Codreum AWSPro User Guide

AWS-Native Network Observability

AI Insights · DNS · DNS Firewall · VPC Flow · Platform · Hosted Zones

Deployment, Prerequisites & Operations

Version 1.0 | Updated June 2026 | Public document

www.codreum.com

Quick Start

From zero to your first dashboard in six steps. Full detail follows in the chapters.

---

1. Configure AWS Credential. Set credentials for the target account in the terminal you will deploy from — the quickest way is to copy the environment-variable block from your AWS access portal (IAM Identity Center) and paste it before you run apply. → *Chapter 6 (6.2).*

2. Prepare your telemetry. AWSPro reads telemetry you already collect — it does not create it. Create a CloudWatch log group and point your DNS resolver query logs (and, as needed, DNS Firewall, VPC Flow, or hosted-zone query logs) at it. → *Chapter 3.*

3. Build your configuration. Open the Config Builder, choose Basic Lite (no license) or Premium (for Premium, enable the License Feature(Network) so VpcD/ZoneD can deploy), map your VPCs/zones to their log groups, pick dashboards and alarms, and download the deployment bundle. → *Chapter 4.*

www.codreum.com/awspro-builder/

4. Deploy with Terraform. Unzip the bundle (it contains main.tf, variables.tf, and your terraform.tfvars) into an empty folder and run: → *Chapter 6.*

```
terraform init
terraform plan      # review what will be created
terraform apply     # deploys in ~8–15 min — let it finish, do not Ctrl+C
```

5. Confirm alerts can reach you. Add your email on the Notifications step, then click the AWS subscription-confirmation email. Until you confirm, you will see a harmless "pending confirmation" notice. → *Chapter 8.*

6. Verify and tune. Open your CloudWatch dashboards (named with your prefix) and confirm widgets populate after a few minutes of traffic. An empty dashboard almost always means a missing prerequisite from step 1. Tune alarm thresholds in your first week. → *Chapters 8 & 11.*

> **Basic Lite vs. Premium**
>
> Evaluating? Start with Basic Lite — no license required — for posture dashboards and starter alarms from just your log groups. Upgrade to Premium (full dashboards, alarm tuning, AI Insights, per-subject routing) by switching the tier in your configuration. See Chapter 7 for licensing.

> **The one rule to remember**
>
> AWSPro never creates your telemetry — you put it in place, AWSPro reads it and builds the monitoring on top. If a dashboard is empty, the telemetry is missing or misrouted (Chapter 3), not the product failing.

Contents

1. Introduction

Codreum AWSPro turns the network telemetry that already exists in your AWS account — DNS resolver logs, DNS Firewall logs, VPC Flow Logs, platform metrics, and Route 53 hosted-zone query logs — into a set of ready-made CloudWatch dashboards, alarms, and optional AI-assisted insights. It deploys entirely inside your own AWS account using Terraform. Nothing about your network leaves your account, and Codreum never sees your data.

> **What AWSPro is**
>
> A Terraform provider you run in your own account. It reads telemetry you already collect and builds operational dashboards and alarms on top of it.
>
> It does not capture traffic, install agents, or send data anywhere. It reads from CloudWatch Logs and CloudWatch Metrics that you point it at.

1.1 What it deploys

AWSPro is organized into two monitoring modules and two licensing tiers.

VpcD — VPC-level DNS, DNS Firewall, VPC Flow, and platform (endpoint/NAT) observability, plus a global cross-VPC summary.

ZoneD — Route 53 hosted-zone query visibility, per zone and across zones.

Tier	What you get
Basic Lite	Global Lite posture dashboards (VpcD and/or ZoneD) plus starter alarms, driven from your DNS / query log groups. No license required.
AWSPro Premium	The full set of dashboards (DNS, DNS Firewall, VPC Flow, Platform, Global, Hosted Zones), per-target alarm tuning, AI Insights, and per-VPC / per-zone notification routing. Requires a license ID and the network entitlement (license_features = Network)

1.2 How you will use this guide

Read the guide in order the first time. The sequence matters: AWSPro consumes telemetry you must put in place first, so the Prerequisites chapter comes before deployment. A typical first deployment looks like this:

1. Confirm the prerequisites for the capabilities you want (Chapter 3). This is where most first-time effort goes — AWSPro does not create log groups or flow logs for you.
2. Build your configuration with the Config Builder (Chapter 4), or copy a template.
3. Apply with Terraform (Chapters 5–6).
4. Verify telemetry is flowing and alarms notify you (Chapter 8).
5. Tune thresholds and operate day to day (Chapter 11).

2. How AWSPro Works

Understanding the data flow makes the prerequisites obvious, so this short chapter comes first.

2.1 The data path

AWSPro reads from two AWS services you already use: **CloudWatch Logs** (your DNS, DNS Firewall, VPC Flow, and zone query logs) and **CloudWatch Metrics** (platform metrics such as VPC endpoint and NAT gateway statistics). From those, it builds CloudWatch dashboards and alarms, and — for Premium — an in-account AI insights layer using Amazon Bedrock.

> **The one rule that explains every prerequisite**
>
> AWSPro never creates your telemetry. It does not create log groups, enable flow logs, turn on query logging, or enable platform metrics. You put the telemetry in place; AWSPro reads it and builds the monitoring on top.
>
> This is by design: it keeps AWSPro least-privilege and means your raw data stays under your control.

2.2 Drift enforcement

AWSPro is declarative. The configuration you apply is the desired state. If someone later edits or deletes an AWSPro-managed dashboard or alarm by hand in the console, the next Terraform apply restores it to match your configuration. You will sometimes see Terraform report that it acted even when your configuration did not change — that is drift being corrected, and it is expected. To intentionally remove something, change the configuration (for example, disable a dashboard) rather than deleting it in the console.

3. Prerequisites — Preparing Your Telemetry

This is the most important chapter. Each capability in AWSPro reads a specific kind of telemetry, and that telemetry must already be flowing to CloudWatch before AWSPro can chart or alarm on it. For every capability below you will find the same four-part pattern:

- **What you must have** — the AWS resource or setting required.
- **How to set it up** — exact console steps.
- **What failure looks like** — so you can recognize a misconfiguration.
- **How to verify** — how to confirm it is working before blaming the product.

> **A CloudWatch log group is the common prerequisite**
> Almost every capability needs your telemetry delivered to a CloudWatch Logs log group that you create. AWSPro reads from these; it does not create them. Section 3.1 shows how to create one — you will reuse those steps throughout this chapter.

3.1 Creating a CloudWatch log group

You will create one or more log groups to receive your DNS, DNS Firewall, VPC Flow, or zone query logs. Do this once per stream of telemetry you want AWSPro to read.

1. Open the AWS Console and go to CloudWatch → Logs → Log groups.
2. Choose Create log group.
3. Give it a clear name (for example, /codreum/dns/prod-vpc or /codreum/vpcflow/prod-vpc). You will reference this exact name or its ARN in your AWSPro configuration.
4. Optionally set a retention period to control cost (for example, 30 days). AWSPro does not require a specific retention.
5. Choose Create.

> **Tip**
> Use a naming convention that encodes the source and scope, such as /codreum/<telemetry>/<vpc-or-zone>. It makes the mapping in your configuration self-explanatory and keeps the console tidy.

3.2 VPC DNS — Resolver query logging

Powers the VPC DNS dashboard and the DNS alarms, and is the minimum telemetry for Basic Lite (VpcD). It captures the DNS queries made from within your VPC by the Route 53 Resolver.

What you must have

- A CloudWatch log group to receive resolver query logs (Section 3.1).
- A Route 53 Resolver query logging configuration that sends your VPC's DNS queries to that log group.

How to set it up

1. Open Route 53 → Resolver → Query logging.
2. Choose Configure query logging.
3. Give the configuration a name.
4. For the destination, choose CloudWatch Logs and select the log group you created in Section 3.1.
5. Under VPCs to log queries for, add the VPC(s) you want AWSPro to monitor.
6. Choose Configure query logging.

How to verify

- Generate some DNS traffic from an instance in the VPC (for example, resolve a few public domains).
- In CloudWatch Logs, open the log group and confirm new log streams and events appear within a few minutes.

> **What failure looks like**
> If the log group is empty after DNS traffic, query logging is not attached to that VPC, or it is pointed at a different log group. AWSPro will show an empty DNS dashboard with no error — an empty chart almost always means missing or misrouted telemetry, not a product fault.

3.3 DNS Firewall logging

Powers the DNS Firewall dashboard and its threat-protection and policy widgets (blocked queries, rule-group hits, DGA and DNS-tunneling protections).

What you must have

- Route 53 Resolver DNS Firewall rule groups associated with your VPC.
- A CloudWatch log group receiving the resolver query logs that include firewall actions (the same resolver query logging path as Section 3.2 carries firewall action fields).
- In your AWSPro configuration, the firewall scope for the VPC — the DNS Firewall rule group association. The Config Builder calls this the **DNS Firewall rule group**, set on the Inventory step.

How to set it up

1. Ensure DNS Firewall is in use: Route 53 → DNS Firewall → Rule groups, and confirm a rule group is associated with your VPC.
2. Ensure resolver query logging is enabled for the VPC (Section 3.2) so firewall actions are logged.
3. In AWSPro, provide the DNS Firewall rule group for that VPC (Inventory step / dns_firewall_scope_map).

How to verify

- With the DNS Firewall dashboard enabled, confirm blocked-query and rule-group widgets populate after some traffic that matches a firewall rule.

> **Requirement reminder**
> The DNS Firewall dashboard and the DNS Firewall alarm both require, for every monitored VPC, a DNS log group AND a DNS Firewall rule group. The Config Builder enforces this and will flag a VPC that is missing either one before you export.

3.4 VPC Flow Logs

Powers the VPC Flow dashboard, the VPC Flow Native dashboard, the flow alarms, and the flow-record widgets in the global summary. This section needs the most care, because the flow-log format determines which consumption path you can use, and the most common mistake here fails silently.

Two ways AWSPro consumes flow logs

Path	What it needs
VPC Flow (normalizer-backed)	Accepts ANY VPC Flow Log format. A normalizer Lambda that AWSPro deploys reads each log's declared format and republishes it normalized. Choose this when you cannot recreate your flow logs in the strict format. Slightly higher cost and latency because it runs a Lambda.
VPC Flow Native	Reads your flow logs directly, with NO Lambda — lower cost and latency. Requires the STRICT 18-field custom format below. Choose this when you control your flow-log format.

The required custom format (for Native, and for Basic Lite flow)

If you use VPC Flow Native — or the Basic Lite flow charts and REJECT alarm — your flow logs must be created with this exact custom format, in this exact field order:

```
${vpc-id} ${subnet-id} ${interface-id} ${instance-id} ${srcaddr}
${dstaddr} ${pkt-srcaddr} ${pkt-dstaddr} ${srcport} ${dstport}
${protocol} ${packets} ${bytes} ${action} ${flow-direction}
${traffic-path} ${tcp-flags} ${log-status}
```

> **The AWS default format does NOT work for Native**
>
> The AWS default flow-log format starts with version and account-id and is missing vpc-id, flow-direction, and traffic-path. If you point Native (or Basic Lite flow) at default-format logs, the metric filters never match: the charts stay empty and AWS reports no error. The same silent failure happens if you accidentally point flow configuration at a DNS log group.
>
> If you cannot recreate your flow logs in the custom format, use the normalizer-backed VPC Flow path instead — it accepts any format.

How to set it up — create flow logs to CloudWatch (custom format)

1. Create a CloudWatch log group for flow logs (Section 3.1).
2. In the VPC console, select the VPC (or subnet/ENI), open the Flow logs tab, and choose Create flow log.
3. Set Destination to Send to CloudWatch Logs and select your flow-log log group.
4. Set Log record format to Custom format, and select the fields in the exact order shown above (or paste the format string).
5. Choose an IAM role that allows publishing to CloudWatch Logs (the console can create one).
6. Create the flow log. In AWSPro, map the VPC to this log group (Inventory step / flow_log_group_map).

Additional setup for the normalizer-backed VPC Flow path

If you choose the normalizer-backed VPC Flow dashboard (any format), AWSPro deploys a normalizer Lambda named **`<prefix>-VpcD-VpcFlow-Normalizer`**. Your flow-log delivery to CloudWatch must be subscribed to that Lambda so it can republish normalized records. After your first apply creates the Lambda, add a CloudWatch Logs subscription filter on your flow-log group that targets the normalizer Lambda's ARN:

1. Apply AWSPro once with the VPC Flow (normalizer) dashboard enabled so the normalizer Lambda is created.
2. In CloudWatch → Logs → Log groups, open your flow-log group, go to Subscription filters, and choose Create → Lambda subscription filter.
3. Select the <prefix>-VpcD-VpcFlow-Normalizer function as the destination.
4. Set the log format / filter pattern appropriate to your flow logs and create the filter. Normalized records are written to /aws/codreum/<prefix>/vpcflow-normalized, which the VPC Flow dashboard reads.

How to verify flow logs

- Confirm metric filters are present on the flow-log group (for the strict/native and Basic paths).
- After ~10 minutes of traffic, confirm VpcFlowRecords datapoints appear (Native publishes under the <prefix>/vpcd/vpcflow-native namespace; Basic Lite under <prefix>/vpcd-lite).
- If charts are still empty after 10 minutes of known traffic, the format does not match — switch to the custom format above, or use the normalizer path.

3.5 Platform metrics (VPC endpoints & NAT gateways)

Powers the VPC Platform dashboard and the platform alarms (endpoint connections and drops, NAT port allocation, NAT delivery ratios, Network Address Usage).

What you must have

- VPC endpoints and/or NAT gateways in the monitored VPC(s) — these publish CloudWatch metrics automatically under AWS-native namespaces; there is no log group to create for platform metrics.
- In AWSPro, the endpoint and/or NAT inventory for the VPC (Inventory step). At least one endpoint is required across all monitored VPCs to enable endpoint widgets/alarms, and at least one NAT gateway across all VPCs to enable NAT widgets/alarms.

How to verify

- Confirm the endpoint/NAT resources exist and are in service.
- With the Platform dashboard enabled, confirm the endpoint and NAT widgets populate (CloudWatch publishes these metrics by default for active resources).

> **Note**
> Platform metrics are emitted by AWS for active VPC endpoints and NAT gateways without any extra logging configuration. The only AWSPro-side prerequisite is listing the endpoint IDs and NAT gateway IDs in your inventory so the dashboards and alarms know what to chart.

Enabling Network Address Usage (NAU) metrics

The Platform dashboard charts Network Address Usage (NAU) — how much of your VPC's IP/ENI capacity is consumed against AWS limits. NAU metrics are published to CloudWatch at the VPC level, but you must turn the metric on for the VPC.

1. Open the VPC console and select the VPC you are monitoring.
2. On the VPC's details, locate the Network Address Usage metrics setting (also available via the CLI as the enableNetworkAddressUsageMetrics VPC attribute).
3. Enable Network Address Usage metrics for the VPC.
4. AWS then publishes NetworkAddressUsage (and peered NAU) metrics for the VPC, which the Platform dashboard and the NAU alarms read automatically.

> **What failure looks like**
> If NAU widgets on the Platform dashboard stay empty while endpoint and NAT widgets populate, NAU metrics are not enabled on the VPC. Enable the VPC attribute above; there is no log group involved.

3.6 Route 53 hosted-zone query logging (ZoneD)

Powers the ZoneD dashboards (per-zone and global), and is the minimum telemetry for Basic Lite (ZoneD). It captures public DNS queries answered by your Route 53 hosted zones.

What you must have

- A CloudWatch log group to receive zone query logs (Section 3.1).
- Query logging enabled on each hosted zone you want to monitor, pointed at that log group.

> **Region requirement**
> Route 53 hosted-zone query logging delivers to CloudWatch Logs in us-east-1. ZoneD therefore requires the AWSPro deployment region to be us-east-1. The Config Builder shows this requirement on the Deployment and Review steps.

How to set it up

1. Create a log group in us-east-1 (Section 3.1).
2. Open Route 53 → Hosted zones, select the zone, and on the Configure query logging action, choose the CloudWatch log group.
3. Repeat for each zone you want to monitor. In AWSPro, map each zone to its query-log group (zone scope / query log group).

How to verify

- Resolve a record in the zone from the public internet and confirm query-log events appear in the log group.

4. Building Your Configuration with the Config Builder

The Config Builder is a free, browser-based tool that generates a valid AWSPro configuration file (terraform.tfvars) through a guided wizard. It is the fastest and safest way to produce a configuration, because it validates your inputs against the same rules the provider enforces. Open it at www.codreum.com/awspro-builder/.

> **Your data never leaves your browser**
>
> The Config Builder runs entirely in your browser. There is no server, no database, and no API. When you import an existing configuration, it is parsed locally on your machine and never uploaded. Saved drafts live in your browser's local storage on your own computer. Codreum never receives or stores anything you enter, including VPC IDs, account numbers, or license keys.

4.1 The wizard at a glance

The builder adapts to your tier. AWSPro Premium shows the full nine-step wizard; Basic Lite shows a shorter five-step wizard with only the steps that apply to Basic. You can switch tiers at any time without losing your work.

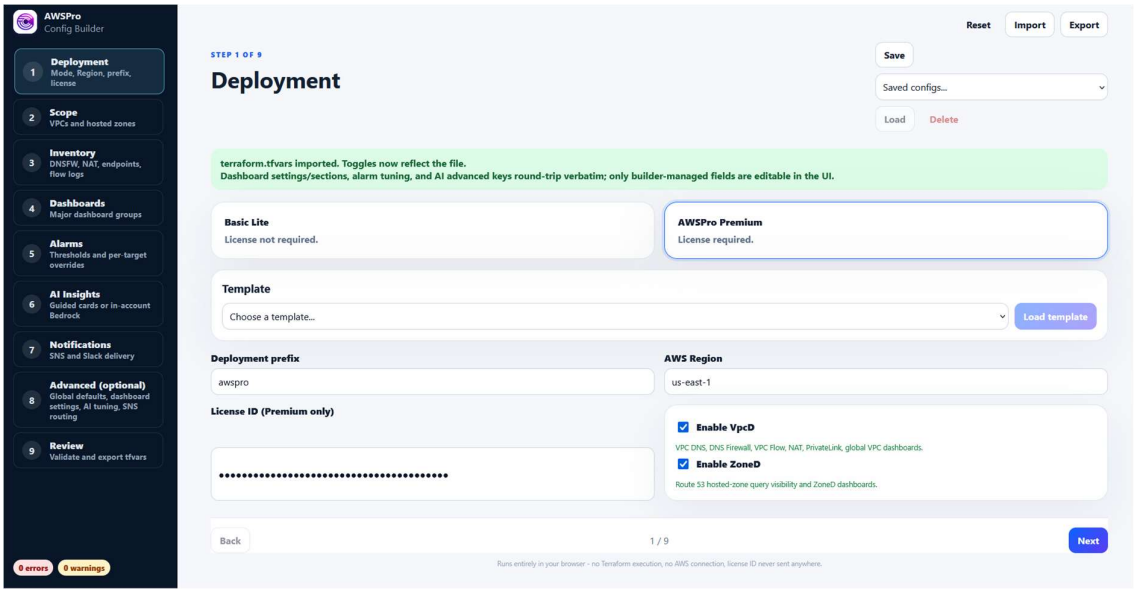

Figure 4.1 — Deployment step (Premium). Choose your tier, region, prefix, and which modules to enable.

4.2 The Premium wizard, step by step

The Premium wizard walks through nine steps. Each step validates as you go and shows any issues inline. The following tour shows the flow; your screens will reflect your own environment.

Step 1 — Deployment

Choose your tier (Basic Lite or AWSPro Premium), AWS region, and a deployment prefix, enter your license ID for Premium, and enable the modules you want (VpcD, ZoneD). The page reacts as you choose — for example, it reminds you that ZoneD requires us-east-1.

For Premium, also turn on the License Feature — it writes license_features = ["network"]. VpcD and ZoneD stay greyed out until it is enabled, because both modules require that entitlement.

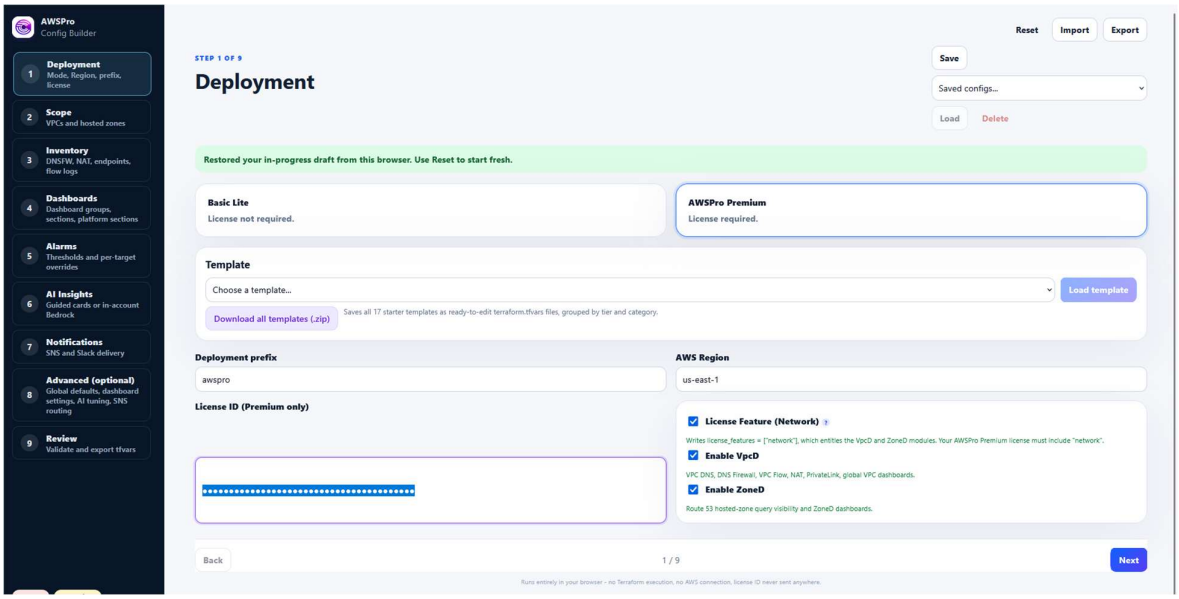

Figure 4.3 — Step 1, Deployment.

Step 2 — Scope

List the VPCs and hosted zones to monitor, each with its DNS or query log group. This is where the telemetry you prepared in Chapter 3 is mapped to subjects AWSPro will chart.

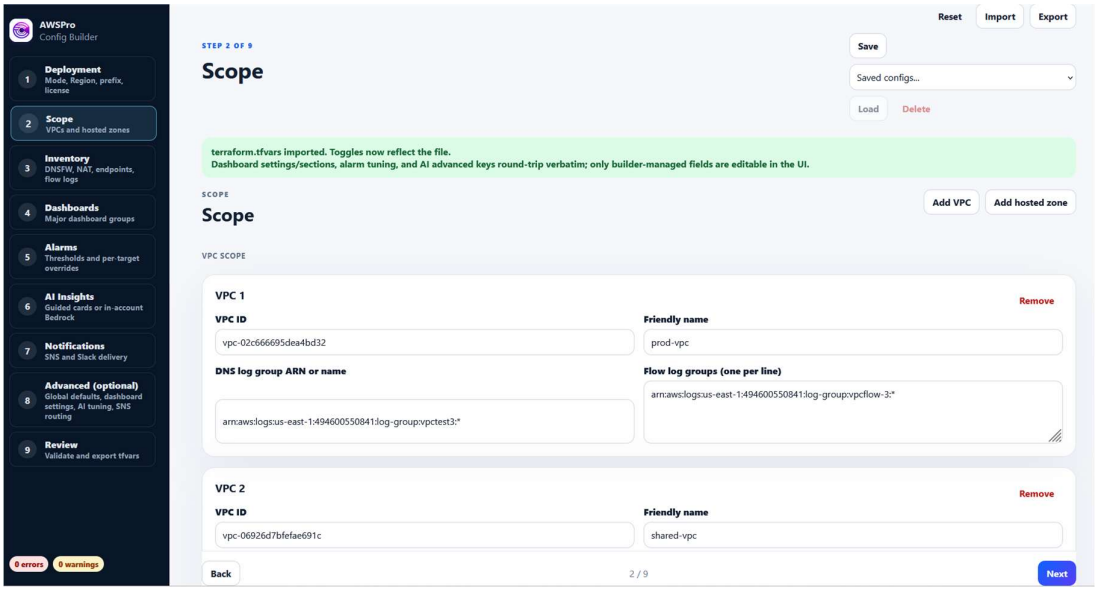

Figure 4.4 — Step 2, Scope.

Step 3 — Inventory

Provide the per-VPC telemetry that premium dashboards and alarms need: flow-log groups, DNS Firewall rule groups, NAT gateways, and VPC endpoints. The builder shows a "?" next to fields that benefit from explanation — for example, the difference between the normalizer-backed VPC Flow and VPC Flow Native paths.

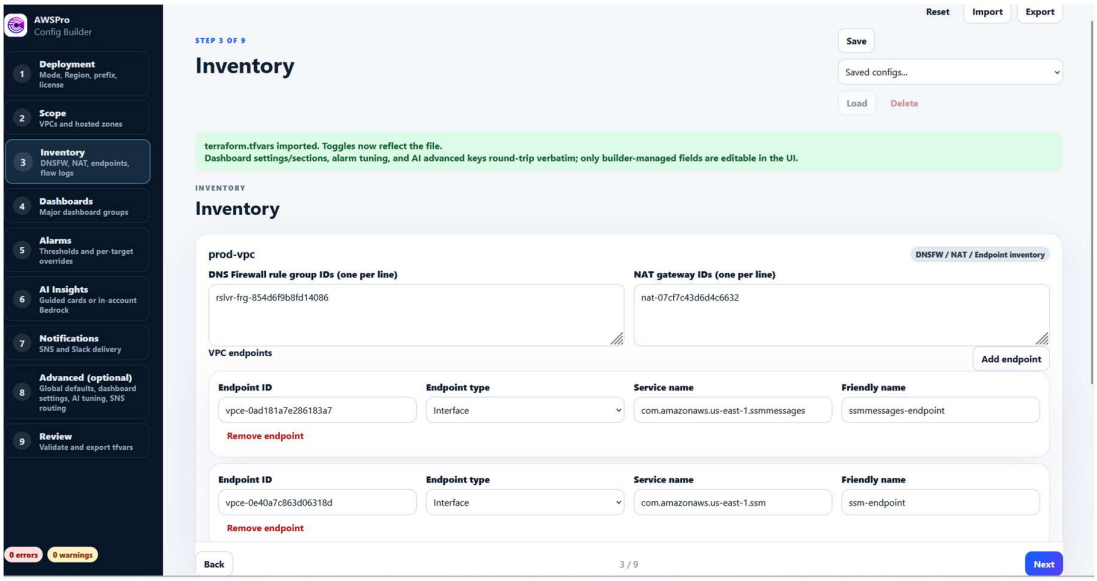

Figure 4.5 — Step 3, Inventory.

Step 4 — Dashboards

Choose which dashboards to deploy. Dashboards whose module is disabled are greyed out, so you only enable what your scope supports.

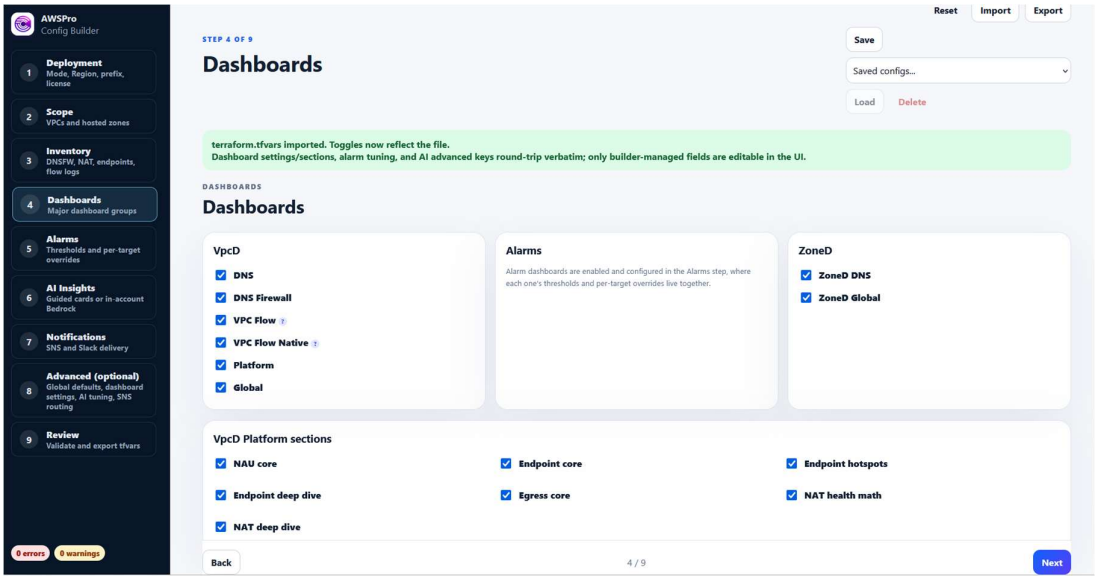

Figure 4.6 — Step 4, Dashboards.

Step 5 — Alarms

Enable alarm dashboards and tune thresholds per target. The provider's default thresholds appear as placeholders; you only fill in a value where you want to override the default, and the builder marks which targets you have modified.

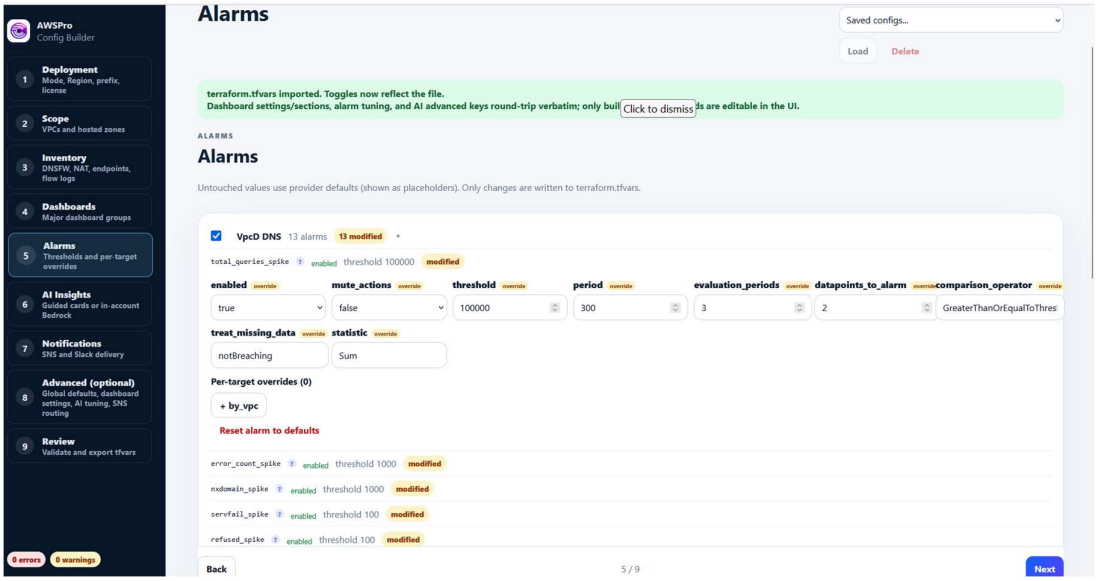

Figure 4.7 — Step 5, Alarms.

Step 6 — AI Insights

Turn on guided triage cards and, optionally, in-account Amazon Bedrock analysis. The default model is shown; you can change it or the region. (See Chapter 14 for how this stays in your account.)

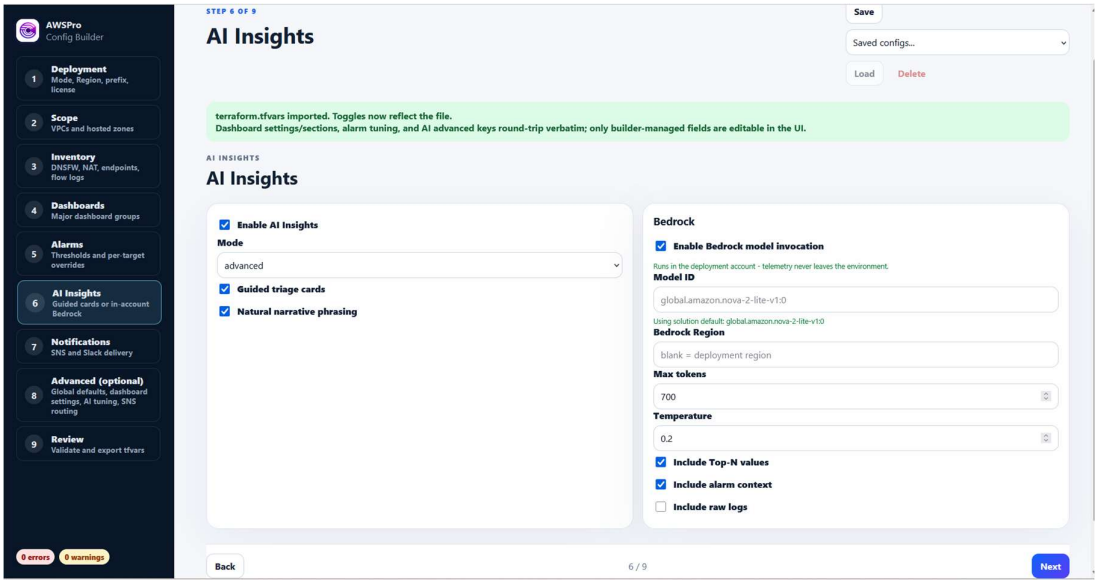

Figure 4.8 — Step 6, AI Insights.

Step 7 — Notifications

Add email, SMS, and HTTPS subscribers for the default alert topic, and enable Slack (Premium). Adding subscribers here is all that is needed to receive alerts.

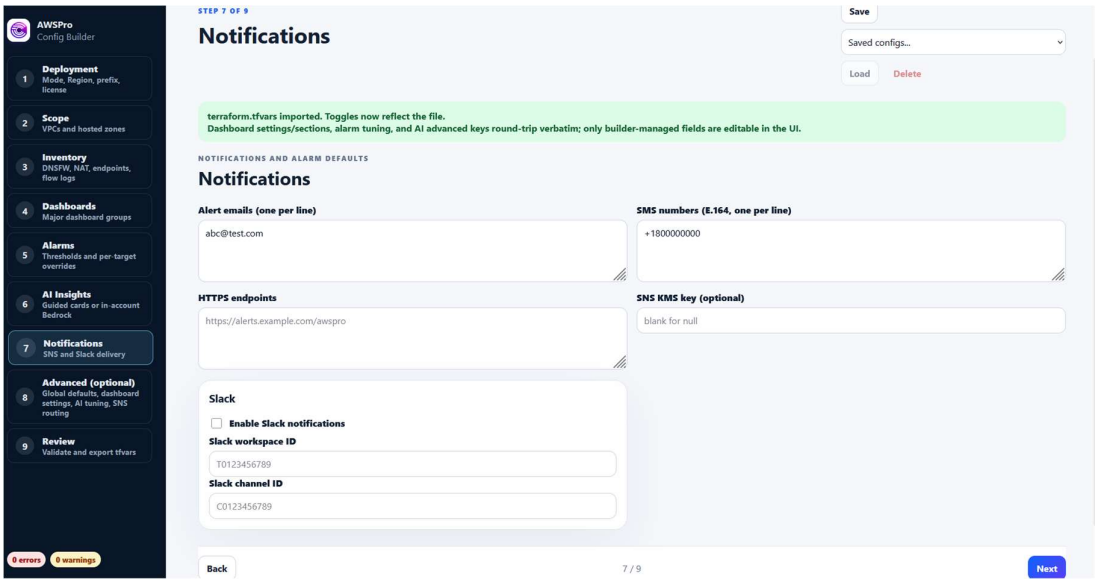

Figure 4.9 — Step 7, Notifications.

Step 8 — Advanced (optional)

Fine-tune dashboard settings, AI tuning, and per-alarm-dashboard SNS routing. Every field has a "?" explaining what it does. Blocks whose dashboard is disabled are greyed out, and provider defaults are shown as placeholders so you only set what you want to change.

The Advanced step also includes Scheduled reports (just below Global defaults): enable it to have AWSPro email a weekly or monthly digest of the period — alarm state, dashboards, and headline metrics — to your alert topic. See Chapter 11.

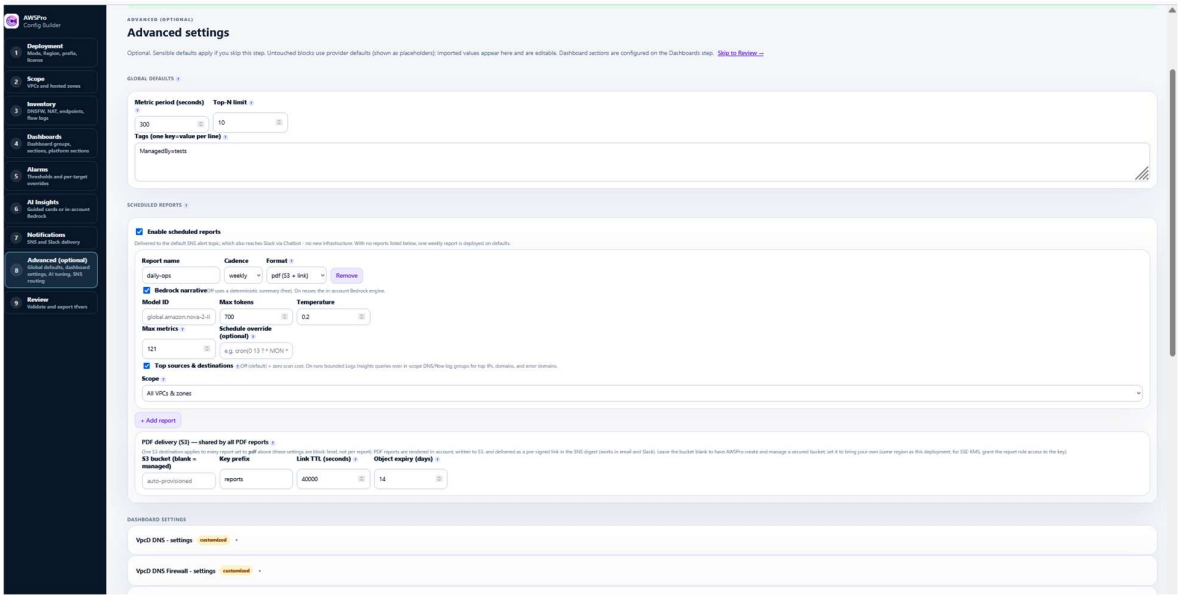

Figure 4.10 — Step 8, Advanced.

Step 9 — Review & export

Review opens with a plain-language summary of exactly what will deploy — mode, region, modules, scope, dashboards, alarms, AI, and notifications — followed by validation results and the generated tfvars. When it is clean, export your configuration.

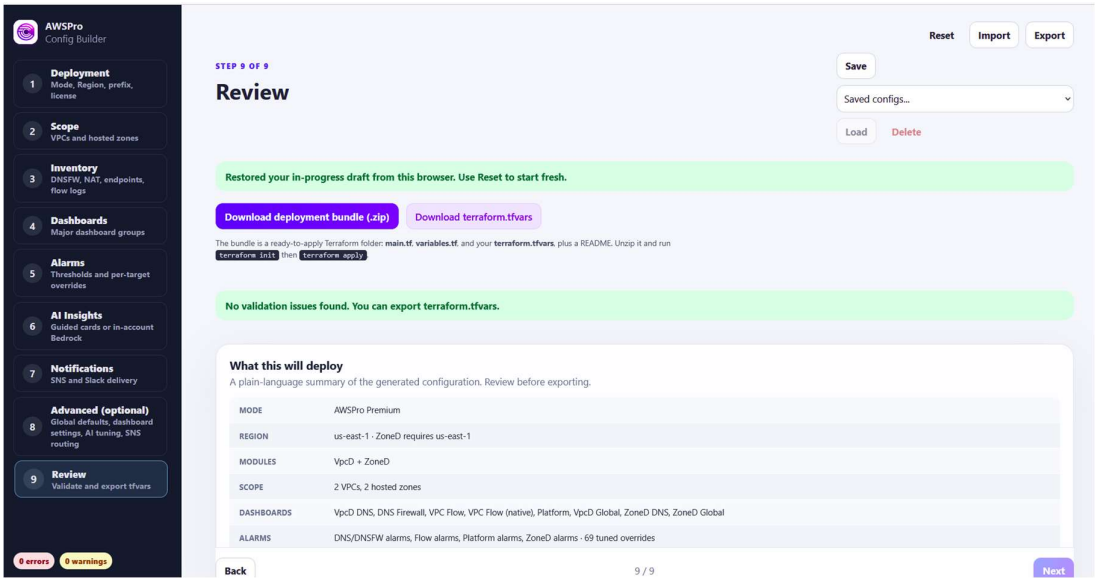

Figure 4.11 — Step 9, Review.

4.3 The Basic Lite wizard

Basic Lite shows a focused five-step wizard — Deployment, Scope, Alarms, Notifications, Review — because the other steps do not apply to Basic. You can switch between tiers at any time; your work is preserved, and the builder only ever exports what the selected tier actually deploys.

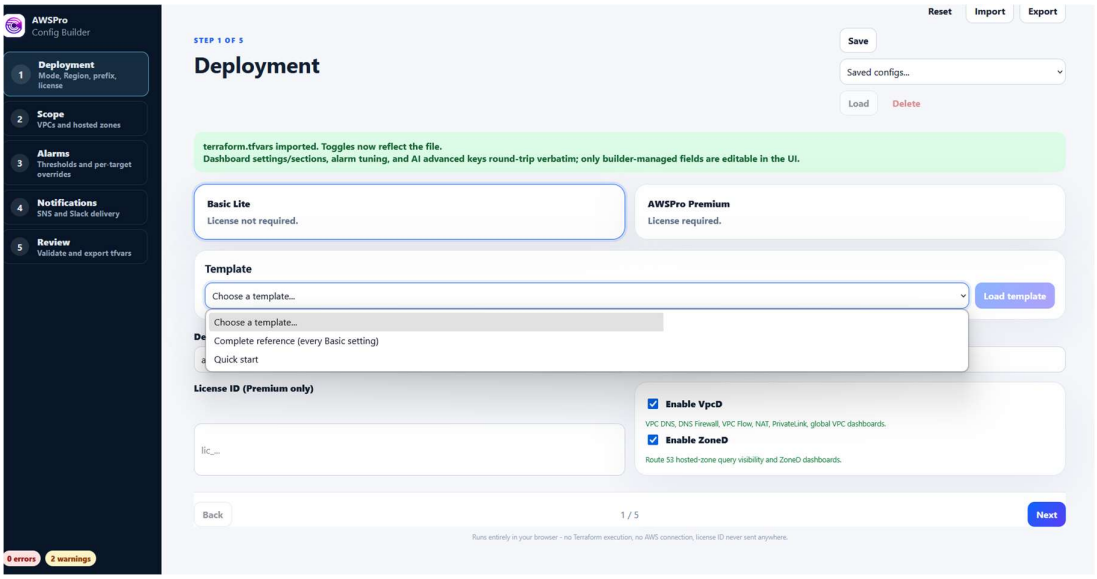

Figure 4.12 — Basic Lite Deployment step (five-step wizard).

4.4 Start from a template

Rather than filling every field from scratch, you can load a template on the Deployment step and adjust it. Templates are complete, valid starting points that differ in which capabilities or rollout patterns they enable. Copy one, replace the example IDs/ARNs/emails with your environment's values, and apply.

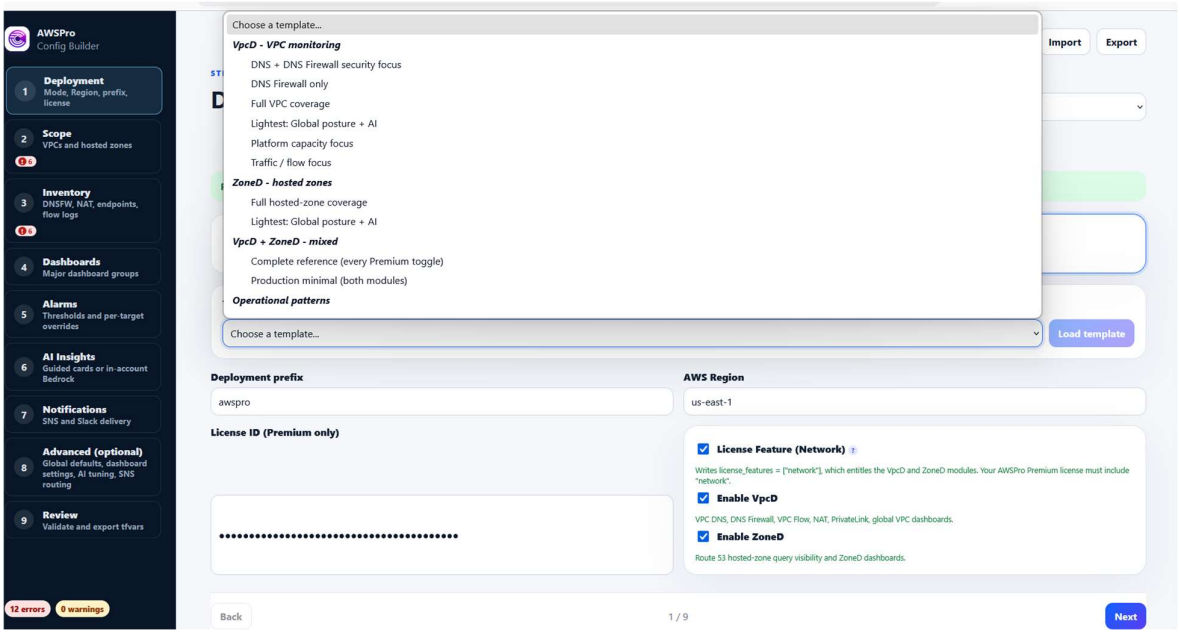

Figure 4.13 — Loading a template gives you a working configuration to adjust.

How to choose, in one decision: Do you have a Premium license? No → start from `basic-lite`. Yes → start from `vpcd-only-premium` (or `zoned-only-premium` for hosted-zone monitoring) and narrow or widen from there. The full-toggle templates are references for tuning, not starting points.

Basic Lite templates

Template	What it deploys / when to use it
`basic-lite`	Lite posture dashboards (VpcD and/or ZoneD Global Lite) + starter alarms + email alerts. The no-license starting point — "is my DNS healthy?" on a dashboard within one apply.
`basic-full-toggle`	Every Basic-honored setting written out at provider defaults: flow telemetry, alarm tuning, the off-switch, all channels. A reference for what Basic supports.

Premium starting points

Template	What it deploys / when to use it
`vpcd-only-premium`	Complete VpcD family (DNS, Global, alarms), no ZoneD. The most common Premium starting point.
`zoned-only-premium`	Complete ZoneD family (Global, per-zone DNS, alarms) + AI + SNS. us-east-1 only.
`vpcd-zoned-production-minimal`	A production-lean cut of VpcD + ZoneD in one stack — a sensible "both modules, no excess" baseline.

Focused Premium templates

Each enables a specific slice of the product — useful when you want exactly one capability area.

Template	Focus
`vpcd-dns-dnsfw-alarms`	DNS investigation + DNS Firewall views + their alarm families + AI + SNS.
`vpcd-dns-firewall-only`	DNS Firewall enforcement views and alarms.
`vpcd-flow-only`	VPC Flow dashboards + flow alarms + AI + SNS.
`vpcd-platform-only`	NAU / endpoint / NAT platform dashboards + platform alarms + AI + SNS.
`vpcd-global-ai`	One executive VpcD Global posture dashboard + AI; no alarms or investigation dashboards.
`zoned-global-ai`	One ZoneD Global posture dashboard + AI. us-east-1 only.
`alerts-only`	AWSPro alarm logic + notification wiring, without investigation dashboards — alerting without the visual estate.
`custom-sns-topic`	Routes specific VPC/zone alarms to your existing SNS topics (subject_sns_topic_map).
`defaults-full-toggle`	Every supported toggle at provider defaults — a reference to copy from; deploys the entire estate as written.

Operational-pattern templates

Three templates support rollout and operational workflows rather than a capability area.

Template	What it deploys / when to use it
`no-alert-actions-muted`	Dashboards and alarms with all notification actions muted. The recommended pattern for your first week (see Chapter 11): observe normal volumes and tune thresholds without alert noise, then unmute by re-enabling actions and re-applying.
`all-dashboards-disabled-all-sections-enabled`	A dry-run scaffold: licensing, telemetry sources, and SNS wiring with every dashboard disabled. Validates the foundation before deploying the dashboard estate.
`multi-prefix-collision-test`	A second parallel stack — same license and scope, different prefix, separate folder. Useful for verifying prefix-scoped naming or running a staging copy alongside production.

4.5 Built-in validation

The builder checks your configuration against the provider's own rules as you go. The sidebar marks any step that has issues, and each step shows its specific errors and warnings inline at the top, so you fix problems where they occur rather than discovering them at apply time. For example, if you enable the DNS Firewall dashboard for a VPC but have not provided its rule group, the builder flags exactly that, on the step where you would fix it.

> **Import an existing configuration**
> Already have a terraform.tfvars? Use Import to load it. The builder reads it, lets you edit it in the wizard, and re-exports it — preserving any settings it does not directly manage. This is the easy way to make a small change to an existing deployment.

4.6 Exporting a complete deployment bundle

When your configuration is clean, export it. The builder produces your terraform.tfvars, and can include the standard main.tf and variables.tf so you receive a complete, ready-to-apply bundle — drop the three files into a folder and run terraform init. See Chapter 5 for what main.tf and variables.tf contain.

5. Your Terraform Project Files

An AWSPro deployment is three files in one folder. The Config Builder produces the first; the other two are standard and rarely change. This chapter shows exactly what each contains so you can deploy with confidence.

File	Purpose
terraform.tfvars	Your configuration — VPCs, zones, dashboards, alarms, notifications. Generated by the Config Builder or copied from a template. This is the file you edit.
main.tf	Declares the Codreum provider and the AWSPro resource, wiring your variables into it. Standard; the same for every customer.
variables.tf	Declares the input variables that terraform.tfvars fills in, with types and validation. Standard.

5.1 main.tf

This file points Terraform at the Codreum private registry and declares the AWSPro resource. Place it in your deployment folder exactly as below. The provider source and version are the same for every customer.

```
terraform {
  required_version = ">= 1.12.0"

  required_providers {
    codreum = {
      source  = "registry.codreum.com/codreum/awspro"
      version = "~> 1.0.0"
    }
  }
}

provider "codreum" {}

resource "codreum_awspro_pack" "network_observability" {
  prefix     = var.prefix
  aws_region = var.aws_region

  license = {
    type       = var.license_type
    license_id = var.license_id
    product    = "awspro"
    vpc_ids    = var.license_vpc_ids
    zone_ids   = var.license_zone_ids
    features   = var.license_features
  }

  modules = {
    vpcd        = { enabled = var.enable_vpcd }
    zoned       = { enabled = var.enable_zoned }
    lite_alarms = var.lite_alarms
  }

  subject_log_group_map = var.subject_log_group_map
  subject_sns_topic_map = var.subject_sns_topic_map

  dns_alert_emails          = var.dns_alert_emails
  dns_alert_sms_numbers     = var.dns_alert_sms_numbers
  dns_alert_https_endpoints = var.dns_alert_https_endpoints
  sns_kms_master_key_id     = var.sns_kms_master_key_id

  enable_slack_notifications = var.enable_slack_notifications
  slack_workspace_id         = var.slack_workspace_id
  slack_channel_id           = var.slack_channel_id

  ai_insights = var.ai_insights
  scheduled_reports = var.scheduled_reports
  vpcd        = var.vpcd
  zoned       = var.zoned
  tags        = var.tags
}
```

About the provider source

`registry.codreum.com/codreum/awspro` is Codreum's private registry. Your Terraform must be able to reach it during `terraform init` to download the provider. Requires Terraform 1.12.0 or newer.

5.2 variables.tf

This file declares every input variable that terraform.tfvars supplies. You normally do not edit it — it defines types, defaults, and validation (for example, that license_type must be awspro or basic, and that the prefix is lowercase alphanumeric). Key variables:

Variable	Meaning
prefix	Deployment prefix; AWSPro resources are named <prefix>-<region>-… Lowercase alphanumeric/hyphen, max 33 chars. Must be unique per account and region; see Section 6.7 for running multiple deployments.
aws_region	Deployment region. ZoneD requires us-east-1.
license_type	awspro (Premium) or basic (Basic Lite).
license_id	Premium license ID; null for Basic. Marked sensitive.
enable_vpcd / enable_zoned	Turn each module on or off.
license_vpc_ids / license_zone_ids	The licensed VPCs and hosted zones in scope.
subject_log_group_map	Maps each VPC/zone to its CloudWatch log group ARN.
dns_alert_emails / _sms_numbers / _https_endpoints	Subscribers on the default SNS alert topic.
ai_insights	AI configuration (guided cards; optional in-account Bedrock).
vpcd / zoned	The module scope, sources, dashboards, and alarms (built by the Config Builder).
license_features	Entitled modules. Now only “Network” is available, which is required to enable VpcD/ZoneD under Premium

Getting the standard files

main.tf and variables.tf are provided with AWSPro and are the same for every deployment. The Config Builder can also include them — see Section 4.6 — so you can download a complete, ready-to-apply bundle (main.tf + variables.tf + terraform.tfvars) in one step.

variables.tf in full

For reference, the complete variables.tf is below. It is standard; you normally do not edit it.

```
variable "prefix" {
  type        = string
  description = "Customer deployment prefix. Resources use <prefix>-<aws_region>-..."
  validation {
    condition     = can(regex("^[a-z0-9]([a-z0-9-]{0,31}[a-z0-9])?$", var.prefix))
    error_message = "prefix must be lowercase alphanumeric/hyphen, start and end with a letter or
number, at most 33 chars."
  }
}

variable "aws_region" {
  type        = string
  description = "AWS Region where AWSPro resources are deployed. ZoneD requires us-east-1."
}

variable "license_type" {
  type        = string
  description = "Use awspro for Premium or basic for Basic Lite."
  validation {
    condition     = contains(["awspro", "basic"], var.license_type)
    error_message = "license_type must be either awspro or basic."
  }
}

variable "license_id" {
  type        = string
  description = "AWSPro Premium license ID. Use null for Basic Lite."
  sensitive   = true
  nullable    = true
  default     = null
}

variable "enable_vpcd" {
  type = bool
  description = "Enable VpcD."
  default = true
}

variable "enable_zoned" {
  type = bool
  description = "Enable ZoneD."
  default = false
}

variable "license_vpc_ids"  {
  type = list(string)
  description = "Licensed VPC IDs."
  default = []
 }

variable "license_zone_ids" {
  type = list(string)
```

```
  description = "Licensed zone IDs."
  default = []
}

variable "license_features" {
  type = list(string)
  description = "Coarse module entitlements. "network" entitles the VpcD and ZoneD modules under
Premium and is required when either is enabled; Basic Lite uses an empty list.
  default = []
}

variable "lite_alarms" {
  type        = any
  description = "Basic Lite starter-alarm overrides. Ignored in Premium mode."
  default     = {}
}

variable "subject_log_group_map" {
  type        = map(string)
  description = "Map of VPC/zone ID to CloudWatch Logs log group ARN."
  default     = {}
}

variable "subject_sns_topic_map" {
  type        = map(string)
  description = "Optional map of subject ID to existing SNS topic ARN for subject-specific
routing."
  default     = {}
}

variable "dns_alert_emails" {
  type = list(string)
  description = "Email subscribers."
  default = []
}

variable "dns_alert_sms_numbers" {
  type = list(string)
  description = "SMS subscribers (E.164)."
  default = []
}

variable "dns_alert_https_endpoints" {
  type = list(string)
  description = "HTTPS subscribers."
  default = []
}

variable "sns_kms_master_key_id" {
  type        = string
  description = "Optional customer-managed KMS key for the SNS topic. Do not use alias/aws/sns."
  nullable    = true
```

```
  default     = null
}

variable "enable_slack_notifications" {
  type = bool
  description = "Enable Slack (Premium)."
  default = false
}

variable "slack_workspace_id" {
  type = string
  nullable = true
  default = null
}

variable "slack_channel_id" {
  type = string
  nullable = true
  default = null
}

variable "ai_insights" {
  type        = any
  description = "AI Insights config. Guided mode adds triage cards; Bedrock mode invokes an in-
account model."
  default = {
    enabled             = true
    mode                = "guided"
    guided_triage_cards = true
    bedrock = {
      enabled               = false
      model_id              = null
      region                = null
      max_tokens            = 700
      temperature           = 0.2
      include_raw_logs      = false
      include_topn_values   = true
      include_alarm_context = true
    }
  }
}

variable "vpcd" {
  type = any
  description = "VpcD scope/source/dashboard/alarm config."
  default = {}
}

variable "zoned" {
  type = any
  description = "ZoneD scope/dashboard/alarm config."
  default = {}
```

```
}

variable "tags"  {
  type = map(string)
  description = "Tags applied to managed resources."
  default = {}
}

variable "scheduled_reports" {
  type        = any
  description = "Scheduled Reports configuration (Premium)."
  default = {
    enabled = false
    reports = []
  }
```

6. Deploying AWSPro

6.1 What you need

- Terraform installed on your machine (version 1.12.0 or newer).
- AWS credentials for the target account with permission to create CloudWatch dashboards, alarms, SNS topics, IAM roles, and (where used) Lambda functions.
- A configuration — either a deployment bundle from the Config Builder, or a terraform.tfvars from a template.
- Your prerequisites in place (Chapter 3) for the capabilities you enabled.

6.2 Configuring AWS Credential

Terraform reads your AWS credentials from the environment when you run apply, so set them in the same terminal session first. The deployment Region comes from your configuration (the aws_region you chose in the Config Builder), so your credentials only need to be valid for the target account. Any standard AWS credential method works; the most common are below.

From the AWS access portal (IAM Identity Center). If you sign in through the AWS access portal, this is the quickest path:

- Open the access portal and expand the account you are deploying into. It lists the roles you have been granted.
- Choose the role you want to deploy with — it must allow the actions in 15.1 — then choose Access keys.
- On the macOS and Linux tab, copy the block under Option 1, Set AWS environment variables (the three export lines). The Windows and PowerShell tabs show the same commands for those shells.
- Paste the lines into the terminal you will deploy from, then run terraform apply in that same terminal.

```
export AWS_ACCESS_KEY_ID="ASIAEXAMPLE0123456789"
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMIEXAMPLEKEY1234567890abcd"
export AWS_SESSION_TOKEN="IQoJb3JpEXAMPLEtemporarysessiontoken..."
```

These portal credentials are temporary and carry a session token, so when they expire you repeat the steps for a fresh set. To avoid copying them each time, run aws configure sso once — the portal shows the SSO start URL and Region to use — and then deploy with AWS_PROFILE set to that profile.

Other methods. If you keep credentials in your AWS credentials file (Option 2 in the same portal panel pastes a named profile into ~/.aws/credentials for you), select it for the deployment by setting AWS_PROFILE to that profile name. For a long-lived IAM user, set AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY, or run aws configure to save them in a profile.

Confirm before you apply. Check that you are pointed at the intended account and identity:

```
aws sts get-caller-identity
```

The Account in the output should be your target account, and the role should be the one you expect. Then run terraform plan and terraform apply in that terminal.

6.3 The fastest path — deploy from a Config Builder bundle

If you used the Config Builder (Chapter 4), the quickest way to deploy is the bundle it produces. The bundle is a complete, ready-to-apply Terraform folder, so there is nothing else to assemble.

1. On the Review step, choose Download deployment bundle (.zip).
2. Unzip it into an empty folder. It contains main.tf, variables.tf, your terraform.tfvars, and a README.
3. In that folder, run terraform init to download the Codreum provider.
4. Run terraform plan to review what will be created, then terraform apply to deploy.

> **That is the whole process**
> Because the bundle already contains main.tf and variables.tf, you do not need to write any Terraform by hand. Unzip, init, apply.

6.4 Deploying from a template instead

If you started from a template rather than the builder, place your terraform.tfvars alongside the standard main.tf and variables.tf (Chapter 5) in one folder, then run the same init / plan / apply sequence.

1. Put terraform.tfvars, main.tf, and variables.tf in the same folder.
2. terraform init
3. terraform plan — confirm it will create the dashboards and alarms you expect.
4. terraform apply

6.5 How long apply takes — and why you should wait

> **Let apply finish — do not interrupt it**
> A full deployment typically takes 8–15 minutes to converge, depending on how many dashboards, alarms, and Lambdas your configuration includes. This is normal. Terraform is creating and wiring many AWS resources in your account.
>
> Do not press Ctrl+C or close the terminal while apply is running. Interrupting a Terraform apply can leave the deployment half-created and require a follow-up apply to reconcile. Start it, then let it complete.

Larger configurations (many VPCs, all dashboards, all alarm families, AI with Bedrock) sit at the upper end of that range; a small Basic Lite deployment is at the lower end. If apply appears to pause, it is usually waiting on AWS to finish creating a resource — that is expected.

6.6 What you will see — drift detection and repair

AWSPro is declarative and self-healing. On every plan and apply, it compares what is actually in your account against your configuration and repairs anything that has drifted. Understanding the output prevents confusion.

"0 changed" but AWSPro still acted

During a refresh, Terraform may report that nothing changed in your configuration, yet note that AWSPro restored something. This means a managed resource (a dashboard or alarm) had been altered or deleted out of band — for example, someone edited it in the console — and AWSPro put it back to match your desired state. The output lists what was restored. This is the system working correctly, not an error.

> **Example: repairable drift**
> Someone deletes an AWSPro-managed alarm in the console. On the next plan/apply, AWSPro detects the alarm is missing, reports that it will recreate it, and restores it. Your configuration did not change — AWSPro simply re-enforced it. No action is needed from you beyond letting the apply run.

Warning-only findings

Some findings are informational and never block or change anything. The most common is an unconfirmed notification subscription: if you added an email address but have not yet clicked the AWS confirmation message, AWSPro surfaces a warning-only finding noting the subscription is pending confirmation (expected vs. actual). It clears the moment you confirm, and it never stops a deployment. Warnings tell you something to look at; they do not indicate failure.

When you want to change something

To intentionally change or remove a resource, edit your configuration and apply — never edit or delete AWSPro-managed resources in the console, because the next apply restores them. For example, to remove a dashboard, disable it in your configuration; to remove a whole module, set its enable flag to false.

6.7 Switching tiers and modes later

You can move between Basic Lite and Premium, or enable/disable modules, by changing your configuration and re-applying. Because AWSPro is declarative, the apply reconciles your account to the new desired state — adding what is now enabled and removing what is now disabled. Disabling a module does not delete the tuning you configured for it; that configuration is preserved in your file and takes effect again if you re-enable the module.

6.8 Running more than one AWSPro deployment in the same account

You can run multiple independent AWSPro deployments side by side in the same AWS account and region — for example, a production stack and a staging stack, or two stacks owned by different teams. Every AWSPro-managed resource is named <prefix>-<region>-… and every metric namespace begins with your prefix, so deployments never touch each other's resources as long as each deployment uses a unique prefix. Keep each deployment in its own Terraform folder with its own state.

A single Premium license may back multiple deployments. Two stacks can reference the same license ID and even the same licensed VPCs and hosted zones, provided their prefixes differ — license validation checks entitlement, not how many stacks use it. The multi-prefix-collision-test template ("Second parallel stack" in the Config Builder) is a ready-made starting point for this: same license and scope, different prefix, separate folder.

One rule: never reuse a prefix. Two deployments with the same prefix in the same account and region are not supported. They would compute identical resource names, and each stack's drift enforcement (Section 2.2) would treat the other's resources as drift — the two applies would continuously overwrite each other. There is no error message for this; it simply misbehaves. If you need a second stack, choose a new prefix.

7. Licensing

This chapter explains how AWSPro licensing works so there are no surprises: which tier needs a license, how validation happens, how often, and what occurs if validation cannot reach Codreum.

7.1 Tiers and licenses

- **Basic Lite** requires no license. Set license_type = basic and license_id = null. It makes no license-validation calls.
- **AWSPro Premium** requires a license ID. Set license_type = awspro and provide your license_id, along with the licensed VPC and zone IDs it covers.

> **Getting a license**
> To obtain or renew an AWSPro Premium license, contact Codreum (see Chapter 18). If you are evaluating, start with Basic Lite — no license required — and upgrade by switching the tier in your configuration.

7.2 What the license covers

Your license entitles two things: a set of subjects and a set of modules.

Subjects — the specific VPCs and hosted zones in license_vpc_ids and license_zone_ids. AWSPro monitors those licensed subjects; keep the IDs in sync with what you actually deploy.

Modules — the license_features list. This is the single source of truth for which Premium modules you may deploy. Today there is one module entitlement, **Network**, which enables both the VpcD and ZoneD modules. A Premium license that uses VpcD and/or ZoneD therefore carries license_features = ["network"].

Under AWSPro Premium you must include "network" in license_features whenever VpcD or ZoneD is enabled. If you enable a module without its entitlement, the apply fails with a clear message telling you to add it — and your license must have been issued with it. Basic Lite is the free tier and is never gated: it uses an empty list (license_features = []) and ignores the field.

Why a separate field? The features list lets your plan grow one flag at a time. Today network entitles VpcD and ZoneD; future modules are added the same way (for example license_features = ["network", "application"]) with no change to how you deploy. Editing your config can only ever *reduce* what deploys — it can never grant a module your license was not issued for

7.3 How validation works

Premium deployments include a small license-status function that runs in your own account. On a schedule, it confirms your entitlement with Codreum's license service and records the result in your account so the dashboards know the license is healthy.

Aspect	Behavior
Where it runs	A Lambda in your own AWS account (not Codreum's).
How often	Every 4 hours, on a schedule AWSPro creates.
What is sent	Only entitlement metadata — account ID, license ID, license type,licensed VPC/zone IDs, and the entitled modules (features) No telemetry (see Chapter 13).
Where the result is stored	Cached in your account (SSM parameter) so dashboards have a recent, signed license status.
Endpoint	`https://api.codreum.com/api/license/validate`

7.4 License health and alarms

AWSPro publishes a license-status signal and creates an alarm on it so you are alerted if the license becomes invalid, expired, or unreachable. Two points worth knowing:

- **Healthy cadence:** validation publishes on the normal 4-hour cadence. A healthy license keeps the status signal green.
- **Tolerance for a missed check:** the license alarm uses a 5-hour window, so a single missed or slow validation does not trip it — only a sustained gap does. This avoids false alarms from transient network blips.

> **If validation cannot reach Codreum**
>
> Because the status is cached in your account, a brief inability to reach the license service does not immediately disrupt your dashboards. A sustained failure (beyond the alarm window) raises the license alarm so you can investigate — typically egress/network access to the license endpoint, or a license that needs renewal.

7.5 Keeping the license endpoint reachable

The license-status Lambda in your account must be able to reach https://api.codreum.com/api/license/validate. If your account restricts outbound network access, ensure that endpoint is allowed so periodic validation can succeed. Basic Lite has no such requirement (it makes no validation calls).

8. Verifying Your Deployment

After applying, confirm telemetry is flowing and that alarms can reach you. This short verification routine catches the common prerequisite gaps before they become surprises.

8.1 Confirm dashboards are populating

1. Open CloudWatch → Dashboards and find your AWSPro dashboards (named with your prefix).
2. For each enabled dashboard, confirm widgets show data after a few minutes of real traffic.
3. An empty dashboard almost always means a missing or misrouted prerequisite — revisit the relevant section of Chapter 3.

8.2 Confirm notifications reach you

AWSPro always creates a default SNS alert topic and points alarms at it. To actually receive alerts you must add and confirm subscriptions.

> **Confirm your email subscription**
> When you add an email address, AWS sends a Subscription Confirmation message that you MUST click. Until you do, AWSPro's drift check will show an informational, warning-only finding noting the subscription is pending confirmation. It is never enforced and clears as soon as you confirm. This is expected — not an error.

End-to-end notification test (proves the whole path from metric to inbox):

1. Add your email to the alert emails, apply, and click the confirmation email.
2. Temporarily lower a starter-alarm threshold (for example, set the NXDOMAIN spike threshold to 1) and apply.
3. From an instance in a monitored VPC, query a few non-existent domains.
4. Within ~10–15 minutes the alarm transitions to ALARM and emails you; recovery sends an OK email.
5. Restore the threshold and apply.

> **SMS troubleshooting**
> SMS has no confirmation step, but delivery depends on your account's SNS SMS settings — spend limit, origination identity / sender ID per region, and whether your account is still in the SMS sandbox (which only delivers to verified numbers). "No SMS received" is almost always account-level SNS configuration, not the product.

9. The Dashboards

This chapter is a visual tour of what AWSPro deploys, so you know what to expect and what each surface is for. Your dashboards will show your own data; the screenshots use sample environments.

9.1 Global VPC Summary

A cross-VPC operational snapshot: health scores, DNS quality and security, flow activity and traffic volume, platform drop risk, and observability coverage — ranked so the VPC that needs attention surfaces first.

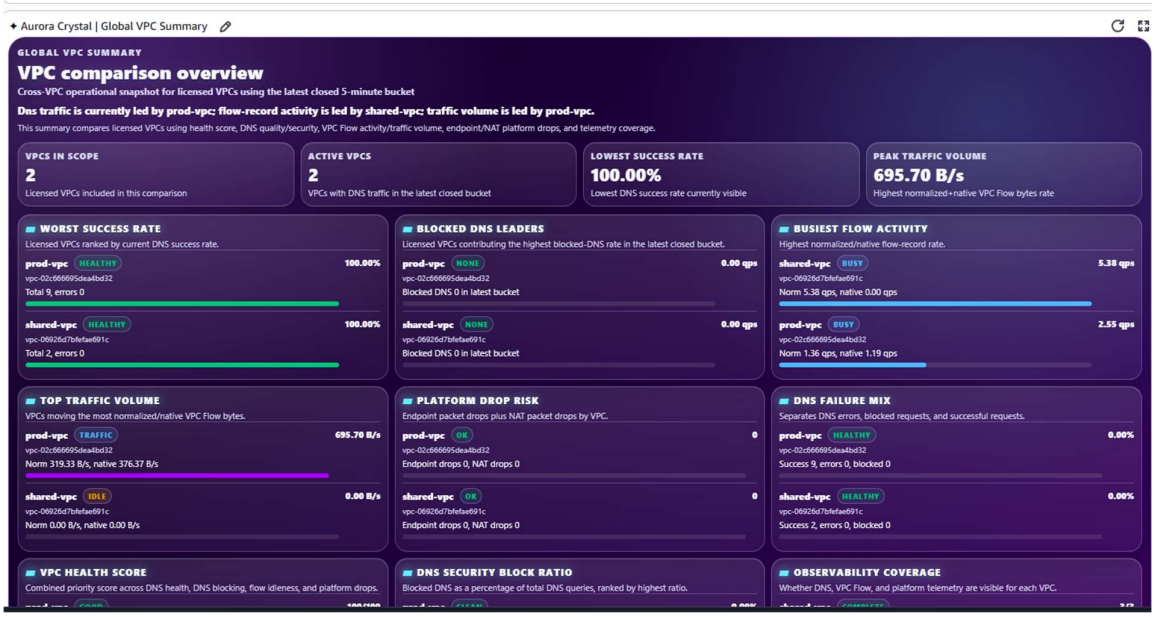

Figure 9.1 — Global VPC Summary: compare every monitored VPC at a glance.

9.2 VPC DNS

Query volume and composition, success and error rates, NXDOMAIN / SERVFAIL / REFUSED trends, transport mix, and the busiest querying sources — the core view of DNS health for a VPC.

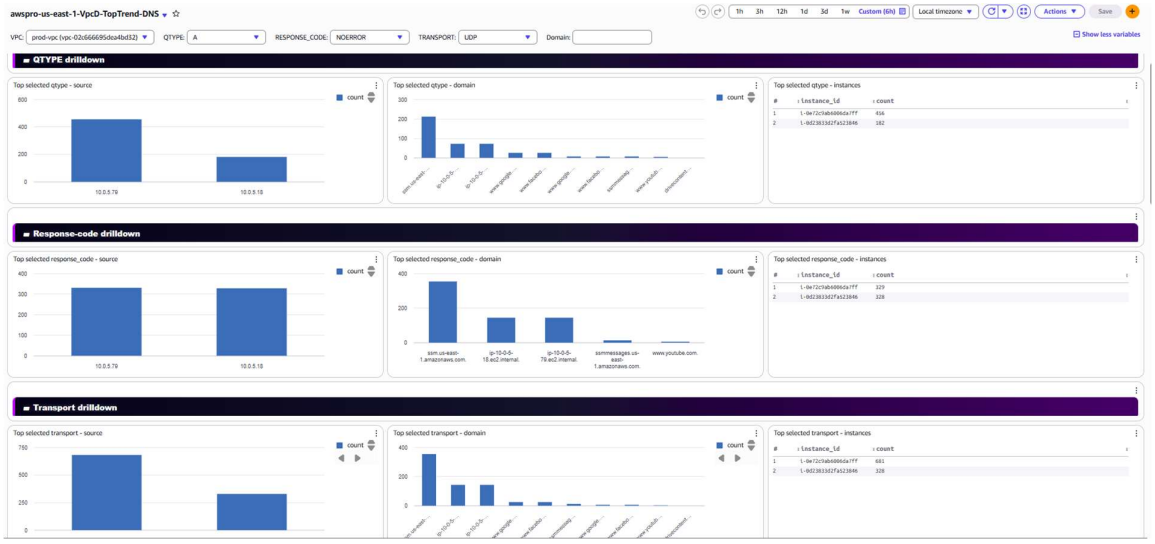

Figure 9.2 — VPC DNS dashboard.

9.3 DNS Firewall

Blocked-query trends, rule-group activity, and threat protections (DGA, dictionary-DGA, DNS tunneling) — the security view of what your DNS Firewall policy is stopping.

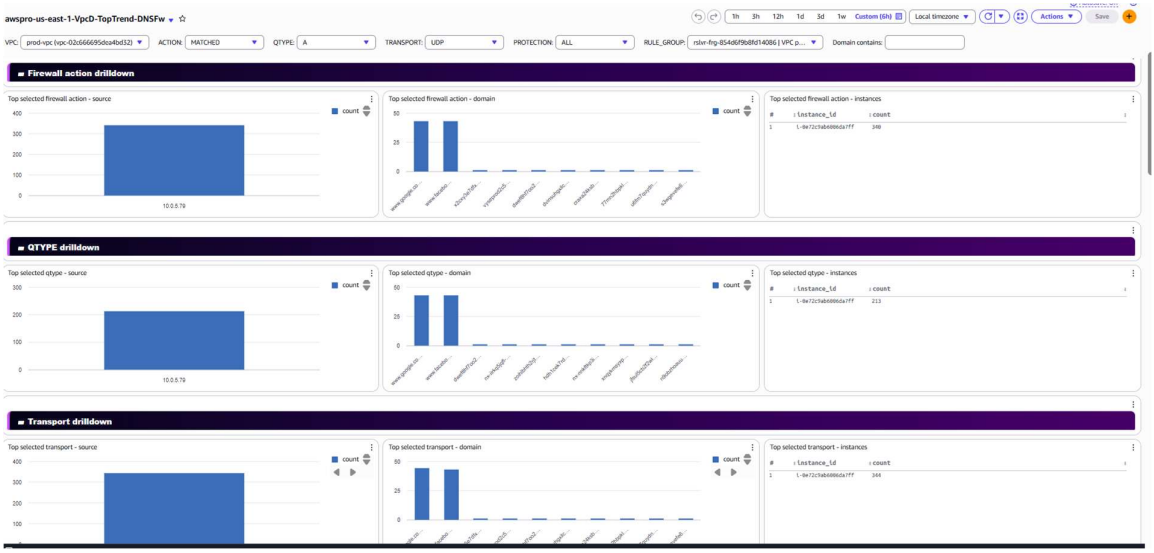

Figure 9.3 — DNS Firewall dashboard.

9.4 VPC Flow

Traffic volume, accepts versus rejects, top talkers, and connection-level signals from your VPC Flow Logs. (See Section 3.4 for the two consumption paths.)

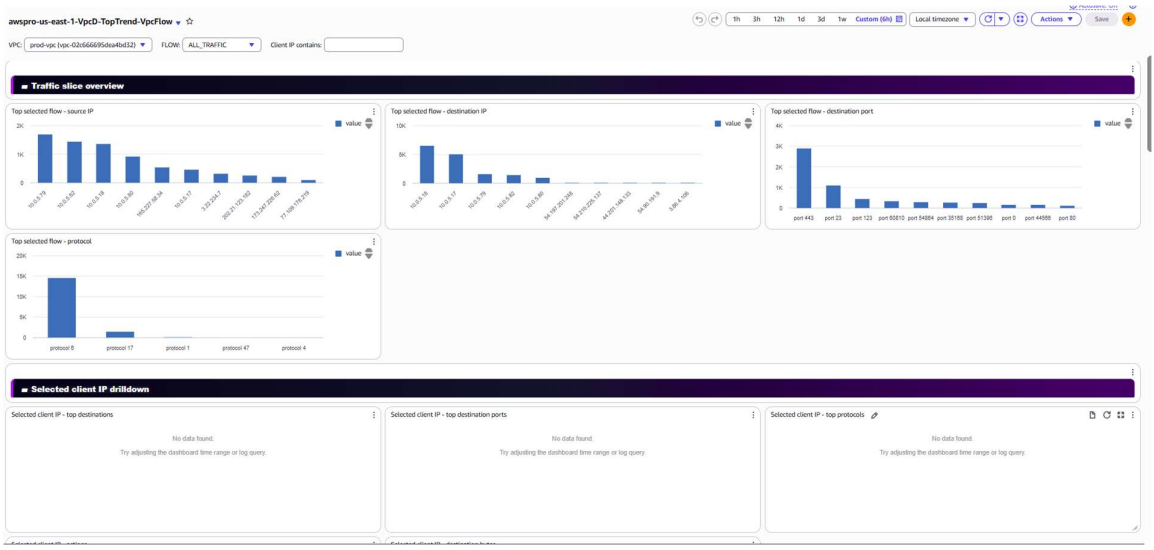

Figure 9.4 — VPC Flow dashboard.

9.5 VPC Platform

VPC endpoint and NAT gateway health: connections, bytes, packet drops, NAT port allocation and delivery ratios, and Network Address Usage against AWS limits.

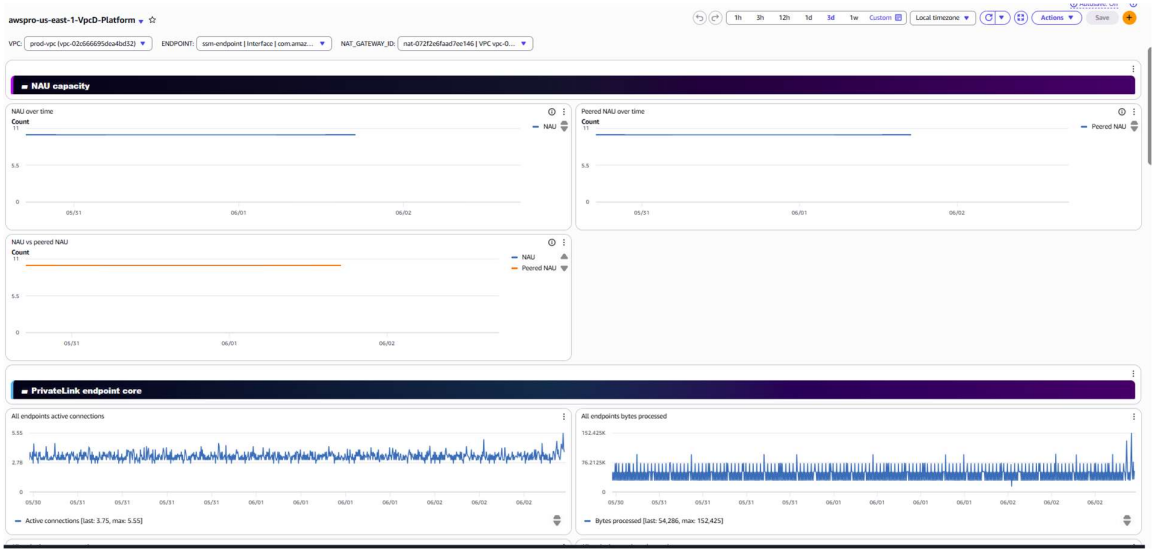

Figure 9.5 — VPC Platform dashboard.

9.6 Hosted Zones (ZoneD)

Per-zone query volume, success rate, NXDOMAIN rate, and query composition, plus a cross-zone global comparison.

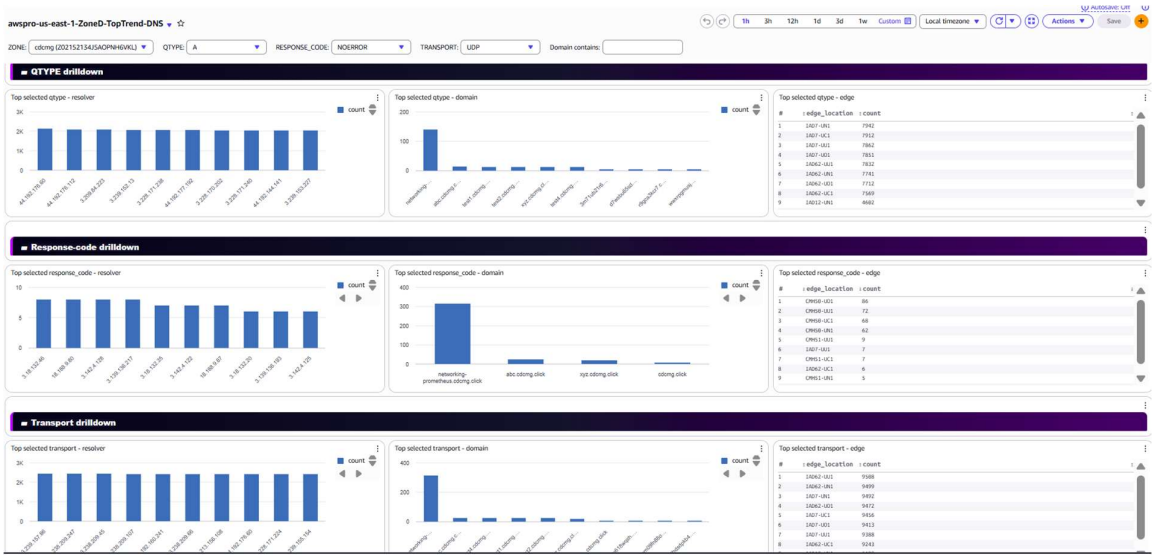

Figure 9.6 — Hosted-zone (ZoneD) dashboard.

9.7 AI Insights (Premium)

Guided triage cards summarize what the dashboards are showing in plain language. With in-account Amazon Bedrock enabled, AWSPro can generate richer analysis and correlation — all within your own account and region. Bedrock model calls happen in your account; Codreum does not run them and does not receive your telemetry.

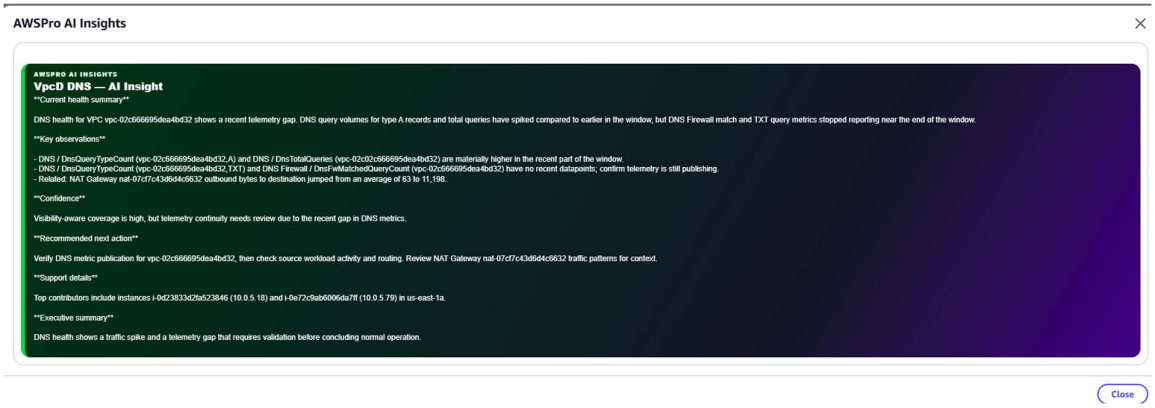

Figure 9.7 — AI Insights summarizing a DNS dashboard.

9.8 Basic Lite dashboards

Basic Lite deploys the Global Lite posture dashboards with a deterministic summary widget and starter alarms — a fast "is my DNS healthy?" view from just your log groups, no license required.

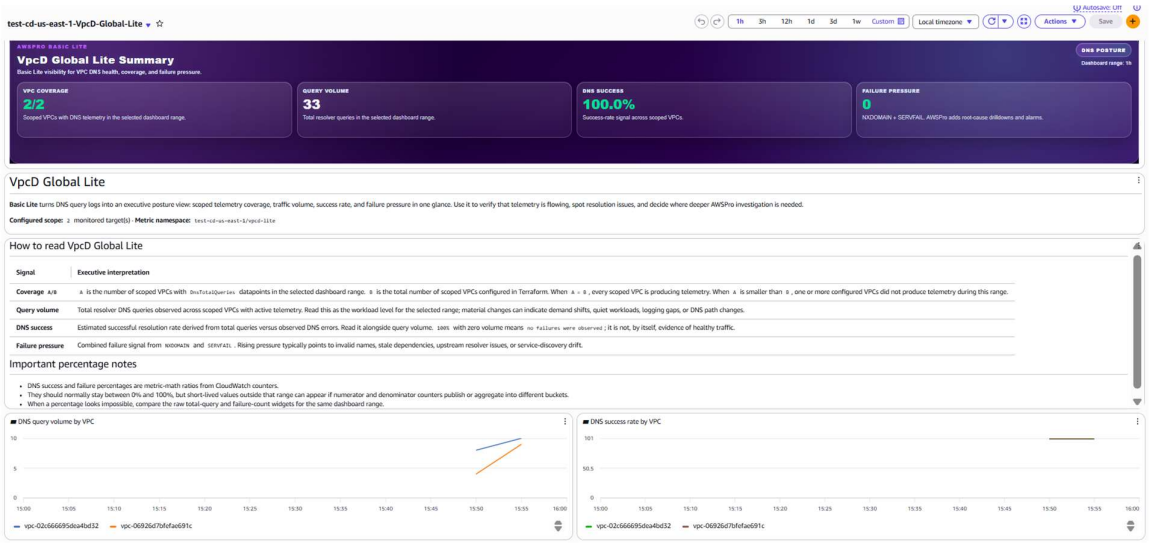

Figure 9.8 — Basic Lite VpcD Global dashboard.

10. AI Insights in Depth

AI Insights adds an interpretation layer on top of your dashboards. Instead of reading every widget yourself, you get a plain-language reading of what the data shows. This chapter explains the two modes, what each setting changes, when to use which, and what to expect — and what not to expect — from the feature.

10.1 The two modes

Mode	What it does
Guided	Generates triage cards deterministically from your dashboard data — no model calls. It summarizes posture (what is healthy, what is elevated, where to look) using the same metrics the dashboards show. Available in both Basic Lite and Premium.
Bedrock	Adds Amazon Bedrock model analysis on top of guided cards: richer narrative explanations and correlation across signals. Premium only. The model runs in your own account and region (see Chapter 13, Security).

> **Start with Guided**
> Guided mode is on by default and costs nothing extra (no model calls). It is the right starting point for everyone. Turn on Bedrock when you want narrative analysis and cross-signal correlation, and you are comfortable with per-token Bedrock cost in your account.

Figure 10.1 — AI Insights summarizing a DNS dashboard.

10.2 Guided mode settings

Setting	What it changes / when to use
`enabled`	Turns AI Insights on or off entirely.
`guided_triage_cards`	Shows the deterministic triage cards on dashboards. Leave on for the at-a-glance posture summary.
`mode`	guided (cards only) or bedrock/advanced (adds model analysis). Basic Lite is always guided.

10.3 Bedrock settings — what each one changes

When you enable Bedrock, these settings control the model, its behavior, and how much context it sees. Defaults are sensible; tune only when you have a reason.

Setting	Default	What it changes / how to tune
`enabled`	false	Turns Bedrock analysis on. Premium only.

Setting	Default	What it changes / how to tune
`model_id`	(provider default)	Which Bedrock model/inference profile to call. Leave blank to use the in-account default global.amazon.nova-2-lite-v1:0 (low cost). Set a different model or inference-profile ARN your account is approved for when you want higher quality.
`region`	deployment region	Which region Bedrock runs in. Leave blank to keep it in your deployment region (recommended — no data crosses a region boundary).
`max_tokens`	700	Maximum length of the model's response (1–4000). Raise for longer narratives; lower to reduce cost/latency. 700 suits concise triage.
`temperature`	0.2	Creativity vs. determinism (0–1). Low (0.2) gives focused, consistent operational analysis. Raise only if you want more varied phrasing; for monitoring, low is best.
`include_topn_values`	true	Lets the model see Top-N tables (top talkers, top rules) for context. Keep on for useful, specific analysis.
`include_alarm_context`	true	Lets the model see current alarm state so its reading reflects what is firing. Keep on.
`include_raw_logs`	false	Whether raw log samples are included in prompts. Off by default for data minimization and cost. Turn on only if you specifically want the model to reason over raw log lines.

10.4 Choosing the model — and overriding the default

By default AWSPro uses **`global.amazon.nova-2-lite-v1:0`** (Amazon Nova 2 Lite, via a Global cross-Region inference profile). To use a different model, set the model_id field — in the Config Builder's AI Insights step, or directly in the ai_insights.bedrock block. Whatever you provide is passed through to Amazon Bedrock as-is, so it must be a model ID or inference-profile ID that is valid and enabled in your Region.

Using the original Nova Lite (lowest cost / widest compatibility)

The original Amazon Nova Lite is roughly ten times cheaper on output tokens than Nova 2 Lite and is available in a very broad set of Regions, including AWS GovCloud (US). It is an excellent choice when you want the lowest possible AI cost, or when you deploy in a Region or partition where Nova 2 Lite is not offered.

In the Config Builder: on the AI Insights step, with Bedrock enabled, set Model ID to one of the values below (instead of leaving it blank).

In terraform.tfvars: set model_id inside the ai_insights.bedrock block.

```
ai_insights = {
  enabled = true
  mode    = "bedrock"

  bedrock = {
    enabled  = true

    # Original Nova Lite — lowest cost. Choose the form valid for your Region:
    #   bare model ID (single-Region):
    model_id = "amazon.nova-lite-v1:0"
    #   or a cross-Region inference profile (recommended for resilience):
    #   model_id = "us.amazon.nova-lite-v1:0"   # US Regions
    #   model_id = "eu.amazon.nova-lite-v1:0"   # EU Regions

    # region = "us-east-1"   # optional: pin Bedrock to a specific Region
  }
}
```

Which model_id form should I use?

Use a cross-Region inference profile form (us. or eu. prefix) when you want automatic resilience across Regions in that geography — this matches how the Nova 2 Lite default works (its global. prefix is a worldwide profile). Use the bare model ID (amazon.nova-lite-v1:0) when you want the model pinned to your single deployment Region, for example to satisfy data-residency requirements. Whichever you choose, it must be enabled for your account in that Region (Bedrock → Model access).

To go back to the default

Leave model_id blank (or remove it) and AWSPro uses `global.amazon.nova-2-lite-v1:0` again.

One model ID to avoid

Do not set model_id to the direct legacy ID `anthropic.claude-3-5-haiku-20241022-v1:0` — it may not support on-demand invocation and AWSPro will reject it at apply time. Use an inference-profile form instead (for example `us.anthropic.claude-haiku-4-5-20251001-v1:0`), or a model your account is approved for.

Verify availability first: model and Region support change over time. Before standardizing on a model, confirm it is enabled for your account in your Region under Bedrock → Model access, and check its current price on the AWS Bedrock pricing page.

10.5 Increasing AI analysis capability and coverage

Out of the box, AI Insights is tuned for fast, low-cost summaries. If you want sharper analysis, broader coverage, or deeper detail, AWSPro exposes the levers below. They live in the ai_insights.bedrock and ai_insights.advanced blocks (the Config Builder surfaces them on the AI Insights and Advanced steps). Group your changes by what you are trying to improve.

Smarter, more capable analysis

- Use a stronger model: set model_id to a more capable Bedrock model or inference profile your account is approved for (for example a Nova Pro or a Claude inference profile). This is the single biggest lever on analysis quality — and the biggest cost lever, so weigh it against the cost guidance in Chapter 16.
- **Allow longer reasoning:** raise bedrock.max_tokens (default 700, range 1–4000) so the model can produce fuller explanations rather than truncating.
- **Keep correlation on:** advanced.correlation_enabled (default on) lets the model relate signals to one another rather than reading each in isolation — leave it enabled for root-cause-style insight.

Broader coverage (more context the model can see)

By default the model receives summarized metric, alarm, and Top-N context, but not raw logs. Widen what it sees with the collect_* flags:

Setting	Default	Effect
`advanced.collect_metric_context`	on	Include summarized metric context. Core context; keep on.
`advanced.collect_alarm_context`	on	Include current alarm state so analysis reflects what is firing.
`advanced.collect_topn_context`	on	Include Top-N tables (top talkers, top rules).
`advanced.collect_log_context`	off	Include log-derived context. Turn on for deeper investigation; increases prompt size, latency, and cost.
`bedrock.include_raw_logs`	off	Include raw log samples in the prompt. The strongest coverage increase — and the largest cost/privacy impact. Enable deliberately.

Deeper detail (how much the model pulls in)

Setting	Default	Range	Effect of raising it
`advanced.max_topn_items`	5	1–20	More entries per Top-N table — e.g. top 15 talkers instead of 5.
`advanced.max_metric_queries`	48	1–200	More metrics gathered per analysis — broader signal base.
`advanced.max_log_queries`	4	1–12	More log queries when log context is on — deeper log coverage.
`advanced.topn_query_mode`	safe	off / safe / full	full runs richer Top-N queries; safe balances cost; off disables.
`advanced.metric_period_seconds`	300	60–3600	Lower = finer-grained metric resolution; higher = coarser, cheaper.
`advanced.max_query_seconds`	12	5–60	Allows longer data-gathering before the analysis runs.

Richer output

- `advanced.output_profile` (operator / support / executive / full): chooses how much the written analysis includes. The default is **full**, the most comprehensive write-up; switch to **operator** for the tightest output, or **support** / **executive** to focus on those audiences.
- `advanced.support_summary_enabled / executive_summary_enabled` (both on by default): keep on to generate support- and executive-oriented summaries alongside the operator view.
- `advanced.operator_notes` (off by default): a switch that adds operator-facing diagnostic notes to the AI output — for example, a note that AI context collection is already at maximum depth and no further Terraform tuning would help. It is off by default so customer-facing widgets stay free of internal configuration vocabulary; turn it on for operator or diagnostic deployments.

Capability vs. cost and latency

Every lever that widens context, deepens detail, lengthens output, or upgrades the model increases prompt size and/or model cost and the time each analysis takes. Raise them deliberately — change one group at a time, observe the improvement and the cost, and stop when the value plateaus. The defaults are a deliberate balance; you do not need to maximize everything.

A sensible "high-capability" starting point

If you want noticeably richer analysis without going to extremes: keep the default model or move one tier up, raise max_tokens to ~1500, and raise max_topn_items to ~10. The output_profile default is already full, so no change is needed there. Turn on collect_log_context only when you are actively investigating, and leave include_raw_logs off unless you specifically need raw lines.

10.6 When to use which

- **Just want posture at a glance, no cost:** Guided only (the default). Triage cards summarize health on each dashboard.
- **Want narrative explanations and root-cause hints:** Enable Bedrock with the default model. Keep temperature low and raw logs off.
- **Want the highest-quality analysis and accept higher cost:** Set a more capable model_id (an approved Bedrock model or inference profile) and consider raising max_tokens.
- **Strict data-minimization environment:** Keep include_raw_logs off (default); the model still gets summarized metric and alarm context, not raw log content.

10.7 What to expect — and what not to

What AI Insights is good at

Summarizing posture in plain language, highlighting which VPC/zone needs attention, explaining what a spike or elevated rate likely means, and correlating signals (Bedrock) so you start your investigation in the right place.

What AI Insights is not

It is not an autonomous remediator — it explains and prioritizes; it does not change your network or your AWS resources. It is not a replacement for the dashboards — it interprets them. And it is not a data exfiltration path: prompts and responses stay in your account and region, and raw logs are excluded by default.

Cost reminder

Guided mode makes no model calls and adds no Bedrock cost. Bedrock mode is billed per token by AWS in your account, and it is small by default — see Section 15.1 for realistic estimates and how to keep it that way. The default lite model keeps cost low; max_tokens caps response size; and excluding raw logs (the default) keeps prompts small.

11. Scheduled Reports (Premium)

Scheduled Reports turns your live dashboards into a recurring written digest. On a daily, weekly, or monthly schedule, AWSPro looks at your deployment, compares the period with the one before it, writes a plain-language summary of how it went, and delivers it to the people who need to see it — without anyone having to open a dashboard. It is built on the same in-account engine as AI Insights, runs entirely inside your account, and by default delivers through the alert topic you already have. A report can go out as a plain-text digest or as a styled, multi-page PDF.

> **Premium, with the network entitlement.** Scheduled Reports reads the signed license-status cache that the Premium license watcher maintains, so it is available on AWSPro Premium only. Because a report summarises your network modules, it also requires the **network** feature entitlement and at least one of VpcD or ZoneD enabled — with neither, there is nothing to report on and the feature stays off (see §11.6). Basic Lite has no watcher and no cache, so the feature stays off there.

11.1 What a report contains

Each run gathers a bounded snapshot of the period, scoped to your deployment (and optionally to specific subjects — see §11.5), and turns it into a short briefing. The report is ordered scope first, then alarms and anomalies, then per-dashboard traffic, then the opt-in top sources, then a per-subject comparison:

Section	What it shows
Cover & health verdict	A one-line health verdict, an alarm-state donut, and KPI tiles (incidents, data gaps, anomalies, largest flow), plus an “at a glance” line and the reporting window.
Scope & inventory	The licensed VPCs and zones the report covers (ids and friendly names) and resource counts — NAT gateways, VPC endpoints, and DNS Firewall groups — for the in-scope subjects.
Incidents & alarm health	Totals by state (OK / ALARM / insufficient data), coverage %, and the alarms currently firing or missing data.
Anomalies & notable changes	Metrics that moved more than the threshold versus the prior period — new signals and large up/down deltas.
Per-dashboard traffic	Curated headline metrics per dashboard — VPC Flow, VPC DNS, DNS Firewall, then Platform/NAT, VPC capacity (NAU) and endpoints — as comparison bars with last-6-period trend sparklines, plus composition pies (traffic-by-path, direction, protocol, query-type, firewall actions and threat protections) and a DNS-Firewall block-rate callout.
Hosted zones (ZoneD)	Hosted-zone query activity, rendered as its own section after the VpcD content so the two never interleave.
Top sources & destinations	(Opt-in.) Top source IPs, destination domains and error domains for DNS and hosted zones, and top source/destination IPs for VPC flow — read from your query and flow logs. See §11.7.
Per-subject comparison	A block per individual VPC and zone: its key metrics this period versus the prior period, with a trend.

Metric activity is discovered from your prefix’s namespaces rather than a fixed list, so any module you add later is summarised automatically, and the curated per-dashboard views surface the signals that matter (DNS errors, flow rejects, SYN counts, firewall blocks) rather than a volume-sorted dump. A report is delivered as the plain-text digest or as a styled PDF; the PDF lays the same content across multiple pages with links back to the matching CloudWatch dashboards and alarms. Format is set per report — see §11.3.

11.2 Cadence and timing

Cadence	When it runs (default)	Period it covers	Compared against
daily	Every day, 13:00 UTC	the last 24 hours	the prior day
weekly	Mondays, 13:00 UTC	the last 7 days	the prior week
monthly	1st of the month, 13:00 UTC	the last 30 days	the prior month

You can run more than one report — a common setup is a daily operations digest plus a monthly executive digest, and the two can use different formats and scopes. If you need a different time, set an explicit schedule override on the report.

A schedule override is a standard **AWS EventBridge** expression, always in **UTC**. A cron has **six** fields — cron(minutes hours day-of-month month day-of-week year) — and exactly one of day-of-month / day-of-week must be a question mark (?). You can also use a simple rate(…). For example:

Expression	Runs
cron(0 0 * * ? *)	Every day at 00:00 UTC (a clean calendar-day window).
cron(0 8 ? * MON-FRI *)	Every weekday at 08:00 UTC.
cron(0 13 ? * MON *)	Mondays at 13:00 UTC (the weekly default).
cron(0 6 1 * ? *)	The 1st of each month at 06:00 UTC.
rate(6 hours)	Every 6 hours.
rate(30 minutes)	Every 30 minutes.

> **Expressions are UTC.** EventBridge does not adjust for time zones or daylight saving, so convert from your local time when you set an override (and re-check twice a year if the local hour matters). Leave the override blank to use the cadence default of 13:00 UTC.

11.3 Report format — text or PDF

Each report sets its format to either text (default) or pdf. Format is per report, so you can run a lightweight text digest daily and a polished PDF weekly or monthly from the same deployment.

- **text** — the plain-language digest delivered in the body of the SNS message (reaching email and Slack). No S3 is involved.
- **pdf** — a multi-page styled report: a cover (health verdict, alarm donut, KPI tiles); then scope & inventory, incidents & alarm health, anomalies & notable changes, the per-dashboard traffic pages (VPC Flow, VPC DNS, DNS Firewall, Platform, NAU, endpoints), the hosted-zone section, the opt-in top-sources section, and the per-VPC / per-zone comparisons. The PDF is written to S3 and linked from the message (see §11.4).

The PDF is rendered inside your account with no external dependencies — there is no Lambda layer to manage — and its sections deep-link back to the matching CloudWatch dashboards and alarms in the console.

11.4 PDF delivery and the report bucket (S3)

PDF reports need somewhere to write the document, so the PDF format uses S3; text reports never touch S3. The S3 settings are set once for the deployment and shared by every PDF report:

Setting	Default	What it does
report_s3_bucket	(blank)	Leave blank to have AWSPro create a secured bucket for you, or set it to bring your own existing bucket.
report_s3_prefix	reports	The key prefix that PDFs are written under.
report_s3_url_ttl	43200 (12h)	Lifetime of the pre-signed download link, in seconds (60–604800).
report_s3_expire_days	14	Lifecycle expiry — PDFs older than this are deleted automatically (1–3650).

When AWSPro creates the bucket, it is secured by default: S3 Block Public Access on, bucket-owner-enforced ownership, SSE-S3 encryption, a bucket policy that denies non-TLS and unencrypted requests, a lifecycle rule that expires old reports, and a deterministic, globally-unique name. If you bring your own bucket, AWSPro writes objects under the prefix and does not change your bucket's settings. The managed bucket is created only when a PDF report will actually run — if reporting is off, or the network modules/entitlement are absent (§11.6), no bucket is created.

> **Set the link lifetime deliberately.** Keep report_s3_url_ttl longer than the gap between delivery and when people actually open the message, but shorter than report_s3_expire_days so a link never outlives its object. An expired link returns an S3 "access denied" — re-run the report or wait for the next scheduled run.

11.5 Scoping a report

By default a report covers every licensed VPC and zone. Use a report's scope to narrow it:

Scope	Covers
["all"]	All licensed VPCs and zones (the default).
["all-vpc"]	All VPCs only.
["all-zone"]	All zones only.
specific ids	Only the listed VPC/zone ids, e.g. ["vpc-0abc1234","Z0123EXAMPLE"] (one or more).

Three rules. The ids must be subjects you have licensed; you cannot mix a keyword (all / all-vpc / all-zone) with specific ids in the same report; and the scope follows the enabled modules — if VpcD is disabled, "all" and "all-vpc" resolve to zones only and a VPC id is not accepted, and if ZoneD is disabled the same applies to zones (see §11.6). When a report is scoped, its metrics, its alarms, and its scope & inventory counts (VPCs, zones, NAT gateways, endpoints and DNS-Firewall groups) are all filtered to the selected subjects — so a "prod-vpc daily" and an "all-zones weekly" can run side by side, each seeing only its own world. A report whose scope resolves to nothing — for example one scoped to a VPC id after VpcD was disabled — is simply not generated.

11.6 What the report includes — modules, dashboards, and sources

The report mirrors what your deployment actually publishes; it does not read your enable switches directly. Three things govern what appears:

Module enablement and entitlement. Reporting summarises the network modules, so it requires the network entitlement and at least one of VpcD / ZoneD enabled. With neither (or the entitlement absent) no reports are generated at all. A disabled module contributes no subjects, so “all” resolves to the enabled modules only (§11.5).

Per-dashboard sections (data-driven). Each per-dashboard section appears when its metrics exist. A dashboard only publishes its metrics while it is enabled, so disabling a dashboard removes its section on its own — disable vpc_flow and the VPC Flow metrics section stops appearing, disable dns and the VPC DNS section goes, disable dns_firewall and the DNS Firewall section (and its block-rate callout) goes. The alarm-health count reflects the alarm dashboards you have enabled, and the “N dashboards” count reflects the dashboards deployed — so the global dashboards affect only that count.

AWS-native sections and top-N (source-map driven). The Platform (NAT), VPC endpoints and NAU sections read AWS-native metrics for whatever you list in the NAT / endpoint scope maps, and the top-sources section (§11.7) reads your raw query and flow logs. These follow their source maps and the report_top_n switch — not the per-dashboard toggles. So, for example, with vpc_flow disabled but a flow log group still mapped and top-N on, the VPC Flow metrics section is gone while the flow top-N rows can still appear.

Report content	Governed by
Whether any report runs at all	the network entitlement, plus at least one of VpcD / ZoneD enabled
VPC Flow / VPC DNS / DNS Firewall / hosted-zone metric sections	the matching dashboard being enabled (its metrics must be published)
Incidents & alarm-health count	the alarm dashboards you have enabled
“N dashboards” count	the dashboards deployed (including the global dashboards)
Platform (NAT) / VPC endpoints / NAU sections	the NAT and endpoint scope maps
Top sources & destinations	report_top_n, plus the query / flow log-group maps

11.7 Top sources & destinations

An opt-in section that turns your query and flow logs into a short “who talked to whom” view. It is **off by default**; set report_top_n = true on a report to enable it.

When on, the report runs a small set of bounded CloudWatch Logs Insights queries against the log groups you have mapped, and lists the top three of each:

Group	Rows
DNS (VPC resolver)	top source IPs · top destination domains · top error domains (NXDOMAIN / SERVFAIL)
Hosted zones	top query names · top error domains · top resolver IPs
VPC Flow	top source IPs · top destination IPs (by bytes)

Rows appear only for the in-scope subjects that have a log group mapped — VPC resolver query logs and hosted-zone query logs feed the DNS and hosted-zone rows, and VPC flow logs feed the flow rows. A deployment that maps none of these, or leaves report_top_n off, shows no top-sources section and runs zero Logs Insights queries (and so incurs zero scan cost — see §11.13). Domain names are shown decoded: a wildcard record that Route 53 writes as \052.example.com is displayed as *.example.com.

> **Top-N reads raw logs directly.** Because it queries your log groups rather than AWSPro metrics, the top-sources section follows the log-group maps and report_top_n, not the per-dashboard toggles (§11.6). It is the one report feature whose cost scales with traffic, so enable it where the source-IP / destination-domain view is worth it.

11.8 Delivery

By default a report is delivered through the default SNS alert topic — the same topic your alarms use. That means it reaches every confirmed email subscriber, and, if you have Slack connected, it also appears in Slack through the existing AWS Chatbot association. One publish reaches both channels, and no new infrastructure is created.

When a report's format is PDF, the Lambda renders the document, writes it to your report bucket, and embeds the time-limited pre-signed download link in the same SNS message — so the link reaches email and Slack exactly as a text digest would, now with a "Download the full PDF" link. PDF delivery degrades safely: if the document cannot be rendered or uploaded for any reason, the report falls back to the full plain-text digest, so a report is always delivered.

> **No email arriving?** A report can only reach an email address that has confirmed its SNS subscription. If a recipient never confirmed, they will not receive reports (or alarms). See Chapter 8, "Confirm notifications reach you."

The ses and slack sinks are reserved as future delivery options; today, sns is the delivery path (and it already covers email and Slack).

11.9 Bedrock narrative vs. deterministic summary

Each report can be written two ways:

Mode	What you get	Cost
Deterministic (default)	A clear, rules-based summary generated from your data — no model calls.	None extra
Bedrock narrative	A richer, plain-language briefing written by the in-account Bedrock model.	One model call per run

Turning on the Bedrock narrative for a report requires AI Insights to be in bedrock or advanced mode with Bedrock enabled (Chapter 10) — reports use the same approved, in-account model. If Bedrock is unavailable for any reason, the report automatically falls back to the deterministic summary, so a report is always delivered.

11.10 Tuning metric coverage (max_metrics)

Each report has a max_metrics setting — by default 120, configurable per report from 1 to 200. It bounds how many distinct metric series the report pulls each run: the candidate pool the analysis draws from, not the length of the digest. The report does not print a long table — it ranks and groups those series into the top-traffic, anomaly, and per-subject views, and the budget is shared fairly across your monitored namespaces so VpcD and ZoneD are both represented. Raising or lowering it therefore changes how complete the analysis is, not how big the report is.

For most deployments, leave it at 120. Cost is not a reason to trim it — CloudWatch metric reads are billed per series, but at these volumes that is a few cents a month even at 200. What changes as you lower it is coverage of the lower-volume signals, as below.

Setting	What it means for the report
Default — 120	Covers a typical small or medium deployment in full, fair-shared across VpcD and ZoneD. Leave it here unless you have a specific reason to change it.
Higher — up to 200	For large fleets: many VPCs multiply the per-protocol and per-flag dimensions into the hundreds of series, so a bigger pool keeps the picture complete.
Lower — 40 to 60	Fine for a scoped report (one or a few specific VPCs or zones); scope-aware discovery spends the whole budget on that subject, so a smaller pool still covers it well.
Too low — 10 to 20	Top traffic and the alarm and incident summary stay reliable, but anomalies and security signals (rejects, SYN floods, flag anomalies) on everything else can be missed. Treat 40 as a practical floor for an all-scope report.

11.11 Enabling Scheduled Reports

In the Config Builder (recommended). On the Advanced step, just below Global defaults, open Scheduled reports and turn it on. Add one report, or several for different audiences. For each report you set:

- **Report name** — a short label (e.g. weekly-ops).
- **Cadence** — daily, weekly, or monthly.
- **Format** — text (the digest in the message) or PDF (styled document linked from the message).
- **Scope** — all VPCs & zones, all VPCs, all zones, or specific subjects (§11.5).
- **Top sources & destinations** — off by default; turn on to add the top source/destination view from your query and flow logs (§11.7).
- **Bedrock narrative** — off for the deterministic summary, on to use the model (shows model id, max tokens, and temperature when on).
- **Max metrics** — how many headline metrics to include (default 120; see §11.10).
- **Schedule override** — optional custom cron(…)/rate(…) in UTC; see §11.2 for the format and examples.

When any report uses the PDF format, set the shared PDF delivery (S3) fields once — bucket (blank to auto-create), prefix, link TTL, and object expiry (§11.4). Leaving the report list empty but the feature enabled deploys a single weekly report on defaults.

> **Reporting follows your network modules.** Scheduled Reports requires the network entitlement and at least one of VpcD / ZoneD enabled. With neither, the feature produces no reports — and no report bucket — regardless of this switch (§11.6).

In terraform.tfvars directly. The same configuration in HCL:

```
scheduled_reports = {
  enabled               = true
  report_s3_bucket      = ""         # blank = AWSPro creates a secured bucket; or your own
bucket name
  report_s3_prefix      = "reports"
  report_s3_url_ttl     = 43200      # pre-signed link lifetime (seconds)
  report_s3_expire_days = 14         # auto-delete PDFs after N days
  reports = [
    {
      name          = "prod-daily"
      period        = "daily"        # every day 13:00 UTC, vs the prior day
      report_format = "pdf"
      report_scope  = ["vpc-0abc1234"]   # one licensed VPC
      report_top_n  = true           # add the top source/destination view (reads query/flow
logs)
      sinks         = ["sns"]        # reaches email + Slack
    },
    {
      name            = "exec-monthly"
      period          = "monthly"    # 1st of month 13:00 UTC, vs the prior month
      report_format   = "pdf"
      report_scope    = ["all"]
      bedrock_enabled = true         # narrative via the in-account model
      max_tokens      = 900
      temperature     = 0.2
      max_metrics     = 120
    },
  ]
}
```

A starter template is included as templates/scheduled-reports-premium.tfvars.example.

11.12 What it reads, and what leaves your account

A report reads only your own telemetry: CloudWatch alarms, dashboards, and metrics under your prefix, plus the license-status cache used to confirm the deployment is healthy. When top sources & destinations is on, it also runs bounded Logs Insights queries against your mapped DNS and flow log groups — reading your raw log content in-account to compute the top talkers. When the Bedrock narrative is on, the prompt and the model response stay inside your account and region, exactly as with AI Insights (Chapter 14). For PDF reports, the document is rendered and stored inside your account, in your own S3 bucket. The only thing that leaves is the finished digest — and, for a PDF, the time-limited pre-signed link to that object — delivered to your own SNS topic and its subscribers. The PDF itself, and your raw log content, are never sent to Codreum or anywhere outside your account. Each run also publishes ReportsSent / ReportsSkipped counters under the codr/reports namespace so you can confirm reports are firing.

11.13 Cost

A deterministic text report costs effectively nothing — a short Lambda run and a handful of CloudWatch reads. A Bedrock-narrative report adds exactly one model call per run (so a weekly report is roughly four to five calls a month, a monthly report is one), which is small with the default low-cost model. A PDF report adds only a little more Lambda runtime to render and S3 storage for small PDFs that the lifecycle rule expires after report_s3_expire_days. CloudWatch metric reads are billed per series but come to pennies a month even at max_metrics = 200.

The one cost that scales with traffic is top sources & destinations. When report_top_n is on, the report runs a few CloudWatch Logs Insights queries, billed per GB scanned (about $0.005/GB), bounded by your log volume and the report's scope and cadence. Leave report_top_n off to avoid it entirely. See §16.1 for how Bedrock pricing works.

11.14 Troubleshooting

Symptom	Likely cause	What to do
No report arrived	An email subscriber never confirmed their SNS subscription	Confirm the subscription (Chapter 8); resend confirmation if needed.
No reports deploy at all	The network entitlement is missing, or neither VpcD nor ZoneD is enabled	Add network to license.features and enable a module (§11.6); reporting needs at least one network module.
No report arrived; ReportsSkipped = 1	The license gate failed (license lapsed, or region/prefix mismatch)	Check license health (§7.4); reports resume automatically once active.
A whole module's content is missing	That module (or its dashboard) is disabled, so its metrics aren't published	Expected — the report follows the enabled modules/dashboards (§11.6). Re-enable to include it.
No "Top sources & destinations" section	report_top_n is off, or no DNS/flow log groups are mapped for the in-scope subjects	Turn on report_top_n and map the query/flow log groups (§11.7).
DNS Firewall section absent	Firewall metrics didn't rank into the period's coverage (low volume)	Expected at low volume; confirm the firewall is publishing, or raise max_metrics (§11.10).
Got the deterministic summary, expected the narrative	Bedrock is off for that report, or the model call failed	Enable AI Insights bedrock/advanced mode and the report's Bedrock narrative; confirm model access in your region.
PDF report arrived as plain text	PDF render or S3 upload failed; the report fell back to text	Check the report Lambda logs; confirm the S3 bucket exists and the role can PutObject.
"Download" link returns access denied	The pre-signed link expired (report_s3_url_ttl elapsed)	Open links sooner, raise the TTL, or wait for the next scheduled report.
No PDF written to the bucket	A bring-your-own bucket the role cannot write, or the wrong prefix	Confirm report_s3_bucket / report_s3_prefix and that apply granted S3 write to that bucket.

12. Operating AWSPro Day to Day

12.1 Tune thresholds in week one

The starter and per-target alarm thresholds are sensible starting points, not final values. Plan to tune them during your first week once you have seen normal volumes for your environment. A common adjustment: public-facing VPCs see far more rejected flows than internal ones, so raise the flow-reject spike threshold substantially (often into the 500–1000+ range) to avoid noise.

12.2 Alarms point at one topic by default

All alarms — including Basic Lite starter alarms — point at the default SNS topic AWSPro creates. Adding email, SMS, or HTTPS subscriptions to that topic is all that is needed to receive notifications. Premium adds per-VPC and per-zone routing to your own topics and Slack delivery if you want finer control.

12.3 Turning things off the right way

To stop an alarm or dashboard, disable it in your configuration and re-apply — do not delete it in the console. Because of drift enforcement (Section 2.2), a console deletion is restored on the next apply. The supported opt-out for the Basic Lite starter alarms, for example, is setting their enabled flag to false in configuration.

12.4 Understanding drift output

If Terraform reports that AWSPro acted even though you did not change your configuration, it corrected drift — something AWSPro-managed had been altered or removed out of band and was restored to match your desired state. This is the system working as intended. The way to make an intentional change is always to change the configuration and apply, never to edit in the console.

13. Troubleshooting

Symptom	Most likely cause & fix
A dashboard is empty	Missing or misrouted telemetry. Confirm the relevant log group is receiving events (Chapter 3) and that it is the same log group referenced in your configuration.
VPC Flow charts are empty	Flow-log format mismatch. Native and Basic Lite require the strict 18-field custom format (Section 3.4). Recreate flow logs in that format, or switch to the normalizer-backed VPC Flow path which accepts any format.
DNS Firewall widgets are empty	The VPC is missing either its DNS log group or its DNS Firewall rule group. Both are required for the firewall dashboard and alarm.
No email alerts	The subscription is unconfirmed. Click the AWS Subscription Confirmation email. Until then a warning-only drift finding will note it is pending.
No SMS alerts	Account-level SNS SMS configuration: spend limit, origination identity, or SMS sandbox (verified numbers only). Not a product issue.
Terraform acted but I changed nothing	Drift correction — an AWSPro-managed resource was altered out of band and restored. Expected. Change configuration to make intentional changes.
Platform widgets are empty	The endpoint/NAT IDs are not listed in inventory, or the resources are not active. Platform metrics are emitted by AWS automatically for active resources.
A premium dashboard will not enable	You are on Basic Lite, or the governing module is disabled. Switch to Premium and enable the module on the Deployment step.
Apply/plan fails: a module requires the "network" entitlement |	Under Premium, VpcD and ZoneD require "network" in license_features. Add it (in the Config Builder, enable the **License Feature (Network)** on the Deployment step) and re-apply. Your AWSPro license must include it; contact Codreum if it doesn't

14. Security & Data Handling

This chapter answers the question security and procurement teams ask first: what data leaves our account, and what can AWSPro do inside it? The short answer is that AWSPro runs entirely within your AWS account, your network telemetry never leaves it, and the only thing sent to Codreum is license-entitlement metadata.

14.1 Where everything runs

AWSPro is a Terraform provider that creates resources in your own AWS account — CloudWatch dashboards and alarms, an SNS alert topic, IAM roles, and (where used) Lambda functions for the VPC Flow normalizer, AI custom widgets, and license status. It reads your telemetry from CloudWatch Logs and CloudWatch Metrics in your account. Your DNS logs, flow logs, query logs, and metrics are never copied out of your account by AWSPro.

> **Your telemetry stays in your account**
> AWSPro does not capture, forward, or transmit your network telemetry anywhere. All dashboards, alarms, and AI analysis are built and run inside your AWS account and region.

14.2 What is sent to Codreum (license validation)

The single outbound call AWSPro makes to Codreum is license validation. A license-status Lambda running in your account periodically calls Codreum's license endpoint to confirm your entitlement. The request contains only the metadata needed to validate a license — no network telemetry of any kind.

Field sent	What it is
Endpoint	`https://api.codreum.com/api/license/validate`
Account ID	Your AWS account ID.
License ID	Your AWSPro license identifier.
License type	awspro or basic.
VPC IDs	The licensed VPC IDs in scope.
Zone IDs	The licensed hosted-zone IDs in scope.
Features	The entitled modules on your license (for example, network)

> **What is NOT sent**
> No DNS queries, no flow records, no query-log content, no metric values, no dashboard data. The validation call carries entitlement metadata only — the identifiers needed to confirm your license covers the VPCs and zones you are monitoring.

Basic Lite requires no license and makes no license-validation call.

14.3 AI Insights and Amazon Bedrock

AI Insights has two parts. Guided triage cards are generated deterministically from your dashboard data, in your account, with no model calls. Optional Bedrock analysis invokes an Amazon Bedrock model — and that invocation happens entirely within your account and region.

- **In-account:** Bedrock is invoked by an AWSPro custom-widget Lambda running under an IAM role in your account, using your account's Bedrock access. Codreum does not run the model and does not receive the prompt or the response.
- **In-region:** the Bedrock region defaults to your deployment region. You can override it, but by default no data crosses a region boundary.
- **Model:** the default is the in-account model global.amazon.nova-2-lite-v1:0. You may select a different Bedrock model or inference profile your account is approved for.
- **Data minimization:** raw logs are excluded from prompts by default (include_raw_logs is false); the analysis uses summarized metric and alarm context. You control what context is included.

Scheduled Reports use the same in-account model path: when a report's Bedrock narrative is enabled, its prompt and response stay inside your account and region, and only the finished digest is delivered to your own SNS topic.

> **Bedrock stays in your account and region**
> When you enable Bedrock, the model call is made by a Lambda in your account against Bedrock in your region. Your telemetry and the AI prompts/responses never reach Codreum.

14.4 Encryption & secrets

- **License ID** is treated as sensitive in Terraform (it will not print in plan/apply output).
- **SNS encryption** is optional: provide a customer-managed KMS key via sns_kms_master_key_id if your policy requires encrypted topics. Do not use the AWS-managed alias/aws/sns.
- **No long-lived Codreum credentials** are stored in your account. AWSPro uses your AWS credentials at apply time and your in-account IAM roles at run time.

14.5 The Config Builder is client-side

The Config Builder that produces your configuration runs entirely in your browser. It has no backend, no database, and no API. Configurations you import (including VPC IDs, account numbers, and license IDs) are parsed locally and never uploaded; saved drafts live in your own browser storage. Hosting the builder on a website does not change this — it serves files to your browser; it never receives what you enter.

15. IAM & Permissions

15.1 Permissions to deploy

The AWS credentials you run terraform apply with must be able to create and manage the resources AWSPro deploys. At a high level this includes:

- CloudWatch dashboards and alarms (create/update/delete).
- SNS topics and subscriptions (for the default alert topic).
- IAM roles and policies (for the AWSPro Lambdas).
- Lambda functions and their log groups (VPC Flow normalizer, AI custom widgets, license-status watcher), where those features are enabled.
- SSM parameters and a DynamoDB table used for license status (Premium).
- Read access to the CloudWatch Logs and CloudWatch Metrics that the dashboards reference.

> **Least privilege at run time**
> The Lambdas AWSPro deploys run under tightly scoped IAM roles it creates. Sensitive actions — Bedrock model invocation, license-watcher inspection — are restricted to the specific resources and account; broad read actions (such as listing Bedrock profiles or describing EC2) are limited to what the widgets need.

15.2 What AWSPro can and cannot do

- **Can:** read the telemetry you point it at and create the monitoring resources described above, in your account.
- **Cannot:** modify your VPCs, security groups, route tables, or workloads; capture packet data; or send your telemetry outside your account.

16. Cost Considerations

AWSPro itself is licensed software; the AWS resources it creates incur normal AWS charges. There are no surprises here, but it is worth knowing what drives cost so you can plan.

Resource	Cost note
CloudWatch dashboards	Charged per dashboard per month beyond the AWS free allotment. Disable dashboards you do not use.
CloudWatch alarms	Charged per alarm per month. Alarm tuning controls how many you deploy.
CloudWatch Logs	Ingestion and storage of your DNS/flow/query logs is your existing cost; set log-group retention to control it.
VPC Flow normalizer Lambda	Only deployed for the normalizer-backed VPC Flow path. Invoked per flow-log batch; the native path avoids this Lambda entirely (lower cost).
AI custom-widget Lambda + Bedrock	Only when AI/Bedrock is enabled. Bedrock is billed per token by AWS in your account; the default model is a low-cost lite model, and raw logs are excluded from prompts by default.
SNS notifications	Email/HTTPS are negligible; SMS is billed by AWS per message and region.
DynamoDB + SSM (license status)	Minimal; a small license-state table and parameter (Premium).

> **Two easy cost controls**
> Prefer the native VPC Flow path over the normalizer when you control your flow-log format — it skips the Lambda. And set sensible CloudWatch Logs retention on your telemetry log groups; AWSPro does not require any particular retention.

16.1 Understanding Bedrock cost (and why it is small by default)

The most common hesitation about AI Insights is fear of a surprise bill from Amazon Bedrock. In practice, AWSPro is configured so that Bedrock cost is small and predictable, and you stay in full control. This section explains why, with realistic estimates.

> **Guided mode is free**
> If you only use Guided mode (the default), there is no Bedrock cost at all — Guided makes no model calls. You can run AI Insights indefinitely at zero AI cost. Bedrock cost only applies when you explicitly enable Bedrock mode.

The default model

Unless you choose otherwise, AWSPro uses `global.amazon.nova-2-lite-v1:0` (Amazon Nova 2 Lite) — a fast, low-cost model in Amazon's own Nova family, invoked through a cross-Region inference profile in your account. It is far cheaper per token than frontier models, which is why it is the default for routine operational summaries. You can change it with the model_id setting if your account is approved for a different model.

How Bedrock is billed

Amazon Bedrock charges per token — small units of text — split into input tokens (the context AWSPro sends: summarized metrics, Top-N values, alarm state) and output tokens (the model's response). AWSPro is tuned to keep both small:

- **Low-cost default model:** Nova 2 Lite is priced for high-volume, low-cost use, far below large frontier models.
- **Capped output:** max_tokens defaults to 700, so each response is short.
- **Lean input:** raw logs are excluded by default; prompts carry compact summaries, not bulk log data.
- **On-demand only:** the model is invoked when an AI widget renders — not continuously. There is no always-on inference.

Realistic estimate

As a recent reference point, Amazon Nova 2 Lite on-demand pricing was about **$0.30 per million input tokens** and **$2.50 per million output tokens**. Always confirm the current figure for your Region on the AWS Bedrock pricing page — model prices change. Using that reference, a single AI widget render (roughly a few thousand input tokens plus up to 700 output tokens) costs on the order of a third of a US cent. Even with frequent viewing, a typical deployment lands in the range of a few US dollars per month — not tens or hundreds.

A Scheduled Report with the Bedrock narrative enabled adds one model call per run (a weekly report ≈ four to five calls a month; a monthly report, one); deterministic reports make no model calls.

> **Worked illustration**
>
> At about $0.30/1M input and $2.50/1M output, a render sending ~3,000 input tokens and ~700 output tokens costs roughly (3,000 × $0.30/1M) + (700 × $2.50/1M) ≈ $0.0009 + $0.00175 ≈ **$0.0027**. So **1,000 renders in a month ≈ $2.70**. This is illustrative — confirm current AWS pricing — but it shows the order of magnitude: dollars, not hundreds of dollars.

What would increase cost — and how you control it

Factor	Effect & how you control it
Choosing a larger model	Frontier models cost many times more per token than the default lite model. You choose the model via model_id — leave it at the default for lowest cost.
Raising max_tokens	Longer responses cost more. The 700 default keeps responses concise; raise it only if you need longer narratives.
Including raw logs	include_raw_logs = true enlarges every prompt. It is off by default; leave it off unless you specifically need it.
How often widgets render	Cost scales with how often AI widgets are viewed/refreshed. Infrequently viewed dashboards cost almost nothing.

How to stay in control

- Start with Guided mode (free) to see the value before enabling Bedrock.
- When you enable Bedrock, keep the default lite model, max_tokens at 700, and raw logs off — the low-cost defaults.
- Set an AWS Budget alert on Bedrock spend, or watch Bedrock cost in AWS Cost Explorer, for the first month so you see the real number for your usage — it is almost always small.
- Because everything runs in your account, you have the normal AWS cost controls (Budgets, Cost Explorer, per-service alarms) over Bedrock just like any other service.

Bottom line

With the defaults, enabling Bedrock adds single-digit-dollars-per-month cost for a typical deployment, fully visible and controllable through standard AWS cost tools. You will not be surprised by a hundred-dollar bill from leaving it on at defaults.

17. Uninstalling AWSPro

Because AWSPro is declarative Terraform, removal is clean and complete.

1. To remove everything AWSPro created, run terraform destroy in your deployment folder. This deletes the dashboards, alarms, SNS topic, IAM roles, and Lambdas AWSPro created.
2. Your telemetry is untouched: the CloudWatch log groups you created, your flow logs, resolver query logging, and hosted-zone query logging are yours and are not removed by AWSPro (it never created them).
3. To remove only part of the deployment, disable that part in your configuration and apply — for example, set enable_zoned = false to remove ZoneD while keeping VpcD.

> **Remember drift enforcement**
> Do not delete AWSPro-managed resources by hand in the console to "uninstall" — the next apply would recreate them. Use terraform destroy, or disable in configuration and apply.

18. Glossary

Term	Definition
VpcD	AWSPro's VPC-level module: DNS, DNS Firewall, VPC Flow, and platform (endpoint/NAT) observability, plus a global cross-VPC summary.
ZoneD	AWSPro's Route 53 hosted-zone module: per-zone and cross-zone query visibility.
Basic Lite	The no-license tier. Deploys Global Lite posture dashboards and starter alarms from your log groups only.
AWSPro Premium	The licensed tier. Full dashboards, per-target alarm tuning, AI Insights, and per-subject notification routing.
Prefix	Your deployment name component; AWSPro resources are named <prefix>-<region>-… Each deployment in an account/region must use a unique prefix.
NAU	Network Address Usage — a VPC metric tracking IP/ENI consumption against AWS limits; charted on the Platform dashboard.
Normalizer	A Lambda AWSPro deploys for the normalizer-backed VPC Flow path; it reads any flow-log format and republishes normalized records.
Native (VPC Flow)	The lower-cost flow path that reads strict 18-field custom-format flow logs directly, with no Lambda.
Drift enforcement	AWSPro restoring a managed resource that was changed or deleted out of band, on the next apply, to match your configuration.
Guided triage cards	Deterministic, in-account plain-language summaries of dashboard state (no model calls).
Bedrock (AI Insights)	Optional in-account, in-region Amazon Bedrock model analysis invoked by an AWSPro Lambda.

19. Support & Resources

19.1 Online resources

Resource	Link
Company website	www.codreum.com
Products	www.codreum.com/products.html
Knowledge base	www.codreum.com/knowledge.html
Documentation	www.codreum.com/docs.html
Config Builder	www.codreum.com/awspro-builder/
Open a support case	www.codreum.com/cases.html
Contact us	www.codreum.com/contact.html

19.2 Getting help

For technical issues, start with the Troubleshooting table in Chapter 13 and confirm your prerequisites in Chapter 3 — most questions trace back to telemetry that is not yet flowing. If you need assistance, open a support case at www.codreum.com/cases.html or reach the team via www.codreum.com/contact.html.