ARDUINO

SERVERLESS WAR STORIES II

TURBO

press start

Nome: **Bob**

Professione: **Junior Consultant**

Azienda: **SprintInc (20 dipendenti)**

Esperienza: **2 anni embedded + 2 anni AWS**

Attivita’ svolte:

Sviluppo tool di devops interni

Qualche piccolo progetto per clienti

Ultimamente addirittura R&D interna!

Fino a quando un giorno……

SprintInc CEO:

- Bob! Il nostro DevOps! Abbiamo una interessante opportunita’ per te!

SprintInc Manager:

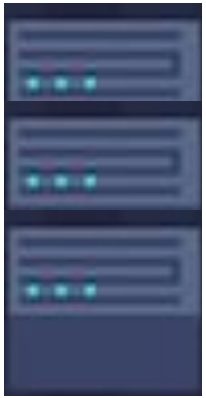
- Bob e’ scoppiata la bomba: **MegaPawaPetrolCorp** vuole (per ieri) per la sua megamigrazione:
  -
    1. Gestione costi e fatturazione del Cloud
    2. Riduzione costi su VM Cloud

SprintInc Manager:

- Ovviamente dovrai gestire tutto tu perche per prendere questo progetto abbiamo proposto un costo di 50 rupie

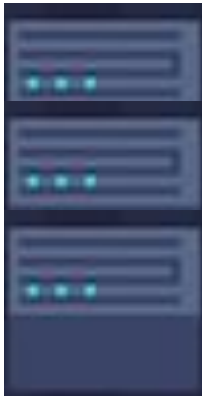
- Inoltre non dimenticarti che domani sei a Firenze per inziare riunioni per altri 15 progetti

- Ah, e visto che sono le 18.30 prepara giusto 35345453 slide per stasera



Capitolo I:

approccio “**serverful**” su **AWS**

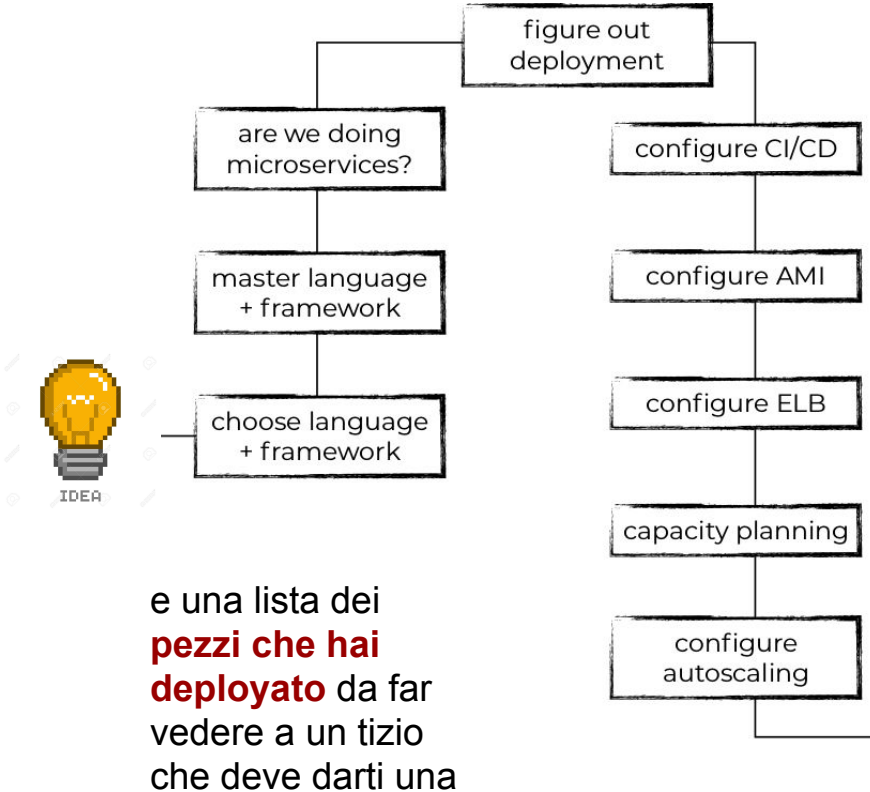


Ok ma...

e il **logging**?

e il **monitoraggio**?

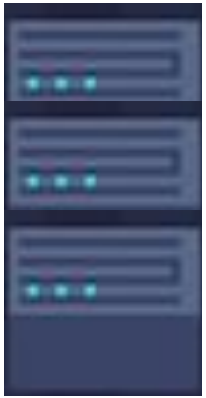
Si ma e i **test**?



e una lista dei **pezzi che hai deployato** da far vedere a un tizio che deve darti una mano?

E quando gli aws account
da 4 (**500 EC2**)
sono diventati
30 (**10000 EC2**)
che hai fatto?


SCORE
$300
STAGE
5-3
TIME 197
P
E

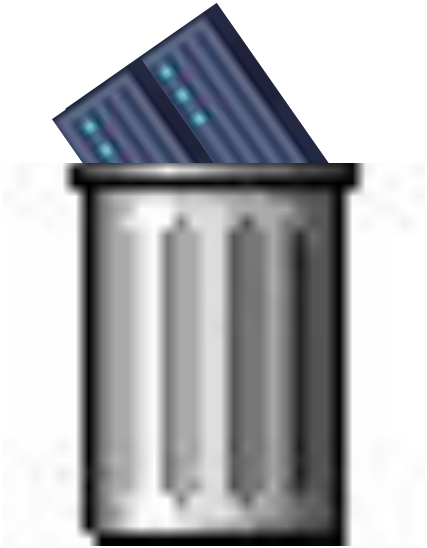


Upgrading your Linux distro
does not provide value to users.

Managing your RabbitMQ servers
does not provide value to users.

Configure your log rotation mechanism **does not provide value to users.**

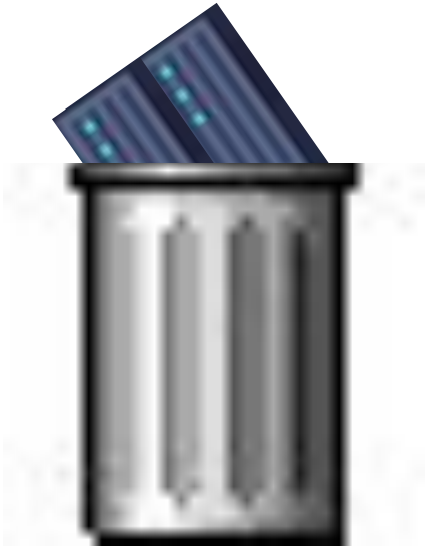
Managing the Security Updates for your fleet
does not provide value to users



Shipping products provides value to users.

focusing your efforts on what provides value to users.

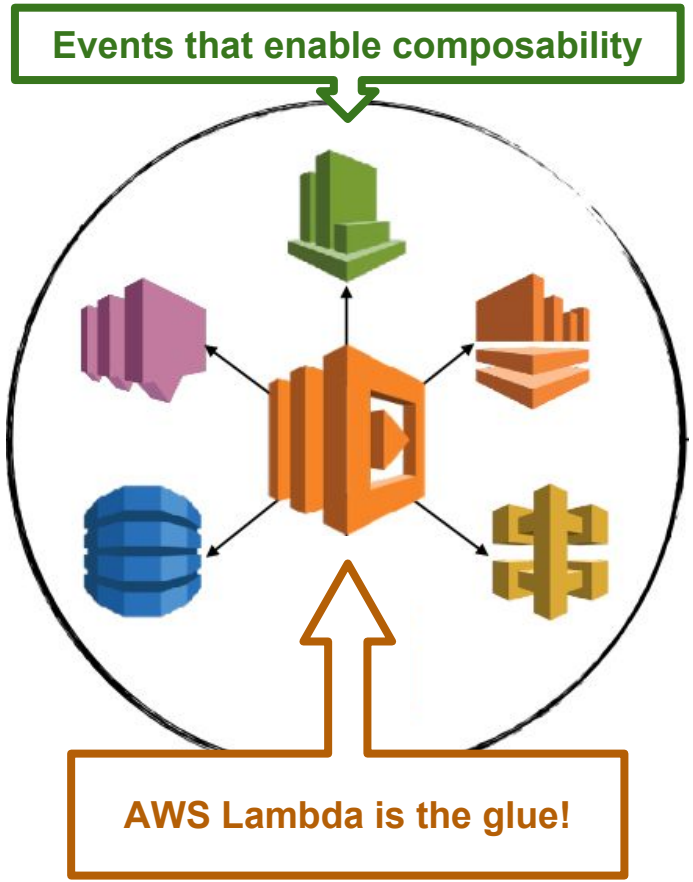
the serverless maxim
focus on business logic, not servers.

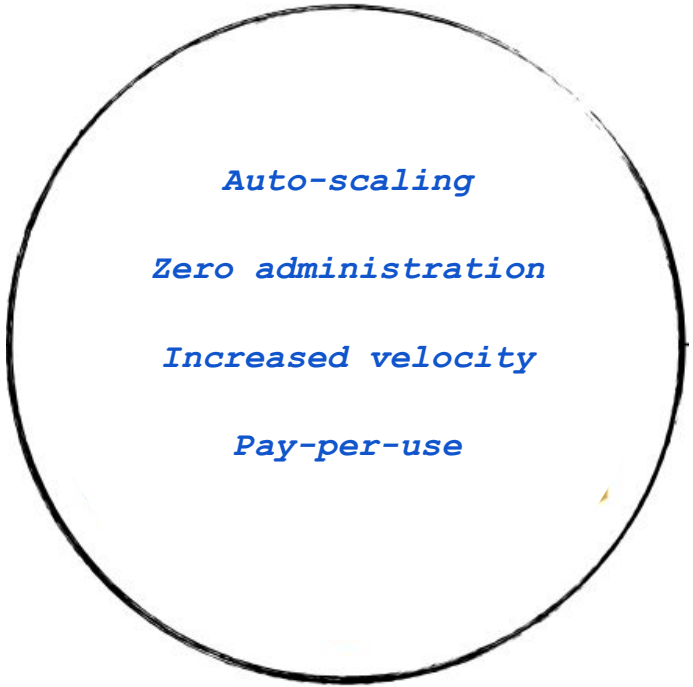


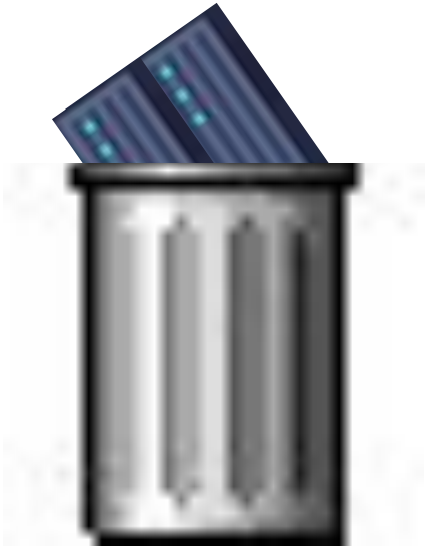
Capitolo II:

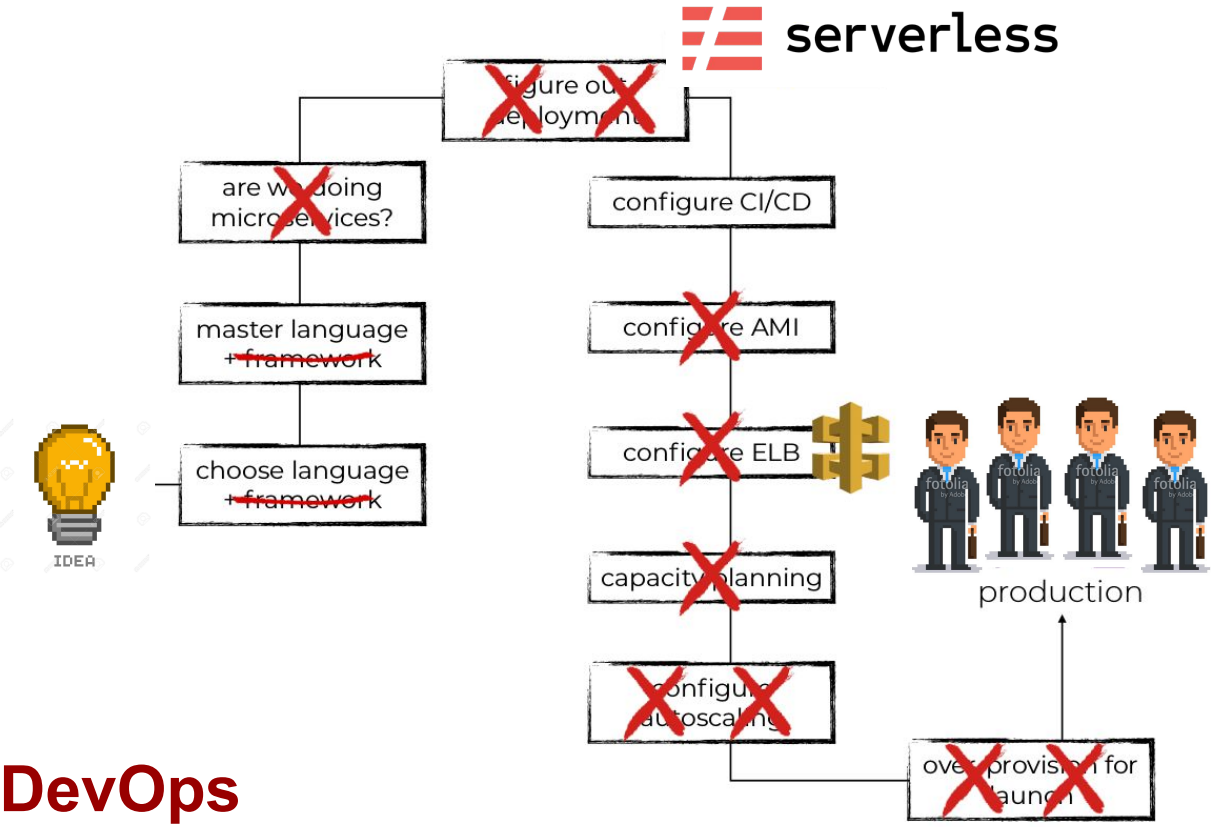
approccio “**serverless**” su **AWS**

What makes an application serverless?



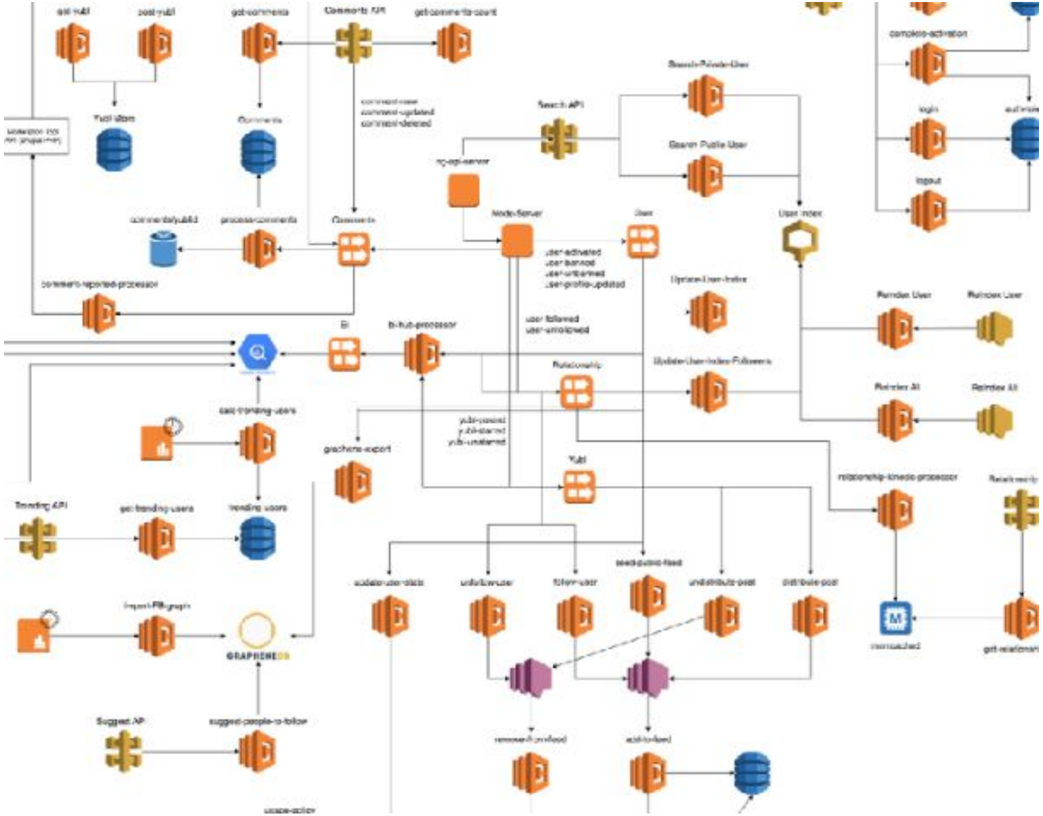






NoOps > DevOps

```yaml
service: serverless-rest-api-with-dynamodb

frameworkVersion: ">=1.1.0 <2.0.0"

provider:
  name: aws
  runtime: nodejs6.10
  environment:
    DYNAMODB_TABLE: ${self:service}-${opt:stage, self:provider.stage}
  iamRoleStatements:
    - Effect: Allow
      Action:
        - dynamodb:Query
        - dynamodb:Scan
        - dynamodb:GetItem
        - dynamodb:PutItem
        - dynamodb:UpdateItem
        - dynamodb:DeleteItem
      Resource: "arn:aws:dynamodb:${opt:region, self:provider.region}:*:table/${self:provider.environment.DYNAMODB_TABLE}"

functions:
  create:
    handler: todos/create.create
    events:
      - http:
          path: todos
          method: post
          cors: true

  list:
    handler: todos/list.list
    events:
      - http:
          path: todos
          method: get
          cors: true

  get:
    handler: todos/get.get
    events:
      - http:
          path: todos/{id}
          method: get
          cors: true

  update:
    handler: todos/update.update
    events:
      - http:
          path: todos/{id}
          method: put
          cors: true

  delete:
    handler: todos/delete.delete
    events:
      - http:
          path: todos/{id}
          method: delete
          cors: true

resources:
  Resources:
    TodosDynamoDbTable:
      Type: 'AWS::DynamoDB::Table'
      DeletionPolicy: Retain
      Properties:
        AttributeDefinitions:
          -
            AttributeName: id
            AttributeType: S
        KeySchema:
          -
            AttributeName: id
            KeyType: HASH
        ProvisionedThroughput:
          ReadCapacityUnits: 1
          WriteCapacityUnits: 1
        TableName: ${self:provider.environment.DYNAMODB_TABLE}
```

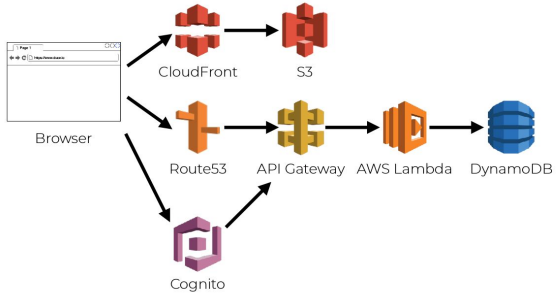


get-comments
Comments API
get-comments-count
Comments
process-comments
comments-reported-processor
Search API
Search Private User
Search Public User
Node-Server
User
User Index
complete-activation
login
logout
user-activated
user-banned
user-unbanned
user-profile-updated
user-followed
user-unfollowed
Update-User-Index
Update-User-Index-Followers
Reindex User
Reindex All
Relationship
BI
bi-hub-processor
calc-trending-users
graphene-export
Trending API
get-trending-users
trending users
relationship-kinesis-processor
follow-user
unfollow-user
distribute-post
GRAPHENEDB
Suggest API
suggest-people-to-follow
add-to-feed
memcached

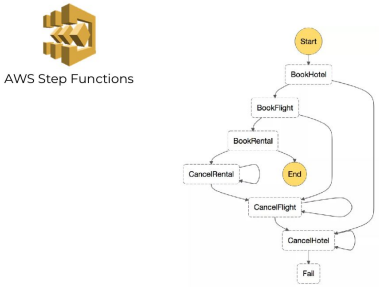
Use cases

cron

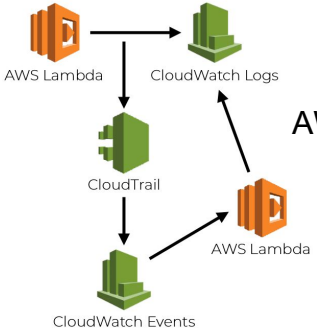
Web App



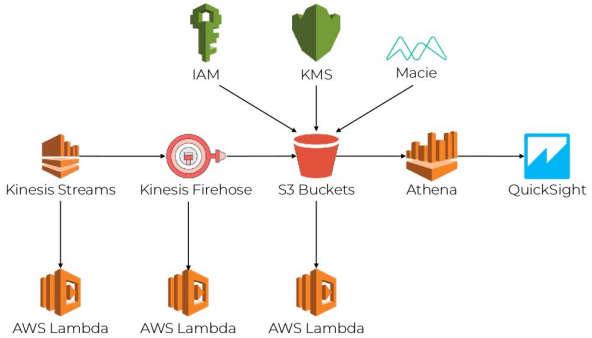
Workflow



AWS ops automation



Data Analysis



Best of Breed

amazon web services

serverless

python

Boto 3

AWS
Serverless Framework
Python3.6

Grazie!