

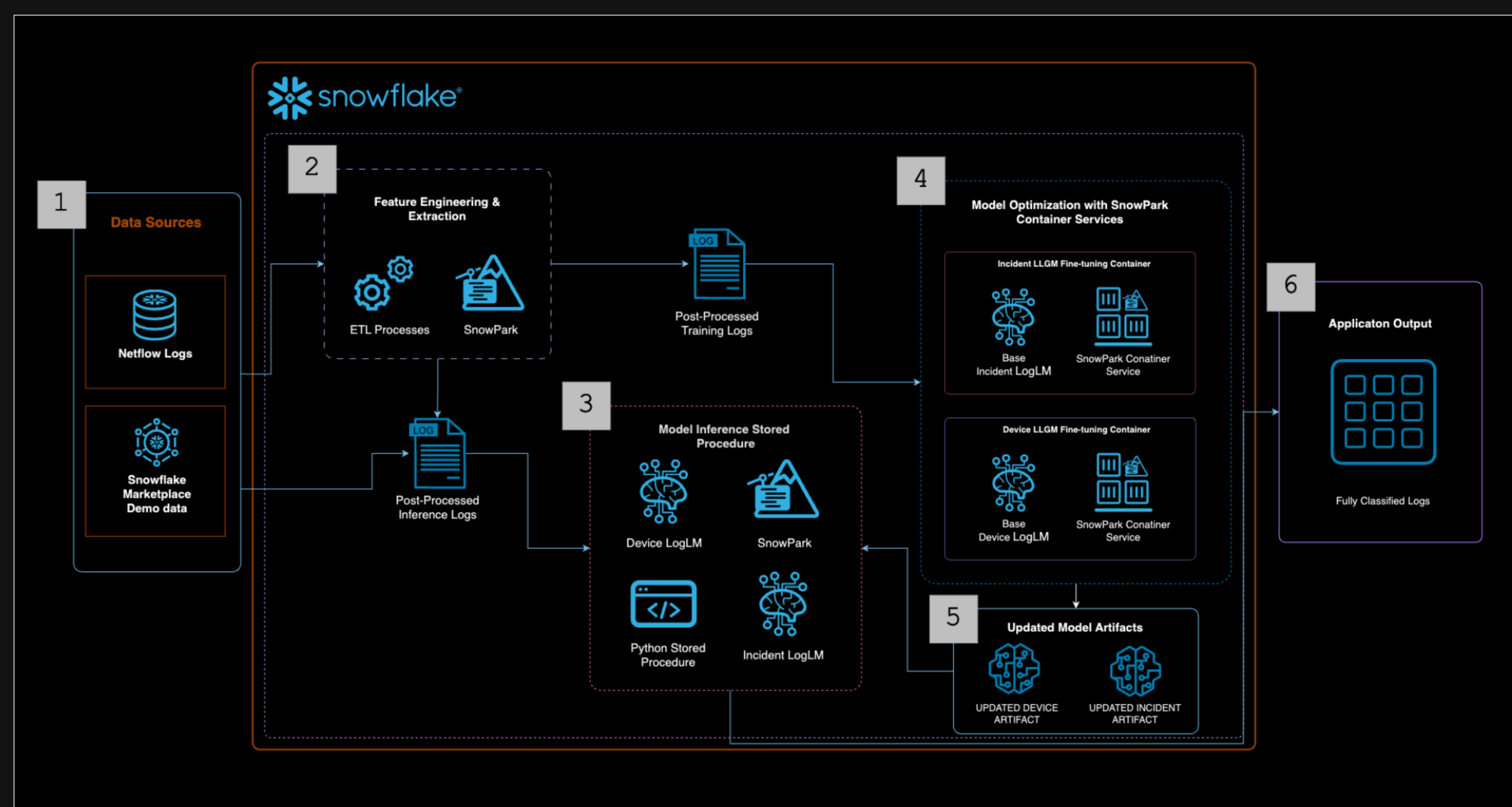
# Getting Started with TEMPO (Using Sample Data)

This guide will walk you through the process of setting up and using the TEMPO Native App in your Snowflake environment with provided sample data. The demo dataset is netflow from approximately 500 boxes on a corporate network, running for one week.

## Cost of Running Data

[Insert cost information here]

## Architecture Diagram



## Step 1: Install the TEMPO Native App

1. Obtain the TEMPO native app from the Snowflake Marketplace.
2. Once installed, the app will be available for use in your Snowflake environment.
3. Grant the app privileges to create the required compute resources:

```
GRANT CREATE COMPUTE POOL ON ACCOUNT TO APPLICATION TEMPO;  
GRANT CREATE WAREHOUSE ON ACCOUNT TO APPLICATION TEMPO;  
GRANT EXECUTE TASK ON ACCOUNT TO APPLICATION TEMPO;  
GRANT EXECUTE MANAGED TASK ON ACCOUNT TO APPLICATION TEMPO;
```

It's recommended to create a dedicated warehouse for the application with the following properties:

```
CREATE WAREHOUSE <YOUR_WAREHOUSE_NAME>  
WAREHOUSE_TYPE = 'SNOWPARK-OPTIMIZED'  
WAREHOUSE_SIZE = 'LARGEMEDIUM' ;  
  
GRANT USAGE ON WAREHOUSE <YOUR_WAREHOUSE_NAME> TO APPLICATION TEMPO;
```

The application comes with its own in-house warehouse (TEMPO\_WH), which will be used for container services and live inference only.

## Step 2: Initialize the Application

Call the startup procedure to initialize the app:

```
CALL TEMPO.MANAGER.STARTUP();
```

## Step 3: Grant Permissions for Sample Data Access

Grant the TEMPO app the necessary permissions to access the sample data:

```
GRANT USAGE ON DATABASE tempo_sample TO APPLICATION TEMPO;  
GRANT USAGE ON SCHEMA tempo_sample.inference_samples TO APPLICATION TEMPO;  
GRANT USAGE ON SCHEMA tempo_sample.training_samples TO APPLICATION TEMPO;  
GRANT SELECT ON ALL TABLES IN SCHEMA tempo_sample.inference_samples TO APPLICATION TEMPO;  
GRANT SELECT ON ALL TABLES IN SCHEMA tempo_sample.training_samples TO APPLICATION TEMPO;
```

## Step 4: Perform Inference (Tempoinference)

Use the `TEMPOINFERENC` stored procedure to perform inference on sample static log data. It takes the table's FQN (fully qualified name), a Boolean to control whether full logs or only anomalies are returned, and the log type, in this order.

```
CALL TEMPO.DETECTION.TEMPOINFERENC('<TABLE_FQN>', <RETURN BOOLEAN>, '<LOG_TYPE>');
```

Example:

```
CALL TEMPO.DETECTION.TEMPOINFERENC('tempo_sample.inference_samples.workstations_inference', True, 'websites');  
CALL TEMPO.DETECTION.TEMPOINFERENC('tempo_sample.inference_samples.webservers_inference', False, 'websites');
```

## Optional Step: Fine-Tuning Models

If you need to adjust the models for better performance with your specific data patterns, you can use the fine-tuning feature. This step is optional and should be performed only if the default models don't meet your specific needs.

### Step 5a: Grant Permissions for Fine-Tuning

If you decide to fine-tune, you will be making use of the in-house compute pool with GPU access and the following configuration:

[Configuration details here]

### Step 5b: Fine-Tune Models

The ideal fine-tuning sequence based on data availability is Device before Anomaly, so that the Anomaly model can use the Device outputs. To fine-tune the models, you would execute the following queries:

```
CALL TEMPO.FINE_TUNE.DEVICE_MODEL('<SERVICE_NAME>');  
CALL TEMPO.FINE_TUNE.WORKSTATION_MODEL('<SERVICE_NAME>');
```

Example:

```
CALL TEMPO.FINE_TUNE.DEVICE_MODEL('first device model fine tuning');
```

### Step 5c: Monitor Fine-Tuning Services

Check the status of fine-tuning services:

```
CALL SYSTEM$GET_SERVICE_STATUS('<FINE_TUNE_SERVICE_NAME>');
```

Example:

```
CALL SYSTEM$GET_SERVICE_STATUS('FINE_TUNE.sample_workstation');
```

Note: Fine-tuning can be a time-consuming process and may incur additional computational costs. Only proceed with fine-tuning if you have specific requirements that aren't met by the default models.

## Step 6: Live Inference

### Step 6a: Setting Up Live Inference

To start live inference, you need to initialize the necessary components such as streams, tasks, and live update tables. Before that, you need to allow change tracking on the specific table you would like live inference on:

```
ALTER TABLE '<TABLE_NAME>' SET CHANGE_TRACKING = TRUE;
```

### Step 6b: Start Live Inference

To start the live inference process, use the `START_LIVE_INFERENCE` procedure. This procedure sets up a data stream, creates a live updates table, and schedules a task that continuously monitors incoming data and applies the inference model.

```
CALL TEMPO.LIVE_INFERENCE.START_LIVE_INFERENCE('<TABLE_PATH>', '<MODEL_TYPE>', '<REFRESH_TIME>');
```

Example:

```
CALL TEMPO.LIVE_INFERENCE.START_LIVE_INFERENCE('mydb.myschema.mytable', 'workstation', '1M');
```

### Step 6c: Monitor, Suspend, and Resume Task

After creating a task, the task name should be visible in the Snowflake interface.

- Monitoring tasks:

```
SELECT *  
FROM TABLE(INFORMATION_SCHEMA.TASK_HISTORY(  
  TASK_NAME => '<Task_name>'  
));
```

- Suspending Task:

```
ALTER TASK <Task_name> SUSPEND;
```

- Resuming tasks:

```
ALTER TASK <Task_name> RESUME;
```

### Step 6d: Shutting Down Live Inference

When you need to stop live inference, use the `SHUTDOWN_LIVE_INFERENCE` procedure to safely terminate the process.

```
CALL TEMPO.LIVE_INFERENCE.SHUTDOWN_LIVE_INFERENCE('<TABLE_NAME>');
```

Example:

```
CALL TEMPO.LIVE_INFERENCE.SHUTDOWN_LIVE_INFERENCE('mytable');
```

After live inference is set up, review the results stored in the live updates table. This table will contain the incoming data along with the model's classification.

```
SELECT * FROM TEMPO.LIVE_INFERENCE.LIVE_UPDATES_ON-<TABLE_NAME>;
```

Example:

```
SELECT * FROM TEMPO.LIVE_INFERENCE.LIVE_UPDATES_ON_mytable;
```

## Additional Notes

**Performance Considerations:** Live inference tasks can consume significant resources. Ensure that your Snowflake environment is properly scaled to handle the workload.

This guide uses sample data provided with TEMPO. For using your own data, please refer to the custom data guide.