DevOpsSchool

SYLLABUS · LEVEL 04 OF 04 · ADVANCED

Microsoft Sentinel Advanced Training

Microsoft Sentinel technical training · Advanced level · Days 1-5

CORPORATE TRAINING (DAY-WISE)	ONLINE LIVE & INTERACTIVE	SELF-PACED VIDEO	CURRICULUM EDITION
5 days · Days 1-5	Max 8-10 hours	1-year LMS access	2026-11-01

Format: Day-wise agenda only; no hourly schedule
Delivery: Instructor-led explanations, demonstrations, guided labs, assessment, and case studies
Primary experience: Microsoft Sentinel in the Microsoft Defender portal

Level structure and cumulative inclusion

Level	New instruction	Cumulative days (corporate)	Entry requirement
Foundation	Day 1	Day 1	No prior Sentinel experience required; basic Azure and security literacy helpful.
Essential	Day 2	Days 1-2	Foundation (Day 1) completed.
Intermediate	Day 3	Days 1-3	Completion of the Essential practical progression gate.
Advanced	Days 4-5	Days 1-5	Completion of the Essential and Intermediate practical progression gates.

This syllabus: Advanced level, covering Days 1-5. Each level includes every earlier day in full so the syllabus is self-contained; in delivery each topic is taught once, and later days build on earlier lab artifacts.

Delivery formats: Corporate training runs day-wise (Foundation 1 day, Essential 2 days, Intermediate 3 days, Advanced 5 days, cumulative). Online live and interactive sessions are a maximum of 8-10 hours. Self-paced video includes one year of LMS access.

Lab baseline: An approved Azure subscription and workspace, instructor-seeded synthetic logs where needed, Microsoft Defender portal access, and permission to create test artifacts. Additional Azure/Logic Apps ingestion and retention costs may apply. Preview or license/region-dependent features should be demonstrated only when available, with a prepared example otherwise.

Reference alignment note: Microsoft Learn currently states that the Azure-portal Sentinel experience will stop being supported after **March 31, 2027**; the course therefore teaches the Defender portal as the standard experience. Feature availability, costs, and licensing should be checked again before delivery.
Source

Five-day learning progression (no repeated teaching days)

Day	Level	Principal new capability	Deliverable
Day 1	Foundation	Explain the platform and deploy a working workspace	Architecture and initial telemetry evidence
Day 2	Essential	Explore logs, enable a basic detection, and triage an incident	Query pack and first incident record

Day	Level	Principal new capability	Deliverable
Day 3	Intermediate	Ingest more complex data, correlate, enrich, and tune	Correlated custom detection and case automation
Day 4	Advanced	Engineer normalized detections, hunt, and orchestrate response	Hunt report and approval-gated playbook
Day 5	Advanced	Operationalize at enterprise scale and demonstrate readiness	Architecture, CI/CD, monitoring, and final capstone

Day 1 - Microsoft Sentinel Essentials, SOC Context, Architecture, and Initial Deployment

Level: Foundation | **Day outcome:** Understand where Sentinel fits in security operations, explain its architecture, and establish a working lab environment.

Day-specific 5W1H

Lens	Questions answered through Day 1 content
What?	What are SIEM, SOAR, XDR, security events, alerts, incidents, and Microsoft Sentinel?
Why?	Why centralize security telemetry, detection, investigation, and coordinated response?
Who?	Who owns the platform, operates the SOC, triages alerts, investigates threats, and governs access?
When?	When is SIEM useful for monitoring, compliance, detection, and incident response?
Where?	Where does Sentinel run, and where do Microsoft, multicloud, and on-premises signals originate?
How?	How do data sources, ingestion, Log Analytics, Sentinel, and the Defender portal work together?

Day 1 agenda - unique topics

ID	Topic	Coverage
F01	The security operations problem	Security visibility gaps, alert fatigue, SOC mission, and lifecycle of a security event.
F02	Fundamental security terminology	Telemetry, log, event, indicator, detection, alert, incident, case, false positive, and response action.
F03	What Microsoft Sentinel provides	Cloud-native SIEM capabilities, unified security operations experience, key benefits, and fundamental limitations.
F04	People and responsibilities	SOC L1/L2/L3 analysts, threat hunters, security engineers, cloud administrators, and service owners; basic ownership boundaries.
F05	Where Sentinel fits and when to adopt it	Microsoft/Azure, multicloud, hybrid, and on-premises contexts; common organizational use cases.
F06	Comparison with alternative approaches	SIEM versus endpoint-only monitoring, EDR/XDR, log management, and traditional SIEM; complementary rather than mutually exclusive tools.
F07	Foundational platform architecture	Data sources → connectors → ingestion/Log Analytics → analytics → incidents → investigation and response; Defender portal interface.
F08	Azure resource dependencies	Microsoft Entra tenant, subscription, resource group, Log Analytics workspace, region, and workspace association.
F09	Prerequisites and deployment planning	Lab scope, account requirements, region selection, permissions, sample data, safe test practices, and awareness of ingestion costs.
F10	Baseline access permissions	Least privilege concepts; basic Azure and Microsoft Sentinel roles; why owner and analyst access differ.
F11	Initial workspace setup	Create or select a Log Analytics workspace, enable Sentinel, connect it to the Microsoft Defender portal, and confirm navigation.
F12	First data connection	Activate one instructor-approved, straightforward Microsoft data source (for example Azure Activity, subject to access), observe connection status, and verify sample events arrive.

Practical exercises

1. Sketch the foundational Sentinel architecture and identify the responsibility of each component.

2. Provision or access a dedicated training workspace, enable Microsoft Sentinel, and open the Defender portal experience.
3. Connect an approved starter data source and record evidence of incoming events; use instructor-provided synthetic telemetry when a licensed source is unavailable.

Best practices, limitations, and troubleshooting focus

- **Best practices:** Use an isolated lab; apply least privilege; document subscription, workspace, and region; start with a small, known log source; use synthetic data instead of confidential real incidents.
- **Common errors:** Incorrect tenant or workspace, insufficient permissions, unsupported source prerequisites, no events yet, and confusion between portal views.
- **Advantages and limits:** Rapid cloud-based deployment and integration versus ongoing data charges, source-specific prerequisites, and the need for ongoing SOC processes.

Real-world case: A small enterprise needs one place to view identity and Azure administration events before developing a full detection program.

Day 1 evidence: Architecture sketch, working Sentinel workspace, documented access model, and proof of first successful data ingestion.

Official reading: What is Microsoft Sentinel? | Deployment guide | Sentinel in the Defender portal

Day 2 - Log Exploration, Basic Detection, and First-Line Incident Handling

Level: Essential | **Day outcome:** Use basic KQL, operationalize built-in detection content, and complete a guided alert-to-incident investigation.

Day-specific 5W1H

Lens	Questions answered through Day 2 content
What?	What are security log tables, KQL queries, analytic-rule templates, incidents, and workbooks?
Why?	Why turn ingested events into meaningful, explainable detections and triage decisions?
Who?	Who reviews initial alerts, validates evidence, and escalates or closes an incident?
When?	When should a SOC query logs, activate a tested template, investigate, or escalate?
Where?	Where are logs, built-in content, detections, incidents, and workbook views accessed?
How?	How does a basic query support a detection, and how does a detected event become an investigated incident?

Day 2 agenda - unique topics

ID	Topic	Coverage
F13	Understanding ingested security records	Tables, columns, event timestamps, event identifiers, data types, and lookback windows.
F14	KQL essentials	KQL pipeline model; `where`, `project`, `take`, `sort`, `count`, `summarize`, and `bin`; basic filters and aggregations.
F15	Basic log validation	Identify relevant events, check timestamps, distinguish no data from no matching records, and save simple reusable queries.
F16	Built-in security content	Navigate Content hub; distinguish solutions, detection templates, hunting content, and workbook templates at a user level.
F17	Activate a starter analytic rule	Create a rule from a suitable template, understand its prerequisite data, choose basic settings, and test expected alert output.
F18	Detection-to-incident fundamentals	Alert versus incident, severity, status, ownership, and the basic incident lifecycle.
F19	Guided first-line triage	Review incident overview, mapped entities, related alerts, chronology, and available evidence.
F20	Basic disposition and escalation	Differentiate benign activity, expected test data, suspicious activity, and confirmed issues; record evidence and hand off.
F21	Consume an out-of-box workbook	Open an existing workbook, interpret charts/filters, and use its visualization for a simple SOC question.
F22	Entry-level SOC operating procedure	Apply a standard checklist for investigating and documenting a new incident consistently.
F23	Entry-level troubleshooting	Diagnose empty query output, mismatched time filters, missing rule prerequisites, rule-not-triggered situations, and missing incident visibility.

Practical exercises

1. Run a starter query pack over the approved lab data to identify significant events and summarize activity over time.
2. Enable a prevalidated detection template against a supported data source or instructor-seeded telemetry, then generate a harmless test event.

3. Review the resulting alert/incident, document evidence and disposition, and interpret a built-in workbook relevant to that data.

Best practices, limitations, and troubleshooting focus

- **Best practices:** Validate data prerequisites before enabling rules; use explicit time windows; preserve query evidence; maintain clear incident notes; avoid creating unsupported detection assumptions.
- **Common errors:** Wrong table name, unexpected time range, a query that returns no rows, a rule that has never had matching data, and analyst access without sufficient investigation rights.
- **Advantages and limits:** Templates accelerate initial coverage, but require source validation and can produce false positives; simple KQL answers basic questions but not multistage correlation.

Real-world case: An analyst triages a suspicious administrative or identity-related event using collected evidence and an approved decision checklist.

Day 2 evidence: Basic KQL query pack, enabled starter detection, annotated test incident, and first-line triage record.

Essential progression gate: Learners can explain the end-to-end architecture, confirm ingestion, write basic KQL, deploy a tested template-based detection, and complete a guided incident investigation. Meeting this gate qualifies them for **the Intermediate level**.

Official reading: Content and solutions | Microsoft Sentinel learning resources | SC-5001 learning path

Day 3 - Intermediate Ingestion, KQL Correlation, Enrichment, and Detection Tuning

Level: Intermediate | **Entry requirement:** Successfully complete Days 1-2 | **Day outcome:** Extend telemetry coverage, correlate events, develop a tuned scheduled detection, and improve repeatable case operations.

Day-specific 5W1H

Lens	Questions answered through Day 3 content
What?	What are agent-based collection, DCRs, multitable KQL correlation, watchlists, UEBA, and custom detections?
Why?	Why collect non-Microsoft signals, enrich results, and tune rules instead of relying only on templates?
Who?	Who operates collection pipelines, owns enrichment data, maintains rules, and manages escalation queues?
When?	When should a new collection pattern, correlation query, exception, or automated assignment be introduced?
Where?	Where do Linux/network/device logs, external cloud sources, watchlists, and behavior context enter the workflow?
How?	How are events collected and mapped, correlated with KQL, enriched, converted into detections, and operationally tuned?

Day 3 agenda - unique topics

ID	Topic	Coverage
I01	Expand data-source coverage	Connector selection by source type; API-based versus agent-based patterns; compatibility and data-quality considerations.
I02	Linux and network telemetry collection	Azure Monitor Agent (AMA), Syslog versus CEF, forwarding architecture, Data Collection Rules (DCRs), and common collection paths.
I03	Third-party and multicloud intake patterns	Integration options for cloud audit, firewall, network, and application security data; source-side dependencies and field mapping.
I04	Ingestion pipeline validation	Check schema, parsing, ingestion delay, duplicate records, timestamps, and missing fields in newly connected sources.
I05	Intermediate KQL composition	`let`, `join`, `union`, `extend`, `parse`, `mv-expand`, and `distinct`; basic cross-table correlation.
I06	Correlation across identities and infrastructure	Join related identity, host, or network events; manage time windows and ambiguous identifiers.
I07	Business-context enrichment with watchlists	Build and maintain a controlled CSV-based watchlist; compare events with high-value assets, known maintenance accounts, or trusted context.
I08	Threat-intelligence indicators	Recognize indicator sources, validity/expiry, matching limitations, and contextual use in investigations or detections.
I09	Introduction to behavior enrichment	Enable or consume supported User and Entity Behavior Analytics (UEBA) context and inspect enriched user/entity evidence.
I10	Author a custom scheduled analytics rule	Convert intermediate KQL into a detection with entity mapping, thresholds, lookback, severity, and relevant custom details.
I11	Tune alert quality and incident flow	Apply exclusions, suppression/grouping choices, ownership conventions, triage queues, and false-positive feedback.
I12	Configure non-playbook automation rules	Automate incident assignment, tagging, severity changes, or other supported low-risk case actions; distinguish this from playbook orchestration reserved for Day 4.

ID	Topic	Coverage
I13	Build a focused workbook	Turn a KQL-based detection or source-health question into a simple custom operational visualization.
I14	Diagnose intermediate failures	Work through DCR/AMA data gaps, CEF versus Syslog format issues, incorrect KQL joins, and excessive or missing alerts.

Practical exercises

1. Onboard a preapproved Linux Syslog/CEF lab feed using AMA and a DCR, or analyze instructor-provided results when lab infrastructure is restricted.
2. Correlate two telemetry sources with intermediate KQL, enrich matches using a watchlist, and document relevant UEBA/threat-intelligence context if available.
3. Publish a custom scheduled detection with entity mapping, tune a known false-positive condition, configure an assignment/tagging automation rule, and build a focused workbook.

Best practices, limitations, and troubleshooting focus

- **Best practices:** Validate log fidelity before correlation; document ownership of watchlists and indicators; limit noisy event sources; tune on replayable benign/malicious test cases; prefer reversible automation for case metadata.
- **Common errors:** AMA/DCR not assigned, unsupported log levels, parse mismatches, many-to-many join explosion, stale watchlists, unsuitable thresholds, and accidental alert storms.
- **Advantages and limits:** Broader collection and contextual correlation improve coverage but introduce pipeline maintenance, data costs, variable source quality, and higher tuning effort.

Real-world case: A suspicious identity event becomes meaningful only after linking it with firewall or Linux activity and determining whether the user belongs to a privileged watchlist.

Day 3 evidence: Additional telemetry source record, correlated KQL query, governed watchlist, tuned custom detection, case assignment automation, and custom workbook.

Intermediate progression gate: Learners can connect or validate a second data source, correlate logs, enrich findings, create and tune a scheduled detection, and apply safe case-management automation. Meeting this gate qualifies them for **the Advanced level**.

Official reading: Data connectors | CEF/Syslog through AMA | Watchlists | Automation rules

Day 4 - Advanced Detection Engineering, Threat Hunting, and SOAR Playbooks

Level: Advanced | **Entry requirement:** Successfully complete Days 1-3 | **Day outcome:** Engineer normalized and higher-fidelity detections, run a hypothesis-led hunt, and automate a controlled response through Logic Apps.

Day-specific 5W1H

Lens	Questions answered through Day 4 content
What?	What are ASIM normalization, detection engineering, proactive threat hunts, NRT rules, and Logic Apps playbooks?
Why?	Why standardize heterogeneous data, investigate attacker behavior proactively, and orchestrate repeatable response?
Who?	Who approves detection logic, hunts for threats, builds playbooks, and authorizes high-impact remediation?
When?	When are high-speed rules, hypothesis-led hunts, escalation playbooks, and human approval gates appropriate?
Where?	Where do normalized queries, hunting evidence, incident triggers, and workflow actions operate?
How?	How are robust detections engineered and tested, hunts converted into findings, and secure playbooks invoked from automation rules?

Day 4 agenda - unique topics

ID	Topic	Coverage
A01	Advanced Security Information Model (ASIM)	Normalized schemas, unifying parsers, source-specific parsers, and why normalization helps multi-vendor detection.
A02	Parser-aware high-performance querying	Use applicable ASIM filter parameters; assess field consistency, malformed input, and query performance.
A03	Multi-stage attack detection engineering	Build correlation logic around sequence, distinct entities, context, and realistic attack chains; define test cases and expected outcomes.
A04	Advanced detection execution modes	Scheduled rules versus near-real-time (NRT) rules; execution/latency constraints, output limits, and suitability.
A05	Detection validation and quality engineering	Measure signal-to-noise, precision proxies, missed detections, false positives, and change safety using benign and synthetic attack tests.
A06	Threat-hunting methodology	Form hypotheses, choose evidence sources, search for attacker behaviors, record findings, and decide whether to operationalize a detection.
A07	MITRE ATT&CK mapping	Link techniques to hypotheses and telemetry coverage; distinguish coverage mapping from proof of effective detection.
A08	Advanced entity and anomaly investigation	Pivot from behavior anomalies and UEBA context to surrounding events; challenge suspicious signals with alternate explanations.
A09	SOAR and playbook architecture	Azure Logic Apps workflow components, Sentinel automation-rule triggers, incident/alert/entity context, and response dependencies.
A10	Build a controlled incident playbook	Compose a workflow that enriches evidence, opens or updates an authorized ticket, and requests approval before disruptive actions.
A11	Secure workflow identity and permissions	Managed identities or securely managed connectors, required roles, scoped resources, and segregation of duties.
A12	Reliable orchestration design	Idempotency, retries, timeouts, rate limits, error branches, duplicate triggers, and safe rollback design.

ID	Topic	Coverage
A13	Response governance and guardrails	Manual versus automated remediation, human-in-the-loop decisions, testing mode, auditability, and approval thresholds.
A14	Advanced response debugging	Examine automation-run outcomes and Logic Apps execution histories; resolve unauthorized actions, failed connectors, and unexpected repetitions.

Practical exercises

1. Translate a vendor-specific query to a supported ASIM schema and evaluate a higher-fidelity detection against a synthetic attack chain.
2. Conduct a structured threat hunt using a documented hypothesis and MITRE ATT&CK technique; capture findings and propose a new detection.
3. Create a Logic Apps playbook triggered through a Microsoft Sentinel automation rule, enrich a simulated incident, request approval, and test safe failure handling.

Best practices, limitations, and troubleshooting focus

- **Best practices:** Use supported normalized parsers when appropriate; prefilter expensive searches; document test datasets; require approval for identity/network containment; avoid secrets in workflow definitions; design idempotent playbooks.
- **Common errors:** Unsupported parser fields, queries exceeding execution constraints, incorrect NRT assumptions, missing Sentinel/Logic Apps permissions, duplicate playbook runs, and unsafe automated remediation.
- **Advantages and limits:** Behavioral hunts and SOAR improve investigation depth and response consistency, but need skilled tuning, workflow ownership, and controls to prevent harmful actions.

Real-world case: Analysts hunt a staged identity-compromise sequence and use an approval-gated playbook to notify the incident commander and update an incident-management ticket.

Day 4 evidence: ASIM-based query, detection test matrix, threat-hunt report, and tested approval-gated Logic Apps playbook.

Official reading: ASIM parsers | Hunting capabilities | NRT rules | Run playbooks

Day 5 - Enterprise Architecture, Data Lake, DevSecOps, Governance, and Capstone

Level: Advanced | **Entry requirement:** Complete Days 1-4 | **Day outcome:** Design a production-ready Sentinel operating model, manage security data at scale, deliver detection content as code, and validate end-to-end readiness.

Day-specific 5W1H

Lens	Questions answered through Day 5 content
What?	What constitutes an enterprise Sentinel architecture, data-governance strategy, content-as-code pipeline, and measurable SOC service?
Why?	Why formalize scale, retention, operational health, cost controls, continuity, and change governance?
Who?	Who owns tenant/workspace design, release approvals, cost management, security compliance, and SOC performance?
When?	When should an organization scale to multiple workspaces, adopt long-term data-lake retention, or formalize CI/CD and production SLOs?
Where?	Where do multitenant boundaries, archival tiers, content repositories, audit trails, and reporting systems live?
How?	How can teams architect, deploy, govern, monitor, optimize, and validate Sentinel as a production security capability?

Day 5 agenda - unique topics

ID	Topic	Coverage
A15	Enterprise workspace and tenant architecture	Centralized versus distributed workspaces; geography, residency, access boundaries, multitenant operations, and Azure Lighthouse where applicable.
A16	Production reference-architecture decisions	Design for multiple business units, cloud accounts, operational isolation, cross-workspace query boundaries, and organizational ownership.
A17	Analytics versus data lake tiers	Detection-ready analytics storage versus cost-oriented long-term lake storage, query/latency differences, retention governance, and investigation needs.
A18	Historical security analytics	Data-lake KQL jobs, historical investigations, optional notebook workflows, and when to promote or summarize retained data.
A19	Security-data economics	Ingestion volume, commitment-tier versus pay-as-you-go considerations, retention charges, extra workflow costs, and budget forecasting.
A20	Cost and coverage optimization	Source prioritization, collection filtering, retention plans, useful-telemetry coverage, and the tradeoff between savings and detection gaps.
A21	Detection/content as code	GitHub or Azure DevOps repository integration, supported Bicep/ARM content formats, branch structure, versioning, and repeatable deployments.
A22	Production release governance	Automated validation, peer review, promotion across environments, change records, artifact ownership, controlled rollback, and separation of duties.
A23	Platform reliability, health, and auditing	Enable and use supported `SentinelHealth` and `SentinelAudit` telemetry; review connector, detection, and automation health; distinguish platform and workflow diagnostics.
A24	SOC service-level and performance management	Establish coverage measures, false-positive trend, MTTD/MTTR, escalation SLAs, improvement backlog, and SOC optimization actions.
A25	Portal modernization and migration planning	Transition planning for the Defender portal, inventory/compatibility assessment, training impacts, and feature availability/licensing checks.

ID	Topic	Coverage
A26	Operational resilience and governance	Incident runbooks, evidence handling, access reviews, recovery procedures, retention/legal controls, and dependency failure scenarios.
A27	Final enterprise capstone	Present an architecture, ingest/quality strategy, threat-detection plan, controlled response design, repository-based release plan, and cost/operability justification.

Practical exercises

1. Design a multiteam or multicloud production topology and justify workspace count, tenant boundaries, retention, access roles, and data-lake placement.
2. Store a previously validated detection as supported declarative content, use an approved GitHub/Azure DevOps workflow to validate and deploy it to a test workspace, and describe a rollback.
3. Diagnose a simulated production degradation using health/audit evidence and present the final capstone: a threat scenario, detection, approved response path, data-retention decisions, cost guardrails, and SOC metrics.

Best practices, limitations, and troubleshooting focus

- **Best practices:** Treat detection content as code; enforce approvals and audit trails; establish per-source value and retention rationale; test production changes in isolation; track quality and budget alongside alert counts.
- **Common errors:** Excessive workspace fragmentation, unintended cross-tenant visibility, unsuitable lake-only retention for real-time detections, repository permission failures, missing deployment parameters, and unmonitored ingestion or rule drift.
- **Advantages and limits:** Cloud-native scale, long-term analytics, and continuous delivery support enterprise SOC operations but require governance, cost ownership, and validated feature availability by license/region.

Real-world case: A regulated multinational operates a unified SOC spanning Azure and third-party systems while maintaining data residency controls, Git-based detection changes, audit evidence, and defensible telemetry costs.

Day 5 evidence: Enterprise design decision record, tier/retention matrix, repository deployment demonstration, monitoring runbook, cost model, and capstone presentation.

Advanced completion gate: Learners defend architecture and governance choices, show a validated advanced detection and safe playbook, explain data-tier tradeoffs, demonstrate change-control and health monitoring, and complete the integrated capstone.

Official reading: Data lake overview | Data tiers and retention | Pricing and billing | Sentinel repositories | Health and audit

Training quality and assessment design

Stage checkpoint	Evidence used for assessment	Next-step rule
End of Day 2 (Essential)	Setup and ingestion evidence, basic KQL, starter alert, incident triage	Pass before entering Day 3.
End of Day 3 (Intermediate)	Second-source collection evidence, correlation, enrichment, tuned rule, safe case automation	Pass before entering Day 4.
End of Day 5 (Advanced)	ASIM/hunt/response demonstration, enterprise blueprint, governed deployment, cost/reliability plan	Final program completion.

Recommended instructor assessment approach: use checklists and observable lab artifacts, not attendance alone. Allow a supervised remediation attempt for learners who miss a prerequisite. Reuse one synthetic incident storyline throughout the course, but introduce **new skills** on each successive day.

5W1H and requested topic coverage map

Requested training lens	Primary day where it is introduced and assessed
What / Why / Who / When / Where / How	Applied to the **distinct new capability on every day**, in each day's 5W1H matrix.
Architecture	Day 1 - core flow; Day 4 - SOAR workflow architecture; Day 5 - enterprise multiworkspace design.
Installation and setup	Day 1 - core workspace; Day 3 - AMA/DCR source integration; Day 4 - playbook setup; Day 5 - automated deployment.
Practical examples and real-world use cases	Fresh scenario and hands-on evidence on every day.
Best practices and troubleshooting	Specific, level-appropriate guidance and failure modes on every day.
Advantages and limitations	Basic platform evaluation on Day 1; feature-specific tradeoffs in later days.
Comparisons with alternatives	Day 1 - SIEM/XDR/log-management choices; Day 3 - ingestion/detection choices; Day 4 - NRT/scheduled and manual/automated actions; Day 5 - architecture and data-tier choices.

Scope note: This is an instructional **agenda**, not a time-blocked schedule or operational authorization to connect production sources. Changes, connector selections, and automated responses must be adapted to the actual tenant, entitlements, and security policy.

Enroll or ask for a corporate batch: https://www.devopsschool.com/courses/microsoft/sentinel/advance.html
contact@DevOpsSchool.com · +91 99057 40781 · +1 469 756 6329 · www.devopsschool.com