Cloud Computing

Renato Panda

renato.panda@ipt.pt

Once again, David is back...

There must be a better way…

Now I understand about virtualization, availability, how to scale as needed and so on…

… but I still have to **buy** and **configure** all the HW/SW required to handle the **peak load**…

…even if that happens only **once a month** or year!

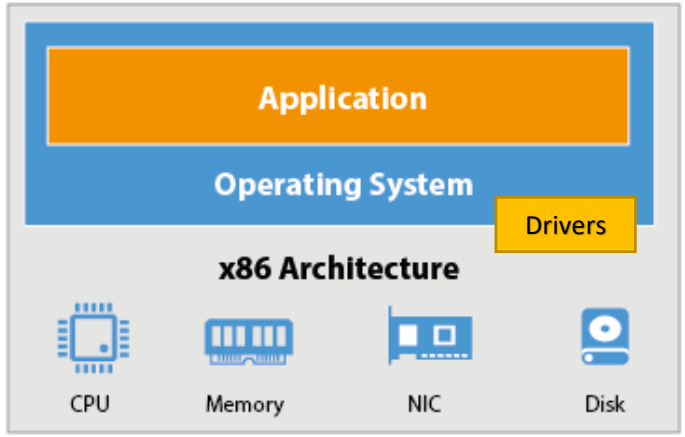

I WAS HOPING FOR A SLIGHTLY MORE DETAILED EXPLANATION OF HOW CLOUD COMPUTING WORKS THAN – "IT'S MAGIC"!
IT
© D.Fletcher for CloudTweaks.com


THANKS TO THE CLOUD ALL MY DEVICES ARE IN SYNC... IT'S ONLY MY BRAIN THAT DOESN'T KNOW WHAT'S GOING ON.


ʊ_ʊ
There is no cloud
it's just someone else's computer

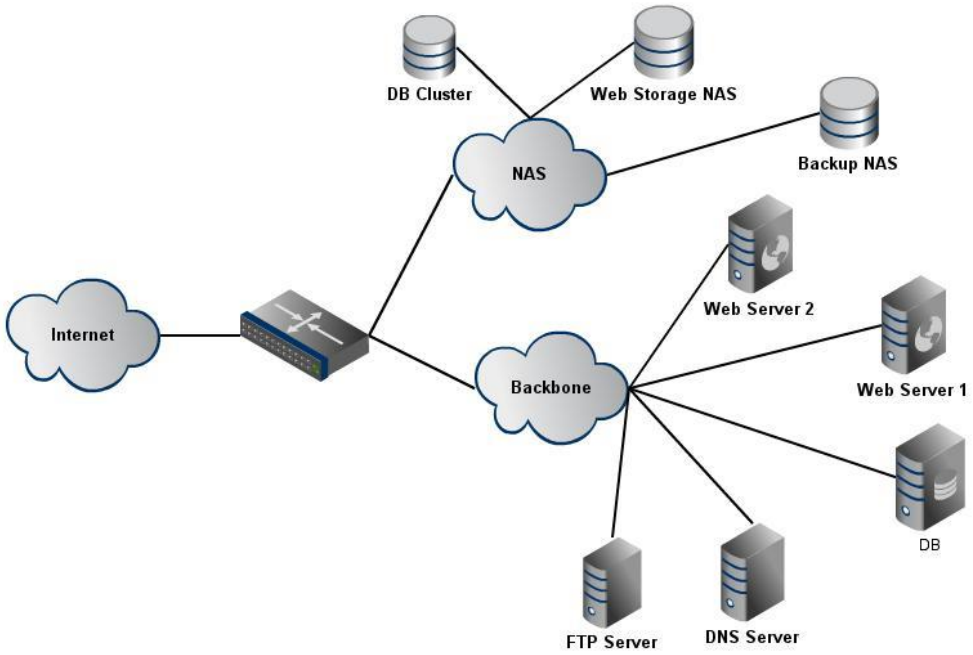
Traditional Server Architecture

- Traditionally applications run on bare-metal servers
  - OS + app stack installed to the host
    - E.g., Ubuntu, Nginx, PostgreSQL, Puma, Redis, libraries, language interpreters and tools…



What about Resources Utilization?

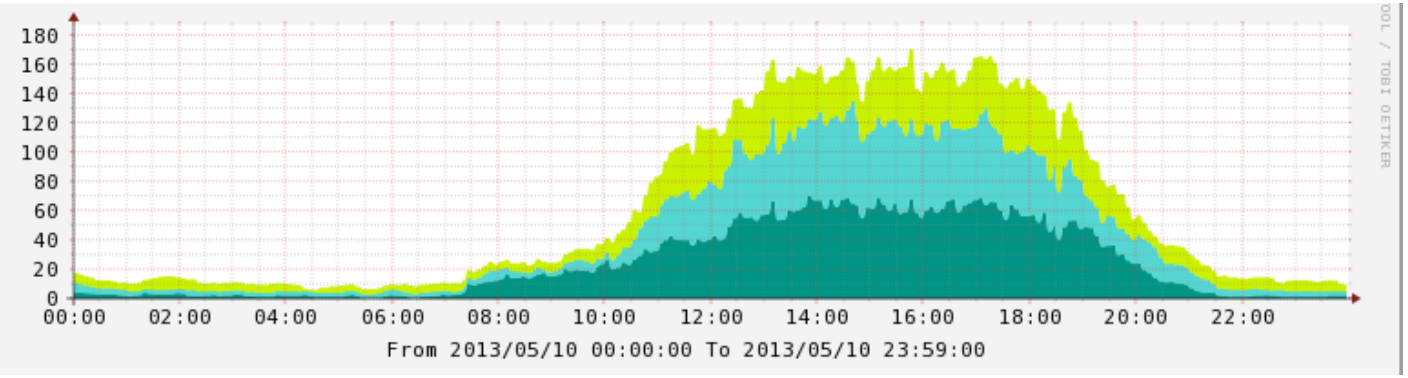
- Many software vendors require dedicated servers to run their apps
  - E.g., accounting, human resources, websites each in a different machine



What about Resources Utilization?

- A typical application
  - Does not need the full power of any modern server
  - Hardware is (should be?) planned to handle peak traffic

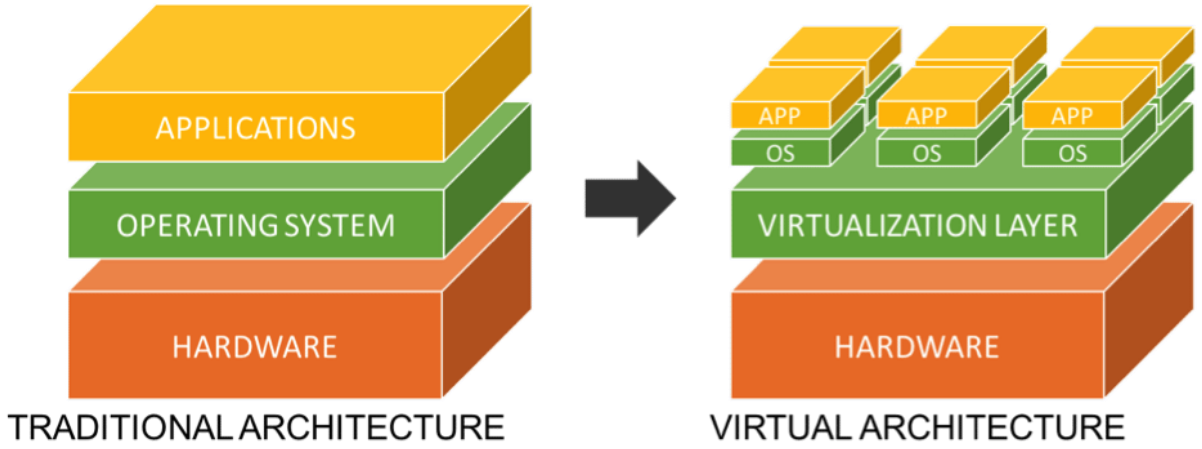
- Underutilized resources



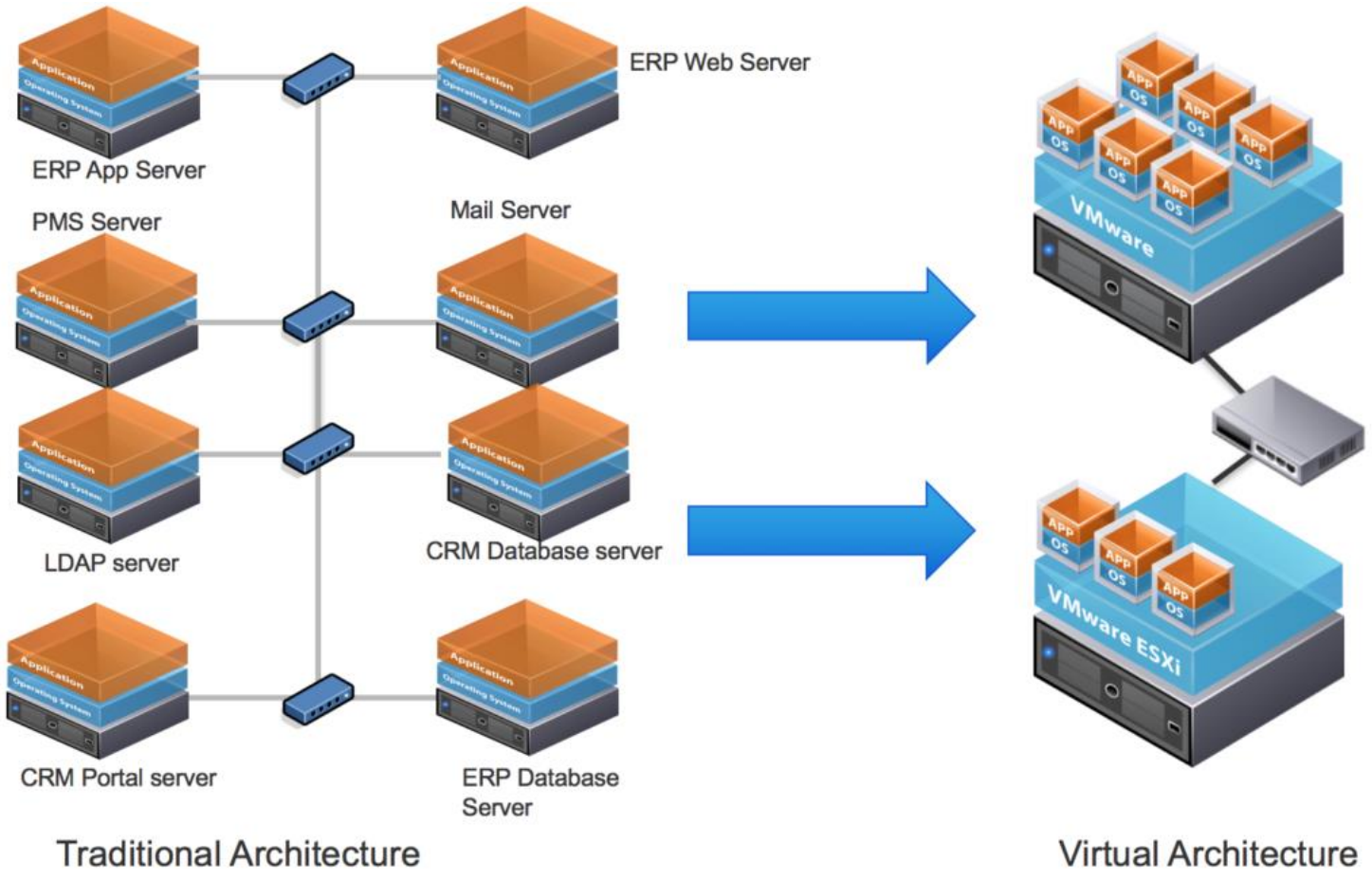
Virtualization

"In computing, **virtualization** refers to the act of **creating a virtual** (rather than actual) **version** of something, including virtual computer **hardware** platforms, **storage** devices, and computer **network** resources"

(Wikipedia)



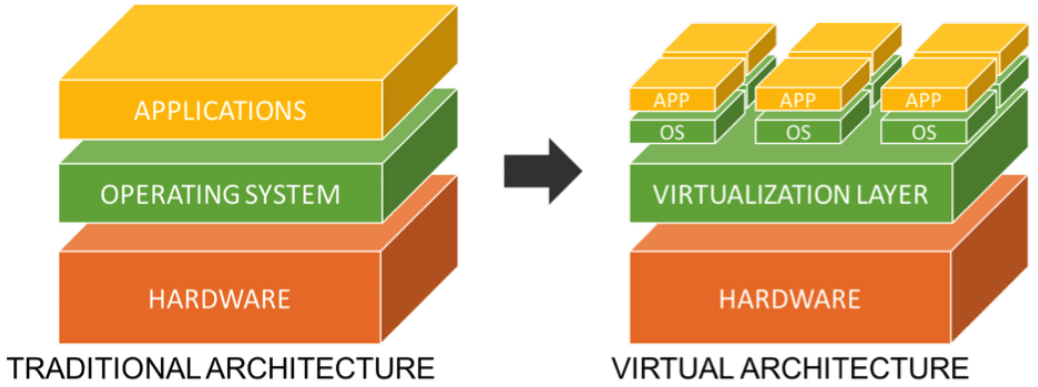
Virtualization Addresses these Issues



Servers consolidation

Virtualization Addresses these Issues

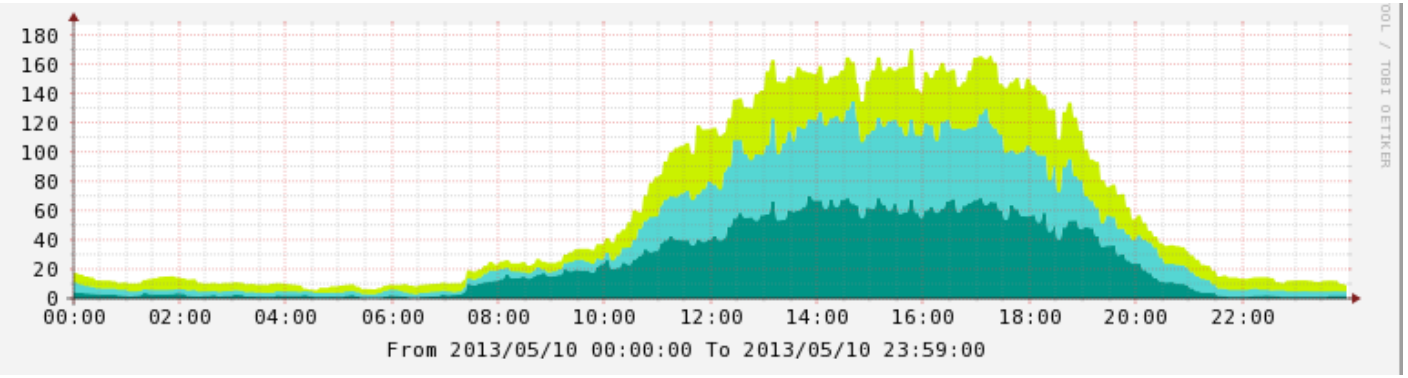
- Virtualization reduces costs
  - Maintenance
  - Power consumption
  - Rack space
- Improves several aspects
  - Disaster recovery
  - Manageability
  - Portability



Widely adopted nowadays in the industry

… but what about resources utilization?

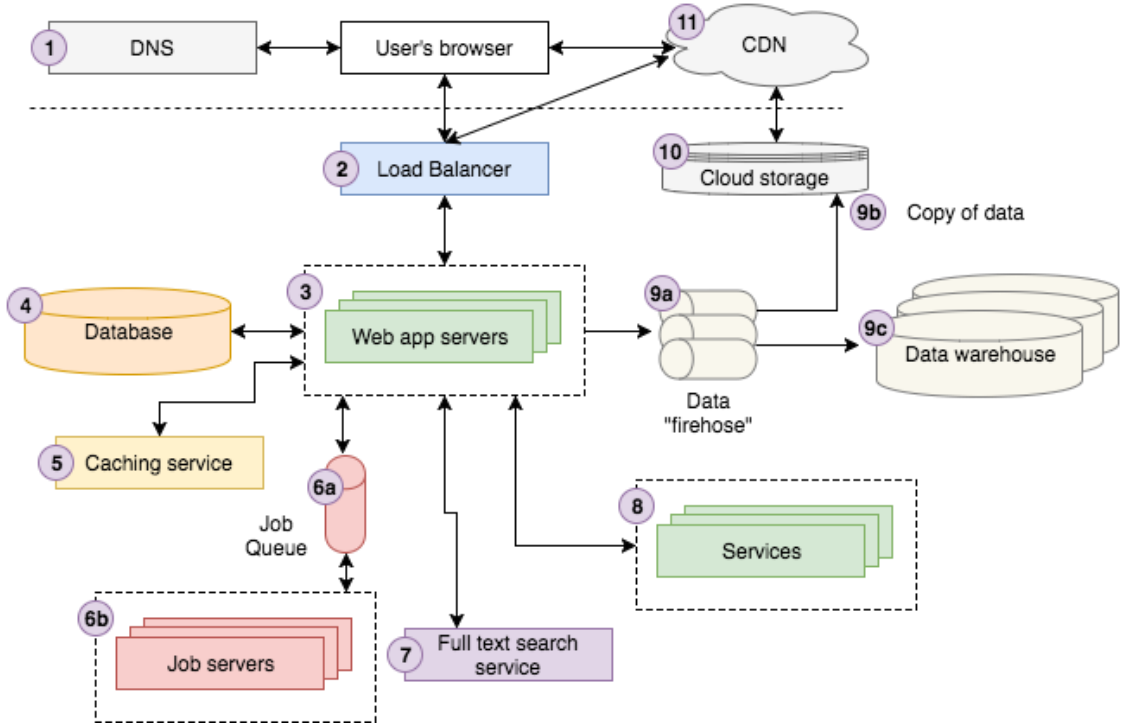
- Virtualization reduces this problem… but does not solve it.
  - We still need to have hardware to handle the peak load periods



… and about setting up and managing all this?

- Systems are increasingly complex
  - We must…
    - Start with infrastructure
    - Buy and configure all the servers
    - Maintain them all secure and patched
    - Design our marvelous product
    - Develop each part of the system
    - Test it, do backups
    - …

My startup is just me, myself and I!!



“There was a time when every household, town, farm, or village had its own water well. Today, shared public utilities give us access to clean water by simply turning on the tap; cloud computing works in a similar fashion.”

Vivek Kundra, former federal CIO, U.S. government

“Just like water from the tap in your kitchen, cloud computing services can be turned on or off quickly as needed. Like at the water company, there is a team of dedicated professionals making sure the service provided is safe, secure, and available on a 24/7 basis. When the tap isn’t on, not only are you saving water, but you aren’t paying for resources you don’t currently need.”

Vivek Kundra, former federal CIO, U.S. government

Cloud Computing

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user.

I get it! Makes sense!!

Is this only useful for specific startup cases?

Nop! Let's Check Some Real Examples…

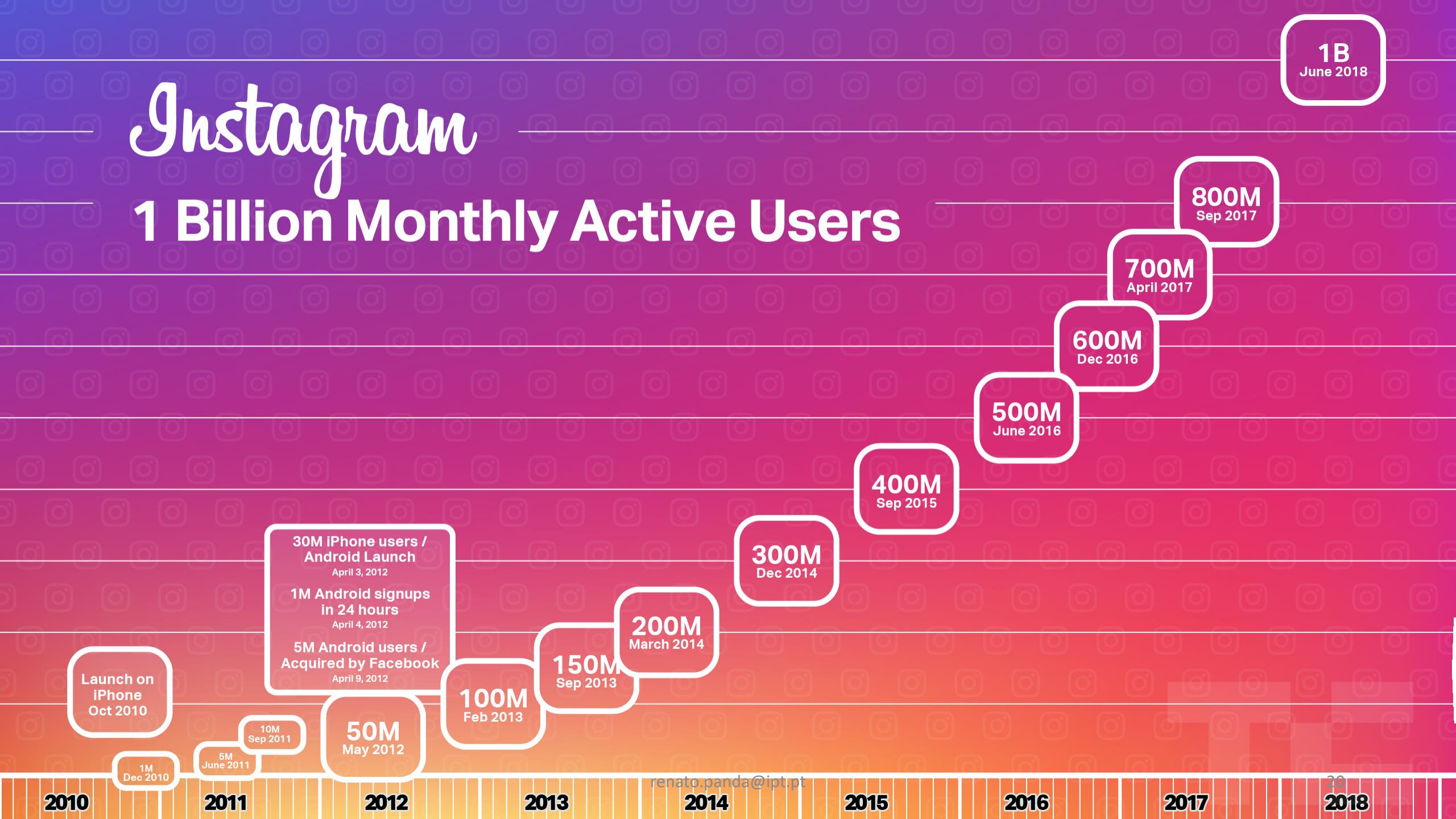
- Pick your favorite flavor
  - Start-Up Case Study: Instagram
  - Established Company Case Study: Netflix
  - Government Case Study: NOAA
  - Not-for-Profit Case Study: Obama Campaign

Case Study: Instagram

- Instagram
  - March 2010 (San Francisco)
    - Kevin Systrom and Mike Krieger got $500,000 seed funding for their check-in project, Burbn
    - Idea was too similar to Foursquare and they focused it more on photo-sharing
  - October 2010
    - Instagram iOS app officially released
    - 25.000 users registered on that first day
  - December 2010
    - Instagram reaches 1 million users

Case Study: Instagram

- Instagram
  - September 2011
    - Instagram reaches 10 million users
    - Still only available to iOS
  - April 2012
    - Hits 30 million users
    - Launches Android version
    - 1 million users registers in 24 hours
    - Acquired by Facebook (US$1 billion)


Instagram
1 Billion Monthly Active Users
1B
June 2018
800M
Sep 2017
700M
April 2017
600M
Dec 2016
500M
June 2016
400M
Sep 2015
300M
Dec 2014
200M
March 2014
150M
Sep 2013
100M
Feb 2013
50M
May 2012
30M iPhone users / Android Launch
April 3, 2012
1M Android signups in 24 hours
April 4, 2012
5M Android users / Acquired by Facebook
April 9, 2012
10M
Sep 2011
5M
June 2011
1M
Dec 2010
Launch on iPhone
Oct 2010
2010
2011
2012
2013
2014
2015
2016
2017
2018

Case Study: Instagram

- Three guys were able to build a billion-dollar product
  - On a bootstrap budget (considering the San Francisco reality, not PT ☺)
  - Entirely on a public cloud

- Would this be possible with a physical, on-premises, datacenter?
  - Could they buy, setup and maintain hardware fast enough to keep up with this growth?

Case Study: Netflix

- What is Netflix?
  - Industry leader in streaming video content over the Internet
- What about its history?
  - 1997 - Reed Hastings and Marc Randolph co-found Netflix to offer online movie rentals
  - 1998 - The first DVD rental and sales site, netflix.com, is launched
  - 1999 - Subscription service, offering unlimited DVD rentals for one low monthly price
  - 2000 - Personalized movie recommendation system is introduced
    - Uses members' ratings to accurately predict choices for all members

Case Study: Netflix

"While you have been listening to me talk, the Netflix service has gone live in nearly every country of the world”

Netflix CEO Reed Hastings, January, 2016
@ Consumer Electronics Show, Las Vegas

- What about its history?
  - 2006 - Netflix grows to 5 million members
  - 2007 - Netflix **introduces streaming**
  - 2016 - Netflix **expands to 130 countries**

"I think you can imagine the amount of work and thinking and architecture design we had to do to open up to 130 countries, and millions of new customers just in that moment; the technical architecture, the research, the billing systems, the kind of people that we needed and the thinking about these kinds of problems in order to make that happen.“

Dave Hahn, Senior Engineer
Netflix's cloud operations and reliability team

Case Study: Netflix

- How was this achieved?
  - 2008 - Netflix was running relational databases in its own data centers (DC)
    - A DC failure shut the entire service down and stopped DVD shipments for three days
    - There were two paths
      A. Turn Netflix into a world-class data center operations company
      B. Move the service to the public cloud (and **focus on the core competencies**)
  - Things to consider
    - Netflix was growing fast
    - Serving millions of customers was already generating an enormous quantity of data
    - It was very challenging to predict traffic
    - Struggle to rack their own servers fast enough to handle the ever-growing volumes

Case Study: Netflix

- How was this achieved?
  - 2008 - Netflix was running relational databases in its own data centers (DC)
    - A DC failure shut the entire service down and stopped DVD shipments for three days
    - There were two paths
      A. Turn Netflix into a world-class data center operations company
      B. Move the service to the public cloud (and **focus on the core competencies**)
  - Advantages of B
    - Netflix engineers could focus on building and improving the business applications
    - Cloud provider would focus on infrastructure
      - Allowing Netflix to add thousands of virtual servers and petabytes of storage within minutes
    - Auto-scaling capabilities would handle traffic peaks and cut costs

Case Study: Netflix

- A migration to the cloud was the clear choice
  - Netflix became a poster child customer for Amazon Web Services (AWS)

- How to migrate their business to AWS?
  A. Forklift all of their monolithic enterprise systems out of the data center and drop them into AWS
  B. Rebuild the Netflix technology in AWS and fundamentally change the way that the company operated

Case Study: Netflix

- A migration to the cloud was the clear choice
  - Netflix became a poster child customer for Amazon Web Services (AWS)

- Consequences of each option
  A. Fast, but would bring all of the old data center problems to the cloud
  B. Slow and complex, but would unleash the full power of a cloud solution

Case Study: Netflix

- Migrate to AWS meant a move to microservices
  - The service was divided into multiple microservices
  - Each managed by their own small team
    - Each understood how their service worked and interacted with other systems

"Any Netflix service team can deploy their service at any time. It requires no coordination, no scheduling, no crucible to get to production.“

Dave Hahn, Senior Engineer
Netflix's cloud operations and reliability team

Case Study: Netflix

- Migrate to AWS meant a move to microservices
  - The service was divided into multiple microservices
  - Each managed by their own small team
    - Each understood how their service worked and interacted with other systems

- The entire process took seven years to complete
  - 2009 - 100% of all customer traffic was run through Netflix's own data center
    - By 2010 many of that traffic was already going into AWS
    - Still, several other operational services persisted
  - 2016 - the last remaining data centers used by the streaming service were shut down

Case Study: Netflix

- Benefits of the migration
  - Improved scalability
  - Improved the velocity by which the company could release new content
    - Also new features, interfaces and interactions
  - Freed up the capacity of engineers
  - Cut the costs of streaming
  - Drastically improved availability

The company is able to scale at amazing levels while controlling costs and reducing the risks of downtime

Case Study: Netflix

- Want to know more?

Netflix uses AWS for nearly all its computing and storage needs, including databases, analytics, recommendation engines, video transcoding, and more—hundreds of functions that in total use more than **100,000 server instances** on AWS.

About **1,000 Amazon Kinesis shards** work in parallel to process streams, "**multiple terabytes of log data** each day, yet events show up in our analytics in seconds,"

John Bennett, senior software engineer at Netflix.

Netflix Gains New Efficiencies Using AWS Lambda

How Netflix Uses AWS Lambda to Build **Rule-Based, Self-Managing Infrastructure** to Increase Efficiency

Netflix Realizes Multi-Region Resiliency Using Amazon Route 53, allowing the team to **evacuate an entire region in less than 40 minutes**.

Case Study: NOAA

- National Oceanic and Atmospheric Administration (NOAA)
  - American federal scientific agency with over 25,000 employees
- NOAA mission
  - Understand and predict change in climate, weather, oceans, and coasts
- NOAA employees
  - Work in all conditions such as in the air, on land, and on sea
  - Rely heavily on Internet-connected devices and collaboration with team members and other agencies
    - E-mail and other collaboration (video conference, messaging, share documents)

Case Study: NOAA

- In 2012 NOAA adopted a cloud-based collaboration solution
  - G Suite by Google Cloud was adopted, includes among which
    - Google services such as Gmail (@noaa), drive, cloud search, calendar, docs, hangouts
    - Administration tools, endpoint security management, vault
    - Work statistics and insights

Solution adopted by several other big names

Case Study: NOAA

- Benefits of migrating to the cloud solution
  - Costs cut in half
  - Removed the burden of managing software and hardware updates
    - Highly distributed and device-heavy environment
- NOAA's staff feedback
  - E-mail and other tools are faster
  - Easier to setup than the on-premises solutions
  - Services themselves were more modern
  - Better overall service at half the price with less effort

Case Study: Obama Campaign

- Obama campaign's technical team faced a very special set of requirements
  - Build a suit of applications, including an e-commerce fund-raising platform
    - Scalable, able to handle over US$1 billion dollars
    - That would run only for six months
    - Expecting a huge spike in the last days
    - After that, back everything up and go away

In other words, six-month shelf-life with one big peak

Case Study: Obama Campaign

- Obama Campaign relied heavily on cloud solutions
  - All service models were used (SaaS, PaaS and IaaS)
    - This will be explained later
    - Several reasons supported these decisions (case by case)
      - Such as lower costs, speed to market, on-demand resources, and scalability
  - E.g., their phone-calling application scaled to 7,000 concurrent users as it peaked on Election Day

Case Study: Obama Campaign

- What about costs?
  - Roughly US$1.5 million was spent
    - Over $1 million of that (66%+!!) was for an on-premises hosting company that managed some of the social media and digital advertising
    - The remaining services (200+ apps) cost less than US$500,000 of cloud infrastructure and services

Source: Architecting the Cloud - Design Decisions for Cloud Computing Service Models book (p40)

Case Studies Summary

Enterprises are **shifting dollars** away from **commercial software** licenses and **hardware** investments **in favor** of a variety of **cloud services** across all three service models.

The secret to success for enterprises will be **picking the right cloud solutions to solve the right business problems**. Understanding the three cloud service models—SaaS, PaaS, and IaaS—is crucial for enterprises to make the right investments in the cloud.

Case Studies Summary

- Want to know more AWS case studies?

By using AWS, Coursera can handle half a petabyte of traffic each month.

Mapbox can collect 100 million miles of telemetry data every day using AWS.

Lyft is going all in on AWS to deliver better customer experiences and focus on innovation.

FINRA collects and analyzes billions of brokerage transactions daily using AWS.

Airbnb benefits from the scalability, agility, and reliability provided by AWS.

Cloud Computing

- Most of you already know a few things about cloud computing
  - **On-demand availability** of computer system resources without direct active management by the user
    - Especially data storage and computing power, but can even be ready-to-use apps
  - May be limited to a single organization or be available to many
    - Enterprise (**private**) vs **public** clouds
  - Relies on **sharing of resources** to achieve coherence and **economies of scale**
    - Allows companies to avoid or **minimize** up-front IT infrastructure **costs**
  - Cloud providers typically use a "**pay-as-you-go**" model
    - Can lead to unexpected operating expenses if used incorrectly
  - Possible due to the several technological advances
    - Availability of high-capacity networks, low-cost computers and storage devices
    - Hardware virtualization, service-oriented architecture and autonomic and utility computing

How did it start?

- A brief journey into the past
  - 1999 – Salesforce is founded as a software as a service (SaaS) company
    - Provides a customer-relationship management (CRM) service
  - 2002 – Amazon launches Amazon Web Services (AWS)
    - In its early stages consisted of only a few disparate tools and services
    - In **2006** AWS introduced its **Elastic Compute Cloud (EC2)**
  - 2008 – Google released the beta version of Google App Engine
    - Platform as a Service, initially only supporting Python
  - 2008 – NASA's OpenNebula
    - First open-source software for deploying private and hybrid clouds (+ federation)
  - 2002-2010+ - Cloud-based applications explode
    - Facebook (2004), Twitter (2006), DropBox (2008), Google Docs (2009), iCloud (2011), …
  - 2010 – Microsoft launches Microsoft Azure
  - 2010 – Rackspace Hosting + NASA launch OpenStack (OSS based on Nebula platform)
  - 2011 – IBM launches SmartCloud Enterprise (IaaS)
  - 2012 – Oracle Cloud, first to provide IaaS, PaaS and SaaS solutions
  - 2012 – Google Compute Engine

The NIST Definition of Cloud Computing (ref)

(National Institute for Standards and Technology)

"Cloud computing is a model for enabling convenient, ***on-demand*** *network access to a* ***shared pool*** *of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be* ***rapidly provisioned*** *and released with* ***minimal management*** *effort or service provider interaction"*

Image source

The NIST Definition of Cloud Computing (ref)

(National Institute for Standards and Technology)

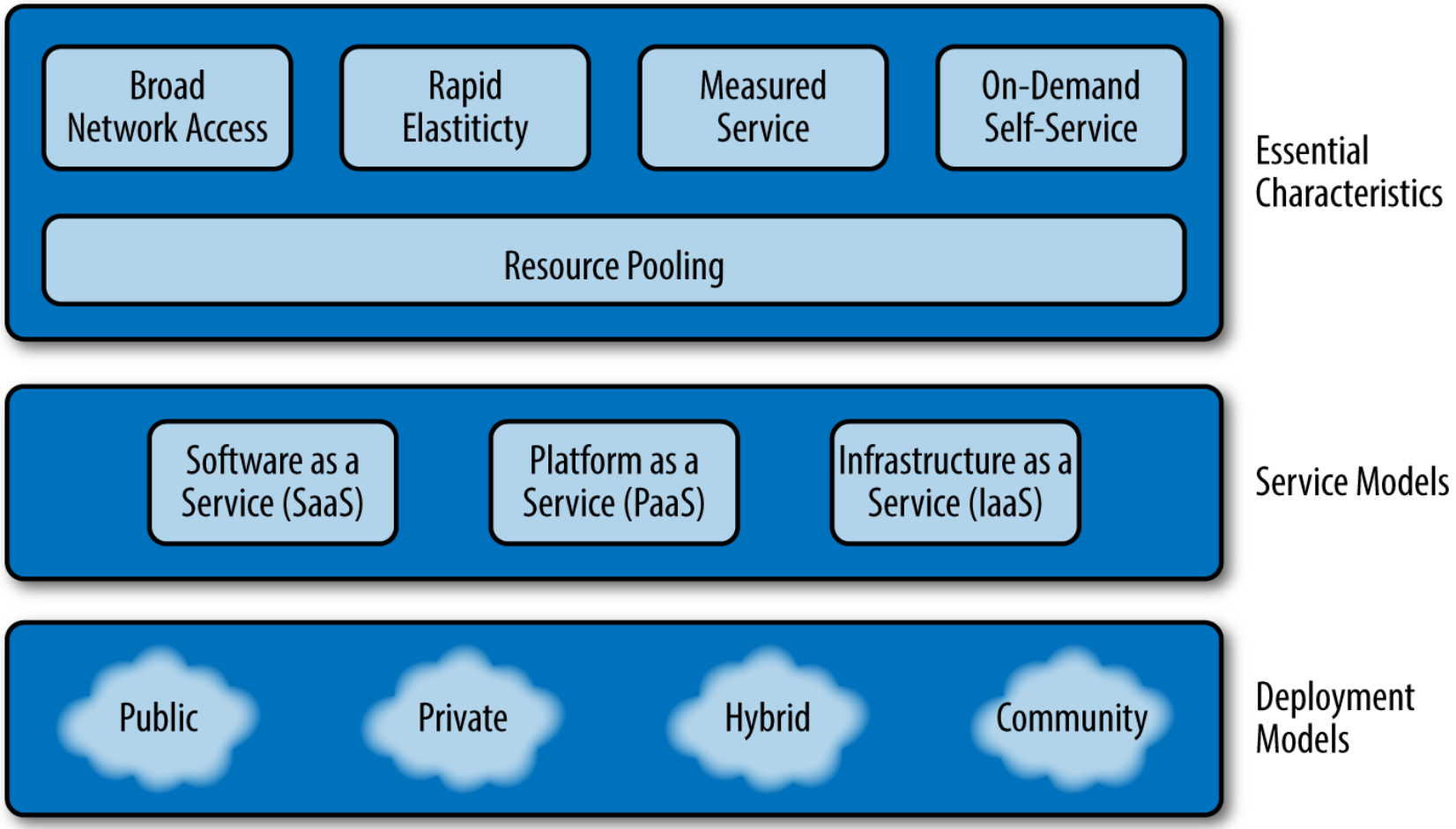


Image source

The NIST Definition of Cloud Computing (ref)

(National Institute for Standards and Technology)

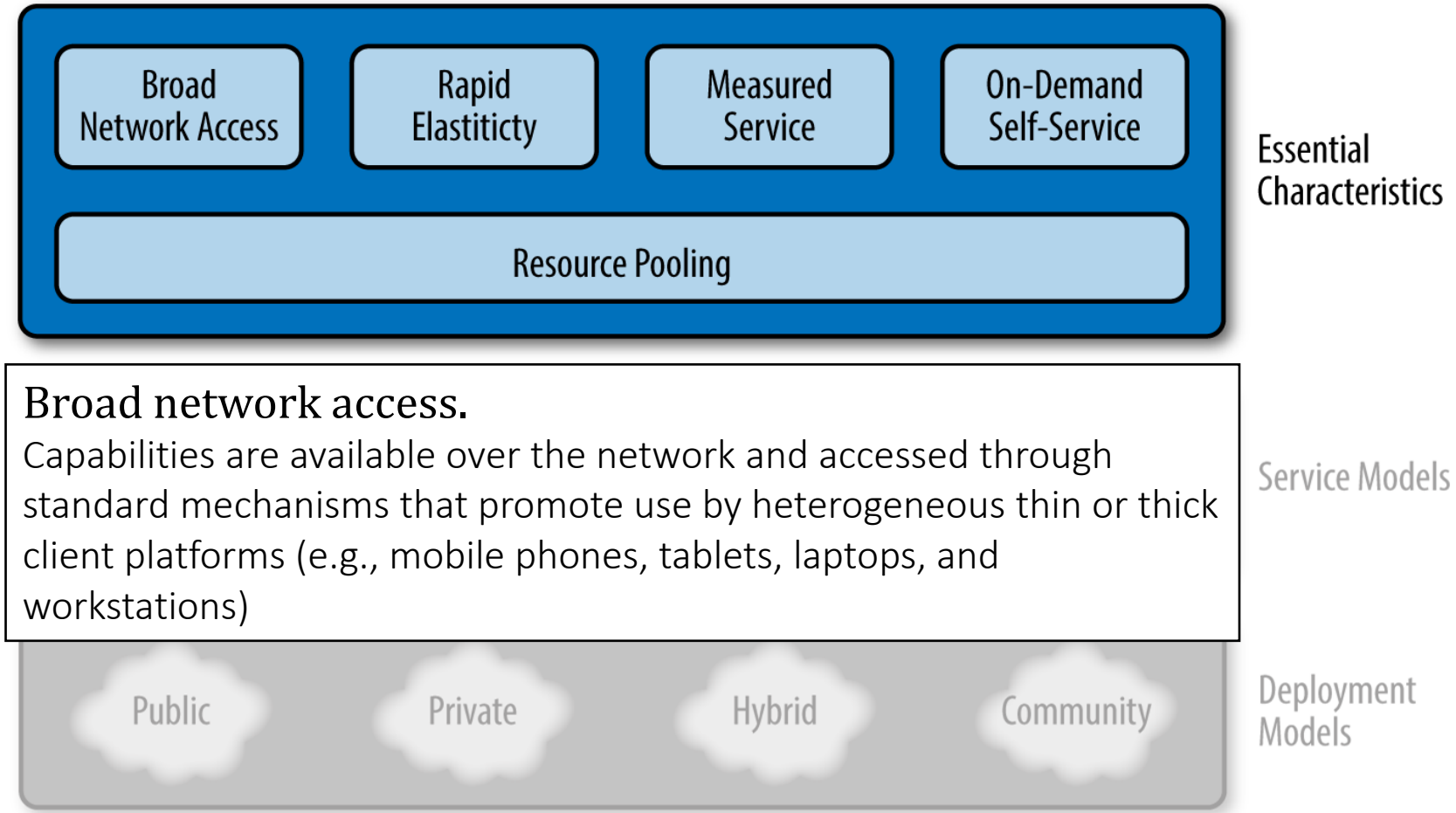
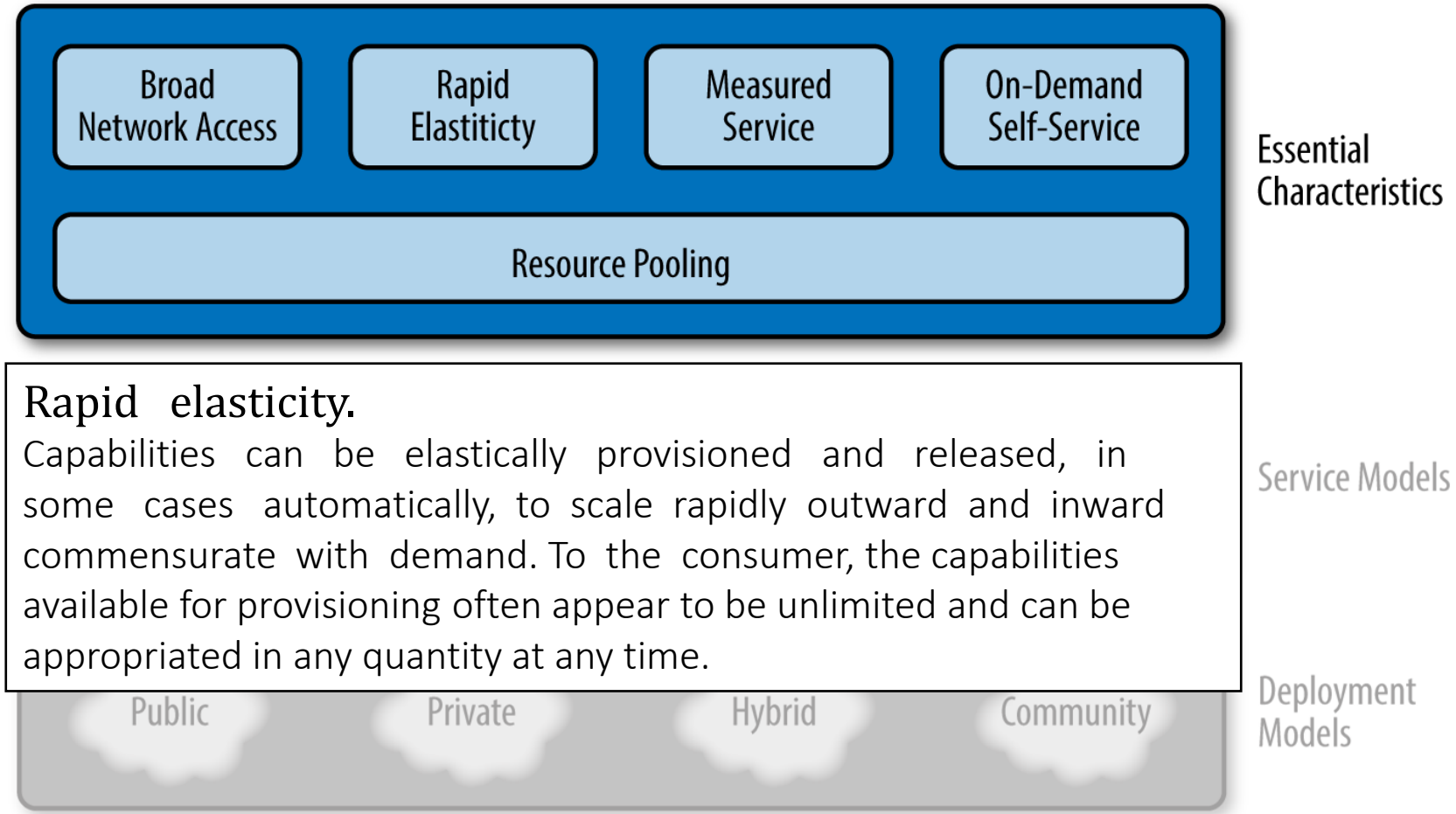


Image source

The NIST Definition of Cloud Computing (ref)

(National Institute for Standards and Technology)

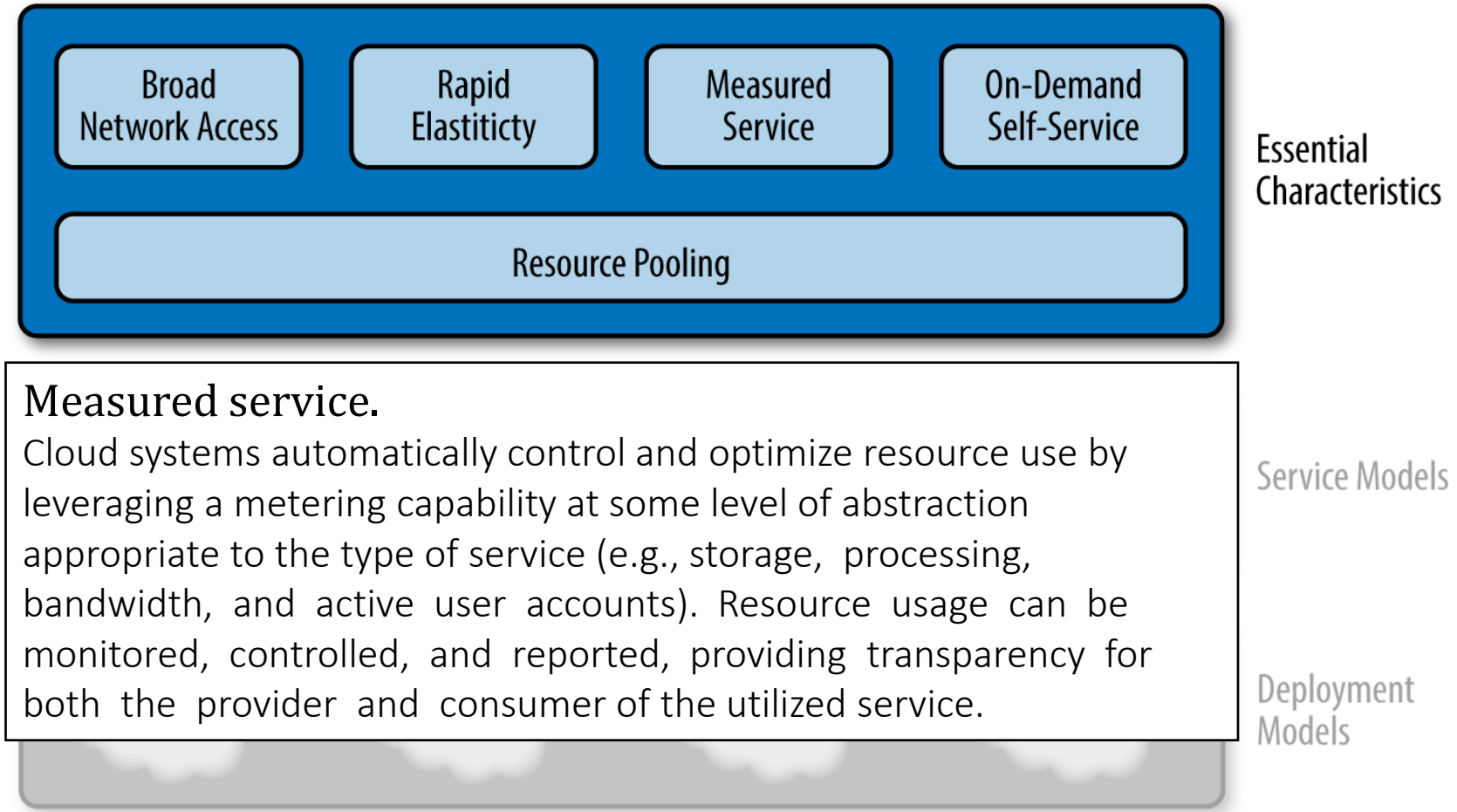


Rapid elasticity.
Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

Image source

The NIST Definition of Cloud Computing (ref)

(National Institute for Standards and Technology)



Measured service.
Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Image source

The NIST Definition of Cloud Computing (ref)

(National Institute for Standards and Technology)

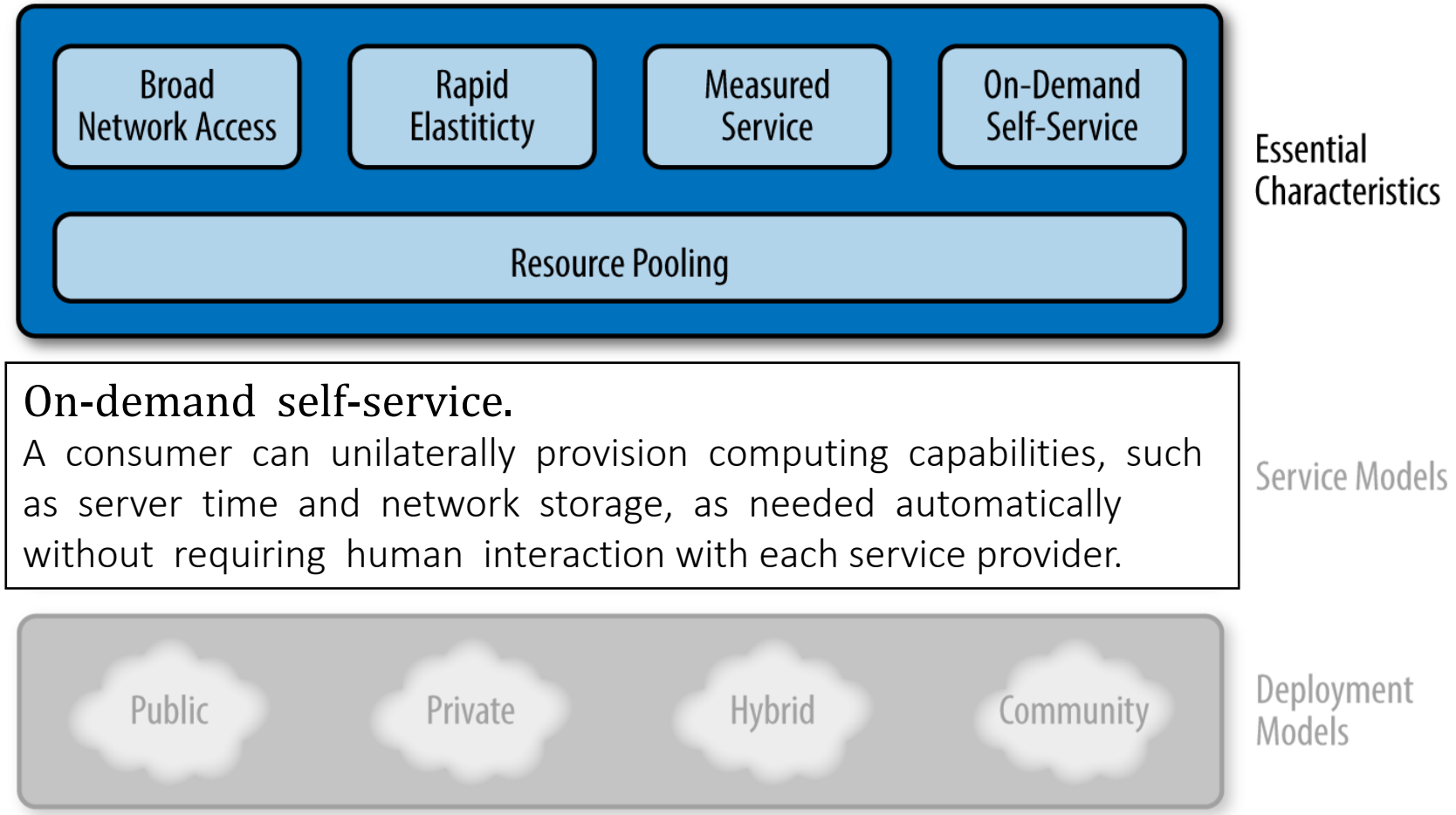


Image source

The NIST Definition of Cloud Computing (ref)

(National Institute for Standards and Technology)

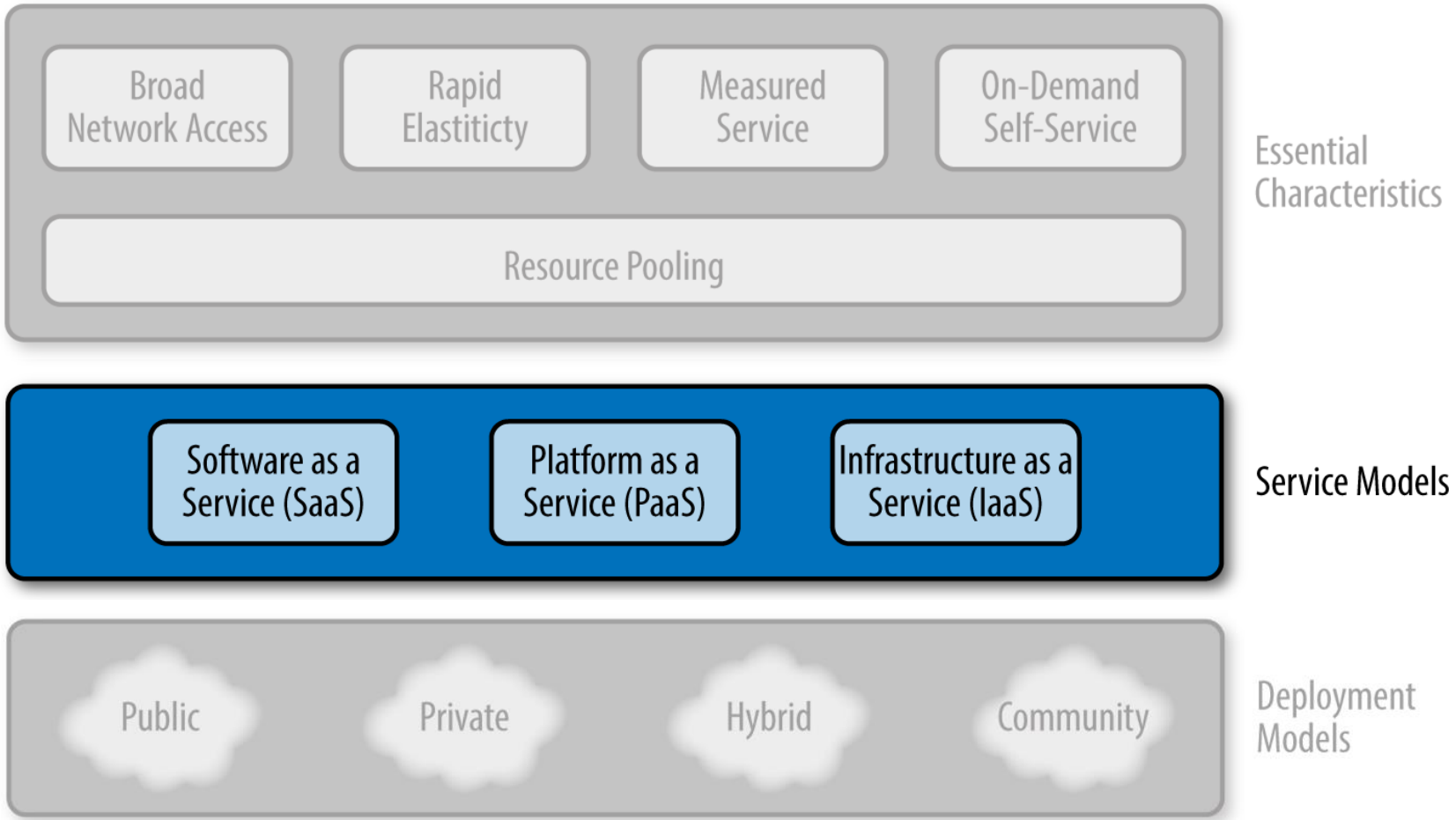
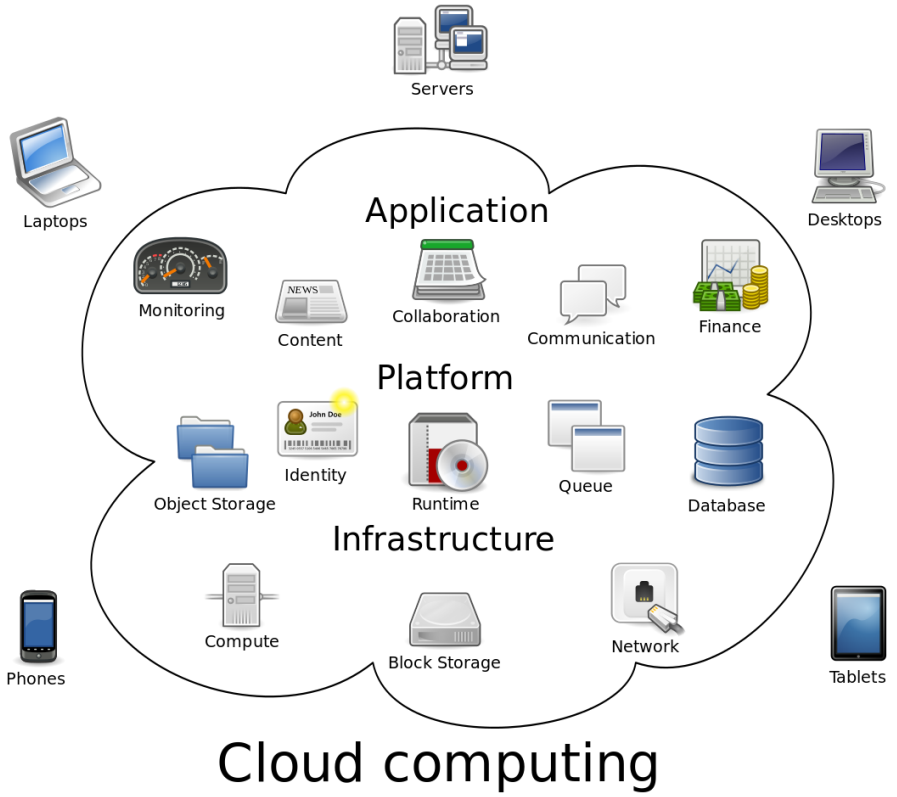


Image source

Cloud Computing Service Models

- Providers offer their "services" according to three different models
  - Infrastructure-as-a-Service (IaaS)
  - Platform-as-a-Service (PaaS)
  - Software-as-a-Service (SaaS)

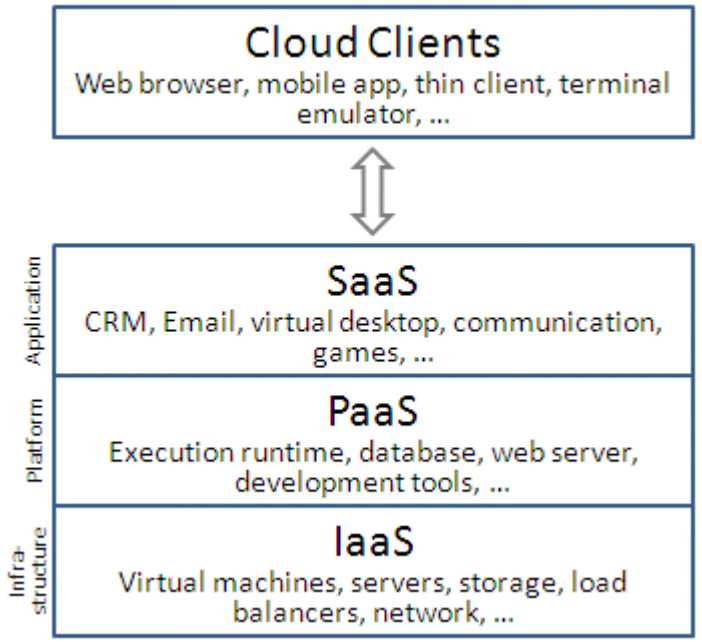
- "…as a service"?
  - Means "managed for you"
  - You focus on the remaining



Cloud Computing Service Models

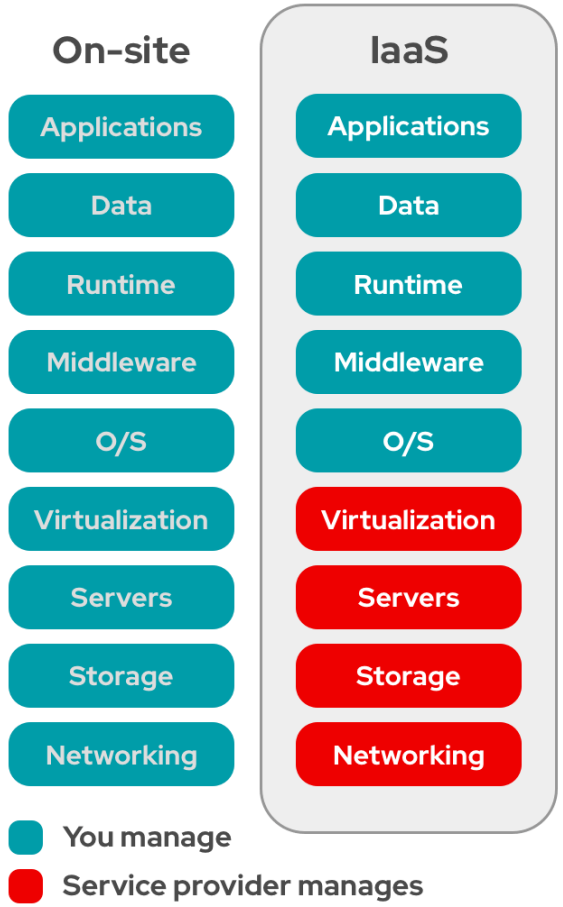
- Providers offer their "services" according to three different models
  - Infrastructure-as-a-Service (IaaS)
  - Platform-as-a-Service (PaaS)
  - Software-as-a-Service (SaaS)

These models offer increasing abstraction
They are thus often portrayed as layers in a stack



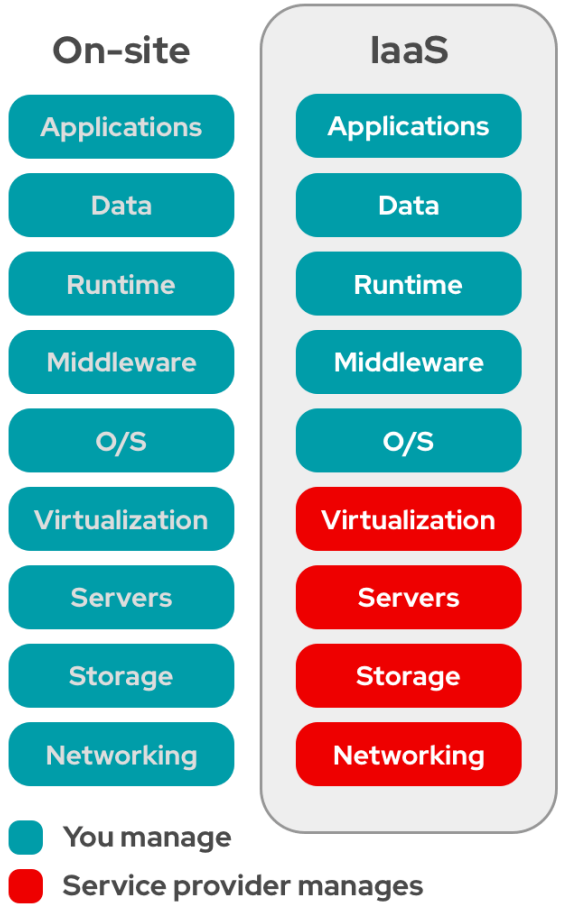
Cloud Computing Service Models

- Infrastructure-as-a-Service (IaaS)
  - Provides the virtualization, storage, network, and servers (“as-a-service”)
    - Managing / maintaining physical infrastructure is abstracted
      - Available as a collection of services accessible using code (API) or web interfaces
    - Developers still have to design and code entire applications
    - Administrators still need to install, manage and patch third-party solutions



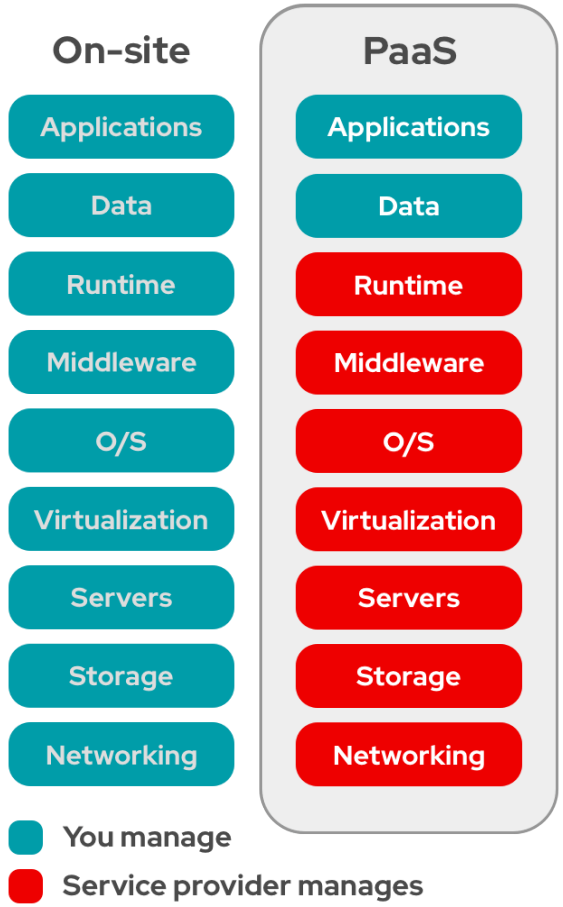
Cloud Computing Service Models

- Infrastructure-as-a-Service (IaaS)
  - Provides the virtualization, storage, network, and servers (“as-a-service”)
  - Examples
    - AWS EC2, Google Compute Engine (GCE)



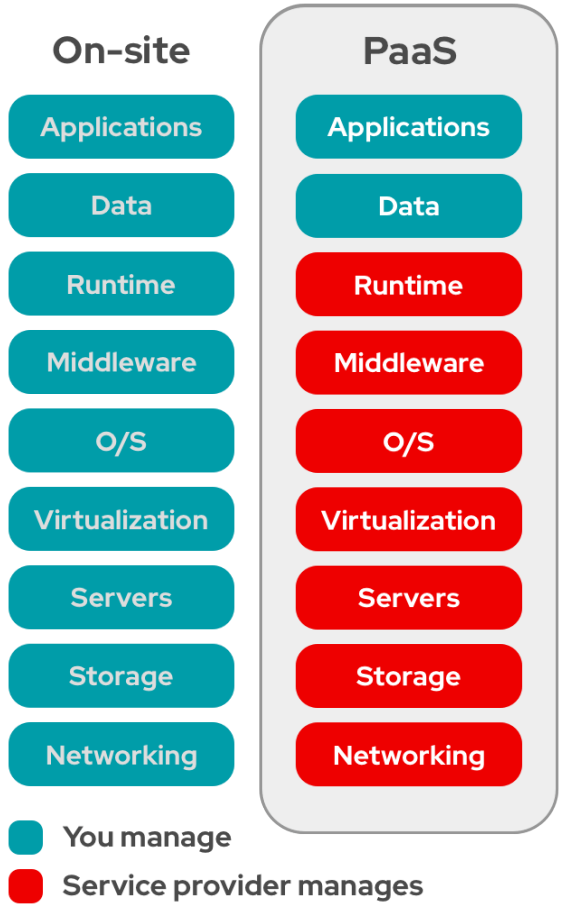
Cloud Computing Service Models

- Platform-as-a-Service (PaaS)
  - Provides infrastructure + specific software stack / platform as an integrated solution (“as-a-service”)
    - Typically includes operating system, programming-language execution environment, database, and web servers
    - Developers can focus just on their code or product
      - Frees time, less distraction (focus on core competence)
    - Setup and maintenance is all handled by the provider
    - Scalability is still possible



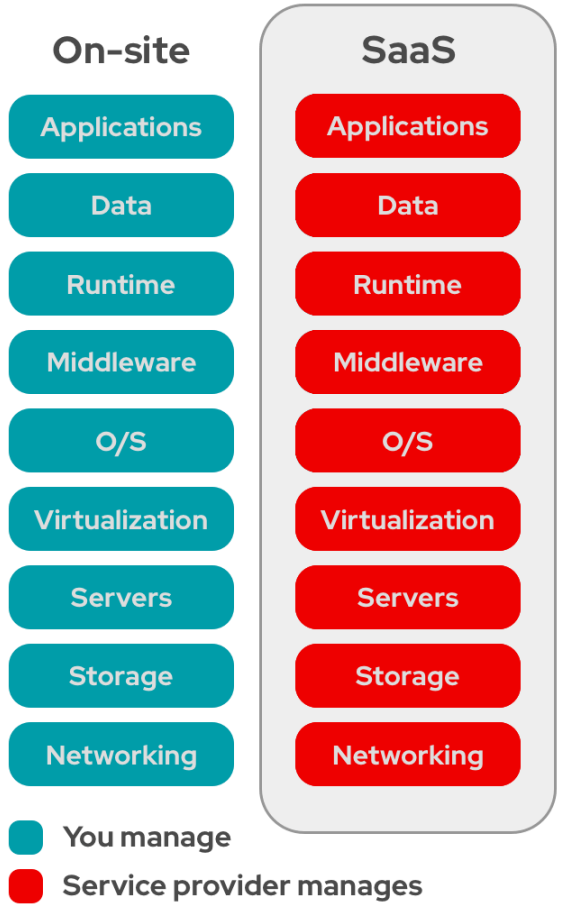
Cloud Computing Service Models

- Platform-as-a-Service (PaaS)
  - Provides infrastructure + specific software stack / platform as an integrated solution (“as-a-service”)
  - Examples
    - AWS Elastic Beanstalk, Heroku, Google App Engine, OpenShift



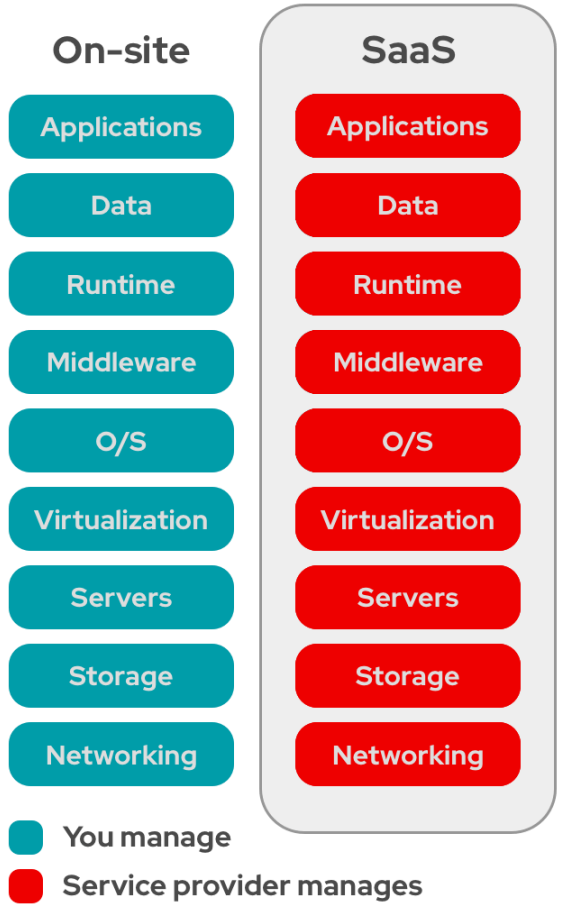
Cloud Computing Service Models

- Software-as-a-Service (SaaS)
  - Delivers an application to users (“as-a-service”)
    - Abstracts all the its underlying layers
      - Eliminates all responsibility of maintaining infrastructure, platforms, and software
    - A good solution for “support services” of a business that require minimal customizations to solve
      - Also reduces initial costs by using subscription models

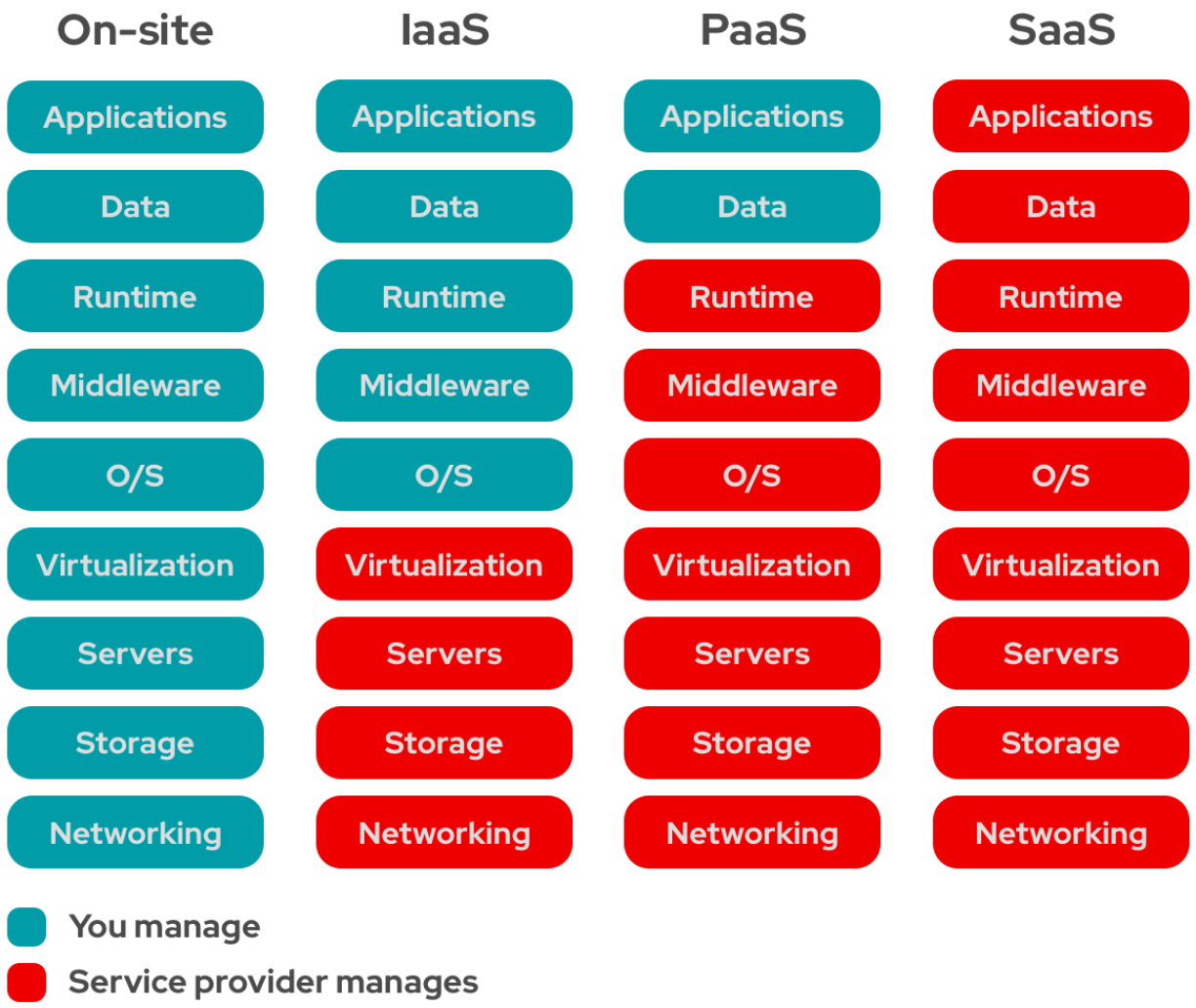


Cloud Computing Service Models

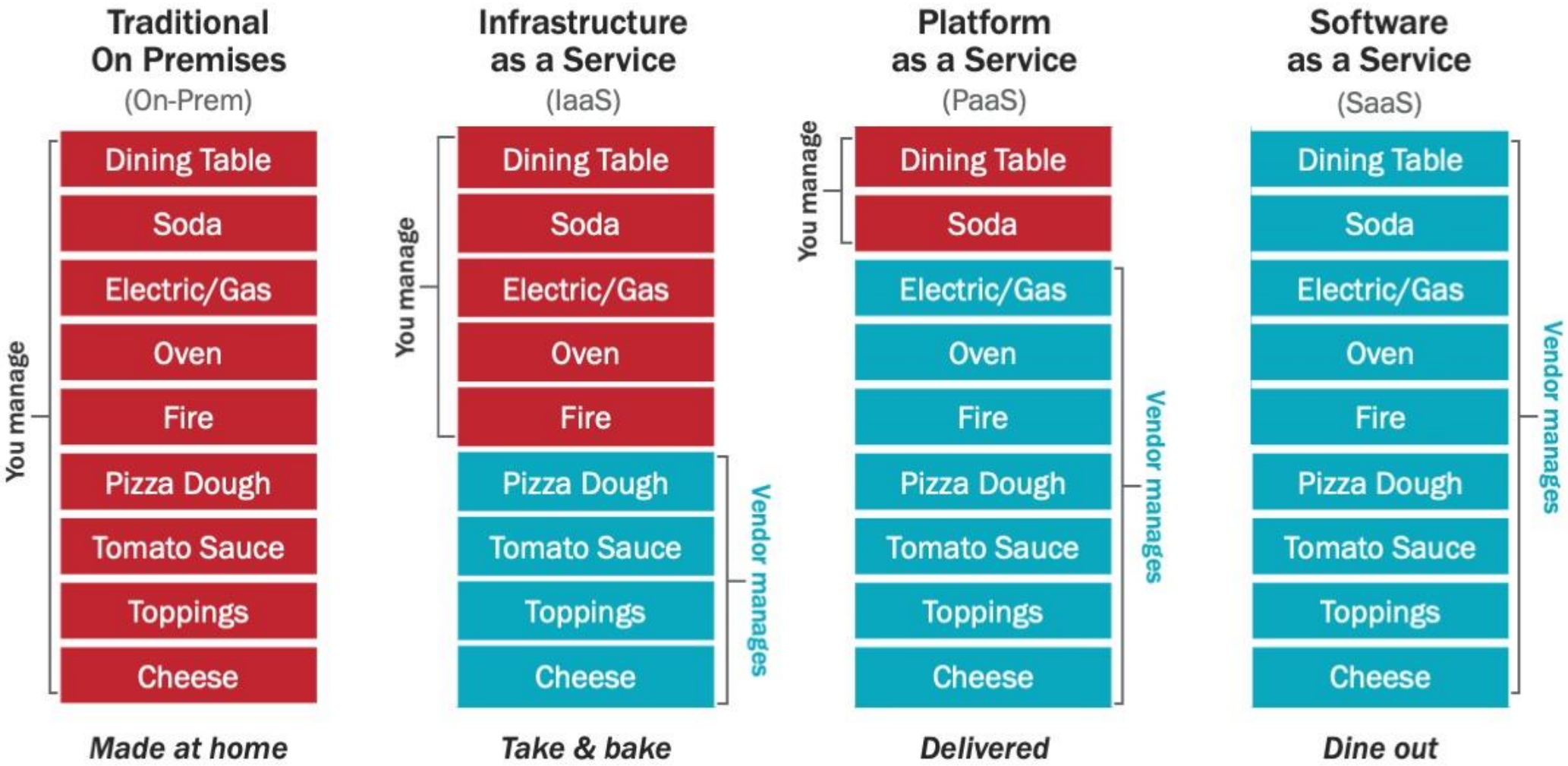
- Software-as-a-Service (SaaS)
  - Delivers an application to users (“as-a-service”)
  - Examples
    - Google Apps/Docs, Office365, FB, YouTube, GitHub, Slack, WordPress (products), …



Cloud Computing Service Models



Pizza as a Service



Travel as a Service

● customer's responsibility ● vendor's responsibility

Independent Travel by Caravan On-Premises	Self-Guided Travel by Plane/Bus/Train IaaS	Partially Guided Travel PaaS	All Inclusive Travel SaaS
Accommodations	Accommodations	Accommodations	Accommodations
Luggage transfer	Luggage transfer	Luggage transfer	Luggage transfer
Transport	Transport	Transport	Transport
Routing	Routing	Routing	Routing
Visa	Visa	Visa	Visa
Medical insurance	Medical insurance	Medical insurance	Medical insurance
Tickets booking	Tickets booking	Tickets booking	Tickets booking
Food + entertainment	Food + entertainment	Food + entertainment	Food + entertainment
Info on sightseeing	Info on sightseeing	Info on sightseeing	Info on sightseeing

Why not just use PaaS for developing our product and SaaS for everything else?

Got it! IaaS, PaaS, SaaS!

Cloud Computing Service Models

- Over the years new models have been proposed
  - Function-as-a-Service (FaaS)
  - Containers-as-a-Service (CaaS)
    - KaaS
  - Mobile backend as a service (MBaaS)

  - They are just other (further?) abstractions of software development concepts

Cloud Computing Service Models

- Function-as-a-Service (FaaS)
  - A way to implement *serverless computation*
  - Event-driven computing execution model
  - Executed inside stateless containers
  - The implemented functions manage server-side logic and state using other services

Cloud Computing Service Models

- Function-as-a-Service (FaaS), in other words…
  - Developer builds and deploys a function (code)
    - E.g., a function triggered by an HTTP POST, that saves data to some database (service)
  - The FaaS service provider executes it on demand
  - Developer does **not** know **how** or **where** it is executed
    - Developer does not care…

Cloud Computing Service Models

- With Function-as-a-Service (FaaS) there are no…
  - Virtual machines to provision and manage
  - Concerns about server capacity and scaling
    - Too many idling servers?
    - Too much requests for the provisioned servers?

All this is **abstracted** away

- You cannot **over** provision or **under** provision execution capacity
- It just executes and **charges you exactly** for that execution

Cloud Computing Service Models

- With Function-as-a-Service (FaaS) there are no…
  - Virtual machines to provision and manage
  - Concerns about server capacity and scaling
    - Too many idling servers?
    - Too much requests for the provisioned servers?

Examples

- AWS Lambda, Google Cloud Functions, Azure Functions

Cloud Computing Service Models

- Containers-as-a-Service (CaaS)
  - Applications (containers) are deployed and managed through container-based abstraction services
  - Container engines, orchestration and the underlying compute resources are delivered to users as a service
    - Can be the cloud but also an on-premises datacenter
  - Somewhere between IaaS and PaaS

Cloud Computing Service Models

- Containers-as-a-Service (CaaS) advantages
  - Deploys quickly and lightly on any infrastructure
  - Provides standardized functionality on-premise or public clouds
    - Different clouds can be linked and treated as one
  - All benefits of open container technology, without the obscurity and limitation of PaaS
    - PaaS systems actually exploit containerization

- Examples
  - Google Container/Kubernetes Engine (GKE), AWS Elastic Container Service (ECS/EKS), Azure Container Service (ACS/AKS)

Cloud Computing Service Models

- Mobile-backend-as-a-Service (MBaaS)
  - Also known as “Backend-as-a-Service”
  - Provides easy backend cloud services to frontend apps (web/mobile)
    - Via custom SDKs and APIs to access several resources
      - Backend cloud storage
      - Databases
      - User management
      - Push notifications

Cloud Computing Service Models

- Why Mobile-backend-as-a-Service (MBaaS)?
  - The typical web or mobile app has very similar backend requirements
    - API to access and store data, notifications, social integrations
  - MBaaS provides a bridge between frontend and these services
  - Like other models, but specially tailored to mobile/web app developers

- Examples
  - Google Firebase, AWS Amplify, Azure Mobile (?)


{* SOFTWARE *}

Early adopters delighted as Microsoft pulls plug on Mobile Backend as a Service. Haha, only joking – they're fuming

'Horrible news' for invested devs

By Tim Anderson 17 Feb 2020 at 11:45 48 SHARE ▼

The NIST Definition of Cloud Computing (ref)

(National Institute for Standards and Technology)

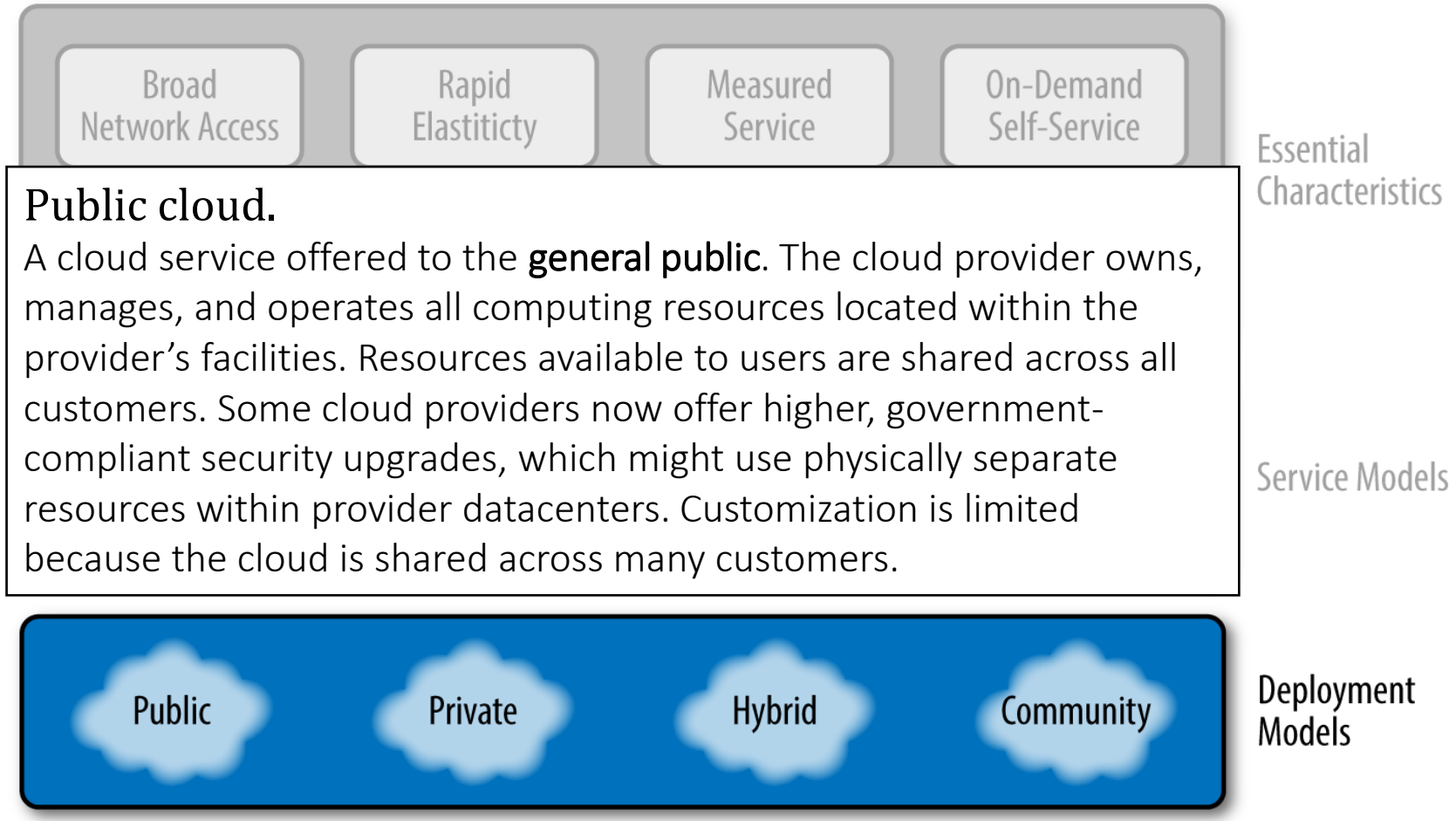


Image source

The NIST Definition of Cloud Computing (ref)

(National Institute for Standards and Technology)

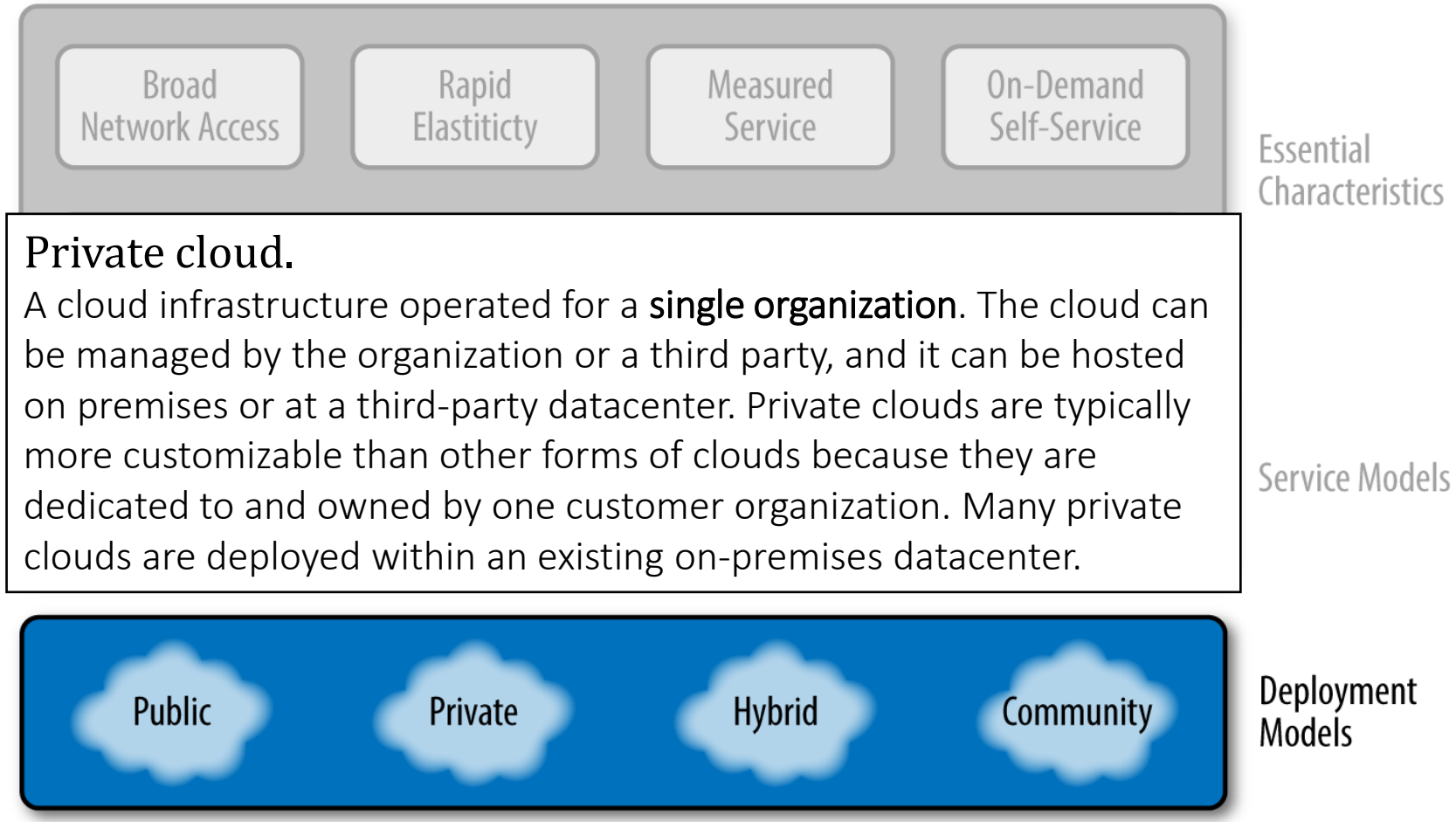
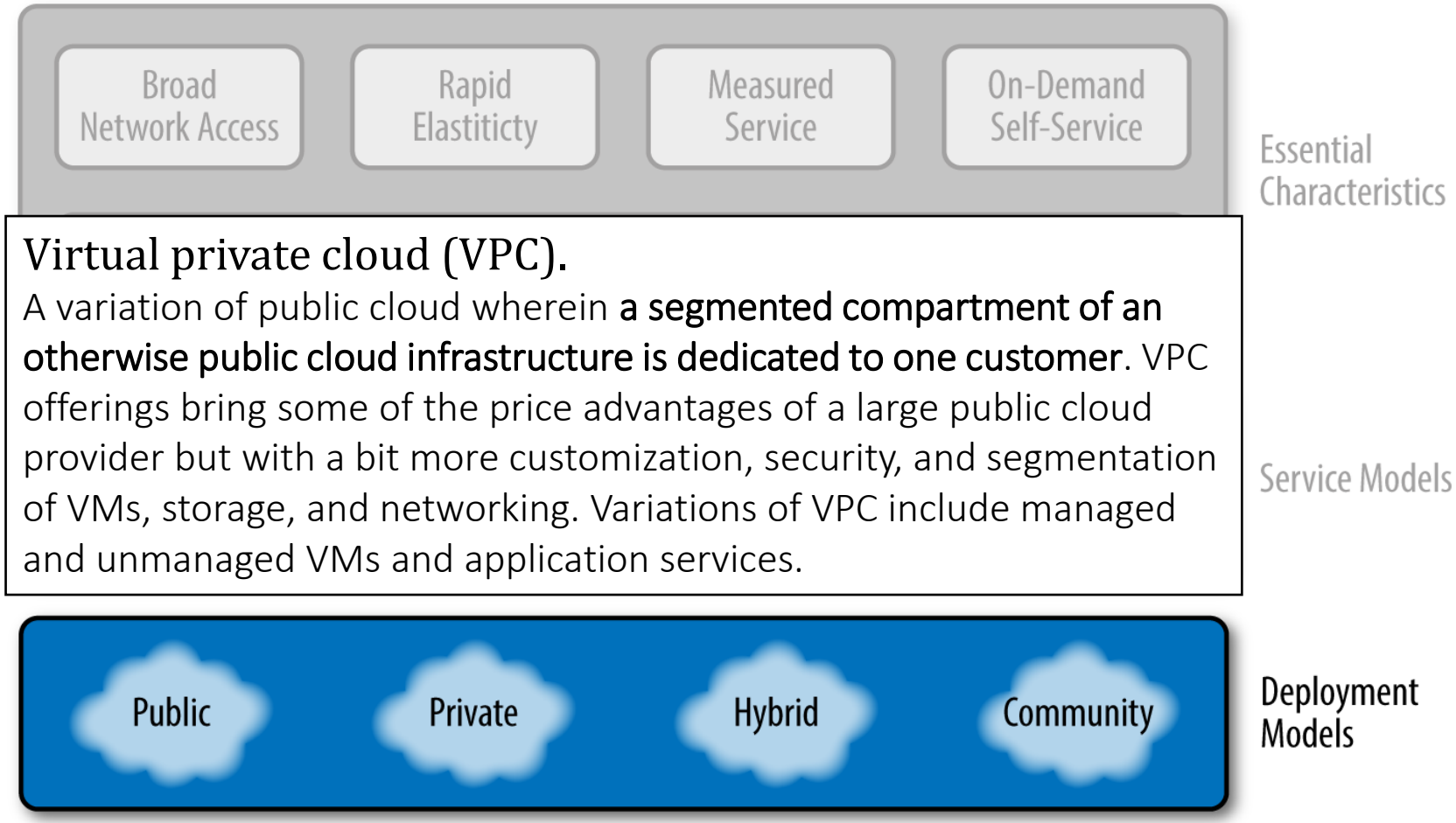


Image source

The NIST Definition of Cloud Computing (ref)

(National Institute for Standards and Technology)



Bonus!

Image source

The NIST Definition of Cloud Computing (ref)

(National Institute for Standards and Technology)

Community cloud.
A cloud service that provides for a **community of users** or organizations with **shared interests** or concerns. The system is managed by one or more of the organizations, by a central provider, or a combination of the two. Organizations utilizing this cloud service have shared missions, governance, security requirements, and policies. Cloud services can be hosted on-premises at the consumer organization, at peer organization facilities, at a provider, or a combination of these. This community cloud term is often used in marketing to explain the target consumers of the service, although **the actual cloud might technically be a VPC, private, or hybrid** cloud model.

Essential Characteristics

Service Models

Deployment Models

Image source

The NIST Definition of Cloud Computing (ref)

(National Institute for Standards and Technology)

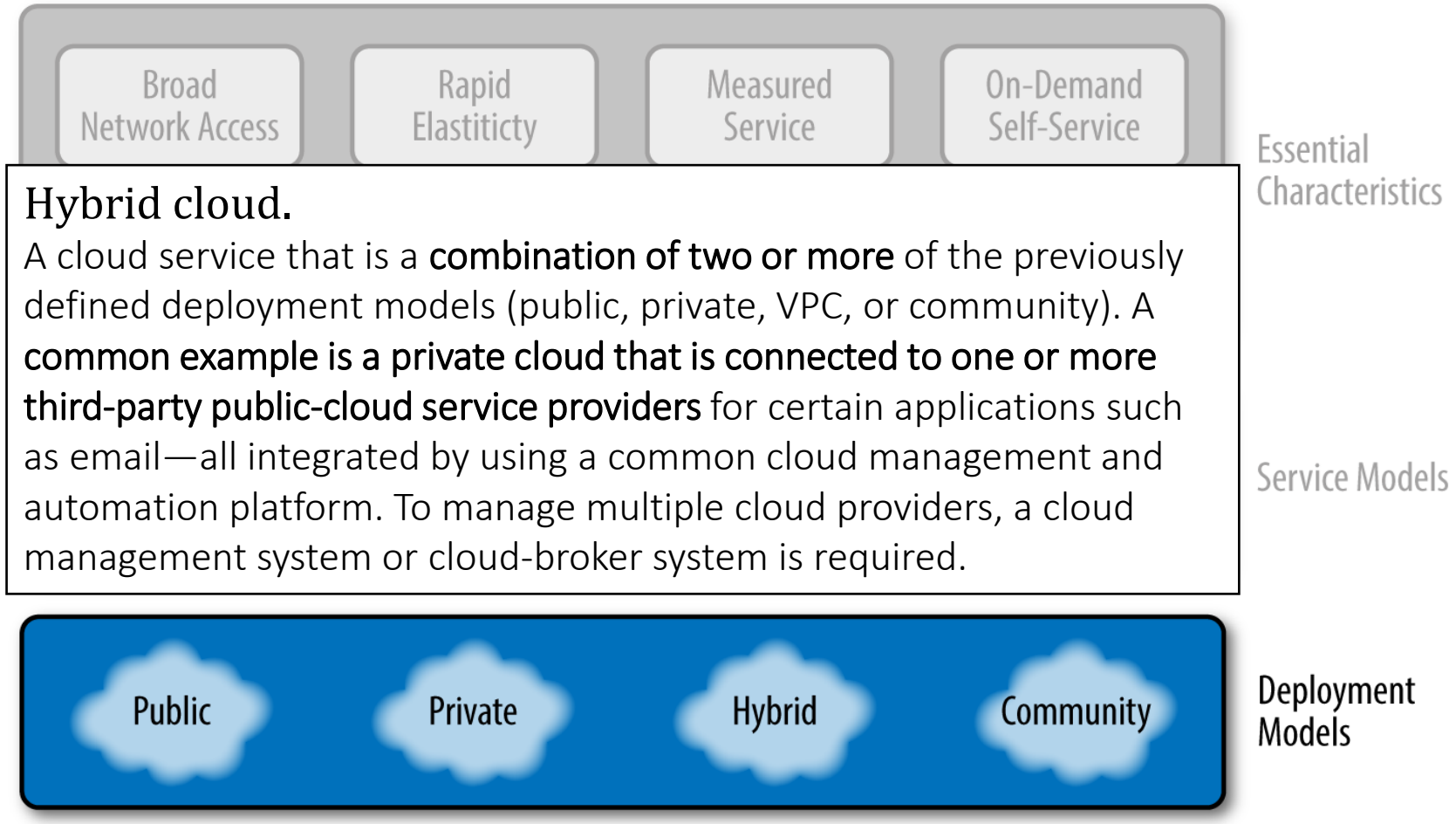
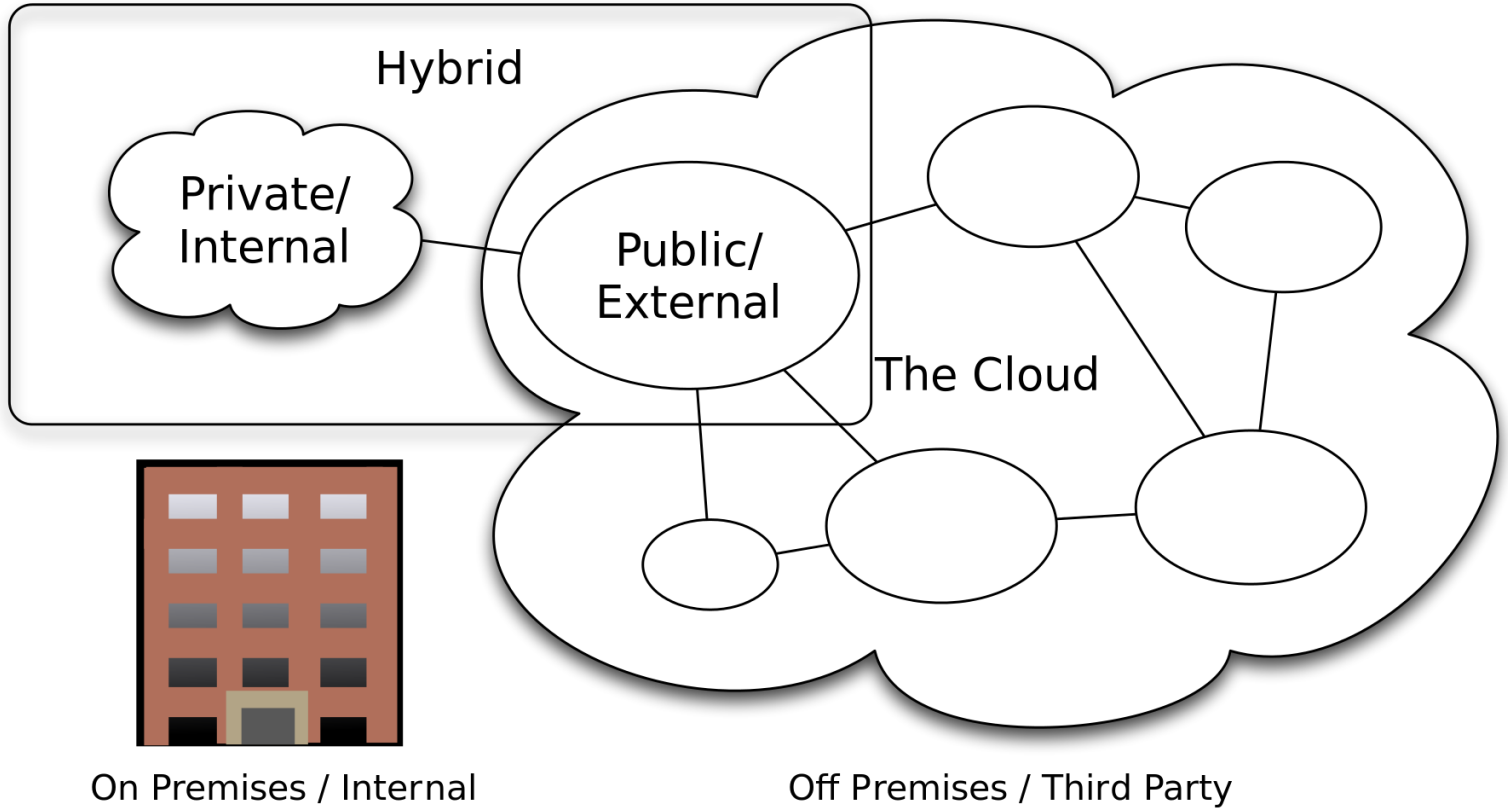


Image source

Cloud Deployment Models



Cloud Computing Types

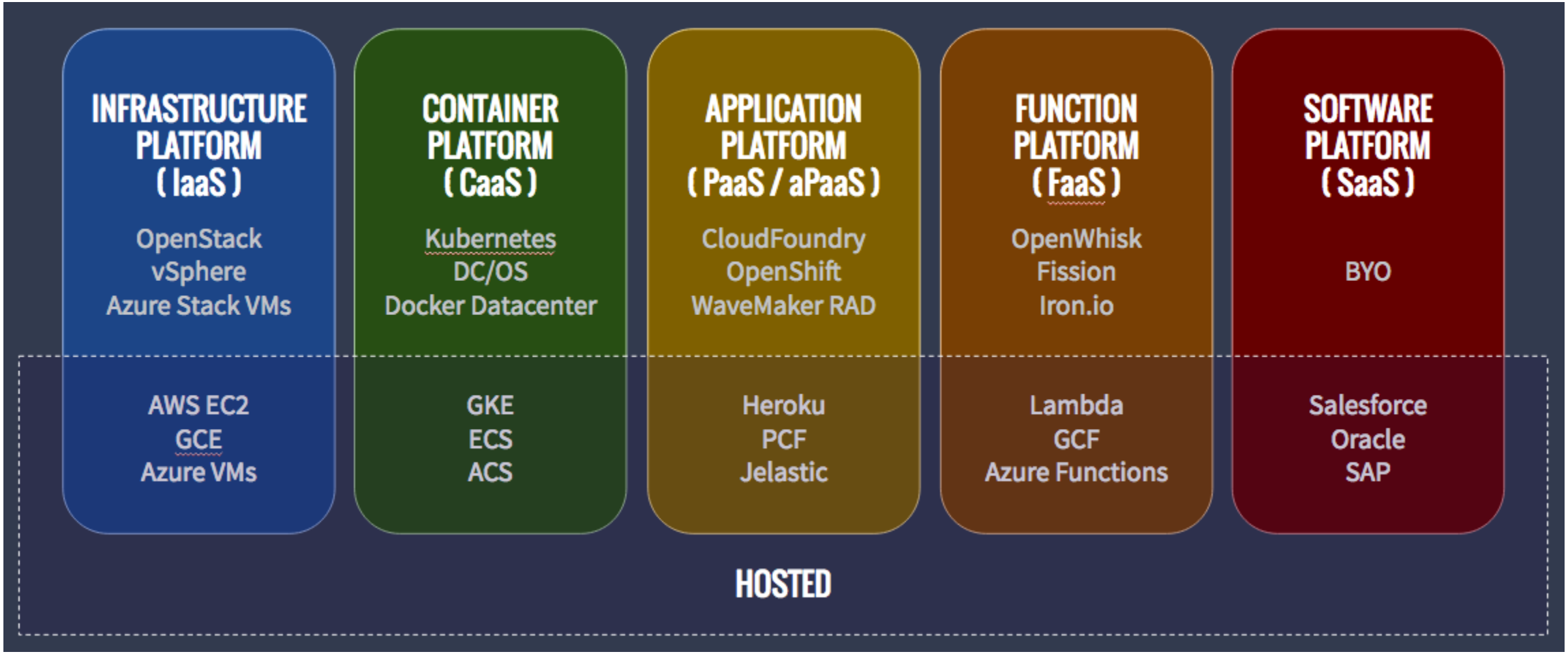
Cloud Deployment Models

- Advantages of public clouds (vs. private clouds)
  - Utility pricing
    - End user pays only for the resources consumed
  - Elasticity
    - Seemingly endless pool of resources that can be dynamically scaled as needed
  - Core competency
    - End user outsources the data center and infrastructure management to the provider
    - End user can focus on its own core competency

Cloud Deployment Models

- Drawbacks of public clouds (vs. private clouds)
  - Control
    - End user must rely on the cloud provider to meet SLAs (tip: architect redundancy properly!)
  - Regulatory issues
    - Data privacy issues might hinder public cloud deployments
    - Hybrid clouds are a possible solution (host the sensible part in a private cloud)
    - Some public clouds started providing certified SaaS solutions for specific components
  - Limited configurations
    - Public clouds have standard sets of infrastructure and product “templates”
    - Very specific requirements (HW/SW) might be impossible to get

Cloud Service Model/Deployment Examples



Selecting the Right Cloud Service Model

What service model should I use? PaaS? SaaS? FaaS? MBaaS? XaaS? …aaS?

Why isn't everyone just using SaaS or… ?

Selecting the Right Cloud Service Model

- Can one cloud service model fit all scenarios?
  - Why pick one model and force-fit all solutions into that?
  - Even in the same application you might end up using all of them

- Several aspects should be considered
  - Technical
  - Financial
  - Strategic
  - Organization
  - Risk

Selecting the Right Cloud Service Model

- Several aspects should be considered
  - Technical
    - Performance, scalability, security, regulation, business continuity, disaster recovery, and so on, e.g.:
    - PaaS handles auto scaling but typically enforces some (high) limits
      - In some cases that might be a limitation (why Netflix, Twitter and others use IaaS instead?)
    - DBaaS handles replication, autoscaling, monitoring, backups, and more
      - The limitation is lack of control over the database

Selecting the Right Cloud Service Model

- Several aspects should be considered
  - Financial
    - What is the total cost of ownership?
    - This often is more than just calculating the price per hour of the cloud service
    - Costs might include
      - Migrating legacy code
      - Rearchitecting solutions
      - Training employees

Selecting the Right Cloud Service Model

- Several aspects should be considered
  - Strategic
    - How important is speed-to-market?
    - How important is having control?
    - Business strategies such as reducing costs, consolidating data centers and others are also very important (and sometimes neglected)

Selecting the Right Cloud Service Model

- Several aspects should be considered
  - Organization
    - Does the team/company have the skills to build solutions in the cloud?
    - The lower down the cloud stack we go, the higher the degree of competence we need

Selecting the Right Cloud Service Model

- Several aspects should be considered
  - Risk
    - How much risk is a company willing to assume?
    - How long can the solution be down?
    - How damaging is a security breach?
    - Can the government seize the data in the cloud with a warrant?

    - Risk (privacy, data ownership, regulation) is also a major factor when choosing a public or private solution

Selecting the Right Cloud Service Model

- When to use SaaS?
  - Outsource applications, features, and services that are not a core competency of the team
    - Assuming it meets the needs and is affordable
- When to use PaaS?
  - Simpler projects, without very high load or complexity, where time-to-market is crucial, small teams/startups
- When to use IaaS?
  - Complex applications, with high scalability and performance requirements that need specific configurations (OS, web servers, DBMS, …)
  - Reduce costs, custom multi-zone redundancy solutions (e.g., Netflix case)

Public Cloud Providers

- There are many (1000+) cloud service providers out there

- Initially their offer was quite different
  - Amazon provided IaaS services (EC2)
  - Azure was suited for Windows solutions
  - Google offered a PaaS solution for Python and Java projects (App Engine)

- Nowadays most provide similar services under different names and APIs

Heroku

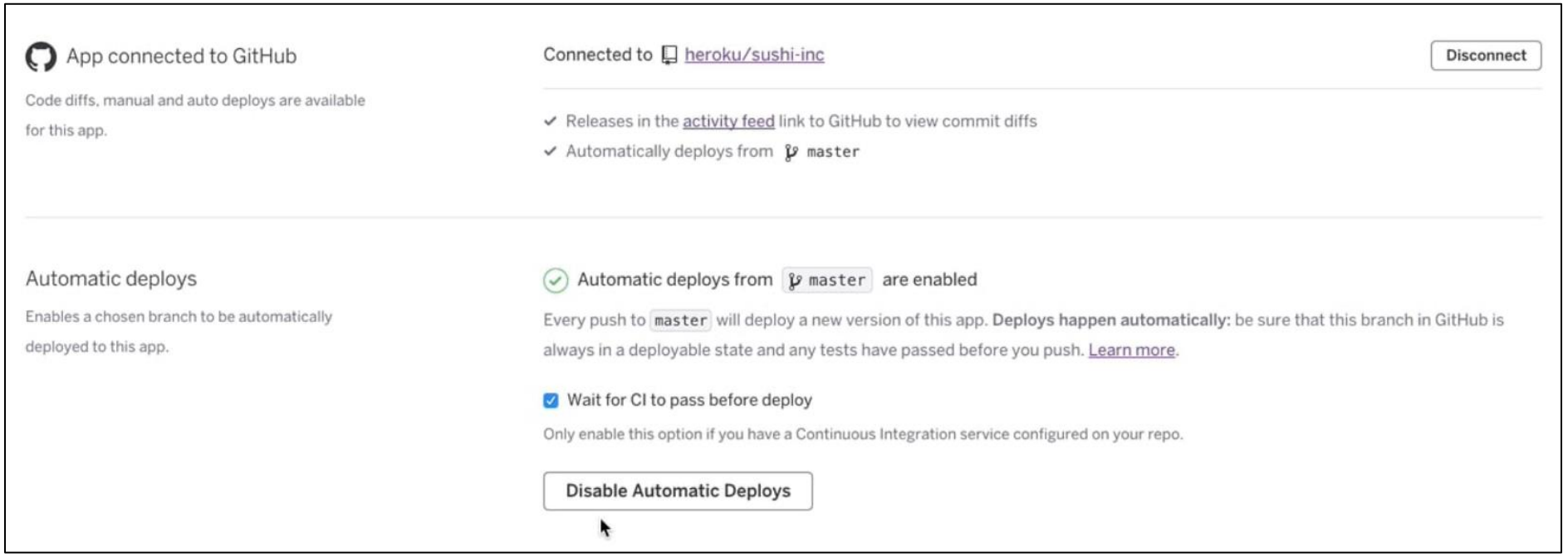
"*Heroku* is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud."

- Engineers can dedicate all their time to
  - Develop applications
  - Deploy and manage them
  - Scale resources (& pay for it ☺) as needed

… instead of spending resources to setup and maintain infrastructure and so on

Heroku

- Heroku has a web interface and also CLI tools
- Extremely simple to use, you can deploy directly from git with a single click



PaaS: Heroku

- Deploying your new app to Heroku is simple:
  - Register an account (free or a paid one) – no more free plans in 2023!
  - Create a “new app” in the Heroku dashboard
  - Follow the guides, example with GitHub:
    1. Connect your GitHub repo and select branch
    2. Press “deploy” button or activate automatic deploy
    3. You might want to use the Web Console/Terminal to seed your DB or create an admin account
  - Similar steps can be done directly from your terminal using Heroku Toolbelt (CLI)

```
$ git push heroku master
$ heroku ps:scale web=3 queue=2
$ heroku ps
== web: 'java lib/foobar.jar $PORT'
web.1: up 2013/02/07 18:59:17 (~ 13m ago)
web.2: up 2013/02/07 18:52:08 (~ 20m ago)
web.3: up 2013/02/07 18:31:14 (~ 41m ago)

== queue: `java lib/queue-processor.jar`
queue.1: up 2013/02/07 18:40:48 (~ 32m ago)
queue.2: up 2013/02/07 18:40:48 (~ 32m ago)
```

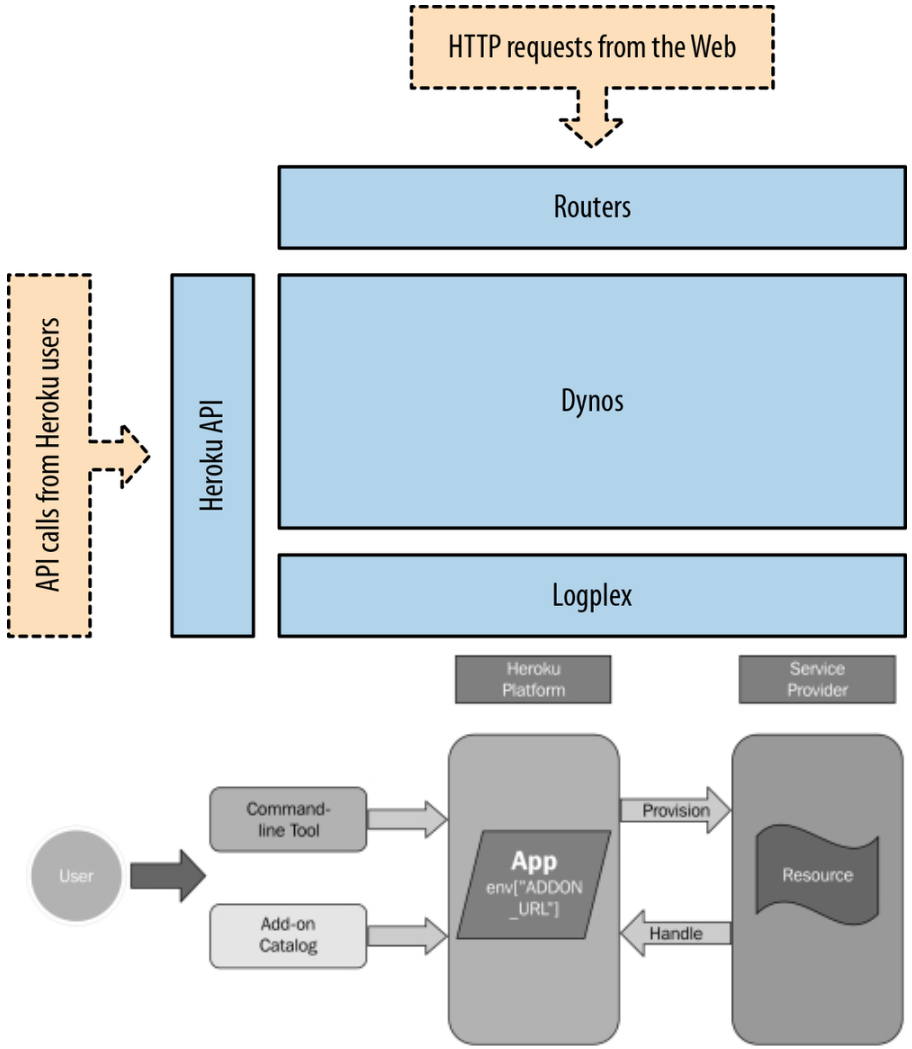
```
$ heroku config:set ENCRYPTION_KEY=my_secret_launch_codes
Adding config vars and restarting demoapp... done, v14
ENCRYPTION_KEY:     my_secret_launch_codes
(...)
$ heroku releases:rollback v102
Rolling back demoapp... done, v102
$ heroku releases
== demoapp Releases
v104 Rollback to v102 jon@heroku.com   2013/01/31 14:11:33 (~15s ago)
v103 Deploy 582fc95   jon@heroku.com   2013/01/31 12:15:35
v102 Deploy 990d916   jon@heroku.com   2013/01/31 12:01:12
```

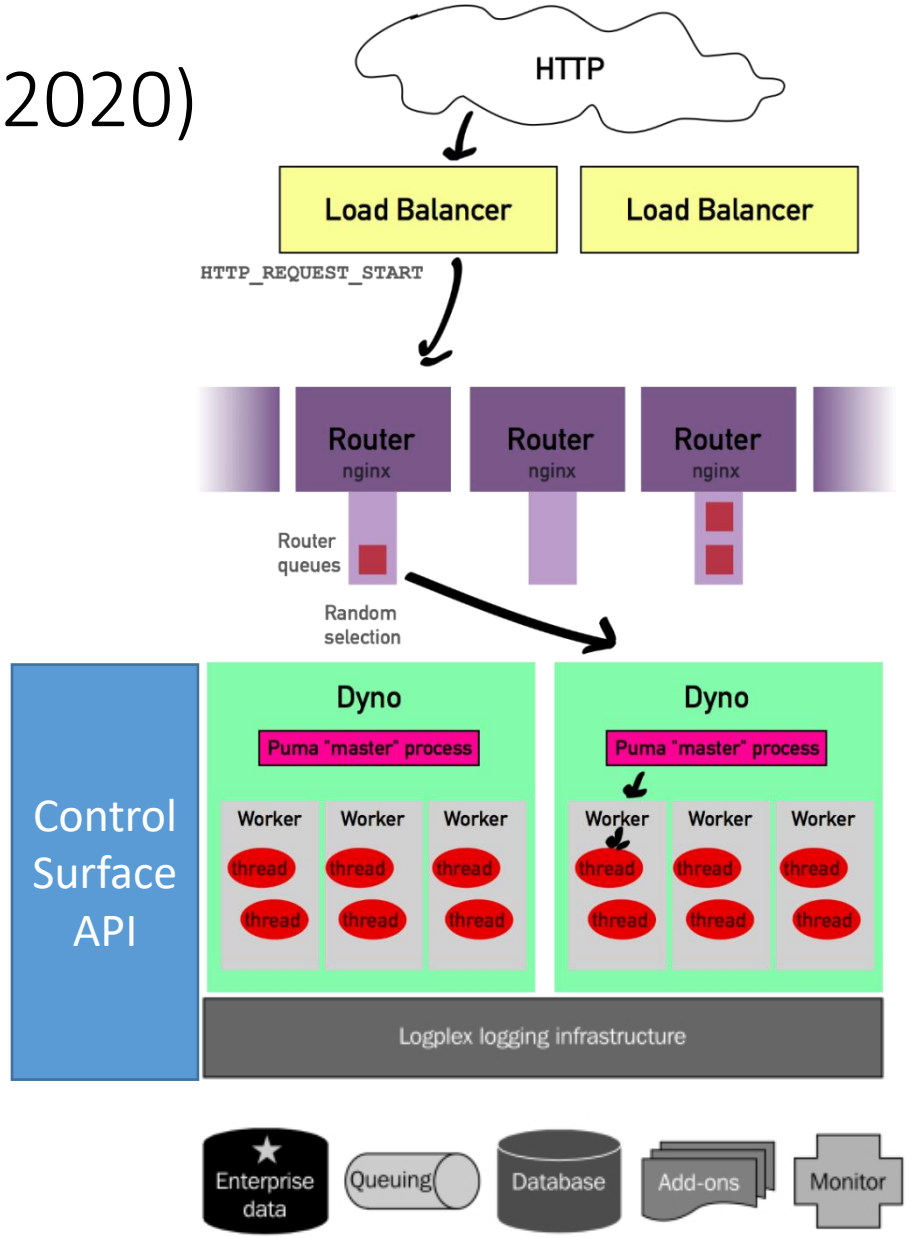
Heroku

- Heroku runs on AWS
  - If Heroku OR AWS fail, your app fails
    - Heroku is limited to US and EU datacenters
  - You pay AWS + Heroku (higher cost)

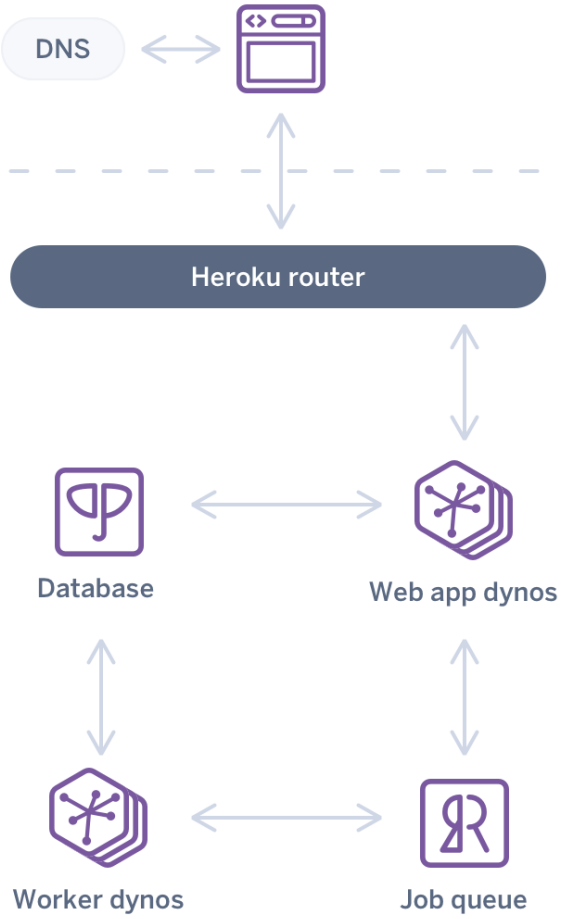
- When should I use Heroku?
  - A project with low computational requirements
  - A team that cannot afford a DevOps (or N) expert with AWS knowledge or similar
  - To deploy and test a minimum viable product
  - Project that can be quickly changed and so on

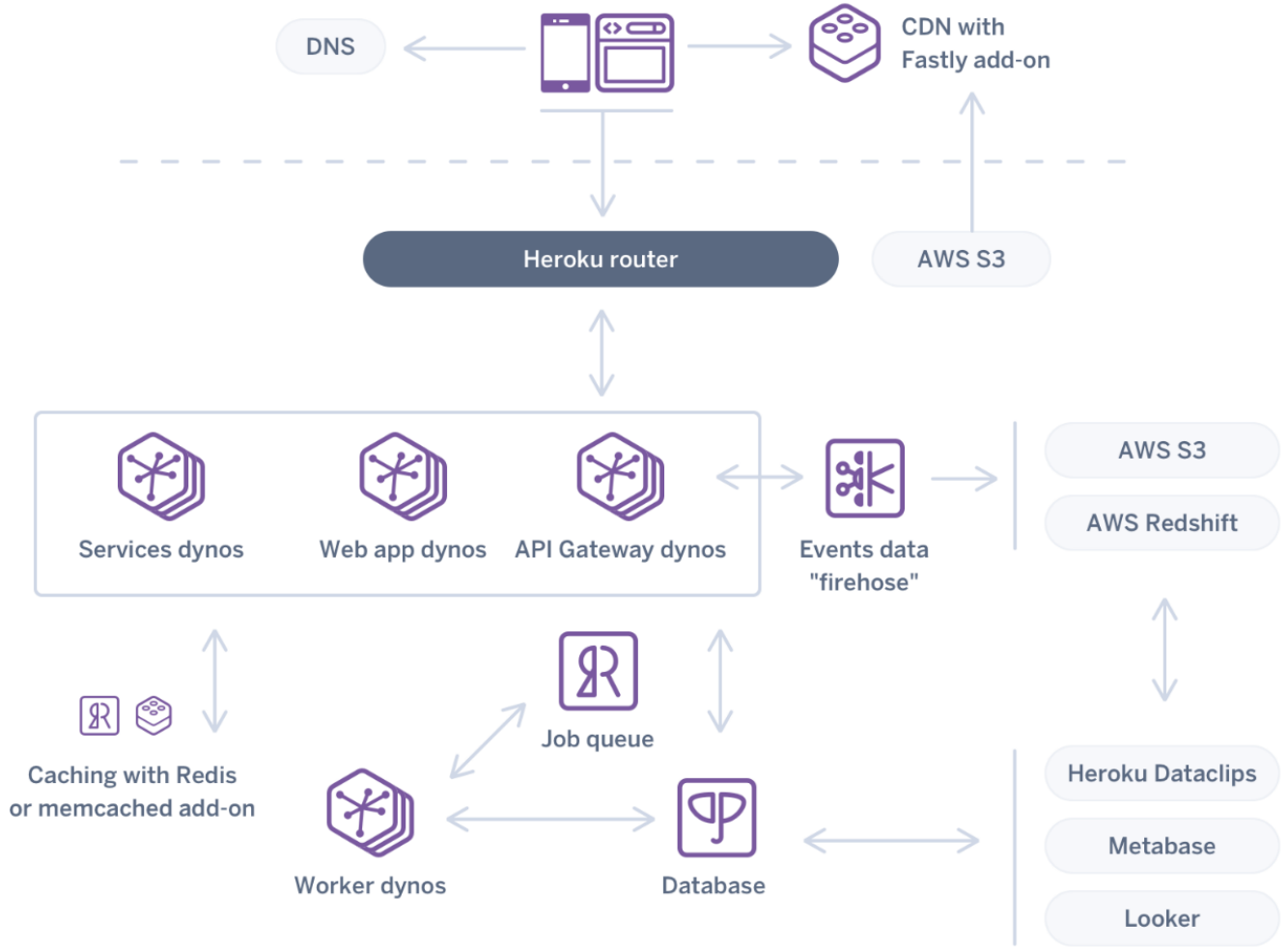
How does Heroku Works? (2020)





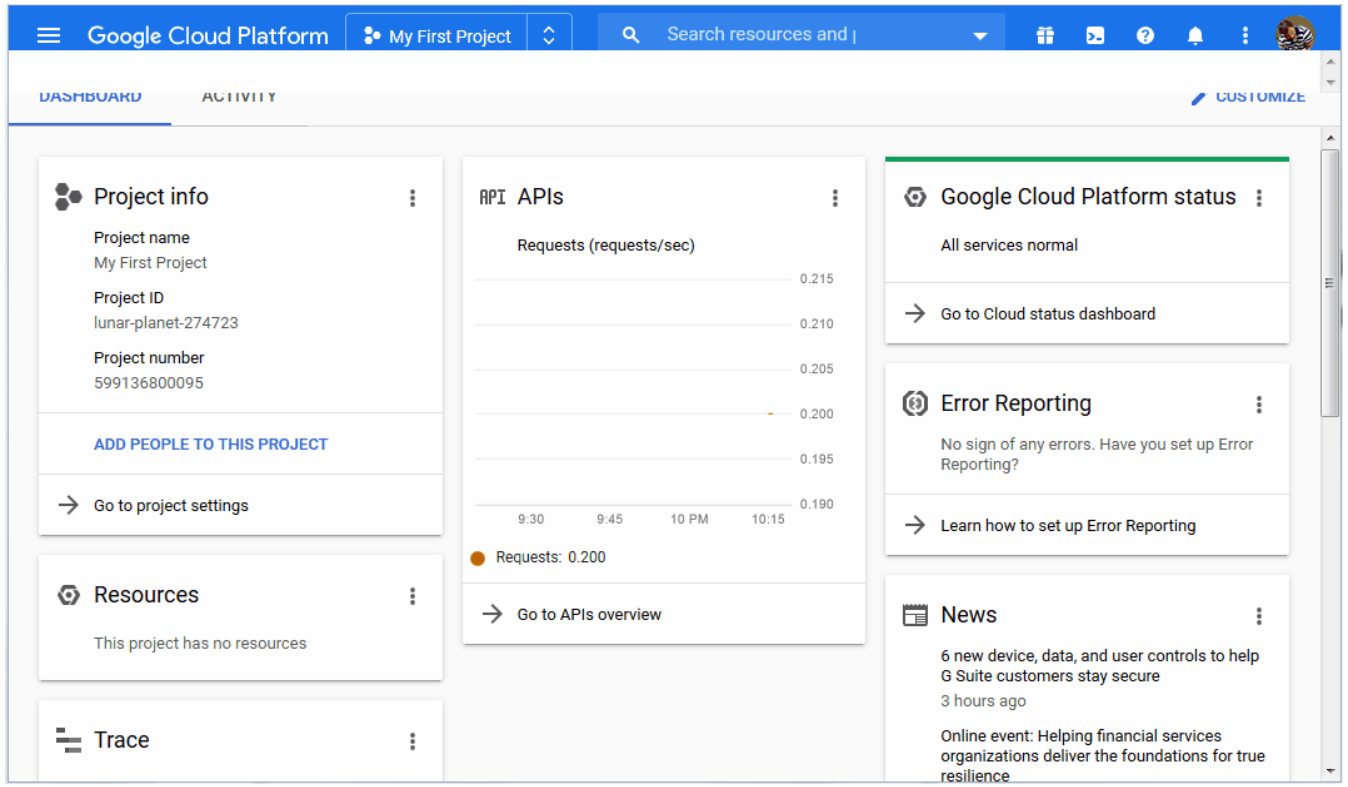
Modern web app on Heroku in 2008 vs 2018





Google Cloud Platform

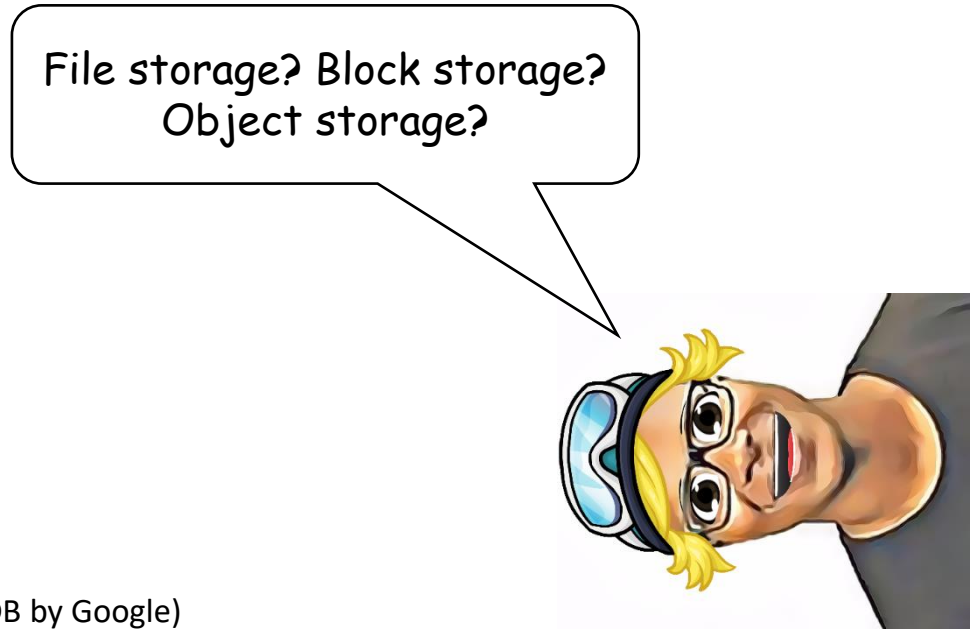
- Google Cloud Platform provides IaaS, PaaS and *serverless* solutions
  - Used by Google itself: Google Search, Gmail and YouTube run on it
  - Offers over 90+ products
    - Compute
    - Storage & Databases
    - Networking
    - Big Data
    - Cloud AI
    - Management Tools
    - Identity & Security
    - IoT
    - API Platform



Google Cloud Platform

- Compute services include
  - **App Engine (PaaS)**
    - PaaS to deploy Java, PHP, Node.js, Python, C#, .Net, Ruby and Go applications
  - **Compute Engine (IaaS)**
    - IaaS to run Microsoft Windows and Linux VMs
  - Kubernetes Engine (GKE) or GKE on-prem (part of Anthos)
    - **CaaS** based on Kubernetes
  - **Cloud Functions (FaaS)**
    - FaaS to run event-driven code written in JS (Node.js), Python or Go
  - Cloud Run
    - Execution environment based on Knative (a Kubernetes-based platform to deploy and manage modern serverless workloads)

Google Cloud Platform

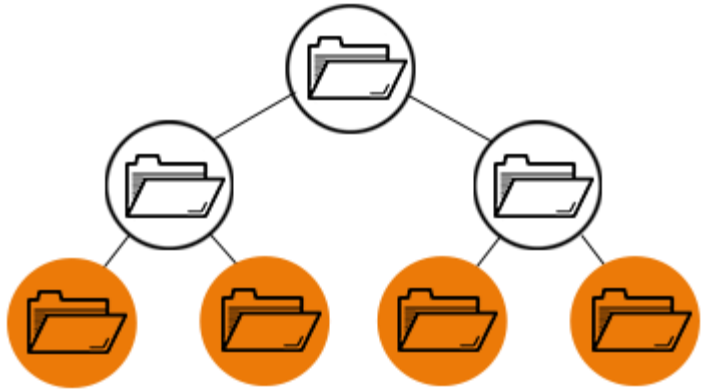


- Storage services (& DBs) include
  - **Cloud Storage**
    - **Object** storage with integrated edge caching to store unstructured data
  - **Cloud SQL**
    - DBaaS based on MySQL and PostgreSQL
  - **Cloud Bigtable**
    - Managed NoSQL database service
  - Cloud Spanner
    - Horizontally scalable, strongly consistent, relational database service (a NewSQL DB by Google)
  - Cloud Datastore
    - NoSQL database for web and mobile applications
  - Persistent Disk
    - **Block** storage for Compute Engine virtual machines
  - Cloud MemoryStore
    - Managed in-memory data store based on Redis
  - Local SSD
    - High performance, transient, local block storage
  - Filestore
    - High performance **file** storage for Google Cloud users

File storage, block storage or object storage?

- Storage formats that hold, organize, and present data in different ways – each with their own capabilities and limitations
  - File storage
    - Organizes and represents data as a **hierarchy of files in folders**
    - The oldest and most widely used data storage system for direct and network-attached storage systems
    - The one that you are used to
  - Downsides

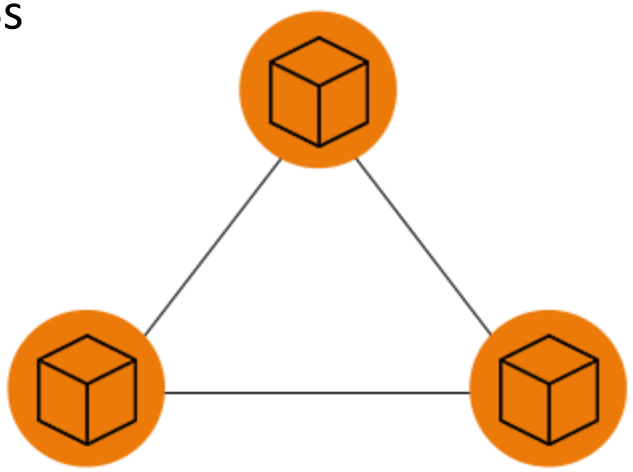
    File-based storage systems must scale by adding more systems, rather than scale up by adding more capacity



File storage, block storage or object storage?

- Storage formats that hold, organize, and present data in different ways – each with their own capabilities and limitations
  - Block storage
    - **Chops data into blocks** (evenly size) and stores them as separate pieces
    - Blocks can even be stored in different envs (e.g., Linux and Windows)
    - Efficient, reliable, fast, used for big transactions and huge DBs
  - Downsides

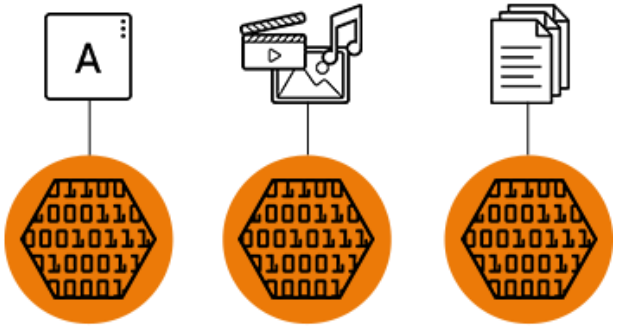
    Expensive. Limited capability to handle metadata, normally dealt with in the application or database level (one extra thing to worry about)



File storage, block storage or object storage?

- Storage formats that hold, organize, and present data in different ways – each with their own capabilities and limitations
  - Object storage
    - Manages data and links it to associated metadata (requires/uses an HTTP API)
    - Flat structure, data is broken into pieces (objects) and spread out among hardware
      - Kept in **a repository**, **each object has a unique id (or url)** to find it and **metadata**
    - Good for static data, scalable, cost efficient
  - Downsides

    Objects **cannot be modified** (written completely at once).
    Writing objects is slow, not suited for traditional databases.



File storage, block storage or object storage?

- Storage formats that hold, organize, and present data in different ways – each with their own capabilities and limitations
  - Block vs object storage

	OBJECT STORAGE	BLOCK STORAGE
PERFORMANCE	Performs best for big content and high stream throughput	Strong performance with database and transactional data
GEOGRAPHY	Data can be stored across multiple regions	The greater the distance between storage and application, the higher the latency
SCALABILITY	Can scale infinitely to petabytes and beyond	Addressing requirements limit scalability
ANALYTICS	Customizable metadata allows data to be easily organized and retrieved	No metadata

Google Cloud Platform

- Networking services include
  - **Virtual Private Cloud** (**VPC**)
    - For managing the software defined network of cloud resources
  - **Cloud Load Balancing**
    - Software-defined, managed service for load balancing the traffic
  - Cloud Armor
    - Web application firewall (against DDoS attacks)
  - Cloud Content Delivery Network (**CDN**)
    - Based on Google's globally distributed edge points of presence
  - Cloud **Interconnect**
    - Service to create a **hybrid cloud** (connect a data center with Google Cloud Platform)
  - **Cloud DNS**
    - Managed, authoritative DNS service
  - Network **Service Tiers**
    - Premium network tier for higher-performing network

Google Cloud Platform

- Big Data services include
  - **BigQuery**
    - Scalable, managed enterprise data warehouse for analytics
  - Cloud Dataflow
    - Managed service based on Apache Beam for stream and batch data processing
  - Cloud Dataproc
    - Big data platform for running **Apache Hadoop and Apache Spark** jobs
  - Cloud Composer
    - Managed workflow orchestration service built on Apache Airflow
  - **Cloud Datalab**
    - Tool for data exploration, analysis, visualization and machine learning (fully managed **Jupyter** Notebook service)
  - Cloud Dataprep
    - Data service based on Trifacta to visually explore, clean, and prepare data for analysis
  - **Cloud Pub/Sub**
    - Scalable event ingestion service based on message queues
  - **Cloud Data Studio**
    - Business intelligence tool to visualize data through dashboards and reports

Google Cloud Platform

- Cloud AI services include
  - Cloud **AutoML**
    - Service to train and deploy custom ML models
  - Cloud **TPU**
    - Accelerators used by Google to train ML models
  - Cloud **Machine Learning Engine**
    - Managed service for training and building ML models based on mainstream frameworks
  - Cloud Job Discovery
    - Service based on Google's search and machine learning capabilities for the recruiting ecosystem
  - Dialogflow Enterprise
    - Development environment for building conversational interfaces
  - Cloud **Natural Language**
    - Text analysis service based on Google Deep Learning models
  - Cloud **Speech-to-Text, Text-to-Speech and Translation API**
    - TTS / STT and translation conversion services using ML
  - **Cloud Vision** API and Video Intelligence
    - Image analysis and video analysis services based on ML

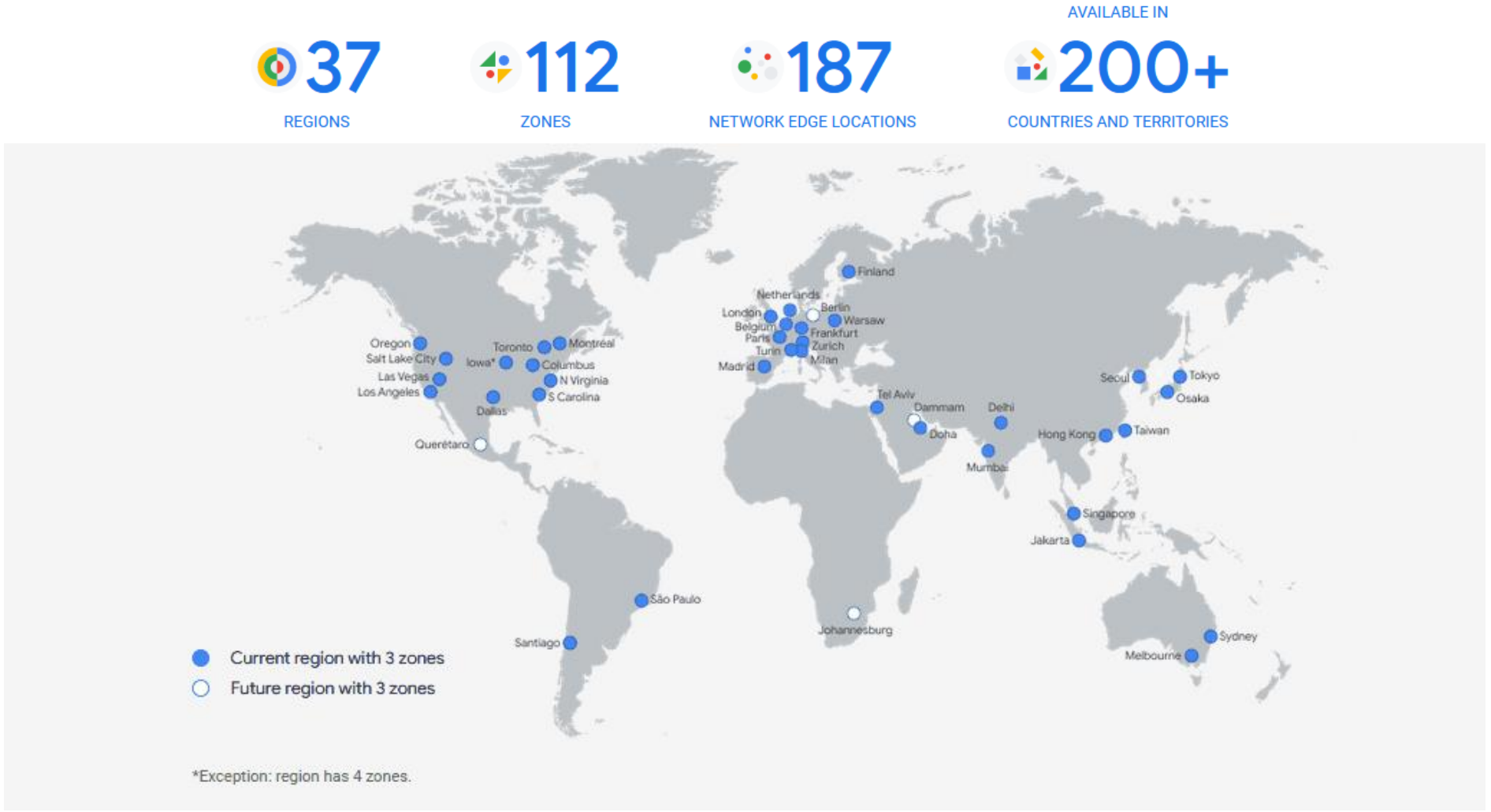
Google Cloud Platform

- Many other services are available
  - Management Tools
    - Stackdriver (monitoring, logging, and diagnostics)
    - Cloud Deployment Manager (using YAML, Python or Jinja2 templates)
    - Cloud **Console**, Cloud **Shell**, Cloud Mobile App (to manage GPC resources)
    - Cloud **APIs** (to programmatically access GPC resources)
  - Identity and Security
    - Cloud Identity (Single sign-on service), Cloud IAM, Cloud Identity-Aware Proxy, Cloud Data Loss Prevention API, Security Key Enforcement, Cloud Key Management Service, Cloud Resource Manager, Cloud Security Command Center, Cloud Security Scanner, Access Transparency, VPC Service Controls (perimeters), …
  - IoT services, API services (maps, analytics, monetization , …)

[Compute Engine API]
[Ansible GCP Guide]

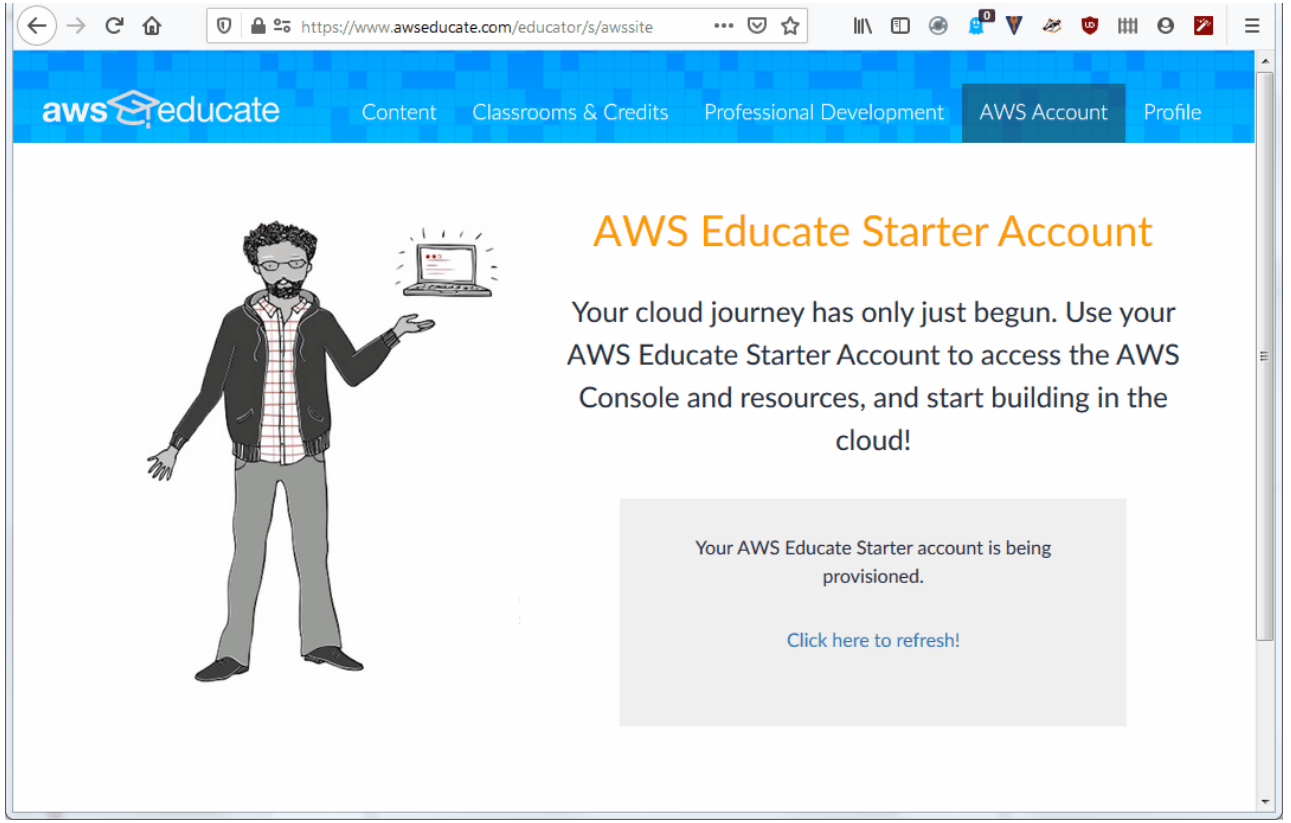
COMING SOON! Google Cloud will continue expanding into the following regions: Berlin (Germany), Dammam (Kingdom of Saudi Arabia), Querétaro (Mexico), Malaysia, Thailand, New Zealand, Greece, Norway, South Africa, Austria and Sweden.

Google Cloud Platform (2023)



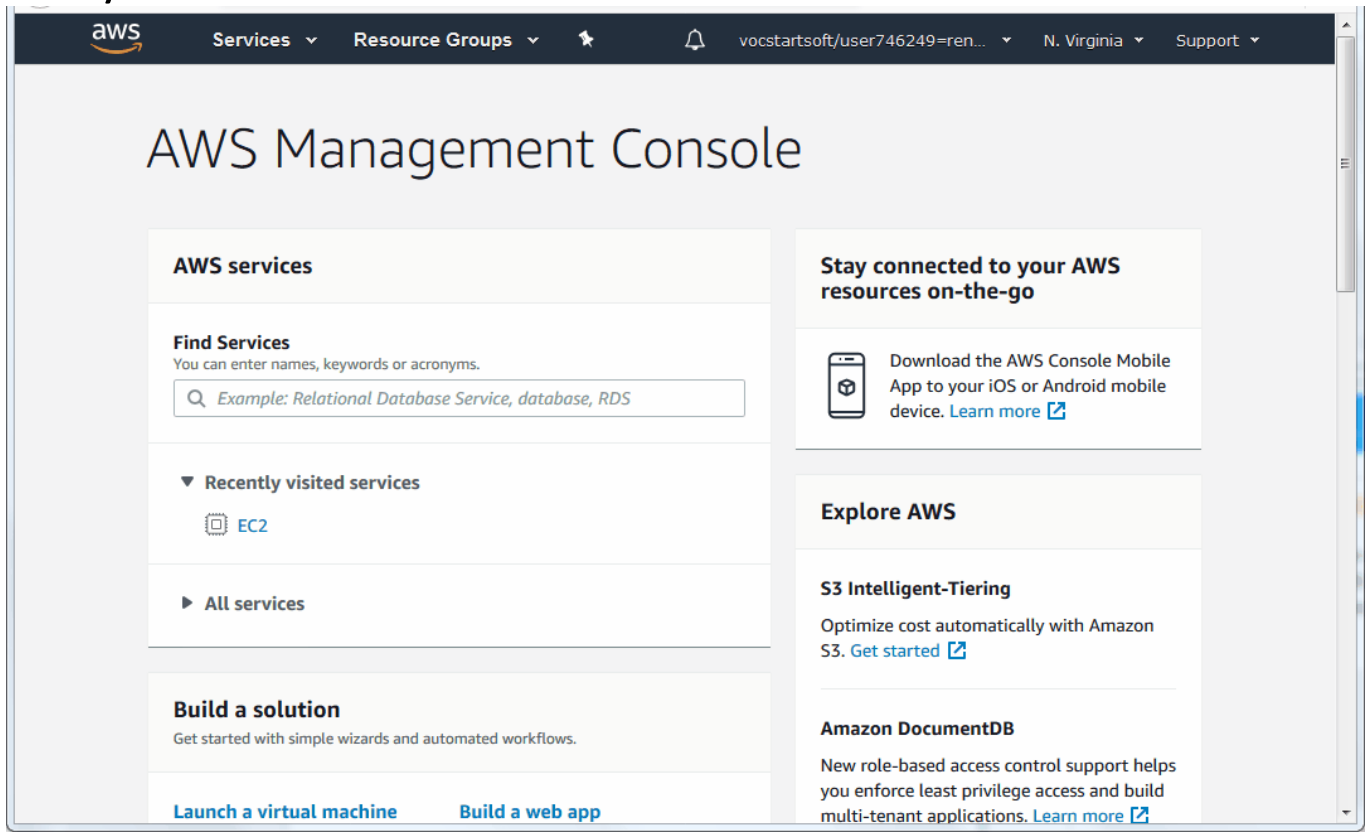
Amazon Web Services (AWS - 2020)

- AWS provides IaaS, PaaS and serverless solutions
  - Biggest / most used cloud provider today
  - Offers over 200+ products
    - Computing
    - Storage and databases
    - Networking
    - Deployment
    - Analytics
    - Artificial Intelligence
    - IoT
    - …



Amazon Web Services (AWS)

- AWS provides IaaS, PaaS and serverless solutions
  - Biggest / most used cloud provider today
  - Offers over 175+ products
    - Computing
    - Storage and databases
    - Networking
    - Deployment
    - Analytics
    - Artificial Intelligence
    - IoT
    - …



Amazon Web Services (AWS)

- AWS Developer Tools
  - AWS **Web Console**
    - Simple web interface for Amazon Web Services
  - AWS **Command Line Tool**
    - CLI to manage and automate AWS services with scripts
  - AWS **Cloud9** Integrated Development Environment (IDE)
    - Write, run, and debug code on a Cloud IDE
  - AWS Tools and Software Development Kit (**SDK**)
    - Simplify coding with language-specific abstracted APIs for AWS services
  - AWS Cloud Development Kit (**CDK**)
    - Define cloud infrastructure using familiar programming languages (infrastructure as code)

Amazon Web Services (AWS)

- AWS Developer Tools services
  - Amazon Corretto
    - Production-ready distribution of OpenJDK
  - AWS CodeBuild
    - Build and Test Code
  - AWS CodeCommit
    - Store Code in Private Git Repositories
  - AWS CodeDeploy
    - Automate Code Deployment
  - AWS CodePipeline
    - Release Software using Continuous Delivery
  - AWS CodeStar
    - Develop and Deploy AWS Applications
  - AWS Device Farm
    - Test Android, iOS, and web apps on real devices in the AWS cloud
  - AWS X-Ray
    - Analyze and debug your applications

Amazon Web Services (AWS)

- Compute services include (1)
  - Amazon Elastic Compute Cloud (**EC2**)
    - Virtual Servers in the Cloud (IaaS)
  - Amazon **EC2 Auto Scaling**
    - Scale Compute Capacity to Meet Demand
  - AWS **Elastic Beanstalk**
    - Run and Manage Web Apps (PaaS)
  - AWS **Lambda**
    - Run your Code in Response to Events (FaaS)
  - Amazon Lightsail
    - Launch and Manage Virtual Private Servers

Amazon Web Services (AWS)

- Compute services include (2)
  - AWS Batch
    - Run Batch Jobs at Any Scale
  - AWS Outposts
    - Run AWS services on-premises
  - AWS Serverless Application Repository
    - Discover, Deploy, and Publish Serverless Applications
  - AWS Wavelength
    - Deliver ultra-low latency applications for 5G devices
  - VMware Cloud on AWS
    - Build a Hybrid Cloud without Custom Hardware

Amazon Web Services (AWS)

- Container services include
  - Amazon Elastic Container Registry (**ECR**)
    - Store and Retrieve Docker Images
  - Amazon Elastic Container Service (**ECS**)
    - Run and Manage Docker Containers
  - Amazon Elastic Kubernetes Service (**EKS**)
    - Run Managed Kubernetes on AWS
  - AWS Fargate
    - Run Containers without Managing Servers or Clusters (*serverless* compute)

Amazon Web Services (AWS)

- Storage services include (1)
  - Amazon Simple Storage Service (**S3**)
    - Scalable Storage in the Cloud (objects)
  - Amazon Elastic Block Store (**EBS**)
    - EC2 block storage volumes (blocks)
  - Amazon Elastic File System (**EFS**)
    - Fully managed file system for EC2 (files)
  - Amazon FSx for Lustre (1) and For Windows (2)
    1. High-performance file system integrated with S3
    2. Fully managed Windows native file system
  - Amazon **S3 Glacier**
    - Low-cost Archive Storage in the Cloud

Amazon Web Services (AWS)

- Storage services include (2)
  - AWS **Backup**
    - Centralized backup across AWS services
  - AWS Snow Family
    - Physical devices to migrate data into and out of AWS
  - AWS Storage Gateway
    - Hybrid Storage Integration
  - CloudEndure Disaster Recovery
    - Highly automated disaster recovery

Amazon Web Services (AWS)

- Database services include (DBaaS) (1)
  - Amazon Aurora
    - High Performance Managed Relational Database (proprietary, closed-source database engine)
  - Amazon Relational Database Service (**RDS**)
    - Managed Relational Database Service for MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB
    - Amazon also provides Amazon RDS on VMware to automate on-premises database management
  - Amazon **DynamoDB**
    - Managed NoSQL Database
  - Amazon DocumentDB (with MongoDB compatibility)
    - Fully managed document database
  - Amazon **ElastiCache**
    - In-memory Caching System

Amazon Web Services (AWS)

- Database services include (DBaaS) (2)
  - Amazon Managed Apache Cassandra Service
    - Managed Cassandra-compatible database
  - Amazon Neptune
    - Fully Managed Graph Database Service
  - Amazon Quantum Ledger Database (QLDB)
    - Fully managed ledger database
  - Amazon **Redshift**
    - Fast, Simple, Cost-effective Data Warehousing
  - Amazon Timestream
    - Fully managed time series database
  - AWS Database Migration Service
    - Migrate Databases with Minimal Downtime

Amazon Web Services (AWS)

- Security and compliance services include (1)
  - AWS RoboMaker
    - Develop, test, and deploy robotics applications
  - AWS Ground Station
    - Fully managed ground station as a service
  - AWS Identity & Access Management (**IAM**)
    - Manage User Access and Encryption Keys
  - Amazon Cognito
    - Identity Management for your Apps
  - Amazon Detective
    - Investigate potential security issues
  - Amazon GuardDuty
    - Managed Threat Detection Service
  - Amazon Inspector
    - Analyze Application Security
  - Amazon Macie
    - Discover, Classify, and Protect your Data

Amazon Web Services (AWS)

- Security and compliance services include (2)
  - AWS Artifact
    - On-demand access to AWS compliance reports
  - AWS **Certificate Manager**
    - Provision, Manage, and Deploy SSL/TLS Certificates
  - AWS CloudHSM
    - Hardware-based Key Storage for Regulatory Compliance
  - AWS Directory Service
    - Host and Manage Active Directory
  - AWS Firewall Manager
    - Central Management of Firewall Rules
  - AWS Key Management Service (**KMS**)
    - Managed Creation and Control of Encryption Keys
  - AWS Resource Access Manager
    - Simple, secure service to share AWS resources

Amazon Web Services (AWS)

- Security and compliance services include (3)
  - AWS Secrets Manager
    - Rotate, Manage, and Retrieve Secrets
  - AWS Security Hub
    - Unified security and compliance center
  - AWS **Shield**
    - DDoS Protection
  - AWS Single Sign-On
    - Cloud Single Sign-On (**SSO**) Service
  - AWS WAF
    - Filter Malicious Web Traffic

Amazon Web Services (AWS)

- Management and governance services include (1)
  - Amazon **CloudWatch**
    - Monitor Resources and Applications
  - AWS **Auto Scaling**
    - Scale Multiple Resources to Meet Demand
  - AWS Chatbot
    - ChatOps for AWS
  - AWS CloudFormation
    - Create and Manage Resources with Templates
  - AWS CloudTrail
    - Track User Activity and API Usage

Amazon Web Services (AWS)

- Management and governance services include (2)
  - AWS Compute Optimizer
    - Identify optimal AWS Compute resources
  - AWS Config
    - Track Resource Inventory and Changes
  - AWS Control Tower
    - Set up and govern a secure, compliant, multi-account environment
  - AWS Console Mobile Application
    - Access resources on the go
  - AWS License Manager
    - Track, manage, and control licenses

Amazon Web Services (AWS)

- Management and governance services include (3)
  - AWS **Management Console**
    - Web-based user interface
  - AWS Managed Services
    - Infrastructure operations management for AWS
  - AWS **OpsWorks**
    - Automate Operations with Chef and Puppet
  - AWS Organizations
    - Central governance and management across AWS accounts
  - AWS Personal Health Dashboard
    - Personalized View of AWS Service Health

Amazon Web Services (AWS)

- Management and governance services include (4)
  - AWS Service Catalog
    - Create and Use Standardized Products
  - AWS Systems Manager
    - Gain Operational Insights and Take Action
  - AWS Trusted Advisor
    - Optimize Performance and Security
  - AWS Well-Architected Tool
    - Review and improve your workloads

Amazon Web Services (AWS)

- Machine learning services include (1)
  - Amazon **SageMaker**
    - Build, Train, and Deploy Machine Learning Models at Scale
  - Amazon Augmented AI
    - Easily implement human review of ML predictions
  - Amazon CodeGuru (Preview)
    - Automate code reviews and identify expensive lines of code
  - Amazon Comprehend
    - Discover Insights and Relationships in Text
  - Amazon Elastic Inference
    - Deep learning inference acceleration
  - Amazon Forecast
    - Increase forecast accuracy using machine learning

Amazon Web Services (AWS)

- Machine learning services include (2)
  - Amazon Fraud Detector (Preview)
    - Detect more online fraud faster
  - Amazon Kendra
    - Reinvent enterprise search with ML
  - Amazon **Lex**
    - Build Voice and Text Chatbots
  - Amazon **Personalize**
    - Build real-time recommendations into your applications
  - Amazon Polly
    - Turn Text into Lifelike Speech
  - Amazon **Rekognition**
    - Analyze Image and Video

Amazon Web Services (AWS)

- Machine learning services include (1)
  - Amazon SageMaker Ground Truth
    - Build accurate ML training datasets
  - Amazon Textract
    - Extract text and data from documents
  - Amazon **Translate**
    - Natural and Fluent Language Translation
  - Amazon **Transcribe**
    - Automatic Speech Recognition
  - AWS **Deep Learning AMIs**
    - Quickly Start Deep Learning on EC2
  - AWS **Deep Learning Containers**
    - Docker images for deep learning

Amazon Web Services (AWS)

- Machine learning services include (1)
  - AWS DeepComposer
    - ML enabled musical keyboard
  - AWS **DeepLens**
    - Deep Learning Enabled Video Camera
  - AWS DeepRacer
    - Autonomous 1/18th scale race car, driven by ML
  - Amazon Inferentia
    - Machine learning inference chip
  - Apache **MXNet** on AWS
    - Scalable, High-performance Deep Learning
  - **TensorFlow** on AWS
    - Open-source Machine Intelligence Library

Amazon Web Services (AWS)

- Analytics services include (1)
  - Amazon Athena
    - Query Data in S3 using SQL
  - Amazon **CloudSearch**
    - Managed Search Service
  - Amazon **Elasticsearch** Service
    - Run and Scale Elasticsearch Clusters
  - Amazon EMR
    - Hosted Hadoop Framework
  - Amazon **Kinesis**
    - Work with Real-time Streaming Data

Amazon Web Services (AWS)

- Analytics services include (2)
  - Amazon QuickSight
    - Fast Business Analytics Service
  - AWS Data Exchange
    - Find, subscribe to, and use third-party data in the cloud
  - AWS Data Pipeline
    - Orchestration Service for Periodic, Data-driven Workflows
  - AWS Glue
    - Prepare and Load Data
  - AWS **Lake Formation**
    - Build a secure data lake in days

Amazon Web Services (AWS)

- Networking and Content Delivery services include (1)
  - Amazon Virtual Private Cloud (**VPC**)
    - Isolated Cloud Resources
  - Amazon **API Gateway**
    - Build, Deploy, and Manage APIs
  - Amazon **CloudFront**
    - Global Content Delivery Network
  - Amazon **Route 53**
    - Scalable Domain Name System
  - AWS PrivateLink
    - Securely Access Services Hosted on AWS

Amazon Web Services (AWS)

- Networking and Content Delivery services include (2)
  - AWS **App Mesh**
    - Monitor and control microservices
  - AWS Cloud Map
    - Application resource registry for microservices
  - AWS **Direct Connect**
    - Dedicated Network Connection to AWS (used for hybrid clouds)
  - Elastic Load Balancing (**ELB**)
    - Distribute incoming traffic across multiple targets

Amazon Web Services (AWS)

- Many additional services are available (1)
  - Game Tech, e.g.,
    - Amazon GameLift – Simple, Fast, Cost-effective Dedicated Game Server Hosting
  - End User Computing, e.g.,
    - Amazon AppStream 2.0 – Stream Desktop Applications Securely to a Browser
  - Customer Engagement, e.g.,
    - Amazon Connect – Cloud-based Contact Center
  - Business Applications, e.g.,
    - Alexa for Business – Empower your Organization with Alexa
    - Amazon WorkMail – Secure and Managed Business Email and Calendaring
  - AWS Cost Management, e.g.,
    - AWS Cost Explorer: Analyze Your AWS Cost and Usage

Amazon Web Services (AWS)

- Many additional services are available (2)
  - Mobile, e.g.,
    - AWS **Amplify** – Build and deploy mobile and web applications
  - Media Services , e.g.,
    - Amazon **Kinesis Video Streams** – Process and Analyze Video Streams
  - Internet of Things , e.g.,
    - AWS IoT Core – Connect Devices to the Cloud
  - Blockchain , e.g.,
    - Amazon Managed Blockchain – Create and manage scalable blockchain networks
  - Quantum Technologies , e.g.,
    - **Amazon Braket** – Explore and experiment with quantum computing
  - Satellite , e.g.,
    - **AWS Ground Station** – Easily control satellites and ingest data with fully managed Ground Station as a Service

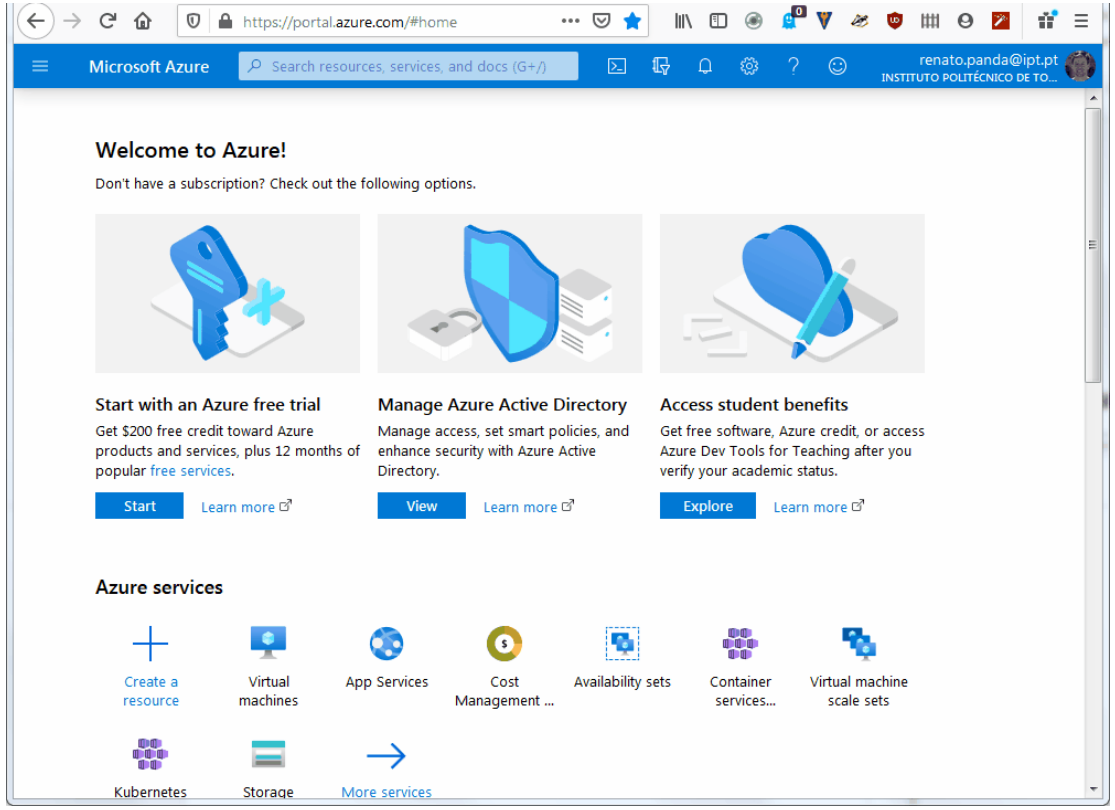
Amazon Web Services (AWS)

In 2020

24 Launched Regions Each with multiple Availability Zones (AZ's)	**3 Announced Regions**	**76 Availability Zones**	**1 Local Zone** For ultralow latency applications
2x More Regions With multiple AZ's than the next largest cloud provider	**245 Countries and Territories Served**	**97 Direct Connect Locations**	**216 Points of Presence** 205 Edge Locations and 11 Regional Edge Caches

Microsoft Azure (2020)

- Azure provides IaaS, PaaS and SaaS services
  - Supports both Microsoft-specific and third-party software and systems
  - Microsoft lists over 600 Azure services
    - Compute
    - Containers
    - Databases
    - Storage
    - Windows Virtual Desktop
    - IoT
    - Analytics
    - AI
    - …



Microsoft Azure

- Compute services include
  - **Virtual Machines** (**IaaS**)
    - Includes Windows and Linux virtual machines
  - Virtual Machine **Scale Sets**
    - Manage and scale up to thousands of VMs
  - **Azure Functions (FaaS)**
    - Process events with *serverless* code
  - **App Service (PaaS)**
    - Supports .NET, .NET Core, Node.js, Java, PHP, Ruby, or Python
  - Web Apps, API Apps, Mobile Apps (PaaS)
    - App Services now replaces all Mobile, Api and Web Apps flavors as a single app framework. Still these remain available and basically resolve into an App Service App
  - **Azure Kubernetes Service (AKS)**
    - Highly available, secure, and fully managed Kubernetes service
  - Container Instances
    - AKS with a different name?

Microsoft Azure

- Storage services include
  - Storage Accounts
    - **Azure Files** – Fully managed cloud file shares, encrypted at rest, in-transit using SMB 3.0 and HTTPS
    - **Azure Disk Storage** – Persistent, secured disk storage for Azure virtual machines
    - **Blob Storage** – Massively-scalable object storage for unstructured data
    - Data Lake Storage – Secure, massively scalable data lake storage (using Blob storage)
    - **Azure Archive Storage** – Low-cost storage for archival (rarely accessed data)
    - Queue Storage – Simple, cost-effective, durable message queueing for large workloads
    - Table Storage – NoSQL key-value store (with Cosmos DB API access / next slide)
  - Azure Backup
    - Simplified one-click backup support for things such as SQL database and virtual machines running in Azure

Microsoft Azure

- Database services include
  - **Azure SQL Database**
    - Intelligent, scalable, cloud DB service that provides the broadest SQL Server engine compatibility
  - SQL Server on Virtual Machines
  - **Azure Cache for Redis**
    - Fully managed, open source–compatible in-memory data store to power fast, scalable applications
  - **Azure Database** for PostgreSQL (also MySQL and MariaDB)
    - Managed PostgreSQL/MySQL/MariaDB database service for app developers
  - **Azure Cosmos DB**
    - Fully managed NoSQL database service with automatic and elastic scaling of throughput and storage
  - Azure API for FHIR
    - Data store following HL7 Fast Healthcare Interoperability Resource (FHIR) standard format

Microsoft Azure

- Network services include
  - **Content Delivery Network (CDN)**
    - Secure and reliable global content delivery and acceleration
  - **Azure DNS**
  - **Traffic Manager**
    - Route incoming traffic for high performance and availability (e.g., DNS load balancing based on geolocation)
  - **Load Balancer**
    - Deliver high availability and network performance to your applications
  - Application Gateway
  - Azure ExpressRoute
    - Dedicated private network fiber connections to Azure (e.g., hybrid scenarios)
  - Several others
    - Azure DDoS Protection, Network Watcher, Azure Firewall, Virtual WAN, Azure Front Door, Azure Private Link…

Microsoft Azure

- AI and ML services include (1)
  - Azure Bot Service
    - Intelligent, *serverless* bot service that scales on demand
  - Azure Databricks
    - Fast, easy, and collaborative Apache Spark-based analytics platform
  - Azure Cognitive Search
    - AI-powered cloud search service for mobile and web app development
  - Bing Autosuggest
    - Give your app intelligent autosuggest options for searches
  - Bing Spell Check
    - Detect and correct spelling mistakes in your app
  - Cognitive Services
    - Add smart API capabilities to enable contextual interactions
  - Computer Vision
    - Distill actionable information from images

Microsoft Azure

- AI and ML services include (2)
  - Bing Custom Search
    - An easy-to-use, ad-free, commercial-grade search tool that lets you deliver the results you want
  - Bing Entity Search
    - Enrich your experiences by identifying and augmenting entity information from the web
  - Bing Image Search
    - Search for images and get comprehensive results
  - Bing News Search
    - Search for news and get comprehensive results
  - Bing Visual Search
    - Get rich insights to help build compelling image applications on the device of your choice.
  - Bing Web Search
    - Get enhanced search details from billions of web documents
  - Bing Video Search
    - Search for videos and get comprehensive results

Microsoft Azure

- AI and ML services include (3)
  - **Content Moderator**
    - Automated image, text, and video moderation
  - Custom Vision
    - Easily customize your own state-of-the-art **computer vision models** for your unique use case
  - Data Science Virtual Machines
    - Rich pre-configured environment for AI development
  - Face
    - Detect, identify, analyze, organize, and **tag faces in photos**
  - **Azure Machine Learning (ML)**
    - Bring AI to everyone with an end-to-end, scalable, trusted platform with experimentation and model management
  - Machine Learning Studio
    - Easily build, deploy, and manage predictive analytics solutions
  - Microsoft Genomics
    - Power **genome sequencing** & research insights

Microsoft Azure

- AI and ML services include (4)
  - Language Understanding
    - Teach your apps to **understand commands** from your users
  - QnA Maker
    - Distill information into conversational, easy-to-navigate answers
  - Speaker Recognition Preview
    - Use speech to **identify and verify individual speakers**
  - Speech Translation
    - Easily integrate **real-time speech translation** to your app
  - **Speech to Text**
    - Convert spoken audio to text for more natural interactions
  - Text Analytics
    - Easily **evaluate sentiment and topics** to understand **what users want**
  - **Text to Speech**
    - Convert text to speech to create more natural, accessible interfaces
  - **Translator Text**
    - Easily conduct machine translation with a simple REST API call

Microsoft Azure

- Many other services are provided
  - Azure **Blockchain** Service and Workbench
    - Simplifies the formation, management, and governance of consortium blockchain networks
  - Dev Tools
    - Visual Studio, SDKs, CLI, Azure Pipelines, App Configuration Storage, Repos, Boards, Monitors and other DevOps tools
  - Identity
    - Azure **Active Directory**
      - Sync with on-premises, enables single sign-on
  - Integration
    - Service Bus
    - Event Grid
    - API Management
  - IoT services (e.g., Azure RTOS, Notification Hubs, Stream Analytics)
  - Management and Governance (i.e., services to simplify and automate cloud management)
  - Media services (e.g., encode, decode, live and on-demand streaming)
  - **Windows Virtual Desktop**
    - Comprehensive desktop and app virtualization service running in the cloud to enable secure remote work

Microsoft Azure (2020)

Up to 1.6 Pbps of bandwidth in a region

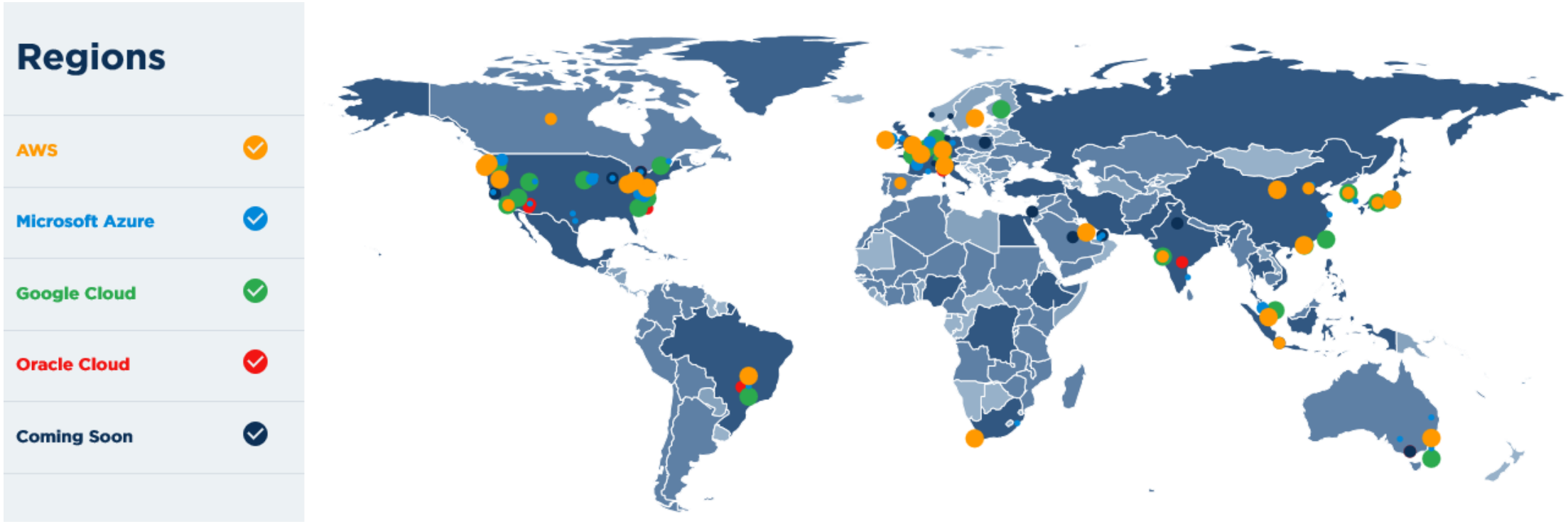
58 regions worldwide **140** available in 140 countries

* Three Azure Government Secret region locations undisclosed

https://interconnected.blog/data-center-coverage-aws-alibaba-azure-gcp/

https://cloudcheckr.com/zonecheckr/

Cloud Providers Comparison (2020)



	Multi-AZ Regions in China	Multi-AZ Regions in North America	Multi-AZ Regions in Europe	Multi-AZ Regions Everywhere Else	Single-AZ Regions	Total Regions
Alibaba	9	2	2	7	1	21
AWS	2	7	5	8	0	22
Azure	0	4	4	2	47	57
GCP	0	7	6	8	0	21

Cloud Providers Comparison

- Many of the services are available in every provider
  - Just under different names and sets of APIs

Service	Google Cloud Platform	Amazon Web Services	Microsoft Azure
IaaS	Google Compute Engine	Amazon EC2	Azure Virtual Machines
PaaS	Google App Engine	AWS Elastic Beanstalk	Azure App Services
FaaS	Google Cloud Functions	AWS Lambda	Azure Functions
NoSQL DB	Google Cloud Bigtable	Amazon DynamoDB	Cosmos DB
Relational DB	Google Cloud SQL	Amazon RDS	Azure Database
Data warehouse	Google BigQuery	Amazon Redshift	Microsoft Azure DataWarehouse
Object storage	Google Cloud Storage	Amazon S3	Azure Blob Storage

Cloud Providers Comparison

- Gartner Report (ref)
  - Magic Quadrant for Cloud Infrastructure as a Service, Worldwide (2019)

The distinctions among providers are apparent in the market for cloud IaaS in terms of worldwide enterprise **adoption**, **capabilities** and service **availability**. Infrastructure and operations leaders should **evaluate providers** with broad capabilities and a positive track record **for customer success**.

Cloud Providers Comparison

- Garner Report provides a review of the biggest cloud providers
  - For each provider, it includes
    - A Summary of the company
    - List of services offered by the provider
    - Datacenter locations
    - Adoption profile (customers that typically adopt a provider's platform)
    - Recommended uses (circumstances under which the provider is recommended)
    - Strengths and Cautions
  - Want the details? Read the report ☺

Cloud Providers Comparison

- Amazon Web Services (from Gartner Report)

  Recommended uses:

  All use cases that run well in a **virtualized environment**. Applications that are potentially challenging to virtualize or run in a multitenant environment — including **highly secure applications**, **strictly compliant** or complex enterprise applications (such as SAP business applications) — require special attention to architecture.

Cloud Providers Comparison

- Google Cloud Platform (from Gartner Report)

  Recommended uses:

  Big data and other **analytics** applications, **machine learning** projects, cloud-native applications, or other applications optimized for **cloud-native** operations.

Cloud Providers Comparison

- Microsoft Cloud (from Gartner Report)

  Recommended uses:

  All use cases that run well in a virtualized environment, particularly for Microsoft-centric organizations.

Cloud Providers Comparison

- Oracle Cloud Infrastructure (from Gartner Report)

  Recommended uses:

  OCI is best suited for enterprises requiring **cloud IaaS for Oracle applications** and for applications that require an **Oracle Database**.

Cloud Providers Comparison

- Alibaba Cloud (from Gartner Report)

  Recommended uses:

  Cloud-first digital business workloads for customers that are (1) **based in China or Southeast Asia** and wish to leverage Alibaba Cloud's relationship with its parent company, or (2) **need to locate cloud infrastructure in China**.

Cloud Providers Comparison

- IBM Cloud (from Gartner Report)

  Recommended uses:

  IBM outsourcing deals that use bare-metal servers as the hosting platform, where the customer has a need for **supplemental basic cloud IaaS**. The infrastructure may also be used as a component of applications built using the **IBM Cloud PaaS capabilities**. It should also be considered in circumstances that require both API control over scalable infrastructure and bare-metal servers in order to meet requirements for performance, regulatory compliance or software licensing.

  Disclaimer: IBM did not respond to requests for supplemental information or for a review of the draft contents of this research. Therefore, Gartner analysis is based on other credible sources, including public information.

If cloud computing is soooo good…

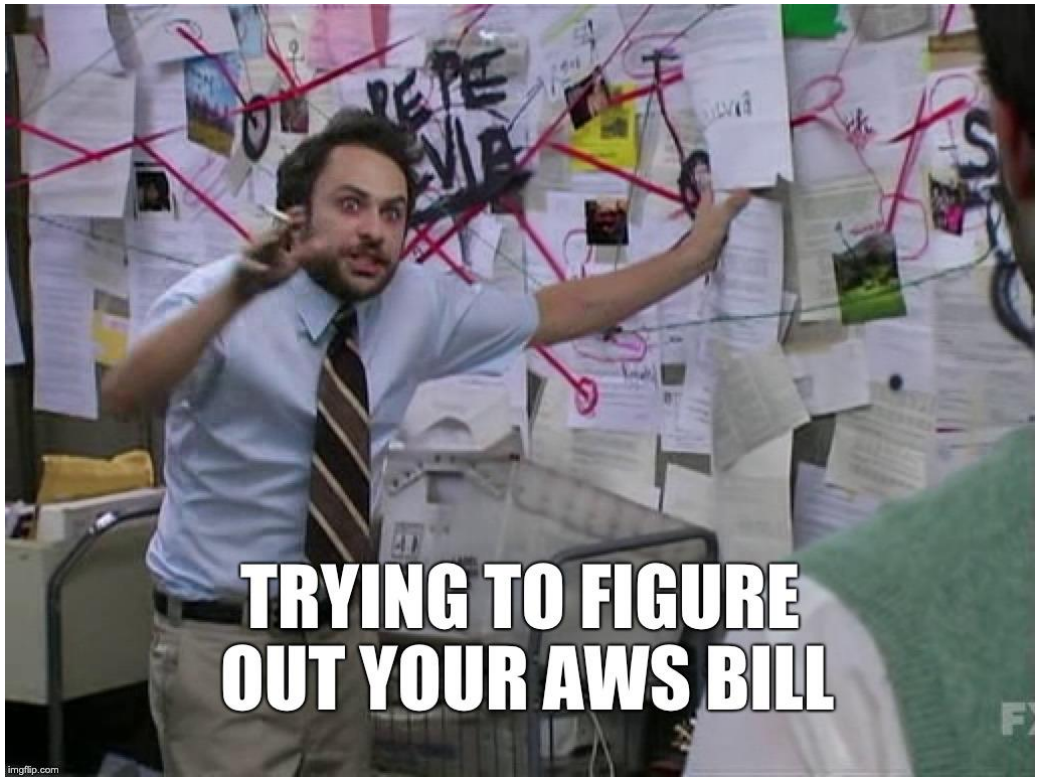
Problems with cloud computing

Where did my money go?

- Cloud computing is often touted as a “cost saver”
  - Variable pricing can appear attractive
  - Turning instances on/off is great

Where did my money go?

- In reality this may not be the cheapest option for your problem
  - Individual charges for everything that happens in your environment start adding up
  - Elasticity eliminates cost predictability
    - Much harder to budget

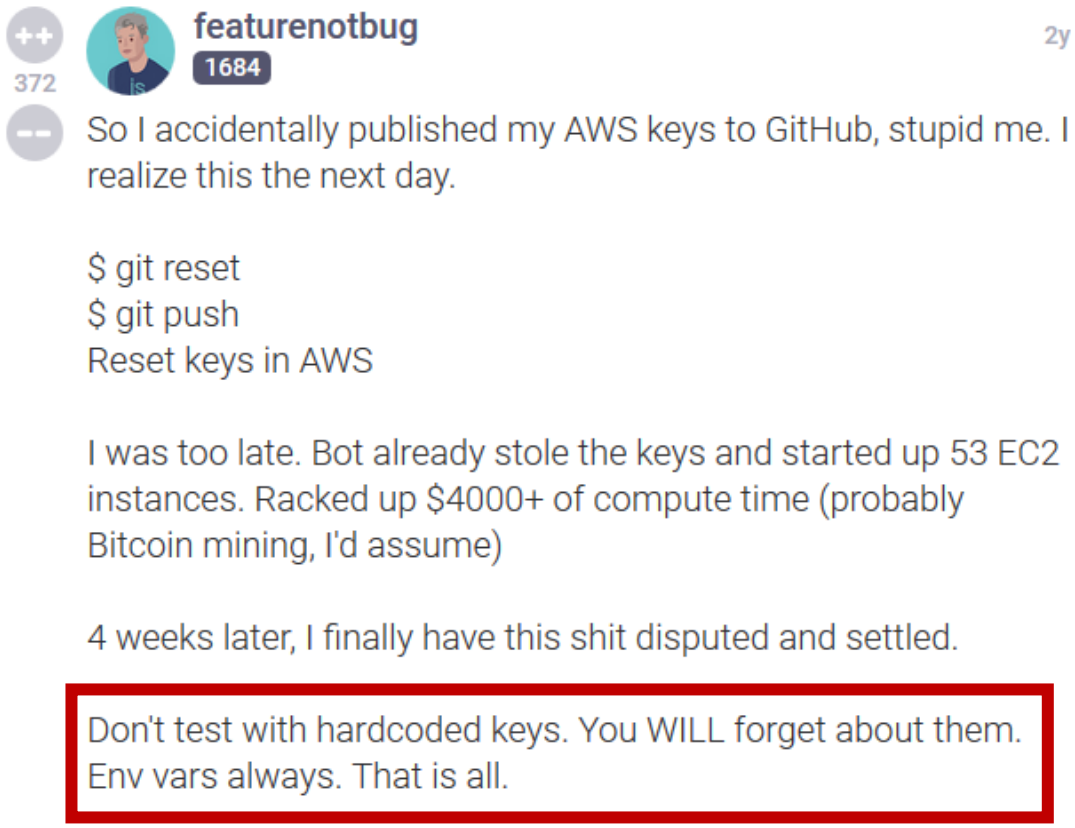


Where did my money go?

- In reality this may not be the cheapest option for your problem
  - Individual charges for everything that happens in your environment start adding up
  - Elasticity eliminates cost predictability
    - Much harder to budget
- Make sure that you
  - Budget and set your limits correctly
  - Implement scaling as efficiently as possible

Where did my money go?

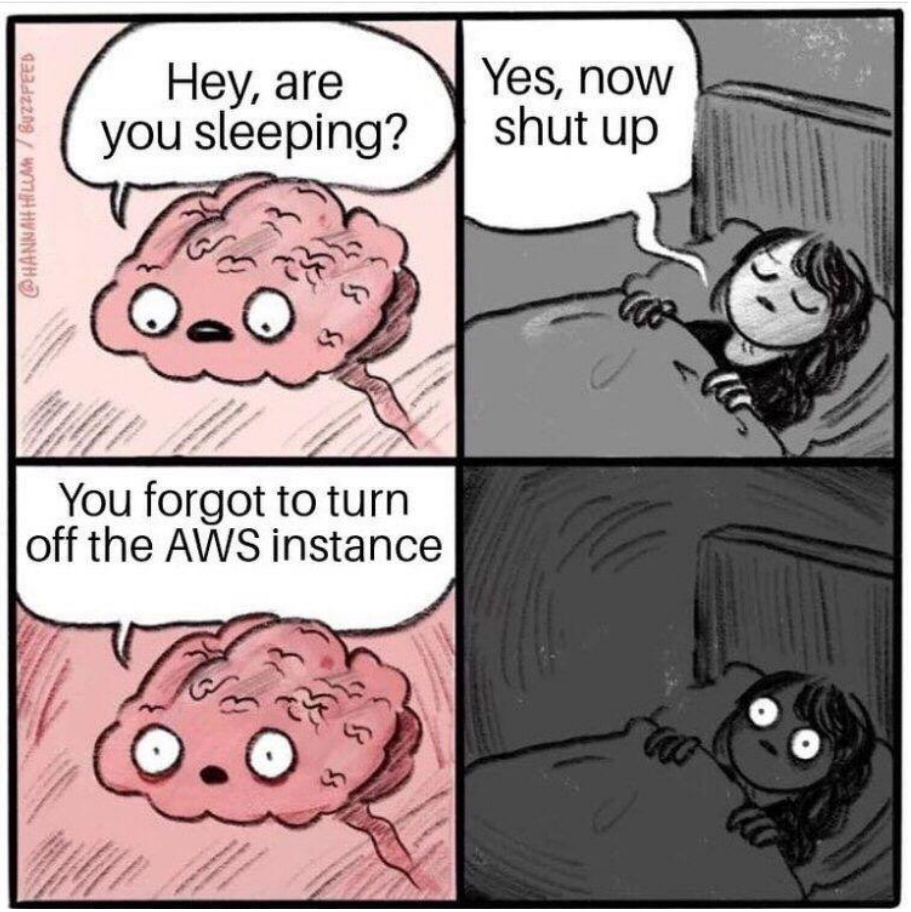
- Also… make sure from the start that it is hard for team members to make big mistakes!



Where did my money go?

- Also… make sure from the start that it is hard for team members to make big mistakes!

- Do not let your unused instances running
  - This is valid for this course too!



Uptime and Service Level Agreement (SLA)

- Cloud providers have high SLAs
  - AWS guarantees 99.99% uptime
    - … or less than 53min per year
  - When that fails, “you will be eligible to receive a Service Credit”
    - Up to 100% if it goes < 95%
      - 95% = 18.25 days per year

- What are the costs if **your product** goes offline for 15 days?
  - Get your requirements straight
  - Multi-region redundancy if needed

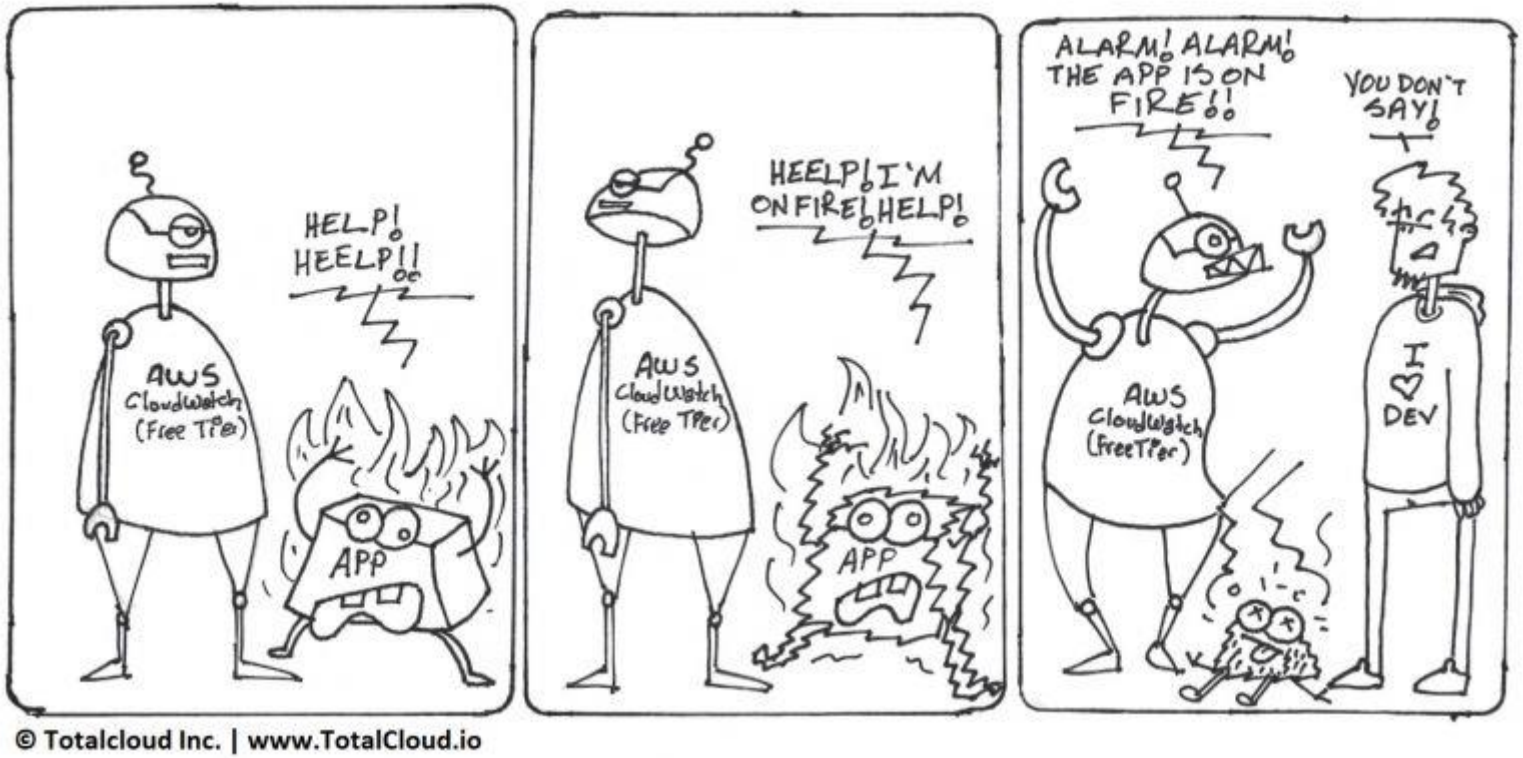
What about performance?

- Cloud providers normally guarantee capacity, not performance
  - E.g., CPUs, memory, disk size, etc.
  - Several reasons such as opaqueness, multi-tenancy, where your data lives (network latency) and others may decrease performance of your instances

- The Top 5 AWS EC2 Performance Issues
  - Unpredictable Elastic Block Store (EBS) Disk I/O
  - EC2 Instance EC2 Compute Unit (ECU) Mismatch and Stolen CPU
  - Running out of EC2 Instance Memory
  - ELB Load Balancing Traffic Latency
  - AWS Maintenance and Service Interruptions

“Many services have a free tier!”

- Which probably is not useful at all in production / real-life scenarios



Cloud Providers Certifications & Training

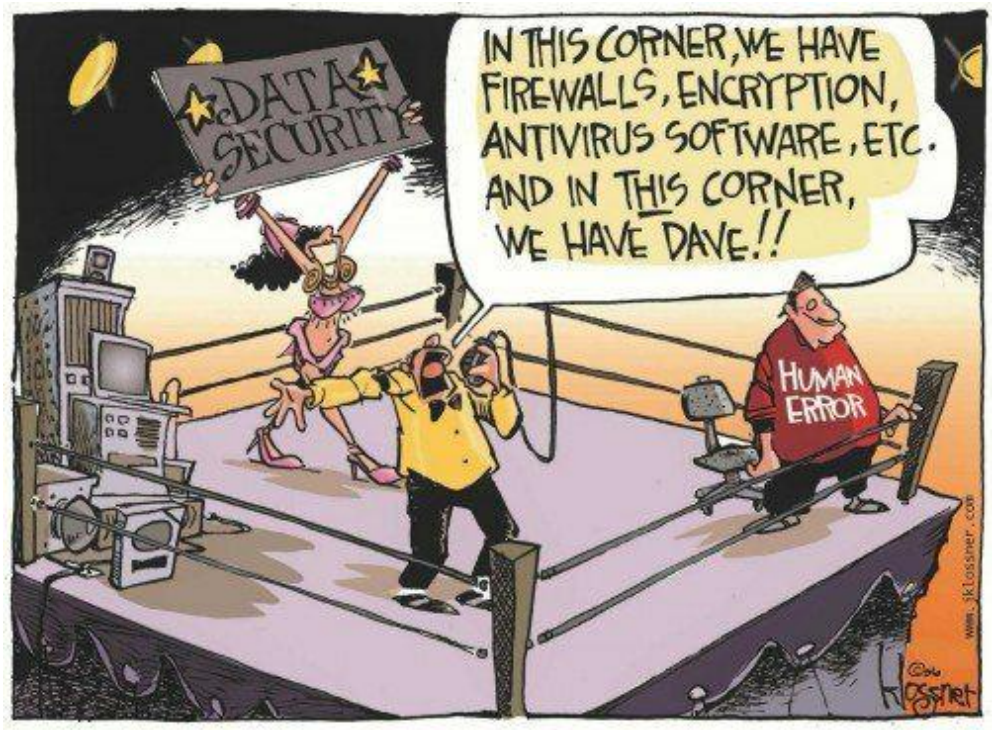
- The number of services you need to grok for full AWS certification

Cloud Computing Random Bits

- Several other questions remain
  - What about security when using the cloud?
  - Where is my data saved?
    - Data considerations (Where is my app data saved? Law and possible issues?)
  - What if my field requires auditing and compliance with specific norms?
  - How can I log and monitor an infrastructure that is ephemeral?

Cloud Computing Security Considerations

- Cloud security is the protection of data, applications, and infrastructures involved in cloud computing
  - Many concerns are similar to any on-premises solution
    - Unauthorized data exposure and leaks
    - Weak access controls
    - Susceptibility to attacks
    - Availability disruptions



Cloud Computing Security Considerations

- Cloud security is the protection of data, applications, and infrastructures involved in cloud computing
  - Like any other solution, we need measures to
    - Know that the data and systems are safe
    - Can see the current state of security
    - Know immediately if anything unusual happens
    - Can trace and respond to unexpected events

Cloud Computing Security Considerations

- Cloud security also has some differences
  - Infrastructure now exists somewhere between amorphous resources sent through the internet and a physical server
  - **Dissolving perimeters**
    - **Traditional** environments usually control access using **a perimeter security model**
    - Cloud envs are **highly connected**, it is easier to bypass traditional perimeter defenses
    - Insecure APIs, weak identity and credentials management, account hijacks, and malicious insiders may pose threats to the system and data

Cloud Computing Security Considerations

- Cloud security also has some differences
  - Infrastructure now exists somewhere between amorphous resources sent through the internet and a physical server
  - **Dissolving perimeters**
    - Preventing unauthorized access in the cloud requires **shifting to a data-centric approach**
      - Encrypt the data
      - Strengthen the authorization process
      - Require strong passwords and 2 factor authentication
      - **Build security into every level**

Cloud Computing Security Considerations

- Cloud security also has some differences
  - Infrastructure know exists somewhere between amorphous resources sent through the internet and a physical server
  - **Dissolving perimeters**
    - Shifting to a data-centric approach
  - **Everything is now in software**
    - Data while at rest and in transit should be protected
  - **Sophisticated threat landscape**
    - Including ones targeting the cloud and used computing stack

Cloud Computing Security Considerations

- Cloud security is a shared responsibility
  - Regardless of the deployment model, you are responsible for securing your own space within that cloud
    - Use trusted software
    - Understand compliance requirements
    - Correctly manage lifecycles (avoid zombie/outdated instances)
    - Considering portability
    - Continuous monitoring
    - Choosing the right people
  - **Are public clouds secure?**
    - Mitigate risk with hybrid cloud

Auditing in the Cloud

"Two thirds of the earth's surface is covered with water, the other third is covered with auditors from headquarters"
—Norman R. Augustine

- Historically, data has been stored behind corporate firewalls
  - It was up to the company to secure the perimeter, harden the infrastructure, and secure the databases
    - Auditors could come on-site and inspect the processes to make their assessments
- Why do I need auditors?
  - They validate that their clients adequately address specific requirements of a given set of constraints as defined by a governing set of laws
  - Many different regulations exist nowadays (e.g., GDPR)

Auditing in the Cloud

- On-premises, an auditor can sit down with a client and map personnel and physical infrastructure to the different controls and processes that are to be audited
  - They have access to physical data centers

- This is not easy to achieve in cloud environments
  - Multitenant environment, multiple customers share compute resources
  - Auditors of one tenant cannot access the infrastructure because it has an obligation to protect the rights of all of the other tenants
  - Major reason why companies build private clouds

Auditing in the Cloud

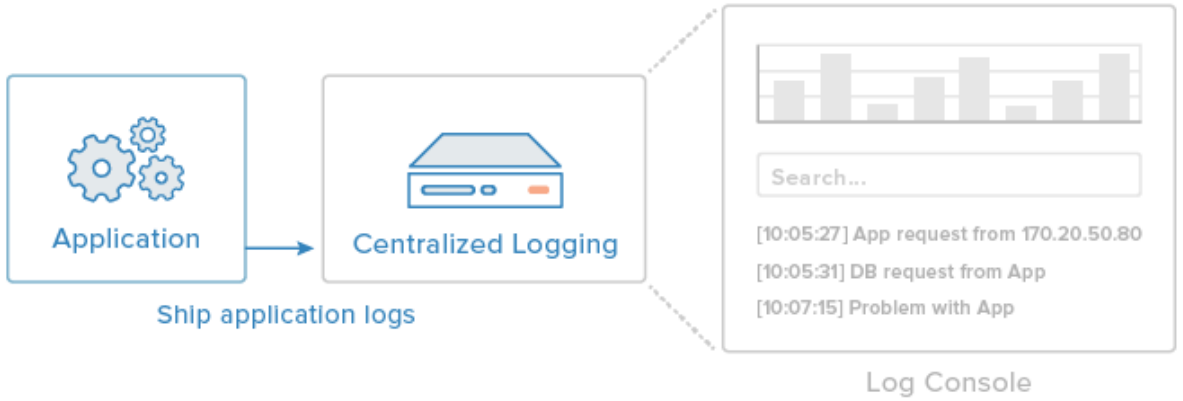
- That is why compliance is such a high priority for cloud providers
  - Many clients require compliance with specific regulations
  - In many cases, this makes it easier and more cost effective
    - E.g., GDPR compliance is already delivered by the cloud provider

- Some examples
  - AWS Compliance Programs
  - General Data Protection Regulation (GDPR) Center

Cloud Monitoring and Logging

- Cloud instances are ephemeral and numerous (due to scale out)
  - Do we want to log events in each instance?
  - How to monitor such distributed and dynamic infrastructure?

- A centralized logging strategy is mandatory
  - Direct logs to an isolated storage area
  - Standardize log formats



Cloud Monitoring and Logging

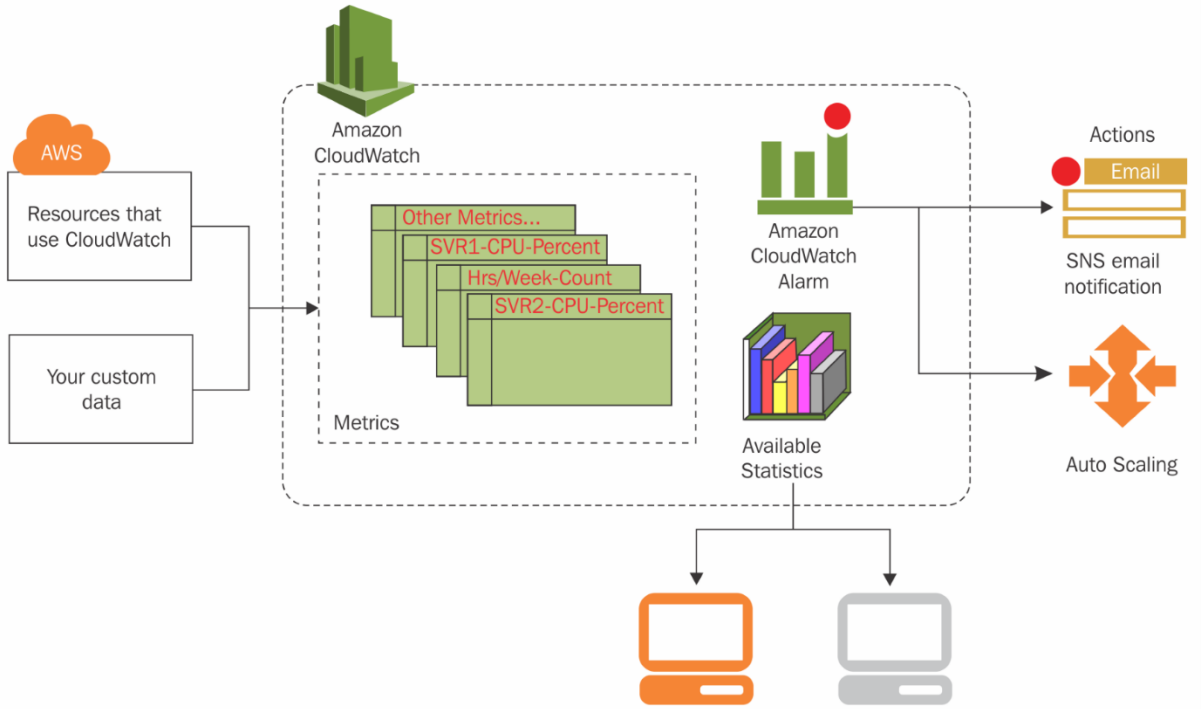
- Cloud instances are ephemeral and numerous (due to scale out)
  - Do we want to log events in each instance?
  - How to monitor such distributed and dynamic infrastructure?

- A centralized logging strategy is mandatory
  - Direct logs to an isolated storage area
  - Standardize log formats

- Remember ELK stack?

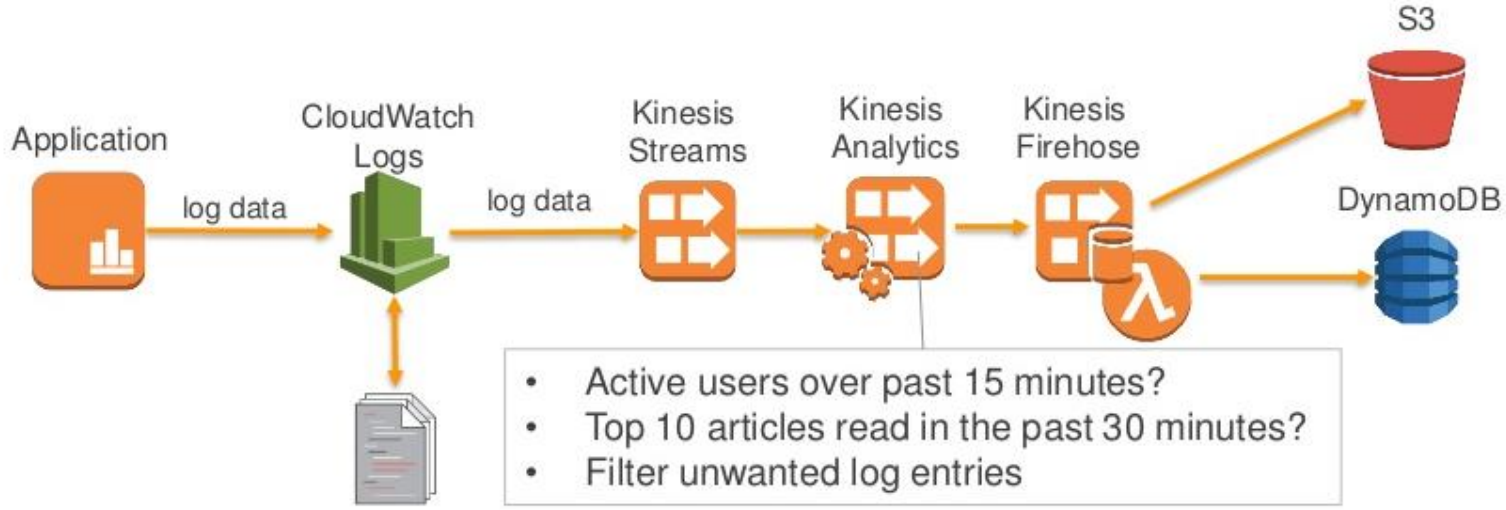
Cloud Monitoring and Logging

- Cloud providers normally have services that achieve this for you
  - AWS provides CloudWatch service
    - Monitor AWS resources in near real-time
    - Monitor custom metrics (app specific)
    - Monitor and store logs
    - Set alarms
    - Data visualization (plots, statistics)
    - Trigger actions on changes



Cloud Monitoring and Logging

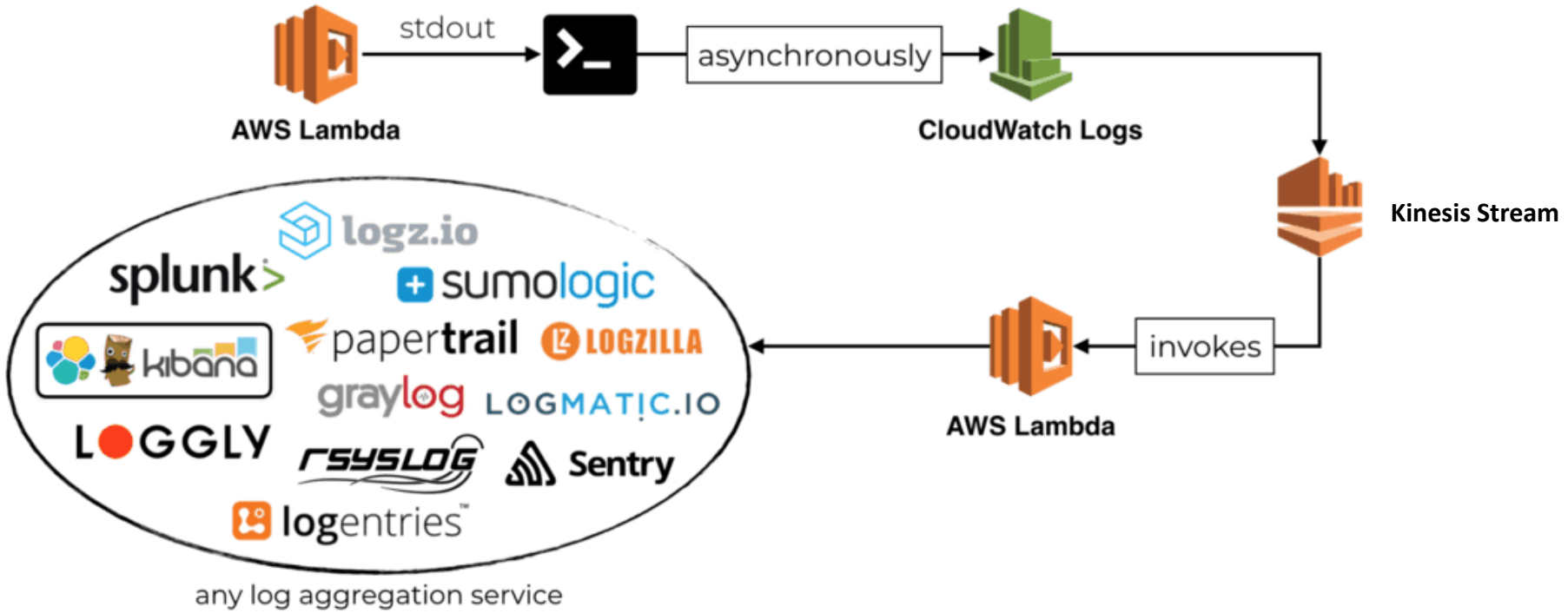
- Cloud providers normally have services that achieve this for you
  - It can be extended, e.g., with AWS Kinesis services



- Use CloudWatch Agent to send application logs to CloudWatch Logs
- Analyze stream with Kinesis Analytics application
- Persist raw log data to durable storage with Kinesis Firehose

Cloud Monitoring and Logging

- Cloud providers normally have services that achieve this for you
  - Or any other custom-made solution that fits your requirements



Cloud Computing Worst Practices

- Typical **misconceptions** when moving to the cloud include
  - Migrating existing applications to the cloud is a **simple** solution that drives **down** costs
    - Few legacy applications are good candidates to move to the cloud in their current architecture
  - Companies often take on cloud computing initiatives with **inflated expectations**
    - Due to impressive stories over the last decade (e.g., Instagram, Netflix)
  - Misinformed about cloud **security**
    - Some believe that cloud computing is **catastrophically insecure**, and data cannot be placed in a public cloud for any reason
    - Others believe that security is **taken care** of for them by the cloud vendors and deploy software and services with gaping security holes

Cloud Computing Worst Practices

- Typical **misconceptions** when moving to the cloud include
  - Selecting a **favorite vendor**, not an appropriate vendor
    - Less of a problem nowadays as the big players offer a very broad and similar set of services
  - Not designing for **failure**, considering outages and out-of-business scenarios
  - **Underestimating** the impacts of organizational change
    - Especially for already established organizations with infrastructure and IT staff
    - Cloud computing is not yet another “IT thing”, it has implications beyond IT, including security, business processes, accounting principles, HR, legal and so on since the data and services exist outside the enterprise

Cloud Computing Worst Practices

- Typical **misconceptions** when moving to the cloud include
  - **Skills** shortage
    - Enterprises often do not have the required **expertise** to build cloud-based solutions
    - Especially with established / stable IT teams with several years working in traditional client-server projects and on-premises systems administration
  - Misunderstanding customer **requirements**
    - IT people neglect the business side of the equation and build the cloud solution that is best for IT
  - **Unexpected** costs
    - Pay-as-you-go model only reduces the costs **if** the software is architected and managed in a way that optimizes the use of cloud services

Cloud Computing & Auto Scaling

- What really means “auto-scaling”?
  - Ensure you have the correct number of resources to handle current requests

- How?
  - Setup a **group** of similar instances
    - E.g., 5 web app nodes
  - Start some of those instances (**desired** capacity)
    - E.g., 2 nodes
  - **Monitor** some parameters in your instances
    - E.g., CPU load of the running instances
  - Define a set of **thresholds**
    - E.g., CPUmax = 0.8, CPUmin = 0.2, minInstances = 1, maxInstances = 5
  - **Start and stop** instances as needed to maintain the thresholds
    - E.g., CPUload = 0.95 for 60s, start an additional instance, register service and update LB

Cloud Computing & Auto Scaling

- How to achieve this?
  - Load balancer (obviously)
  - Set of VMs provisioned (or provision templates)
  - Mechanism to monitor active instances
    - E.g., simple SNMP, consul monitoring + telemetry service, Prometheus + actions
  - API to start and stop existing VMs
  - Service to monitor the instances and react based on a defined policy
    - E.g., simple script?

Cloud Computing & Auto Scaling

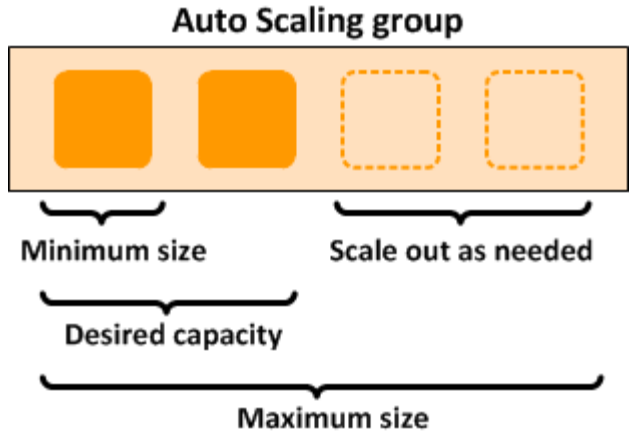
- Auto scaling uses a ***reactive*** decision approach for dealing with traffic scaling
  - Scaling only happens in response to real-time changes in metrics
  - Sometimes this is not enough

- Alternative auto scaling decision approaches
  - **Scheduled** auto scaling approach
    - Changes made at specific times of day
  - **Predictive** auto scaling
    - Using predictive analytics, based on various sources o data to predict future usage
      - E.g., recent usage trends and historical usage data

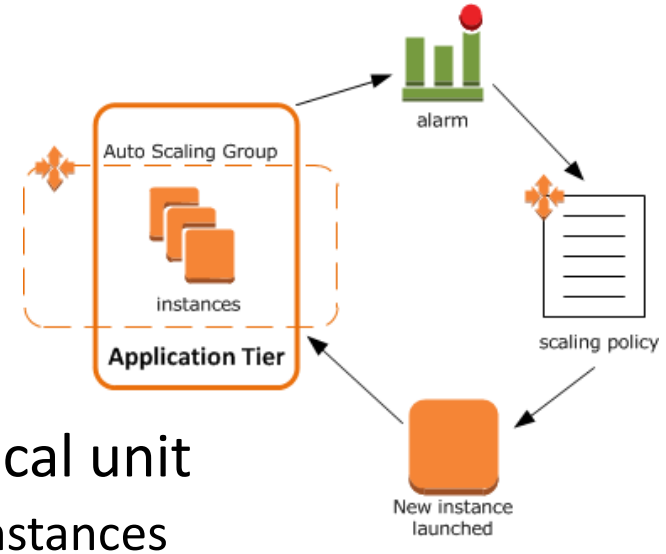
Cloud Computing & Auto Scaling

- Nowadays most cloud providers offer auto scaling services
  - Kubernetes also provides a Horizontal Pod Autoscaler

- For example, Amazon EC2 Auto Scaling
  - Auto scaling groups with minimum, desired and maximum size
  - Instances are scaled out as needed based on the criteria defined in the scaling policy



Cloud Computing & Auto Scaling



- Amazon EC2 auto scaling components

Groups – EC2 are organized into groups, treated as a logical unit

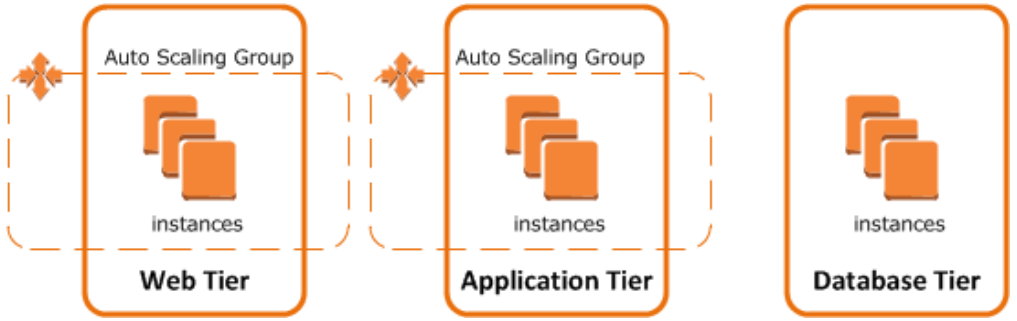
- A group has minimum, maximum and desired number of EC2 instances

Configuration templates – *launch template* used by the group instances

- E.g., ID of the Amazon Machine Image (AMI), instance type, key pair, security groups

Scaling options – rules used for scaling

- Manually, demand, predictive, schedule



Just one more question…

How do I create a private cloud?

Building Clouds

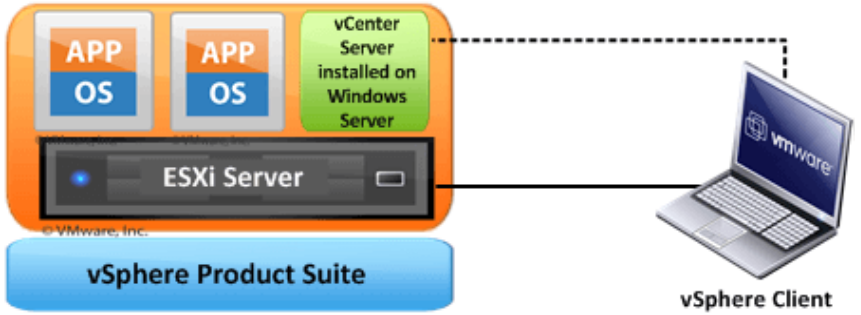
- What is required?
  - It all starts with hardware: hypervisors, network infrastructure, storage,…
  - A base operating system (e.g., Linux, ESXi, …)
  - The cloud will abstract, pool, and share scalable resources across a network
  - But… this can include bare-metal, virtualization or container

- I have a cloud if I have an IT system that
  - Can be accessed by other computers through a network
  - Contains a repository of IT resources
  - Can be provisioned and scaled quickly by me

Building Clouds

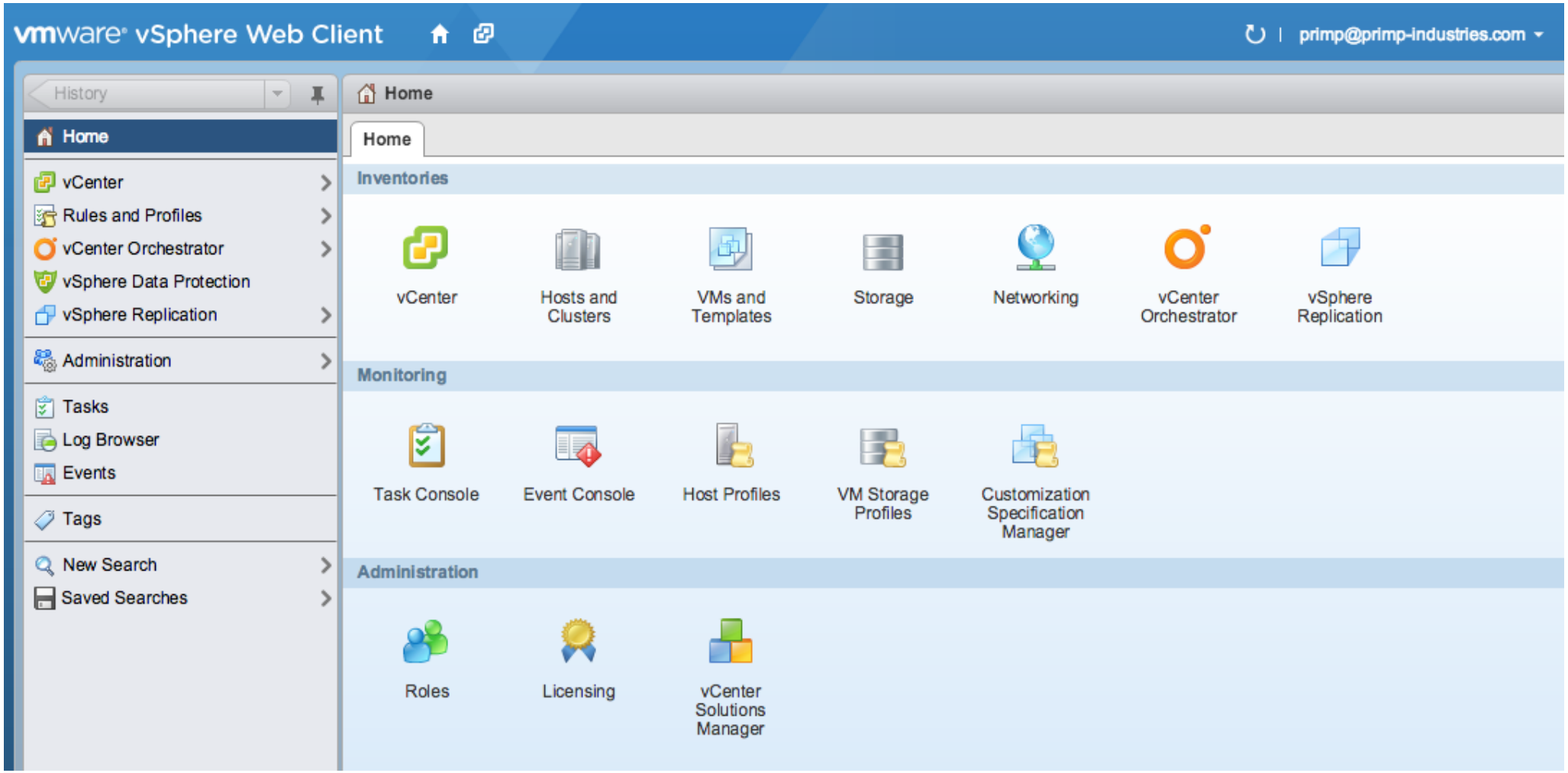
- Normally it starts with type 1 hypervisors (part of a DC)
  - VMware ESXi
    - Bare-metal hypervisor that installs directly onto your physical server
    - Direct access to and control of underlying resources (partitions hardware)
  - Linux KVM
  - Microsoft Hyper-V

Building Clouds

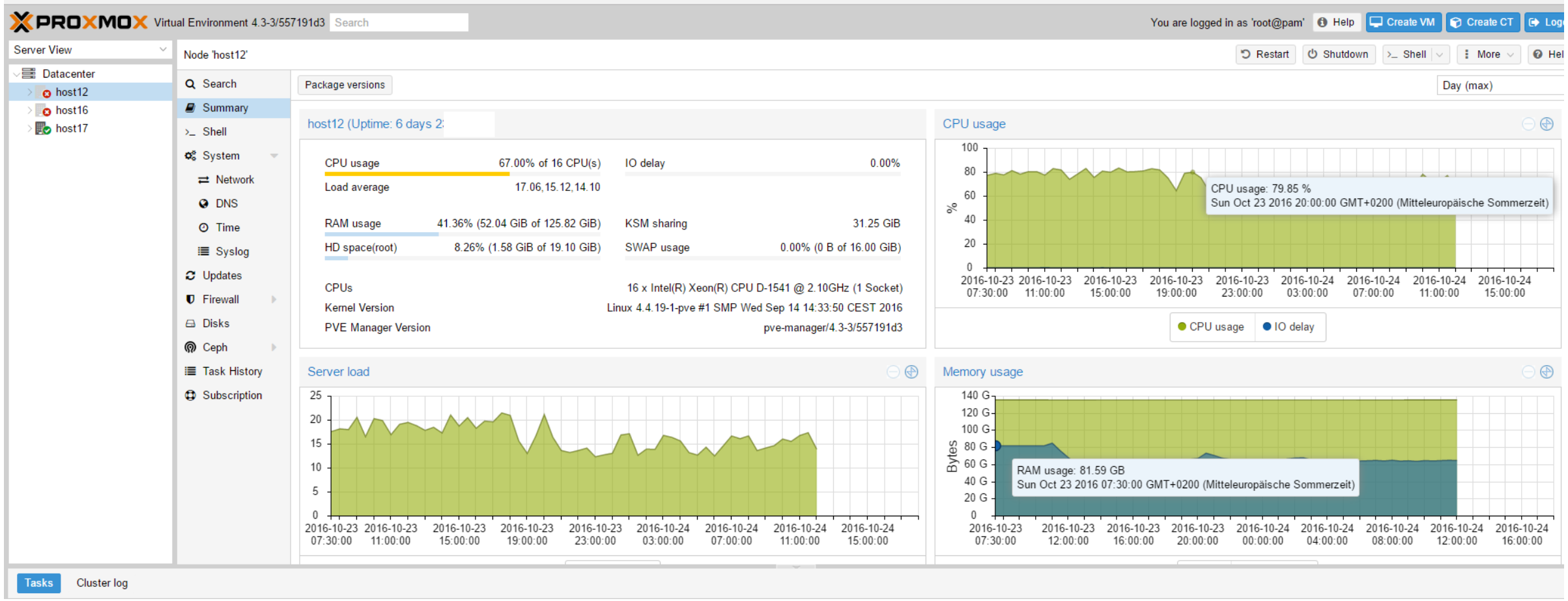


- These HV are managed with specific tools
  - VMware vCenter
    - Centralized management application to manage VMs and ESXi hosts centrally
    - ESXi, vSphere client and vCenter are components of the VMware vSphere suite
  - Proxmox VE
    - Open-source server virtualization management solution based on QEMU/KVM and LXC
    - Can manage VMs, containers, highly available clusters, storage and networks
    - Provides a web interface and CLI
  - oVirt
    - Open-source distributed virtualization solution to manage your entire infrastructure
    - Uses KVM hypervisor and is built upon OSS projects, including libvirt, Gluster, PatternFly, and Ansible

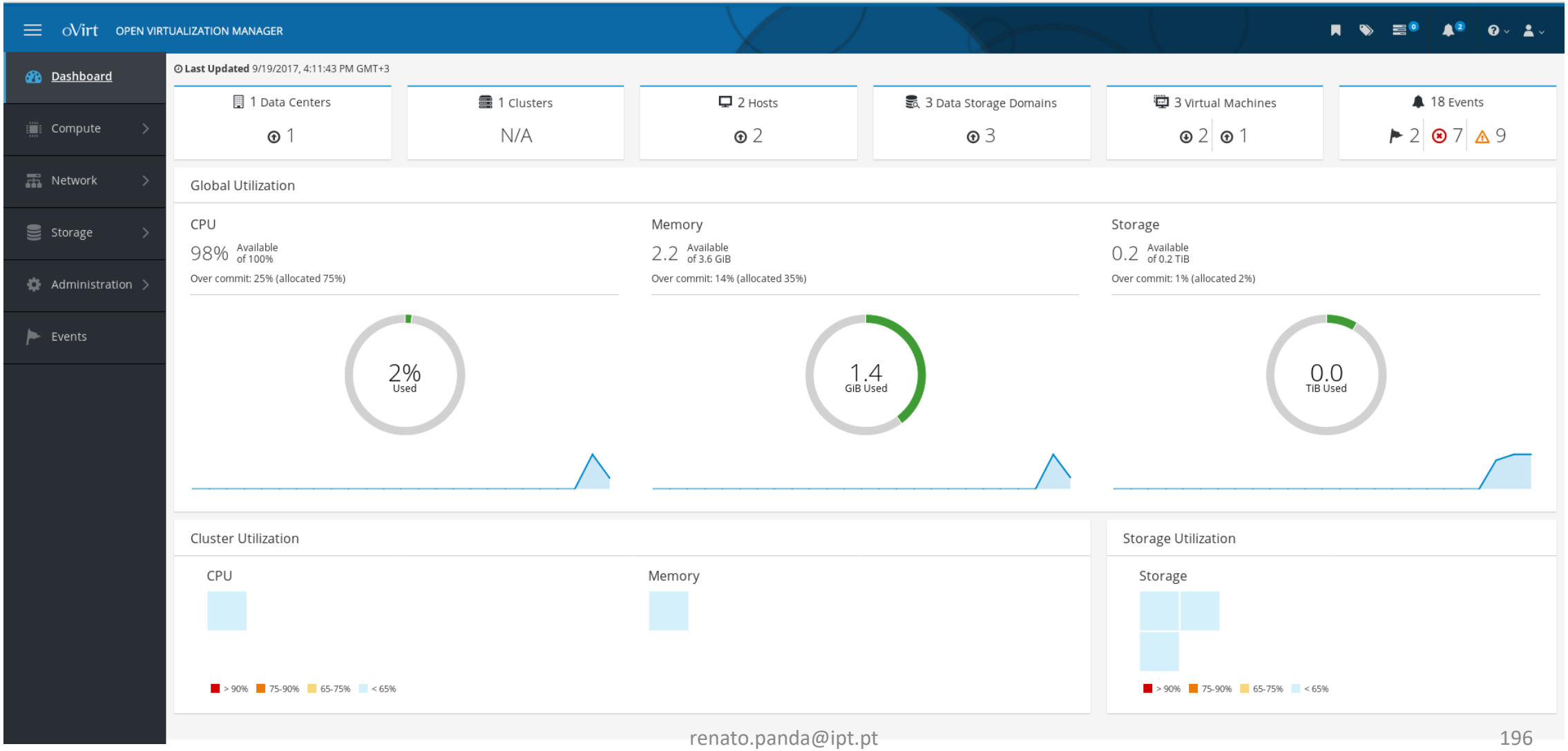
VMware vSphere Web Interface



Proxmox VE Web Interface



oVirt Web Interface



Building Clouds

- vSphere, oVirt and Proxmox are used to manage sets of hypervisors
  - Have a set of hosts and storage area network (SAN)? Then these are your first stop!
  - They already provide some neat features
    - Web-based user interfaces for both admin and non-admin users
    - Integrated management of hosts, storage, and network configuration
    - Live migration of virtual machines and disks between hosts and storage
    - High availability of virtual machines in the event of host failure

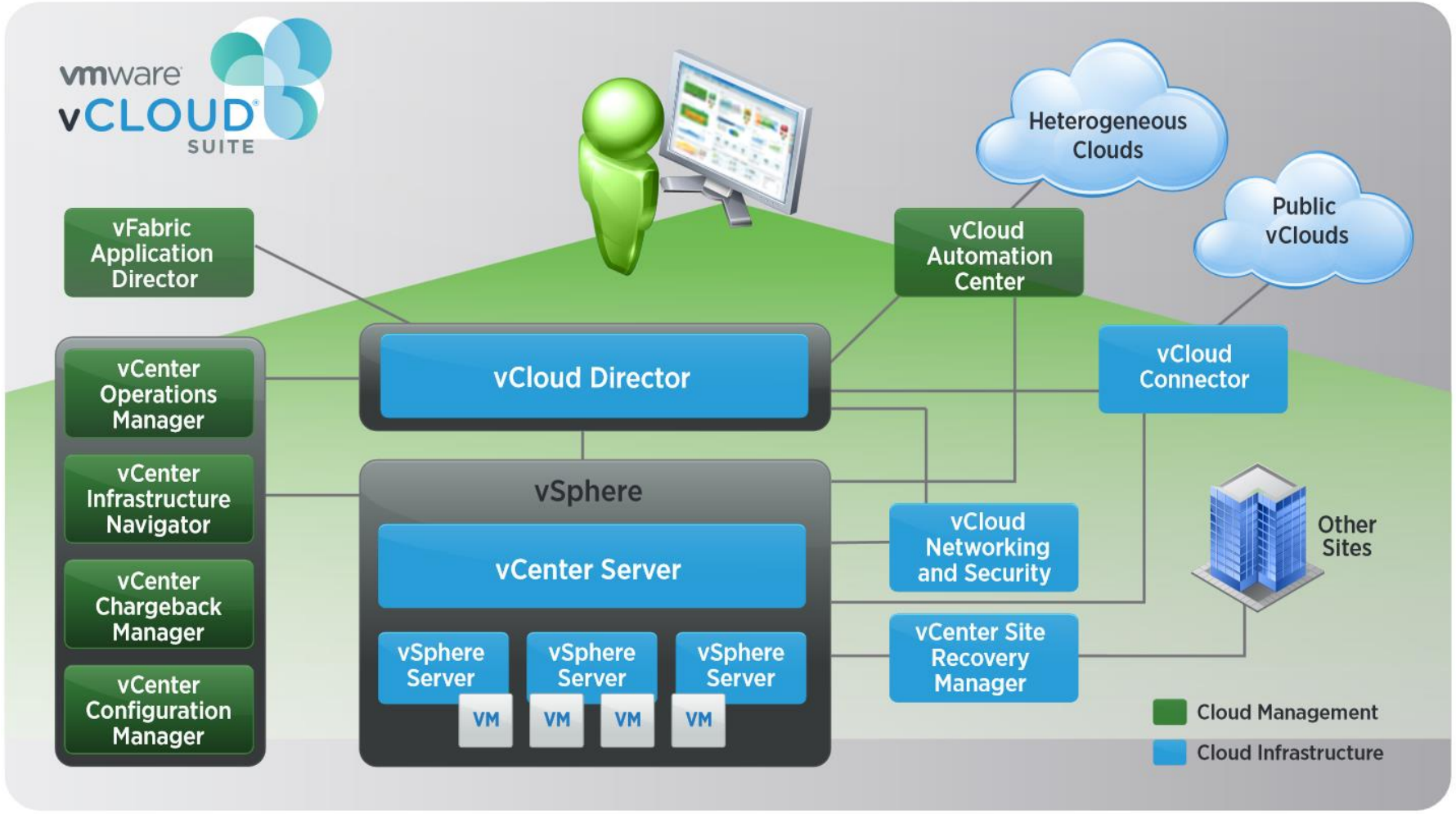
Building Clouds

- As your infrastructure grows and you want to provide more automation and APIs you get to more complex tools
  - VMware vCloud
    - Automation system for public and private clouds that relies on vSphere as its infrastructure
  - RedHat OpenStack
  - Apache CloudStack
  - OpenNebula

- These are tools to create private/public clouds of VMs
  - Think of them as Amazon EC2 and similar services

VMware vCloud Suite

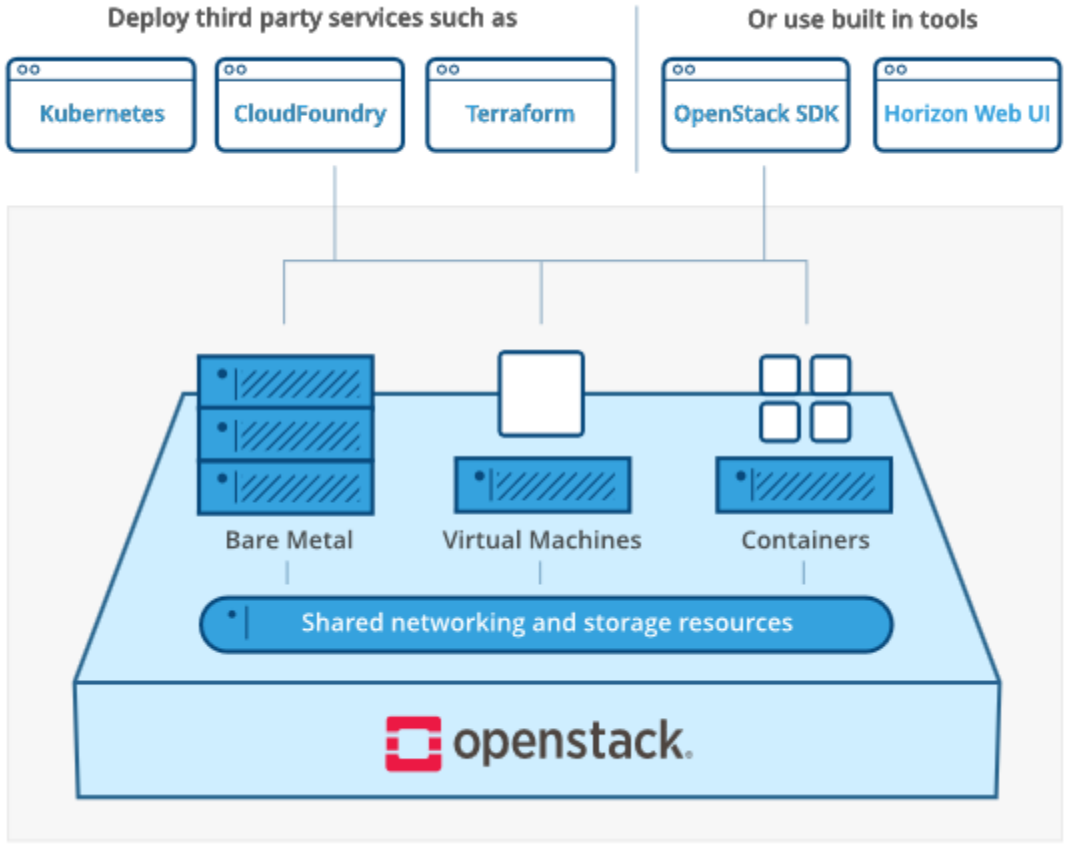
Your IT Infrastructure upgraded to a Software-Defined Data Center



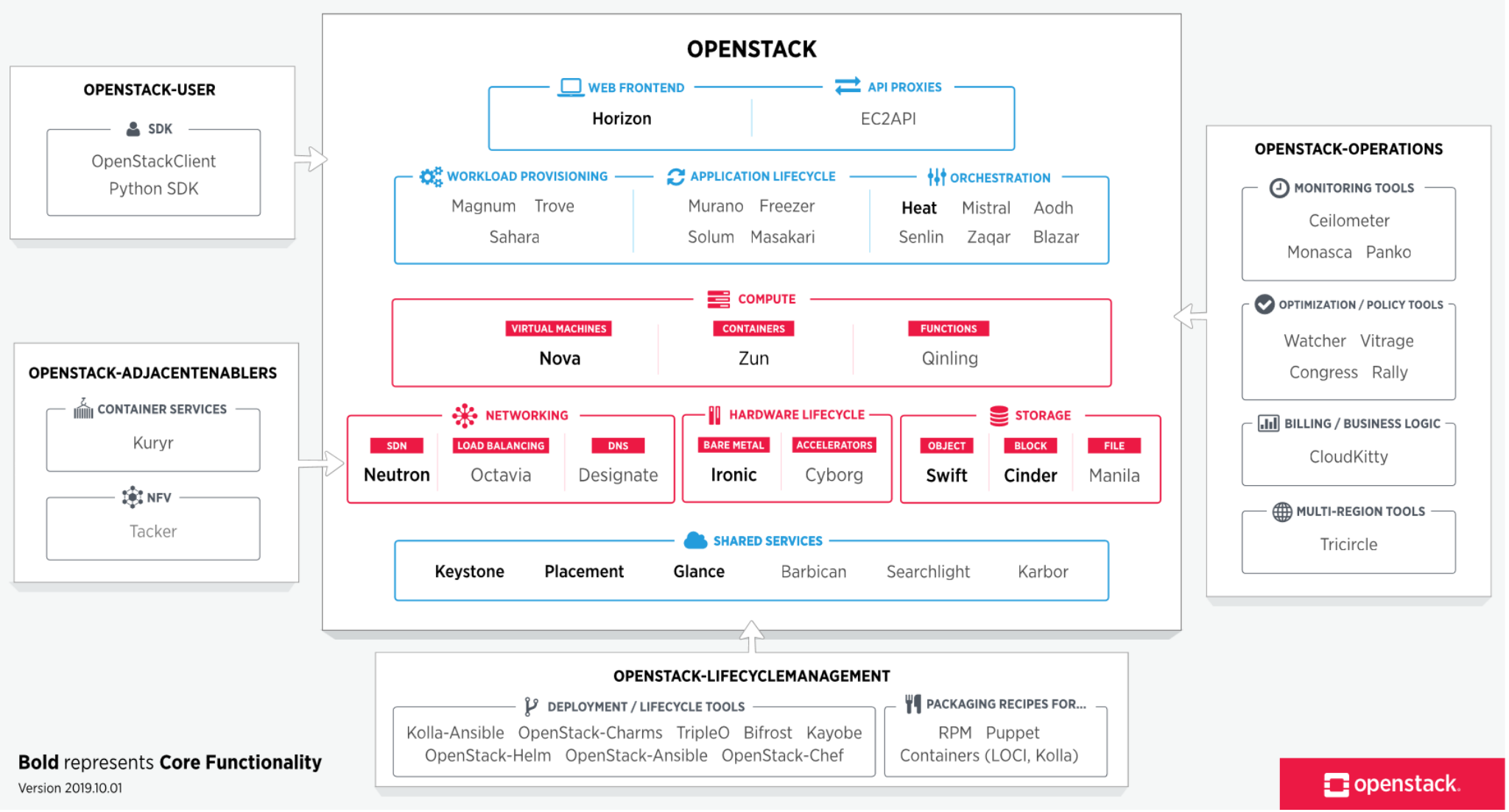
https://blogs.vmware.com/vsphere/2013/01/vcloud-suite-poster.html

RedHat OpenStack

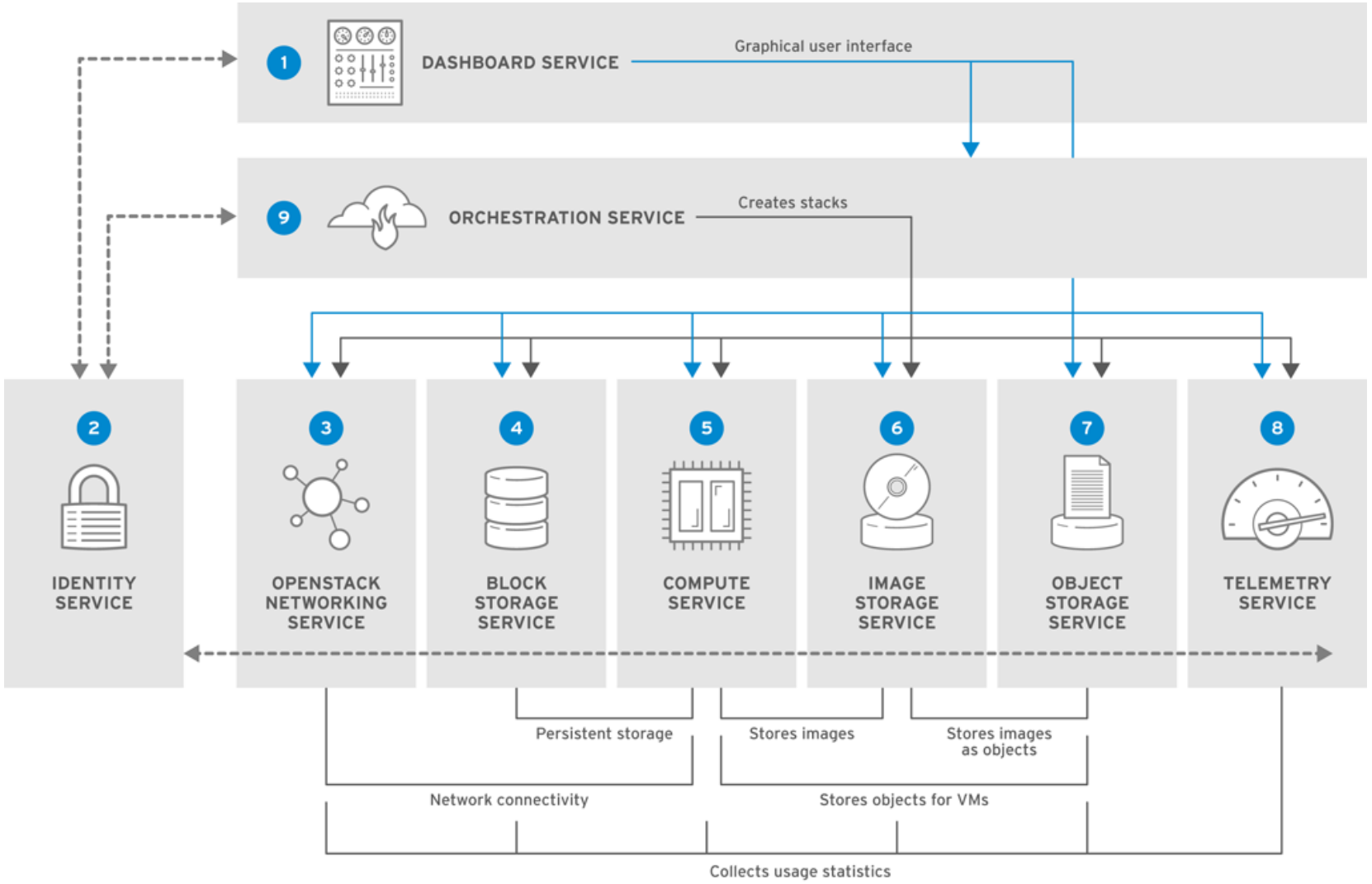
- What is OpenStack?
  - Cloud OS that controls large pools of compute, storage, and networking resources throughout a datacenter
    - All resources are managed and provisioned through APIs with common authentication mechanisms
    - Includes IaaS functionality, orchestration, fault management and service management for HA



The OpenStack Landscape



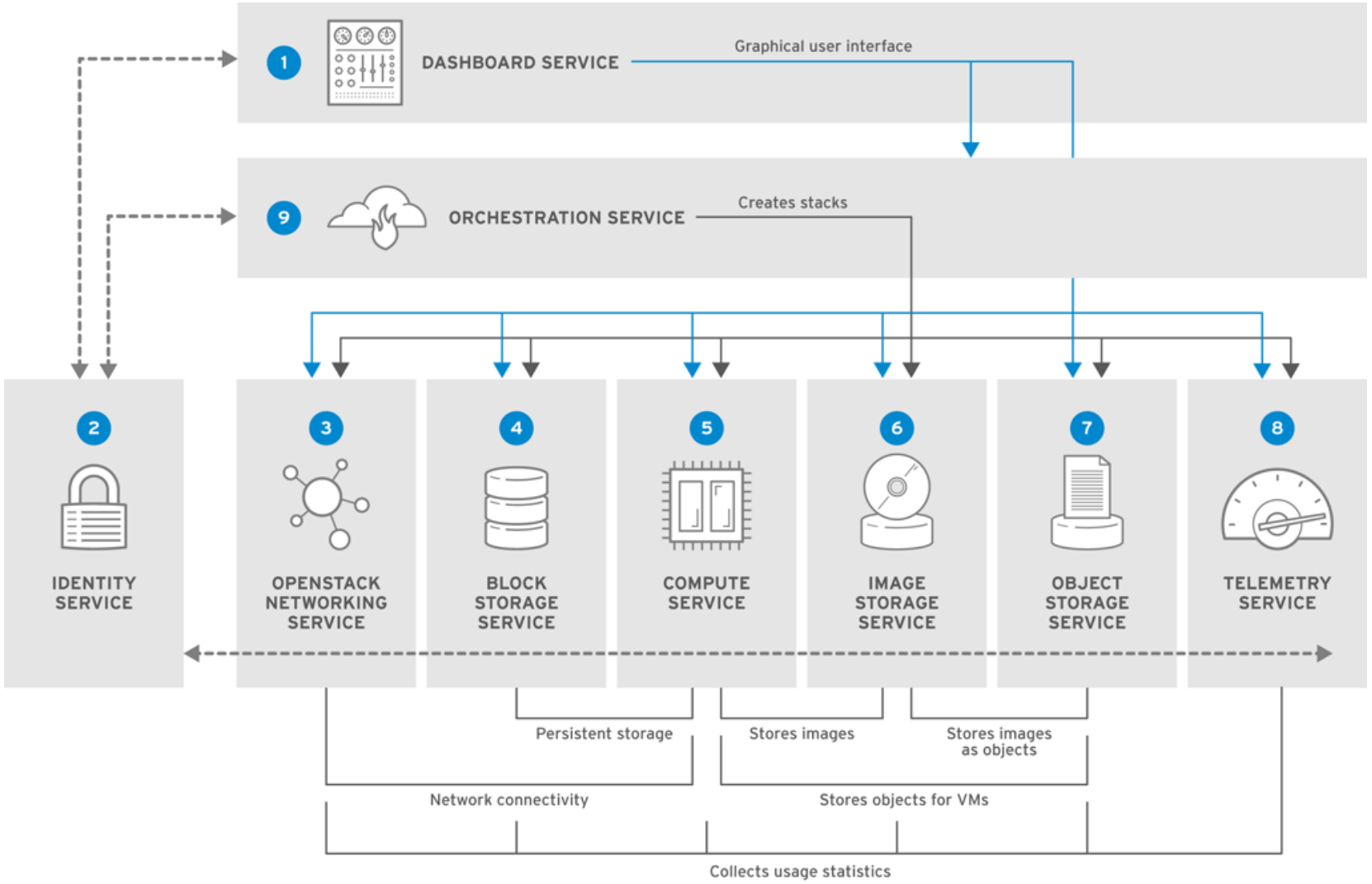
RedHat OpenStack



1) Dashboard (horizon):

Web browser-based dashboard that you use to manage OpenStack services

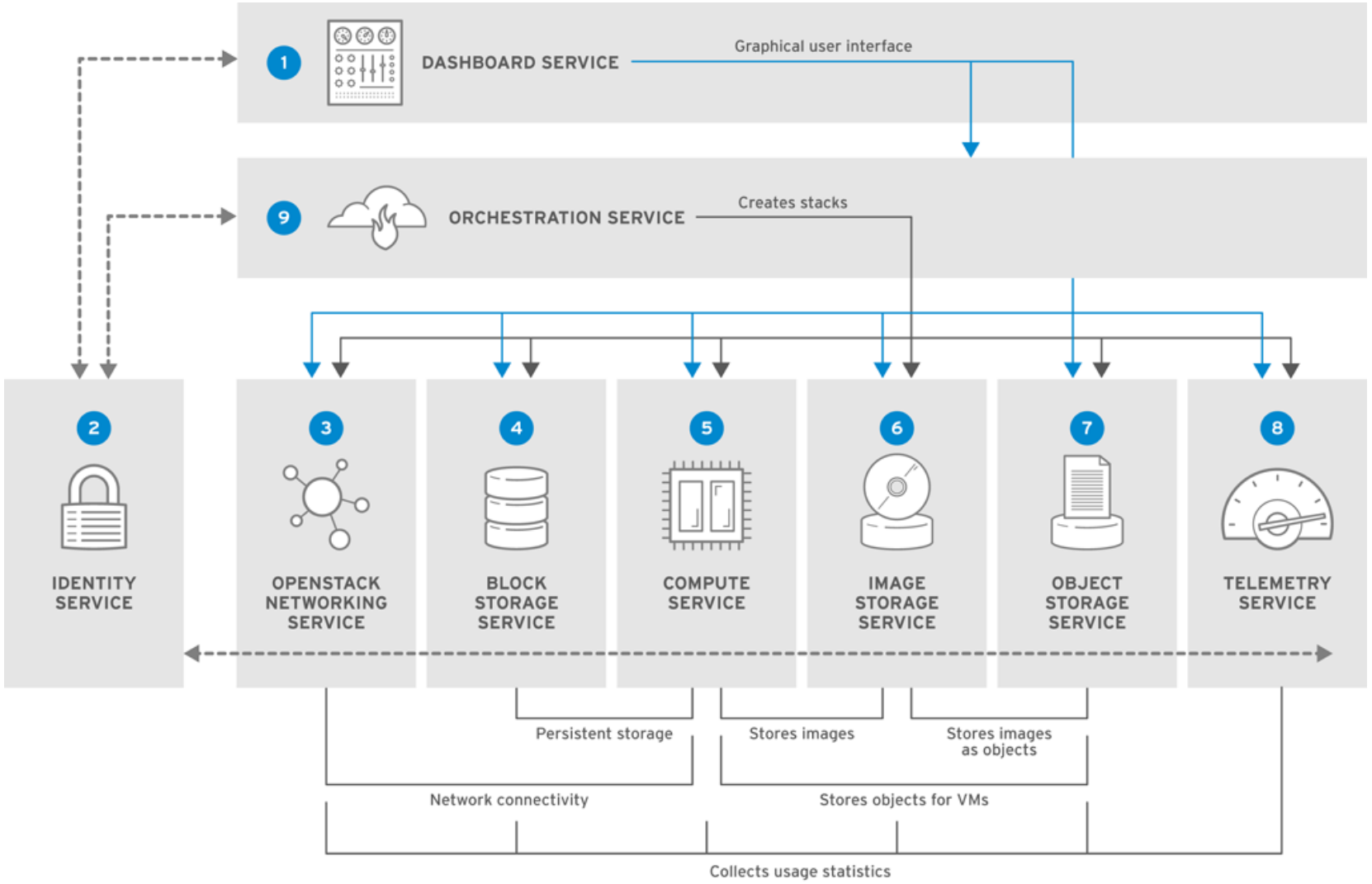
RedHat OpenStack



2) Identity (keystone)

Centralized service for authentication and authorization of OpenStack services and for managing users, projects, and roles

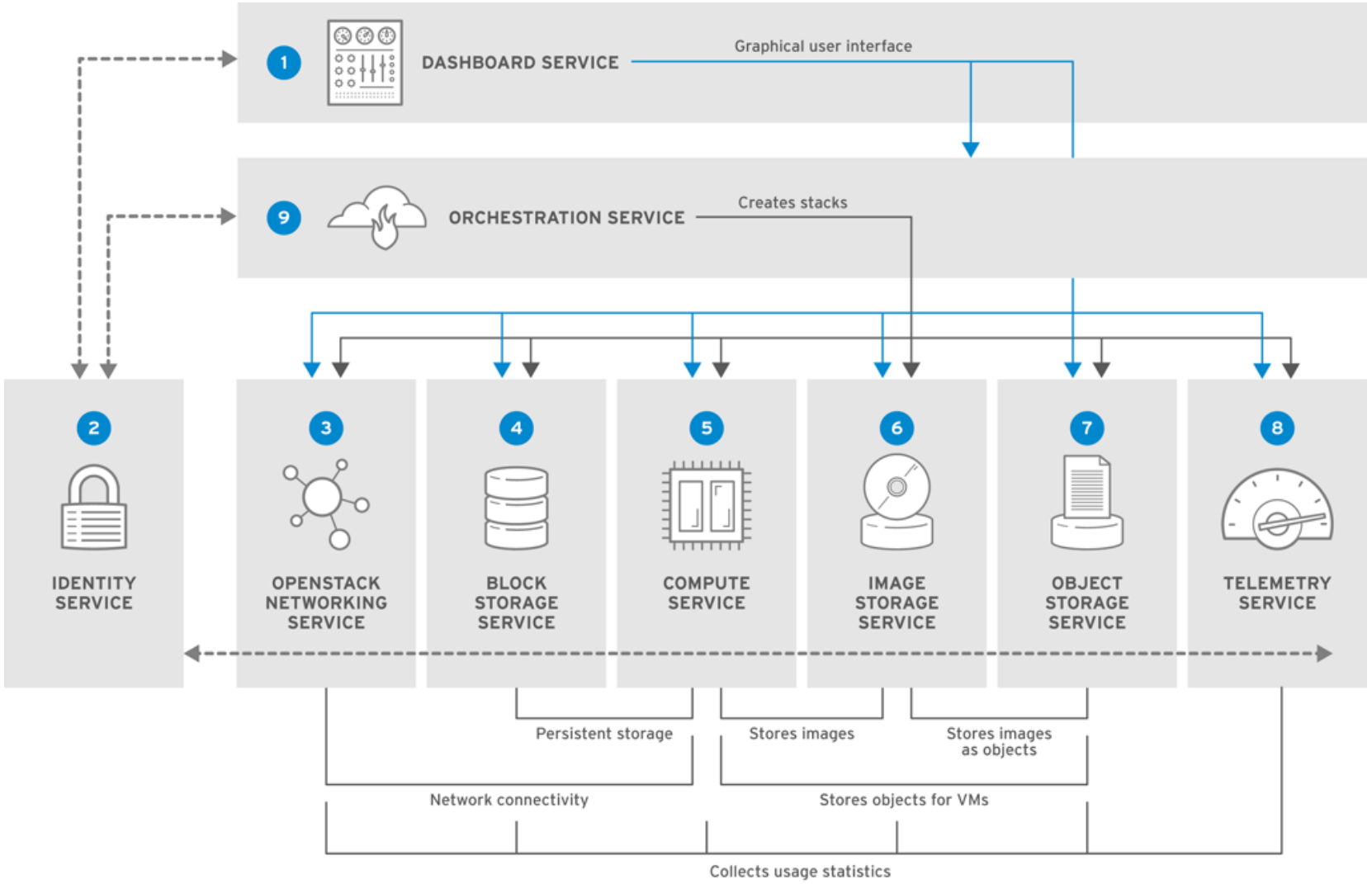
RedHat OpenStack



3) OpenStack Networking (neutron)

Provides connectivity between the interfaces of OpenStack services

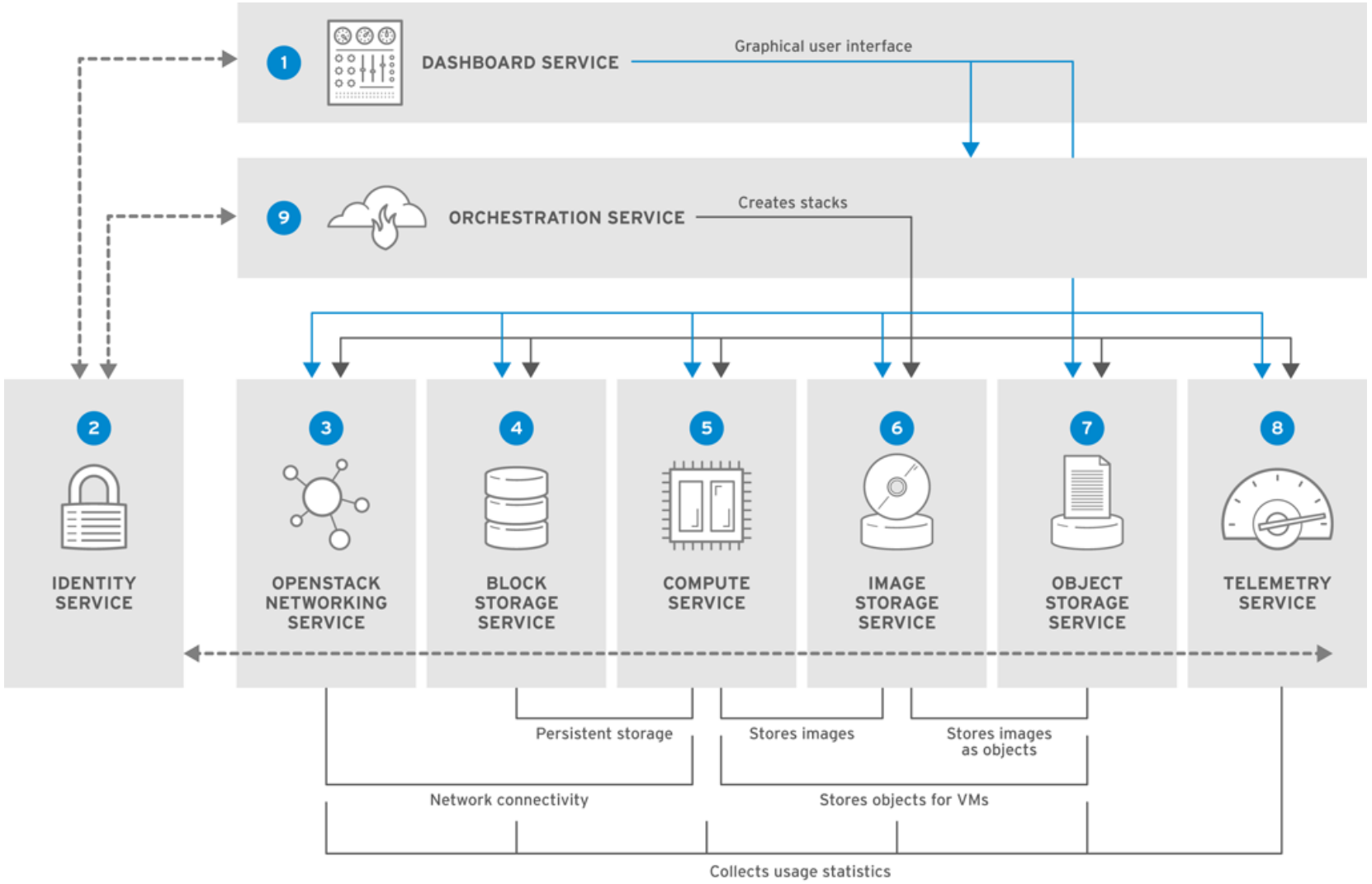
RedHat OpenStack



4) Block Storage (cinder)

Manages persistent block storage volumes for virtual machines

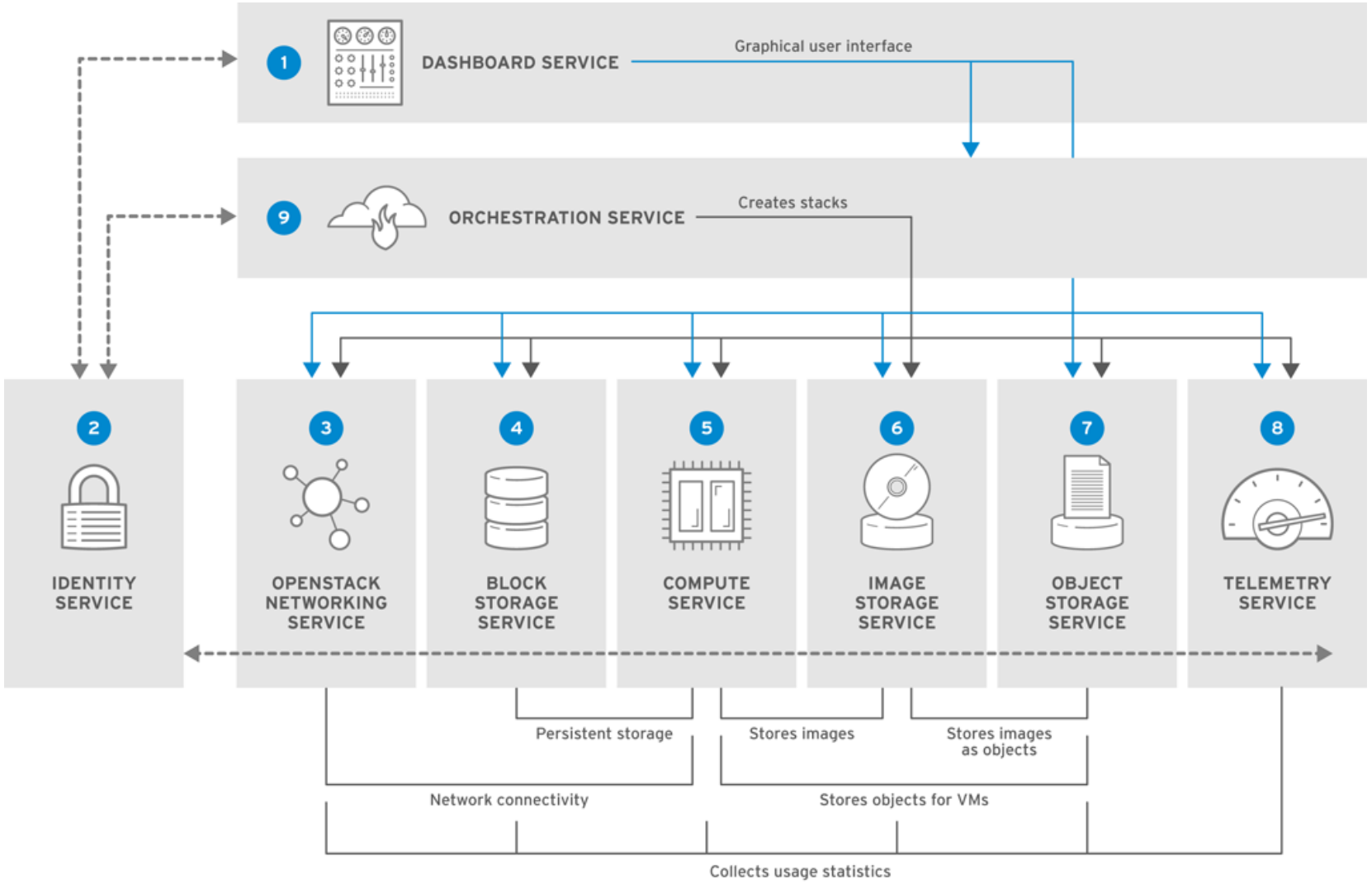
RedHat OpenStack



5) Compute (nova)

Manages and provisions virtual machines running on hypervisor nodes.

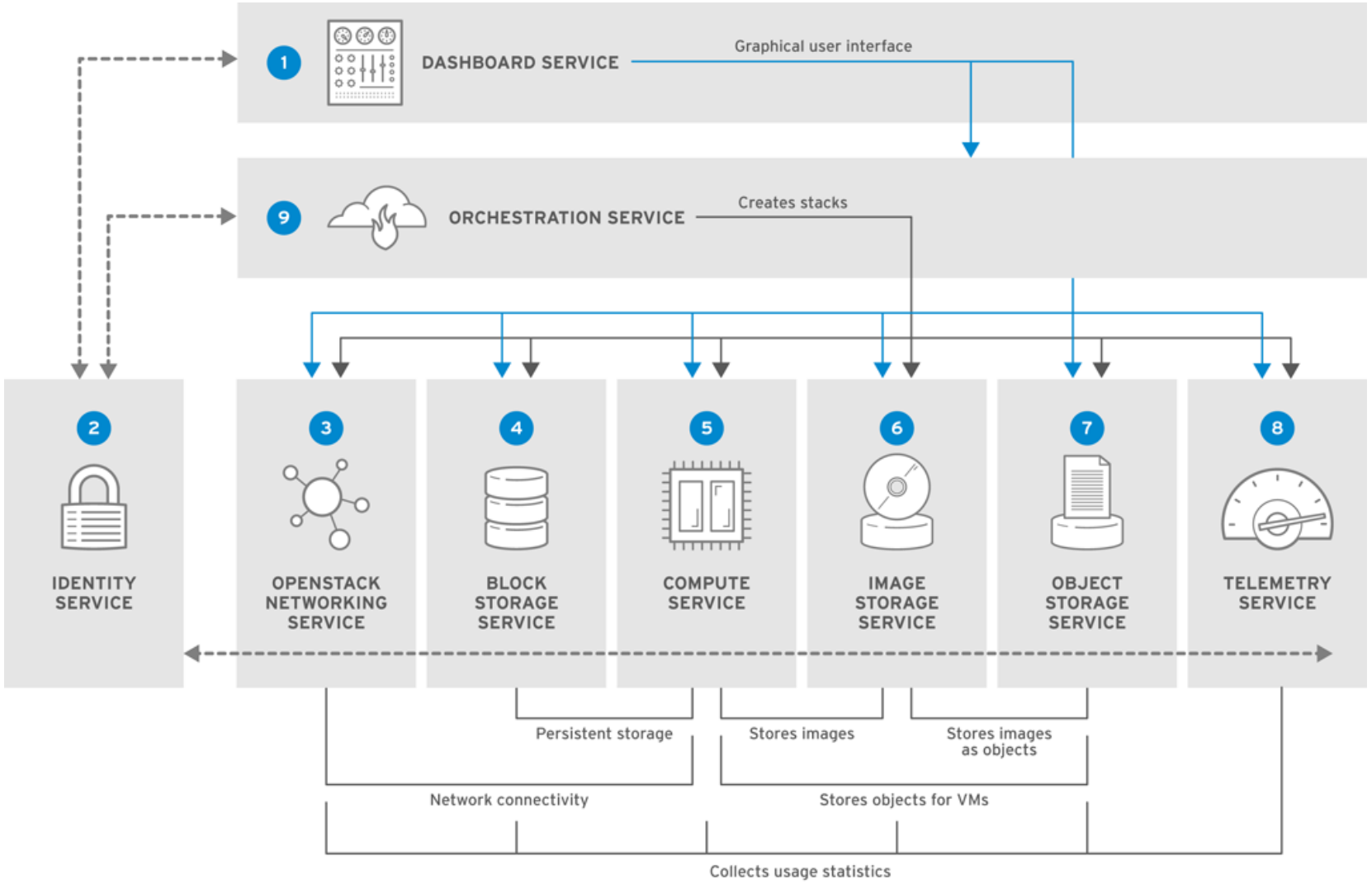
RedHat OpenStack



6) Image (glance)

Registry service that you use to store resources such as virtual machine images and volume snapshots.

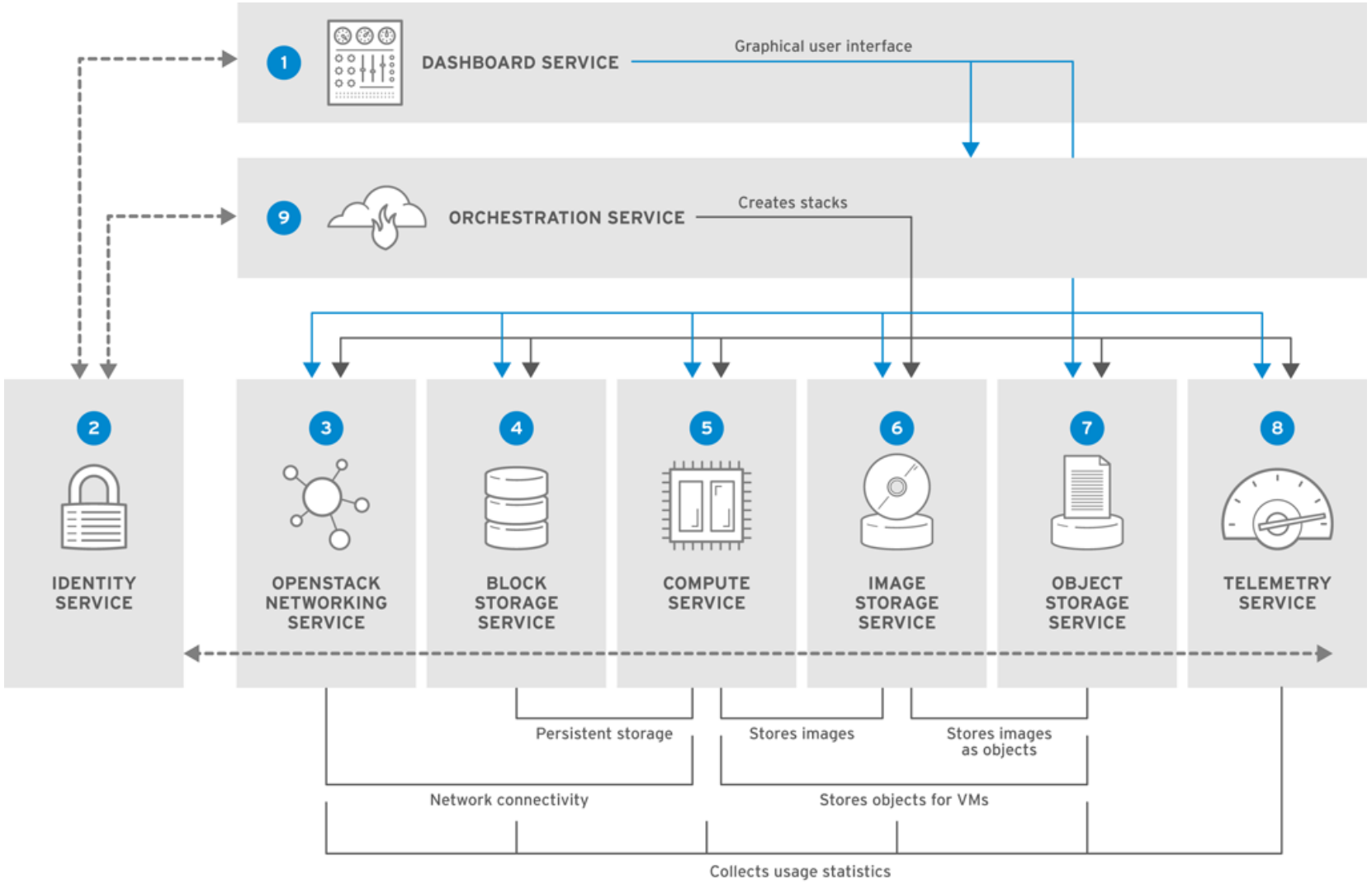
RedHat OpenStack



7) Object Storage (swift)

Allows users to store and retrieve files and arbitrary data.

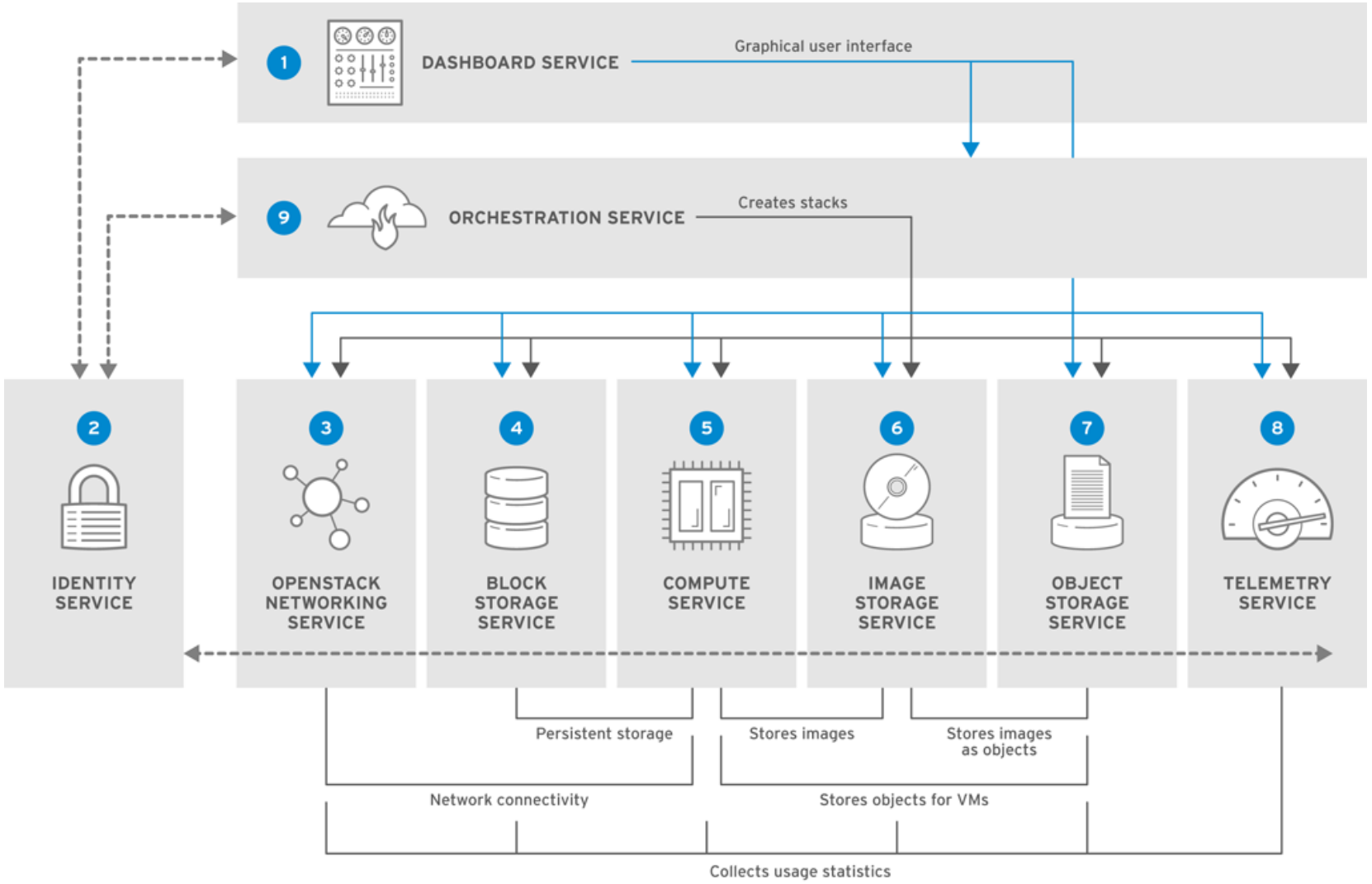
RedHat OpenStack



8) Telemetry (ceilometer)

Provides measurements of cloud resources.

RedHat OpenStack

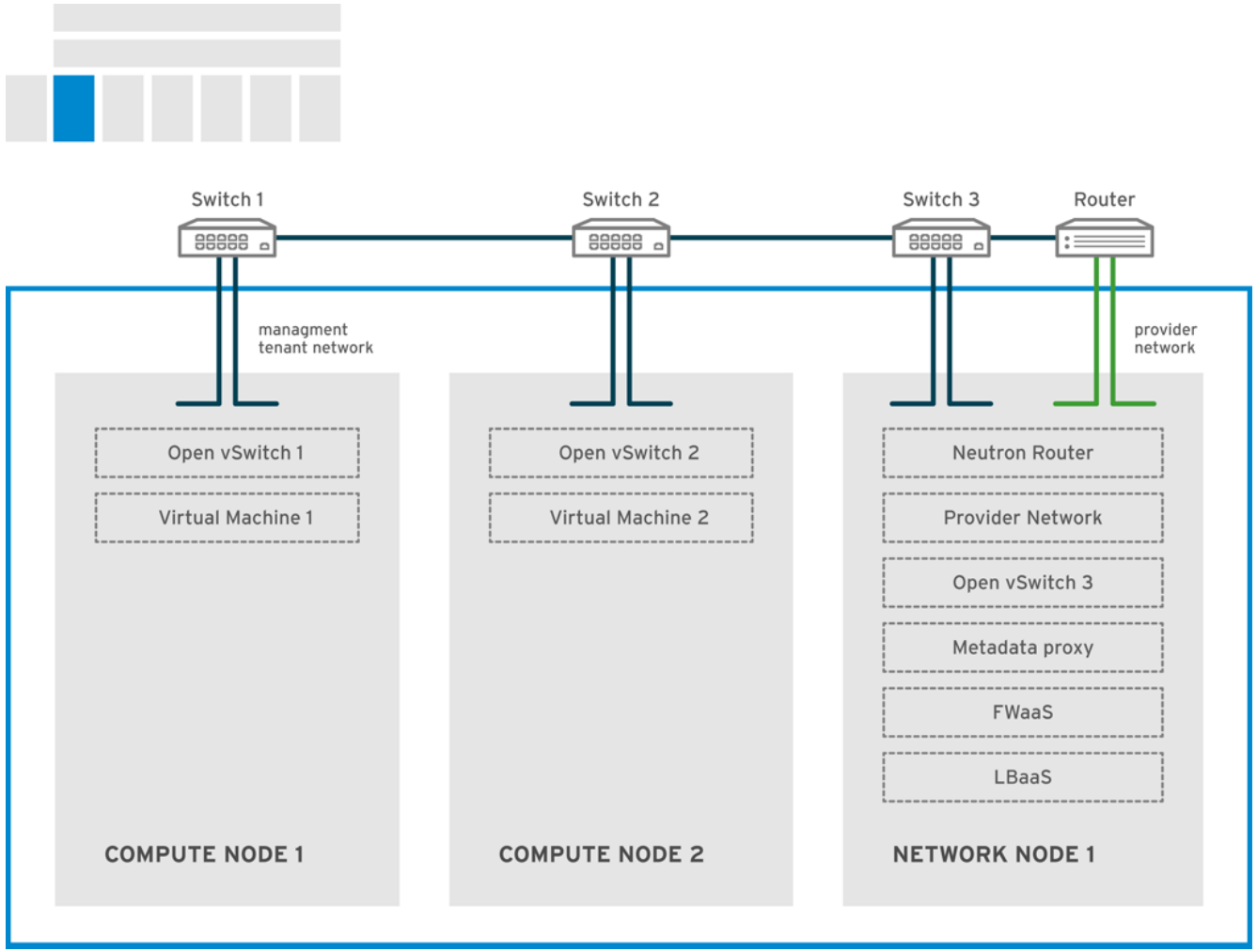


9) Orchestration (heat)

Template-based orchestration engine that supports automatic creation of resource stacks.

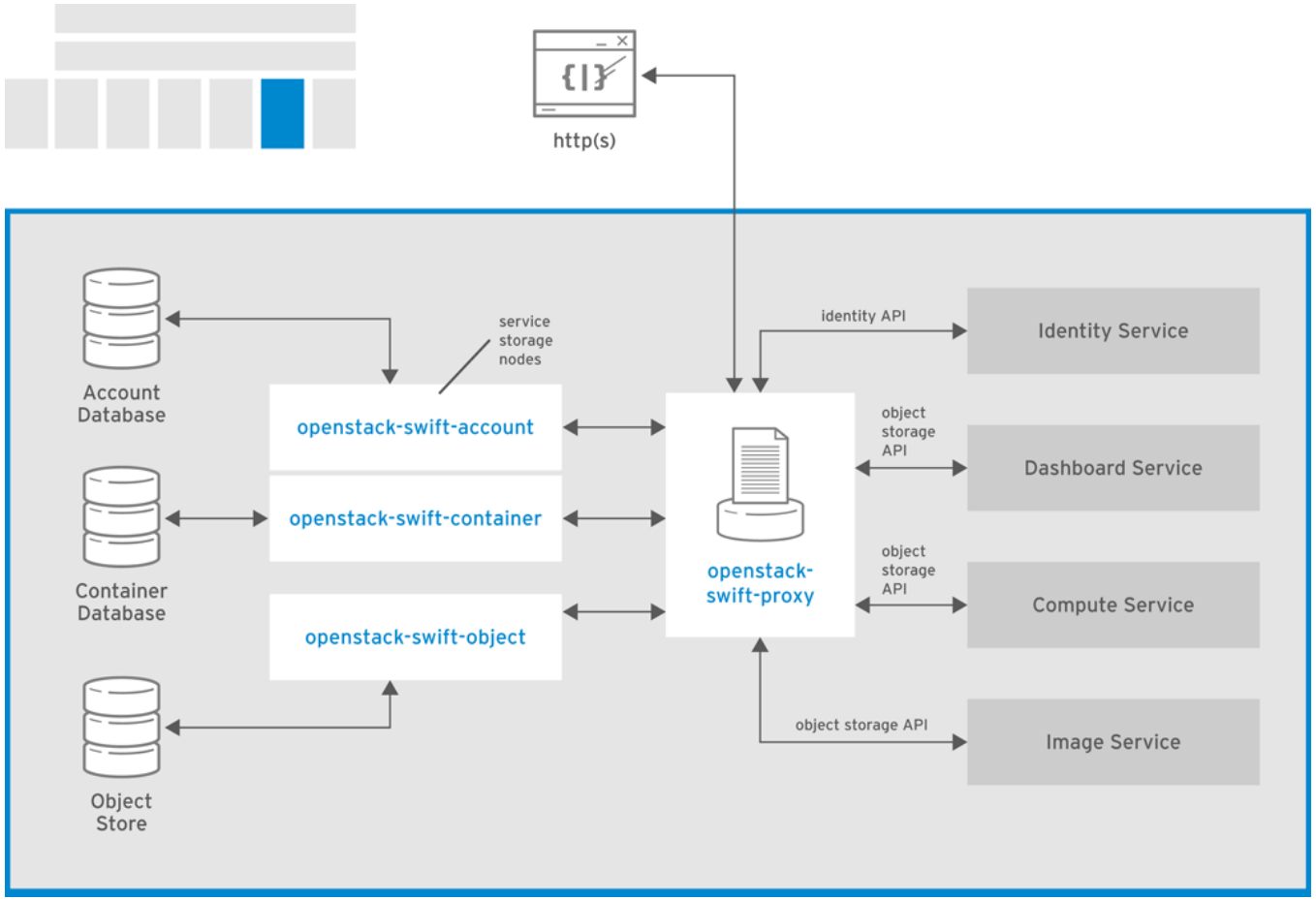
Want to know more about OpenStack?

- How does the networking service work?



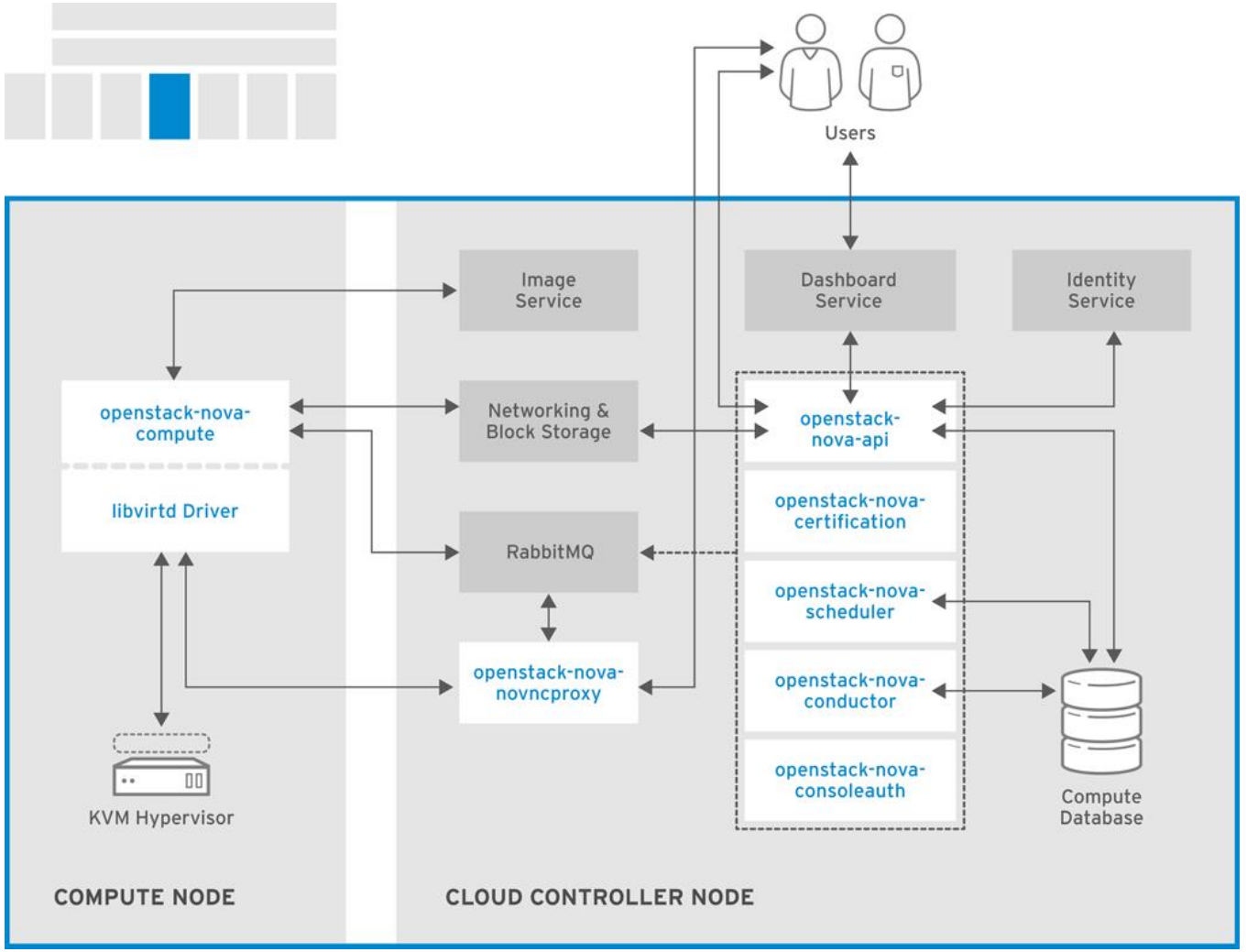
Want to know more about OpenStack?

- How does the networking service work?
- What about the object storage?



Want to know more about OpenStack?

- How does the networking service work?
- What about the object storage?
- How about the VMs, images and templates?
- Check the documentation!



OpenNebula

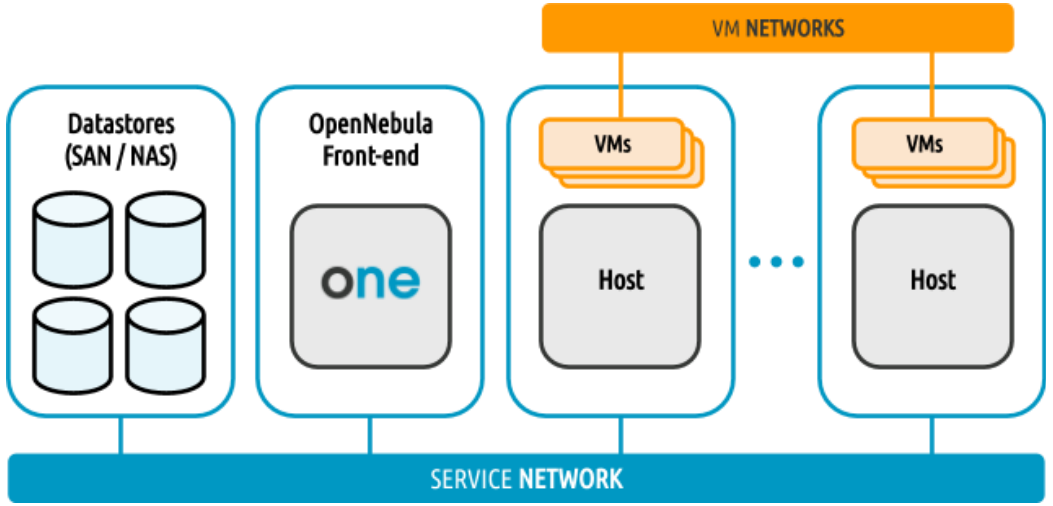
- What is OpenNebula
  - Cloud computing platform for managing **heterogeneous** distributed data center infrastructures
  - Orchestrates storage, network, virtualization, monitoring, and security technologies to deploy multi-tier services (e.g., compute clusters) as virtual machines on distributed infrastructures
  - A less complex / overkill version of OpenStack

OpenNebula

- OpenNebula combines
  - **VMware and KVM** virtual machines for fully virtualized clouds
  - **LXD system containers** for containerized clouds
  - Firecracker micro-VMs for ***serverless*** **deployments**
  - Integrate with cloud providers like AWS and Azure to create flexible **hybrid and edge cloud** infrastructures

OpenNebula Architecture

- Physical infrastructure adopts a classical cluster-like architecture
  - **Front-end** that executes the OpenNebula services
  - Hypervisor-enabled **hosts** that provide the resources needed by the VMs
  - Physical **networks** interconnect the storage servers and OpenNebula control operations, and VLANs for the VMs
  - **Datastores** – hold the base images of the VMs



Cloud & Unexpected Costs

- https://www.troyhunt.com/how-i-got-pwned-by-my-cloud-costs/