Platform Engineering Playbook

Building Internal Developer Platforms with Backstage, Scorecards & Golden Paths

Version 1.0 - June 2026

1. Introduction: Why Platform Engineering

Platform engineering is the discipline of building internal developer platforms (IDPs) that abstract infrastructure complexity and provide self-service capabilities to development teams. It is not about building a UI on top of Kubernetes - it is about creating a product that makes developers productive, secure, and happy without needing deep infrastructure knowledge.

- Developers waste 30-40% of their time on infrastructure and operational concerns
- Security and compliance teams bottleneck deployments without automated guardrails
- Cloud complexity grows faster than teams can keep up - multi-cloud, K8s, 100+ services
- Organisations with mature platform engineering ship 2x faster with 50% fewer incidents (DORA 2025)

You need a platform team when: your engineering org exceeds 20 developers, you have multiple product teams sharing the same infrastructure, or deployment velocity is slowing despite DevOps investment.

2. The Four Pillars of an IDP

An Internal Developer Platform rests on four pillars that together create a cohesive developer experience.

- Pillar 1: Developer Portal - The front door. A single UI where developers discover services, request resources, view documentation, and manage their applications. Backstage (by Spotify) is the market leader with 10,000+ organisations using it.

- Pillar 2: Golden Paths / Scaffolding - Pre-defined, approved templates that create production-ready applications with built-in compliance. A 'Node.js microservice' golden path creates a repo with CI/CD, Dockerfile, Terraform, monitoring, and security scans.

- Pillar 3: Scorecards & Governance - Automated quality gates that score services against organisational standards. Examples: 80% test coverage, no critical vulnerabilities, documented runbook.

- Pillar 4: Self-Service Infrastructure - Developers provision infrastructure through the portal without needing cloud console access. Terraform, Crossplane, or Pulumi manages resources; the portal provides the interface.

3. Backstage: The Developer Portal Backbone

Backstage is an open-source platform for building developer portals. Originally created by Spotify, it is now a CNCF graduated project with a large plugin ecosystem. The core concepts are:

- Software Catalog - a unified view of all services, resources, and teams
- TechDocs - documentation-as-code with Markdown, hosted in Backstage
- Software Templates - self-service scaffolding for new projects
- Plugins - 150+ community plugins for CI/CD, monitoring, security, and more
- Scaffolder - customisable action workflows for software templates

Backstage runs on Node.js/TypeScript with a PostgreSQL backend. It can be self-hosted or used via a managed offering. The plugin architecture makes it extensible.

4. Scorecards & Governance

Scorecards transform governance from a bottleneck into an enabler. Instead of a Cloud Center of Excellence manually reviewing every deployment, automated checks run continuously and developers can see their score improving in real-time.

- Start with 5-10 checks per service type. Not 50. Iterate.
- Each check must be actionable: 'Fix this now' not 'This is important'
- Score out of 100 with clear thresholds: Red (<60), Amber (60-80), Green (>80)
- Review scorecards quarterly - remove checks that are always green

Example: Security: No critical vulnerabilities (30pts) | Observability: Logs, metrics, traces (20pts) | Reliability: Multi-region failover tested (20pts) | Compliance: Encryption at rest+in transit (15pts) | Operations: On-call runbook, SLIs/SLOs defined (15pts)

5. Golden Paths

Golden paths are approved, production-ready templates that encode organisational best practices.
- Source code template with recommended structure, linting, testing framework
- Infrastructure-as-code (Terraform / Bicep) for networking, compute, storage
- CI/CD pipeline (GitHub Actions / Azure DevOps) with security scanning
- Monitoring and alerting configuration (Prometheus / Azure Monitor)
- Documentation stub with runbook template

Maturity Model: Level 1 - Copy-paste instructions | Level 2 - Scripts and templates | Level 3 - Portal-integrated (Backstage template creates everything in one click) | Level 4 - Self-healing (platform auto-detects drift and remediates)

6. Measuring Platform Success

Adopt DORA metrics for deployment velocity and SPACE framework for developer satisfaction.

- Deployment Frequency - How often do teams deploy to production?
- Lead Time for Changes - From commit to production, how long?
- Change Failure Rate - What percentage of deployments cause incidents?
- Time to Restore Service - How long to recover from failures?
- Self-service adoption: % of infrastructure provisioned via portal
- Scorecard compliance: % of services meeting minimum thresholds
- Developer satisfaction: Net Promoter Score for the platform

7. Getting Started: The 90-Day Plan

A phased approach to building your internal developer platform:

- Days 1-30: Foundation - Identify your platform team (3-5 people). Deploy Backstage with Software Catalog. Create first 3 golden paths. Set up initial scorecards.

- Days 31-60: Expansion - Integrate CI/CD pipelines. Enable self-service infrastructure for dev/test. Add TechDocs. Run developer feedback sessions.

- Days 61-90: Production-Ready - Extend golden paths to production with compliance guardrails. Build custom Backstage plugins. Launch to engineering org.

8. Common Pitfalls

Avoid these frequent mistakes when building a platform:

- Building a Platform Nobody Uses - The #1 failure mode. Solution: embed with product teams, treat the platform as a product with user research.

- Too Many Golden Paths Too Fast - Start with 3, get adoption, iterate. Each golden path needs ongoing maintenance.

- Scorecard Bloat - 50-point scorecards with unactionable checks. Keep scorecards lean, actionable, reviewed quarterly.

- Treating It as an Engineering Project - No product manager, no roadmap = a platform that accumulates features nobody asked for.

9. Tool Comparison: Backstage vs Alternatives

Choosing the right platform tooling for your organisation:

- Backstage - Open-source portal, 150+ plugins, self-hosted. Best for enterprise platforms with custom needs.
- Port - Managed SaaS portal with catalog and scaffolding. Best for teams wanting to start fast.
- Humanitec - Platform orchestrator with 20+ integrations. Best for Kubernetes-native platforms.
- AWS Proton - AWS portal with limited catalog. Best for AWS-only shops.
- Custom in-house - Full control, heavy maintenance. Best for unique requirements at large scale.

Recommendation: Start with Backstage for most enterprises. Largest community, most plugins, avoids vendor lock-in.

10. Glossary & References

Key terms and further reading:

- IDP - Internal Developer Platform. The product platform teams build for developers.
- Backstage - CNCF-graduated developer portal by Spotify.
- Golden Path - Pre-approved, production-ready template for creating services.
- Scorecard - Automated quality checks that score services against standards.
- DORA - DevOps Research and Assessment metrics.
- SPACE - Framework for measuring developer productivity.

References: 'Team Topologies' by Skelton & Pais | 'Accelerate' by Forsgren, Humble & Kim | backstage.io/docs | dora.dev | CNCF Platform Engineering Maturity Model