**Index**

**A**

Accessibility

Agile management

Agile methodologies

architecture

behavior-driven development

feature-driven development

hypothesis driven development

principles

test-drive development

transition

waterfall methodology

Always be architecting principle (AbAP)

Amazon Elastic Kubernetes Service (EKS)

Amdahl’s law

Anti-corruption pattern

Anti-Money Laundering (AML)

Anti-pattern

Application programming interface (API)

design principles

Application-specific integrated circuity (ASIC)

Architecture/design principles

container

configuration file

image immutability principle (IIP)

lifecycle conformance

observability

process disposability

runtime confinement

self-containment principle (SCP)

single concern

criteria

decision-making

definition

design

*See*Design principles

engineering principles

PNPP

shift-left

orthogonality

runtime

be smart with state

deploy independently

design for failure

isolate failure

location-independent

microservice failure

security principles

defense in depth

security by design

software quality principles

SOLID principles

TOGAF

Artificial intelligence (AI)

characteristics

definition

driven development

*See*Driven development tools (AI)

microservices architecture

speech implementations

acoustic analysis

algorithm

intelligent agent

ML model

preprocessing

recognition modules

vertical components

subcategories

Artificial intelligence for operations (AIOps)

analytics

anomaly detection

automation

benefits

capabilities

data gathering

definition

DevOps pipeline

enabled operation

event correlation

failure metrics

functions

IT Service Management (ITSM)

key IT KPIs

lifecycle management

traditional operations

transformation

use cases

Assessment techniques

automation

business aspects

capabilities/elements

cloud-native

*See*Cloud-native assessment

definition

detailed architecture

disconnected enterprise organization

execution approach/activities

findings/recommendations

methods

objectives

Asynchronous nonblocking I/O pattern

Amdahl’s law

blocking processing

BLTR relationship

characteristics

synchronous messaging

Atomicity, consistency, isolation, and durability (ACID)

Automate infrastructure

concepts

DevOps pipeline

features

IaC

*See*Infrastructure as code (IaC)

IT tasks

Automation

collaborative environment

eliminating waste

principles/practices

site reliability engineering

Automation maturity assessment

definition

enterprises

evaluation models

maturity assessment model

templates

Azure Kubernetes Services (AKS)

**B**

Bare-metal server/physical server

Behavior-driven development (BDD)

benefits

collaborative approach

concepts

disadvantages

principles/practices

solve problems

specification

techniques/principles

transition

Be smart with state principle (BSSP)

Big data management

*See*Mesh implementation

Blockchain as a service (BaaS)

Box-and port-style architecture

Broker topology

Bulkhead design pattern

Bulkhead pattern

Business continuity/disaster recovery (BC/DR)

**C**

Central processing units (CPUs)

Change Data Capture (CDC)

implementation

log-based approach

process

Chat-based operation (ChatOps)

benefits

bot integration

categories

chatbot architecture

collaboration model

concept and technology

group chat application

services

use cases

Circuit breaker pattern

Client-server service

Client-side discovery pattern

Cloud and cloud-native maturity model (CCNMM)

Cloud DevOps solution

AWS development services

Azure DevOps server

backup restoration

benefits

comparison chart

configuration

Google cloud (GCP)

infrastructure

monitoring/observability services

traditional on-prem solution

Cloud-native architecture

agile development approach

Agile management

benefits

cloud computing

cloud-enabled applications

containerization

CRUDoperations

database

document store

event-driven architecture

graph data store

key-value data store

RDBMS decision flow

relational database

search engine

time-series

types

wide column

database selection

data processing

definition

design principles/patterns

DevSecOps

elements

event-driven

industries

accelerated stage

approach/priorities

migration

scale/innovation

industries/geographies

intelligent data governance

journey

advantages

re-engineering approach

timeline/risks

transformational benefits

key characteristics

lift/shift

maturity model

advantage

cloud optimization

culture/innovation

definition

elements

enablement wave

organizational model

scalability/flexibility

transformation wave

mesh implementation

microservice approach

mobile computing applications

Monolithic legacy application

objectives

objects/files/blocks

organization/culture

organizations

polyglot persistence layer

polylithic and polyglot

processing/analysis

reduces operational expenditures

replication methods

serverless

software architecture

technical/nontechnical duties

traditional approach

Cloud-native assessment

assessment model

comparison

maturity model

Cloud Native Computing Foundation (CNCF)

Cloud steering committees (CSCs)

Cohesion

coincidental cohesion

communicational cohesion

definition

functional process

high and low level

high cohesion

logical cohesion

procedural cohesion

qualitative measure

sequential operation

temporal operation

Coincidental cohesion

Collateral management service

architecture

architecture principles

functional architecture

high-level interaction

Command and query responsibility segregation (CQRS)

application layers

benefits

CRUD operations

databases

definition

disadvantages

issues/use cases

traditional architecture

Communicational cohesion

Complex event processing CEP)

Computing deployment models

architecture

community

hybrid model

private/on-premises

public cloud

Concurrency control pattern

*See*Optimistic concurrency control

Configuration as code (CaC)

Consumer first principle (CFP)

Container as a service (CaaS) model

Containerization

adoption

architecture

benefits

blue/green deployment

client-server architecture

culture

definition

enterprise

environments

image characteristics

image process

immutable

infrastructure

Kubernetes environment

Linux Containers (LXC)

log management

monitoring

Most Value Product (MVP)

networking

orchestration

*See*Orchestration

patterns

principles

privileged container

registry

rolling deployment

security

stateful sessions

technology disruption

Container revolution

Continuous delivery (CD)

Continuous deployment (CD)

Continuous integration (CI)

Control coupling

Coupling

common coupling

content-level coupling

control data sharing

data model

dependency injection

external devices

highly coupled

law of Demeter

loose coupling

loosely coupled

message

modules/microservices

no coupling

stamp coupling

tightly coupled

types of

Cross-Site Scripting (XSS)

Culture of automation principle (CAP)

**D**

Database as a service (DaaS)

Database replication

ETL process

logical database replication

CDC process

full load refresh

methods

partial refresh replication

methods

physical replication

Data coupling

Data management patterns

CQRS pattern

event sourcing pattern

partition

replication

DataOps analytics

data lifecycle

data operations pipeline

definition

pillars

principles

stages

Data-Structured Coupling

Decentralize everything principle (DEP)

Decoupling

approaches

architecture

change data capture

cloud-native architecture

data accessible

architecture

databases/data movement

events

methods

data process

definition

event store

legacy systems

MVP

principles

stream processing

techniques

*See*Decoupling techniques

transaction databases

Decoupling techniques

approaches

big-bang approach

business case

combination

continuous modernization

DDD

*See*Domain-driven design (DDD)

event storming

forces/technological innovation

monolithic legacy application

architecture

business layer

organization’s approach

unicorns/traditional organizations

monolithic legacy applications

definition

legacy services

present-day architecture

principles

requirements

@Scale IT

strategies

technical debt

*See*Technical debt

Defense in depth principle (DiDP)

Dependency inversion principle (DSP)

Deploy independently principle (DIP)

Design for failure principle (DFFP)

Design patterns

Design principles

always be architecting principle

application programming interface

consumer first principle

culture of automation principle

decentralize everything principle

digital decoupling

event storming

evolutionary design

interoperability principle

modeled with business domain principle

monolithic application

polyglot persistence principle

polylithic architecture principle

single source of truth

Detailed architecture assessment model

architecture

capabilities

capture content phase

definition

planning phase

recommendation phase

rules management

workshop phase

DevNetOps

agility and quality

network operation

network pipeline details

NRE principles

pipeline

virtualized network services

DevOps

automation solution

code management

Codota

continuous feedback loops

DeepCode

Kubeflow

ML model pipeline

monitoring alerts

observability

process

*See*DevOps process

quality assessment results

shift-left security approach

software engineering lifecycle

source code tools

Testim.io

transformation

challenges

considerations

development operations

journey

key factors

perspectives

DevOps process

AIOps pipeline

Automate infrastructure

continuous

definition

development process

DevSecOps

*See*DevSecOps

innovation

journey

pillars

DevSecOps

benefits

continuous delivery

continuous deployment

continuous integration (CI)

features

journey

principles

transformation

Digital decoupling principle (DDP)

Digital twin platform

aggregation

definition

environment setup

implementation

instance

manufacturing/business

prototype

Distributed Application Runtime (Dapr)

Docker Swarm

Domain-driven design (DDD)

BBoM system

complexities

concepts

domains

event storming

*See*Event storming

goals

model workshop

practices and guiding principles

single diagram

strategic model

tactical model

Don’t repeat yourself (DRY) principle

Driven development tools (AI)

approaches

concepts

definition

deploy/industrialize

DevOps

enable tools

evolution

framework

governance

hub/spoke model

identification

long-term value

measurement

methodology

principles

process

project execution

proof of concept (PoC)

requirements

stakeholders

unique challenges

value identification process

Dynamic code analysis (DAST)

Dynamic security system testing (DAST)

**E**

Edge computing

Enterprise computing service

Enterprise service bus (ESB)

Event-driven architecture (EDA)

asynchronous communication

box-and port-style component

characteristics

cloud-native architecture

components

Dapr architecture

decoupling

DevOps

encrypting events

events

brokers

business

enterprises/systems

governance

inter/intradomain communication

internal/external inputs

processing

technical events

evolution

FTP, RPC, TCP/IP protocol

interaction

maturity model

message queues

messaging model

payment platform

processing styles

real-time interactivity

security

sender/receiver

serverless architectures

streams/message queue

topologies

transaction management

Event-driven patterns

Event mesh

across cloud providers

architectural layer

brokers

capabilities

characteristics

control plane

elements

implementation

Event sourcing pattern

CRUD operations

definition

event source

stream

Event storming

activities

aggregates

aggregators

API model

benefits

bounded context

communication

domain events and commands

tactical implementation

ubiquitous language

business process

business value

disadvantages

domain model

entity

events

four-step approach

key roles

microservices identification

microservices model

misconceptions

objectives

process

relationship (command/events)

value objects

Event stream processing (ESP)

Evolutionary design principle (EDP)

Extended reality (XR)

augmented reality (AR)

definition

5G network

mixed reality (MR)

reality virtuality existence

virtual reality (VR)

External coupling

Extract, Transfer, and Load (ETL)

**F**

Fail fast implementation

Failure as a service (FaaS)

Feature-driven development (FDD)

benefits

communication

definition

disadvantages

feature specification

processes

Field programming gate array (FPGA)

Fifth-generation (5G) technology

advantages

definition

features and capabilities

frequency ranges and operates

3GPP mobile network

network slicing

trends

File Transfer Protocol (FTP)

Fitness function (-ilities)

automated execution

categories

code package

compliance

coupling/cohesion

definition

extensibility/reusability/maintainability

genetic algorithm

identification

identification/calculation

manual/continual execution

metrics

objectives/quantifiable results

observability

performance

resiliency

scalability

strategies

**G**

Genome sequencing

Global data coupling

Google Kubernetes Engine (GKE)

Governance

change management

decentralization approach

framework

intelligent tooling

objectives

operating model

security

strategies and functions

Graphics processing units (GPUs)

**H**

HashiCorp Configuration Language (HCL)

Header versioning approach

Hexagonal architecture

High availability (HA)

High Observability Principle (HOP)

Homomorphic encryption (HE)

Horizontal partitioning/sharding

HTTP/3

Hypothesis driven development (HDD)

concepts

culture

ecommerce application

evolutionary approach

framing process

objective criteria

scientific method

scientific steps

sidebar link

**I**

Idempotent service operation

-Ilities (software design)

accessibility

active-active deployment

active-passive deployments

automation

bath tub curve

communication failure

customer-centric design

dependencies

deployment environment

designing/developing security

availability

CIA triad

cloud-native security

compliance as code

confidentiality

decentralized approach

defense in depth

failures

homomorphic encryption

integrity

password policy

policy as code

secure API

shift-left security

single pane/audit

threat modeling

validating input

zero-trust model

domain requirements

elasticity

ethics

event-driven architecture

failures

fitness function

*See*Fitness function (-ilities)

functional requirements

high availability configuration

infrastructure

interoperability

maintainability

objectives

observability

Pareto analysis

partial list

portability

reliability

resilient application

software engineering methodology

sustainability

UI architecture

usability

Image immutability principle (IIP)

Infrastructure

automation

*See*Automate infrastructure

characteristics

cloud-native infrastructure

containerization

Kubernetes

requirement

Infrastructure as a service (IaaS)

Infrastructure as code (IaC)

Ansible

AWS Cloud Formation

capture requirements

CFEngine

chef

code structure

coding language

comparison

elements

GitOps

implementation

network and storage

pipeline automation

provisioning infrastructure

Puppet

SaltStack

service deployment

terraform

terraform and Google Cloud

text-based file

tools

Infrastructure services

client-server

cloud services

communication/internet

computing adoption

enterprise computing

evolution

IaaS

mainframe services

Metcalfe’s law

minicomputers

mobile computing services

Moore’s law

PaaS

personal computer

SaaS

servers

*See*Server evolution

services

stages

storing digital information

Init containers

Integrated development environment (IDE)

Integrated real-time monitoring

Intelligent operations

AIOps

*See*Artificial intelligence for operations (AIOps)

applied intelligence

ChatOps

cloud enablement

data-driven approach

elements

enterprises

inefficiencies

IT operates

post-production deployment

right talent and skill

service approach

smart partnership

traditional approach

unicorns and modernized competitors

Interactions

choreography-based approach

communication

event mesh

loosely coupled services

mesh architecture

orchestration

service mesh implementation

SOA requests

Interface segregation principle (ISP)

Interoperability principle

Isolate failure principle (IFP)

**J**

JEVONS paradox theory

**K**

Keep it short and simple (KISS) principle

Key management system (KMS)

Kubernetes as a service (KaaS)

Amazon Elastic Kubernetes Service (EKS)

Azure Kubernetes Services (AKS)

capabilities

definition

Google Kubernetes Engine (GKE)

Red Hat OpenShift

VMware technologies

Kubernetes framework

application locking

architecture

automated placement

building confidence

containers

declarative deployment

deployment

elements/practices

features

health check process

initialization

Kubernetes as a service (KaaS)

mature and streamlined containers

maturity model

measurement and control

multitenant cluster

networking host

orchestration tool comparison

pod deployment and management

predictable demands

preparation phase

principles and patterns

secrets

selecting factors

service mesh pattern

sidecar

singleton service

stateless network

transform phase

**L**

Law of Demeter (LoD)

Lifecycle conformance principle (LCP)

Liskov substitution principle (LSP)

Location-independent principle (LIP)

Logical cohesion

Logical database replication

CDC process

full load refresh

partial refresh replication

Long-Term Evolution (LTE)

Loose coupling design

**M**

Machine learning (ML)

DevOps

evolution

Mainframe services

Mediator pattern

Mediator topology

architecture

components

coordination/orchestration

payment use case

Mesh implementation

architecture

data disruption

data lake architecture

data pipeline implementation

decoupling approach

domain-based pipeline

domain owns and serves

governance

infrastructure as a platform

monolithic lake platform

principles

self-service data

Mesos

Message-oriented middleware (MOM)

Message Queues (MQ)/stream system

AMQ/Kafka configuration file

capabilities

collaboration

flight reservation

interaction

Messaging model

Metcalfe’s law

Microservice architecture approach

Microservices

approaches

architecture

artificial intelligence

automation

autonomous

business capabilities

characteristics

collateral management

containerization

Conway, Melvin

CPU utilization

database/NoSQL

decentralization

definition

design consideration

distributed state

asynchronous

caching

collateral management

Kafka streams

responsibilities

state management

synchronous

elasticity

event mesh

evolution

failure/stability

Governance

hexagonal architecture

implementations/problems

independently dependency

interaction

living continuous design

payment processing

polyglot architecture

resilience

self-healing service

service mesh

smart endpoints/dumb pipes

technological capabilities

trade finance

traditional architecture

user interface

Minicomputer services

Minimum viable product (MVP)

Mobile computing applications

Modeled with business domain principle (MBDP)

Model-View-Controller (MVC) pattern

Molecular modeling

Monolithic architecture principle (MAP)

Moore’s law

Multitiered applications

**N**

Network address translation (NAT) services

Network-attached storage (NAS)

Network operations (NetOps)

Network reliability engineering (NRE)

Network slicing

**O**

Observability

AIOps process

algorithms

benefits

characteristics

cloud-native services

clustering/correlating

data collection

DevOps

features

full-stack observability

administrators/management

capabilities/services

characteristics

data sources

evolution

visualization

Kubernetes

meaning

microservices

monitoring

monitoring tools

scenarios

workflow steps

Open Application Model (OAM)

Open Authorization (OAuth 2)

Open-closed principle

Open Web Application Security Project (OWASP)

Optimistic concurrency control

concurrency control

ETag

meaning

modifications

pessimistic control

single request

Orchestration

Docker Swarm

environments

Kubernetes

*See*Kubernetes framework

Mesos

tools

Orthogonal principles

architecture

cohesion

coupling

intersect option

**P**

Pareto chart analysis

Partitioning pattern

hash sharding

list partitioning concept

range-based sharding

RDBMS/NoSQL

round-robin portioning

sharding/horizontal pattern

strategies

technologies

vertical partitioning

Pathological coupling

Pattern

anti-pattern

architecture style/design

data management

*See*Data management patterns

definition

design patterns

event-driven patterns

microservices

runtime

software architecture

Personal computing (PC) service

Pessimistic concurrency control

Platform as a service (PaaS)

architecture styles

deployment model

infrastructure

limitations/concerns

taxonomy

Polyglot persistence principle (PPP)

Polylithic architecture principle (PAP)

architecture

definition

granular subsystems

properties

Principle of Least Knowledge

Procedural cohesion

Process disposability principle (PDP)

Processing/analyzing data

components

definition

Kappa architecture

Lambda architecture

real-time data

Products not projects principle (PNPP)

**Q**

Quantum as a service (QaaS)

Quantum computing

definition

intensive workloads

potential

**R**

Radiation therapy

Range-based sharding

Red Hat OpenShift

Re-engineering approach

Remote procedure calls (RPCs)

Replication

definition

leader-based/leader-followers

asynchronous/synchronous

candidates

client request

clusters/data centers

election process

multileader

replication lag

single leader

quorum disk

cluster

definition

formula

optimal servers

tolerate unavailable nodes

write/read/read repair

requirements

Resilient application

approaches

bulkhead pattern

circuit breaker pattern

definition

events/conditions

failure responses

key capabilities

retry option

scenario

stateless services

throttling/rate-limiting technique

timeout

Retry pattern

Return on investment (ROI)

RSocket/reactive streams

Runtime confinement principle (RCP)

Runtime patterns

**S**

Saga pattern

ACID properties

choreography approach

compensatory transaction

definition

microservices transaction

orchestration

sequence steps

Security by design principle (SBDP)

designing and developing application

meaning

OWASP security risks

SQL injection

XSS prevention sheet

Self-containment principle (SCP)

Self-healing service

Separation of concern (SoC)

Sequential cohesion

Server evolution

bare-metal/physical server

containers

virtualization

Serverless architecture

advantages

asynchronous/multimedia processing

BaaS architecture

business logic process

characteristics

computing process

data transformation function

definition

design principles

backing services

cold-start time

concurrency

push-based/event-driven pipelines

stateless function

store config

disadvantages

essential components

event-driven approach

evolution

FaaS events

function deployment pipeline

independent reusable function

monolith/microservices

notification functions

operational attributes

operations management

parallel computing

production environment

technical solutions

virtualization technologies

Service level agreements (SLAs)

Service level objectives (SLOs)

Service-oriented architecture (SOA)

Service registry

Services

Service versioning

Sharding

hash technique

horizontal pattern

range partition

Shift-left principle (SLP)

DevOps

security

shifting performance testing

Sidecar pattern

Simple event processing (SEP)

Single concern principle (SCP)

Single responsibility principle

Single source of truth principle (SSOTP)

Site reliability engineering (SRE)

Software as a Service (SaaS)

architecture

custom *vs.* platform

definition

indicators

limitations

software-defined network (SDN)

Software engineering

Agile methodologies

*See*Agile methodologies

behavior/feature-driven development

capabilities

hypothesis-driven development

intelligent engineering

methodology

organizations transform

project/product mindset

test-driven development

traditional application engineering

Software quality principles

architecture

complex/unmanageable system

definition

DRY principle

information hiding

isolation

KISS principle

separation of concern

use layering principle

YAGNI principle

SOLID principles

dependency inversion

interface segregation

Liskov substitution

open-closed principle

single responsibility principle

software design

Stamp coupling

Static code analysis (SAST)

Static security system testing (SAST)

Strangulation patterns

Streaming platforms

Stream processing pipeline

Subject matter experts (SMEs)

Sustainability approaches

**T**

Technical debt

accumulation

decision-making

decoupling model

definition

enterprise

quadrant

Temporal cohesion

Test-drive development (TDD)

benefits

concepts

cycle

disadvantages

factors

higher-quality software

primary goal

steps

Third-Generation Partnership Project (3GPP)

Trade finance project

definition

ecosystem

functional components/architecture

implementation

letter of credit flow

microservice architecture

Traditional architecture layering approach

Transaction management

ACID

cloud-native service

event store

monolithic system

polyglot persistence

sequence diagram

transactions

two-phase commit (2PC)

Transaction security

Transport Layer Security (TLS)

Trends (maturity and industry)

actor model

architecture/design

concepts

Dapr architecture

digital twin platform

edge computing

event-driven architecture

extended reality (XR)

fifth-generation technology

gateways streamline

GitOps

HTTP/3

-ilities

Kubernetes

low-code tools

OAM specification

quantum computing

RSocket/reactive streams

Web Assembly (WASM)

**U**

Unified Modeling Language (UML)

URI versioning

Use layering (UL) principle

User interface

communication

composition

front-end monolith

implementation

integration

micro front ends

pros/cons

routing technique

**V**

Virtualization

applications/services

benefits

concepts

definition

hypervisor controls

technology

virtual machine

working process

Virtual machines (VM)/virtualized servers

adoption

architecture

benefits

cloud platform

container comparison

disadvantages

Virtual network functions (VNFs)

Virtual private cloud (VPC)

VMware technologies

**W, X**

Web Accessibility Initiative (WAI)

Web Assembly (WASM)

**Y**

You Aren’t Gonna Need It YAGNI

**Z**

Zero trust architecture (ZTA)

Zero-trust model