Protect Foundations - Fundamentals

PingOne Protect

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

Related Delivery Kit Assets

- **Protect Foundations – Getting Started**
- **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**
- **Protect Foundations – Delivery Playbook**

Table of Contents

1. Audience & Scope 3

2. Core Concepts 4

2.1 What PingOne Protect Does 4

2.2 Key Data Objects 4

3. Prerequisites & Environment Basics 5

4. Adding PingOne Protect to an Environment 6

4.1 What “Adding Protect” Actually Does 6

4.2 Steps – Add PingOne Protect 6

5. Risk Policies & Predictors 7

5.1 Default Risk Policy 7

5.2 Viewing & Editing Risk Policies 7

5.3 Creating a New Risk Policy 7

5.4 Predictor Behaviour & Fallbacks 8

6. Worker Applications 8

6.1 Why You Need a Worker Application 8

6.2 Creating a Worker Application 9

6.3 Assigning Roles to a Worker Application 9

6.4 Capturing Worker Credentials 9

7. Threat Protection / Protect Dashboard 10

7.1 What the Dashboard Shows 10

7.2 First Sanity-Check with the Dashboard 10

8. Signals (Protect) SDK 11

8.1 Why the SDK Matters 11

8.2 Where the SDK Is Used 11

8.3 Conceptual Flow 11

8.4 High-Level Implementation Steps 12

9. Putting It Together – First Risk Evaluation 12

9.1 Checklist Before You Start 12

9.2 Using the Sample App (Trial Path) 12

9.3 Using Your Own Flow (Integration Path) 13

Protect Foundations - Fundamentals

This document explains the core building blocks of PingOne Protect and provides detailed configuration guidance for each.

It is designed as a technical reference to support delivery and should be used alongside:

- **Protect Foundations - Delivery Playbook** (for delivery flow)
- **Protect Foundations - Getting Started Guide** (for environment setup)
- **Protect Foundations - Integration Guides** (for platform-specific implementation)

---

1. Audience & Scope

Audience

- Partner / customer technical leads
- Ping / partner consultants and architects
- Administrators responsible for PingOne configuration

You should be able to:

- Add PingOne Protect to an environment.
- Create and permission a worker application.
- Understand and configure risk policies and predictors.
- View and interpret risk evaluations and dashboards.
- Understand where the Signals (Protect) SDK fits in and when it's required.

This guide deliberately stops before platform-specific integration (PingFederate, DaVinci or PingAM/AIC) – those are covered in the **Protect Foundation - Integration Guides**.

This guide does not define delivery order. For end-to-end delivery, follow the **Protect Foundations - Delivery Playbook.**

2. Core Concepts

This section provides a high-level understanding of Protect concepts. Detailed tuning and production guidance is covered in the Best Practices guide.

2.1 What PingOne Protect Does

PingOne Protect is a risk and fraud detection service that:

- Collects network, device, location, and behavioral signals.
- Runs them through predictors and ML models.
- Produces a risk evaluation (score + level + recommended actions).
- Lets your journeys adapt: skip MFA for low risk, step-up for medium, or block for high risk.

You can use Protect in:

- **Authentication** flows (login, step-up)
- **Registration** flows (new account fraud)
- **Account recovery** (ATO recovery hardening)
- **High-risk transactions** (payments, profile changes, approvals)

2.2 Key Data Objects

Predictors
A predictor looks at a single dimension of risk and returns a per-predictor result (for example, HIGH/MEDIUM/LOW). Examples: anonymous network, geovelocity, IP reputation, new device, bot detection, suspicious device, user-based risk behavior, traffic anomaly, email reputation, etc.

Risk Policies
A risk policy combines multiple predictors and produces an overall risk level (LOW / MEDIUM / HIGH) and score. Policies also define thresholds and optional mitigation rules.

Risk Evaluations
A **risk** evaluation is a single decision instance, created when your journey calls Protect (via DaVinci connector, PF IK, AIC nodes, or API). It includes:

- `result.level` – LOW / MEDIUM / HIGH
- `result.score` – numeric score behind the level
- `result.recommendedAction` – hints like BOT_MITIGATION, ACCOUNT_RECOVERY, DENY, TEMP_EMAIL_MITIGATION
- Predictor-level details and any custom attributes you passed in

Threat Protection / Protect Dashboard
The dashboard shows:

- Event volumes and distributions by risk level
- Top predictors contributing to High risk
- Breakdown by IP, country, device, etc.
- Tools to drill into specific evaluations and troubleshoot noise or coverage gaps

Signals (Protect) SDK
The SDK (sometimes called Signals or skrisk) collects rich device and behavioral data from browsers and apps. Some predictors require SDK data to evaluate correctly.

In simple terms: predictors generate signals, policies combine them, and evaluations produce the final decision used by your application.

---

3. Prerequisites & Environment Basics

To work with Protect, you need:

- A PingOne tenant (trial or production).
- At least one environment (for example, *CIAM-DEV*, *CIAM-TEST*, *CIAM-PROD*).
- An admin account with:
    - Environment Admin and
    - Identity Data Admin
      (or a custom role combination that covers Protect administration).

For most partners, we recommend:

- A dedicated non-production environment (DEV/TEST) to start.
- Separate customer environments per project where possible.

4. Adding PingOne Protect to an Environment

This section provides reference steps for adding Protect to an environment. For initial environment setup, follow the Protect Foundations Getting Started Guide.

4.1 What “Adding Protect” Actually Does

When you add Protect to an environment:

- PingOne creates the Protect service for that environment.
- A default risk policy is created automatically.
- Threat Protection / Protect views become available in the console.

This is the foundation for everything else.

4.2 Steps – Add PingOne Protect

1. Sign in to the PingOne admin console.
2. In the top nav, select your target environment (DEV/TEST).
3. Go to Overview.
4. In the Services area, click Add (plus icon).
5. In the list of services, select PingOne Protect.
6. Click Finish / Add.

You are done when:

- You see PingOne Protect listed in the side navigation under Threat Protection.
- Under Risk Policies, there is at least one default risk policy for this environment.

5. Risk Policies & Predictors

5.1 Default Risk Policy

When you add Protect, PingOne creates a default risk policy:

- Uses a score-based approach with all out-of-the-box predictors.
- Assigns different scores to High results depending on predictor type (e.g., IP-only predictors might score 50, while behavioral/device predictors might be 75) to reflect their relative strength.
- Is generally suitable as an initial policy for new use cases.

You can continue to use this policy for early testing and **"learn mode"** before heavy tuning.

5.2 Viewing & Editing Risk Policies

Navigate to Risk Policies

1. In PingOne, ensure you've selected the right environment.
2. Go to Threat Protection → Risk Policies (or similar Protect section).
3. You'll see the default policy and any additional custom policies.

View a Policy

- Click the policy name to open it.
- You should see:
    - The list of predictors in use.
    - Thresholds or scores determining LOW/MEDIUM/HIGH.
    - Any fallback values and overrides.

5.3 Creating a New Risk Policy

You might want a separate policy per journey (e.g., Registration, Authentication, Account Recovery, High-Risk Transaction).

Steps (high-level):

1. In the Risk Policies section, click Add / New Policy.
2. Give the policy a name and description (e.g., "CIAM – Registration Policy").
3. Choose predictors:
    - For Registration, emphasize:
        - Email reputation, new device, bot detection, suspicious device, traffic anomaly.
    - For Login, emphasize:
        - User-based risk behavior, geovelocity, user velocity, IP reputation, anonymous network, new device.
4. Configure scores/thresholds:
    - Start close to the default policy; adjust later with real data.
5. Save the policy.

Assigning a Policy in Integrations

- In DaVinci, you can provide a Risk Policy ID override when calling the Protect connector.
- In APIs, you can pass the policy ID when creating evaluations.
- In PF/AIC, the integration kit / nodes will typically use the default policy unless configured otherwise.

5.4 Predictor Behaviour & Fallbacks

Each predictor:

- Has prerequisites (e.g., IP vs SDK data).
- May need a training window before returning reliable scores.
- Can be configured with fallback values for missing data, but some types (e.g., boolean geovelocity predictors) don't support every fallback (e.g., "Medium" may not be valid).
- Predictors rely on the quality and completeness of input data (device, behavioural, network, and contextual signals). Missing or incomplete data will directly reduce the accuracy of risk evaluations and may impact enforcement decisions.

In general:

- Use default predictor settings to start.
- Only adjust fallback values and scores once you've:
    - Collected at least 1–3 weeks of production traffic (workforce) or 2–4 weeks (CIAM).
    - Identified clear false positives or coverage gaps in the dashboard.

(Policy tuning is covered in more depth in the **Protect Foundations - Best Practices**)

---

6. Worker Applications

6.1 Why You Need a Worker Application

Protect is called by server-side components (DaVinci connector, PF IK, AIC nodes, custom services) using PingOne admin APIs. Rather than using a user's login, PingOne uses a worker application:

- A userless OIDC client.
- Holds the client ID + secret used for OAuth2 client credentials grants.
- Is granted admin roles at the environment level so it can:
    - Call risk evaluation APIs,
    - Access risk policies,
    - Interact with other services as required.

6.2 Creating a Worker Application

1. In the PingOne admin console, go to Applications → Applications.
2. Click Add.
3. In the Add Application panel:
    - Set Application name – e.g., `Protect Worker – CIAM DEV`.
    - Optionally add a description and icon for clarity.
    - For Application Type, choose Worker.
4. Click Save / Create.

At this point, you have a worker app without roles.

6.3 Assigning Roles to a Worker Application

Worker apps have no roles by default. You must assign roles:

1. Open your new worker app.
2. Go to the Roles tab.
3. Click Grant Roles.
4. Assign:
    - Environment Admin
    - Identity Data Admin
      (or a more restricted custom role set if you've defined one that still permits Protect operations).
5. Save your changes.
6. Ensure the application is enabled (toggle on) on the Overview tab.

Security note: For stricter environments, you can create a dedicated role that only includes the permissions needed for Protect (and related services) and assign that instead of full Env Admin, as long as integration docs' requirements are met.

6.4 Capturing Worker Credentials

From the worker application's Overview tab, record:

- **Client ID**
- **Client Secret**
- **Environment ID** (for the environment containing this application)

You'll provide these to:

- DaVinci's PingOne Protect connector configuration.
- PingAM/AIC PingOne Worker Service configuration.
- PingFederate integration kit or custom API clients as needed.

7. Threat Protection / Protect Dashboard

7.1 What the Dashboard Shows

The Threat Protection / Protect dashboard provides:

- High-level metrics: total evaluations, distribution across LOW/MEDIUM/HIGH.
- Charts to drill into:
    - Risk events by country, IP, device, OS/browser.
    - Predictor contributions to High risk.
    - Specific evaluations and Resource IDs for deep debugging.

It's the primary tool for:

- Validating that Protect is receiving data.
- Understanding where false positives come from.
- Finding under-protected flows.

7.2 First Sanity-Check with the Dashboard

After you've:

- Added Protect to an environment, and
- Run a few test journeys or used the sample app

Complete the following:

1. In PingOne, open your target environment.
2. Navigate to the Threat Protection / Protect dashboard.
3. Filter to the correct time window (last 15 minutes / last hour).
4. Confirm you can see:
    - At least some risk events.
    - Their risk levels and scores.
    - At least a couple of predictors contributing to the decision.

If the dashboard is empty despite test traffic, don't jump to tuning – first verify:

- Integration calls are actually creating evaluations (DaVinci connector, PF IK, AIC nodes).
- The worker app credentials and environment IDs are correct.

8. Signals (Protect) SDK

The SDK is optional but strongly recommended for production use, as it enables more accurate device and behavioural risk detection.

8.1 Why the SDK Matters

Some predictors (notably new device, behavioral models, suspicious device, some bot-related classifications) depend on client-side signals that the SDK collects:

- Device characteristics
- Browser attributes
- User interaction / behavior
- Persistent identifiers

You can run Protect without SDK payloads, but:

- Certain predictors will return “not evaluated”.
- Overall quality of risk decisions may be lower.

8.2 Where the SDK Is Used

The Signals (Protect) SDK can be:

- Embedded in web flows via:
    - DaVinci HTTP connector skrisk component / ProtectCollector.
    - PF integration kit scripts/templates.
- Embedded in mobile apps via:
    - Native Android/iOS SDKs (Protect/Signals libraries).

DaVinci and AIC integration docs provide exact configuration steps; here we focus on concepts.

Important - The Signals SDK is critical to the effectiveness of PingOne Protect. Without it, risk evaluations will be limited and certain predictors will not function correctly.

8.3 Conceptual Flow

1. Initialize SDK early in the journey (or app startup) so it has time to collect data.
2. SDK collects:
    - Device, browser, network, and behavioral signals.
3. When your journey initiates a risk evaluation, the SDK payload is:
    - Sent via DaVinci connector / PF IK / AIC nodes / API as part of the evaluation input.
4. Protect uses this to evaluate device and behavioral predictors and compute a more accurate score.

8.4 High-Level Implementation Steps

The precise steps depend on your integration surface, but generally:

- **Web (DaVinci):**
    - Install the required connector versions (ProtectCollector, forms connector).
    - Configure the skrisk / ProtectCollector component with Environment ID and optional behavior toggles.
    - Ensure the Protect connector's **"**Risk input from device"/similar configuration references the variable holding SDK data.

- **Mobile (SDKs):**
    - Add the Protect / Signals SDK dependency for Android or iOS.
    - Initialize the SDK in the app's lifecycle (e.g., app start / login screen).
    - Send device payload and context to your Protect integration (DaVinci connector or custom API path).

The detailed DaVinci / PF / AIC specifics are covered in respective Protect Foundations - Integration Guides.

---

9. Putting It Together – First Risk Evaluation

This section illustrates how Protect components work together in practice. For structured delivery, follow the Protect Foundations Delivery Playbook.

9.1 Checklist Before You Start

You should now have:

- Environment with PingOne Protect added.
- At least one risk policy (default is fine).
- A worker application with the right roles and its:
    - Client ID
    - Client Secret
    - Environment ID
- Access to the Threat Protection / Protect dashboard.

9.2 Using the Sample App (Trial Path)

If you're using a PingOne trial with a solution designer:

1. When you create a Customer solution environment, PingOne can auto-provision a sample application.
2. This sample app includes:

    - Pre-wired authentication flows,
    - The ability to simulate risk events and observe predictors.
3. Use the sample app's Authentication flow to:
    - Perform a few logins from different browsers/locations if possible.
    - Trigger risk evaluations that hit your default policy.
4. Then open the Protect dashboard and confirm events are visible as described earlier.

9.3 Using Your Own Flow (Integration Path)

If you're integrating via DaVinci / PF / AIC / API:

- Follow the respective **Protect Foundations - Integration Guides** to:
    - Call Create Risk Evaluation at the decision point in the flow.
    - Call Update Risk Evaluation near the end of the flow to send SUCCESS / FAILED status back to Protect.

Once integration is wired:

1. Run a few test attempts (successful logins, maybe some failed ones).
2. Open the Protect dashboard.
3. Confirm:
    - Evaluations exist for your environment.
    - You can see the risk level, score, and predictor breakdown.

If all of the above is complete, you're at the "Protect-ready environment" stage defined in the **Protect Foundations - Getting Started**