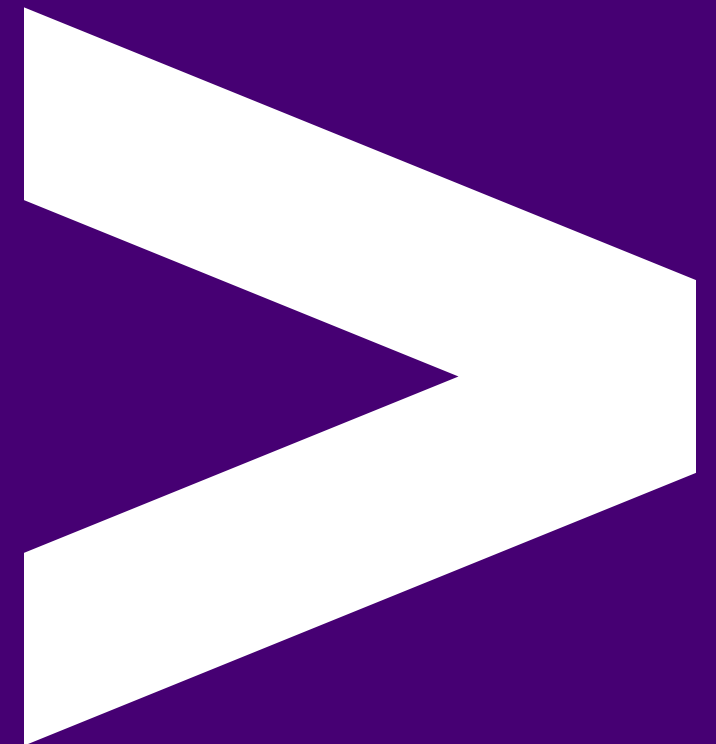







AWS 05 - Introduction to AWS Lambda



AWS sessions list

- AWS 01 AWS + Cloud Intro  1.5hrs
- AWS 02 AWS CLI Setup  1.5hrs
- AWS 03 S3 Storage (Console)  1.5hrs
- AWS 04 CloudFormation Intro + S3 Storage (IaC)  1.5hrs
- AWS 05 Lambda Intro  1.5hrs
- AWS 06 Lambda (IaC) 1.5hrs
- AWS 07 Redshift (IaC) 1.5hrs
- AWS 08 EC2 (IaC) + Grafana setup 1.5hrs

Overview

- Code running as a service
- Lambda as a form of Compute in AWS
- Creating a basic Lambda
- Event Triggers
- Testing lambdas
- Configuration including Environment Variables

Learning Objectives

- What code as a service is
- What Serverless Applications are
- Python code structure in Lambdas
- How to Trigger a Lambda
- Using the Event object
- Configuration like Environment variables

Lambda



Lambda

- 100% code, 0% infrastructure
- Run code without worrying about OS, patching, scaling, any physical hardware
- Never worry about capacity again
- Lambdas run in response to events such as data changes in S3, DB record being inserted
- You can even call them from through HTTP requests, SDK, or the AWS CLI

Lambda Triggers

A lambda function is automatically invoked when one of its triggers is activated.

For example:

- When a record has been inserted into a DB table
- When a file has been uploaded to S3
- When a commit is pushed onto a repo hosted in CodeCommit (Git for AWS)
- When a monitoring alarm goes off

Lambda Pricing Model

Number of requests: First 1 million requests per month are free, \$0.20 per 1 million after (cheap!)

Duration: Calculated from the time your code begins until it terminates, up to the millisecond. The price depends on how much memory you allocate. Roughly \$0.0000166667 for every GB-second used. The first 400,000 are free per month.

Example:

Lambda function allocates 0.5 GB memory & runs for 2 seconds

$\text{GB-seconds} = 0.5 \text{ GB} \times 2 \text{ seconds} = 1 \text{ GB-second}$

So the cost is ~\$0.0000166667

Limitations

- Cold starts: Time it takes to kick off an instance (it's a container under the hood)
- Difficult to scale without understanding the concurrency execution model
- Tightly integrated to work with other AWS services so may have potential 'lock-in'
- Can be difficult to develop locally
- Unsuitable for tasks that take 15+ minutes

Use Cases

- Tasks that take less than 15 minutes to complete
- Asynchronous, event-driven workloads
- Consistent level of traffic

Quiz Time! 🧐

Which of the following best describes AWS Lambda?

1. An AWS service for managing server infrastructure
2. A serverless compute service that runs your code in response to events
3. A tool for automatically scaling EC2 instances
4. A container orchestration service for running containerized applications

Answer: 2

After what time will an AWS Lambda function timeout?

- 1. 5 minutes
- 2. 10 minutes
- 3. 15 minutes
- 4. 30 minutes

Answer: 3

What is a potential drawback of AWS Lambda?

1. Limited integration with other AWS services
2. Difficulty in scaling without understanding the concurrency execution model
3. Inability to handle asynchronous, event-driven workloads
4. Inability to run code in response to events

Answer: 2

Which of the following is a use case for AWS Lambda?

1. Tasks that take more than 15 minutes to complete
2. Running code without worrying about underlying infrastructure
3. Deploying and managing containerized applications
4. Scaling and managing server infrastructure

Answer: 2

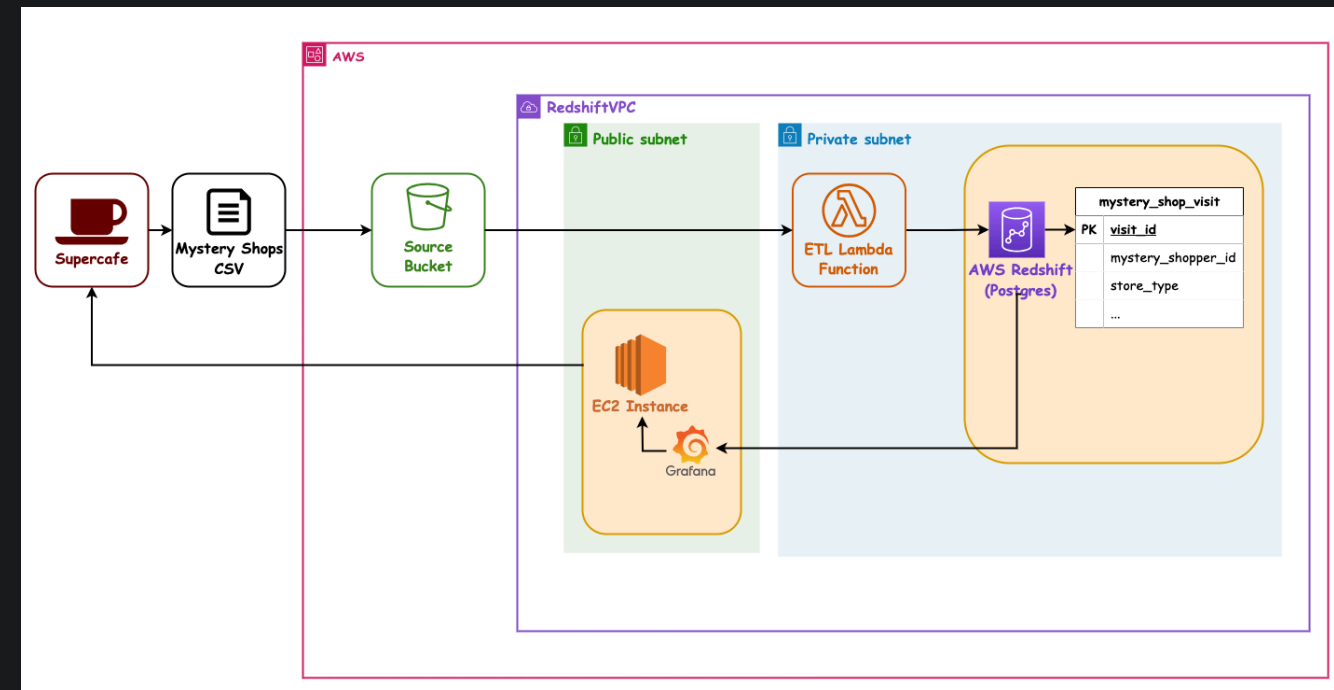
5. What is a "cold start" in the context of AWS Lambda?

1. The time it takes for a Lambda function to scale up
2. The time it takes to kick off an instance (container) to run a Lambda function
3. The process of initialising a new Lambda function
4. The process of stopping an unused Lambda function

Answer: 2

Proposed Pipeline Architecture

Lets revisit our Mystery Shopper target setup:



Our next user story

As a SuperCafe senior manager

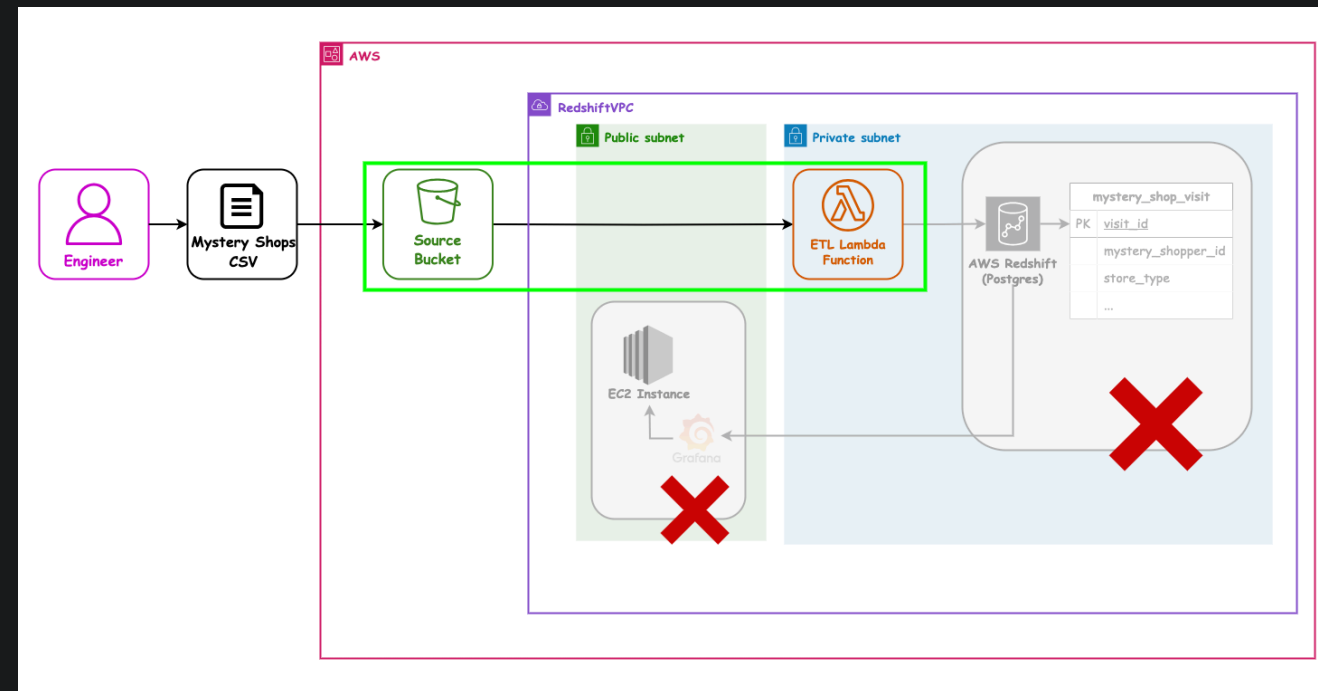
I want the Mystery Shopper data processed automatically

So that the data can be analysed

And the pipeline can run daily

Our next user story - Architecture

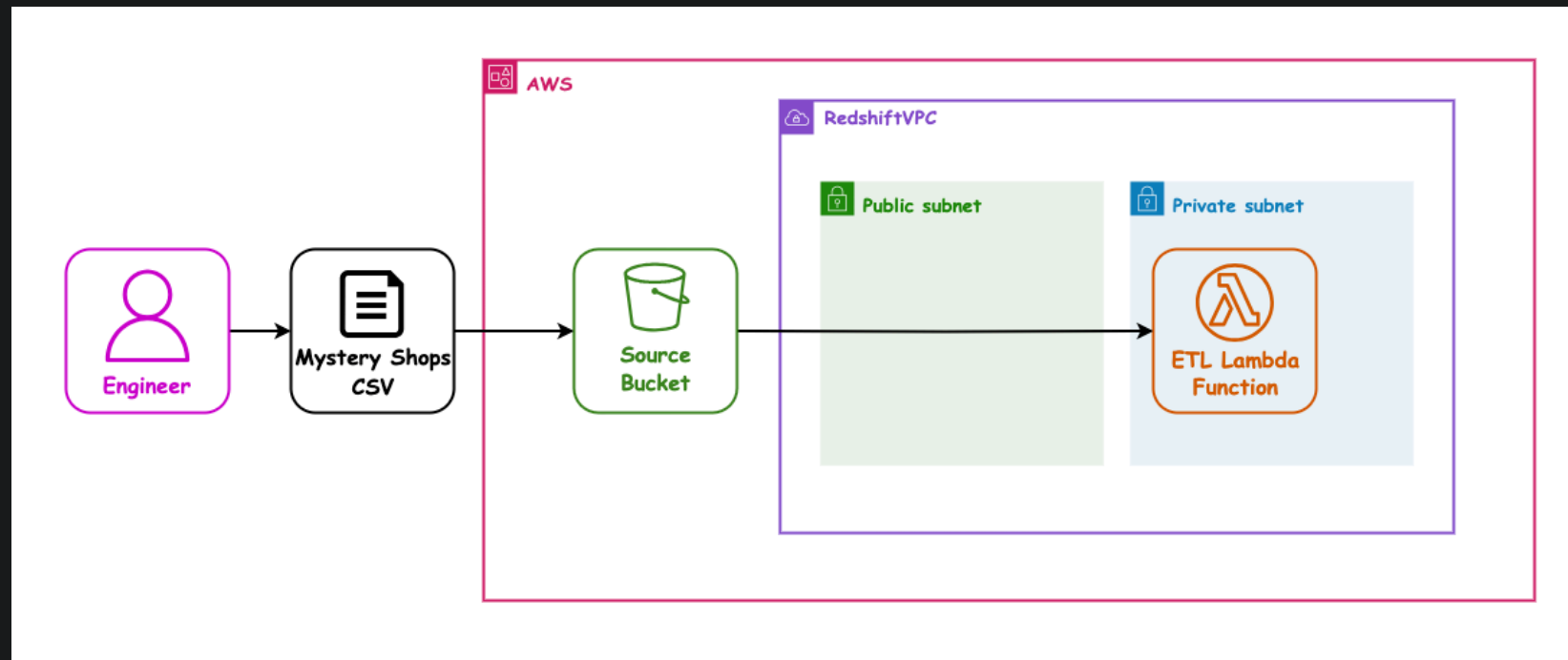
We need to learn about how Lambda works, so that next session we can do this "properly":



This session - we will code a Lambda, so next session we can set it up with IaC.

Our next user story - Architecture

So this session we only need to use these parts of AWS:



Code along - creating a lambda

Lets have a look at how to make a Python Lambda from scratch.

(Details on next slide).

Code along - creating a lambda 1

- Find the **Lambda** service
- Ensure you are in the correct region **eu-west-1**
- Click on "Create Function"
- Select "Author from scratch"
- Enter a function name e.g. **your-name-de-demo-lambda**
- Select the most recent Python Runtime version you can
 - Notice anything about the available versions?
- More on next slide...

Code along - creating a lambda 2

- Under **Change default execution role**, select **Use an existing role** and enter **nja-lambda-execution-role**
- Create the function
- Once the function is created, go to the **Configuration** tab and select **Tags** at the side
- Add a new tag with key **Name** and value of the function name

Code along - events 1

Lets have a look at how Events work and logging them.

- Make a new test event with the `Hello-World` template
- Save it for later
- Trigger the Lambda with your test event
- Check the logs in the Lambda page
- Click through to the logs in CloudWatch

Code along - events 2

Lets all log our `event` object.

- Add code to Log (`print()`) the event object
- (Re)deploy the lambda
- Trigger the lambda with your saved Test Event
- Check the CloudWatch logs now have more in them

Demo - configuration with env vars

Lets have a look at how Environment Variables ("env vars") work.

- **Never** use these for passwords!
- Add an env var e.g. `FAVOURITE_MOVIE` with a suitable value
- Add code to import `os`
- Add code to put the env var in a variable
 - e.g. `fave_movie = os.environ['FAVOURITE_MOVIE']`
- Add code to print the variable
- (Re)deploy the lambda
- Re-Test the lambda and recheck the logs

Code along - configuration with env vars

Lets all add an Environment Variable and use it in our code.

- Repeat the steps the Instructor just did for yourself
- **Never** use these for passwords!

Demo - updating the return value

The return value of a Lambda is used to indicate success/failure to the caller, and convey extra information.

- Demo making a basic Hello message JSON for the return value
- (Re)Deploy the lambda
- Re-Test the lambda
- Check the return value in the logs

Code along - updating the return value

Update your lambda return value to say Hello to yourself

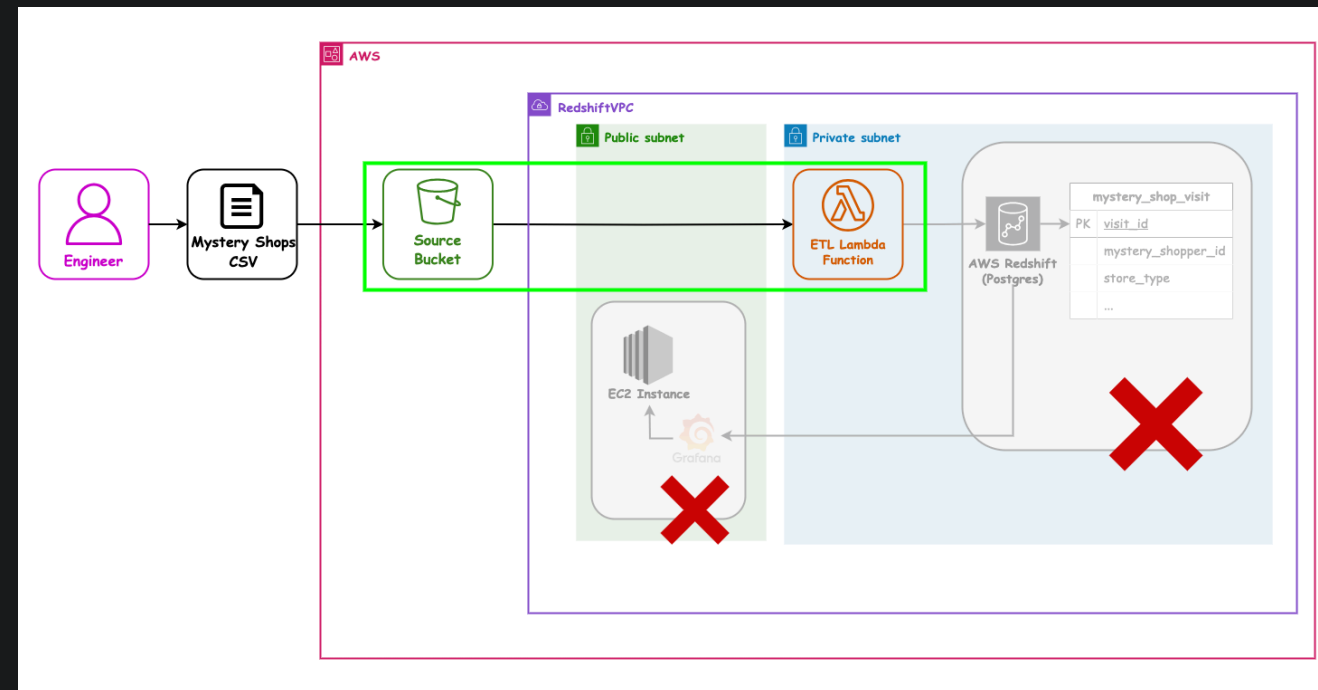
- Repeat what the Instructor just showed you
- (Re)Deploy the lambda
- Re-Test the lambda
- Check the return value in the logs

Sample handler code

A sample of what our code might now look at is in the [./handouts/sample_python_lambda.py](#) file.

Our next user story - Architecture

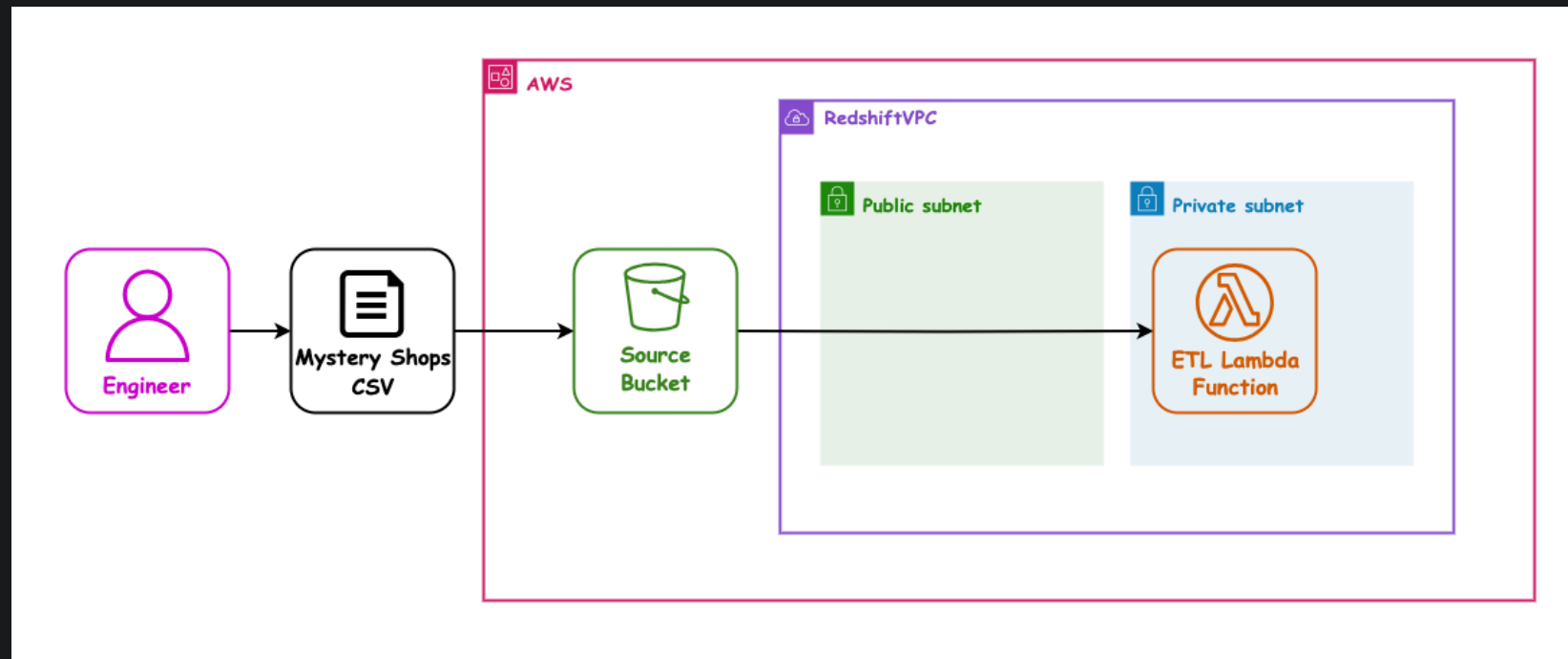
We have experimented with Lambda code, so next session we can do this with IaC:



This session - we will find out about Lambda, so next session we can set it up with IaC.

Our next user story - Architecture

This session we used these parts of AWS, and will again next session:



This session - we will find out about Lambda, so next session we can set it up with IaC.

Overview - recap

- Code running as a service
- Lambda as a form of Compute in AWS
- Creating a basic Lambda
- Event Triggers
- Testing lambdas
- Configuration including Environment Variables

Learning Objectives - recap

- What code as a service is
- What Serverless Applications are
- Python code structure in Lambdas
- How to Trigger an Lambda
- Using the Event object
- Configuration like Environment variables

Emoji Check:

On a high level, do you think you understand the main concepts of this session? Say so if not!

1. 🥲 Haven't a clue, please help!
2. 😞 I'm starting to get it but need to go over some of it please
3. 😐 Ok. With a bit of help and practice, yes
4. 😊 Yes, with team collaboration could try it
5. 😄 Yes, enough to start working on it collaboratively