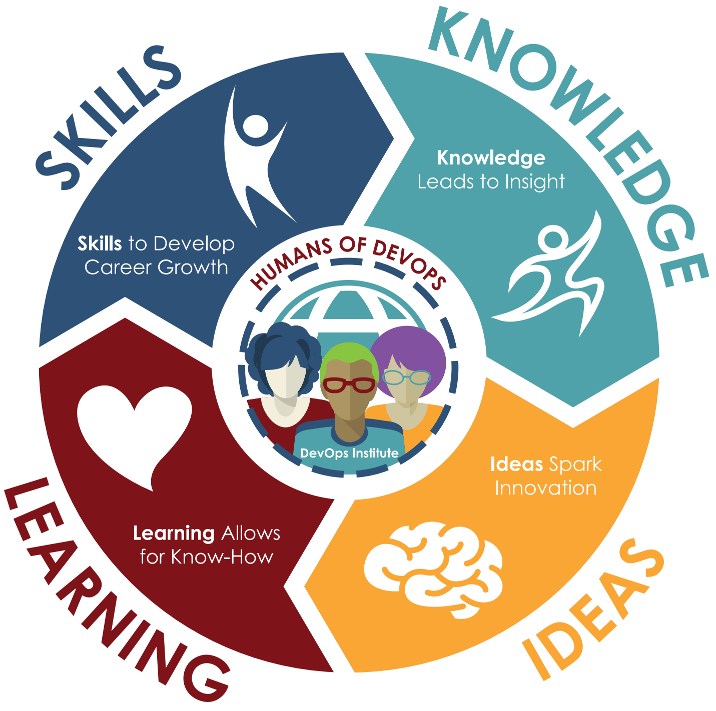
DevOps Institute

Observability Foundation℠ Exam Study Guide


SKILLS
KNOWLEDGE
IDEAS
LEARNING
Skills to Develop Career Growth
Knowledge Leads to Insight
Ideas Spark Innovation
Learning Allows for Know-How
HUMANS OF DEVOPS
DevOps Institute

DevOps Institute's SKIL Framework

DevOps Institute is dedicated to advancing the human elements of DevOps success through our human-centered SKIL framework of Skills, Knowledge, Ideas and Learning.

We develop, accredit and orchestrate SKIL through certifications, research, learning opportunities, events and community connections.

Visit us at www.devopsinstitute.com to learn more.

Join Us!

Become a member and join the fastest growing global community of DevOps practitioners and professionals.

The DevOps Institute continuous learning community is your go-to hub for all things DevOps, so get ready to learn, engage, and inspire.

Visit www.devopsinstitute.com/membership/ to join today.

You belong.

DevOps Institute

DevOps Institute is dedicated to advancing the human elements of DevOps success. We fulfill our mission through our SKIL framework of Skills, Knowledge, Ideas and Learning.

Certification is one means of showcasing your skills. While we strongly support formal training as the best learning experience and method for certification preparation, DevOps Institute also recognizes that humans learn in different ways from different resources and experiences. As the defacto certification body for DevOps, DevOps Institute has now removed the barrier to certification by removing formal training prerequisites and opening our testing program to anyone who believes that they have the topical knowledge and experience to pass one or more of our certification exams.

This examination study guide will help test-takers prepare by defining the scope of the exam and includes the following:

- Course Description
- Examination Requirements
- DevOps Glossary of Terms
- Value Added Resources
- Sample Exam(s) with Answer Key

These assets provide a guideline for the topics, concepts, vocabulary and definitions that the exam candidate is expected to know and understand in order to pass the exam. The knowledge itself will need to be gained on its own or through training by one of our Global Education Partners.

Test-takers who successfully pass the exam will also receive a certificate and digital badge from DevOps Institute, acknowledging their achievement, that can be shared with their professional online networks.

If you have any questions, please contact our DevOps Institute Customer Service team at CustomerService@DevOpsInstitute.com.

Observability Foundation^{SM}
Course Description

DURATION - 16 Hours

OVERVIEW

Microservices and Cloud-Native architectures have been goals of many organizations to help increase speed and agility, but as complexity grows, systems become increasingly challenging to observe. When issues occur, these issues are often difficult to triage and identify the root causes. This course introduces a range of practices for advancing resilience and how to architect end-to-end Observability for Cloud-Native applications. The advantages of building full-stack metrics, events, logs, and distributed tracing are introduced, along with the impact of DevSecOps on Observability and how AIOPs enhance Observability capabilities. This course also covers how Network and Security Observability plays a key role in building reliability, the key aspects of security operations and automated responses are covered,

The course aims to equip participants with the practices, methods, and tools to engage people across the organization involved in Observability by using real-life scenarios and case stories. Upon completion of the course, participants will have tangible takeaways to leverage situations such as implementing MELT models effectively, that fit their organizational context, building distributed tracing and resiliency by design.

The course is developed by leveraging key experts in the fields of telemetry, sources of knowledge and engaging with thought-leaders in the Observability space, and working with organizations who have crossed the chasm of modern Observability to extract real-life best practices.

This course positions learners to successfully complete the Observability Foundation certification exam.

COURSE OBJECTIVES

At the end of the course, the following learning objectives are expected to be achieved:

1. Practical view of how to successfully implement a flourishing Observability culture in your organization
2. The underlying principles of Observability and an understanding why monitoring on its own will not provide the required results in microservices based containerized environments
3. Understanding the three pillars of Observability
4. Adopting open Telemetry standards helps achieve innovation and distributed tracing in a seamless manner
5. Observability Maturity Model and the measurement of practical observability
6. Implementing full stack Observability and distributed tracing will enable a DevSecOps culture

7. Curating Observability using AI to move from reactive to proactive and predictive incident management. Also, how you use DataOps to build a clean data lineage of observable data.
8. Implementing Network, Container level Observability and why is security a first class citizen in building the Observability culture
9. What is Time based Topology, and how does it add value in Observability for a distributed environment
10. The Data paradox,and how we address data issues using a systematic approach (DataOps) to build a clean Observability pipeline
11. How do we feedforward DevSecOps wisdom into Observability
12. Observability practices for DevSecOps and SRE

AUDIENCE

The target audience for the Observability Foundation course are professionals including:

- Anyone focused on large-scale service scalability and reliability
- Anyone interested in modern IT leadership and organizational change approaches
- Business Managers
- Business Stakeholders
- Change Agents
- Consultants
- DevOps Practitioners
- IT Directors
- IT Managers
- IT Team Leaders
- Product Owners
- Scrum Masters
- Software Engineers
- Site Reliability Engineers
- System Integrators
- Tool Providers

LEARNER MATERIALS

- Sixteen (16) hours of instructor-led training and discussion facilitation
- Learner Manual (excellent post-class reference)
- Participation in unique exercises designed to apply concepts
- Sample exam
- Glossary
- Access to additional value-added resources and communities

PREREQUISITES

It is highly recommended that learners attend the SRE Foundation course with an accredited DevOps Institute Education Partner and earn the SRE Foundation certification prior to attending the Observability Foundation course and exam. An understanding and knowledge of common SRE terminology, concepts, principles and related work experience are recommended**.**

CERTIFICATION EXAM

Successfully passing (65%) the 60-minute examination, consisting of 40 multiple-choice questions, leads to the Observability Foundation certificate. The certification is governed and maintained by DevOps Institute.

COURSE OUTLINE

Course Introduction

Module 1: Exploring Observability

Module 2: Pillars of Observability

Module 3: Open Source Landscape for Observability

Module 4: Service Maps and Topology

Module 5: **DataOps Helps Get Observability Right**

Module 6: Building Observability with AIOps

Module 7: Security and Networking with Observability

Module 8: Observability Practices for DevOps and SRE

DevOps Institute

Observability Foundation®

Examination Requirements

Observability (OBSF) Foundation℠ Certificate

Observability (OBSF) Foundation is a freestanding certification from DevOps Institute. The purpose of this certification and its associated course is to impart, test and validate knowledge of Observability basic vocabulary, principles and practices. Observability Foundation is intended to provide individuals an understanding of basic Observability concepts and how observability may be used to improve operational activities by applying observability principles and practices.

Eligibility for Examination

Although there are no formal prerequisites for the exam, DevOps Institute highly recommends the following to prepare candidates for the exam leading to Observability Foundation certification:

- It is recommended that candidates complete at least 16 contact hours (instruction and labs) as part of a formal, approved training course delivered by an accredited Education Partner of DevOps Institute

Examination Administration

The Observability Foundation certification is accredited, managed and administered under the strict protocols and standards of DevOps Institute.

Level of Difficulty

The Observability Foundation certification uses the Bloom Taxonomy of Educational Objectives in the construction of both the content and the examination.

- The Observability Foundation exam contains Bloom 1 questions that test learners' ***knowledge*** of observability concepts and vocabulary terms (see list below)
- The exam also contains Bloom 2 questions that test learners' ***comprehension*** of these concepts in context

Format of the Examination

Candidates must achieve a passing score to gain the Observability Foundation Certificate.

Exam Type	40 multiple choice questions
Duration	60 minutes
Prerequisites	It is recommended that candidates complete the Observability (OBSF) Foundation course from an accredited DevOps Institute Education Partner
Supervised	No
Open Book	Yes
Passing Score	65%
Delivery	Web-based
Badge	Observability Foundation Certified

Exam Topic Areas and Question Weighting

The Observability Foundation exam requires knowledge of the topic areas described below.

Topic Area	Description	Max Questions
OBSF – 1: Exploring Observability	Observability definition, importance of observability, limitations of traditional monitoring and why it is not enough, and Observability Maturity Model	7
OBSF – 2: Pillars of Observability	Definition and application of telemetry including the Three Pillars of Observability (logs, metrics, tracing)	8
OBSF – 3: Open Source Landscape	Elements of observability, clarifying OpenTelemetry and understanding the open source ecosystem	7
OBSF – 4: Service Maps and Topology	Understanding service maps, defining topology, and understanding the benefits of time travel topology, and reading escalation graphs	2
OBSF – 5: DataOps Helps Get Observability Right	Observability and data paradox, understanding why Observability needs DataOps, understanding data ownership and governance along with data privacy and observability	4
OBSF – 6: Building Observability with AIOps	Enterprise platforms and AIOps and AIOps using artificial intelligence and machine learning technologies	3
OBSF – 7: Security and Networking with Observability	Observing security, container security, network observability, and visibility and integration of securities	5
OBSF-8: Observability Practices for DevOps and SRE	Understanding observability indicators, dashboards and visualization with observability, and chaos engineering	4

Concept and Terminology List

The candidate is expected to understand, comprehend and apply the following Observability Foundation concepts and vocabulary at a Blooms 1 (Knowledge) and 2 (Comprehension)

Application Performance Management (APM)	Metrics
Automatic Instrumentation	Monitoring
Cardinality	Observability
Causal Observability	Observability Backend
Chaos Engineering	Observability Enablement
CI/CD Pipeline Scanning	Observability Center of Excellence
CIA (Confidentiality, Integrity, Availability) Triad	Observability Driven Design (ODD)
Cloud Native Computing Foundation (CNCF)	Observability Maturity Model
Cloud Networking	OpenTelemetry (OTel)
Container Scanning	Open Telemetry Protocol (OTLP)
Data Governance	Predictive Analytics
Data Paradox	RED (Rate, Errors, Duration) Metrics View
DataOps	Registry Scanning
DEI (Distributed, Immutable, Ephemeral) Triad	Root Span
Dependency Relations	Sample Rate
Distributed Debugging	Sampling
Distributed Logging	Saturation
Distributed Traces	Service Map
Domain-Agnostic Solutions	Signals
Domain-Centric Solutions	Service Level Agreement (SLA)
eBPF	Service Level Indicator (SLI)
ELK (Elasticsearch, Logstash, Kibana) Stack	Service Level Objective (SLO)
Escalation Graph	Span
Events	Span ID
Filtering Metrics	Structured Logs
Filtering Traces	Telemetry
Instrumentation	Temporal Relations
Instrumenting with Agents	Three Pillars of Observability
Instrumenting with Libraries	Time Travel Topology
Kubernetes	Topology
Latency	Trace ID
Local Development Scanning	Trace Propagation
Log Analysis	Traces
Log Management	USE (Utilization, Errors, Duration) Metrics
Logs	User Interface (UI)
MELT	W3C Standard

DevOps Institute

DEVOPS GLOSSARY OF TERMS

This glossary is provided for reference only as it contains key terms that may or may not be examinable.

DevOps Glossary of Terms

Term	Definition	Course Appearances
12-Factor App Design	A methodology for building modern, scalable, maintainable software-as-a-service applications.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
2-Factor *or* 2-Step Authentication	Two-Factor Authentication, also known as 2FA or TFA or Two-Step Authentication is when a user provides two authentication factors; usually, firstly a password and then a second layer of verification such as a code texted to their device, shared secret, physical token, or biometrics.	DevSecOps Foundation
A/B Testing	Deploy different versions of an EUT to different customers and let the customer feedback determine which is best.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
A3 Problem Solving	A structured problem-solving approach that uses a lean tool called the A3 Problem-Solving Report. The term "A3" represents the paper size historically used for the report (a size roughly equivalent to 11" x 17").	DevOps Foundation
Access Management	Granting an authenticated identity access to an authorized resource (e.g., data, service, environment) based on defined criteria (e.g., a mapped role), while preventing unauthorized identity access to a resource.	DevSecOps Foundation, DevOps Engineering Foundation
Access Provisioning	Access provisioning is the process of coordinating the creation of user accounts, e-mail authorizations in the form of rules and roles, and other tasks such as provisioning of physical resources associated with enabling new users to systems or environments.	DevSecOps Foundation, DevOps Engineering Foundation
Administration Testing	The purpose of the test is to determine if an End User Test (EUT) is able to process administration tasks as expected.	Continuous Delivery Ecosystem Foundation

Advice Process	Any person making a decision must seek advice from everyone meaningfully affected by the decision and people with expertise in the matter. Advice received must be taken into consideration, though it does not have to be accepted or followed. The objective of the advice process is not to form a consensus, but to inform the decision-maker so that they can make the best decision possible. Failure to follow the advice process undermines trust and unnecessarily introduces risk to the business.	DevSecOps Foundation
Agile	A work management method for complex endeavors that divides tasks into small "sprints" of work with frequent reassessment and adaptation of plans.	Certified Agile Service Manager, DevOps Foundation, Site Reliability Engineering, Value Stream Management Foundation, DevOps Engineering Foundation
Agile (adjective)	Able to move quickly and easily; well-coordinated. Able to think and understand quickly; able to solve problems and have new ideas.	Certified Agile Service Manager, DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Agile Coach	Help teams master Agile development and DevOps practices; enables productive ways of working and collaboration.	DevOps Leader, Value Stream Management Foundation
Agile Enterprise	A fast-moving, flexible, and robust company capable of rapid response to unexpected challenges, events, and opportunities.	DevOps Foundation, DevSecOps Foundation
Agile Manifesto	A formal proclamation of values and principles to guide an iterative and people-centric approach to software development. http://agilemanifesto.org	Certified Agile Service Manager, DevOps Foundation, DevOps Engineering Foundation

Agile Portfolio Management	Involves evaluating in-flight projects and proposed future initiatives to shape and govern the ongoing investment in projects and discretionary work. CA's Agile Central and VersionOne are examples.	Site Reliability Engineering
Agile Practice Owner	A role accountable for the overall quality of a service management practice and owner of the Practice Backlog.	Certified Agile Service Manager
Agile Principles	The twelve principles that underpin the Agile Manifesto.	Certified Agile Service Manager
Agile Process	Delivers "just enough" structure and control to enable the organization to achieve its service outcomes in the most expeditious, effective, and efficient way possible. It is easy to understand, easy to follow, and prizes its collaboration and outcomes more than its artifacts.	Certified Agile Service Manager
Agile Process Engineering	An iterative and incremental approach to designing a process with short, iterative designs of potentially shippable process increments or microprocesses.	Certified Agile Service Manager
Agile Process Improvement	Ensures that IT Service Management agility introduced through Agile Process Engineering is continually reviewed and adjusted as part of IT Service Management's commitment to continual improvement.	Certified Agile Service Manager
Agile Service Management	A framework that ensures that ITSM processes reflect Agile values and are designed with "just enough" control and structure in order to effectively and efficiently deliver services that facilitate customer outcomes when and how they are needed.	Certified Agile Service Manager
Agile Service Management Artifacts	Practice Backlog, Sprint Backlog, Increment	Certified Agile Service Manager
Agile Service Management Events	Practice/microprocess Planning, The Sprint, Sprint Planning, Process Standup, Sprint Review, Sprint Retrospective	Certified Agile Service Manager

Agile Service Management Roles	Agile Practice Owner, Agile Service Management Team, Agile Service Manager	Certified Agile Service Manager
Agile Service Management Team	A team of at least 3 people (including a customer or practitioner) that is accountable for a single microprocess or a complete service management practice.	Certified Agile Service Manager
Agile Service Manager	An Agile Service Management subject matter expert who is the coach and protector of the Agile Service Management Team.	Certified Agile Service Manager
Agile Software Development	Group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. Usually applied using the Scrum or Scaled Agile Framework approach.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation
Alerting	The process of notifying operators or developers when a system is behaving unexpectedly or has an issue. Alerts can be triggered based on metrics, logs, or other signals.	Observability Foundation
Amazon Web Services (AWS)	Amazon Web Services (*AWS*) is a secure cloud services platform, offering compute power, database storage, content delivery, and other functionality to help businesses scale and grow.	DevSecOps Foundation, Site Reliability Engineering, DevOps Engineering Foundation
Analytics	Test results processed and presented in an organized manner in accordance with analysis methods and criteria.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Anomaly detection	The process of identifying unusual or unexpected behavior within a system. Anomaly detection can be used to identify issues before they cause significant problems.	Observability Foundation
Andon	A system gives an assembly line worker the ability, and moreover the empowerment, to stop production when a defect is found, and immediately call for assistance.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation

Anti-pattern	A commonly reinvented but poor solution to a problem.	DevOps Foundation, DevOps Engineering Foundation
Anti-fragility	Antifragility is a property of systems that increases its capability to thrive as a result of stressors, shocks, volatility, noise, mistakes, faults, attacks, or failures.	DevOps Foundation, Site Reliability Engineering, DevOps Engineering Foundation
API Testing	The purpose of the test is to determine if an API for an EUT functions as expected.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Application Performance Management (APM)	APM is the monitoring and management of the performance and availability of software applications. APM strives to detect and diagnose complex application performance problems to maintain an expected level of service.	Site Reliability Engineering, Value Stream Management Foundation, DevOps Engineering Foundation, Observability Foundation
Application Programming Interface (API)	A set of protocols used to create applications for a specific OS or as an interface between modules or applications.	DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Application Programming Interface (API) Testing	The purpose of the test is to determine if an API for an EUT functions as expected.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Application Release	Controlled continuous delivery pipeline capabilities including automation (release upon code commit).	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation

Application Release Automation (ARA) or Orchestration (ARO)	Controlled continuous delivery pipeline capabilities including automation (release upon code commit), environment modeling (end-to-end pipeline stages, and deploy application binaries, packages, or other artifacts to target environments), and release coordination (project, calendar, and scheduling management, integrate with change control and/or IT service support management).	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Application Test-Driven Development (ATDD)	Acceptance Test-Driven Development (ATDD) is a practice in which the whole team collaboratively discusses acceptance criteria, with examples, and then distills them into a set of concrete acceptance tests before development begins.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Application Testing	The purpose of the test is to determine if an application is performing according to its requirements and expected behaviors.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Application Under Test (AUT)	The EUT is a software application. E.g. Business application is being tested.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Architecture	The fundamental underlying design of computer hardware, software, or both in combination.	DevSecOps Foundation, DevOps Engineering Foundation
Artifact	Any element in a software development project including documentation, test plans, images, data files, and executable modules.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Artifact Repository	Store for binaries, reports, and metadata. Example tools include JFrog Artifactory, Sonatype Nexus.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation

Attack path	The chain of weaknesses a threat may exploit to achieve the attacker's objective. For example, an attack path may start by compromising a user's credentials, which are then used in a vulnerable system to escalate privileges, which in turn is used to access a protected database of information, which is copied out to an attacker's own server(s).	DevSecOps Foundation
Audit Management	The use of automated tools to ensure products and services are auditable, including keeping audit logs of build, test and deploy activities, auditing configurations, and users, as well as log files from production operations.	Site Reliability Engineering
Authentication	The process of verifying an asserted identity. Authentication can be based on what you know (e.g., password or PIN), what you have (token or one-time code), what you are (biometrics), or contextual information.	DevSecOps Foundation
Authorization	The process of granting roles to users to have access to resources.	DevSecOps Foundation
Auto-DevOps	Auto DevOps brings DevOps best practices to your project by automatically configuring software development lifecycles. It automatically detects, builds, tests, deploys, and monitors applications.	Site Reliability Engineering
Auto-scaling	The ability to automatically and elastically scale and de-scale infrastructure depending on traffic and capacity variations while maintaining control of costs.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Automatic Instrumentation	A feature in OpenTelemetry that allows developers to easily monitor the performance and behavior of their applications without having to manually instrument the code with the OpenTelemetry SDKs.	Observability Foundation

Automated rollback	If a failure is detected during a deployment, an operator (or an automated process) will verify the failure and roll back the failing release to the previous known working state.	Site Reliability Engineering, DevOps Engineering Foundation
Availability	Availability is the proportion of time a system is in a functioning condition and therefore available (to users) to be used.	Site Reliability Engineering, DevOps Engineering Foundation
Backdoor	A backdoor bypasses the usual authentication used to access a system. Its purpose is to grant the cybercriminals future access to the system even if the organization has remediated the vulnerability initially used to attack the system.	DevSecOps Foundation
Backlog	Requirements for a system expressed as a prioritized list of product backlog items usually in the form of 'User Stories'. The product backlog is prioritized by the Product Owner and should include functional, non-functional, and technical team-generated requirements.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Basic Security Hygiene	A common set of minimum-security practices that must be applied to all environments without exception. Practices include basic network security (firewalls and monitoring), hardening, vulnerability and patch management, logging and monitoring, basic policies and enforcement (may be implemented under a "policies as code" approach), and identity and access management.	DevSecOps Foundation
Batch Sizes	Refers to the volume of features involved in a single code release.	DevOps Leader, Value Stream Management Foundation
Bateson Stakeholder Map	A tool for mapping stakeholder's engagement with the initiative in progress.	DevOps Leader
Behavior Driven Development (BDD)	Test cases are created by simulating an EUT's externally observable inputs, and outputs. Example tool: Cucumber.	Continuous Delivery Ecosystem Foundation, Value Stream Management Foundation, DevOps Engineering Foundation

Beyond Budgeting	A management model that looks beyond command-and-control towards a more empowered and adaptive state.	DevOps Leader
Black-Box	Test case only uses knowledge of externally observable behaviors of an EUT.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Blameless post mortems	A process through which engineers whose actions have contributed to a service incident can give a detailed account of what they did without fear of punishment or retribution.	Site Reliability Engineering, DevOps Engineering Foundation
Blast Radius	Used for impact analysis of service incidents. When a particular IT service fails, the users, customers, other dependent services that are affected.	Site Reliability Engineering
Blue/Green Testing or Deployments	Taking software from the final stage of testing to live production using two environments labeled Blue and Green. Once the software is working in the green environment, switch the router so that all incoming requests go to the green environment - the blue one is now idle.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Bug	An error or defect in software that results in an unexpected or system-degrading condition.	DevSecOps Foundation, DevOps Engineering Foundation
Bureaucratic Culture	Bureaucratic organizations are likely to use standard channels or procedures which may be insufficient in a crisis (Westrum).	DevOps Leader
Bursting	Public cloud resources are added as needed to temporarily increase the total computing capacity of a private cloud.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Business Case	Justification for a proposed project or undertaking on the basis of its expected commercial benefit.	DevOps Leader

Business Continuity	Business continuity is an organization's ability to ensure operations and core business functions are not severely impacted by a disaster or unplanned incident that takes critical services offline.	Site Reliability Engineering, DevOps Engineering Foundation
Business Transformation	Changing how the business functions. Making this a reality means changing culture, processes, and technologies in order to better align everyone around delivering on the organization's mission.	DevSecOps Foundation
Business Value	In management, an informal term that includes all forms of value that determine the health and well-being of the firm in the long run.	DevOps Leader, Value Stream Management Foundation
Cadence	Flow or rhythm of events.	DevOps Foundation, DevOps Leader, DevSecOps Foundation
CALMS Model	Considered the pillars or values of DevOps: Culture, Automation, Lean, Measurement, Sharing (as put forth by John Willis, Damon Edwards, and Jez Humble).	DevOps Foundation, DevOps Engineering Foundation
Canary Testing	A canary (also called a canary test) is a push of code changes to a small number of end-users who have not volunteered to test anything. Similar to incremental rollout, it is where a small portion of the user base is updated to a new version first. This subset, the canaries, then serve as the proverbial “canary in the coal mine”. If something goes wrong then a release is rolled back and only a small subset of the users are impacted.	Continuous Delivery Ecosystem Foundation, Site Reliability Engineering, DevOps Engineering Foundation
Capacity	An estimate of the total amount of engineering time available for a given Sprint.	Certified Agile Service Manager, DevOps Engineering Foundation
Capacity Test	The purpose of the test is to determine if the EUT can handle expected loads such as number of users, number of sessions, aggregate bandwidth.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation

Capture-Replay	Test cases are created by capturing live interactions with the EUT, in a format that can be replayed by a tool. E.g. Selenium	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Cardinality	A mathematical term identifying the number of possible values or the number of unique values within a data dimension.	Observability Foundation
Carrots	Positive incentives, for encouraging and rewarding desired behaviors.	DevSecOps Foundation
Causal Observability	When you can see which events caused what other events. It is level 3 of the Observability Maturity Model and requires correlation of telemetry (metrics, events, logs, traces) and topology at every moment in time.	Observability Foundation
Chain of Goals	A method designed by Roman Pichler of ensuring that goals are linked and shared at all levels through the product development process.	DevOps Leader
Change	Addition, modification, or removal of anything that could have an effect on IT services. (ITIL® definition)	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Change Failure Rate	A measure of the percentage of failed/rolled back changes.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Change Fatigue	A general sense of apathy or passive resignation towards organizational changes by individuals or teams.	DevSecOps Foundation
Change Lead Time	A measure of the time from a request for a change to the delivery of the change.	DevOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation

Change Leader Development Model	Jim Canterucci's model for five levels of change leader capability.	DevOps Leader
Change Management	The process that controls all changes throughout their lifecycle. (ITIL definition)	DevOps Foundation, DevOps Leader, DevSecOps Foundation, DevOps Engineering Foundation
Change Management (Organizational)	An approach to shifting or transitioning individuals, teams & organizations from a current state to a desired future state. Includes the process, tools & techniques to manage the people-side of change to achieve the required business outcome(s).	DevOps Leader
Change-based Test Selection Method	Tests are selected according to a criterion that matches attributes of tests to attributes of the code that is changed in a build.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Chaos Engineering	The discipline of experimenting on a software system in production in order to build confidence in the system's capability to withstand turbulent and unexpected conditions.	Site Reliability Engineering, DevOps Engineering Foundation, Observability Foundation
Chapter Lead	A squad line manager in the Spotify model who is responsible for traditional people management duties is involved in day-to-day work, and grows individual and chapter competence.	DevOps Leader
Chapters	A small family of people having similar skills and who work within the same general competency area within the same tribe. Chapters meet regularly to discuss challenges and areas of expertise in order to promote sharing, skill development, re-use, and problem-solving.	DevOps Leader

ChatOps	An approach to managing technical and business operations (coined by GitHub) that involves a combination of group chat and integration with DevOps tools. Example tools include Atlassian HipChat/Stride, Microsoft Teams, Slack.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, Continuous Testing Foundation, Site Reliability Engineering, Value Stream Management Foundation, DevOps Engineering Foundation
Check-in	The action of submitting a software change into a system version management system.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
CIA (Confidentiality, Integrity, Availability) Triad	An information security model that can guide an organization's efforts and policies aimed at keeping its data secure. The three letters in "CIA Triad" stand for Confidentiality, Integrity, and Availability.	Observability Foundation
CI/CD Pipeline Scanning	This pipeline execution process provides an opportunity to observe potential security issues you could have in your images. Most CI/CD tools allow integration of additional security scanners.	Observability Foundation
CI Regression Test	A subset of regression tests that are run immediately after a software component is built. Same as Smoke Test.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Clear-Box	Same as Glass-Box Testing and White-Box Testing.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Cloud Computing	The practice of using remote servers hosted on the internet to host applications rather than local servers in a private data center.	DevSecOps Foundation, Site Reliability Engineering, DevOps Engineering Foundation
Cloud-Native	Native cloud applications (NCA) are designed for cloud computing.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation

Cloud Native Computing Foundation (CNCF)	A Linux Foundation project that supports many useful open-source projects that was founded to help advance container technology.	Observability Foundation
Cloud Networking	A subset of network monitoring which offers connectivity to applications and workloads across clouds, cloud services and so on.	Observability Foundation
Cloudbees	Cloudbees is a commercially supported proprietary automation framework tool that works with and enhances Jenkins by providing enterprise levels support and add-on functionality.	Continuous Testing Foundation, DevOps Engineering Foundation
Cluster Cost Optimization	Tools like Kubecost, Replex, Cloudability use monitoring to analyze container clusters and optimize the resource deployment model.	Site Reliability Engineering
Cluster Monitoring	Tools that let you know the health of your deployment environments running in clusters such as Kubernetes.	Site Reliability Engineering
Clustering	A group of computers (called nodes or members) work together as a cluster connected through a fast network acting as a single system.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Code Coverage	A measure of white box test coverage by counting code units that are executed by a test. The code unit may be a code statement, a code branch, or control path or data path through a code module.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Code Quality	See also static code analysis, Sonar and Checkmarks are examples of tools that automatically check the seven main dimensions of code quality – comments, architecture, duplication, unit test coverage, complexity, potential defects, language rules.	Site Reliability Engineering, DevOps Engineering Foundation
Code Repository	A repository where developers can commit and collaborate on their code. It also tracks historical versions and potentially identifies conflicting versions of the same code. Also referred to as "repository" or "repo."	DevSecOps Foundation, DevOps Engineering Foundation

Code Review	Software engineers inspect each other's source code to detect coding or code formatting errors.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Cognitive Bias	Cognitive bias is a limitation in objective thinking that is caused by the tendency for the human brain to perceive information through a filter of personal experience and preferences: a systematic pattern of deviation from norm or rationality in judgment.	DevOps Leader
Collaboration	People jointly working with others towards a common goal.	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Collaborative Culture	A culture that applies to everyone which incorporates an expected set of behaviors, language, and accepted ways of working with each other reinforcement by leadership.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Compatibility Test	Test with the purpose to determine if an EUT interoperates with another EUT such as peer-to-peer applications or protocols.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Configuration Management	Configuration management (CM) is a systems engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Conformance Test	The purpose of the test is to determine if an EUT complies with a standard.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Constraint	Limitation or restriction; something that constrains. See also *bottleneck*.	DevOps Foundation, DevSecOps Foundation

Container	A way of packaging software into lightweight, stand-alone, executable packages including everything needed to run it (code, runtime, system tools, system libraries, settings) for development, shipment, and deployment.	DevOps Foundation, DevSecOps Foundation, Site Reliability Engineering, DevOps Engineering Foundation
Container Network Security	Used to prove that any app that can be run on a container cluster with any other app can be confident that there is no unintended use of the other app or any unintended network traffic between them.	Site Reliability Engineering
Container Registry	Secure and private registry for Container images. Typically allowing for easy upload and download of images from the build tools. Docker Hub, Artifactory, Nexus are examples.	Site Reliability Engineering
Container Scanning	When building a Container image for your application, tools can run a security scan to ensure it does not have any known vulnerability in the environment where your code is shipped. Blackduck, Synopsis, Synk, Claire, and Klar are examples.	Site Reliability Engineering
Continual Service Improvement (CSI)	One of the ITIL Core publications and a stage of the service lifecycle.	DevOps Foundation
Continuous Delivery (CD)	A methodology that focuses on making sure software is always in a releasable state throughout its lifecycle.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, DevSecOps Foundation, Continuous Testing Foundation, DevOps Engineering Foundation, Observability Foundation
Continuous Delivery (CD) Architect	A person who is responsible to guide the implementation and best practices for a continuous delivery pipeline.	Continuous Delivery Ecosystem Foundation

Continuous Delivery Pipeline	A continuous delivery pipeline refers to the series of processes that are performed on product changes in stages. A change is injected at the beginning of the pipeline. A change may be new versions of code, data, or images for applications. Each stage processes the artifacts resulting from the prior stage. The last stage results in deployment to production.	Continuous Delivery Ecosystem Foundation, DevOps Foundation Course, DevOps Leader, Value Stream Management Foundation, DevOps Engineering Foundation
Continuous Delivery Pipeline Stage	Each process in a continuous delivery pipeline. These are not standard. Examples are Design: determine implementation changes; Creation: implement an unintegrated version of design changes; Integration: merge	Continuous Delivery Ecosystem Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Continuous Deployment	A set of practices that enable every change that passes automated tests to be automatically deployed to production.	DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Continuous Flow	Smoothly moving people or products from the first step of a process to the last with minimal (or no) buffers between steps.	DevOps Foundation, DevOps Leader, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Continuous Improvement	Based on Deming's Plan-Do-Check-Act, a model for ensuring ongoing efforts to improve products, processes, and services.	DevOps Foundation, DevOps Leader, DevOps Engineering Foundation
Continuous Integration (CI)	A development practice that requires developers to merge their code into trunk or master ideally at least daily and perform tests (i.e. unit, integration, and acceptance) at every code commit.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, Continuous Testing Foundation, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation

Continuous Integration Tools	Tools that provide an immediate feedback loop by regularly merging, building, and testing code. Example tools include Atlassian Bamboo, Jenkins, Microsoft VSTS/Azure DevOps, TeamCity.	DevOps Foundation, DevOps Leader, Value Stream Management Foundation, DevOps Engineering Foundation
Continuous Monitoring (CM)	This is a class of terms relevant to logging, notifications, alerts, displays, and analysis of test results information.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Continuous Testing (CT)	This is a class of terms relevant to the testing and verification of an EUT in a DevOps environment.	DevOps Foundation, Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Conversation Café	Conversation Cafés are open, hosted conversations in cafés as well as conferences and classrooms—anywhere people gather to make sense of our world.	DevOps Leader
Conway's Law	Organizations that design systems are constrained to produce designs that are copies of the communication structures of these organizations.	Continuous Delivery Ecosystem Foundation, DevOps Leader, DevOps Engineering Foundation
Cooperation vs. Competition	The key cultural value shift toward being highly collaborative and cooperative, and away from internal competitiveness and divisiveness.	DevSecOps Foundation
COTS	Commercial-off-the-shelf solution	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Critical Success Factor (CSF)	Something that must happen for an IT service, process, plan, project or other activity to succeed.	DevSecOps Foundation
Cultural Iceberg	A metaphor that visualizes the difference between observable (above the water) and non-observable (below the waterline) elements of culture.	DevOps Leader, Value Stream Management Foundation

Culture (Organizational Culture)	The values and behaviors that contribute to the unique psychosocial environment of an organization.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Cumulative Flow Diagram	A cumulative flow diagram is a tool used in agile software development and lean product development. It is an area graph that depicts the quantity of work in a given state, showing arrivals, time in queue, quantity in a queue, and departure.	DevOps Leader, Value Stream Management Foundation
Current State Map	A form of value stream map that helps you identify how the current process works and where the disconnects are.	DevOps Leader, Value Stream Management Foundation
Customer Reliability Engineer (CRE)	CRE is what you get when you take the principles and lessons of SRE and apply them to customers.	Site Reliability Engineering
Cycle Time	A measure of the time from the start of work to ready for delivery.	DevOps Foundation, DevOps Leader. DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Daily Scrum	Daily timeboxed event of 15 minutes or less for the Team to replan the next day of work during a Sprint.	DevOps Foundation, Value Stream Management Foundation
Dashboard	Graphical display of summarized data e.g., deployment frequency, velocity, test results. A visual representation of key metrics and data related to a system. Dashboards can be used to monitor the health and performance of a system in real-time.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, Value Stream Management Foundation, DevOps Engineering Foundation, Observability Foundation

DAST (Dynamic Application Security Testing)	Dynamic application security testing (DAST) is a process of testing an application or software product in an operating state.	DevSecOps Foundation, Site Reliability Engineering, DevOps Engineering Foundation
Data Governance	The specification of decision rights and an accountability framework to ensure the appropriate behavior in the valuation, creation, consumption and control of data and analytics.	Observability Foundation
Data Loss Protection (DLP)	Tools that prevent files and content from being removed from within a service environment or organization.	Site Reliability Engineering
DataOps	Defined as a combination of data and engineering practices. It aims in making the data useful by bringing in governance, ingestion guidelines, processing, and consuming structures.	Observability Foundation
Data Paradox	This means the more data you have, the less information you have.	Observability Foundation
Database Reliability Engineer (DBRE)	A person responsible for keeping database systems that support all user-facing services in production running smoothly.	Site Reliability Engineering
Debugging	The process of finding and fixing problems in a system. Observability tools can be used to help debug issues by providing insights into the system's behavior.	Observability Foundation
Defect Density	The number of faults found in a unit E.g. # defects per KLOC, # defects per change.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Definition of Done	A shared understanding of expectations that an Increment or backlog item must live up to.	Certified Agile Service Manager, DevOps Leader, Value Stream Management Foundation
DEI (Distributed, Immutable, Ephemeral) Triad	A new paradigm of security and stands for distributed, immutable and ephemeral.	Observability Foundation

Delivery Cadence	The frequency of deliveries. E.g. # deliveries per day, per week, etc.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Delivery Package	Set of release items (files, images, etc.) that are packaged for deployment.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Deming Cycle	A four-stage cycle for process management, attributed to W. Edwards Deming. Also called Plan-Do-Check-Act (PDCA).	DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation
Dependency Firewall	Many projects depend on packages that may come from unknown or unverified providers, introducing potential security vulnerabilities. There are tools to scan dependencies but that is after they are downloaded. These tools prevent those vulnerabilities from being downloaded to begin with.	Site Reliability Engineering
Dependency Proxy	For many organizations, it is desirable to have a local proxy for frequently used upstream images/packages. In the case of CI/CD, the proxy is responsible for receiving a request and returning the upstream image from a registry, acting as a pull-through cache.	Site Reliability Engineering
Dependency Relations	The type of relation on an escalation graph. It describes how components, broadly speaking, affect each other.	Observability Foundation
Dependency Scanning	Used to automatically find security vulnerabilities in your dependencies while you are developing and testing your applications. Synopsys, Gemnasium, Retire.js, and bundler-audit are popular tools in this area.	Site Reliability Engineering

Deployment	The installation of a specified version of software to a given environment (e.g., promoting a new build into production).	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Design for Testability	An EUT is designed with features that enable it to be tested.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Design Principles	Principles for designing, organizing, and managing a DevOps delivery operating model.	DevOps Leader
Dev	Individuals involved in software development activities such as application and software engineers.	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Developer (Dev)	An individual who has the responsibility to develop changes for an EUT. Alternate: Individuals involved in software development activities such as application and software engineers.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Development Test	Ensuring that the developer's test environment is a good representation of the production test environment.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Device Under Test (DUT)	The DUT is a device (e.g. router or switch) being tested.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
DevOps	A cultural and professional movement that stresses communication, collaboration, and integration between software developers and IT operations professionals while automating the process of software delivery and infrastructure changes. It aims at establishing a culture and environment where building, testing, and releasing software, can happen rapidly, frequently, and more reliably." (Wikipedia)	Certified Agile Service Manager, DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation

DevOps Coach	Help teams master Agile development and DevOps practices; enables productive ways of working and collaboration.	DevOps Leader
DevOps Infrastructure	The entire set of tools and facilities that make up the DevOps system. Includes CI, CT, CM, and CD tools.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
DevOps Kaizen	Kaizen is a Japanese word that closely translates to "change for better," the idea of continuous improvement—large or small—involving all employees and crossing organizational boundaries. Damon Edwards' DevOps Kaizen shows how making small, incremental improvements (little J's) has an improved impact on productivity long term.	DevOps Leader
DevOps Pipeline	The entire set of interconnected processes that make up a DevOps Infrastructure.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
DevOps Score	A metric showing DevOps adoption across an organization and the corresponding impact on delivery velocity.	Site Reliability Engineering
DevOps Toolchain	The tools needed to support a DevOps continuous development and delivery cycle from idea to value realization.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, DevSecOps Foundation, Continuous Testing Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
DevSecOps	A mindset that "everyone is responsible for security" with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation

Digital Transformation	The adoption of digital technology by a company to improve business processes, value for customers, and innovation.	DevOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Digital Value Stream	A value stream is anything that delivers a product or a service. A digital value stream is one that delivers a digital product or service.	Value Stream Management Foundation
Distributed Debugging	The process of identifying the root cause of an unexpected behavior in a distributed system.	Observability Foundation
Distributed Logging	Combines logs from many programs. It includes log aggregation.	Observability Foundation
Distributed Traces	Traces record the paths taken by requests (made by an application or end-user) as they propagate through multi-service architectures. It is a diagnostic technique that helps localize failures and performance issues within applications, especially those that may be distributed across multiple machines or processes. Tracing that spans multiple services and systems. Distributed tracing allows you to understand how requests flow through a complex system.	Observability Foundation
Distributed Version Control System (DVCS)	The software revisions are stored in a distributed revision control system (DRCS), also known as a distributed version control system (DVCS).	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
DMZ (De-Militarized Zone)	A DMZ in network security parlance is a network zone in between the public internet and internal protected resources. Any application, server, or service (including APIs) that need to be exposed externally are typically placed in a DMZ. It is not uncommon to have multiple DMZs in parallel.	DevSecOps Foundation

Domain-Agnostic Solutions	Solutions focus is on solving problems across multiple domains. It operates more broadly and works across domains, monitoring, logging, cloud, infrastructure, etc., These tools operate on vast amounts of IT data ingested from all domains/tools and they come up with models from this data to provide more accurate inferences and decisions.	Observability Foundation
Domain-Centric Solutions	Focused on solving a problem within a specific domain. Apply AIOps for a certain domain like network monitoring, log monitoring, application monitoring, or log collection.	Observability Foundation
Dynamic Analysis	Dynamic analysis is the testing of an application by executing data in real-time with the objective of detecting defects while it is in operation, rather than by repeatedly examining the code offline.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Dynamic Application Security Testing (DAST)	Dynamic application security testing (DAST) is a process of testing an application or software product in an operating state.	DevSecOps Foundation, DevOps Engineering Foundation
eBPF	Previously stood for Extended Berkeley Packet Filter (eBPF) which is a kernel technology that allows programs to run without requiring changes to the kernel source code or the addition of new modules. eBPF is useful in getting network sights and is immensely useful when instrumentation of tracing is not implemented across all components.	Observability Foundation
EggPlant	Automated function and regression testing of enterprise applications. Licensed by Test Plant.	Continuous Testing Foundation, DevOps Engineering Foundation

Elastic Infrastructure	Elasticity is a term typically used in cloud computing, to describe the ability of an IT infrastructure to quickly expand or cut back capacity and services without hindering or jeopardizing the infrastructure's stability, performance, security, governance, or compliance protocols.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
ELK (Elasticsearch, Logstash, Kibana) Stack	An acronym used to describe a stack that comprises three popular projects: Elasticsearch, Logstash, and Kibana. All of the software components of ELK are free and open source.	Observability Foundation
eNPS	Employee Net Promoter Score (eNPS) is a way for organizations to measure employee loyalty. The Net Promoter Score, originally a customer service tool, was later used internally on employees instead of customers.	DevOps Foundation, DevOps Leader
Entity Under Test (EUT)	This is a class of terms that refers to the names of types of entities that are being tested. These terms are often abbreviated to the form xUT where "x" represents a type of entity under test.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Ephemeral Elastic Infrastructure	The concept of infrastructure being transitory, existing only briefly as needed to serve the needs of a DevOps process that needs infrastructure while it is executing.	DevOps Engineering Foundation
Erickson (Stages of Psychosocial Development)	Erik Erikson (1950, 1963) proposed a psychoanalytic theory of psychosocial development comprising eight stages from infancy to adulthood. During each stage, the person experiences a psychosocial crisis which could have a positive or negative outcome for personality development.	DevSecOps Foundation
Error Budget	The error budget provides a clear, objective metric that determines how unreliable a service is allowed to be within a specific time period.	Site Reliability Engineering, DevOps Engineering Foundation

Error Budget Policies	An error budget policy enumerates the activity a team takes when they've exhausted their error budget for a particular service in a particular time period.	Site Reliability Engineering, DevOps Engineering Foundation
Error Tracking	Tools to easily discover and show the errors that the application may be generating, along with the associated data.	Site Reliability Engineering
Escalation Graph	A temporal graph, which consists of nodes and relations. The nodes of the escalation graph represent IT components at specific times.	Observability Foundation
Events	These are specific sequences of occurrences that take place within a system being monitored.	Observability Foundation
External Automation	Scripts and automation outside of a service that is intended to reduce toil.	Site Reliability Engineering
Fail Early	A DevOps tenet referring to the preference to find critical problems as early as possible in a development and delivery pipeline.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Fail Often	A DevOps tenet which emphasizes a preference to find critical problems as fast as possible and therefore frequently.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Failure Rate	Fail verdicts per unit of time.	DevOps Foundation, Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
False Negative	A test incorrectly reports a verdict of "fail" when the EUT actually passed the purpose of the test.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation

False Positive	A test incorrectly reports a verdict of "pass" when the EUT actually failed the purpose of the test.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Feature Toggle	The practice of using software switches to hide or activate features. This enables continuous integration and testing a feature with selected stakeholders.	DevOps Foundation, Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Federated Identity	A central identity used for access to a wide range of applications, systems, and services, but with a particular skew toward web-based applications. Also, often referenced as Identity-as-a-Service (IDaas). Any identity that can be reused across multiple sites, particularly via SAML or OAuth authentication mechanisms.	DevSecOps Foundation
Filtering Metrics	These are quantifiable data points around the application and/or infrastructure. Example - spike in CPU usage etc.	Observability Foundation
Filtering Traces	It limits the events collected in a trace. If a filter is not set, all events of the selected event classes are returned in the trace output. It can optimize traces for lower cost in distributed cloud-native applications.	Observability Foundation
Fire Drills	A planned failure testing process focussed on the operation of live services including service failure testing as well as communication, documentation, and other human factor testing.	Site Reliability Engineering, DevOps Engineering Foundation
Flow	How people, products, or information move through a process. Flow is the first way of The Three Ways.	DevOps Foundation, DevOps Leader, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation

Flow of Value	A form of map that shows the end-to-end value stream. This view is usually not available within the enterprise.	DevOps Leader. Value Stream Management Foundation
Framework	The backbone for plugging in tools. Launches automated tasks, collects results from automated tasks.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Freedom and Responsibility	A core cultural value that with the freedom of self-management (such as afforded by DevOps) comes the responsibility to be diligent, to follow the advice process, and to take ownership of both successes and failures.	DevSecOps Foundation
Frequency	How often an application is released.	DevOps Leader, DevOps Engineering Foundation
Functional Testing	Tests to determine if the functional operation of the service is as expected.	Site Reliability Engineering, DevOps Engineering Foundation
Future State Map	A form of value stream map that helps you develop and communicate what the target end state should look like and how to tackle the necessary changes.	DevOps Leader, Value Stream Management Foundation, DevOps Engineering Foundation
Fuzzing	Fuzzing or fuzz testing is an automated software testing practice that inputs invalid, unexpected, or random data into applications.	DevSecOps Foundation, DevOps Engineering Foundation
Gated Commits	Define and obtain consensus for the criterion of changes promoted between all CD pipeline stages such as Dev to CI stage / CI to packaging/delivery stage / Delivery to Deployment/Production stage.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Generative (DevOps) Culture	In a generative organization, alignment takes place through identification with the mission. The individual "buys into" what he or she is supposed to do and its effect on the outcome. Generative organizations tend to be proactive in getting the information to the right people by any means. necessary. (Westrum)	DevOps Leader

Generativity	A cultural view wherein long-term outcomes are of primary focus, which in turn drives investments and cooperation that enable an organization to achieve those outcomes.	DevSecOps Foundation
Glass-Box	Same as Clear-Box Testing and White-Box Testing.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Goal-seeking tests	The purpose of the test is to determine an EUT's performance boundaries, using incrementally stresses until the EUT reaches peak performance. E.g. Determine the maximum throughput that can be handled without errors.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Golden Circle	A model by Simon Sinek that emphasizes an understanding of the business' "why" before focusing on the "what" and "how".	DevOps Foundation
Golden Image	A template for a virtual machine (VM), virtual desktop, server, or hard disk drive. (TechTarget)	DevSecOps Foundation
Goleman's Six Styles of Leadership	Daniel Goleman (2002) created the Six Leadership Styles and found, in his research, that leaders used one of these styles at any one time.	DevOps Leader
Governance, Risk Management and Compliance (GRC)	A team or software platform intended for concentrating governance, compliance, and risk management data, including policies, compliance requirements, vulnerability data, and sometimes asset inventory, business continuity plans, etc. In essence, a specialized document and data repository for security governance. Or a team of people who specialize in IT/security governance, risk management, and compliance activities. Most often non-technical business analyst resources.	DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation

Gray-Box	Test cases use a limited knowledge of the internal design structure of the EUT.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
GUI testing	The purpose of the test is to determine if the graphical user interface operates as expected.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Guilds	A "community of interest" group that welcomes anyone and usually cuts across an entire organization. Similar to a Community of Practice.	DevOps Foundation, DevOps Leader
Hand Offs	The procedure for transferring the responsibility of a particular task from one individual or team to another.	DevOps Foundation, DevOps Leader, Value Stream Management Foundation
Hardening	Securing a server or infrastructure environment by removing or disabling unnecessary software, updating to known good versions of the operating system, restricting network-level access to only that which is needed, configuring logging in order to capture alerts, configuring appropriate access management, and installing appropriate security tools.	DevSecOps Foundation
Helm Chart Registry	Helm charts are what describe related Kubernetes resources. Artifactory and Codefresh support a registry for maintaining master records of Helm Charts.	Site Reliability Engineering
Heritage Reliability Engineer (HRE)	Applying the principles and practices of SRE to legacy applications and environments.	Site Reliability Engineering
High-Trust Culture	Organizations with a high-trust culture encourage good information flow, cross-functional collaboration, shared responsibilities, learning from failures and new ideas.	DevOps Foundation

Horizontal Scaling	Computing resources are scaled wider to increase the volume of processing. E.g. Add more computers and run more tasks in parallel.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Hypothesis-Backlog	A collection of requirements expressed as experiments.	Value Stream Management Foundation
Hypothesis-Driven Development (HDD)	A prototype methodology that allows product designers to develop, test, and rebuild a product until it's acceptable to the users.	Value Stream Management Foundation
Idempotent	CM tools (e.g., Puppet, Chef, Ansible, and Salt) claim that they are 'idempotent' by allowing the desired state of a server to be defined as code or declarations and automate steps necessary to consistently achieve the defined state time-after-time.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Identity	The unique name of a person, device, or the combination of both that is recognized by a digital system. Also referred to as an "account" or "user."	DevSecOps Foundation
Identity and Access Management (IAM)	Policies, procedures, and tools for ensuring the right people have the right access to technology resources.	DevSecOps Foundation
Identity as a Service (IDaaS)	Identity and access management services that are offered through the cloud or on a subscription basis.	DevSecOps Foundation
Image-based test selection method	Build images are pre-assigned test cases. Tests cases are selected for a build by matching the image changes resulting from a build.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Immersive learning	A learning approach that guides teams with coaching and practice to help them learn to work in a new way.	DevOps Leader
Immutable	An immutable object is an object whose state cannot be modified after it is created. The antonym is a mutable object, which can be modified after it is created.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation

Immutable Infrastructures	Instead of instantiating an instance (server, container, etc.), with error-prone, time-consuming patches and upgrades (i.e. mutations), replace it with another instance to introduce changes or ensure proper behavior.	Continuous Delivery Ecosystem Foundation, Site Reliability Engineering
Impact-Driven Development (IDD)	A software development methodology that takes small steps towards achieving both impact and vision.	Value Stream Management Foundation
Implementation Under Test	The EUT is a software implementation. E.g. Embedded program is being tested.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Improvement Kata	A structured way to create a culture of continuous learning and improvement. (In Japanese business, Kata is the idea of doing things the "correct" way. An organization's culture can be characterized as its Kata through its consistent role modeling, teaching and coaching.)	DevOps Foundation, Value Stream Management Foundation
Incentive model	A system designed to motivate people to complete tasks toward achieving objectives. The system may employ either positive or negative consequences for motivation.	DevSecOps Foundation
Incident	Any unplanned interruption to an IT service or reduction in the quality of an IT service. Includes events that disrupt or could disrupt the service. (ITIL definition)	DevSecOps Foundation
Incident Management	A process that restores normal service operation as quickly as possible to minimize business impact and ensure that agreed levels of service quality are maintained. (ITIL definition). Involves capturing the who, what, when of service incidents and the onward use of this data in ensuring service level objectives are being met.	DevSecOps Foundation, Site Reliability Engineering, DevOps Engineering Foundation
Incident Response	An organized approach to addressing and managing the aftermath of a security breach or attack (also known as an incident). The goal is to handle the situation in a way that limits damage and reduces recovery time and costs.	DevSecOps Foundation, Site Reliability Engineering, DevOps Engineering Foundation

Increment	Potentially shippable completed work that is the outcome of a Sprint.	Certified Agile Service Manager, DevOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Incremental Rollout	Deploying many small, gradual changes to a service instead of a few large changes. Users are incrementally moved across to the new version of the service until eventually all users are moved across. Sometimes referred to by colored environments e.g. Blue/green deployment.	Site Reliability Engineering, DevOps Engineering Foundation
Infrastructure	All of the hardware, software, networks, facilities, etc., required to develop, test, deliver, monitor and control or support IT services. The term IT infrastructure includes all of the information technology but not the associated people, processes, and documentation. (ITIL definition)	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Infrastructure as Code (IaC)	The practice of using code (scripts) to configure and manage infrastructure.	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Infrastructure Test	The purpose of the test is to verify the framework for EUT operating. E.g. verify specific operating system utilities function as expected in the target environment.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Infrastructure-as-a-Service (IaaS)	On-demand access to a shared pool of configurable computing resources.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Insights Driven	An insight-driven organization embeds analysis, data, and reasoning into the decision-making process, every day.	Value Stream Management Foundation
Instrumentation	Code that emits telemetry. The process of adding code to a system to collect data and provide observability. Instrumentation can include things like adding logging statements or implementing tracing.	Observability Foundation

Instrumenting with Agents	Any services used for monitoring. Agents sit outside the code, monitoring at the server or operating system level. It is “more complete” in the sense that it can work for pieces of the system whose code you don’t control, but less in-depth than in-app instrumentation with libraries.	Observability Foundation
Instrumenting with Libraries	Refers to the process of adding code to an application that allows it to emit information about its behavior and performance. This information can include logs, metrics, and traces which can be used to monitor, diagnose, and resolve issues in real-time.	Observability Foundation
Integrated development environment (IDE)	An integrated development environment (IDE) is a software suite that consolidates the basic tools developers need to write and test software. Typically, an IDE contains a code editor, a compiler or interpreter, and a debugger that the developer accesses through a single graphical user interface (GUI). An IDE may be a standalone application, or it may be included as part of one or more existing and compatible applications. (TechTarget)	DevSecOps Foundation, DevOps Engineering Foundation
Integrated development environment (IDE) 'lint' checks	Linting is the process of running a program that will analyze code for potential errors (e.g., formatting discrepancies, non-adherence to coding standards and conventions, logical errors).	DevSecOps Foundation
Internet of Things	A network of physical devices that connect to the internet and potentially to each other through web-based wireless services.	DevOps Foundation, DevSecOps Foundation
Internal Automation	Scripts and automation delivered as part of the service that is intended to reduce toil.	Site Reliability Engineering
INVEST	A mnemonic was created by Bill Wake as a reminder of the characteristics of a quality user story.	Certified Agile Service Manager
ISO 31000	A family of standards that provide principles and generic guidelines on risk management.	DevSecOps Foundation

Issue Management	A process for capturing, tracking, and resolving bugs and issues throughout the software development lifecycle.	DevSecOps Foundation
IT Service Management (ITSM)	Adopting a process approach towards management, focusing on customer needs and IT services for customers rather than IT systems, and stressing continual improvement. (Wikipedia)	Certified Agile Service Manager, DevOps Foundation, Site Reliability Engineering, Value Stream Management Foundation, DevOps Engineering Foundation, DevOps Engineering Foundation
iTest	Tool licensed by Spirent Communications for creating automated test cases.	Continuous Testing Foundation
ITIL	Provides a best practices framework that organizations can adapt to deliver and maintain IT services to provide optimal value for all stakeholders, including the customer.	Certified Agile Service Manager, DevOps Foundation, Site Reliability Engineering, Value Stream Management Foundation, DevOps Engineering Foundation
Jenkins	Jenkins is a freeware tool. It is the most popular master automation framework tool, especially for continuous integration task automation. Jenkins task automation centers around timed processes. Many test tools and other tools offer plugins to simplify integration with Jenkins.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Kaizen	The practice of continuous improvement.	DevOps Foundation, Value Stream Management Foundation
Kanban	Method of work that pulls the flow of work through a process at a manageable pace.	Certified Agile Service Manager, DevOps Foundation
Kanban Board	Tool that helps teams organize, visualize and manage work.	DevOps Foundation

Karpman Drama Triangle	The drama triangle is a social model of human interaction. The triangle maps a type of destructive interaction that can occur between people in conflict.	DevOps Leader
Key Metrics	Something that is measured and reported upon to help manage a process, IT service or activity.	DevOps Foundation, DevOps Leader, DevOps Engineering Foundation
Key Performance Indicator (KPI)	Key performance indicators are the critical indicators of progress toward an intended result, providing a focus for improvement, and on what matters most.	Value Stream Management Foundation
Keywords-Based	Test cases are created using pre-defined names that reference programs useful for testing.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Knowledge Management	A process that ensures the right information is delivered to the right place or person at the right time to enable an informed decision.	DevOps Foundation, DevSecOps Foundation
Known Error	Problem with a documented root cause and a workaround. (ITIL definition)	DevSecOps Foundation
Kolb's Learning Styles	David Kolb published his learning styles model in 1984; his experiential learning theory works on two levels: a four-stage cycle of learning and four separate learning styles.	DevOps Leader
Kotter's Dual Operating System	John Kotter describes the need for a dual operating system that combines the entrepreneurial capability of a network with the organizational efficiency of traditional hierarchy.	DevOps Leader
Kubernetes	Kubernetes is an open-source container-orchestration system for automating application deployment, scaling, and management. It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation.	Site Reliability Engineering, DevOps Engineering Foundation, Observability Foundation

Kubler-Ross Change Curve	Describes and predicts the stages of personal and organizational reaction to major changes.	DevOps Foundation
Lab-as-a-Service (LaaS)	Category of cloud computing services that provides a laboratory allowing customers to test applications without the complexity of building and maintaining the lab infrastructure.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Laloux (Culture Models)	Frederic Laloux created a model for understanding organizational culture.	DevSecOps Foundation
Latency	Latency is the delay incurred in communicating a message, the time a message spends "on the wire" between the initial request being received e.g. by a server, and the response being received e.g. by a client.	Site Reliability Engineering, Observability Foundation
Laws of Systems Thinking	In his book, 'The Fifth Discipline', Peter Senge outlines eleven laws that will help the understanding of business systems and to identify behaviors for addressing complex business problems.	DevOps Leader, Value Stream Management Foundation
Lean	Production philosophy that focuses on reducing waste and improving the flow of processes to improve overall customer value.	Certified Agile Service Manager, DevOps Foundation, DevOps Leader, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Lean (adjective)	Spare, economical. Lacking richness or abundance.	DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Lean Canvas	Lean Canvas is a 1-page business plan template.	DevOps Leader, Value Stream Management Foundation

Lean Enterprise	An organization that strategically applies the key ideas behind lean production across the enterprise.	DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation
Lean IT	Applying the key ideas behind lean production to the development and management of IT products and services.	DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation
Lean Manufacturing	Lean production philosophy derived mostly from the Toyota Production System.	DevOps Foundation, DevSecOps Foundation
Lean Product Development	Lean Product Development, or LPD, utilizes Lean principles to meet the challenges of Product Development.	DevOps Leader
Lean Startup	A system for developing a business or product in the most efficient way possible to reduce the risk of failure.	DevOps Leader
License Scanning	Tools, such as Blackduck and Synopsis, that check that licenses of your dependencies are compatible with your application, and approve or blacklist them.	Site Reliability Engineering
Little's Law	A theorem by John Little that states that the long-term average number L of customers in a stationary system is equal to the long-term average effective arrival rate λ multiplied by the average time W that a customer spends in the system.	DevOps Leader, Value Stream Management Foundation
LoadRunner	A tool used to test applications, measuring system behavior, and performance under load. Licensed by HP.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Local Development Scanning	Security scanning here is limited to the tools used by developers to scan code when creating or modifying container images. This practice should be a “must” to avoid future security issues in the Continuous Integration process. This is the more proactive moment to apply a security scanning.	Observability Foundation

Logs	Serialized report of details such as test activities and EUT console logs. A record of events that have occurred within a system. Logs can be used to troubleshoot issues and provide insight into how a system is behaving.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation, Observability Foundation
Log Analysis	Tools that allow searches, count, and sometimes correlate data from logs.	Observability Foundation
Log Management	The collective processes and policies used to administer and facilitate the generation, transmission, analysis, storage, archiving, and ultimate disposal of the large volumes of log data created within an information system.	DevSecOps Foundation, Observability Foundation
Logging	The capture, aggregation, and storage of all logs associated with system performance including, but not limited to, process calls, events, user data, responses, error, and status codes. Logstash and Nagios are popular examples.	Site Reliability Engineering, DevOps Engineering Foundation
Logic Bomb (Slag Code)	A string of malicious code used to cause harm to a system when the programmed conditions are met.	DevSecOps Foundation
Longevity Test	The purpose of the test is to determine if a complete system performs as expected over an extended period of time	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Machine Learning	Data analysis that uses algorithms that learn from data.	DevOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Malware	A program designed to gain access to computer systems, normally for the benefit of some third party, without the user's permission	DevSecOps Foundation
Many-factor Authentication	The practice of using at least 2 factors for authentication. The two factors can be of the same class.	DevSecOps Foundation

Mean Time Between Deploys	Used to measure deployment frequency.	DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Mean Time Between Failures (MTBF)	The average time that a CI or IT service can perform its agreed function without interruption. Often used to measure reliability. Measured from when the CI or service starts working, until the time it fails (uptime). (ITIL definition)	DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Mean Time to Detect Defects (MTTD)	Average time required to detect a failed component or device.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, DevSecOps Foundation, Site Reliability Engineering, Value Stream Management Foundation, DevOps Engineering Foundation
Mean Time to Discovery	How long a vulnerability or software bug/defect exists before it's identified.	DevSecOps Foundation
Mean Time to Patch	How long it takes to apply patches to environments once a vulnerability has been identified.	DevSecOps Foundation
Mean Time to Repair/Recover (MTTR)	Average time required to repair/recover a failed component or device. MTTR does not include the time required to recover or restore service.	DevOps Foundation, DevSecOps Foundation, Site Reliability Engineering Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Mean Time to Restore Service (MTRS)	Used to measure time from when the CI or IT service fails until it is fully restored and delivering its normal functionality (downtime). Often used to measure maintainability. (ITIL definition).	DevOps Foundation, DevSecOps Foundation, Site Reliability Engineering, Value Stream Management Foundation, DevOps Engineering Foundation

MELT	An acronym that stands for Metrics, Events, Logs, and Traces. These are the essential data types for Observability. When we instrument everything and use this acronym to form a fundamental, working knowledge of connections—the relationships and dependencies within our system—as well as its detailed performance and health, exemplifies Observability.	Observability Foundation
Mental Models	A mental model is an explanation of someone's thought process about how something works in the real world.	DevOps Leader
Merge	The action of integrating software changes together into a software version management system.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Metric	Something that is measured and reported upon to help manage a process, IT service, or activity. Quantitative measurements that provide insight into how a system is behaving. Metrics can include things like CPU usage, memory usage, network traffic, and error rates.	DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Metrics	This is a class of terms relevant to measurements used to monitor the health of a product or infrastructure.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation, Observability Foundation
Microprocess	A distinct activity that can be defined, designed, implemented, and managed independently and is generally associated with a primary service management practice. A microprocess may be integrated with other service management practices.	Certified Agile Service Manager
Microprocess Architecture	A collection of integrated microprocesses that collectively perform all of the activities necessary for an end-to-end service management practice to be successful.	Certified Agile Service Manager

Microservices	A software architecture that is composed of smaller modules that interact through APIs and can be updated without affecting the entire system.	DevOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Mindset	A person's usual attitude or mental state is their mindset.	DevOps Leader
Minimum Viable Process	The least amount needed in order for this process or microprocess to meet its Definition of Done.	Certified Agile Service Manager
Minimum Viable Product	Most minimal version of a product that can be released and still provide enough value that people are willing to use it.	DevOps Leader
Mock Object	Mock is a method/object that simulates the behavior of a real method/object in controlled ways. Mock objects are used in unit testing. Often a method under a test calls other external services or methods within it. These are called dependencies.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Model	Representation of a system, process, IT service, CI, etc. that is used to help understand or predict future behavior. In the context of processes, models represent pre-defined steps for handling specific types of transactions.	DevSecOps Foundation, DevOps Engineering Foundation
Model-Based	Test cases are automatically derived from a model of the entity under test. Example tool: Tricentis	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Monitoring	The use of a hardware or software component to monitor the system resources and performance of a computer service.	Site Reliability Engineering, Value Stream Management Foundation, DevOps Engineering Foundation, Observability Foundation
Monitoring Tools	Tools that allow IT organizations to identify specific issues of specific releases and to understand the impact on end-users.	DevOps Leader, Value Stream Management Foundation, DevOps Engineering Foundation

Monolithic	A software system is called "monolithic" if it has a monolithic architecture, in which functionally distinguishable aspects (for example data input and output, data processing, error handling, and the user interface) are all interwoven, rather than containing architecturally separate components.	Continuous Delivery Ecosystem Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Multi-factor Authentication	The practice of using 2 or more factors for authentication. Often used synonymously with 2-factor Authentication.	DevSecOps Foundation
Multi-cloud	Multi-cloud DevOps solutions provide on-demand multi-tenant access to development and test environments.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Network Reliability Engineer (NRE)	Someone who applies a reliability engineering approach to measure and automate the reliability of networks.	Site Reliability Engineering, DevOps Engineering Foundation
Neuroplasticity	Describes the ability of the brain to form and reorganize synaptic connections, especially in response to learning or experience or following injury.	DevOps Leader
Neuroscience	The study of the brain and nervous system.	DevOps Leader
Non-functional requirements	Requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors or functions (e.g., availability, reliability, maintainability, supportability); qualities of a system.	DevOps Foundation, DevOps Engineering Foundation
Non-functional tests	Defined as a type of service testing intending to check non-functional aspects such as performance, usability, and reliability of a software service.	Site Reliability Engineering
Object Under Test (OUT)	The EUT is a software object or class of objects.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation

Observability	Observability is focused on externalizing as much data as you can about the whole service allowing us to infer what the current state of that service from its external behavior.	Site Reliability Engineering, Value Stream Management Foundation, DevOps Engineering Foundation, Observability Foundation
Observability Backend	The data access layer to include storage and UI (user interface). There are three main components that the backend is responsible for: to include data storage, query service and visualization.	Observability Foundation
Observability Center of Excellence	The highest level of observability patterns within an organization with a unit that focuses on best practices and standards. Support, education, and training is also provided to individuals and teams within the organization.	Observability Foundation
Observability Driven Design (ODD)	Enables Observability of a system throughout the entire development cycle; it's about encouraging developers to wrap their code with breadcrumbs that can be traced and increase logging and events at the coding level. It’s an approach to shift left observability to the earliest stage of the software development life cycle.	Observability Foundation
Observability Enablement	An introductory level of observability patterns within an organization controlled by experienced DevOps Engineers or SREs. Usually, this enablement process collects and documents best practices along with tool management guidance.	Observability Foundation
Observability Maturity Model	A model that defines four distinct levels in the evolution of observability.	Observability Foundation
Objectives and Key Results (OKRs)	Objectives and key results is a goal-setting framework used by individuals, teams, and organizations to define measurable goals and track their outcomes.	Value Stream Management Foundation

On-call	Being on-call means someone being available during a set period of time, and being ready to respond to production incidents during that time with appropriate urgency.	Site Reliability Engineering
Open Source	Software that is distributed with its source code so that end-user organizations and vendors can modify it for their own purposes.	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
OpenTelemetry (OTel)	The de facto open standard for instrumentation. It is the project that provides a set of APIs, SDKs, libraries, and agents to enable developers to instrument their application once and send correlated traces and metrics to a destination of their choice. It supports 11 languages (so far) and automatic instrumentation agents, including Java, Javascript, Python, .NET, and more.	Observability Foundation
Open Telemetry Protocol (OTLP)	A general-purpose telemetry data delivery protocol designed in the scope of OpenTelemetry project. The specification describes the encoding, transport, and delivery mechanism of telemetry data between telemetry sources, intermediate nodes such as collectors and telemetry backends.	Observability Foundation
Operations (Ops)	Individuals involved in the daily operational activities needed to deploy and manage systems and services such as quality assurance analysts, release managers, system and network administrators, information security officers, IT operations specialists, and service desk analysts.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Operations Management	The function that performs the daily activities needed to deliver and support IT services and the supporting IT infrastructure at the agreed levels. (ITIL)	DevSecOps Foundation

Ops	Individuals involved in the daily operational activities needed to deploy and manage systems and services such as quality assurance analysts, release managers, system and network administrators, information security officers, IT operations specialists, and service desk analysts.	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Orchestration	An approach to building automation that interfaces or "orchestrates" multiple tools together to form a toolchain.	DevOps Foundation, DevSecOps Foundation
Organization Culture	A system of shared values, assumptions, beliefs, and norms that unite the members of an organization.	DevOps Leader, DevOps Engineering Foundation
Organization Model	For DevOps, an approach that is not a dominator hierarchy but instead a Distributed Autonomous Organization (DAO).	DevOps Leader, Value Stream Management Foundation
Organizational Change	Efforts to adapt the behavior of humans within an organization to meet new structures, processes, or requirements.	DevOps Foundation, DevSecOps Foundation
OS Virtualization	A method for splitting a server into multiple partitions called "containers" or "virtual environments" in order to prevent applications from interfering with each other.	DevOps Foundation
Outcome	Intended or actual results.	DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation
Outcome Mapping	A methodology for planning, monitoring, and evaluating development initiatives in order to bring about sustainable change.	Value Stream Management Foundation
Package Registry	A repository for software packages, artifacts, and their corresponding metadata. Can store files produced by an organization itself or for third-party binaries. Artifactory and Nexus are amongst the most popular.	Site Reliability Engineering

Pages	Something for creating supporting web pages automatically as part of a CI/CD pipeline.	Site Reliability Engineering
Patch	A software update designed to address (mitigate/remediate) a bug or weakness.	DevSecOps Foundation
Patch management	The process of identifying and implementing patches.	DevSecOps Foundation
Pathological Culture	Pathological cultures tend to view information as a personal resource, to be used in political power struggles (Westrum).	DevOps Leader, Site Reliability Engineering
Penetration Testing	An authorized simulated attack on a computer system that looks for security weaknesses, potentially gaining access to the system's features and data.	DevSecOps Foundation, DevOps Engineering Foundation
People Changes	Focuses on changing attitudes, behaviors, skills, or performance of employees.	DevOps Leader
Performance Test	The purpose of the test is to determine an EUT meets its system performance criterion or to determine what a system's performance capabilities are.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Plan-Do-Check-Act	A four-stage cycle for process management and improvement attributed to W. Edwards Deming. Sometimes called the Deming Cycle or PDCA.	Certified Agile Service Manager, DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation
Platform-as-a-Service (PaaS)	Category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Plugin	A pre-programmed integration between an orchestration tool and other tools. For example, many tools offer plugins to integrate with Jenkins.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation

Policies	Formal documents that define boundaries in terms of what the organization may or may not do as part of its operations.	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Policy as Code	The notion that security principles and concepts can be articulated in code (e.g., software, configuration management, automation) to a sufficient degree that the need for an extensive traditional policy framework is greatly reduced. Standards and guidelines should be implemented in code and configuration, automatically enforced, and automatically reported on in terms of compliance, variance, or suspected violations.	DevSecOps Foundation, DevOps Engineering Foundation
Practice	A complete end-to-end capability for managing a specific aspect of service delivery (e.g. changes, incidents, service levels).	Certified Agile Service Manager, Value Stream Management Foundation, DevOps Engineering Foundation
Practice Backlog	A prioritized list of everything that needs to be designed or improved for a practice including current and future requirements.	Certified Agile Service Manager
Practice/Microprocess Planning	A high-level event to define the goals, objectives, inputs, outcomes, activities, stakeholders, tools, and other aspects of a practice or microprocess. This meeting is not timeboxed.	Certified Agile Service Manager
Predictive Analytics	A branch of advanced analytics that makes predictions about future outcomes. It can be used to mine performance data to identify potential problems. This insight can provide guidance to SRE.	Observability Foundation
Pre-Flight	This is a class of terms that refers to names of activities and processes that are conducted on an EUT prior to integration into the trunk branch.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Priority	The relative importance of an incident, problem, or change; based on impact and urgency. (ITIL definition)	DevSecOps Foundation

Privileged Access Management (PAM)	Technologies that help organizations provide secured privileged access to critical assets and meet compliance requirements by securing, managing, and monitoring privileged accounts and access. (Gartner)	DevSecOps Foundation, DevOps Engineering Foundation
Problem	The underlying cause of one or more incidents. (ITIL definition)	DevOps Foundation, DevSecOps Foundation
Process	A structured set of activities designed to accomplish a specific objective. A process takes inputs and turns them into defined outputs. Related work activities that take specific inputs and produce specific outputs that are of value to a customer.	Certified Agile Service Manager, DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Process Changes	Focuses on changes to standard IT processes, such as software development practices, ITIL processes, change management, approvals, etc.	DevOps Leader
Process Owner	A role accountable for the overall quality of a process. It may be assigned to the same person who carries out the Process Manager role, but the two roles may be separate in larger organizations. (ITIL definition)	DevSecOps Foundation
Process Standup	A time-boxed event of 15 minutes to inspect progress towards the Sprint Goal and identify impediments as quickly as possible.	Certified Agile Service Manager
Processing Time	The period during which one or more inputs are transformed into a finished product by a manufacturing or development procedure. (Business Dictionary)	DevOps Leader, Value Stream Management Foundation
Product Backlog	Prioritized list of functional and non-functional requirements for a system usually expressed as user stories.	DevOps Foundation
Product Owner	An individual responsible for maximizing the value of a product and for managing the product backlog. Prioritizes, grooms, and owns the backlog. Gives the squad purpose.	DevOps Foundation, DevOps Leader, Value Stream Management Foundation, DevOps Engineering Foundation

Programming-Based	Test cases are created by writing code in a programming language. E.g. JavaScript, Python, TCL, Ruby	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Project to Product	Changing ways of working from a large batch, waterfall project led approach, to a small batch, agile product (or value stream) approach.	Value Stream Management Foundation
Provision Platforms	Tools that provide platforms for provisioning infrastructure (e.g., Puppet, Chef, Salt).	DevOps Leader
Psychological Safety	Psychological safety is a shared belief that the team is safe for interpersonal risk-taking.	DevOps Leader
QTP	Quick Test Professional is a functional and regression test automation tool for software applications. Licensed by HP.	Continuous Testing Foundation
Quality Management	Tools that handle test case planning, test execution, defect tracking (often into backlogs), severity, and priority analysis. CA's Agile Central	Site Reliability Engineering
Ranorex	GUI test automation framework for testing of desktop, web-based and mobile applications. Licensed by Ranorex.	Continuous Testing Foundation, DevOps Engineering Foundation
Ransomware	Encrypts the files on a user's device or a network's storage devices. To restore access to the encrypted files, the user must pay a "ransom" to the cybercriminals, typically through a tough-to-trace electronic payment method such as Bitcoin.	DevSecOps Foundation
RASP	Runtime Application Self-Protection	DevSecOps Foundation
RED (Rate, Errors, Duration) Metrics View	A monitoring acronym that stands for rate, errors and duration. It is a subset of the golden signals but provides basic steps for monitoring web applications.	Observability Foundation

Registry Scanning	It's required that scanning tools are connected to the container registry or provided by itself. It's an important step to observe security vulnerabilities that could be found in container Images and in other Cloud Native artifacts.	Observability Foundation
Regression testing	The purpose of the test is to determine if a new version of an EUT has broken some things that worked previously.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Regulatory compliance testing	The purpose of the test is to determine if an EUT conforms to specific regulatory requirements. E.g. verify an EUT satisfies government regulations for consumer credit card processing.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Release	Software that is built, tested, and deployed into the production environment.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Release Acceptance Criteria	Measurable attributes for a release package that determine whether a release candidate is acceptable for deployment to customers.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Release Candidate	A release package that has been prepared for deployment, may or may not have passed the Release.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Release Governance	Release Governance is all about the controls and automation (security, compliance, or otherwise) that ensure your releases are managed in an auditable and trackable way, in order to meet the need of the business to understand what is changing.	Site Reliability Engineering, DevOps Engineering Foundation

Release Management	The process that manages releases and underpins Continuous Delivery and the Deployment Pipeline.	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Release Orchestration	Typically a deployment pipeline used to detect any changes that will lead to problems in production. Orchestrating other tools will identify performance, security, or usability issues. Tools like Jenkins and Gitlab CI can "orchestrate" releases.	Site Reliability Engineering, DevOps Engineering Foundation
Relevance	A Continuous Testing tenet which emphasizes a preference to focus on the most important tests and test results	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Reliability	A measure of how long a service, component, or CI can perform its agreed function without interruption. Usually measured as MTBF or MTBSI. (ITIL definition)	DevOps Foundation, DevSecOps Foundation, Site Reliability Engineering, DevOps Engineering Foundation
Reliability Test	The purpose of the test is to determine if a complete system performs as expected under stressful and loaded conditions over an extended period of time.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Remediation	Action to resolve a problem found during DevOps processes. E.g. Roll-back changes for an EUT change that resulted in a CT test case fail verdict.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Remediation Plan	A plan that determines the actions to take after a failed change or release. (ITIL definition)	DevOps Foundation, DevSecOps Foundation
Request for Change (RFC)	Formal proposal to make a change. The term RFC is often misused to mean a change record, or the change itself. (ITIL definition)	DevOps Foundation
Requirements Management	Tools that handle requirements definition, traceability, hierarchies & dependency. Often also handles code requirements and test cases for requirements.	Site Reliability Engineering

Resilience	Building an environment or organization that is tolerant to change and incidents.	DevSecOps Foundation, Site Reliability Engineering
Response Time	Response time is the total time it takes from when a user makes a request until they receive a response.	Site Reliability Engineering
REST	Representation State Transfer. The software architecture style of the worldwide web.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Restful API	Representational state transfer (REST) or RESTful services on a network, such as HTTP, provide scalable interoperability for requesting systems to quickly and reliably access and manipulate textual representations (XML, HTML, JSON) of resources using stateless operations (GET, POST, PUT, DELETE, etc.).	Continuous Delivery Ecosystem Foundation
RESTful interface testing	The purpose of the test is to determine if an API satisfies its design criterion and the expectations of the REST architecture.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Return on Investment (ROI)	The difference between the benefit achieved and the cost to achieve that benefit, expressed as a percentage.	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Review Apps	Allow code to be committed and launched in real-time – environments are spun up to allow developers to review their application.	Site Reliability Engineering
Rework	The time and effort required to correct defects (waste).	DevOps Leader

Risk	A possible event that could cause harm or loss or affect an organization's ability to achieve its objectives. The management of risk consists of three activities: identifying risks, analyzing risks, and managing risks. The probable frequency and probable magnitude of future loss. Pertains to a possible event that could cause harm or loss or affect an organization's ability to execute or achieve its objectives.	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Risk Event	A possible event that could cause harm or loss or affect an organization's ability to achieve its objectives. The management of risk consists of three activities: identifying risks, analyzing risks, and managing risks.	DevOps Leader
Risk Management Process	The process by which "risk" is contextualized, assessed and treated. From ISO 31000: 1) Establish context, 2) Assess risk, 3) Treat risk (remediate, reduce or accept).	DevSecOps Foundation
Robot Framework	TDD framework created and supported by Google.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Role	Set of responsibilities, activities, and authorities granted to a person or team. A role is defined by a process. One person or team may have multiple roles. A set of permissions assigned to a user or group of users to allow a user to perform actions within a system or application.	DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Role-based Access Control (RBAC)	An approach to restricting system access to authorized users.	DevSecOps Foundation
Roll-back	Software changes which have been integrated are removed from the integration.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Root Cause Analysis (RCA)	Actions take to identify the underlying cause of a problem or incident.	DevOps Foundation, DevSecOps Foundation

Root Span	Created to represent the whole activity. It is the only span without a parent and represents how a service request is started, and when the response is sent.	Observability Foundation
Rugged Development (DevOps)	Rugged Development (DevOps) is a method that includes security practices as early in the continuous delivery pipeline as possible to increase cybersecurity, speed, and quality of releases beyond what DevOps practices can yield alone.	DevOps Foundation, DevSecOps Foundation
Rugged DevOps	Rugged DevOps is a method that includes security practices as early in the continuous delivery pipeline as possible to increase cybersecurity, speed, and quality of releases beyond what DevOps practices can yield alone.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Runbooks	A collection of procedures necessary for the smooth operation of a service. Previously manual in nature they are now usually automated with tools like Ansible.	Site Reliability Engineering
Runtime Application Self Protection (RASP)	Tools that actively monitor and block threats in the production environment before they can exploit vulnerabilities.	DevSecOps Foundation, Site Reliability Engineering
Sample Rate	The number of spans represented by this particular span, when some portion of spans are dropped to save storage. This makes spans and traces meaningfully countable without keeping each one.	Observability Foundation
Sampling	The concept of selecting a few elements from a large collection and learning about the entire collection by extrapolating from the selected set. It helps to reduce the volume of tracing data.	Observability Foundation
Sanity Test	A very basic set of tests that determine if a software is functional at all.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Saturation	The degree to which the resource has extra work which it can't service, often queued (backlog).	Observability Foundation

Scalability	Scalability is a characteristic of a service that describes its capability to cope and perform under an increased or expanding load.	Site Reliability Engineering, DevOps Engineering Foundation
Scaled Agile Framework (SAFE)	A proven, publicly available, framework for applying Lean-Agile principles and practices at an enterprise scale.	DevOps Foundation, DevOps Engineering Foundation
SCARF Model	A summary of important discoveries from neuroscience about the way people interact socially.	DevOps Leader
Scheduling	Scheduling: the process of planning to release changes into production.	DevOps Leader
Scrum	A simple framework for effective team collaboration on complex projects. Scrum provides a small set of rules that create "just enough" structure for teams to be able to focus their innovation on solving what might otherwise be an insurmountable challenge. (Scrum.org)	Certified Agile Service Manager, DevOps Foundation, DevOps Engineering Foundation
Scrum Pillars	Pillars that uphold the Scrum framework include Transparency, Inspection, and Adaption.	Certified Agile Service Manager, Value Stream Management Foundation
Scrum Team	A self-organizing, cross-functional team that uses the Scrum framework to deliver products iteratively and incrementally. The Scrum Team consists of a Product Owner, Developers, and a Scrum Master.	DevOps Foundation, DevOps Engineering Foundation
Scrum Values	A set of fundamental values and qualities underpinning the Scrum framework: commitment, focus, openness, respect and courage.	Certified Agile Service Manager
Scrum Master	An individual who provides process leadership for Scrum (i.e., ensures Scrum practices are understood and followed) and who supports the Scrum Team by removing impediments.	Certified Agile Service Manager, DevOps Foundation

Secret Detection	Secret Detection aims to prevent that sensitive information, like passwords, authentication tokens, and private keys are unintentionally leaked as part of the repository content.	Site Reliability Engineering, DevOps Engineering Foundation
Secrets Management	Secrets management refers to the tools and methods for managing digital authentication credentials (secrets), including passwords, keys, APIs, and tokens for use in applications, services, privileged accounts, and other sensitive parts of the IT ecosystem.	Site Reliability Engineering, DevSecOps Foundation
Secure Automation	Secure automation removes the chance of human error (and wilful sabotage) by securing the tooling used across the delivery pipeline.	Site Reliability Engineering
Security (Information Security)	Practices intended to protect the confidentiality, integrity, and availability of computer system data from those with malicious intentions.	DevOps Foundation, DevSecOps Foundation
Security as Code	Automating and building security into DevOps tools and practices, making it an essential part of toolchains and workflows.	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation, DevOps Engineering Foundation
Security tests	The purpose of the test is to determine if an EUT meets its security requirements. An example is a test that determines if an EUT processes login credentials properly.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Selenium	Popular open-source tool for software testing GUI and web applications.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Self-healing	Self-healing means the ability of services and underlying environments to detect and resolve problems automatically. It eliminates the need for manual human intervention.	, DevOps Engineering Foundation

Serverless	A code execution paradigm where no underlying infrastructure or dependencies are needed, moreover, a piece of code is executed by a service provider (typically cloud) who takes over the creation of the execution environment. Lambda functions in AWS and Azure Functions are examples.	Site Reliability Engineering, DevOps Engineering Foundation
Service	Enables the ability to do something when and how it is needed or desired. It enables its customers to achieve their objectives more efficiently and/or more effectively than they could without the service.	Certified Agile Service Manager, DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Service Desk	Single point of contact between the service provider and the users. Tools like Service Now are used for managing the lifecycle of services as well as internal and external stakeholder engagement.	DevOps Foundation
Service Level Agreement (SLA)	Written agreement between an IT service provider and its customer(s) that defines key service targets and responsibilities of both parties. An SLA may cover multiple services or customers. (ITIL definition)	DevOps Engineering Foundation, Site Reliability Engineering, Observability Foundation
Service Level Indicator (SLI)	SLI's are used to communicate quantitative data about services, typically to measure how the service is performing against an SLO.	Site Reliability Engineering, DevOps Engineering Foundation, Observability Foundation
Service Level Objective (SLO)	An SLO is a goal for how well a product or service should operate. SLO's are set based on what an organization is expecting from a service.	Site Reliability Engineering, DevOps Engineering Foundation, Observability Foundation
Service Map	Used to discover and track IT applications and components in an organization, outlining how each connects to business services.	Observability Foundation

Seven Pillars of DevOps	Seven distinct "pillars" provide a foundation for DevOps systems which include Collaborative Culture, Design for DevOps, Continuous Integration, Continuous Testing, Continuous Delivery and Deployment, Continuous Monitoring, and Elastic Infrastructure and Tools.	Continuous Delivery Ecosystem Foundation
Shift Left	An approach that strives to build quality into the software development process by incorporating testing early and often. This notion extends to security architecture, hardening images, application security testing, and beyond.	DevOps Foundation, DevSecOps Foundation, DevOps Engineering Foundation
Signals	Forms of communication that are emitted from traces, logs, events and metrics.	Observability Foundation
SilkTest	Automated function and regression testing of enterprise applications. Licensed by Borland.	Continuous Testing Foundation, , DevOps Engineering Foundation
Simian Army	The Simian Army is a suite of failure-inducing tools designed by Netflix. The most famous example is Chaos Monkey which randomly terminates services in production as part of a Chaos Engineering approach.	Site Reliability Engineering, DevOps Engineering Foundation
Single Point of Failure (SPOF)	A single point of failure (SPOF) is a part of a system that, if it fails, will stop the entire system from working.	DevOps Foundation
Site Reliability Engineering (SRE)	The discipline that incorporates aspects of software engineering and applies them to infrastructure and operations problems. The main goals are to create scalable and highly reliable software systems.	Site Reliability Engineering, DevOps Engineering Foundation
Smoke Test	A basic set of functional tests that are run immediately after a software component is built. Same as CI Regression Test.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation

Snapshot	Report of pass/fail results for a specific build.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Snippets	Stored and shared code snippets to allow collaboration around specific pieces of code. Also allows code snippets to be used in other code-bases. BitBucket and GitLab allow this.	Site Reliability Engineering
SOAP	Simple Object Access Protocol (SOAP) is an XML-based messaging protocol for exchanging information among computers.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Software Composition Analysis	A tool that checks for libraries or functions in source code that have known vulnerabilities.	DevSecOps Foundation, DevOps Engineering Foundation
Software Defined Networking (SDN)	Software-Defined Networking (SDN) is a network architecture approach that enables the network to be intelligently and centrally controlled, or 'programmed,' using software applications.	Site Reliability Engineering, DevOps Engineering Foundation
Software Delivery Lifecycle (SDLC)	The process used to design, develop and test high quality software.	DevOps Leader, Site Reliability Engineering, DevOps Engineering Foundation
Software Version Management System	A repository tool which is used to manage software changes. Examples are: Azure DevOps, BitBucket, Git, GitHub, GitLab, VSTS.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Software-as-a-Service (SaaS)	Category of cloud computing services in which software is licensed on a subscription basis.	DevOps Foundation, Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation

Source Code Tools	Repositories for controlling source code for key assets (application and infrastructure) as a single source of truth.	DevOps Foundation, DevOps Leader, Value Stream Management Foundation, DevOps Engineering Foundation
Span	A single operation with a name, timestamp, duration, tags, and other metadata. It represents a single unit of work within a trace.	Observability Foundation
Span ID	A short universally unique identifier (UUID) to identify a span, usually 16 hex characters.	Observability Foundation
Spotify Squad Model	An organizational model that helps teams in large organizations behave like startups and be nimble.	DevOps Foundation, DevOps Leader
Sprint	A period of 2-4 weeks during which an increment of product work is completed.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, Value Stream Management Foundation
Sprint (Scrum)	A time-boxed iteration of work during which an increment of product functionality is implemented.	DevOps Foundation
Sprint Backlog	Subset of the backlog that represents the work that must be completed to realize the Sprint Goal.	Certified Agile Service Manager, DevOps Foundation
Sprint Goal	The purpose and objective of a Sprint, often expressed as a business problem that is going to be solved.	Certified Agile Service Manager, DevOps Foundation, Value Stream Management Foundation
Sprint Planning	A 4 to 8-hour time-boxed event that defines the Sprint Goal, the increment of the Practice Backlog that will be completed during the Sprint, and how it will be completed.	Certified Agile Service Manager

Sprint Retrospective	A 1.5 to 3-hour time-boxed event during which the Team reviews the last Sprint and identifies and prioritizes improvements for the next Sprint.	Certified Agile Service Manager
Sprint Review	A time-boxed event of 4 hours or less where the Team and stakeholders inspect the work resulting from the Sprint and update the Practice Backlog.	Certified Agile Service Manager
Spyware	Software that is installed in a computer without the user's knowledge and transmits information about the user's computer activities over back to the threat agent.	DevSecOps Foundation
Squads	A cross-functional, co-located, autonomous, self-directed team.	DevOps Leader
Stakeholder	Person who has an interest in an organization, project or IT service. Stakeholders may include customers, users and suppliers. (ITIL definition).	DevOps Foundation, DevSecOps Foundation
Stability	The sensitivity a service has to accept changes and the negative impact that may be caused by system changes. Services may have reliability, in that if functions over a long period of time, but may not be easy to change and so does not have stability.	Site Reliability Engineering, DevOps Engineering Foundation
Standard Change	Pre-approved, low risk change that follows a procedure or work instruction. (ITIL definition)	DevOps Foundation, DevSecOps Foundation
Static Application Security Testing (SAST)	A type of testing that checks source code for bugs and weaknesses.	DevSecOps Foundation, DevOps Engineering Foundation
Static Code Analysis	The purpose of the test is to detect source code logic errors and omissions such as memory leaks, unutilized variables, unutilized pointers.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Status Page	Service pages that easily communicate the status of services to customers and users.	Site Reliability Engineering
Sticks	Negative incentives, for discouraging or punishing undesired behaviors.	DevSecOps Foundation

Storage Security	A specialty area of security that is concerned with securing data storage systems and ecosystems and the data that resides on these systems.	Site Reliability Engineering
Stormstack	A commercial orchestration tool based on event triggers instead of time-based.	Continuous Testing Foundation
StoStaKee	This stands for stop, start, and keep: this is an interactive time-boxed exercise focused on past events.	DevOps Leader
Strategic Sprint	A <4 week timeboxed Sprint during which strategic elements that were defined during Practice Planning are completed so that the Team can move on to designing the activities of the process.	Certified Agile Service Manager
Stream-Aligned Team	A team aligned to a single, valuable stream of work; this might be a single product or service, a single user story, or a single user persona.	Value Stream Management Foundation
Structural Changes	Changes in the hierarchy of authority, goals, structural characteristics, administrative procedures, and management systems.	DevOps Leader
Structured Logs	Forms of logs that are computer-readable with many attributes (not just messages).	Observability Foundation
Supplier	External (third party) supplier, manufacturer, or vendor responsible for supplying goods or services that are required to deliver IT services.	DevOps Foundation
Synthetic Monitoring	Synthetic monitoring (also known as active monitoring, or semantic monitoring) runs a subset of an application's automated tests against the system on a regular basis. The results are pushed into the monitoring service, which triggers alerts in case of failures.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
System of Record	A system of record is the authoritative data source for a data element or data entity.	DevOps Foundation, DevSecOps Foundation

System Test	The purpose of the test is to determine if a complete system performs as expected in its intended configurations.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
System Under Test (SUT)	The EUT is an entire system. E.g. Bank teller machine is being tested.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Tag-Based Test Selection Method	Tests and Code modules are pre-assigned tags. Tests are selected for a build matching pre-assigned tags.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Target Operating Model	A description of the desired state of the operating model of an organization.	DevOps Leader, Value Stream Management Foundation
Teal Organization	An emerging organizational paradigm that advocates a level of consciousness including all previous world views within the operations of an organization.	DevOps Leader
Team Dynamics	A measurement of how a team works together. Includes team culture, communication styles, decision-making ability, trust between members, and the willingness of the team to change.	DevOps Leader
Team Topologies	An approach to organizing business and technology teams for fast flow, providing a practical, step-by-step, adaptive model for organizational design and team interaction.	Value Stream Management Foundation, DevOps Engineering Foundation
Techno-Economic Paradigm Shifts	Techno-economic paradigm shifts are at the core of the general, innovation-based theory of economic and societal development as conceived by Carlota Perez.	DevOps Leader, Value Stream Management Foundation
Telemetry	Telemetry is the collection of measurements or other data at remote or inaccessible points and their automatic transmission to receiving equipment for monitoring.	Site Reliability Engineering, DevOps Engineering Foundation, Observability Foundation

Temporal Relations	Describes timelines for each component. It always only connects components with their past self each time a significant event occurs (in terms of the problem).	Observability Foundation
Test Architect	Person who has responsibility for defining the overall end-to-end test strategy for an EUT.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Test Artifact Repository	Database of files used for testing.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Test Campaign	A test campaign may include one or more test sessions.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test Case	Set of test steps together with data and configuration information. A test case has a specific purpose to test at least one attribute of the EUT.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test Creation Methods	This is a class of test terms that refers to the methodology used to create test cases.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation

Test-Driven Development (TDD)	Test-driven development (TDD) is a software development process in which the developer writes a test before composing code. They then follow this process: 1. Write the test 2. Run the test and any others that are relevant and see them fail 3. Write the code 4. Run test(s) 5. Refactor code if needed 6. Repeat Unit level tests and/or application tests are created ahead of the code that is to be tested.	Continuous Delivery Ecosystem Foundation, DevOps Foundation, Continuous Testing Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Test Duration	The time it takes to run a test. E.g. # hours per test	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Test Environment	The test environment refers to the operating system (e.g. Linus, windows version, etc.), the configuration of software (e.g. parameter options), dynamic conditions (e.g. CPU and memory utilization), and physical environment (e.g. power, cooling) in which the tests are performed.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test Fast	A CT tenet referring to accelerated testing.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test Framework	A set of processes, procedures, abstract concepts, and environments in which automated tests are designed and implemented.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test Harness	A tool which enables the automation of tests. It refers to the system test drivers and other supporting tools that requires to execute tests. It provides stubs and drivers which are small programs that interact with the software under test.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation

Test Hierarchy	This is a class of terms describes the organization of tests into groups.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test Methodology	This class of terms identifies the general methodology used by a test. Examples are White Box, Black Box	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test result repository	Database of test results.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Test Results Trend-based	A matrix of correlation factors correlates test cases and code modules according to test results (verdict).	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test Roles	This class of terms identifies general roles and responsibilities for people relevant to testing.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test Script	Automated test case. A single test script may be implemented with one or more test cases depending on the data.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test Selection Method	This class of terms refers to the method used to select tests to be executed on a version of an EUT.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test Session	Set of one or more test suites that are run together on a single build at a specific time.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation

Test Suite	Set of test cases that are run together on a single build at a specific time.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test Trend	History of verdicts.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test Type	The class which indicates the purpose of the test.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Test Version	The version of files used to test a specific build.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Tester	An individual who has the responsibility to test a system or service.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Testing Tools	Tools that verify code quality before passing the build.	DevOps Leader, DevOps Engineering Foundation
The Advice Process	Any person deciding must seek advice from everyone meaningfully affected by the decision and people with expertise in the matter. Advice received must be taken into consideration, though it does not have to be accepted or followed. The objective of the advice process is not to form a consensus, but to inform the decision-maker so that they can make the best decision possible. Failure to follow the advice process undermines trust and unnecessarily introduces risk to the business.	DevSecOps Foundation

The Checkbox Trap	The situation wherein an audit-centric perspective focuses exclusively on "checking the box" on compliance requirements without consideration for overall security objectives.	DevSecOps Foundation
The Power of TED	The Power of TED* offers an alternative to the Karpman Drama Triangle with its roles of Victim, Persecutor, and Rescuer. The Empowerment Dynamic (TED) provides the antidote roles of Creator, Challenger, and Coach and a more positive approach to life's challenges.	DevOps Leader
The Sprint	A period of <4 weeks during which an increment of work is completed.	Certified Agile Service Manager, Value Stream Management Foundation
The Three Pillars of Empiricism	Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation.	Value Stream Management Foundation
The Three Ways	Key principles of DevOps – Flow, Feedback, Continuous experimentation, and learning.	DevOps Foundation, DevSecOps Foundation, Site Reliability Engineering, Value Stream Management Foundation, DevOps Engineering Foundation
Theory of Constraints	Methodology for identifying the most important limiting factor (i.e., constraint) that stands in the way of achieving a goal and then systematically improving that constraint until it is no longer the limiting factor.	DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation
Thomas Kilmann Inventory (TKI)	Measures a person's behavioral choices under certain conflict situations.	DevOps Foundation
Threat Agent	An actor, human or automated, that acts against a system with intent to harm or compromise that system. Sometimes also called a "Threat Actor."	DevSecOps Foundation

Threat Detection	Refers to the ability to detect, report, and support the ability to respond to attacks. Intrusion detection systems and denial-of-service systems allow for some level of threat detection and prevention.	DevSecOps Foundation
Threat Intelligence	Information pertaining to the nature of a threat or the actions a threat may be known to be perpetrating. May also include "indicators of compromise" related to a given threat's actions, as well as a "course of action" describing how to remediate the given threat action.	DevSecOps Foundation
Threat Modeling	A method that ranks and models potential threats so that the risk can be understood and mitigated in the context of the value of the application(s) to which they pertain.	DevSecOps Foundation
Three Pillars of Observability	The three types of telemetry which include logs, traces and metrics.	Observability Foundation
Time-series data	A series of data points collected over time. Time-series data provide insight into how a system is behaving over time. Data that is collected over time and can be used to analyze trends and patterns in a system's behavior.	Observability Foundation
Time to Insight Actioned	The time between having an idea, delivering it to the customer, learning and actioning the insight from that learning.	Value Stream Management Foundation
Time to Learning	The time between conceiving an idea and learning how it was received based on customer feedback.	Value Stream Management Foundation
Time to Market	The period of time between when an idea is conceived and when it is available to customers.	DevOps Leader, Value Stream Management Foundation, DevOps Engineering Foundation
Time to Value	The measure of the time it takes for the business to realize value from a feature or service.	DevOps Foundation, DevSecOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation

Time Tracking	Tools that allow for time to be tracked, either against individual issues or other work or project types.	Site Reliability Engineering
Time Travel Topology	Topology map of an IT environment that keeps changing while the problem is happening. A change is often the cause of a problem so having time helps.	Observability Foundation
Timebox	The maximum duration of a Scrum event.	Certified Agile Service Manager
Toil	A kind of work tied to running a production service that tends to be manual, repetitive, automatable, tactical, devoid of enduring value.	Site Reliability Engineering, DevOps Engineering Foundation
Tool	This class describes tools that orchestrate, automate, simulate and monitor EUT's and infrastructures.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Toolchain	A philosophy that involves using an integrated set of complimentary task-specific tools to automate an end-to-end process (vs. a single-vendor solution).	DevOps Foundation, DevOps Engineering Foundation
Topology	A map that describes the set of relationships and dependencies between discrete components in an environment.	Observability Foundation
Touch Time	In a Lean Production system the touch time is the time that the product is actually being worked on, and value is being added.	DevOps Leader, Value Stream Management Foundation
Traces	A collection of operations that represents a unique transaction handled by an application and its constituent services. The process of following a request or transaction through a system to understand how it is processed, in data that is visualized for a person. Tracing can be used to identify performance bottlenecks and diagnose issues.	Observability Foundation
Trace ID	A generated identifier (usually 32 hex characters) to identify a particular trace. Every span in the trace includes the same Trace ID.	Observability Foundation

Trace Propagation	The mechanism that moves trace context (Trace ID and Span ID) between services and processes. Between processes, it allows the downstream service to continue the trace.	Observability Foundation
Tracing	Tracing provides insight into the performance and health of a deployed application, tracking each function or microservice which handles a given request.	Site Reliability Engineering, DevOps Engineering Foundation
Traffic Volume	The amount of data sent and received by visitors to a service (e.g. a website or API).	Site Reliability Engineering
Training From the Back of the Room	An accelerated learning model in line with agile values and principles using the 4Cs instructional design "map" (Connection, Concept, Concrete Practice, Conclusion).	DevOps Leader
Transformational Leadership	A leadership model in which leaders inspire and motivate followers to achieve higher performance by appealing to their values and sense of purpose, facilitating wide-scale organizational change (State of DevOps Report, 2017).	DevOps Leader
Tribe Lead	A senior technical leader that has broad and deep technical expertise across all the squads' technical areas. A group of squads working together on a common feature set, product, or service is a tribe in Spotify's definitions.	DevOps Leader
Tribes	A collection of squads with a long-term mission that work on/in a related business capability.	DevOps Leader
Trojan (horses)	Malware that carries out malicious operations under the appearance of a desired operation such as playing an online game. A Trojan horse differs from a virus because the Trojan binds itself to non-executable files, such as image files, audio files whereas a virus requires an executable file to operate.	DevSecOps Foundation

Trunk	The primary source code integration repository for a software product.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation
Unit Test	The purpose of the test is to verify code logic.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
Usability Test	The purpose of the test is to determine if humans have a satisfactory experience when using an EUT.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation
USE (Utilization, Errors, Duration) Metrics	A methodology for analyzing the performance of any system. It directs the construction of a checklist, which for server analysis can be used for quickly identifying resource bottlenecks or errors.	Observability Foundation
User	Consumer of IT services. Or, the identity asserted during authentication (aka username).	DevOps Foundation, DevSecOps Foundation
User and Entity Behavior Analytics (UEBA)	A machine learning technique to analyze normal and “abnormal” user behavior with the aim of preventing the latter.	Site Reliability Engineering
User Interface (UI)	The point of human-computer interaction and communication.	Observability Foundation
User Story	A brief statement used to describe a requirement from a user’s perspective. User stories are used to facilitate communication, planning, and negotiation activities between the stakeholders and the Agile Service Management Team.	Certified Agile Service Manager, DevOps Engineering Foundation
Value Added Time	The amount of time spent on an activity that creates value (e.g., development, testing).	DevOps Leader
Value Cycle	The lifecycle stages of the value stream from ideation to value realization.	Value Stream Management Foundation
Value Efficiency	Being able to produce value with the minimum amount of time and resources.	DevOps Leader

Value Stream	All of the activities needed to go from a customer request to a delivered product or service.	DevOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Value Stream Map	Visually depicts the end-to-end flow of activities from the initial request to value creation for the customer.	Certified Agile Service Manager, Value Stream Management Foundation, DevOps Engineering Foundation
Value Stream Mapping	A lean tool that depicts the flow of information, materials, and work across functional silos with an emphasis on quantifying waste, including time and quality.	DevOps Foundation, Value Stream Management Foundation, DevOps Engineering Foundation
Value Stream Management	Value Stream Management is a combination of people, processes, and technology that maps, optimizes, visualizes, measures, and governs business value flow through heterogeneous software delivery pipelines from idea through development and into production.	Site Reliability Engineering, Value Stream Management Foundation DevOps Engineering Foundation
Value Stream Management Platform	Software that manages value streams.	Value Stream Management Foundation, DevOps Engineering Foundation
Variable Speed IT	An approach where traditional and digital processes co-exist within an organization while moving at their own speed.	DevOps Foundation
Velocity	The measure of the quantity of work done in a pre-defined interval. The amount of work an individual or team can complete in a given amount of time.	Certified Agile Service Manager, DevOps Foundation, DevSecOps Foundation, Site Reliability Engineering
Verdict	Test result classified as Fail, Pass, or Inconclusive.	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation

Version control tools	Ensure a 'single source of truth' and enable change control and tracking for all production artifacts.	DevOps Foundation, DevOps Engineering Foundation, DevOps Engineering Foundation
Vertical Scaling	Computing resources are scaled higher to increase processing speed e.g. using faster computers to run more tasks faster.	Continuous Testing Foundation, DevOps Engineering Foundation
Virus (Computer)	Malicious executable code attached to a file that spreads when an infected file is passed from system to system that could be harmless (but annoying), or it could modify or delete data.	DevSecOps Foundation
Visualization	The process of presenting data in a visual format to help engineers understand and analyze a system's behavior. Visualization can include graphs, charts, and dashboards.	Observability Foundation
Voice of the Customer (VOC)	A process that captures and analyzes customer requirements and feedback to understand what the customer wants.	DevOps Foundation
Vulnerability	A weakness in a design, system, or application that can be exploited by an attacker.	DevSecOps Foundation, DevOps Engineering Foundation
Vulnerability Intelligence	Information describing a known vulnerability, including affected software by version, the relative severity of the vulnerability (for example, does it result in an escalation of privileges for a user role, or does it cause a denial of service), the exploitability of the vulnerability (how easy/hard it is to exploit), and sometimes current rate of exploitation in the wild (is it being actively exploited or is it just theoretical). This information will also often include guidance on what software versions are known to have remediated the described vulnerability.	DevSecOps Foundation
Vulnerability management	The process of identifying and remediating vulnerabilities.	DevSecOps Foundation

DevOps Glossary of Terms

Wait Time	The amount of time wasted on waiting for work (e.g., waiting for development and test infrastructure, waiting for resources, waiting for management approval).	DevOps Leader, Value Stream Management Foundation, DevOps Engineering Foundation
Waste (Lean Manufacturing)	Any activity that does not add value to a process, product or service.	Certified Agile Service Manager, DevOps Foundation, DevOps Leader, Value Stream Management Foundation
Water-scrum-fall	A hybrid approach to application lifecycle management that combines waterfall and Scrum development can complete in a given amount of time.	Continuous Delivery Ecosystem Foundation
Waterfall (Project Management)	A linear and sequential approach to managing software design and development projects in which progress is seen as flowing steadily (and sequentially) downwards (like a waterfall).	Certified Agile Service Manager, Continuous Delivery Ecosystem Foundation, DevOps Foundation
Weakness	An error in software that can be exploited by an attacker to compromise the application, system, or the data contained therein. Also called a vulnerability.	DevSecOps Foundation
Web Application Firewall (WAF)	Tools that examine traffic being sent to an application and can block anything that looks malicious.	Site Reliability Engineering
Web IDE	Tools that have a web client integrated development environment. Enables developer productivity without having to use a local development tool.	Site Reliability Engineering
Westrum (Organization Types)	Ron Westrum developed a typology of organizational cultures that includes three types of organizations: Pathological (power-oriented), Bureaucratic (rule-oriented) and Generative (performance-oriented).	DevSecOps Foundation, Site Reliability Engineering
White-Box Testing (or Clear-, Glass-, Transparent-Box Testing or Structural Testing)	Test cases use extensive knowledge of the internal design structure or workings of an application, as opposed to its functionality (i.e. Black-Box Testing).	Continuous Delivery Ecosystem Foundation, Continuous Testing Foundation, DevOps Engineering Foundation

Whitelisting	Application whitelisting is the practice of specifying an index of approved software applications that are permitted to be present and active on a computer system.	Continuous Delivery Ecosystem Foundation, DevOps Engineering Foundation
Wicked Questions	Wicked questions are used to expose the assumptions which shape our actions and choices. They are questions that articulate the embedded, and often contradictory assumptions, we hold about an issue, a problem or a context.	DevOps Leader
Wiki	Knowledge sharing can be enabled by using tools like Confluence which create a rich Wiki of content	Site Reliability Engineering
Wilber's Quadrants	A model that recognises four modes of general approach for human beings. Two axes are used: on one axis people tend towards individuality OR collectivity.	DevOps Leader
Work in Progress (WIP)	Any work that has been started but has not been completed.	DevOps Foundation, Value Stream Management Foundation
Workaround	A temporary way to reduce or eliminate the impact of incidents or problems. May be logged as a known error in the Known Error Database. (ITIL definition).	DevOps Foundation, DevSecOps Foundation
World Café	Is a structured conversational process for knowledge sharing in which groups of people discuss a topic at several tables, with individuals switching tables periodically and getting introduced to the previous discussion at their new table by a "table host".	DevOps Leader
Worms (Computer)	Worms replicate themselves on a system by attaching themselves to different files and looking for pathways between computers. They usually slow down networks and can run by themselves (where viruses need a host program to run).	DevSecOps Foundation

W3C Standard	W3C standards define an **Open Web Platform** for application development that has the unprecedented potential to enable developers to build rich interactive experiences, powered by vast data stores, that are available on any device.	Observability Foundation

Observability Foundation^{SM} Course: Value Added Resources

This document provides links to articles and videos related to the Observability Foundation course from the DevOps Institute. This information is provided to enhance your understanding of Observability concepts and terms and is not examinable. Of course, there is a wealth of other videos, blogs and case studies on the web.
We welcome suggestions for additions.

Videos Featured in the Course

Module	Title & Description	Link
Module 1: Exploring Observability	What is Observability? by Jason Yee (06:02)	https://youtu.be/orsxOxQNzDQ?t=71
Module 2: Pillars of Observability	Examine a Trace in Jaeger by Jessica Kerr (03:01)	https://youtu.be/OILhs8G6hIO
Module 3: Open Source Landscape	OpenTelemetry Components by "Is it Observable" (12:09-16:20)	https://youtu.be/dfpGU9al_i4?t=729
Module 4: Service Maps and Topology	Why a Time-Traveling Topology is Crucial for Doing Effective RCA by Lodewijk Bogaards (1:04)	https://youtu.be/vdeczj_fuHg
Module 5: DataOps Helps Get Observability Right	Artificial Intelligence in Operations (AIOps) by Stylianos Kampakis (11:26)	https://www.youtube.com/watch?v=GSS_rTXkpFU&t=203s
Module 6: Building Observability with AIOps	Observability in Action: A Conversation with Helen Beal (04:06)I	https://youtu.be/ucnZEuwN9OM?t=382
Module 7: Security and Networking with Observability	Security Observability with Dan Gordon by SANS Cloud Security (1:01:30)	https://youtu.be/-npISgT-3KE?t=2460
Module 8: Observability Practices for DevOps and SRE	Burn Your Dashboards The Case for SLO First Monitoring by Austin Parker (05:38)	https://www.youtube.com/watch?v=5maHQiEIGgY

Case Stories Featured in the Course

Company	Module	Link
EMA	Module 1: Exploring Observability	https://www.solarwinds.com/resources/case-study/$11b-european-retailer-reduces-opex-by-$478000-moving-to-solarwinds-full-stack-observability/delivery
Guance Cloud	Module 2: Pillars of Observability	https://github.com/GuanceCloud
GitHub	Module 3: Open Source Landscape	https://lightstep.com/case-studies/github
Accenture	Module 4: Topology and Service Maps	https://youtu.be/3VAyoIgnM_o?list=TLGGwejRCd6n4lQxNjEyMjAyMg&t=202
FlipKart	Module 5: DataOps Helps Get Observability Right	https://iide.co/case-studies/flipkart-marketing-strategy-case-study/
ServiceNow	Module 6: Building Observability with AIOps	https://www.servicenow.com/content/dam/servicenow-assets/public/en-us/doc-type/resource-center/ebook/learn-how-AIOps-delivers-high-performance-business-services.pdf
Flexport	Module 7: Security and Networking with Observability	Uptycs Security Observability Case Study
Medallia	Module 8: Observability Practices for DevOps and SRE	https://www.youtube.com/watch?v=giBO7OPz3K8

Additional Links of Interest

Module 1	Video Monitoring vs. Observability	https://www.youtube.com/watch?v=1k0aOwsQZpw
Module 1	Observability vs. Monitoring: What's the Difference in DevOps *	https://www.instana.com/blog/observability-vs-monitoring/
Module 1	Accelerating Digital Transformation In Turbulent Times	https://bit.ly/accelturbulent
Module 1	AppDynamics Provides Comprehensive Observability	https://bit.ly/obs-serverless
Module 1	Central Nervous System (and AIOps strategy)	https://www.appdynamics.com/central-nervoussystem/
Module 1	Slowdown is the New Outage	https://bit.ly/slowdownoutage
Module 1	Doctors Without Borders	https://bit.ly/msf_dwb
Module 1	Video reference for Module 1 A guide to move from Monitoring to Observability	https://www.brighttalk.com/webcast/18840/490280?utm_source=brighttalk-portal&utm_medium=web&utm_content=AIOps%20Instana&utm_term=search-result-2&utm_campaign=webcasts-search-results-feed
Module 1	5 Key Observability Trends 2022	https://www.infoworld.com/article/3648053/5-key-observability-trends-for-2022.html
Module 1	5 Key Observability Trends 2022	https://thenewstack.io/5-key-observability-trends-for-2022/
Module 1	Advanced Observability Teams See Big Efficiency Gains	https://www.apmdigest.com/advanced-observability-2
Module 1	Observability Driven Design (ODD)	https://devops.com/observability-driven-development-from-software-development-to-devops-and-beyond/
Module 1	Case Study for Module 1 (alternative)	https://terazo.com/case-studies/how-a-healthcare-company-improved-customer-service-by-boosting-observability/

Module 5	7 steps to get Data right	https://content.dataiku.com/data-privacy/7-steps-DS-project
Module 2	Observability versus monitoring in software development	https://circleci.com/blog/observability-vs-monitoring/?utm_source=google&utm_medium=sem&utm_campaign=sem-google-dg—emea-en-dsa-maxConv-auth-nb&utm_term=g_-_c__dsa_&utm_content=&gclid=Cj0KCQiAg_KbBhDLARIsANx7wAxh4MHT0vj3vUTQ4M8gHjs5-PpRS8a1EgDoT4Ehsz-GZjFijSa_GCUaArvzEALw_wcB
Module 2	Demystifying M.E.L.T. - the key data for business observability	https://itbrief.com.au/story/demystifying-m-e-l-t-the-key-data-for-business-observability#:~:text=The%20acronym%20M.E.L.T.%20is%20used,%2C%20events%2C%20logs%20and%20traces.&text=Metrics%20are%20the%20starting%20point,or%20collected%20at%20regular%20intervals.
Module 2	Magpie: Online Modelling and Performance-Aware Systems	https://www.usenix.org/legacy/publications/library/proceedings/hotos03/tech/full_papers/barham/barham_html/paper.html
Module 4	"What is Happening: Attempting to Understand Our Systems" by Beau Lyddon Observability_pipeline.pdf	"What is Happening: Attempting to Understand Our Systems" by Beau Lyddon
Module 4	The Observability Pipeline	Observability_pipeline.pdf
Module 4	Data Management – Industry Best Practices	https://cloudian.com/guides/data-protection/data-protection-and-privacy-7-ways-to-protect-user-data/
Module 5	Using Platform Engineering & AIOps AI in Ops with Stylianos Kampakis	https://youtu.be/GSS_rTXkpFU?t=203
Module 7	Securing the Software Supply Chain with Observability	https://www.rtinsights.com/observability-and-software-supply-chain-security/
Module 7	Log4j-vulnerability-highlights-the-need-for-observability	https://www.rtinsights.com/log4j-vulnerability-highlights-the-need-for-observability/

Module 7	Software-supply-chain-observability-with-grafeas-and-kritis	https://www.slideshare.net/aysylu/software-supply-chain-observability-with-grafeas-and-kritis
Module 7	Supply-chain-security/supply-chain-security-paper	https://github.com/cncf/tag-security/blob/main/supply-chain-security/supply-chain-security-paper/CNCF_SSCP_v1.pdf
Module 7	Common Vulnerabilities and Exposures	Center for Internet Security (CIS), the National Vulnerability Database https://content.microfocus.com/national-vulnerability-database-report-tb/software-vulnerabilities-cve-appsec?lx=1dKb4k&utm_source=techbeacon&utm_medium=techbeacon&utm_campaign=7014J000000dVA4QAM
Module 7	Kubernetes Breach case study	https://www.infosecurity-magazine.com/opinions/tesla-hack-cryptojacking-warning/
Module 7	Flexport Empowers DevOps Teams with Security Observability	Uptycs Security Observability Case Study
Module 4	Reducing Ticket time using 4Ts	https://youtu.be/3VAyoIgnM_o?list=TLGGwejRCd6n4lQxNjEyMjAyMg&t=202
Module 3	Implementing OpenTelemetry	https://developer.newrelic.com/opentelemetry-masterclass/hands-on/
Module 3	Implementing OpenTelemetry standards	https://storiesfromtheherd.com/opentelemetry-hands-on-instrumentation-f1b423c323c0
Module 3	GitHub integration with OpenTelemetry	https://lightstep.com/case-studies/github
Module 2	Pillars of Observability	https://youtu.be/0pxbZ10fJYY?t=11tions/tip/The-3-pillars-of-observability-Logs-metrics-and-traces
Module 2	Understanding Observability & Monitoring from a opensource standard context	https://www.youtube.com/watch?v=1k0a0wsQZpw
Module 6	SRE & Incident Response Management Runbook Automation with Damon Edwards	https://youtu.be/uyJ-FJXD5co?t=140

Module 8	Observability Trends in the near Future	https://newrelic.com/blog/best-practices/prioritize-observability#toc-show-them-the-value-of-observability-and-that-it-s-on-the-rise-
Module 8	Chaos Engineering	https://www.youtube.com/watch?t=733&v=w_Y6C0QgmL0&feature=youtu.be
Module 7	Security-observability-vs-visibility-and-monitoring	https://www.techtarget.com/searchsecurity/feature/Security-observability-vs-visibility-and-monitoring
Module 7	Next-gen-network-management-difficult-without-AIOp	https://www.techtarget.com/searchnetworking/opinion/Next-gen-network-management-difficult-without-AIOps
Module 7	Observability-and-security-monitoring-in-containers-and-containerized-applications	https://socfortress.medium.com/observability-and-security-monitoring-in-containers-and-containerized-applications-a781dae17799
Module 7	Observability-security-work-together	https://coralogix.com/blog/observability-security-work-together/
Module 7	Vulnerability-management	https://newrelic.com/blog/nerdlog/vulnerability-management-public-preview
Module 8	Observability and DevSecOps	https://youtu.be/T01ge8byOoU?t=25
Module 8	DevOps metrics to measure Observability success	https://semaphoreci.medium.com/17-devops-metrics-to-measure-success-cca66a9e79c
Module 8	Is it Observable?	https://isitobservable.io/site-reliability-engineering/non-standard-slos-beyond-availability
Module 8	Where to start your Observability journey? Observability Maturity Architecture Resource Center	https://github.com/newrelic/oma-resource-center
Module 8	Forecast of Observability Trends	https://newrelic.com/observability-forecast/2022/state-of-observability
Module 8	Observability-is-more-than-logs-metrics-traces	https://xeraa.net/talks/observability-is-more-than-logs-metrics-traces/
Module 8	Observability Engineering	https://www.oreilly.com/library/view/observability-engineering/9781492076438/ch04.html
Module 8	Observability-maturity-model	https://docs.newrelic.com/docs/new-relic-solutions/observability-maturity/operational-efficiency/observability-coe/

Module 1	Defense in Depth works for Reliability – Monzo Bank	https://youtu.be/OUYTNywPk-s?t=148
Module 8	SLO is the Proxy for Customer Happiness Home Depot	https://sre.google/workbook/slo-engineering-case-studies/
Module 8	SLO is the Proxy for Customer Happiness Kudos Engineering	https://youtu.be/KmVDkBmnb4U?t=63
Module 8	SLO is the Proxy for Customer Happiness Cloud SLAs	AWS summary of SLAs SLAs for Microsoft Azure Google Cloud Platform SLAs
Module 7	Building Secure and Reliable Systems Google Chrome Security Team	https://youtu.be/fNyT7HNKQfk?t=332 https://learning.oreilly.com/library/view/building-secure-and/9781492083115/ch19.html#onenine_case_study_chrome_security_team
Module 2	Full Stack Observability Planet Case Study	https://youtu.be/5aNeNhKNIUM?t=1283
Module 6	Using AIOps How FedEx uses AIOps to improve Operational Efficiencies’	https://opusresearch.net/wordpress/2017/10/02/case-study-how-fedex-is-leveraging-intelligent-assistants-ai-and-natural-language-understanding/
Module 6	Using AIOps How 3M Modernized IT Event Management and Alerting Using AIOPs	https://www.splunk.com/en_us/form/how-3m-modernized-it-event-management-and-alerting-with-splunk.html
Module 6	SRE & Incident Response Management HCL helps its customers better manage and monitor their modern IT environments	https://www.moogsoft.com/resources/aiops/case-study/moogsoft-hcl-technologies-case-study/
Module 2	Pitfalls on the Road to Creating a Successful Observability	https://www.usenix.org/conference/lisa17/conference-program/presentation/bisset
Module 3	Medallia’s Centralized Observability Journey	https://grafana.com/go/grafanaconline/2022/medallia-centralized-observability-use-case/
Module 3	A Collection of Best Practices for Production Services	https://sre.google/sre-book/service-best-practices/

Module 1	Differences between Synthetic Monitoring and Real user Monitoring	https://stackify.com/rum-vs-synthetic-monitoring/
Module 2	Good article on Observability and Monitoring	https://youtu.be/pY44UX8j4Pc?t=26
Module 2	Observability — A 3-Year Retrospective'	https://thenewstack.io/observability-a-3-year-retrospective/
Module 1	Monitoring and Observability — What's the Difference and Why Does It Matter?	https://thenewstack.io/monitoring-and-observability-whats-the-difference-and-why-does-it-matter/
Module 1	Observability at Google	https://www.oreilly.com/library/view/observability-at-google/0636920424239/video329911.html
Module 6	Building Self healing with AIOps	https://www.dynatrace.com/news/blog/shift-left-sre-building-self-healing-into-your-cloud-delivery-pipeline/
Module 5	The Seven Steps to Implement #DataOps	https://www.youtube.com/watch?v=muhs8zJnETM
Module 6	Proactively Detect Unusual Behavior	https://newrelic.com/lp/aiops?utm_campaign=AIOps-Emerging&utm_medium=cpc&utm_source=google&utm_content=AIO_LP&fiscal_year=FY21&quarter=Q4&gtm=OPS&program=aiops&ad_type=None&geo=EMERGING&utm_term=aiops&utm_device=c&_bt=504169602696&_bm=e&_bn=g&gclid=Cj0KCQjw5PGFBhC2ARIsAIFIMNeggGr9EMZ_f9XTeybtxJldKxxZEFp1g4oMXpPyaOq5NUBHT5FyWFUaAn6uEALw_wcB
Module 6	Gartner Market Guide for AIOps Platforms, 2021	https://digitate.com/market-guide-for-aiops-platforms/?utm_source=google&utm_medium=search&utm_campaign=corporate-aiops&utm_content=aiops-gartner-Aiops
Module 6	A successful digital transformation requires AIOps	https://sciencelogic.com/solutions/aiops
Module 6	The rise of Platform Engineering	https://softwareengineeringdaily.com/2020/02/13/setting-the-stage-for-platform-engineering/

Module 6	Top 5 things to know about Platform Engineering	https://www.techrepublic.com/article/top-5-things-to-know-about-platform-engineering/
Module 8	This is How to Use ITIL, DevOps, and SRE Best Practices	https://www.blameless.com/blog/itil-devops-sre-work-together
Module 8	Incident Response Training PagerDuty Academy	https://response.pagerduty.com/training/courses/incident_response/
Module 8	Tracking Every Release	https://codeascraft.com/2010/12/08/track-every-release/
Module 8	Accelerating SREs to On-Call and Beyond	https://sre.google/sre-book/accelerating-sre-on-call/
Module 8	Stress Testing the Cloud	https://www2.deloitte.com/us/en/pages/consulting/articles/chaos-engineering-stress-testing-the-cloud-sre-devops-cloud-value-devops-reliability-risk-management-test-management.html
Module 8	Disaster Recovery Testing (DiRT) Test Template	https://docs.google.com/document/d/1nxYuX62SvKst9YuozJCsBEWU9AltvtV9mgnd-HbzBrA/edit#heading=h.hpgc9ckwivdb
Module 8	Security Chaos Engineering	https://www.oreilly.com/library/view/security-chaos-engineering/9781492080350/
Module 8	Principles of Chaos Engineering	https://principlesofchaos.org/
Module 8	Chaos Engineering: the principles and practices	https://www.gremlin.com/community/tutorials/chaos-engineering-the-history-**principles-**and-practice/
Module 8	How to Use Chaos Engineering to Break Things Productively	https://www.infoq.com/articles/chaos-engineering-security-networking/
Module 8	GameDay involves Observability	https://queue.acm.org/detail.cfm?id=2371297
Module 8	Security Chaos Engineering	https://www.rochestersecurity.org/wp-content/uploads/2018/10/RSS2018-B1.pdf

Module 8	SRE Essentials	https://learning.oreilly.com/playlists/7b526ba0-0ba2-4d89-baac-25e9f3877d7f/
Module 8	Observability Essentials	https://learning.oreilly.com/library/view/97-things-every/9781492081487/
Module 8	Building an Observability based Organisation	https://www.slideshare.net/FranklinAngulo1/building-an-sre-organization-squarespace
Module 8	A day in the life of SRE	https://blog.newrelic.com/engineering/what-does-an-sre-do/
Module 8	The Evolving Engagement Model	https://sre.google/sre-book/evolving-sre-engagement-model/
Module 8	You Can Too! —An Observability Manual	https://youtu.be/Cg877bv_xig?t=1027
Module 8	I'm an observability SRE Lead! Now What? How to Bootstrap and Organize Your SRE Team	https://www.youtube.com/watch?v=KbKfAwPbQgk
Module 8	Psychological Safety	https://www.oreilly.com/library/view/seeking-sre/9781491978856/ch27.html#psychological_safety_in_sre

Publication References

Site Reliability Engineering	https://landing.google.com/sre/sre-book/toc/index.html
The Site Reliability Workbook	https://landing.google.com/sre/workbook/toc/ https://learning.oreilly.com/library/view/site-reliability-engineering/9781491929117/
SRE Essentials	https://learning.oreilly.com/playlists/7b526ba0-0ba2-4d89-baac-25e9f3877d7f/
Seeking SRE	https://learning.oreilly.com/library/view/seeking-sre/9781491978856/
Building Secure and Reliable Systems	https://learning.oreilly.com/library/view/building-secure-and/9781492083115/

Database Reliability Engineering	https://learning.oreilly.com/library/view/database-reliability-engineering/9781491925935/
Practical Site Reliability Engineering	https://learning.oreilly.com/library/view/practical-site-reliability/9781788839563/
Chaos Engineering	https://learning.oreilly.com/library/view/chaos-engineering/9781492043850/
Security Chaos Engineering	https://www.oreilly.com/library/view/security-chaos-engineering/9781492080350/
Real world SRE	https://www.amazon.com/Real-World-SRE-Survival-Responding-Maximizing/dp/1788628888?asin=1788628888&revisionId=&format=4&depth=1

DevOps Institute

Observability Foundation v1.0 Sample Examination with Answer Key

Feb 2023

1. Wrapping code with breadcrumbs to help enable Observability throughout the entire development lifecycle is called ________________.

A. Topology
B. Observability Success (OS)
C. Observability Driven Design (ODD)
D. Causal Observability

2. During which scanning process provides the best moment to execute a proactive security scanning on container images?

A. At CI/CD Pipeline
B. At Container Registry
C. At Local Development Environment
D. Once image has been deployed in Cloud

3. What is automatic instrumentation?

A. Code that emits telemetry, that you did not have to write
B. Enhanced data flow management
C. Breaking things in a controlled environment and through strictly-planned experiments
D. Application-specific code to add custom attributes

4. Which of the following does NOT require Time Travel Topology?

A. Very high pace of innovation from multiple teams
B. Very diverse set of runtimes, supporting software and databases
C. Very static execution environment in which services are consistently delivered
D. Very diverse set of languages

5. Which container image scanning would allow the integration of additional security scanners be added?

A. Local development
B. Registry
C. Network update
D. CI/CD pipeline

6. Which of the following are components of cloud networking?

A. Instance Network Interfaces and Routes Apathy
B. Load Balancers and Gateways
C. Network Virtual Appliances and VPNs (Virtual Private Networks)
D. All of the above

7. Which of the following team patterns displays Observability within an organization?

A. Observability Enablement
B. Observability Center of Excellence
C. Observability Teams
D. All of the above

8. All of the following are considered Pillars of Observability except _____.

A. Metrics
B. Events
C. Logs
D. Traces

9. What is the biggest project under Cloud Native Compute Foundation (CNCF)?

A. Kubernetes
B. OpenTelemetry
C. FinOps
D. MLOps

10. All of the following are Golden Signals for networking observability except

A. Connectivity
B. Network uptime
C. Packet loss
D. Security Events

11. Which instrumentation is highly context driven and uses an inside out approach?

A. Instrumenting with Agents
B. Instrumenting with Libraries
C. Instrumenting with AI/ML Ops
D. Refactoring

12. What is the most advanced level of Observability today?

A. Proactive Observability
B. Monitoring
C. Causal Observability
D. Observability

13. Modern Logging does NOT include:

A. Staggered Logging
B. Log Analysis Tools
C. Distributed Logging
D. Structured Logs

14. What are some struggles with using distributed tracing in Modern Observability?

A. No standards exist for transmitting trace data
B. Trace data is cheaper to store than logs
C. Tracing is harder to implement than logs
D. Metrics are better for debugging

15. Which of the following is NOT a reason that Observability needs DataOps?

A. Increasing data security
B. Enhanced data flow management
C. User-friendly automation
D. Authenticating users

16. What does the acronym CSOC stand for?

A. Common Security Operations Center
B. Centralized Security Operations Center
C. Cryptography Security Operations Center
D. Command Services Operations Center

17. Between processes, passing along just enough trace context to allow the downstream service to continue the trace is known as____________________?

A. Trace Distribution
B. Trace Propagation
C. Distraction
D. Storytelling

18. Which of these is a multi-layered map that shows how everything in the IT environment is related?

A. Trend Analysis
B. Causal Determination
C. Upstream-Downstream relationship
D. Topology

19. What attribute on a span identifies how long it took?

A. Span ID
B. Service Name
C. Duration
D. Trace ID

20. What is the specification of decision rights and an accountability framework to ensure the appropriate behavior in the valuation, creation and consumption and control of data and analytics?

A. Data Foundation
B. Data Sharing
C. Connected Governance
D. Data Governance

21. What does data paradox mean?

A. Without care, more data means less information
B. Without data governance, we would be clueless
C. Without data agility, full stack observability is not possible
D. None of the above

22. What AI/ML Use Case can be used to provide guidance to SRE?

A. Root Cause Analysis
B. Predictive Analytics
C. Anomaly Detection
D. Event Resolution

23. What measure is used to decide whether an application functioned well enough in a particular instance?

A. Uptime
B. Service Level Objective
C. Chaos Engineering
D. Service Level Indicator

24. Under the Observability Maturity Model what does Level 3 indicate?

A. Observability
B. Causal Observability
C. Monitoring
D. Proactive Observability with AIOps

25. What makes distributed traces special, compared to traces that aren't distributed?

A. Distributed traces show the execution across services
B. Distributed traces reveal what happened concurrently and sequentially
C. Any number of attributes can be added to distributed traces to show context
D. Distributed traces clearly identify the bottleneck

26. In SRE, what is NOT a common issue that can arise when implementing Observability?

A. Environment inconsistency (dev, prod, test are all observed differently)
B. Generating too many alerts
C. Unable to trace until the end / incomplete instrumentation
D. Logging in to multiple servers to retrieve logs

27. What are the industry metrics good practices for workloads that treats the service as a black box?

A. USE (Utilization, Saturation, Errors)
B. ACE (Availability, Conversion, Engagement)
C. RED (Rate, Errors, Duration)
D. LES (Latency, Errors, Saturation)

28. Which Pillar of Observability is the most modern?

A. Metrics
B. Logs
C. Traces
D. Events

29. Which of the following questions is NOT a goal of frontend Observability?

A. How well is it working?
B. What is wrong?
C. What part of the funnel did customers drop off in?
D. What is slow?

30. In the community-governed standard for transmission of telemetry, what does the OTLP abbreviation stand for?

A. Over The Limit Protocol
B. OpenTelemetry Protocol
C. Observability Transmission Line
D. Observable Telemetry Light Push

31. What is the name of the logs that are designed to be computer-readable with many attributes?

A. Distributed Logs
B. Structured Logs
C. Log Aggregation
D. Spans

32. What is an example of a Business Metric view?

A. Customer Churn
B. Distributed Tracing
C. RED (Rate, Errors, Duration) Metrics
D. USE (Utilization, Saturation, Errors) Metrics

33. What is a widely used information security model that can guide an organization's efforts and policies to help keep its data secure?

A. Data Governance
B. CIA Triad (Confidentiality, Integrity, Availability)
C. Automation
D. DataOps Security

34. What should the focus be on in order to build meaningful quality gates?

A. Focus on Transactional Metrics
B. Focus on Aggregational Metrics
C. Focus on Holistic Metrics
D. Focus on Business Metrics

35. Which of these people would you NOT consult with when creating a Service Level Objective?

A. Stakeholders, Project Managers
B. Engineer Manager, Software Engineers
C. Focus group of users
D. SREs

36. Which of the following is a commonly used connectivity channel to connect incoming security events into Observability solutions?

A. Security Events
B. SEIM (Security Incident and Event Management)
C. Command Line Outputs
D. API

37. Which data type would be best for auditing, with the ability to see a record of every time something happened?

A. Logs
B. Metrics
C. Monitoring
D. Traces

38. Metrics can identify which of the following?

A. The piece of code where the bug is located
B. Which calls were made concurrently
C. Which customers are experiencing high latency
D. When something is wrong

39. What are some common stack options for metrics?

A. Prometheus and Grafana
B. Prometheus and Jaeger
C. Grafana and Jaeger
D. Jaeger and Zipkin

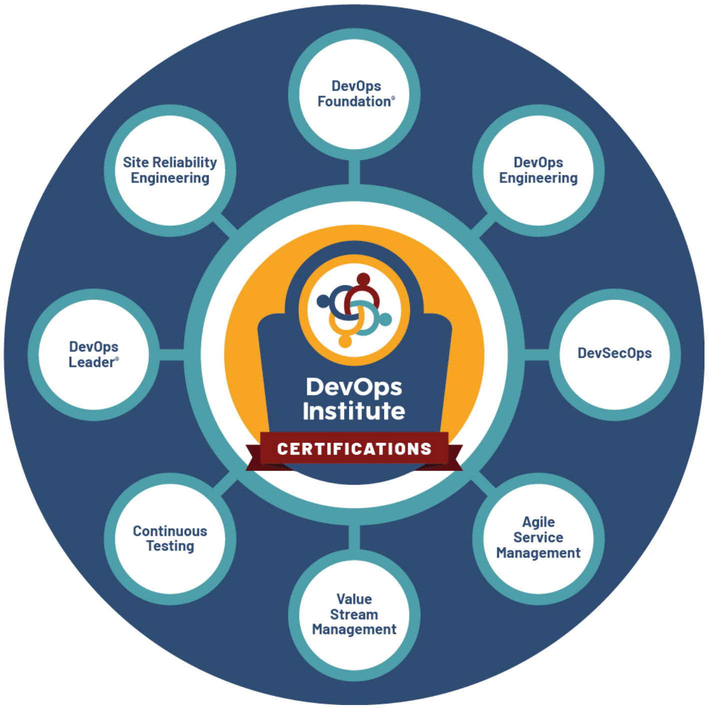
40. Which one of these codes qualifies as a Service Level Objective (SLO)?

A. 98% of checkout requests will return success within 800 ms, over 28 days
B. 99.9% uptime
C. Every request completes within 500ms
D. Failure is not an option

ANSWER KEY

Question	Correct Answer	Topic Area
1	C	1: Exploring Observability
2	C	7: Security and Networking with Observability
3	A	3: Open Source Landscape
4	C	4: Service Maps and Topology
5	D	7: Security and Networking with Observability
6	D	7: Security and Networking with Observability
7	D	8: Observability Practices for DevOps and SRE
8	B	2: Pillars of Observability
9	A	3: Open Source Landscape
10	D	7: Security and Networking with Observability
11	B	1: Exploring Observability
12	A	1: Exploring Observability
13	A	2: Pillars of Observability
14	C	3: Open Source Landscape
15	D	5: DataOps Helps Get Observability Right
16	B	6: Building Observability with AIOps
17	B	2: Pillars of Observability
18	D	4: Service Maps and Topology
19	C	2: Pillars of Observability
20	D	5: DataOps Helps Get Observability Right
21	A	5: DataOps Helps Get Observability Right

22	B	6: Building Observability with AIOps
23	D	8: Observability Practices for DevOps and SRE
24	B	1: Exploring Observability
25	A	2: Pillars of Observability
26	D	1: Exploring Observability
27	C	1: Exploring Observability
28	C	2: Pillars of Observability
29	C	3: Open Source Landscape
30	B	3: Open Source Landscape
31	B	2: Pillars of Observability
32	A	1: Exploring Observability
33	B	5: DataOps Helps Get Observability Right
34	D	6: Building Observability with AIOps
35	C	8: Observability Practices for DevOps and SRE
36	D	7: Security and Networking with Observability
37	A	2: Pillars of Observability
38	D	3: Open Source Landscape
39	A	3: Open Source Landscape
40	A	8: Observability Practices for DevOps and SRE



Your Path to

DevOps Success

DevOps Institute is dedicated to advancing the human elements of DevOps success. Our goal is to help advance careers and support emerging practices using a role-based approach to certification which focuses on the most modern competencies and hireable skills required by today's organizations adopting DevOps.

Take the next steps in your learning and certification journey to DevOps success.

Click on a certification or visit www.devopsinstitute.com/certifications to learn more.

Become a Member

Join the fastest growing global communityof DevOps practitioners and professionals and gain access to invaluable learning content, the latest news, events, emerging practices, developyour network and advance your career.

You belong.

(www.devopsinstitute.com/membership)