

Creating Serverless Applications



Dror Helper

@dhelper www.helpercode.com

Module Overview



Lambda configuration and properties

Triggering Lambdas

- Using S3 events
- API Gateway for HTTP calls
- DynamoDB Streams to track activity
- SNS as a Lambda trigger

```
def handler_name(event, context)
    # insert code here
    return result;
```

Reminder: Lambda Function (Python)

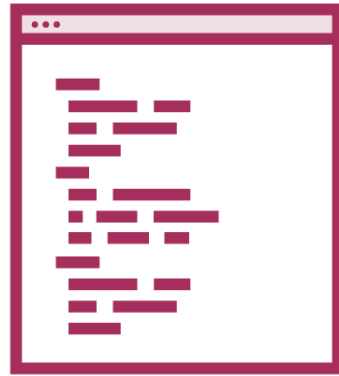
Parameters:

- Runtime information passed via the *context* parameter
- Event data (usually *dict*) in *event* parameter

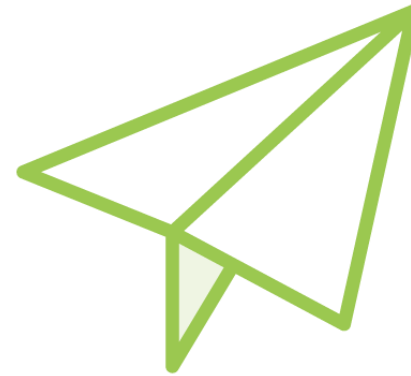
Using Event Sources



Event sources
can be an AWS
service



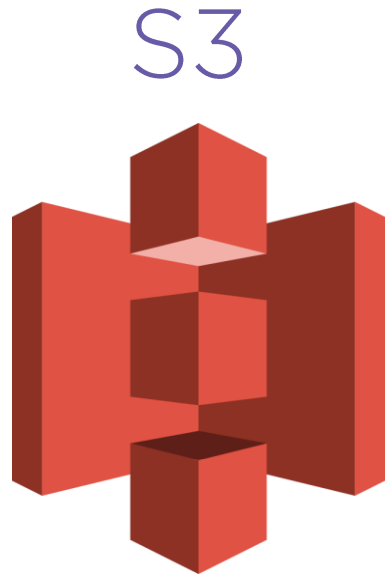
Event data
depends on
event type



Async/Sync
invocation



Multiple events
can run
concurrently



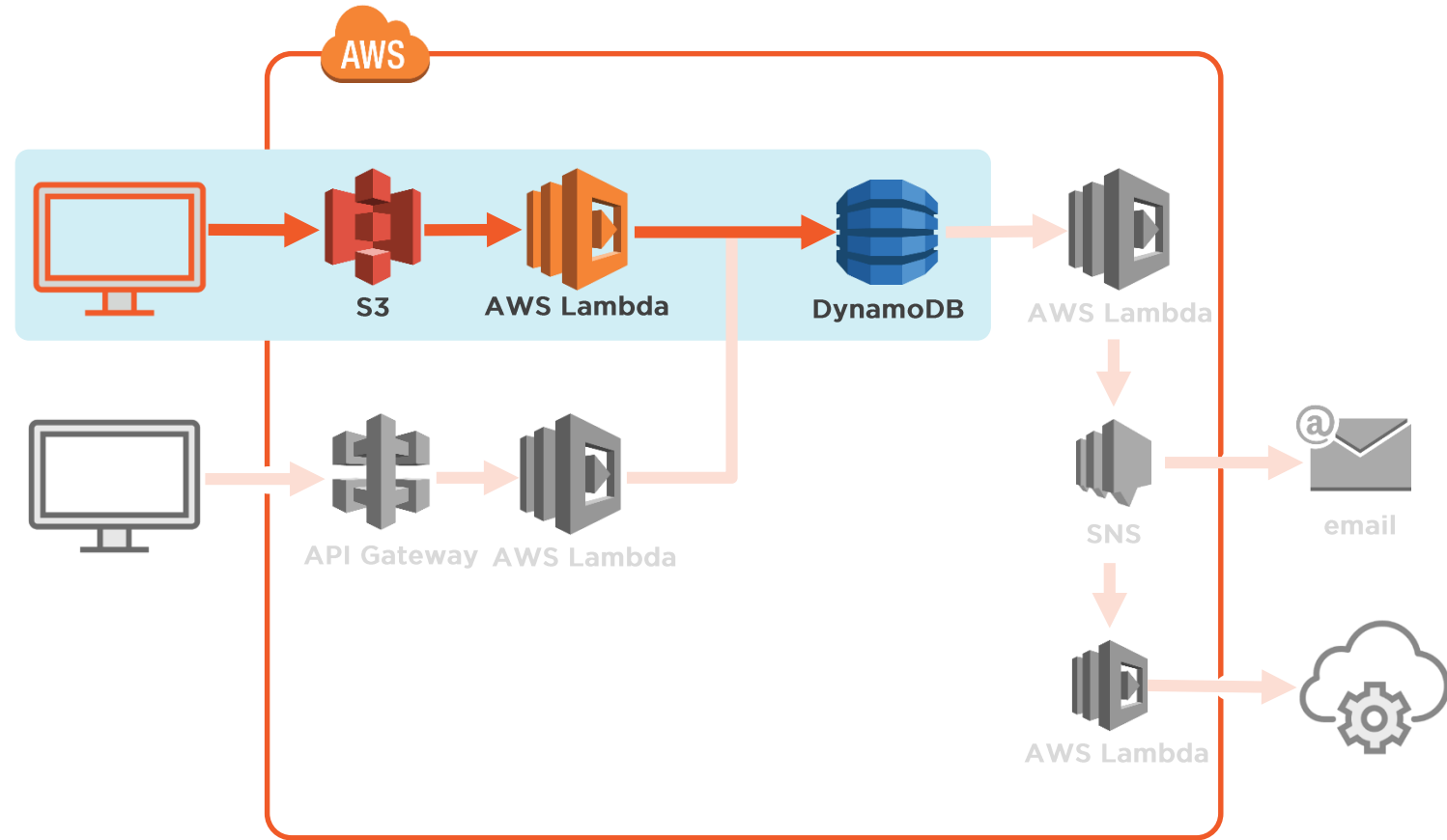
Simple bucket storage

- Manage buckets
- Store and retrieve data(files) in buckets
- Support versioning

Supported events

- New object created
- Object deleted
- RRS object lost

Demo



API Gateway



Create and maintain secure RESTful APIs

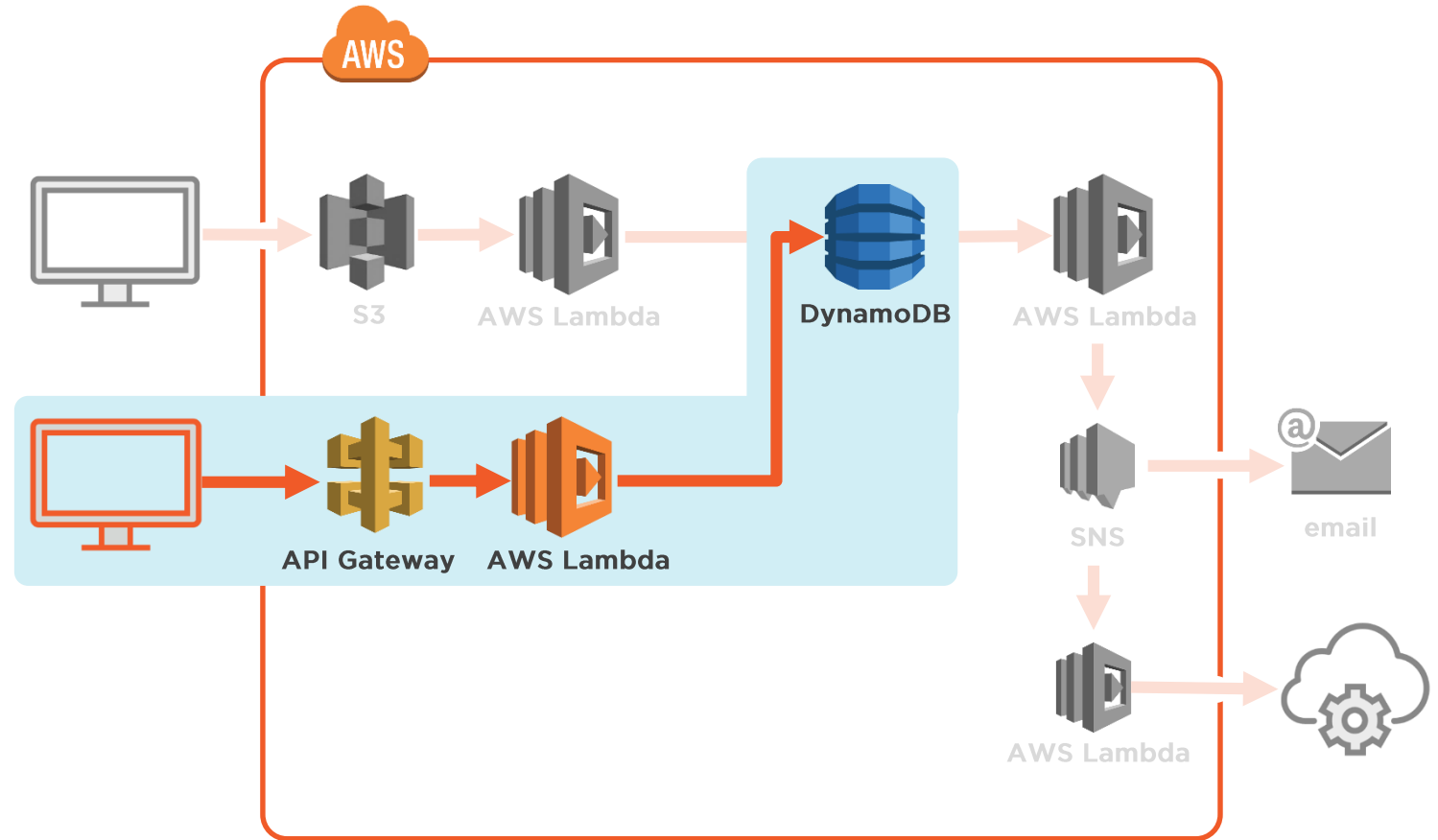
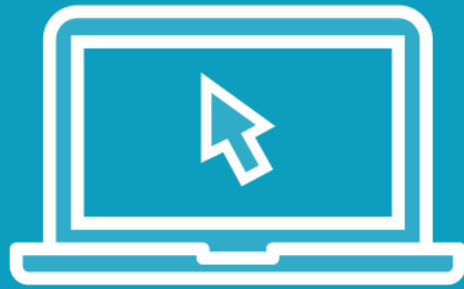
Lambda scenarios

- Proxy for other services
- Backend for API calls

Event types

- Pass through the entire request
- Catch-all methods
- Catch-all resources

Demo



DynamoDB



Fast and scalable NoSQL database

Store and retrieve items using keys

DynamoDB streams

- Capture changes in tables
- Ordered
- Stored up to 24 hours
- Enabled per table

SNS



Push a single message to multiple targets

Uses the “publish/subscribe” model

- Using topics and messages

Supports multiple subscription types

- Can trigger Lambda functions

```
import os
```

```
def lambda_handler(event, context):  
    snsArn = os.environ['sns_arn']
```

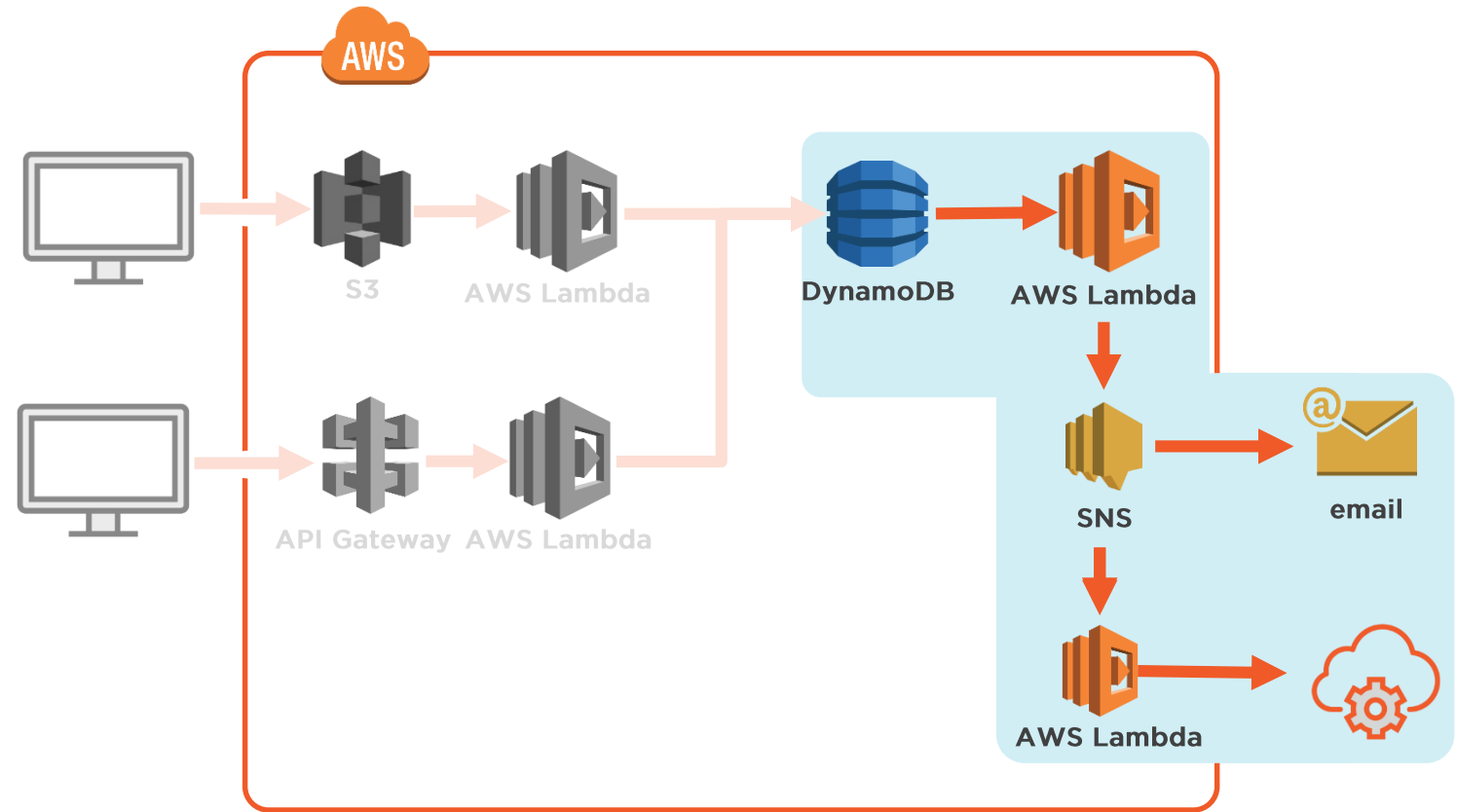
Environment Variables

Defined during Lambda Creation

- Key-value pairs
- Accessible from within the code

Used for configuration/settings

Demo





Concurrency Limits

Scaling indefinitely is usually not desired

- Not all resources can scale accordingly
- Manage infrastructure costs
- Regulate execution time

1000 concurrent executions per region

- Can be raised by contacting AWS support
- Can reserve limit per function
 - Deducted from total region limit

Invocations are throttled after limit

- Handling depends on event source and invocation type

Concurrent invocation can be monitored

Demo



Lambda concurrency levels

Summary



Building a serverless application

- Using event sources
 - Create IAM Role
 - Create a Lambda and test it
 - Add an event source

Passing data to lambda functions

- Using event data
- Using environment variables

Concurrency limits

- Changing default behavior
- Monitoring