Admissions & Support: support@mockattempt.com
Phone / WhatsApp: +91 98765 43210
Course Code: SREM • **Updated:** 02 Nov 2026
https://mockattempt.com/training/cloud-devops-kubernetes-sre-master-program

SREM • CLOUD & DEVOPS ENGINEERING **BEGINNER TO EXPERT • JOB-GUARANTEED***

100% PLACEMENT GUARANTEED* CAREER PROGRAM (6-MO REFUND*)

Cloud DevOps, Kubernetes & Site Reliability Engineering (SRE) Master Program

Complete Zero-to-Expert OS, Linux Admin, Networking, Nginx, Bash & Python From Scratch to AWS Cloud, Docker, Kubernetes (EKS), Terraform, ArgoCD, DevSecOps & SRE | 100% Placement Guaranteed* (6 Months Fees Refund*)

Duration: 28w (420h) • Weekend Only (2h/Day) (448 Hrs) • **Modules:** 22 • **Live Topics:** 86 • **Subtopics:** 688+ • **Labs & Tasks:** 172 • **Mocks / AI Vivas:** 172 / 172 • **Projects:** 22 Module + 1 Live Client Prod

Fixed Non-Negotiable Tuition Fee: Base Fee **₹49,999** + 18% GST (₹9,000) = **Total Incl. GST: ₹58,999** • **Effective ROI:** ~₹112/Guided Hr (~₹581/Live Topic)

Faculty: 100% Active Working IT Professionals • **Weekend-Only Live Batch (7:00 AM – 8:00 PM IST Flexible Slots):** Weekend-Only Career Live Batch (Sat – Sun • 2 Hrs/Day • 7:00 AM – 8:00 PM) (Saturday & Sunday Only (2 Hours Live Session / Day) • 07:00 AM – 08:00 PM IST (Students Can Choose Any 2-Hour Weekend Slot Based on Working Professional Trainer Availability) • 28 Weeks (448 Guided Live + Lab Hours))

✓ **Trainer LeetCode & HackerRank Mentorship (Python & SRE Automation Scripting): Live trainer support to crack LeetCode & HackerRank problems, optimize complexity, and handle edge cases.**

✓ **6-Month Placement Guarantee* + Online AI Assessment (OA) Support: Complete 100% assigned repositories & score ≥90% across all Topic Mock Tests & Voice AI Interviews → 100+ verified IT job openings within 6 months with active Senior Trainer coaching to crack employer Online AI Assessments (OA), or 100% net tuition refund*.**

Core Technologies & Engineering Stack: Linux, Networking & Bash/Python • AWS Cloud & Hardened Docker • Kubernetes (EKS), Helm & ArgoCD • Terraform, DevSecOps & SRE • Nginx / HAProxy & TLS Security • GitHub Actions & Jenkins CI/CD • Ansible Configuration Management • Trivy, SBOM, HashiCorp Vault & Falco • Prometheus, Grafana & Loki • OpenTelemetry Distributed Tracing • SLIs, SLOs, k6 Load & Chaos Mesh • Platform Engineering & System Design

1. PROGRAM MASTERY OUTCOMES & ELIGIBILITY

End-to-End Engineering Outcomes:

1. Start from Day-1 zero coding experience and master every foundation, framework, database, cloud, and system design layer up to expert level
2. Build and ship 22 real-world production GitHub repositories + 1 Live Client Production Cloud Deployment (zero dummy to-do apps)
3. Achieve ≥90% mastery across every Topic Mock Test and Spoken MockAttempt AI Technical Interview (unlimited free re-attempts allowed below 90%)
4. Earn Dual Barcode-Verifiable Credentials: Course Excellence Certificate + Official Project Internship Confirmation Letter
5. Unlock 6-Month Placement Support guaranteeing 100 curated relevant IT job openings — backed by our 100% Tuition Fee Refund* policy
6. Crack LeetCode and HackerRank problems in Python & SRE Automation Scripting with live trainer mentorship on algorithmic/query patterns, edge cases, and optimal complexity.
7. Receive active Senior Trainer coaching during your 6-month (180-day) Placement Tenure to crack employer Online AI Assessments (OA), timed coding rounds, and automated screening tests.
8. Deep Linux kernel/networking troubleshooting, Bash/Python SRE automation, and hardened multi-stage container builds.

Eligibility & Prerequisites: Zero prior programming or IT experience required — Module 1 starts from Day-1 setup, logic building, and plain-English fundamentals • Basic computer literacy and commitment to attend Weekend-Only Live Sessions (Saturday & Sunday, 2 Hours/Day between 7:00 AM – 8:00 PM IST taught by active Working IT Professionals) plus guided weekday real-repository sprints • Dedication to complete 100% of assigned tasks and re-attempt assessments until crossing ≥90% mastery • Ideal for B.Tech / BCA / MCA / B.Sc IT students, non-CS beginners, career switchers, and working developers upgrading to 2026 production standards. • Dedication to write original code in your GitHub portfolio and complete the mentor-guided Live Client Production Sprint.

2. DETAILED ZERO-TO-EXPERT MODULE, TOPIC & SUBTOPIC SYLLABUS (22 MODULES • 86 LIVE TOPICS • 688+ SUBTOPICS)

Progressive Zero-to-Expert Mastery Stages: Modules 1–5: Stage 1: Day-1 Foundations & Core Mental Models (20 Topics / 160 Subtopics) • Modules 6–12: Stage 2: Applied Engineering, Data & API Workflows (28 Topics / 224 Subtopics) • Modules 13–17: Stage 3: Internals, Concurrency, Security & Performance (20 Topics / 160 Subtopics) • Modules 18–22: Stage 4: Cloud Scale, System Design & ≥90% Placement Gate (18 Topics / 144 Subtopics)

MODULE 1 OF 22 • STAGE 1 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Computing, OS Internals & Linux Terminal Mastery From Absolute Scratch

Beginner-to-expert real-repository module covering Computing, OS Internals & Linux Terminal Mastery From Absolute Scratch with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Computer Architecture & OS Fundamentals: User Space vs Kernel Space, Syscalls, CPU Scheduling & Virtual RAM — Purpose: Understand what actually happens inside an operating system when applications allocate memory or open files. • ✓ Linux Distributions (Ubuntu, Debian, RHEL/AL2023), Shell Navigation & Complete File/Directory CLI Mastery — Purpose: Move effortlessly across production Linux servers from Day 1, even with zero prior Linux experience. • ✓ Streams, Pipes, Redirection, Regex & High-Speed Log Parsing with grep, awk, sed, cut, sort, uniq & jq — Purpose: Extract error rates, top offending IPs, and slow endpoints from gigabyte log files in one pipeline. • ✓ Users, Groups, File Permissions, SUID/SGID Bits, ACLs, sudo Policies & Package Management (apt / dnf) — Purpose: Configure multi-user server environments cleanly with strict least-privilege file access.

1.1 Computer Architecture & OS Fundamentals: User Space vs Kernel Space, Syscalls, CPU Scheduling & Virtual RAM [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for C... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Computer Architecture • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Computer Architecture & OS Fundamentals: User Space vs Kernel Space, Syscalls, CPU Scheduling & Virtual RAM in real Cloud DevOps, Kubernetes & Sit...

What Topic Is Next → Next Topic 1.2: Linux Distributions (Ubuntu, Debian, RHEL/AL2023), Shell Navigation & Complete File/Directory CLI Mastery

1.2 Linux Distributions (Ubuntu, Debian, RHEL/AL2023), Shell Navigation & Complete File/Directory CLI Mastery [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for L... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Linux Distributions (Ubuntu, Debian, RHEL/AL2023) • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Linux Distributions (Ubuntu, Debian, RHEL/AL2023), Shell Navigation & Complete File/Directory CLI Mastery in real Cloud DevOps, Kubernetes & Site...

What Topic Is Next → Next Topic 1.3: Streams, Pipes, Redirection, Regex & High-Speed Log Parsing with grep, awk, sed, cut, sort, uniq & jq

1.3 Streams, Pipes, Redirection, Regex & High-Speed Log Parsing with grep, awk, sed, cut, sort, uniq & jq [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for S... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Streams • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Streams, Pipes, Redirection, Regex & High-Speed Log Parsing with grep, awk, sed, cut, sort, uniq & jq in real Cloud DevOps, Kubernetes & Site Reli...

What Topic Is Next → Next Topic 1.4: Users, Groups, File Permissions, SUID/SGID Bits, ACLs, sudo Policies & Package Management (apt / dnf)

1.4 Users, Groups, File Permissions, SUID/SGID Bits, ACLs, sudo Policies & Package Management (apt / dnf) [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for U... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Users • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Users, Groups, File Permissions, SUID/SGID Bits, ACLs, sudo Policies & Package Management (apt / dnf) in real Cloud DevOps, Kubernetes & Site Reli...

What Topic Is Next → Module 1 Capstone (kubepulse-m01-os-linux-cli-foundations) → Next Topic 2.1: Process Tree, Signals, zombie vs orphan Processes, /proc Inspection & Load Average Diagnostics (top/htop/pidstat)

Module 1 Hands-On Capstone Project (100 Marks): kubepulse-m01-os-linux-cli-foundations — Architect, implement, test, and push the production GitHub repository 'kubepulse-m01-os-linux-cli-foundations' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 1:...

MODULE 2 OF 22 • STAGE 1 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Enterprise Linux System Administration: Processes, systemd, Storage & cgroups

Beginner-to-expert real-repository module covering Enterprise Linux System Administration: Processes, systemd, Storage & cgroups with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Process Tree, Signals, zombie vs orphan Processes, /proc Inspection & Load Average Diagnostics (top/htop/pidstat) — Purpose: Pinpoint whether a slow Linux host is bottlenecked on CPU run-queues, RAM paging, or disk I/O wait. • ✓ systemd Service Units, Restart Policies, Resource Slices, Timers & Structured journalctl Forensics — Purpose: Run background daemons reliably with memory/CPU caps and instant crash recovery. • ✓ Block Devices, Partitioning, LVM Logical Volumes, XFS/ext4 Filesystems, Inodes & Disk Full Emergency Recovery — Purpose: Resolve 100%-disk-full and deleted-open-file (lsof) production incidents without server downtime. • ✓ Linux Namespaces (pid, net, mnt, uts, user) & Control Groups (cgroups v2) — How Containers Actually Work — Purpose: Demystify Docker and Kubernetes forever by understanding the native Linux kernel primitives underneath them.

2.1 Process Tree, Signals, zombie vs orphan Processes, /proc Inspection & Load Average Diagnostics (top/htop/pidstat) [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for P... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Process Tree • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Process Tree, Signals, zombie vs orphan Processes, /proc Inspection & Load Average Diagnostics (top/htop/pidstat) in real Cloud DevOps, Kubernetes...

What Topic Is Next → Next Topic 2.2: systemd Service Units, Restart Policies, Resource Slices, Timers & Structured journalctl Forensics

2.2 systemd Service Units, Restart Policies, Resource Slices, Timers & Structured journalctl Forensics [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for s... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for systemd Service Units • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply systemd Service Units, Restart Policies, Resource Slices, Timers & Structured journalctl Forensics in real Cloud DevOps, Kubernetes & Site Reliabi...

What Topic Is Next → Next Topic 2.3: Block Devices, Partitioning, LVM Logical Volumes, XFS/ext4 Filesystems, Inodes & Disk Full Emergency Recovery

2.3 Block Devices, Partitioning, LVM Logical Volumes, XFS/ext4 Filesystems, Inodes & Disk Full Emergency Recovery [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for B... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Block Devices • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Block Devices, Partitioning, LVM Logical Volumes, XFS/ext4 Filesystems, Inodes & Disk Full Emergency Recovery in real Cloud DevOps, Kubernetes & S...

What Topic Is Next → Next Topic 2.4: Linux Namespaces (pid, net, mnt, uts, user) & Control Groups (cgroups v2) — How Containers Actually Work

2.4 Linux Namespaces (pid, net, mnt, uts, user) & Control Groups (cgroups v2) — How Containers Actually Work [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for L... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Linux Namespaces (pid, net, mnt, uts, user) • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Linux Namespaces (pid, net, mnt, uts, user) & Control Groups (cgroups v2) — How Containers Actually Work in real Cloud DevOps, Kubernetes & Site R...

What Topic Is Next → Module 2 Capstone (kubepulse-m02-linux-systemd-cgroups) → Next Topic 3.1: SSH Public-Key Cryptography, SSH Agent Forwarding Risks, Bastion Jump Hosts & Hardened sshd_config

Module 2 Hands-On Capstone Project (100 Marks): kubepulse-m02-linux-systemd-cgroups — Architect, implement, test, and push the production GitHub repository 'kubepulse-m02-linux-systemd-cgroups' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 2: En...

MODULE 3 OF 22 • STAGE 1 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Linux Security Hardening, SSH Bastions, Host Firewalls & Live Troubleshooting

Beginner-to-expert real-repository module covering Linux Security Hardening, SSH Bastions, Host Firewalls & Live Troubleshooting with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ SSH Public-Key Cryptography, SSH Agent Forwarding Risks, Bastion Jump Hosts & Hardened sshd_config — Purpose: Eliminate brute-force SSH attacks and enforce audited key/certificate access to cloud nodes. • ✓ Linux Host Firewalls (ufw, iptables, nftables), Fail2ban & Mandatory Access Control (AppArmor / SELinux) — Purpose: Contain compromised processes at the kernel boundary before lateral movement can occur. • ✓ Kernel Tuning via sysctl (File Descriptors, Ephemeral Ports, TCP Backlog) for High-Concurrency Servers — Purpose: Tune Linux nodes to sustain 100,000+ concurrent sockets without "Too many open files" crashes. • ✓ Live Incident Triage with strace, lsof, vmstat, iostat & dmesg Out-Of-Memory (OOM-Killer) Analysis — Purpose: Diagnose mysterious process hangs and kernel OOM kills like a seasoned Site Reliability Engineer.

3.1 SSH Public-Key Cryptography, SSH Agent Forwarding Risks, Bastion Jump Hosts & Hardened sshd_config [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for S... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines,

canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for SSH Public-Key Cryptography • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply SSH Public-Key Cryptography, SSH Agent Forwarding Risks, Bastion Jump Hosts & Hardened sshd_config in real Cloud DevOps, Kubernetes & Site Reliabi...

What Topic Is Next → Next Topic 3.2: Linux Host Firewalls (ufw, iptables, nftables), Fail2ban & Mandatory Access Control (AppArmor / SELinux)

3.2 Linux Host Firewalls (ufw, iptables, nftables), Fail2ban & Mandatory Access Control (AppArmor / SELinux) [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for L... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Linux Host Firewalls (ufw, iptables, nftables) • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Linux Host Firewalls (ufw, iptables, nftables), Fail2ban & Mandatory Access Control (AppArmor / SELinux) in real Cloud DevOps, Kubernetes & Site R...

What Topic Is Next → Next Topic 3.3: Kernel Tuning via sysctl (File Descriptors, Ephemeral Ports, TCP Backlog) for High-Concurrency Servers

3.3 Kernel Tuning via sysctl (File Descriptors, Ephemeral Ports, TCP Backlog) for High-Concurrency Servers [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for K... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Kernel Tuning via sysctl (File Descriptors, Ephemera... • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Kernel Tuning via sysctl (File Descriptors, Ephemeral Ports, TCP Backlog) for High-Concurrency Servers in real Cloud DevOps, Kubernetes & Site Rel...

What Topic Is Next → Next Topic 3.4: Live Incident Triage with strace, lsof, vmstat, iostat & dmesg Out-Of-Memory (OOM-Killer) Analysis

3.4 Live Incident Triage with strace, lsof, vmstat, iostat & dmesg Out-Of-Memory (OOM-Killer) Analysis [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for L... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Live Incident Triage with strace • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Live Incident Triage with strace, lsof, vmstat, iostat & dmesg Out-Of-Memory (OOM-Killer) Analysis in real Cloud DevOps, Kubernetes & Site Reliabi...

What Topic Is Next → Module 3 Capstone (kubepulse-m03-linux-security-hardening) → Next Topic 4.1: OSI & TCP/IP Layers, IPv4/IPv6 CIDR Subnetting, Routing Tables, NAT & Private RFC1918 Address Planning

Module 3 Hands-On Capstone Project (100 Marks): kubepulse-m03-linux-security-hardening — Architect, implement, test, and push the production GitHub repository 'kubepulse-m03-linux-security-hardening' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 3:...

MODULE 4 OF 22 • STAGE 1 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Computer Networking From Scratch: TCP/IP, DNS, TLS, Subnets & Packet Forensics

Beginner-to-expert real-repository module covering Computer Networking From Scratch: TCP/IP, DNS, TLS, Subnets & Packet Forensics with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ OSI & TCP/IP Layers, IPv4/IPv6 CIDR Subnetting, Routing Tables, NAT & Private RFC1918 Address Planning — Purpose: Calculate cloud and Kubernetes pod/service CIDR ranges accurately without IP exhaustion overlaps. • ✓ TCP Handshakes, Retransmissions, Keepalives, TIME_WAIT, UDP & Socket State Inspection (ss / netstat) — Purpose: Debug intermittent connection resets, latency spikes, and exhausted ephemeral port pools. • ✓ DNS Resolution Internals, Authoritative vs Recursive Resolvers, CoreDNS & dig / nslookup Troubleshooting — Purpose: Fix the #1 root cause of cloud/Kubernetes outages — broken DNS resolution and stale TTLs. • ✓ TLS 1.3 Handshakes, X.509 Certificate Chains, mTLS, Let’s Encrypt ACME & Packet Capture with tcpdump — Purpose: Inspect live wire traffic and prevent midnight production outages caused by expired TLS certificates.

4.1 OSI & TCP/IP Layers, IPv4/IPv6 CIDR Subnetting, Routing Tables, NAT & Private RFC1918 Address Planning [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for T... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for TCP/IP Layers • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply OSI & TCP/IP Layers, IPv4/IPv6 CIDR Subnetting, Routing Tables, NAT & Private RFC1918 Address Planning in real Cloud DevOps, Kubernetes & Site Rel...

What Topic Is Next → Next Topic 4.2: TCP Handshakes, Retransmissions, Keepalives, TIME_WAIT, UDP & Socket State Inspection (ss / netstat)

4.2 TCP Handshakes, Retransmissions, Keepalives, TIME_WAIT, UDP & Socket State Inspection (ss / netstat) [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for T... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for TCP Handshakes • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply TCP Handshakes, Retransmissions, Keepalives, TIME_WAIT, UDP & Socket State Inspection (ss / netstat) in real Cloud DevOps, Kubernetes & Site Relia...

What Topic Is Next → Next Topic 4.3: DNS Resolution Internals, Authoritative vs Recursive Resolvers, CoreDNS & dig / nslookup Troubleshooting

4.3 DNS Resolution Internals, Authoritative vs Recursive Resolvers, CoreDNS & dig / nslookup Troubleshooting [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for D... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for DNS Resolution Internals • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply DNS Resolution Internals, Authoritative vs Recursive Resolvers, CoreDNS & dig / nslookup Troubleshooting in real Cloud DevOps, Kubernetes & Site R...

What Topic Is Next → Next Topic 4.4: TLS 1.3 Handshakes, X.509 Certificate Chains, mTLS, Let's Encrypt ACME & Packet Capture with tcpdump

4.4 TLS 1.3 Handshakes, X.509 Certificate Chains, mTLS, Let's Encrypt ACME & Packet Capture with tcpdump [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for T... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for TLS 1.3 Handshakes • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply TLS 1.3 Handshakes, X.509 Certificate Chains, mTLS, Let's Encrypt ACME & Packet Capture with tcpdump in real Cloud DevOps, Kubernetes & Site Relia...

What Topic Is Next → Module 4 Capstone (kubepulse-m04-networking-dns-tls-packets) → Next Topic 5.1: HTTP/1.1, HTTP/2, HTTP/3 (QUIC), Status Codes, Caching Headers, CORS & Keep-Alive Connection Reuse

Module 4 Hands-On Capstone Project (100 Marks): kubepulse-m04-networking-dns-tls-packets — Architect, implement, test, and push the production GitHub repository 'kubepulse-m04-networking-dns-tls-packets' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module...

MODULE 5 OF 22 • STAGE 1 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Web Servers, Reverse Proxies & Load Balancers: Nginx, HAProxy & Traffic Routing

Beginner-to-expert real-repository module covering Web Servers, Reverse Proxies & Load Balancers: Nginx, HAProxy & Traffic Routing with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ HTTP/1.1, HTTP/2, HTTP/3 (QUIC), Status Codes, Caching Headers, CORS & Keep-Alive Connection Reuse — Purpose: Master web protocol semantics essential for debugging API gateways, CDNs, and ingress controllers. • ✓ Nginx Reverse Proxy Architecture: Worker Event Loop, Virtual Hosts, TLS Termination, Gzip/Brotli & Buffering — Purpose: Configure high-throughput Nginx edge proxies in front of Node.js, Python, Go, and Java backends. • ✓ Layer-4 vs Layer-7 Load Balancing with HAProxy & Nginx: Health Checks, Connection Draining & Rate Limiting — Purpose: Shed abusive traffic bursts at the proxy edge and drain backend servers gracefully during deploys. • ✓ Zero-Downtime Reloads, Access/Error Log Correlation & Debugging 502 / 503 / 504 Gateway Errors — Purpose: Instantly distinguish upstream crash (502), overload (503), and slow upstream timeout (504) incidents.

5.1 HTTP/1.1, HTTP/2, HTTP/3 (QUIC), Status Codes, Caching Headers, CORS & Keep-Alive Connection Reuse [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for H... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for HTTP/1.1 • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply HTTP/1.1, HTTP/2, HTTP/3 (QUIC), Status Codes, Caching Headers, CORS & Keep-Alive Connection Reuse in real Cloud DevOps, Kubernetes & Site Reliabi...

What Topic Is Next → Next Topic 5.2: Nginx Reverse Proxy Architecture: Worker Event Loop, Virtual Hosts, TLS Termination, Gzip/Brotli & Buffering

5.2 Nginx Reverse Proxy Architecture: Worker Event Loop, Virtual Hosts, TLS Termination, Gzip/Brotli & Buffering [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for N... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Nginx Reverse Proxy Architecture • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Nginx Reverse Proxy Architecture: Worker Event Loop, Virtual Hosts, TLS Termination, Gzip/Brotli & Buffering in real Cloud DevOps, Kubernetes & Si...

What Topic Is Next → Next Topic 5.3: Layer-4 vs Layer-7 Load Balancing with HAProxy & Nginx: Health Checks, Connection Draining & Rate Limiting

5.3 Layer-4 vs Layer-7 Load Balancing with HAProxy & Nginx: Health Checks, Connection Draining & Rate Limiting [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for L... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Layer-4 vs Layer-7 Load Balancing with HAProxy • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Layer-4 vs Layer-7 Load Balancing with HAProxy & Nginx: Health Checks, Connection Draining & Rate Limiting in real Cloud DevOps, Kubernetes & Site...

What Topic Is Next → Next Topic 5.4: Zero-Downtime Reloads, Access/Error Log Correlation & Debugging 502 / 503 / 504 Gateway Errors

5.4 Zero-Downtime Reloads, Access/Error Log Correlation & Debugging 502 / 503 / 504 Gateway Errors [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Z... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Zero-Downtime Reloads • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Zero-Downtime Reloads, Access/Error Log Correlation & Debugging 502 / 503 / 504 Gateway Errors in real Cloud DevOps, Kubernetes & Site Reliability...

What Topic Is Next → Module 5 Capstone (kubepulse-m05-nginx-haproxy-proxies) → Next Topic 6.1: Git Object Model (.git Internals), Branching, Merging vs Rebasing, Cherry-Picking & git bisect Bug Hunting

Module 5 Hands-On Capstone Project (100 Marks): kubepulse-m05-nginx-haproxy-proxies — Architect, implement, test, and push the production GitHub repository 'kubepulse-m05-nginx-haproxy-proxies' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 5: We...

MODULE 6 OF 22 • STAGE 2 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Git Version Control Internals, Branching Workflows, JSON/YAML & Developer Tooling

Beginner-to-expert real-repository module covering Git Version Control Internals, Branching Workflows, JSON/YAML & Developer Tooling with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Git Object Model (.git Internals), Branching, Merging vs Rebasing, Cherry-Picking & git bisect Bug Hunting — Purpose: Recover cleanly from any Git state and pinpoint the exact commit that broke staging in minutes. • ✓ Trunk-Based Development vs GitFlow, Protected Branches, CODEOWNERS, Signed Commits & PR Review Standards — Purpose: Establish high-velocity engineering collaboration workflows across application and DevOps teams. • ✓ Deep YAML, JSON & TOML Fluency, jq / yq CLI Transformations & Schema Linting for Cloud/K8s Configs — Purpose: Manipulate complex Kubernetes, Helm, GitHub Actions, and Ansible YAML files without whitespace errors. • ✓ Makefiles / Taskfiles, Pre-Commit Secret Scanners (gitleaks) & Reproducible Developer Setup Scripts — Purpose: Stop AWS keys and private tokens from ever entering Git history in the first place.

6.1 Git Object Model (.git Internals), Branching, Merging vs Rebasing, Cherry-Picking & git bisect Bug Hunting [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for G... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Git Object Model (.git Internals) • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Git Object Model (.git Internals), Branching, Merging vs Rebasing, Cherry-Picking & git bisect Bug Hunting in real Cloud DevOps, Kubernetes & Site...

What Topic Is Next → Next Topic 6.2: Trunk-Based Development vs GitFlow, Protected Branches, CODEOWNERS, Signed Commits & PR Review Standards

6.2 Trunk-Based Development vs GitFlow, Protected Branches, CODEOWNERS, Signed Commits & PR Review Standards [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for T... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Trunk-Based Development vs GitFlow • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Trunk-Based Development vs GitFlow, Protected Branches, CODEOWNERS, Signed Commits & PR Review Standards in real Cloud DevOps, Kubernetes & Site R...

What Topic Is Next → Next Topic 6.3: Deep YAML, JSON & TOML Fluency, jq / yq CLI Transformations & Schema Linting for Cloud/K8s Configs

6.3 Deep YAML, JSON & TOML Fluency, jq / yq CLI Transformations & Schema Linting for Cloud/K8s Configs [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for D... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Deep YAML • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Deep YAML, JSON & TOML Fluency, jq / yq CLI Transformations & Schema Linting for Cloud/K8s Configs in real Cloud DevOps, Kubernetes & Site Reliabi...

What Topic Is Next → Next Topic 6.4: Makefiles / Taskfiles, Pre-Commit Secret Scanners (gitleaks) & Reproducible Developer Setup Scripts

6.4 Makefiles / Taskfiles, Pre-Commit Secret Scanners (gitleaks) & Reproducible Developer Setup Scripts [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for M... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Makefiles / Taskfiles • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Makefiles / Taskfiles, Pre-Commit Secret Scanners (gitleaks) & Reproducible Developer Setup Scripts in real Cloud DevOps, Kubernetes & Site Reliab...

What Topic Is Next → Module 6 Capstone (kubepulse-m06-git-yaml-workflows) → Next Topic 7.1: Bash Scripting Foundations: Variables, Quoting Rules, Arrays, Conditionals, Loops & Functions From Scratch

Module 6 Hands-On Capstone Project (100 Marks): kubepulse-m06-git-yaml-workflows — Architect, implement, test, and push the production GitHub repository 'kubepulse-m06-git-yaml-workflows' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 6: Git V...

MODULE 7 OF 22 • STAGE 2 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Bash Shell Scripting From Zero to Idempotent Production Ops Automation

Beginner-to-expert real-repository module covering Bash Shell Scripting From Zero to Idempotent Production Ops Automation with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Bash Scripting Foundations: Variables, Quoting Rules, Arrays, Conditionals, Loops & Functions From Scratch — Purpose: Write clean, portable shell scripts free from word-splitting and unquoted-path bugs. • ✓ Defensive Bash Engineering: Strict Mode (set -euo pipefail), Signal Traps, Lockfiles (flock) & Temp Cleanup — Purpose: Ensure cron scripts never overlap concurrently or leave half-written corrupted files on failure. • ✓ CLI Argument Parsing (getopts), REST API Calls with curl + jq, Retry Backoff Loops & Slack/Webhook Alerts — Purpose: Automate health probes, deployment smoke checks, and operational notifications reliably. • ✓ Testing Shell Scripts with ShellCheck + Bats & Automating Log Archival, DB Backups & Cert Checks — Purpose: Ship linted, idempotent ops scripts trusted to run unattended on production fleets.

7.1 Bash Scripting Foundations: Variables, Quoting Rules, Arrays, Conditionals, Loops & Functions From Scratch [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for B... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Bash Scripting Foundations • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Bash Scripting Foundations: Variables, Quoting Rules, Arrays, Conditionals, Loops & Functions From Scratch in real Cloud DevOps, Kubernetes & Site...

What Topic Is Next → Next Topic 7.2: Defensive Bash Engineering: Strict Mode (set -euo pipefail), Signal Traps, Lockfiles (flock) & Temp Cleanup

7.2 Defensive Bash Engineering: Strict Mode (set -euo pipefail), Signal Traps, Lockfiles (flock) & Temp Cleanup [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for D... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Defensive Bash Engineering • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Defensive Bash Engineering: Strict Mode (set -euo pipefail), Signal Traps, Lockfiles (flock) & Temp Cleanup in real Cloud DevOps, Kubernetes & Sit...

What Topic Is Next → Next Topic 7.3: CLI Argument Parsing (getopts), REST API Calls with curl + jq, Retry Backoff Loops & Slack/Webhook Alerts

7.3 CLI Argument Parsing (getopts), REST API Calls with curl + jq, Retry Backoff Loops & Slack/Webhook Alerts [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for C... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for CLI Argument Parsing (getopts) • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply CLI Argument Parsing (getopts), REST API Calls with curl + jq, Retry Backoff Loops & Slack/Webhook Alerts in real Cloud DevOps, Kubernetes & Site...

What Topic Is Next → Next Topic 7.4: Testing Shell Scripts with ShellCheck + Bats & Automating Log Archival, DB Backups & Cert Checks

7.4 Testing Shell Scripts with ShellCheck + Bats & Automating Log Archival, DB Backups & Cert Checks [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for T... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Testing Shell Scripts with ShellCheck + Bats • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Testing Shell Scripts with ShellCheck + Bats & Automating Log Archival, DB Backups & Cert Checks in real Cloud DevOps, Kubernetes & Site Reliabili...

What Topic Is Next → Module 7 Capstone (kubepulse-m07-bash-sre-automation) → Next Topic 8.1: Python 3 From Zero: Syntax, Lists, Dicts, Sets, Loops, Functions, Modules, venv & Exception Handling

Module 7 Hands-On Capstone Project (100 Marks): kubepulse-m07-bash-sre-automation — Architect, implement, test, and push the production GitHub repository 'kubepulse-m07-bash-sre-automation' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 7: Bash...

MODULE 8 OF 22 • STAGE 2 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Python 3 Programming From Scratch for DevOps, CLI Tools, APIs & AWS Boto3

Beginner-to-expert real-repository module covering Python 3 Programming From Scratch for DevOps, CLI Tools, APIs & AWS Boto3 with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Python 3 From Zero: Syntax, Lists, Dicts, Sets, Loops, Functions, Modules, venv & Exception Handling — Purpose: Gain complete Python programming fluency starting from basics — tailored specifically for DevOps & SREs. • ✓ File I/O, JSON/YAML/CSV Parsing, subprocess Execution, Regular Expressions & Structured Logging — Purpose: Replace fragile 500-line Bash scripts with readable, maintainable, unit-tested Python automation. • ✓ Building DevOps CLI Tools (Click / Typer), Consuming REST APIs (httpx) & Exposing FastAPI Webhook Bots — Purpose: Create internal developer CLIs and ChatOps automation services used across engineering teams. • ✓ Cloud Automation with AWS Boto3 SDK: IAM Audits, Orphaned EBS/Snapshot Cleanup, EC2/EKS Tagging & Pytest — Purpose: Automate FinOps cost cleanup and security compliance checks across AWS accounts with tested Python code.

8.1 Python 3 From Zero: Syntax, Lists, Dicts, Sets, Loops, Functions, Modules, venv & Exception Handling [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for P... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Python 3 From Zero • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Python 3 From Zero: Syntax, Lists, Dicts, Sets, Loops, Functions, Modules, venv & Exception Handling in real Cloud DevOps, Kubernetes & Site Relia...

What Topic Is Next → Next Topic 8.2: File I/O, JSON/YAML/CSV Parsing, subprocess Execution, Regular Expressions & Structured Logging

8.2 File I/O, JSON/YAML/CSV Parsing, subprocess Execution, Regular Expressions & Structured Logging [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for F... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for File I/O • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply File I/O, JSON/YAML/CSV Parsing, subprocess Execution, Regular Expressions & Structured Logging in real Cloud DevOps, Kubernetes & Site Reliabilit...

What Topic Is Next → Next Topic 8.3: Building DevOps CLI Tools (Click / Typer), Consuming REST APIs (httpx) & Exposing FastAPI Webhook Bots

8.3 Building DevOps CLI Tools (Click / Typer), Consuming REST APIs (httpx) & Exposing FastAPI Webhook Bots [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for B... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Building DevOps CLI Tools (Click / Typer) • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Building DevOps CLI Tools (Click / Typer), Consuming REST APIs (httpx) & Exposing FastAPI Webhook Bots in real Cloud DevOps, Kubernetes & Site Rel...

What Topic Is Next → Next Topic 8.4: Cloud Automation with AWS Boto3 SDK: IAM Audits, Orphaned EBS/Snapshot Cleanup, EC2/EKS Tagging & Pytest

8.4 Cloud Automation with AWS Boto3 SDK: IAM Audits, Orphaned EBS/Snapshot Cleanup, EC2/EKS Tagging & Pytest [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for C... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Cloud Automation with AWS Boto3 SDK • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Cloud Automation with AWS Boto3 SDK: IAM Audits, Orphaned EBS/Snapshot Cleanup, EC2/EKS Tagging & Pytest in real Cloud DevOps, Kubernetes & Site R...

What Topic Is Next → Module 8 Capstone (kubepulse-m08-python-devops-boto3) → Next Topic 9.1: AWS Foundations & Zero-Trust IAM: Roles, Policies, STS AssumeRole, OIDC Federation, KMS & Secrets Manager

Module 8 Hands-On Capstone Project (100 Marks): kubepulse-m08-python-devops-boto3 — Architect, implement, test, and push the production GitHub repository 'kubepulse-m08-python-devops-boto3' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 8: Pyth...

MODULE 9 OF 22 • STAGE 2 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

AWS Cloud Infrastructure From Scratch: IAM, Multi-AZ VPC, EC2, ALB, S3 & RDS

Beginner-to-expert real-repository module covering AWS Cloud Infrastructure From Scratch: IAM, Multi-AZ VPC, EC2, ALB, S3 & RDS with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ AWS Foundations & Zero-Trust IAM: Roles, Policies, STS AssumeRole, OIDC Federation, KMS & Secrets Manager — Purpose: Configure secure multi-environment AWS foundations with zero static access keys. • ✓ Enterprise AWS Networking: Multi-AZ VPCs, Public/Private Subnets, NAT, Security Groups, VPC Endpoints & Peering — Purpose: Design isolated private cloud networks ready to host production Kubernetes and database clusters. • ✓ Compute & Edge: EC2, Auto Scaling Groups, Application/Network Load Balancers, Route 53 TLS DNS & CloudFront — Purpose: Route global HTTPS user traffic reliably into resilient Multi-AZ cloud compute fleets. • ✓ Managed Persistence & Queues: Amazon S3, RDS PostgreSQL Multi-AZ, ElastiCache Redis, ECR & SQS/SNS — Purpose: Operate managed AWS stateful services with automated backups, encryption, and failover readiness.

9.1 AWS Foundations & Zero-Trust IAM: Roles, Policies, STS AssumeRole, OIDC Federation, KMS & Secrets Manager [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for A... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines,

canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for AWS Foundations • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply AWS Foundations & Zero-Trust IAM: Roles, Policies, STS AssumeRole, OIDC Federation, KMS & Secrets Manager in real Cloud DevOps, Kubernetes & Site...

What Topic Is Next → Next Topic 9.2: Enterprise AWS Networking: Multi-AZ VPCs, Public/Private Subnets, NAT, Security Groups, VPC Endpoints & Peering

9.2 Enterprise AWS Networking: Multi-AZ VPCs, Public/Private Subnets, NAT, Security Groups, VPC Endpoints & Peering [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for E... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Enterprise AWS Networking • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Enterprise AWS Networking: Multi-AZ VPCs, Public/Private Subnets, NAT, Security Groups, VPC Endpoints & Peering in real Cloud DevOps, Kubernetes &...

What Topic Is Next → Next Topic 9.3: Compute & Edge: EC2, Auto Scaling Groups, Application/Network Load Balancers, Route 53 TLS DNS & CloudFront

9.3 Compute & Edge: EC2, Auto Scaling Groups, Application/Network Load Balancers, Route 53 TLS DNS & CloudFront [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for C... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Compute • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Compute & Edge: EC2, Auto Scaling Groups, Application/Network Load Balancers, Route 53 TLS DNS & CloudFront in real Cloud DevOps, Kubernetes & Sit...

What Topic Is Next → Next Topic 9.4: Managed Persistence & Queues: Amazon S3, RDS PostgreSQL Multi-AZ, ElastiCache Redis, ECR & SQS/SNS

9.4 Managed Persistence & Queues: Amazon S3, RDS PostgreSQL Multi-AZ, ElastiCache Redis, ECR & SQS/SNS [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for M... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Managed Persistence • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Managed Persistence & Queues: Amazon S3, RDS PostgreSQL Multi-AZ, ElastiCache Redis, ECR & SQS/SNS in real Cloud DevOps, Kubernetes & Site Reliabi...

What Topic Is Next → Module 9 Capstone (kubepulse-m09-aws-cloud-production) → Next Topic 10.1: How Modern Apps Are Built: Packaging Node.js/TypeScript, Python, Java (Maven/Gradle) & Go Services

Module 9 Hands-On Capstone Project (100 Marks): kubepulse-m09-aws-cloud-production — Architect, implement, test, and push the production GitHub repository 'kubepulse-m09-aws-cloud-production' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 9: AWS...

MODULE 10 OF 22 • STAGE 2 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Application Build Systems, Artifact Registries & Continuous Integration Foundations

Beginner-to-expert real-repository module covering Application Build Systems, Artifact Registries & Continuous Integration Foundations with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ How Modern Apps Are Built: Packaging Node.js/TypeScript, Python, Java (Maven/Gradle) & Go Services — Purpose: Understand every language ecosystem's build, dependency lockfile, and runtime artifact model as a DevOps Lead. • ✓ Twelve-Factor App Architecture: Env Config, Stateless Processes, Port Binding, Disposability & Dev/Prod Parity — Purpose: Guide developers to structure cloud-native microservices that scale and deploy cleanly. • ✓ Continuous Integration Principles: Fast Unit/Integration Suites, Dependency Caching, Matrix Builds & Linters — Purpose: Keep CI feedback loops under 5 minutes while catching regressions before merge. • ✓ Semantic Versioning (SemVer), Immutable Artifact Repositories (ECR / GHCR) & Build Provenance Metadata — Purpose: Guarantee every production deployment maps 1:1 to a traceable Git SHA and immutable artifact digest.

10.1 How Modern Apps Are Built: Packaging Node.js/TypeScript, Python, Java (Maven/Gradle) & Go Services [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for H... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for How Modern Apps Are Built • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply How Modern Apps Are Built: Packaging Node.js/TypeScript, Python, Java (Maven/Gradle) & Go Services in real Cloud DevOps, Kubernetes & Site Reliabi...

What Topic Is Next → Next Topic 10.2: Twelve-Factor App Architecture: Env Config, Stateless Processes, Port Binding, Disposability & Dev/Prod Parity

10.2 Twelve-Factor App Architecture: Env Config, Stateless Processes, Port Binding, Disposability & Dev/Prod Parity [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for T... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for

Twelve-Factor App Architecture • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Twelve-Factor App Architecture: Env Config, Stateless Processes, Port Binding, Disposability & Dev/Prod Parity in real Cloud DevOps, Kubernetes &...

What Topic Is Next → Next Topic 10.3: Continuous Integration Principles: Fast Unit/Integration Suites, Dependency Caching, Matrix Builds & Linters

10.3 Continuous Integration Principles: Fast Unit/Integration Suites, Dependency Caching, Matrix Builds & Linters [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for C... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Continuous Integration Principles • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Continuous Integration Principles: Fast Unit/Integration Suites, Dependency Caching, Matrix Builds & Linters in real Cloud DevOps, Kubernetes & Si...

What Topic Is Next → Next Topic 10.4: Semantic Versioning (SemVer), Immutable Artifact Repositories (ECR / GHCR) & Build Provenance Metadata

10.4 Semantic Versioning (SemVer), Immutable Artifact Repositories (ECR / GHCR) & Build Provenance Metadata [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for S... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Semantic Versioning (SemVer) • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Semantic Versioning (SemVer), Immutable Artifact Repositories (ECR / GHCR) & Build Provenance Metadata in real Cloud DevOps, Kubernetes & Site Rel...

What Topic Is Next → Module 10 Capstone (kubepulse-m10-build-systems-ci-core) → Next Topic 11.1: Docker Architecture From Scratch: OCI Images, UnionFS Layers, Containers, Registries & Essential CLI Workflows

Module 10 Hands-On Capstone Project (100 Marks): kubepulse-m10-build-systems-ci-core — Architect, implement, test, and push the production GitHub repository 'kubepulse-m10-build-systems-ci-core' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 10: A...

MODULE 11 OF 22 • STAGE 2 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Docker Containerization From First Container to Hardened Multi-Stage Images

Beginner-to-expert real-repository module covering Docker Containerization From First Container to Hardened Multi-Stage Images with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Docker Architecture From Scratch: OCI Images, UnionFS Layers, Containers, Registries & Essential CLI Workflows — Purpose: Build, tag, inspect, and debug containers from first principles with crystal-clear mental models. • ✓ Production Dockerfile Engineering: Layer Caching, .dockerignore & Multi-Stage Alpine / Distroless Builds — Purpose: Shrink bloated 1.2 GB images down to hardened 60 MB runtime images with minimal attack surface. • ✓ Container Security & Runtime Limits: Non-Root USER, Read-Only RootFS, Dropped Linux Capabilities & Memory/CPU Caps — Purpose: Prevent container breakout vulnerabilities and noisy-neighbor memory starvation. • ✓ Docker Networking (Bridge, Host, Overlay), Volumes & Multi-Service Local Topologies with Docker Compose — Purpose: Orchestrate APIs, databases, caches, and reverse proxies locally with healthcheck dependencies.

11.1 Docker Architecture From Scratch: OCI Images, UnionFS Layers, Containers, Registries & Essential CLI Workflows [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for D... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Docker Architecture From Scratch • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Docker Architecture From Scratch: OCI Images, UnionFS Layers, Containers, Registries & Essential CLI Workflows in real Cloud DevOps, Kubernetes &...

What Topic Is Next → Next Topic 11.2: Production Dockerfile Engineering: Layer Caching, .dockerignore & Multi-Stage Alpine / Distroless Builds

11.2 Production Dockerfile Engineering: Layer Caching, .dockerignore & Multi-Stage Alpine / Distroless Builds [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for P... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Production Dockerfile Engineering • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Production Dockerfile Engineering: Layer Caching, .dockerignore & Multi-Stage Alpine / Distroless Builds in real Cloud DevOps, Kubernetes & Site R...

What Topic Is Next → Next Topic 11.3: Container Security & Runtime Limits: Non-Root USER, Read-Only RootFS, Dropped Linux Capabilities & Memory/CPU Caps

11.3 Container Security & Runtime Limits: Non-Root USER, Read-Only RootFS, Dropped Linux Capabilities & Memory/CPU Caps [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for C... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Container Security • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Container Security & Runtime Limits: Non-Root USER, Read-Only RootFS, Dropped Linux Capabilities & Memory/CPU Caps in real Cloud DevOps, Kubernete...

What Topic Is Next → Next Topic 11.4: Docker Networking (Bridge, Host, Overlay), Volumes & Multi-Service Local Topologies with Docker Compose

11.4 Docker Networking (Bridge, Host, Overlay), Volumes & Multi-Service Local Topologies with Docker Compose [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for D... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Docker Networking (Bridge, Host, Overlay) • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Docker Networking (Bridge, Host, Overlay), Volumes & Multi-Service Local Topologies with Docker Compose in real Cloud DevOps, Kubernetes & Site Re...

What Topic Is Next → Module 11 Capstone (kubepulse-m11-docker-hardened-containers) → Next Topic 12.1: Kubernetes Control Plane (kube-apiserver, etcd, scheduler, controller-manager) & Node (kubelet, kube-proxy, CRI)

Module 11 Hands-On Capstone Project (100 Marks): kubepulse-m11-docker-hardened-containers — Architect, implement, test, and push the production GitHub repository 'kubepulse-m11-docker-hardened-containers' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module...

MODULE 12 OF 22 • STAGE 2 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Kubernetes From Scratch: Architecture, Pods, Deployments, Services & Namespaces

Beginner-to-expert real-repository module covering Kubernetes From Scratch: Architecture, Pods, Deployments, Services & Namespaces with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Kubernetes Control Plane (kube-apiserver, etcd, scheduler, controller-manager) & Node (kubelet, kube-proxy, CRI) — Purpose: Understand every moving part of a Kubernetes cluster so troubleshooting feels logical and fast. • ✓ Pods, Multi-Container Patterns (Init, Sidecar), Labels, Selectors, Annotations & Namespace Isolation — Purpose: Author clean Kubernetes Pod specs and organize multi-team environments with ResourceQuotas. • ✓ Declarative ReplicaSets, Deployments, RollingUpdate MaxSurge/MaxUnavailable, Rollbacks & DaemonSets / Jobs — Purpose: Roll out application updates with zero dropped traffic and schedule cluster-wide agents and batch jobs. • ✓ Kubernetes Networking Model: ClusterIP, NodePort, LoadBalancer Services, Headless DNS & CoreDNS Discovery — Purpose: Connect frontend, backend, and worker pods reliably across cluster nodes via Kubernetes service DNS.

12.1 Kubernetes Control Plane (kube-apiserver, etcd, scheduler, controller-manager) & Node (kubelet, kube-proxy, CRI) [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for K... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Kubernetes Control Plane (kube-apiserver, etcd, sche... • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Kubernetes Control Plane (kube-apiserver, etcd, scheduler, controller-manager) & Node (kubelet, kube-proxy, CRI) in real Cloud DevOps, Kubernetes...

What Topic Is Next → Next Topic 12.2: Pods, Multi-Container Patterns (Init, Sidecar), Labels, Selectors, Annotations & Namespace Isolation

12.2 Pods, Multi-Container Patterns (Init, Sidecar), Labels, Selectors, Annotations & Namespace Isolation [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for P... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Pods • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Pods, Multi-Container Patterns (Init, Sidecar), Labels, Selectors, Annotations & Namespace Isolation in real Cloud DevOps, Kubernetes & Site Relia...

What Topic Is Next → Next Topic 12.3: Declarative ReplicaSets, Deployments, RollingUpdate MaxSurge/MaxUnavailable, Rollbacks & DaemonSets / Jobs

12.3 Declarative ReplicaSets, Deployments, RollingUpdate MaxSurge/MaxUnavailable, Rollbacks & DaemonSets / Jobs [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for D... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Declarative ReplicaSets • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Declarative ReplicaSets, Deployments, RollingUpdate MaxSurge/MaxUnavailable, Rollbacks & DaemonSets / Jobs in real Cloud DevOps, Kubernetes & Site...

What Topic Is Next → Next Topic 12.4: Kubernetes Networking Model: ClusterIP, NodePort, LoadBalancer Services, Headless DNS & CoreDNS Discovery

12.4 Kubernetes Networking Model: ClusterIP, NodePort, LoadBalancer Services, Headless DNS & CoreDNS Discovery [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for K... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Kubernetes Networking Model • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Kubernetes Networking Model: ClusterIP, NodePort, LoadBalancer Services, Headless DNS & CoreDNS Discovery in real Cloud DevOps, Kubernetes & Site...

What Topic Is Next → Module 12 Capstone (kubepulse-m12-kubernetes-core-workloads) → Next Topic 13.1: Configuration & Secret Management: ConfigMaps, Kubernetes Secrets, External Secrets Operator & Env Injection

Module 12 Hands-On Capstone Project (100 Marks): kubepulse-m12-kubernetes-core-workloads — Architect, implement, test, and push the production GitHub repository 'kubepulse-m12-kubernetes-core-workloads' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 1...

MODULE 13 OF 22 • STAGE 3 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Stateful Kubernetes, Storage CSI, ConfigMaps / Secrets, Probes & Ingress TLS

Beginner-to-expert real-repository module covering Stateful Kubernetes, Storage CSI, ConfigMaps / Secrets, Probes & Ingress TLS with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Configuration & Secret Management: ConfigMaps, Kubernetes Secrets, External Secrets Operator & Env Injection — Purpose: Sync secrets securely from AWS Secrets Manager / Vault into pods without committing base64 secrets to Git. • ✓ Application Lifecycle Hooks: Startup, Readiness & Liveness Probes, preStop Hooks & Graceful SIGTERM Drain — Purpose: Eliminate 502 errors during rolling deployments by coordinating probe timing and load-balancer drain. • ✓ Persistent Storage in Kubernetes: StorageClasses, PersistentVolumeClaims (PVC), EBS/EFS CSI & StatefulSets — Purpose: Run stateful workloads (databases, Kafka, Redis) with stable network identities and durable volumes. • ✓ Ingress Controllers (NGINX / AWS ALB), Gateway API & Automated TLS Certificates with cert-manager — Purpose: Expose dozens of HTTPS microservices cleanly behind path/host routing and auto-renewed Let’s Encrypt/ACM certs.

13.1 Configuration & Secret Management: ConfigMaps, Kubernetes Secrets, External Secrets Operator & Env Injection

[90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for C... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Configuration • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Configuration & Secret Management: ConfigMaps, Kubernetes Secrets, External Secrets Operator & Env Injection in real Cloud DevOps, Kubernetes & Si...

What Topic Is Next → Next Topic 13.2: Application Lifecycle Hooks: Startup, Readiness & Liveness Probes, preStop Hooks & Graceful SIGTERM Drain

13.2 Application Lifecycle Hooks: Startup, Readiness & Liveness Probes, preStop Hooks & Graceful SIGTERM Drain

[90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for A... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Application Lifecycle Hooks • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Application Lifecycle Hooks: Startup, Readiness & Liveness Probes, preStop Hooks & Graceful SIGTERM Drain in real Cloud DevOps, Kubernetes & Site...

What Topic Is Next → Next Topic 13.3: Persistent Storage in Kubernetes: StorageClasses, PersistentVolumeClaims (PVC), EBS/EFS CSI & StatefulSets

13.3 Persistent Storage in Kubernetes: StorageClasses, PersistentVolumeClaims (PVC), EBS/EFS CSI & StatefulSets

[90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for P... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Persistent Storage in Kubernetes • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Persistent Storage in Kubernetes: StorageClasses, PersistentVolumeClaims (PVC), EBS/EFS CSI & StatefulSets in real Cloud DevOps, Kubernetes & Site...

What Topic Is Next → Next Topic 13.4: Ingress Controllers (NGINX / AWS ALB), Gateway API & Automated TLS Certificates with cert-manager

13.4 Ingress Controllers (NGINX / AWS ALB), Gateway API & Automated TLS Certificates with cert-manager

[90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for I... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Ingress Controllers (NGINX / AWS ALB) • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Ingress Controllers (NGINX / AWS ALB), Gateway API & Automated TLS Certificates with cert-manager in real Cloud DevOps, Kubernetes & Site Reliabil...

What Topic Is Next → Module 13 Capstone (kubepulse-m13-k8s-storage-ingress-probes) → Next Topic 14.1: Kubernetes Access Control: ServiceAccounts, Roles, ClusterRoles, RoleBindings, AWS IAM -> EKS RBAC & IRSA

Module 13 Hands-On Capstone Project (100 Marks): kubepulse-m13-k8s-storage-ingress-probes — Architect, implement, test, and push the production GitHub repository 'kubepulse-m13-k8s-storage-ingress-probes' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module...

MODULE 14 OF 22 • STAGE 3 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Production Amazon EKS: RBAC, NetworkPolicies, Autoscaling (HPA/Karpenter) & Helm

Beginner-to-expert real-repository module covering Production Amazon EKS: RBAC, NetworkPolicies, Autoscaling (HPA/Karpenter) & Helm with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Kubernetes Access Control: ServiceAccounts, Roles, ClusterRoles, RoleBindings, AWS IAM -> EKS RBAC & IRSA — Purpose: Enforce strict human and pod-level least privilege across production Amazon EKS clusters. • ✓ Zero-Trust Cluster Security: Pod Security Standards (Restricted), CNI NetworkPolicies & PodDisruptionBudgets — Purpose: Block unauthorized pod-to-pod east-west traffic and guarantee minimum availability during node upgrades. • ✓ Multi-Dimensional Autoscaling: Resource Requests/Limits, HPA, KEDA Event Scaling & Karpenter Node Autoscaling — Purpose: Scale pods on queue depth/RPS and provision right-sized Spot/On-Demand nodes in under 45 seconds. • ✓ Packaging & Templating Kubernetes Releases with Helm Charts, Kustomize Overlays & Safe Chart Rollbacks — Purpose: Manage dev, staging, and prod Kubernetes configurations cleanly without copy-pasting thousands of YAML lines.

14.1 Kubernetes Access Control: ServiceAccounts, Roles, ClusterRoles, RoleBindings, AWS IAM -> EKS RBAC & IRSA [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for K... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Kubernetes Access Control • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Kubernetes Access Control: ServiceAccounts, Roles, ClusterRoles, RoleBindings, AWS IAM -> EKS RBAC & IRSA in real Cloud DevOps, Kubernetes & Site...

What Topic Is Next → Next Topic 14.2: Zero-Trust Cluster Security: Pod Security Standards (Restricted), CNI NetworkPolicies & PodDisruptionBudgets

14.2 Zero-Trust Cluster Security: Pod Security Standards (Restricted), CNI NetworkPolicies & PodDisruptionBudgets [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Z... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Zero-Trust Cluster Security • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Zero-Trust Cluster Security: Pod Security Standards (Restricted), CNI NetworkPolicies & PodDisruptionBudgets in real Cloud DevOps, Kubernetes & Si...

What Topic Is Next → Next Topic 14.3: Multi-Dimensional Autoscaling: Resource Requests/Limits, HPA, KEDA Event Scaling & Karpenter Node Autoscaling

14.3 Multi-Dimensional Autoscaling: Resource Requests/Limits, HPA, KEDA Event Scaling & Karpenter Node Autoscaling [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for M... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Multi-Dimensional Autoscaling • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Multi-Dimensional Autoscaling: Resource Requests/Limits, HPA, KEDA Event Scaling & Karpenter Node Autoscaling in real Cloud DevOps, Kubernetes & S...

What Topic Is Next → Next Topic 14.4: Packaging & Templating Kubernetes Releases with Helm Charts, Kustomize Overlays & Safe Chart Rollbacks

14.4 Packaging & Templating Kubernetes Releases with Helm Charts, Kustomize Overlays & Safe Chart Rollbacks [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for P... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Packaging • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Packaging & Templating Kubernetes Releases with Helm Charts, Kustomize Overlays & Safe Chart Rollbacks in real Cloud DevOps, Kubernetes & Site Rel...

What Topic Is Next → Module 14 Capstone (kubepulse-m14-eks-security-helm-scaling) → Next Topic 15.1: Terraform From Scratch: HCL Syntax, Providers, Remote S3 State Locking, Variables, Outputs & Expressions

Module 14 Hands-On Capstone Project (100 Marks): kubepulse-m14-eks-security-helm-scaling — Architect, implement, test, and push the production GitHub repository 'kubepulse-m14-eks-security-helm-scaling' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 1...

MODULE 15 OF 22 • STAGE 3 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Infrastructure as Code with Terraform & Server Configuration with Ansible

Beginner-to-expert real-repository module covering Infrastructure as Code with Terraform & Server Configuration with Ansible with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Terraform From Scratch: HCL Syntax, Providers, Remote S3 State Locking, Variables, Outputs & Expressions — Purpose: Provision cloud networks, IAM, databases, and EKS clusters reproducibly from version-controlled HCL. • ✓ Reusable Terraform Modules, Multi-Environment Directory Layouts (Terragrunt / Workspaces) & Drift Remediation — Purpose: Standardize infrastructure blueprints across staging and production with zero manual console drift. • ✓ Ansible Configuration Management: Agentless SSH Inventories, Idempotent Playbooks, Roles & Jinja2 Templates — Purpose: Automate OS patching, security CIS hardening, and hybrid fleet configuration across 100+ Linux hosts. • ✓ Ansible Vault Secrets, Dynamic Cloud Inventories & Policy-as-Code IaC Scanning (tfsec / Checkov) — Purpose: Combine Terraform provisioning with Ansible host hardening and automated pre-merge security checks.

15.1 Terraform From Scratch: HCL Syntax, Providers, Remote S3 State Locking, Variables, Outputs & Expressions [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for T... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Terraform From Scratch • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Terraform From Scratch: HCL Syntax, Providers, Remote S3 State Locking, Variables, Outputs & Expressions in real Cloud DevOps, Kubernetes & Site R...

What Topic Is Next → Next Topic 15.2: Reusable Terraform Modules, Multi-Environment Directory Layouts (Terragrunt / Workspaces) & Drift Remediation

15.2 Reusable Terraform Modules, Multi-Environment Directory Layouts (Terragrunt / Workspaces) & Drift Remediation [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for R... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Reusable Terraform Modules • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Reusable Terraform Modules, Multi-Environment Directory Layouts (Terragrunt / Workspaces) & Drift Remediation in real Cloud DevOps, Kubernetes & S...

What Topic Is Next → Next Topic 15.3: Ansible Configuration Management: Agentless SSH Inventories, Idempotent Playbooks, Roles & Jinja2 Templates

15.3 Ansible Configuration Management: Agentless SSH Inventories, Idempotent Playbooks, Roles & Jinja2 Templates [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for A... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Ansible Configuration Management • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Ansible Configuration Management: Agentless SSH Inventories, Idempotent Playbooks, Roles & Jinja2 Templates in real Cloud DevOps, Kubernetes & Sit...

What Topic Is Next → Next Topic 15.4: Ansible Vault Secrets, Dynamic Cloud Inventories & Policy-as-Code IaC Scanning (tfsec / Checkov)

15.4 Ansible Vault Secrets, Dynamic Cloud Inventories & Policy-as-Code IaC Scanning (tfsec / Checkov) [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for A... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Ansible Vault Secrets • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Ansible Vault Secrets, Dynamic Cloud Inventories & Policy-as-Code IaC Scanning (tfsec / Checkov) in real Cloud DevOps, Kubernetes & Site Reliabili...

What Topic Is Next → Module 15 Capstone (kubepulse-m15-terraform-ansible-iac) → Next Topic 16.1: GitHub Actions Enterprise Workflows: Reusable Workflows, Caching, Matrix Testing & Keyless AWS OIDC

Module 15 Hands-On Capstone Project (100 Marks): kubepulse-m15-terraform-ansible-iac — Architect, implement, test, and push the production GitHub repository 'kubepulse-m15-terraform-ansible-iac' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 15: I...

MODULE 16 OF 22 • STAGE 3 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Enterprise CI/CD (GitHub Actions & Jenkins) & GitOps Progressive Delivery (ArgoCD)

Beginner-to-expert real-repository module covering Enterprise CI/CD (GitHub Actions & Jenkins) & GitOps Progressive Delivery (ArgoCD) with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ GitHub Actions Enterprise Workflows: Reusable Workflows, Caching, Matrix Testing & Keyless AWS OIDC — Purpose: Build fast, modular CI pipelines used by modern cloud-native SaaS and product engineering teams. • ✓ Jenkins Declarative Pipelines (Jenkinsfile), Shared Libraries, Ephemeral Kubernetes Build Agents & Credentials — Purpose: Master enterprise Jenkins pipelines still widely tested in MNC, banking, and GCC DevOps interviews. • ✓ GitOps Continuous Delivery with ArgoCD: App-of-Apps, ApplicationSets, Sync Waves & Automated Drift Self-Healing — Purpose: Make Git the single source of truth for every Kubernetes cluster — zero manual kubectl apply in prod. • ✓ Progressive Delivery with Argo Rollouts: Automated Canary Traffic Shifting, Metric Analysis & Blue-Green Releases — Purpose: Shift 5% -> 25% -> 100% traffic automatically and abort rollouts instantly if Prometheus error rates rise.

16.1 GitHub Actions Enterprise Workflows: Reusable Workflows, Caching, Matrix Testing & Keyless AWS OIDC [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for G... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for GitHub Actions Enterprise Workflows • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply GitHub Actions Enterprise Workflows: Reusable Workflows, Caching, Matrix Testing & Keyless AWS OIDC in real Cloud DevOps, Kubernetes & Site Reliab...

What Topic Is Next → Next Topic 16.2: Jenkins Declarative Pipelines (Jenkinsfile), Shared Libraries, Ephemeral Kubernetes Build Agents & Credentials

16.2 Jenkins Declarative Pipelines (Jenkinsfile), Shared Libraries, Ephemeral Kubernetes Build Agents & Credentials [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for J... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Jenkins Declarative Pipelines (Jenkinsfile) • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Jenkins Declarative Pipelines (Jenkinsfile), Shared Libraries, Ephemeral Kubernetes Build Agents & Credentials in real Cloud DevOps, Kubernetes &...

What Topic Is Next → Next Topic 16.3: GitOps Continuous Delivery with ArgoCD: App-of-Apps, ApplicationSets, Sync Waves & Automated Drift Self-Healing

16.3 GitOps Continuous Delivery with ArgoCD: App-of-Apps, ApplicationSets, Sync Waves & Automated Drift Self-Healing [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for G... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for GitOps Continuous Delivery with ArgoCD • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply GitOps Continuous Delivery with ArgoCD: App-of-Apps, ApplicationSets, Sync Waves & Automated Drift Self-Healing in real Cloud DevOps, Kubernetes &...

What Topic Is Next → Next Topic 16.4: Progressive Delivery with Argo Rollouts: Automated Canary Traffic Shifting, Metric Analysis & Blue-Green Releases

16.4 Progressive Delivery with Argo Rollouts: Automated Canary Traffic Shifting, Metric Analysis & Blue-Green Releases [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for P... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Progressive Delivery with Argo Rollouts • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Progressive Delivery with Argo Rollouts: Automated Canary Traffic Shifting, Metric Analysis & Blue-Green Releases in real Cloud DevOps, Kubernetes...

What Topic Is Next → Module 16 Capstone (kubepulse-m16-cicd-jenkins-argocd-gitops) → Next Topic 17.1: Shift-Left CI Security Gates: Gitleaks Secret Scanning, SonarQube/Semgrep SAST & Dependency SCA Audits

Module 16 Hands-On Capstone Project (100 Marks): kubepulse-m16-cicd-jenkins-argocd-gitops — Architect, implement, test, and push the production GitHub repository 'kubepulse-m16-cicd-jenkins-argocd-gitops' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module...

MODULE 17 OF 22 • STAGE 3 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

DevSecOps Supply Chain Security: SAST, Trivy, SBOMs, Vault & Runtime Defense

Beginner-to-expert real-repository module covering DevSecOps Supply Chain Security: SAST, Trivy, SBOMs, Vault & Runtime Defense with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Shift-Left CI Security Gates: Gitleaks Secret Scanning, SonarQube/Semgrep SAST & Dependency SCA Audits — Purpose: Catch leaked tokens, SQLi flaws, and vulnerable transitive libraries automatically on pull requests. • ✓ Container Supply Chain Security: Trivy CVE Gates, Syft SBOM Generation & Cosign Keyless Image Signing — Purpose: Verify cryptographically that only CI-built, vulnerability-scanned images are allowed into production. • ✓ Centralized Secrets Management with HashiCorp Vault: Dynamic DB Credentials, PKI & Kubernetes Auth Injection — Purpose: Issue short-lived auto-expiring database credentials instead of static long-lived passwords. • ✓ Kubernetes Admission & Runtime Security: Kyverno / OPA Gatekeeper Policies & eBPF Threat Detection with Falco — Purpose: Block privileged pods at admission and alert in real time if an attacker spawns a shell inside a container.

17.1 Shift-Left CI Security Gates: Gitleaks Secret Scanning, SonarQube/Semgrep SAST & Dependency SCA Audits [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for S... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Shift-Left CI Security Gates • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Shift-Left CI Security Gates: Gitleaks Secret Scanning, SonarQube/Semgrep SAST & Dependency SCA Audits in real Cloud DevOps, Kubernetes & Site Rel...

What Topic Is Next → Next Topic 17.2: Container Supply Chain Security: Trivy CVE Gates, Syft SBOM Generation & Cosign Keyless Image Signing

17.2 Container Supply Chain Security: Trivy CVE Gates, Syft SBOM Generation & Cosign Keyless Image Signing [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for C... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Container Supply Chain Security • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Container Supply Chain Security: Trivy CVE Gates, Syft SBOM Generation & Cosign Keyless Image Signing in real Cloud DevOps, Kubernetes & Site Reli...

What Topic Is Next → Next Topic 17.3: Centralized Secrets Management with HashiCorp Vault: Dynamic DB Credentials, PKI & Kubernetes Auth Injection

17.3 Centralized Secrets Management with HashiCorp Vault: Dynamic DB Credentials, PKI & Kubernetes Auth Injection [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for C... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Centralized Secrets Management with HashiCorp Vault • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Centralized Secrets Management with HashiCorp Vault: Dynamic DB Credentials, PKI & Kubernetes Auth Injection in real Cloud DevOps, Kubernetes & Si...

What Topic Is Next → Next Topic 17.4: Kubernetes Admission & Runtime Security: Kyverno / OPA Gatekeeper Policies & eBPF Threat Detection with Falco

17.4 Kubernetes Admission & Runtime Security: Kyverno / OPA Gatekeeper Policies & eBPF Threat Detection with Falco [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for K... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Kubernetes Admission • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Kubernetes Admission & Runtime Security: Kyverno / OPA Gatekeeper Policies & eBPF Threat Detection with Falco in real Cloud DevOps, Kubernetes & S...

What Topic Is Next → Module 17 Capstone (kubepulse-m17-devsecops-vault-falco) → Next Topic 18.1: Observability Pillars (Metrics, Logs, Traces) & Prometheus Architecture: Scraping, Exporters & kube-prometheus-stack

Module 17 Hands-On Capstone Project (100 Marks): kubepulse-m17-devsecops-vault-falco — Architect, implement, test, and push the production GitHub repository 'kubepulse-m17-devsecops-vault-falco' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 17: D...

MODULE 18 OF 22 • STAGE 4 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Full-Stack Observability: Prometheus, PromQL, Grafana, Loki & OpenTelemetry

Beginner-to-expert real-repository module covering Full-Stack Observability: Prometheus, PromQL, Grafana, Loki & OpenTelemetry with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Observability Pillars (Metrics, Logs, Traces) & Prometheus Architecture: Scraping, Exporters & kube-prometheus-stack — Purpose: Instrument Linux nodes, Kubernetes control planes, databases, and custom microservices end-to-end. • ✓ Mastering PromQL Queries (rate, histogram_quantile, sum by), Recording Rules & Alertmanager Deduplication — Purpose: Compute accurate p50/p95/p99 latency percentiles and route noise-free alerts to Slack/PagerDuty. • ✓ Executive & On-Call Grafana Dashboards (RED & USE Methods) + Centralized Log Aggregation with Grafana Loki — Purpose: Jump from a latency spike graph directly to correlated container log streams in seconds using LogQL. • ✓ Distributed Tracing with OpenTelemetry Collector & Grafana Tempo / Jaeger Across Microservices — Purpose: Visualize cross-service request waterfalls and pinpoint slow downstream SQL or API hops instantly.

18.1 Observability Pillars (Metrics, Logs, Traces) & Prometheus Architecture: Scraping, Exporters & kube-prometheus-stack [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for O... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Observability Pillars (Metrics, Logs, Traces) • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Observability Pillars (Metrics, Logs, Traces) & Prometheus Architecture: Scraping, Exporters & kube-prometheus-stack in real Cloud DevOps, Kuberne...

What Topic Is Next → Next Topic 18.2: Mastering PromQL Queries (rate, histogram_quantile, sum by), Recording Rules & Alertmanager Deduplication

18.2 Mastering PromQL Queries (rate, histogram_quantile, sum by), Recording Rules & Alertmanager Deduplication [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for M... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Mastering PromQL Queries (rate, histogram_quantile,... • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Mastering PromQL Queries (rate, histogram_quantile, sum by), Recording Rules & Alertmanager Deduplication in real Cloud DevOps, Kubernetes & Site...

What Topic Is Next → Next Topic 18.3: Executive & On-Call Grafana Dashboards (RED & USE Methods) + Centralized Log Aggregation with Grafana Loki

18.3 Executive & On-Call Grafana Dashboards (RED & USE Methods) + Centralized Log Aggregation with Grafana Loki [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for E... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Executive • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Executive & On-Call Grafana Dashboards (RED & USE Methods) + Centralized Log Aggregation with Grafana Loki in real Cloud DevOps, Kubernetes & Site...

What Topic Is Next → Next Topic 18.4: Distributed Tracing with OpenTelemetry Collector & Grafana Tempo / Jaeger Across Microservices

18.4 Distributed Tracing with OpenTelemetry Collector & Grafana Tempo / Jaeger Across Microservices [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for D... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Distributed Tracing with OpenTelemetry Collector • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Distributed Tracing with OpenTelemetry Collector & Grafana Tempo / Jaeger Across Microservices in real Cloud DevOps, Kubernetes & Site Reliability...

What Topic Is Next → Module 18 Capstone (kubepulse-m18-prometheus-grafana-otel) → Next Topic 19.1: Core SRE Principles: Defining User-Centric SLIs, SLOs, SLAs & Multi-Window Error-Budget Burn-Rate Alerts

Module 18 Hands-On Capstone Project (100 Marks): kubepulse-m18-prometheus-grafana-otel — Architect, implement, test, and push the production GitHub repository 'kubepulse-m18-prometheus-grafana-otel' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 18:...

MODULE 19 OF 22 • STAGE 4 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Site Reliability Engineering (SRE): SLIs, SLOs, Error Budgets, Load & Chaos

Beginner-to-expert real-repository module covering Site Reliability Engineering (SRE): SLIs, SLOs, Error Budgets, Load & Chaos with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Core SRE Principles: Defining User-Centric SLIs, SLOs, SLAs & Multi-Window Error-Budget Burn-Rate Alerts — Purpose: Balance release velocity with reliability mathematically instead of paging engineers on false alarms. • ✓ Capacity Planning & Load Testing with k6: Stress, Spike & Soak Tests Against Autoscaled EKS Services — Purpose: Find breaking points, connection-pool ceilings, and HPA lag in staging before Black Friday traffic hits. • ✓ Chaos Engineering GameDays (Chaos Mesh / Litmus): Pod Kills, AZ Network Partitions, Latency & DNS Faults — Purpose: Prove resilient self-healing behavior empirically under controlled fault injection. • ✓ Incident Command, Triage Playbooks, Blameless Postmortems (RCAs) & Eliminating Operational Toil in Code — Purpose: Lead high-severity production war rooms calmly and turn every outage into permanent automation.

19.1 Core SRE Principles: Defining User-Centric SLIs, SLOs, SLAs & Multi-Window Error-Budget Burn-Rate Alerts [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for C... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Core SRE Principles • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Core SRE Principles: Defining User-Centric SLIs, SLOs, SLAs & Multi-Window Error-Budget Burn-Rate Alerts in real Cloud DevOps, Kubernetes & Site R...

What Topic Is Next → Next Topic 19.2: Capacity Planning & Load Testing with k6: Stress, Spike & Soak Tests Against Autoscaled EKS Services

19.2 Capacity Planning & Load Testing with k6: Stress, Spike & Soak Tests Against Autoscaled EKS Services [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for C... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Capacity Planning • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Capacity Planning & Load Testing with k6: Stress, Spike & Soak Tests Against Autoscaled EKS Services in real Cloud DevOps, Kubernetes & Site Relia...

What Topic Is Next → Next Topic 19.3: Chaos Engineering GameDays (Chaos Mesh / Litmus): Pod Kills, AZ Network Partitions, Latency & DNS Faults

19.3 Chaos Engineering GameDays (Chaos Mesh / Litmus): Pod Kills, AZ Network Partitions, Latency & DNS Faults [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for C... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Chaos Engineering GameDays (Chaos Mesh / Litmus) • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Chaos Engineering GameDays (Chaos Mesh / Litmus): Pod Kills, AZ Network Partitions, Latency & DNS Faults in real Cloud DevOps, Kubernetes & Site R...

What Topic Is Next → Next Topic 19.4: Incident Command, Triage Playbooks, Blameless Postmortems (RCAs) & Eliminating Operational Toil in Code

19.4 Incident Command, Triage Playbooks, Blameless Postmortems (RCAs) & Eliminating Operational Toil in Code [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for I... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Incident Command • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Incident Command, Triage Playbooks, Blameless Postmortems (RCAs) & Eliminating Operational Toil in Code in real Cloud DevOps, Kubernetes & Site Re...

What Topic Is Next → Module 19 Capstone (kubepulse-m19-sre-slos-chaos-incidents) → Next Topic 20.1: Internal Developer Platforms (IDP), Golden Paths & High-Scale Cloud/DevOps System Design Interviews

Module 19 Hands-On Capstone Project (100 Marks): kubepulse-m19-sre-slos-chaos-incidents — Architect, implement, test, and push the production GitHub repository 'kubepulse-m19-sre-slos-chaos-incidents' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Module 19...

MODULE 20 OF 22 • STAGE 4 • 21 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Platform Engineering, System Design & KubePulse Live Client Production Cutover

Beginner-to-expert real-repository module covering Platform Engineering, System Design & KubePulse Live Client Production Cutover with supervised live coding, ≥90% MockAttempt AI Viva Gates, and production GitHub PR verification.

Objectives: ✓ Internal Developer Platforms (IDP), Golden Paths & High-Scale Cloud/DevOps System Design Interviews — Purpose: Design global CI/CD platforms, multi-region Kubernetes architectures, and observability pipelines on whiteboards. • ✓ End-to-End Delivery of KubePulse Zero-Downtime GitOps EKS Platform, DevSecOps Pipeline & SRE Mesh — Purpose: Unite Terraform, EKS, ArgoCD Canary rollouts, Trivy/Cosign, and Prometheus/Loki/OTel in one live stack. • ✓ Live Chaos Failover GameDay, ATS-Optimized DevOps/SRE Resume & 20-Repository GitHub Portfolio Defense — Purpose: Demonstrate live 99.95% uptime, self-healing rollbacks, and earn your Verified Internship Letter. • ✓ Linux/K8s Live Troubleshooting Labs, Spoken MockAttempt AI Vivas & ≥90% Placement Guarantee Activation — Purpose: Clear all ≥90% topic mocks and AI vivas (unlimited free re-attempts) to unlock 100 job openings in 6 months.

20.1 Internal Developer Platforms (IDP), Golden Paths & High-Scale Cloud/DevOps System Design Interviews [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for I... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Internal Developer Platforms (IDP) • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Internal Developer Platforms (IDP), Golden Paths & High-Scale Cloud/DevOps System Design Interviews in real Cloud DevOps, Kubernetes & Site Reliab...

What Topic Is Next → Next Topic 20.2: End-to-End Delivery of KubePulse Zero-Downtime GitOps EKS Platform, DevSecOps Pipeline & SRE Mesh

20.2 End-to-End Delivery of KubePulse Zero-Downtime GitOps EKS Platform, DevSecOps Pipeline & SRE Mesh [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for E... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for End-to-End Delivery of KubePulse Zero-Downtime GitOp... • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply End-to-End Delivery of KubePulse Zero-Downtime GitOps EKS Platform, DevSecOps Pipeline & SRE Mesh in real Cloud DevOps, Kubernetes & Site Reliabil...

What Topic Is Next → Next Topic 20.3: Live Chaos Failover GameDay, ATS-Optimized DevOps/SRE Resume & 20-Repository GitHub Portfolio Defense

20.3 Live Chaos Failover GameDay, ATS-Optimized DevOps/SRE Resume & 20-Repository GitHub Portfolio Defense [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for L... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Live Chaos Failover GameDay • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Live Chaos Failover GameDay, ATS-Optimized DevOps/SRE Resume & 20-Repository GitHub Portfolio Defense in real Cloud DevOps, Kubernetes & Site Reli...

What Topic Is Next → Next Topic 20.4: Linux/K8s Live Troubleshooting Labs, Spoken MockAttempt AI Vivas & ≥90% Placement Guarantee Activation

20.4 Linux/K8s Live Troubleshooting Labs, Spoken MockAttempt AI Vivas & ≥90% Placement Guarantee Activation [90 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Input Validation, Edge • Modular Clean • Idiomatic SREM Implementation • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for L... • Runnable Code Implementation (YAML) • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Linux/K8s Live Troubleshooting Labs • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Linux/K8s Live Troubleshooting Labs, Spoken MockAttempt AI Vivas & ≥90% Placement Guarantee Activation in real Cloud DevOps, Kubernetes & Site Rel...

What Topic Is Next → Module 20 Capstone (kubepulse-m20-live-client-production-capstone) → Next Topic 21.1: Declarative Kubernetes Platform Engineering with Helm, Kustomize & Argo CD GitOps

Module 20 Hands-On Capstone Project (100 Marks): kubepulse-m20-live-client-production-capstone — Architect, implement, test, and push the production GitHub repository 'kubepulse-m20-live-client-production-capstone' (strictly zero dummy scaffolding) demonstrating end-to-end mastery of Mo...

MODULE 21 OF 22 • STAGE 4 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Production Kubernetes GitOps, Service Mesh Concepts & DevSecOps Supply Chain

Self-explanatory list-and-bullet production mastery module covering Production Kubernetes GitOps, Service Mesh Concepts & DevSecOps Supply Chain in Cloud DevOps, Kubernetes & Site Reliability Engineering (SRE) Master Program, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken MockAttempt AI Viva.

Objectives: ✓ Declarative Kubernetes Platform Engineering with Helm, Kustomize & Argo CD GitOps — Purpose: Automate multi-cluster rollouts, drift healing, and progressive canary traffic shifts. • ✓ Container & Cluster Hardening: Trivy, SBOMs, Kyverno/OPA Policies & NetworkPolicies — Purpose: Enforce non-root

workloads, signed images, and zero-trust pod networking automatically. • ✓ Autoscaling at Scale: HPA, KEDA Event Scalers, Karpenter/Cluster Autoscaler & PDBs — Purpose: Handle 10x traffic surges smoothly while keeping cloud compute utilization efficient. • ✓ Build, verify, and document 'Argo CD GitOps + Terraform EKS + Trivy/OPA Policy-Gated Platform Repository' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + MockAttempt AI Technical Interviews.

21.1 Declarative Kubernetes Platform Engineering with Helm, Kustomize & Argo CD GitOps [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Security Hygiene & Data Safety Rules for Declarative Kubernetes Platform Engineering with Helm, Kustomize & Argo CD... • Performance Optimization, Complexity & Latency Impact of Declarative Kubernetes Platform Engineering with Helm, Kus... • Live Debugging, Stack • Runnable Code Implementation (YAML) • Hands • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Declarative Kubernetes Platform Engineering with Hel... • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Declarative Kubernetes Platform Engineering with Helm, Kustomize & Argo CD GitOps in real Cloud DevOps, Kubernetes & Site Reliability Engineering...

What Topic Is Next → Next Topic 21.2: Container & Cluster Hardening: Trivy, SBOMs, Kyverno/OPA Policies & NetworkPolicies

21.2 Container & Cluster Hardening: Trivy, SBOMs, Kyverno/OPA Policies & NetworkPolicies [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Security Hygiene & Data Safety Rules for Container & Cluster Hardening • Performance Optimization, Complexity & Latency Impact of Container & Cluster Hardening • Live Debugging, Stack • Runnable Code Implementation (YAML) • Hands • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Container • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Container & Cluster Hardening: Trivy, SBOMs, Kyverno/OPA Policies & NetworkPolicies in real Cloud DevOps, Kubernetes & Site Reliability Engineerin...

What Topic Is Next → Next Topic 21.3: Autoscaling at Scale: HPA, KEDA Event Scalers, Karpenter/Cluster Autoscaler & PDBs

21.3 Autoscaling at Scale: HPA, KEDA Event Scalers, Karpenter/Cluster Autoscaler & PDBs [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Security Hygiene & Data Safety Rules for Autoscaling at Scale • Performance Optimization, Complexity & Latency Impact of Autoscaling at Scale • Live Debugging, Stack • Runnable Code Implementation (YAML) • Hands • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Autoscaling at Scale • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Autoscaling at Scale: HPA, KEDA Event Scalers, Karpenter/Cluster Autoscaler & PDBs in real Cloud DevOps, Kubernetes & Site Reliability Engineering...

What Topic Is Next → Module 21 Capstone (Argo CD GitOps + Terraform EKS + Trivy/OPA Policy-Gated Platform Repository) → Next Topic 22.1: Unified Logs, Metrics & Distributed Traces (OpenTelemetry, Prometheus, Grafana, Loki)

Module 21 Hands-On Capstone Project (100 Marks): Argo CD GitOps + Terraform EKS + Trivy/OPA Policy-Gated Platform Repository — Architect, implement, test, and document 'Argo CD GitOps + Terraform EKS + Trivy/OPA Policy-Gated Platform Repository' applying every bullet-wise engineering concept from Module 21 (Producti...

MODULE 22 OF 22 • STAGE 4 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

OpenTelemetry Observability, Chaos/Incident Engineering & SRE Placement Defense

Self-explanatory list-and-bullet production mastery module covering OpenTelemetry Observability, Chaos/Incident Engineering & SRE Placement Defense in Cloud DevOps, Kubernetes & Site Reliability Engineering (SRE) Master Program, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken MockAttempt AI Viva.

Objectives: ✓ Unified Logs, Metrics & Distributed Traces (OpenTelemetry, Prometheus, Grafana, Loki) — Purpose: Correlate customer-facing latency spikes to exact pod, query, or network root causes. • ✓ SLIs, SLOs, Burn-Rate Alerting, Automated Runbooks & Chaos Failure Injection — Purpose: Verify system self-healing before real outages strike and eliminate alert fatigue. • ✓ Linux Kernel/Networking Troubleshooting & Senior DevOps/SRE Placement Interviews — Purpose: Debug DNS, TLS, iptables, OOMKills, and distributed outages live in SRE interviews. • ✓ Build, verify, and document 'Prometheus + Grafana + Loki + Tempo SLO Error-Budget & Chaos GameDay Repository' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + MockAttempt AI Technical Interviews.

22.1 Unified Logs, Metrics & Distributed Traces (OpenTelemetry, Prometheus, Grafana, Loki) [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Security Hygiene & Data Safety Rules for Unified Logs, Metrics & Distributed Traces (OpenTelemetry, Prometheus, Gra... • Performance Optimization, Complexity & Latency Impact of Unified Logs, Metrics & Distributed Traces (OpenTelemetry,... • Live Debugging, Stack • Runnable Code Implementation (YAML) • Hands • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Unified Logs • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Unified Logs, Metrics & Distributed Traces (OpenTelemetry, Prometheus, Grafana, Loki) in real Cloud DevOps, Kubernetes & Site Reliability Engineer...

What Topic Is Next → Next Topic 22.2: SLIs, SLOs, Burn-Rate Alerting, Automated Runbooks & Chaos Failure Injection

22.2 SLIs, SLOs, Burn-Rate Alerting, Automated Runbooks & Chaos Failure Injection [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Security Hygiene & Data Safety Rules for SLIs, SLOs, Burn • Performance Optimization, Complexity & Latency Impact of SLIs, SLOs, Burn • Live Debugging, Stack • Runnable Code Implementation (YAML) • Hands • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for SLIs • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply SLIs, SLOs, Burn-Rate Alerting, Automated Runbooks & Chaos Failure Injection in real Cloud DevOps, Kubernetes & Site Reliability Engineering (SRE)...

What Topic Is Next → Next Topic 22.3: Linux Kernel/Networking Troubleshooting & Senior DevOps/SRE Placement Interviews

22.3 Linux Kernel/Networking Troubleshooting & Senior DevOps/SRE Placement Interviews [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Security Hygiene & Data Safety Rules for Linux Kernel/Networking Troubleshooting & Senior DevOps/SRE Placement Inte... • Performance Optimization, Complexity & Latency Impact of Linux Kernel/Networking Troubleshooting & Senior DevOps/SR... • Live Debugging, Stack • Runnable Code Implementation (YAML) • Hands • Idiomatic SREM Implementation: immutable GitOps pipelines, canary/blue-green rollouts & SLO error budgets • Linux Kernel, GitHub Actions CI/CD, Terraform, Docker/Kubernetes & Observability Execution & Memory Lifecycle for Linux Kernel/Networking Troubleshooting • Production Hardening & Security: OIDC secretless CI, SBOM/Cosign image signing, Vault & policy-as-code

Purpose: To understand and apply Linux Kernel/Networking Troubleshooting & Senior DevOps/SRE Placement Interviews in real Cloud DevOps, Kubernetes & Site Reliability Engineering (...

What Topic Is Next → Module 22 Capstone → End-of-Course Live Client Project: KubePulse Zero-Downtime GitOps EKS Platform, DevSecOps Supply Chain & SRE Observability Mesh

Module 22 Hands-On Capstone Project (100 Marks): Prometheus + Grafana + Loki + Tempo SLO Error-Budget & Chaos GameDay Repository — Architect, implement, test, and document 'Prometheus + Grafana + Loki + Tempo SLO Error-Budget & Chaos GameDay Repository' applying every bullet-wise engineering concept from Module 22 (Open...

3. LIVE CLIENT PRODUCTION PROJECT, GLOBAL CERTIFICATIONS & CAREER SCOPE

End-of-Course Live Client Production Project + Official Internship Confirmation Letter

Project: KubePulse Zero-Downtime GitOps EKS Platform, DevSecOps Supply Chain & SRE Observability Mesh • **Confirmed Role:** Cloud DevOps, Kubernetes & Site Reliability Engineering (SRE) Resident

Domain & Production Infra: High-Traffic E-Commerce & Cloud-Native Platform Client (Linux + AWS EKS + Docker + ArgoCD + Terraform + Prometheus/Grafana/Loki + Trivy)

- Hardened Linux workloads and multi-stage distroless Docker images with automated SAST/Trivy CVE gates in CI.
- Provisioned AWS EKS clusters via Terraform and automated progressive Canary/Blue-Green rollouts with Helm & ArgoCD GitOps.
- Built full-stack Prometheus, Grafana, Loki, and OpenTelemetry SLO error-budget burn-rate alerting dashboards.
- Executed chaos-engineering failover drills and Python/Bash self-healing runbooks achieving 99.95% production uptime.

Aligned Global Vendor Certifications: **CKA** (Certified Kubernetes Administrator (CKA)) • **CKAD** (Certified Kubernetes Application Developer (CKAD)) • **DCA** (Docker Certified Associate (Container Engineering)) • **TA-003** (HashiCorp Certified: Terraform Associate (003)) • **DOP-C02** (AWS Certified DevOps Engineer – Professional)

Target Job Roles: Cloud DevOps Engineer, Site Reliability Engineer (SRE), Kubernetes & Platform Engineer, DevSecOps & GitOps Automation Engineer

Realistic Annual CTC Scope: **Placement-Guaranteed DevOps / Junior SRE (0–1 Yr):** ₹6.5 LPA – ₹14.0 LPA **Mid-Level Kubernetes DevOps / SRE-2 (1–3 Yrs):** ₹15.0 LPA – ₹29.0 LPA **Principal SRE / Staff Platform Architect (3–6+ Yrs):** ₹30.0 LPA – ₹60.0+ LPA