

Cloud Computing Theory and Practice

INSY 5345 & INSY 4307

DR. SANTOSO BUDIMAN



INSY 5345 Cloud Computing Theory and Practice



**Dr. Santoso Budiman
COBA 511
santoso.budiman@uta.edu**

Topics

Section 1: Cloud Computing Introduction

Section 2: Computer Basics

Section 3: Virtual Machines

Section 4: Networking and IaC

Section 5: Elasticity & High Availability

Section 6: Cloud Storages

Section 7: Cloud Databases

Section 8: Containers

Section 9: Kubernetes & Infrastructure as Code

Section 10: Serverless

Cloud Computing Introduction

SECTION 1

Section 1

- Cloud Introduction
 - Digital Transformation
 - What is Cloud Computing
 - Pyramid Service Model
 - Cloud Computing Reliability
 - Cloud Migration
- Public Cloud Provide – we use aws in this class
- Cloud Computing Vs. Edge Computing
- Cloud Engineer Roles
- Cloud Architecting
 - AWS Well-Architected Framework (Guide to design a solution)
 - Best practices for building solutions

Digital Transformation

Digital Transformation

The adoption of digital technology by a company to improve business processes

Digital technologies: electronic systems that generate, store or process data

Digital Technology continues to evolve

Organizations must keep up

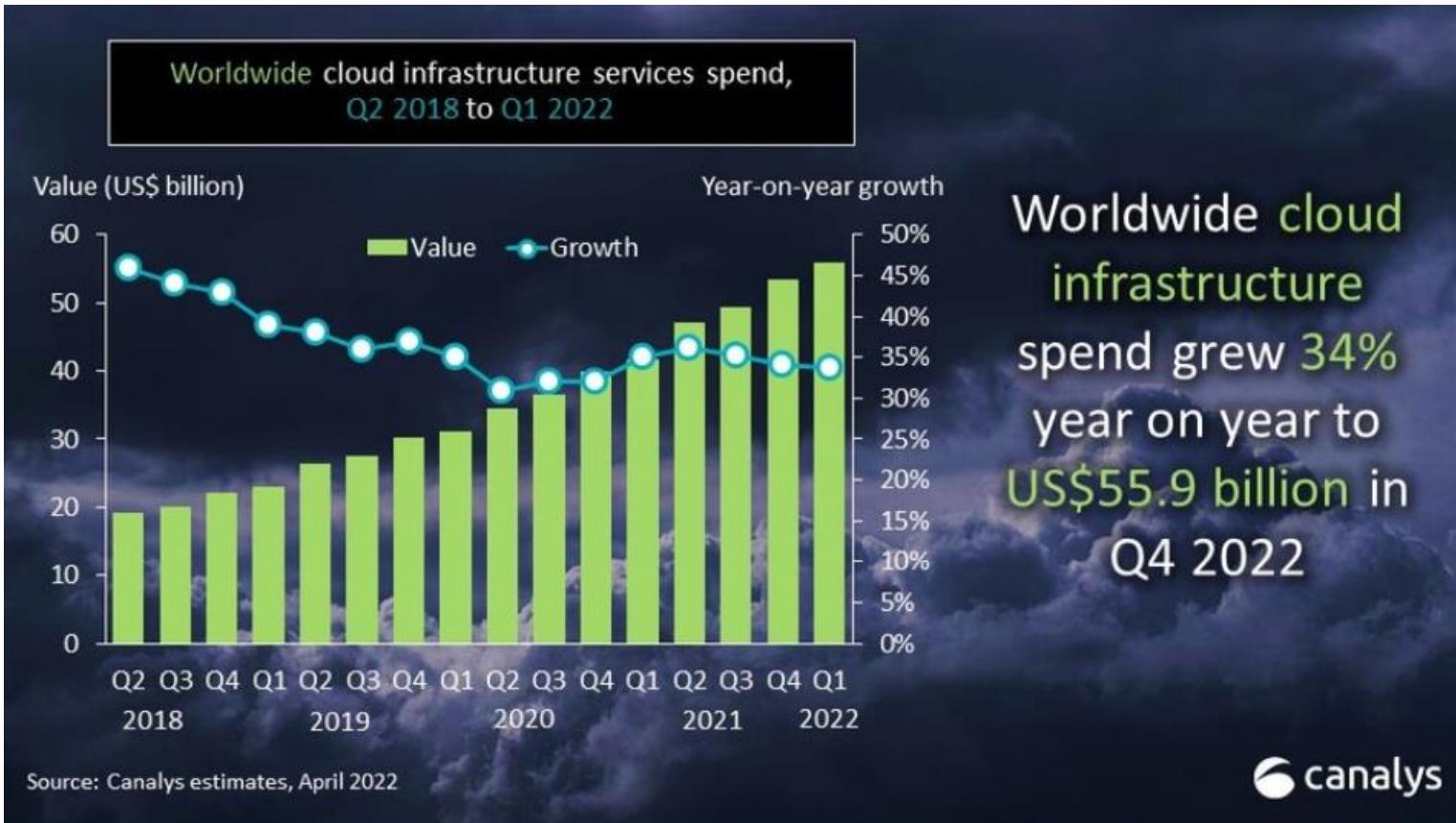
Cloud Computing is a Core component of Digital Transformation

Cloud Computing – Digital Transformation

“Cloud infrastructure services provide the core components needed to support businesses’ digital transformation initiatives around building new customer experiences, deploying IoT to transform processes, using big data and analytics for better insights, and embedding machine learning and AI for automation,” Canalys Principal Analyst Matthew Ball.

https://www.canalys.com/static/press_release/2019/pr20190204.pdf

Cloud Global growth



<https://www.canalys.com/newsroom/global-cloud-services-Q1-2022>

Data volume in zettabytes

150

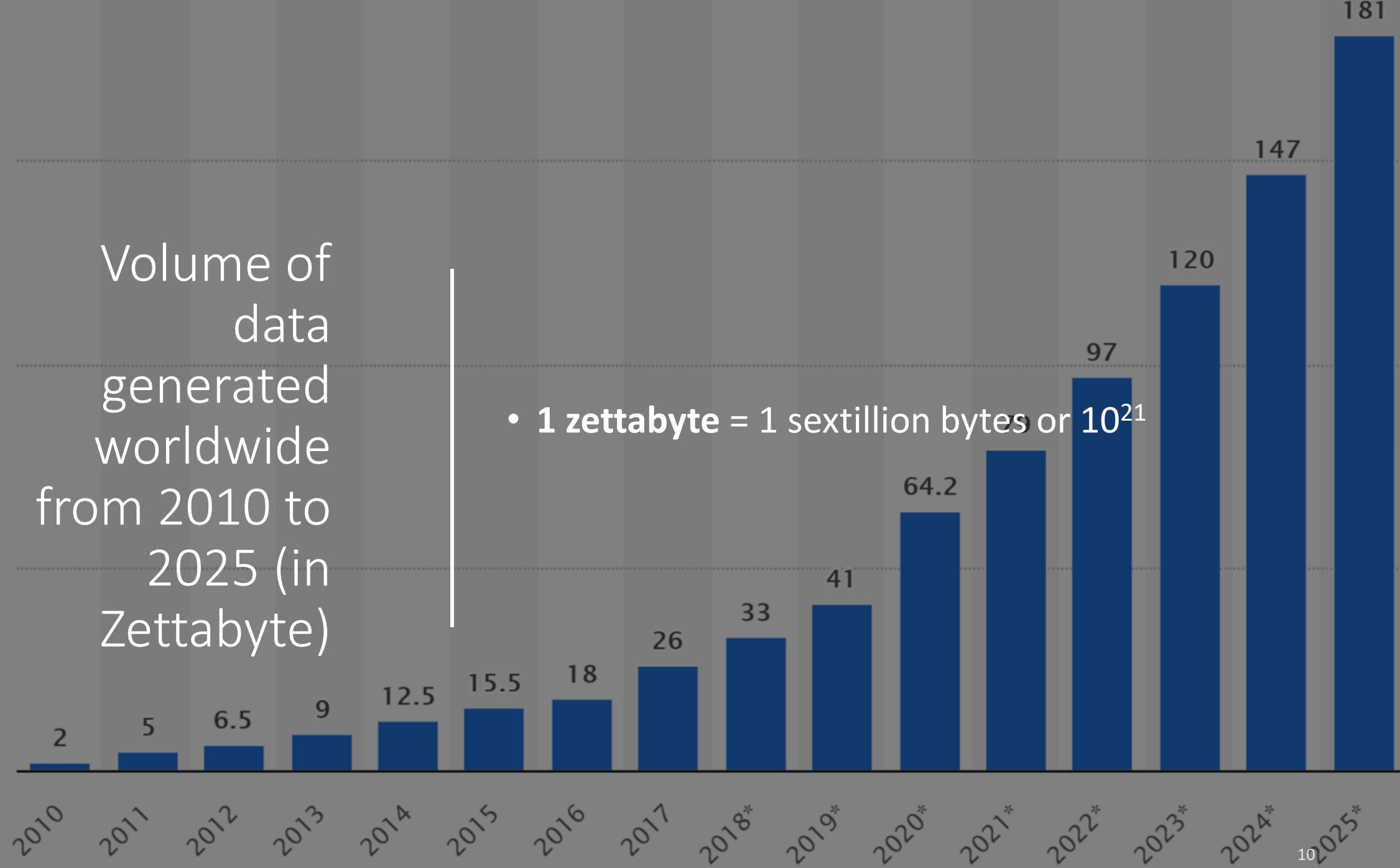
Volume of
data
generated
worldwide
from 2010 to
2025 (in
Zettabyte)

100

50

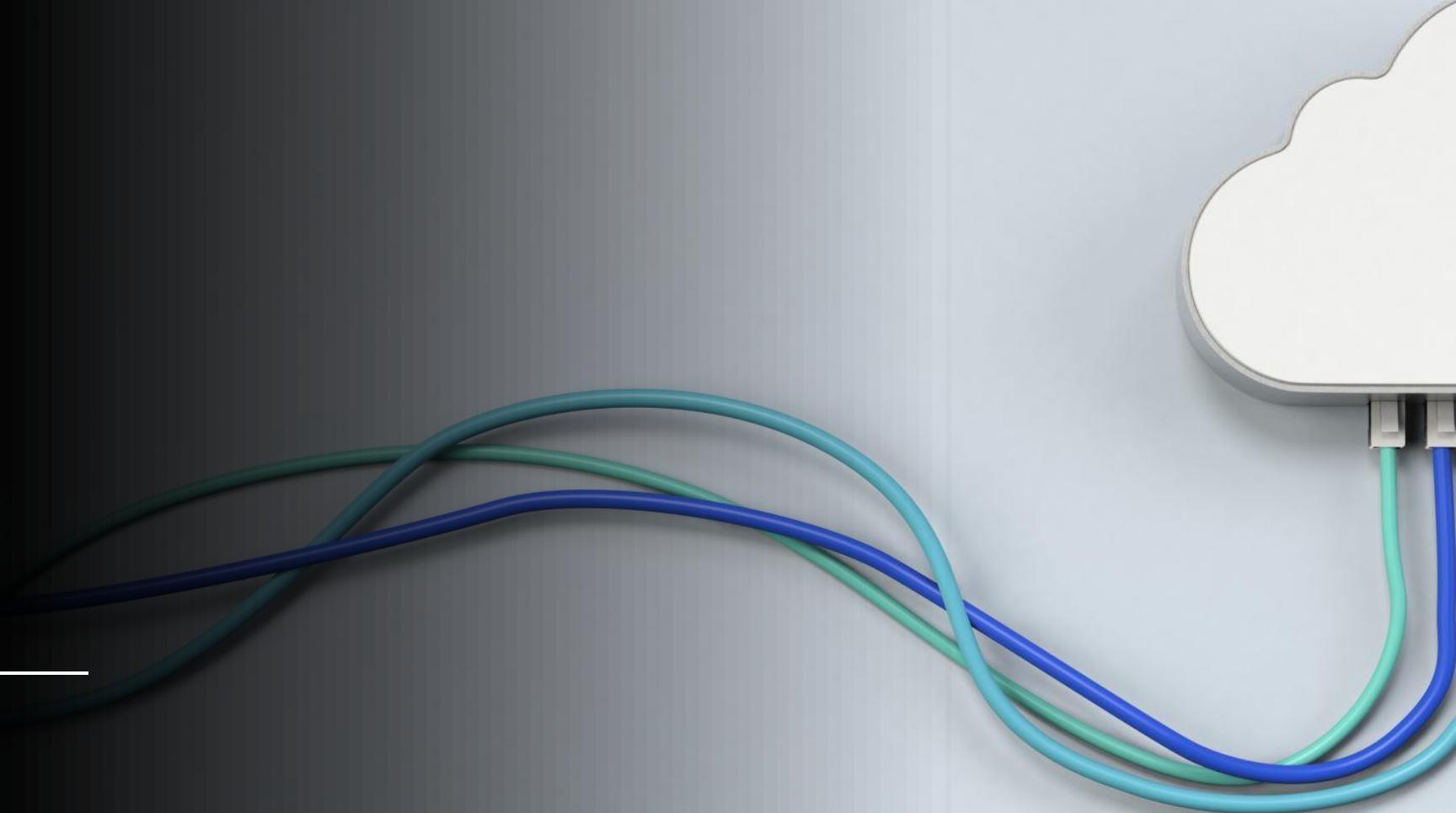
0

- **1 zettabyte = 1 sextillion bytes or 10^{21}**





What is Cloud Computing



What is Cloud Computing

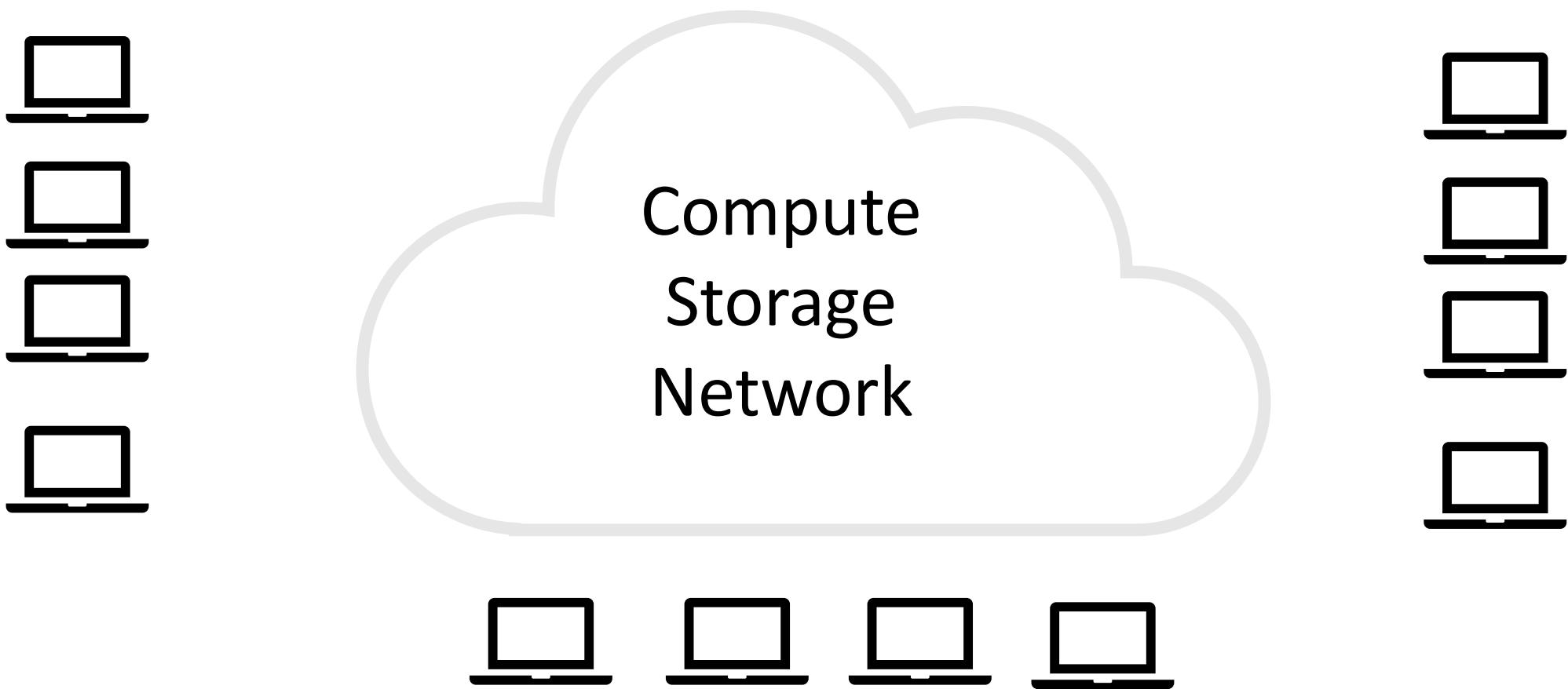
“Ubiquitous, convenient, on-demand access to shared computing resources that can be rapidly provisioned and released with minimal management effort”

<https://csrc.nist.gov/publications/detail/sp/800-145/final>

-- Cloud computing is the on-demand delivery of IT resources available from the Internet with a pay-as-you-go model --



Cloud Computing



Cloud Characteristics

A Cloud must have the following 5 characteristics (NIST)

Rapid Elasticity	Measured Service	On Demand Self Service	Ubiquitous Network Access	Resource Pooling
<ul style="list-style-type: none">• The ability to scale resources• To consumers, Cloud resources are infinite	<ul style="list-style-type: none">• Services are controlled and measured by the provider• Customers are billed based on that	<ul style="list-style-type: none">• Consumers can use Cloud services without human interaction	<ul style="list-style-type: none">• Capabilities are available over the internet• Can be accessed by both thick and thin clients	<ul style="list-style-type: none">• Can do multi-tenant model• Resources can be assigned based on demand• A sense of location independent

Elasticity and Scalability

Elasticity is defined as the ability to scale resources in and out

Elastic computing allows compute resources to vary dynamically to meet variable workload

Cloud elasticity – through resource pooling

Elasticity allows an organization to scale resources in and out as required

example: Coca-Cola

<https://www.youtube.com/watch?v=jm94IAqQ7gk>

Cloud Computing Benefits

Financial	Capacity	Operations and Management	Business Needs
<ul style="list-style-type: none">• CAPEX vs OPEX• Less upfront cost• Shared cost• Pay as you go	<ul style="list-style-type: none">• Big Data• Elasticity• Scalability	<ul style="list-style-type: none">• Focus on application• Remove headache on hardware• Performance• Security• Integration complexity• Service-Level Agreements (SLA)	<ul style="list-style-type: none">• Digital Transformation• Customer Experience• Business Analytics• IOT• Time to deployment• Compliance regulatory• Geographical reach

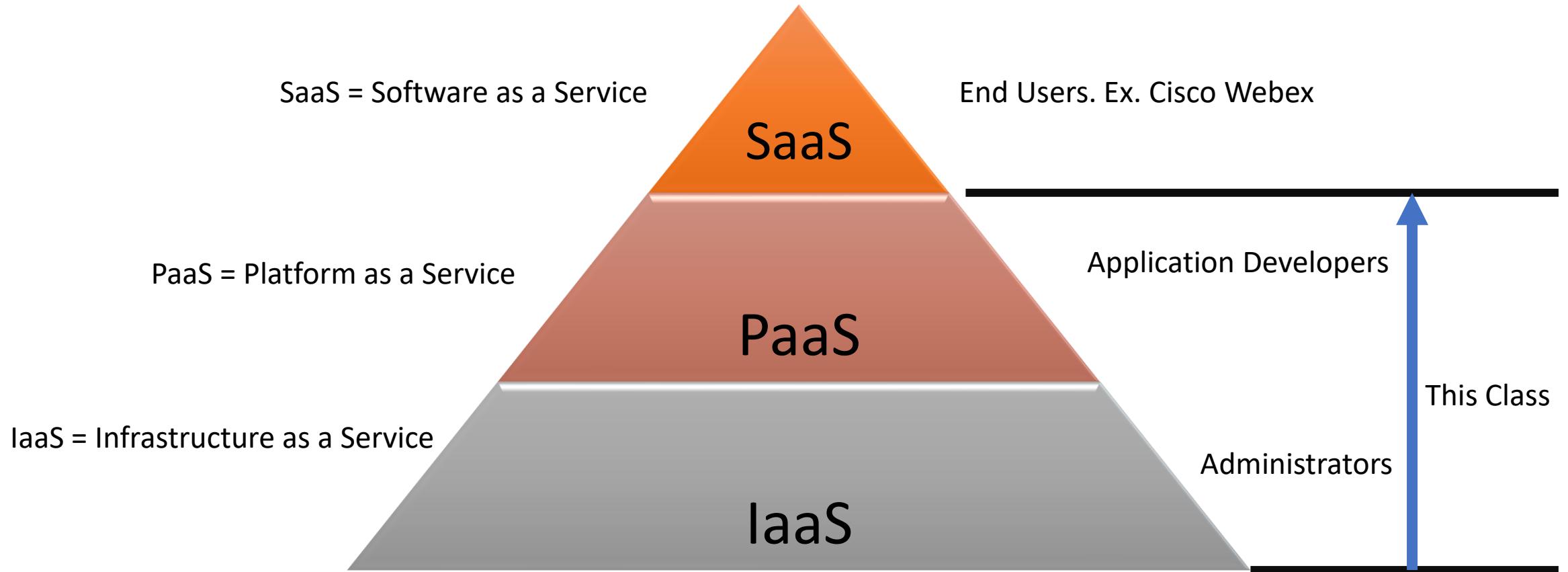


Cloud Pyramid Service Model

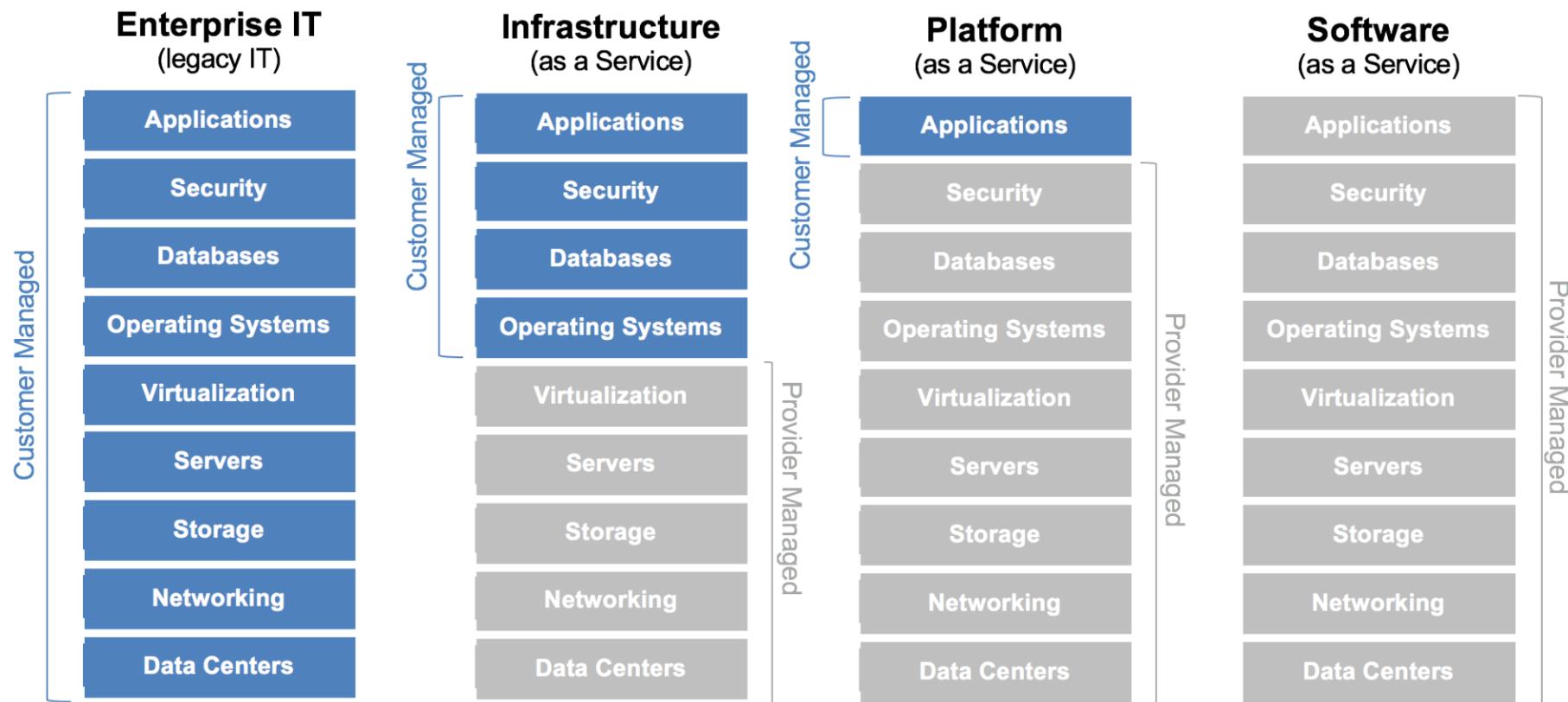


Cloud Computing – Service Model

A Cloud service model is a set of IT-related services offered by a cloud provider

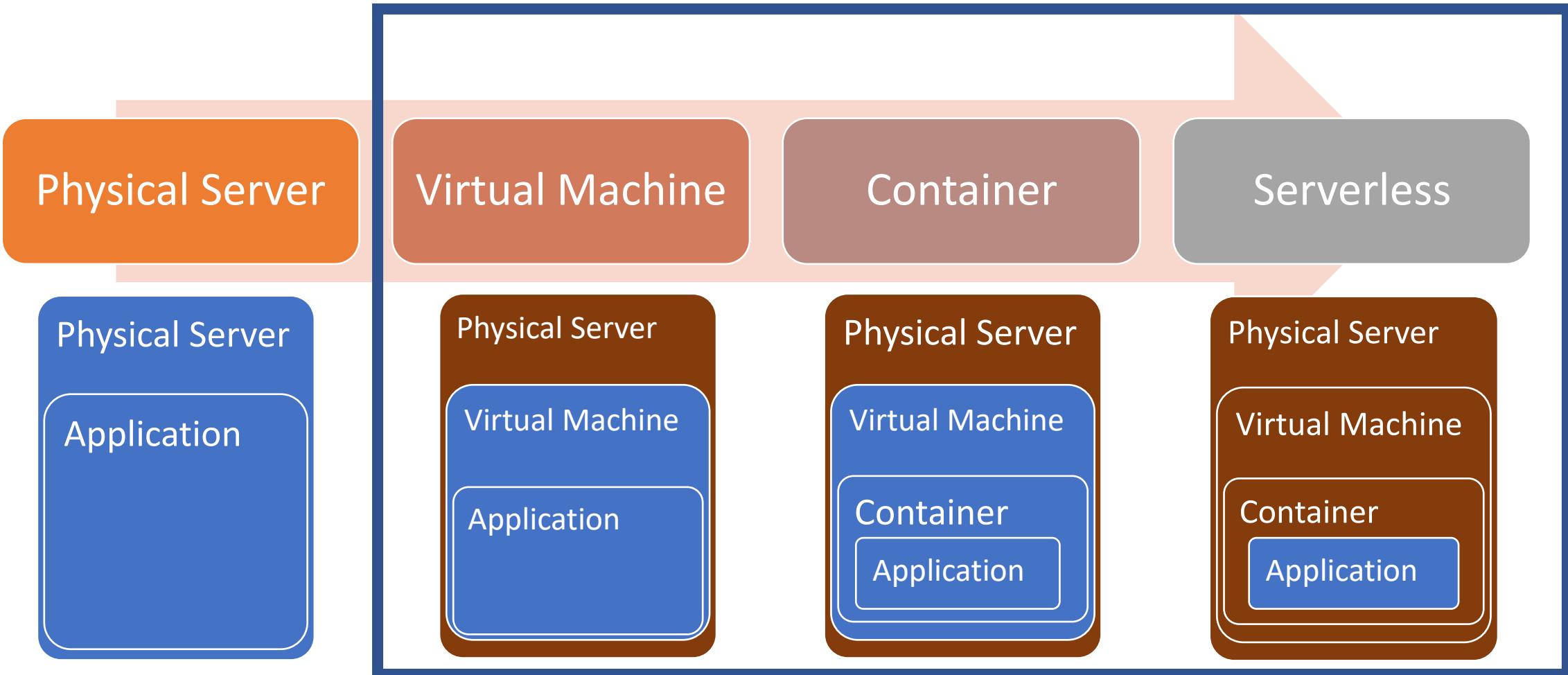


Legacy to SaaS

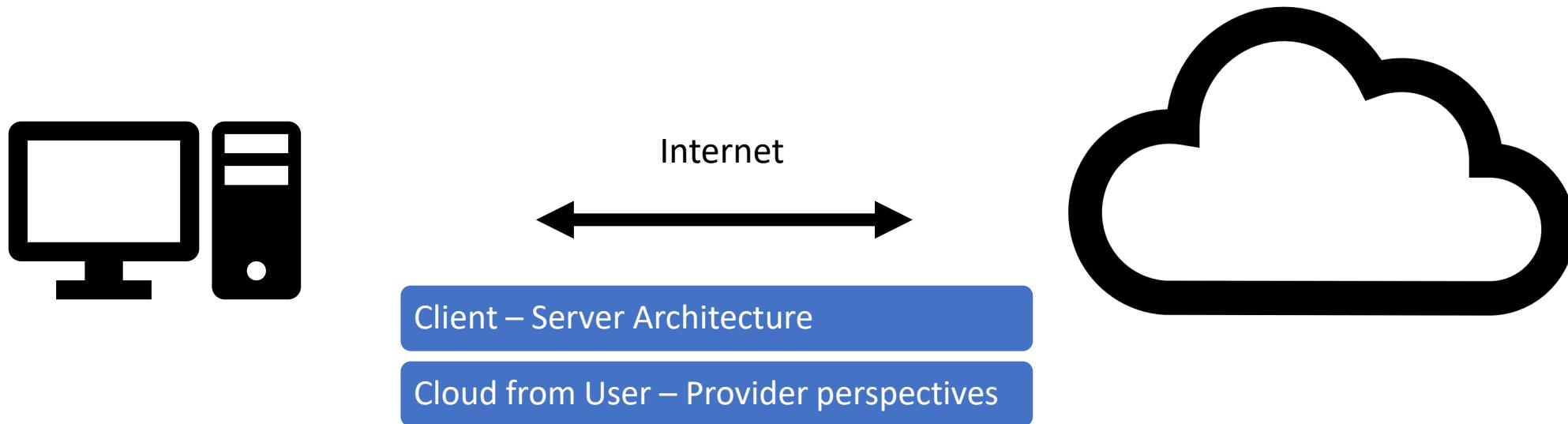


<https://mycloudblog7.files.wordpress.com/2013/06/screen-shot-2015-06-09-at-2-13-05-pm1.png>

Infrastructure - Application



Client - Server



Internet has revolutionized the way computers are used.

Client–Server model is a **distributed application structure** that partitions tasks or workloads between the providers of a resource or service - called servers, and service requesters - called clients.

- Server – bigger computers that can simultaneously support multiple users
- Client – computers that are generally used by individuals: desktops, laptops, smartphones

Cloud Computing Reliability

Cloud Computing Availability

Service Level Agreement (SLA) : a contract between a customer and the service provider

- The service provider is required to provide services per the negotiated quantity and quality ranges for a given price

Availability : the ratio of uptime/total time.

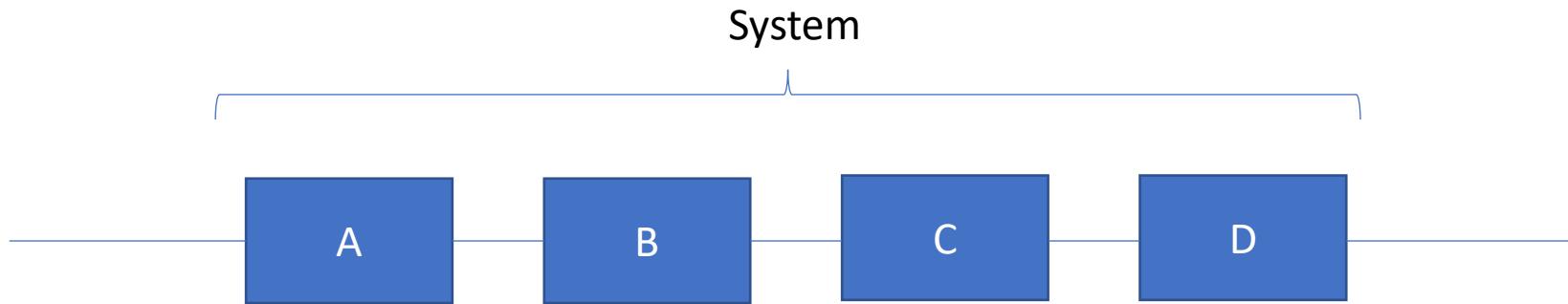
xx.xxxxxx%

- **Uptime** is the total time minus downtime
- Availability ratio in percentage is measured by 9's.
 - 90% has one 9
 - 99% has two 9s

9's	Availability(90%)	Downtime per year	Examples
1	90.0	36 days 12 h	Personal Computers
2	99.0	87 h 36 min	Entry level business
3	99.9	8 h 45.6 min	ISPs, mainstream business
4	99.99	52 min 33.6 s	Data Centers
5	99.999	5 min 15.4 s	Banking, medical
6	99.9999	31.5 s	Military defense

5 9s downtime =0.001%
 $(365 \times 24 \times 60 \times 60) = 315.36 \text{ s} = 5 \text{ min } 15.36 \text{ s}$

System Availability – Series Systems



A Availability = A_a – ex 50%

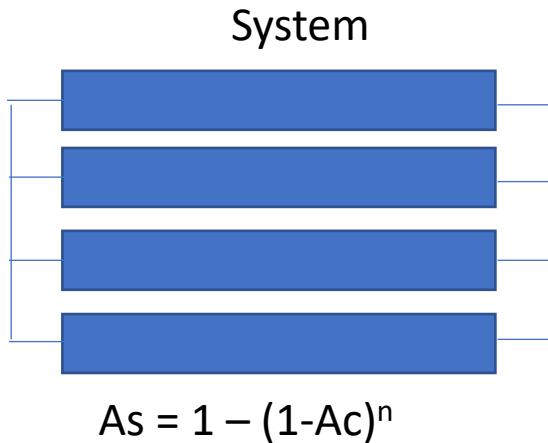
B Availability = A_b – ex 60%

C Availability = A_c – ex 70%

D Availability = R_d – ex 80%

System Availability (A_s) = $A_a \times A_b \times A_c \times A_d = 50\% \times 60\% \times 70\% \times 80\% = 16.8\%$

System Availability – Parallel Systems

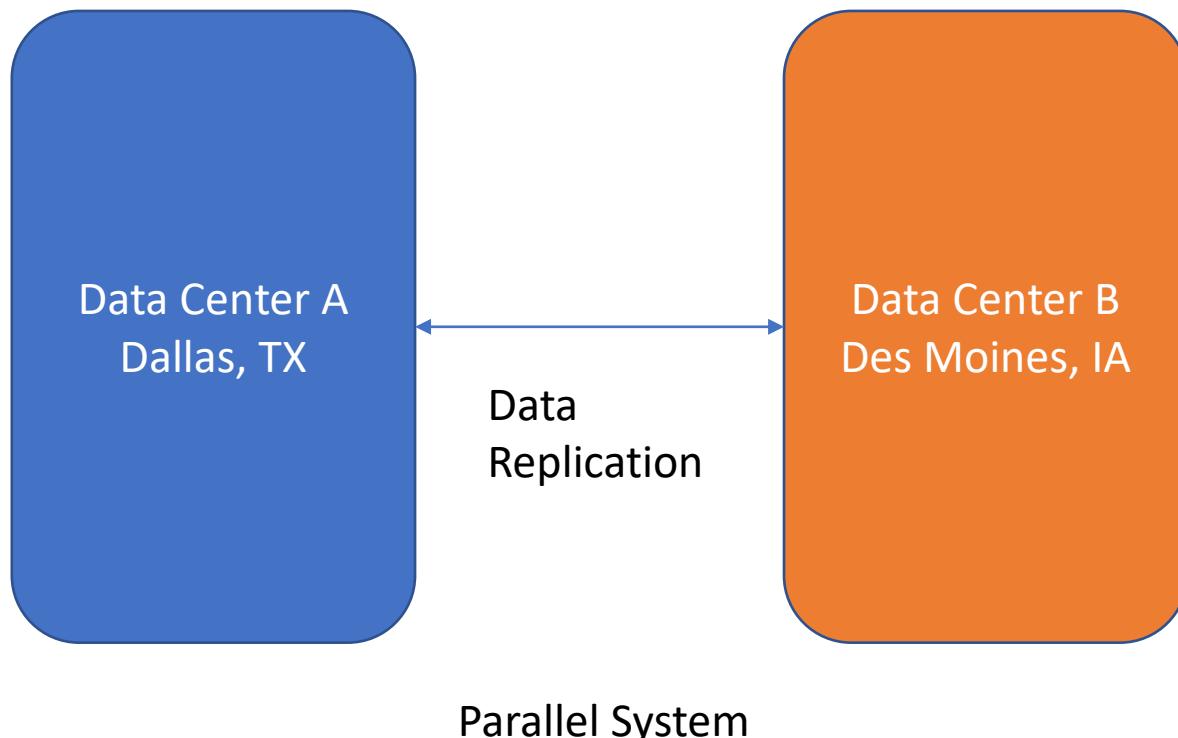


For example:

- To increase availability (High Availability), 4 components are connected in parallel configuration
- If component is down, the others can be used
- The Availability of each component is **Ac**. For example 0.9
- The Availability of the system (**As**) is
 - $1 - (1-0.9)^4 = 1-(0.1)^4 = 0.9999$
- So availability improves from 1 9 to 4 9s

Geo-redundancy

Geo-redundancy signifies a computer system operating at two or more geographical locations as a **redundancy** in case the primary system fails due to any reason.

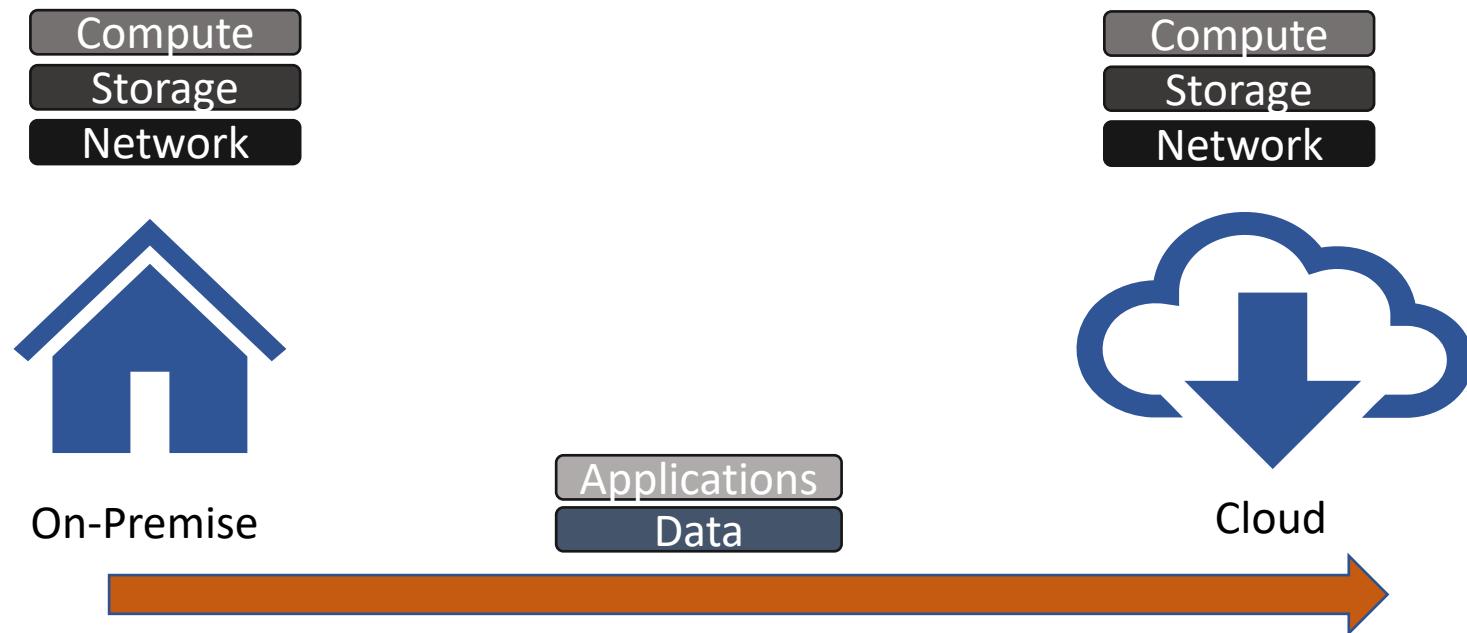




Cloud Migration



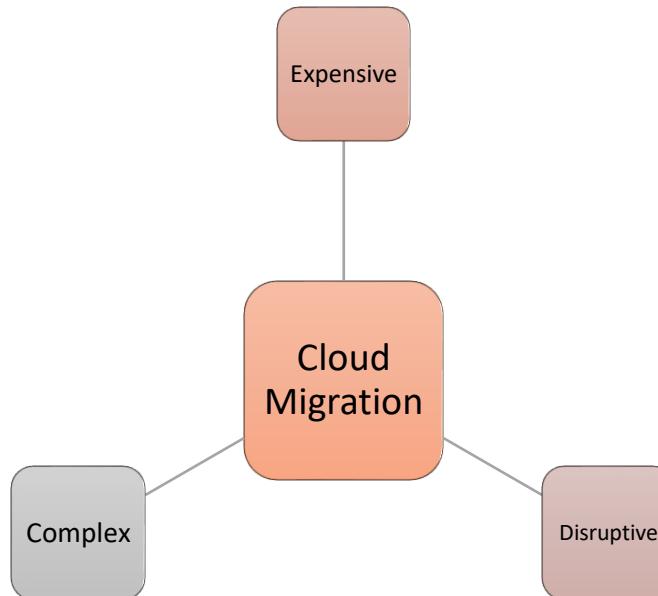
Migration to Cloud



Cloud Migration

Cloud migration is the process of moving data, applications or other business elements to a **cloud** computing environment.

Migration must be planned properly



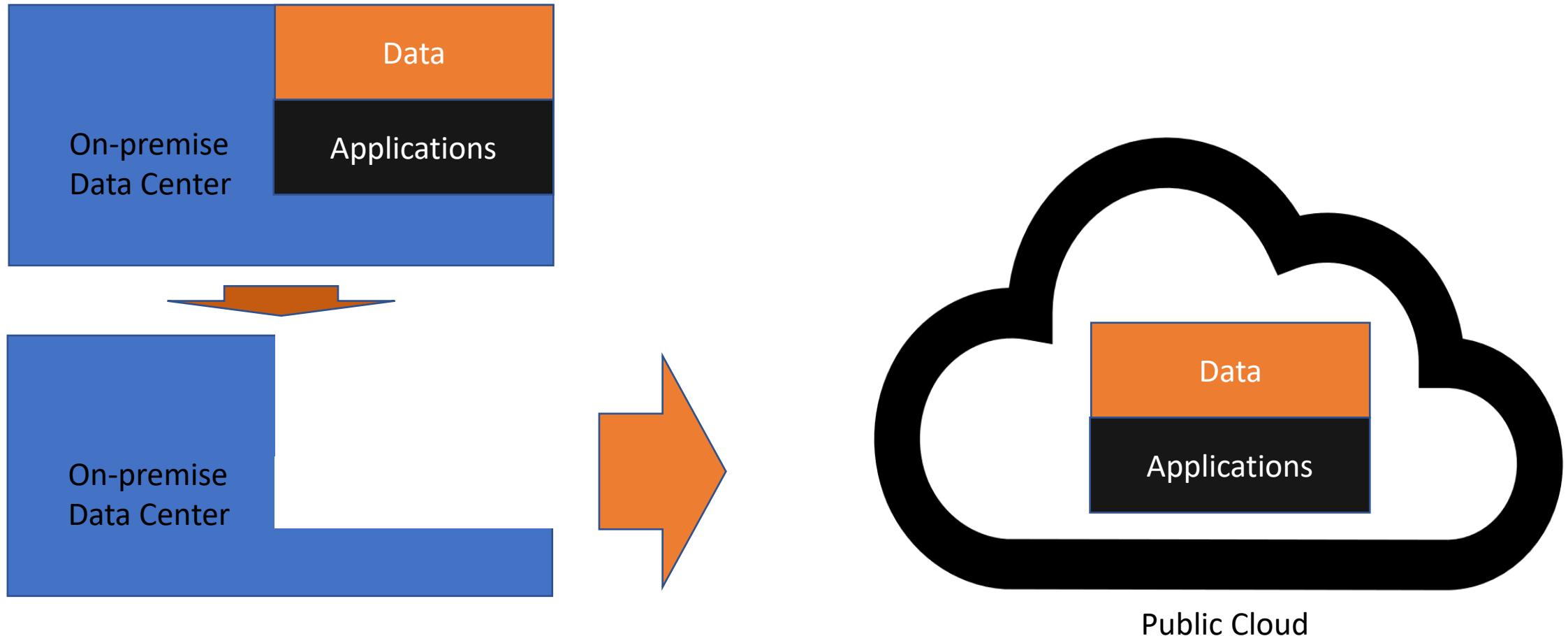
When to migrate to Cloud Computing

Due Diligence must be performed to weigh benefits and risks

Business Case As-Is Vs.
To-Be

Migrate after performing
Due Diligence with clear
objectives

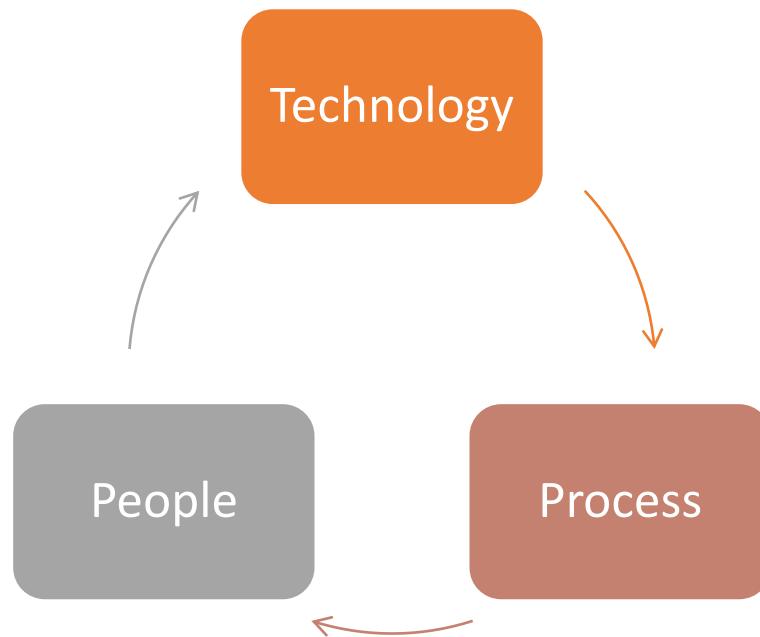
Migration – On-Prem to Cloud



Impact on Organization

A Paradigm Shift

- Impacts everyone





Data Centers

- There are over 1 M Data Centers all over the US (from 1 server to over 1M sq ft area).
- Google Data Center youtube video (the above picture is not Google):
 - <https://www.youtube.com/watch?v=XZmGGAbHqa0>
- Facebook Data Center Fort worth (450K sqft on 150 acres campus):
 - <https://www.facebook.com/FortWorthDataCenter/videos/fort-worth-data-center-grand-opening/1370391999721199/>

Areas to be addressed to be competitive with internal dedicated servers (On-Premise)

Network Latency

- Customer demand high bandwidth
- Weakest link must be identified and fixed

Fine-grained Migration and Provisioning

- Plan to cope with massive data during migration

Standards and Open-source solutions for Cloud SW

- Customer should be able to migrate from provider to provider

Offline versus Online synchronization of data

- Internet and power are not 24x7 guaranteed in some markets such as emerging markets

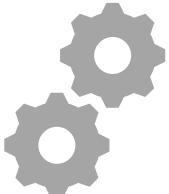
Migration Transition and Transformation Stages



Transition

Status quo –As-Is (process and system)

Preparation for change (new process creation, implementation (not activated yet), etc.)



Transformation

The change of process and system
Both old and new processes and systems are going in parallel



Normal Operation

Cut-over
New process and system are the only thing working

Design Migration: Application Migration Strategies

Rehosting	Replatforming	Repurchasing	Refactoring/Re-architecting	Retire
<ul style="list-style-type: none">• Lift and shift• Apply legacy systems to a new platform• Many can be automated• May not be the optimum (not utilizing Cloud Native capabilities), may want to modernize	<ul style="list-style-type: none">• Core architecture still maintained• Optimizations is performed	<ul style="list-style-type: none">• Moving to a new product• For example, move to SaaS platform	<ul style="list-style-type: none">• Core architecture are modified to enable new features (modernization).• the most advanced method of migrating applications to the cloud.• Re-architecture application fully exploits cloud innovation.	<ul style="list-style-type: none">• The legacy application is no longer needed.• Maybe the function is no longer needed or moved to a new product (Repurchasing)

Design Migration: Cloud Migration issues

When designing migration plan these must be considered:

Down Time (potential down time during migration to be planned and avoided)

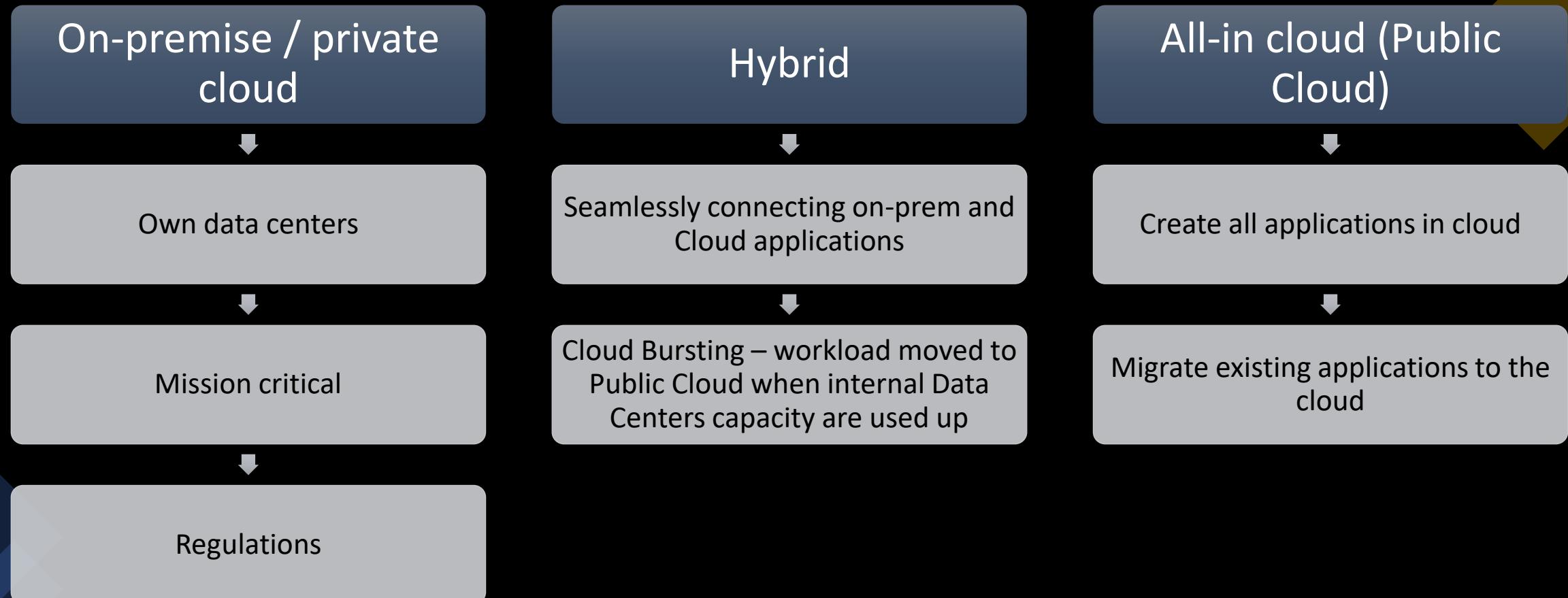
Data Protection and Security (systems maybe vulnerable during migration)

Interoperability (applications ability to communicate with each other)

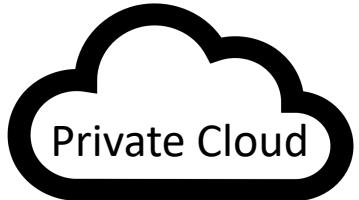
Adaptability (process, organization readiness for the change)

Fall Back Plan (In case of problems what needs to be done)

3 Cloud Computing Deployment Models



Cloud Deployment – Private Cloud



Owned by a single organization

Enables central access to IT resources from many locations

Utilizes internal resources

Similar benefits as public cloud without relinquishing control, security, and recurring costs to a cloud provider.

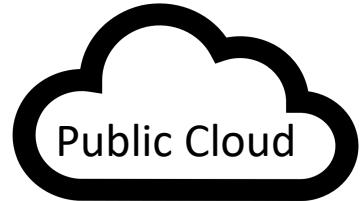
Driven by the need to maintain control of the environment because regulatory or business reasons (Health care, banks, etc.)

Combination of virtualization, data center automation, chargeback metering and identity-based security.

Virtualization allows easy scalability, flexible resource management, maximum hardware utilization.

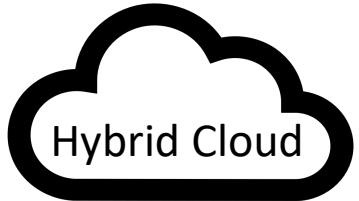
Some organizations use private clouds to share storage between internal departments (Private Cloud Space (PCS)).

Cloud Deployment – Public Cloud



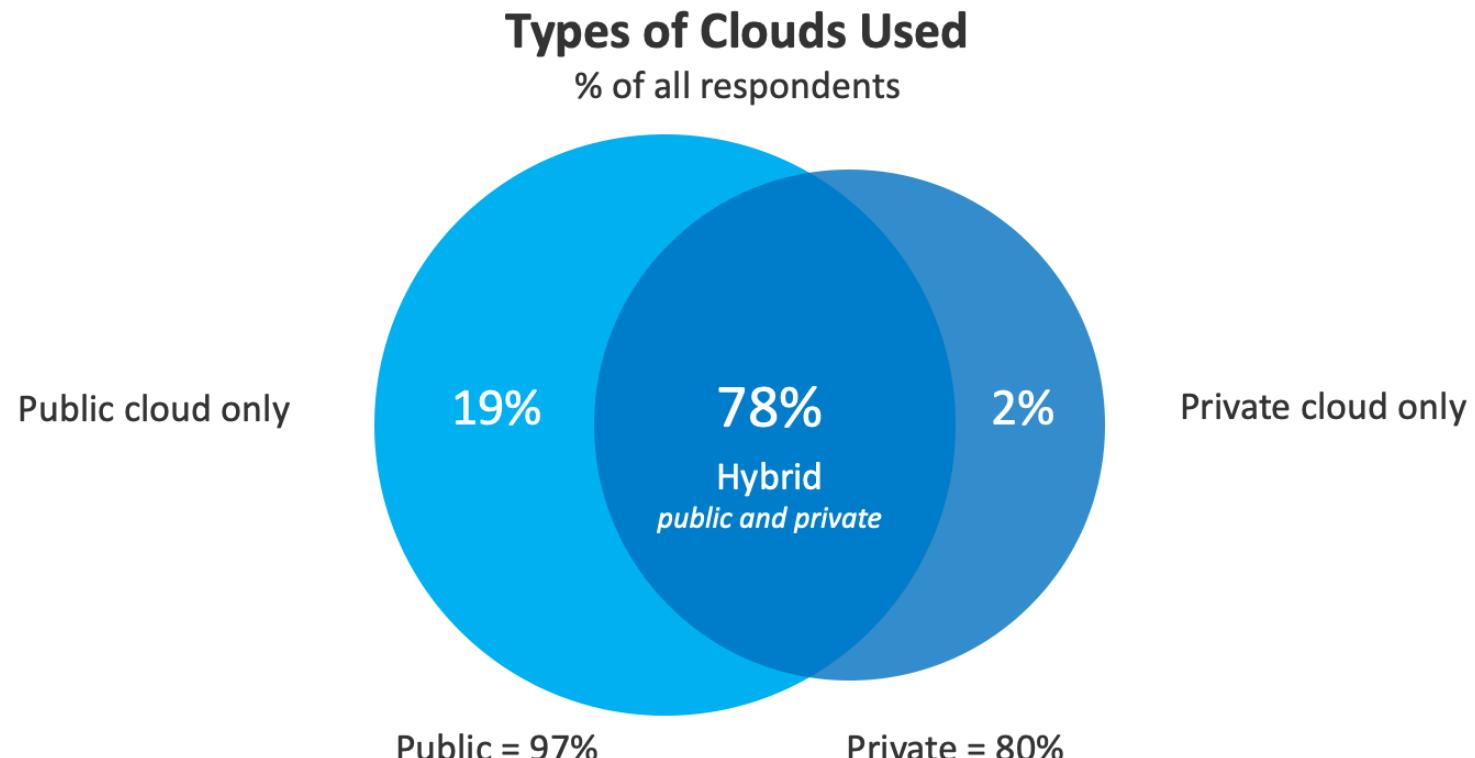
- Pool of computing services delivered over the internet via a cloud provider
- Pay-as-you-go model
- Easy and inexpensive setup because the hardware, application, and bandwidth costs are covered and maintained by the cloud provider
- Public clouds are used typically when organization less likely need the level of infrastructure and security offered by private clouds
- Offer scalability – cloud resources are available on demand from vast pool of resources
- Public cloud providers: Microsoft Windows Azure, Google Cloud Platform, Amazon Web Services, etc.

Cloud Deployment – Hybrid Cloud



- Cloud service that utilizes both private and public clouds to perform distinct functions with the same organization
- An organization continues to provide and manage some resources internally while other resources are provided externally
- Hybrid cloud allows an organization to take advantage of the scalability and cost effectiveness of a public cloud without exposing mission-critical data to a public cloud provider.
- **Not** considered a **hybrid** if an organization uses **SaaS** applications (such as CRM, ERP, Finance) and does not move the data to a private cloud or internal data center
- A cloud environment is labeled a **hybrid cloud only if there is a combination of private/on-premise and public clouds or if data is moved between the internal data center and public cloud.**

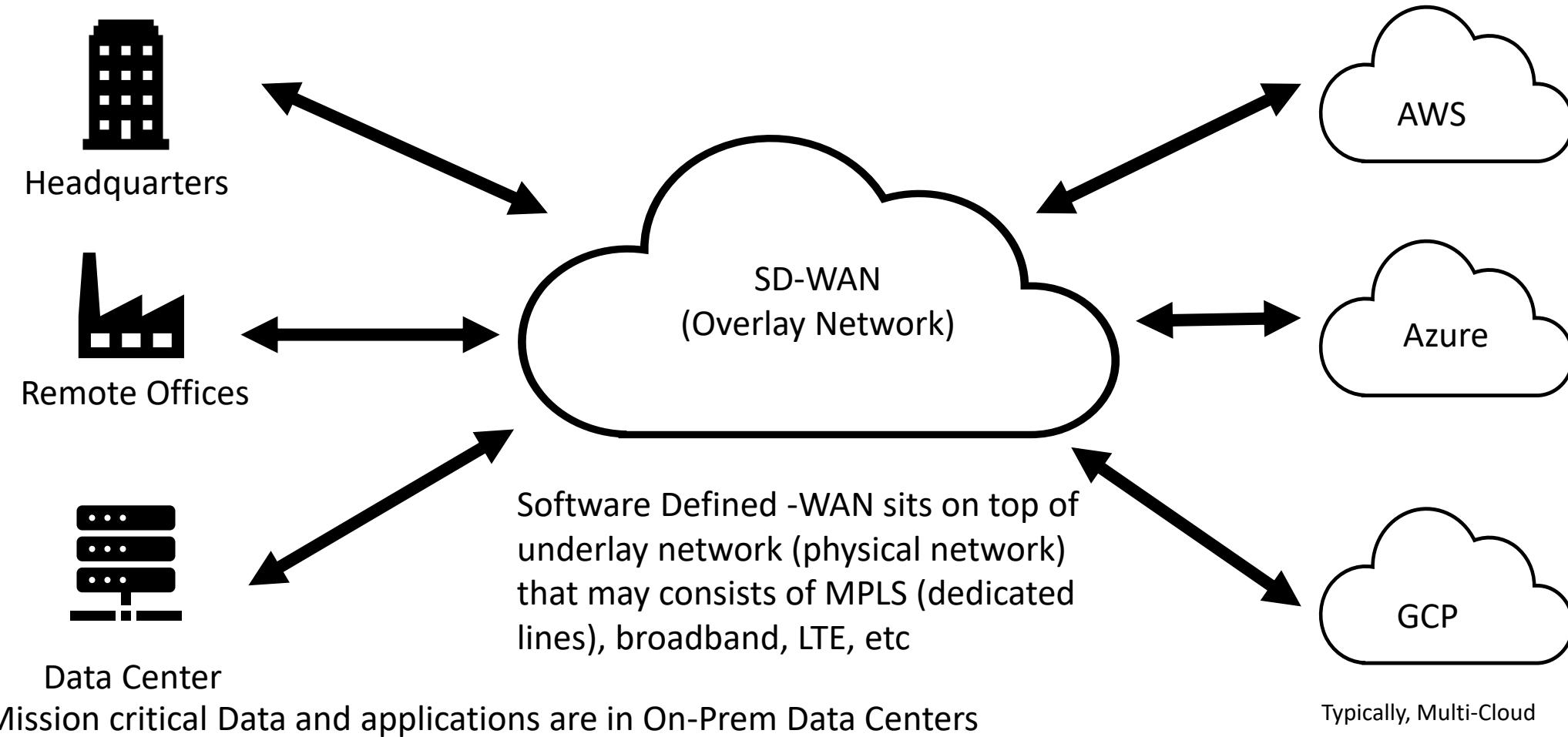
2021 Survey (750 US companies)



Source: Flexera 2021 State of the Cloud Report

https://info.flexera.com/CM-REPORT-State-of-the-Cloud?lead_source=PPC&utm_source=google&utm_medium=cpc&utm_campaign=NB-State-Of-Cloud&campaignid=10760212027&adgroupid=125385947491&utm_term=state%20of%20the%20cloud%20report&https://info.flexera.com/CM-REPORT-State-of-the-Cloud%3Flead_source%3DPPC&_bt=507839023626&_bk=state%20of%20the%20cloud%20report&_bm=b&_bn=g&_bg=125385947491&gclid=EA1aIQobChMlrl5-K87rY8AlVzJ6zCh3gEwaPEAAAYyAAEgJo6fD_BwE

Typical Enterprise Network



2021 State Of Multi-Cloud - Turbonomic

Highlights

1. Multicloud leaders aspire to leverage the right apps and services to differentiate their business, but **in 2021 they value business leverage** over the benefits of any one cloud.
2. In 2021 **leaders prioritize launching new internal services**, indicating progress on the digital transformation journey.
3. **For 61% of organizations** containerization will play a strategic role within 18 months, today it is already strategic for nearly 20%.
4. **56% are using at least one container platform today**, 79% of which are using commercial Kubernetes.
5. 56% of those on their container/cloud native journey are running containers in production, but **complexity at scale is hindering advanced use cases**.
6. **Optimization (25%) is the most important initiative** for organizations adopting public cloud in the coming year, followed by advancing a multicloud strategy (21%).
7. 30% of organizations are using **3 or more clouds** today.
8. **83% of organizations** are using at least one cloud managed service today
9. **For 62% of organizations** public cloud PaaS will play a strategic role for their business within 18 months.
10. **Complexity at 38%** is the leading barrier by far to edge computing becoming conventional.

Multi-Cloud Reasoning:

1. Want to use advance services from each provider
2. Mergers (don't want to migrate).
3. Diversification.

January of 2021 and garnered responses from 819 participants, multiple industries.

Cloud Computing Potential Risks

Below are some of the potential risks

Lose direct knowledge and control of the underlying hardware

Noisy neighbors or other VMs sharing the same hardware can affect your VM performance

Hard to diagnose performance issues, due to limited visibility and virtualization (problem isolation)

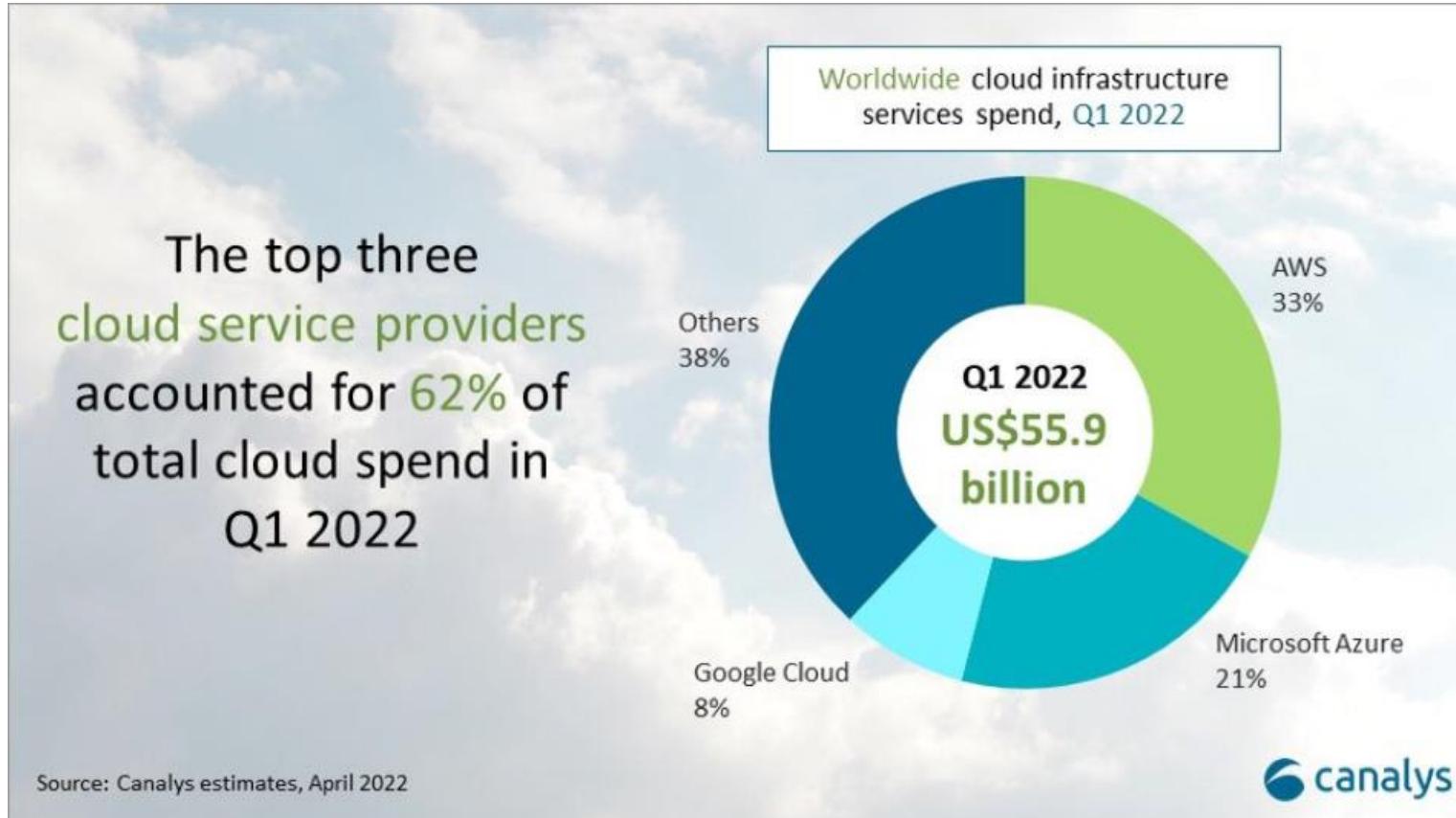
Potential security risks of placing mission critical data on remote servers

Vendor lock-in means getting stuck with a Cloud provider who has your data



Public Cloud Providers

Global Top 3



Global market share:

- AWS (33%)
- Azure (21%)
- GCP (8%)

<https://www.canalys.com/newsroom/global-cloud-services-Q1-2022>

AWS Global Infrastructure 2022

26 Launched Regions

Each with multiple Availability Zones (AZ's)

84 Availability Zones

17 Local Zones

25 Wavelength Zones

For ultralow latency applications

8 Announced Regions

32 Announced Local Zones

2x More Regions

With multiple AZ's than the next largest cloud provider

245 Countries and Territories Served

108 Direct Connect Locations

310+ Points of Presence

300+ Edge Locations and 13 Regional Edge Caches

<https://aws.amazon.com/about-aws/global-infrastructure/>

AWS Regions (2022)

AWS Global Infrastructure Map

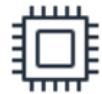
The AWS Cloud spans 84 Availability Zones within 26 geographic regions around the world, with announced plans for 24 more Availability Zones and 8 more AWS Regions in Australia, Canada, India, Israel, New Zealand, Spain, Switzerland, and United Arab Emirates (UAE).



AWS Product Groups

AWS has **175** services grouped under the followings:

<https://aws.amazon.com/products/>



Compute



Storage



Database



Networking &
Content Delivery



Analytics



Machine
Learning



Security, Identity,
& Compliance

AWS Services Offered Vary Per Region

The screenshot shows the AWS Global Infrastructure Regional Table page. The top navigation bar includes links for Contact Sales, Support, English, My Account, and Sign In to the Console. Below the navigation is a secondary navigation bar with links for Global Infrastructure (Overview, Regions and AZs, Local Zones, Regional Table, Sustainability), Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Events, Explore More, and a search bar.

The main content area features a "Region Table" section with a "PAGE CONTENT" link and a "Region Table" link. A note indicates the data was last updated on June 26, 2020. On the left, a sidebar lists various AWS services: AWS Edge Network Locations, Amazon Cloudfront China, AWS China Regions*, Asia Pacific (Osaka) Local Region**, AWS Identity and Access Management (IAM), AWS Mobile Hub, AWS Organizations, AWS Support in AWS GovCloud (US), and AWS Systems Manager.

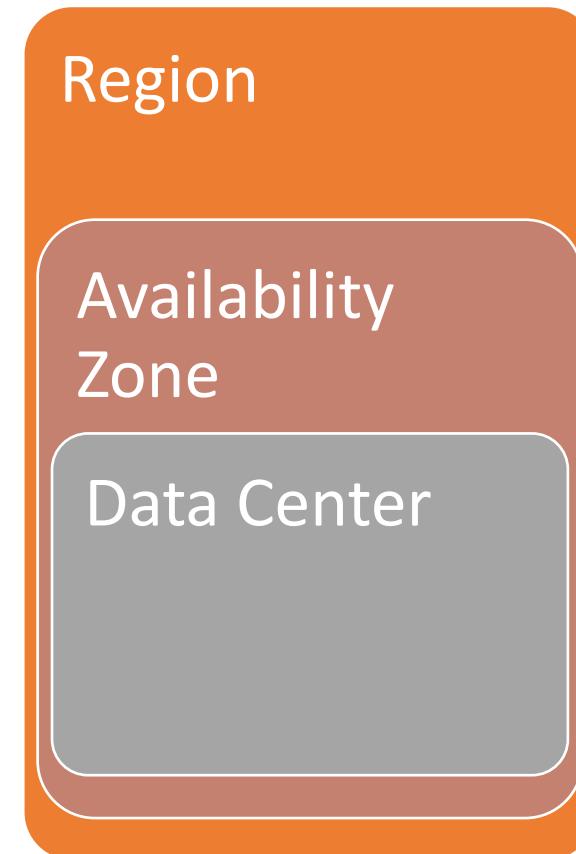
The central part of the page displays a table titled "Region Table" with columns for the Americas, Europe / Middle East / Africa, and Asia Pacific regions. The table lists various AWS services and their availability across different AWS locations. The table structure is as follows:

Services Offered:	Northern Virginia	Ohio	Oregon	Northern California	Montreal	São Paulo	AWS GovCloud (US-West)	AWS GovCloud (US-East)
Alexa for Business	✓							
Amazon API Gateway	✓	✓	✓	✓	✓	✓	✓	✓
Amazon AppStream 2.0	✓			✓			✓	
Amazon Athena	✓	✓	✓	✓	✓	✓	✓	✓
Amazon AppFlow	✓	✓	✓	✓	✓	✓		
Amazon								

<https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/>

AWS

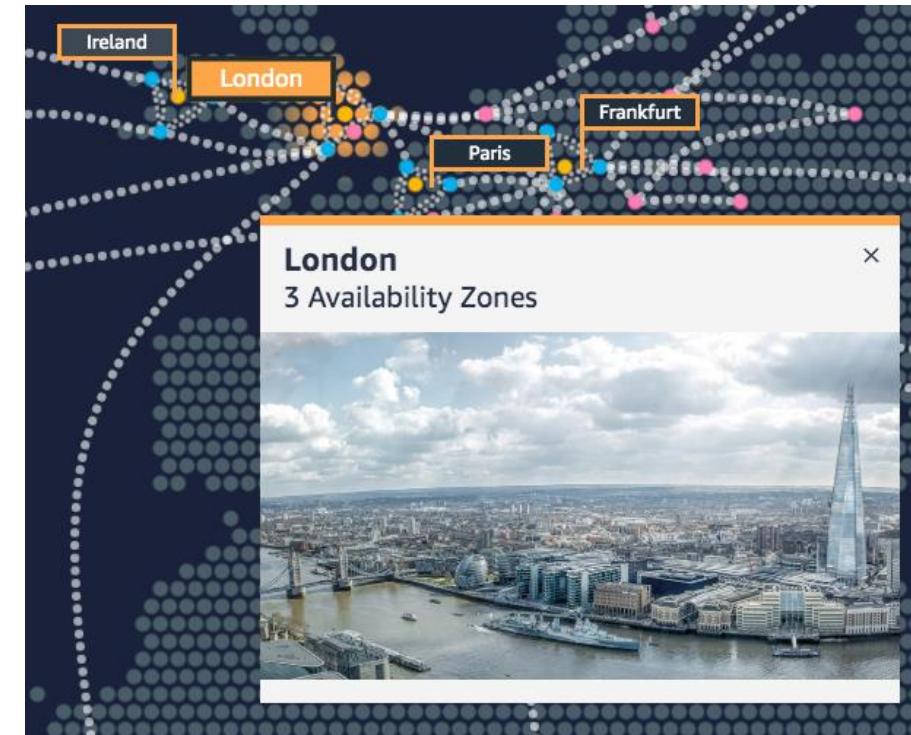
- Can choose region/multiple regions when using AWS service.
- **Region** is a physical location around the world.
- Each region contains **Availability Zones (AZ)** – **Logical Data Centers**
- AZ:
 - **Consists of 1-6 data centers**
 - A single data center can be part of only 1 AZ
 - Each AZ is located in a different floodplain
 - Power grids are designed to not impact multiple AZs in case of disaster
 - Have redundant power supplies and networking connectivity
 - AZs are engineered to be insulated from failures in other AZs
 - A data center may host thousands of servers
- AWS offers edge locations/ Points Of Presence (POPs)
 - In most major cities globally
 - To reduce latency and provide fast performance.
 - This usually where DNS and Content Delivery Services are located



Region & Number of Availability Zones	
US East	China
N. Virginia (6),	Beijing (2),
Ohio (3)	Ningxia (3)
US West	Europe
N. California (3),	Frankfurt (3),
Oregon (4)	Ireland (3),
Asia Pacific	London (3),
Mumbai (2),	Paris (3),
Seoul (2),	Stockholm (3)
Singapore (3),	South America
Sydney (3),	São Paulo (3)
Tokyo (4),	GovCloud (US)
Osaka-Local (1) ¹	US-East (3),
Canada	US-West (3)
Central (2)	

AWS Regions

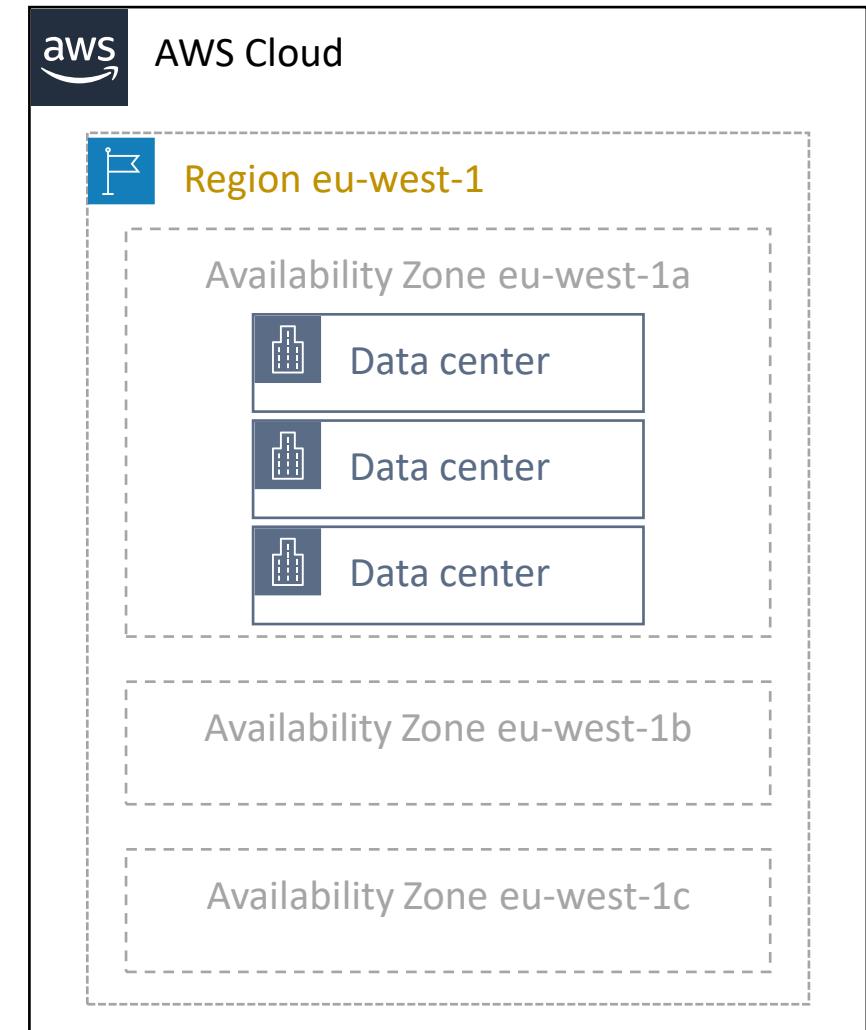
- An **AWS Region** is a geographical area
- Each AWS Region consists of **two or more Availability Zones**
- Communication between Regions uses **AWS backbone network** infrastructure
- You enable and control **data replication** across Regions



Example: London Region

AWS Availability Zones

- Each Availability Zone is –
 - Made up of **one or more** data centers
 - Designed for **fault isolation**
 - Interconnected with other Availability Zones in a Region using high-speed **private** links
- For certain services, you can choose your Availability Zones
- AWS recommends replicating across Availability Zones for resiliency



Example N. Virginia Region

Region & Number of Availability

Zones

US East

N. Virginia (6),
Ohio (3)

China

Beijing (2),
Ningxia (3)

US West

N. California (3),
Oregon (4)

Europe

Frankfurt (3),
Ireland (3),
London (3),

Asia Pacific

Mumbai (2),
Seoul (2),

Paris (3), Stockholm (3)

Singapore (3),

Sydney (3),
Tokyo (4),
Osaka-Local (1)¹

South America

São Paulo (3)

GovCloud (US)

US-East (3),
US-West (3)

Canada

Central (2)

AWS Services ▾ Search for services, features, marketplace products, and docs [Alt+S] UTSantosoAWS ▾ N. Virginia ▾ Support ▾

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances 1 Launch into Auto Scaling Group ⓘ

Purchasing option ⓘ Request Spot instances

Network ⓘ vpc-472bcd3a (default) C Create new VPC

Subnet ⓘ No preference (default subnet in any Availability Zone) C Create new subnet

Auto-assign Public IP ⓘ No preference (default subnet in any Availability Zone)

Placement group ⓘ subnet-82d101a3 | Default in us-east-1b

Capacity Reservation ⓘ subnet-ff2ca0f1 | Default in us-east-1f

Domain join directory ⓘ subnet-8c6ccb3d | Default in us-east-1d

IAM role ⓘ subnet-b50cc6d3 | Default in us-east-1a

Shutdown behavior ⓘ subnet-8f2a4fc2 | Default in us-east-1c

Stop - Hibernate behavior ⓘ subnet-460e1778 | Default in us-east-1e

Enable termination protection ⓘ Protect against accidental termination

Monitoring ⓘ Enable CloudWatch detailed monitoring Additional charges apply.

Tenancy ⓘ Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.

Elastic Inference ⓘ Add an Elastic Inference accelerator Additional charges apply.

Credit specification ⓘ Unlimited Additional charges may apply

AWS Local Zones

- AWS Local Zones are a type of infrastructure deployment that places compute, storage, database, and other select AWS services close to large population and industry centers.
- Enable you to run latency-sensitive portions of applications closer to end users and resources in a specific geography
- Are an extension of an AWS Region where you can use AWS services in geographic proximity to end users
- Let you place AWS compute, storage, database, and other select services closer to large population, industry, and IT centers where no Region exists today
- Are managed and supported by AWS
- Los Angeles (LA) AWS Local Zone is available by invitation

Local Zones

US West (Oregon) Local Zones

Parent Region	Zone Name	Location (metro area)
US West (Oregon)	us-west-2-las-1a	Las Vegas
US West (Oregon)	us-west-2-lax-1a	Los Angeles
US West (Oregon)	us-west-2-lax-1b	Los Angeles
US West (Oregon)	us-west-2-phx-1a	Phoenix

AWS Point Of Presence (PoP)

A point of presence (PoP) is a demarcation point, access point, or physical location at which two or more networks or communication devices share a connection.

A PoP can be a simple “single server” or a full deployed advanced infrastructure (Edge locations).

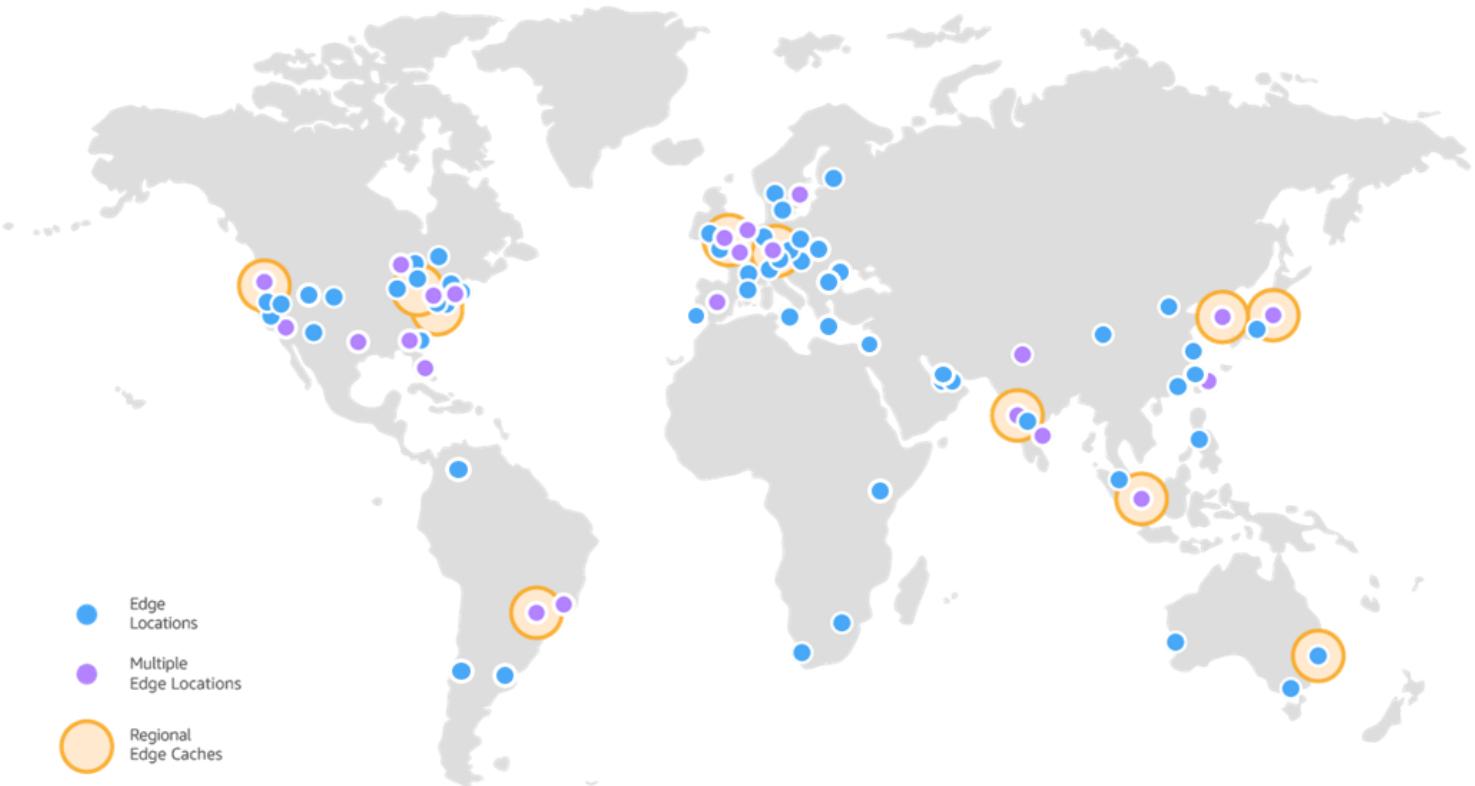
The aws PoPs are used for both **AWS CloudFront to deliver content to end users at high speeds**, and Lambda@Edge to run Lambda functions with the lowest possible latency.

CloudFront is an aws content delivery network (CDN) service.

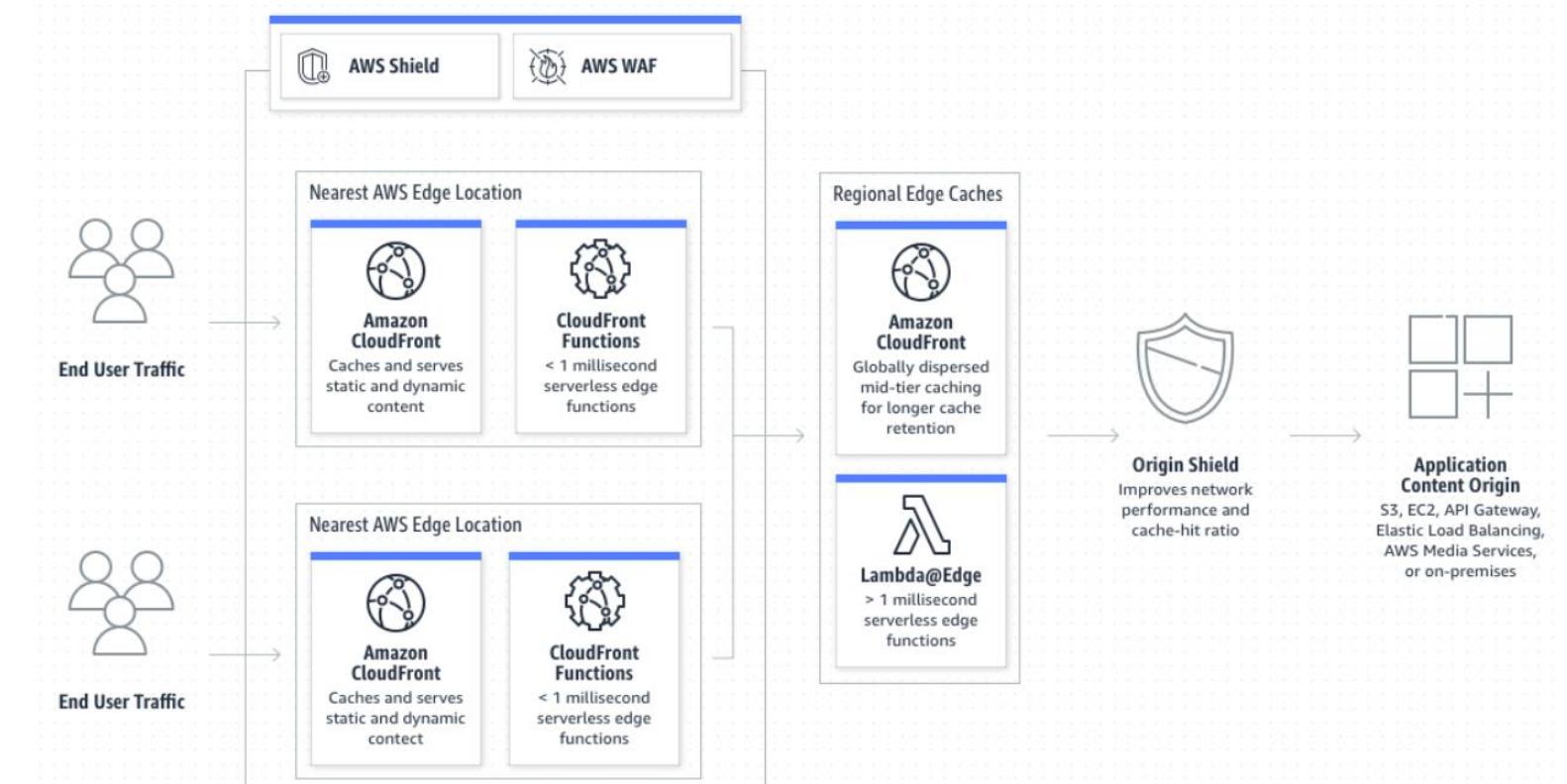
A CDN is a network of servers that distributes content from an “origin” server throughout the world by caching content close to where each end user is accessing the internet via a web-enabled device.

Edge locations and Regional edge caches improve performance by caching content closer to users.

AWS Points of Presence



AWS CloudFront for CDN



<https://aws.amazon.com/cloudfront/>

AWS Region Considerations



Compliance

Law and Regulations



Latency



Price

Price is influenced by local tax, etc.



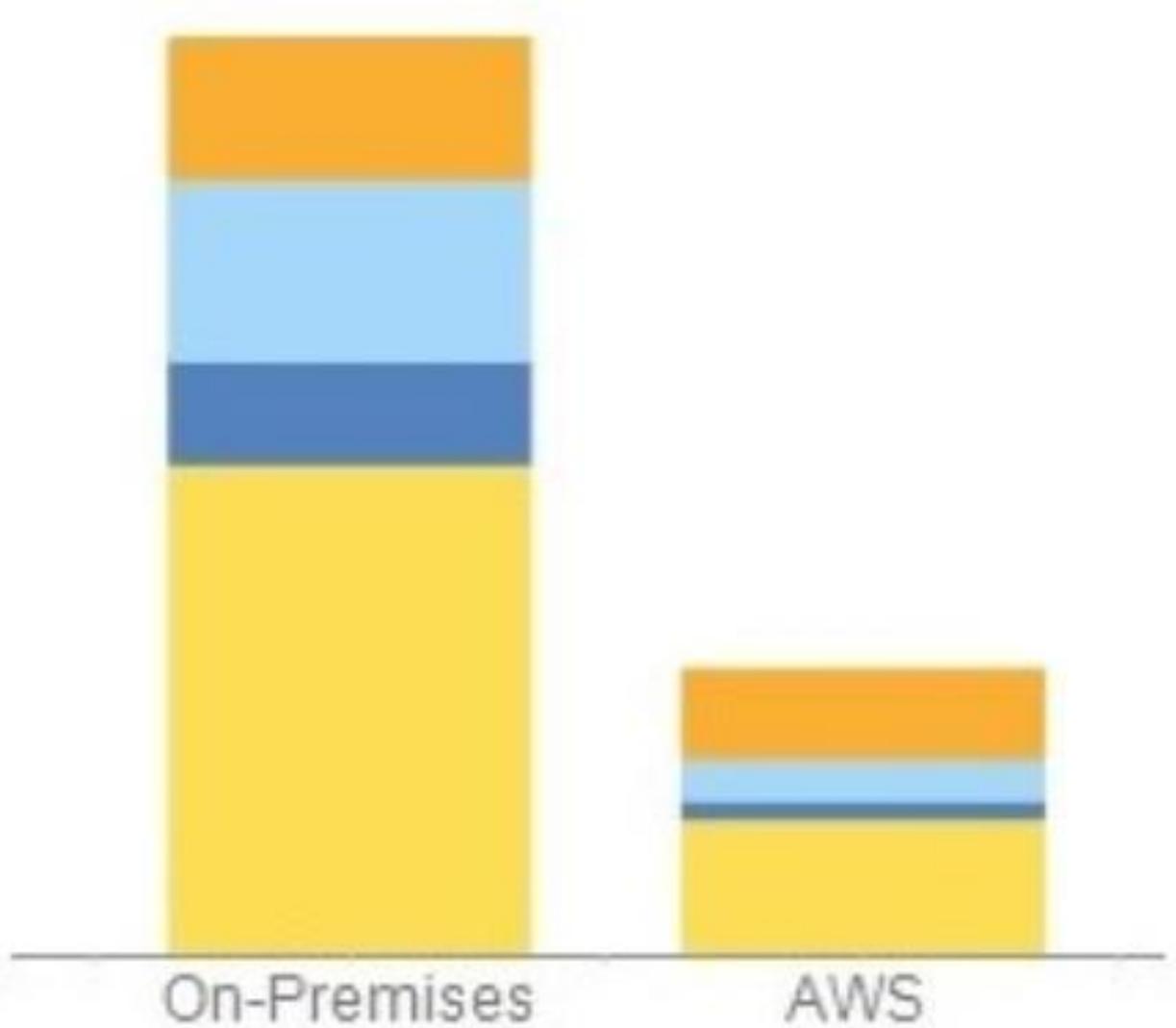
Service Availability

AWS Security and Compliance

- AWS follows shared security model:
 - AWS is responsible for security of the cloud
 - Physical security of data centers, hardware, compute, storage, virtualization, networking, etc.
 - Metaphor: House
 - Customers are responsible for the security in the cloud
 - Securing the application, data being hosted, etc.
 - Customers choose the level of security for their application
 - Metaphor: inside the rooms of the house.

Total Cost of Ownership (TCO)

- Provide savings as high as 69% per year
- AWS TCO Calculator
- <https://awstcoccalculator.com/>



AWS Cost Calculators

Calculate Your Savings

Simple Monthly Calculator



Whether you are running a single instance or dozens of individual services, you can estimate your monthly bill using your monthly bill using AWS Simple Monthly Calculator. The calculator allows you to estimate individual or multiple prices and use templates to appraise complete solutions.

[AWS Simple Monthly Calculator](#)

TCO Calculator



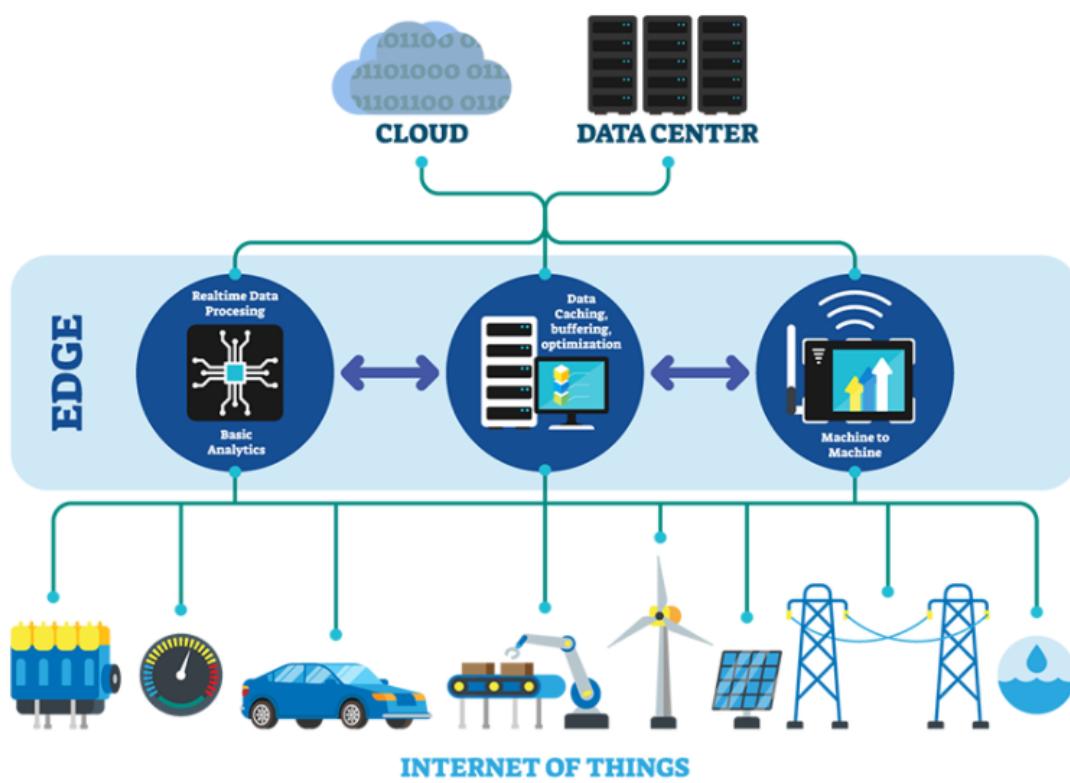
The AWS TCO calculator gives you the option to evaluate the savings from using AWS and comparing against on premises and co-location environments. The TCO calculator matches your current infrastructure to the most cost effective AWS offering. This tool takes into consideration all the costs to run a solution, including physical facilities, power and cooling, providing a realistic end-to-end comparison of your costs.

[AWS TCO Calculator](#)

https://aws.amazon.com/pricing/?pg=WIAWS-N&tile=learn_more

Cloud Computing Vs Edge Computing

EDGE Computing



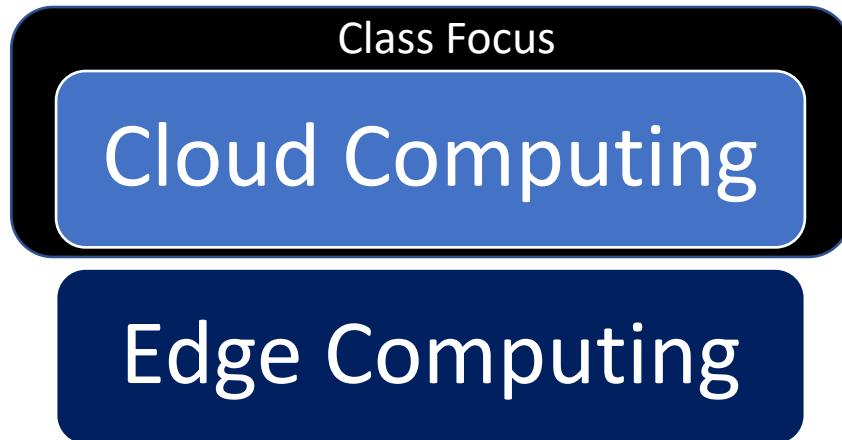
Edge computing is a distributed computing paradigm that brings computation and data storage closer to the location where it is needed to improve latency and save bandwidth (a topology not a technology).

- Edge computing can be in a device, hub, etc.

Internet Of Things -the network of physical objects—“things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.

<https://innovationatwork.ieee.org/real-life-edge-computing-use-cases/>

AWS Edge



Infrastructure

- AWS Local Zones
- AWS Outposts
- AWS Wavelength

Rugged & Disconnected Edge Devices

- AWS Snowball Edge
- AWS Snowcone

AWS Local Zones: an AWS infrastructure deployment that places AWS compute, storage, database, and other select services close to large population, industry, and IT centers.

- US locations: Los Angeles, Boston, Houston, Miami

AWS Wavelength: embedded AWS compute and storage inside telco provider's 5G network.

AWS Outpost: fully managed and configurable compute and storage racks built with AWS-designed hardware that allow customers to run compute and storage on-premises.

<https://aws.amazon.com/edge/>

Cloud Engineer Roles

Cloud Engineer Demand

What Skills Are Most in Demand in the IT Industry?

Here are 22 popular IT skills that you can seize in 2022 to unlock your potential and advance your IT career.

Infrastructure



1. Networking

Nearly everything we do starts with the network, so a well-rounded IT pro knows that it is important to understand the technology behind [computer networking](#).



2. Cloud computing

As remote work has increased, so has the demand for cloud computing. [Cloud computing](#) skills – such as configuration, deployment, [security](#), management and troubleshooting – are essential to IT pros since. This includes everything from end user support for [software as a service \(SaaS\)](#) solutions to analyzing, managing, maintaining and deploying [cloud platforms](#) like Microsoft Azure or Amazon Web Services (AWS).



<https://www.comptia.org/blog/top-it-skills-in-demand>

<https://searchcloudcomputing.techtarget.com/definition/cloud-engineer>

Cloud Engineer Roles

Cloud Architect

(design cloud system, integrate tools and services)

Cloud Software Developer

(build and maintain software applications and databases)

Cloud Security Engineers

(build and maintain security for platforms and applications)

Cloud Administrators

(develop, maintain, and troubleshoot networks of cloud platforms and computing resources)

Cloud Network Engineers

(manage and support network infrastructures)

Cloud Automation Engineers

(automation, orchestration and integration (DevOps), include IaC)



Cloud architect

- Stays up-to-date with new technologies, helps decide which to use
- Provides documentation, processes, and tooling to developers
- Gives developers freedom to innovate
- Common challenges include –
 - Resource management
 - Cost optimization
 - Defining best practices for performance, reliability, and security

Job titles: Cloud Architect, Systems Engineer, Systems Analyst

Cloud architecture is the practice of applying cloud characteristics to a solution that uses cloud services and features to meet an organization's technical needs and business use cases.

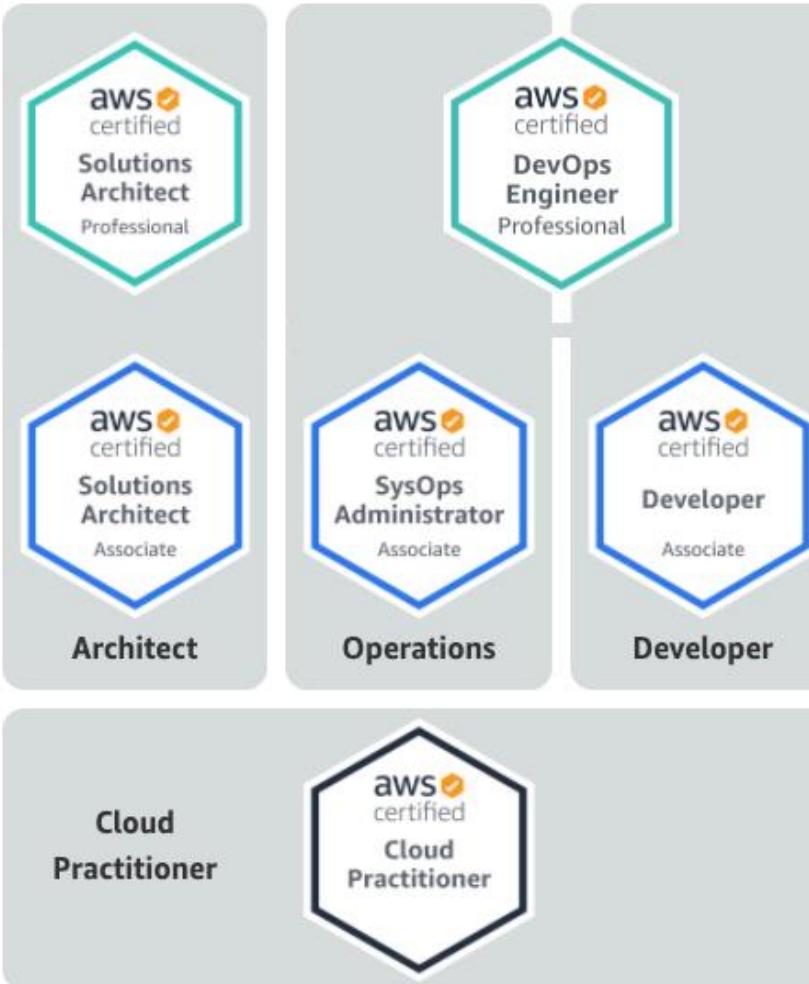
AWS Certification exams



Available AWS Certifications

Professional

Two years of comprehensive experience designing, operating, and troubleshooting solutions using the AWS Cloud



Associate

One year of experience solving problems and implementing solutions using the AWS Cloud

Foundational

Six months of fundamental AWS Cloud and industry knowledge

aws certified

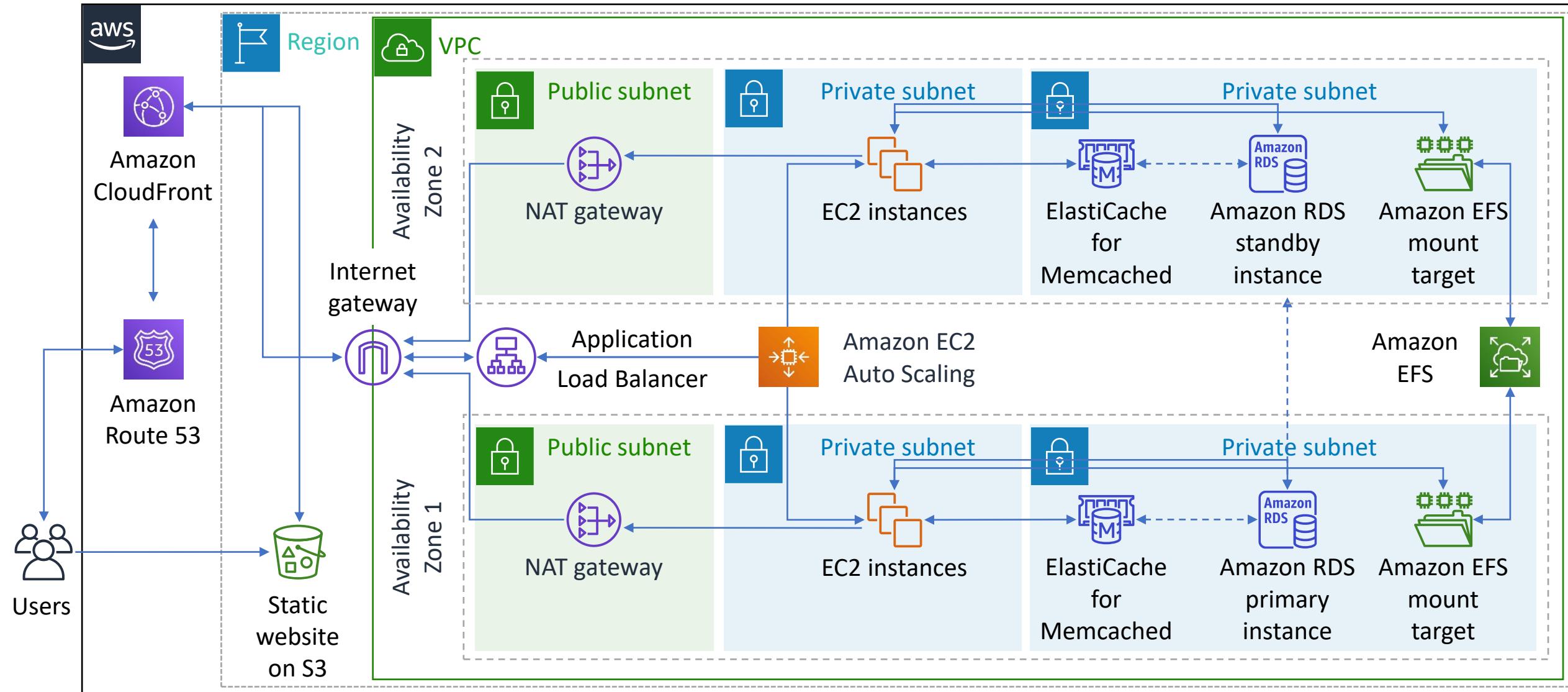
Updated May 2019

Specialty

Technical AWS Cloud experience in the Specialty domain as specified in the **exam guide**



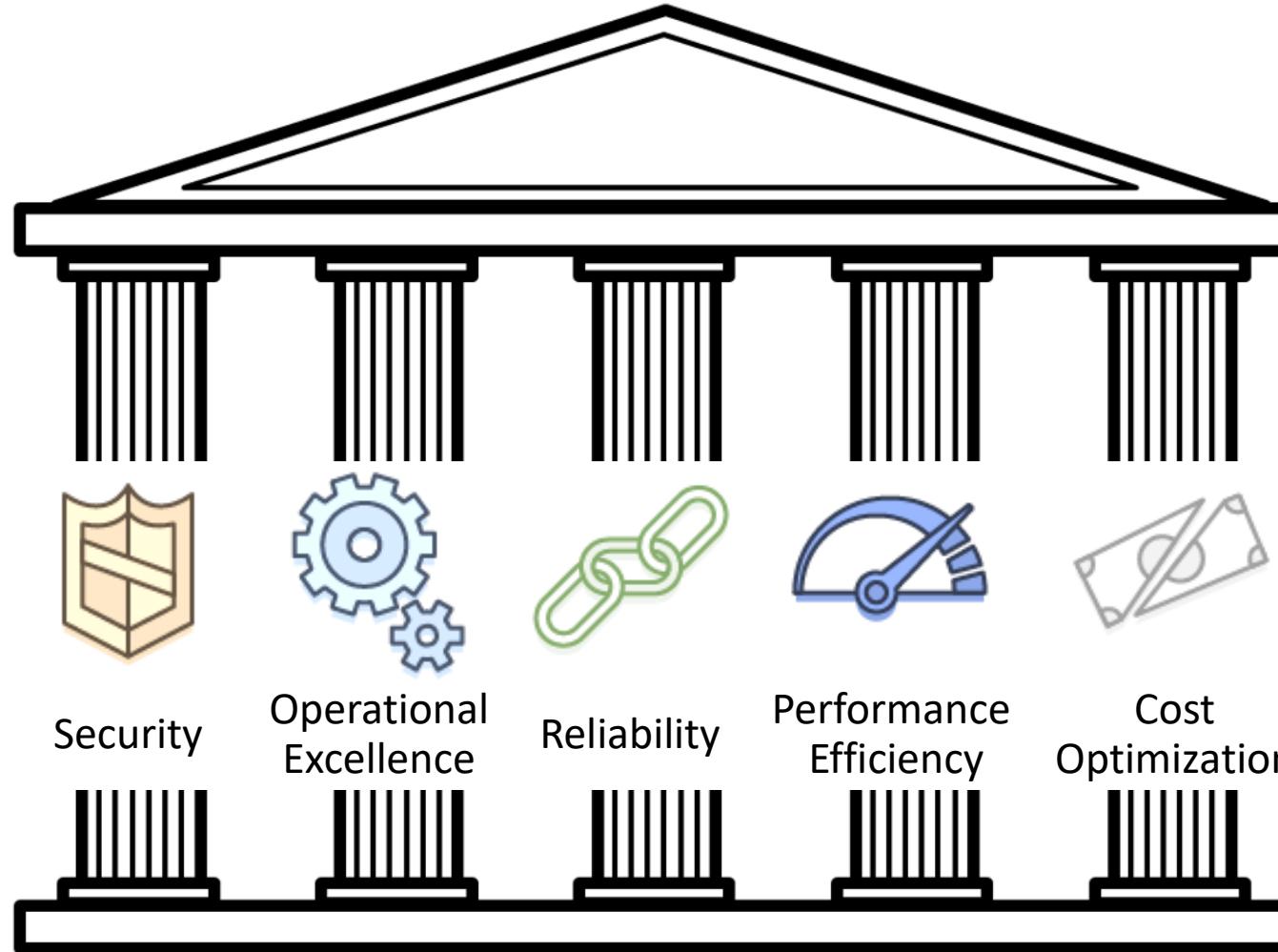
A large architecture



AWS Well-Architected Framework

A Design Guide

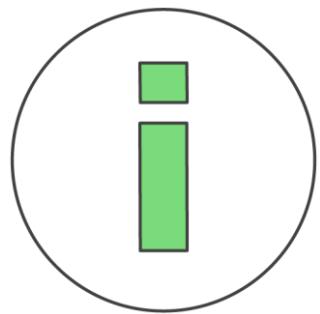
The AWS Well-Architected Framework (Design Guide)



To easier to memorize: CROPS

The [AWS Well-Architected Framework](#) is a guide to build cloud applications and workloads, based on 5 pillars.

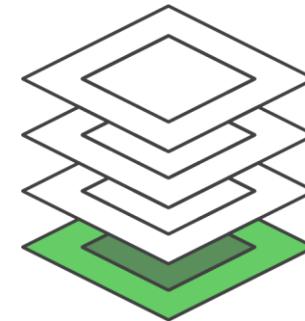
Security pillar



Identity foundation



Traceability



Security at all layers



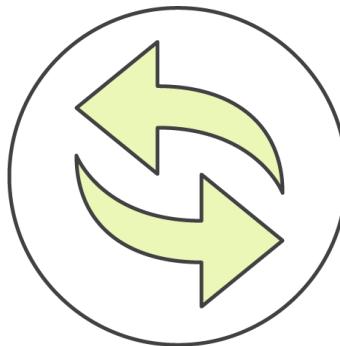
Risk assessment and
mitigation strategies

Operational Excellence pillar

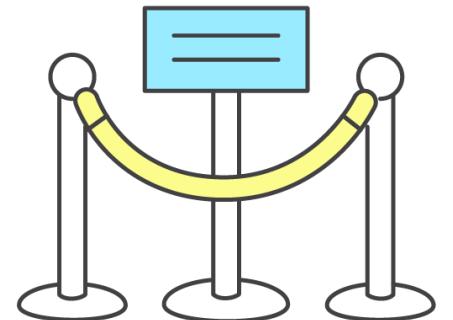
- The ability to run and monitor systems
- To continuously improve supporting process and procedures



Deployed



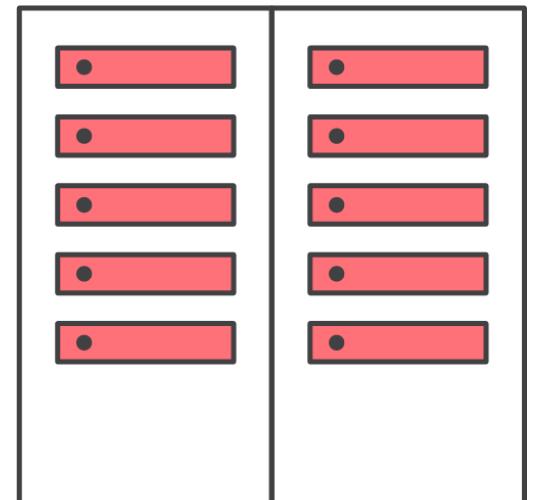
Updated



Operated

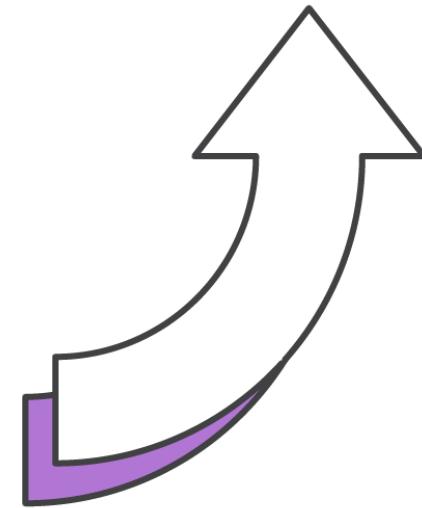
Reliability pillar

- Recover quickly from infrastructure or service disruptions
- Dynamically acquire computing resources to meet demand
- Mitigate disruptions such as:
 - Misconfigurations
 - Transient network issues



Performance Efficiency pillar

- Choose efficient resources and maintain their efficiency as demand changes
- Democratize advanced technologies
- Employ mechanical sympathy



Cost Optimization pillar

- Measure efficiency
- Eliminate unneeded expense
- Consider using managed services



The AWS Well-Architected Tool



AWS
Well-Architected
Tool

- Helps you review the state of your workloads and compares them to the latest AWS architectural best practices
- Gives you access to knowledge and best practices used by AWS architects, when you need it
- Delivers an action plan with step-by-step guidance on how to build better workloads for the cloud
- Provides a consistent process for you to review and measure your cloud architectures
- The earlier the better
- Continuous practice
- <https://www.youtube.com/watch?v=yb9CH3UbMbw>

Design tradeoffs

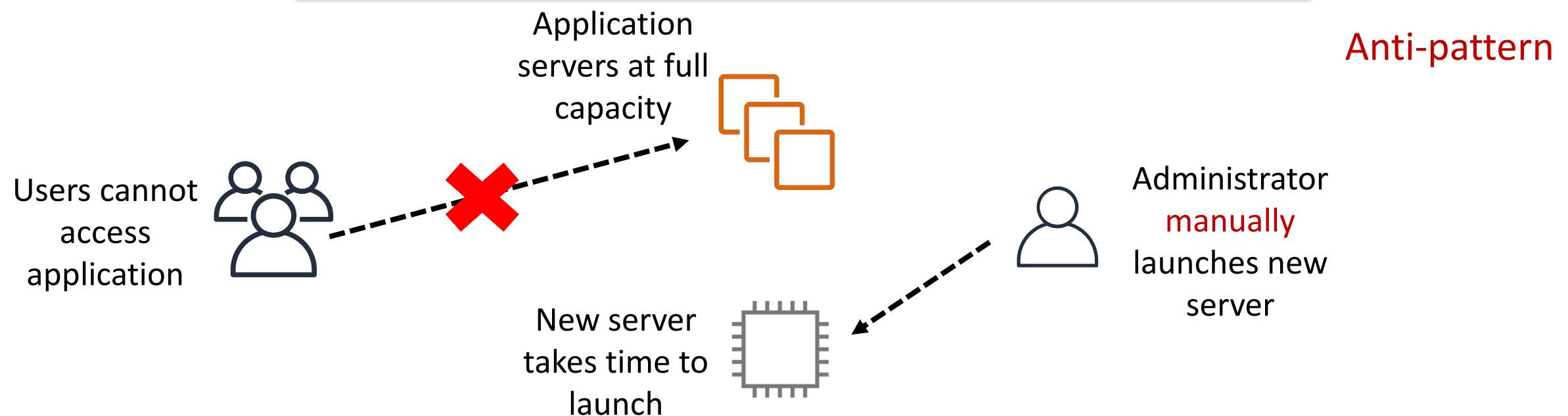
- Evaluate tradeoffs so you can select an optimal approach
- Examples of tradeoffs include:
 - Trade consistency, durability, and space for time and latency to deliver higher performance
 - Prioritize speed to market of new features over cost
- Base design decisions on empirical data

Best practices for building solutions on AWS

- As you design solutions, evaluate tradeoffs and base your decisions on empirical data
- Follow these best practices when building solutions on AWS –
 1. Enable scalability
 2. Automate your environment
 3. Treat resources as disposable
 4. Use loosely-coupled components
 5. Design services, not servers
 6. Choose the right database solution
 7. Avoid single points of failure
 8. Optimize for cost
 9. Use caching
 10. Secure your entire infrastructure

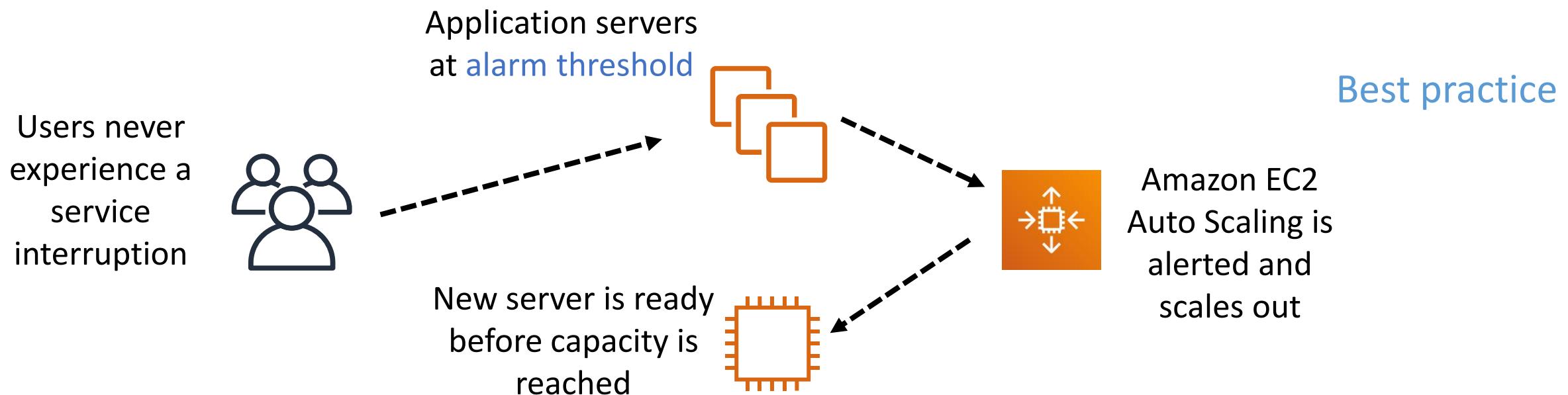
1. Enable scalability (1 of 2)

Ensure that your architecture can handle changes in demand.



1. Enable scalability (2 of 2)

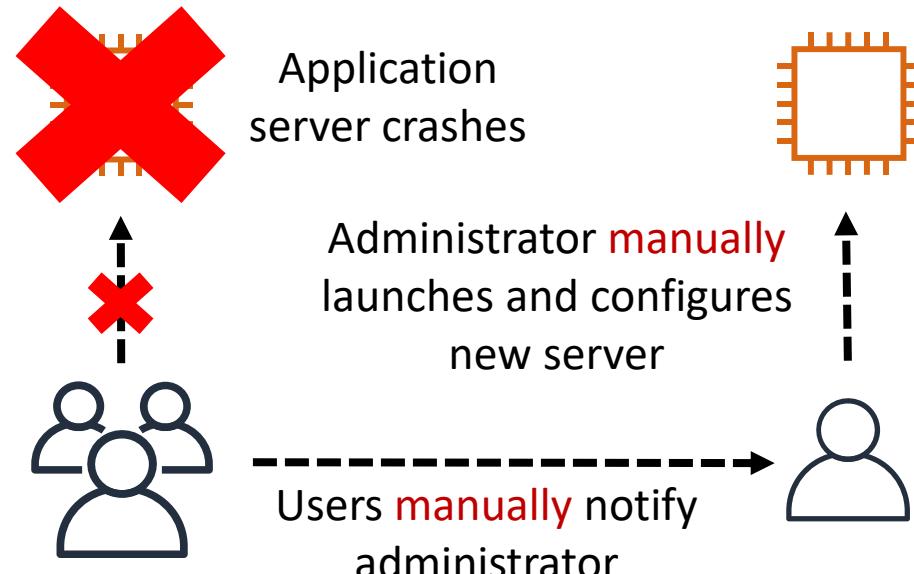
Ensure that your architecture can handle changes in demand.



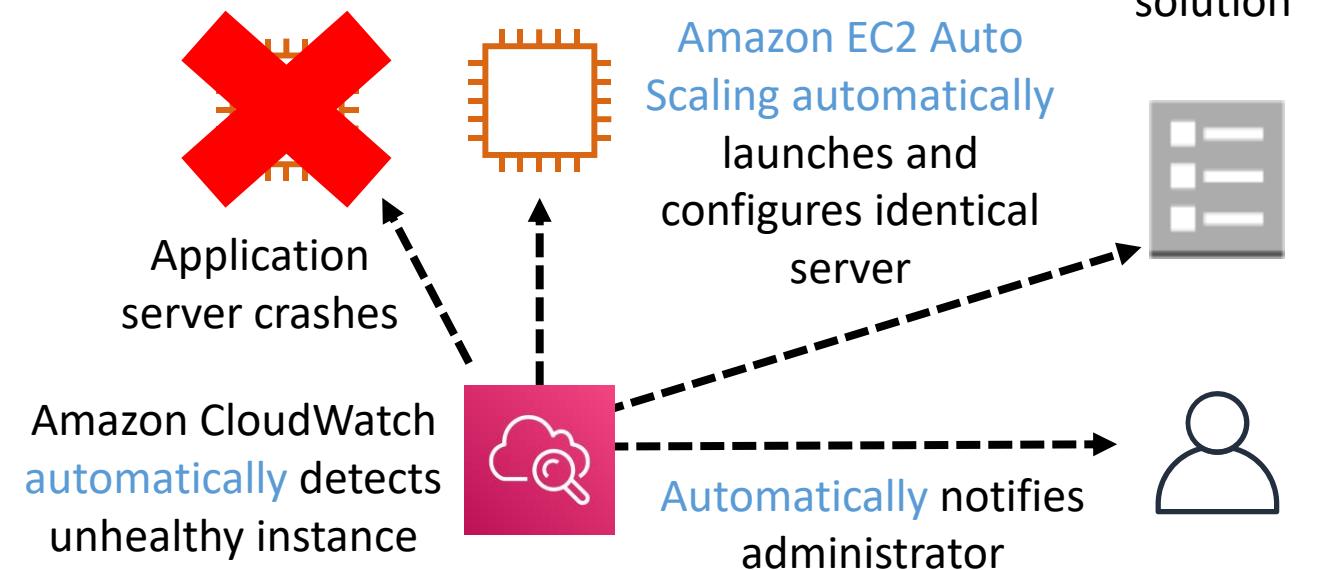
2. Automate your environment

Where possible, automate the provisioning, termination, and configuration of resources.

Anti-pattern



Best practice



3. Treat resources as disposable



Take advantage of the dynamically provisioned nature of cloud computing.

Anti-pattern

- Over time, different servers end up with different configurations
- Resources run when they're not needed
- Hardcoded IP addresses prevent flexibility
- It can be difficult or inconvenient to test new updates on hardware that's in use

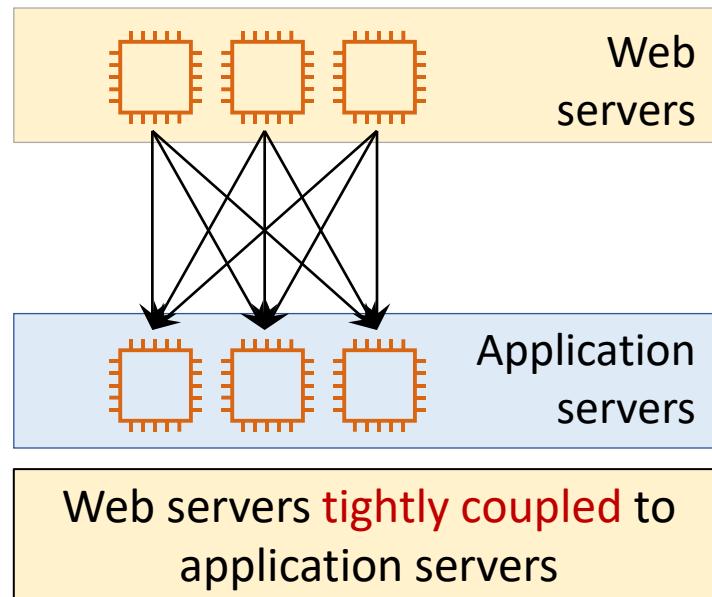
Best practice

- Automate deployment of new resources with identical configurations
- Terminate resources that are not in use
- Switch to new IP addresses automatically
- Test updates on new resources, and then replace old resources with updated ones

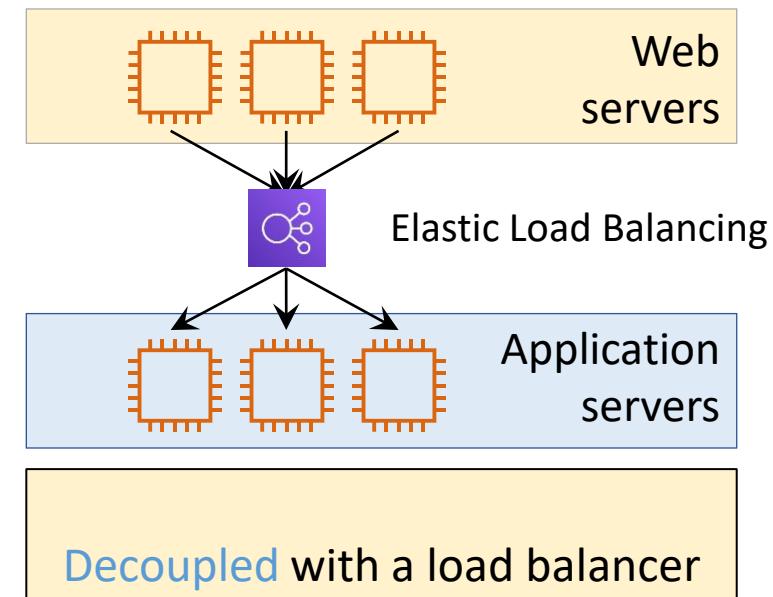
4. Use loosely coupled components

Design architectures with independent components.

Anti-pattern



Best practice



5. Design services, not servers



*Use the breadth of AWS services.
Don't limit your infrastructure to servers.*

Anti-pattern

- Simple applications run on persistent servers
- Applications communicate directly with one another
- Static web assets are stored locally on instances
- Backend servers handle user authentication and user state storage

Best practice

- When appropriate, consider using containers or a serverless solution
- Message queues handle communication between applications
- Static web assets are stored externally, such as on Amazon Simple Storage Service (Amazon S3)
- User authentication and user state storage are handled by managed AWS services

6. Choose the right database solution



Match technology to the workload, not the other way around.

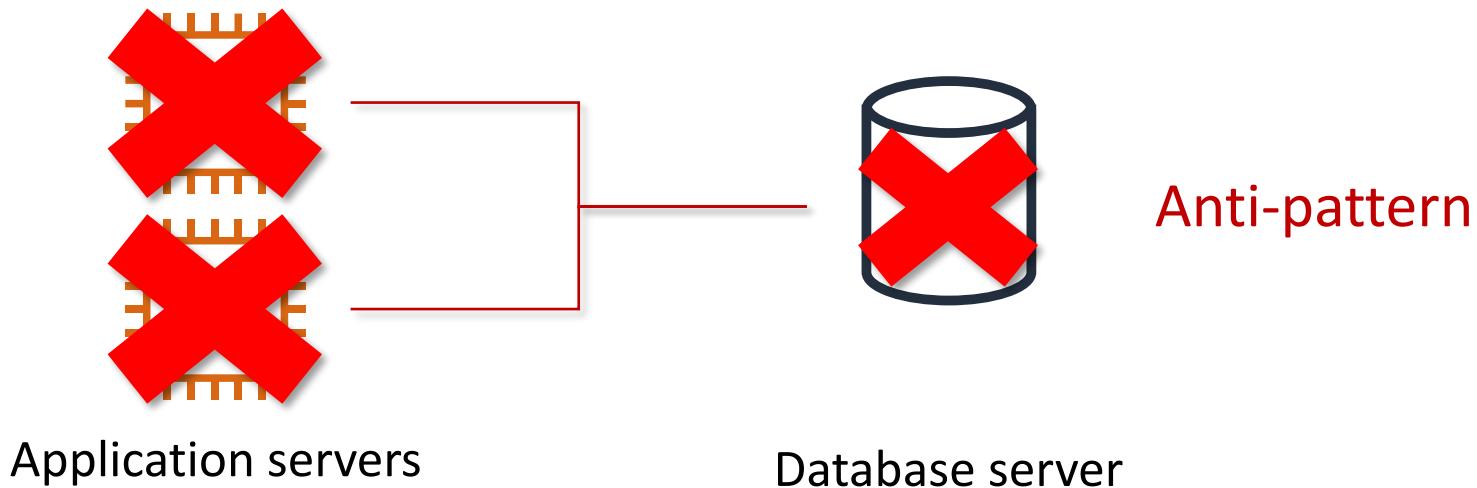
Things to consider:

- Read and write needs
- Total storage requirements
- Typical object size and nature of access to these objects
- Durability requirements
- Latency requirements
- Maximum concurrent users to support
- Nature of queries
- Required strength of integrity controls

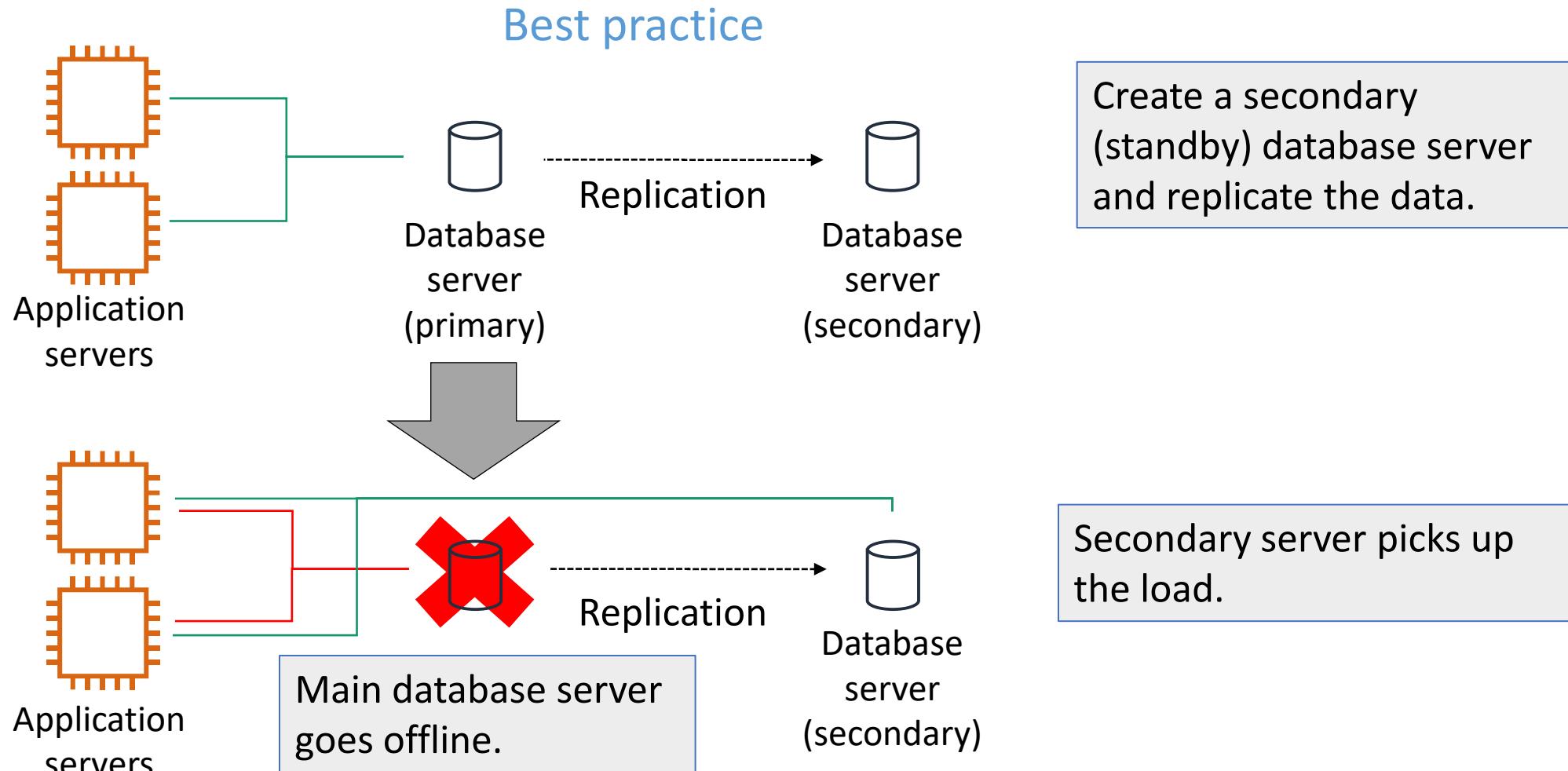
7. Avoid single points of failure (1 of 2)

*Assume everything fails.
Then, design backward.*

Where possible, use redundancy to prevent single points from bringing down an entire system.



7. Avoid single points of failure (2 of 2)



8. Optimize for cost



*Take advantage of the flexibility of AWS
to increase your cost efficiency.*

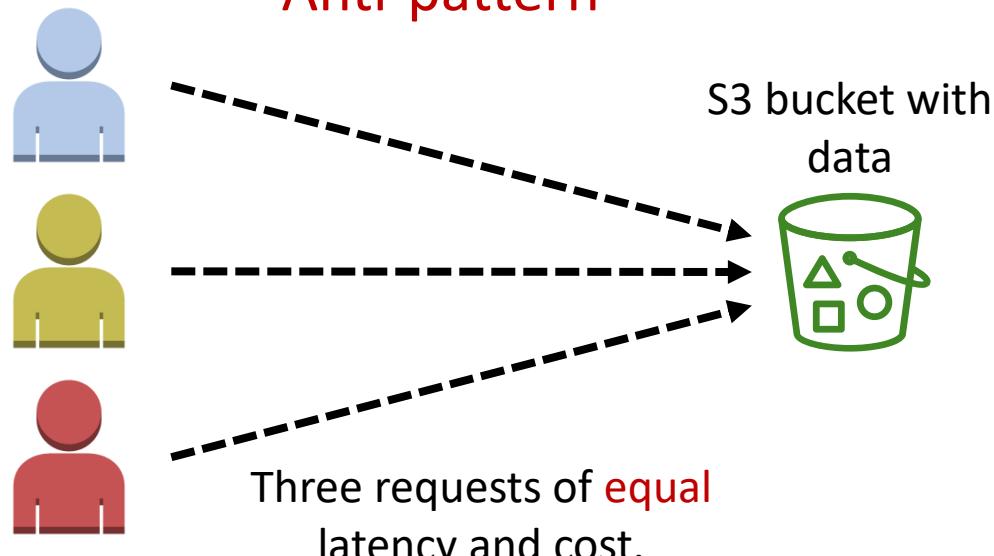
Things to consider:

- Are my resources the right size and type for the job?
- What metrics should I monitor?
- How do I make sure to turn off resources that are not in use?
- How often will I need to use this resource?
- Can I replace any of my servers with managed services?

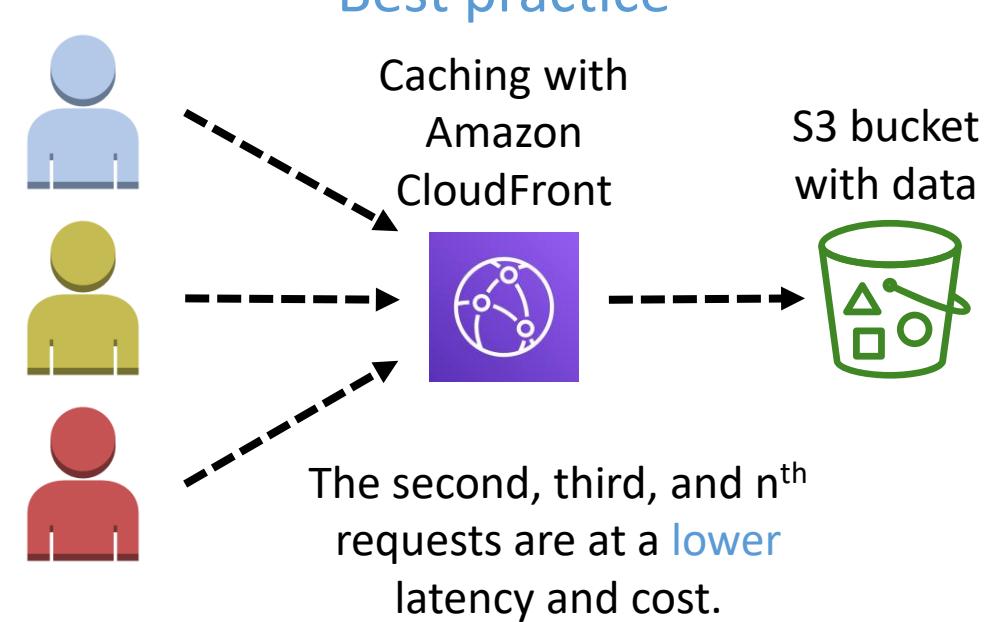
9. Use caching

Caching minimizes redundant data retrieval operations, improving performance and cost.

Anti-pattern



Best practice



10. Secure your entire infrastructure



Build security into every layer of your infrastructure.

Things to consider:

- Isolate parts of your infrastructure
- Encrypt data in transit and at rest
- Enforce access control granularly, using the principle of least privilege
- Use multi-factor authentication (MFA)
- Use managed services
- Log access of resources
- Automate your deployments to keep security consistent

Assignment

- Create a regular aws account with your UTA email address.
- They will ask your credit card.
- They have changed process. I will request credit for you and they will provide the credit directly to your account. Hope this will work.
- You will be responsible for all charges.
 - Remember, aws will charge only on the services you use.
 - Remove all resources you create after you are done with homework.
 - Check frequently. We will use “Free Tier” services as much as we can and if you do everything properly, the credit should be sufficient to cover.

End Section 1

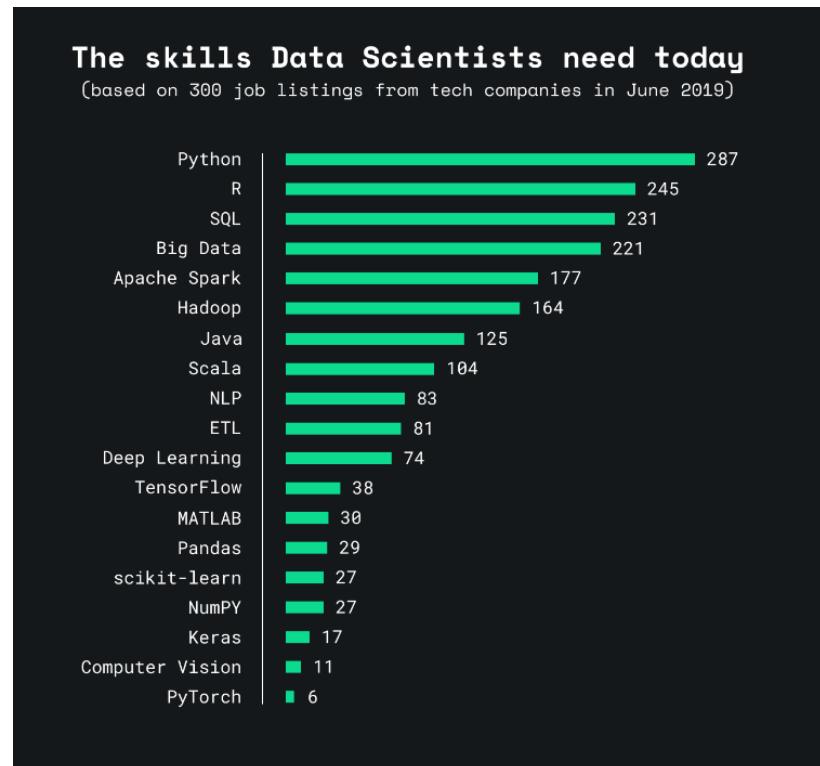
Software & Tools for this class

Python

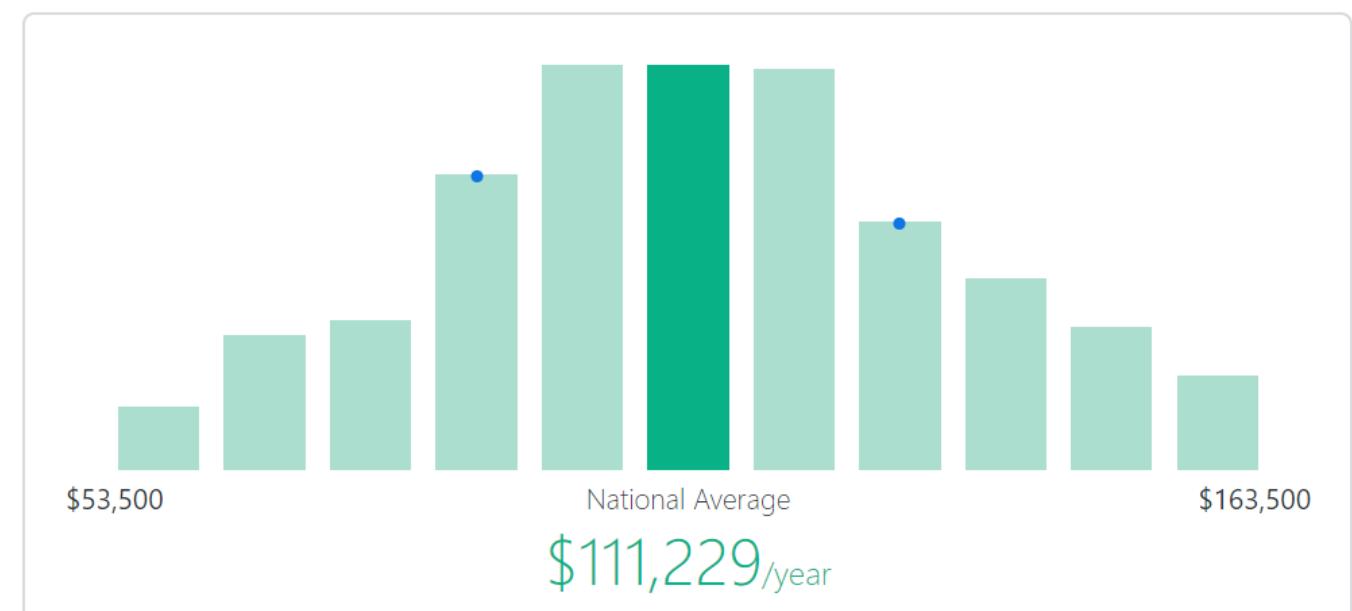
Python

- Python is a general-purpose scripting language
- It is used for web development, data analysis, artificial intelligence, and scientific computing
- The difference between programming and scripting language is that **scripting languages** do not require the compilation step and are rather interpreted. For example, normally, a C **program** needs to be compiled before running whereas normally, a **scripting language** like JavaScript need not be compiled.
- Python has become one of the most popular language
 - Open source
 - Object Oriented
 - Simple
 - Large standard library
- A module is a python.py file (contains python code)

Data Scientists Skills



Average Salary of Python Developer Jobs



<https://www.ziprecruiter.com/Salaries/Python-Developer-Salary>

<https://cvcompiler.com/blog/how-to-become-more-marketable-as-a-data-scientist/>

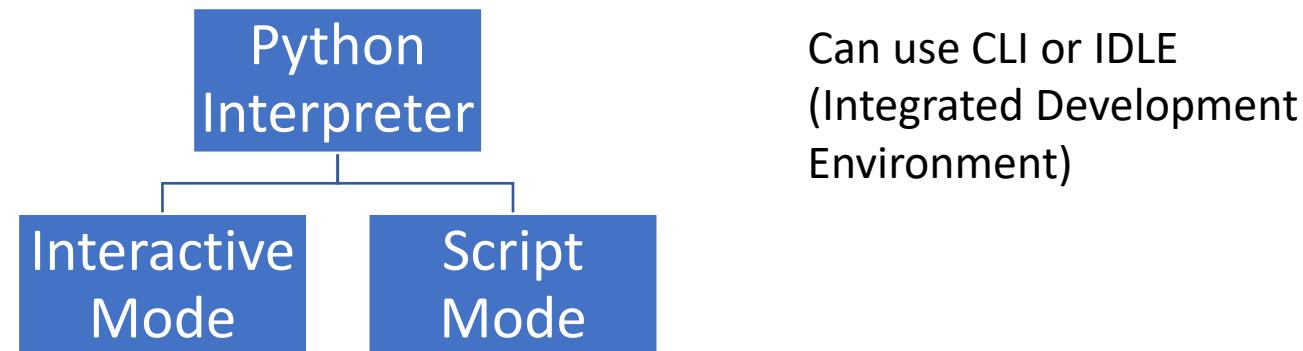
Compiler Vs. Interpreter

Compiler

- Translates a high-level language program into a separate machine language program
- The machine language program can be executed any time.

Interpreter

- Translates and executes the instructions in a high-level language



Can use CLI or IDLE
(Integrated Development Environment)

Download Python

- <https://www.python.org/>

The screenshot shows the Python.org homepage with a dark blue header featuring navigation links for Python, PSF, Docs, PyPI, Jobs, and Community. Below the header is the Python logo and a search bar with a 'GO' button. A sidebar on the left contains a code snippet demonstrating a Fibonacci series generator:

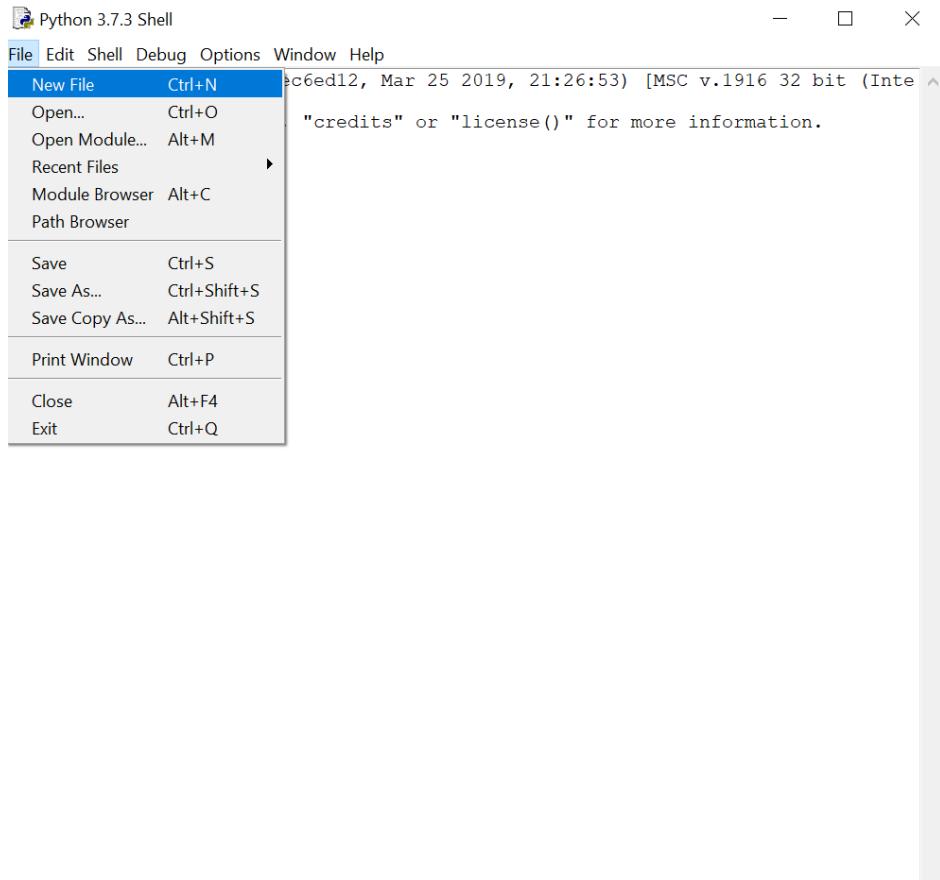
```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

A yellow callout box highlights the first line of the code. To the right, a section titled "Functions Defined" explains the core of extensible programming and provides a link to "More about defining functions in Python 3". At the bottom of the page, a footer states: "Python is a programming language that lets you work quickly".

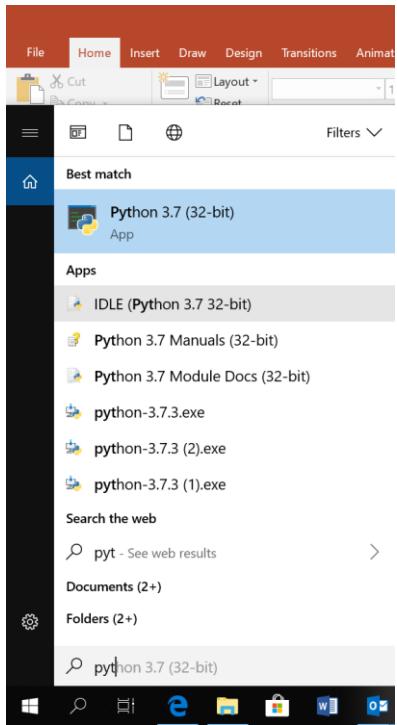
Python Cheat Sheet

<https://ehmatthes.github.io/pcc/cheatsheets/README.html>

Python Script Mode



Python Interactive Mode

A screenshot of the Python 3.7.3 Shell window. The title bar reads "Python 3.7.3 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the Python interpreter's prompt and some sample code. The text in the window is:

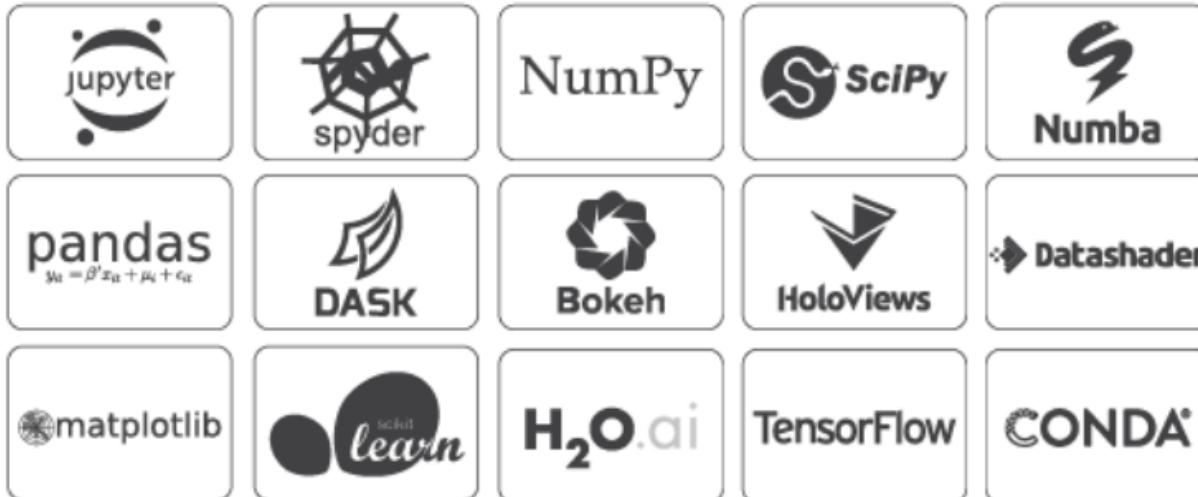
```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> # This is Python interactive mode
>>> 3*5
15
>>> print ("Hello World")
Hello World
>>> |
```

The text "This is Python interactive mode" is displayed in red, indicating the current state of the shell.

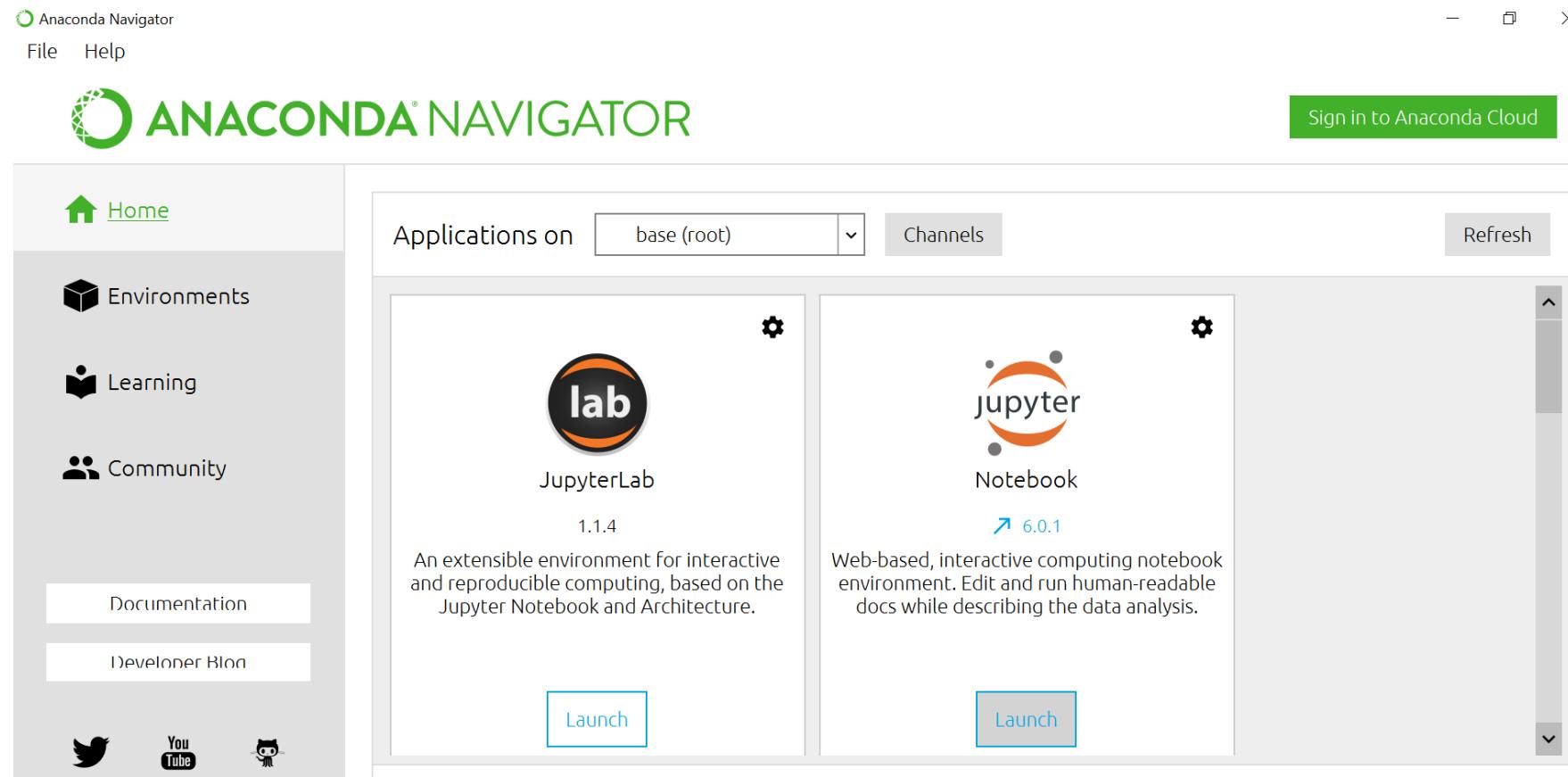
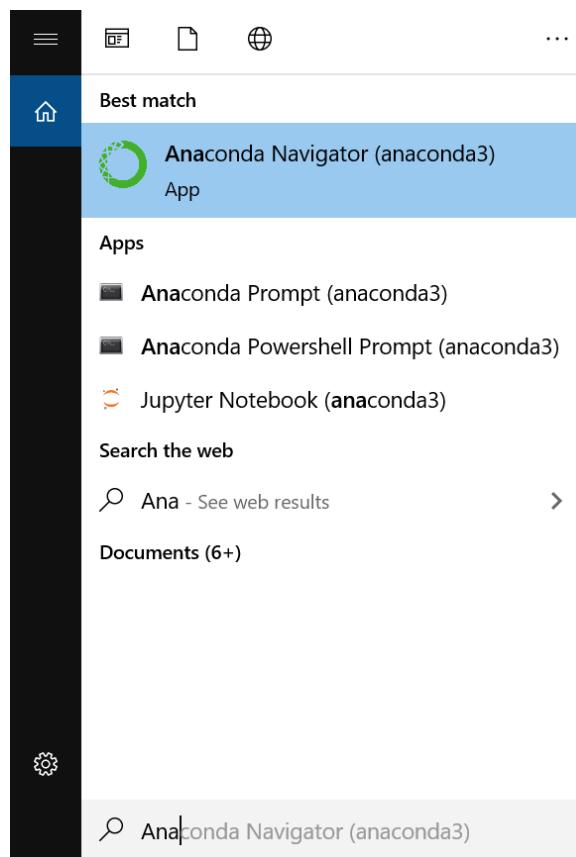
Anaconda (Recommended)

- Anaconda - **a package manager, an environment manager, a Python/R data science distribution, and a collection of over 1,500+ open-source packages.**
- Jupyter Notebook - a web-based interactive computational environment for creating Jupyter notebook documents.
- Anaconda Navigator - a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands.

<https://docs.anaconda.com/anaconda/>



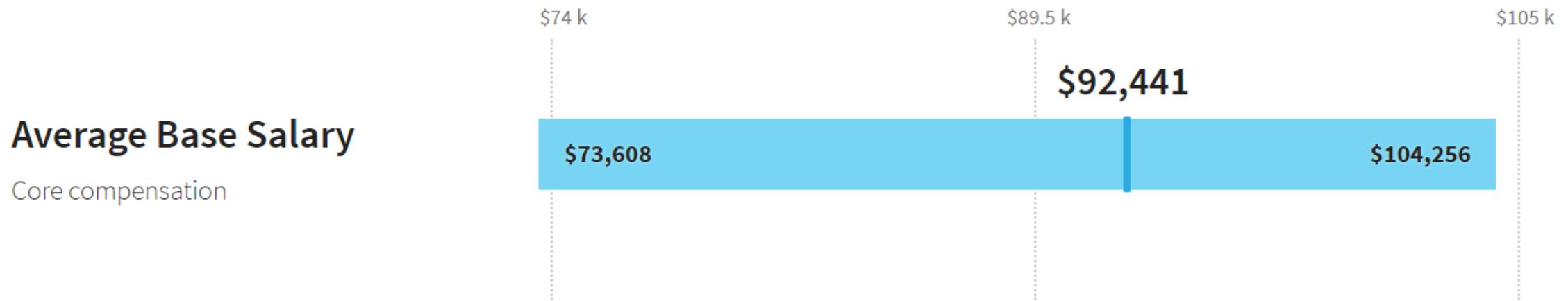
Anaconda Navigator



Software & Tools for this class

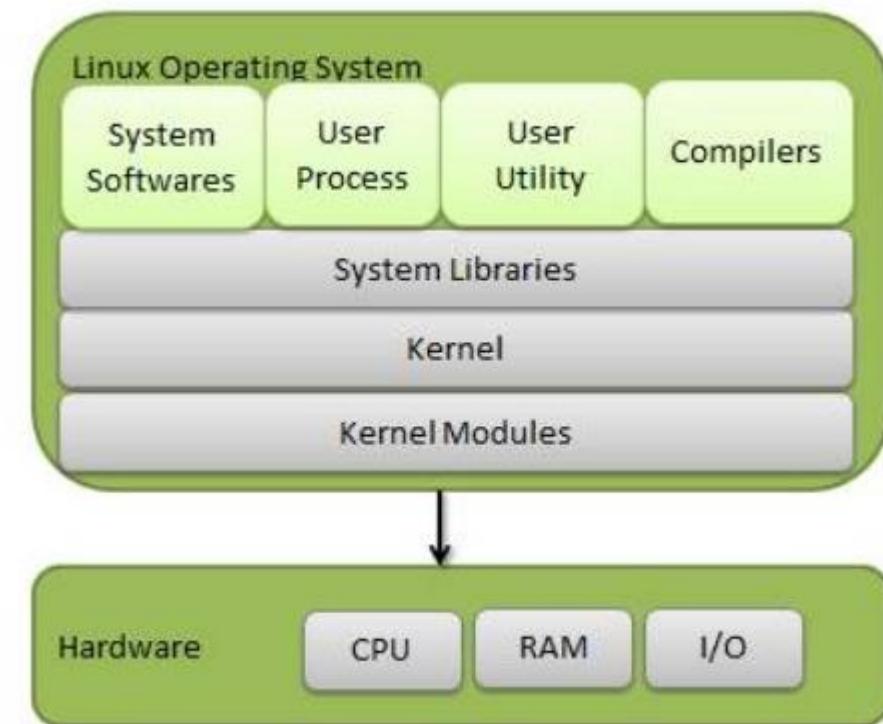
Linux

Linux Administrator - Salary



Operating System is a system software.

- Kernel is part of the operating system that interface between applications and hardware.
- The other parts of the OS sits on top of the Kernel and actually acts similar to other application processes.
 - Ex: File system

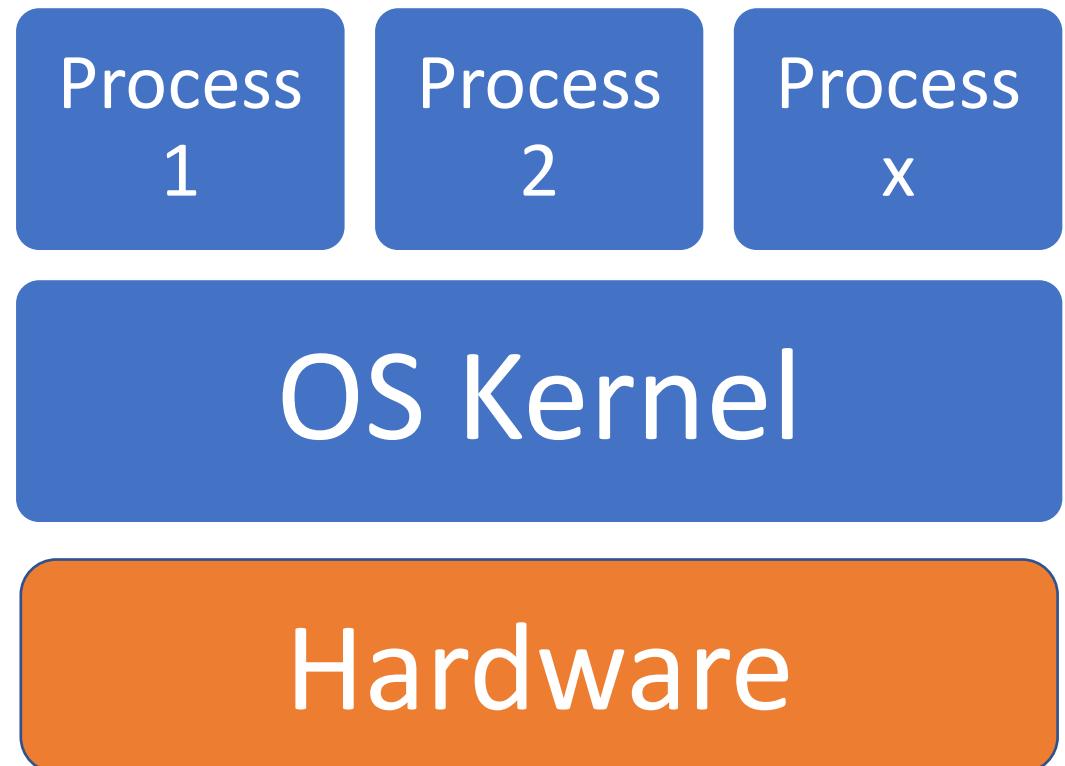


OS objectives:

- Protection and Privacy : Processes can not access each other's data
- Abstraction: hides details of hardware
- Resource management: which process has access to which part of hardware.

A program is a collection of instructions.

A process is an instance of a program that is being executed.



UNIX And LINUX

- Linux is an open-source unix-like operating system
 - Linux >< Unix
 - Unix was created by AT&T employee in 1969
 - Linux was created by Linus Torvalds in 1991
 - Layers:
 - Linux kernel (core of the operating system)
 - GNU shell utilities (including the terminal interface)
- Linux Distributions (**Distro**)
 - Produced by different organizations.
 - A Linux distribution combines software and compile it into a single operating system so the user can boot up and install.
 - Linux distribution is usually built around a package management system plus the Linux kernel.

Linux distributions have different philosophy, for example:

- Fedora **refuses** to include **closed-source** software
- Mint **includes closed-source software**

Different Linux distributions are suited for different purposes and personal preferences.

Some Linux Distributions

- Debian, **Ubuntu** (part of Debian family), Mint, Fedora, openSUSE, Mageia, CentOS
- https://en.wikipedia.org/wiki/Linux_distribution

Linux Notes

Linux is an OS (between hardware and applications)

- Open Source – source code can be read (not necessarily free)

Shell – terminal user interfaces with the OS – mainly CLI

- Bash shell (bourne again shell) - default login shell for most Linux distributions & macOS

Root – the highest level of something

- Root user can do everything

Capitalization matters (Case sensitive)

Most distro has 2 versions:

- Server (stripped down)
- Desktop (more friendly (GUI))
- Linux is good for servers (not as good for desktops (not too friendly))

Some file editors: nano, vi, vim, emacs

Shell (bash)

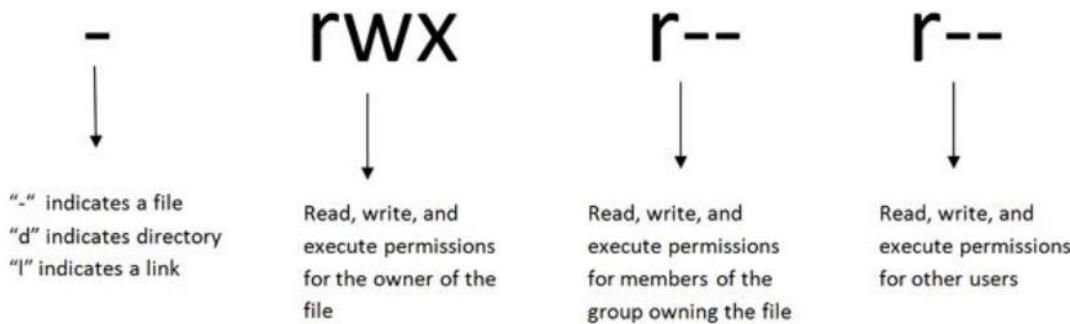
Kernel

Hardware

Linux file system

- Everything is a file; if something is not a file, it is a process
- There is no difference between a file and a directory, since
 - a directory is just a file containing names of other files.
 - Programs, services, texts, images, and so forth, are all files.
 - Input and output devices, and generally all devices, are considered to be files, according to the system
- The tree of the file system starts at the trunk or *slash*
- File permission

The permissions are broken into 4 sections.

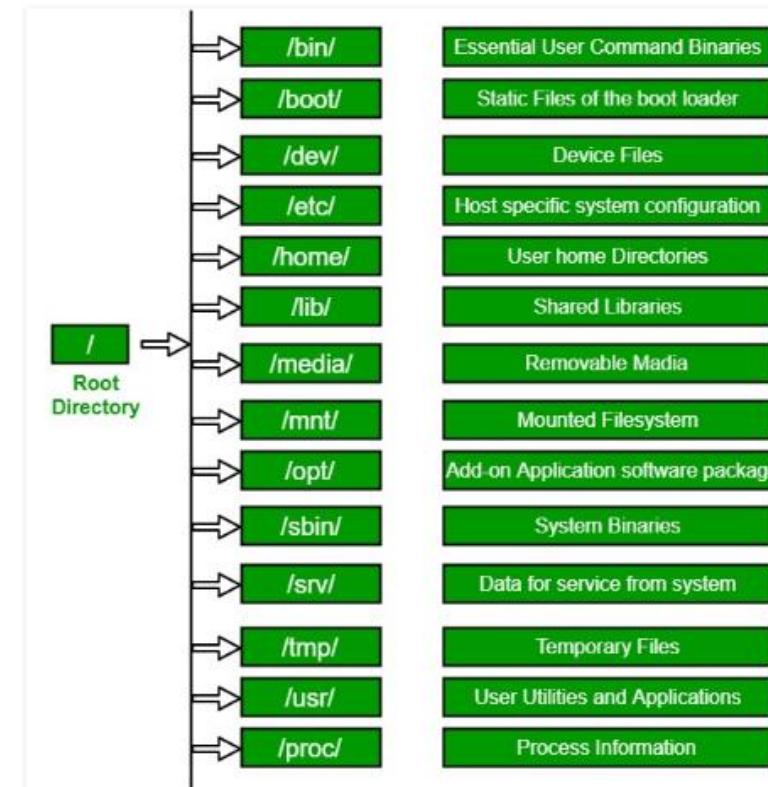


Symbol	Meaning
-	Regular file
d	Directory
l	Link
c	Special file
s	Socket
p	Named pipe
b	Block device

- https://www.tldp.org/LDP/intro-linux/html/sect_03_01.html

Linux File Hierarchy Structure (FHS)

- The Linux File Hierarchy Structure or the Filesystem Hierarchy Standard (FHS) defines the directory structure and directory contents in Unix-like operating system.
- It is maintained by the Linux Foundation.



```
santoso_budiman@cloudshell:/home (test1-244521)$ cd /
santoso_budiman@cloudshell:/ (test1-244521)$ ls
bin  boot  dev  etc  google  home  lib  lib32  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
santoso_budiman@cloudshell:/ (test1-244521)$ 
```

Example

```
santoso_budiman@cloudshell:/ (test1-244521)$ ls -l
total 88
drwxr-xr-x  1 root root 4096 Jun 19 11:19 bin
drwxr-xr-x  2 root root 4096 Mar 28 04:12 boot
drwxr-xr-x 10 root root 1340 Jun 22 16:32 dev
drwxr-xr-x  1 root root 4096 Jun 22 16:32 etc
drwxr-xr-x  1 root root 4096 Jun 22 16:32 google
drwxr-xr-x  4 root root 4096 Jun 22 16:32 home
drwxr-xr-x  1 root root 4096 Jun 19 11:13 lib
drwxr-xr-x  2 root root 4096 Jun 19 11:11 lib32
drwxr-xr-x  2 root root 4096 Jun  9 19:00 lib64
drwxr-xr-x  2 root root 4096 Jun  9 19:00 media
drwxr-xr-x  2 root root 4096 Jun  9 19:00 mnt
drwxr-xr-x  1 root root 4096 Jun 22 16:32 opt
dr-xr-xr-x 136 root root   0 Jun 22 16:32 proc
drwxr-xr-x  2 root root 4096 Jun 22 16:32 root
drwxr-xr-x  1 root root 4096 Jun 22 16:32 run
drwxr-xr-x  1 root root 4096 Jun 19 11:16 sbin
drwxr-xr-x  2 root root 4096 Jun  9 19:00 srv
dr-xr-xr-x  12 root root   0 Jun 22 16:32 sys
drwxrwxrwt  1 root root 4096 Jun 22 16:46 tmp
drwxr-xr-x  1 root root 4096 Jun 19 11:09 usr
drwxr-xr-x  1 root root 4096 Jun  9 19:00 var
santoso_budiman@cloudshell:/ (test1-244521)$ 
```

Linux Command Syntax

command [-argument] [--long-argument] file

- Case sensitive
- --help,
 - ex: pwd –help

■ Some basic commands

- pwd – print current working directory
- ls – list given directory
- cat – print the content of a text file
- cd – change directory
- clear – clear screen
- history – historical list of commands
- cp – copy from and to
- mv – move/rename file/folder
- rm – remove/delete
- chmod – change permission

Some Linux commands

FILE SYSTEM	SYSTEM	PROCESS MANAGEMENT	PERMISSIONS	NETWORKING
<code>ls</code> — list items in current directory	<code>shutdown</code> — shut down machine	<code>ps</code> — display your currently active processes	<code>ls -l</code> — list items in current directory and show permissions	<code>wget file</code> — download a file
<code>ls -l</code> — list items in current directory and show in long format to see permissions, size, and modification date	<code>reboot</code> — restart machine	<code>top</code> — display all running processes	<code>chmod ugo file</code> — change permissions of file to ugo - u is the user's permissions, g is the group's permissions, and o is everyone else's permissions. The values of u, g, and o can be any number between 0 and 7.	<code>curl file</code> — download a file
<code>ls -a</code> — list all items in current directory, including hidden files	<code>date</code> — show the current date and time	<code>kill pid</code> — kill process id pid	<code>7</code> — full permissions	<code>scp user@host:file dir</code> — secure copy a file from remote server to the dir directory on your machine
<code>ls -F</code> — list all items in current directory and show directories with a slash and executables with a star	<code>whoami</code> — who you are logged in as	<code>kill -9 pid</code> — force kill process id pid	<code>6</code> — read and write only	<code>scp file user@host:dir</code> — secure copy a file from your machine to the dir directory on a remote server
<code>ls dir</code> — list all items in directory dir	<code>finger user</code> — display information about user		<code>5</code> — read and execute only	<code>scp -r user@host:dir dir</code> — secure copy the directory dir from remote server to the directory dir on your machine
<code>cd dir</code> — change directory to dir	<code>man command</code> — show the manual for command		<code>4</code> — read only	<code>ssh user@host</code> — connect to host as user
<code>cd ..</code> — go up one directory	<code>df</code> — show disk usage		<code>3</code> — write and execute only	<code>ssh -p port user@host</code> — connect to host on port as user
<code>cd /</code> — go to the root directory	<code>du</code> — show directory space usage		<code>2</code> — write only	<code>ssh-copy-id user@host</code> — add your key to host for user to enable a keyed or passwordless login
<code>cd ~</code> — go to your home directory	<code>free</code> — show memory and swap usage		<code>1</code> — execute only	<code>ping host</code> — ping host and output results
<code>cd -</code> — go to the last directory you were just in	<code>whereis app</code> — show possible locations of app		<code>0</code> — no permissions	<code>whois domain</code> — get information for domain
<code>pwd</code> — show present working directory	<code>which app</code> — show which app will be run by default		<code>chmod 600 file</code> — you can read and write - good for files	<code>dig domain</code> — get DNS information for domain
<code>mkdir dir</code> — make directory dir			<code>chmod 700 file</code> — you can read, write, and execute - good for scripts	<code>dig -x host</code> — reverse lookup host
<code>rm file</code> — remove file			<code>chmod 644 file</code> — you can read and write, and everyone else can only read - good for web pages	<code>lsof -i tcp:1337</code> — list all processes running on port 1337
<code>rm -r dir</code> — remove directory dir recursively			<code>chmod 755 file</code> — you can read, write, and execute, and everyone else can read and execute - good for programs that you want to share	
<code>cp file1 file2</code> — copy file1 to file2				
<code>cp -r dir1 dir2</code> — copy directory dir1 to dir2 recursively				
<code>mv file1 file2</code> — move (rename) file1 to file2				
<code>ln -s file link</code> — create symbolic link to file				
<code>touch file</code> — create or update file				
<code>cat file</code> — output the contents of file				
<code>less file</code> — view file with page navigation				
<code>head file</code> — output the first 10 lines of file				
<code>tail file</code> — output the last 10 lines of file				
<code>tail -f file</code> — output the contents of file as it grows, starting with the last 10 lines				
<code>vim file</code> — edit file				
<code>alias name 'command'</code> — create an alias for a command				
COMPRESSION		SEARCHING		SHORTCUTS
<code>tar cf file.tar files</code> — create a tar named file.tar containing files		<code>grep pattern files</code> — search for pattern in files		<code>ctrl+a</code> — move cursor to beginning of line
<code>tar xf file.tar</code> — extract the files from file.tar		<code>grep -r pattern dir</code> — search recursively for pattern in dir		<code>ctrl+f</code> — move cursor to end of line
<code>tar czf file.tar.gz files</code> — create a tar with Gzip compression		<code>grep -rn pattern dir</code> — search recursively for pattern in dir and show the line number found		<code>alt+f</code> — move cursor forward 1 word
<code>tar xzf file.tar.gz</code> — extract a tar using Gzip		<code>grep -r pattern dir --include='*.ext'</code> — search recursively for pattern in dir and only search in files with .ext extension		<code>alt+b</code> — move cursor backward 1 word
<code>gzip file</code> — compresses file and renames it to file.gz		<code>command grep pattern</code> — search for pattern in the output of command		
<code>gzip -d file.gz</code> — decompresses file.gz back to file		<code>find file</code> — find all instances of file in real system		
		<code>locate file</code> — find all instances of file using indexed database built from the updatedb command. Much faster than find		
		<code>sed -i 's/day/night/g' file</code> — find all occurrences of day in a file and replace them with night - s means substitute and g means global - sed also supports regular expressions		

<http://cheatsheetworld.com/programming/unix-linux-cheat-sheet/>

Linux-sudo command

- To run administrative applications (changing system configuration):
 - Super user (su command) – super user can do anything
 - Substitute user do/ super user do (sudo command) – running as if an administrator
 - Normal users do not have permission to avoid unwanted/intended change of the system – deleting system files by accident, etc.
- sudo
 - allows a user to run a program/ root commands as another user (mostly as root user)
 - Security purposes
 - A standard user does not have permission to certain commands, such as installing an application. sudo command to be put in front of the command will allow it.
 - There is a way to define who can use the sudo command.
 - Different Linux distributions may differ

Unix Command	Acronym translation	Description
touch [OPTION] <FILE>	-	Touches a file. If the file doesn't exist already it will create a file with the specified name. If it already exists it will update the date of the file
mv [OPTION] <FILE> <destination directory or another filename>	Move	Moves a file to a specified directory. It can also be used to rename files.
rm [OPTION] <FILE>	Remove	Removes specified file in current directory. This command can also be used to remove non-empty directories.
cp [OPTION] <FILE> <destination directory or another file>	Copy	Works a lot like mv, but moves a copy of the file instead. Can also be used copy the content of one file to another file.
cat [OPTION] <FILE>	Concatenate	Concatenates files and displays it in standard output. If used on one file, the content of that file is displayed in the command line interface.
head [OPTION] <FILE>	-	Outputs the first part of a file
tail [OPTION] <FILE>	-	Outputs the last part of a file
less <FILE>	-	Shows a screenfull of the file. This is a useful command for viewing big files as it loads at smalls segments at a time. q --> quit , space --> scroll forward one page , b --> scroll backward one page. Arrow keys can be used to scroll up and down one line at a time.
wc [OPTION] <FILE>	Word count	Counts the lines and words in the file/files, but can also count other things based on the options you give it.
paste [OPTION] <FILE>	-	Merges lines from different files.
cut [OPTION] <FILE>	-	Removes different parts of a file depending on what is specified in the option.
echo [OPTION] <STRING>	-	Outputs the string to your command line interface. In computer language, a string is just a sequence of characters.
wget [OPTION] <URL>	web get	A non-interactive network downloader used to download files located at the URL.
curl [OPTION] <URL>	client URL	Similarly to wget , it is used to download files at the specified URL. This is an alternative MAC OS users, where wget doesn't work.
tee [OPTION] <FILE>	It's named after the 'T-splitter' used in plumbing.	Splits output so that it can be outputted to both the terminal and a file.

Piping and Redirecting in Linux

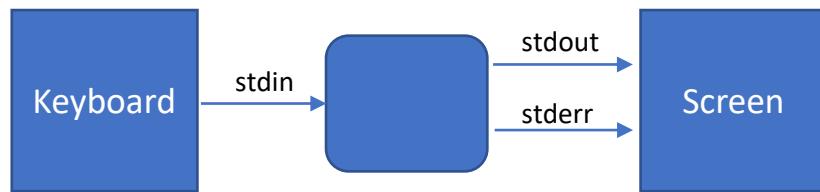
Piping

- A pipe is a form of redirection (transfer of standard output to some other destination)
- The Unix/Linux systems allow stdout of a command to be connected to stdin of another command.
- Using the pipe character '|'.
 - Pipe is used to **combine two or more commands**
 - command_1 | command_2 | command_3 | | command_N
 - Output of command_1 is input to command_2, etc.
 - Ex: \$ cat sample2.txt | head -7 | tail -5
- <https://www.geeksforgeeks.org/piping-in-unix-or-linux/>

Redirecting

- **Redirection** is a feature in **Linux** such that when executing a command, you can change the standard input/output devices.
- Redirection from stdout:
 - > redirects the output of a command to a file, replacing the existing contents of the file.
 - >> redirects the output of a command to a file, appending the output to the existing contents of the file.
- <https://www.howtogeek.com/299219/how-to-save-the-output-of-a-command-to-a-file-in-bash-aka-the-linux-and-macos-terminal/>

Stdin, stdout, stderr, redirection



- stdin, stdout, and stderr are three data streams
 - Standard input (stdin) –keyboard
 - Standard output and error (stdout, stderr) - screen
- Handled as files:
 - 0: stdin
 - 1: stdout
 - 2: stderr
- Stdout and stderr can be piped / redirected
- <https://www.howtogeek.com/435903/what-are-stdin-stdout-and-stderr-on-linux/>