NLPipe

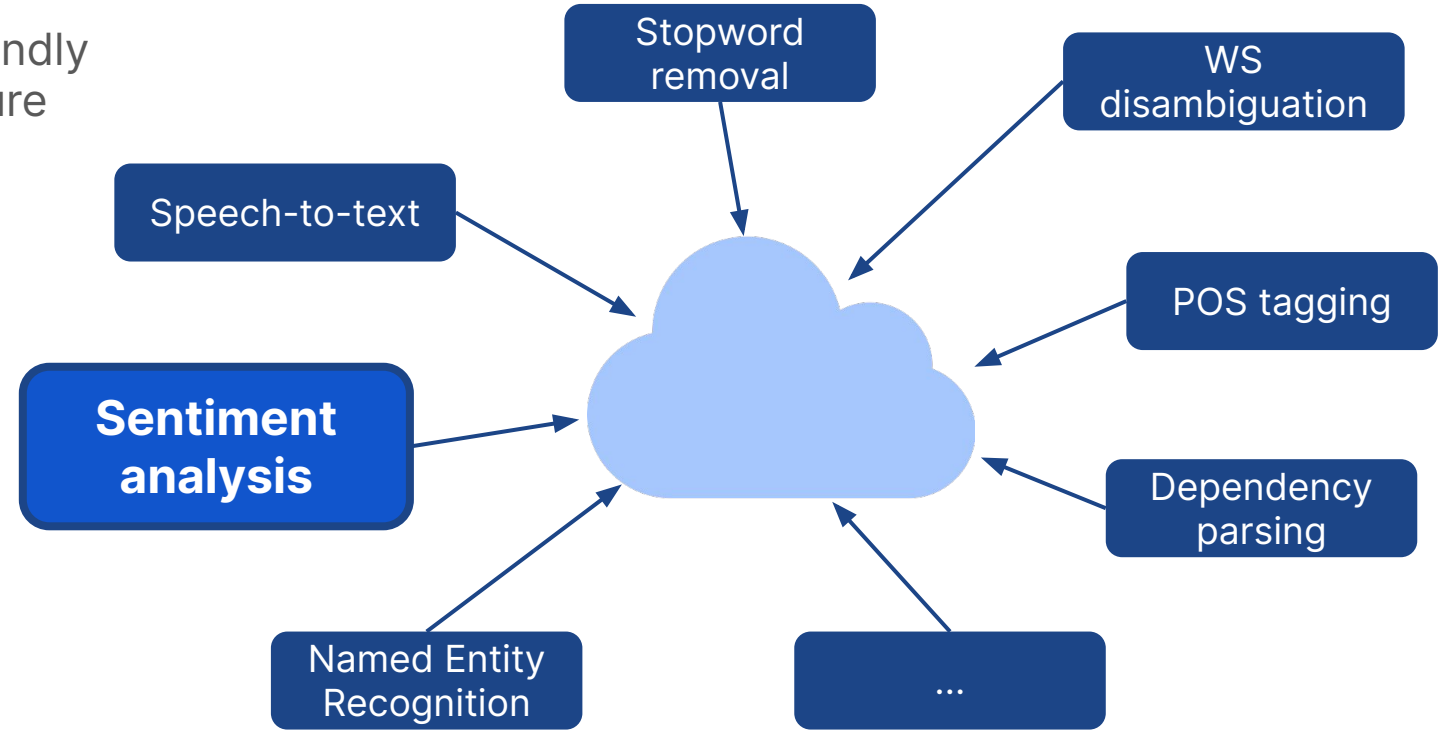
A scalable infrastructure
to deploy NLP solvers

Edoardo Gabrielli, Davide Quaranta, Daniele Solombrino

NLPipe

The idea

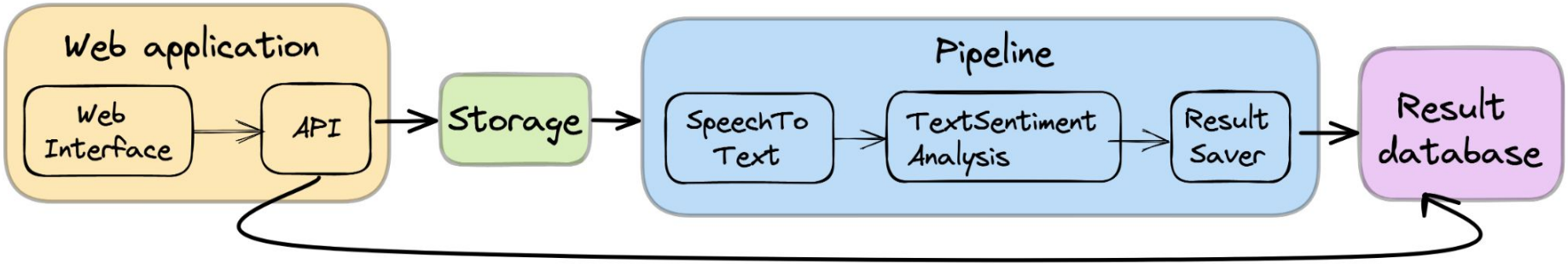
Create a NLP-friendly cloud infrastructure



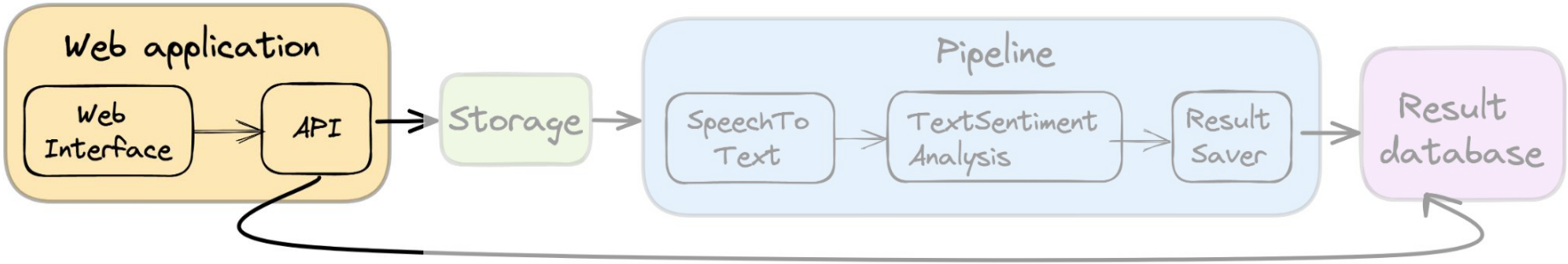
Driving principles

- **Scalability**
- **Modularity**
- **Programming language agnosticity**
- Adaptable to other NLP tasks
- Minimum privilege design

Design: the big picture



Design: web application



NLPipe

Sentiment Analysis aaS

Upload a recording of someone talking, and get the sentiment of the speech, via a lightweight serverless service 🚀

Sfoglia... Nessun file selezionato. UPLOAD

NLPipe

Sentiment Analysis aaS

Your request

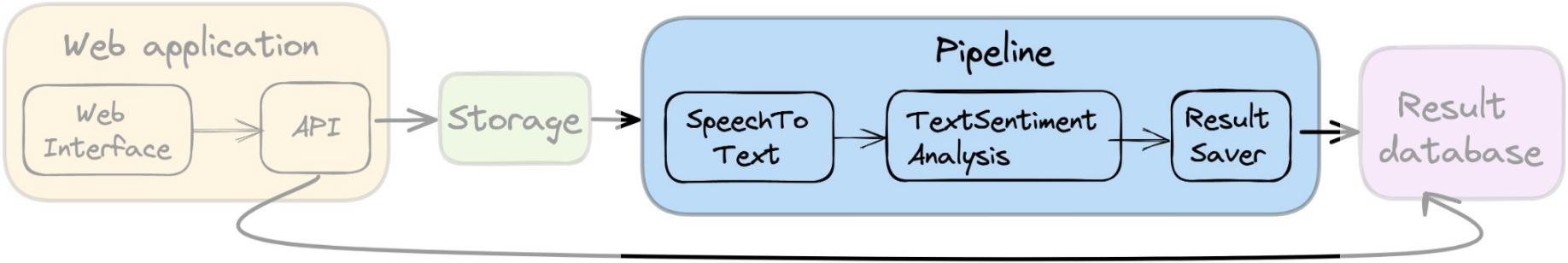
ID: 8a845cbc-6023-4f3f-8a84-0f9eedceab10

Status: **finished**

Sentiment: **positive**

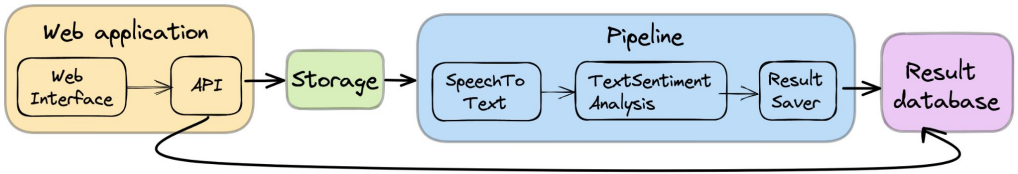
Last update at **Sat Jun 26 2021 16:55:15**

Design: pipeline



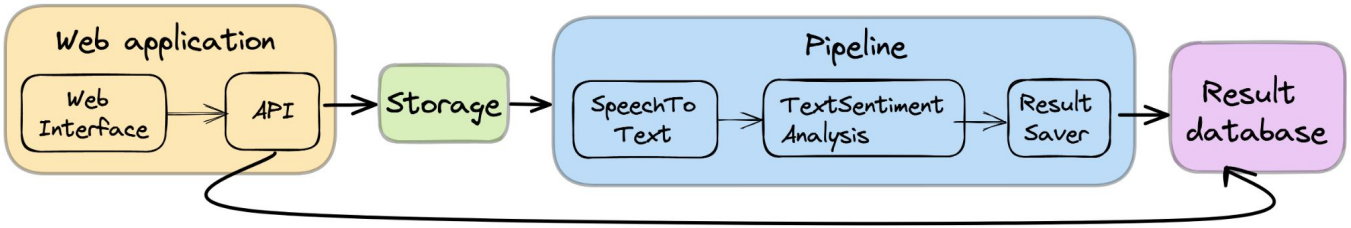
Implementation

- Local development
- **Web app**: Go
- **Pipeline**: Python 3
- **Storage**: s3-mock
- **DynamoDB**: dynamodb-local
- Docker Compose

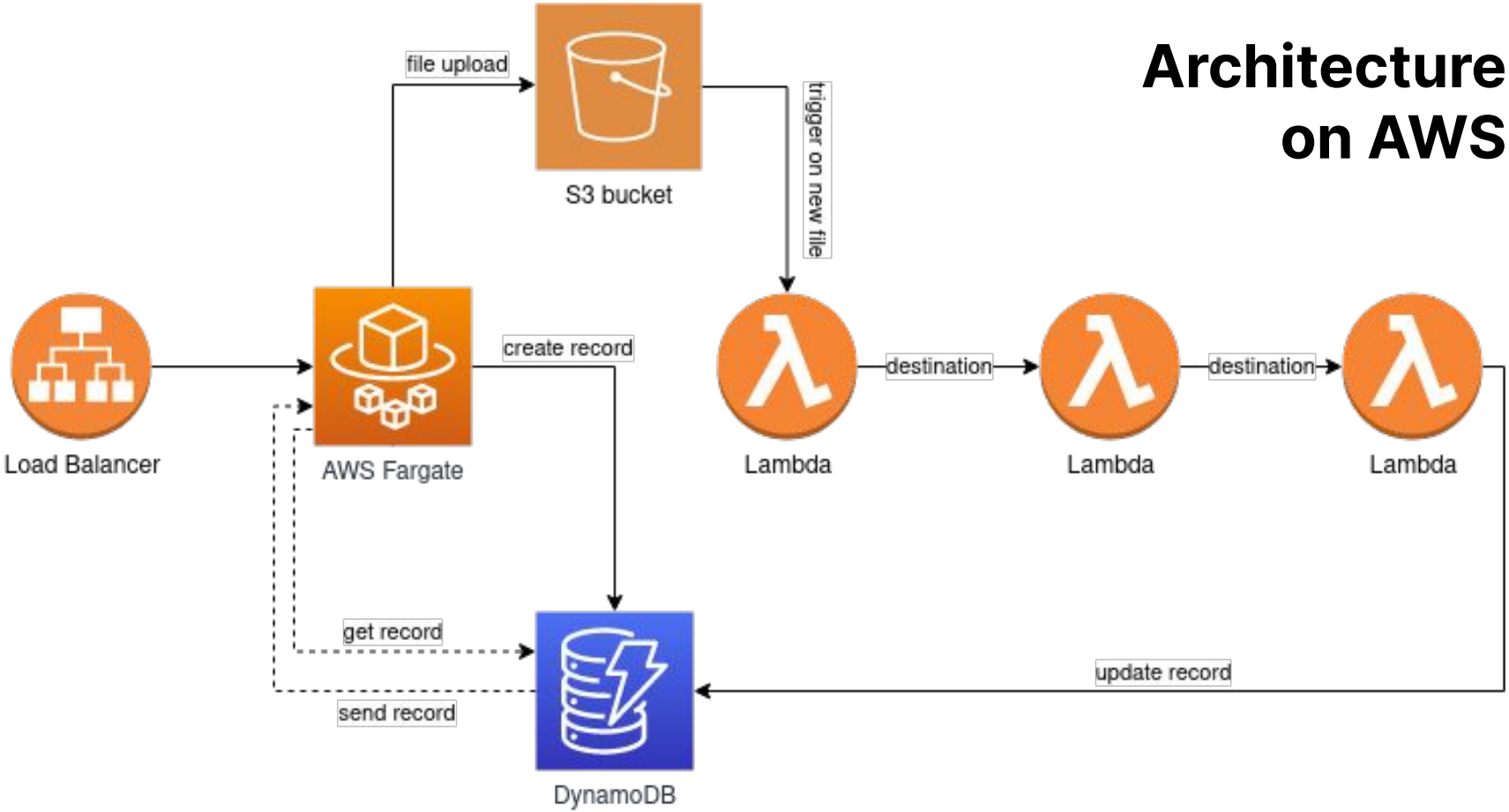


Deployment on AWS

- **Web application** → AWS **Fargate**
- **Storage** → Amazon **S3**
- **Pipeline** → AWS **Lambda** functions
- **Database** → Amazon **DynamoDB**

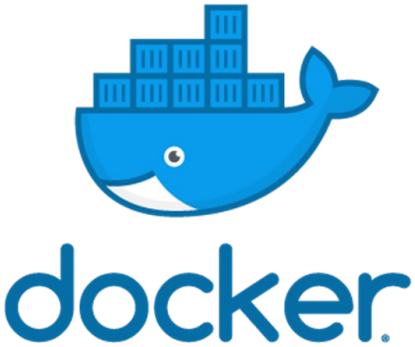


Architecture on AWS



Web application on AWS Fargate

```
## STAGE 1
FROM golang:1.15 as builder

RUN apt update && apt install -y --no-install-recommends ca-certificates

ENV GO111MODULE=on \
    CGO_ENABLED=0 \
    GOOS=linux \
    GOARCH=amd64

WORKDIR /app
COPY . .
RUN go mod download

RUN go build

## STAGE 2
FROM scratch
COPY --from=builder /app/nlpipe /app/
COPY --from=builder /app/html /app/html
COPY --from=builder /etc/ssl/certs/ca-certificates.crt /etc/ssl/certs/ca-certificates.crt

EXPOSE 8001

WORKDIR /app
ENTRYPOINT ["./nlpipe"]
```



Web application on AWS Fargate

Port Mappings

Host Port	Container Port	Protocol
8001	8001	tcp

Environment Variables

Key	Value/ValueFrom
DYNAMODB_ENDPOINT	https://dynamodb.us-east-1.amazonaws.com
DYNAMODB_TABLE	nlpipe-results
REGION	us-east-1
S3_BUCKET	uploads.nlpipe
S3_ENDPOINT	https://s3.us-east-1.amazonaws.com

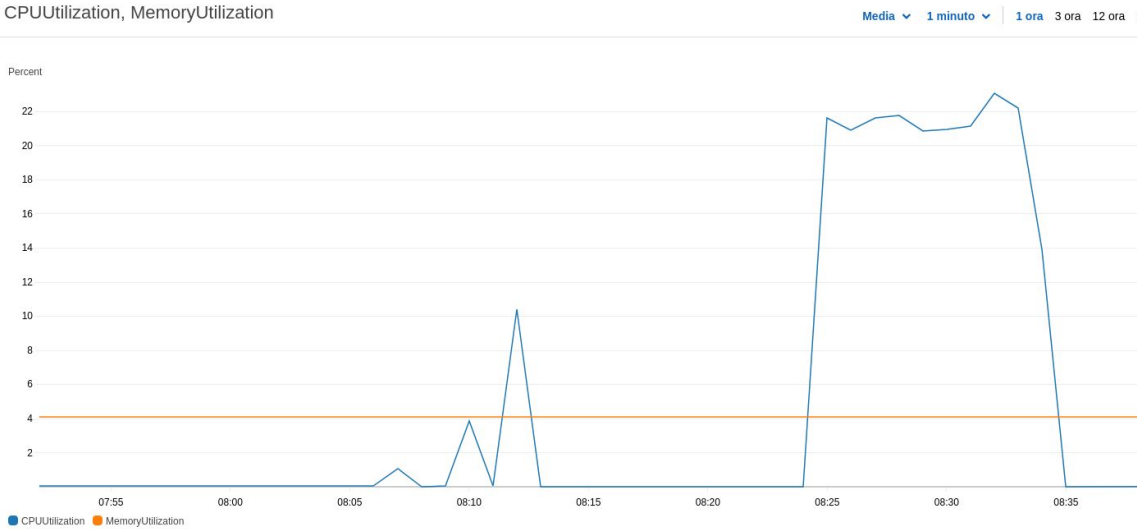
- ECS Repository
- Task Definition
- Container Definition

Autoscaling and balancing AWS Fargate

- Integrated in AWS Fargate
- **Service** definition
- Application Load Balancer
- Port mapping
- Autoscaling policy

CloudWatch

	Gruppo di log
☐	/aws/lambda/dynamodb-writer
☐	/aws/lambda/sentiment-analysis
☐	/aws/lambda/speech2text
☐	/ecs/nlpipe-app



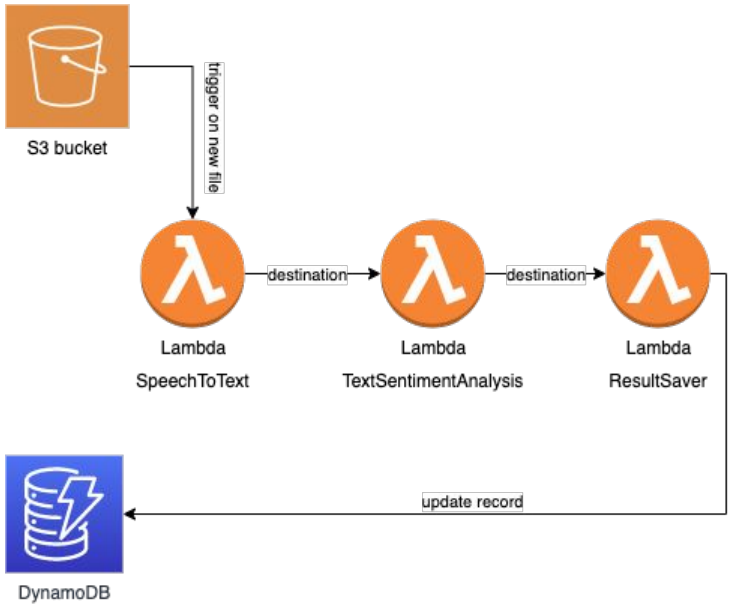
Gruppo di log: /ecs/nlpipe-app

#	@timestamp	@message
▶ 1	2021-07-11T11:23:08.908Z	time="2021-07-11T11:23:08Z" level=info msg="Configuration from environment variables: {\"Region\":\"us-east-1\",\"DynamoDbEndpoint\":\"https
▶ 2	2021-07-11T11:23:08.908Z	time="2021-07-11T11:23:08Z" level=info msg="API listening on :8001"
▶ 3	2021-07-11T09:33:50.752Z	time="2021-07-11T09:33:50Z" level=info msg="Configuration from environment variables: {\"Region\":\"us-east-1\",\"DynamoDbEndpoint\":\"https
▶ 4	2021-07-11T09:33:50.752Z	time="2021-07-11T09:33:50Z" level=info msg="API listening on :8001"
▼ 5	2021-07-11T08:06:47.664Z	time="2021-07-11T08:06:47Z" level=warning msg="Got a non audio file. Aborting."
	@ingestionTime	1625990809207
	@log	070236449957:/ecs/nlpipe-app
	@logStream	ecs/nlpipe/937a2d83269e4efb90bd855b44554527
	@message	time="2021-07-11T08:06:47Z" level=warning msg="Got a non audio file. Aborting."

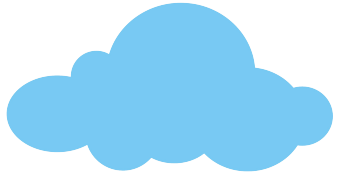
Pipeline on AWS Lambda

Tackled **challenges**

- Autoscaling
  - Managed
- Data passing
  - **Context** and **Environment**
  - AWS SDK
- Pipelining
  - AWS **Events** and **Destinations**
- External libs and files
  - **Layers**



Testing (1/2)



Hypothesis to prove:

- ☁️ **Cloud** → better scaling options than **Monolith**
- ⏱️ Lambdas → execution **time** depends on **RAM**
- 💰 **Bad** capacity planning → **higher** costs

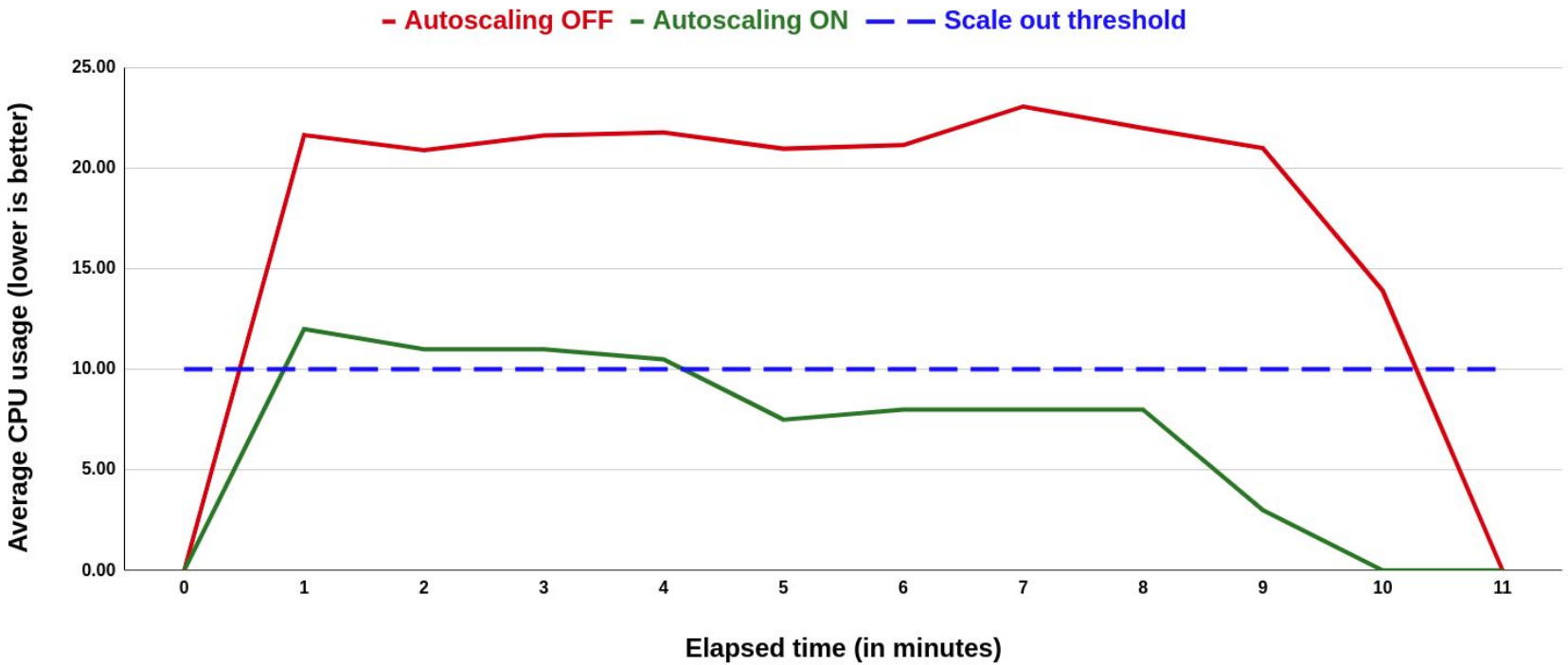
Testing (2/2)

Tests structure:

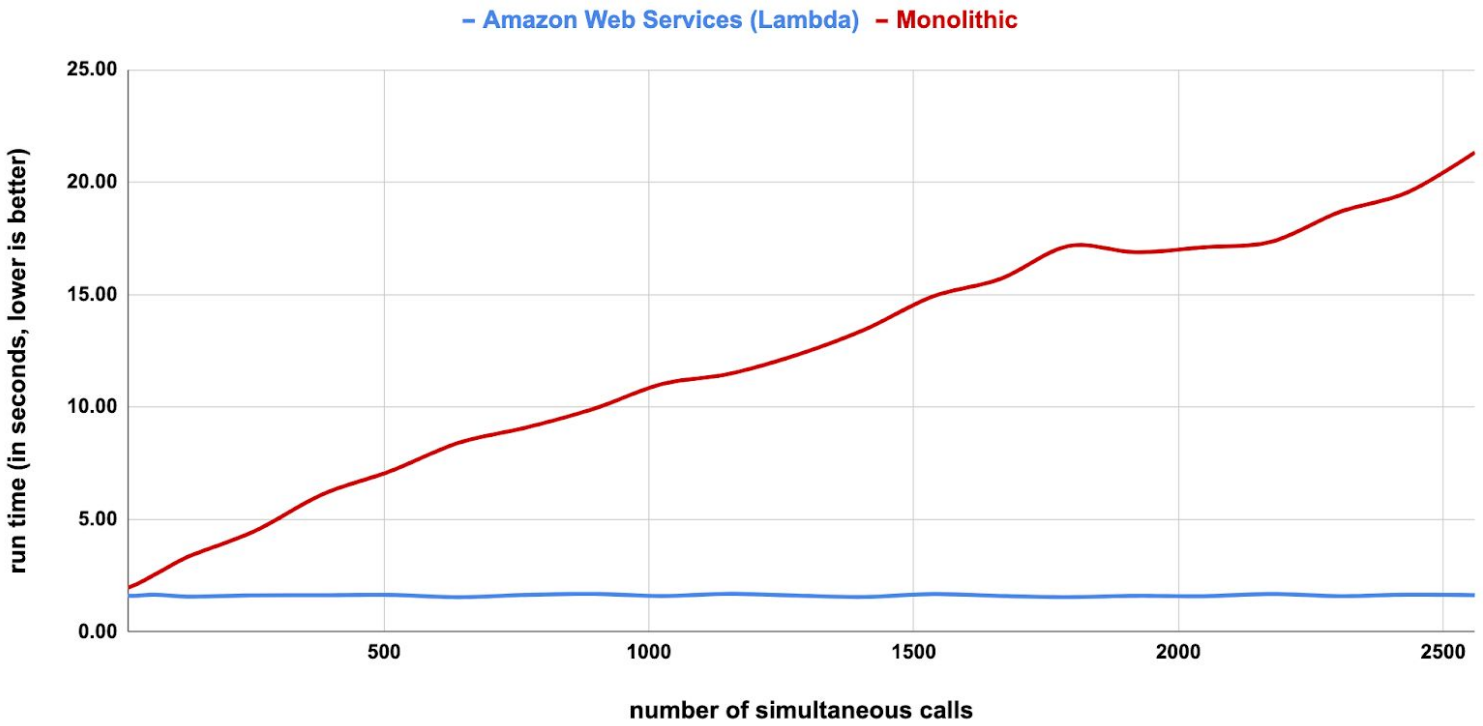
- **Batch size** → **number** of **simultaneous** function **calls**
- **Increasing** batch size (up to 2560)
- **Bash** + **AWS CLI** + **Postman** + **Newman**
- Gather **data** to test **hypothesis**
  - Execution times
  - CPU usage
  - RAM usage
  - Billing costs

POSTMAN

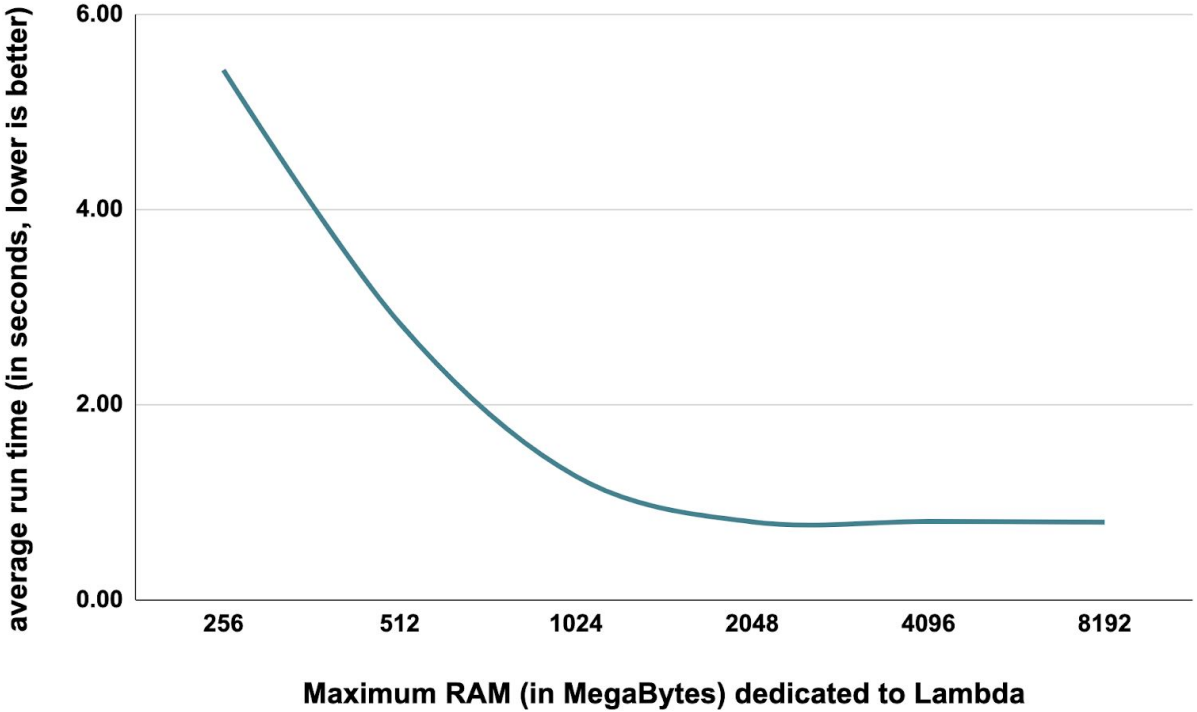
Validation (1/4): Fargate autoscaling



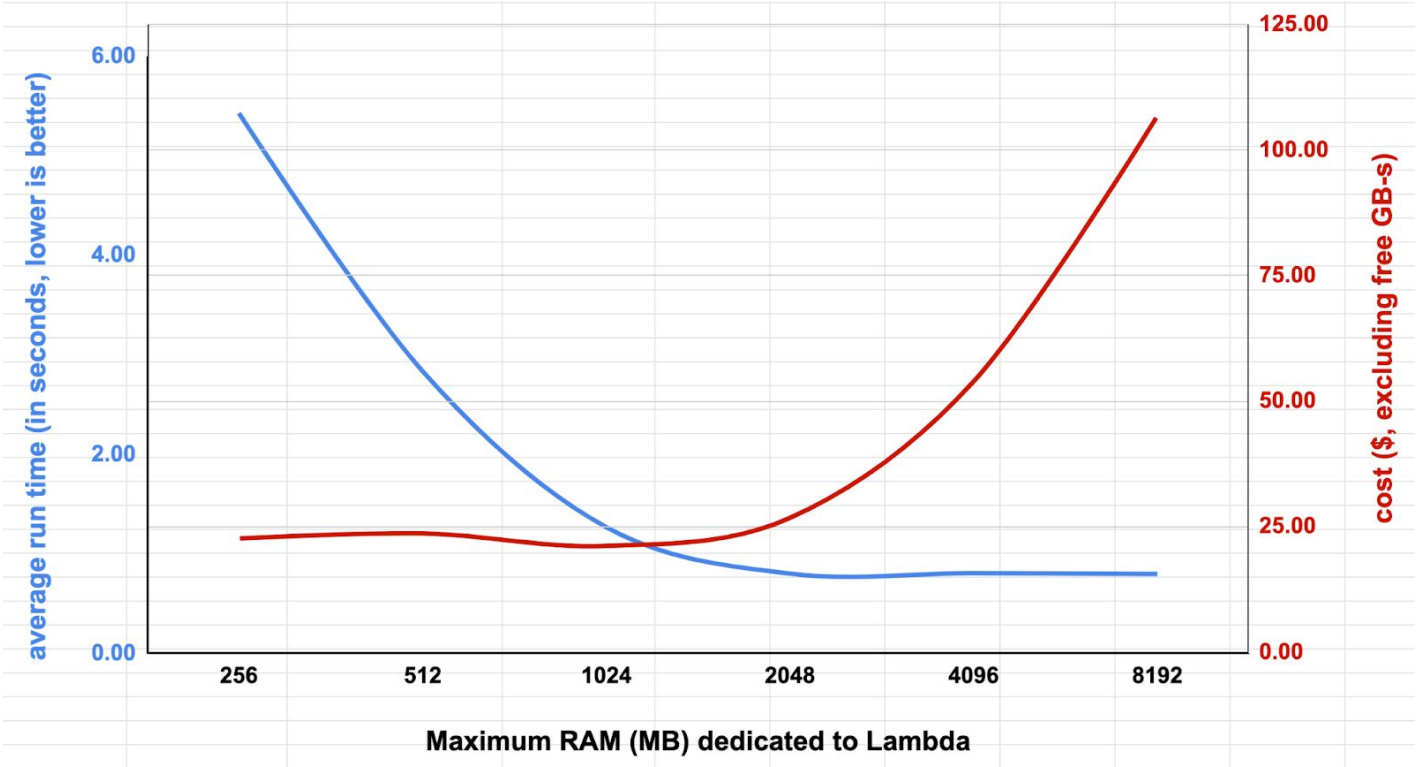
Validation (2/4): Lambda vs. monolithic



Validation (3/4): Lambda RAM vs. time correlation



Validation (4/4): economic cost of bad planning



Conclusion

- NLPipe: **scalable** and **flexible cloud** infrastructure for NLP solvers
- **Pipeline** design
- Collaboration between **independent components**
- Local implementation
- **AWS** deployment
- **Formulate test** hypothesis
- **Evaluate test** results