UNIVERSITY of York
Europe Campus
CITY College

PaaS

Heroku

CCS3341 Cloud Computing

Dr S. Veloudis

PaaS

- Platform as a Service
- But what is a platform?
  - A platform here refers to a software framework through which software solutions can be developed
    - It provides a complete operational and development environment for a multitude of programming languages
    - It offers core software functionality for developing a cloud application which would otherwise need to be engineered from scratch
  - A cloud platform therefore serves as a launch pad for cloud software
  - PaaS offerings also include the infrastructure on which cloud applications are deployed
    - This infrastructure may be provided by a 3rd party IaaS provider

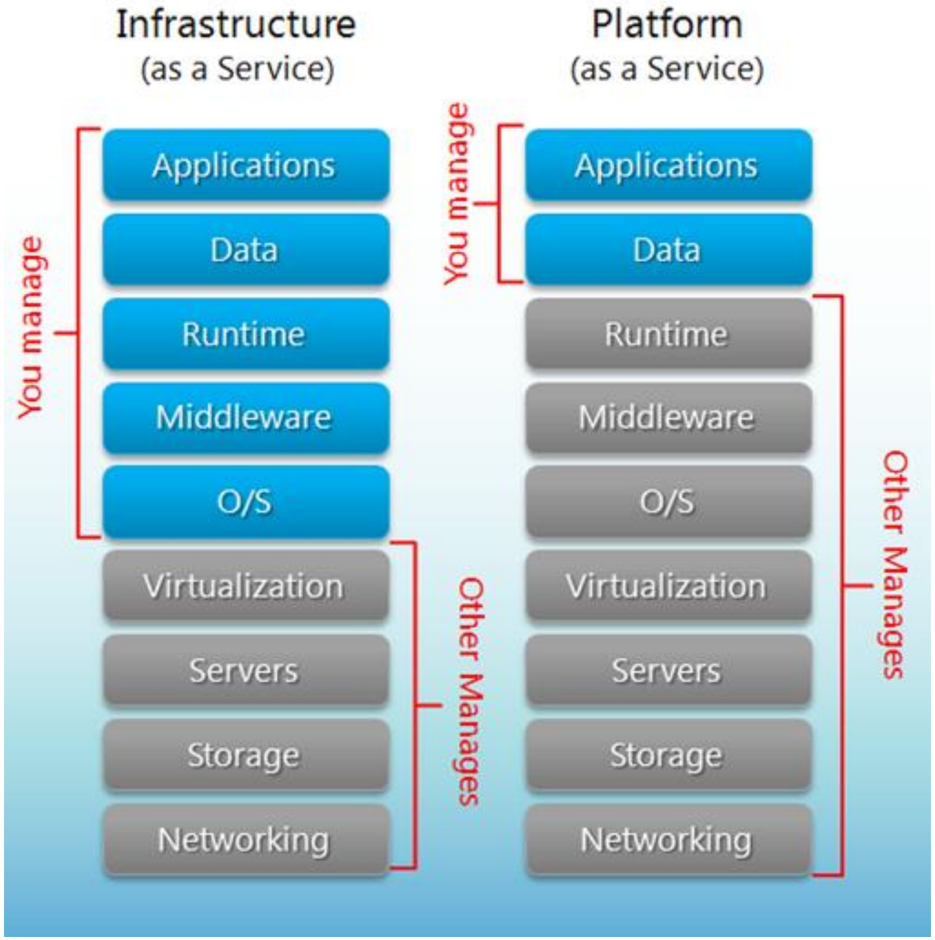
Note: A PaaS provider may at the same time be an IaaS consumer!

University of York Europe Campus
CITY College

PaaS

contd.

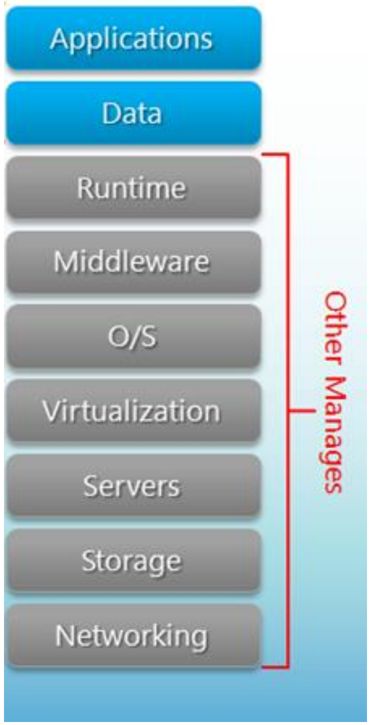
- PaaS vs IaaS



source: blogs.technet.microsoft.com

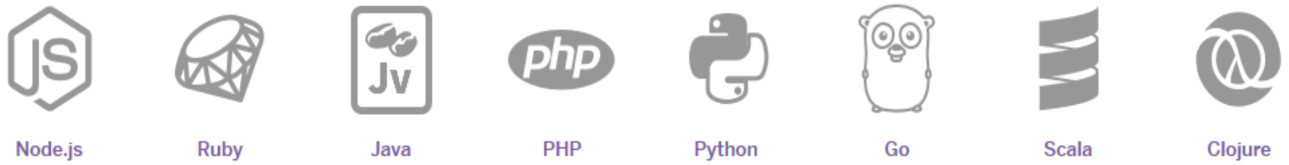
PaaS

- Cloud platforms are offered as a service
  - No need to deploy and maintain the platform on proprietary equipment
  - The PaaS user simply codes up a software solution and doesn’t worry about platform deployment details
- PaaS abstracts away interaction with the ‘bare metal’
  - Frees developers from the burden of administering the OS
- PaaS customers are charged for the infrastructure on which the PaaS-developed cloud application is deployed
  - Clearly, the more resources this infrastructure is equipped with, the higher the charges



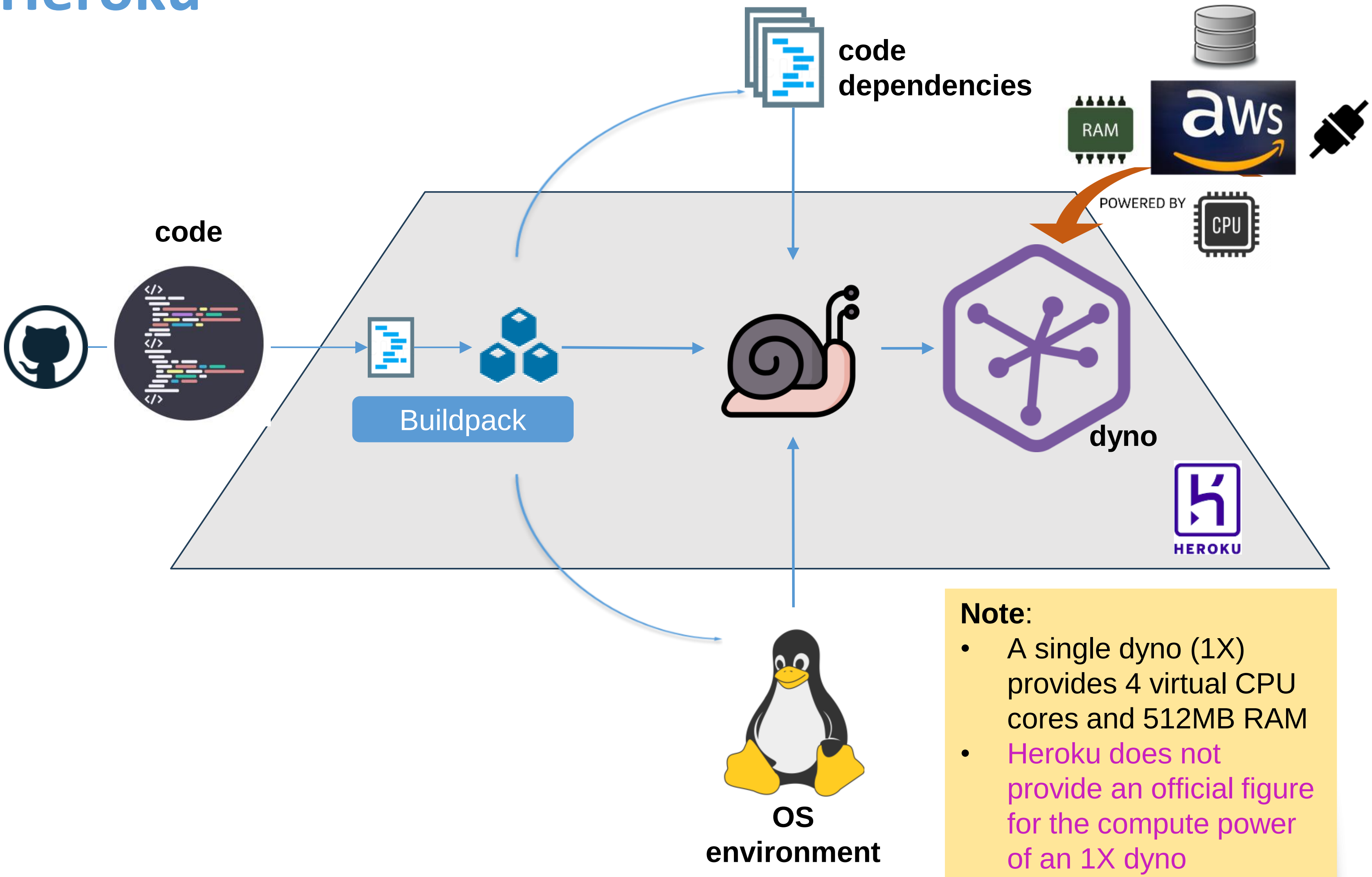
PaaS – Heroku Platform contd.

- Supports a multitude of programming languages



- Provides an operational environment in which applications developed in the supported languages can execute
- It also provides a development environment to assist developers in writing applications (see later)
- But how does it work?

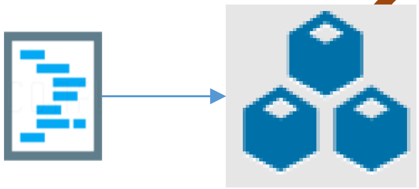
Heroku



Note:

- A single dyno (1X) provides 4 virtual CPU cores and 512MB RAM
- Heroku does not provide an official figure for the compute power of an 1X dyno

Heroku: Buildpacks



Buildpack

- Core concept on Heroku platform

Provides a simple and flexible way to package and deploy applications

- Builds the application
- Sets up appropriate **runtime environment** for the application to execute

Two kinds:

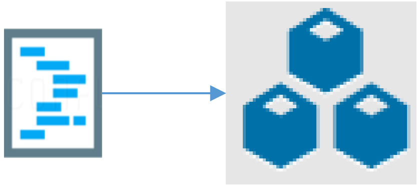
Official

- Heroku provides a collection of officially supported buildpacks for popular languages and frameworks such as Node.js, Ruby, Python, Java, etc.
- Maintained by Heroku
- Designed to be used out-of-the-box for most common use cases

3rd party

- Developers provide custom buildpacks
- Customize the build and deployment process of an application

Heroku: Buildpacks



Buildpack

By invoking an appropriate runtime (e.g., JRE and hence javac)

- Building an app

Linking

- Combining multiple object files and libraries into a single executable
- Resolves references to functions and variables defined in other parts of the code

- Linking is a dynamic process merging bytecode files (.class, .jar) **on demand** (at runtime)
- Merged files are either internal to the codebase or from external repos (declared in the POM file)

Fetched by: maven

Compilation:

Transforming a program *P* written in a language *L* to a semantically-equivalent program *P’* expressed in some object code language *L’*

In Java *L’* is called **bytecode**: a platform independent non-executable format

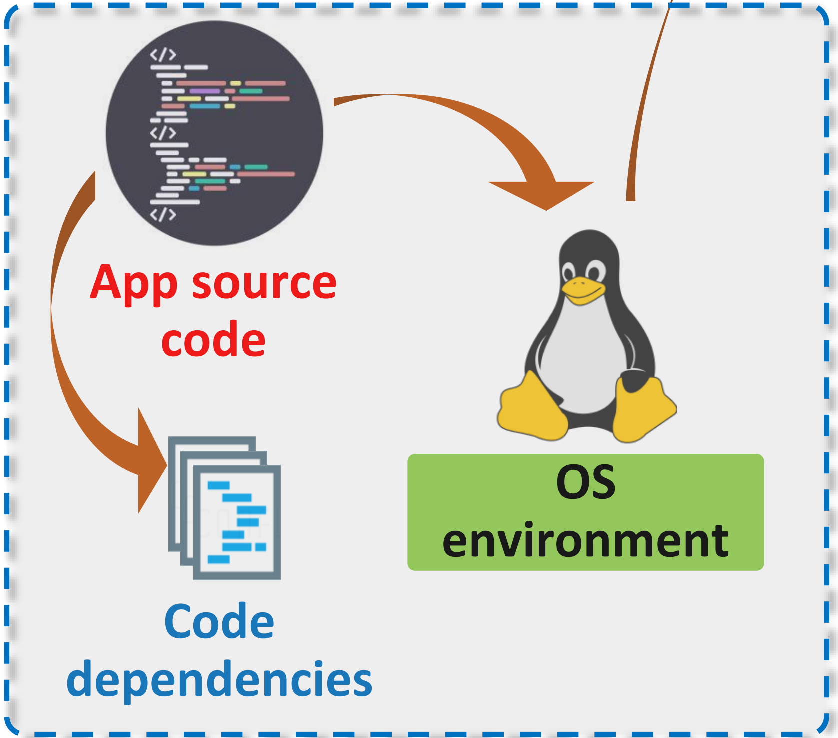
Java (and .NET) uses Just-In-Time (JIT) compilation: javac compiles source code into bytecode which is then **interpreted**

As opposed to (ahead of time, AOT) compilation, **interpretation** is performed on an instruction-by-instruction basis

AOT compilation leads to shorter execution times and JIT compilation to better portability and shorter building times

Heroku: Slugs

- Bundle comprising



A **containerised** environment where the app runs

Includes:

Language runtime (e.g., JRE)

- Compiles and/or interprets source code
- Executes source code

Lightweight Linux-based OS

- Provides necessary OS runtime to support execution

Heroku: Slugs

In-slug OS provides **drivers** for accessing disks and other devices

- In-slug OS manages **network access**
- Implements TCP/IP stack
- Opens sockets (endpoints), establishes connections, manages data transfers

Through appropriate APIs that are called from within the app

- **Interrupt handling**

- In-slug OS maintains **page tables** to map virtual memory space into physical
- In-slug OS manages only memory allocated to containing dyno
- Ensures **process isolation**

Of utmost importance for isolating different dynos

- In-slug OS implements the **upper layers** of the TCP/IP stack (http, https, app specific communication protocols)
- Lower stack layers are handled by the Heroku OS with the support of the **Heroku router**

Heroku: Router

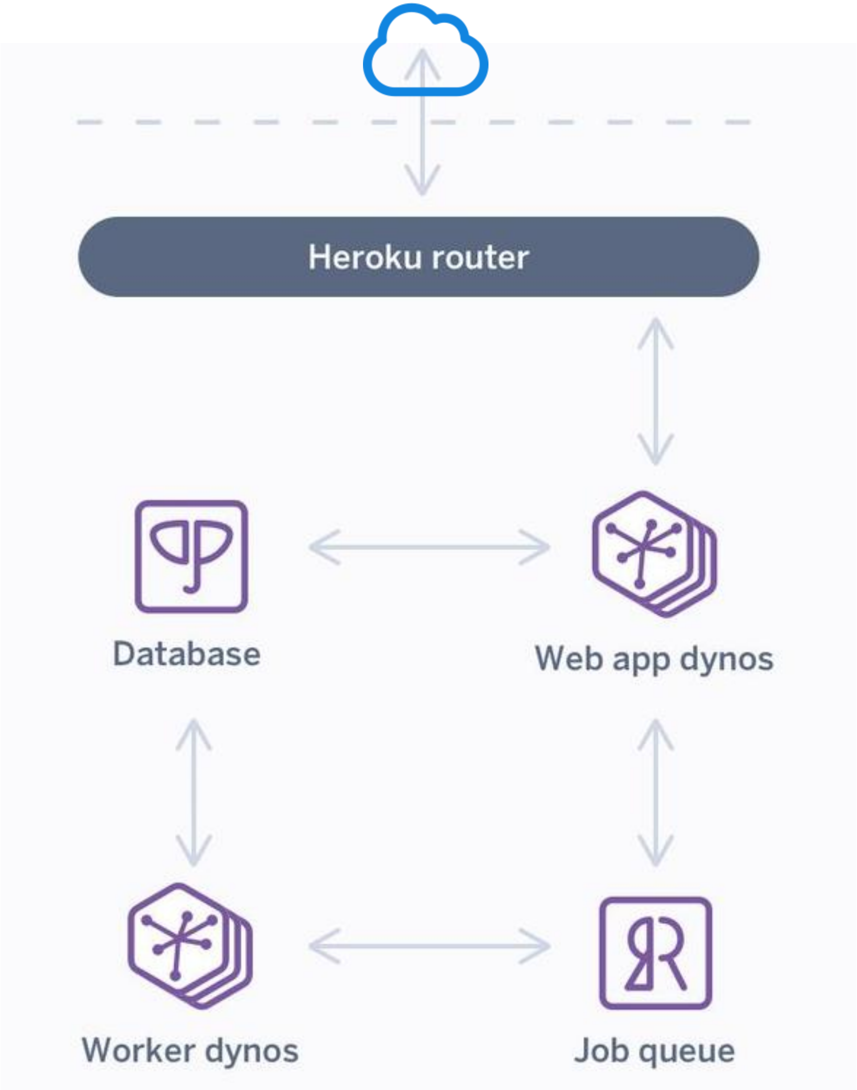
Heroku Router

Ingress traffic

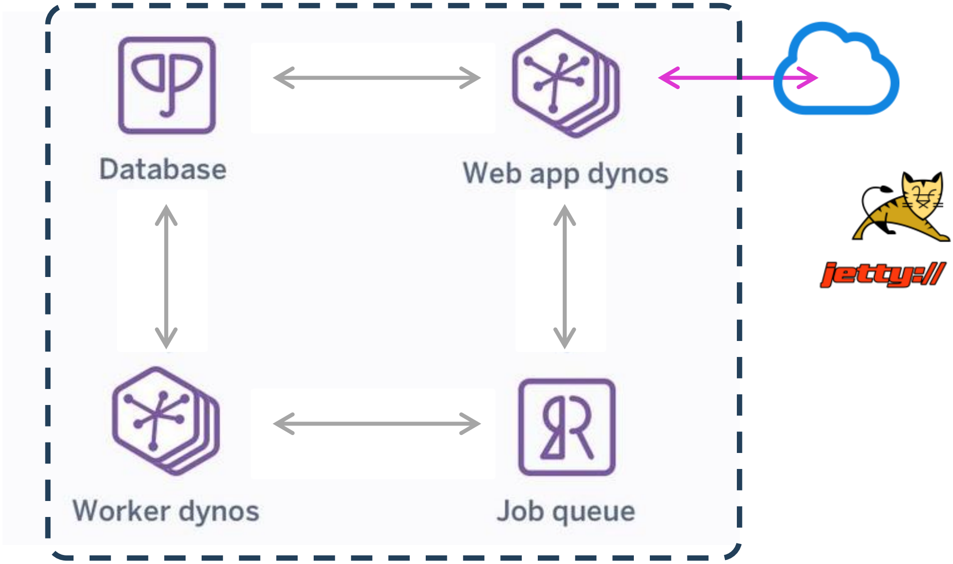
- Dynamic request routing to appropriate dyno
- Load balancing
- SSL/TLS termination (decrypting incoming traffic)
- DDoS protection
- Firewall

Egress traffic

- Routing based on IP tables
- Firewall



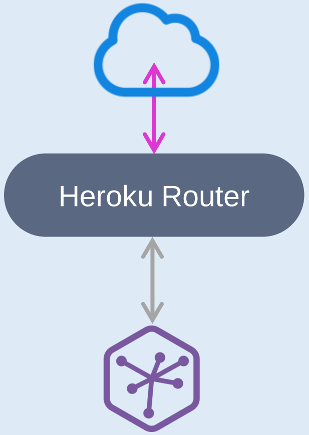
Heroku: Ingress and Egress Traffic



- Requests passed over to the upper layers of the TCP/IP stack

- Web server unmarshals the request which is now ready for processing

Ingress traffic

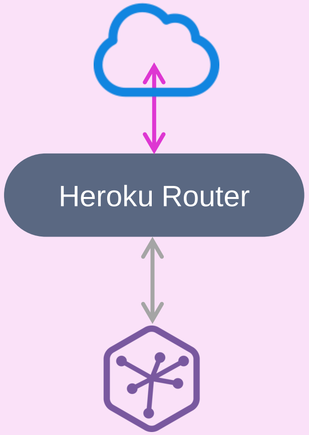


Receives incoming requests and determines the appropriate destination for each request based on routing rules and possibly load balancing

Egress traffic

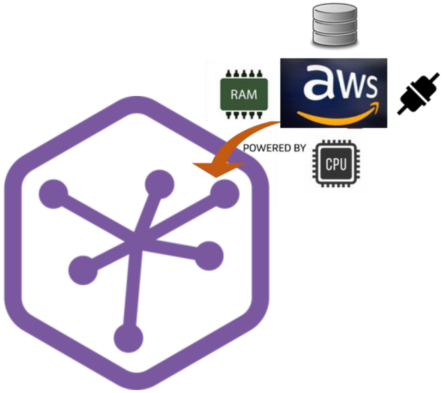
- Web server marshals response

- Upper layers of the TCP/IP stack pass over response to Heroku Router



Receives a response and routes it towards its destination using IP tables

Heroku: Dynos



An app must always comprise at least one **web dyno**

- Two kinds of dynos

Web dynos

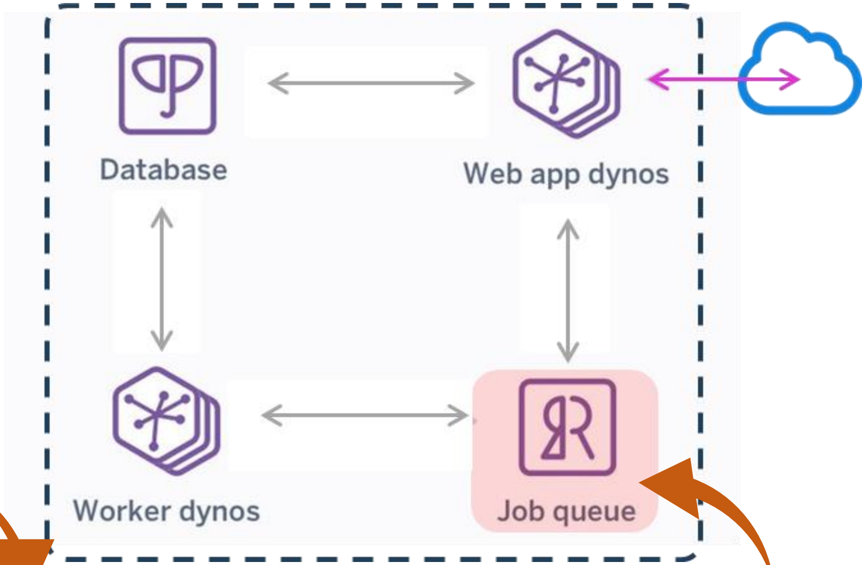
intended for serving web requests

Worker dynos

Execute background tasks

A web dyno is one whose slug includes a prepackaged webserver

Prebuilt functionality to **unmarshal** incoming requests and **marshal** responses



- No prepackaged web server
- Used for tasks that can be executed **asynchronously** and independently of incoming web requests
- Typically connect to a database

Heroku: Dynos - Procfile

A text file with commands for starting dynos

Procfiles are not necessary for simple apps (e.g., apps comprising a single web dyno)

```
web: java -Dserver.port=$PORT -Xmx512m -Dlogging.config=logback.xml -jar
target/myapp.jar
worker: java -jar target/myapp.jar –workers
scheduler: java -jar target/myapp.jar --scheduler
```

Defines three processes (dynos)

web: specifies the command to invoke a web process, on the web server at port $PORT, the amount of RAM assigned to JVM for this app, and a custom logback

- **worker:** specifies the command to invoke a worker
- **scheduler:** specifies the command to invoke a scheduler (also a worker but this time with the label “scheduler”)

By default Heroku runs 1 web dyno automatically for a newly deployed app

Heroku: Add-ons

A modular piece of software extending the functionality of an app

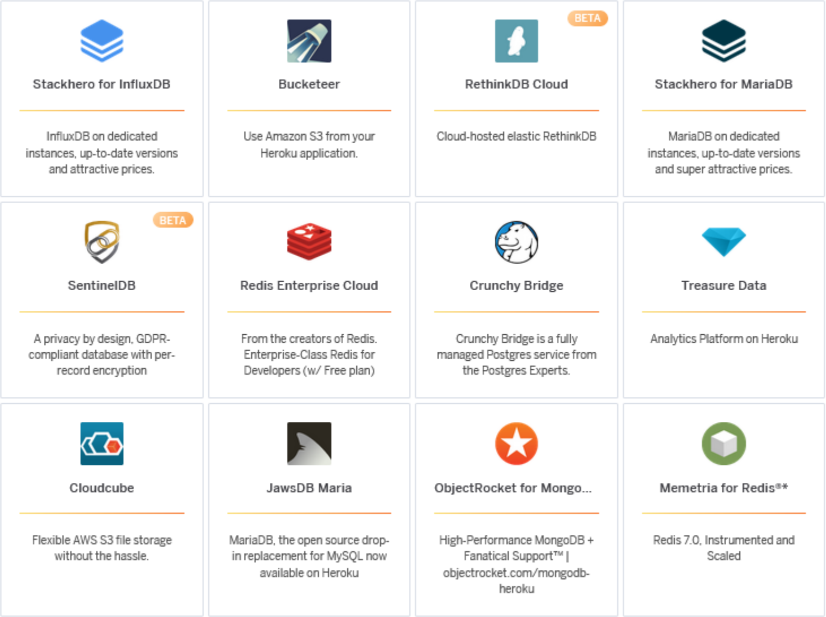
- Heroku offers a comprehensive list of add–ons
- These are either proprietary (owned by Heroku) or 3rd party

Example

- Suppose a Java app that uses a database
- A popular option is to get the database as service from Heroku
- In such a case, the database must use an addon to connect to the database

Example

- Suppose an app is deployed as a web and a worker dyno
- The two dynos interact via a message queue
- The message queue may be provided as an addon



Stackhero for InfluxDB
InfluxDB on dedicated instances, up-to-date versions and attractive prices.

Bucketeer
Use Amazon S3 from your Heroku application.

BETA
RethinkDB Cloud
Cloud-hosted elastic RethinkDB

Stackhero for MariaDB
MariaDB on dedicated instances, up-to-date versions and super attractive prices.

BETA
SentinelDB
A privacy by design, GDPR-compliant database with per-record encryption

Redis Enterprise Cloud
From the creators of Redis. Enterprise-Class Redis for Developers (w/ Free plan)

Crunchy Bridge
Crunchy Bridge is a fully managed Postgres service from the Postgres Experts.

Treasure Data
Analytics Platform on Heroku

Cloudcube
Flexible AWS S3 file storage without the hassle.

JawsDB Maria
MariaDB, the open source drop-in replacement for MySQL now available on Heroku

ObjectRocket for Mongo...
High-Performance MongoDB + Fanatical Support™ | objectrocket.com/mongodb-heroku

Memetria for Redis**
Redis 7.0, Instrumented and Scaled

and many more…

PaaS

- Benefits
  - Cost savings
    - Leveraging economies of scale allows lower infrastructure setting up, running and maintenance costs
    - The flexible pay-as-you-go model incurs lower costs as it allows usage to be monitored, measured, and billed transparently based on utilisation
  - Agility and increased productivity
    - No need to spend time on planning and purchasing servers, networking equipment, etc.
    - Easy and cost-effective to quickly setup the infrastructure required for testing up new ideas
    - Development can be greatly facilitated through the use of 3rd party services
    - Faster time to market
    - Focus on innovation rather than on code deployment, infrastructure maintenance and update issues
    - Location independence

PaaS

contd.

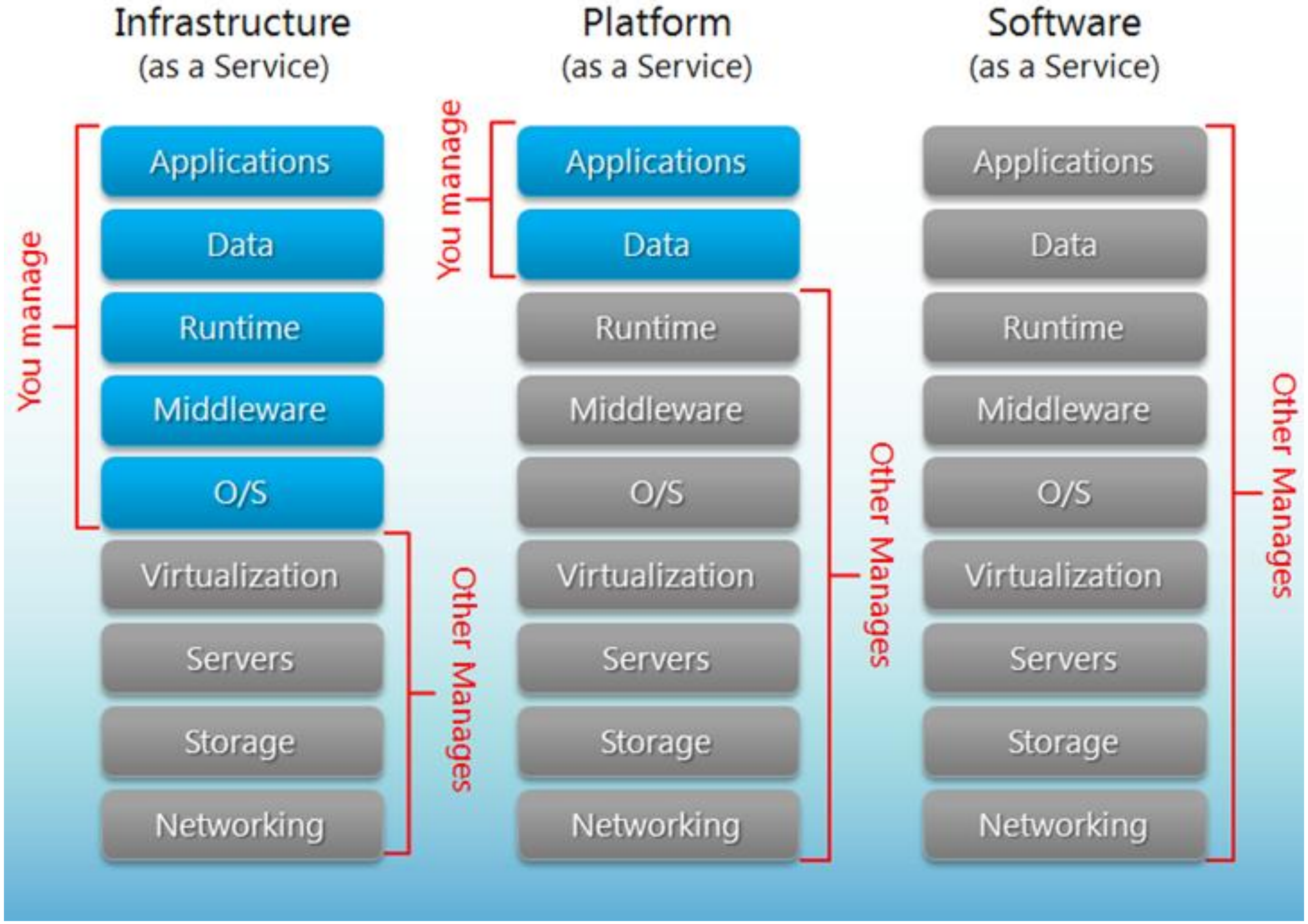
- Benefits(contd.)
  - Performance and availability
    - Scalability to cope with increased demand loads
  - Reliability
    - Provides support for disaster recovery and business continuity
  - Power-efficient
    - Multi-tenancy and virtualisation permit less power consumption
- Criticisms
  - Security
  - Lock-in effect
    - Applications developed on a particular PaaS platform are designed to be offered on that platform
    - This also depends on the amount of 3rd party services used

SaaS

- Software as a Service
- Refers to software (services and applications) that are delivered over the Internet on an on-demand basis
  - No need to go through a sales process to gain access to the software
  - The software is made available through a web browser
    - No need to install the software locally
    - Allows for the use of thin clients for accessing possibly ‘fat applications’ (i.e. hefty applications that wouldn’t be able to run on a thin client)
    - Allows for ubiquitous access
  - The software is charged on a pay-as-you-go basis
    - The customer is charged only for the parts of the service used and for as long as they are used
    - The customer is typically billed on a recurring basis
  - The software and the entire software stack on which it operates is maintained by the vendor

SaaS

- SaaS vs PaaS vs IaaS



source: blogs.technet.microsoft.com

SaaS

- SaaS applications have a much lower barrier to entry than their locally installed competitors
- Software types that lend themselves to the SaaS model include:
  - Customer relationship management (CRM)
  - Video conferencing
  - Accounting
  - Web analytics
  - Web content management
  - Collaboration software

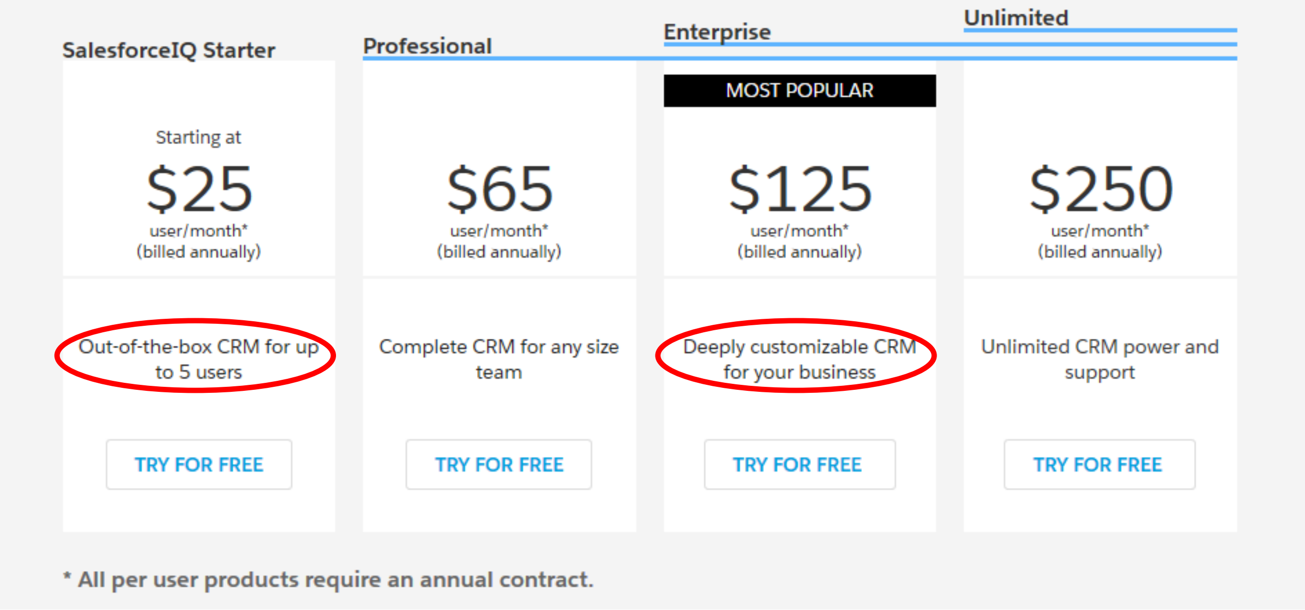
<u>Note</u>: Well known SaaS examples include Gmail, Google docs, …

- SaaS applications may be used out-of-the-box
  - No need to make extensive changes or integrate with other systems
- SaaS applications are also typically offered in easily-customisable versions
  - Provide the opportunity to clients to create added value by customising the applications in the way that suits their businesses better

SaaS

Example

Salesforce



SalesforceIQ Starter	Professional	Enterprise	Unlimited
		MOST POPULAR	
Starting at $25 user/month* (billed annually)	$65 user/month* (billed annually)	$125 user/month* (billed annually)	$250 user/month* (billed annually)
Out-of-the-box CRM for up to 5 users	Complete CRM for any size team	Deeply customizable CRM for your business	Unlimited CRM power and support
TRY FOR FREE	TRY FOR FREE	TRY FOR FREE	TRY FOR FREE

* All per user products require an annual contract.

Introducing SalesforceIQ

Discover a smart, easy CRM solution built for small businesses. SalesforceIQ is an out-of-the box application powered by Relationship Intelligence that can be set up in minutes.

salesforceIQ

SaaS

- Benefits
  - Cost savings
    - Leveraging economies of scale allows lower infrastructure setting up, running and maintenance costs
    - The flexible pay-as-you-go model incurs lower costs as it allows usage to be monitored, measured, and billed transparently based on utilisation
  - Agility and increased productivity
    - No need to spend time on planning and purchasing servers, networking equipment, etc.
    - Faster time to market
    - Location independence
    - High-degree of customisability
  - Performance and availability
    - Scalability to cope with increased demand loads
  - Reliability
    - Provides support for disaster recovery and business continuity

SaaS

- Benefits$_{\text{(contd.)}}$
  - Power-efficient
    - Multi-tenancy and virtualisation permit less power consumption
- Criticisms
  - Lock-in effect
    - Depends on the degree of customisation performed…