

EC2 instance setup

Video link for reference : <https://www.youtube.com/watch?v=qUtg59N6ph>

how to setup EC2 instance :

1. select ubuntu machine for ami:

Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-a9d276c9

choose instance type .

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

3. configure instance :

Number of instances : 1

4. Step 4: Add Storage

Volume Type ; Root

Device : /dev/sda1

Snapshot : snap-826344d5

Size (GiB) : 8

Volume Type ; General Purpose SSD (GP2)

IOPS : 100 / 3000

5. security group

group name : spotcrunch-node

Description : launch-wizard-4 created 2016-11-25T10:30:26.875+05:30

type : ssh protocol : tcp port range : 22

Set following inbound ips for security group of created instance :

HTTP	TCP	80	0.0.0.0/0
Custom TCP Rule	TCP	9000	0.0.0.0/0
SSH	TCP	22	206.183.111.25/32
SSH	TCP	22	122.106.156.73/32
Custom TCP Rule	TCP	3002	0.0.0.0/0
Custom TCP Rule	TCP	9005	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0

sshkey : spotcrunch_user
pem file : spotcrunch_user.pem

// now instance is created with following id

Instance ID created is :

i-fb10776f

6) create elastic ip : eg. foll

public dns : http://ec2-35-163-78-247.us-west-2.compute.amazonaws.com

elastic ip : 35.163.78.247

private ip : 172.31.45.25

RDS setup :

DB name for production : spotcrunchglobal

username : crunchmaster01

password : Sp0tcrunCH01

endpoint : seesharecrunch.crs3fmrqbgr5.ap-southeast-2.rds.amazonaws.com

databasename for stg : spotcrunchdbstg

username : crunchmaster01

password : Sp0tcrunCH01

endpoint : seesharecrunch.crs3fmrqbgr5.ap-southeast-2.rds.amazonaws.com

login to mysql server from command line :

```
mysql -h seesharecrunch.crs3fmrqbgr5.ap-southeast-2.rds.amazonaws.com -u crunchmaster01  
-p
```

command to view all previliages granted to user :

```
> show grants for 'crunchmaster01';
```

How to edit database connection details

Path for datasources.json : /var/www/html/zapdish/stg/loopback/server/datasources.json

```

"mysqlDs": {
  "host": "localhost",
  "port": 3306,
  "database": "flikbuzzapp-dev",
  "password": "*****",
  "name": "mysqlDs",
  "user": "root",
  "connector": "mysql",
  "createDatabase": true
},

```

For changing username update 'user' key value and for password update 'password' key value

For Email smtp details update following details in datasources.json file

```

"emailDS": {
  "name": "emailDS",
  "connector": "mail",
  "transports": [
    {
      "type": "smtp",
      "host": "smtp.wwindia.com",
      "secure": false,
      "port": 25,
      "tls": {
        "rejectUnauthorized": false
      },
      "auth": {
        "user": "Test1@wwindia.com",
        "pass": "abc123!@#"
      }
    }
  ]
}

```

For Changing zomato api key in config.json file

Path : var/www/html/zapdish/stg/loopback/server/config.json

```

"zomato": {
  "user_key": "759dc0b51d8f108efd14315d90bf537c"
},
"strip": {
  "client_api_key": "sk_test_hE50MM6Ez2I88cTtmVDR LHGu"
},

```

Push notification :

For sending push notification on **android**, we are using firebase. And its configuration defined var/www/html/spotcrunch/stg/loopback/server/credentials/zapdish-firebase-adminsdk-gka8y-5213ae4627.json

For sending push notification on **ios** device, add pem files at following location

Path : var/www/html/spotcrunch/stg/loopback/server/credentials/apns_cert_dev.pem

Path : var/www/html/spotcrunch/stg/loopback/server/credentials/apns_cert_key.pem

Amazon images container's name

We have container's name mentioned in config.json file

```
"AWSS3Buckets": {  
  "product_images": "spotcrunch-local",  
  "bucket_name": "spotcrunch-local",  
  "s3bucket_url": "https://s3.ap-south-1.amazonaws.com/spotcrunch-local/",  
  "containers": {  
    "flick": "flick_images/",  
    "product": "product_images/",  
    "restaurant": "restaurant_images/",  
    "app_user_pics": "profile_pics",  
    "promotion_pic": "promotion_pic",  
    "cuisine_pic": "cuisine_pic",  
    "product_feature_image_pic": "product_feature_image_pic"  
  }  
}
```

Amazon s3 bucket

```
"AWSConfig": {  
  "accessKeyId": "AKIAI7QC4M3LCW3XAS3Q",  
  "secretAccessKey": "Yg809aNUfilrq/t4KRN9fxTt4dHuacubR32OAOu",  
  "region": "us-west-2",  
  "signatureVersion": "v4"  
}
```

Angular js code constants

Path : /var/www/html/zapdish/stg/angular/client/app/core/core.constant.js

```
"baseUrl" : "http://13.55.91.237:9005/stgapi",
'rootUrl' : "stg.spotcrunch.com",
"awsS3Bukcet" : {
    "product_images" : "flickbuzzlocal",
    "bucket_name" : "spotcrunch-stg",
    "s3bucket_url" : "https://s3-ap-southeast-2.amazonaws.com/spotcrunch-stg/",
    "containers": {
        "flick": "flick_images/",
        "product": "product_images/",
        "restaurant": "restaurant_images/",
        "app_user_pics": "profile_pics",
        "promotion_pic" : "promotion_pic",
        "cuisine_pic": "cuisine_pic/",
        "product_feature_image_pic": "product_feature_image_pic/"
    }
},
"stripe":{
    "key":{
        "api":"sk_test_eb5QZRTBuntQSTA45gwigIX9",
        "publish":"pk_test_KwbxHzO8Nlkm3qHhSBEGjeRn"
    }
}
```

Node Application Dependencies :

Need following dependencies to be installed on server to run loopback application:

- 1) nodejs
- 2) strongloop
- 3) grunt-cli
- 4) pm2 or forever
- 5) git

All instructions needed to run loopback application are listed in README file

Angular Application Dependencies :

Need following dependencies to be installed on server to run loopback application:

- 1) nodejs
- 2) yo
- 3) bower
- 4) gulp-cli
- 5) ruby
- 6) saas
- 7) git

All instructions needed to run angular application are listed in README file

Zapdish application setup :

Application is running in two environment : **Staging** and **Production**

Staging environment loopback application path : /var/www/html/zapdish/stg/loopback

Staging environment angular application path : /var/www/html/zapdish/stg/angular/dist

Production environment loopback application path : /var/www/html/zapdish/prod/loopback

Production environment angular application path : /var/www/html/zapdish/prod/angular/dist

nginx server setup :

Angular application and nodejs applications(reverse proxy) are serving through using nginx server We have created sub-domain on nginx server, that you can find at following path :
/etc/nginx/sites-available

Following configuration files are used for serving angular and nodejs api :

default : Production angular application

v1-app-update.spotcrunch.com : production server's v1 version apis

spotcrunchdev : staging server's angular application

stg-api.spotcrunch.com : staging server's api

Stg api explorer : <http://stg-api.spotcrunch.com/explorer/>

prod api path :

<http://v1-app-update.zapdish.me/api/> or <http://35.163.78.247:9000/api/>

staging api path :

<http://stg-api.zapdish.me/stgapi/> or <http://13.55.91.237:9005/stgapi/>

Production site path : <http://client.zapdish.me/>

Stg site path : <http://stg.zapdish.me>

////////zapdish live////////

Admin login :

"username": "ZapDishAdmin",

"password" : "zapdishAdmin123!@#"

////////zapdish stg////////

admin login

"username": "sptcrunchadmin",

"password": "sptcrunchadmin123",

Command for start, stop and check status of nginx server

1. Check nginx server status : **sudo service nginx status**

2. Start nginx server : **sudo service nginx start**

3. Stop nginx server : **sudo service nginx stop**

Git

We have project added on gitlab.com

Url : <https://gitlab.com>

Login details :

Username : sandip.ghadge@wwindia.com

Password : sandy4690

Repositories :

Loopback application : <https://gitlab.com/rameshnagy/spotcrunch-loopback>

Angular application : <https://gitlab.com/rameshnagy/angular>

We have created 3 branches dev, stg and master. On dev branch we are doing development, stg is used for stg site and master for production

Instructions to setup application can be found in readme file in respective project's git repository code.

How to updated code on stg site

On local machine we have three git branches, **master**, **stg** and **dev** , all functionality is first done on dev branch and when functionality is done, move code to stg branch and when all things are tested it will be moved on master branch.

Follow following steps to update angular code on stg server :

- 1) if new request comes or need code changes then do changes on local machine and once all changes done, push code to dev branch
- 2) checkout in stg branch merge dev branch code in stg and push code to stg branch
- 2) login to server using ssh and cd in angularjs code path of staging site
- 3) take a code pull on using `git pull rorigin stg` command
- 3) now use gulp command to rebuild js and css file file, using the following command
> gulp serve-dist
- 5) done

Follow following steps to update loopback code on stg server :

- 1) if new request comes or need code changes then do changes on local machine and once all changes done, push code to dev branch
- 2) checkout in stg branch merge dev branch code in stg and push code to stg branch
- 2) login to server using ssh and cd in to angularjs code path of staging site
- 3) take a code pull on using `git pull rorigin stg` command
- 3) now restart node application using following command

(To list process id use following command,

> pm2 list

It will show following result :

App name	id	mode	pid	status	restart	uptime	cpu	mem	watching
prod	3	fork	21422	online	0	31m	0%	125.1 MB	disabled
stg	0	fork	16214	online	0	78m	0%	111.9 MB	disabled

Use the following command to check which code instance is running at process id 3
> pm2 desc 3

It will show result like this :

Describing process with id 3 - name server

status		online	
name		server	
restarts		0	
uptime		23h	
script path		/var/www/html/zapdish/prod/loopback/server/server.js	
script args		N/A	
error log path		/home/ubuntu/.pm2/logs/server-error-0.log	
out log path		/home/ubuntu/.pm2/logs/server-out-0.log	
pid path		/home/ubuntu/.pm2/pids/server-0.pid	
interpreter		node	
interpreter args		N/A	
script id		0	
exec cwd		/var/www/html/zapdish/prod/loopback	
exec mode		fork_mode	
node.js version		6.9.1	
watch & reload		✗	
unstable restarts		0	
created at		2017-05-08T07:02:17.179Z	

Revision control metadata

revision control		git	
remote url		http://gitlab.neosofttech.in/Other/Flikbuzz_node.git	

repository root	/var/www/html/spotcrunch/prod/loopback	
last update	2017-05-09T06:10:40.351Z	
revision	db80e220fd2ddde30090405259219c6629d8f0c5	
comment	updated api	
branch	master	

Code metrics value

Loop delay	0.6ms
------------	-------

Add your own code metrics: <http://bit.ly/code-metrics>

Use `pm2 logs server [--lines 1000]` to display logs

Use `pm2 monit` to monitor CPU and Memory usage server

)

> pm2 stop process_id

> pm2 start process_id

5) done.

How to updated code on production site

Follow following steps to update angular code on production server

- 1) Checkout in master branch on local machine
- 2) Merge stg branch code to master branch and push code changes
- 3) Login to server using ssh and cd in angular application's root path
- 4) Pull code pull on from master branch
- 5) Use following command to rebuild minified css and js file
- 6) > gulp serve-dist
- 7) Done

Follow following steps to update loopback code on production server

- 8) Checkout in master branch on local machine
- 9) Merge stg branch code to master branch and push code changes
- 10) Login to server using ssh and cd in loopback application's root path

- 11) Pull code on from master branch
- 12) Use following command to reflect changes
- 13) > pm2 stop process_id
 > pm2 start process_id
- 14) Done.

PM2

We are using pm2 to keep loopack application in backend. Find more information on same on link : <http://pm2.keymetrics.io/>

Production server's pm2 is running with id : 3

Stg server's pm2 is running with id : 0

How to restart node js service after it down after system upgrade

- A) First list pm2 service instances using following command**
> pm2 list

It will show following result :

App name	id	mode	pid	status	restart	uptime	cpu	mem	watching
prod	3	fork	26464	online	0	31m	0%	122.7 MB	disabled
stg	0	fork	26458	online	0	31m	0%	115.0 MB	disabled

This mean now there are two up(status = online) and process running

- B) Lets check which code instance is running at instance 0 using following command**
> pm2 desc 3

It will show some result, if it contain following line, then its running production server on process id 3

script path | **/var/www/html/zapdish/prod/loopback/server/server.js**

C) IF we get process status as **stopped** , in pm2 list command, use the following command to start

process (eg. if process id is 3 for stopped process)

> pm2 start 3

D) In case server is upgraded both stg(process id 1) and prod(process id 0) will be stopped

In that case we have to restart process using following command

> pm2 start 0

> pm2 start 1

E) In case there is no process running when checked with **`pm2 list`** command, using we have to start node processes for stg and production instances:

For stg instance, go to path : **/var/www/html/zapdish/stg/loopback**

And run following command :

> pm2 start server/server.js

For production instance, go to path : **/var/www/html/zapdish/production/loopback**

And run following command :

> pm2 start server/server.js

Note : There is no change needed after server upgrade for angular code. Angular files will be served from dist directory of stg and production code instances as earlier.

Cronjobs :

Cronjob for sending push notification once order has been delivered

Cronjob's list can be viewed by running the following command :

Crontab -l

Order status

Following are all order status :

"R = Order Received,

P = Order Processing,

DI = Order Dispatched,

DE = Order Delivered,

RE= Rejected

PE = Pending,

BO= booked,

RP=Ready for pickup"

Dine in types :

1 : dine_in ,

2 : pickup

3 : deliver"

Dine_in (1):

Once order is place it goes to Pending(PE) status

1) Order Accepted then order changes cycle is like

PE → R → 'P' → BO → DE

2) if order is Rejected , then order status change cycle like

PE → RE

Pickup (2):

Once order is place it goes to Pending(PE) status

1) Order Accepted then order changes cycle is like

RE → PE → R → P → RP → DE

2) if order is Rejected , then order status change cycle like

PE → RE

Delivery (3):

Once order is place it goes to Pending(PE) status

1) Order Accepted then order changes cycle is like

RE → PE → R → P → DI → DE

2) if order is Rejected , then order status change cycle like

PE → RE

Default Zapdish user for default chef rating and review:

```
{  
  "country_id": 1,  
  "mobile_no": "+9199XXXXXX99",  
  "first_name": "Zapdish",  
  "full_name": "Zapdish",  
  "last_name": "",  
  "email": "zapdish-team@zapdish.com",  
  "user_icon": "",  
  "handle": "",  
  "active": true,  
  "username": "zapdish-team",  
  "password": "zapdish-team123!@#",  
  "referral_code": "zapdish-team"  
}
```

Platform components including technology used

Backend :

Technology used :

1. nodejs
2. Strongloop
- 3 grunt

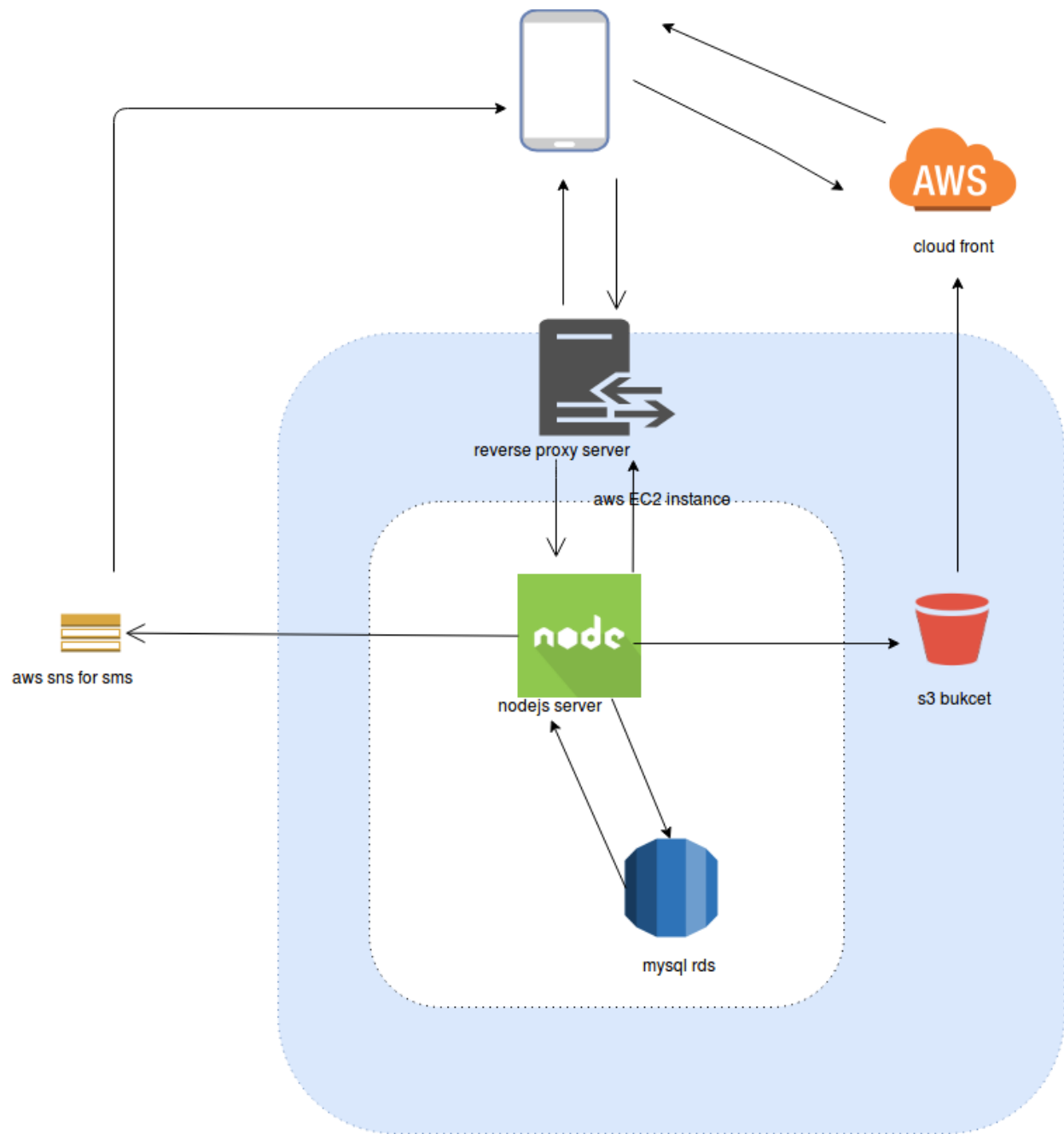
Components used :

1. Aws's sns : for sending sms
2. Aws s3 bucket for saving images
3. Mysql RDS - for managing database
4. Zomato integration for searching restaurant
5. Stripe payment

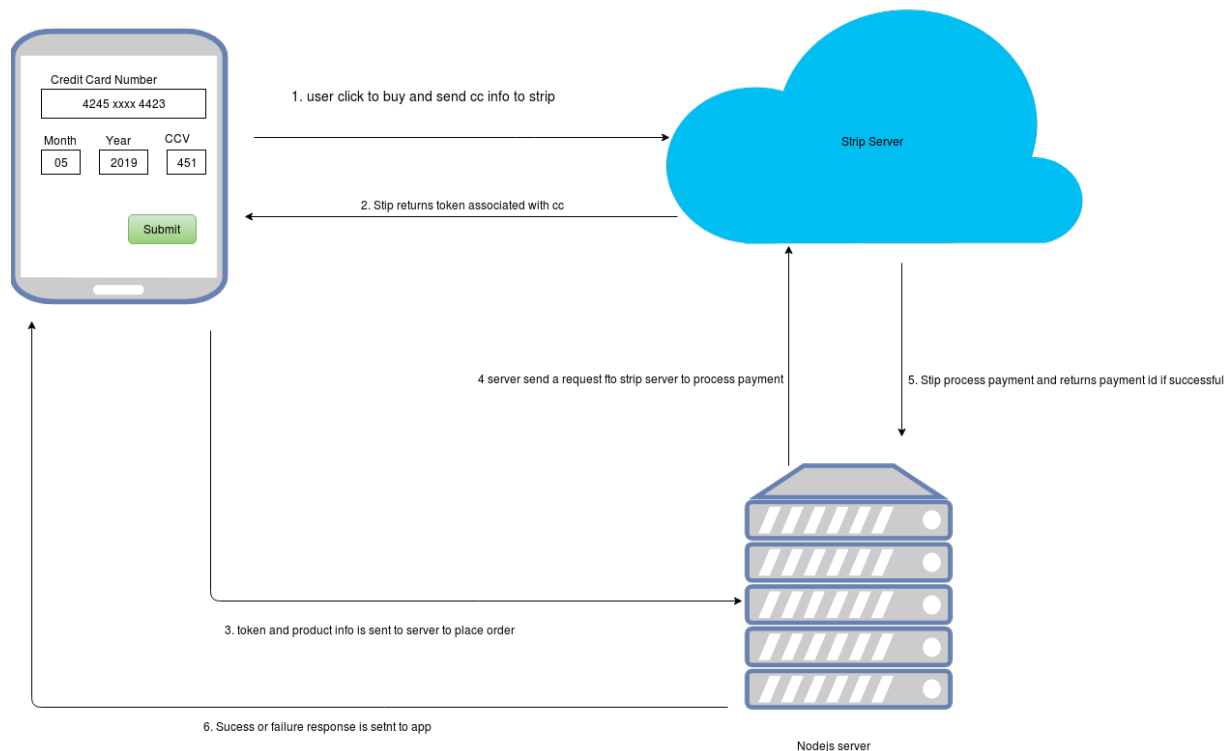
Frontend :

We are using angularjs on frontend for sysadmin and restaurant login

Logical diagram :



Payment architecture



Session Management including security - interaction between app and server and dB

- All apis are accessed through accesstoken which user receive after login in response and that access token is valid for some time frame.

Logging and log management

- Whatever logs we are showing on app side are not saved on server. If there are any exception caught then they are logged in error.log file

Database management and setup

- Database schema file is already shared with client which is a mysql workbench file containing all tables and their association with others

Third party integration and dependency or zomato API etc

- We have used zomato api, strip for payment and aws s3 bucket for saving images, Its all configurations are defined in server/config.json file.

Release Management including v1-app and auto update, staging server, code pull from git repository.

As discussed with client we are currently having v1-app that is first version of app. In future we create a new version of app and want to keep v1 app running then we have to create a another ec2 instance and deploy loopback application there with new changes as v2

Instructions for code updation are mentioned above in doc file.

Api documentation :

Loopback api documentation generated using swagger can be found here :

<http://v1-app-update.zapdish.me/explorer/>

It contains all apis available

How Default Review set for chef?

We have added a default user in database directly from sql with following details:

Email : zapdish-team@zapdish.me

First_name: Zapdish

Country_code: 1

Mobile_no: +9199XXXXXX99

Handle : zapdish-team

Active: 1

Username: zapdish-team

Full name: Zapdish

Once new chef is registered (actually once mobile no is verified), we add review by from user to chef by using review set by admin from sysadmin configuration with review rating 5.

