UPC DRaaS™

UNITEDLAYER®

E-BOOK · THE COMPLETE GUIDE

The Continuity Handbook.

Why most disaster-recovery plans are partial, what that actually costs, and how to build one you can afford to apply everywhere — and afford to test.

12 chapters 3 frameworks 1 decision

CONTENTS

What this handbook covers

This guide is for the people who own the consequences of an outage: CIOs, heads of infrastructure, business-continuity leads, CISOs, and the finance partners who approve what all of it costs. It assumes you have a DR plan. The question it answers is whether it covers what it should — and whether you could prove it does.

01 **Why DR plans are partial** — 03
The economics behind every scope decision

02 **What you are actually buying** — 04
Readiness, capacity, and the gap between them

03 **Framework 1 — the consequence test** — 05
Setting RPO and RTO from business impact

04 **Tiering without over-buying** — 06
Platinum, Gold, Silver applied honestly

05 **Why untested DR is an assumption** — 07
Drill cadence as the real maturity signal

06 **Ransomware changed recovery** — 08
Immutability, clean rooms, validated points

07 **The AI stack has no plan** — 09
Registries, inference data, vectors, serving

08 **Framework 2 — the continuity maturity model** — 10
Five levels from documented to demonstrated

09 **Topologies and what they cost** — 11
Active-Active, Active-Passive, Active-Replicated

10 **DR as a migration bridge** — 12
Turning a VMware exit into a resilience upgrade

11 **Framework 3 — the DR readiness checklist** — 13
Twelve questions before you sign anything

12 **Getting started** — 14
Scoping, tiering, and the first drill

THREE FRAMEWORKS YOU CAN REUSE

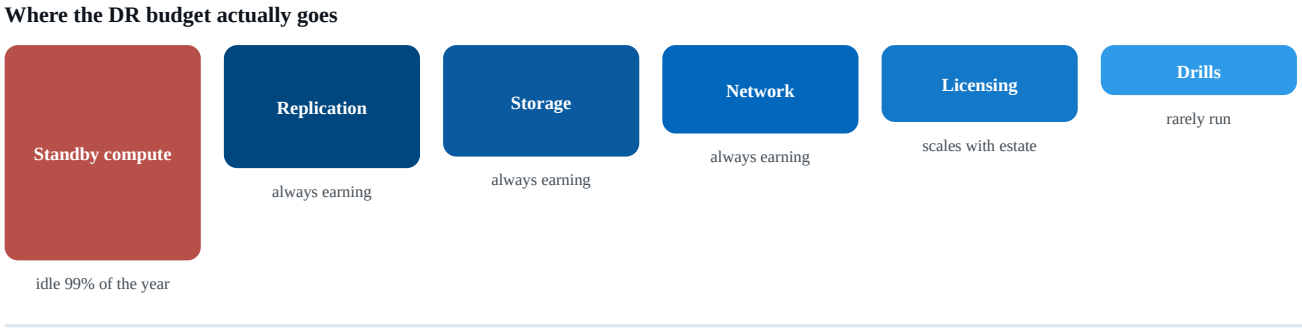
- **The consequence test** — deriving RPO and RTO from business impact rather than infrastructure capability.
- **The continuity maturity model** — five levels, each defined by what it still cannot demonstrate.
- **The DR readiness checklist** — twelve questions worth answering before signing any DR contract.

01 CHAPTER ONE

Why DR plans are partial

Ask an infrastructure team why a particular application is not covered by disaster recovery and you will rarely hear that the risk was assessed and accepted. You will hear that it did not make the list.

That phrasing is worth pausing on, because it describes a budget decision wearing the clothes of a risk decision. In the conventional model, protecting an application means standing up capacity to run it somewhere else — capacity that must exist continuously, be licensed continuously, and be paid for continuously, against an event that may never occur. Protection cost therefore scales almost linearly with the size of the protected estate, and the list of covered systems gets drawn where the budget runs out.



The largest line item is the one that does no work until the day you need it — and the smallest is the one that proves the rest will function.

FIGURE 1 — THE SHAPE OF A CONVENTIONAL DR BILL

The consequence of a cost-driven scope

Two things follow, and both are predictable. First, the applications excluded are the ones ranked below the cut line — which is a proxy for importance, but a poor one, since dependency rarely follows org-chart significance. Second, the exclusions are invisible until an incident, at which point the organisation discovers that the tier-one system it protected depends on a tier-two system it did not.

> ***Nobody decides to leave a system unprotected. They decide to protect the ones they can afford, and the rest is what is left over.***

02 CHAPTER TWO

What you are actually buying

A disaster-recovery service sells two distinct things that conventional pricing bundles together: readiness, and capacity. Separating them is the single change that makes most other improvements affordable.

Readiness is continuous. Capacity is occasional.

Readiness is the state of being able to recover: current replicas, a validated configuration, tested runbooks, a committed recovery target. It has to exist every second of every day, because the event it protects against is unscheduled. It is also comparatively cheap — storage, replication bandwidth, orchestration.

Capacity is the compute required to actually run the workload after failover. It is expensive, and it is needed on precisely two kinds of day: a real incident, and a drill. On every other day it produces nothing.

COMPONENT	WHEN IT EARNS ITS COST	CONVENTIONAL MODEL	ON-DEMAND MODEL
Replication	Continuously — the replica must be current	Charged continuously	Charged continuously
Storage	Continuously — the data must exist	Charged continuously	Charged continuously
Orchestration	Continuously — policy and runbooks live	Charged continuously	Charged continuously
Compute	Only on activation or drill	Charged continuously	Charged on activation
Licensing	Follows compute in most models	Charged continuously	Follows activation

Why the distinction matters commercially

Once compute is decoupled from readiness, the marginal cost of protecting one more application falls dramatically — it becomes storage and replication rather than a second machine. That is what allows scope to be set by consequence instead of budget, and it is also what makes frequent testing viable, since a drill costs the compute it consumes rather than a year of standby.

THE QUESTION TO ASK ANY DR PROVIDER

- Which components do I pay for continuously, and which do I pay for only when I activate or test? If the answer is "all of them, continuously," you are funding idle capacity regardless of what it is called.

03 FRAMEWORK ONE

The consequence test

Recovery objectives are too often inherited from what the infrastructure can do. They should be derived from what the business loses while the system is down.

THE RULE

- **Set RPO from the cost of re-creating lost work. Set RTO from the cost of the outage itself. Where either cost is non-financial — safety, regulatory, reputational — state it explicitly rather than converting it to a number nobody believes.**

Applying it

For each application, answer two questions with the business owner rather than the infrastructure team. First: if we lost the last N minutes of transactions, what would it take to reconstruct them, and could we? That answer sets RPO. Second: what happens in the first hour, the first four hours, and the first day of this being unavailable? That answer sets RTO — and it is frequently the more revealing of the two, because the discussion surfaces manual fallbacks that either exist or do not.

APPLICATION TYPE	RPO DRIVER	RTO DRIVER	TYPICAL TIER
Payments, trading, checkout	Transactions cannot be reconstructed	Direct revenue loss per minute	Platinum
Clinical records	Clinical decisions depend on currency	Patient-safety exposure within a shift	Platinum
Order management, CRM	Re-keyable but expensive	Customer-visible within the hour	Gold
Reporting, analytics	Re-computable from source	Internal impact only	Silver
Internal tooling	Low — little unique state	Tolerable for a working day	Silver

The test does two useful things. It moves the conversation out of IT, which is where the consequences actually land. And it produces a defensible rationale for each tier — which is what a regulator, auditor, or board asks for when they ask why a system was protected the way it was.

04 CHAPTER FOUR

Tiering without over-buying

A single recovery target across an estate is always wrong in both directions at once: it over-protects the systems nobody would miss for a day, and under-protects the ones measured in minutes.

Matching commitment to consequence

Platinum	**RPO near-zero** Ask: what does an hour of this being down actually cost?	**RTO 15 minutes**	Revenue-bearing, safety-critical, regulated transaction systems
Gold	**RPO < 15 minutes** Ask: can the business run manually for an hour?	**RTO 1 hour**	Core business applications and customer-facing services
Silver	**RPO < 1 hour** Ask: would anyone outside IT notice within a shift?	**RTO 4 hours**	Supporting systems, reporting, internal tooling

FIGURE 2 — THE TIERING DECISION

The discipline tiering requires

Tiering only saves money if the tiers are applied honestly, and the common failure is grade inflation — every business owner asked whether their system is critical says yes. Two practices prevent it. First, require the consequence test from Chapter 3 to be completed with the business owner and recorded; a tier without a stated rationale defaults down. Second, publish the relative cost of each tier, so choosing Platinum for a reporting system is a visible budget decision rather than an invisible default.

SIGNS YOUR TIERING IS NOT REAL

- More than a third of applications are in the top tier.
- No application has ever been **moved down** a tier after review.
- Tier assignment was made by IT without a named business owner.
- The cost difference between tiers is not visible to the people choosing.

05 CHAPTER FIVE

Why untested DR is an assumption

A recovery plan that has never been executed end to end is a hypothesis. It may be a well-researched one, but it has not been subjected to the only experiment that matters.

The reasons testing slips are consistent and rarely about willingness. A full drill in a conventional model requires production-equivalent compute to be available, which either means duplicating capacity or borrowing it from somewhere that needs it. It requires staff time during a window when nothing else is happening. And in seasonal or safety-critical businesses, it collides with change freezes exactly when the outcome would be most informative.

> ***The most expensive part of most DR programmes is the standby capacity. The most valuable part is the drill. Conventional pricing makes you buy the first and skip the second.***

What a real drill surfaces

Organisations that move from annual to quarterly testing report a consistent pattern: the first full drill finds problems in dependency order, credentials, DNS, and licensing rather than in replication. Those are precisely the failures that documentation cannot predict and that a partial test — restoring one system in isolation — will never surface.

TEST TYPE	WHAT IT PROVES	WHAT IT MISSES
Backup restore	The data is readable	Whether the application starts, and in what order
Single-system failover	One workload can run elsewhere	Cross-system dependencies and shared services
Tabletop exercise	People know the plan	Whether the plan is technically correct
Full failover drill	The estate recovers, in order, within target	Only behaviour under genuine incident stress

Cadence is the practical signal of maturity. Annual testing tells you the plan was correct a year ago, against an estate that has since changed. Quarterly testing tells you it is correct now — and in most organisations the estate changes enough in a quarter that this distinction is not academic.

06 CHAPTER SIX

Ransomware changed what recovery means

Traditional disaster recovery was designed for events that are indifferent to your data: a power failure, a flood, a hardware fault. Ransomware is not indifferent. It targets the data, and increasingly it targets the backups first.

Speed is no longer sufficient

In a conventional disaster, the most recent recovery point is the best one. Under a cyber incident it may be the worst, because the payload was probably present before the encryption was noticed. Dwell time — the period between compromise and detection — is frequently measured in weeks, which means a fast restore from the latest snapshot can reintroduce the attacker along with the data.

Recovery therefore acquires three requirements it did not previously have: backups that cannot be altered or deleted, a place to restore into that is not the compromised environment, and a way to establish that a given recovery point is actually clean before it is promoted to production.

WHAT CYBER-RESILIENT RECOVERY REQUIRES

- **Immutability by default** — protected copies that cannot be modified or deleted within their retention window, including by an administrator.
- **Isolation** — recovery data held separately from production, so a compromise of one does not reach the other.
- **Clean-room recovery** — a separate environment to restore into and validate before anything returns to production.
- **Recovery-point validation** — a means of testing candidate points for integrity rather than assuming the latest is safe.
- **Model integrity protection** — for AI estates, assurance that model artefacts have not been tampered with.

Why this belongs in DR rather than beside it

Organisations often buy cyber-recovery capability separately from disaster recovery, and end up with two recovery paths, two runbooks, and two sets of assumptions about which to use under pressure. Since the decision between them has to be made during an incident, by people under stress, with incomplete information about what kind of event they are in, the practical argument for a single platform covering both is not efficiency — it is that there is one procedure to get right.

07 CHAPTER SEVEN

The AI stack has no plan

Enterprise AI moved from experiment to production dependency faster than continuity planning could follow. In most organisations the applications are in the DR plan and the AI serving them is not.

The new dependencies

An AI-enabled application depends on components that did not exist in the last continuity review: a model registry holding versioned artefacts, training and inference datasets, vector indexes supporting retrieval, serving endpoints, and the observability layer that makes any of it debuggable. If the application is tier one, those dependencies are tier one too — regardless of which team owns them or which budget bought them.

> ***If your copilot is in the critical path and your model registry is not in the DR plan, your copilot is not actually protected.***

AI COMPONENT	WHY IT NEEDS RECOVERY	COMMON GAP
Model registry	Versioned artefacts the application binds to	Treated as a build artefact, not production state
Training & inference data	Expensive or impossible to reconstruct	Covered by backup, not by recovery objectives
Vector indexes	Retrieval fails silently without them	Assumed rebuildable; rebuild time rarely measured
Serving endpoints	The application's runtime dependency	Owned by a platform team outside the DR programme
Observability	Without it, recovery cannot be validated	Recovered last, when it is needed first

A practical first step

Take the applications already in your top DR tier and trace every AI dependency each one has. In most estates this exercise takes an afternoon and produces a list nobody had written down before — which is itself the finding. Whether those components then get Platinum or Gold is a tiering conversation; whether they appear in the plan at all should not be.

08 FRAMEWORK TWO

The continuity maturity model

Each level is defined by what it cannot yet demonstrate. Most organisations sit at level two or three and believe they are at four.

1	**Documented**	A DR plan exists, with runbooks and named owners. **Cannot demonstrate:** that the plan works, since it has not been executed.
2	**Partial**	Critical applications are replicated and individually restorable. **Cannot demonstrate:** that the estate recovers together, in dependency order.
3	**Tested**	Full failover has been executed at least once, with findings closed. **Cannot demonstrate:** that it still works against an estate that has changed since.
4	**Routine**	Drills run on a regular cadence with measured recovery times against committed tiers. **Cannot demonstrate:** clean recovery from a cyber incident, as distinct from an outage.
5	**Resilient**	Cyber recovery validated through clean-room restoration, the AI stack in scope, and every tier evidenced by recent drill data. **Cannot demonstrate:** nothing material — recovery is proven rather than planned.

How to use it

The levels are sequential and the common stall is between three and four, for a reason that is commercial rather than technical: a single full test can be justified as a project, while a recurring quarterly one has to be funded as an operating cost. That is precisely the cost that on-demand compute removes, which is why the pricing model and the maturity ceiling are more closely related than they first appear.

A FOUR-QUESTION SELF-ASSESSMENT

- Has your full estate ever failed over together, in dependency order? **(Level 3)**
- When was the last drill, and what recovery time did you actually measure? **(Level 4)**
- Could you demonstrate that a chosen recovery point is free of a payload? **(Level 5)**
- Is your model registry in the same plan as the applications that depend on it? **(Level 5)**

09 CHAPTER NINE

Topologies and what they cost

Three recovery topologies cover nearly every requirement. Choosing between them is mostly a question of how much you are willing to pay to compress recovery time.

TOPOLOGY	HOW IT WORKS	BEST FOR	TRADE-OFF
Active-Active	Both sites carry live workload; failure removes capacity rather than service	Mission-critical environments needing the lowest RTO	Highest cost; requires applications that tolerate multi-site operation
Active-Passive	A standby environment kept current, brought up on activation	Core business applications on Gold-tier commitments	Recovery measured in minutes to an hour, not seconds
Active-Replicated	Data replicated to a cold site; compute stood up when required	Systems that need not be recovered instantly	Longest recovery; lowest ongoing cost

The under-used option

Active-Passive has a property that is often overlooked: the standby environment does not have to be idle between incidents. Using it for test and development means the capacity earns something during the ninety-nine percent of its life when no disaster is occurring — which also has the useful side effect of keeping it exercised and current, rather than discovering during an incident that it drifted from production months ago.

CHOOSING BETWEEN THEM

- Start from the **tier**, not the topology. Platinum usually implies Active-Active; Gold implies Active-Passive; Silver implies Active-Replicated.
- Check whether the **application** supports the topology — multi-site operation is an application property, not an infrastructure one.
- Ask what the standby will do **between incidents**; an environment with a second purpose stays healthier.

10 CHAPTER TEN

DR as a migration bridge

Organisations reassessing their virtualisation platform face a problem with an unexpected solution: the work required to build a recovery estate is very nearly the work required to migrate.

The logic is straightforward. A DR programme replicates production into a second environment, validates that workloads run there, and rehearses the cutover. A migration does exactly the same things, with one difference — the cutover is intended to be permanent. An organisation that has built a tested recovery environment on a modern hypervisor has, whether or not it planned to, built a migration target it has already proven.

> ***A recovery environment you have failed over to successfully is a migration target you have already tested.***

Why this sequencing reduces risk

Treating migration as a series of rehearsed failovers rather than a single cutover changes the risk profile. Each system moves when its recovery has been demonstrated, the original estate stays available as rollback until the new one holds, and the team executing the migration is using a procedure they have already practised. The alternative — a big-bang cutover with rollback existing only on paper — concentrates all the risk into one window.

STAGE	DR OBJECTIVE	MIGRATION OBJECTIVE
Replicate	Maintain a current copy for recovery	Establish the target environment
Validate	Prove workloads run at the recovery site	Prove workloads run on the new platform
Drill	Rehearse failover and measure recovery time	Rehearse cutover and identify dependencies
Activate	Fail over during an incident, then fail back	Cut over permanently, retaining rollback

The practical caveat is that this works best where the recovery platform is one you would choose on its own merits. A migration bridge to somewhere you did not want to be is just a slower version of the wrong decision.

11 FRAMEWORK THREE

The DR readiness checklist

Twelve questions worth answering — about your current arrangement as much as any proposed one. The useful signal is whether each answer is a number, a document, or a hope.

- [ ] Which components bill continuously, and which bill on activation?
- [ ] When did the full estate last fail over together?
- [ ] Which applications are out of scope, and who decided?
- [ ] Is there a clean room to restore into after a cyber incident?
- [ ] Is the model registry and inference data in the plan?
- [ ] What does the standby environment do between incidents?
- [ ] What is the committed RPO and RTO, per application?
- [ ] What recovery time was measured, as opposed to targeted?
- [ ] Are backups immutable, and for how long?
- [ ] How is a recovery point validated as clean before promotion?
- [ ] What does a drill cost, and how often can you afford one?
- [ ] Who is accountable for the plan being current, not just correct?

Scoring the answers

Sort each response into one of three categories. **Measured** answers cite a figure from a recent test. **Documented** answers cite a plan or contract. **Assumed** answers are reasoning rather than evidence — "it should come up in about an hour." A healthy programme produces measured answers to the questions about recovery time and drill cadence, and documented answers to the rest. Any assumed answer to a question about a tier-one system is the most valuable finding on the page, because it is the gap you did not know you had.

> ***Ask for the number from the last test. If the answer is a target rather than a measurement, you have found the work.***

12 CHAPTER TWELVE

Getting started

A continuity programme does not need to begin with a platform decision. It begins with an inventory and an honest conversation about consequence.

STEP	WHAT TO DO	WHAT IT PRODUCES
1 · Inventory	List every application, with its business owner and its dependencies — including AI components	The actual scope, usually larger than the current plan
2 · Consequence	Run Chapter 3's two questions with each business owner and record the answers	A defensible RPO and RTO per application
3 · Tier	Assign Platinum, Gold, or Silver from the recorded rationale, with costs visible	A tiering that survives review
4 · Model the cost	Compare continuous versus on-activation pricing across the full scope	Whether full coverage is affordable — usually yes
5 · Drill	Run one full failover and measure what actually happens	Your real recovery time, and the findings list
6 · Set cadence	Fix a recurring drill schedule and hold it	Movement from level three to level four

Where UPC DRaaS fits

The Pay-on-Activation model is what makes steps four, five, and six affordable in sequence: readiness billed continuously across the whole scope, heavy compute billed only when a drill or an incident calls for it. Three tiers support the output of step three, three topologies match commitment to application capability, and immutability with clean-room recovery covers the cyber case from Chapter 6 within the same platform and the same runbook.

THE FIVE THINGS WORTH TAKING AWAY

- **Scope is usually a budget decision.** Ask what you would protect if cost were not the constraint — then check what it would actually cost.
- **Readiness and capacity are different purchases.** Only one of them needs to be continuous.
- **Untested is unproven.** Cadence, not documentation, is the maturity signal.
- **Fast recovery can restore the attacker.** Immutability and a clean room are now part of DR, not adjacent to it.
- **Trace your AI dependencies.** The registry and the indexes belong in the same tier as the application they serve.

IN CLOSING

Always ready, never idle

Disaster recovery has always been sold as insurance, and insurance is a poor metaphor — it implies a premium paid against a payout, with nothing in between. A well-built continuity capability is not dormant. It is a tested, exercised, continuously current second copy of the estate, and the organisations that get most from it are the ones that use it.

That is the practical argument for separating readiness from capacity. Readiness, kept continuously, is what lets you answer a supervisor, satisfy a customer's security review, and sleep before a peak trading period. Capacity, invoked on demand, is what lets you test often enough for that readiness to mean something — and what removes the reason to leave systems out of scope in the first place.

BOTTOM LINE

Always ready, never idle — disaster recovery that stays committed 24/7/365 and bills for compute only when you need it.

Let's build your continuity plan.

Speak with a specialist, or ask us to run a scope, tiering, and DR cost assessment against your current estate.

UPC DRaaS™
UNITEDLAYER.COM

EMAIL
contact@unitedlayer.com

PHONE
+1-888-853-7733

X
x.com/unitedlayer

LINKEDIN
linkedin.com/company/unitedla

This handbook reflects UnitedLayer published materials for UPC DRaaS together with general industry practice in business continuity and disaster recovery. Frameworks presented are analytical tools, not benchmarks, and do not constitute regulatory or legal advice. Service tiers, availability figures, regional coverage, and comparative cost figures vary by configuration and contract and should be confirmed for your deployment. UnitedLayer®, United Private Cloud®, and UPC DRaaS™ are marks of UnitedLayer.

© 2026 UNITEDLAYER. ALL RIGHTS RESERVED.