

The Rise of Mobility



Guido
Nebiolo



Matteo
Madeddu



- Il Problema
- La Soluzione
- Core
- API
- Sviluppi
- Q&A

Il Problema

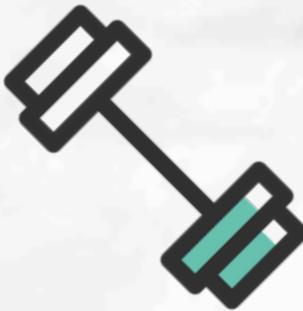
Questo è... {{your_name}}



Matteo viaggia molto



Non è troppo in forma



Non è troppo in forma



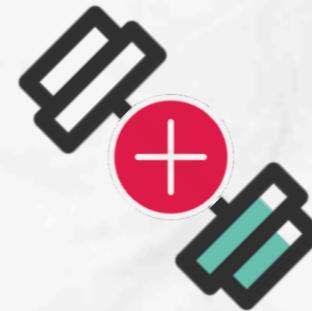
E' sempre nelle curve



Lancia spesso i pm i



In pratica, Matteo
ha un po' di guai da risolvere



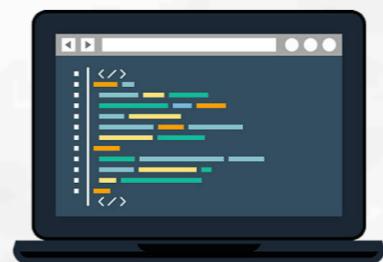
In pratica, Matteo
ha un po' di guai da risolvere



Idea: portable code
environment in cloud!



Idea: portable code
environment in cloud!



Idea: portable code
environment in cloud!



e IoT Button
come Laptop



La soluzione

Soluzione

Implementazione



Button



API



Core













La soluzione
Core

Soluzione

Implementazione



Button



API



Core

Visual Code Studio - Server

github.com/cdr/code-server

Creazione dell'ambiente
di sviluppo

Step 1: Creazione Della Macchina

MyCodeServerInstance:

Type: "AWS::EC2::Instance"

Properties:

ImageId: String	# OS image to run on instance
InstanceType: String	# CPU, Ram, network, etc
KeyName: String	# KeyPair name
SubnetId: String	# Subnet ID
UserData: String	# Run commands
MeteData: String	# cfn-init

Volumes:	# Volume attached
- Volume	

Step 2: Configurazione dell'Ambiente di Sviluppo

```
UserData:                                # UserData section
  !Base64:
    !Join:                                # Steps to run the init session
      - ""
      - "#!/bin/bash -xe\n"
      - "/opt/aws/bin/cfn-init -v "
      - "          --stack "
      - !Ref AWS::StackName
      - "          --resource CodeServerPublicInstance"
      - "          --configsets end-to-end"
      - "          --region "
      - !Ref AWS::Region
      - "\n"

Metadata:                                # MetaData section
  AWS::CloudFormation::Init:
    configSets:
      end-to-end:
        - Install                         # Steps to install packages
        - Setup                           # Steps to setup development environment
        - Service                          # Steps to run the service

Install:
  packages:
    yum:
      docker: []                      # Install base packages
      wget: []
      gzip: []
      git: []
      ....
```

```
# Steps to setup development environment
```

```
.....
```

```
Setup:
```

```
# Retrieve of secrets from AWS vault
```

```
files:
```

```
  "/etc/pki/CA/private/cert1.pem":
```

```
    content:
```

```
      Fn::Join:
```

```
      - ""
```

```
      - - ""
```

```
      - !Ref CertificateParameterName
```

```
    mode: "000600"
```

```
    owner: root
```

```
    group: root
```

```
  "/etc/pki/CA/private/privkey1.pem":
```

```
    content:
```

```
      Fn::Join:
```

```
      - ""
```

```
      - - ""
```

```
      - !Ref PrivateKeyParameterName
```

```
    mode: "000600"
```

```
    owner: root
```

```
    group: root
```

```
# Steps to setup development environment
```

```
.....  
Setup:
```

```
.....  
# Commands to install environment  
commands:
```

```
020-wget:
```

```
command: wget https://github.com/cdr/code-server/releases/  
download/1.1156-vsc1.33.1/code-server1.1156-vsc1.33.1-linux-  
x64.tar.gz  
cwd: "~"
```

```
030-untar-cleanup:
```

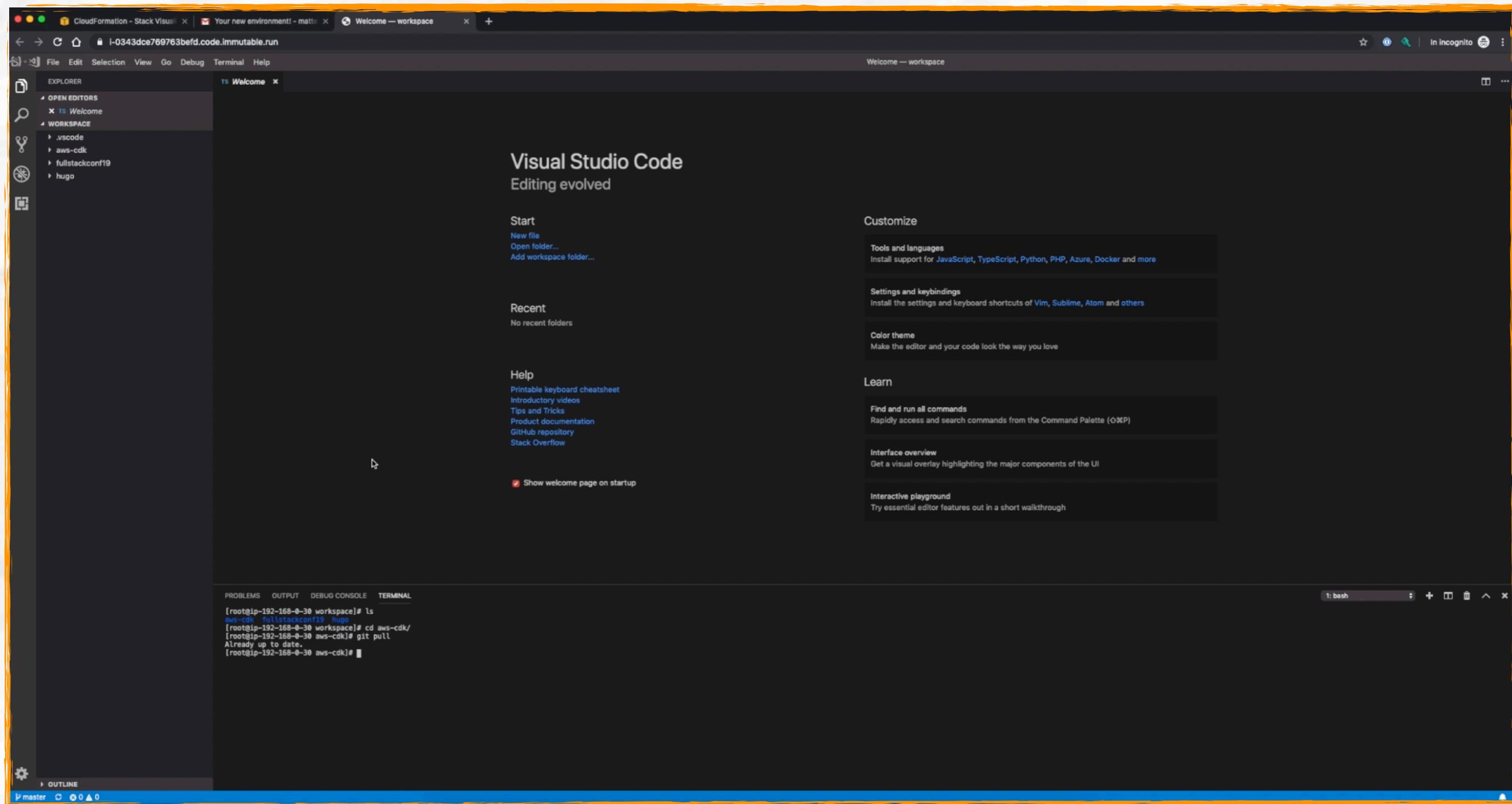
```
command: tar -xvzf code-server1.1156-vsc1.33.1-linux-  
x64.tar.gz && cp code-server1.1156-vsc1.33.1-linux-x64/code-  
server /usr/bin/ && rm -rf code-server1.1156-vsc1.33.1-linux-  
x64*  
cwd: "~"
```

```
040-chmod-bin:
```

```
command: chmod +x /usr/bin/code-server  
cwd: "~"
```

```
.....
```

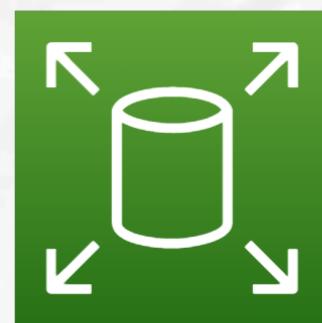
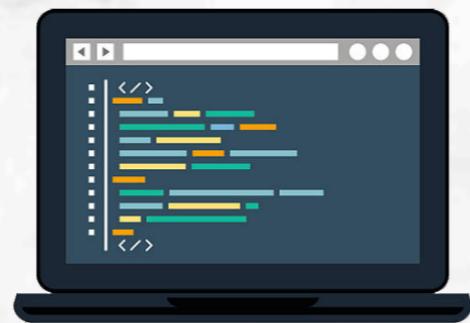
Step 3: Accesso Web



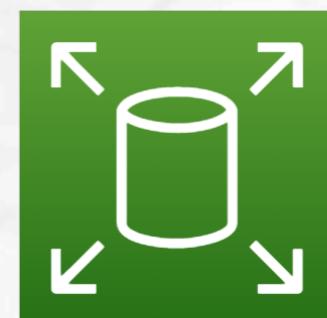
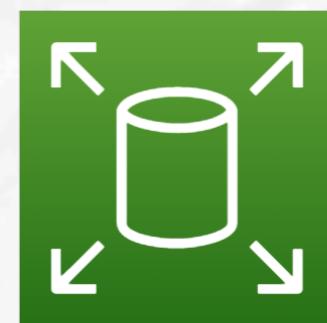
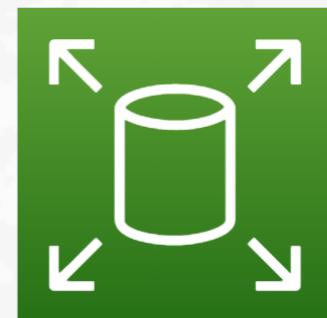
Parametri

- Chiavi SSH
- Configurazione Certificati
- Password di Accesso
- Domini
- Volumi

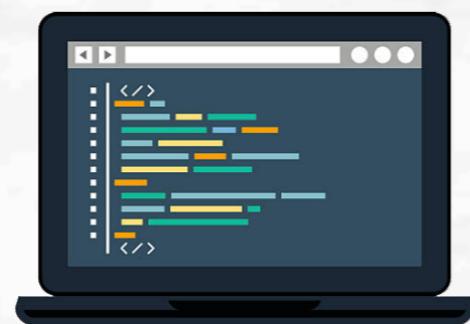
Un volume



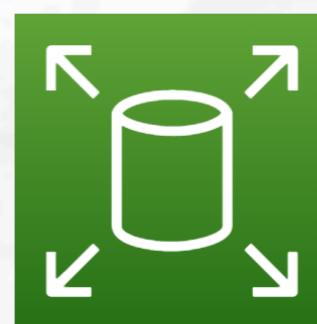
Più volumi



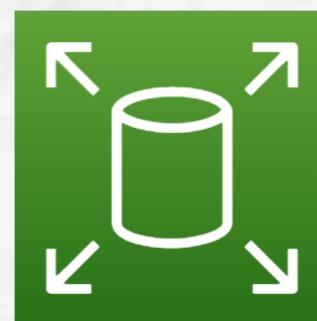
Diversi progetti



workspace:/
project A

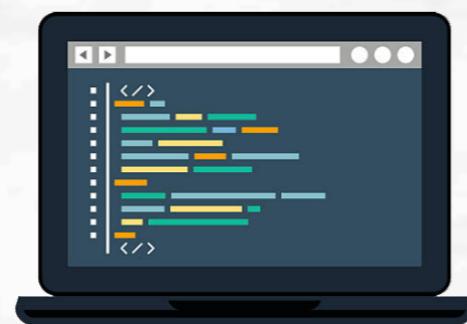


workspace:/
project B

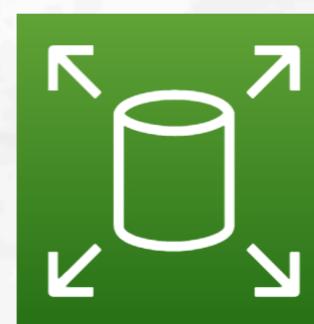


workspace:/
project C

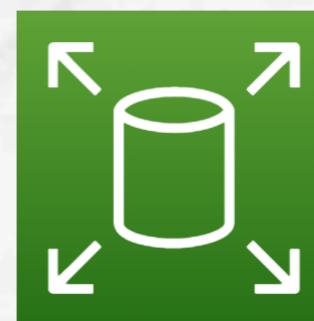
Diversi clienti



workspace:/
customer 1



workspace:/
customer 2

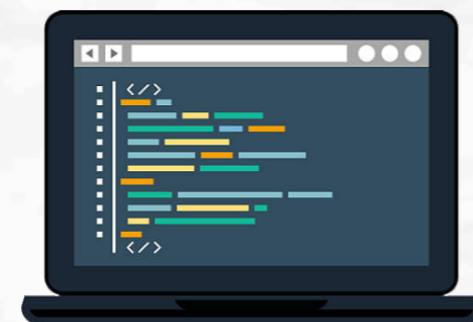


workspace:/
customer 3

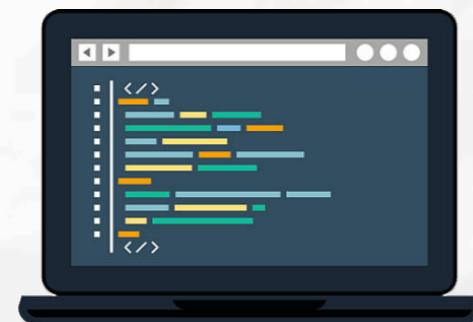
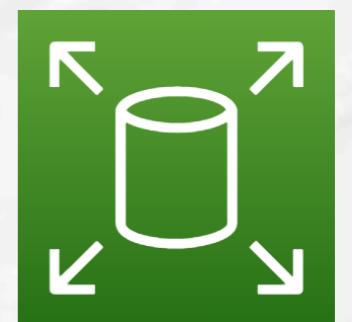
Diversi scopi



small type:/
dev env

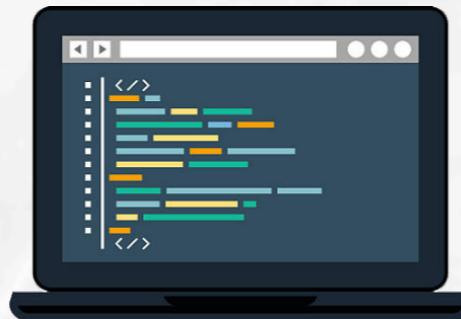


medium type:/
test env

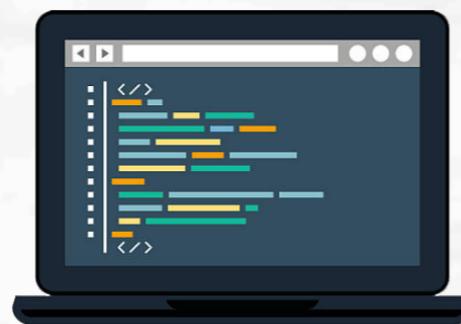


data type:/
prod env

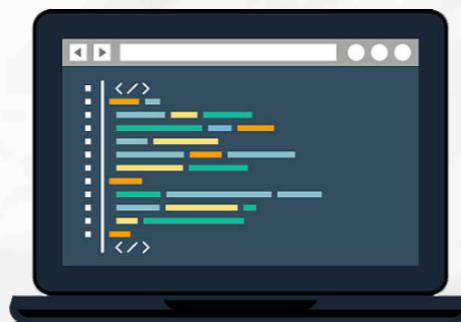
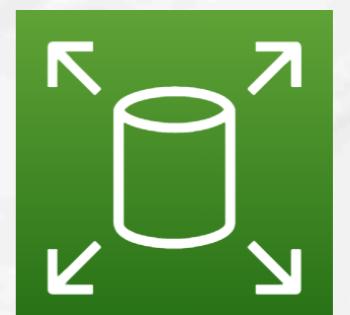
Diversi scopi



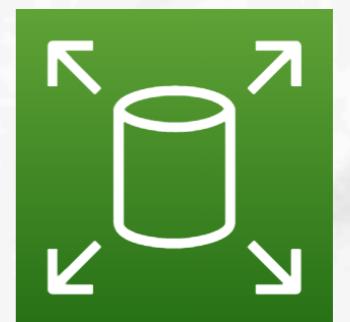
small type:/
dev env



medium type:/
test env



data type:/
prod env



La soluzione
API

Soluzione



Button

Implementazione



API

Core

Soluzione

Implementazione



Button



API



Core

Soluzione

Implementazione



Button

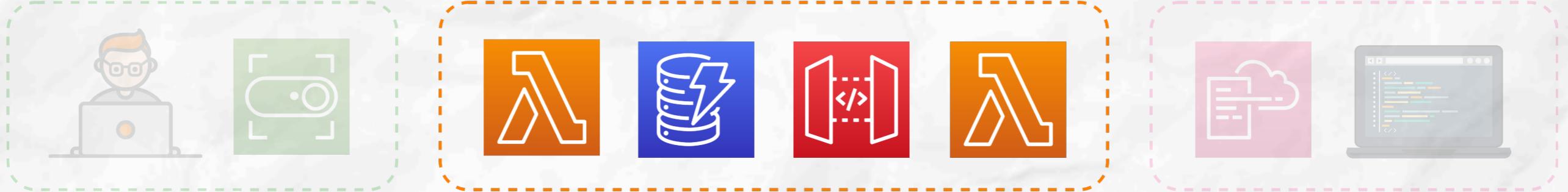


API



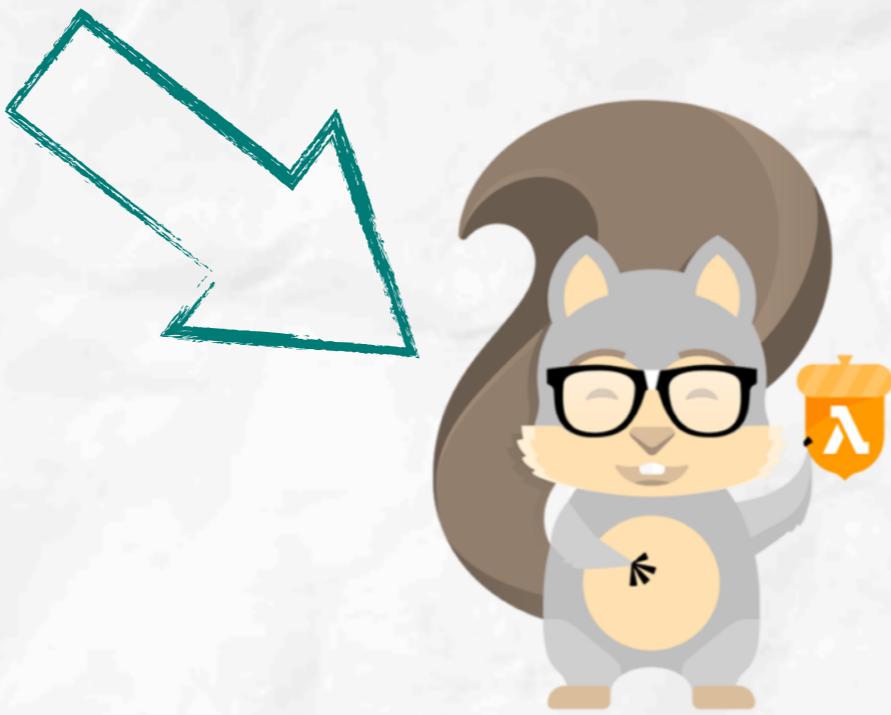
Core

Demo!



API

Questo è... SAM





.....

Resources:

API:

Type: AWS::Serverless::Api

Properties:

StageName: !Ref Stage

Auth:

DefaultAuthorizer: AWS_IAM

InvokeRole: NONE

Microservice:

Type: AWS::Serverless::Function

Properties:

CodeUri: MicroserviceAPI/

Handler: index.handler

Runtime: nodejs10.x

.....

Lambda Button

```
def launch_stack(configuration_item):

    stackdata = None

    # fingers crossed
    try:

        # generate a `minimum_password_length` random password if not
        # defined in configuration item parameter
        configuration_item["app_password"] = "".join(
            random.choices(string.ascii_letters + string.digits,
                           k=minimum_password_length)) \
            if "app_password" not in configuration_item.keys() \
            else configuration_item["app_password"]

        # map configuration item parameters to stack parameters
        parameters = list(map(lambda x: {
            "ParameterKey": f"{snake_to_camelcase(x)[0].upper()}{snake_to_camelcase(x)[1:]}",

            "ParameterValue": configuration_item[x]
        }, configuration_item.keys()))

        response = microservice_api.create_environment(parameters)
        stackdata = response.text

    except Exception as e:
        logging.error(str(e))

    return stackdata
```

Lambda API

Lambda API

github.com/jeremydaly/lambda-api

```
const api = require('lambda-api')();
const _ = require('lodash')
const stackLib = require('./stack')

api.post('/environments', async (req, res) => {
  // Check if a body is provided
  if (req.body === undefined || req.body == "") {
    throw Error("No required stack parameters specified.");
  }
  // Isolate required UserId property and throw error if not defined
  var userId = _.get(_.find(req.body, (parameter) =>
    parameter.ParameterKey == "UserId"), "ParameterValue", undefined);
  if (userId === undefined || userId == "") {
    throw Error("No user id specified.");
  }

  return await stackLib.createStack(userId, req.body);
})

api.delete('/environments/:environment', async (req, res) => {
  return await stackLib.deleteStack(req.params.environment);
})

api.get('/environments/:environment', async (req, res) => {
  result = await stackLib.getStack(req.params.environment);
  if (condition) res.error(404, 'Not found');
  else return result;
})

module.exports = api
```

Demo

Siluppi

Frontend per amministrazione

ALEX THOMPSON

Dashboard

Billed Hours per Day

↑ 55% increase in today usage.

🕒 updated 4 minutes ago

Daily Bill \$6,24

Last 24 Hours

Global User Space 23/500 GB

Get More Space...

Connected Users 4

Updated 3 minutes ago

Your Environments

ID	Family	Cost	Actions
i-0009ace46faa3dd1e	t3.small	\$4,342	Oud-Turnhout
i-00114f400ed271d12	t3.small	\$4,342	Oud-Turnhout
i-00144861ba4a87f84	t3.small	\$4,342	Oud-Turnhout
i-002079b2d1cb3b6a1	t3.small	\$4,342	Oud-Turnhout
i-0035e4a725ffdc125	t3.small	\$4,342	Oud-Turnhout
i-0046e4f725eecc142	t3.small	\$4,342	Oud-Turnhout

Create New Environment

Instance Family

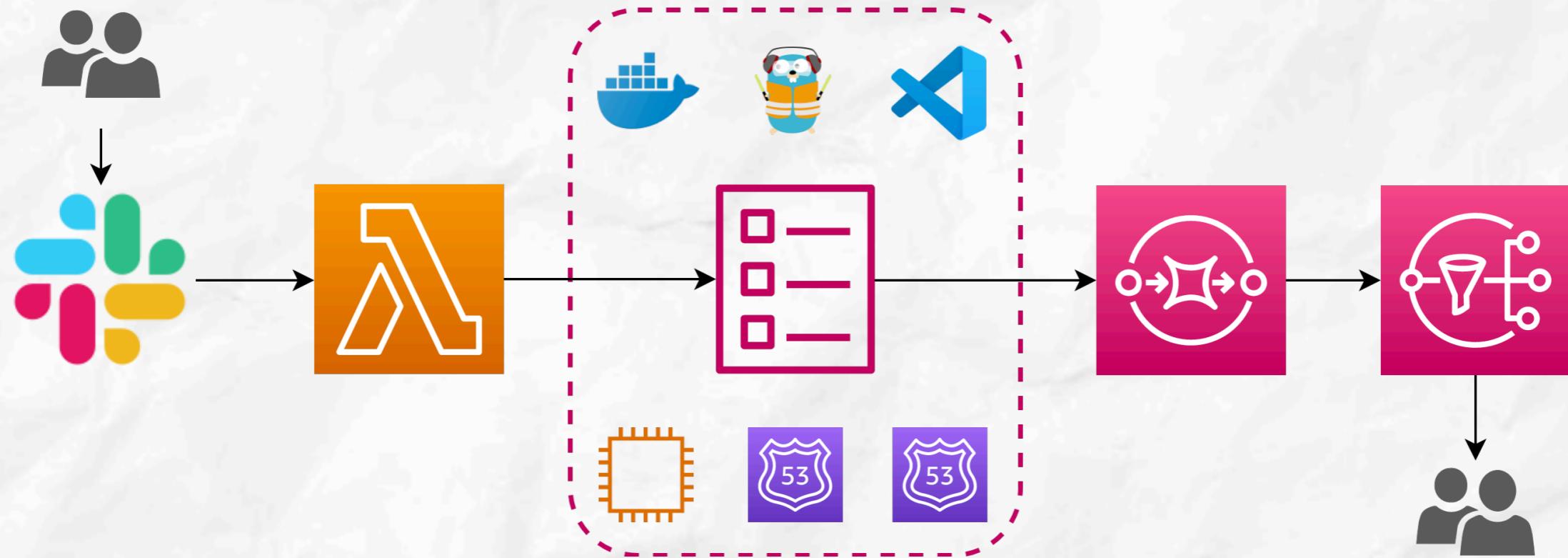
Workspace (Optional)

Disk Size in GB (Optional)

30

CREATE

Chatbot & Multitenancy



Codice, materiali, slide

github.com/made2591/fullstackconf19



Guido
Nebiolo



Matteo
Madeddu



Grazie

Q ≠ A