Ballerina
Swan Lake

Data Integration with ETL

October 2024

Hello!

Lakshan Weerasinghe

lakshanw@wso2.com | Senior Software Engineer **|** **@ballerinalang** | **WSO2**

Gayal Dassnayake

gayald@wso2.com | Senior Software Engineer **|** **@ballerinalang** | **WSO2**

About this Session

Coming Up

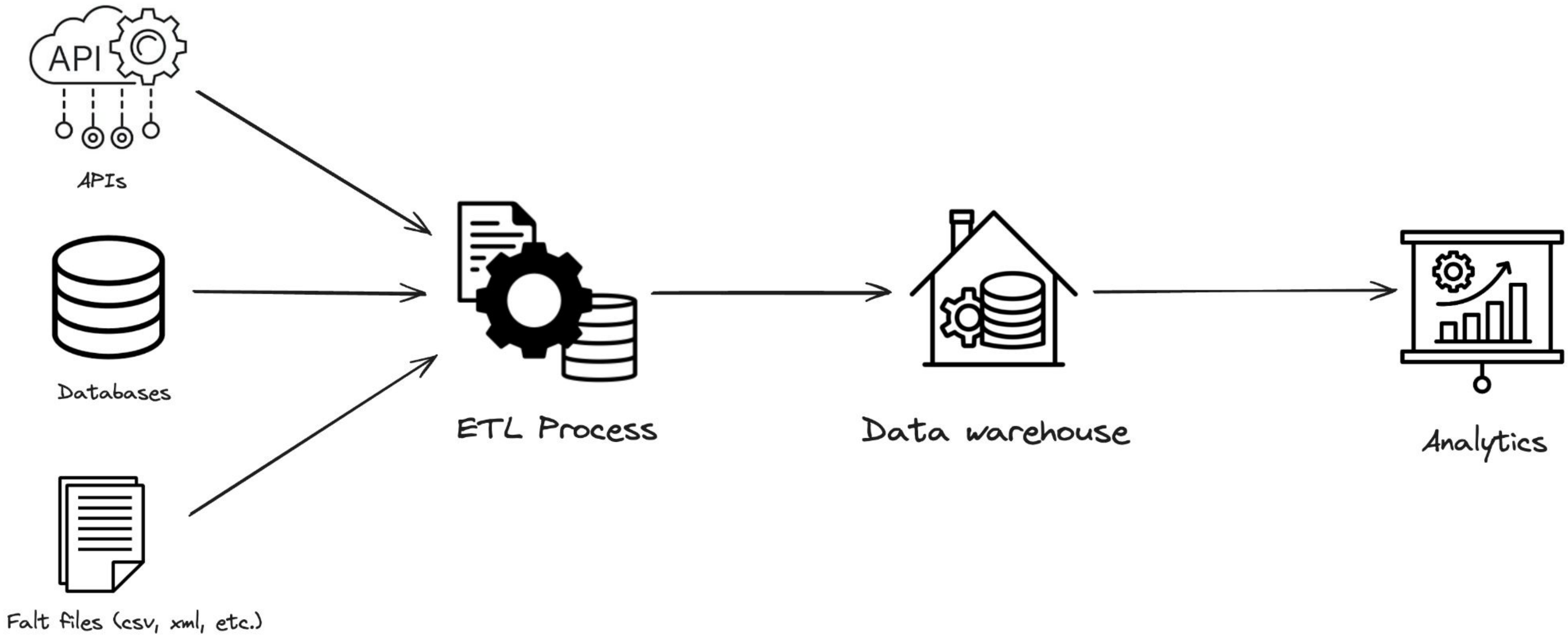
Introduction to ETL

Modern ETL Tools

Ballerina for ETL

Hands-on Session

Introduction



APIs

Databases

Falt files (csv, xml, etc.)

ETL Process

Data warehouse

Analytics

Data sources

Extract(E)

Programmatically read data from a data source

E.g.:

- Database
- API
- Flat file

Transform(T)

Apply data transform techniques on data, such as

- Removing duplicates
- Normalizing
- Filtering
- Enriching
- Changing the data format

Load(L)

Store the transformed data in a database

E.g.:

- Google sheets
- S3 buckets
- Databases
- Data warehouse

Modern ETL Tools

Modern ETL Tools

- Google cloud dataflow
- Azure data factory
- AWS Glue
- Informatica
- Tableau

Ballerina for ETL

Ballerina is a statically typed, data oriented, concurrent, graphical programming language designed for Integration.

What comes with Ballerina for ETL

Java

```java
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import java.io.*;
import java.lang.reflect.Type;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.util.List;

class HttpClientExample {
   public static void main(String[] args) throws Exception {
       Gson gson = new Gson();
       HttpURLConnection getConnection = (HttpURLConnection) new
URL("http://localhost:9090/albums").openConnection();
       getConnection.setRequestMethod("GET");
       String albumsJson = new BufferedReader(new InputStreamReader(getConnection.getInputStream())).lines()
               .reduce("", (acc, line) -> acc + line);
       Type albumListType = new TypeToken<List<Album>>() {}.getType();
       List<Album> albums = gson.fromJson(albumsJson, albumListType);
       HttpURLConnection postConnection = (HttpURLConnection) new
URL("http://localhost:9090/albums").openConnection();
       postConnection.setRequestMethod("POST");
       postConnection.setRequestProperty("Content-Type", "application/json; utf-8");
       postConnection.setDoOutput(true);
       Album newAlbum = new Album("Sarah Vaughan and Clifford Brown", "Sarah Vaughan");
       try (OutputStream os = postConnection.getOutputStream()) {
           os.write(gson.toJson(newAlbum).getBytes(StandardCharsets.UTF_8));
       }
       String postedAlbumJson = new BufferedReader(new
InputStreamReader(postConnection.getInputStream())).lines()
               .reduce("", (acc, line) -> acc + line);
       Album postedAlbum = gson.fromJson(postedAlbumJson, Album.class);
   }
}

class Album {
   String title;
   String artist;
   public Album(String title, String artist) { this.title = title; this.artist = artist; }
}
```

Ballerina

```ballerina
import ballerina/http;

type Album readonly & record {
   string title;
   string artist;
};

public function main() returns error? {
   http:Client albumClient = check new
("localhost:9090");

   // Sends a `GET` request to the "/albums" resource.
   Album[] albums = check albumClient->/albums;

   // Sends a `POST` request to the "/albums" resource.
   Album album = check albumClient->/albums.post({
       title: "Sarah Vaughan and Clifford Brown",
       artist: "Sarah Vaughan"
   });
}
```

First class support for data extraction from APIs

```
// http
http:Client albumClient = check new ("localhost:9090");
Album[] albums = check albumClient->/albums;

// graphql
graphql:Client graphqlClient = check new ("localhost:9090/graphql");
string document = "{ profile { name, age } }";
ProfileResponse response = check graphqlClient->execute(document);

// github
github:Client github = check new (gitHubConfig);
github:Repository[] userRepos = check github->/user/repos(visibility = "private", 'type = ());
```

Support more than 500+ SAAS connectors

- Salesforce
- GitHub
- SAP
- Stripe
- Docusign
- Slack
- Gmail
- Discord
- Kafka

Connectors for SQL and NoSQL databases

- Oracle Database
- SQL Server
- Mysql
- PostgreSQL
- Redis
- MongoDB

First class support for xml and json data formats

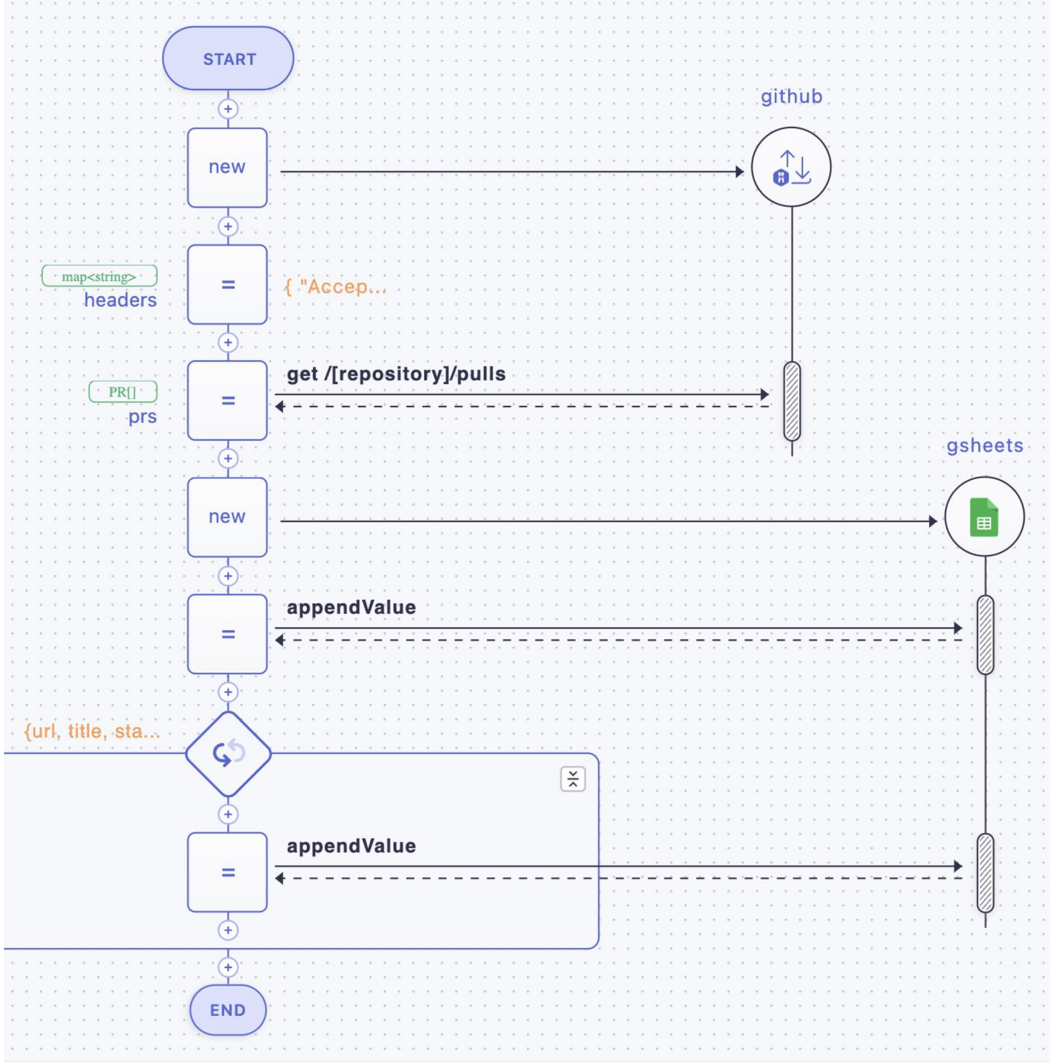
```
json[] users = [
   {
       user: {
          name: {
             firstName: "John",
             lastname: "Smith"
          },
          age: 24
       }
    },
    null
];

json firstUserName = check users[0].user.name;

string firstName = check firstUserName.firstName;
```

Language integrated queries for data transformation

```
Order[] orders = [
   {orderId: 1, itemName: "A", price: 23.4, quantity: 2},
   {orderId: 1, itemName: "A", price: 20.4, quantity: 1},
   {orderId: 2, itemName: "B", price: 21.5, quantity: 3},
   {orderId: 1, itemName: "B", price: 21.5, quantity: 3}
];

float income = from var {price, quantity} in orders
   let var totPrice = price * quantity
   collect sum(totPrice);

var quantities = from var {itemName, quantity} in orders
   group by itemName
   select {itemName, quantity: sum(quantity)};
```

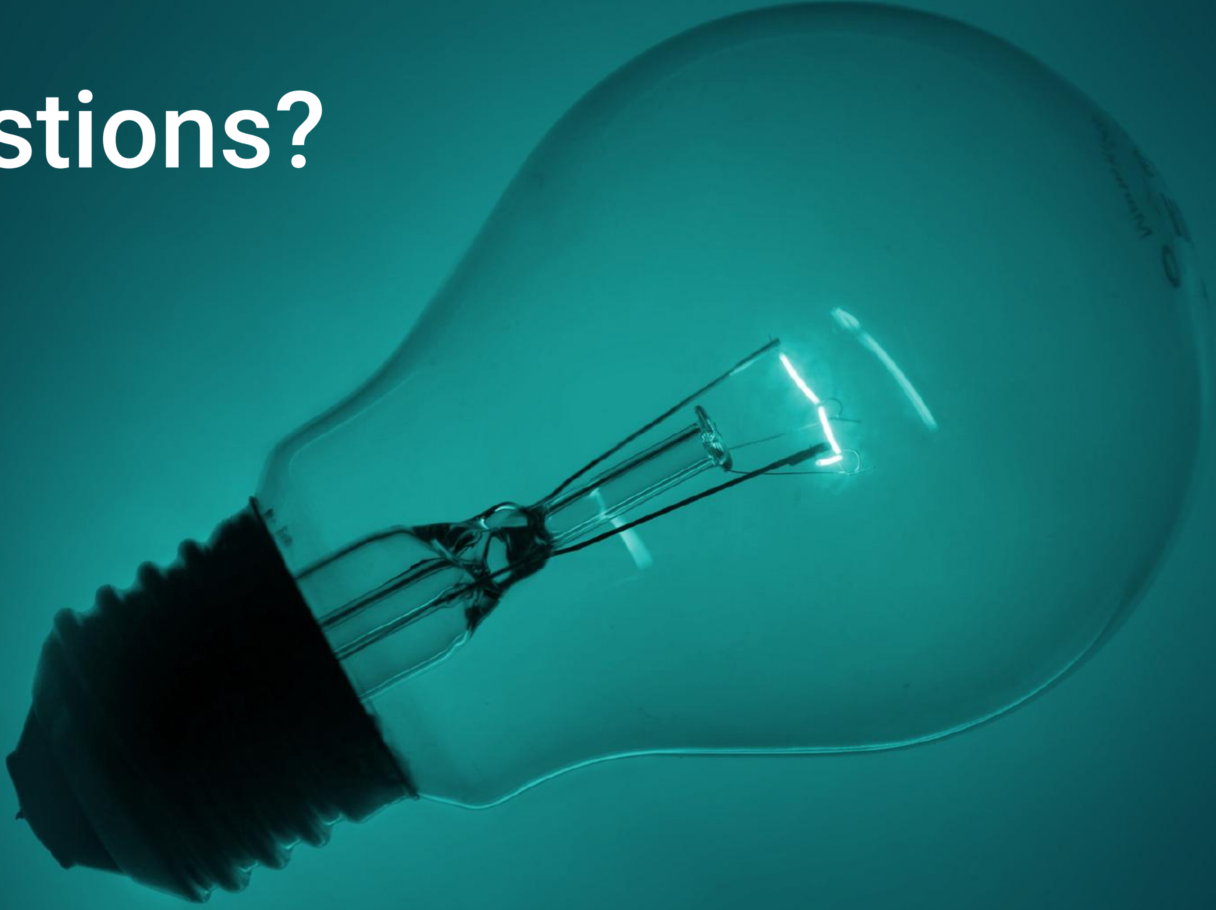
Visual Representations

```
public function main() returns error? {
   http:Client github = check new
("https://api.github.com/repos");
   map<string> headers = {
       "Accept": "application/vnd.github.v3+json",
       "Authorization": "token " + githubPAT
   };
   PR[] prs = check
github->/[repository]/pulls(headers);

   sheets:Client gsheets = check new ({auth: {token:
sheetsAccessToken}});
   _ = check gsheets->appendValue(spreadSheetId,
["Issue", "Title", "State", "Created At", "Updated
At"],
                {sheetName: sheetName});

   foreach var {url, title, state, created_at,
updated_at} in prs {
       _ = check gsheets->appendValue(spreadSheetId,
[url, title, state, created_at, updated_at],
                    {sheetName: sheetName});
   }
}
```



Ballerina
Swan Lake

Questions?



Learning resources

- Ballerina documentation and tutorials
    - Ballerina for ETL
        - https://ballerina.io/use-cases/etl/
    - Learn Guide
        - ballerina.io/learn/
    - Ballerina by example
        - ballerina.io/learn/by-example
    - YT Training Series
- WSO2 Certified Ballerina Developer - Swan Lake

Community Channels

https://github.com/ballerina-platform/

stackoverflow https://stackoverflow.com/questions/tagged/ballerina

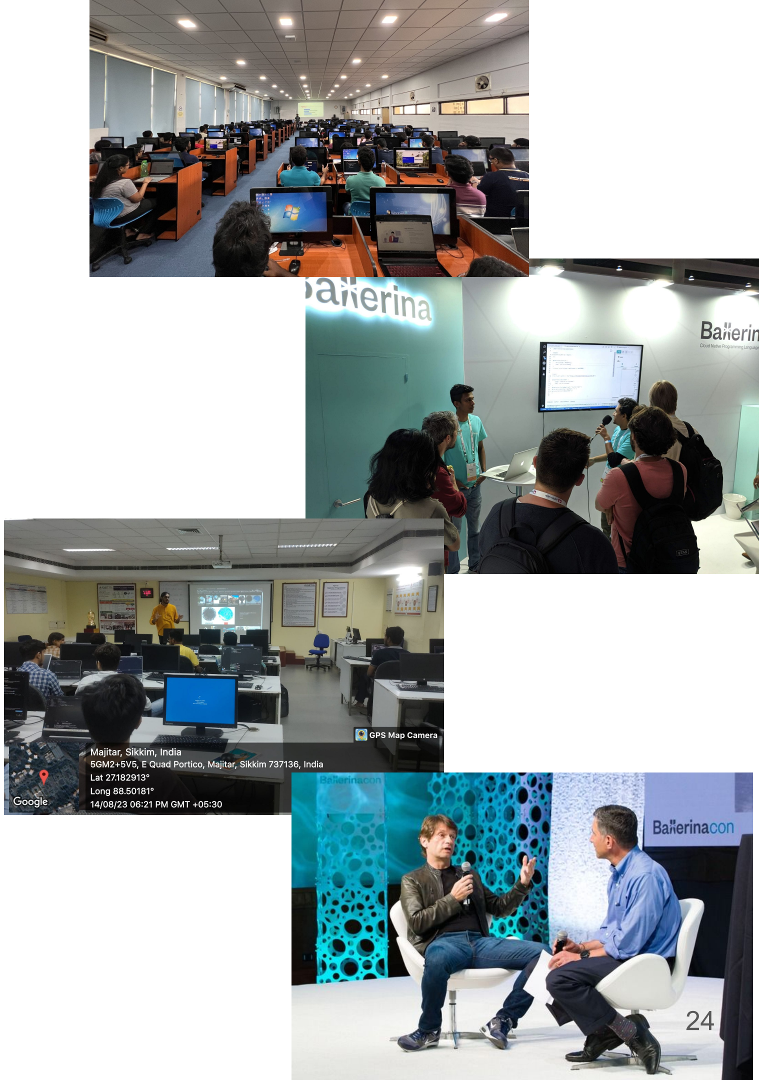
Discord https://discord.com/invite/ballerinalang

https://twitter.com/ballerinalang

Ballerina student program

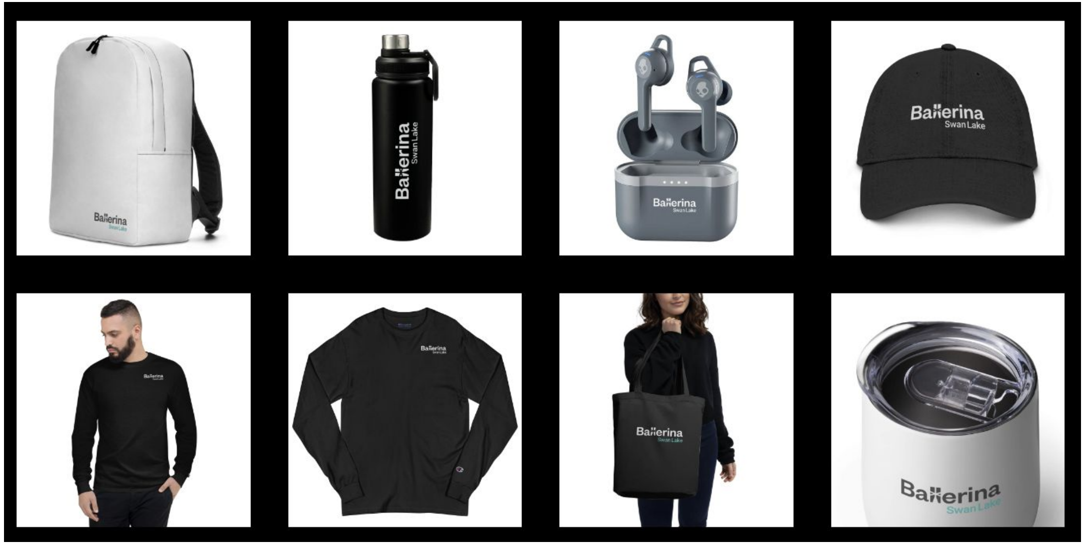
- Ballerina student engagement program
  https://ballerina.io/community/student-program/

- Ballerina ambassador program
  https://ballerina.io/community/ambassadors/



Hacktoberfest

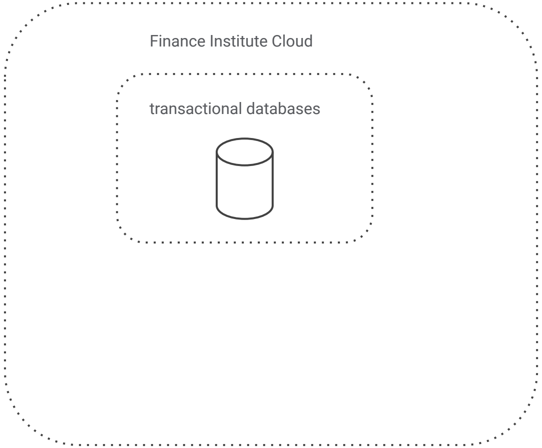
- Register before October 31st
- https://ballerina.io/hacktoberfest/



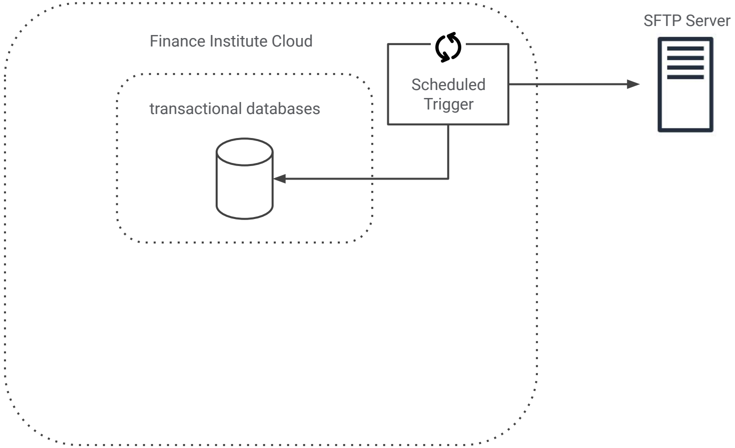
Hands-on Session

Demonstration

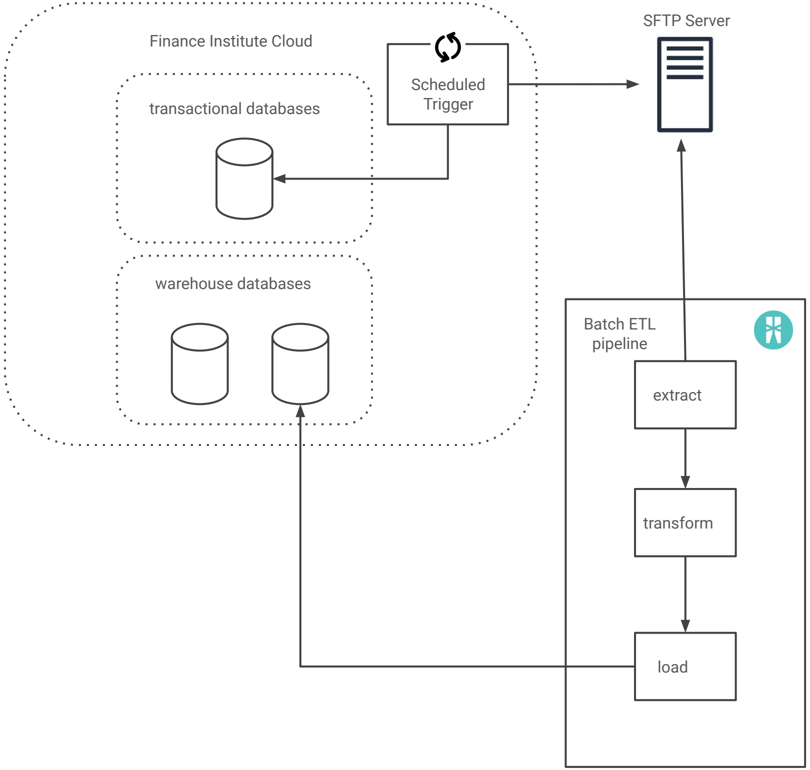
Demonstration



Demonstration



Demonstration

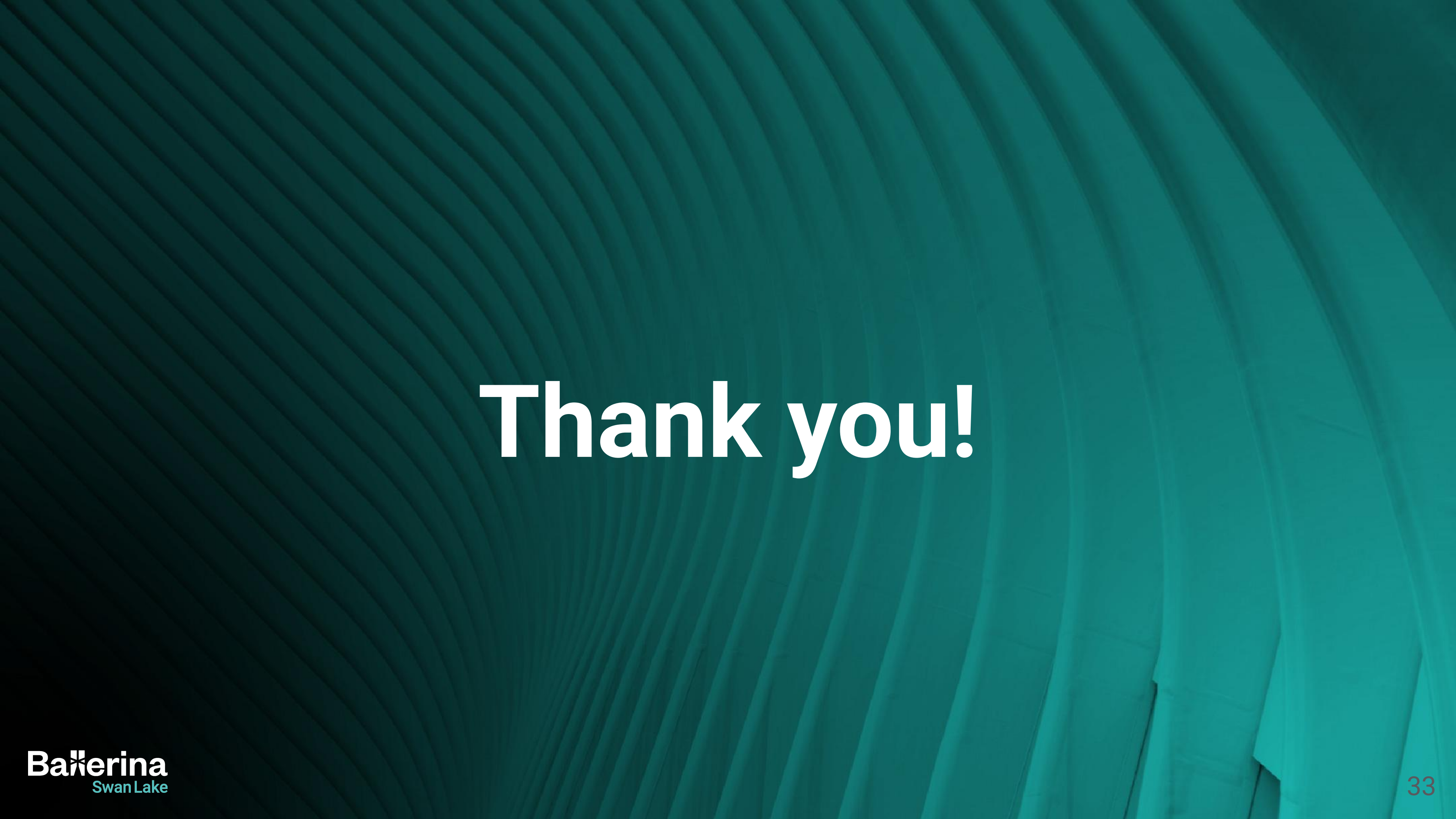


Ballerina

Setting Up Ballerina Development Environment

- Download the Ballerina distribution
- Install Ballerina
- Setup VSCode
- Go to https://ballerina.io/learn/get-started/ for more information

github.com/LakshanWeerasinghe/batch-etl-pipeline-demo



Thank you!