IMPLEMENTATION PLAN

On-premises and private deployment

What gets stood up, in what order, and who does each part.

Pattern The whole platform, inside your perimeter

For Sovereign, air-gapped and regulated on-premises estates

IRON FORT YOU TOGETHER

You run everything, so this reads more like a platform deployment than a SaaS onboarding. Our system integrator partners do this work on request. Sequence matters more here than anywhere else: the supply chain and the directory have to be settled before anything useful can be deployed.

BEFORE YOU START

- A target platform: OpenShift, self-managed Kubernetes, or a Linux VM cluster.
- A registry you already run — Docker Hub, Amazon ECR, Azure ACR, GitLab Registry or Azure DevOps Artifacts.
- A directory: Microsoft Entra ID, Google Workspace, Okta, or any SAML 2.0 provider.
- PostgreSQL, and storage for evidence — object or filesystem.
- A named platform team, or an engagement with one of our system integrator partners.

1

Supply chain YOU

- Mirror our signed images into your registry and verify the signatures.
- Agree the promotion path: who moves an image from staging to production, and on what schedule.
- For air-gapped estates, confirm the media transfer process before you need it. This is the step that surprises people.

DONE WHEN

Images are in your registry and you can reproduce the pull without us.

2

Data services and identity YOU

- Provision PostgreSQL with your own backup and restore process — and test the restore.
- Provision evidence storage: an S3-compatible object store, or simply a mounted filesystem on NFS, a SAN volume or local disk.
- Configure sign-in against your directory over SAML 2.0 or OIDC. A disconnected site cannot reach Microsoft or Google, which is why this is the one pattern where the directory is yours.

DONE WHEN

A test user signs in against your directory and the database survives a restore drill.

3

Deploy the platform TOGETHER

- Deploy the application containers by Helm chart or Operator, or as systemd units on a VM cluster.
- Wire in your secrets backend — Vault, Kubernetes secrets or an HSM.
- Point ingress at your own certificates and DNS. Nothing about this deployment reaches outside your perimeter.

DONE WHEN

The application is reachable internally and holds no configuration that points outward.

4

Map the network zones YOU

- List the segmented zones that need collecting: perimeter, virtualisation, core network, and whatever else your estate is divided into.
- For each, decide the worker runtime — a VM, a pod or a container — and who owns it.
- Confirm each zone can reach the control plane. That single egress rule is what removes the need to open firewalls between segments.

DONE WHEN

A zone map exists with an owner and a runtime per zone.

5

Deploy the workers TOGETHER

- Deploy a ScanOps worker into a DMZ in each zone, starting with the least sensitive.
- Give each read-only credentials scoped to its own zone and nothing else.
- Confirm each connects out to your control plane, and that collection covers what the zone owner expects.

DONE WHEN

Every zone on the map is collecting.

6

Operate it YOU

- Ship logs to your SIEM and add the platform to your existing monitoring.
- Add it to your patch and upgrade cycle. You control when versions land, which is the point — and also the obligation.
- Run a restore drill and a worker-failure drill before you depend on either.

DONE WHEN

The platform is in your runbooks and your on-call rota like any other internal service.

Sequence, not schedule. The phases are ordered by dependency. Elapsed time depends on your change process, your approvals and how much of the estate is in scope, so this plan does not guess at it. Ask us and we will estimate it against your specifics.

Iron Fort Solutions · goironfort.com/reference-architecture