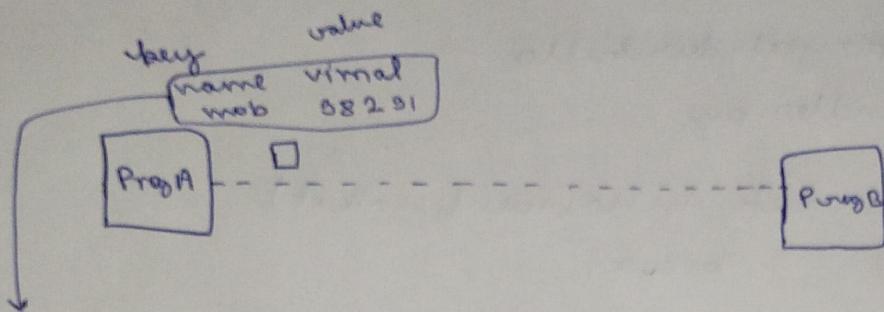


YAML :-

Yet Another markup language



we should have a particular format to send data, and same format should be understood by the receiver too.

JSON Notation

```

{
  name: vimal,
  mob: 1234
}
  
```

"we use curly braces here"

YAML notation :-

↳ `key : Value`
`x : 5` ← Assigning value to a variable

Array in YAML :-

List in python :-

```

name = ["Gragan", "Sonam", "Aman"]
  
```

YAML

Indentation

```

name:
  - "Gragan"
  - "Pop"
  - "Tom"
  
```

DAY-2: AWS :-

Name of lambda^m is same as service name.

Go inside directory directory file

aws-python-project-dev-hello.
↳ handler.py

serverless

Lambda
code

```
def hello(event, context):
    body = {
        "message": "Go serverless ...",
        "input": event
    }
    return {"statusCode": 200, "body": json.dumps(
        body
    )}
```

Changes made inside lambda is performed by
yaml serverless file

service: aws-python-project

frameworkVersion: '3'

provider:

- name: aws
 - runtime: python3.8
 - region: mumbai.ap-south-1

} we are asking to
make changes

functions:

visual:

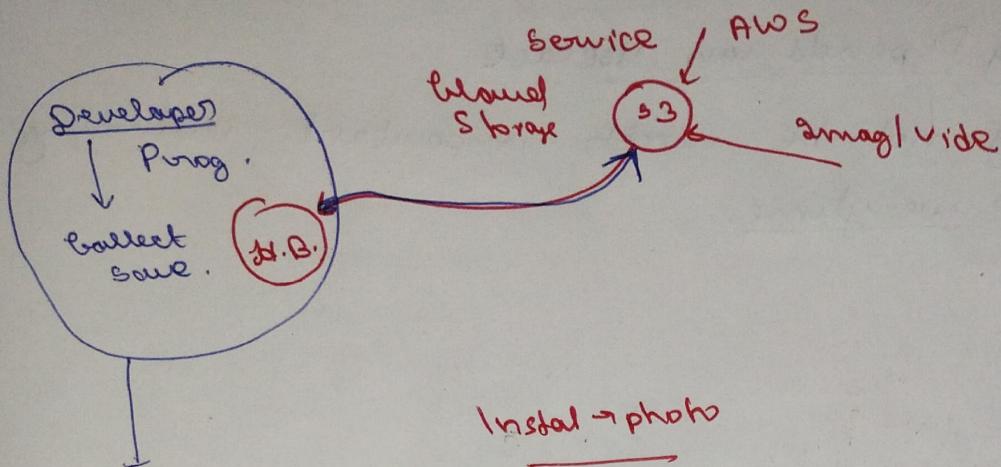
handler: myfile.visual.

④ Refer to official document for more
key words

provider:

~~stage~~
Stage: dev

functions:



This program will save the image in S3 &
S3 is managed by Aws guy.

* We never use manually a service, like we manually upload the pic

But prog. will automatically upload the code.

*

④ Developer write the lambda | serverless code.

functions:

virtual:

handler: virtual.myfunc
 ↓
 filename for name

lambda fn structure
 virtual.py
 {
 def myfunc()
 {

* \$ serverless deploy

Testing the app:

M2 serverless invoke --function virtual

M2 Graphical approach:

by Go to Lambda portal & do the changes

⑩ When to run if? { TRIGGERING }

↳ Running Depends on usecase

↳ It may not be worth important running the if everytime.

↳

Client

Lambda

GET

/data

API Gateway :-

→ Go to serverless.com & get some example code

Link to page :-

Edit the serverless file

functions:

vimal:

handler:

events:

- http:

| path: /data

| method: GET

↳ They will automatically
create API Gateway,

↳ Create
and connect the API
Gateway with this
Lambda.

④ serverless deploy

↳ They will sharel create an API Gateway URL now,

④ we can check by visiting the API gateway page.

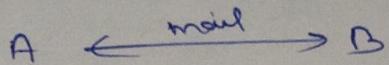
⑤ Serverless deploy - function vimal

↳ only focus on changes

↳ It won't recreate re-create
the entire setup.

Proj :-

"CREATING OWN MAIL SERVER"



- M1 :- \otimes EC2 service & install app app on its file
postfix

M2 :- Use or AWS service (SES) we for
the mailing related activity

Service Name:-

AMAZON SES

↳ Create identity using gmail.com

↳ Verify the email by logging in to

- \otimes Create 2 identities with diff email addresses

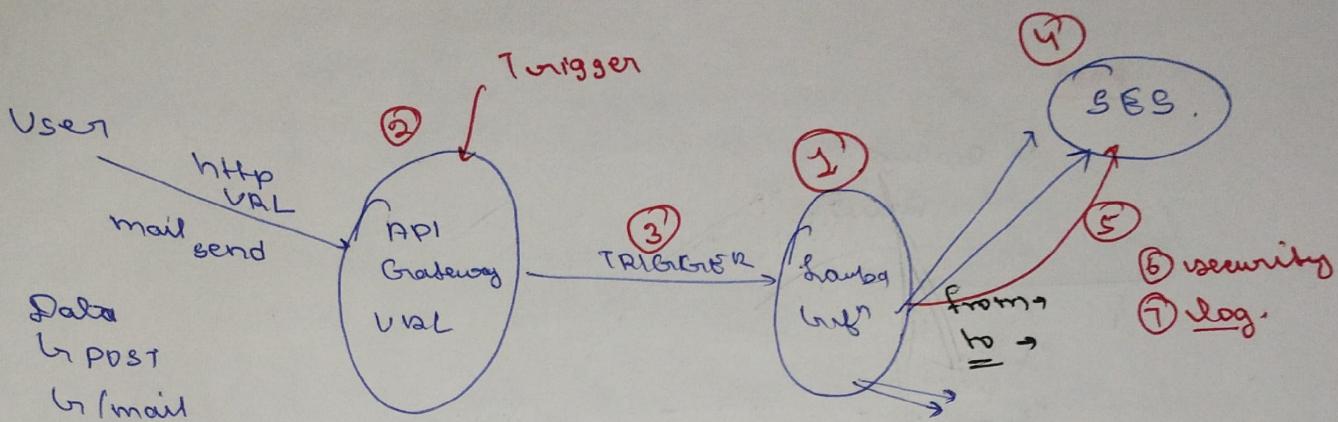
* Sending AWS SES :-

↳ Send Test Email

- \otimes You would have received mail in
another email

\$4 SDS ↵

AWS - Node.js - Standard.



* we want to pass the email via Lambda so that we can keep a count of it.

⑧ Code for lambda :-

Get it from the workshop's github page.

↳ Create new SES project →

YAML :-

functions:

createContact:

handler: handler.createContact.

http events:

⑨ By default, the data is received w/ inside ~~with~~ the event keyword.

④ Send mail using ~~postman~~ Postman.

run `aws command` & get the endpoint.

POST

- ↳ Body
- ↳ Raw.

{

"to":

"from":

"Subject": "Inv from lambda",

"message": "Awesome project"

}

⑤ Test {

: "message failed to send"

} : success.



☞ TROUBLESHOOTING FAIL:-

↳ error in lambda code.

↳ Go to monitor

↳ our lambda fn wasn't called.

Lambda

- ↳ record
- ↳ log.

AWS

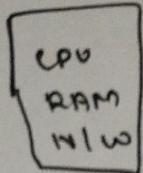
↳ CloudWatch (monitoring service)

↳ log.

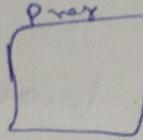
↳ all the things going on
our AWS account is
recorded here,

Cloud Watch

Metric



client

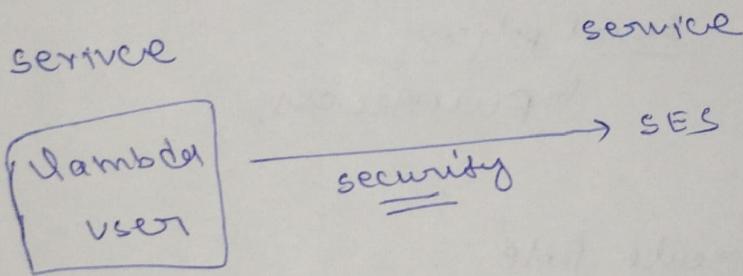


"Program related data is called logs."

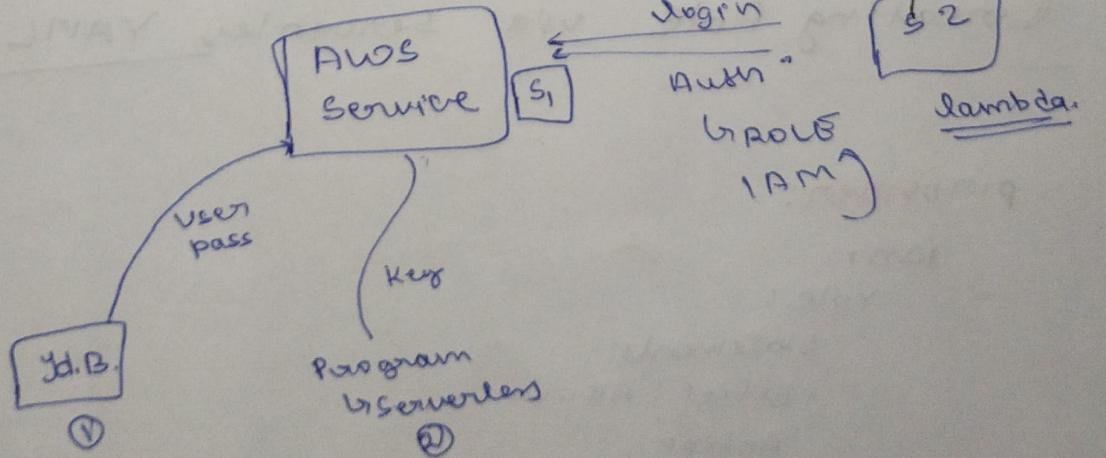
④ Cloud watch are the monitoring tools used

⑤ we can see all the logs in "View logs in cloud watch"

⑥



⑥ ~~this~~ SES won't allow any other service to access its resources.



Role :- When one service tries to use another service.

Creating role via ~~serverless~~ Web UI

↳ Role can be ~~not~~ created from usecase to usecase.

↳ Select sys service eg. Lambda that will access 2 blocks Next

↳ Select sys service whose service is to be accessed.

Select policy

↳ FullAccess.

↳ Create Role.

↳ Now we will find that power & permission have been attached to the "Lambda".

Creating Role via ~~serverless~~ YAML

provider:

```
iam:  
  role:  
    statements:  
      - Effect: Allow  
        Action:  
          - "sts:*"
```

```
Resource:  
  - "x"
```