Protect Foundations – Getting Started

PingOne Protect

Field	Value
Version	1.0
Date	2026-04-01
Owner	Partner Delivery Architects
Intended Audience	Technical Consultants/Project Managers
Distribution	Internal/Partner

Related Delivery Kit Assets

- **Protect Foundations – Fundamentals**
- **Protect Foundations – Best Practices**
- **Protect Foundations – PingFederate Integration Guide**
- **Protect Foundations – DaVinci Integration Guide**
- **Protect Foundations – PingAM / AIC Integration Guide**
- **Protect Foundations – Delivery Roadmap Template**

Table of Contents

Protect Foundations - Getting Started

This document takes a customer from no Protect to a “Protect-ready environment” so the delivery team can move directly into the **Protect Foundations - Integration Guides** and **Delivery Playbook**.

If the reader is new to PingOne or Protect configuration, they should use this together with **Protect Foundations - Fundamentals** for detailed console steps and screen-by-screen guidance.

This guide prepares the environment for delivery. Full effectiveness of Protect is achieved through correct integration, data collection, and ongoing tuning.

---

1. Who This Is For

- Primary: partner / customer technical lead preparing an environment for a Protect Foundations project.
- Also relevant: PS lead consultant, customer IAM/platform owner.
- If they have not configured PingOne or Protect before, they should follow **Protect Foundations - Fundamentals** alongside this document.

---

2. What Good Looks Like

A successful Protect Foundations delivery should result in:

- At least one core journey integrated with Protect (e.g. login)
- Signals SDK or device profiling implemented where applicable
- Risk-based branching implemented (Low / Medium / High)
- Risk evaluations visible and validated in PingOne
- Initial tuning completed based on observed behaviour
- Customer able to monitor and operate Protect post-handover

3. Expected Outcomes

By the time this document is complete, the environment should:

- Have PingOne Protect enabled in the correct PingOne environment.
- Have at least one risk policy configured and selected for initial use (default is fine).
- Be producing events on the Threat Protection / Protect dashboard (from a sample app or test flow).
- Have a chosen integration path:
    - PingFederate, DaVinci, PingAM/AIC, or API/SDK.
- Have a worker application and credentials ready for the integration team.

If any of these are unclear, the detailed “how-to” lives in **Protect Foundations - Fundamentals**.

---

4. Prerequisites Checklist

Before treating an environment as “ready to integrate”:

PingOne tenant exists and you can sign in to the admin console.
A target environment (e.g., CIAM-DEV, WF-DEV) is created.
Your user (or a sponsoring admin) has:

- Environment Admin
- Identity Data Admin
  (If unsure how to verify or assign, see “Admin roles” in PingOne Protect Fundamentals.)

In-scope journeys are identified (e.g., CIAM login, registration, password reset, workforce SSO).
A primary integration surface for the Protect Foundations project is chosen:

- PingFederate
- PingOne DaVinci
- PingOne Advanced Identity Cloud / PingAM
- Custom app / API / SDK

5. Integration Steps

Step 1 – Add Protect to the Environment

1. In the PingOne admin console, select the target environment.
2. Go to Overview → Services → Add.
3. Choose PingOne Protect and complete the wizard.
4. Confirm:
    - Protect appears in the environment sidebar.
    - A default risk policy has been created for that environment.

If any of these steps are unfamiliar, see **Protect Foundations – Fundamentals** → Add PingOne Protect to an Environment.

5. Step 2 – Baseline Orientation & Sanity Check

These concepts are explained in detail in **Protect Foundations - Fundamentals** and are included here for quick orientation only.

Concepts to align on:

- **Predictors** – individual risk signals (IP reputation, geovelocity, new device, bot detection, etc.).
- **Risk policies** – combine predictors to produce an overall Low / Medium / High risk decision.
- **Risk evaluations** – per-transaction records with level, score, and predictor details.

Signals/Device Data Collection (Critical):

For DaVinci web flows, use supported collectors:

- **ProtectCollector (skrisk)** via:
    - HTTP Connector
    - PingOne Forms Connector

Ensure the collector is:

- Initialized as early as possible in the user journey
- Allowed sufficient time to collect data before risk evaluation is triggered

If device data is not collected early:

- Risk evaluations will be incomplete
- Risk scoring accuracy will be reduced

Sanity check:

- Use either:
    - The PingOne sample app (from a trial environment), or
    - A simple test journey set up via Fundamentals.
- Perform a few logins or flows.

- Open the **Threat Protection / Protect dashboard** and confirm:
  - Events appear for the right environment.
  - Risk levels and scores are populated.
  - At least some predictor details are visible.

Do **not** tune policies at this stage. Tuning and rollout are handled later via **Protect Foundations - Best Practices**.

If events are visible in the dashboard, the environment is correctly configured and ready to proceed.

Step 3 – Choose the Integration Path

Decide where the first Protect Foundations integration will happen. The selected integration path should align with the overall delivery approach defined in the **Protect Foundations - Delivery Playbook**.

Integration Surface	Typical Scenario	Next Document to Open
PingFederate	Existing PF SSO/auth policies already in place	Protect + PingFederate Integration Guide
DaVinci	New or orchestrated CIAM flows, API-first journeys	Protect + DaVinci Integration Guide
PingAM / AIC	Using AIC/PingAM journeys for access/auth	Protect + PingAM / AIC Integration Guide
Custom app / API / SDK	Direct app integration, heavy mobile/native usage	Protect + API/SDK guidance (if provided)

Only one surface is needed to start; others can be added in later phases.

Step 4 – Prepare Handoff to the Integration Team

Before an integrator opens any **Protect + <Platform>** guide, ensure the following are ready and documented.

Environment & Access

- Environment name and region
- Environment ID
- Confirm admin access for:
    - PingOne Protect console.
    - Integration platform (PF / DaVinci / AIC / app).

Worker Application

A Worker application in PingOne (used for secure server-side API calls to Protect):

- Worker app name
- Client ID
- Client Secret
- Environment ID
- Roles assigned (e.g., Env Admin + Identity Data Admin or equivalent).

If the team doesn't know how to create or permission this, point them **to Protect Foundations - Fundamentals** → Create & Configure Worker Application.

Risk Policy to Use Initially

- Initial risk policy name / ID (often the default)
- Initial enforcement mode:
    - ☐ Learn / non-blocking (recommended)
    - ☐ Partial enforcement
    - ☐ Full enforcement (for limited, controlled flows only)

Handoff Package

Bundle the following for the implementers:

- This **Getting Started** document (completed).
- Environment and worker app details (no secrets in email/plain text where possible).
- Pointer to the chosen **Protect + <Platform> Integration Guide**.
- Any customer-specific constraints (network, logging, SSO patterns).

6. “Integration-Ready” Exit Criteria

An environment is considered “integration-ready” when all the following are true:

- PingOne Protect is enabled in the correct environment.
- At least one risk policy exists and is selected for initial use.
- The Threat Protection / Protect dashboard shows real or sample events.
- A primary integration path (PingFederate / DaVinci / PingAM / API) has been selected and agreed.
- A worker application with the right roles is created and working credentials are available to the integration team.
- The integrator has the required console access and the agreed handoff package.

Once these boxes are ticked, move straight into the relevant **Protect + <Platform> Integration Guide** (to implement), the Protect Foundations Delivery Playbook (to guide the overall delivery flow) and use **Protect Foundations - Best Practices** to shape policy design and tuning.