Replaying parallel requests to break a monolith

Use this architecture to break your monolith with confidence by setting up a parallel run strategy combined with the strangler fig pattern. First proxy the methods to be replaced with a microservice, storing a copy of user requests and monolith responses in a time-series database. Then replay the requests on your new microservice to compare the responses between the legacy monolith and the replacement microservice.

1. **AWS Migration Hub Refactor Spaces**
   - Users → Amazon API Gateway → strangler fig
   - Legacy system → proxy

2. **AWS Lambda proxy recorder**
   - AWS Step Functions manual replay flow
   - AWS Lambda database reset
   - Amazon S3 monolith responses
   - Amazon Kinesis Data Firehose
   - Amazon Kinesis Data Streams
   - Amazon S3 connector
   - AWS Lambda sorter function
   - Amazon Timestream ordered requests
   - AWS Lambda sequencer function
   - Amazon SQS FIFO queue
   - Amazon RDS temporary DB
   - AWS Lambda request repeater
   - AWS Lambda microservice responses
   - Amazon SNS alarm
   - AWS Lambda response comparer

3. **AWS Lambda proxy recorder**
   - AWS Step Functions periodic snapshot flow
   - AWS Backup snapshot
   - Amazon S3 bucket
   - Amazon RDS database
   - Amazon EC2 application
   - Amazon RDS backup
   - AWS Backup restore
   - AWS Step Functions workflow to replay
   - AWS Lambda microservice temporary databases
   - AWS Lambda microservice API
   - Amazon SNS alarm
   - AWS Lambda response comparer

4. **AWS Lambda proxy recorder**
   - AWS Step Functions periodic snapshot flow
   - AWS Lambda proxy recorder
   - AWS Lambda request repeater
   - AWS Lambda microservice responses
   - Amazon S3 bucket
   - Amazon RDS database
   - Amazon EC2 application
   - Amazon RDS backup
   - AWS Backup restore
   - AWS Step Functions workflow to replay
   - AWS Lambda microservice temporary databases
   - AWS Lambda microservice API
   - Amazon SNS alarm
   - AWS Lambda response comparer

5. **AWS Lambda proxy recorder**
   - AWS Step Functions periodic snapshot flow
   - AWS Lambda proxy recorder
   - AWS Lambda request repeater
   - AWS Lambda microservice responses
   - Amazon S3 bucket
   - Amazon RDS database
   - Amazon EC2 application
   - Amazon RDS backup
   - AWS Backup restore
   - AWS Step Functions workflow to replay
   - AWS Lambda microservice temporary databases
   - AWS Lambda microservice API
   - Amazon SNS alarm
   - AWS Lambda response comparer

6. **AWS Lambda proxy recorder**
   - AWS Step Functions periodic snapshot flow
   - AWS Lambda proxy recorder
   - AWS Lambda request repeater
   - AWS Lambda microservice responses
   - Amazon S3 bucket
   - Amazon RDS database
   - Amazon EC2 application
   - Amazon RDS backup
   - AWS Backup restore
   - AWS Step Functions workflow to replay
   - AWS Lambda microservice temporary databases
   - AWS Lambda microservice API
   - Amazon SNS alarm
   - AWS Lambda response comparer