ENGINEERING SYSTEMS | OPEN-SOURCE TOOLING

Production-minded engineering, shown in the work.

CaeliCode Solutions brings together public work across cloud infrastructure, DevOps/SRE, security automation, developer tooling, and reliable software delivery. Capability claims are paired with inspectable implementation wherever the underlying work can be shared.

7	3	216	OPEN
public repositories	released tools	status-platform tests	source and history

Capability map

Systems behind reliable software.

01

Cloud platforms

Infrastructure code, identity boundaries, network foundations, and operable environments.

02

DevOps and SRE

Delivery pipelines, observability, reliability practice, and incident readiness.

03

Security engineering

Threat-aware automation, guardrails, evidence, provenance, and reviewable controls.

04

Software delivery

APIs, internal tools, product surfaces, integrations, and maintainable handoff.

Selected public engineering work

PROJECT	WHAT IT DEMONSTRATES	EVIDENCE
Status platform	Monitoring, tested component state, workflow automation, Pages publishing, and optional incident synchronization.	**Python 216 tests**
WSL builder	Profile-driven environments, pinned tools, verified updates, rollback, DNS handling, and SSH agent forwarding.	**Shell releases**
SSH action	Native OpenSSH automation with multi-host execution, bastion support, timeouts, and captured output.	**Shell releases**
Code review action	Configurable pull-request review with focused inline feedback and low-noise output.	**JavaScript source**

REPOSITORY-BACKED DEMONSTRATION

A reliability workflow, shown end to end.

The status platform connects synthetic monitoring, tested state evaluation, automation, and versioned publishing. The workflow is presented as public engineering evidence, not as an anonymous client or production claim.

01	02	03	04
Observe	**Evaluate**	**Publish**	**Synchronize**
Grafana synthetic checks	Tested component-state logic	GitHub Actions and Pages	Optional incident integration

Evidence standard

Claims should resolve to something inspectable.

01	**Source**	Implementation and commit history
02	**Verification**	Tests, releases, and automated workflows
03	**Operations**	Failure handling, observability, and recovery paths
04	**Communication**	Decisions, boundaries, and supporting documentation

START HERE

A concise engineering review

1. Open selected work.
2. Trace one workflow.
3. Review capability boundaries.
4. Follow the supporting reasoning.

PUBLIC LINKS

caelicode.com
github.com/caelicode
caelicode.github.io

Engineering inquiries: support@caelicode.com