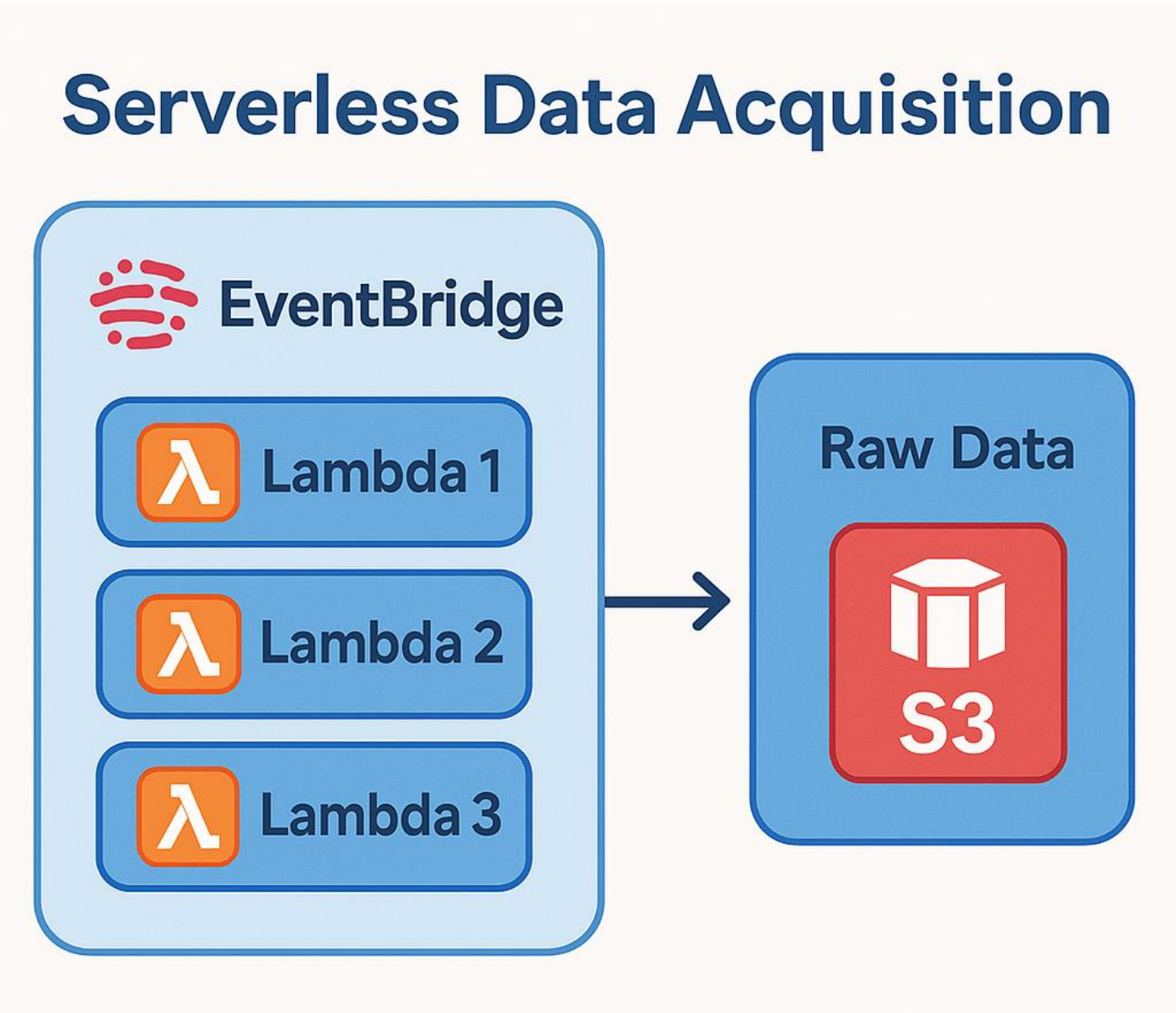


◇ STEP 1: Set Up AWS Account and S3 Bucket

1. ✅ **Create AWS Account** (if not already)
2. ✅ Go to **S3 → Create Bucket**
   a. Bucket name: `data-hackathon-smit-saad`
   b. Enable versioning (optional but good)
3. ✅ Inside bucket, create folder structure:

```
/raw/yahoofinance/
/raw/coinmarketcap/
/raw/openexchangerates/
```

◇ STEP 2: Set Up IAM Role for Lambda

1. ✅ Go to **IAM > Roles > Create Role**

2. ✅ Choose **Lambda** as service
3. ✅ Attach policies:
   a. `AmazonS3FullAccess`
   b. `CloudWatchLogsFullAccess`
4. ✅ Name it: `lambda-s3-role`

◇ STEP 3: Create Lambda Function – Yahoo Finance

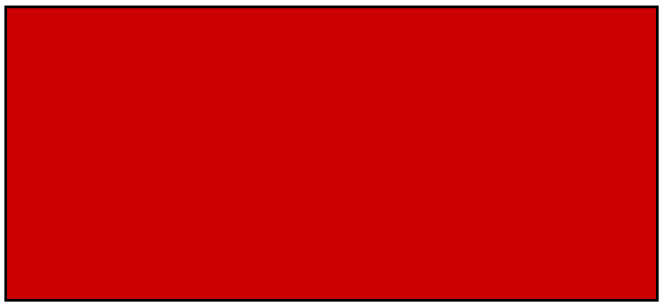
1. ✅ Go to **Lambda > Create Function**
   a. Name: `lambda_yahoofinance`
   b. Runtime: Python 3.9
   c. Role: Use `lambda-s3-role`
2. ✅ Write Code (sample below)
3. ✅ Set Timeout = **60 seconds**

📦 Sample Code – `lambda_yahoofinance.py`

```
import yfinance as yf
import json
import boto3
from datetime import datetime
import pytz

def lambda_handler(event, context):
    s3 = boto3.client('s3')
    bucket = 'data-hackathon-smit-saad'
    now = datetime.now(pytz.UTC)
    symbols = ['AAPL', 'MSFT', 'GOOGL']  # Short list for demo

    for symbol in symbols:
        data = yf.download(tickers=symbol, interval='1m', period='1d')
        if not data.empty:
            latest = data.iloc[-1]
            output = {
                "timestamp": now.isoformat(),
                "source": "Yahoo Finance",
                "symbol": symbol,
                "ohlcv": {
                    "open": latest['Open'],
                    "high": latest['High'],
                    "low": latest['Low'],
                    "close": latest['Close'],
```



```
                    "volume": int(latest['Volume'])
                },
                "status": "success"
            }
            path =
f"raw/yahoofinance/{now.strftime('%Y/%m/%d/%H%M')}_{symbol}.json"
            s3.put_object(
                Bucket=bucket,
                Key=path,
                Body=json.dumps(output)
            )
```

◇ STEP 4: Add Eve[redacted]e

1. ✅ Go to **Amazo[redacted]**
   a. Name: t[redacted]
   b. Schedule[redacted]
   c. Target: S[redacted]

◇ ST[redacted]change Rates

☑ lam[redacted]

- [redacted]
- Scrape CoinMarketCap → All Crypto
- Extract Top 10 by market cap
- Store in `raw/coinmarketcap/YYYY/MM/DD/HHMM.json`

☑ *lambda_openexchangerates.py*

- Call API using App ID
- Save result in: `raw/openexchangerates/YYYY/MM/DD/HHMM.json`

◇ STEP 6: Test and Validate

1. ✅ Trigger each Lambda manually
2. ✅ Check S3 bucket – is data stored correctly?
3. ✅ Check **CloudWatch Logs** for errors
4. ✅ Confirm files include:
   a. Timestamp
   b. Source name

c. Symbol
d. OHLCV / exchange data
e. Status