Developer to CTO

CS 7002

Romain Untereiner

Developer to CTO
SESSION 10

Infrastructure

Where to begin to launch reliable services?

To ensure **reliability** and **efficiency** of development we need:

- a safe and efficient coding process

- stable and adaptive infrastructure

- ways to test and debug

- method to apply lightweight changes on services and infrastructure

How to be efficient?

- Don’t rewrite always the same things
- Be able to test afap
- Have a reproducible run scope
- Make your computer work for what it needs

How to be reliable?

- Have single source of truth
- Have reproductible behaviours
- Automate code validations / tests
- Have staging processes
- Have a seamless upgrade system

What is to be DRY ?

Don't Repeat Yourself

Opposed to WET processes: Write Every Time

It implies:

- **normalizing** everything
- **modularizing** the build
- enabling **environment-driven** run

Coding process

WET	DRY
- Run your code on your machine’s host	- Encapsulate the run env
- Hardcode things	- Get env-specific variables from env
- Rewrite common functions	- Modularize everything
- Test isolated software	- Test in-context software
- Release with local validation	- Release with online validation

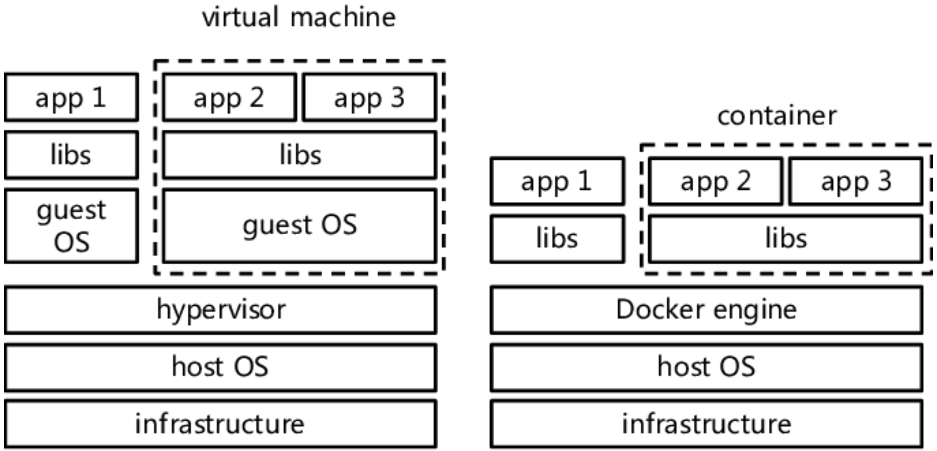
DRY coding process

- Containerization
- Continuous Integration
- Code stacks, run them locally
- Have Development - Staging - Production environment

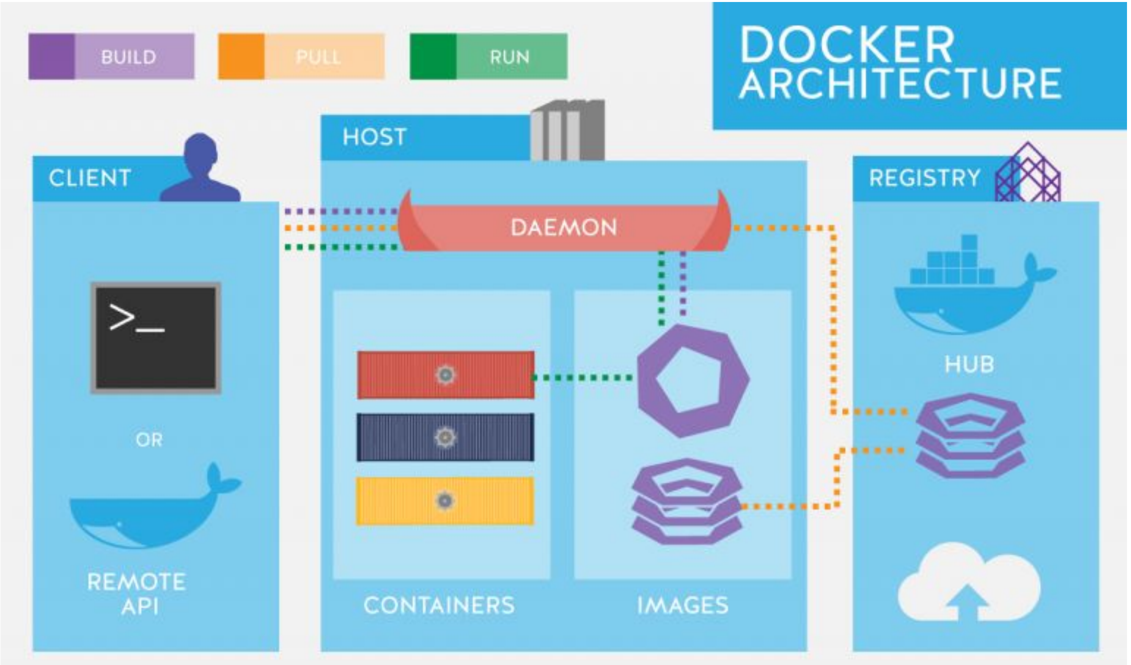
What are a VM and a container?

It’s yet harder, but try to guess

What are a VM and a container?

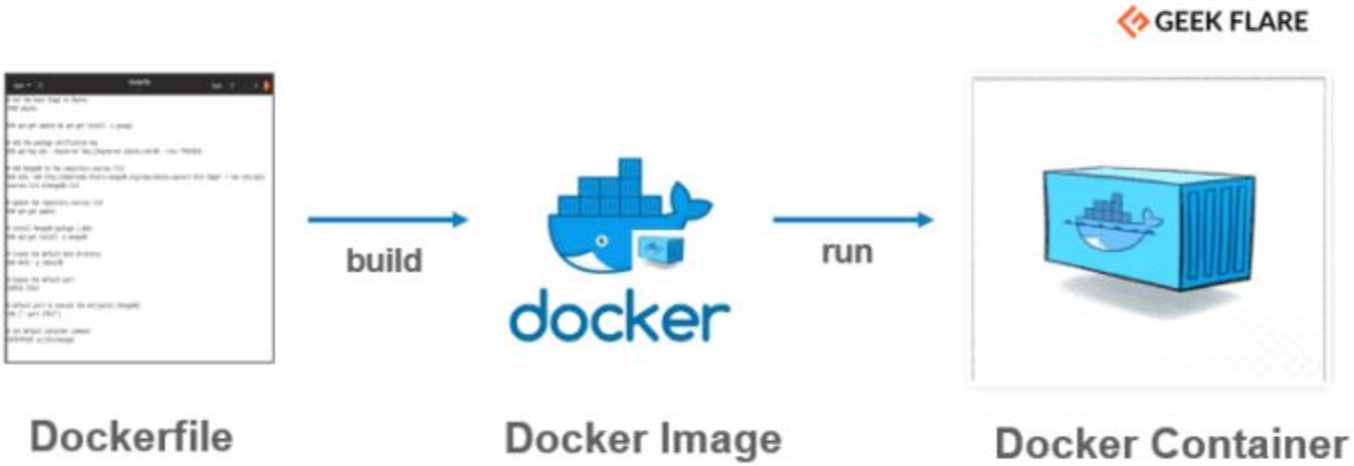


Docker

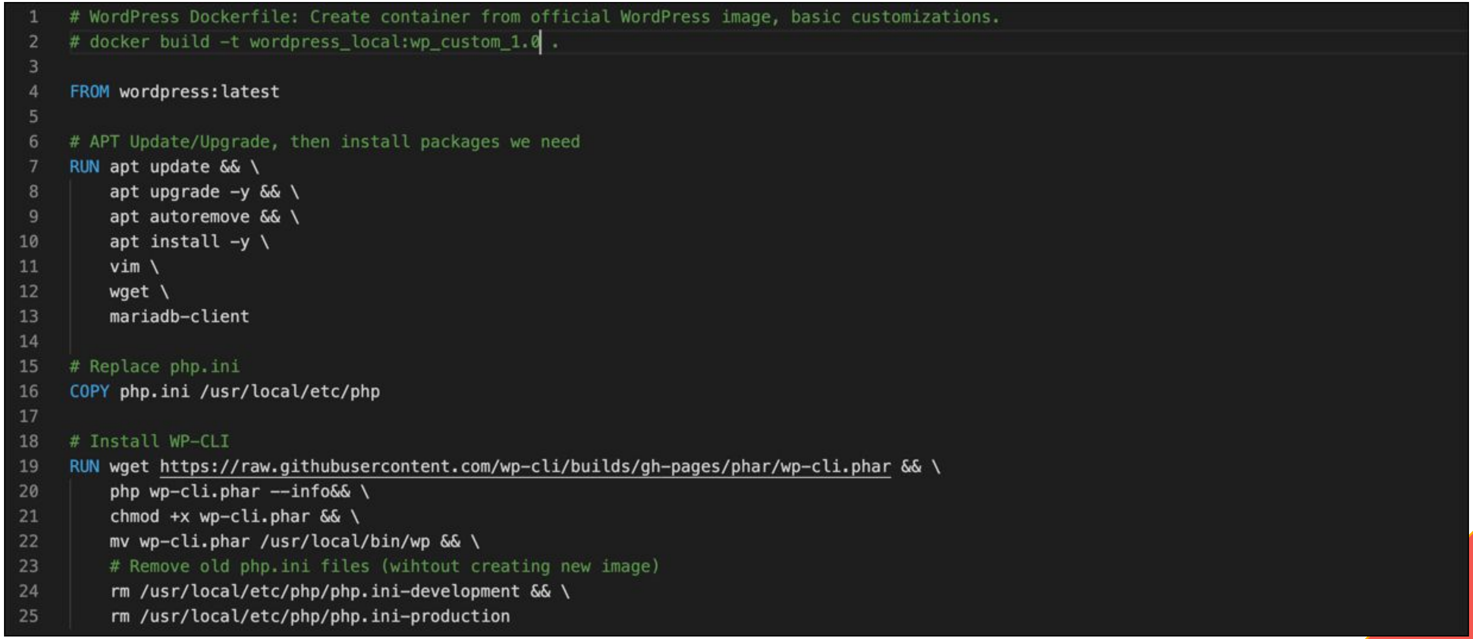


BUILD
PULL
RUN
DOCKER ARCHITECTURE
CLIENT
>_
OR
REMOTE API
HOST
DAEMON
CONTAINERS
IMAGES
REGISTRY
HUB

Docker workflow

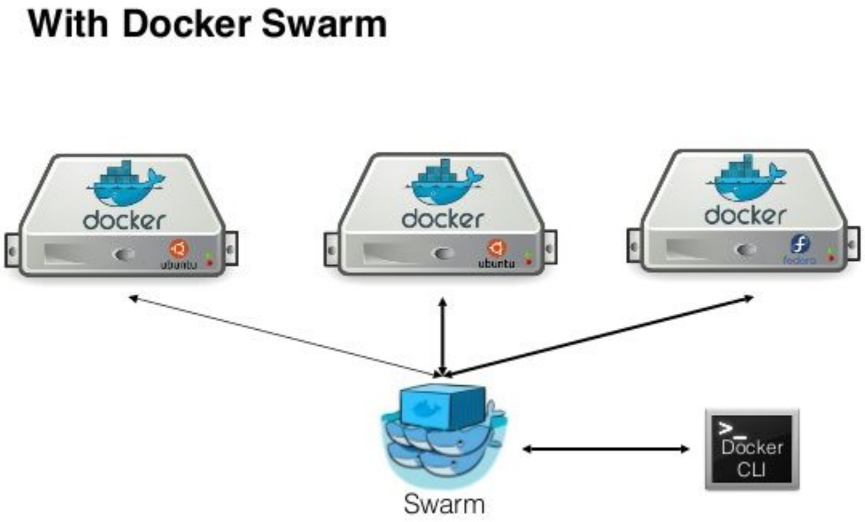


Dockerfile

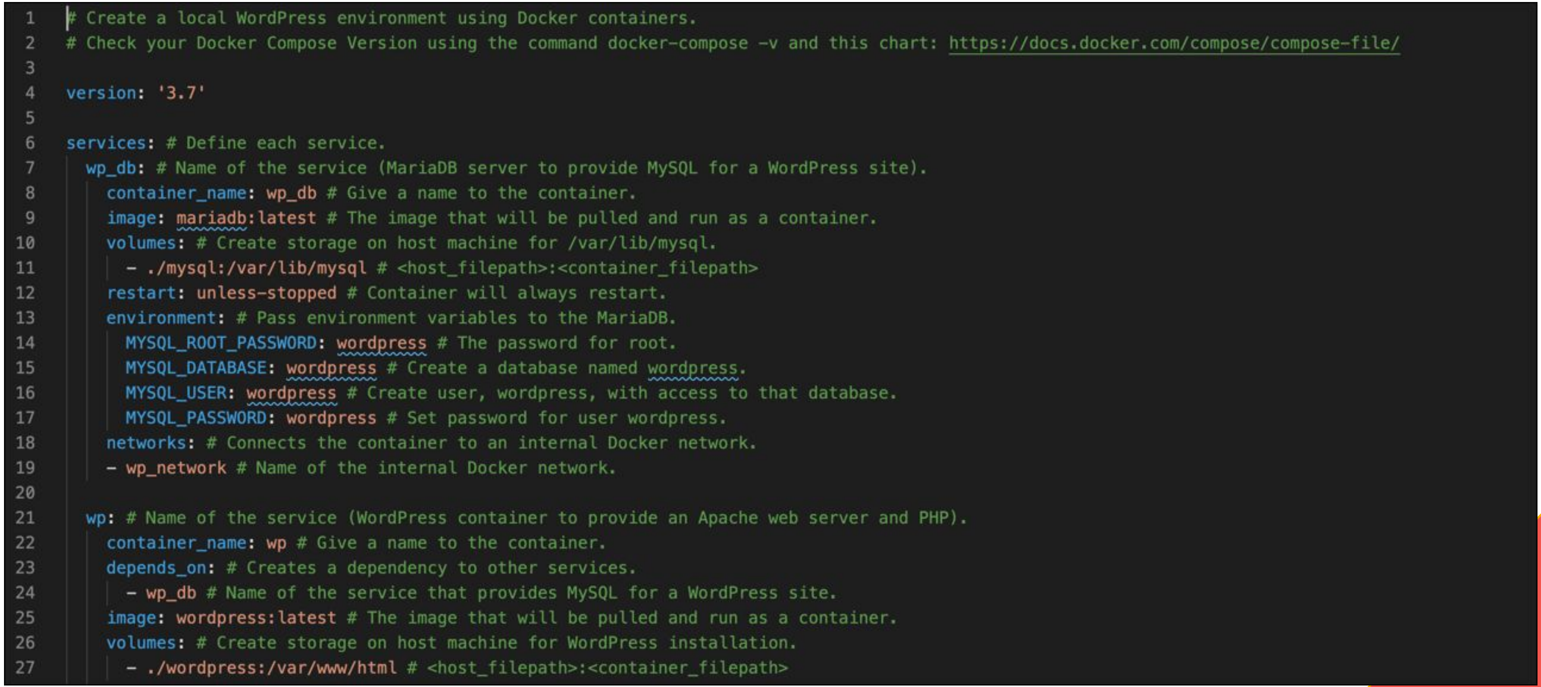


```
# WordPress Dockerfile: Create container from official WordPress image, basic customizations.
# docker build -t wordpress_local:wp_custom_1.0 .

FROM wordpress:latest

# APT Update/Upgrade, then install packages we need
RUN apt update && \
    apt upgrade -y && \
    apt autoremove && \
    apt install -y \
    vim \
    wget \
    mariadb-client

# Replace php.ini
COPY php.ini /usr/local/etc/php

# Install WP-CLI
RUN wget https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar && \
    php wp-cli.phar --info&& \
    chmod +x wp-cli.phar && \
    mv wp-cli.phar /usr/local/bin/wp && \
    # Remove old php.ini files (wihtout creating new image)
    rm /usr/local/etc/php/php.ini-development && \
    rm /usr/local/etc/php/php.ini-production
```

Run stacks

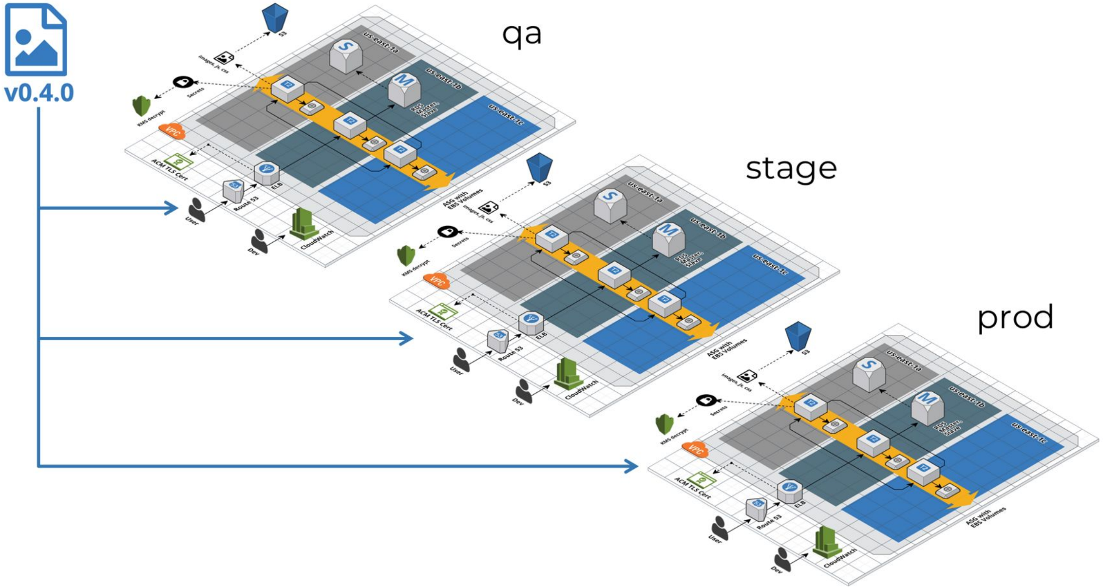

With Docker Swarm
docker
ubuntu
docker
ubuntu
docker
fedora
Swarm
>_
Docker
CLI

Docker compose file



```yaml
# Create a local WordPress environment using Docker containers.
# Check your Docker Compose Version using the command docker-compose -v and this chart: https://docs.docker.com/compose/compose-file/

version: '3.7'

services: # Define each service.
  wp_db: # Name of the service (MariaDB server to provide MySQL for a WordPress site).
    container_name: wp_db # Give a name to the container.
    image: mariadb:latest # The image that will be pulled and run as a container.
    volumes: # Create storage on host machine for /var/lib/mysql.
      - ./mysql:/var/lib/mysql # <host_filepath>:<container_filepath>
    restart: unless-stopped # Container will always restart.
    environment: # Pass environment variables to the MariaDB.
      MYSQL_ROOT_PASSWORD: wordpress # The password for root.
      MYSQL_DATABASE: wordpress # Create a database named wordpress.
      MYSQL_USER: wordpress # Create user, wordpress, with access to that database.
      MYSQL_PASSWORD: wordpress # Set password for user wordpress.
    networks: # Connects the container to an internal Docker network.
    - wp_network # Name of the internal Docker network.

  wp: # Name of the service (WordPress container to provide an Apache web server and PHP).
    container_name: wp # Give a name to the container.
    depends_on: # Creates a dependency to other services.
      - wp_db # Name of the service that provides MySQL for a WordPress site.
    image: wordpress:latest # The image that will be pulled and run as a container.
    volumes: # Create storage on host machine for WordPress installation.
      - ./wordpress:/var/www/html # <host_filepath>:<container_filepath>
```

Setup environments



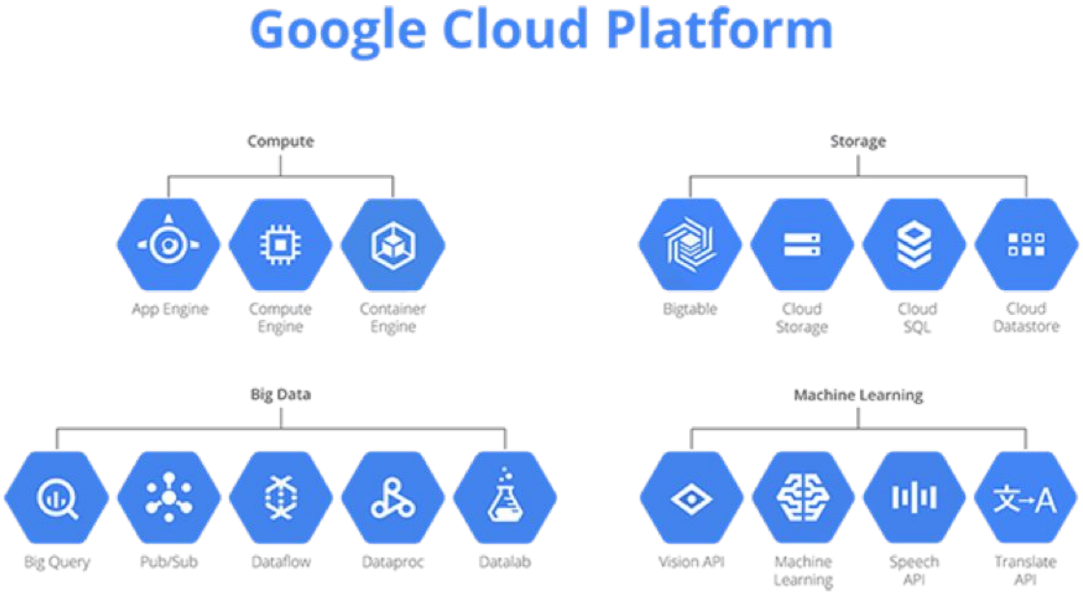
Infrastructure

WET	DRY
- Setup infrastructure manually	- Use the cloud and IaaC
- Change server configurations manually	- Containers & Clustering
- Need disruptive upgrades	- Continuous deployment
- Be locked by the context	- Put context into environment
- Scale manually	- Auto-scaling

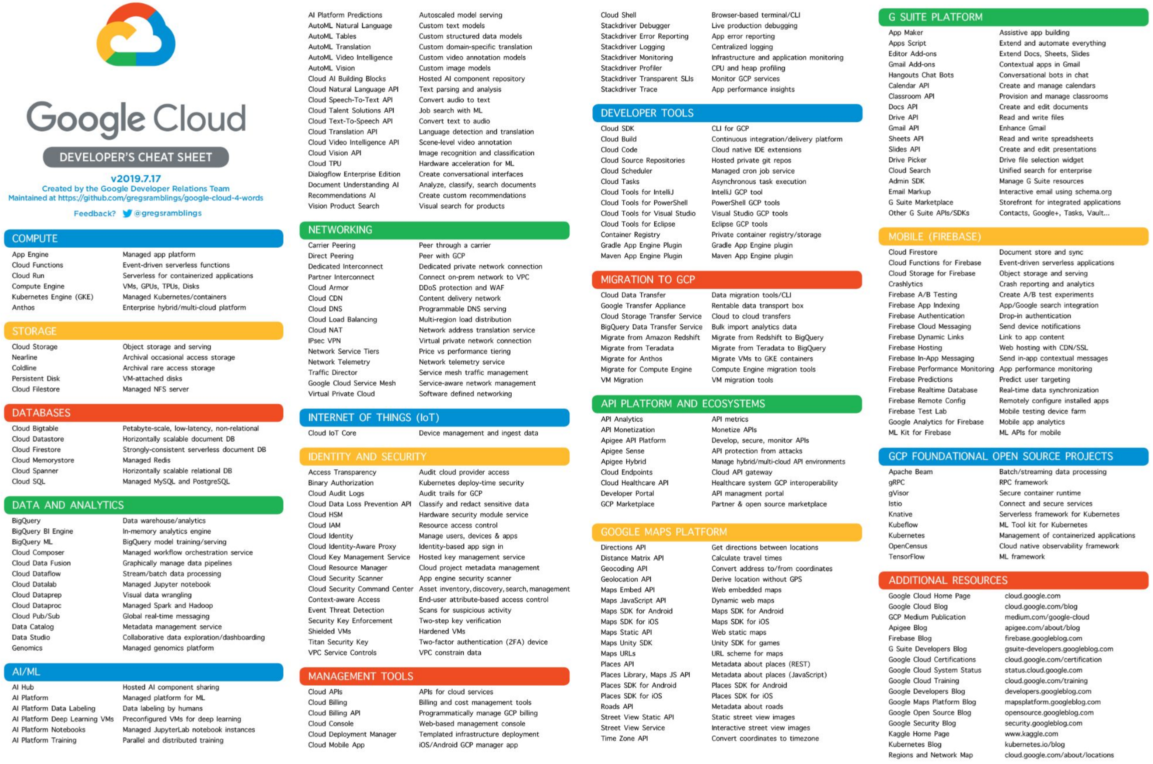
DRY infrastructure

- Containerization & Clustering
- Continuous Deployment
- Infrastructure As A Code
- Cloud-agnosticism

Cloud


Google Cloud Platform
Compute
App Engine
Compute Engine
Container Engine
Storage
Bigtable
Cloud Storage
Cloud SQL
Cloud Datastore
Big Data
Big Query
Pub/Sub
Dataflow
Dataproc
Datalab
Machine Learning
Vision API
Machine Learning
Speech API
Translate API

Cloud



Google Cloud

DEVELOPER'S CHEAT SHEET

v2019.7.17

Created by the Google Developer Relations Team

Maintained at https://github.com/gregsramblings/google-cloud-4-words

Feedback? @gregsramblings

COMPUTE

App Engine	Managed app platform
Cloud Functions	Event-driven serverless functions
Cloud Run	Serverless for containerized applications
Compute Engine	VMs, GPUs, TPUs, Disks
Kubernetes Engine (GKE)	Managed Kubernetes/containers
Anthos	Enterprise hybrid/multi-cloud platform

STORAGE

Cloud Storage	Object storage and serving
Nearline	Archival occasional access storage
Coldline	Archival rare access storage
Persistent Disk	VM-attached disks
Cloud Filestore	Managed NFS server

DATABASES

Cloud Bigtable	Petabyte-scale, low-latency, non-relational
Cloud Datastore	Horizontally scalable document DB
Cloud Firestore	Strongly-consistent serverless document DB
Cloud Memorystore	Managed Redis
Cloud Spanner	Horizontally scalable relational DB
Cloud SQL	Managed MySQL and PostgreSQL

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 Datalab	Managed Jupyter notebook
Cloud Dataprep	Visual data wrangling
Cloud Dataproc	Managed Spark and Hadoop
Cloud Pub/Sub	Global real-time messaging
Data Catalog	Metadata management service
Data Studio	Collaborative data exploration/dashboarding
Genomics	Managed genomics platform

AI/ML

AI Hub	Hosted AI component sharing
AI Platform	Managed platform for ML
AI Platform Data Labeling	Data labeling by humans
AI Platform Deep Learning VMs	Preconfigured VMs for deep learning
AI Platform Notebooks	Managed JupyterLab notebook instances
AI Platform Training	Parallel and distributed training
AI Platform Predictions	Autoscaled model serving
AutoML Natural Language	Custom text models
AutoML Tables	Custom structured data models
AutoML Translation	Custom domain-specific translation
AutoML Video Intelligence	Custom video annotation models
AutoML Vision	Custom image models
Cloud AI Building Blocks	Hosted AI component repository
Cloud Natural Language API	Text parsing and analysis
Cloud Speech-To-Text API	Convert audio to text
Cloud Talent Solutions API	Job search with ML
Cloud Text-To-Speech API	Convert text to audio
Cloud Translation API	Language detection and translation
Cloud Video Intelligence API	Scene-level video annotation
Cloud Vision API	Image recognition and classification
Cloud TPU	Hardware acceleration for ML
Dialogflow Enterprise Edition	Create conversational interfaces
Document Understanding AI	Analyze, classify, search documents
Recommendations AI	Create custom recommendations
Vision Product Search	Visual search for products

NETWORKING

Carrier Peering	Peer through a carrier
Direct Peering	Peer with GCP
Dedicated Interconnect	Dedicated private network connection
Partner Interconnect	Connect on-prem network to VPC
Cloud Armor	DDoS protection and WAF
Cloud CDN	Content delivery network
Cloud DNS	Programmable DNS serving
Cloud Load Balancing	Multi-region load distribution
Cloud NAT	Network address translation service
IPsec VPN	Virtual private network connection
Network Service Tiers	Price vs performance tiering
Network Telemetry	Network telemetry service
Traffic Director	Service mesh traffic management
Google Cloud Service Mesh	Service-aware network management
Virtual Private Cloud	Software defined networking

INTERNET OF THINGS (IoT)

Cloud IoT Core	Device management and ingest data

IDENTITY AND SECURITY

Access Transparency	Audit cloud provider access
Binary Authorization	Kubernetes deploy-time security
Cloud Audit Logs	Audit trails for GCP
Cloud Data Loss Prevention API	Classify and redact sensitive data
Cloud HSM	Hardware security module service
Cloud IAM	Resource access control
Cloud Identity	Manage users, devices & apps
Cloud Identity-Aware Proxy	Identity-based app sign in
Cloud Key Management Service	Hosted key management service
Cloud Resource Manager	Cloud project metadata management
Cloud Security Scanner	App engine security scanner
Cloud Security Command Center	Asset inventory, discovery, search, management
Context-aware Access	End-user attribute-based access control
Event Threat Detection	Scans for suspicious activity
Security Key Enforcement	Two-step key verification
Shielded VMs	Hardened VMs
Titan Security Key	Two-factor authentication (2FA) device
VPC Service Controls	VPC constrain data

MANAGEMENT TOOLS

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

DEVELOPER TOOLS

Cloud SDK	CLI for GCP
Cloud Build	Continuous integration/delivery platform
Cloud Code	Cloud native IDE extensions
Cloud Source Repositories	Hosted private git repos
Cloud Scheduler	Managed cron job service
Cloud Tasks	Asynchronous task execution
Cloud Tools for IntelliJ	IntelliJ GCP tool
Cloud Tools for PowerShell	PowerShell GCP tools
Cloud Tools for Visual Studio	Visual Studio GCP tools
Cloud Tools for Eclipse	Eclipse GCP tools
Container Registry	Private container registry/storage
Gradle App Engine Plugin	Gradle App Engine plugin
Maven App Engine Plugin	Maven App Engine plugin

MIGRATION TO GCP

Cloud Data Transfer	Data migration tools/CLI
Google Transfer Appliance	Rentable data transport box
Cloud Storage Transfer Service	Cloud to cloud transfers
BigQuery Data Transfer Service	Bulk import analytics data
Migrate from Amazon Redshift	Migrate from Redshift to BigQuery
Migrate from Teradata	Migrate from Teradata to BigQuery
Migrate for Anthos	Migrate VMs to GKE containers
Migrate for Compute Engine	Compute Engine migration tools
VM Migration	VM migration tools

API PLATFORM AND ECOSYSTEMS

API Analytics	API metrics
API Monetization	Monetize APIs
Apigee API Platform	Develop, secure, monitor APIs
Apigee Sense	API protection from attacks
Apigee Hybrid	Manage hybrid/multi-cloud API environments
Cloud Endpoints	Cloud API gateway
Cloud Healthcare API	Healthcare system GCP interoperability
Developer Portal	API managment portal
GCP Marketplace	Partner & open source marketplace

GOOGLE MAPS PLATFORM

Directions API	Get directions between locations
Distance Matrix API	Calculate travel times
Geocoding API	Convert address to/from coordinates
Geolocation API	Derive location without GPS
Maps Embed API	Web embedded maps
Maps JavaScript API	Dynamic web maps
Maps SDK for Android	Maps SDK for Android
Maps SDK for iOS	Maps SDK for iOS
Maps Static API	Web static maps
Maps Unity SDK	Unity SDK for games
Maps URLs	URL scheme for maps
Places API	Metadata about places (REST)
Places Library, Maps JS API	Metadata about places (JavaScript)
Places SDK for Android	Places SDK for Android
Places SDK for iOS	Places SDK for iOS
Roads API	Metadata about roads
Street View Static API	Static street view images
Street View Service	Interactive street view images
Time Zone API	Convert coordinates to timezone

G SUITE PLATFORM

App Maker	Assistive app building
Apps Script	Extend and automate everything
Editor Add-ons	Extend Docs, Sheets, Slides
Gmail Add-ons	Contextual apps in Gmail
Hangouts Chat Bots	Conversational bots in chat
Calendar API	Create and manage calendars
Classroom API	Provision and manage classrooms
Docs API	Create and edit documents
Drive API	Read and write files
Gmail API	Enhance Gmail
Sheets API	Read and write spreadsheets
Slides API	Create and edit presentations
Drive Picker	Drive file selection widget
Cloud Search	Unified search for enterprise
Admin SDK	Manage G Suite resources
Email Markup	Interactive email using schema.org
G Suite Marketplace	Storefront for integrated applications
Other G Suite APIs/SDKs	Contacts, Google+, Tasks, Vault...

MOBILE (FIREBASE)

Cloud Firestore	Document store and sync
Cloud Functions for Firebase	Event-driven serverless applications
Cloud Storage for Firebase	Object storage and serving
Crashlytics	Crash reporting and analytics
Firebase A/B Testing	Create A/B test experiments
Firebase App Indexing	App/Google search integration
Firebase Authentication	Drop-in authentication
Firebase Cloud Messaging	Send device notifications
Firebase Dynamic Links	Link to app content
Firebase Hosting	Web hosting with CDN/SSL
Firebase In-App Messaging	Send in-app contextual messages
Firebase Performance Monitoring	App performance monitoring
Firebase Predictions	Predict user targeting
Firebase Realtime Database	Real-time data synchronization
Firebase Remote Config	Remotely configure installed apps
Firebase Test Lab	Mobile testing device farm
Google Analytics for Firebase	Mobile app analytics
ML Kit for Firebase	ML APIs for mobile

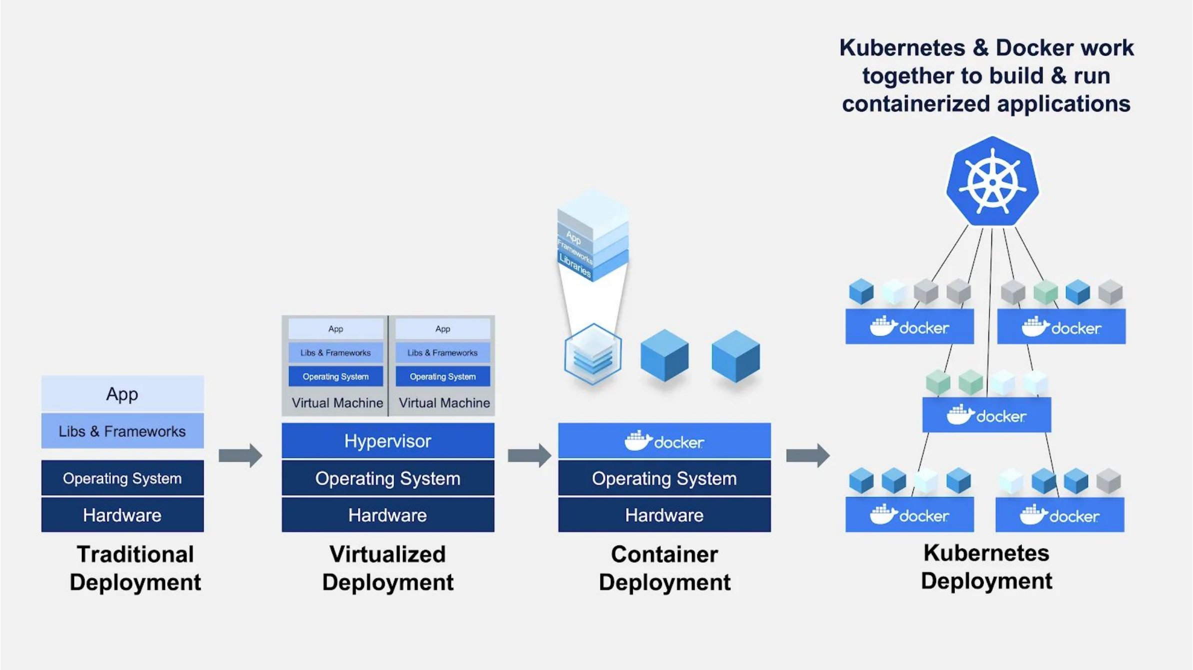
GCP FOUNDATIONAL OPEN SOURCE PROJECTS

Apache Beam	Batch/streaming data processing
gRPC	RPC framework
gVisor	Secure container runtime
Istio	Connect and secure services
Knative	Serverless framework for Kubernetes
Kubeflow	ML Tool kit for Kubernetes
Kubernetes	Management of containerized applications
OpenCensus	Cloud native observability framework
TensorFlow	ML framework

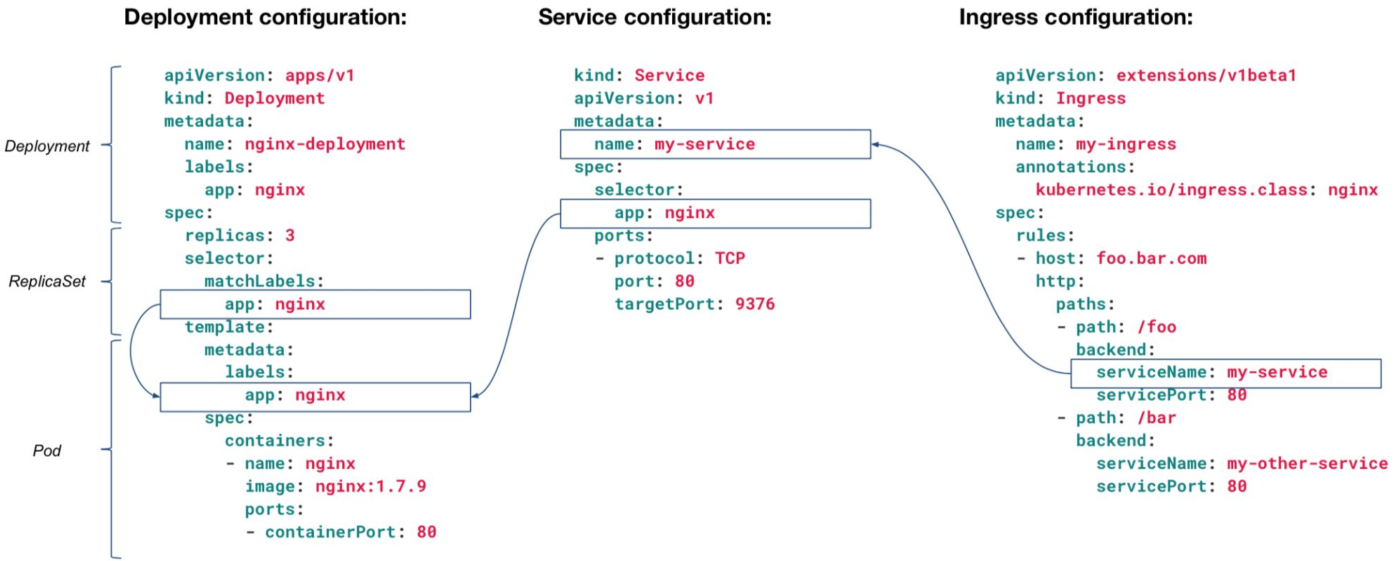
ADDITIONAL RESOURCES

Google Cloud Home Page	cloud.google.com
Google Cloud Blog	cloud.google.com/blog
GCP Medium Publication	medium.com/google-cloud
Apigee Blog	apigee.com/about/blog
Firebase Blog	firebase.googleblog.com
G Suite Developers Blog	gsuite-developers.googleblog.com
Google Cloud Certifications	cloud.google.com/certification
Google Cloud System Status	status.cloud.google.com
Google Cloud Training	cloud.google.com/training
Google Developers Blog	developers.googleblog.com
Google Maps Platform Blog	mapsplatform.googleblog.com
Google Open Source Blog	opensource.googleblog.com
Google Security Blog	security.googleblog.com
Kaggle Home Page	www.kaggle.com
Kubernetes Blog	kubernetes.io/blog
Regions and Network Map	cloud.google.com/about/locations

Containers in Kubernetes



Kubernetes manifests



Deployment configuration:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

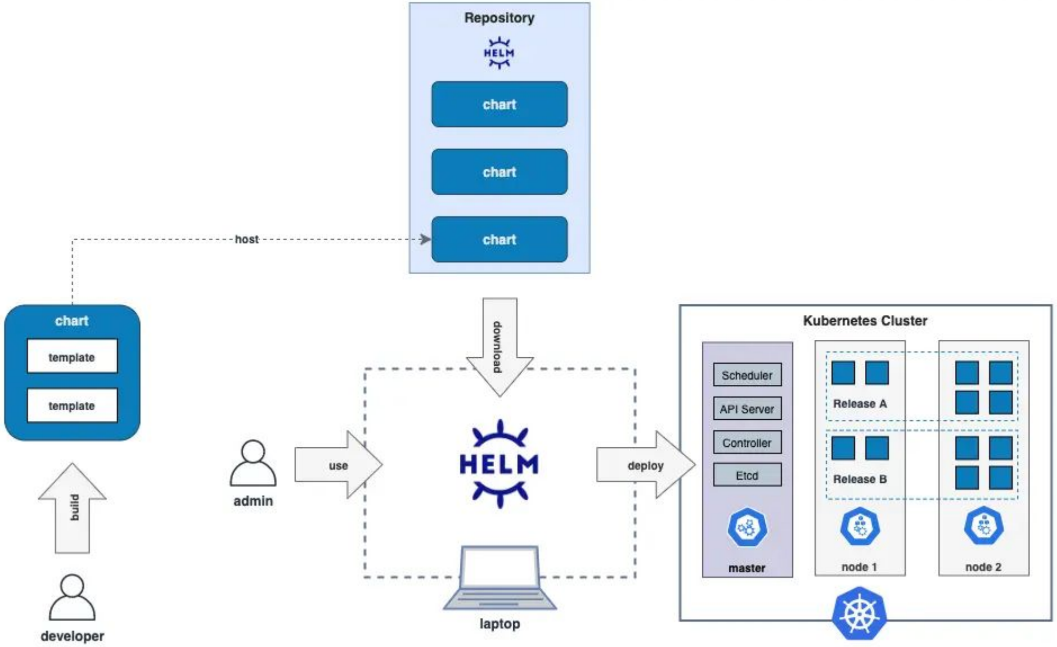
Service configuration:

```
kind: Service
apiVersion: v1
metadata:
  name: my-service
spec:
  selector:
    app: nginx
  ports:
  - protocol: TCP
    port: 80
    targetPort: 9376
```

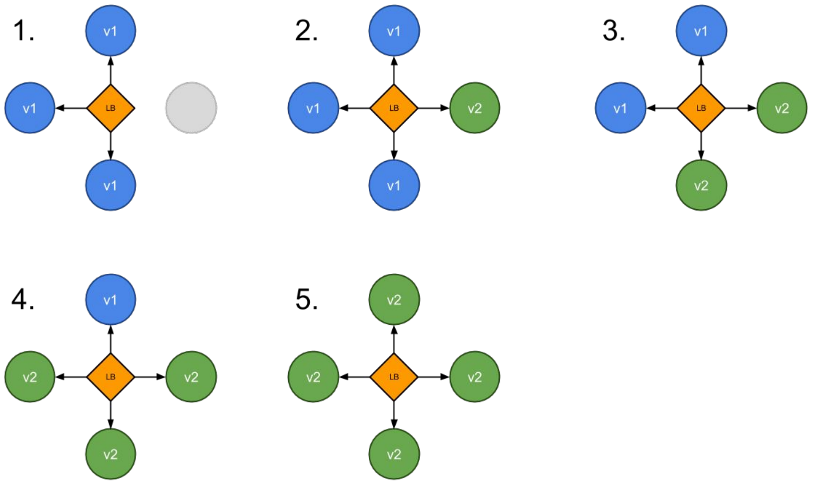
Ingress configuration:

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: my-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /foo
        backend:
          serviceName: my-service
          servicePort: 80
      - path: /bar
        backend:
          serviceName: my-other-service
          servicePort: 80
```

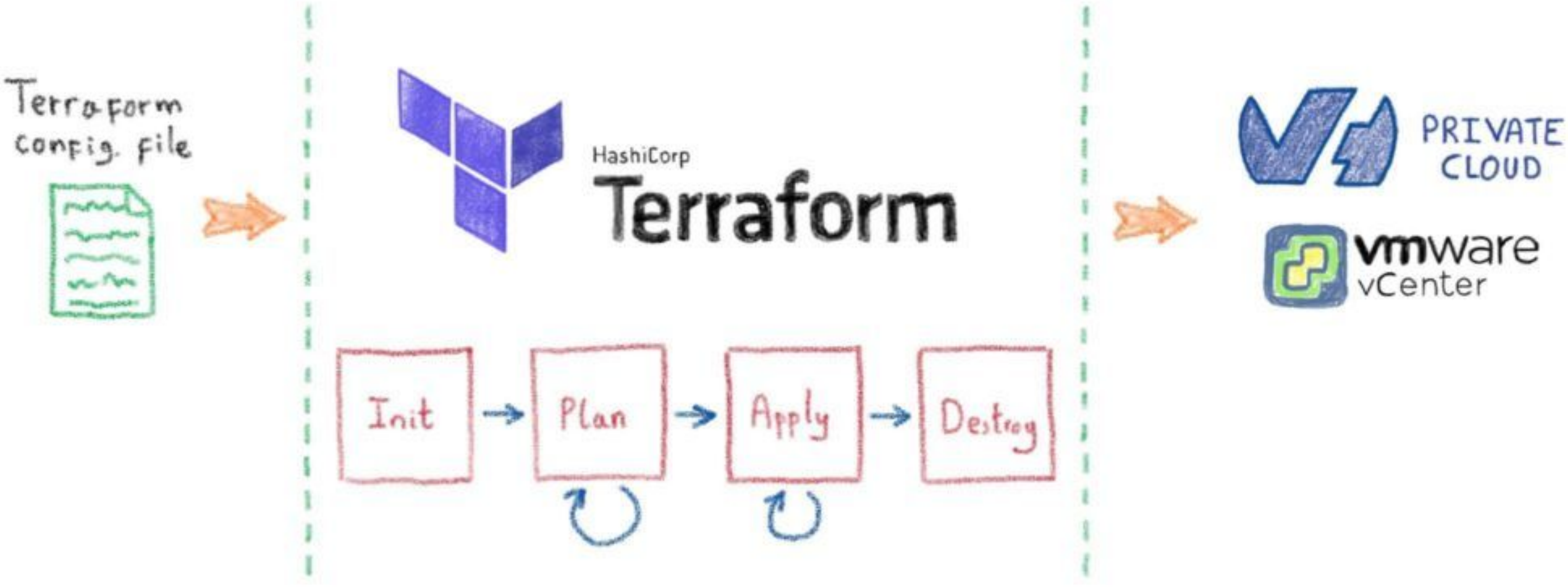
Helm


Repository
HELM
chart
chart
chart
host
chart
template
template
build
developer
admin
use
download
HELM
laptop
deploy
Kubernetes Cluster
Scheduler
API Server
Controller
Etcd
master
Release A
Release B
node 1
node 2

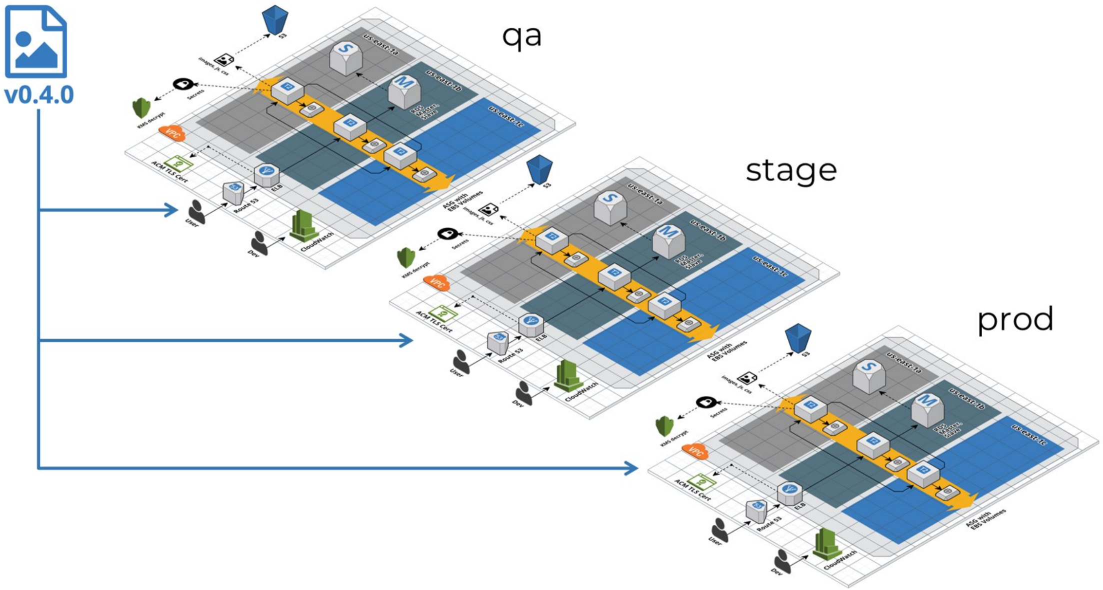
Rolling upgrade



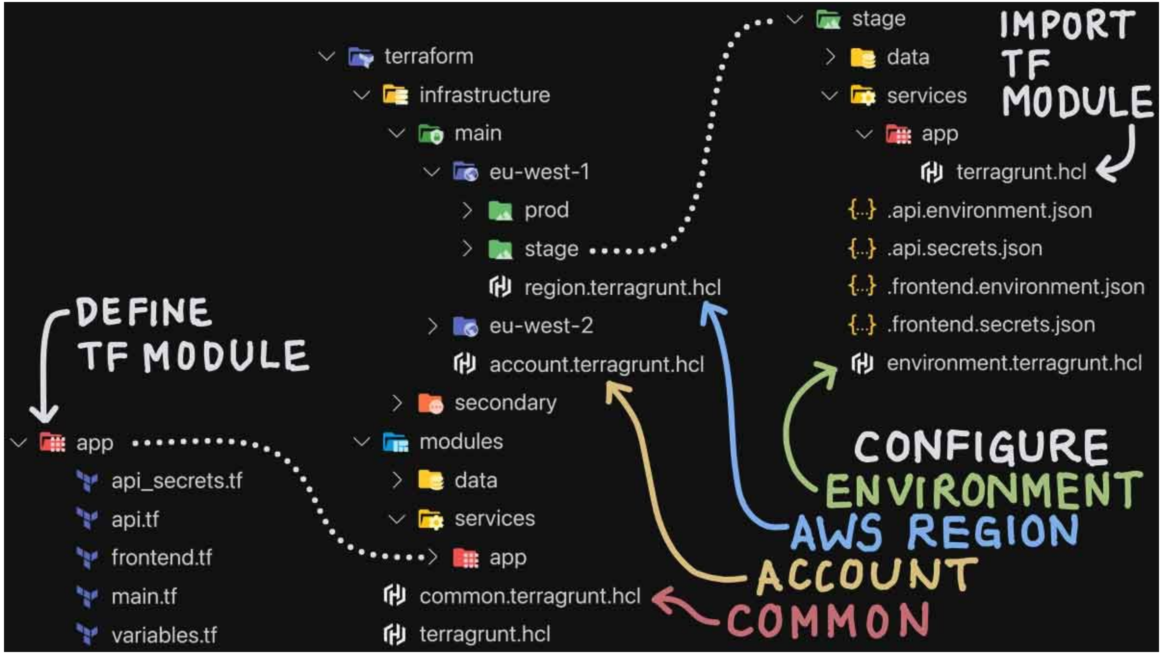
IaaC: terraform



IaaC: terragrunt



IaaC: terragrunt



```
⌄ terraform
  ⌄ infrastructure
    ⌄ main
      ⌄ eu-west-1
        › prod
        › stage
          region.terragrunt.hcl
      › eu-west-2
        account.terragrunt.hcl
    › secondary
  ⌄ modules
    › data
    ⌄ services
      › app
    common.terragrunt.hcl
    terragrunt.hcl
```

```
⌄ app
    api_secrets.tf
    api.tf
    frontend.tf
    main.tf
    variables.tf
```

```
⌄ stage
  › data
  ⌄ services
    ⌄ app
        terragrunt.hcl
    .api.environment.json
    .api.secrets.json
    .frontend.environment.json
    .frontend.secrets.json
    environment.terragrunt.hcl
```

Summary

Have at least **2 “codebases”: software & infra**

- Prepare your cluster(s) and cloud services

- Code the software (code & Dockerfile)
- Prepare the stack (k8s manifests, compose files)
- Build, test and validate locally (Docker, Docker Swarm)
- Commit
  - -> triggers CI (build, test, validation)
  - -> triggers CD (test online, in dev env)

- Merge
- Create version
- Push to production