INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

Dharwad
ज्ञानेन विकासः

PaaS and CaaS (Google – Docker)

Dr. Animesh Chaturvedi

Assistant Professor: IIIT Dharwad

Young Researcher: Heidelberg Laureate Forum

Postdoc: King's College London & The Alan Turing Institute

PhD: IIT Indore MTech: IIITDM Jabalpur

HEIDELBERG LAUREATE FORUM

KING'S College LONDON

Indian Institute of Technology Indore
भारतीय प्रौद्योगिकी संस्थान इंदौर

The Alan Turing Institute

PDPM
Indian Institute of Information Technology, Design and Manufacturing, Jabalpur

PaaS and Container as a Service (CaaS)

A. **PaaS – Google App Engine (GAE)**
B. Container as a Service (CaaS) - DockerHub

Platform as a service (PaaS)

- PaaS allow customers to do application
  - development,
  - execution, and
  - management

- PaaS removes the complexity of building and maintaining the infrastructure for app
  - development and
  - deployment

https://en.m.wikipedia.org/wiki/Platform_as_a_service

Examples of PaaS Clouds

- Amazon Web Services
- Abiquo Enterprise Edition
- CloudStack
- Citrix Cloud
- CtrlS
- DigitalOcean
- EMC Atmos
- Eucalyptus
- Fujitsu
- GoGrid
- Google Cloud Platform
- GreenButton
- Helion
- GE Predix
- Google App Engine
- GreenQloud
- Heroku
- IBM Cloud
- Inktank
- Jelastic
- Mendix
- Microsoft Azure
- MindSphere
- Netlify
- Oracle Cloud
- OutSystems
- openQRM
- OpenShift
- PythonAnywhere
- RightScale
- Scalr
- Force.com
- SAP Cloud Platform
- VMware vCloud Air
- WaveMaker

https://en.m.wikipedia.org/wiki/Platform_as_a_service

GAE: Modern Web Applications

Web app using Google App Engine (GAE) and Google Cloud.

- Quickly reach customers and end users by deploying web apps on App Engine.
- Zero-config deployments and zero server management.
- Only writing code for App Engine.
- App Engine automatically scales to support sudden traffic spikes without provisioning, patching, or monitoring.

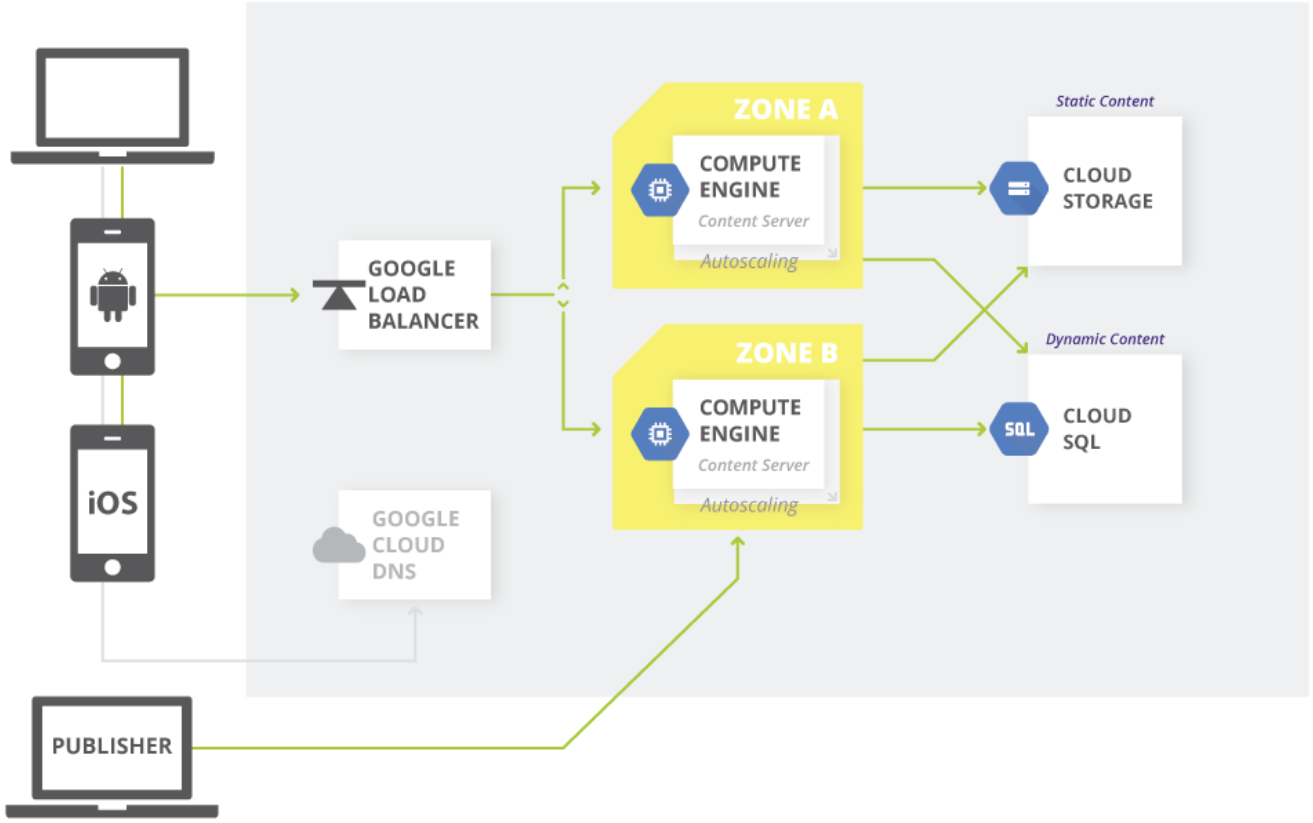
https://cloud.google.com/appengine

Google Content Management

- Web information, marketing campaigns or social media.
- Personalized for individual users or groups.
- Google Cloud Platform (GCP) components and services to create a Content Management system.
- Google Load Balancer to support traffic routed to multiple zones for high availability.
- Google's Cloud DNS provides a robust DNS manages the domain.
- Static content → Cloud Storage
- Dynamic content→ Cloud SQL implementation.

https://cloud.google.com/solutions/architecture/contentmanagement accessed 15 Jan 2015

Google Content Management



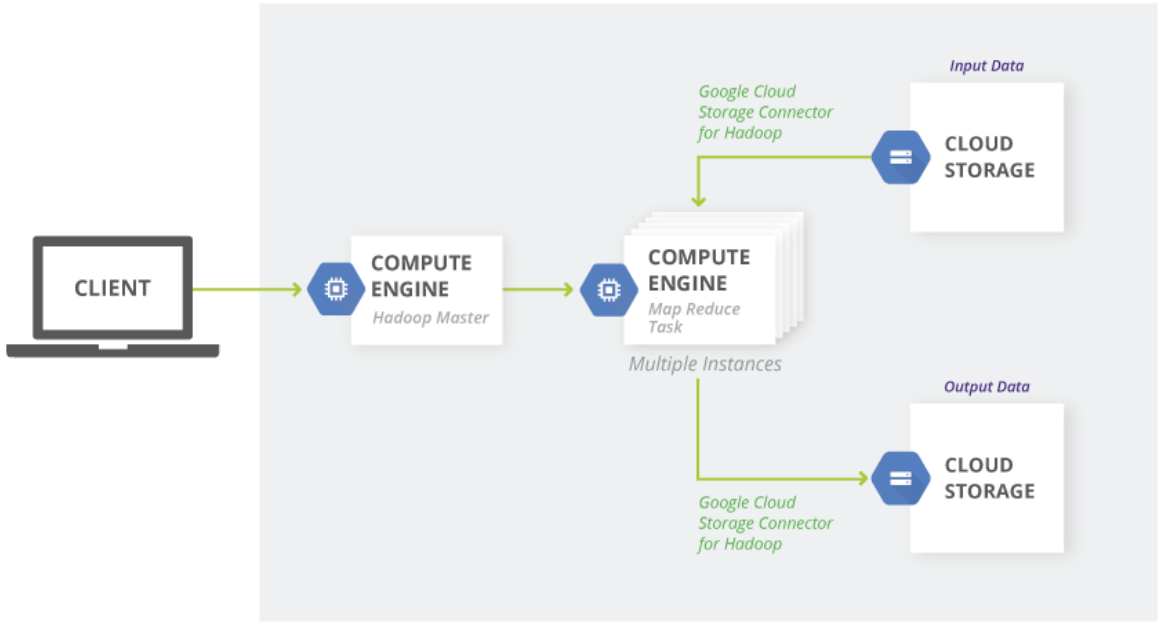
https://cloud.google.com/solutions/architecture/contentmanagement accessed 15 Jan 2015

Architecture: Hadoop on Google Cloud Platform

- Infrastructure for MapReduce using Hadoop.
- Compute power and Cloud Storage to store the input and output of the MapReduce jobs.
- Hadoop Master: includes the HDFS NameNode and the MapReduce JobTracker.
- Nodes in the cluster will run MapReduce tasks with DataNode and MapReduce TaskTracker.
- Backing-up the storage through Google Cloud Storage Connector for Hadoop. HDFS, can be used, Google's Cloud Storage.

https://cloud.google.com/solutions/architecture/hadoop accessed 15 Jan 2015

Architecture: Hadoop on Google Cloud Platform



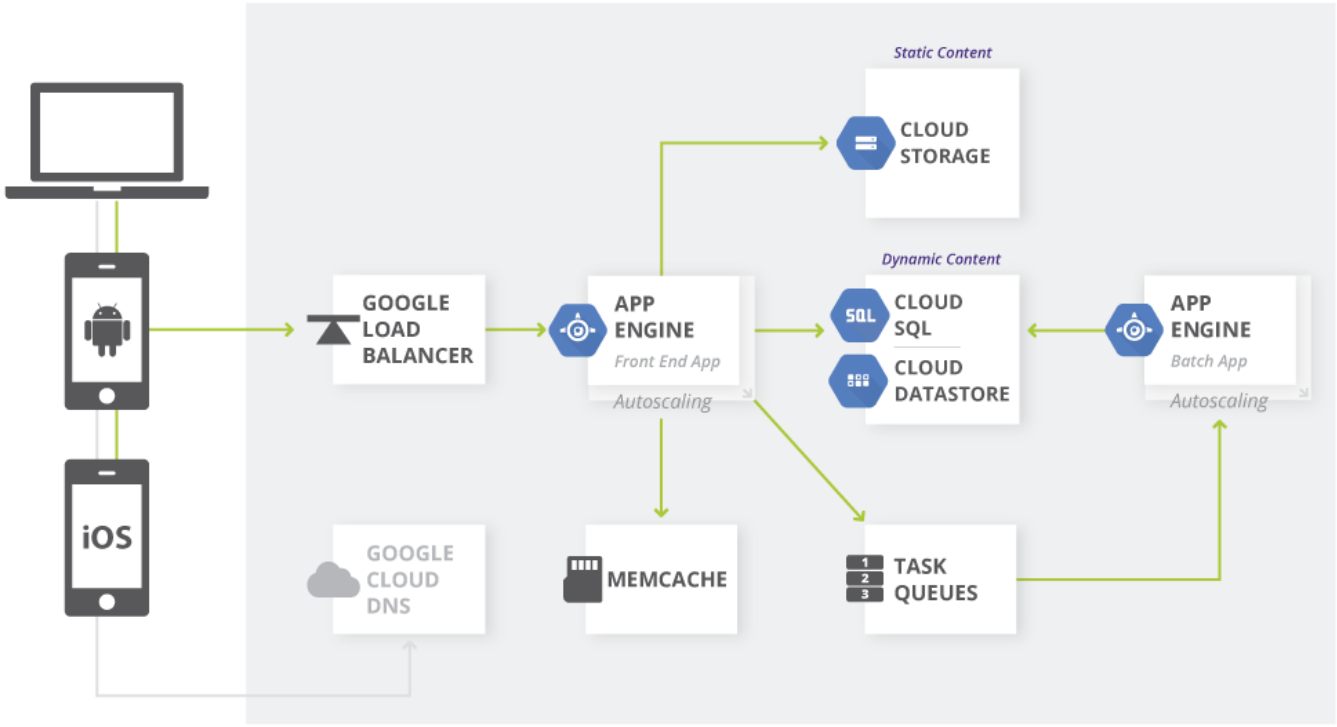
https://cloud.google.com/solutions/architecture/hadoop accessed 15 Jan 2015

Architecture: Web Application on Google App Engine

- Simple development and deployment of Web Applications with autoscaling compute power as well as the integrated features like distributed in-memory cache, task queues and datastore, to create robust applications quickly and easily.
- For applications written in Java, Python, PHP and Go.
- Supports multiple application versions which support A/B testing.
- Memcache is an in-memory cache to provide extremely high speed access to information cached by the web server (e.g. authentication or account information).
- Task Queues provide a mechanism to offload longer running tasks to backend servers, freeing the front end servers to service new user requests.
- Google Load Balancer which provides transparent load balancing to applications.
- Google's Cloud DNS is used to manage DNS domain of user.

https://cloud.google.com/solutions/architecture/webapp accessed 15 Jan 2015

Architecture: Web Application on Google App Engine



https://cloud.google.com/solutions/architecture/webapp accessed 15 Jan 2015

Google Web Tool - Kit

- GWT gives us API to design rich web applications.
- GWT is a Swing-like Java framework. Developer can write web application without writing HTML or JavaScript code.
- GWT is a development environment similar to any Web-Server-Code or Desktop-app development environment.
- GWT helps to debug, re-factor and unit test a Web-Client.
- GWT provides a so-called hosted mode, that allows developers to debug Java code, as well as a web mode which executes the GWT-generated JavaScript code.
- **Google uses GWT for its Sites:** Google Docs, Google AdSense, Google Wallet
- **Other Sites:** gogrid.com, Scenechronize, Google Moderator, Whirled. See more at http://gwtgallery.appspot.com/

GAE: Modern Web Applications

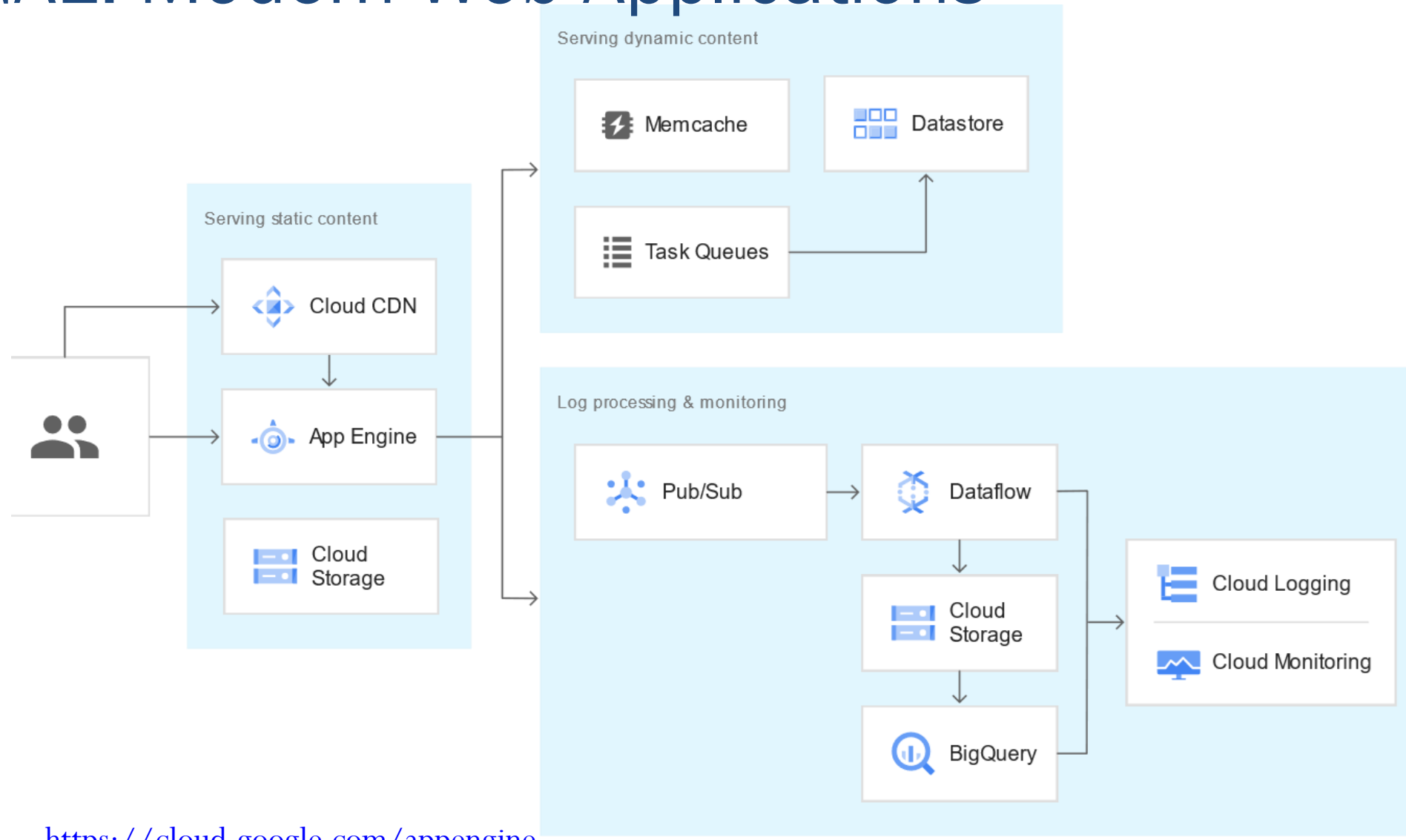
Web app using Google App Engine (GAE) and Google Cloud.

- Quickly reach customers and end users by deploying web apps on App Engine.
- Zero-config deployments and zero server management.
- Only writing code for App Engine.
- App Engine automatically scales to support sudden traffic spikes without provisioning, patching, or monitoring.

https://cloud.google.com/appengine

Please refer to Slide 2

GAE: Modern Web Applications

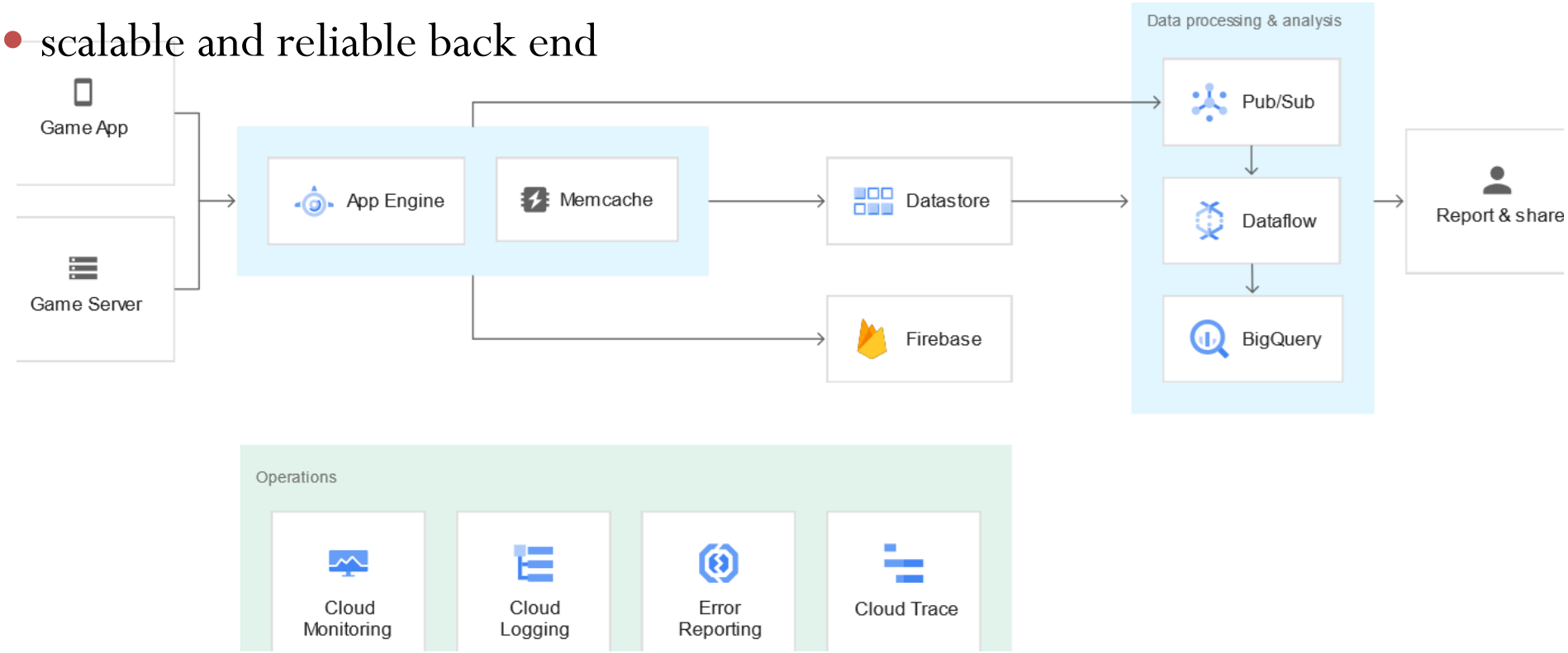


https://cloud.google.com/appengine

GAE: Scalable Mobile Back-ends

Mobile app built with Firebase and Google services.

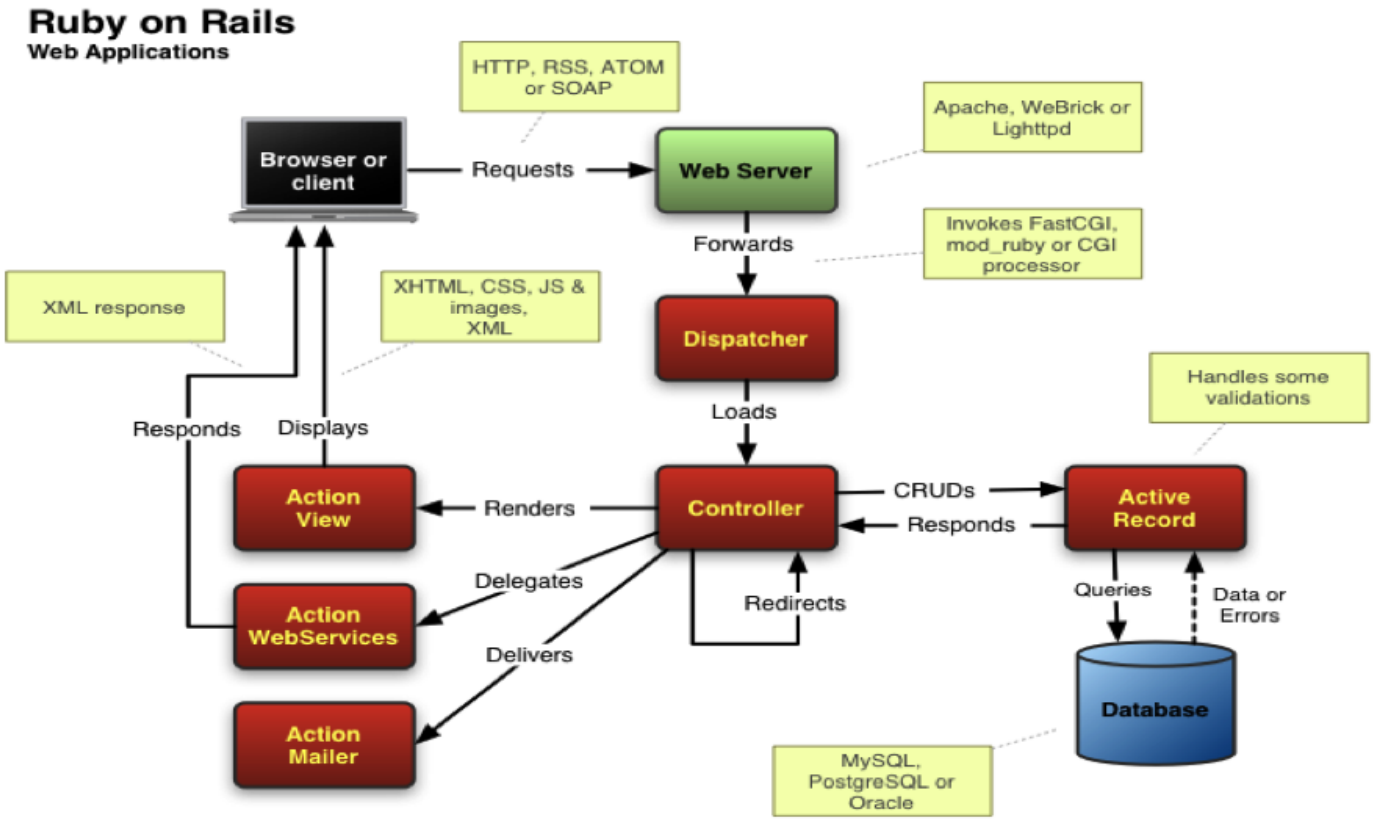
- automatically scales the hosting environment.
- an easy-to-use frontend mobile platform
- scalable and reliable back end



https://cloud.google.com/appengine

Ruby on Rails

- A web applications development framework. Very popular in the agile development with the help of convention over configuration management. Ruby on Rails uses the Model-View-Controller (MVC)



http://adrianmejia.com/blog/2011/08/11/ruby-on-rails-architectural-design/

Ruby on Rails

RESTful Architecture

- Representational State Transfer (REST) is an alternative to web services, such as SOAP and WSDL.
- Works on HTTP protocol for operations: Create, Read, Update and Delete (CRUD).
- RESTful is useful when it is required stateless, limited bandwidth (specially for mobile devices no overhead of SOAP), and when service provider and user have complete information of operations.

Ruby on Rails

- MVC separates business logic from HTML views. Architectural pattern in order to improve the maintainability of the application.
- **Model:** Carries the business logic and rules to manipulate the data.
  - Models represent the information in the database.
  - Manages interaction with database.
- **View:** Front-end of the application, representing the user interface.
  - HTML files with embedded Ruby code. Used to display data in the form of views
  - Formats, such as HTML, PDF, XML, RSS and more.
- **Controller:** it interact with models and views.
  - Incoming requests are processed by the controllers.
  - Controller process the data from the models and pass it to the views for presentation.

PaaS and Container as a Service (CaaS)

1. PaaS – Google App Engine (GAE)
2. **Container as a Service (CaaS) – DockerHub**

OS-level virtualization

OS-level virtual image looks like real computers from the point of view of programs running on

- **Containers** (LXC, Solaris containers, Docker),
- **Zones** (Solaris containers), **Partitions**,
- **Virtual private servers** (OpenVZ), **Virtual environments** (VEs), **Virtual kernels** (DragonFly BSD), or
- All resources (connected devices, files and folders, network shares, CPU power, quantifiable hardware capabilities).
- Programs running inside of a container can only see the container's contents and devices assigned to the container.
- *Container* refer to OS-level virtualization systems, e.g. Microsoft's Hyper-V containers.

https://en.m.wikipedia.org/wiki/OS-level_virtualization

Container image

- Container is a running process interacting with its own private filesystem provided by a **Container image**.
- Container image runs an application with
  - the code or binary,
  - runtime dependencies, and
  - any other filesystem objects required.
- Docker is a platform for developers and sysadmins to **build, run, and share** applications with containers.

https://docs.docker.com/get-started/

DockerHub is a CaaS

- Solomon Hykes and Sebastien Pahl
- OS-level virtualization to deliver software in packages called containers.
  - hosts the containers is called **Docker Engine**
- Docker can package an application and its dependencies in a virtual container that can run on any Linux server.
- Containers are isolated from one another and bundle their own software, libraries, and configuration files;
- All containers are run by a single operating system kernel and therefore use fewer resources than VMs.

https://en.m.wikipedia.org/wiki/Docker_(software)

Docker Computational reproducibility

Docker implement and deployment of containers, includes:

(1) cross-platform portability,

(2) modular/component re-using and sharing,

(3) Linux container (LXC) based OS level virtualization,

(4) portable across platforms, and

(5) archiving and versioning of container images.

Boettiger, Carl. "An introduction to Docker for reproducible research." *ACM SIGOPS Operating Systems Review* 49.1 (2015): 71-79.

Dependency and Coupling

- Interdependence and relationships between software modules;
  - Connected two or more routines or modules;
  - Coupling can be "low/loose" or "high/tight"
  - Low coupling provides a well-structure and good design software,
- Disadvantage of High/Tightly coupled:
  - Change in one module forces changes in other dependent modules.
  - Modularity require effort or time due to inter-module dependency.
  - Harder to reuse and test modules due to dependent modules.
- Performance reduction by message and parameter
  - request/response messages require CPU and memory in
  - message creation, transmission, translation (e.g. marshaling) and interpretation (string, array or data structure)

https://en.wikipedia.org/wiki/Coupling_(computer_programming)

Docker images: resolve Dependencies

- Docker image is based on a Linux system with Linux-compatible software including
  - R,
  - Python,
  - Matlab, and
  - most other programming environments.

- Docker image and Virtual Machine (VM) image
  - Similarity: resolves the dependency with a pre-installed and pre-configured binary image of dependencies.
  - Dissimilarities: Docker images share the Linux kernel with the host machine.

Boettiger, Carl. "An introduction to Docker for reproducible research." *ACM SIGOPS Operating Systems Review* 49.1 (2015): 71-79.

Dockerfile

- Docker can build images automatically by reading the instructions from a *Dockerfile*.
- Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.
- Docker build create an automated build that executes several command-line instructions.

  docker build **-f** /path/to/a/Dockerfile

- Copy and paste to pull this image

  docker pull hello-world

Boettiger, Carl. "An introduction to Docker for reproducible research." *ACM SIGOPS Operating Systems Review* 49.1 (2015): 71-79.
https://hub.docker.com/_/hello-world

Docker concepts

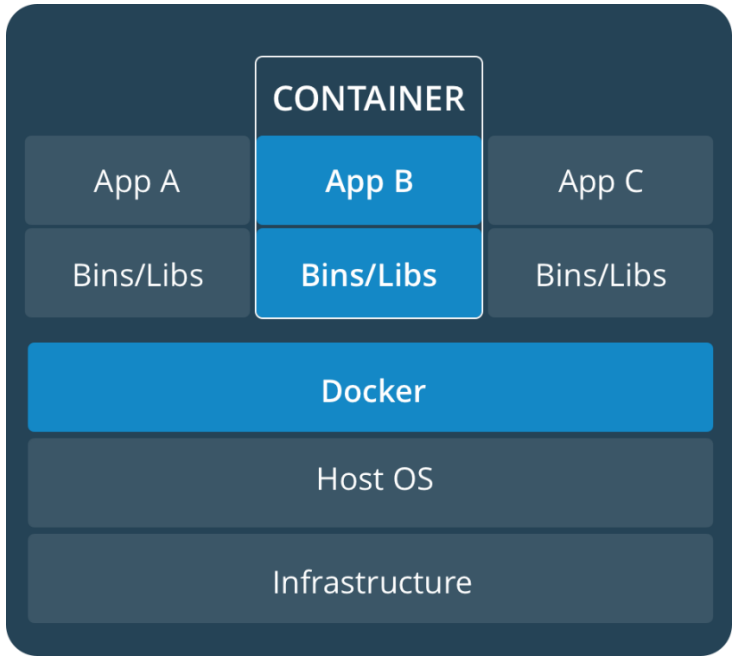
The use of containers to deploy applications is called *containerization*.

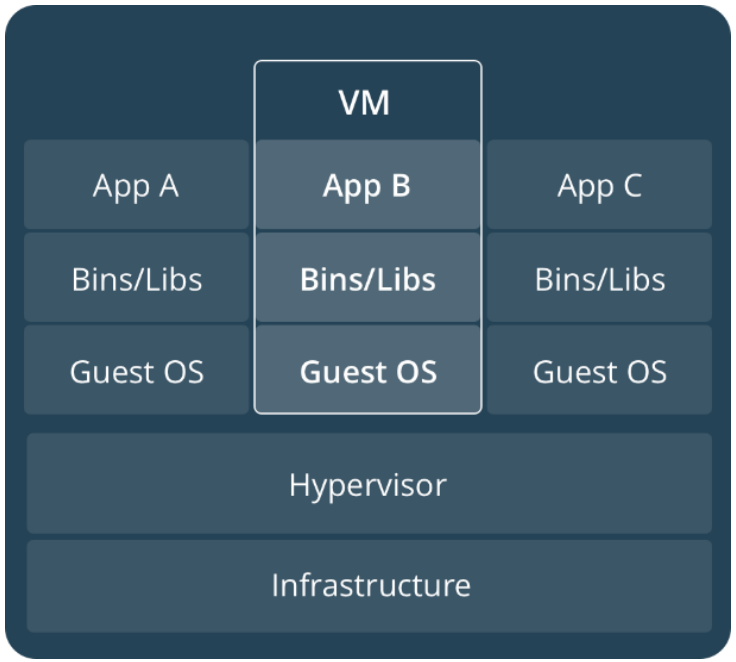
- **Flexible**: Complex applications can be containerized.
- **Lightweight**: Containers leverage and share the host kernel, making them efficient resource usage as compared to VMs.
- **Portable**: Build locally, deploy to the cloud, and run anywhere.
- **Loosely coupled**: Containers are highly self sufficient and encapsulated allow to replace or upgrade.
- **Scalable**: Distribute container replicas across a datacenter.
- **Secure**: Constraints and isolations to processes without any configuration required on the part of the user.

https://docs.docker.com/get-started/

Containers & Virtual machine (VM)

- Container runs *natively* on Linux and shares the kernel of the host machine with other containers. It runs a discrete process, taking memory like an executable.
- VM runs a full "guest OS" with hypervisor access to host resources. VM uses overhead resources that are not consumed by application logic.





https://docs.docker.com/get-started/

Build and run your Docker image

git clone https://github.com/dockersamples/node-bulletin-board

cd node-bulletin-board/bulletin-board-app

docker build **--tag** bulletinboard:1.0

docker run **--publish 8000:8080 --detach --name bb** bulletinboard:1.0

```
# Use the official image as a parent image.
FROM node:current-slim

# Set the working directory.
WORKDIR /usr/src/app

# Copy the file from your host to your current location.
COPY package.json .

# Run the command inside your image filesystem.
RUN npm install

# Add metadata to the image to describe which port the container is listening on at runtime.
EXPOSE 8080

# Run the specified command within the container.
CMD [ "npm", "start" ]

# Copy the rest of your app's source code from your host to your image filesystem.
COPY . .
```

https://docs.docker.com/get-started/part2/

Why PaaS or CaaS

Programming tools, Deployment tools, and Application hosting

- CaaS include stack of tools for deploying containers.
- CaaS do not include development tools.
- CaaS is a subset of PaaS functionality
- CaaS main use case is containerized applications
- CaaS is an incomplete form of a PaaS.
- PaaS and CaaS use-cases:
  - PaaS: integrated solution for develop and deploy applications
  - CaaS: easy to set-up and manage a container environment
- Benefit from a development at PaaS or not? Yes, then PaaS
- Pre-developed application for public deployment or not?Yes, then CaaS

https://containerjournal.com/features/paas-vs-caas-wrong-question-ask/

PaaS and CaaS Hybrid

- PaaS-CaaS hybrid provides both development environment and deployment container
  - Amazon Elastic Container Service (ECS),
  - Ruby-on-Rails (EngineYard) and
  - Google App Engine etc.

https://containerjournal.com/features/paas-vs-caas-wrong-question-ask/
https://aws.amazon.com/ecs/
https://www.engineyard.com/
https://cloud.google.com/appengine

תודה רבה
Hebrew

Ευχαριστώ
Greek

Danke
German

Спасибо
Russian

धन्यवादः
Sanskrit

Merci
French

ধন্যবাদ
Bangla

நன்றி
Tamil

شكراً
Arabic

ಧನ್ಯವಾದಗಳು
Kannada

Thank You
English

നന്ദി
Malayalam

Grazie
Italian

多謝
Traditional Chinese

ధన్యవాదాలు
Telugu

આભાર
Gujarati

ਧੰਨਵਾਦ
Punjabi

धन्यवाद
Hindi & Marathi

Gracias
Spanish

多谢
Simplified Chinese

https://sites.google.com/site/animeshchaturvedi07

Obrigado
Portuguese

ありがとうございました
Japanese

ขอบคุณ
Thai

감사합니다
Korean