

START WHERE YOU ARE:

A Practical Guide to IaC Security Maturity





TABLE OF CONTENTS

04	Why Most Orgs Are (Still) at Level 2
05	The Power of Picking Your Own Path
06	Level-by-Level Pathways
07	Level 1: Emergent
09	Level 2: Reactive
11	Level 3: Proactive
13	Level 4: Adaptive
15	Level 5: Resilient
17	Gomboc's Role Across the Maturity Model
20	Business Impact at Every Stage



Infrastructure as Code (IaC) has reshaped how teams manage cloud environments, bringing speed, repeatability, and control.

But despite its benefits, IaC isn't secure by default. Misconfigurations can still slip through, creating risk, even in well-written code.

You've likely identified where your organization stands if you've read the IaC Security Maturity Model. Maybe you've adopted version control and basic scanning. Perhaps you're stuck between stages, unsure how to move forward. The natural question is: What now?

This guide offers a practical next step, not a leap to perfection, but a map for progress. We'll show how to evolve your security posture from wherever you are today with real-world examples, common pitfalls, and actionable guidance.

It's designed for DevOps leaders, cloud engineers, CISOs, compliance teams, and anyone navigating the tension between speed and security in a fast-moving cloud environment. Maturity is a journey. Let's take the next step together.





WHY MOST ORGS ARE (STILL) AT LEVEL 2

After years of experimentation and iteration, many organizations have taken the first real steps into Infrastructure as Code.

At Level 2, Reactive, IaC is no longer a proof of concept.

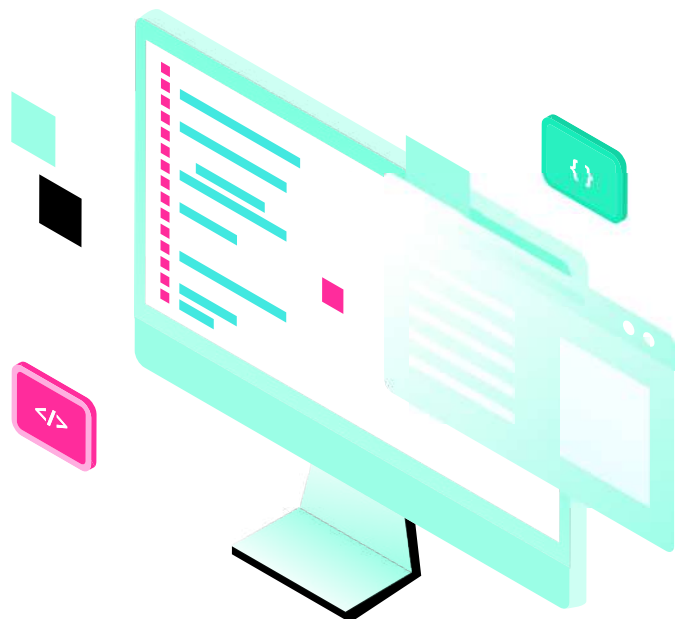
Teams are writing and storing their infrastructure in code, usually in version control systems like Git. That's meaningful progress. But it's also where progress often stalls.

At this stage, infrastructure changes are still deployed manually or triggered by scripts without proper CI/CD integration. There's no consistent process to validate what gets deployed or enforce how it should be written. Security, if addressed at all, shows up late after a Jira ticket, an alert, or worse, an incident.

The result is an uncomfortable tension. Developers are responsible for infrastructure but lack the context or time to get security right. Security teams flag issues but rarely have a clear or scalable way to fix them. Misconfigurations stack up. Alerts become noise. And the backlog of unresolved security findings grows faster than anyone can clear it.

Here's the thing: this is normal. Most organizations find themselves at this inflection point. They've built enough structure to see what's broken, but not enough automation or process to fix it at scale. Level 2 isn't a failure; it's a signal. It means the foundation is in place, and the opportunity to evolve is right in front of you.

With the right guidance, the leap from reactive to proactive isn't about overhauling everything at once. It's about making smart, incremental improvements, integrating security into pipelines, automating fixes, and shifting from alerts to action. And it starts right here.





THE POWER OF PICKING YOUR OWN PATH

It's tempting to view security maturity as a ladder that must be climbed step by step, in perfect order, without slipping. But in reality, the journey isn't that clean. Most organizations don't move uniformly across every area. You might have version-controlled infrastructure but no CI/CD integration. You might enforce encryption everywhere but still rely on manual remediation. Maturity doesn't happen all at once, and that's okay.

That's why we encourage a different perspective: don't chase perfection. Start where you are and pick the next step that makes the most sense for you.

Security maturity isn't about checking every box or reaching some distant ideal. It's about making sustainable progress from where you are, focusing on what's actionable, practical, and tailored to your environment.

By reframing the journey as a guided path, not a rigid hierarchy, you gain permission to act without being overwhelmed. You stop waiting for a perfect state that may never come and instead begin improving with what you already have.





LEVEL-BY-LEVEL PATHWAYS

When advancing IaC security maturity, there's no universal playbook, but there are clear patterns. Across hundreds of teams, we've seen common signals defining each growth stage. By understanding these stages in practice, you can identify your current state and make informed decisions about where to go next.

This section is designed to help you do exactly that. We break down each maturity model level with a real-world lens, what it looks like on the ground, what's likely getting in your way, and what success could look like just one step ahead.

You'll also find smart, achievable next steps for each level. These aren't sweeping overhauls or idealized best practices. They're practical moves grounded in how real teams work, build, and deploy.

At each stage, we'll show you how Gomboc can help accelerate progress by automating painful manual work, reducing risk, and embedding security directly into your development workflows.

Whether you're still applying infrastructure changes by hand or building a full CI/CD pipeline with policy-as-code, there's always a next step.

You can advance with clarity, purpose, and momentum by mapping that step to where you are today, not where you wish you were.





LEVEL 1: EMERGENT

BUSINESS GOAL

Establish foundational control over infrastructure through code and reduce risk from untracked manual changes.

PATH FORWARD:

Begin adopting Infrastructure as Code tools and storing configurations in version control.

IMPACT

Enables repeatable, reviewable changes and lays the groundwork for automation and security enforcement.

Every journey has a beginning; for many teams, that starting point is the Emergent stage. At this level, infrastructure is still largely manual. Engineers spin up cloud resources through web consoles or CLI commands. There's little to no version control, and changes often happen directly in production. Without a clear infrastructure management system, visibility and risk are high.

This stage isn't about negligence. It's about exploration. Teams are moving fast, testing ideas, and building products. But as environments grow and become more complex, the cracks start to show. Manual changes become harder to track.

Teams add manual change control processes to handle the increased fragility, but still, drift creeps in as manual processes cannot keep up with the speed of development. Misconfigurations can go unnoticed without guardrails until something breaks or, worse, becomes exposed.

The good news? **The path forward is clear.**

The first smart move is adopting Infrastructure as Code tools like Terraform or CloudFormation. These give you the structure to define infrastructure declaratively, making it easier to track, review, and reuse.

Gomboc helps you build secure cloud infrastructure from the start, automatically analyzing your code and generating precise, infrastructure-as-code fixes as you write. Whether starting your first IaC project or scaling a mature environment, Gomboc integrates directly into your development workflow (including IDE plugins) to prevent misconfigurations before they're deployed. Instead of flooding developers with tickets, Gomboc delivers ready-to-merge pull requests that align with best practices, enabling teams to avoid painful rewrites and ship confidently, without needing deep security expertise..



LEVEL 1: EMERGENT

If you're at Level 1, you don't need to solve everything at once. But you do need to start building a foundation.

MATURITY FACTOR	LEVEL 1 FOCUS	GOMBOC'S ROLE
Infrastructure Mgmt	Manual creation via console or CLI; little to no repeatability	Gomboc analyzes environments and generates IaC-based fixes to bootstrap the structure
Versioning	Minimal or no version control; infrastructure is not tracked in Git	Gomboc provides guidance on how to structure your IaC and get the most of your SCM integration.
Delivery Automation	No CI/CD; changes applied manually	No pipeline required—Gomboc fixes are delivered in a way that teams can adopt incrementally
Security Integration	No security validation until after deployment (if at all)	Gomboc identifies misconfigs early and suggests secure-by-default IaC
Policy & Governance	No formal guardrails or enforcement mechanisms	Gomboc introduces structure with pre-built policy rules aligned with well-known standards
Remediation Approach	Ad hoc, reactive manual fixes when issues are noticed	Gomboc automates fixes via copy-pasteable or ready-to-merge IaC recommendations



LEVEL 2: REACTIVE

BUSINESS GOAL

Reduce the risk of manual changes breaking deployments and slow remediation of security issues.

PATH FORWARD:

Integrate infrastructure changes into CI pipelines and automate remediation for common misconfigurations.

IMPACT

Improves security posture and dev velocity by resolving issues earlier and reducing security ticket backlog.

By the time a team reaches Level 2: Reactive, they've made real progress. Infrastructure is now written as code and stored in Git. There's a shared repository, a growing number of contributors, and a stronger sense of control over how environments are defined. But while version control has brought order to the chaos, delivery is still largely manual, and security remains an afterthought.

Deployments rely on ad hoc scripts. Code is tracked, but without consistent automation to validate or enforce changes. Security reviews happen after the fact, if they happen at all. A misconfiguration gets flagged during an audit or by an alert, resulting in a ticket. And another ticket. And another.


Delivery deadlines are a prime driver of these problems. To meet deadlines, companies accept delayed remediation of security issues with the intent of returning for later remediation. These promises are not always kept, building tech debt in the form of tickets.

Over time, those tickets start to pile up. Backlogs grow. Engineering slows. Security frustrations mount. This is the most common stage we see across teams of all sizes. They've embraced IaC but haven't yet connected it to the systems that keep infrastructure secure and scalable. And that disconnection is precisely where the opportunity lies.




LEVEL 2: REACTIVE


The goals at this stage are clear:



Shrink the security backlog by resolving repeat issues quickly.



Shift security left by integrating checks into the CI pipeline.



Automate remediation of common misconfigurations to reduce developer burden.

That's where Gomboc makes an immediate impact. Instead of flooding teams with alerts or handing off ambiguous tickets, Gomboc transforms security findings into ready-to-merge pull requests. These PRs are context-aware, tailored to your environment, and provide explanations so developers know what's being fixed and why. With each approved PR, you reduce the backlog, improve security posture, and build a faster feedback loop between code and compliance. Security becomes just another part of the workflow, not a blocker or burden.

Level 2 is a tipping point. You've got the foundations in place. Now is the time to scale your efforts without scaling your pain. And with the proper tooling, you can move from reactive fixes to proactive resilience, one automated PR at a time.

FACTOR	LEVEL 2 FOCUS	GOMBOC'S ROLE
Infrastructure Mgmt	Code in Git, but deployed manually	Delivers PRs into Git; no infra overhaul needed
Versioning	Shared repo, branching, growing contributors	PR-based fixes with explanations
Delivery Automation	Ad hoc scripts, no consistent pipeline	Gomboc adds structure without needing full CI/CD
Security Integration	Alert → ticket → backlog	Eliminates tickets with automated PRs
Policy & Governance	Limited or inconsistent enforcement	Policies are optional at this stage
Remediation Approach	Developer manually fixes misconfigs	Gomboc remediates automatically with full developer control



LEVEL 3: PROACTIVE

BUSINESS GOAL

Prevent security drift and reduce time to remediate policy violations.

PATH FORWARD:

Embed policy checks and auto-remediation into CI/CD pipelines with infrastructure-aware tools like Gomboc.

IMPACT

Ensures changes stay aligned with security expectations, slashing MTTR and improving compliance adherence.

At Level 3, the foundations of Infrastructure as Code are no longer experimental; they're operational. CI/CD pipelines are beginning to take shape. Code changes trigger builds, and infrastructure updates are starting to flow through more consistent, automated processes. This is where IaC starts to feel like a real software discipline for many teams.

Security is no longer entirely reactive. You've likely introduced basic policy-as-code or used static analysis tools to catch common misconfigurations.

Preventive controls might flag issues before they reach production, and your team is starting to think seriously about metrics like Mean Time to Remediate (MTTR). The question shifts from "Is our infrastructure secure?" to "How quickly can we secure it when something breaks?"

But even with these improvements, gaps remain. Policy checks might run in CI, but enforcement is often inconsistent. Drift between the intended and deployed state is still a risk. While issues may be caught earlier, resolving them usually depends on developers manually researching and rewriting infrastructure code, delaying fixes, and reintroducing the backlog problem in a new form.

This is the moment to move from identifying issues to resolving them automatically. Gomboc helps teams at this stage make that leap. By integrating into your pipelines and source control, Gomboc continuously scans your infrastructure, the code, and the live environment. It detects policy violations and drift and then generates compliant-by-default pull requests to remediate issues before they're deployed or diverged.



LEVEL 3: PROACTIVE

You define the policies. Gomboc enforces them at scale without slowing down your team. Level 3 is about momentum. You're no longer fixing things after the fact. You're building systems that stay secure by design. With automation, drift detection, and integrated fixes, you can begin to shift security left in a way that sticks and finally close the gap between policy and practice.

MATURITY FACTOR	LEVEL 3 FOCUS	GOMBOC'S ROLE
Infrastructure Mgmt	IaC is standardized and lives in Git; infra changes flow through structured code reviews	Gomboc continuously scans your code to proposes IaC fixes as PRs, aligned with team standards and source-controlled workflows
Versioning	Mature Git usage with branching, reviews, and approvals	Gomboc integrates directly into GitHub/GitLab/etc., tagging owners and including context with each PR
Delivery Automation	CI/CD pipelines trigger builds and deployments automatically	Gomboc scans pipeline-triggered changes and remediates issues contextually before they ship
Security Integration	Security begins to shift left; checks run during CI, but remediation is still manual	Gomboc closes the loop between detection and remediation by integrating with your IDE to suggest policy-aligned fixes as you code.
Policy & Governance	Early policy-as-code or compliance baselines are enforced, but not uniformly	Gomboc applies policy-aligned fixes that reinforce early governance goals
Remediation Approach	Issues are found earlier, but are still resolved manually by developers	Gomboc generates ready-to-merge, compliant-by-default IaC fixes that reduce MTTR and dev burden



LEVEL 4: ADAPTIVE

BUSINESS GOAL

Scale security governance across multiple teams and pipelines without slowing down delivery.

PATH FORWARD:

Implement policy-as-code and automated guardrails with auditable enforcement at the pull request level.

IMPACT

Security becomes embedded and trusted, reducing friction while enabling consistent policy enforcement at scale.

At Level 4, infrastructure and security processes are no longer in conflict. They're in sync. CI/CD workflows are fully automated, and policy-as-code is integral to the development lifecycle. Every infrastructure change flows through a pipeline that enforces security and compliance without slowing delivery. Teams at this level have started to balance speed with control.

This is where maturity begins to show. DORA metrics like deployment frequency, lead time for changes, and mean time to restore are tracked and actively used to guide decisions. Teams understand the tradeoffs between risk and velocity and have systems to manage both. The focus shifts from visibility to governance at scale.

That scale introduces new challenges. As more teams adopt IaC and pipelines multiply, enforcing consistency across environments becomes harder. Security teams can't review every change manually, and developers need enough autonomy to move quickly without compromising standards.

This is where guardrail automation becomes critical. Rather than gating progress, security needs to run alongside it, flagging violations and applying corrections early, ideally before the code even reaches production. However, it also needs to be flexible: not every violation is fatal, and not every environment requires the same rules.

Gomboc enables this balance. At the Adaptive stage, Gomboc acts as a force multiplier, scaling IaC governance across teams and pipelines. It continuously enforces policies and remediates violations with automated, auditable pull requests. Every fix includes context and traceability, giving security teams the assurance they need while leaving final approval in the hands of human reviewers.



LEVEL 4: ADAPTIVE

The result is a system where developers retain ownership but operate within a secure, self-correcting framework. Gomboc becomes the safety net that lets teams move fast without falling off the edge, automating enforcement, eliminating drift, and providing audit-ready reporting for every change.

In Level 4, the organization moves from doing security well to doing it reliably at scale. Governance becomes embedded, automation becomes trusted, and security becomes a seamless part of how infrastructure is built, shipped, and maintained.

MATURITY FACTOR	LEVEL 4 FOCUS	GOMBOC'S ROLE
Infrastructure Mgmt	Fully automated, multi-team aC workflows; standardized modules and templates used org-wide	Gomboc integrates seamlessly across teams, environments, and IaC tools with consistent, reusable fixes
Versioning	All changes are tracked via Git; cross-team collaboration and enforcement are critical	Gomboc provides audit-trail PRs with full policy alignment and change context for transparent approvals
Delivery Automation	Mature CI/CD processes; pipelines are the only way code is shipped	Gomboc enforces policy compliance pre-merge or pre-deploy through CI/CD integrations (e.g., Jenkins, GitHub Actions)
Security Integration	Security as part of every pipeline; preventive controls are expected at the commit or PR level	Gomboc applies guardrails and remediations before drift or risk can propagate, automatically matching CSPM alerts to code and eliminating alert limbo.
Policy & Governance	Policy-as-code is enforced org-wide; traceability and audit readiness become primary concerns	Gomboc auto-enforces policies via IaC fixes and generates audit-ready records for every remediation. Gomboc enables you to codify custom policies that match specific conditions to your enterprise environment.
Remediation Approach	Fixes must be fast, auditable, and automated—manual security review no longer scales.	Gomboc delivers pre-approved, policy-compliant fixes that scale across teams without human bottlenecks.



LEVEL 5: RESILIENT

BUSINESS GOAL

Maintain continuous, audit-ready compliance and minimize risk without increasing overhead.

PATH FORWARD:

Operationalize self-healing infrastructure using drift correction, policy updates, and end-to-end traceability.

IMPACT

Infrastructure becomes compliant and secure by default, freeing teams to innovate without rework or oversight bottlenecks.

At Level 5, Infrastructure as Code isn't just a workflow. It's the operating model. Every environment, every change, and every security control flows through code. IaC has become the single source of truth across the organization, governing how infrastructure is provisioned, validated, and secured. What once required coordination across multiple teams is now codified, automated, and scalable.

At this level of maturity, automated drift detection and correction are no longer nice to have but essential.

They ensure infrastructure remains aligned with policy and prevent configuration changes from introducing risk or compliance gaps. If a resource changes outside the expected path, whether through a console click, a misfired script, or a third-party tool, it's automatically flagged and reconciled. Infrastructure is not just deployed securely. It's maintained that way continuously.

And yet, maturity here doesn't mean rigidity. In resilient organizations, security doesn't slow teams down. It enables them. Developers operate within self-service environments, provisioning and iterating quickly but always within defined guardrails. Those boundaries aren't manually policed; they're enforced through code and context, with automated validation and correction ensuring the environment stays compliant without intervention.

The real hallmark of Level 5 is alignment. Security, operations, and development are no longer separate lanes. They share code, context, and goals. Infrastructure changes are reviewed and approved in the same systems as the application code. Policies are written in a way that both security engineers and developers understand. Compliance is demonstrated continuously, not just during audits.



LEVEL 5: RESILIENT

Reaching Level 5 doesn't mean the journey is over. It means you've built a system that can adapt, recover, and improve on its own, an infrastructure security program that's not just compliant or automated but truly resilient.

MATURITY FACTOR	LEVEL 5 FOCUS	GOMBOC'S ROLE
Infrastructure Mgmt	Infrastructure is fully defined, deployed, and governed through code; IaC is the operating model	Gomboc maintains infrastructure integrity by enforcing policy-compliant, IaC-based changes continuously
Versioning	Everything is version-controlled; change management is standardized across teams	Gomboc contributes fully traceable PRs with policy references and contextual documentation
Delivery Automation	Pipelines handle all changes; self-service provisioning is enabled within guardrails	Gomboc works behind the scenes in CI/CD to detect and fix issues pre- and post-deploy without friction
Security Integration	Security is invisible but ever-present—embedded, automated, and self-healing	Gomboc identifies and corrects drift automatically, minimizing risk exposure without developer involvement
Policy & Governance	Compliance is enforced continuously, demonstrated through audit trails and policy alignment	Gomboc ensures continuous enforcement from IDE to production, integrates updates from evolving frameworks, and provides audit reporting
Remediation Approach	Security fixes are self-applying; systems correct themselves in real-time	Gomboc enforces the desired state, generates fixes automatically, and minimizes MTTR to near zero



GOMBOC'S ROLE ACROSS THE MATURITY MODEL

No matter where you are on the IaC maturity spectrum, from getting started with version control to managing fully automated pipelines, Gomboc delivers tangible impact at every stage. At early levels, it helps you adopt secure practices from the outset by guiding and correcting infrastructure code in real time. As maturity increases, Gomboc shifts from supportive to strategic: automating complex remediations, maintaining continuous compliance, and scaling policy enforcement across environments. This progression makes it easy to map specific Gomboc capabilities to each stage of IaC maturity, enabling a clear view of how it accelerates growth, reduces operational friction, and supports long-term security outcomes.

MATURITY LEVEL	MTTR	OVERHEAD	RISK	DEV VELOCITY	SECURITY BACKLOG	HOW GOMBOC HELPS
Level 1: Emergent	High (Manual fixes)	High (reactive work)	High (no visibility)	Low (high friction)	Growing rapidly	<ul style="list-style-type: none"> ■ Auto-generates secure IaC fixes ■ Removes need for deep expertise ■ Kickstarts remediation safely
Level 2: Reactive	Moderate	Lower (some tooling)	Lower (basic checks)	Moderate	Plateaued	<ul style="list-style-type: none"> ■ Converts findings to PRs ■ Explains changes inline ■ Delivers fixes into existing Git workflows
Level 3: Proactive	Reduced via automation	Lower (fewer reworks)	Medium (preventive controls)	Improved	Actively shrinking	<ul style="list-style-type: none"> ■ Integrates with pipelines ■ Auto-remediates misconfigs ■ Fixes drift continuously



GOMBOC'S ROLE ACROSS THE MATURITY MODEL

MATURITY LEVEL	MTTR	OVERHEAD	RISK	DEV VELOCITY	SECURITY BACKLOG	HOW GOMBOC HELPS
Level 4: Adaptive	Low	Significantly reduced	Low (guardrails in place)	High	Minimal	<ul style="list-style-type: none"> Enforces policy-as-code Generates audit-ready fixes Scales security across teams
Level 5: Resilient	Near Zero	Optimized	Very low (continuous enforcement)	Very high (self-service)	Zero to near-zero	<ul style="list-style-type: none"> Continuous drift correction Auto-updates with new policies Secures infra with zero friction

At **Level 1**, Gomboc helps you take the first real step toward security. Analyzing your environment and generating ready-to-use Infrastructure as Code fixes give teams without deep security expertise a safe, fast way to begin remediating code misconfigurations without starting from scratch.

At **Level 2**, where infrastructure lives in Git but deployments are still manual, and security tickets pile up, Gomboc flips the script. Instead of flagging issues and leaving them for engineers to solve, it translates security findings into developer-ready pull requests. Each change is explained, policy-aligned, and delivered straight into your workflow, whether you use GitHub, GitLab, Bitbucket, or another SCM.

At **Level 3**, Gomboc becomes a proactive part of your CI/CD process. It integrates into build pipelines like Jenkins or GitHub Actions to catch misconfigurations early. More importantly, it pairs detection with automated remediation and drift correction, helping your infrastructure stay secure even as it changes.



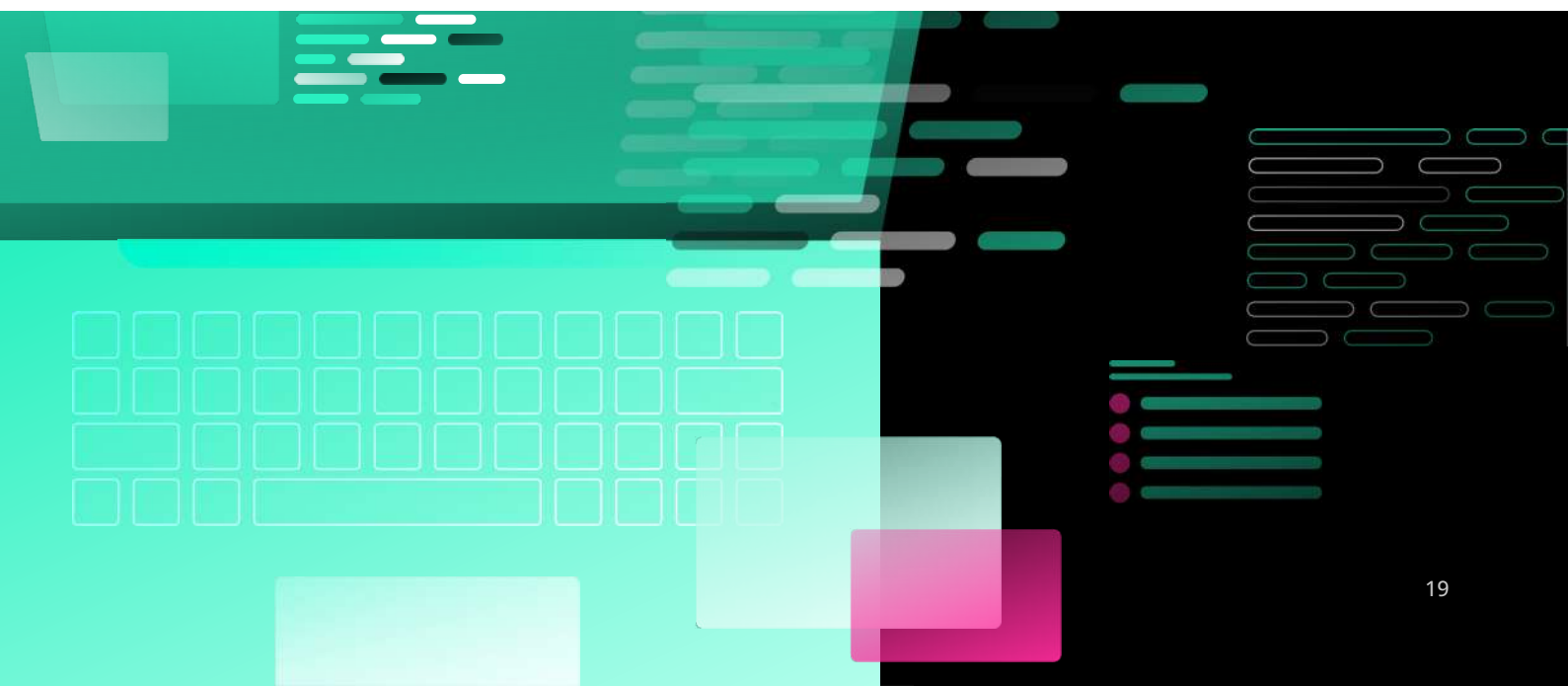
GOMBOC'S ROLE ACROSS THE MATURITY MODEL

At **Level 4**, Gomboc helps you scale governance. With policy-as-code enforcement, audit-ready reports, and guardrails that operate automatically but leave approval in your hands, Gomboc enables consistent enforcement across teams without sacrificing developer velocity. Every change is traceable, every fix is reviewable, and every step aligns with your security and compliance goals.

By **Level 5**, Gomboc becomes part of your operating fabric. Its “self-righting” capabilities enforce the desired state continuously, catching and correcting drift, updating policies with new benchmarks, and embedding security directly into how teams work. With centralized logging, policy traceability, and integration across your toolchain, Gomboc ensures your infrastructure remains compliant, resilient, and responsive without adding friction.

By **Level 5**, Gomboc becomes part of your operating fabric. Its “self-righting” capabilities enforce the desired state continuously, catching and correcting drift, updating policies with new benchmarks, and embedding security directly into how teams work. With centralized logging, policy traceability, and integration across your toolchain, Gomboc ensures your infrastructure remains compliant, resilient, and responsive without adding friction.

From the first steps to full automation, Gomboc is not just a tool. It's a partner in your security maturity journey, closing the gap between intention and implementation without slowing down your team.





BUSINESS IMPACT AT EVERY STAGE

Investing in IaC security maturity isn't just about improving your technical posture. It's about driving real business results. At every journey stage, the impact is measurable: faster remediation, lower operational costs, reduced risk exposure, and happier, more productive teams.

As organizations mature, their mean time to remediate (MTTR) shrinks. What once took days or weeks to fix is now resolved automatically through pull requests. Costs drop as manual rework is replaced by policy-driven automation, and security incidents are prevented rather than cleaned up. Risk is no longer something you respond to. It's something you control.

Development velocity improves, too. When developers don't have to sift through noisy alerts or unclear tickets, they can stay focused on building. Security shifts from a blocker to a silent partner, working in the background, surfacing only what matters, and offering fixes right where engineers work.

Perhaps most importantly, the backlog disappears. Gomboc replaces endless ticket queues with clear, actionable PRs, turning the security backlog into a thing of the past. Instead of asking developers to do more, it empowers them to do less because the right thing is already done.

At each stage, Gomboc helps teams progress with fewer tickets, faster recovery, and stronger compliance. It's not about being perfect. It's about getting better, one step at a time.

So, wherever you are today, the next step is within reach.



Book a Gomboc demo to see how automated remediation works in action



Try it yourself to explore how Gomboc can help your team move faster, securely

Security maturity isn't just about where you're going. It's about knowing where to begin. And with the right partner, that next step becomes much easier.