

de Porfiodeacises
acvited
gendifo/ Penel)
ecesattlnajof
ystund.
acesstlgrgstconserter
agelock
odalreteressconserter
upcatonMattide
ogallerargerconserter
eosettings
odallertsterconserter
ofiun: fastizing
ocesstigrperconserter
reldings.
ocesttigrApoina
chonaplag to
l forpahin
acestti conserter
cr/onLT
ocester orius.
ollinh
ogeseti
acesallg conter
each
goecatigres
e:
acesallgracnafres,
dure
acesstigsterconserter
essude..
ecagalloystappararter
engs.
acesetlgrgerconserterton
oytasting:
togafl lgo:
died.co repany



Golang (Go) Programming Language

SULTAN - 56189 - BS/DS-5_1



Introduction to Go

Developed by Google

Created in 2007, released 2009 by
Griesemer, Pike, Thompson.

Efficiency & Simplicity

Combines C's efficiency with
Python's simplicity.

Scalability & Performance

Statically typed, compiled for
modern cloud, server, distributed
apps.

Uses of Golang



Backend Web Development

Building robust and scalable server-side applications.



Cloud Infrastructure

Powering cloud services and platforms.



DevOps Tools

Creating efficient tools for development and operations.



Network Programming

Ideal for high-performance network applications.

Go's concurrency with goroutines and channels handles multiple tasks efficiently.

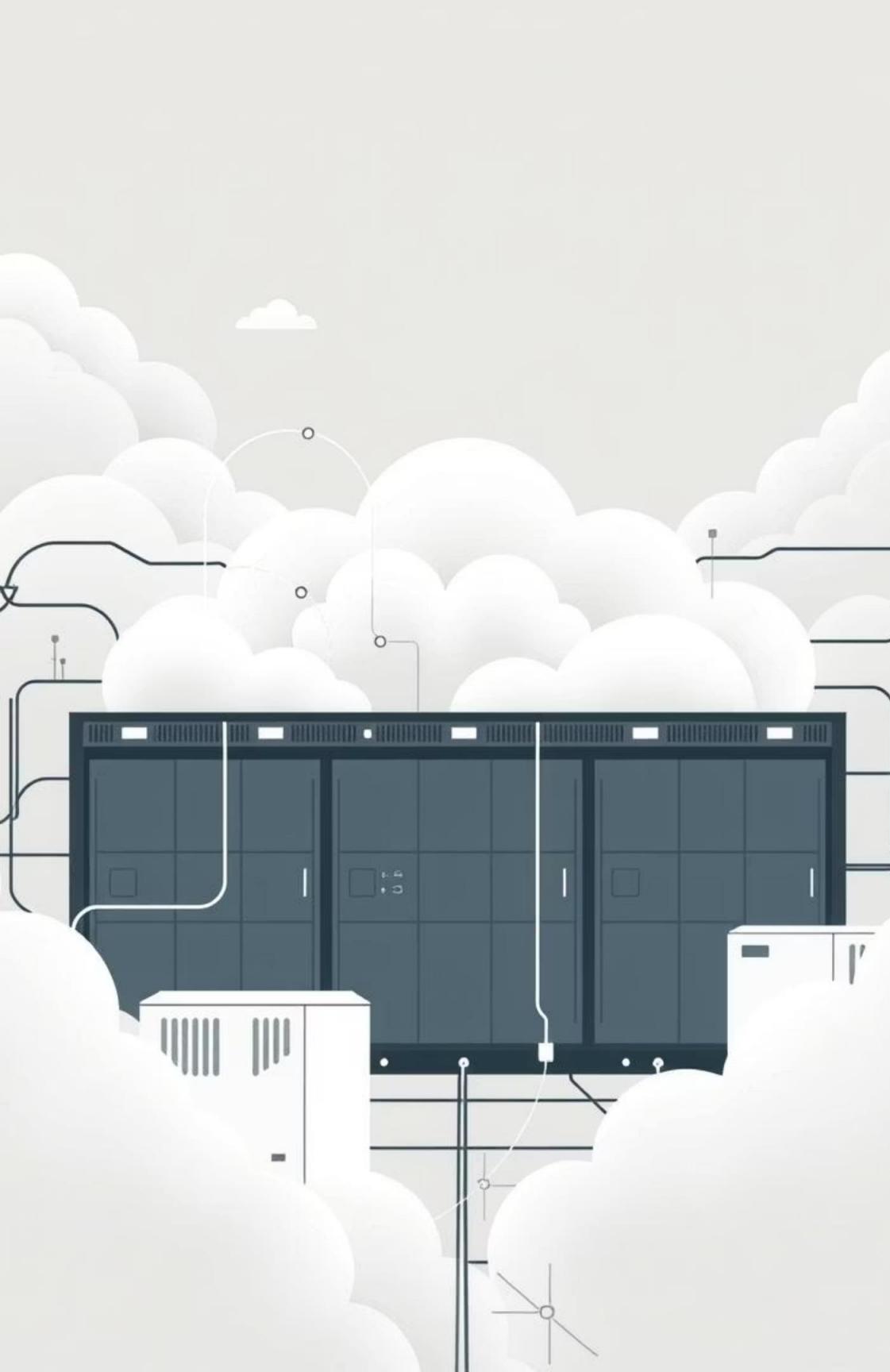
Go in Data Science

While not as dominant as Python, Go is gaining traction for:

- High-performance data processing
- Real-time analytics
- Building scalable data pipelines

Its speed and concurrency are suitable for big data and streaming microservices.





Real-Life Example: Docker & Kubernetes



Docker

Containerization platform built with Go.



Kubernetes

Orchestration system also written in Go.

These Go-powered technologies revolutionized cloud computing and container orchestration, showcasing Go's simplicity, speed, and scalability for high-performance backend systems.

Comparison: Go vs Python vs C++

Typing	Static	Dynamic	Static
Compilation	Compiled	Interpreted	Compiled
Speed	Fast	Moderate	Very Fast
Ease of Use	Easy	Very Easy	Moderate
Concurrency	Built-in (goroutines)	Threading (limited)	Threading/Manual
Use Case	Cloud, Backend, Networking	AI, Data Science, Automation	System Software, Games

Sample Golang Syntax

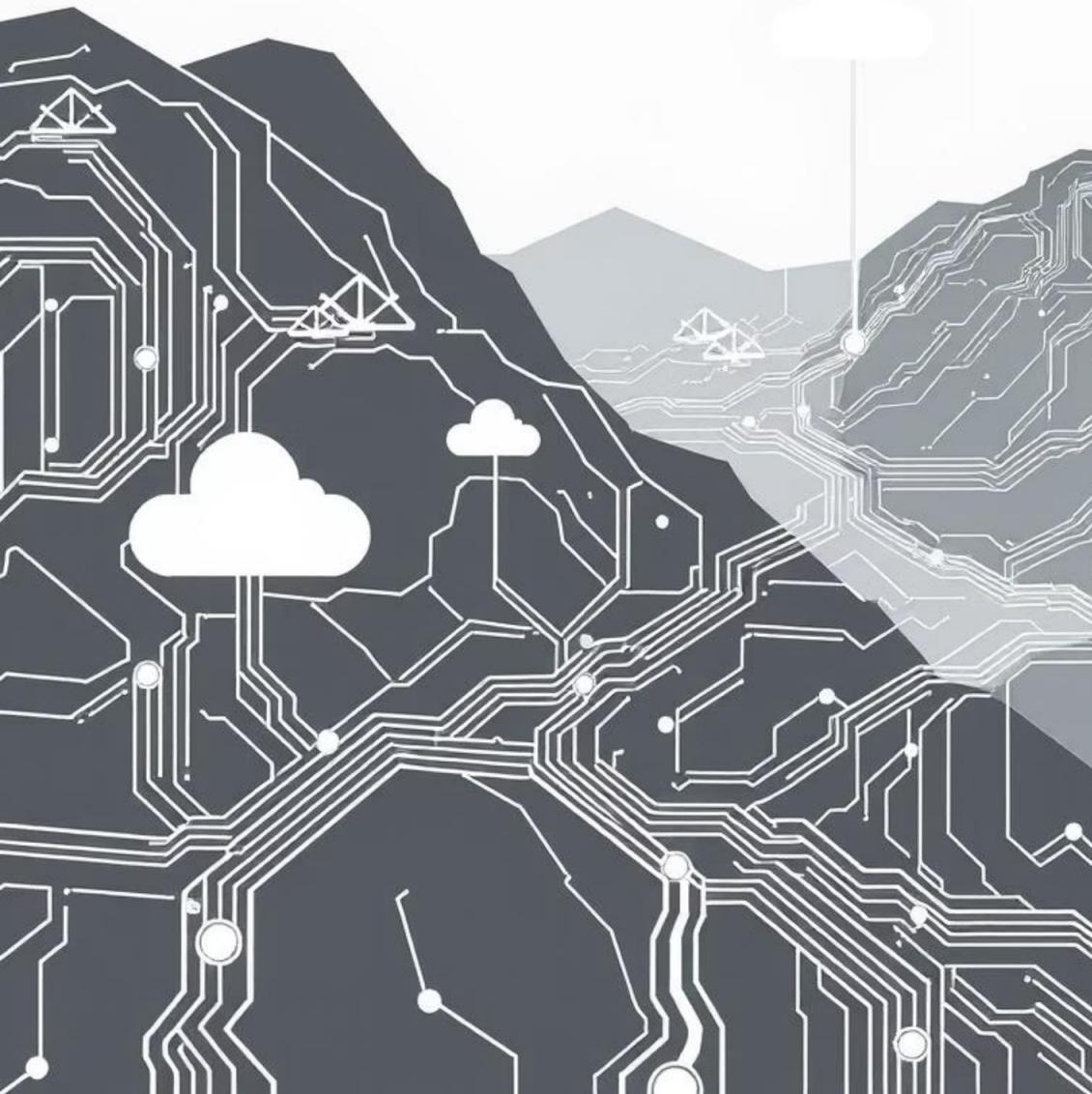
```
package mainimport "fmt"func main() {  
    fmt.Println("Hello World")}
```

This simple "Hello World" program demonstrates Go's clean and concise syntax.

```
7      that putaled Weepe:  
8      Correcteticats: wart  
9      qually:  
10  
11  
12      purcaon ellle, -enecies ()  
13      { goo  
14      ( accourallerr-clanentles:  
15      / accoMeldrar-porhacio:  
16      / coo  
17      / accouralltart larp  
18      / econWaltanendmiste, cho-coclonnet: any / ebueleniond tape,storl  
19      / rendllar:  
20      { accoWelore -carrectnent)  
21      / accolocat/ogae-Airt paretute cuso.lo:  
22  
23  
24      / accourol/ISby:  
25      . auact/elcadil, munition:
```

Hello World

Conclusion



Powerful & Efficient

Bridges simplicity and performance.

Modern Design

Concurrency model, cross-platform capabilities.

Ideal for Scalable Apps

Cloud, networking, backend systems.

Emerging in Data Science

Immense potential in data engineering.

GITHUB LINK: <https://github.com/sultanali543/INTRO-TO-GOLANG-GO>

**THANK
YOU
FOR YOUR
TIME**

