User Guide

Prism - Predictive Intelligence Platform

Version 1.0.0 | April 2026

Table of Contents

1. Introduction

1.1 What is Prism?

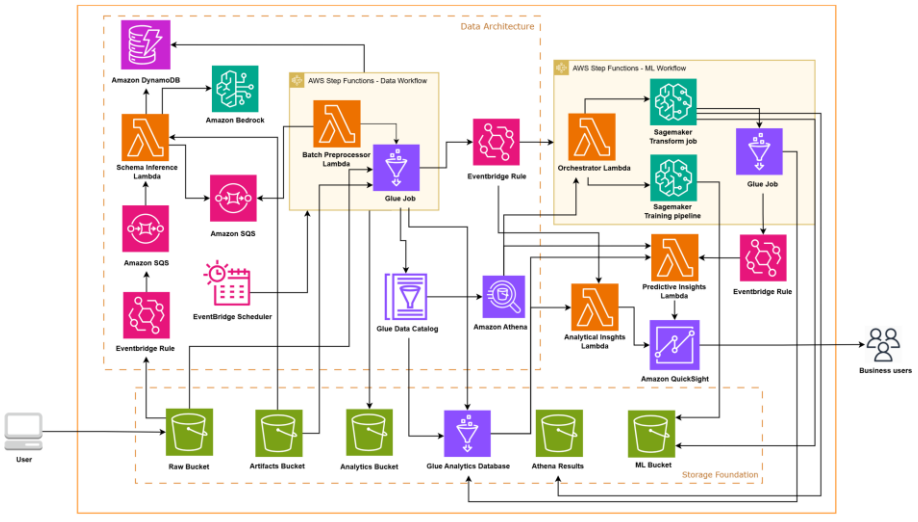
Prism (Predictive Intelligence Platform) is a customer-deployed, event-driven data Lakehouse and MLOps platform delivered as AWS CDK infrastructure. It provides end-to-end data processing, machine learning model training, and business intelligence capabilities within your own AWS account.

1.2 Key Features

- **Automated Data Processing** – Simply upload your data to S3, and the platform automatically processes it through the entire pipeline without manual intervention.
- **Smart Data Organization** – Your data is stored in a modern data lakehouse format (Apache Iceberg) that supports safe updates, tracks all changes over time, and lets you query historical data whenever needed.
- **Intelligent Schema Mapping** – The platform uses Amazon Bedrock AI to automatically understand your data structure and map it to the required format, saving you time and reducing errors.
- **Reliable Workflow Management** – All data processing runs through AWS Step Functions, which automatically handles retries and ensures your workflows complete successfully even if temporary issues occur.
- **Automated Machine Learning** – Once your data is ready, the platform automatically trains, evaluates, and deploys ML models using Amazon SageMaker—no data science expertise required.
- **Ready-to-Use Dashboards** – Beautiful QuickSight dashboards are automatically created for you, showing predictions, trends, and actionable insights from your data.

1.3 Architecture Overview

The following diagram illustrates the end-to-end architecture of Prism:



Prism: Predictive Intelligence Platform (Default VPC)

2. Churn Data Requirements

2.1 Overview

Prism Churn Prediction is specifically designed for telecom industry datasets. To ensure accurate predictions, your data must include specific customer and usage information.

2.2 Required Columns

Your telecom churn dataset must contain the following columns:

Column Name	Type	Description
year	Integer	Calendar year for the record (used for time analysis/partitioning).
month	Integer	Calendar month for the record (1–12).
customer_id	String	Unique identifier for each customer.
churn_date	Date	Date when customer churned (blank/null if not churned).
city	String	Customer city (or primary location).
plan_type	String	Customer plan type (e.g., prepaid/postpaid, bundle name).
has_phone_service	Boolean	Whether the customer has phone/voice service.
has_internet_service	Boolean	Whether the customer has internet/data service.
internet_service_type	String	Internet service technology/type (if applicable).
contract_duration_category	String	Contract duration bucket (e.g., month-to-month, 12-month, 24-month).
contract_start_date	Date	Contract start date.
contract_end_date	Date	Contract end date.
customer_join_date	Date	Date the customer joined (used for tenure calculations).
monthly_total_bill_amount	Float	Total billed amount for the month.
total_lifetime_charges	Float	Total charges accumulated over customer lifetime.
payment_method	String	Payment method used by the customer.
monthly_data_usage	Float	Monthly data usage (e.g., GB/MB based on your convention).
monthly_support_calls	Integer	Count of support calls/tickets in the month.
monthly_total_recharge_amount	Float	Total prepaid recharge amount in the month (if applicable).
churn_label	Integer	Target label (1 = churned, 0 = not churned).

2.3 Data Format Guidelines

- **File Format**: CSV (comma-separated values).

- **Encoding**: UTF-8.
- **Header Row**: First row must contain column names.
- **Missing Values**: Use empty strings or "NA" for missing values.
- **Decimal Separator**: Use period (.) for decimal numbers.
- **Column Names**: Ensure there are no repeated columns, and the column names are descriptive and semantically similar to the column names mentioned in Section 2.2.
- **Optional Columns**: Your dataset may include additional columns beyond the required ones. The platform will process them accordingly.

3. Prerequisites

3.1 AWS Account Requirements

Requirement	Details
AWS Account	Active AWS account with billing enabled.
IAM Permissions	Administrator access or equivalent permissions.
AWS Region	Region with all required services.

3.2 Required AWS Services

The following AWS services must be available and enabled in your account:

Service	Purpose	Cost Type
Amazon S3	Data storage	Pay-per-use
AWS Glue	ETL processing & data catalog	Pay-per-use
Amazon Athena	SQL query engine	Pay-per-query
Amazon SageMaker	ML model training & inference	Pay-per-use
Amazon QuickSight	Business intelligence dashboards	Subscription
Amazon Bedrock	AI schema inference	Pay-per-use
AWS Lambda	Serverless compute	Pay-per-use
Amazon EventBridge	Event routing	Pay-per-use
Amazon SQS	Message queuing	Pay-per-use
Amazon DynamoDB	Schema registry	Pay-per-use
AWS Step Functions	Workflow orchestration	Pay-per-use

Bedrock Quota Requirements

Before deploying, ensure Amazon Bedrock Nova Pro has sufficient token quota in your account:

1. Open the AWS Console and navigate to **Amazon Bedrock → Service Quotas.**
2. Search for **"Nova Pro"** and look for **“Cross-region model inference tokens per minute for Amazon Nova Pro”** in the list.
3. Check the **TPM (Tokens Per Minute)** value. It must be **2,000,000 (2M) or higher.**
4. If it shows 0 or less than 2M, click **Request quota increase** and request a minimum of **2,000,000 TPM.**
5. Quota increase requests are typically approved within a few business days.

Warning: Deploying before this quota is approved will cause the platform's AI schema analysis feature to fail at runtime.

SageMaker Instance Quota Requirements

Before deploying, ensure your account has sufficient SageMaker instance quotas for processing, training, and transform jobs:

Instance Type	Quota Name to Search	Minimum Required
ml.m5.large	"**ml.m5.large** for processing job usage"	16
ml.m5.xlarge	"**ml.m5.xlarge** for processing job usage"	16
ml.m5.2xlarge	"**ml.m5.2xlarge** for processing job usage"	8
ml.m5.4xlarge	"**ml.m5.4xlarge** for processing job usage"	4
ml.m5.large	"**ml.m5.large** for transform job usage"	16
ml.m5.xlarge	"**ml.m5.xlarge** for transform job usage"	8
ml.m5.2xlarge	"**ml.m5.2xlarge** for transform job usage"	4
ml.m5.4xlarge	"**ml.m5.4xlarge** for transform job usage"	2
ml.m5.xlarge	"**ml.m5.xlarge** for training job usage"	30
ml.m5.2xlarge	"**ml.m5.2xlarge** for training job usage"	30

`ml.m5.4xlarge`	"`ml.m5.4xlarge` for training job usage"	30

The Default Quotas are as follows:

Service	Quota	Default	Action
Lambda	Concurrent executions	1,000	Request increase for high volume
SQS	Messages in queue	Unlimited	-
S3	Buckets per account	100	Request increase if needed
Glue	Max DPUs per account	100	Request increase for parallel jobs
Bedrock	Nova Pro ("Cross-region model inference tokens per minute for Amazon Nova Pro") TPM (Tokens Per Minute)	Varies	Must be ≥ 2,000,000 TPM
QuickSight	SPICE capacity	10GB	Purchase additional capacity
Step Functions	State transitions/second	2,000	Usually sufficient

If you want to increase the quota usage:

1. Go to **Service Quotas Console.**
2. Select the AWS service.
3. Find the quota you need to increase.
4. Click **Request quota increase.**
5. Enter desired value and submit.

3.3 CDK Bootstrap Guide

CDK bootstrap creates a CloudFormation stack called **CDKToolkit** in your AWS account that provisions:

Resource	Purpose
`S3 Bucket`	Stores deployment assets (Lambda code, Glue scripts, etc.)
`ECR Repository`	Stores container images (if needed)
`IAM Roles (5)`	Allow CDK to deploy resources on your behalf

SSM Parameter	Tracks the bootstrap version

These resources are used automatically by the Prism installer during deployment.

Prerequisites

- An active AWS account with billing enabled
- A user or role with **AdministratorAccess** (or equivalent permissions for CloudFormation, S3, ECR, IAM, and SSM)
- Access to a terminal (Command Prompt, PowerShell, Terminal, or CloudShell

Option A: AWS CloudShell (Recommended)

AWS CloudShell is a browser-based terminal available directly in the AWS Management Console. The AWS CLI, Node.js, and the CDK CLI are pre-installed there is nothing to set up on your local machine.

You can run the bootstrap command from a CloudShell session in any region. The target region is specified in the command itself.

Step	Action
1	Sign in to the **AWS Management Console** with an IAM user or role that has the required permissions.
2	Click the **CloudShell** icon (**>_**) in the top navigation bar, or search for "CloudShell" in the services search bar. Wait a few seconds for the session to initialize.

3	Copy and paste the following commands into the CloudShell terminal:
	`ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)`
	`cdk bootstrap aws://$ACCOUNT_ID/us-east-1`
	The first command retrieves your 12-digit AWS account ID. The second command performs the bootstrap. This typically takes one to two minutes.

4	Run the following command to confirm the CDKToolkit stack was created successfully:
	`aws cloudformation describe-stacks \`
	`--stack-name CDKToolkit \`
	`--region us-east-1 \`
	`--query "Stacks[0].StackStatus" \`
	`--output text`
	A result of **CREATE_COMPLETE** or **UPDATE_COMPLETE** confirms the bootstrap was successful. You may now proceed with the Prism deployment.

Option B: From Your Local Machine

Use this option if you prefer to work from your own computer. This requires installing the AWS CLI, Node.js, and the CDK CLI.

Step	Action	Commands

1	Install the AWS CLI	**macOS:**
		`curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"`
		`sudo installer -pkg AWSCLIV2.pkg -target /`

		Windows: Download and run the installer from https://awscli.amazonaws.com/AWSCLIV2.msi

		Linux:
		`curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"`
		`unzip awscliv2.zip`
		`sudo ./aws/install`

2	Configure Your AWS Credentials	Run:
		`aws configure`
		When prompted, enter the following:
		`AWS Access Key ID [None]: <your-access-key>`
		`AWS Secret Access Key [None]: <your-secret-key>`
		`Default region name [None]: us-east-1`
		`Default output format [None]: json`
		To find your access keys, go to **IAM Console** > **Users** > select your user > **Security credentials** tab > **Access keys**. Click **Create access key** if you do not have one.
		Verify your credentials:
		`aws sts get-caller-identity`

		You should see a response containing your account ID, user ARN, and user ID.
3	Install Node.js	**macOS:** `brew install node@20`
		Windows: Download the LTS installer from https://nodejs.org/en/download

		Linux (Ubuntu/Debian):
		`curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -`
		`sudo apt-get install -y nodejs`

4	Install the CDK CLI	`npm install -g aws-cdk`
		Verify the installation:
		`cdk --version`

5	Run CDK Bootstrap	`ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)`
		`cdk bootstrap aws://$ACCOUNT_ID/us-east-1`
		This typically takes one to two minutes.

6	Verify the Bootstrap	`aws cloudformation describe-stacks \`
		`--stack-name CDKToolkit \`
		`--region us-east-1 \`
		`--query "Stacks[0].StackStatus" \`
		`--output text`
		A result of **CREATE_COMPLETE** or **UPDATE_COMPLETE** confirms the bootstrap was successful.

Verification Checklist

After bootstrapping, you can confirm all resources are in place by running the following commands:

What to Check	Command	Expected Result
CDKToolkit stack	`aws cloudformation describe-stacks --stack-name CDKToolkit --region us-east-1 --query "Stacks[0].StackStatus" --output text`	CREATE_COMPLETE or UPDATE_COMPLETE
Assets bucket	`aws s3 ls --region us-east-1 | grep cdk-hnb659fds-assets`	A bucket name containing your account ID
Bootstrap version	`aws ssm get-parameter --name "/cdk-bootstrap/hnb659fds/version" --region us-east-1 --query "Parameter.Value" --output text`	A version number (e.g., 22)

If all checks return the expected results, your environment is ready for Prism deployment.

Troubleshooting

Issue	Resolution
"Unable to resolve AWS account"	Your AWS credentials are not configured or have expired. Run aws configure and verify with `aws sts get-caller-identity`.
"Access Denied" during bootstrap	Your IAM identity does not have sufficient permissions. Use **AdministratorAccess** or ensure the permissions listed in the Required Permissions section are in place.
"CDKToolkit already exists"	No action is needed. Your account is already bootstrapped. The command will update the stack automatically if a newer version is available.
Bootstrap succeeded but deployment fails	Confirm you bootstrapped in **us-east-1**. CDK bootstrap is region-specific, and Prism requires the bootstrap resources in us-east-1.
"npm: command not found"	Node.js is not installed or not in your system PATH. Follow Step 3 in Option B to install it.

3.4 Before You Launch Checklist

Complete these steps before launching the deployment:

Step	Action	How to Verify
0	CDK Bootstrap setup	Detailed guide in section 3.3.

1	Activate Amazon QuickSight	QuickSight Console shows your account is active.
2	Verify Bedrock Nova Pro quota	Service Quotas → Bedrock → Nova Pro TPM: “Cross-region model inference tokens per minute for Amazon Nova Pro” ≥ 2,000,000 (see Section 3.2).
3	Verify SageMaker instance quotas	Service Quotas → SageMaker → Check processing, training, and transform job quotas (see Section 3.2).
4	Request quota increases if needed	Service Quotas Console → Submit requests for Bedrock and SageMaker.
5	(Optional) Create SageMaker Domain	Required only if you want to view ML pipelines in SageMaker AI Studio (see Section 5.6)

4. Setup Guide

4.1 AWS Marketplace Deployment

Follow these steps to deploy Prism from AWS Marketplace:

Step 1: Subscribe to Prism

1. Navigate to AWS Marketplace.
2. Search for "Prism" or "Predictive Intelligence Platform".
3. Click **View purchase options.**
4. Select your preferred **contract renewal option.**
5. Choose the **license** that matches your intended usage.
6. Review the **EULA and Offer** documents before proceeding.
7. Click **Subscribe** to complete your subscription.

Step 2: Configure the Deployment

1. Choose the latest software version.
2. Review and copy the usage instructions - you may need these when filling in the deployment parameters.
3. Click **Launch with CloudFormation.**
4. You will be redirected to the CloudFormation console.

Step 3: Load CloudFormation Template

1. Keep the default selections. **"Choose an existing template"** and **"S3 URL"** should already be filled in. Do not change these. Click **Next.**

Step 4: Configure Stack Parameters

Now you should see the **Specify stack details** page. Configure the stack parameters below. Use the usage instructions copied earlier as a reference or refer to the following parameter descriptions for guidance on each field.

Provide a Stack Name

- Recommended stack name is prism.

Compute Configuration

- **Training Dataset Size:** Select your dataset size tier first before choosing any instance types. The stack will fail if instance types are too small for your data. Options: SM for datasets under 500MB, MD for 500MB–2GB, LG for 2GB–5GB.

Tier	Data Size	Processing	Training	Inference
SM	Under 500 MB	`ml.m5.xlarge`	`ml.m5.xlarge`	`ml.m5.xlarge`
MD	500 MB - 2 GB	`ml.m5.xlarge`	`ml.m5.2xlarge`	`ml.m5.2xlarge`
LG	2 GB - 5 GB	`ml.m5.2xlarge`	`ml.m5.4xlarge`	`ml.m5.4xlarge`

Warning: The stack will fail if instances are too small for the selected tier. Choose appropriately based on your dataset size.

- **Max Glue ETL Workers:** Number of parallel workers for the ETL job. Default is 5. Increase up to 15 if you have a larger dataset (2-5 GBs) and want faster processing.

Schedule & Model Quality

- **Pipeline Schedule:** How frequently the pipeline runs automatically to ingest new data. Default rate is 1 hour. Adjust based on how often your data updates.
- **Minimum AUC Threshold:** The minimum model quality score (AUC) required before a newly trained model is deployed. Default is 0.65. A higher value enforces stricter model quality but may prevent deployment if data quality is low.

Environment Configuration

- **QuickSight Service Role Name:** The IAM role QuickSight uses to access your data. The default value aws-quicksight-service-role-v0 is created automatically when you first activate QuickSight. Only change this if your account uses a custom QuickSight role.
- **QuickSight Username:** Your QuickSight user identifier. To find it, open QuickSight, click the profile icon in the top right, and select **Manage QuickSight**. Can be either your username or the email address associated with your QuickSight account.
- **Admin Email for SNS Notifications:** Email address that will receive operational alerts and pipeline notifications. Replace the placeholder with a valid email address. You will receive a confirmation email from AWS SNS after deployment. You must confirm the subscription to receive alerts.

Installer EC2 Configuration

- **AMI ID:** Pre-configured by 10Pearls. Do not change.
- **Installer EC2 Instance Type:** Pre-configured by 10Pearls. Do not change.
- **Installer VPC CIDR Block:** CIDR block for the temporary installer **VPC**. Default is 10.0.0.0/16. No changes are required.

Assets Configuration

- **Seller Assets S3 Bucket Name:** Pre-configured by 10Pearls. Do not change.
- **Customer Assets S3 Bucket Name:** Pre-configured by 10Pearls. Do not change.

Step 5: Review and Create

1. Review all parameters.
2. Acknowledge IAM resource creation by checking the required boxes.
3. Click **Create stack.**

Step 6: Monitor Deployment

1. The stack will show **CREATE_IN_PROGRESS** until the installer signals completion (20-40 minutes).
2. To monitor progress in real-time:
    - Go to **CloudFormation Console** → **Outputs** tab.
    - Find the **InstallerInstanceId.**
    - Open **CloudShell** and connect to the instance via SSM:

```
aws ssm start-session --target <instance-id> --region us-east-1
```

    - Stream logs with:

```
sudo journalctl -u prism-deployer.service -f
```

3. The installer instance **self-terminates on success.**
4. If the instance remains running after 60 minutes, there may be an error - check the deployment logs.
5. Once complete, the stack status will show **CREATE_COMPLETE.**
6. Check the **Outputs** tab for important resource information (bucket names, dashboard URLs, etc.).

Step 7: Upload Your Data

1. Navigate to S3 in the AWS Console.
2. Find the raw data bucket: prism-raw-{account}-{region}.
3. Navigate to the churn-data/ folder.
4. Upload your telecom churn CSV file.

5. Note that the pipeline will trigger automatically based on the schedule you selected during. deployment. For example, if you chose rate (1 hour), processing will not begin until the first scheduled interval (1 hour) has passed. To trigger the pipeline immediately, navigate to **Step Functions → State Machines → prism-processing-workflow → Start execution.**

Warning: Do not start multiple executions in parallel. Wait for the first execution to complete before executing the state machine again.

Step 8: View Results

1. Once processing completes, open QuickSight.
2. Navigate to **Dashboards** to view your personalized dashboard.

4.2 VPC Deployment Options

Prism supports three VPC deployment modes:

USE_DEFAULT (Simplest Option)

Uses the **default VPC** in your AWS account.

Behaviour:

- Deploys resources into the AWS default VPC.
- Uses public subnets.
- No VPC endpoints are created.

When to use:

- Quickest setup with minimal configuration.
- Lowest complexity.
- Recommended if you are unfamiliar with VPC networking.

To deploy with default VPC:

1. During CloudFormation stack creation, set **VpcDeploymentMode** to USE_DEFAULT.
2. No additional VPC configuration is required.

Pros: Simplest setup, lowest cost, no VPC expertise required.
Cons: Less network isolation uses public subnets.

CREATE_NEW (Recommended for Production)

Creates a new VPC with private subnets and VPC endpoints for AWS services. This provides better isolation and security.

Required parameters:

- **`VpcCidr:`** CIDR block for the new VPC (example: 10.0.0.0/16).
- **`MaxAzs:`** Number of Availability Zones (keep in us-east-1).
- **`NatGateways:`** Number of NAT Gateways (0 or 1 allowed).

Behavior:

- A new VPC with public and private subnets is created.
- If `NatGateways = 1`, NAT Gateway is created.
- If `NatGateways = 0`, required **interface endpoints** are created for the following AWS services:
  - `Amazon SQS.`
  - `Amazon Bedrock Runtime.`
  - `Amazon SageMaker Runtime.`
  - `Amazon SageMaker API.`
  - `AWS Step Functions.`
  - `AWS Glue.`
  - `Amazon CloudWatch Logs.`
  - `Amazon EventBridge.`
  - `Amazon Athena.`

To deploy with a new VPC:

1. Set **VpcDeploymentMode** to CREATE_NEW.
2. Configure the following parameters:
   - **`VpcCidr:`** CIDR block for the new VPC (default: 10.0.0.0/16).
   - **`NatGateways:`** Set to 0 to use VPC Endpoints (recommended for cost savings), or 1 for NAT Gateway.
   - **`MaxAzs:`** Maximum availability zones (default: 1).

Recommendation: If you choose **CREATE_NEW**, keep **NatGateways = 0** unless you specifically require NAT Gateways. This avoids unnecessary networking costs.

Pros: Better security, private subnets, VPC endpoints.
Cons: Slightly higher cost if using NAT Gateway.

USE_EXISTING

Uses an existing VPC in your account. Ideal when you have specific networking requirements or compliance needs.

Prerequisites for USE_EXISTING:

Before selecting this mode, ensure your VPC meets the following requirements:

1. **DNS Settings must be enabled:** These are required for internal service communication.
   - Enable **DNS resolution.**
   - Enable **DNS hostnames.**
2. **Private Subnets:** You must have private subnets available in the VPC.
3. **Route Tables:** Route tables must be properly configured for your subnets.
4. **Internet Access:** If you want to use an existing NAT Gateway, set **NatGateways** to 1 and ensure it is configured for your subnets.

To deploy with an existing VPC:

1. Set **VpcDeploymentMode** to USE_EXISTING.
2. Provide the following required parameters:

- **VpcId**: Your existing VPC ID (e.g., vpc-0a1b2c3d).
- **AvailabilityZones**: Comma-separated list of availability zones (e.g., us-east-1a,us-east-1b).
- **PrivateSubnetIds**: Comma-separated list of private subnet IDs (e.g., subnet-0a1b2c3d,subnet-0e5f6a7b).
- **RouteTableIds**: Comma-separated list of route table IDs.

3. Set **NatGateways** to 1 if you want to use your own NAT Gateway instead of VPC endpoints.

Required Security Group Configuration:

If you provide existing security groups for **SageMaker**, **Glue**, or **Lambda**, they must allow **self-referencing traffic** so jobs can communicate within the service.

Example configuration:

```
Allow inbound traffic:
Source: same security group
Protocol: All traffic
```

Equivalent behavior to:

```
SecurityGroupIngress:
  Source: self
  Port: all
  Protocol: all
```

This allows distributed processing jobs (such as SageMaker training and Glue processing) to communicate internally.

Using Existing Security Groups:

If you enable existing security groups for any of the following services:

Service	Parameter
Glue	use_existing_security_groups_glue = true
Lambda	use_existing_security_groups_lambda = true
SageMaker	use_existing_security_groups_sagemaker = true

You must also provide the corresponding security group IDs (e.g., sg-0a1b2c3d).

If this option is **false**, the platform will automatically create and manage security groups for you.

NAT Gateway Behavior:

- **NatGateways = 0**
  - The platform creates required **VPC endpoints** for AWS services.
  - This is the **lowest cost configuration.**
- **NatGateways = 1**
  - The deployment assumes a **NAT Gateway already exists.**
  - The NAT Gateway must be correctly attached to the provided VPC and route tables.

- The platform will route outbound traffic through this NAT Gateway instead of creating endpoints.

Pros: Use your existing network infrastructure, comply with organization policies.
Cons: Requires more configuration and VPC expertise.

VPC Recommendation

If you are **not familiar with VPC networking**, the simplest options are:

- **USE_DEFAULT** – No configuration needed.
- **CREATE_NEW** – Platform handles VPC creation.

In both cases, set:

```
nat_gateways = 0
```

This enables **VPC endpoints instead of NAT Gateways**, which significantly reduces networking costs.

This configuration works out of the box and prevents unnecessary networking costs.

4.3 Post-Deployment Verification

After deployment completes, verify the following:

Check S3 Buckets:

1. Go to **S3 Console**
2. Verify these buckets exist:
   - prism-raw-{account}-{region} – for raw data uploads (upload to churn-data/ folder).
   - prism-analytics-{account}-{region} – for processed data.
   - prism-ml-{account}-{region} – for ML artifacts.

Check Glue Resources:

1. Go to **AWS Glue Console.**
2. Verify the database prism_db exists.
3. Check that Glue jobs are created.

Check Step Functions:

1. Go to **Step Functions Console.**
2. Verify the state machine is created.
3. Check recent executions (if any).

Check QuickSight:

1. Open **QuickSight Console.**
2. Navigate to **Datasets** and verify datasets are created.
3. Navigate to **Dashboards** and verify dashboard is available.

5. Data Ingestion

5.1 Understanding the Two-Upload Workflow

The platform is designed around a **two-upload workflow**:

Upload	Purpose	What Happens
First Upload	Historical data for training	Full historical dataset used to train the churn prediction model
Second Upload	Inference data	Data you want predictions on - triggers batch inference

5.2 Upload Location

After deployment, upload your CSV files to the churn-data/ folder in the raw bucket:

```
s3://prism-raw-{account}-{region}/churn-data/
```

For example: s3://prism-raw-123456789012-us-east-1/churn-data/

The platform automatically:

- Detects uploads.
- Infers the schema using Amazon Bedrock.
- Starts the pipeline based on the configured schedule.

Note: If the schedule is set to 1 hour (default), the pipeline runs in an hour. See Section 5.5 for how to trigger it sooner.

5.3 First Upload: Training Data

Upload your **full historical dataset** for training. This should contain all columns including the churn target variable.

Processing Timeline:

Stage	Duration	Description
Schema Inference	~1 minute	Amazon Bedrock analyses your data structure
Glue ETL	~2-5 minutes	Transforms data to Iceberg format (varies by size: 1MB-2GB typical)
SageMaker Training Pipeline	~20-50 minutes	Model training, evaluation, and deployment

What to Expect:

1. Once Gluejob completes, your **historical analytics QuickSight dashboard** will be available within 1-5 minutes.
2. Monitor training progress in **Sagemaker AI Console** → **SageMaker Studio** → **Pipelines** (a domain is required to access SageMaker Studio, refer to section 5.6).
3. Wait for the SageMaker training pipeline to complete successfully before uploading inference data.

5.4 Second Upload: Inference Data

After training completes, upload the data you want predictions on. This data goes through the same ingestion process.

Processing Timeline:

Stage	Duration	Description
Schema Inference	~1 minute	Validates data structure
Glue ETL	~2-5 minutes	Processes inference data
SageMaker Batch Transform	Varies	Generates predictions (varies by dataset size)
Dashboard Update	~1-5 minutes	QuickSight dashboard updated to include predictions sheet

What to Expect:

1. Once Glue completes, it automatically triggers a SageMaker Batch Transform job.
2. Monitor progress in **SageMaker AI Console** → **Batch Transform jobs.**
3. When inference finishes, a second Glue job loads the predictions.
4. Your **QuickSight prediction insights dashboard** is automatically updated.

5.5 Triggering the Pipeline Manually

If you prefer not to wait for the next scheduled run, you can trigger the pipeline immediately by starting the workflow directly. This method is more straightforward and avoids the need to modify complex scheduling rules:

Step	Action
1	Open the AWS Step Functions Console.
2	**Navigate to "State machines"** in the left-hand menu**.**

3	**Search for the pipeline** (typically named with a `prism-processing-` prefix).
4	**Select the state machine** and click the **Start execution** button.
5	**Confirm the execution:** You can optionally provide a unique name for the run, or simply click **Start execution** again to use the default settings.
6	**Monitor Progress:** The console will display a visual graph of the workflow. The run is complete when the status reaches **Succeeded**.

5.6 Monitoring Ingestion

Check Step Functions Status:

The platform uses two Step Functions state machines:

Step	Action
1	Go to **Step Functions Console.**
2	Find and monitor the following state machines: ● prism-processing-workflow – Main orchestration workflow for data processing. ● prism-ml-workflow – ML training and inference workflow.
3	View the latest execution.
4	Check execution status and details.

Check Glue Job Status:

The platform uses multiple Glue jobs:

Step	Action
1	Go to **AWS Glue Console.**
2	Navigate to **ETL Jobs.**
3	Monitor the following jobs: ● prism-etl-churn-job – Main ETL job for processing churn data. ● prism-churn-ingest-predictions-job – Job for loading inference results.
4	View job run history and status.

Check SageMaker Pipeline Status:

Step	Action
1	Go to **SageMaker AI Console.**
2	Open **SageMaker Studio** (a domain is required to access SageMaker Studio, refer to the section below)
3	Navigate to **Pipelines.**
4	Select your pipeline (t will have a “prism” prefix).
5	View execution status and logs.

Creating a SageMaker Domain (Optional):

To view ML pipelines with a visual interface in SageMaker AI Studio:

Step	Action
1	Go to **SageMaker AI Console.**
2	Click **Domains** in the left navigation.
3	Click **Create domain.**
4	Choose **Quick setup** for the fastest configuration.
6	Click **Submit.**
7	Wait for domain creation to complete (5-10 minutes).
8	Once created, click **Open Studio** to access the visual pipeline interface.

Note: Creating a SageMaker Domain is optional and not required for the platform to function. It only provides a visual interface for monitoring ML pipelines.

Check Batch Transform Status:

Step	Action
1	Go to **SageMaker AI Console.**
2	Navigate to **Batch Transform jobs.**
3	View job status and completion time.

Check SageMaker Training and Processing Jobs:

Step	Action
1	Go to **SageMaker AI Console.**
2	Navigate to **Training → Training jobs** to view model training status.
3	Navigate to **Processing → Processing jobs** to view data processing status.

5.7 Data Validation

The platform performs automatic validation including:

- Column presence verification.
- Data type checking.
- Value range validation.
- Duplicate detection.

If validation fails, check CloudWatch logs for detailed error messages.

Tip: Review Section 2 (Churn Data Requirements) before uploading to ensure your data has all required columns in the correct format.

6. Workflows & Use Cases

6.1 Churn Prediction and Analytics Workflow

The churn prediction and analytics workflow processes telecom customer data to generate insights and predict which customers are likely to churn.

Workflow Steps:

Step	Action
1	**Data Upload** – You upload customer data to the S3 raw bucket.
2	**Event Trigger** – S3 upload triggers EventBridge event.
3	**Schema Inference** – Amazon Bedrock analyses and maps your data schema.
4	**Data Processing** – AWS Glue transforms data into Iceberg format.
5	**Feature Engineering** – Relevant features are extracted for ML.
6	**Model Training** – SageMaker trains the churn prediction model.
7	**Batch Inference** – Model generates predictions for all customers upon data upload.
8	**Dashboard Update** – QuickSight displays predictions and insights.

Note: Results are not automatically generated for all historical records, rather part of the next ingestion lifecycle on new data upload.

6.2 Viewing Predictions

After processing completes:

Step	Action
1	Open **QuickSight Console.**
2	Navigate to **Dashboards.**
3	Select the **Prism Customer Churn Dashboard.**
4	View the 3 sheets: • Churn Overview and risk. • Revenue and Segments • Prediction Analysis (predictions are here)

6.3 Querying Data with Athena

You can run custom SQL queries against your data:

Step	Action
1	Go to **Amazon Athena Console.**
2	Select the workgroup **prism-workgroup.**
3	Select database **prism_analytics_db**.
4	Run queries against available tables: • churn_historical • churn_predictions

7. QuickSight Dashboards

7.1 Accessing the Dashboard

Step	Action
1	Navigate to **QuickSight Console.**
2	Click **Dashboards** in the left navigation.
3	Select the **Prism Customer Churn Dashboard.**

The dashboard contains multiple sheets mentioned in **Section 6.2**.

7.2 Dashboard Components

The churn prediction dashboard includes:

Churn Risk Overview

- Total customers analyzed.
- Customers at high risk (>70% probability).
- Customers at medium risk (40-70% probability).
- Customers at low risk (<40% probability).

Risk Distribution Chart

- Visual breakdown of churn risk categories.
- Trend over time (if historical data available).

Top Risk Factors

- Key features contributing to churn.
- Feature importance visualization.

Customer List

- Sortable table of all customers.
- Risk score for each customer.
- Filter by risk level, contract type, etc.

7.3 Creating Custom Visuals in Dashboard

While the default visuals on the Dashboard are generated automatically for required columns, you can also create your own custom visuals directly from the QuickSight interface, and it's simple to do.

To create a custom visual:

Step	Action
1	Open the **Analyses** tab from the QuickSight side panel and choose '**Prism Customer Churn Analysis**'.
2	Click **Add Visual** from the **Insert** tab and choose a chart type from the visuals panel onto the sheet.
3	Select your desired dimensions and measures from the available fields.
4	Customize the chart type, colours, and labels as needed.

5	Click **Done editing** to save your visual to the sheet.
6	Publish the Analysis as Dashboard.

This gives you full flexibility to explore the data beyond the default views, whether you want to compare additional columns, try different chart types, or build visuals tailored to your specific analysis needs.

7.4 Sharing Dashboards

Share with specific users:

Step	Action
1	Open the dashboard.
2	Click **Share** icon in the top right.
3	Enter email addresses of QuickSight users.
4	Set permission level (Viewer or Co-owner).
5	Click **Share.**

Publish to larger audience:

Step	Action
1	Click **Share** → **Publish dashboard.**
2	Configure sharing settings.
3	Copy the dashboard link to distribute.

7.5 Natural Language Querying with QuickSight Q Topics

Prism automatically creates a QuickSight Q Topic as part of the analytics pipeline. A Q Topic enables natural language querying instead of navigating dashboard filters, you can ask questions about your churn data in plain English directly within the QuickSight console.

For example, you can type questions like:

- "How many customers churned last month?"
- "What is the average monthly bill for churned customers?"
- "Show me customers with fiber optic internet who are on month-to-month contracts"
- "What is the total lifetime value of customers by payment method?"

To Access Q topics follow the steps:

Step	Action
1	Open the QuickSight Console.
2	Click the Q search bar at the top of the page (or navigate to Q Topics from the left navigation).
3	Select the Customer Churn Analysis Topic.
4	Type a question in plain English and press Enter.
5	QuickSight Q returns an answer as a visual (chart, table, or number) that you can pin to a dashboard.

8. Monitoring

8.1 CloudWatch Monitoring

View Processing Metrics:

1. Go to **CloudWatch Console.**
2. Navigate to **Metrics.**
3. Find metrics under:
    - `AWS/Lambda.`
    - `AWS/Glue.`
    - `AWS/States.`

8.2 Logs

Lambda Logs:

Step	Action
1	Go to **CloudWatch Console.**
2	Navigate to **Logs** → **Log groups.**
3	Find log groups starting with /aws/lambda/prism-.

Glue Job Logs:

Step	Action

1	Go to **AWS Glue** console**.**
2	Click ETL Jobs.
3	Find the glue jobs “prism-etl-churn-job" or “prism-churn-ingest-predictions-job”
4	Click on **runs -> logs** to see execution logs.

Step Functions Logs:

Step	Action
1	Go to **Step Functions Console.**
2	Select the state machine (**prism-processing-workflow** or **prism-ml-workflow**).
3	Click on an execution to view details and logs.

SageMaker Logs:

You can view SageMaker logs in three ways:

Via SageMaker Pipeline:

Step	Action
1	Go to **SageMaker AI Console.**
2	Open **SageMaker Studio** (a domain is required to access SageMaker Studio, refer to section 5.6)
3	Navigate to **Pipelines.**
4	Select your pipeline execution (with the prefix **prism**).
5	Click on any step to view its logs.

Via Individual Jobs:

Step	Action
1	Go to **SageMaker AI Console.**
2	Navigate to **Training → Training jobs** or **Processing → Processing jobs.**
3	Select the job.
4	Click **View logs** to open CloudWatch logs.

Via Cloudwatch Log group:

Job Type	Cloudwatch Log Group Path
Processing Jobs	/aws/sagemaker/ProcessingJobs
Training Jobs	/aws/sagemaker/TrainingJobs
Batch Transform	/aws/sagemaker/TransformJobs

9. Security Configuration

9.1 Encryption

S3 Encryption:

- All S3 buckets use Server-Side Encryption (SSE-S3) by default.
- You can upgrade to SSE-KMS for additional control.

DynamoDB Encryption:

- Tables are encrypted at rest using AWS managed keys.

In Transit:

- All data transfer uses TLS 1.2+.

9.2 IAM Security

Least Privilege:

- All IAM roles follow the principle of least privilege.
- Roles are scoped to specific resources where possible.

Lambda Function Isolation:

- Each Lambda function has its own dedicated IAM execution role.
- Roles are individually scoped to only the permissions that specific function requires.
- Note: The QuickSight role is shared between inference and analytics functions for dashboard operations.

Service Roles:

- Glue jobs use a dedicated service role with ETL-specific permissions.
- SageMaker uses a specific ML role for training and inference.
- Each Step Functions state machine has its own dedicated orchestration role.

9.3 Network Security

VPC Security (when using CREATE_NEW or USE_EXISTING):

- Private subnets for compute resources.
- VPC endpoints for AWS service access.
- Security groups restrict traffic.

S3 Access:

- Bucket policies restrict access to authorized principals.
- Block public access is enabled.

9.4 Access Control

QuickSight Access:

- Users must be invited to view dashboards.
- Row-level security can restrict data access.

Athena Access:

- Workgroup permissions control query access.
- IAM policies restrict database/table access.

10. Cost Management

10.1 Cost Components

The following are some of the major cost components. This is not an exhaustive list:

Service	Cost Driver	Typical Range
S3	Storage + requests	$0.023/GB/month
Glue	DPU-hours	$0.44/DPU-hour
Athena	Data scanned	$5/TB scanned
SageMaker	Instance hours	Varies by instance
Lambda	Invocations + duration	$0.20/million requests
QuickSight	User licenses	$9-18/user/month
Step Functions	State transitions	$0.025/1000 transitions
VPC	NAT Gateway + data transfer	$0.045/hour + data costs
VPC Endpoints	Per endpoint per AZ	$0.01/hour per endpoint
EC2 (Installer)	Instance hours	Temporary, terminates after deployment
DynamoDB	Read/write capacity	Pay-per-request

Note: Using `NatGateways = 0` with VPC Endpoints is recommended to reduce networking costs.

10.2 Cost Monitoring

To ensure you are only monitoring costs associated with this application, use the standardized resource tags during budget creation:

1. **Open the Billing Console:** Navigate to the AWS Budgets dashboard.
2. **Create Budget:** Click **Create budget**, select **Cost budget**, and click **Next**.
3. **Define Budget Amount:** Enter a name (e.g., `Prism_Monthly_Budget`) and set your desired period and amount.
4. Filter by Prism Tag (Critical Step).
5. Define your notification triggers (e.g., notify when actual costs exceed 80% of the budget) and add your email address.

Use Cost Explorer:

1. Go to **Cost Explorer.**
2. Filter by service or tag to see costs for S3, Glue, SageMaker, QuickSight, etc.
3. Analyse costs over time.

11. Troubleshooting

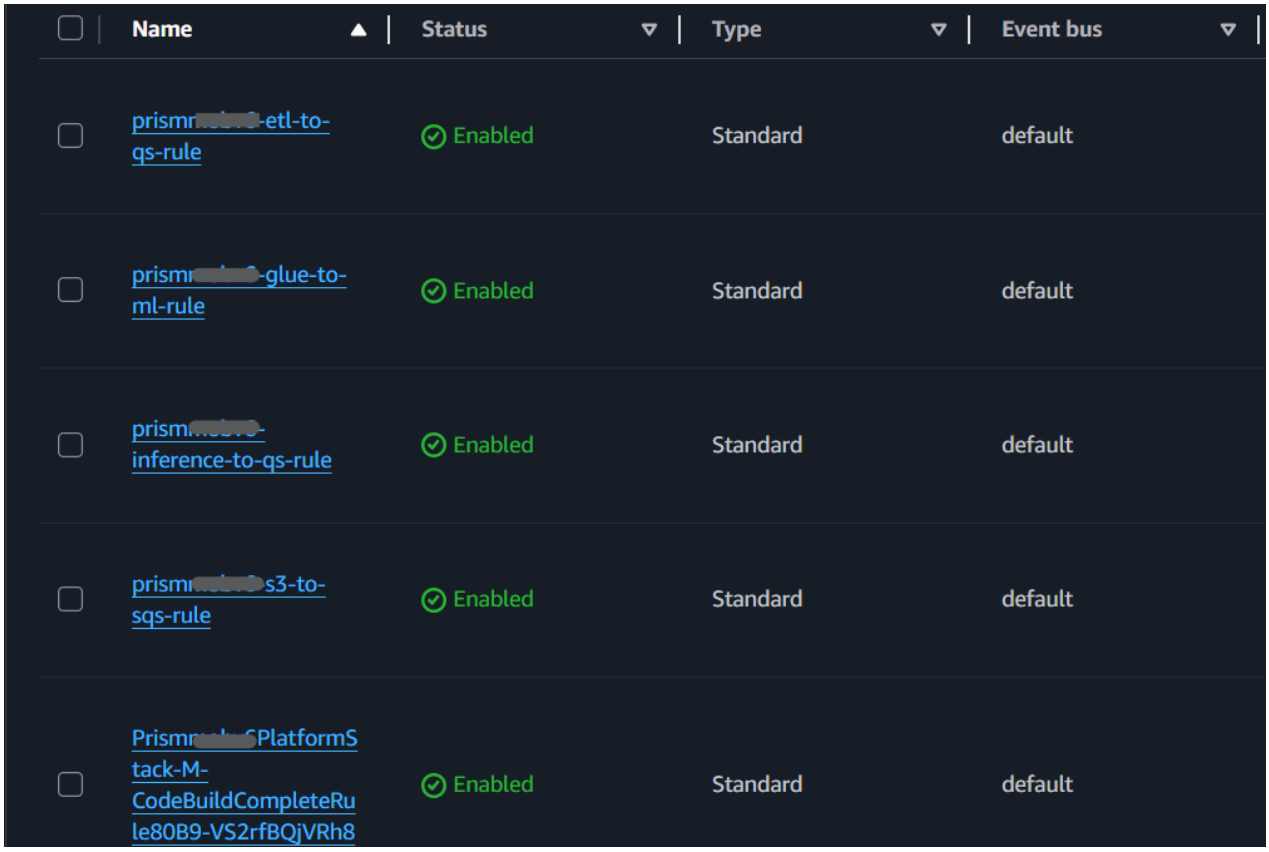
11.1 Common Issues

Data Not Processing

Symptoms: Uploaded file not processed, no Step Functions execution.

Solutions:

1. Check if EventBridge rules are enabled.
    - Go to **`EventBridge Console.`**
    - Verify all 5 “prism” prefix rules are in ENABLED state.



Note: The prefix may change depending on the stack name that is configured during installation.

2. Check Lambda invocation logs.

- Go to **CloudWatch Console → Logs.**
- Check /aws/lambda/prism-schema-reader for checking the errors. Some common errors could be:
  - your data failed to fulfill the schema requirements
  - your license expired
  - model unavailable: rate limit exceeded

Glue Job Failures

Symptoms: Glue job timed out. Glue job failed.

Solutions:

1. Check Glue job error logs in CloudWatch.
2. Verify input data format matches expected datatypes specified in the schema .

QuickSight Dashboard Empty

Symptoms: Dashboard shows no data or loading forever.

Solutions:

1. Verify Athena can query the data tables.

2. Check QuickSight SPICE dataset status.
3. Refresh the dataset manually:
    - Go to **QuickSight** → **Datasets**
    - Select dataset → **Refresh now**
4. Verify QuickSight role can access S3.

SageMaker Training Failures

Symptoms: ML model training fails.

Solutions:

1. Check SageMaker training job logs.
2. Verify sufficient instance quota.
3. Check training data volume is adequate.
4. Review hyperparameters for validity.

11.2 Getting Support

If issues persist:

1. Gather relevant logs and error messages.
2. Note the timestamps when the issue occurred.
3. Contact support (see Section 15).

12. Upgrades & Updates

12.1 Update Process

Before updating:

Review release notes for breaking changes.

Update via AWS Marketplace:

1. Go to **AWS Marketplace Subscriptions.**
2. Find your Prism subscription.
3. Click **Update Stack** when a new version is available.
4. Review parameter changes.
5. Execute stack update.

12.2 Rollback Procedure

If an update causes issues:

1. Go to **CloudFormation Console.**
2. Select the **Prism installer** stack.
3. Click **Stack actions** → **Roll back.**
4. Wait for rollback to complete.
5. Verify system functionality.

13. Backup & Recovery

13.1 Data Backup

S3 Versioning:

All S3 buckets have versioning enabled by default. To recover deleted files:

1. Go to **S3 Console.**
2. Navigate to the bucket.
3. Click **Show versions** toggle.
4. Find the version you want to restore.
5. Download or copy the previous version.

DynamoDB Point-in-Time Recovery:

Schema registry supports point-in-time recovery:

1. Go to **DynamoDB Console.**
2. Select the schema registry table.
3. Click **Backups** tab.
4. Use **Restore to point-in-time** to restore.

13.2 Iceberg Time Travel

Apache Iceberg supports time travel queries, allowing you to access historical data. You can query data as it existed at a specific point in time using Athena:

```
-- Query data as of a specific snapshot
SELECT * FROM prism_analytics_db.churn_historical
FOR VERSION AS OF <snapshot-id>
```

To find available snapshots along with timestamps:

```
SELECT snapshot_id, committed_at
FROM "prism_analytics_db"."churn_historical$snapshots"
ORDER BY committed_at DESC
```

14. Uninstallation

14.1 Before Uninstalling

Critical: Review what data you want to retain before uninstalling.

1. Save QuickSight dashboards (export as PDF or republish to another account).
2. Cancel AWS Marketplace subscription to stop billing.

Note: The raw S3 bucket has a retention policy and will not be deleted automatically, so your uploaded data will be preserved.

14.2 Delete CloudFormation Stacks

You need to delete two CloudFormation stacks:

Delete the Installer Stack:

1. Go to **CloudFormation Console.**
2. Find and select the installer stack (the name you chose during deployment).
3. Click **Delete Stack.**
4. Confirm deletion.
5. Wait for deletion to complete.

Delete the PrismPlatform Stack:

1. Go to **CloudFormation Console.**
2. Select the "`PrismPlatformStack`" stack (created by the installer).
3. Click **Delete Stack.**
4. Confirm deletion.
5. Wait for deletion to complete (monitor status).

Note: **PrismPlatformStack** is the parent stack and initiating its deletion will automatically trigger the removal of all associated nested stacks.

14.3 Verify Complete Removal

1. Go to **Resource Groups Console.**
2. Search for resources with names starting with prism-*.
3. Verify no resources remain.

Note: Raw S3 Bucket created by the stack will need to be manually removed by the user.

15. Support

15.1 Support Channels

Channel	Use Case	Response Time
Documentation	Self-service help	Immediate
AWS Marketplace Support	Subscription issues	24-48 hours

15.2 Contacting Support

10Pearls Support:

- Email: aws-platform-support@10pearls.com
- Website: https://10pearls.com

AWS Marketplace Support:

- Via AWS Console: Support Centre.

15.3 Information to Include

When contacting aws support, provide:

1. AWS Account ID (not credentials).
2. Error messages and timestamps.
3. Steps to reproduce the issue.
4. CloudWatch logs (relevant excerpts).
5. Screenshots of the issue (if applicable).

15.4 Self-Service Resources

Resource	Location
This User Guide	Included with product
EULA	EULA

AWS Marketplace Listing	Per subscription
AWS Documentation	docs.aws.amazon.com

Release Notes

Version 1.0.0 (April 2026)

Initial Release

- Telecom Churn Prediction workflow.
- Apache Iceberg data Lakehouse.
- AI-powered schema inference (Amazon Bedrock).
- SageMaker ML pipelines.
- QuickSight programmatic dashboards.
- Event-driven processing architecture.
- VPC deployment options (default, new, existing).

Appendix A: Prism System Configurations

This section consolidates all configuration options for Prism.

A.1 Core Configuration

Parameter	Type	Default	Description
QuickSightUsername	String	*required*	QuickSight user email for dashboard ownership
QuickSightRoleName	String	aws-quicksight-service-role-v0	QuickSight IAM service role
AdminEmail	String	*required*	Email address for SNS operational alerts

A.2 Lambda Configuration

Setting	Value	Description
Runtime	Python 3.11	Lambda runtime environment
Architecture	x86_64	Processor architecture
Memory	256-512 MB	Memory allocation (varies by function)
Timeout	30-900 seconds	Function timeout (varies by function)
Tracing	Active	X-Ray tracing enabled

VPC	Configured	Runs in VPC when using CREATE_NEW or USE_EXISTING modes

A.3 Training Dataset Size Tiers

Select the appropriate tier based on your dataset size. This automatically configures the correct instance types:

Tier	Data Size	Processing Instance	Training Instance	Inference Instance
SM (Small)	Under 500 MB	ml.m5.xlarge	ml.m5.xlarge	ml.m5.xlarge
MD (Medium)	500 MB - 2 GB	ml.m5.xlarge	ml.m5.2xlarge	ml.m5.2xlarge
LG (Large)	2 GB - 5 GB	ml.m5.2xlarge	ml.m5.4xlarge	ml.m5.4xlarge

Important: Selecting a tier that's too small for your data will cause the deployment to fail. When in doubt, choose the next tier up.

A.4 Compute Configuration

Parameter	Type	Default	Description
GlueWorkerType	String	G.1X	Glue worker size (G.1X, G.2X, G.4X, G.8X)
GlueMaxWorkers	Number	5	Maximum Glue workers for auto-scaling
InferenceInstanceType	String	ml.m5.large	SageMaker batch inference instance (auto-set by tier)
TrainingInstanceType	String	ml.m5.xlarge	SageMaker training instance (auto-set by tier)

A.5 Scheduling Configuration

Parameter	Type	Default	Description
ScheduleExpression	String	rate(1 hour)	EventBridge schedule expression for batch pipeline
SchedulerEnabled	String	true	Enable/disable scheduled processing

A.6 Model Quality Configuration

Parameter	Type	Default	Description
MinimumAUC	Number	0.65	Minimum AUC threshold for model deployment (range: 0.50-0.80). The model only deploys if it meets this score on the test set.

A.7 VPC Configuration

Parameter	Type	Default	Description
VpcDeploymentMode	String	USE_DEFAULT	CREATE_NEW, USE_EXISTING, or USE_DEFAULT
VpcCidr	String	10.0.0.0/16	VPC CIDR block (for CREATE_NEW)
NatGateways	Number	0	Number of NAT gateways. Set to 0 to use VPC Endpoints (lower cost). Set to 1 for NAT Gateway.
MaxAzs	Number	1	Maximum availability zones (for CREATE_NEW, keep in us-east-1)
VpcId	String	-	Required for USE_EXISTING mode
AvailabilityZones	String	-	Required for USE_EXISTING mode (comma-separated list)
PrivateSubnetIds	String	-	Required for USE_EXISTING mode (comma-separated list)
RouteTableIds	String	-	Required for USE_EXISTING mode (comma-separated list)

A.8 S3 Configuration

Setting	Default	Description
Versioning	Enabled	All buckets have versioning enabled for data protection
Encryption	SSE-S3	Server-side encryption enabled by default
Lifecycle	Configured	Older versions are transitioned to cheaper storage classes

A.9 Glue Configuration

Setting	Value	Description
Job Name	`prism-etl-churn-job`	ETL job for processing churn analytics data
Type	`Spark`	Spark-based ETL job
Glue Version	`5.0`	Supports Spark 3.5, Scala 2, Python 3
Language	`Python 3`	Python runtime
Worker Type	`G.1X`	4 vCPU and 16GB RAM per worker
Auto-scaling	`Enabled`	Dynamically scales workers up and down
Maximum Workers	`5`	Maximum number of workers when auto-scaling
Job Timeout	`30 minutes`	Maximum job runtime
Number of Retries	`1`	Automatic retry on failure
Maximum Concurrency	`1`	Only one job run at a time
Job Bookmark	`Disabled`	Does not track previously processed data
Data Lake Format	`Iceberg`	Apache Iceberg table format

A.10 SageMaker Configuration

Setting	Default	Description
Training Instance	`ml.m5.xlarge`	Instance type for model training
Inference Instance	`ml.m5.large`	Instance type for batch inference
Model Framework	`XGBoost`	ML algorithm used for churn prediction

A.11 QuickSight Configuration

Setting	Default	Description
Dataset Refresh	On data update	Datasets refresh when new data is processed
SPICE	Enabled	In-memory acceleration for fast queries
Row-Level Security	Optional	Can be configured for multi-tenant deployments

https://www.10pearls.com

info@10pearls.com