Cloud Computing

Guess: Where is the World's Largest Datacenter?

Where is the World's Largest Datacenter?

- (2020) Range International Info Group, China, 6.3 Million sq. ft.
- (2018) China Telecom. 10.7 Million sq. ft.
- (2017) "The Citadel" Nevada. 7.2 Million sq. ft.
- (2015) In Chicago!
  - 350 East Cermak, Chicago, 1.1 MILLION sq. ft.
  - Shared by many different "carriers"
  - Critical to Chicago Mercantile Exchange
- See:
  - http://ict-price.com/top-10-biggest-data-centres-from-around-the-world/
  - https://www.gigabitmagazine.com/top10/top-10-biggest-data-centres-world
  - https://www.racksolutions.com/news/data-center-news/top-10-largest-data-centers-world/


Google
App Engine
amazon
web services
RIGHTSCALE
ORACLE
GridGain
2.0
Elastic Server
VirtualBox
virtualLogix
vmware
force.com
platform as a service
CITRIX
OpenVZ
KVM
10gen
Microsoft
parascale
Powering Cloud Storage
hp
Xen
GIGASPACES
Parallels
Optimized Computing
RIGHTSCALE
elastra
enomaly
ORACLE
BUNGEEconnect
PowerVM
Virtualiron

Customers Save Time and $$$

- Dave Power, Associate Information Consultant at Eli Lilly and Company: “With AWS, Powers said, a new server can be up and running in **three minutes** (it used to take Eli Lilly **seven and a half weeks** to deploy a server internally) and a **64-node Linux cluster** can be online in five minutes (compared with three months internally). … It's just shy of instantaneous.”

- Ingo Elfering, Vice President of Information Technology Strategy, GlaxoSmithKline: “With Online Services, we are able to reduce our IT **operational costs** by roughly **30%** of what we’re spending”

- Jim Swartz, CIO, Sybase: “At Sybase, a private cloud of virtual servers inside its datacenter has saved nearly **$US2 million annually** since 2006, Swartz says, because the company can share computing power and storage resources across servers.”

- 100s of startups in Silicon Valley can harness large computing resources without buying their own machines.

Many Cloud Providers

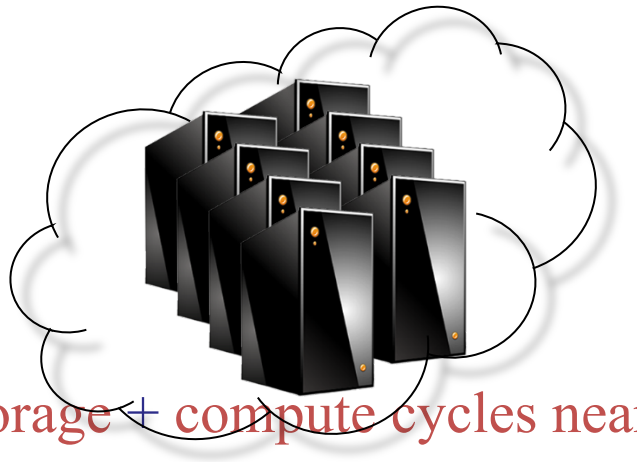
- AWS: Amazon Web Services
  - EC2: Elastic Compute Cloud
  - S3: Simple Storage Service
  - EBS: Elastic Block Storage
- Microsoft Azure
- Google Cloud/Compute Engine/AppEngine
- Rightscale, Salesforce, EMC, Gigaspaces, 10gen, Datastax, Oracle, VMWare, Yahoo, Cloudera
- And many many more!

But what exactly IS a cloud?

What is a Cloud?

- It's a cluster!
- It's a supercomputer!
- It's a datastore!
- It's superman!

- None of the above
- All of the above

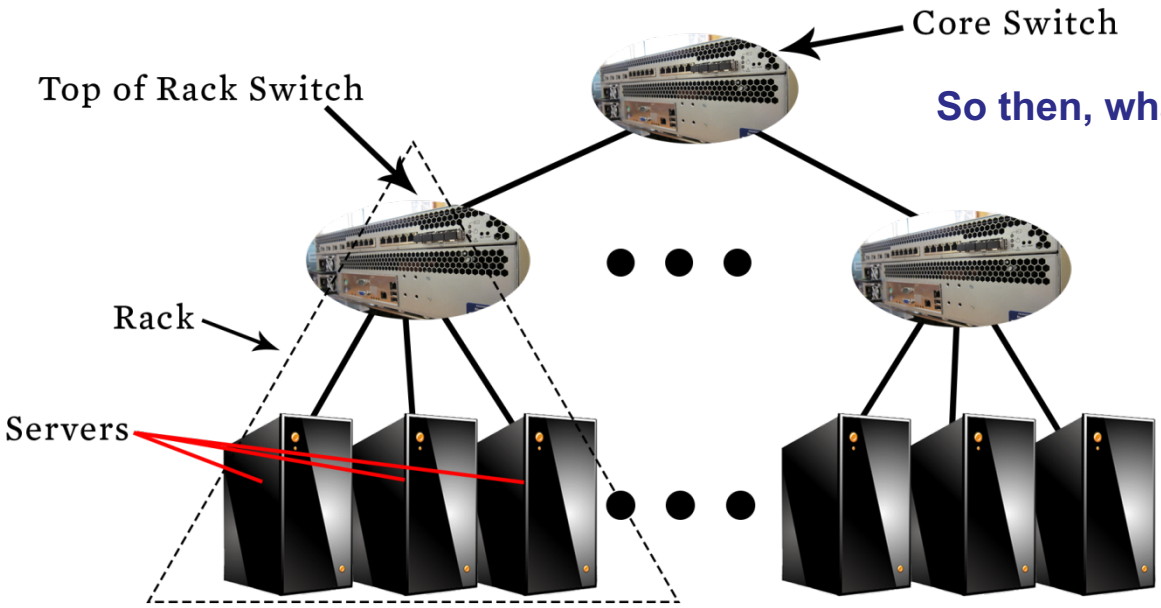


- Cloud = Lots of storage + compute cycles nearby

What is a Cloud?

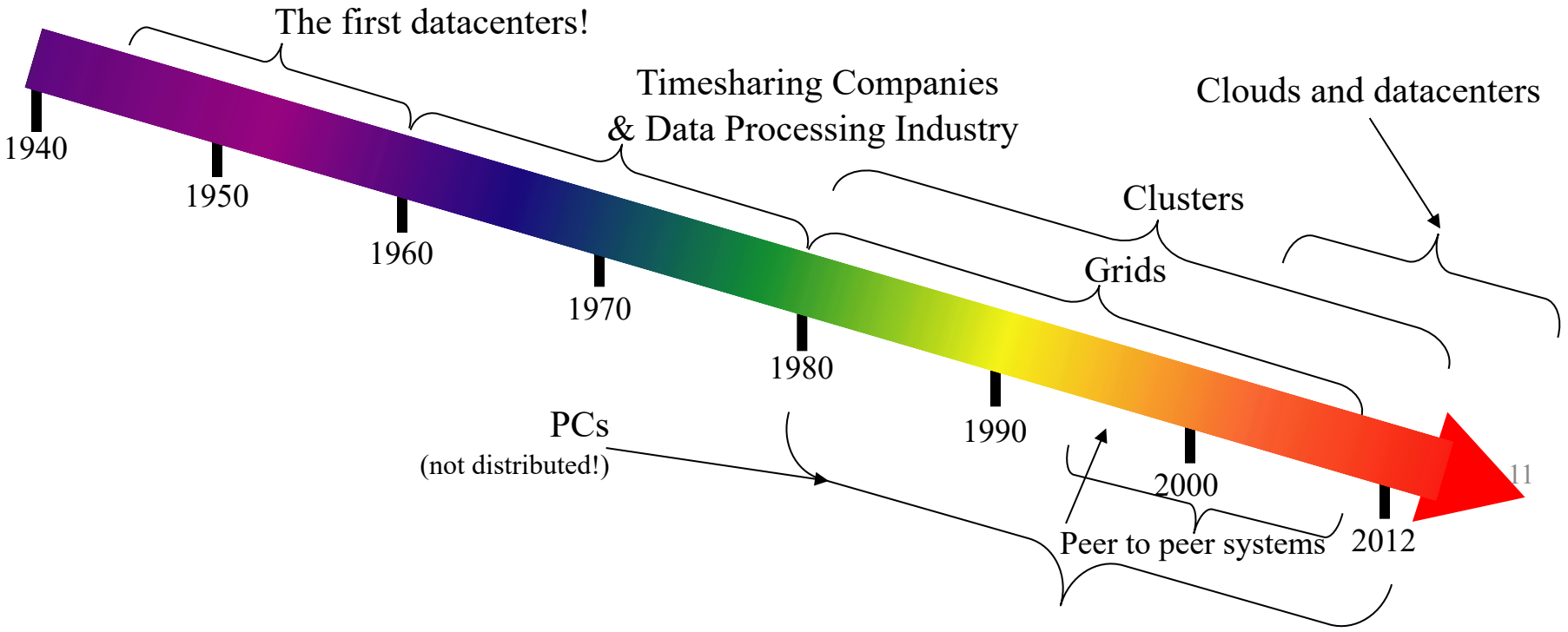
- A single-site cloud (aka "Datacenter") consists of
  - Compute nodes (grouped into racks) (2)
  - Switches, connecting the racks
  - A network topology, e.g., hierarchical
  - Storage (backend) nodes connected to the network (3)
  - Front-end for submitting jobs and receiving client requests (1)
  - (1-3: Often called "three-tier architecture")
  - Software Services
- A geographically distributed cloud consists of
  - Multiple such sites
  - Each site perhaps with a different structure and services

A Sample Cloud Topology



So then, what is a cluster?

"A Cloudy History of Time"



Trends: Technology

- Doubling Periods – storage: 12 mos, bandwidth: 9 mos, and (what law is this?) cpu compute capacity: 18 mos
- Then and Now
  - Bandwidth
    - 1985: mostly 56Kbps links nationwide
    - 2015: Tbps links widespread
  - Disk capacity
    - Today's PCs have TBs, far more than a 1990 supercomputer

Trends: Users

- Then and Now

  Biologists:

  – 1990: were running small single-molecule simulations
  – Today: CERN's Large Hadron Collider producing many PB/year

Prophecies

- In 1965, MIT's Fernando Corbató and the other designers of the Multics operating system envisioned a computer facility operating “like a power company or water company”.

- Plug your thin client into the computing Utility and Play your favorite Intensive Compute & Communicate Application
  - Have today’s clouds brought us closer to this reality? Think about it.

Four Features New in Today's Clouds

I. Massive scale.

II. On-demand access: Pay-as-you-go, no upfront commitment.
- And anyone can access it

III. Data-intensive Nature: What was MBs has now become TBs, PBs and XBs.
- Daily logs, forensics, Web data, etc.
- Humans have data numbness: Wikipedia (large) compressed is only about 10 GB!

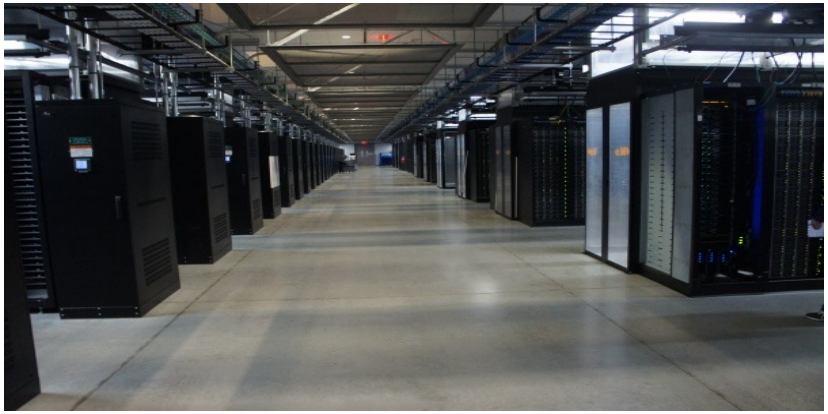
IV. New Cloud Programming Paradigms: MapReduce/Hadoop, NoSQL/Cassandra/MongoDB and many others.
- High in accessibility and ease of programmability
- Lots of open-source

Combination of one or more of these gives rise to novel and unsolved distributed computing problems in cloud computing.

I. Massive Scale

- Facebook [GigaOm, 2012]
  - 30K in 2009 -> 60K in 2010 -> 180K in 2012
- Microsoft [NYTimes, 2008]
  - 150K machines
  - Growth rate of 10K per month
  - 80K total running Bing
  - In 2013, Microsoft Cosmos had 110K machines (4 sites)
- Yahoo! [2009]:
  - 100K
  - Split into clusters of 4000
- AWS EC2 [Randy Bias, 2009]
  - 40K machines
  - 8 cores/machine
- eBay [2012]: 50K machines
- HP [2012]: 380K in 180 DCs
- Google [2011, Data Center Knowledge] : 900K

Servers



Front

Back

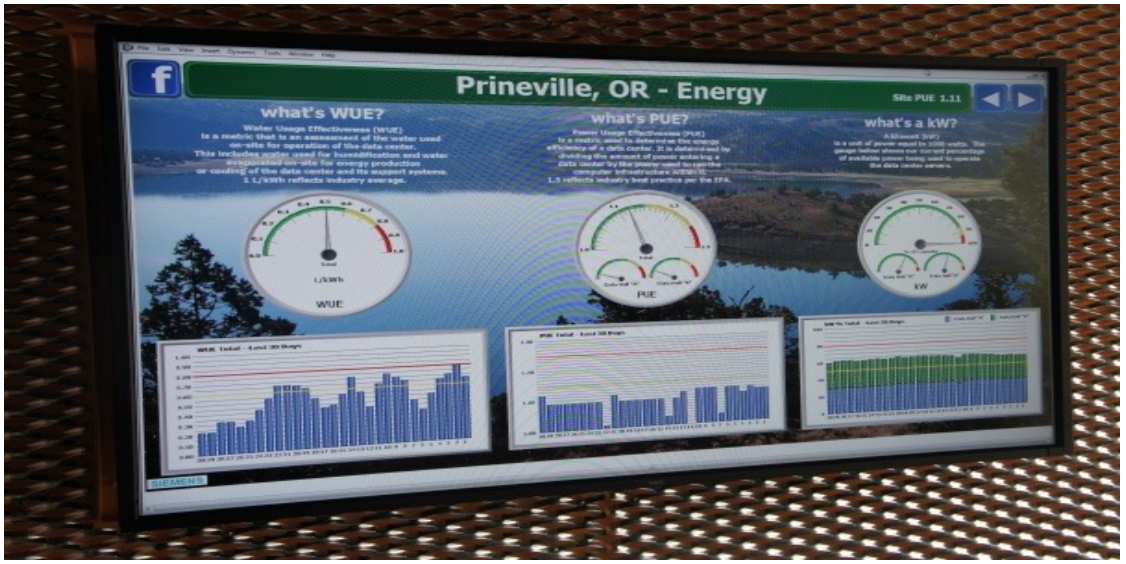
In

Some highly secure (e.g., financial info)

Power

Off-site

On-site

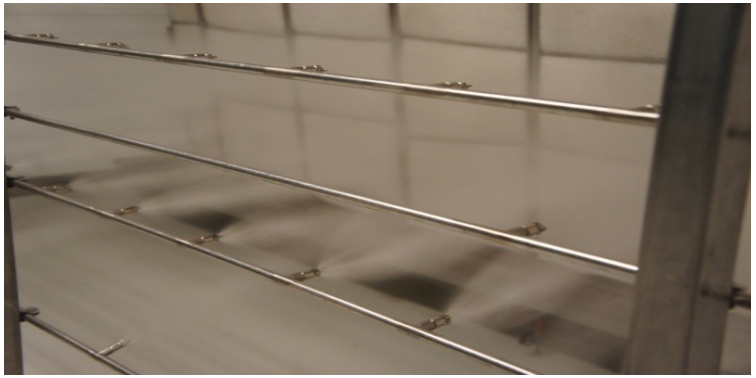


- WUE = Annual Water Usage / IT Equipment Energy (L/kWh) – low is good
- PUE = Total facility Power / IT Equipment Power – low is good (e.g., Google~1.1)

Cooling

Air sucked in from top (also, Bugzappers)

Water purified



Water sprayed into air

15 motors per server bank

II. On-demand access: *aaS Classification

On-demand: renting a cab vs. (previously) renting a car, or buying one. E.g.:

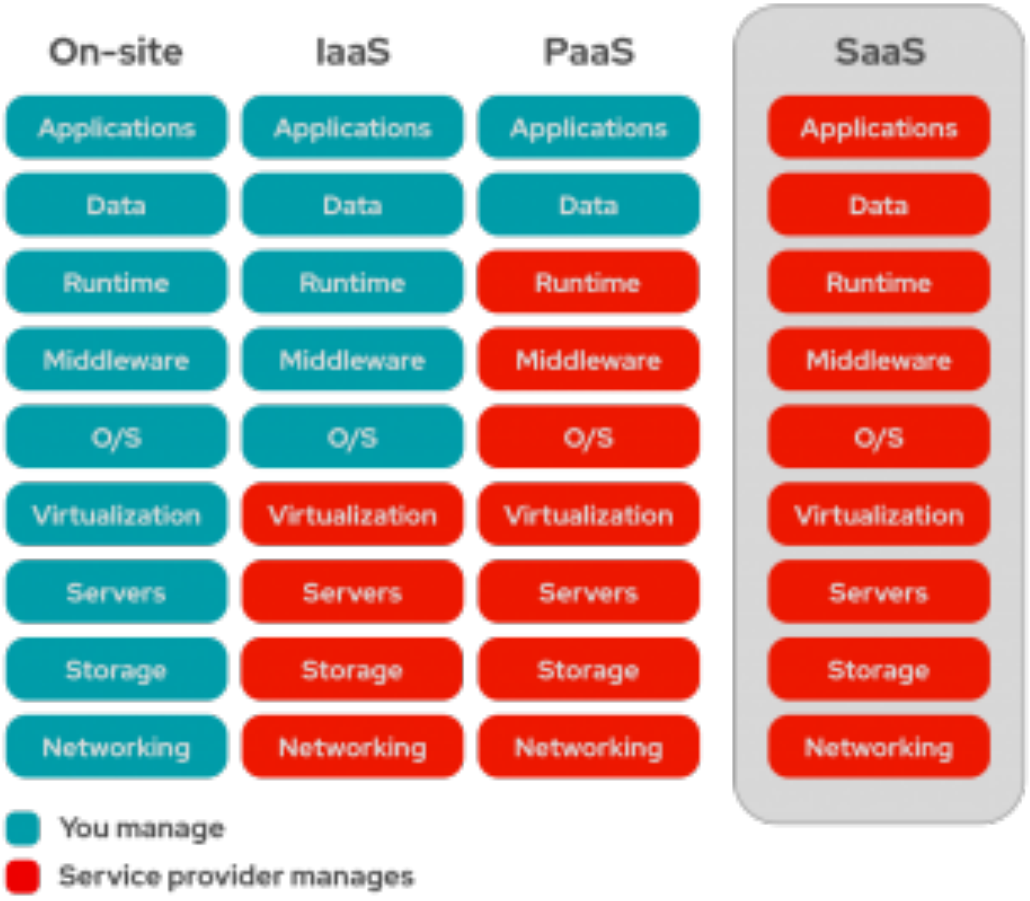
- AWS Elastic Compute Cloud (EC2): a few cents to a few $ per CPU hour
- AWS Simple Storage Service (S3): a few cents per GB-month

- HaaS: Hardware as a Service
  - You get access to barebones hardware machines, do whatever you want with them, Ex: Your own cluster
  - Not always a good idea because of security risks
- IaaS: Infrastructure as a Service
  - You get access to flexible computing and storage infrastructure. Virtualization is one way of achieving this (cgroups, Kubernetes, Dockers, VMs,…). Often said to subsume HaaS.
  - Ex: Amazon Web Services (AWS: EC2 and S3), OpenStack, Eucalyptus, Rightscale, Microsoft Azure, Google Cloud.

II. On-demand access: *aaS Classification

- PaaS: Platform as a Service
  - You get access to flexible computing and storage infrastructure, coupled with a software platform (often tightly coupled)
  - Ex: Google's AppEngine (Python, Java, Go)
- SaaS: Software as a Service
  - You get access to software services, when you need them. Often said to subsume SOA (Service Oriented Architectures).
  - Ex: Google docs, MS Office 365 Online

II. On-demand access: *aaS Classification



III. Data-intensive Computing

- Computation-Intensive Computing
  - Example areas: MPI-based, High-performance computing, Grids
  - Typically run on supercomputers (e.g., NCSA Blue Waters)
- Data-Intensive
  - Typically store data at datacenters
  - Use compute nodes nearby
  - Compute nodes run computation services
- In data-intensive computing, the focus shifts from computation to the data: CPU utilization no longer the most important resource metric, instead I/O is (disk and/or network)

IV. New Cloud Programming Paradigms

- Easy to write and run highly parallel programs in new cloud programming paradigms:
  - Google: MapReduce and Sawzall
  - Amazon: Elastic MapReduce service (pay-as-you-go)
  - Google (MapReduce)
    - Indexing: a chain of 24 MapReduce jobs
    - ~200K jobs processing 50PB/month (in 2006)
  - Yahoo! (Hadoop + Pig)
    - WebMap: a chain of several MapReduce jobs
    - 300 TB of data, 10K cores, many tens of hours (~2008)
  - Facebook (Hadoop + Hive)
    - ~300TB total, adding 2TB/day (in 2008)
    - 3K jobs processing 55TB/day
  - Similar numbers from other companies, e.g., Yieldex, eharmony.com, etc.
  - NoSQL: MySQL is an industry standard, but Cassandra is 2400 times faster!

Two Categories of Clouds

- Can be either a (i) public cloud, or (ii) private cloud
- Private clouds are accessible only to company employees
- Public clouds provide service to any paying customer:
  - Amazon S3 (Simple Storage Service): store arbitrary datasets, pay per GB-month stored
    - As of 2019: 0.4c-3 c per GB month
  - Amazon EC2 (Elastic Compute Cloud): upload and run arbitrary OS images, pay per CPU hour used
    - As of 2019: 0.2 c per CPU hr to $7.2 per CPU hr (depending on strength)
  - Google cloud: similar pricing as above
  - Google AppEngine/Compute Engine: develop applications within their appengine framework, upload data that will be imported into their format, and run

Single site Cloud: to Outsource or Own?

- Medium-sized organization: wishes to run a service for *M* months
  - Service requires 128 servers (1024 cores) and 524 TB
  - Same as UIUC CCT (Cloud Computing Testbed) cloud site (bought in 2009, now decommissioned)
- Outsource (e.g., via AWS): *monthly* cost
  - S3 costs: $0.12 per GB month. EC2 costs: $0.10 per CPU hour (costs from 2009)
  - Storage = $ 0.12 X 524 X 1000 ~ $62 K
  - Total = Storage + CPUs = $62 K + $0.10 X 1024 X 24 X 30 ~ $136 K
- Own: monthly cost
  - Storage ~ $349 K / *M*
  - Total ~ $ 1555 K / *M* + 7.5 K (includes 1 sysadmin / 100 nodes)
    - using 0.45:0.4:0.15 split for hardware:power:network and 3 year lifetime of hardware

Single site Cloud: to Outsource or Own?

- Breakeven analysis: more preferable to own if:
  - $349 K / *M* < $62 K (storage)
  - $ 1555 K / *M* + 7.5 K < $136 K (overall)

  Breakeven points
  - *M* > 5.55 months (storage)
  - *M* > 12 months (overall)

- As a result
  - Startups use clouds a lot
  - Cloud providers benefit monetarily most from storage

Public Research Clouds

- Accessible to researchers with a qualifying grant
- Chameleon Cloud: https://www.chameleoncloud.org/
  - HaaS
  - OpenStack (~AWS)
- CloudLab: https://www.cloudlab.us/
  - Build your own cloud on their hardware

Summary

- Clouds build on many previous generations of distributed systems
- Especially the timesharing and data processing industry of the 1960-70s.
- Need to identify unique aspects of a problem to classify it as a new cloud computing problem
  - Scale, On-demand access, data-intensive, new programming
- Otherwise, the solutions to your problem may already exist!

Virtualization and Containerization

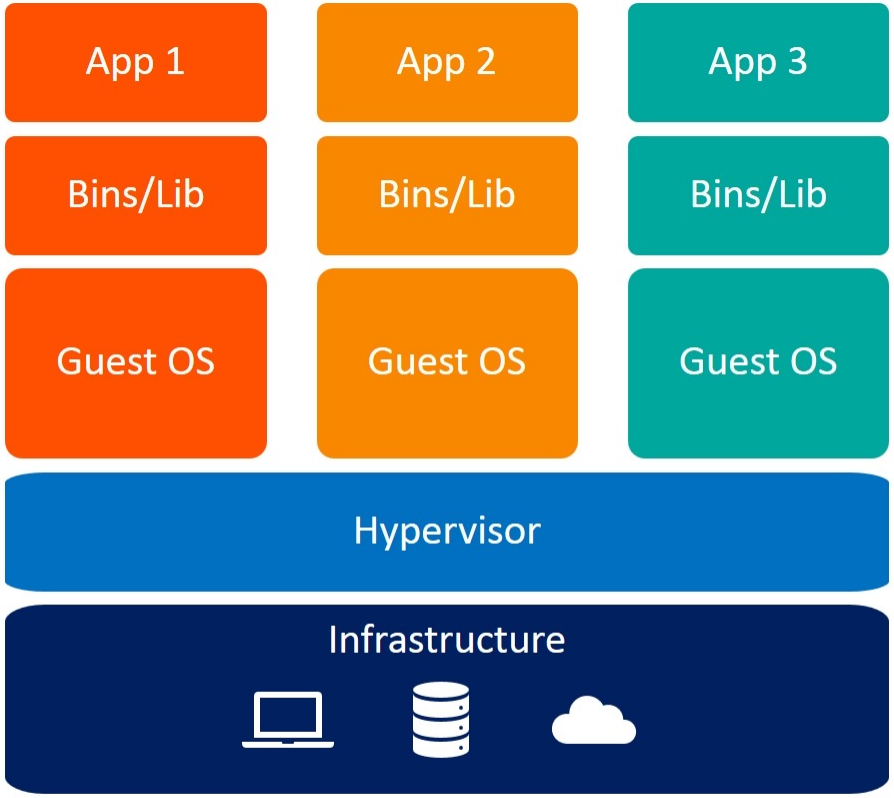
Virtualization vs. Containerization

- Virtualization creates software-based or virtual versions of a computer resource
  - Hypervisors take physical resources and divide them up
  - Run different operating systems on the same hardware
  - Each guest OS is controlled through elevated security measures
- Containerization is a lightweight alternative to virtualization. Containerization encapsulate an application in a container with its own operating environment.
  - Each container is an executable package of software that runs on top of a host OS.
  - Containerization evolved from a Linux feature known as cgroups, which is designed for isolating and controlling resource usage.

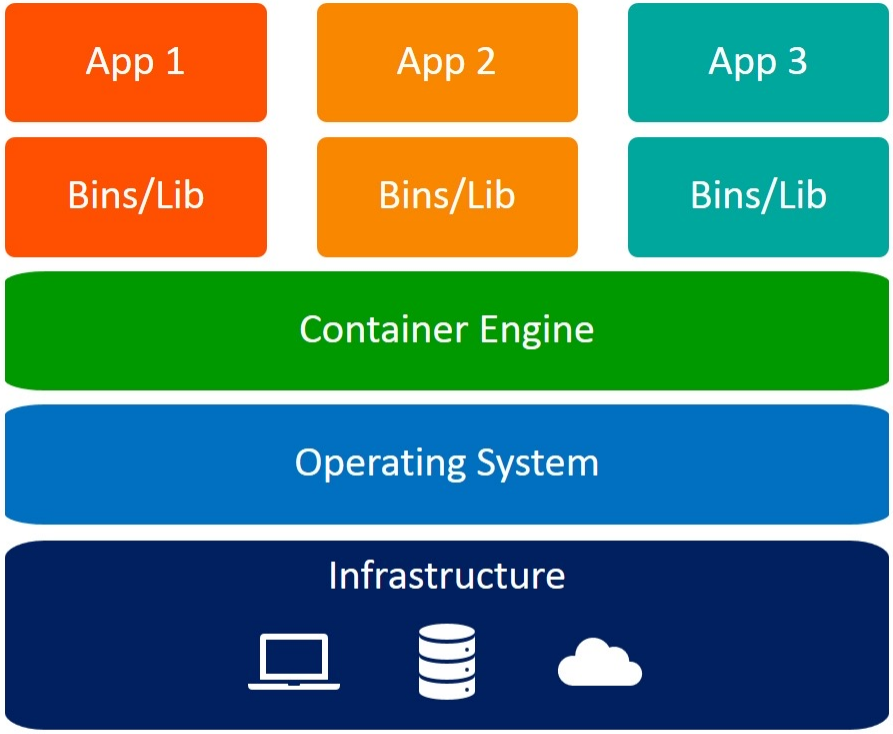
Virtualization vs. Containerization

Area	Virtualization	Containerization
Isolation	Provides complete isolation from the host operating system and the other VMs	Typically provides lightweight isolation from the host and other containers, but doesn't provide as strong a security boundary as a VM
Operating System	Runs a complete operating system including the kernel, thus requiring more system resources such as CPU, memory, and storage	Runs the user-mode portion of an operating system, and can be tailored to contain just the needed services for your app using fewer system resources
Guest compatibility	Runs just about any operating system inside the virtual machine	Runs on the same operating system version as the host
Deployment	Deploy individual VMs by using Hypervisor software	Deploy individual containers by using Docker or deploy multiple containers by using an orchestrator such as Kubernetes
Persistent storage	Use a Virtual Hard Disk (VHD) for local storage for a single VM or a Server Message Block (SMB) file share for storage shared by multiple servers	Use local disks for local storage for a single node or SMB for storage shared by multiple nodes or servers
Load balancing	Virtual machine load balancing is done by running VMs in other servers in a failover cluster	An orchestrator can automatically start or stop containers on cluster nodes to manage changes in load and availability.

VM vs. Container



Virtual Machines



Containers

Cloud Native

- *Cloud-native technologies empower organizations to build and run scalable applications in **modern, dynamic environments** such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.*

- *These techniques enable **loosely coupled systems** that are **resilient, manageable, and observable**. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.*

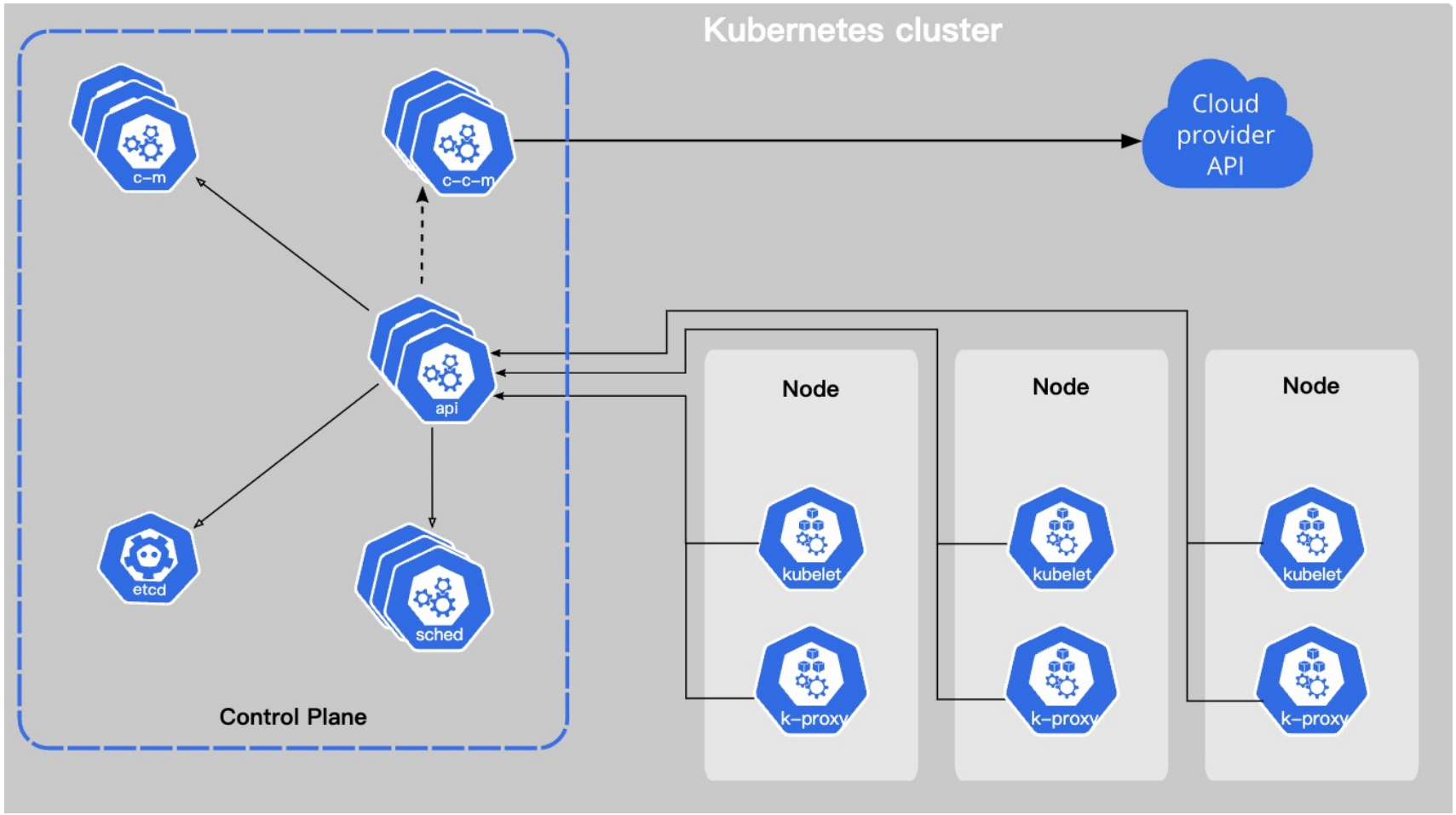
Why Should You Use Containers?

- Cloud-native environments
- Microservices
- DevOps workflow
- Job management
- Potability and usability

Kubernetes

- open-source orchestration systems for containers developed by Google and open-sourced in 2014
- All major cloud providers support Kubernetes
- Has a variety of good features
  - High availability
  - Auto-scaling
  - Rich ecosystem
  - Service discovery
  - Container health management
  - Configuration management

Kubernetes


Kubernetes cluster
c-m
c-c-m
Cloud provider API
api
etcd
sched
Control Plane
Node
Node
Node
kubelet
kubelet
kubelet
k-proxy
k-proxy
k-proxy

Microservices

- Serverless: a cloud-native development model that allows developers to build and run applications without having to manage servers.
- Characteristics:
  - Small
  - Autonomous
  - Specialized
- Benefits:
  - Speed
  - Simplicity
  - Maps logic
  - Reliability

Exercise

- Check the GitHub repo https://github.com/noahgift/container-from-scratch-python
  - Build the container yourself and run it

```
jingzhuhe@Jingzhus-MacBook-Pro container-from-scratch-pyth
on % docker run -it hello-duke-cli-210 python app.py --nam
e "Jingzhu"
Hello Jingzhu!
```

  - Pull an image and run it


```
root@6cdaf3f6ba8a:/app# python app.py --name "Jingzhu"
Hello Jingzhu!
```

Exercise

- Pass a command and run it

```
jingzhuhe@Jingzhus-MacBook-Pro container-from-scratch-pyth
on % docker run -it noahgift/cloudapp python app.py --name
 "Jingzhu"
WARNING: The requested image's platform (linux/amd64) does
 not match the detected host platform (linux/arm64/v8) and
 no specific platform was requested
Hello Jingzhu!
```

Search In Use only

NAME		TAG	IMAGE ID	CREATED	SIZE
hello-duke-cli-210	IN USE	latest	97816dab25ad	13 minutes ago	902.89 MB
noahgift/cloudapp	IN USE	latest	2a92410f810e	over 1 year ago	951.85 MB