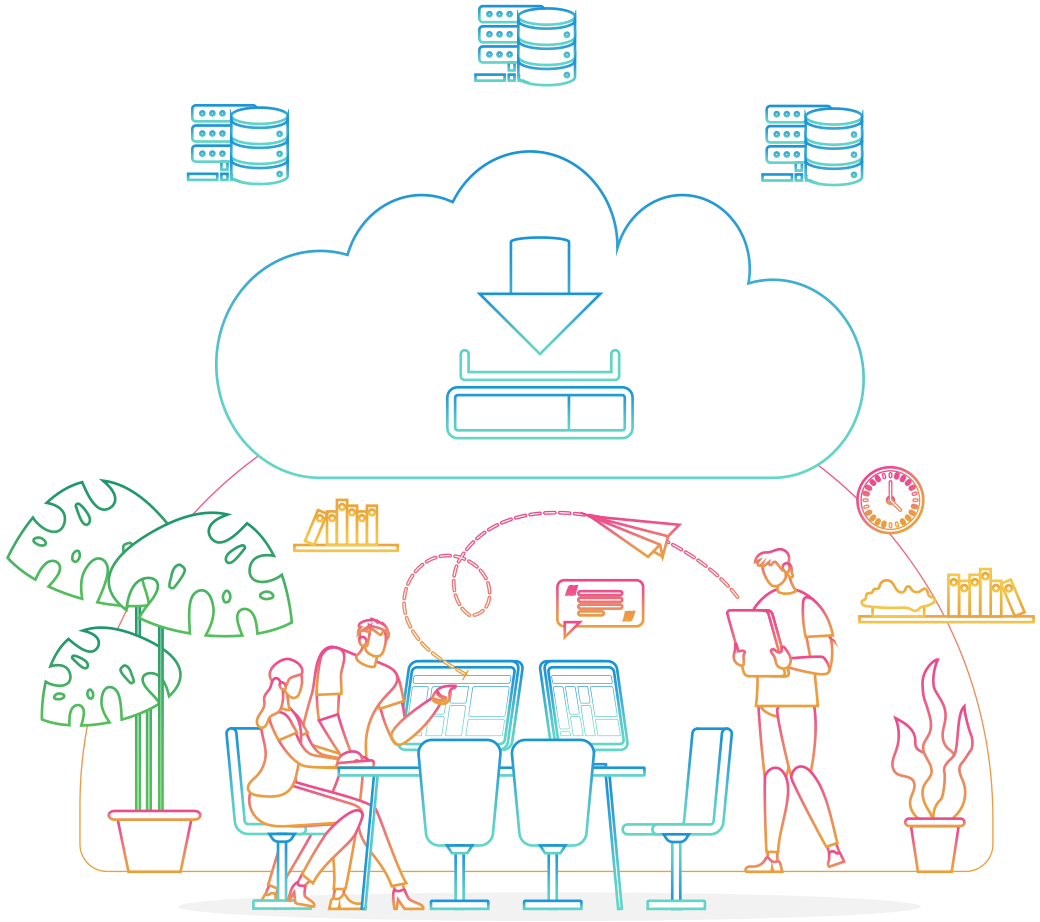
KodeKloud

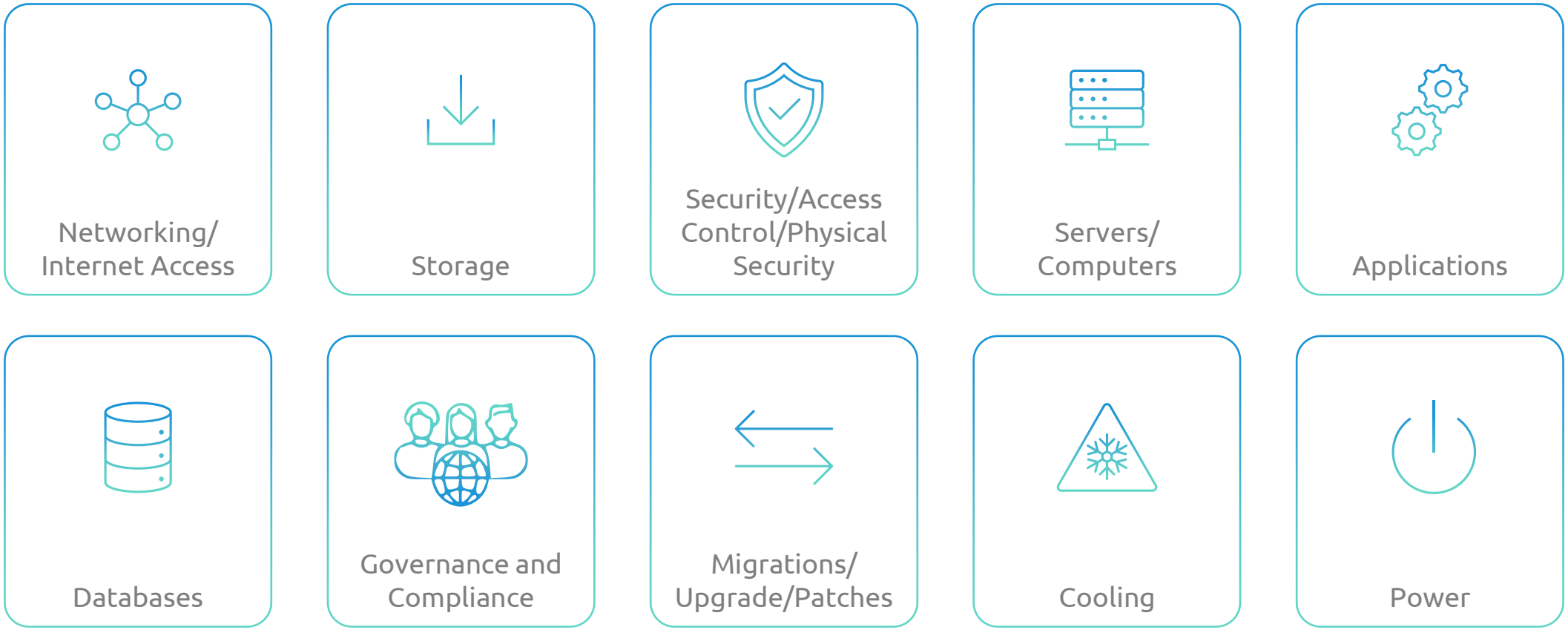
What Is Cloud Computing?

What Are Traditional IT Services Like?

- Imagine producing an idea with new tech!
- You request three servers
- This sparks an order for:
  - Hardware, Power, Cooling, Space
  - Operational and security installation
  - Access granted to the requestor
  - Software installation by the requestor

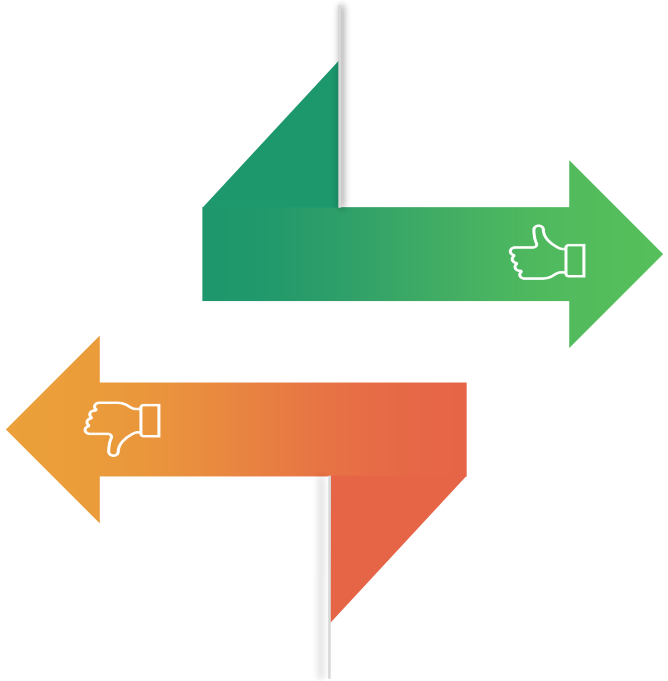


What Are Traditional IT Services Like?



Traditional IT – Pros and Cons

- Increased costs/Poor ROI
- Limited scalability
- Responsible for Everything
- Limited Locations
- More Personnel
- Long Provisioning Times



- Improved security
- Greater customization
- More control

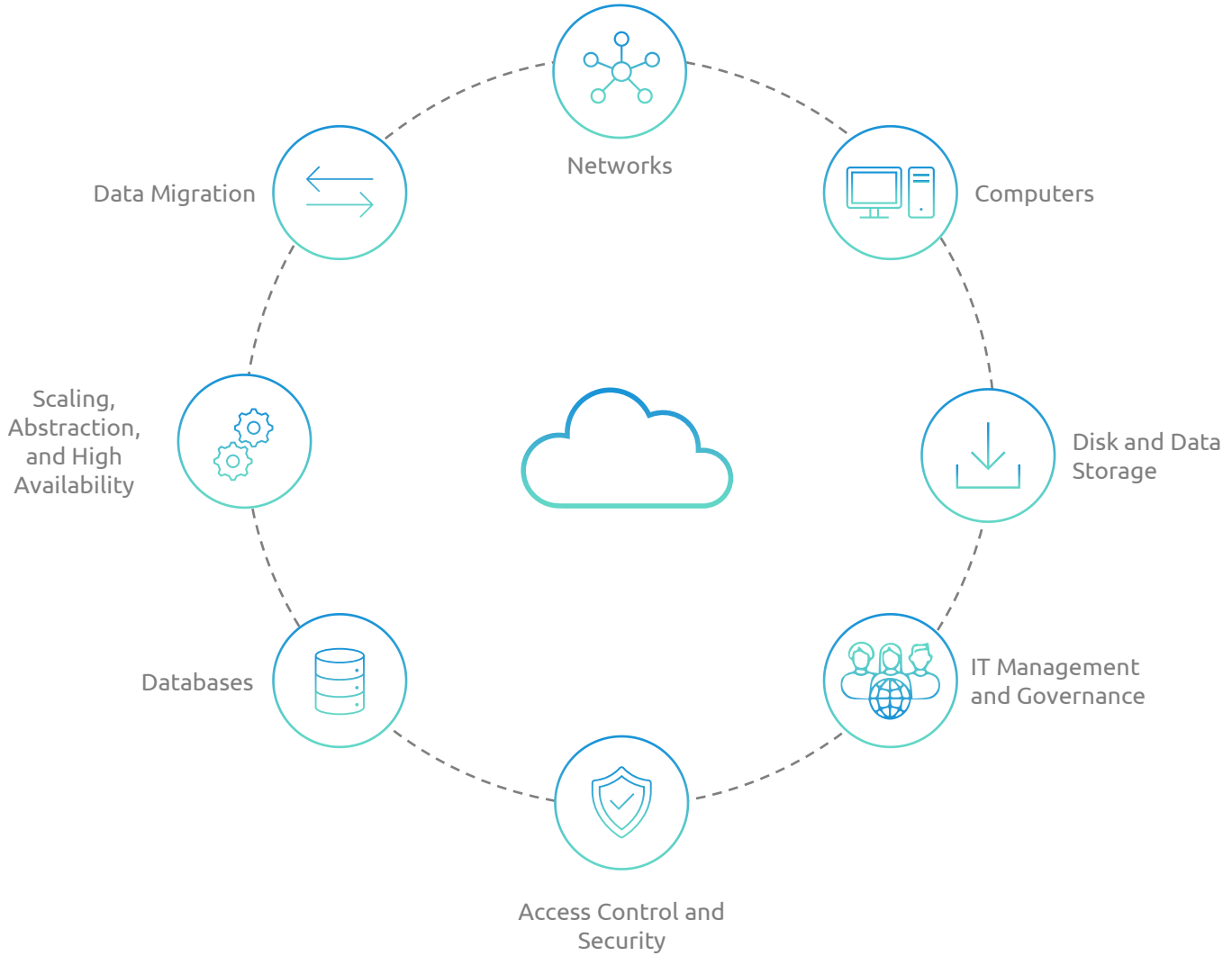
What Is Cloud Computing?

Cloud Computing is the on-demand delivery of IT resources, particularly compute power, application hosting, database application, networking, and more

Website

Command Line

Programming Language



What Is Cloud Computing?

Client

Works on a Client-Server model of computing

Client Request

Website/CLI/API Call

Request for a database

Response

Response for a database

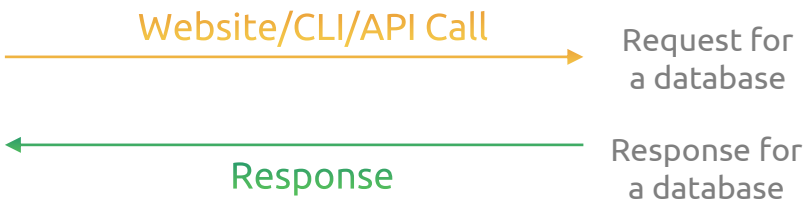
Server Response

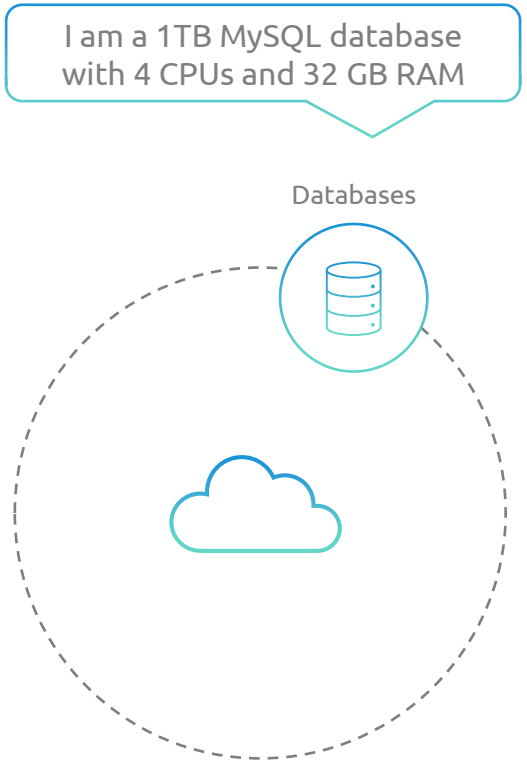
Client Request/ Server Response

Server

What Is Cloud Computing?

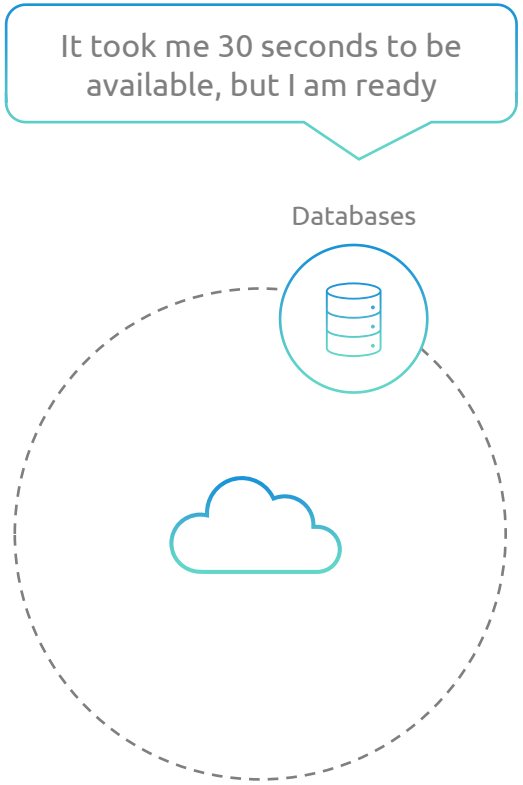
Pay only for what you request or use



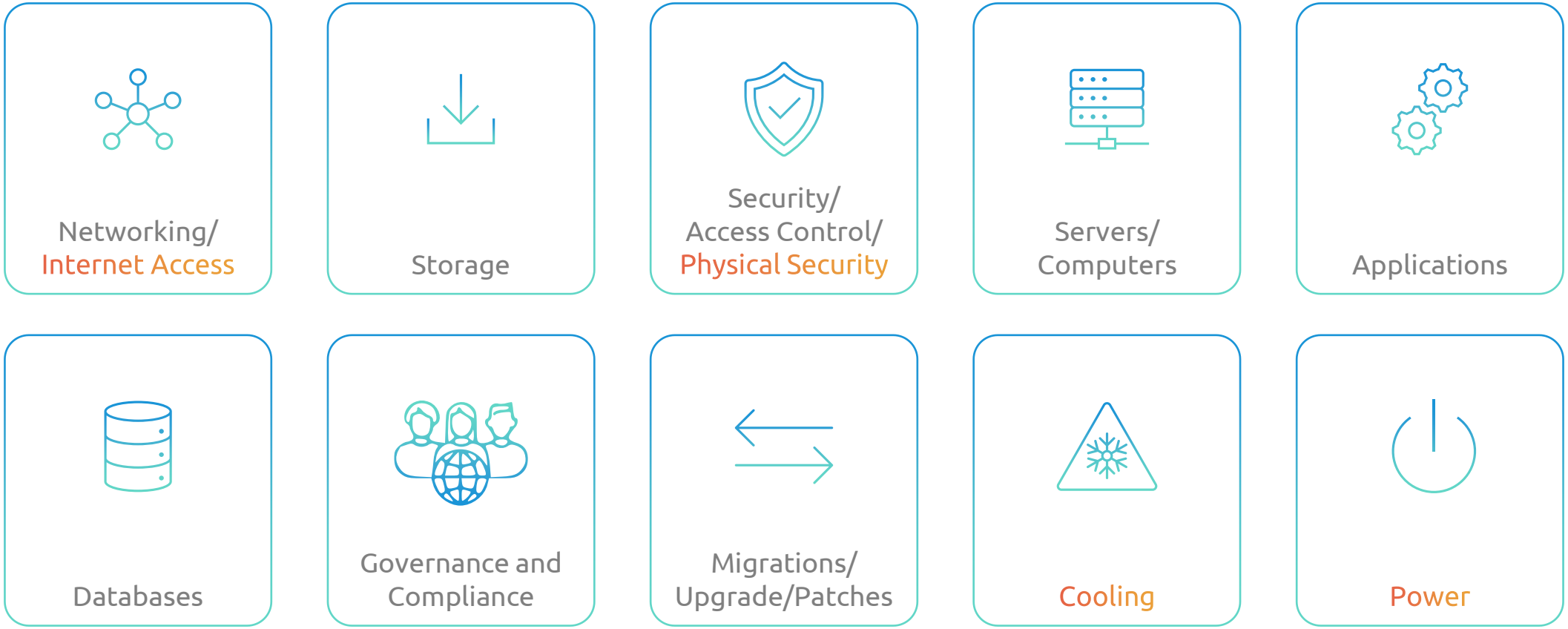


What Is Cloud Computing?

Access your requested resources in seconds or minutes, but sometimes instantly!



How Does Cloud Computing Help You?

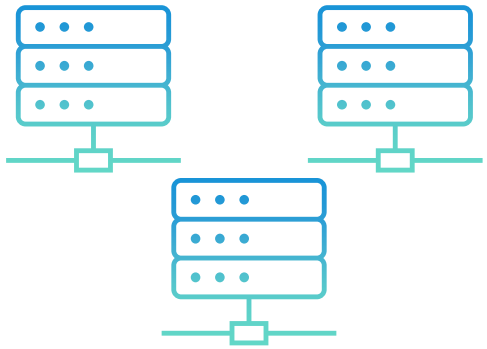


Cloud Computing Models

Cloud

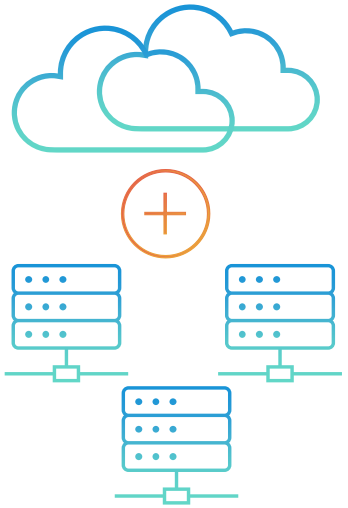
All in cloud

On-Premises



Not in cloud

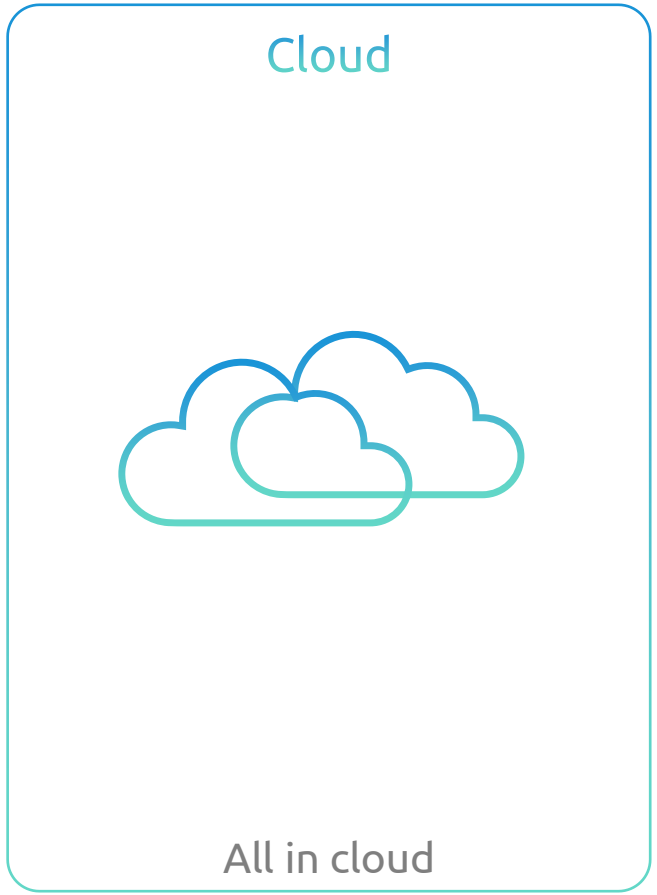
Hybrid



Some in cloud and some not in cloud

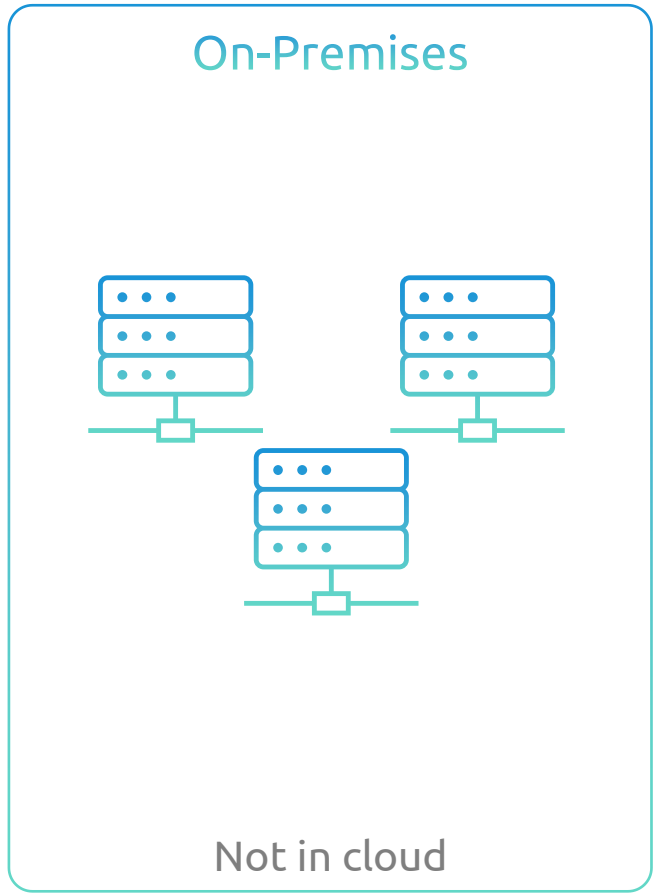
Cloud Computing Models – Cloud (only)

- Usually includes startups of all sizes; most businesses started after 2011
- Runs everything “in the cloud”
- Migrates existing projects to the cloud (if they exist)
- All new projects are in the cloud
- Shared Responsibility for Security and Operations



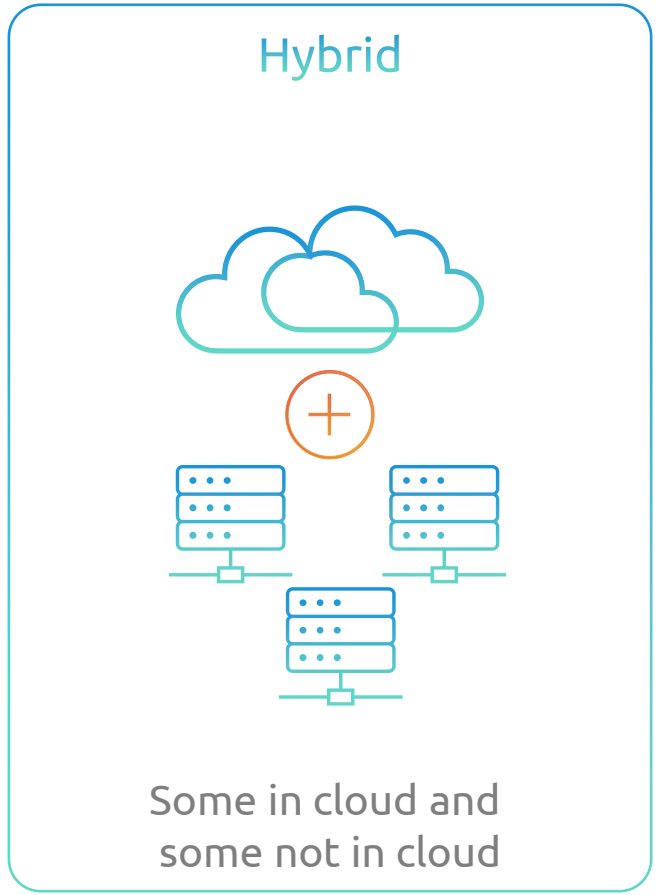
Cloud Computing Models – On-Premises

- Minimal to no Cloud Computing usage
- All Projects run in a owned data center or a rented data center
- Responsible for all security and operations
- Can be legacy companies that haven't had a reason to move to the cloud
- Can be companies that need strict control and security over the entire infrastructure



Cloud Computing Models – Hybrid

- Runs some parts of the application in cloud; other parts are on-premise
- Migrates existing applications to the cloud; may leave legacy applications on-premise
- Most new applications are designed and built for the cloud
- Usually fast connection between on-premise and cloud resources

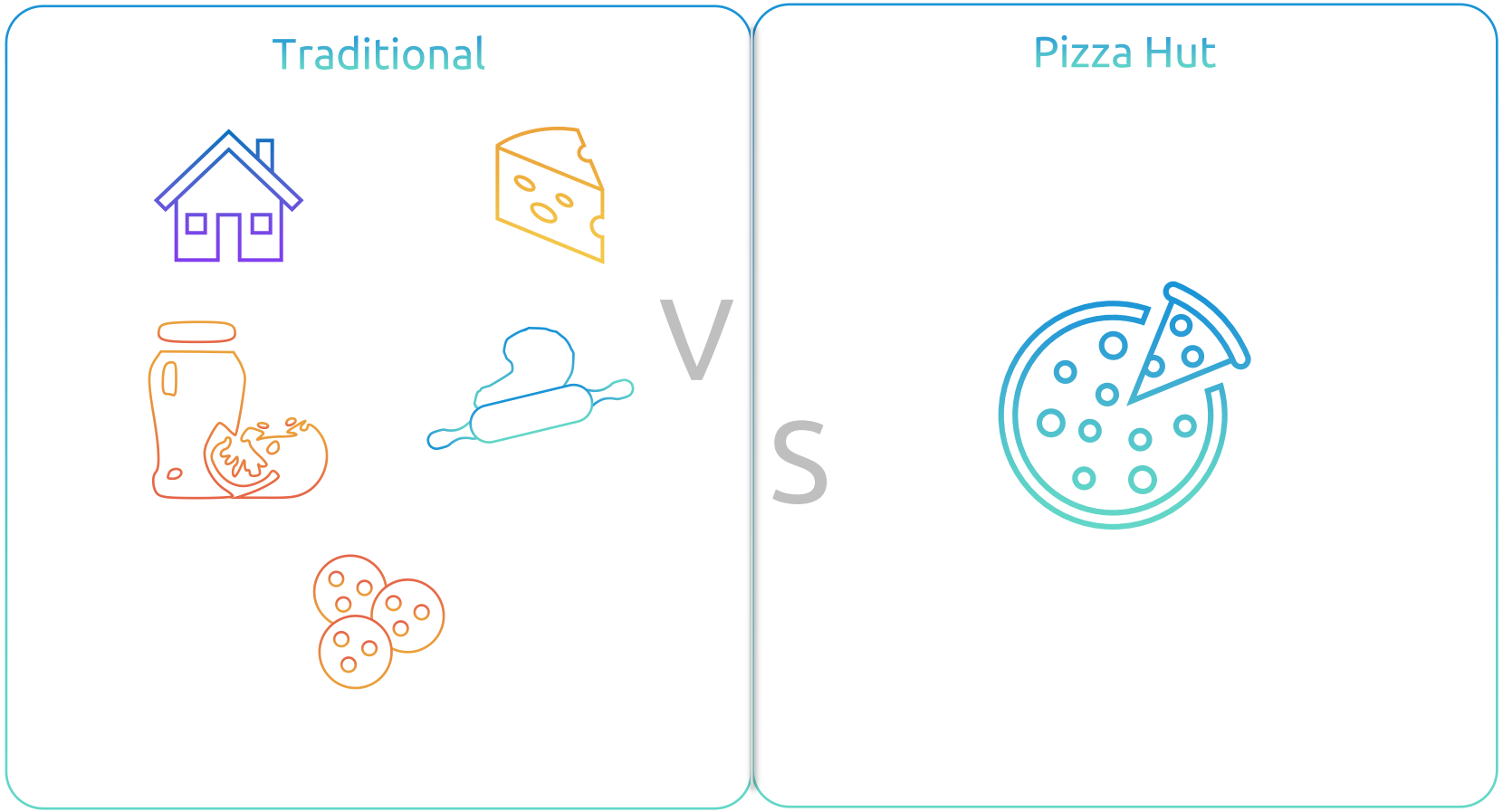


What Is Cloud Computing? - Summary

- Cloud Computing is the on-demand delivery of IT resources, particularly compute power, application hosting, database application, networking, and more
- Three models of deployment: Cloud, On-Premises, and Hybrid
- Works on a Client-Server model
- Provides almost instant pay-as-you-go access to compute resources/ app hosting

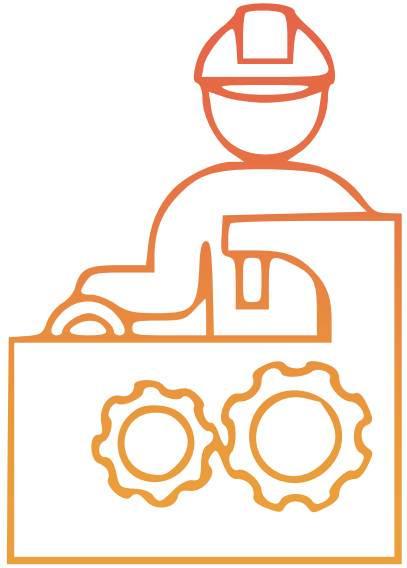
What Is Amazon Web Services?

Amazon Web Services Versus Traditional IT



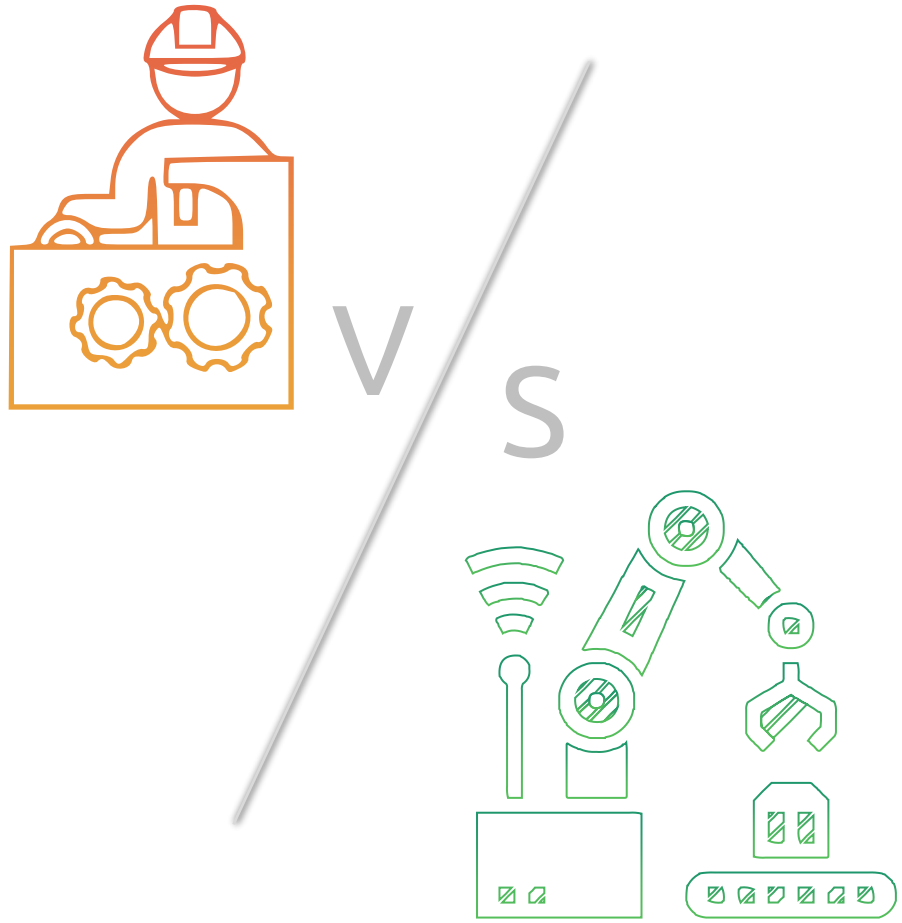
Amazon Web Services Versus Traditional IT

- Imagine producing an idea with new tech, but this time we are using AWS
- In our original Traditional IT request, you had to request for:
  - Hardware (optional)
  - Network cable (optional)
  - Power cable (optional)
  - Server placement with cooling (optional)
  - Operational and security installation (optional)
  - Access granted to requestor (3-6 weeks or 3 days)
  - Software installation by requestor (???)

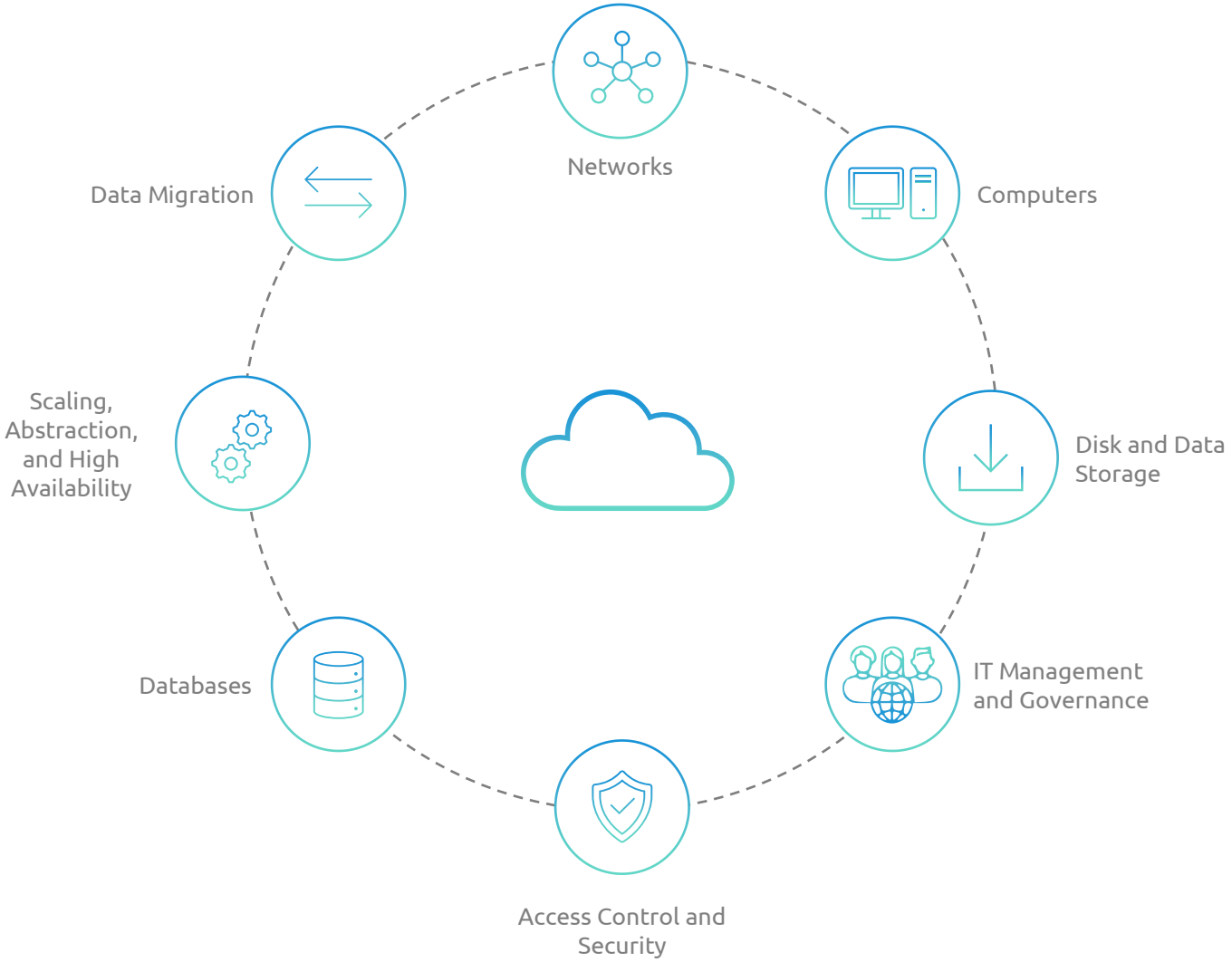


Amazon Web Services Versus Traditional IT

- Imagine if instead of provisioning hardware, you provisioned a hardware service that included networking and power
- Instead of managing physical objects, you get access to services that you can configure with ease

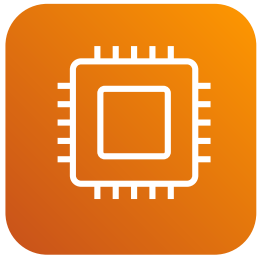


What Is Cloud Computing?



What Is Amazon Web Services?

AWS Core Service Categories



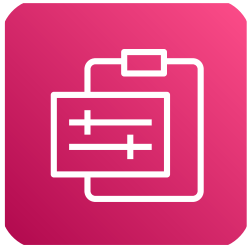
Compute

Networking and
Content Delivery

Storage

Database

Security, Identity,
and Compliance

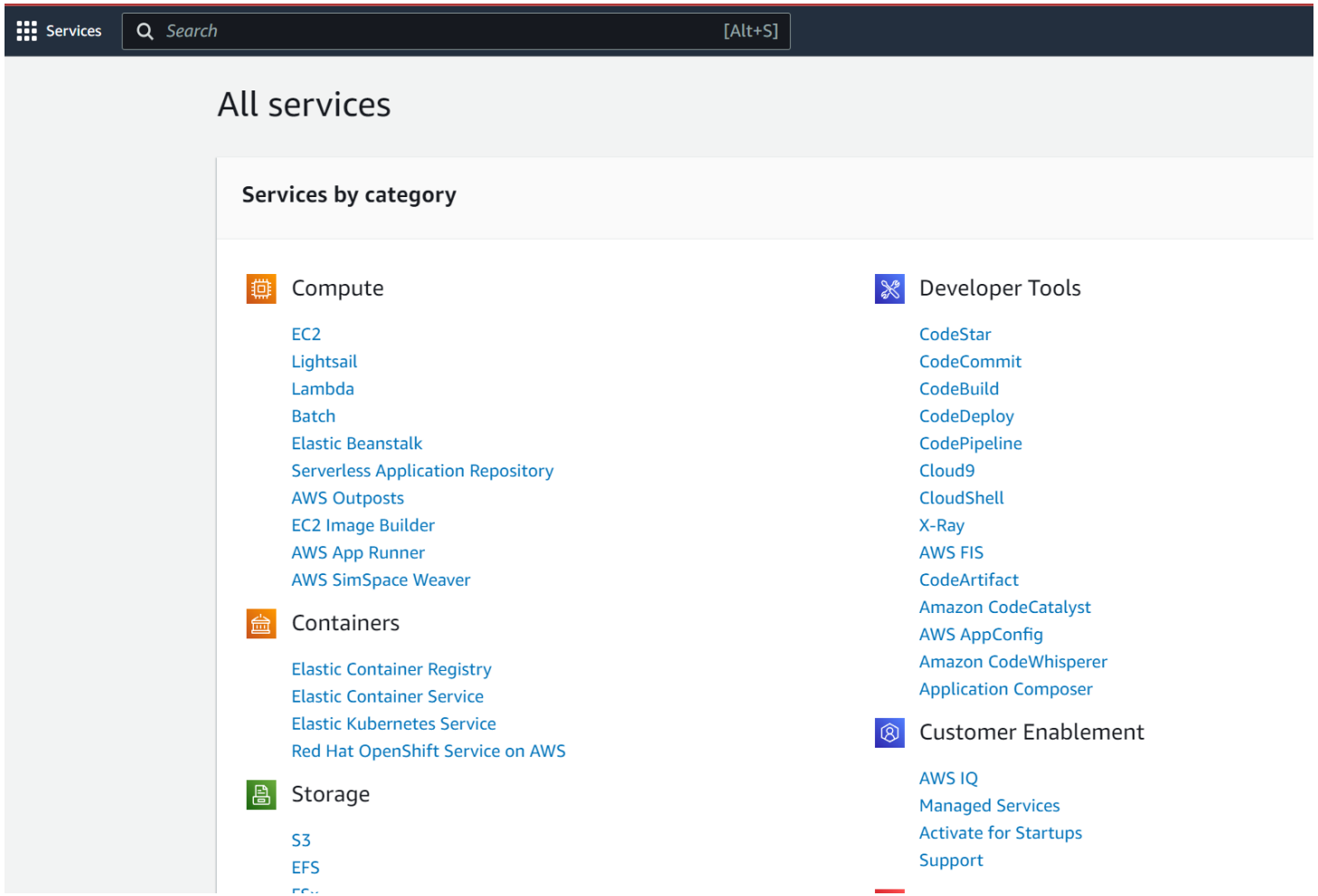


Management
and Governance

Three Ways to Interact With AWS

1 **Console**

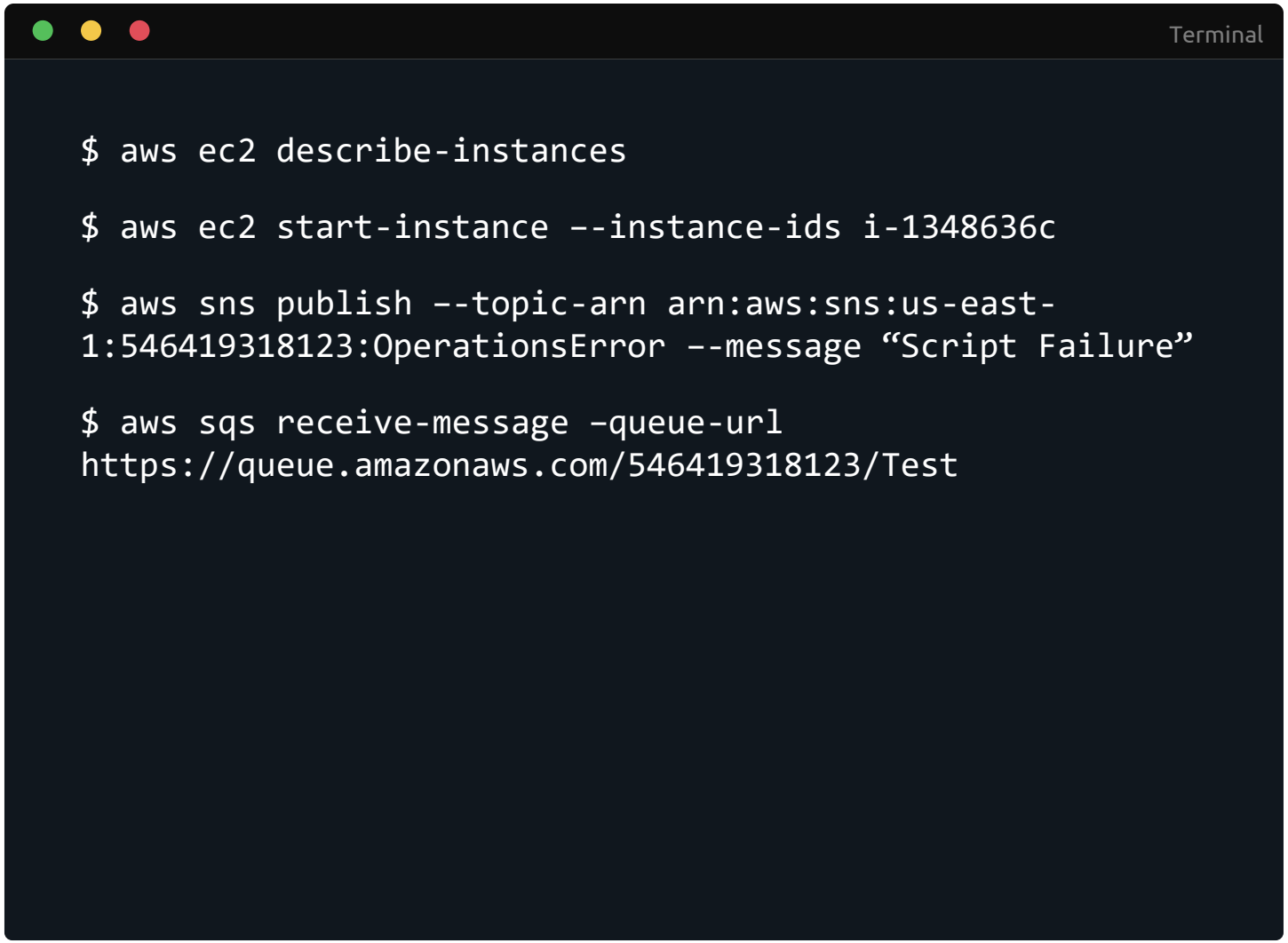
Great to learn and confirm



Three Ways to Interact With AWS

2 AWS CLI

Great for engineers



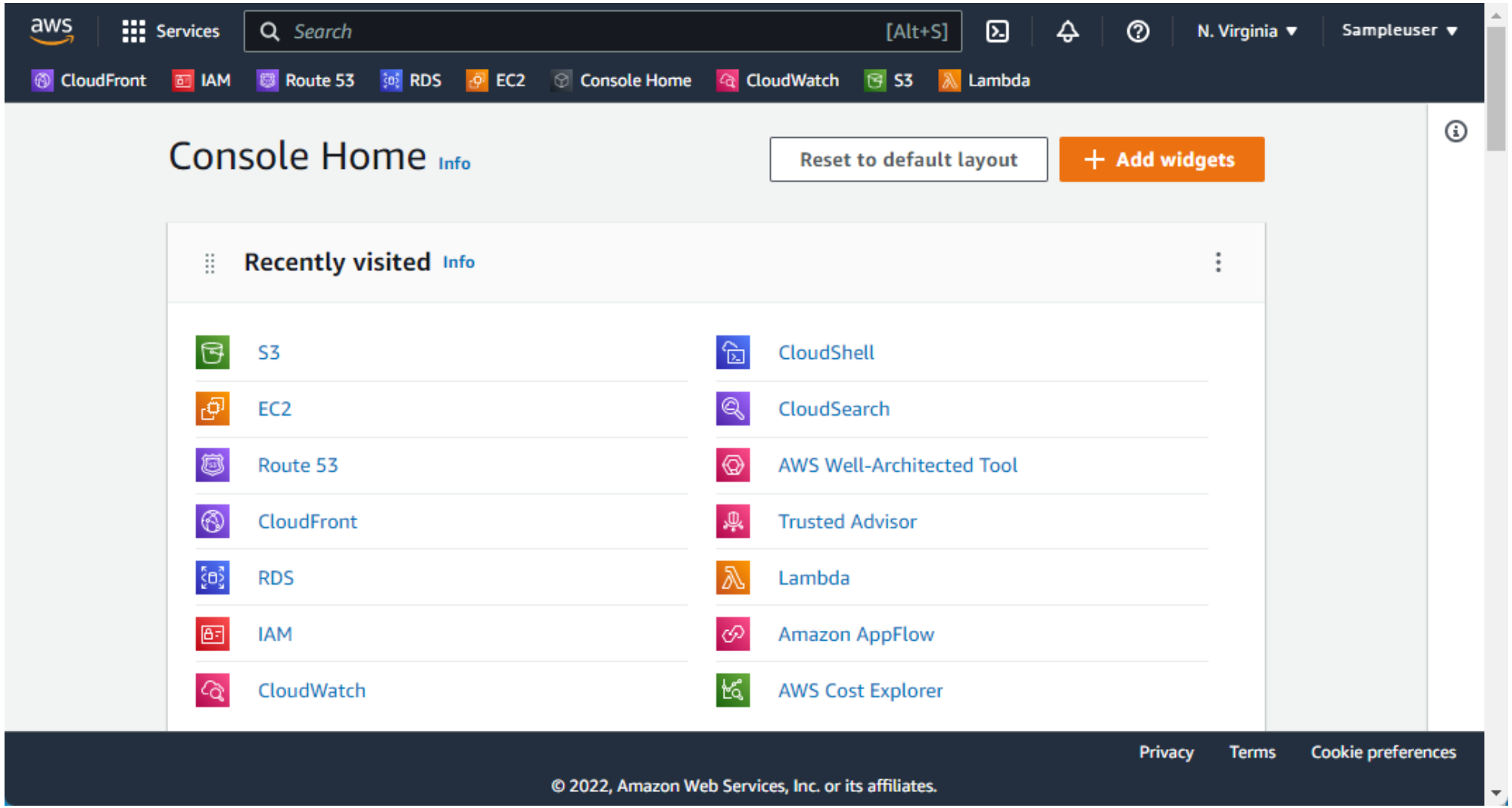
Three Ways to Interact With AWS

3

AWS SDK

For developers and different types of coders

Let’s Take a Brief yet Informative Tour!



What Is AWS? - Summary

- Amazon Web Services or AWS was the first large-scale cloud provider
- AWS was launched in 2006, with S3 as the first service
- Since then, AWS has grown to 300+ services
- Signing up is free; all services usually are pay-to-use
- AWS has one of the largest communities, market positioning, and growth in the industry

Benefits of Cloud

Trade Upfront Expense for Variable Expense

Upfront Expenses (CAPEX)

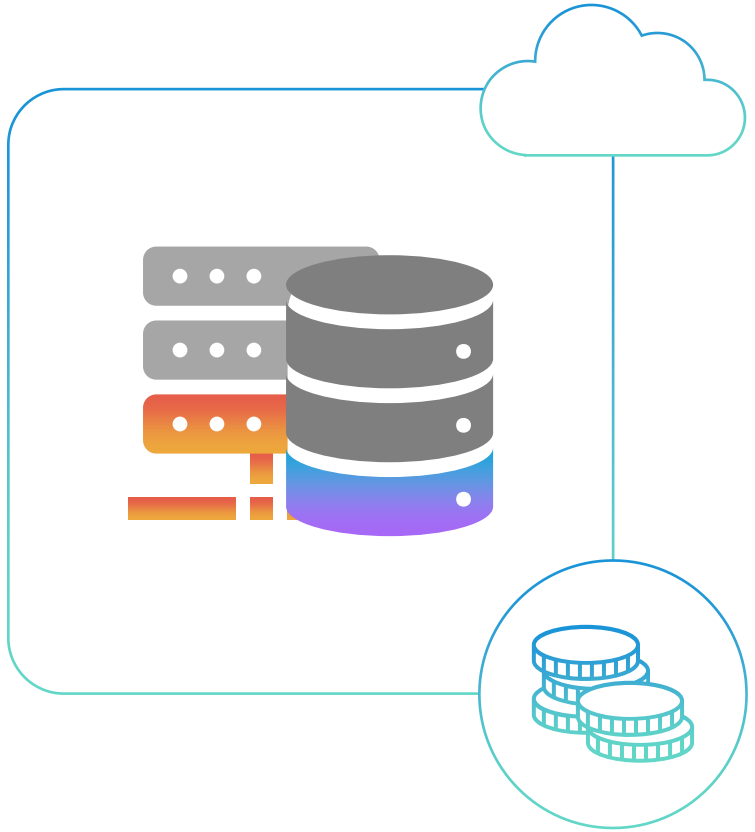
Large upfront investment in hardware before use

Trade Upfront Expense for Variable Expense

Variable Expenses (OPEX)

Pay for usage and requests

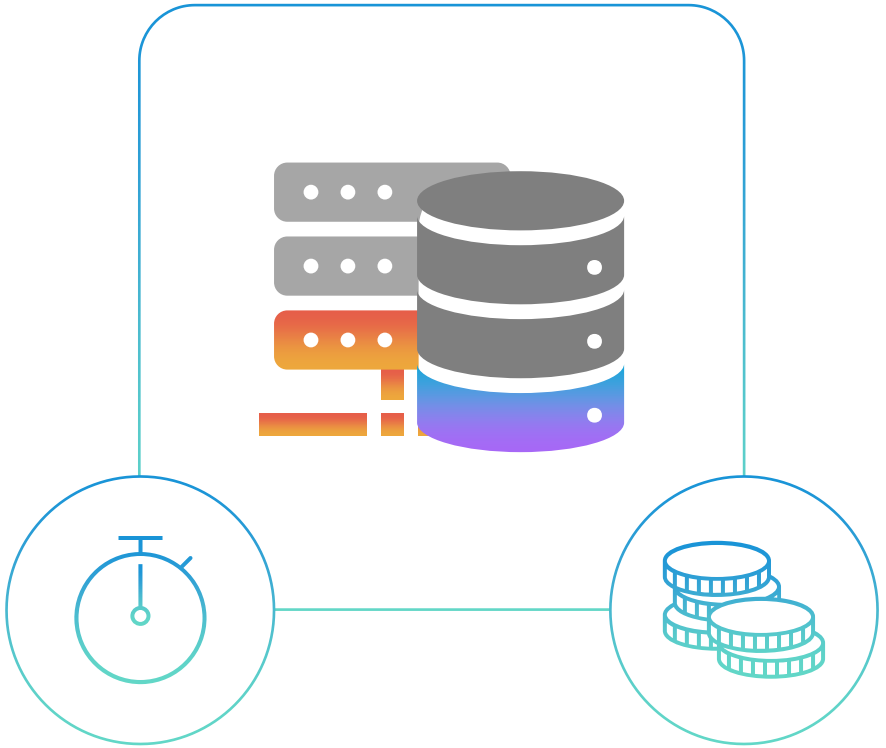
Give back what you don't need



Stop Focusing on Data Centers

Focus on Data Centers

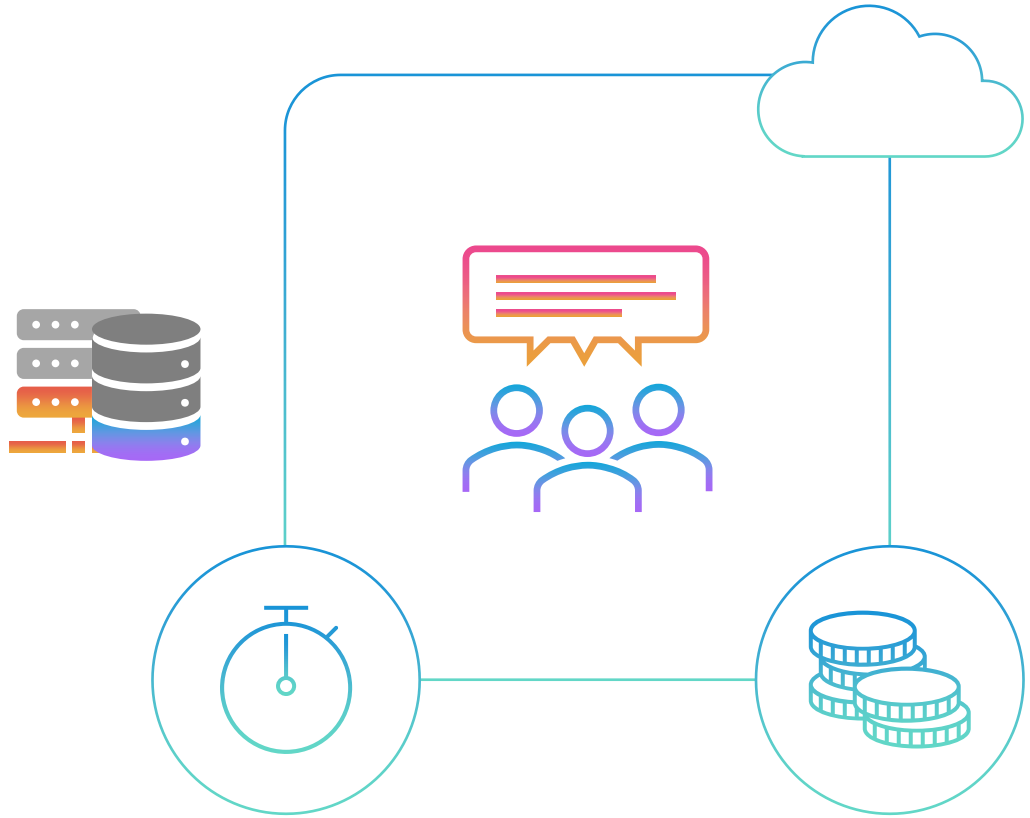
Invest time and money; focus on data centers



Stop Focusing on Data Centers

Focus on Customers

Invest time and money; focus on customers and applications



Stop Guessing Capacity (or easily fix bad guesses)

Limited Scalability

Bound by hardware limitations or vendor supply limits

Stop Guessing Capacity (or easily fix bad guesses)

Extreme Scalability

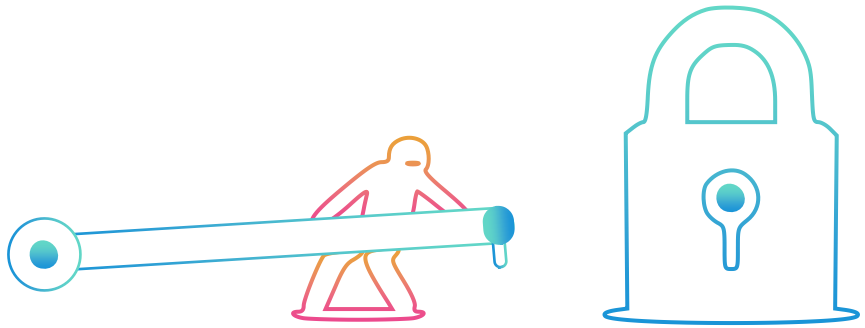
Scales in and out as needed; experiments with load and performance as needed

Stop Guessing Capacity (or easily fix bad guesses)

Benefit From Massive Economies of Scale

Smaller Scale

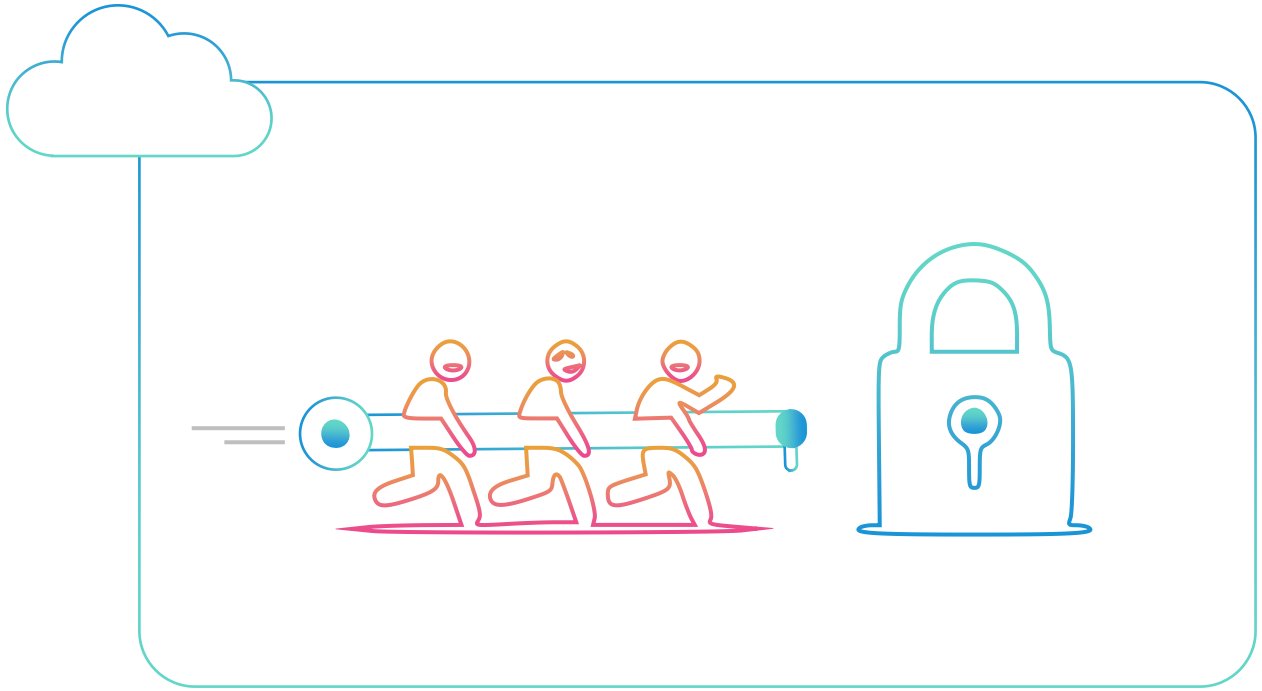
Pay higher prices based on only your own usage, with no price drops per unit



Benefit From Massive Economies of Scale

Economies of Scale

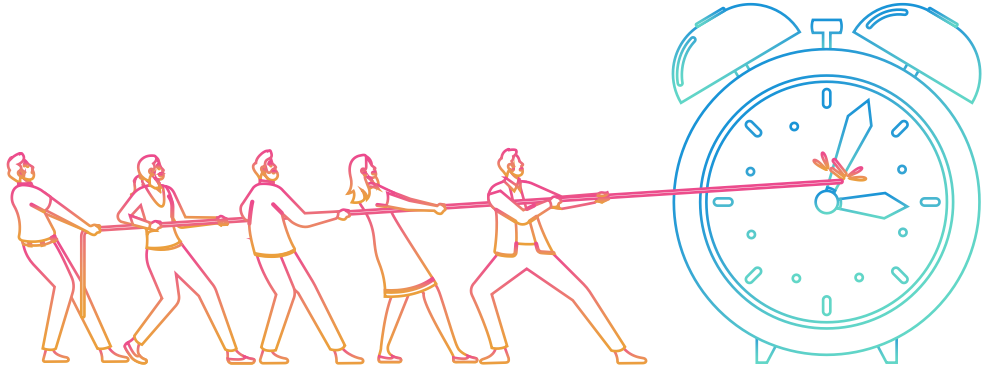
Benefit from customers' aggregated usage, with price drops per unit used



Increase Provision Speed and Business Agility

Data Center Requests

Often provision in days or weeks or months



Increase Provision Speed and Business Agility

Cloud Computing Requests

Always provision in seconds or minutes

Go Global (with your app) in Minutes!

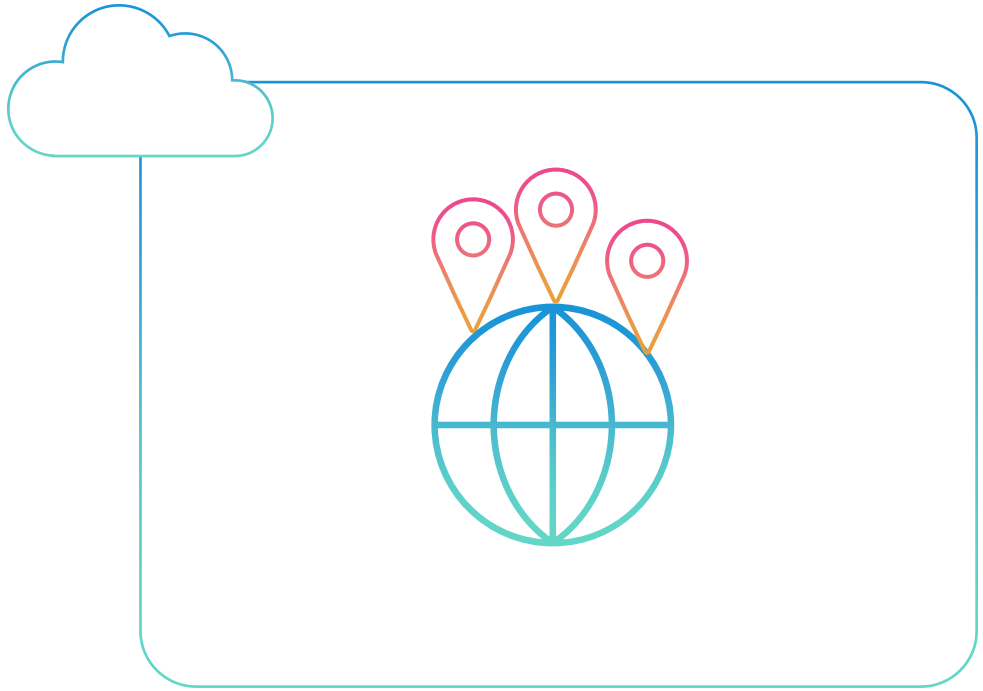
Limited Deployment Options

Limited to places where data centers are available

Go Global (with your app) in Minutes!

Global Deployment Options

AWS global infrastructure available at 31 locations globally



Benefits of Cloud Computing - Summary

- Trade upfront expense for variable expense
- Stop focusing on data centers
- Stop guessing capacity
- Benefit from economies of scale
- Increase your speed and agility
- Go global in minutes

Cloud Computing – The Economics

Economic Models of AWS – Overview

Free Tier

Volume Discounts

On-demand

Price Drops

Reservations

Free Tier

On-Demand

= Size

On-Demand

= Size

On-Demand

2.1 GB = 2.1 GB 

On-Demand

2 Processor

16 Gigabyte

10 Hours

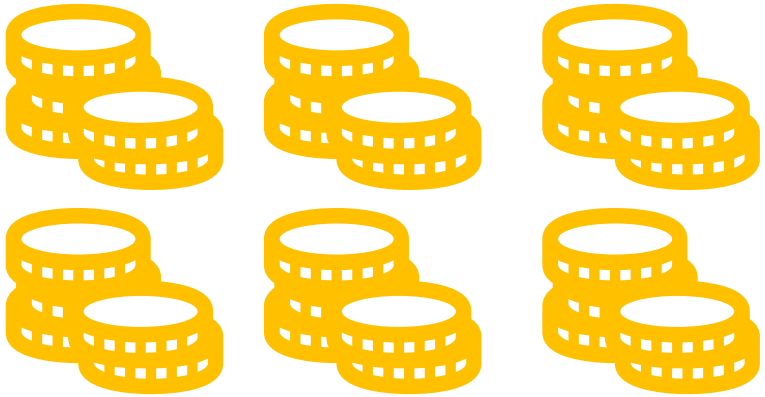
Reservations

Reservations

Reservations

Volume Discounts

Volume Discounts



Volume Discounts (S3 Standard)

Volume Discounts (S3 Standard)

1 cent per GB at:

First 50 TB per month

Next 450 TB per month

Greater than 500 TB per month

Price Drops

Price Drops

Price Drops

Price Drops

2006 — 2023

129 Price Drops

Price Drops

0.15 USD per GB

2006

0.023-0.026 USD per GB

2023

us-east-1

Cloud Computing Economics - Summary

- Free Tier means that certain services are always free, and some are free for 12 months after new account creation
- On-Demand is full pricing and pay-as-you-go, but no contract and very flexible
- Reservations are contracts that you enter with Amazon for 1-3 years
- Volume Discounts are pay less per unit as you use more
- Price Drops are random price cuts that AWS does every few years on its services

Cloud Design Principles

Principles of Cloud Native Design for AWS – Where We Are Headed!

- Design for failure
- Decouple components
- Implement elasticity
- Think parallel

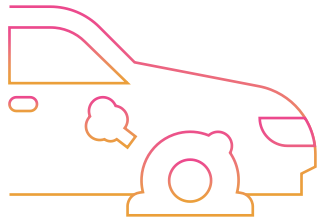
Principles of Cloud Native Design for AWS – Before We Dive in….

- There is a deeper framework for AWS Architecture, but it is not needed at the Cloud Practitioner level

- It is called the AWS Well Architected Framework

Design for Failure

Single Points of Failure



What should happen when my component fails?

Increased Resiliency

Resiliency and Auto-Recovery

Be intentional about failure and add redundancy

"Everything fails all the time." – Werner Vogels

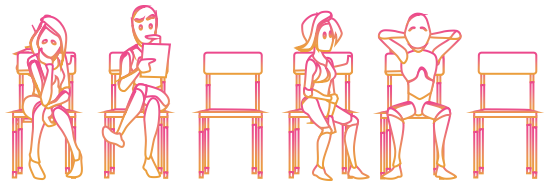
Decouple Components

Tight Coupling

What happens in case of a surge in customer requests?

No Data Loss

Loosely Coupled With Queue



Put a queue or scaling layer in between components

"Watch for Coupling that is too tight" – Steve McConnell, *Code Complete*

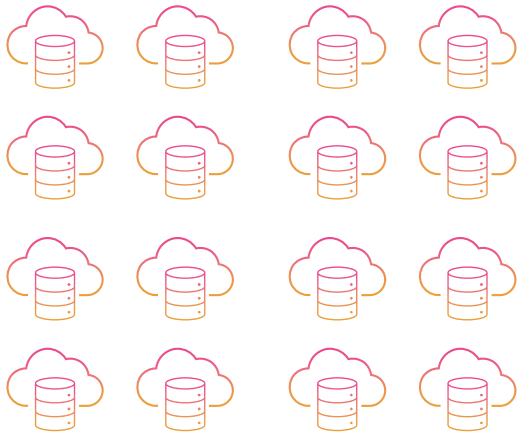
Implement Elasticity

Limited Expansion and Contraction

Hardware boundaries are small

Better Costs;
Better Performance

Large Expansion and Contraction



Hardware boundaries are large

"The ability to acquire resources as you need them and release resources when you no longer need them. In the cloud, you want to do this automatically." – AWS Docs

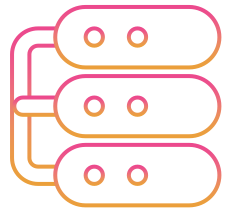
Think Parallel

Do Things in Series

Do you want 1 server to do it in 36 hours?

Increased Concurrency

Do Things in Parallel



Do you want 3 servers to do it in 12 hours?

Cloud Design Principles - Summary

- Design for failure
- Decouple components
- Implement elasticity
- Think parallel

KodeKloud