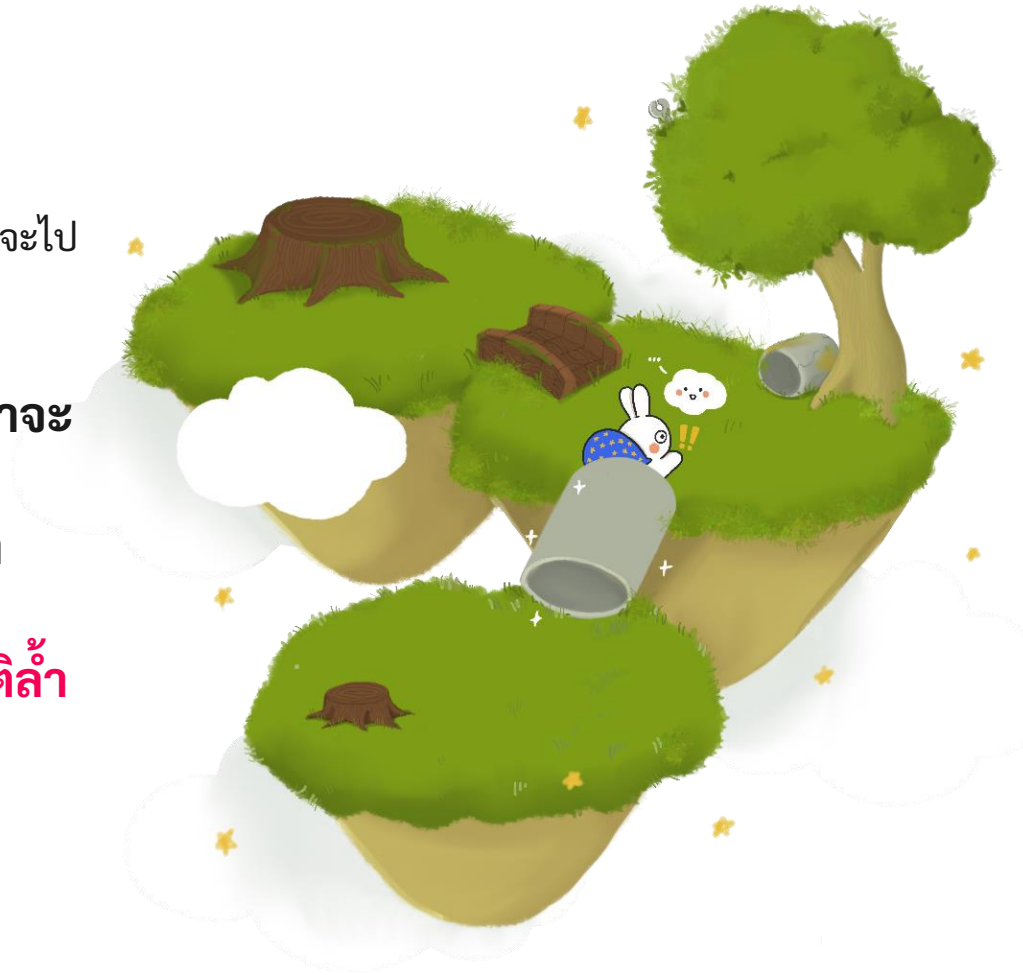
Chapter 3: Cloud Computing

ปลดล็อกทุกขีดจำกัด ด้วยการประมวลผลบน Cloud

ความเดิมตอนที่แล้ว: ไรต์ดี กระต่ายน้อยที่มีความฝันอยากจะไปดาวดีอี ทำความสะอาดท้องจนเสร็จ

ไรต์ดี มุดไปอีกฝั่งของท่อด้วยความตื่นเต้นว่าจะเจอกับทางไปดาวดีอี เธอได้พบกับเมฆน้อย คลาวดี ที่บอกกับเธอด้วยน้ำเสียงหวานใส่ว่า

“สวัสดีกระต่ายน้อย ในท่อใต้ต้นไม้นี้มีสมบัติล้ำค่าอยู่ เธอมาช่วยซ่อมท่อให้หน่อยสิ งานง่าย ๆ แค่นี้ทำแป๊บเดียวก็เสร็จเนอะ ไม่ต้องคิดเงินหรอกคนกันเอง”





Intro to Cloud Computing



DATATH

On-premise คืออะไร



“On-premises software” (หรือ on-prem) หมายถึงซอฟต์แวร์ที่ทำงานบน physical server ที่ตั้งอยู่ภายในบริษัท โดยที่ไม่ได้เชื่อมต่อกับ cloud หรือ server ภายนอก

พัฒนาการ การใช้เซิร์ฟเวอร์ในองค์กร



Virtual Machine (VM) /
Virtual Private Server
(VPS)

VPS คือ บริการเช่าส่วนหนึ่ง
ของ Dedicated Server



Dedicated Server

การซื้อเซิร์ฟเวอร์ของเป็น
ขององค์กร



Data Center

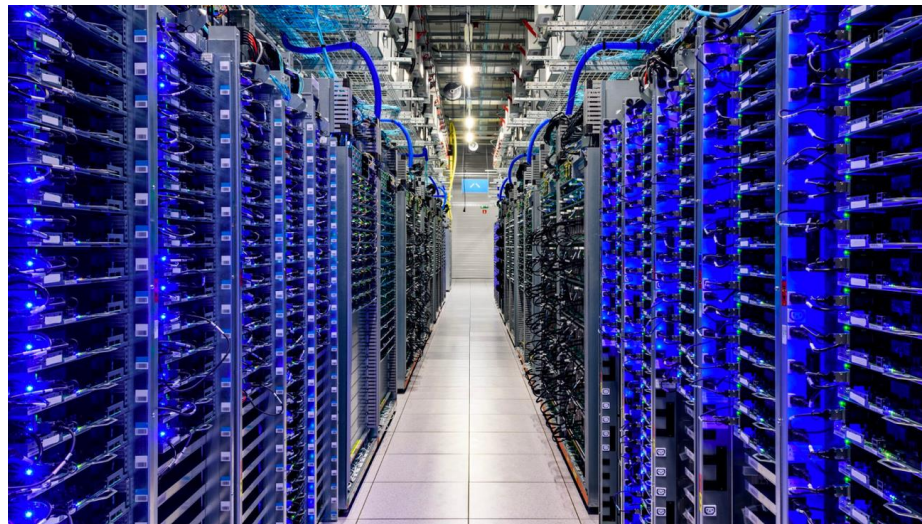
การสร้าง Data Center เพื่อ
จัดการเซิร์ฟเวอร์จำนวนมาก



Data Center ของ Google



บรรยากาศด้านนอก Data Center



บรรยากาศด้านใน Data Center

<https://www.google.com/about/datacenters/gallery/>

Cloud Computing คือ อะไร



บริการเข้าใช้ส่วนหนึ่งของ Data Center ผ่านระบบอินเทอร์เน็ต โดยมีการให้บริการในหลายรูปแบบ เช่น

- VM (Virtual Machine)
- Data Lake / Storage ที่เก็บข้อมูล
- Database ฐานข้อมูล
- Data Warehouse
- ฯลฯ



On-prem vs Cloud Computing (1)



	On-Premise	Cloud Computing
ค่าใช้จ่าย	มีค่าใช้จ่ายในการจัดซื้อติดตั้ง Server อาจจะต้องทำ data center หรือ server room ขึ้นมา (เราเรียกว่า CapEX - Capital Expenditure หรือเงินลงทุน)	ค่าใช้จ่ายเป็นลักษณะการบริการรายเดือน จ่ายเท่าที่ใช้ (pay as you go) เราเรียกว่า OpEx หรือ Operational Expenditure ค่าใช้จ่ายในการดำเนินงาน
สถานที่ตั้ง	สามารถติดตั้งเซิร์ฟเวอร์บนพื้นที่ขององค์กรได้	สามารถเลือกที่ตั้งเซิร์ฟเวอร์ของเราได้จาก Data Center ของผู้ให้บริการ Cloud มีหลายที่ทั่วโลก
ความปลอดภัยด้าน Physical	องค์กรจะต้องลงทุนด้านความปลอดภัยเอง เช่น จ้างรปภ. หรือสร้างระบบรักษาความปลอดภัยในตึก	ผู้ให้บริการ Cloud จะจัดการด้านความปลอดภัยให้ทั้งหมด ซึ่งได้รับมาตรฐานความปลอดภัยสำหรับ Data Center





On-prem vs Cloud Computing (2)



	On-Premise	Cloud Computing
ประสิทธิภาพ	ประสิทธิภาพจะขึ้นอยู่กับจำนวนและสเปคของเครื่องใน data center ที่สร้างขึ้น	สามารถเลือกสเปคเครื่อง และตำแหน่ง data center ได้หลายรูปแบบ สามารถสร้าง cluster ขนาดใหญ่ได้อย่างรวดเร็ว หรือเพิ่ม GPU ได้อีกด้วย
การจัดการข้อผิดพลาด	ต้องมีแผนจัดการรับมือ เมื่อเกิดข้อผิดพลาดขึ้นใน data center เช่น การมี site สำรอง	สามารถสำรองข้อมูลไปที่ Data Center อื่นทั่วโลก (Replication) และมี SLA (Service Level Agreement) เป็นการรับประกันว่าระบบจะไม่ล่มบ่อย
ซอฟต์แวร์ที่ให้บริการ	มี Service ตามที่เราติดตั้งไว้ หากต้องการซอฟต์แวร์ใหม่ จะต้องมีการทีมคอยดูแล	มี Service ต่าง ๆ ให้เลือกใช่มากมาย และอัปเดตอัตโนมัติอย่างสม่ำเสมอ



ประเภทของ Cloud Computing

1. Public Cloud
2. Private Cloud
3. Hybrid Cloud



1. Public Cloud: Cloud Platform providers



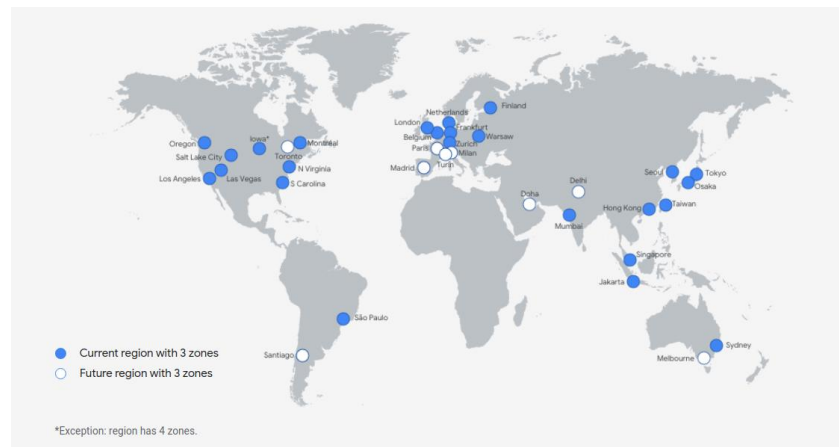
และอื่น ๆ เช่น IBM Cloud, Alibaba Cloud, Huawei Cloud, Tencent Cloud



ข้อดีของ Public Cloud Platform

- มีบริการหลากหลายรูปแบบ
เช่น VM, Kubernetes, Serverless, Database, Data Warehouse, Load balancer ฯลฯ
- มี Data Center ในหลายประเทศทั่วโลก
- สามารถขยายตามต้องการได้ (Scalability)
- สามารถปรับแต่งสเปคตามต้องการได้ (OS, RAM, CPU, GPU, Disk)
- มี SLA (Service Level Agreement) ที่รับประกันว่าระบบจะไม่ล่มบ่อย ได้รับความไว้วางใจจากองค์กรใหญ่ ๆ มากมาย

ที่ตั้ง Server ของ Google Cloud:



DATATH

<https://cloud.google.com/about/locations>

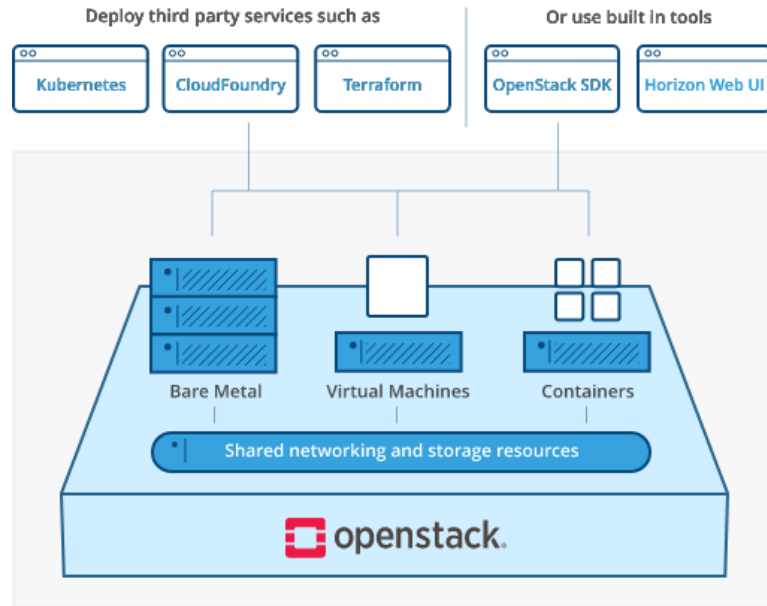
2. Private Cloud: สร้าง cloud platform ส่วนตัว



การสร้าง cloud ส่วนตัว ภายใน data center ขององค์กร

โดยจะต้องจัดซื้อ และ maintenance ระบบ รวมถึงจัดการด้าน network เอง

อาจมีค่าใช้จ่ายที่สูงในการซื้อ Hardware และการสร้าง data center ให้ได้ตามมาตรฐาน แต่สามารถมั่นใจได้ว่า ข้อมูลจะอยู่ภายใน data center และ network ขององค์กรเท่านั้น



OPENSHIFT



openstack.



DATATH

3. Hybrid Cloud: On-premise + Cloud



Hybrid Cloud เป็นการรวมระบบ Cloud (Public & Private)
กับ ระบบ On-Premise เข้าด้วยกัน

2 Use Cases หลัก ของ Hybrid Cloud:

1. เราต้องการจัดการเซิร์ฟเวอร์บน On-Premise ผ่านระบบ Cloud เช่น Google Cloud Anthos ที่สามารถควบคุมระบบเซิร์ฟเวอร์ที่ติดตั้ง Kubernetes ได้
2. เราต้องการนำระบบ Cloud มาใช้บน On-Premise เช่น การวางเครื่อง Amazon Outpost ของ AWS เข้าไปใน Data Center ของบริษัท



Google Cloud Anthos



Amazon Outpost

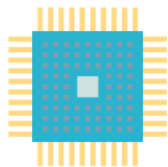




Cloud Computing Concepts



บริการหลัก 2 แบบ: Compute & Storage/Database



Compute คือ หน่วยประมวลผล
เช่น

- CPU
- RAM
- GPU



Storage คือ ที่เก็บข้อมูล
เช่น

- Persistent disk (Disk ใน VM)
- Blob storage (Data Lake)
- Database
- Data warehouse

การเพิ่มประสิทธิภาพ Compute: Vertical Scaling vs Horizontal Scaling

Vertical Scaling

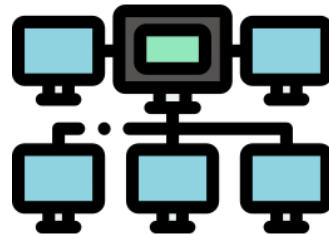
เปลี่ยนไปใช้เซิร์ฟเวอร์ที่สเปคดีขึ้น



เหมาะกับระบบที่ต้องทำงานบนเซิร์ฟเวอร์
เครื่องเดียว เช่น ระบบ Database ส่วนใหญ่

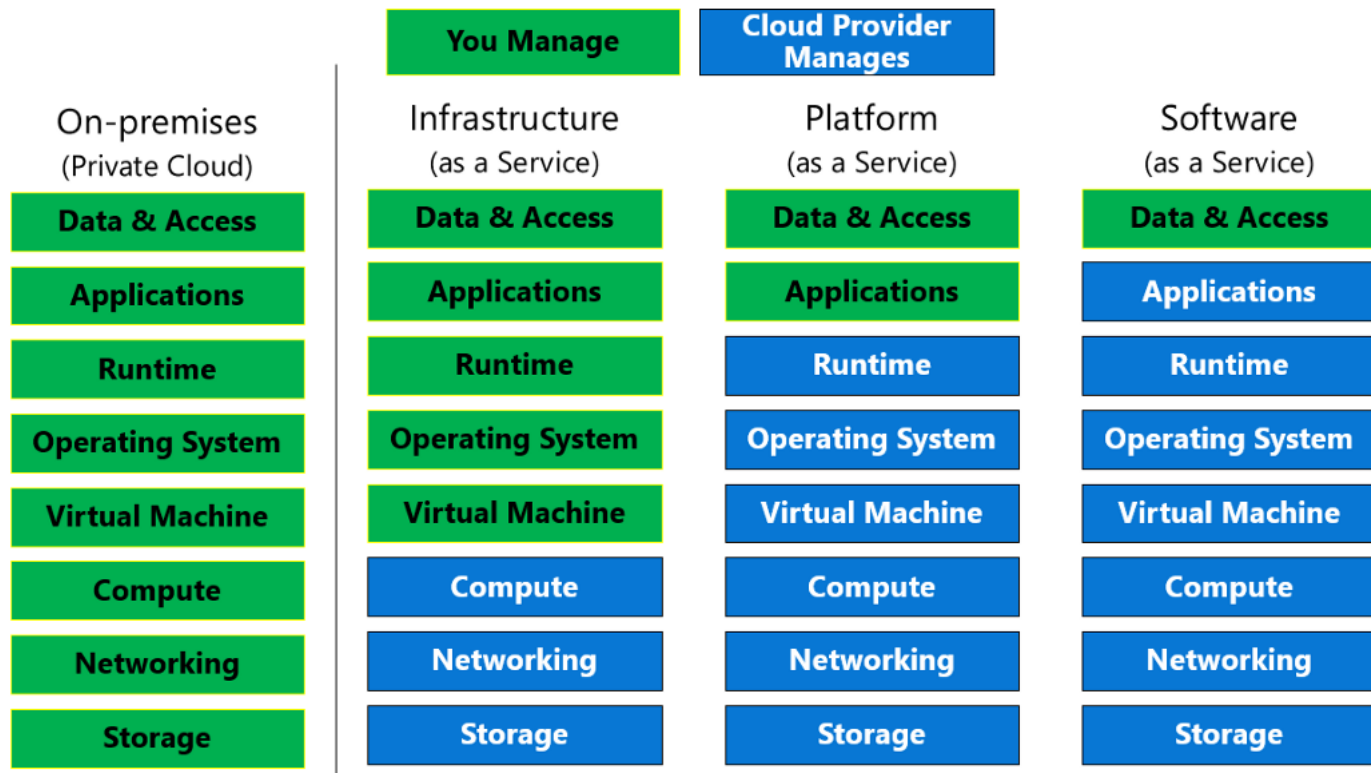
Horizontal Scaling

เพิ่มจำนวนเครื่องเซิร์ฟเวอร์



เหมาะกับงานที่ประมวลผลเป็น
cluster ได้ (Distributed Computing)
เช่น Hadoop, Spark

กลุ่มบริการประเภทต่าง ๆ บน Cloud (as a service)



บริการประเภทต่าง ๆ บน Cloud (as a service)

- **Infrastructure as a Service**

การสร้าง (provision) infrastructure เช่น Virtual Machine (VM), Virtual Networking, Google Cloud Storage

- **Platform as a Service**

การให้บริการ platform ที่สามารถนำไปพัฒนาแอปพลิเคชันต่อได้ทันที เช่น Google AppEngine, Cloud SQL

- **Software as a Service**

การให้บริการ software แบบใช้ได้ทันที เช่น Google Drive, Gmail, Google Colab



2 ประเภทหลัก ๆ ของ Platform as a Service: Managed Service & Serverless



Managed Service

Cloud Computing จัดการเรื่องการติดตั้งให้ และดูแลระบบให้ส่วนหนึ่ง ผู้ใช้ยังสามารถเข้าไปจัดการเองได้ถ้าต้องการ

เช่น Google Cloud Dataproc = Managed Hadoop
 Google Cloud Composer = Managed Airflow
 Google Kubernetes Engine = Managed Kubernetes



Serverless

Cloud Computing จัดการระบบให้ทั้งหมด ผู้ใช้ไม่สามารถเข้าไปจัดการการทำงานของระบบภายในบริการนั้น

เช่น Google Cloud Function = Serverless Function
 Google Cloud Run = Serverless Container
 Google BigQuery = Serverless Data Warehouse



DATA TH

ปัญหาของการใช้ Cloud: Vendor lock-in



Vendor Lock-in = การติดอยู่ใน ecosystem

การที่เลือกใช้ service ของ cloud provider ของ platform หนึ่ง แล้ว service นั้นมีฟีเจอร์ที่เฉพาะเจาะจงที่เจ้าอื่นไม่มี ทำให้ไม่สามารถย้ายไปเจ้าอื่นได้ หรือทำให้ไม่สามารถย้ายระบบกลับมา on-premise ได้

กลายเป็นว่าเราโดน Lock in อยู่กับ Cloud เจ้านั้น

Cloud Agnostic คือ ใช้ได้กับหลาย cloud



Cloud Agnostic = บริการที่รองรับ cloud หลายเจ้า

Cloud Agnostic คือ สามารถเลือกใช้บริการนี้กับ cloud ยี่ห้ออื่น ๆ ได้ ไม่เฉพาะเจาะจงผู้ให้บริการ Cloud

เช่น Snowflake, Databricks





มาใช้ GCP กันเถอะ!

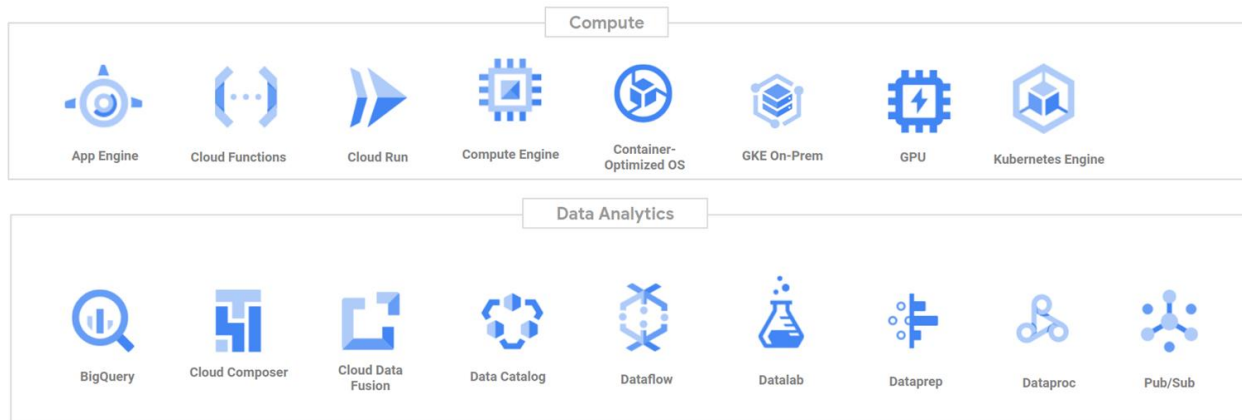


Google Cloud มีอะไรให้ใช้บ้าง

Google Cloud มีมากกว่า 100+
บริการ ในหลากหลายหมวดหมู่

- Compute
- Storage & DBs
- Big Data
- ML & AI products
- Identity & Security
- Networking
- Management tools
- Developer tools
- IoT
- API Platform

ตัวอย่างบริการบน Google Cloud



Technology Stack overview: Cloud

Data Lake



Amazon S3 Google Cloud Storage Azure Blob Storage

Data Warehouse



Google BigQuery Azure Synapse Apache Hive



Amazon Redshift Snowflake

Data Processing



Apache Spark



Apache Beam



Amazon Glue



Cloud Dataflow



Cloud Dataproc

Data Pipeline



Azure Data Factory



Apache Airflow



Apache Oozie



Luigi

บริการ Compute ประเภทต่าง ๆ ของ Google Cloud

IAAS : Infrastructure
as a service

CAAS : Container
as a service

PAAS : Platform
as a service

FAAS : Function
as a service



**Google
Compute
Engine**

Virtual Machine



**Google
Kubernetes
Engine (GKE)**

Managed Kubernetes
platform



**Google
Cloud Run**

Serverless container
cluster



**Google
App Engine**

Serverless application



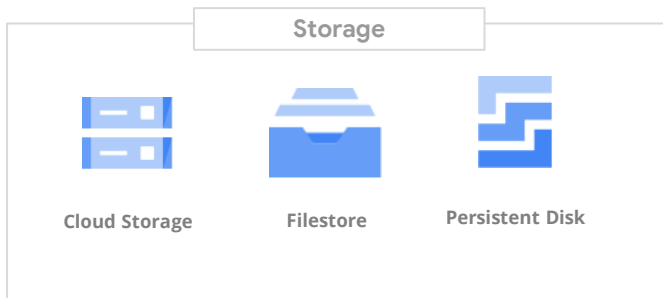
**Google
Cloud
Function**

Serverless function

ที่เก็บข้อมูลบน Google Cloud: Storage & Database

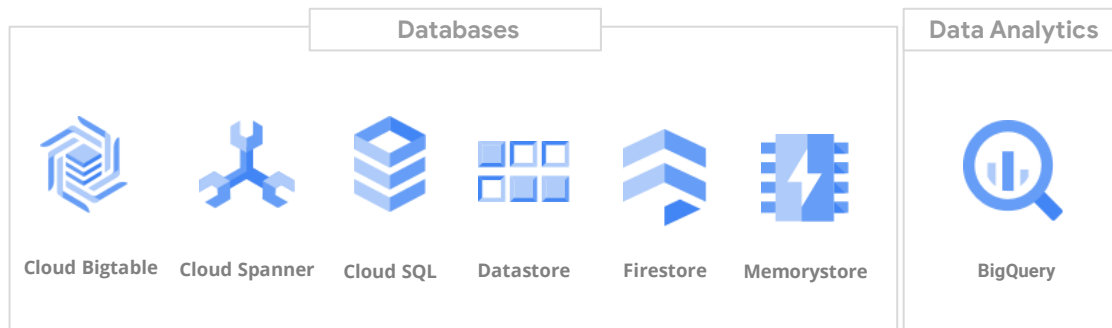
Storage (Data Lake)

- ลักษณะเก็บเป็นไฟล์



Database / Data Warehouse

- เก็บเป็นแถวข้อมูล
- มี Database ให้เลือกหลากหลายประเภท ทั้ง SQL, Document-store, Key-value, Wide-column store ฯลฯ



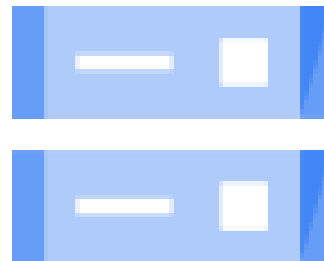
Google Cloud Storage บริการเก็บไฟล์ราคาไม่แพง

GCS เป็นบริการของ Google Cloud ที่ทำให้องค์กรสามารถเก็บไฟล์แบบ Blob (Binary Large Object) ขนาดเท่าไรก็ได้ และดึงไปใช้ตอนไหนก็ได้ เหมือนเป็น Harddrive ส่วนตัวขององค์กร

จุดเด่น:

- โอกาสไฟล์หายน้อยมาก สามารถเพิ่มความทนทาน (Durability) ได้ถึง 99.999999999%
- ค่าใช้จ่ายไม่แพง

จุดด้อย: ทำได้เฉพาะการอัปโหลด - ดาวน์โหลดแบบพื้นฐาน ไม่มีฟีเจอร์หุ้รหุ้รแบบ Google Drive หรือ Dropbox



Google
Cloud Storage

บริการบน Google Cloud สำหรับทำ Data Processing & Data Pipeline

Google Cloud
Service



Cloud
Dataproc



Cloud
Dataflow

Framework
เบื้องหลัง



Data Pipeline



Cloud
Composer



DATATH

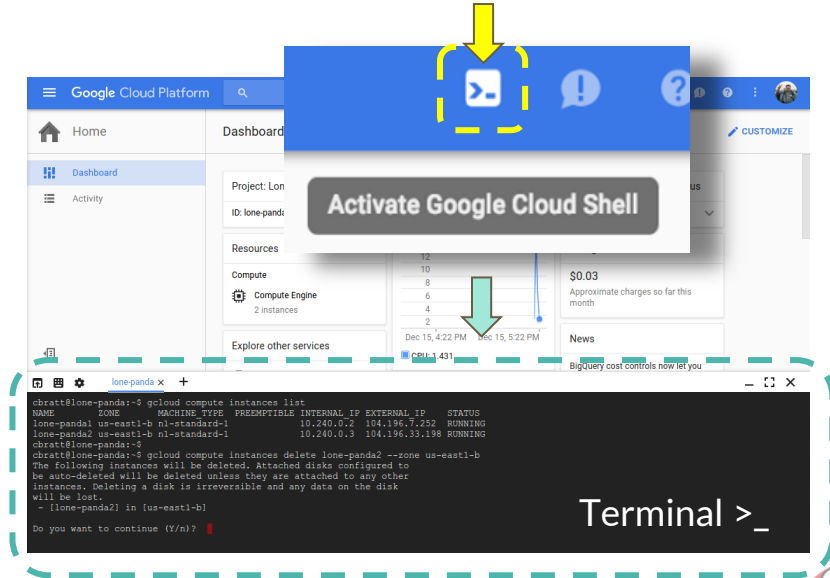
Bonus: Cloud Shell

Cloud Shell เป็นเครื่องมือฟรี ที่ Google Cloud ช่วยให้ใช้งาน command-line interface (CLI) ผ่าน web browser ได้สะดวกยิ่งขึ้น

โดย Cloud Shell จะเป็นเครื่อง VM ชั่วคราว ที่สามารถใช้งาน command line ได้ผ่าน terminal พร้อมด้วย tool พื้นฐานจำเป็น เช่น Python, MySQL, Docker, Git

พร้อม Cloud Shell Editor ที่ทำให้สามารถแก้ไขไฟล์ออนไลน์ได้ คล้ายคลึงกับโปรแกรม VS Code

Cloud Shell



The screenshot shows the Google Cloud Platform dashboard. A blue overlay with a white terminal icon and the text "Activate Google Cloud Shell" is prominently displayed. Below this, a terminal window is open, showing the command prompt "Terminal >_" and the output of the command "gcloud compute instances list". The output shows a table of instances:

NAME	ZONE	MACHINE TYPE	PREEMPTIBLE	INTERNAL IP	EXTERNAL IP	STATUS
lone-panda1	us-east1-b	n1-standard-1		10.240.0.2	104.196.7.252	RUNNING
lone-panda2	us-east1-b	n1-standard-1		10.240.0.3	104.196.33.198	RUNNING

The terminal also shows the command "gcloud compute instances delete lone-panda2 --zone us-east1-b" and the confirmation prompt "Do you want to continue (Y/n)?".

เราสามารถ upload และ download ไฟล์ไปยัง Cloud Shell ได้ โดยมีพื้นที่ให้ใช้ได้ 5GB



DATATH



Google Cloud

DEVELOPER'S CHEAT SHEET

v2021.3.3

Created by the Google Developer Relations Team
Maintained at <https://4words.dev>

Feedback? [@gregsramblings](#) [@pvergadia](#)

COMPUTE

Cloud Functions	Event-driven serverless functions
App Engine	Managed app platform
Cloud Run	Serverless for containerized applications
Kubernetes Engine (GKE)	Managed Kubernetes/containers
Compute Engine	VMs, GPUs, TPUs, Disks
Bare Metal Solution	Hardware for specialized workloads
Preemptible VMs	Short-lived compute instances
Shielded VMs	Hardened VMs
Sole-tenant Nodes	Dedicated physical servers
VMware Engine	VMware on Compute Engine

Cloud Filestore	Managed NFS server
Cloud Storage	Multi-class multi-region object storage
Persistent Disk	Block storage for VMs
Local SSD	VM locally attached SSDs

Cloud Bigtable	Petabyte-scale, low-latency, non-relational
Cloud Firestore	Serverless NoSQL document DB
Cloud Memorystore	Managed Redis and Memcached
Cloud Spanner	Horizontally scalable relational DB
Cloud SQL	Managed MySQL/PostgreSQL/SQL Server
Database Migration Service	Migrate to Cloud SQL
DB Insights	SQL Inspector

DATA AND ANALYTICS

BigQuery	Data warehouse/analytics
BigQuery BI Engine	In-memory analytics engine
BigQuery ML	BigQuery model training/serving
Cloud Composer	Managed workflow orchestration service
Cloud Data Fusion	Graphically manage data pipelines
Cloud Dataflow	Stream/batch data processing
Cloud Dataprep	Visual data wrangling
Cloud Datastore	Managed Spark and Hadoop
Cloud Pub/Sub	Global real-time messaging
Data Catalog	Metadata management service
Data Studio	Collaborative data exploration/dashboarding

HYBRID AND MULTI-CLOUD

Anthos	Enterprise hybrid/multi-cloud platform
Anthos Clusters	Hybrid/on-prem Kubernetes Engine
Anthos Config Management	Policy and security automation
Anthos Service Mesh	Managed service mesh (Istio)
Cloud Run for Anthos	Serverless development for Anthos
GCP Marketplace for Anthos	Pre-configured containerized apps
Migrate for Anthos	Migrate VMs to Kubernetes Engine
Operations	Monitoring, logging, troubleshooting
Cloud Build	Continuous integration/delivery platform
Traffic Director	Service mesh traffic management
Apigee API Management	API management, development, security

[illegible]

Carrier Peering	Peer through a carrier
Direct Peering	Peer with GGP
Dedicated Interconnect	Dedicated private network connection
Partner Interconnect	Connect on-prem network to VPC
Cloud Armor	DDoS protection and WAF
Cloud DNS	Cloud delivery network
Cloud DNS	Programmable DNS serving
Cloud Load Balancing	Multi-region load distribution/balancing
Cloud NAT	Network address translation service
Cloud Router	VPC/on-prem network route exchange (BGP)
VPC VPN (HA)	VPN (Virtual private network connection)
Network Service Tiers	Use vs performance (pricing)
VPN Gateway	Network telemetry service
Traffic Director	Service mesh traffic management
Google Cloud Service Mesh	Service-aware network management
Virtual Private Cloud	Software defined networking
VPN	Security perimeter for API-based services
Network Intelligence Center	Network monitoring and topology

GAMING

Google Cloud Game Servers	Orchestrate Agones clusters
---------------------------	-----------------------------

INTERNET OF THINGS (IOT)

Cloud IoT Core	Max
----------------	-----

IDENTITY AND SECURITY

Access Workspaces	Audit: doc provider access
Assured Transpans	Workload compliance controls
Binary Authorization	Kubernetes deployment time security
Cloud Access Management Service	Managed private OSs
Cloud Asset Inventory	All assets, one place
Cloud Audit Logs	Audit trails for GCP
Cloud Build	Cloudify and reduce sensitive data
Cloud IAM	Hardware security module service
Cloud DNS	External keys you control
Cloud IAM	Resource access controls
Cloud Key Management Service	Manage secrets, devices & apps
Cloud Identity Identity Aware Proxy	Identity-based app access
Cloud KMS	Hosted key management service
Cloud Resource Manager	Cloud project metadata management
Cloud Security Command Center	Security management & risk platform
Cloud Security Scanner	App security management
Configuring Computing	Encrypt data at rest
Cloud Storage	End-user attribute-based access control
Event-Driven Detection	Scrap for suspicious activity
Managed Service for Microsoft Exchange	Managed Microsoft Active Directory
Secret Manager	Store and manage secrets

IDENTITY AND SECURITY (CONT.)

Security Key Enforcement	Two-step key verification
Shielded VMs	Hardened VMs
Titan Security Key	Two-factor authentication (2FA) device
VPC Service Controls	VPC data constraints

Cloud APIs	APIs for cloud services
Cloud Billing API	Programmatically manage GCP billing
Cloud Billing	Billing and cost management tools
Cloud Console	Web-based management console
Cloud Deployment Manager	Templated infrastructure deployment
Cloud Mobile App	iOS/Android GCP manager app
Private Catalog	Internal Services Catalog
Cloud Debugger	Live production debugging
Error Reporting	Error reporting
Cloud Logging	Centralized logging
Cloud Monitoring	Infrastructure and application monitoring
Cloud Profiler	CPU and heap profiling
Cloud Trace	App performance insights
Transparent SLIs	Monitor GCP services

DEVELOPER TOOLS

Cloud Build	Continuous integration/delivery platform
Cloud Code for IntelliJ	IntelliJ GCP tools
Cloud Code for VS Code	VS Code GCP tools
Cloud Code for Eclipse	Cloud native IDE extensions
Cloud Tools for Eclipse	Eclipse GCP tools
Cloud Tools for Visual Studio	Visual Studio GCP tools
Grade App Engine plugin	Grade App Engine plugin
Maven App Engine plugin	Maven App Engine plugin
Cloud SDK	CLI for GCP
Cloud Shell	Browse-based terminal/CLI
Artifact Registry	Universal package manager
Cloud Source Repositories	Hosted private git repos
Cloud Storage	Private container registry/storage
Container Analysis	Automated security scanning
Eventarc	Event-driven Cloud Run services
Cloud Scheduler	Managed cron job service
Cloud Tasks	Asynchronous task execution
Cloud Workflows	HTTIP services orchestration

BigQuery Data Transfer Service	Bulk import analytics data
Cloud Data Transfer	Data migration tools/CLI
Google Transfer Appliance	Rentable data transport box
Storage Transfer Service	Online/on-premises data transfer
Migrate to Anthos	Migrate VMs to GKE containers
Migrate for Compute Engine	Compute Engine migration tools
Migrate from Amazon Redshift	Migrate from Redshift to BigQuery
Migrate from Teradata	Migrate from Teradata to BigQuery
VM Migration	VM migration tools
Cloud Foundation Toolkit	Infrastructure as Code templates
KF	Cloud Foundry to Kubernetes

API PLATFORM AND ECOSYSTEMS

API Analytics	API metrics
API Monetization	Monetize APIs
Apigee API Platform	Develop, secure, monitor APIs
API Gateway	Fully managed API Gateway
Apigee Hybrid	Managed hybrid/multi-cloud API environments
Apigee Sense	API protection from attacks
Cloud Endpoints	Cloud API gateway
Cloud Healthcare API	Healthcare system GCP interoperability
Developer Portal	API management portal
GCP Marketplace	Partner & open source marketplace
AppSheet	No-code App creation

GOOGLE MAPS PLATFORM

Directions API	Get directions between locations
Distance Matrix API	Multi-origin/destination travel times
Geocoding API	Convert address to/from coordinates
Geolocation API	Derive location without GPS
Maps Embed API	Display frame embedded maps
Maps JavaScript API	Dynamic web maps
Maps SDK for Android	Maps for Android apps
Maps SDK for iOS	Maps for iOS apps
Maps Static API	Display static map images

GOOGLE MAPS PLATFORM (CONT.)

Maps SDK for Unity	Unity SDK for games
Maps URLs	URL scheme for maps
Places API	Rest-based Places features
Places Library, Maps JS API	Places features for web
Places SDK for Android	Places features for Android
Places SDK for iOS	Places feature for iOS
Roads API	Convert coordinates to roads
Street View Static API	Static street view images
Street View Service	Street view for JavaScript
Time Zone API	Convert coordinates to timezone

GOOGLE WORKSPACE

Admin SDK	Manage Workspace resources
AMP for Email	Dynamic interactive email
Apps Script	Extend and automate everything
Calendar API	Create and manage calendars
Custom Search API	Provision and manage calendars
Cloud Search	Unified search for enterprise
Docs API	Create and edit documents
Drive Activity API	Retrieve Google Drive activity
Drive API	Read and write files
Drive Picker	Drive file selection wizard
Email Markup	Interactive email using scheming
Workspace Add-ons	Extend G Suite apps
Workspace Marketplace	Storefront for integrated applications
Email	Enhance Gmail
Hangouts Chat Bots	Conversational bots in chat
People API	Manage user's Contacts
Sheets API	Read and write spreadsheets
Slides API	Create and edit presentations
Tasks API	Search, read & update Tasks
Vault API	Manage your organization's documents

Cloud Firestore	Document store and sync
Cloud Functions for Firebase	Event-driven serverless applications
Cloud Storage for Firebase	Object storage and serving
Craigslist	Crawl reporting and analytics
Crashlytics	Crash & ASB test experiments
Database In-Memory Testing	Trusted test early access
Firebase App Distribution	Drop-in authentication
Firebase Authentication	Send device notifications
Firebase Cloud Messaging	Link to app content
Firebase Dynamic Links	Pre-packaged development solutions
Firebase Extensions	Web hosting with CDN/SSL
Firebase Hosting	Send in-app contextual messages
Firebase In-App Messaging	App/web performance monitoring
Firebase Performance Monitoring	Predict user targeting
Firebase Predictions	Real-time data synchronization
Firebase Realtime Database	Remotely configure installed apps
Firebase Remote Config	Mobile testing device farm
Firebase Test Lab	Mobile test app analytics
Google Analytics for Firebase	ML APIs for mobile
ML Kit for Firebase	

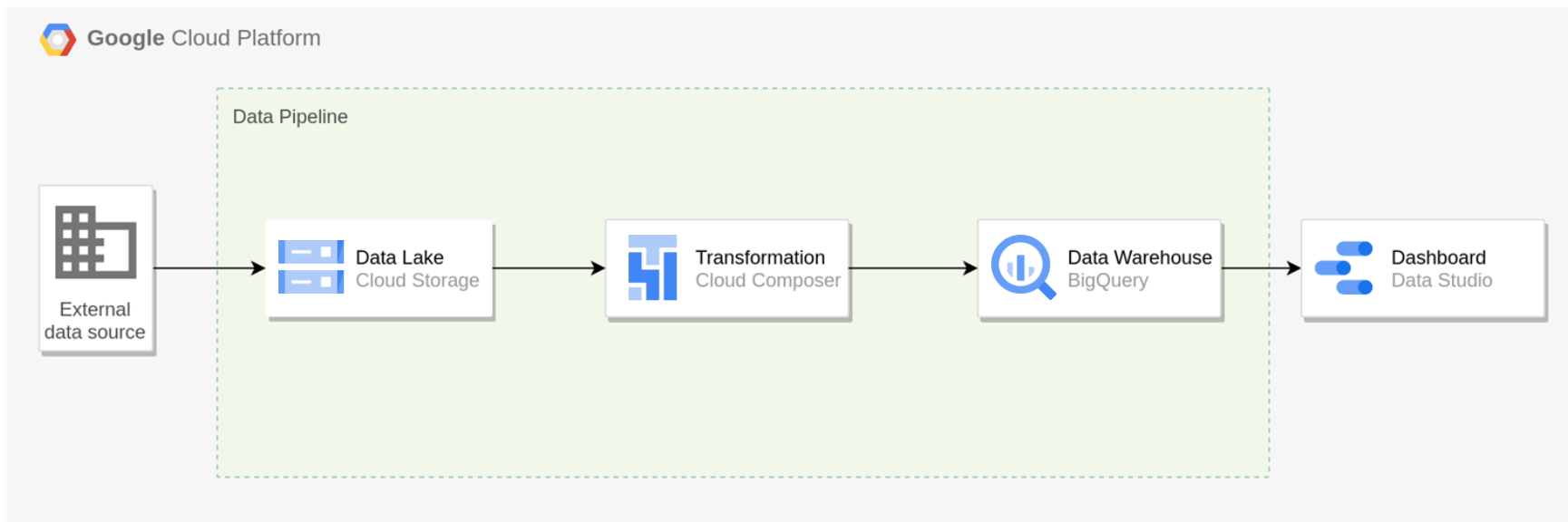
Google Cloud Home Page	cloud.google.com
Google Cloud Blog	cloud.google.com/blog
Google Cloud Platform Podcast	cloud.google.com/blog/podcasts
Kubernetes Podcast from Google	kubernetespodcast.com
Google Cloud Open Source	opensource.google.com/projects/first/
Q&A Medium Publication	medium.com/google-cloud
Google Cloud about:blog	google.com/about:blog
FreePress Blog	freepress.googleblog.com
Workshop Developers Blog	gwst-dev-developers.googleblog.com
Google Cloud Security Blog	cloudsecurity.googleblog.com
Workshop Twitter	twitter.com/gcloudsws
Google Cloud Certifications	cloud.google.com/certification
Google Cloud System Status	cloud.google.com/status
Google Cloud Training	cloud.google.com/training
Google Developers Blog	developers.googleblog.com
Google Maps Platform Blog	mapsplatform.googleblog.com
Google Cloud Source Blog	opensource.googleblog.com
Google Security Blog	www.google.com/securityblog
Kaggle Home Page	www.kaggle.com
Google Kubernetes Blog	kubernetes.io/blog
Regions and Network Map	cloud.google.com/about/locations
DORA - Software & Delivery Statistics	cloud.google.com/devops



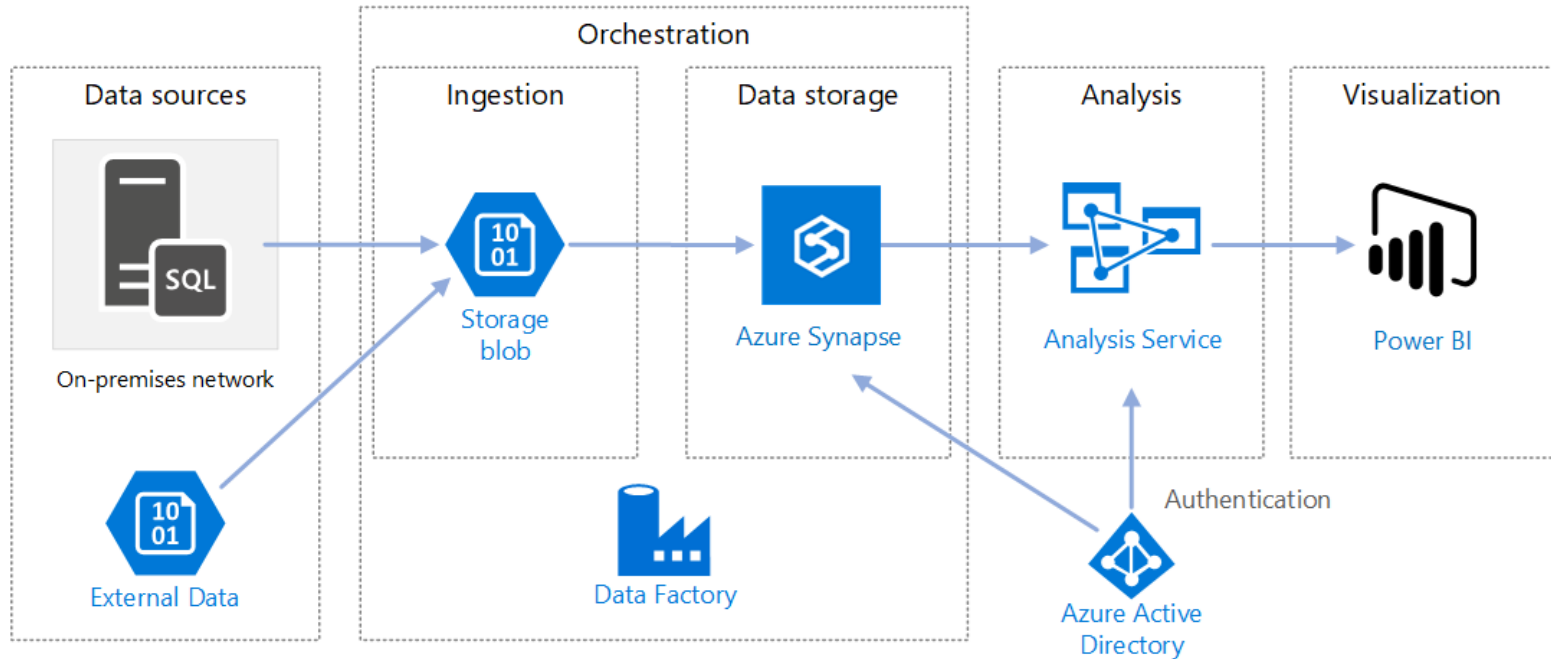
DATA TH

Reference Architecture (GCP)

Architecture: Batch Ingest, Transform, Load



Reference Architecture (Azure)



ใช้ Cloud เจ้าไหนดีที่สุด?



Google Cloud



<https://cloud.google.com/docs/compare/aws>



<https://cloud.google.com/docs/compare/azure>



DATATH



Intro to Bash [Linux command line]



เรียนรู้ Linux Command line เบื้องต้น ที่จำเป็น

Linux ต่างกับ Unix ยังไง

UNIX[®]
An Open Group Standard

หรือ *nix เป็น ระบบปฏิบัติการที่มีรากฐานมาจาก Unix



Bash คือ อะไร



Bash มาจากคำว่า Bourne-again shell ซึ่งเป็น Unix shell ประเภทหนึ่ง ที่นิยมใช้กันทั่วไป สำหรับ Unix และ Linux

Bash เป็นภาษา command language ที่ใช้ในการสั่งการระบบปฏิบัติการ ผ่านโปรแกรม terminal ซึ่งภาษาในตระกูลนี้จะเรียกรวม ๆ ได้ว่า **Shell** ซึ่งมีลักษณะเป็น command line Shell อื่น ๆ ที่นิยมใช้ นอกจาก Bash เช่น Zsh (ที่มากับ MacOS ในปัจจุบัน)



การใช้ Bash command line



- Bash command line จะใช้งานผ่านโปรแกรม Terminal ที่มากับเครื่อง สำหรับ Linux หรือ Mac อยู่แล้ว
- ถ้าใน Windows สามารถใช้ WSL (Windows Subsystem for Linux) หรือ Git Bash เพื่อใช้งาน Terminal ได้
- ถ้าใช้งานผ่านเว็บ Google Cloud Console จะสามารถใช้งาน Terminal ผ่าน Cloud Shell (>_) ได้
- สัญลักษณ์ประจำ Bash คือ \$_ เพราะว่า ทุกคำสั่ง จะมี \$ ขึ้นต้นมาให้เสมอ แทนว่าบรรทัดนั้นคือ command ตัวอย่าง

```
$ ls  
README-cloudshell.txt
```



\$ command

Output

พิมพ์ command หลัง \$

output ของ command นั้น

การใช้งานแบบ command-line มีอีกชื่อหนึ่งเรียกว่า command-line interface หรือ CLI นั่นเอง

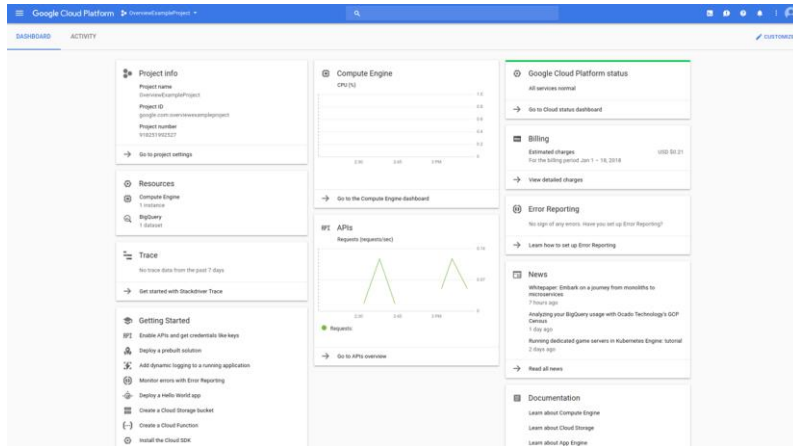


DATATH

มาเริ่มต้นใช้งาน Cloud Shell Terminal กัน

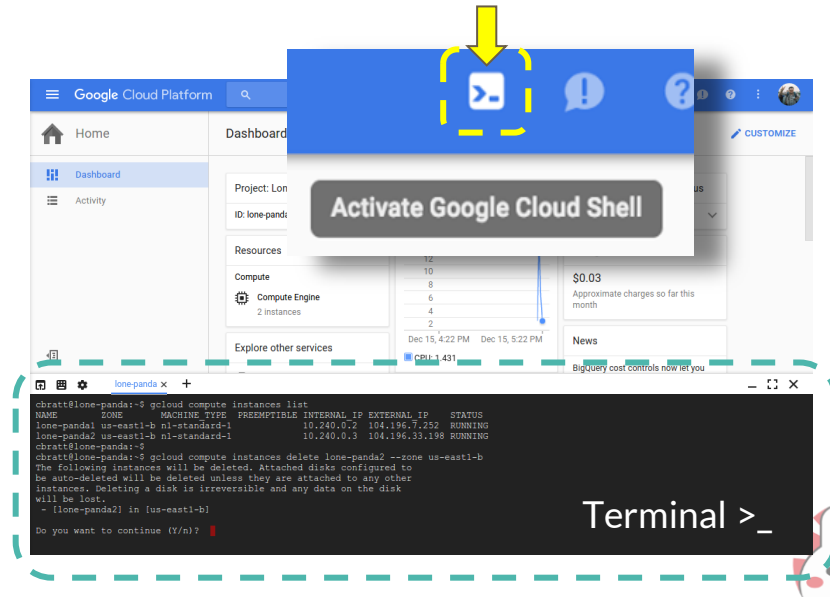
Cloud Console

หน้าเว็บ UI ของ Google Cloud
<https://console.cloud.google.com>



Cloud Shell

เครื่อง VM ชั่วคราว เพื่อใช้รัน command line



DATATH

คำสั่ง Bash ที่เราจะเรียนกันวันนี้



1. คำสั่งพื้นฐานสำหรับห้องโลก

- ls
- pwd
- cd

2. Print, สร้าง/อ่านไฟล์

- echo
- cat
- more / less
- touch

3. การสร้าง directory & การจัดการกับไฟล์

- mkdir
- cp
- mv
- rm

4. คำสั่งอื่น ๆ แกรมเพิ่มเติม

- wget
- unzip / zip
- wc
- Bonus: man
- Bonus: วิธีออกจาก vim



1. คำสั่งพื้นฐาน: ท่องโลก terminal



BASH
THE BOURNE-AGAIN SHELL

- ls
- pwd
- cd

ls : list ดูว่ามีไฟล์อะไรบ้าง

คำสั่ง ls สำหรับ list file ใน directory ปัจจุบัน จากในตัวอย่างจะเห็นว่ามี 1 ไฟล์

```
$ ls
README-cloudshell.txt
```

คำสั่ง ls -l สำหรับ list file แบบละเอียดมากขึ้น ที่บอกถึง mode ของไฟล์ เจ้าของไฟล์ เวลาแก้ไขล่าสุด

```
$ ls -l
-rw-r--r-- 1 user user 913 Sep 1 16:08 README-cloudshell.txt
```

Mode ของไฟล์
เจ้าของ/กรุ๊ป/คนอื่นๆ

Owner ของไฟล์
Group ของไฟล์

ขนาดไฟล์
(byte)

วันที่ที่
แก้ไข

ชื่อไฟล์

d: directory
r: read
w: write
x: execute

เกร็ดเสริม -h หรือ human readable จะอ่านขนาดของไฟล์เป็น KB, MB, GB เพื่อความเข้าใจง่าย
-a หรือ all จะโชว์ไฟล์ที่ซ่อนอยู่ได้ โดยไฟล์ที่ซ่อน จะเป็นไฟล์ที่ขึ้นต้นด้วยจุด (.)



pwd : print working directory

คำสั่ง pwd สำหรับ print full path ว่าตอนนี้อยู่ที่ directory อะไร

```
$ pwd  
/home/username
```

โดยปกติแล้ว เมื่อเริ่มต้น จะเริ่มที่ **HOME (/home/username)** มักแทนด้วยสัญลักษณ์ ~ (tilde)

Full path ของระบบ Unix จะเริ่มที่ **/ (root)** เสมอ โฟลเดอร์ที่อยู่ภายใต้ root มักเป็น folder ที่เกี่ยวข้องกับระบบ และไม่สามารถแก้ไขได้ด้วยสิทธิ์ของ user ทั่วไป

Relative path คือ path ที่อ้างอิงจาก path ปัจจุบัน หรือ สามารถอ้างอิงจาก HOME (~) ได้เช่นกัน



cd : change directory

คำสั่ง cd สำหรับการเปลี่ยน directory ไปยัง path อื่นๆ คำสั่งนี้ มักใช้คู่กับ ls และ pwd

```
$ cd [PATH]
$ cd ~
$ pwd
/home/username
```

สัญลักษณ์พิเศษที่ใช้บ่อย

~	HOME	(~ Tilde)
/	ROOT	(/ Slash)
.	Current Directory	
..	Parent Directory	

Home (~) คือ directory หลักของ user ใช้ในการอ้างอิง path

Root (/) คือ จุดเริ่มต้นของ path โดย full path จะเริ่มจาก / เสมอ

. แทน directory ปัจจุบันที่ใช้งานอยู่

.. แทน directory ก่อนหน้าขึ้นไปหนึ่งชั้น (เหมือนถอยออกจากโฟลเดอร์)



2. คำสั่งพื้นฐาน: การ print, สร้าง/อ่าน ไฟล์



BASH
THE BOURNE-AGAIN SHELL

- echo
- cat
- more / less
- touch

echo : print ออกมาบน Terminal

คำสั่ง echo สำหรับการแสดงข้อความหรือตัวแปรออกมาบน Terminal

```
$ echo "Hello World!"  
Hello World!
```

```
$ echo $USER  
username
```

เกร็ดเสริม:

`$Variable` หรือ `${var}`

`$` ใช้หน้าชื่อตัวแปร

คำสั่ง echo ยังใช้สำหรับการ print ข้อความ ลงไปต่อท้ายไฟล์ โดยใช้เครื่องหมาย `>>` ได้อีกด้วย

```
$ echo "Hello World! in a file" >> hello.txt  
$ ls  
README-cloudshell.txt      hello.txt
```

cat : concatenate อ่านไฟล์ออกมาบน Terminal

คำสั่ง cat สำหรับ print เนื้อหาในไฟล์ออกมา

```
$ cat [FILE]...  
$ cat hello.txt  
Hello World! in a file
```

ที่เรียกว่า concatenate เพราะ ถ้าใส่ชื่อไฟล์หลายไฟล์ใน command เดียวกัน จะนำชื่อไฟล์ทั้งหมดมาต่อกัน และยังใช้ร่วมกับเครื่องหมาย >> เพื่อนำ output ไปใส่ไว้ในไฟล์

```
$ cat hello.txt hello.txt >> new_file.txt  
$ cat new_file.txt  
Hello World! in a file  
Hello World! in a file
```

more / less : เปิดไฟล์ เป็นหน้า ๆ

บางครั้ง ถ้าไฟล์มีขนาดใหญ่ การใช้คำสั่ง cat สำหรับ print เนื้อหาไฟล์ จะอ่านยาก เพราะจะแสดงผลเนื้อหาของทั้งไฟล์ออกมา ในกรณีนี้ สามารถใช้คำสั่ง more หรือ less แทนได้

```
$ more [FILENAME]
```

```
$ less [FILENAME]
```

ทั้งสองคำสั่ง มีความคล้ายคลึงกัน คือ จะแสดงผลเนื้อหาทีละหน้า

- กด space bar เพื่อที่จะดูหน้าถัดไป
- กด q เพื่อออก

ข้อแตกต่างคือ more จะ print เนื้อหาลงใน terminal ทันที เมื่อจบ command จะเห็นเนื้อหาไฟล์นั้นอยู่ ส่วน less จะแยกการแสดงผลออกเป็นอีกหน้า ทำให้ย้อนดูเนื้อหาก่อนหน้าได้ด้วย และมีฟีเจอร์เยอะกว่า



touch : สร้างไฟล์เปล่า

คำสั่ง touch สำหรับการสร้างไฟล์เปล่า ขึ้นมาหนึ่งไฟล์

```
$ touch [FILENAME]  
$ touch blank.txt  
$ cat touch.txt
```

เมื่อลอง cat คุณก็จะไม่เห็นอะไร เพราะว่าเป็นไฟล์เปล่านั่นเอง

คำสั่งนี้อาจจะไม่ค่อยได้ใช้มาก แต่ก็มีประโยชน์ในการสร้างไฟล์อย่างรวดเร็ว



คำสั่ง Bash ที่เราเรียนไปแล้ว



1. คำสั่งพื้นฐานสำหรับห้องโลก

- ls
- pwd
- cd

2. Print, สร้าง/อ่านไฟล์

- echo
- cat
- more / less
- touch

3. การสร้าง directory & การจัดการกับไฟล์

- mkdir
- cp
- mv
- rm

4. คำสั่งอื่น ๆ แถมเพิ่มเติม

- wget
- unzip / zip
- wc
- Bonus: man
- Bonus: วิธีออกจาก vim



3. คำสั่งพื้นฐาน: การสร้าง directory & การจัดการกับไฟล์



BASH
THE BOURNE-AGAIN SHELL

- mkdir
- cp
- mv
- rm

mkdir : make directory สร้างโฟลเดอร์ใหม่

คำสั่ง mkdir หรือ make directory สำหรับการสร้างโฟลเดอร์ใหม่

```
$ mkdir [DIRECTORY_NAME]
$ mkdir new_folder
$ ls
blank.txt  hello.txt  new_file.txt  new_folder  README-cloudshell.txt
```

เมื่อลอง ls ดู จะเห็น new_folder ที่เป็นตัวหนา และสีที่แตกต่าง บ่งบอกว่าเป็น directory

mkdir ยังสามารถสร้าง folder ที่ซ้อนกันได้หลายชั้น โดยใช้ -p (p: parent คือสร้าง parent ด้วยถ้าไม่มี)

```
$ mkdir -p new_folder/in/folder
```

cp : copy ก๊อปปี้ไฟล์

คำสั่ง cp สำหรับ copy ไฟล์ จากต้นทางไปปลายทาง

ถ้าปลายทางเป็น directory สามารถ copy หลายไฟล์เข้าไปใน directory นั้นได้

```
$ cp [SOURCE]... [DESTINATION]
```

```
$ cp hello.txt new_folder/
```

```
$ ls new_folder
```

```
in hello.txt
```

ถ้าต้องการ copy โฟลเดอร์ ต้องใส่ -r (r: recursive) เพื่อให้ copy ไปยังทุก ๆ ไฟล์ในโฟลเดอร์ต้นทาง

```
$ cp -r new_folder/ new_folder2
```

```
$ ls new_folder2
```

```
in hello.txt
```

mv : move ย้าย / เปลี่ยนชื่อไฟล์ หรือ directory

คำสั่ง mv หรือ move ใช้สำหรับการย้ายไฟล์ หรือ directory ไปอีกที่หนึ่ง เราใช้ mv ในการเปลี่ยนชื่อได้ด้วย

```
$ mv [SOURCE] [DESTINATION]
$ mv new_file.txt new_folder/           # ย้ายไฟล์เข้าไปใน folder new_folder
$ mv new_folder2 good_name              # เปลี่ยนชื่อ folder เป็น good_name
$ ls
blank.txt  good_name  hello.txt  new_folder  README-cloudshell.txt
$ ls new_folder
hello.txt  in  new_file.txt
```

rm : remove ลบไฟล์ หายถาวร

คำสั่ง rm สำหรับ remove เพื่อลบไฟล์ออก

```
$ rm [FILENAME]
$ rm blank.txt
$ ls
$ good_name hello.txt new_folder README-cloudshell.txt
```

ถ้าต้องการลบ directory ต้องใช้ -r (r: recursive) เพื่อให้ไปลบไฟล์ย่อยข้างในด้วย

```
$ rm -r new_folder
$ ls
$ good_name hello.txt README-cloudshell.txt
```

4. คำสั่งอื่น ๆ เพิ่มเติม



- `wget`
- `unzip / zip`
- `wc`
- Bonus: `man`
- Bonus: วิธีออกจาก vim

wget : download ไฟล์

คำสั่ง wget สำหรับ การ download ไฟล์จาก URL
-O ใช้สำหรับการตั้งชื่อไฟล์ที่จะโหลดมา

```
$ wget [-O output_name] [URL]
$ wget -O data.zip https://file.designil.com/bhXYol+
data.zip 100%[=====>] 6.39M 1.62MB/s in 3.7s
$ ls *.zip
data.zip
```

ls *.zip ตัว * คือ wildcard ที่ใช้แทน string อะไรก็ได้ ในที่นี้คือ หาไฟล์อะไรก็ได้.zip แล้วได้ผลลัพธ์ออกมาเป็น data.zip

นอกจากคำสั่ง wget ที่ใช้ download file แล้ว

ยังมีคำสั่ง curl ที่ใช้ในการ download file และยิง API ได้เช่นกัน



unzip : แยกไฟล์ zip

คำสั่ง unzip สำหรับการแตกไฟล์ zip ด้วย command-line

```
$ unzip [ZIP_FILE]
$ unzip data.zip
Archive:  data.zip
  inflating:  ws2-output.csv
```

zip : สร้างไฟล์ zip

คำสั่ง zip สำหรับการสร้างไฟล์ zip ด้วย command-line

```
$ zip [ZIP_FILE.zip] [FILE]...
$ zip new_data.zip  ws2_data.csv
adding: ws2_data.csv (deflated 76%)
```

wc : word count นับจำนวนบรรทัด และจำนวนคำ

คำสั่ง wc สำหรับนับจำนวนบรรทัด, จำนวนคำ และ จำนวน byte สามารถใส่ชื่อไฟล์ได้หลายไฟล์

```
$ wc [FILE]...
```

```
$ wc ws2-output.csv
```

```
500001 1126750 29570221 ws2-output.csv
```

จำนวน line
(บรรทัด)

จำนวน word
(คำ)

จำนวน byte

การใช้ wc ในการนับจำนวนบรรทัดเท่านั้น

```
$ wc -l ws2-output.csv
```

```
500001 ws2-output.csv
```

-l คือ แสดงผลแค่ line count ว่าไฟล์มีกี่บรรทัด



DATA TH

Bonus : man อยากรู้คำสั่งไหน เปิดดูคู่มือได้เลย

man มาจาก manual ใช้เพื่อดูคู่มือของคำสั่ง นั้น ๆ ได้ เมื่อดูเสร็จ กด q เพื่อ ออก

```
$ man [COMMAND]
```

ตัวอย่าง

```
$ man wc
```

```
WC(1) User Commands
NAME
  wc - print newline, word, and byte counts for each file

SYNOPSIS
  wc [OPTION]... [FILE]...
  wc [OPTION]... --files0-from=F

DESCRIPTION
  Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified.
  acters delimited by white space.

  With no FILE, or when FILE is -, read standard input.

  The options below may be used to select which counts are printed, always in the following order: newline,

  -c, --bytes
        print the byte counts

  -m, --chars
        print the character counts
```



Bonus : ออกจาก vim ยังไง

vim คือ โปรแกรม text editor ใน terminal ที่มีชื่อเสียงโปรแกรมหนึ่ง พัฒนามาจาก vi (vim - Vi IMproved)
หนึ่งในชื่อเสียงของ vim คือ เข้าแล้วออกไม่ได้

วิธีการเข้า

```
$ vim hello.txt
```

วิธีการออก

```
<ESC> :q
```

ถ้าต้องการแก้ไขไฟล์ จากใน terminal ด้วยโปรแกรมที่ใช้งานง่ายกว่า vim

แนะนำโปรแกรมชื่อ nano (เวลาออก กด ctrl +x)



คำสั่ง Bash ที่เราเรียนกันไปแล้ว



1. คำสั่งพื้นฐานสำหรับห้องโลก

- ls
- pwd
- cd

2. Print, สร้าง/อ่านไฟล์

- echo
- cat
- more / less
- touch

3. การสร้าง directory & การจัดการกับไฟล์

- mkdir
- cp
- mv
- rm

4. คำสั่งอื่น ๆ แถมเพิ่มเติม

- wget
- unzip / zip
- wc
- Bonus: man
- Bonus: วิธีออกจาก vim



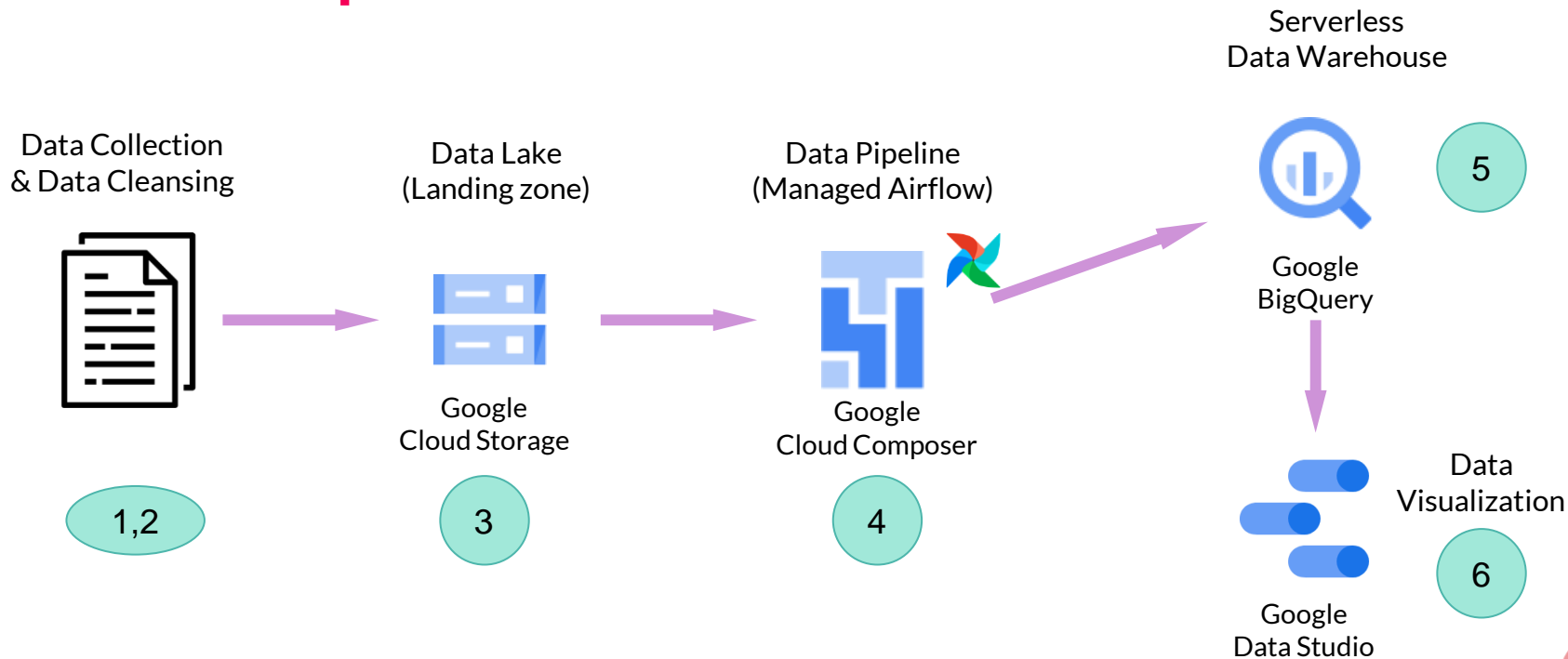
Workshop 3:

Upload to Data Lake



DATATH

Workshops ที่งูมด



วิธีการสมัครสมาชิก Google Cloud และรับเครดิตฟรี

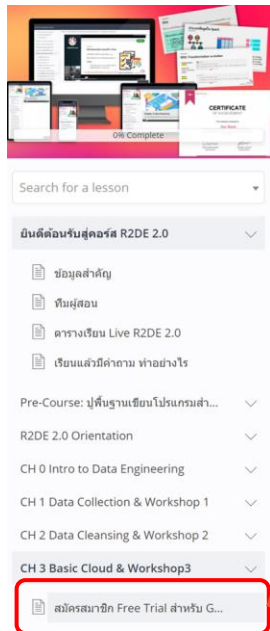
สำหรับลูกค้าใหม่ของ Google Cloud จะได้รับเครดิตฟรี \$300 ใช้ได้ใน 90 วันแรก

วิธีการสมัครสมาชิก Google Cloud

>>

(ต้องใช้บัตรเครดิต หรือบัตรเดบิต ในการสมัครสมาชิก)

* ใช้ในการเรียนบทที่ 3 ถึง 6



CH 3 Basic Cloud & Workshop3 / สมัครสมาชิก Free Trial สำหรับ Google Cloud ...

สำหรับใน Workshop นี้ จำเป็นต้องสมัคร Free Trial จาก Google Cloud สำหรับ Workshop 3 - 6 ผู้สามารถสมัครได้ตั้งแต่วันที่ <https://cloud.google.com/>

เงื่อนไขการสมัคร

1. มีบัญชี Google Account หรือ Gmail ที่ยังไม่ได้ใช้ Free Trial หรือขอ free credit มาก่อน
2. มีบัตรเครดิต เพื่อยืนยันตัวตนและผูกไว้กับบัญชี ในตอนเริ่มต้นอาจจะมีการคิดเงิน 1 USD สำหรับการยืนยันตัวตน

วิดีโอสรุปภายใน 2 นาที

วิธีสมัครสมาชิก Google Cloud (GCP) เพื่อรับฟรีเครดิต \$300 อายุ 90 วัน (2020)



ขั้นตอนในการสมัคร:

1. เข้าไปที่ <https://cloud.google.com/> แล้วกด Get started for free จากนั้นจะขึ้นหน้า Log in ด้วย Google Account



DATA TH



ไฟล์สำหรับ Workshop 3

ลิงค์ไฟล์สำหรับเรียน Workshop 3



Road to Data Engineer 2.0

0% Complete

CERTIFICATE OF ACHIEVEMENT

Search for a lesson

ขั้นตอนรับสู่ออร์ส R2DE 2.0

- ข้อมูลสำคัญ
- ทีมผู้สอน
- ตารางเรียน Live R2DE 2.0
- เรียนแล้วมีคำถาม ทำอย่างไร

Pre-Course: ปูพื้นฐานเขียนโปรแกรมสำ... ▾

R2DE 2.0 Orientation ▾

CH 0 Intro to Data Engineering ▾

CH 1 Data Collection & Workshop 1 ▾

CH 2 Data Cleansing & Workshop 2 ▾

CH 3 Basic Cloud & Workshop3 ▾

สมัครเรียนฟรี Free Trial ส่วนชั้น G

ไฟล์สำหรับเรียน Workshop 3

Workshop 3 – Upload to Data Lake

Upload ไฟล์ข้อมูล เข้า Google Cloud Storage ที่เราจะใช้เป็น Data Lake



Prerequisite:

1. สมัครสมาชิก Google Cloud
2. สร้าง Project (Google Cloud อาจจะสร้างให้อัตโนมัติ)
3. สร้าง Bucket เก็บข้อมูล

Input:

- CSV file จาก workshop 2



Output:

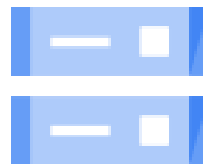
- ข้อมูลเข้าไปอยู่ใน Cloud Storage



gsutil คำสั่งในการใช้งานร่วมกับ Google Cloud

gsutil เป็น command ที่มาพร้อมกับ Cloud SDK ที่ต้อง install เพิ่มถ้าต้องการใช้งานผ่าน command line บน local หรือสามารถใช้งานผ่าน Cloud Shell ได้ทันที

การอ้างอิง path ใน GCS จะต้องขึ้นต้นด้วย gs:// เสมอ



Google
Cloud Storage

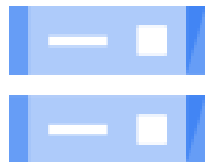
Documentation การใช้งาน gsutil: <https://cloud.google.com/storage/docs/how-to>

gsutil Cheatsheet: <https://bit.ly/gsutil-gist>

Install Cloud SDK: <https://cloud.google.com/sdk/docs/install>



วิธี Upload Data เข้า Google Cloud Storage



Google
Cloud Storage

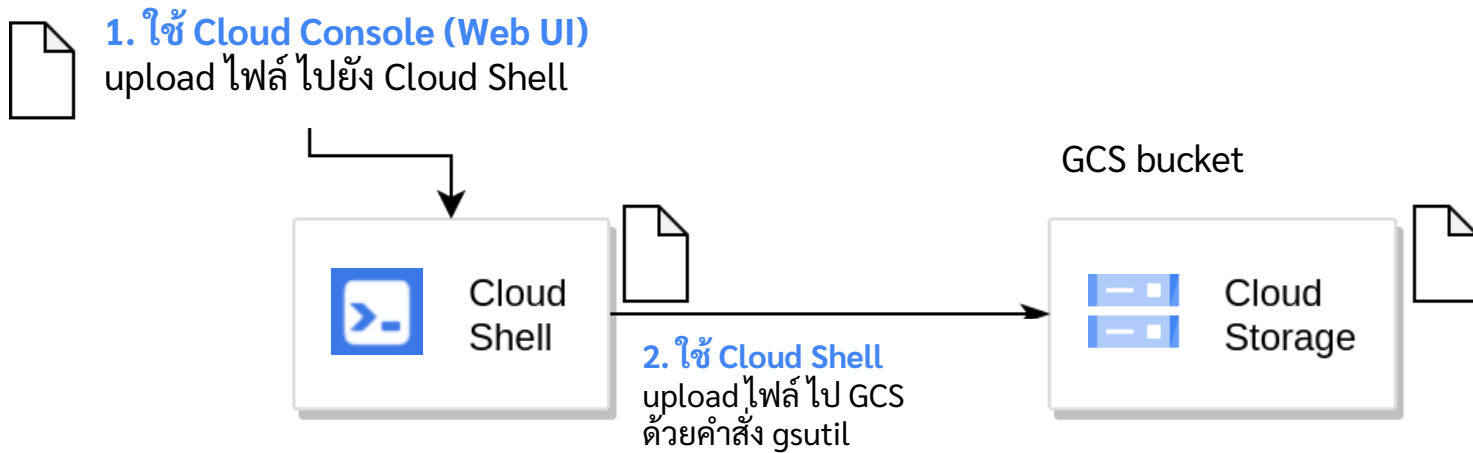
Google Cloud Storage (GCS)

1. สร้าง **Bucket** ทำได้หลายวิธีดังนี้
 - Cloud Console บนเว็บ UI
 - gsutil command ผ่าน Cloud Shell
2. อัปโหลดข้อมูล ทำได้หลายวิธีดังนี้
 - อัปโหลดผ่าน Cloud Console บนเว็บ UI
 - ใช้คำสั่ง gsutil ผ่าน Cloud Shell
 - ใช้โค้ด Python ผ่าน Python SDK library

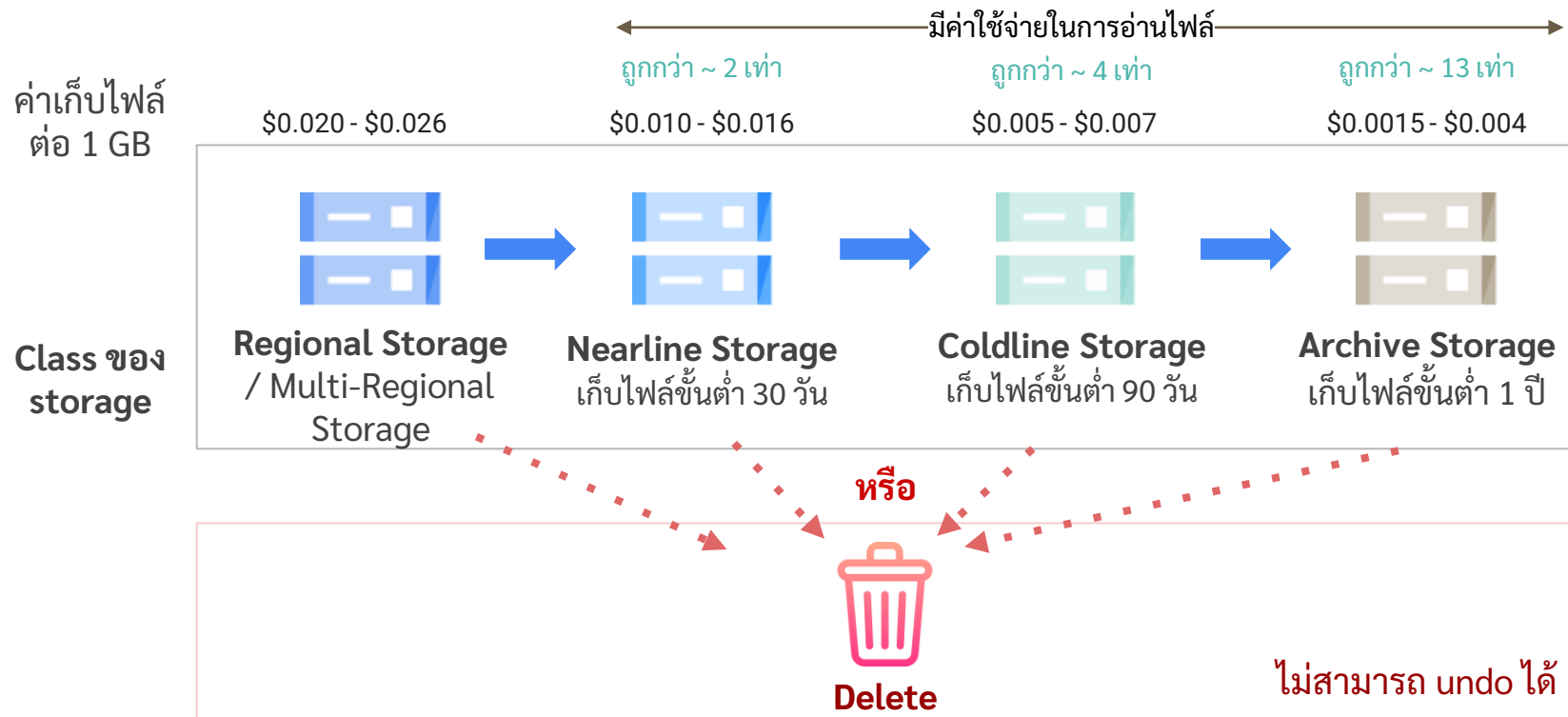
Upload https://cloud.google.com/storage/docs/samples/storage-upload-file#storage_upload_file-python

Download <https://cloud.google.com/storage/docs/downloading-objects#storage-download-object-python>

การใช้ Cloud Shell รับคำสั่ง gsutil เพื่อจัดการ GCS



Bonus: Storage Object Lifecycle จัดการ storage ระยะยาว



รู้จักกับ Storage class ต่างๆ <https://cloud.google.com/storage/docs/storage-classes>

อ้างอิงราคา <https://cloud.google.com/storage/pricing#asia> , <https://cloud.google.com/storage/pricing#archival-pricing>



DATATH