gart.

# Self-Assessment: IT Infrastructure

A practical diagnostic for CTOs, founders, and tech leaders

This assessment is designed to surface structural risks, operational bottlenecks, and hidden inefficiencies in modern IT infrastructure — before they turn into incidents, outages, or runaway cloud costs.

⏱️ 5–7 minutes

📊 Instant results

🔍 Vendor-agnostic, implementation-driven

🎯 Built for SaaS, digital products, and data-driven businesses

SECTION 1

# Architecture & Design

## Q1. How standardized, documented, and transferable is your infrastructure architecture?

*Choose one of the options, which is the most relevant to your case.*

01

### 1) Fragmented & tribal knowledge–driven

- No up-to-date architecture diagrams
- Knowledge lives in individuals' heads
- New engineers struggle to understand system design
- High onboarding and handover risk

02

### 2) Partially documented, inconsistently applied

- Some diagrams or docs exist, but outdated
- Different teams use different patterns
- Documentation is not used in daily decision-making

03

### 3) Mostly standardized, but incomplete or siloed

- Core components follow agreed patterns
- Documentation exists but lacks depth or context
- Changes are not always reflected in architecture docs

04

### 4) Fully standardized, documented, and operationalized

- Clear reference architectures per workload
- Documentation is living, reviewed, and versioned
- Architecture decisions are explicit and repeatable

# Q2. How resilient is your infrastructure to failures of individual components or services?

*Choose one of the options, which is the most relevant to your case.*

### 1) Single points of failure are common

- One node, one database, one region dependencies
- Failures cause full or partial outages
- Recovery is manual and stressful

### 2) Partial redundancy without real confidence

- Some failover exists, but rarely tested
- Dependencies between components are unclear
- Recovery time is unpredictable

### 3) Designed for resilience, but with known gaps

- Redundancy for critical components
- Failover exists but not automated everywhere
- Regular concerns during peak load or incidents

### 4) Designed for failure and continuously validated

- No critical single points of failure
- Failover is automated and tested
- Teams are confident during incidents

SECTION 2

# Reliability & Observability

## Q3. How effectively are incidents detected, surfaced, and prioritized?

*Choose one of the options, which is the most relevant to your case.*

1

### 1) Users detect problems first

- No meaningful alerts
- Incidents discovered via support tickets or social media
- No clear signal vs noise

2

### 2) Alerts exist but are noisy or unreliable

- Alert fatigue is common
- Important issues are buried among false positives
- On-call response is reactive

3

### 3) Alerts are useful but incomplete

- Core systems are monitored
- Some blind spots remain
- Root cause analysis still takes time

4

### 4) Proactive, signal-driven observability

- Clear SLIs/SLOs are defined
- Issues are detected before user impact
- Monitoring supports fast diagnosis

# Q4. How well do you understand and manage dependencies across infrastructure and services?

*Choose one of the options, which is the most relevant to your case.*

### 1) Dependencies are largely unknown

- Failures cascade unexpectedly
- Changes cause unpredictable side effects

### 2) Dependencies known informally

- Senior engineers "just know"
- High risk when people are unavailable

### 3) Dependencies documented but static

- Diagrams exist but become outdated
- Limited visibility during incidents

### 4) Dependencies mapped and observable in real time

- Service maps and dependency graphs exist
- Teams understand blast radius before changes
- Faster incident containment

SECTION 3

# Automation & Operations

## Q5. How is infrastructure provisioned, changed, and reviewed?

*Choose one of the options, which is the most relevant to your case.*

### 1) Manual changes on live systems

- High risk of configuration drift
- Changes are hard to audit or rollback

### 2) Scripts exist but lack discipline

- Automation varies by engineer or team
- Inconsistent environments

### 3) Infrastructure as Code is used, but not enforced

- IaC exists but not mandatory
- Manual fixes still happen under pressure

### 4) Fully automated, versioned, and controlled

- All changes go through code review
- Rollbacks are predictable
- Environments are reproducible

# Q6. How consistent and repeatable are environments (dev / test / staging / prod)?

*Choose one of the options, which is the most relevant to your case.*

### 1) Each environment behaves differently

- "Works in staging, fails in prod" is common
- Debugging takes excessive time

### 2) Mostly similar, but manually adjusted

- Differences are undocumented
- Drift accumulates over time

### 3) Mostly consistent with known exceptions

- Drift is tracked but not fully prevented
- Some manual steps remain

### 4) Fully reproducible and drift-controlled

- Environments are identical by design
- Drift detection is automated

SECTION 4

# Security & Access Control

## Q7. How is access to infrastructure and production systems governed?

*Choose one of the options, which is the most relevant to your case.*

1

### 1) Broad or shared access

- Shared accounts or credentials
- High insider-risk exposure

2

### 2) Basic role separation

- Roles exist but rarely reviewed
- Over-permission is common

3

### 3) Controlled access with periodic review

- Access tied to roles
- Reviews are manual or infrequent

4

### 4) Least-privilege by default

- Access is time-bound and audited
- Strong separation of duties

# Q8. How are secrets, credentials, and sensitive configurations handled?

*Choose one of the options, which is the most relevant to your case.*

### 1) Hardcoded or manually shared

- Secrets in code or chat tools
- High breach risk

### 2) Partially centralized

- Multiple secret storage methods
- Inconsistent enforcement

### 3) Centralized but not fully automated

- Secrets manager exists
- Rotation and audits are irregular

### 4) Secure, centralized, and continuously audited

- Automated rotation
- Access logging and compliance readiness

SECTION 5

# Cost & Scalability

## Q9. How visible, predictable, and controllable are infrastructure costs?

*Choose one of the options, which is the most relevant to your case.*

1

### Costs are understood only after invoices arrive

- No ownership or accountability
- Budget surprises are frequent

2

### Basic reporting without actionability

- Cost data exists but is not used
- Optimization is ad-hoc

3

### Cost tracking with limited automation

- Budgets and alerts exist
- Optimization is periodic

4

### Real-time visibility and cost governance

- Costs tied to teams and services
- Continuous optimization

# Q10. How well does your infrastructure scale with growth, traffic spikes, or new workloads?

*Choose one of the options, which is the most relevant to your case.*

1

**Scaling is manual and risky**

- Performance degrades under load
- Scaling causes outages

2

**Scaling works but destabilizes systems**

- Requires firefighting
- Side effects are common

3

**Scaling is reliable but resource-heavy**

- Works with planning and effort
- Efficiency is not optimal

4

**Predictable, automated, and cost-aware scaling**

- Elastic by default
- Supports growth and experimentation

# Self-Assessment Result Interpretation

## 🔴 0–10

### Fragile Infrastructure

**What this means:**

- High operational and business risk
- Growth amplifies instability

**Recommended next step:** IT Infrastructure Audit

## 🟠 11–18

### Reactive Infrastructure

**What this means:**

- Systems work, but issues are handled after impact
- Hidden cost, security, and reliability risks

**Recommended next step:** Stabilization & Automation Program

## 🟢 19–24

### Stable but Inefficient

**What this means:**

- Solid foundation
- Optimization and scalability potential not fully unlocked

**Recommended next step:** Cost Optimization & DevOps Improvements

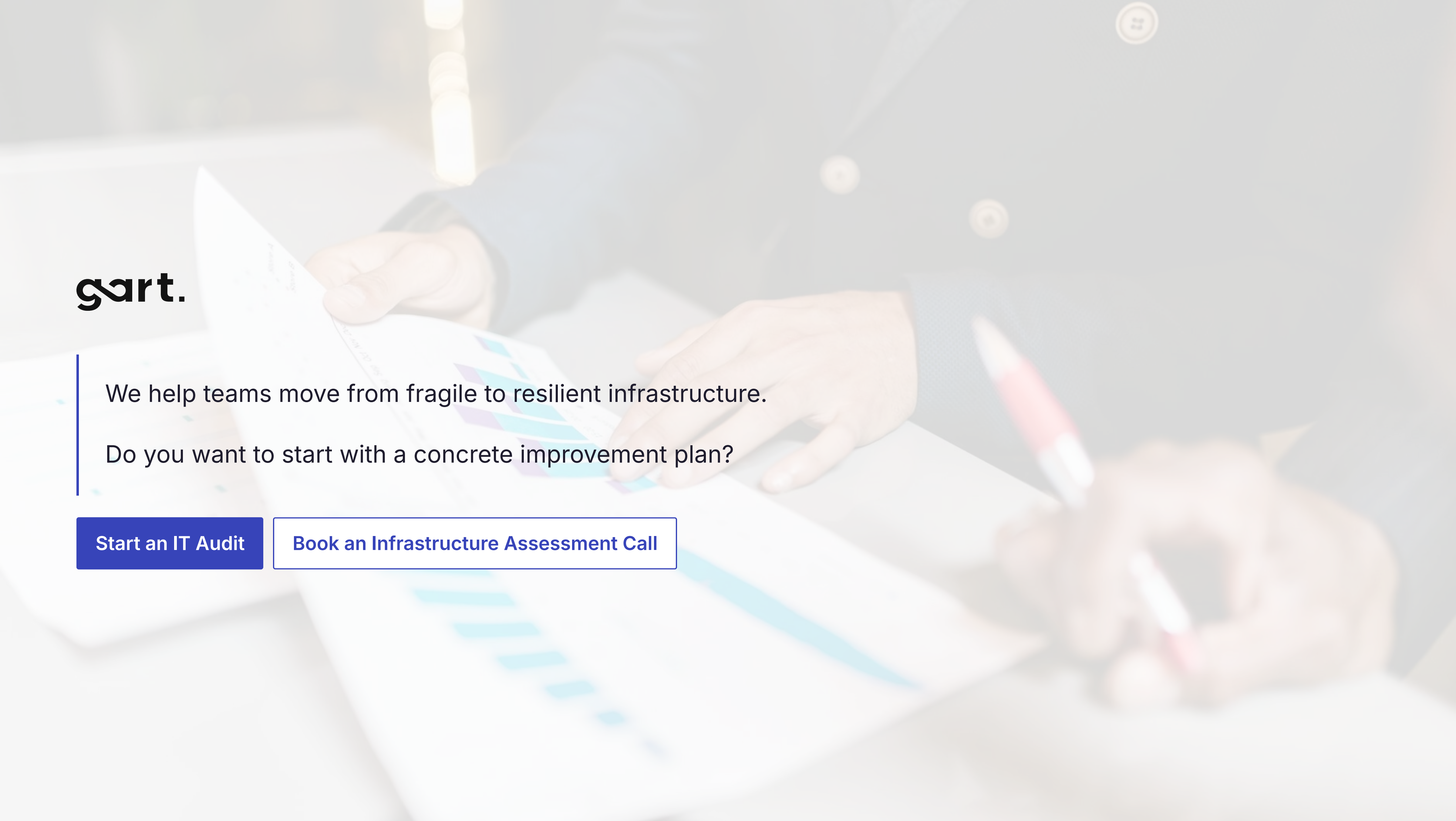
## 🔵 25–30

### Resilient & Scalable

**What this means:**

- Infrastructure supports growth, audits, and advanced workloads
- Ready for AI, data, and rapid scaling

**Recommended next step:** Advanced Optimization or SRE Practices



gart.

> We help teams move from fragile to resilient infrastructure.
>
> Do you want to start with a concrete improvement plan?

Start an IT Audit

Book an Infrastructure Assessment Call