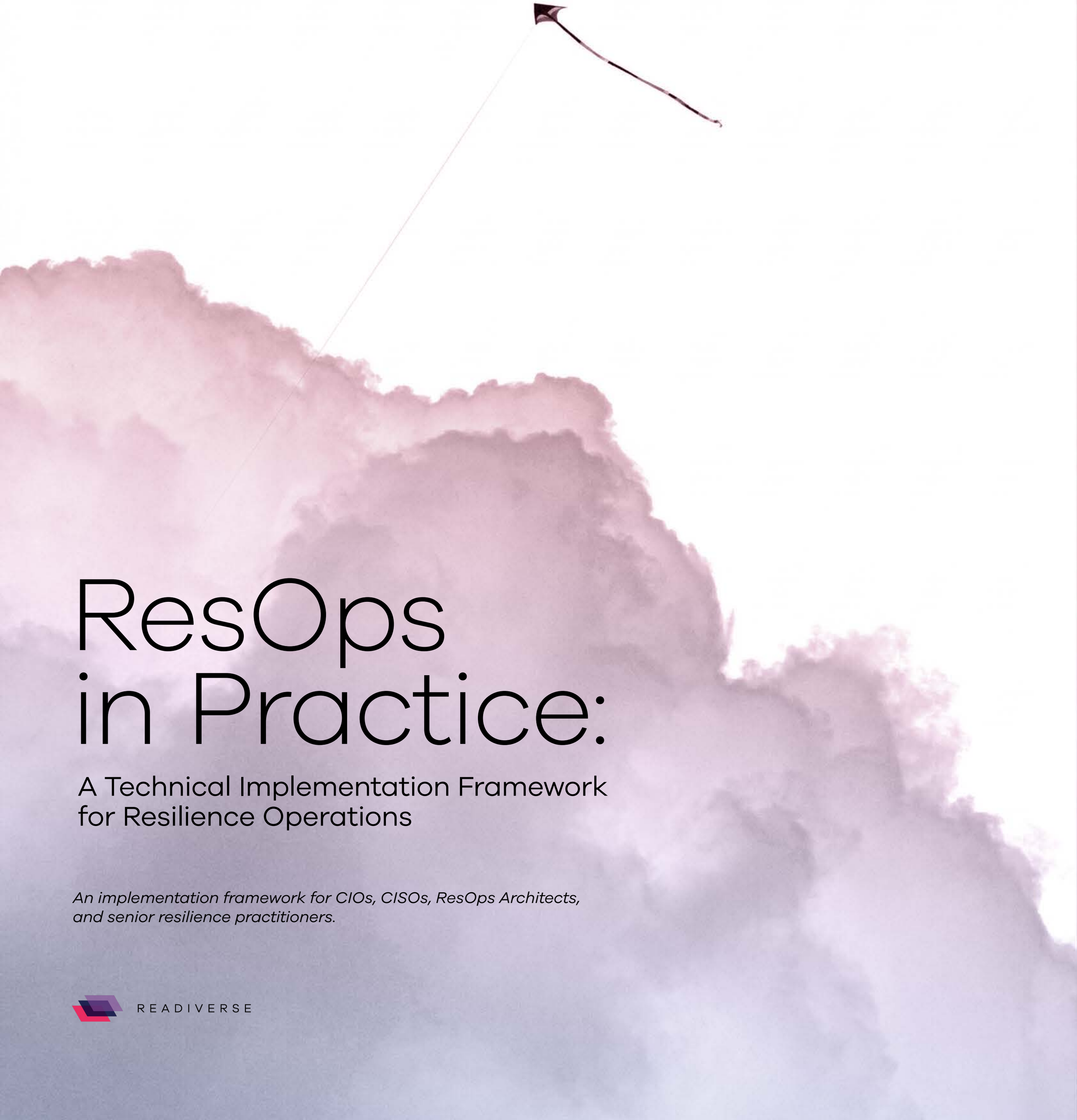

ResOps in Practice:

A Technical Implementation Framework for Resilience Operations

An implementation framework for CIOs, CISOs, ResOps Architects, and senior resilience practitioners.

READIVERSE

Executive Summary

Resilience Operations (ResOps) is the operational discipline that coordinates how an enterprise withstands disruption, limits impact on critical services, and restores those services within defined impact tolerances. It is not an extension of backup or security operations. It is a distinct, cross-functional operating model with its own disciplines, governance structure, and measurement framework.

ResOps is organized across five domains: resilience governance, recovery planning, recovery architecture, recovery assurance, and resilience measurement. Continuous improvement runs across all five domains as a cross-cutting practice – addressed via the resilience backlog through normal governance processes.

Each domain encompasses multiple disciplines – the operational practices that make each domain real. The disciplines presented in this framework represent a practitioner starting point, developed by Commvault and its partners based on where enterprises most commonly need to build capability first. ResOps is an emerging operational discipline, and we expect the community to define, expand, and refine these over time. We welcome that input.

This framework provides both the strategic context that CIOs, CISOs, and their teams need to understand why each discipline matters, and the implementation detail that architects, engineers, and practitioners need to build and operate it. The seven disciplines presented here mirror the structure of well-established reliability frameworks, offering a map that defines a mature ResOps practice.

The starting insight is simple: Resilience is not a property of having the right tools. It is a property of knowing, with evidence, that critical services can be restored within defined business tolerances when disruption occurs. The disciplines in this framework are the starting point for building that evidence.

The 5 ResOps Domains & 7 Disciplines Covered in this Framework

Domains	Disciplines
1. Resilience Governance	1. Resilience Strategy & Governance 2. Operations & Crisis Management Integration
2. Recovery Planning	3. Critical Services & Dependency Mapping
3. Recovery Architecture	4. Protection, Isolation, & Containment 5. Assured Recovery & Rebuild
4. Recovery Assurance	6. Exercises & Validation
5. Resilience Measurement	7. Measurement & Service Resilience Indicators

How to Read This Framework

Each discipline section includes an executive framing that explains why the capability matters at the leadership level, an implementation actions table that specifies the concrete steps practitioners take to build it, and supporting detail on key concepts within the discipline. The guide concludes with ResOps Maturity Model and Roles framework, which provide context for where an organization currently sits and who needs to own each part of the model.

Core Principle

ResOps assumes disruption is inevitable. Its mandate is not to prevent all failures, but to verify that when failures occur, the organization responds according to design rather than improvisation – and recovers within the tolerances that matter to customers, regulators, and the business.

1 Resilience Strategy & Governance

Define what resilience means for the business and establish the governance rhythm that makes it operational.

Resilience without governance is aspiration. The first discipline establishes the institutional infrastructure that gives every other ResOps capability its authority, funding, and accountability. Without it, recovery capabilities remain fragmented across teams, investment is driven by technical preference rather than business risk, and no one can answer the recoverability question with confidence.

The governance model has three components: a clear charter that defines scope and ownership, impact tolerances that translate business risk appetite into measurable recovery targets, and an operating rhythm that tracks progress and surfaces gaps to leadership continuously.

Implementation actions

Establish a ResOps charter, scope, and ownership – including a cross-functional ResOps lead or office with executive mandate.

Define impact tolerances for each critical service: maximum tolerable downtime, maximum tolerable data loss, and minimum viable service mode.

Align resilience objectives with enterprise risk appetite, applicable regulatory expectations, and strategic priorities

Integrate resilience investment into enterprise risk and funding processes – policy ties funding to tested impact tolerances.

Establish a monthly ResOps Council review to track backlog progress and unblock systemic issues.

Implement quarterly board reporting covering tolerance attainment, key resilience risks, and financial implications.

Publish a resilience backlog that links directly to the enterprise risk register and is tracked alongside regular delivery work.

Operationalizing the Governance Rhythm

The governance rhythm is what separates a ResOps program from a ResOps operating model. The program exists when there is a charter and a set of capabilities. The operating model exists when those capabilities are continuously measured, governed, and improved through a predictable institutional rhythm.

Monthly council reviews should cover: resilience backlog status by service, exercise findings and remediation progress, any new gaps identified through telemetry or incidents, and investment prioritization decisions. Quarterly board reporting should translate these technical findings into business terms: which critical services are recoverable to tolerance today, which are not, what the financial and regulatory exposure is, and what the path to closing the gaps looks like.

2 Critical Services & Dependency Mapping

Know exactly what must survive and what it depends on.

You cannot build recovery capability for services you haven't identified or protect dependencies you haven't mapped. The second discipline is the intelligence foundation of the entire ResOps model: a maintained, operational map of which business services are critical, what they depend on, and where the single points of failure are.

The mapping is practical, not academic. Its purpose is not exhaustive documentation but targeted investment – surfacing the dependencies and failure modes that most directly threaten recovery within defined impact tolerances.

Implementation actions

Identify critical business services and value chains (e.g., customer onboarding, trade settlement, manufacturing line operations).

Map underlying dependencies for each critical service: applications, data stores, infrastructure, identities, third parties, physical locations, and key people.

Classify services and dependencies by criticality tier (Tier 0 through Tier 3) to drive differentiated resilience investment.

Identify single points of failure in each critical service's dependency chain.

Assign a Business Resilience Owner to each Tier 0 and Tier 1 service, accountable for dependency map accuracy.

Establish a cadence for refreshing dependency maps – at minimum, quarterly and following significant architecture changes.

Integrate dependency mapping with configuration management database (CMDB), IT service management (ITSM), and risk register systems to help reduce manual maintenance.

What Good Dependency Mapping Reveals

Organizations that complete a thorough dependency map for the first time consistently find surprises: critical services with dependencies on systems classified as low-priority, third-party providers that represent single points of failure across multiple critical services, and key-person dependencies where recovery knowledge lives in the heads of two or three individuals.

These findings drive the initial ResOps backlog – the prioritized list of improvements that governance tracks to closure. The dependency map is not a one-time exercise. It must be maintained as an operational asset, updated when architecture changes, new services are onboarded, or exercises reveal gaps between the map and reality.

3 Protection, Isolation, & Containment

Limit blast radius and protect recovery paths from the same attack that hits production.

One of the most consequential shifts in modern resilience design is the recognition that attackers have learned to target recovery infrastructure first. Ransomware operators now routinely identify and destroy backup systems before encrypting production data.

If recovery assets are not protected in a hardened, isolated environment, the organization's ability to recover is compromised at the same moment the production environment is compromised.

This discipline is about architecture. Its objective is to verify that when disruption reaches production systems, recovery paths remain intact, clean, and accessible.

Implementation actions

Implement defense-in-depth for critical data and systems: network segmentation, strong identity controls, zero-trust access patterns.

Design and operate hardened, logically isolated protection domains for backups, replicas, and golden images – including air-gapped and immutable storage for highest-criticality services.

Standardize resilient architecture patterns for critical workloads: multi-zone and multi-region deployment, redundancy, fault isolation, and controlled blast-radius boundaries.

Define and validate containment controls that prevent production compromises from propagating to recovery assets: privileged access controls, isolation boundaries, and drift prevention.

Implement immutable backup configurations that help prevent deletion or modification by compromised credentials.

Design identity resilience: Confirm that recovery workflows can execute even if primary identity systems are compromised.

Establish data portability and instant repatriation capabilities for critical workloads across cloud providers.

Resilience in Depth

The principle underlying this discipline is resilience in depth: multiple independent layers of protection, designed so that when one layer fails, the next absorbs the impact without cascading failure. The goal is controlled degradation, not catastrophic failure.

Technology resilience addresses systems and data. Process resilience addresses the procedures and decision rights that govern incident response. People resilience addresses the human dependencies – enabling recovery to proceed even when key individuals are unavailable, under stress, or operating with degraded tools and information. All three layers must be designed explicitly and validated regularly.

4 Assured Recovery & Rebuild

Verify that you can restore business-critical services within tolerance, even from total loss.

Assured recovery is the technical heart of ResOps. It answers the question that backup and disaster recovery have always implied but rarely proven: If we lose everything – if the entire production environment is compromised, corrupted, or destroyed – can we rebuild to a known-good state within the window that matters to customers and regulators?

The distinction between having recovery capability and having assured recovery is evidence. Assured recovery means the organization has tested the full restoration path, validated that recovery points are clean and complete, and documented the runbooks that guide execution when tools and people are degraded.

Implementation actions

Define and document recovery architectures and runbooks for each critical service, covering data restoration, application rebuild, environment reconstitution, and dependency sequencing.

Maintain clean, validated recovery points via immutable, malware-scanned backups and snapshots logically separated from production blast radius.

Design infrastructure-as-code (IaC) and configuration-as-code capabilities that enable clean environment rebuilds without manual configuration.

Define and document rebuild-from-scratch scenarios for catastrophic compromise: how the environment is reconstituted, in what sequence, and by whom.

Perform periodic, service-scoped recovery drills that restore from clean recovery points into isolated or cleanroom environments – not into production.

Capture recovery drill results as evidence against defined impact tolerances, and feed findings into the resilience backlog.

Validate that golden images, container registries, and IaC repositories are maintained, accessible, and current – separate from production systems.

The Cleanroom Principle

One of the most important practices in assured recovery is the cleanroom restoration: restoring critical services from immutable recovery points into an isolated environment that is logically disconnected from the potentially compromised production environment. This approach allows the organization to validate recovery completeness, scan for malware in restored data, and confirm that the restored environment is functional before reconnecting it to production operations.

Organizations that skip cleanroom validation often discover during an actual incident that their recovery data was corrupted or infected before the backup was taken – a finding that is devastating when discovered mid-incident and preventable when discovered through regular drills.

5 Exercises & Validation

Turn assumptions into evidence through realistic testing and continuous validation.

Resilience that has not been tested is hope, not capability. The fifth discipline is the mechanism that converts the investments made in the other disciplines into evidence: structured exercises, at multiple levels of scope and realism, designed to expose gaps between assumed and actual recovery capability without disrupting production operations.

The exercise portfolio is designed to be non-disruptive by default. Testing occurs in isolated environments, with carefully scoped scenarios, so day-to-day operations continue. The goal is not one high-risk annual disaster recovery test – it is a continuous rhythm of validation that keeps the resilience backlog current and confidence in recoverability from any disruption grounded in evidence.

Implementation actions

Run scenario-based exercises that simulate realistic, end-to-end events: ransomware, cloud region outage, key SaaS provider loss, insider threat, AI model compromise.

Implement automated resilience tests: recovery drills, failover tests, chaos experiments, and crisis playbook rehearsals in non-production environments.

Track all exercise outcomes against defined impact tolerances and Service Resilience Indicators; use findings to update architecture, controls, runbooks, and training.

Conduct quarterly executive tabletops that walk leadership through major disruption scenarios – validating decision rights and escalation paths without touching production.

Conduct quarterly technical tabletops where engineering and operations teams step through recovery processes end-to-end in simulation.

Run bi-annual live recovery drills: full restoration from clean, immutable recovery points into isolated environments, with results captured as formal evidence.

Include AI adversarial exercises that stress-test model behavior, training pipelines, and control enforcement in segmented environments.

Conduct third-party failover tests during agreed low-risk windows to validate supplier recovery obligations.

The Resilience Backlog

Every exercise produces output beyond a report. It produces a resilience backlog: a prioritized list of specific improvements, each owned by a named team or individual, each tracked through normal delivery and governance processes until resolved. The backlog is what connects exercise findings to organizational improvement – without it, the same gaps appear in every exercise.

The backlog is prioritized by impact tolerance gap: Improvements that close the largest distance between defined tolerance and demonstrated capability come first. This helps verify that resilience investment is driven by business risk rather than technical preference.

6 Operations & Crisis Management Integration

Operate as a single system across site reliability engineering (SRE), security, and crisis management.

One of the most consistent failure modes in real disruptions is the transition from normal operations to crisis-state operations. Teams that operate effectively day to day discover in a real incident that their crisis processes, escalation paths, and decision rights are unclear, untested, or incompatible with how the incident unfolds.

The sixth discipline addresses this directly: designing, documenting, and regularly rehearsing the operating model that governs the transition from business-as-usual to crisis operations and back again. The objective is that when a real disruption occurs, the organization responds according to a practiced design – not improvisation.

Implementation actions

Define the explicit transition model from business-as-usual operations (network operations center/security operations center/SRE) to crisis-state operations (crisis management team/war room): clear triggers, decision rights, and handoff protocols.

Standardize incident taxonomies and severities that bridge cyber, technology, business continuity, and physical events – a single classification system that all teams share.

Maintain joint runbooks and communication protocols (internal and external) tested in cross-functional exercises, not just within team boundaries.

Establish a shared operating picture: consistent status reporting, service-impact views, and a single source of truth for actions and decisions during incidents.

Define who can make which decisions during a declared incident: who can freeze change, isolate systems, fail over workloads, or shut down access – and what conditions trigger those authorities.

Design for human constraints: build for fatigue, missing context, and unavailable experts through automation, checklists, cross-training, backup authority structures, and simple defaults that prevent cascading mistakes.

Practice the operating rhythm you will use in a crisis – use the same incident command patterns in exercises that you will use in real events.

Designing for Degraded Conditions

Crisis operations rarely occur under ideal conditions. Key personnel are unavailable, communication tools may be compromised, and the team is working with incomplete information under significant time pressure. Effective crisis management design accounts for these realities explicitly.

Runbooks must be written for someone who is not the primary expert – detailed enough to be followed by any of the multiple people trained as backup, simple enough to be executed under stress. Decision rights must be pre-authorized clearly enough that execution doesn't stop while someone tries to reach an unavailable decision-maker. Communication protocols must function when primary tools (Slack, email, ITSM systems) are compromised or inaccessible.

7 Measurement & Service Resilience Indicators

Provide executive clarity through measurable, continuously updated resilience indicators.

If resilience cannot be measured, it cannot be managed. The seventh discipline is the measurement framework that makes the entire ResOps operating model visible – to the engineering teams building recovery capabilities, to the council governing them, and to the board demanding evidence that critical services can withstand disruption.

Service Resilience Indicators (SRIs) are the core mechanism. They translate the technical state of recovery capability into business-relevant signals, updated continuously through telemetry and testing, and surfaced in a single dashboard view that answers the recoverability question without requiring manual investigation.

Implementation actions

Define SRIs for each critical service, aligned to defined impact tolerances: time to recover to minimum viable service, clean restore confidence score, dependency health indicators, and tolerance attainment.

Implement dashboards that surface SRIs, key risk indicators, exercise results, and major resilience gaps to senior leadership and the board.

Embed SRIs into planning and change decisions: major releases, cloud migrations, vendor selections, and investment prioritization.

Automate SRI data collection through integration with backup metadata, configuration drift monitoring, IaC repositories, and identity and access management (IAM) posture tools.

Define thresholds that trigger escalation to the ResOps Council when SRIs fall below defined minimums.

Use SRIs to prioritize the resilience backlog: Investments are sequenced by the size of the gap between current SRI and defined tolerance.

Report SRI trends to demonstrate that the ResOps program is producing measurable improvement in recovery capability.

What SRIs Measure

Reality		What It Captures
Recoverability confidence	→	Evidence-based confidence that a critical service can be restored within its defined impact tolerances, based on the most recent exercise results.
Data integrity score	→	Validation that recovery points are complete, current, clean, and free from malware or corruption.
Dependency health	→	The resilience posture of the services and systems that a critical service depends on – including third parties.
Tolerance confidence	→	The gap, if any, between defined impact tolerances and demonstrated recovery performance in the most recent exercise.
Coverage rate	→	The percentage of Tier 0 and Tier 1 services with current, validated recovery evidence – vs. those operating on assumption.
Mean Time to Clean Recovery (MTCR)	→	The average time required to restore critical business services from data that is verifiably clean – measuring recovery integrity, not just recovery speed. Where recovery time objective (RTO) measures how fast and recovery point objective (RPO) measures how recent, MTCR measures how trustworthy.

MTCR complements RTO and RPO to form a complete triad of recovery assurance – speed, recency, and integrity together constitute a full picture of recovery readiness.

ResOps Maturity Model

The ResOps Maturity Model provides a practical framework for assessing how well an organization can currently withstand disruption and restore critical services within defined impact tolerances. Maturity is measured by the ability to demonstrate recoverability with evidence – not by the number of tools deployed or documents produced.

Organizations should use the model to identify where they are today, understand what the next level requires, and structure their ResOps backlog to close the gap.

Maturity Level		Characteristics and Outcome
Level 0 – Fragmented recovery	→	Resilience capabilities exist but are fragmented across backup, security, IT operations, and business continuity. Critical services are not clearly defined. Dependencies are incomplete or outdated. Recovery relies on manual effort and individual expertise. Leadership cannot confidently answer whether critical services are recoverable right now. **Outcome:** Recovery may succeed, but only through improvisation.
Level 1 – Basic discipline	→	The organization has identified business-critical services and established basic recovery objectives. Ownership exists, and some documentation and testing are in place. Dependency mapping is partial. RTO/RPO exist but are not validated end to end. Recovery runbooks exist but vary in quality. Testing is periodic and largely technical. **Outcome:** The organization believes it can recover, but confidence is based on plans rather than proof.
Level 2 – Integrated ResOps	→	ResOps is established as a cross-functional operating discipline. Impact tolerances and SRIs are defined for critical services. Realistic, scenario-based validation is performed regularly. Clear service ownership drives recovery design. Dependencies are mapped and maintained. Exercise findings feed a managed resilience backlog. Executives receive a consolidated resilience posture view. **Outcome:** The organization can demonstrate recoverability for critical services in known scenarios.
Level 3 – Evidence-driven resilience	→	Resilience is continuously validated and actively managed as a business capability. Evidence from testing and telemetry guides investment, change approval, and board-level reporting. Validation is frequent and automated where possible. Rebuild-from-scratch scenarios are feasible for critical services. SRIs are embedded in planning and change decisions. Leaders can answer, with evidence, whether critical services can withstand disruption and recover within tolerance. **Outcome:** Resilience is no longer assumed – it is measured, governed, and improved continuously.

Using the model

Most organizations beginning a formal ResOps program sit at Level 0 or Level 1. The transition from Level 1 to Level 2 – moving from plans to proof – is typically the most significant transformation, requiring formal governance, structured exercises, and SRI measurement to be established for the first time.

ResOps Roles and the Council

No single team owns resilience. ResOps coordinates and enforces the connections between accountable owners, engineering execution, validation, and reporting into a service-level view. In practice, most ResOps roles are filled by existing practitioners – infrastructure leaders, security architects, SREs, and risk managers – who become fluent in ResOps and can reason across domain boundaries.

Head of ResOps/Resilience Operations Leader

Mission

Owns the single question: "Are we recoverable right now, for our most critical services?" Turns that into a measurable, continuously governed function across the enterprise. Typically reports to CIO or CISO, with a dotted line to the other.

Core responsibilities

- Define the ResOps charter, operating model, and scope across all clouds, data centers, SaaS, and business units.
- Prioritize critical business services and map them to underlying apps, infrastructure, data, and third-party services.
- Approve recovery service-level agreement (SLA) tiers and policy: RPO, RTO, coverage, test frequency, and isolation requirements.
- Sponsor the ResOps discipline and the single source of truth for recoverability posture.
- Escalate gaps to CIO and CISO: "These are the services that cannot be recovered to SLA today, and why."

Key success metrics

- Time to answer "are we recoverable?" for any critical service.
- Percentage of Tier 0 and Tier 1 services demonstrably recoverable to SLA.
- Reduction in unknowns and manual work during recovery tests and incidents.

ResOps Architect/ Resilience Architect

Mission

Designs how resilience is built and proved: architectures, patterns, data flows, and the telemetry needed to answer the recoverability question with evidence.

Core responsibilities

- Define reference architectures for resilient business services.
- Specify how data, configurations, secrets, identities, and dependencies are protected and reconstructed.
- Design integration patterns between security tools, backup platforms, disaster recovery tooling, cloud providers, and SaaS platforms.
- Decide what telemetry is required for recoverability evidence: backup metadata, configuration drift, IaC state, SaaS export capabilities, IAM posture.
- Shape the resilience scoring model used by CISO and CIO.

Key success metrics

- Coverage of services under standardized resilience patterns.
- Reduction of one-off custom disaster recovery solutions.
- Number of services with full dependency and protection mapping.

ResOps Engineer/ Automation Engineer

Mission

Automates the entire ResOps lifecycle: discovery, data collection, verification, testing, and evidence generation. Makes recoverability a pipeline, not a spreadsheet.

Core responsibilities

- Deploy and maintain collectors that pull metadata from security tools (security information and event management, endpoint detection and response, cloud security posture management), backup systems, cloud infrastructure, and SaaS APIs.
- Build pipelines that correlate data into a single model of "what is protected, how, and to what SLA."
- Automate recovery tests, game days, and on-demand simulate-recovery workflows.
- Integrate ResOps outputs with ITSM; CMDB; governance, risk, and compliance; and risk register systems.

Key success metrics

- Percentage of data collection automated vs. manual.
- Frequency and reliability of automated recovery tests.
- Mean time to generate a recoverability report for any critical service.

ResOps Analyst/ Recoverability SRE

Mission

Turns raw signals into a clear recoverability posture view: by business service, by SLA, by risk. During incidents, acts as the recovery navigator.

Core responsibilities

- Monitor ResOps dashboards and alerts across backup coverage, configuration drift, test results, and environmental risk.
- Investigate gaps: unprotected data, missing configurations, failed restore tests, fragile dependencies.
- Work with service owners to remediate issues and track them to closure.
- Prepare board, regulator, and executive-level reports that explain resilience in business terms.

Key success metrics

- Reduction in uncovered systems and stale backups.
- Time to identify and classify a resilience gap.
- Improvement in resilience scores over time for critical services.

ResOps Governance and Risk Lead

Mission

Makes recoverability a governed obligation, not a best-effort activity. Ties ResOps into policy, audit, and regulatory frameworks.

Core responsibilities

- Translate regulatory and industry expectations for operational resilience into concrete ResOps controls.
- Define policies for recovery testing cadence, evidence retention, and sign-off.
- Align ResOps metrics with enterprise risk management, business continuity, and audit programs.
- Prevent new projects from going live without meeting minimum ResOps criteria.

Key success metrics

- Number of critical services with signed-off recovery evidence.
- Reduction of resilience-related audit and regulatory findings.
- Adoption of ResOps requirements in new project and vendor onboarding processes.

Business Resilience Owner/ Service Owner

Mission

Owns recoverability for a specific business service or application. Acts as the accountable voice for that service, supported by the ResOps function.

Core responsibilities

- Define what "recovered" means in business terms: tolerable data loss, downtime, and degraded operating modes.
- Maintain the dependency mapping for their service across applications, infrastructure, data stores, and third parties.
- Accept or challenge the ResOps assessment for their service and sponsor remediation work.
- Participate in recovery exercises and maintain accuracy of runbooks.

Key success metrics

- Successful recovery tests for their service to SLA.
- Time to business functionality restoration in incidents or exercises.
- Completeness and freshness of dependency mapping.

Conclusion

The ResOps disciplines in this framework represent a practitioner starting point for building the capability across the five ResOps domains – from the governance structures that provide authority and accountability, to the technical architectures that protect recovery paths, to the measurement frameworks that make recoverability visible and improvable.

No organization builds all seven disciplines simultaneously. The path to evidence-driven resilience is iterative: Establish governance and identify critical services first, then build protection and recovery capability for your highest-priority services, and then validate continuously and use the evidence to drive improvement across the broader portfolio.

The organizations that invest in this operating model now – before the next significant disruption – are building a durable competitive advantage. When the disruption comes, they are better positioned to recover faster, with less improvisation, and with the evidence to demonstrate recoverability to customers, regulators, and boards.

ResOps provides a framework for building provable resilience. These disciplines are where enterprises begin to build it – and the starting point for a broader community conversation about what the full model becomes.

Related Resources on Readiverse

Resource		Description
Evidence Over Hope: The Executive Case for ResOps	→	The core Readiness Report for CIOs and CISOs, including the ResOps Council setup framework and Impact Tolerance Framework.
ResOps: The Future of Resilient Business in the Era of AI	→	Foundational overview of the ResOps discipline.
From Minimum Viability to Operational Resilience	→	Co-authored with Deloitte. Goes beyond recovery readiness to the harder question: how do you prove it? Covers the ResOps maturity model, impact tolerances, service resilience indicators, and Mean-Time-to-Clean Recovery — the metrics that make resilience a continuous discipline, not a periodic exercise.

Available at readiverse.com/resops