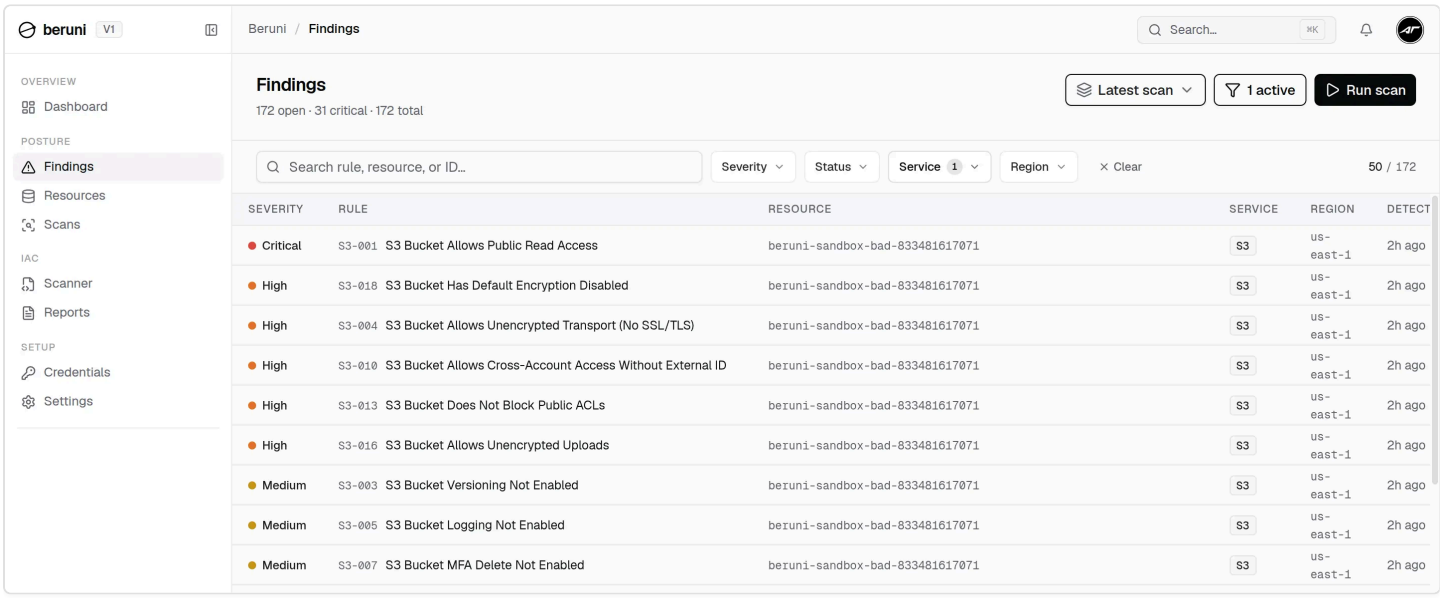
FOR AWS-NATIVE SAAS TEAMS · 20–200 PEOPLE · PRE-SERIES B

The AWS scan for startups, not enterprises.

Built for the team that can't justify Wiz.

Beruni assumes a read-only IAM role in your AWS account, evaluates IAM, S3, EC2, VPC, and RDS configurations against a curated rule library, and returns ranked findings in about a minute on a 50-resource account. No agents, no sidecars, no access keys. The role uses AWS's `SecurityAudit` managed policy, pinned by an ExternalId you can rotate to revoke us instantly. Every finding cites the exact resource, the exact rule, and the AWS CLI command to close it.



FINDINGS · SCANNED AGAINST A SEEDED DEMO ACCOUNT SEVERITY · RULE · RESOURCE · SERVICE · REGION · TIME

01 / SETUP

Connected in a CloudFormation stack.

Deploy a read-only role pinned to your ExternalId. No agents in your account, no access keys to rotate.

02 / SURFACE

Findings in about a minute.

IAM, S3, EC2, VPC, RDS — the five services behind most AWS breaches. ~1 min on a 50-resource account.

03 / TRUST

Open rule library, auditable.

Every check is a JSON file with the AWS field it inspects and the controls it traces to. No proprietary DSL.

IAM

Over-permissioned roles. `Action: "*"` nobody flagged.

S3

Public buckets. The five-minute test someone forgot to close.

EC2

Open ingress. Port 22 to `0.0.0.0/0`.

VPC

Security-group mistakes exposing what you thought was private.

RDS

Public databases, no encryption. `publicly_accessible = true`.

04 / START

Free during closed beta. No credit card.

A read-only role, an ExternalId, and the first scan returns in about a minute.

`beruni.app — Join the waitlist →`

Or talk first — `demo.beruni.app`

05 / CAPABILITIES — SHIPPED

Agentless scanning	Read-only IAM role assumed via STS. Nothing installed in your account, no sidecars, no access keys to rotate.
Five AWS services	IAM, S3, EC2, VPC, RDS — the surface area where most AWS breaches happen, and roughly the entire stack a SaaS startup runs day one.
Findings & triage	Every row links to the exact resource, the rule it fired on, the AWS field that tripped it, and the AWS CLI commands to close it.
Compliance citations	Each finding traces to controls in CIS AWS Foundations, AWS FSBP, PCI DSS, SOC 2, HIPAA, and GDPR.
Open rule library	Every check is a JSON file. No proprietary expression language. Audit, fork, or extend the rules.

05.1 / ON THE ROADMAP

IaC scanning ROADMAP	Terraform and CloudFormation templates checked against the same rule library — before they ship.
Notifications ROADMAP	Slack, email, and webhook alerts when new findings appear or severity escalates.
Scheduled reports ROADMAP	Weekly findings digests and compliance-ready exports for auditors.

07 / COMPLIANCE — FINDINGS CITE

- CIS AWS Foundations v3.0.0
- AWS FSBP
- PCI DSS v3.2.1
- SOC 2
- HIPAA
- GDPR

08 / TRUST — WHAT THE ROLE CAN

Read-only IAM
AWS's `SecurityAudit` managed policy. `List*`, `Describe*`, `Get*` on the five services. Nothing that changes state.

ExternalId pinning
Your role only trusts our scanner principal when called with your ExternalId. Rotate it and we're locked out instantly.

Nothing written
We read configuration and emit findings. Your AWS account is never modified by us.

Nothing exfiltrated
We store the configuration we inspected. Object data inside your buckets is never opened.

Open by design
The rule library is auditable JSON. No proprietary DSL. Fork, extend, or audit the checks yourself.

06 / HOW IT WORKS — CONNECT · SCAN · FIX

01 / CONNECT
Deploy.
Launch our CloudFormation stack. AWS creates a read-only role pinned to your ExternalId.

02 / SCAN
Inspect.
We `AssumeRole`, collect configuration across five services, evaluate against the rule library.

03 / FIX
Close.
Open a finding: rule, AWS field that tripped, risk in plain English, AWS CLI command to remediate.

09 / SEVERITY — COLOR CARRIES SEVERITY AND SEVERITY ONLY

CRITICAL
Public exposure, missing encryption on production data, world-open ingress.

HIGH
Over-permissioned IAM, unencrypted backups, missing access logging.

MEDIUM
Weak password policy, MFA gaps on non-privileged users, stale keys.

LOW
Hardening recommendations, tagging gaps, lifecycle policy hygiene.