

Web Programming 2

Adder / Digital Circuit / CPU

Cloud Computing (AWS)

Http Request / JSON

Power of 2s

Power of 2	Exact Value (X)	Approx. Value	X Bytes into MB, GB, etc.
7	128		
8	256		
10	1,024	1 thousand	1 K
16	65,536		64 K
20	1,048,576	1 million	1 MB
30	1,073,741,824	1 billion	1 GB
32	4,294,967,296		4 GB
40	1,099,511,627,776	1 trillion	1 TB

Practice - BitWise

- https://pconrad.github.io/old_pconrad_cs16/topics/bitOps/

Bitwise AND (&)

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

Bitwise AND (&)

13 & 11

13	0001101	# 13 = 8 + 4 + 1
11	<u>0001011</u>	# 11 = 8 + 2 + 1
13 & 11 = 9	0001001	# 8 + 1 = 9

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

Bitwise OR (|)

A	B	A B
0	0	0
0	1	1
1	0	1
1	1	1

Bitwise OR (|)

13 | 11

13 0001101

11 0001011

13 | 11 = 15 **0001111**

13 = 8 + 4 + 1

11 = 8 + 2 + 1

8 + 4 + 2 + 1 = **15**

A	B	A B
0	0	0
0	1	1
1	0	1
1	1	1

Bitwise XOR (^)

A	B	A ^ B
0	0	0
0	1	1
1	0	1
1	1	0

Bitwise XOR (^)

13 ^ 11

13	0001101	# 13 = 8 + 4 + 1
11	<u>0001011</u>	# 11 = 8 + 2 + 1
13 ^ 11 = 15	0000110	# 4 + 2 = 6

A	B	A ^ B
0	0	0
0	1	1
1	0	1
1	1	0

Bitwise Left Shift (<<)

5 << 1

5	0000101	# 5 = 4 + 1
10	0001010	# 10 = 8 + 2



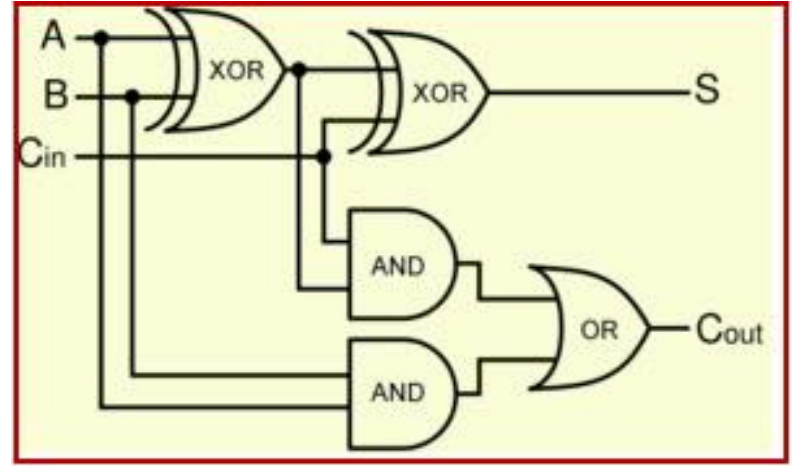
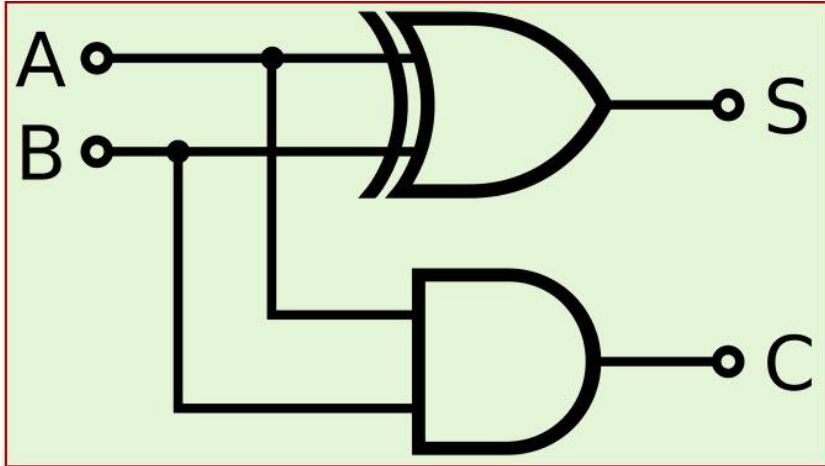
Bitwise Right Shift (>>)

13 >> 2

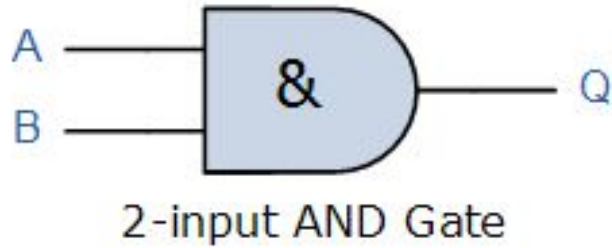
13	0001101	# 13 = 8 + 4 + 1
3	0000011	# 3 = 2 + 1



Logic gates

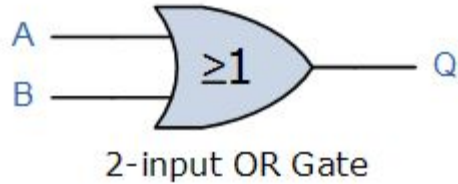


Bitwise AND (&)



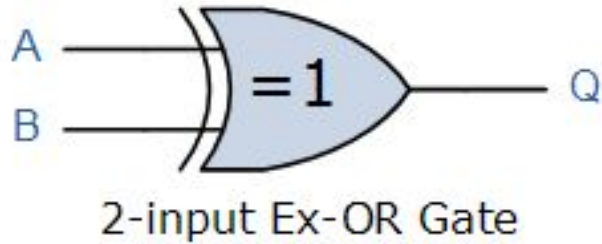
A	B	A & B (= Q)
0	0	0
0	1	0
1	0	0
1	1	1

Bitwise OR (|)



A	B	A B (= Q)
0	0	0
0	1	1
1	0	1
1	1	1

Bitwise XOR (^)

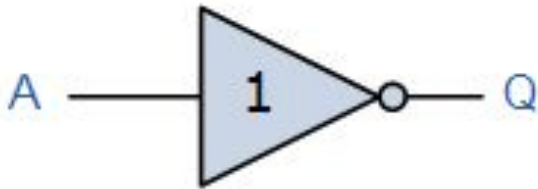


A	B	A ^ B
0	0	0
0	1	1
1	0	1
1	1	0

Bitwise NOT (~)

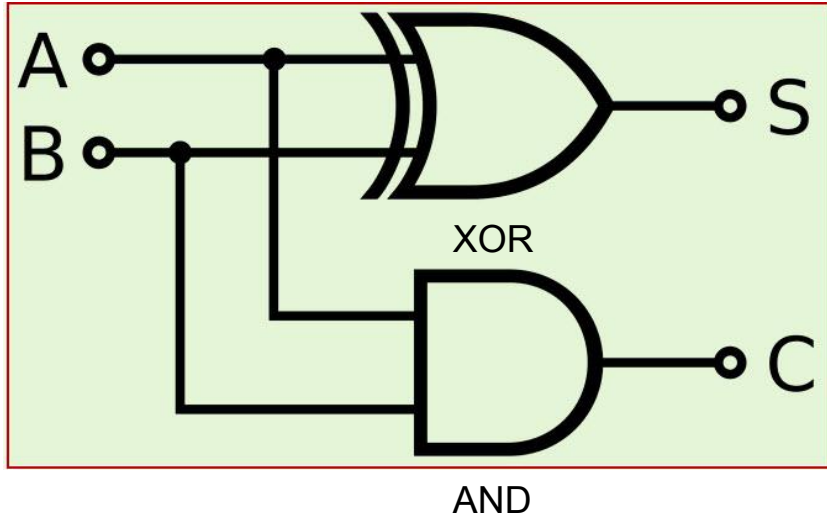
- The complement of X (Flip the bit)
 - $0 \rightarrow 1$
 - $1 \rightarrow 0$

A	$\sim A$ (= Q)
0	1
1	0



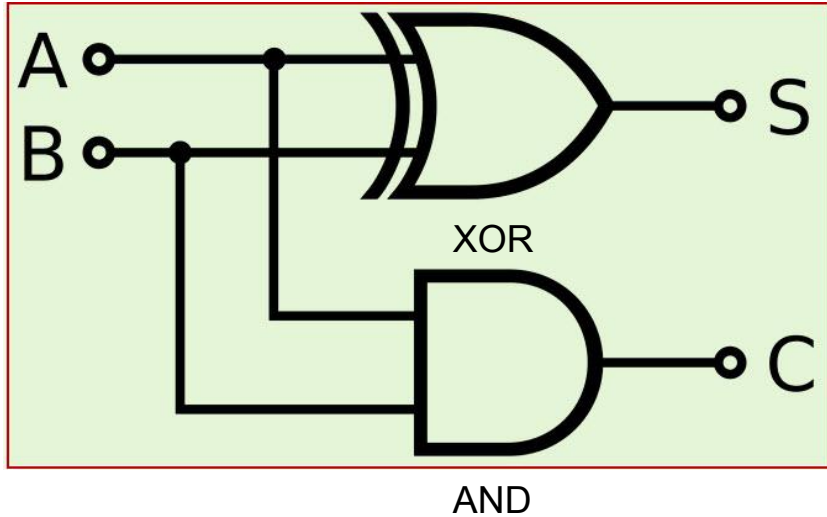
Inverter or NOT Gate

Half Adder



A	B	$A \wedge B$ (= S)	$A \& B$ (= C)
0	0		
0	1		
1	0		
1	1		

Half Adder



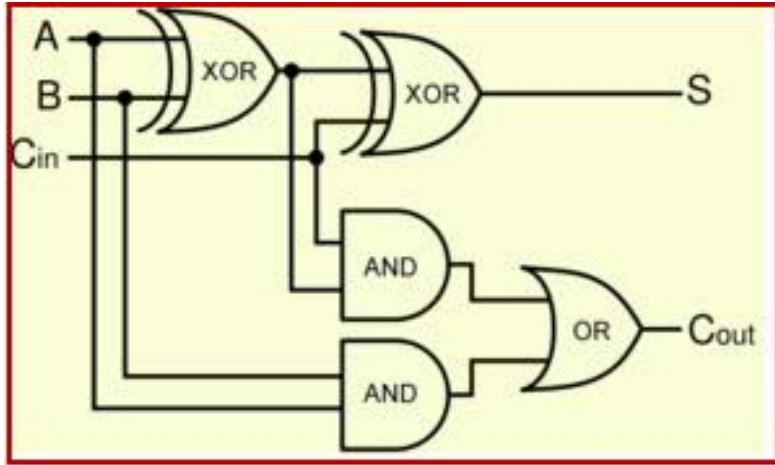
Ex)

A 0001
B 0001

Sum 0000
Carry 0010

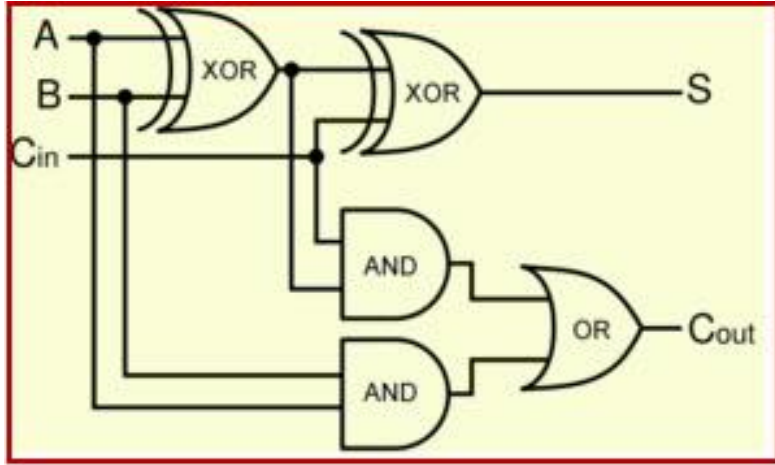
A	B	$A \wedge B$ (= S)	$A \& B$ (= C)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

Full Adder



A	B	C	S	Cout
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Full Adder



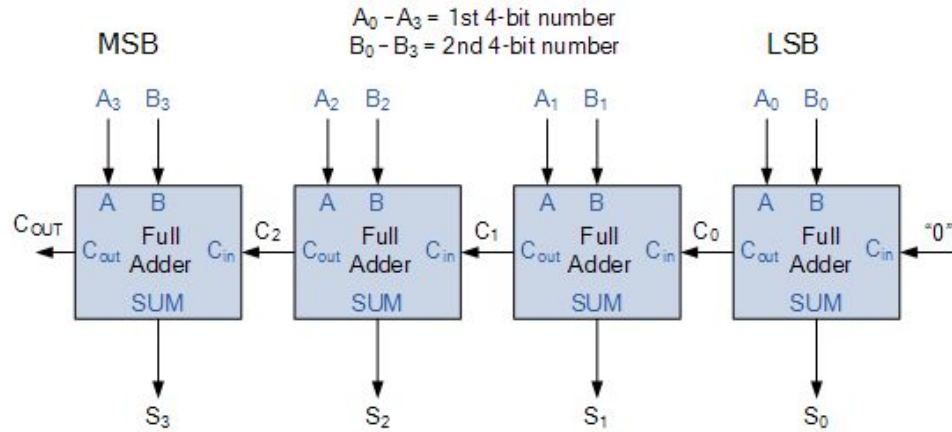
Ex)

Cin	0001
A	0001
B	0001

Sum	0001
Cout	0010

A	B	C	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

4-bit Full Adder

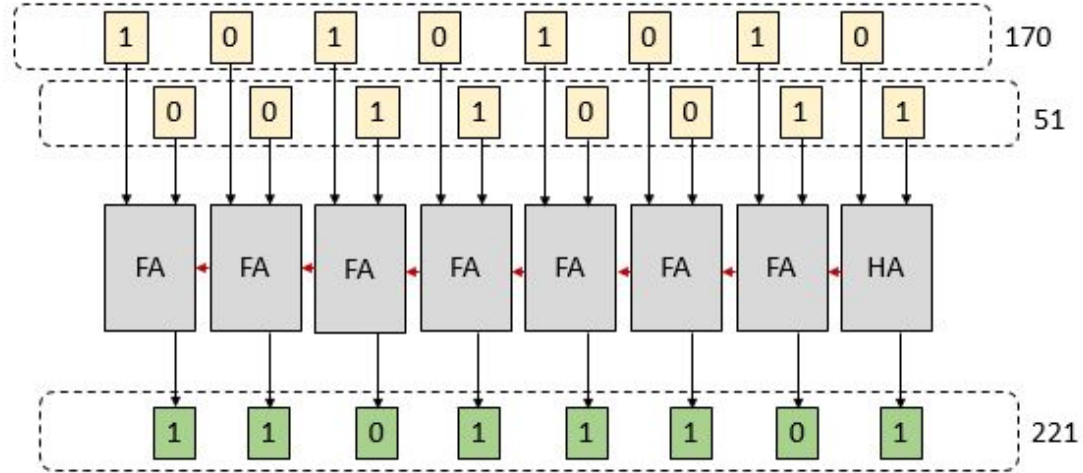


Ex)

A	0101
B	0011

Sum	1000

8-bit Adder



Ex)

A	10101010
B	00110011
<hr/>	
Sum	11011101

Digital Circuits

NTE NTE4008B - IC-CMOS 4-BIT Adder

\$2 [online](#)



Save

[Browse Integrated Circuits & Chips »](#)

The NTE4008B is a 4-bit full adder in a 16-Lead DIP type package constructed with MOS P-Channel and N-Channel enhancement mode devices in a single monolithic structure. This device consists of four full adders with fast internal look-ahead carry output. It is useful in binary addition and other arithmetic applications. The fast parallel carry output bit allows high-speed operation when used with other adders in a system. [« less](#)



Texas Instruments - Logic Adder & Subtractor

from Mouser Electronics

Texas Instruments SN74LS283N Logic Adder & Subtractor LS,PDIP-16,- 0.4 mA,8 mA

\$1.60

+ \$9.99 shipping, No tax

Mouser Electronics

90% positive (352)

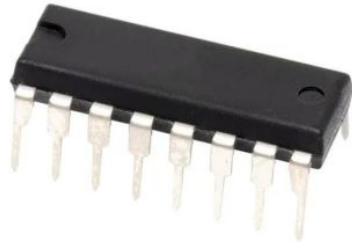
[Visit site](#)

\$1.60 [Digi-Key](#)

\$1.42 [Arrow.com](#)

\$0.63 [Verical - an Arrow company](#)

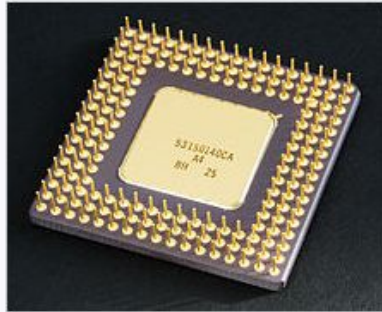
[Compare prices from 5+ stores](#)



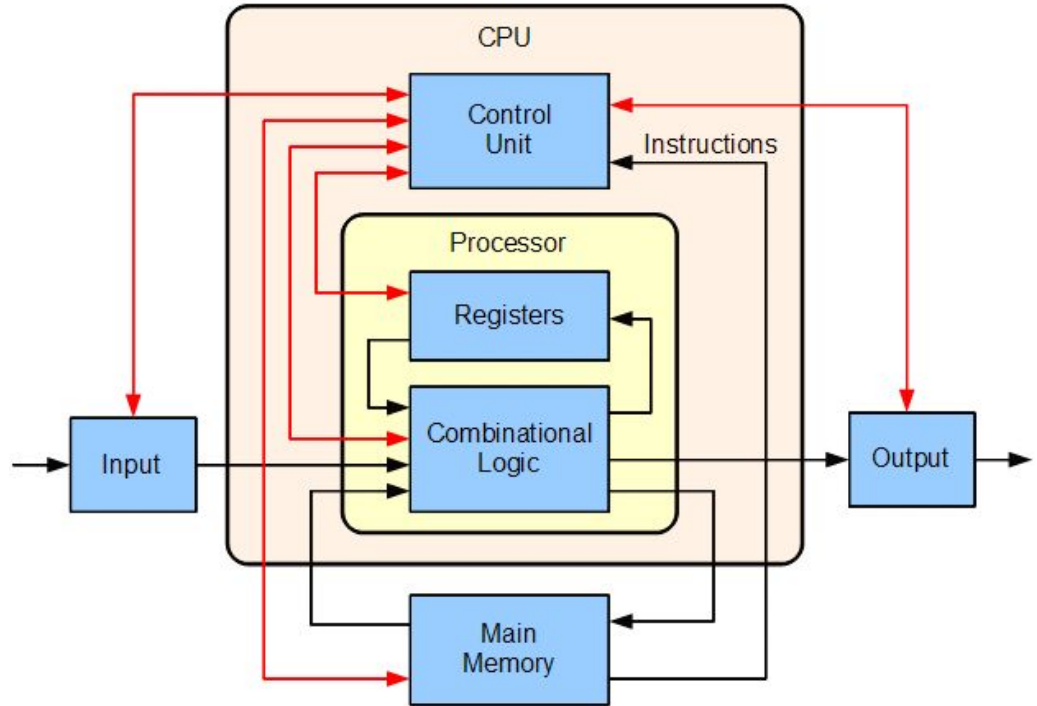
CPU - Central Processing Unit



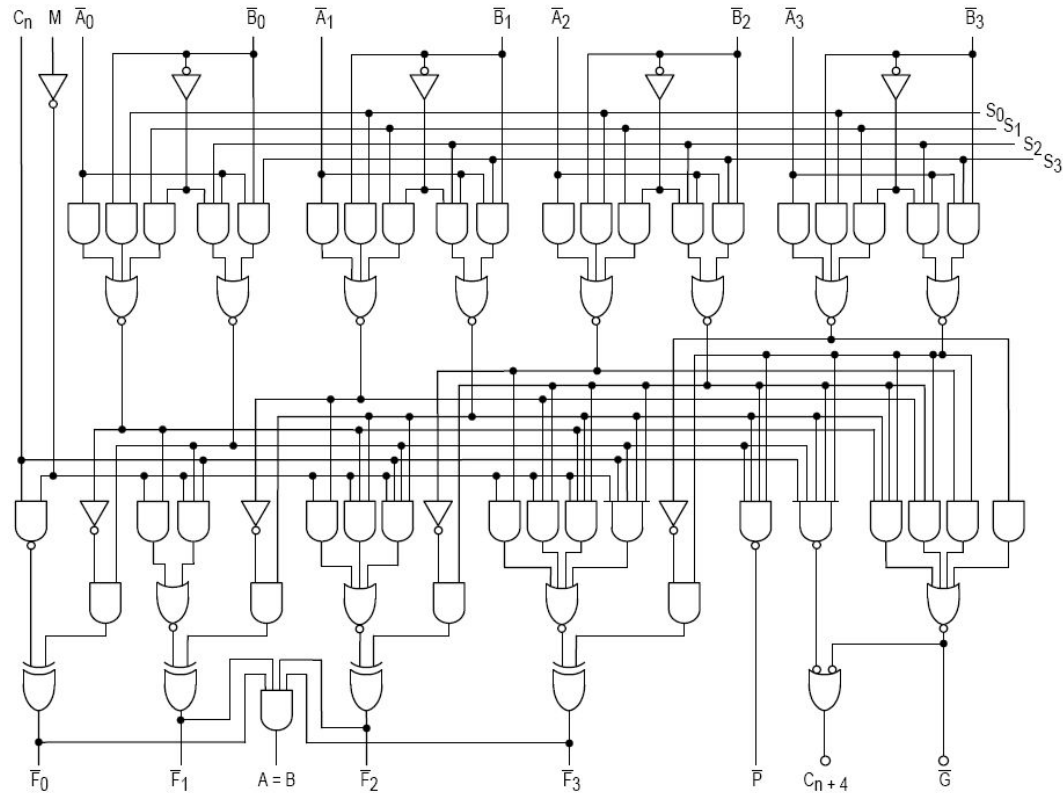
An Intel 80486DX2 CPU, as seen from above



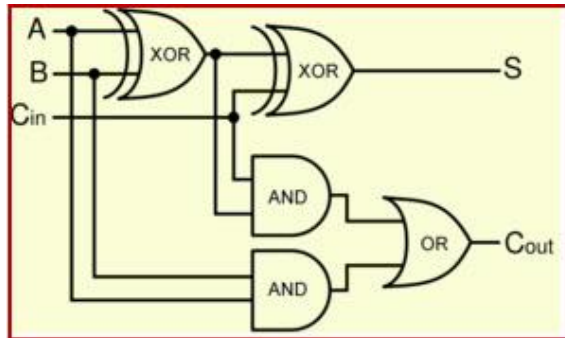
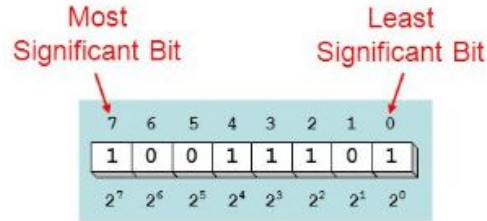
Bottom side of an Intel 80486DX2, showing its pins



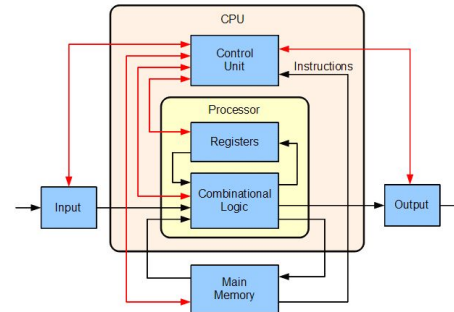
CPU -> Processor -> Combinational Logic (4-bit)



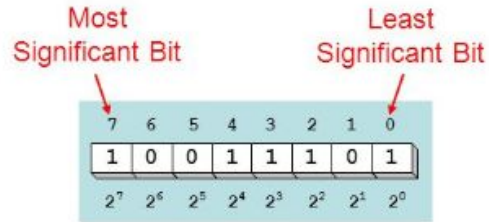
Turing-Complete



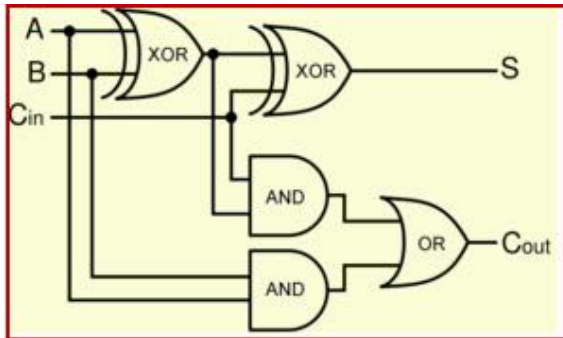
- Read and write data in memory,
- Perform conditional branching: if a memory address has a given value, jump to another point in the program



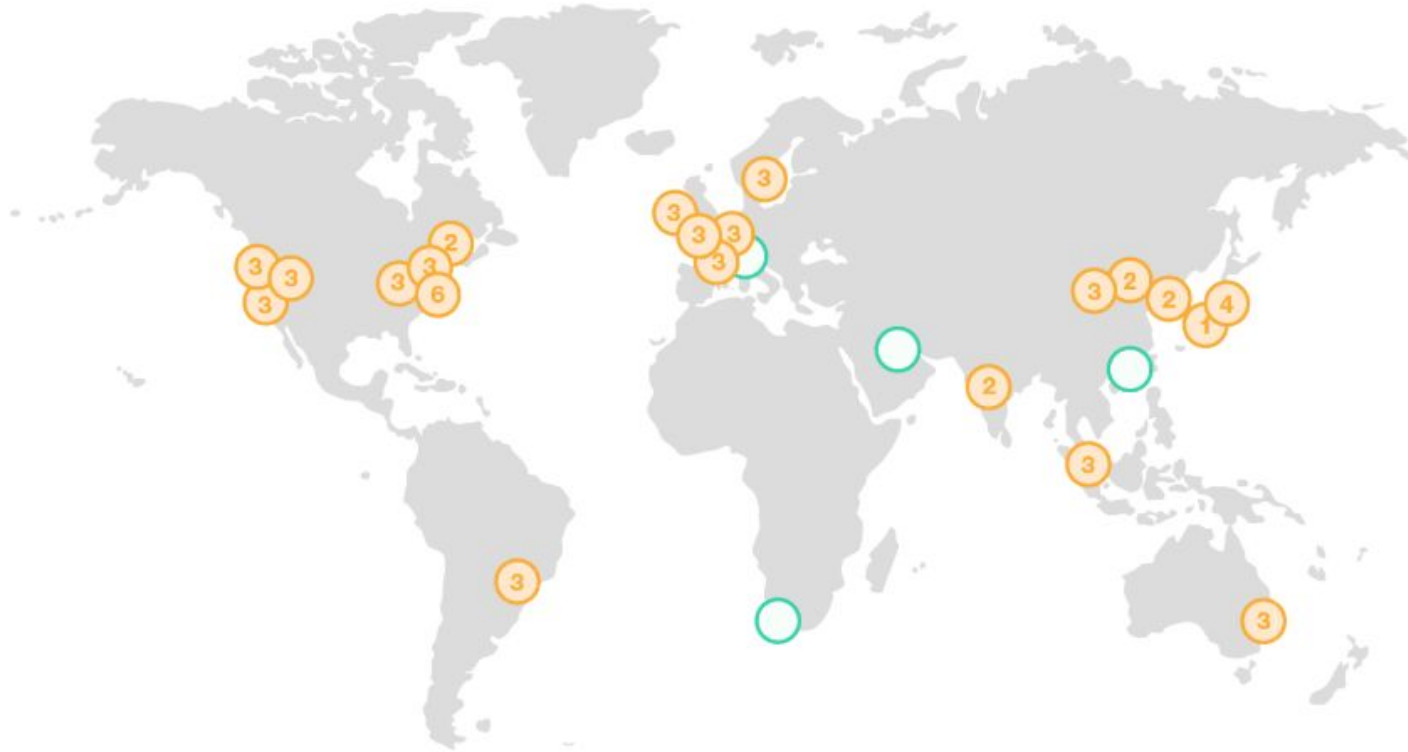
Turing-Complete



- Read and write data in memory,
- Perform conditional branching: if a memory address has a given value, jump to another point in the program



Cloud Computing - Global Infrastructure

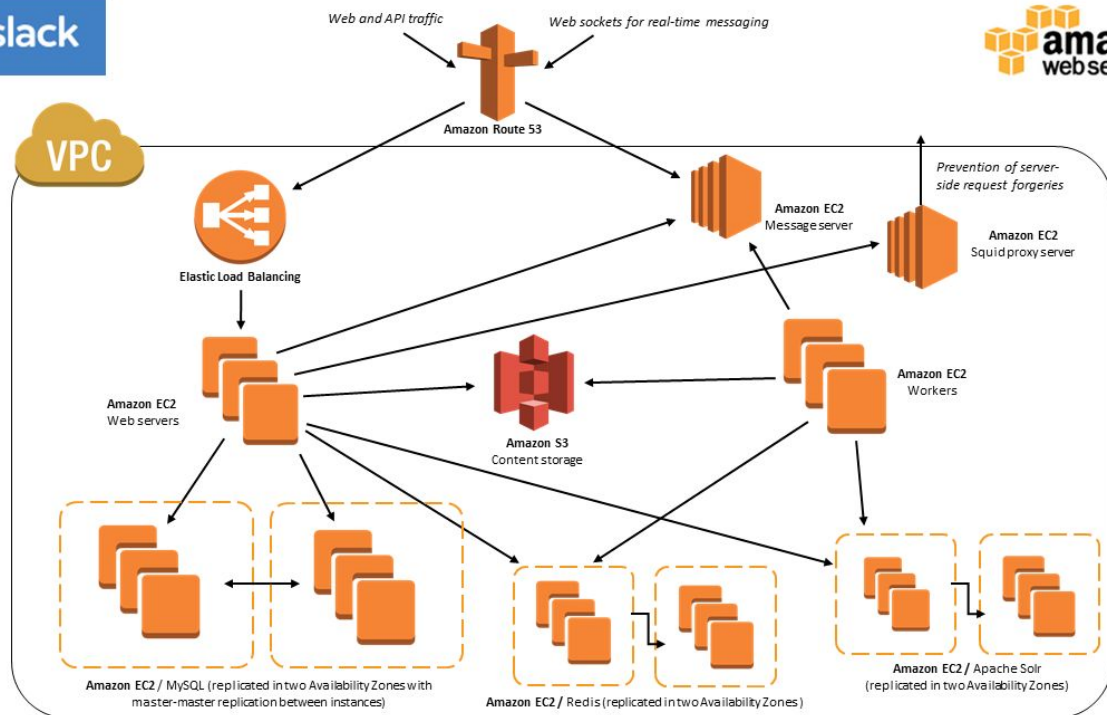


Source: <https://aws.amazon.com/about-aws/global-infrastructure/>

What is Cloud Computing?

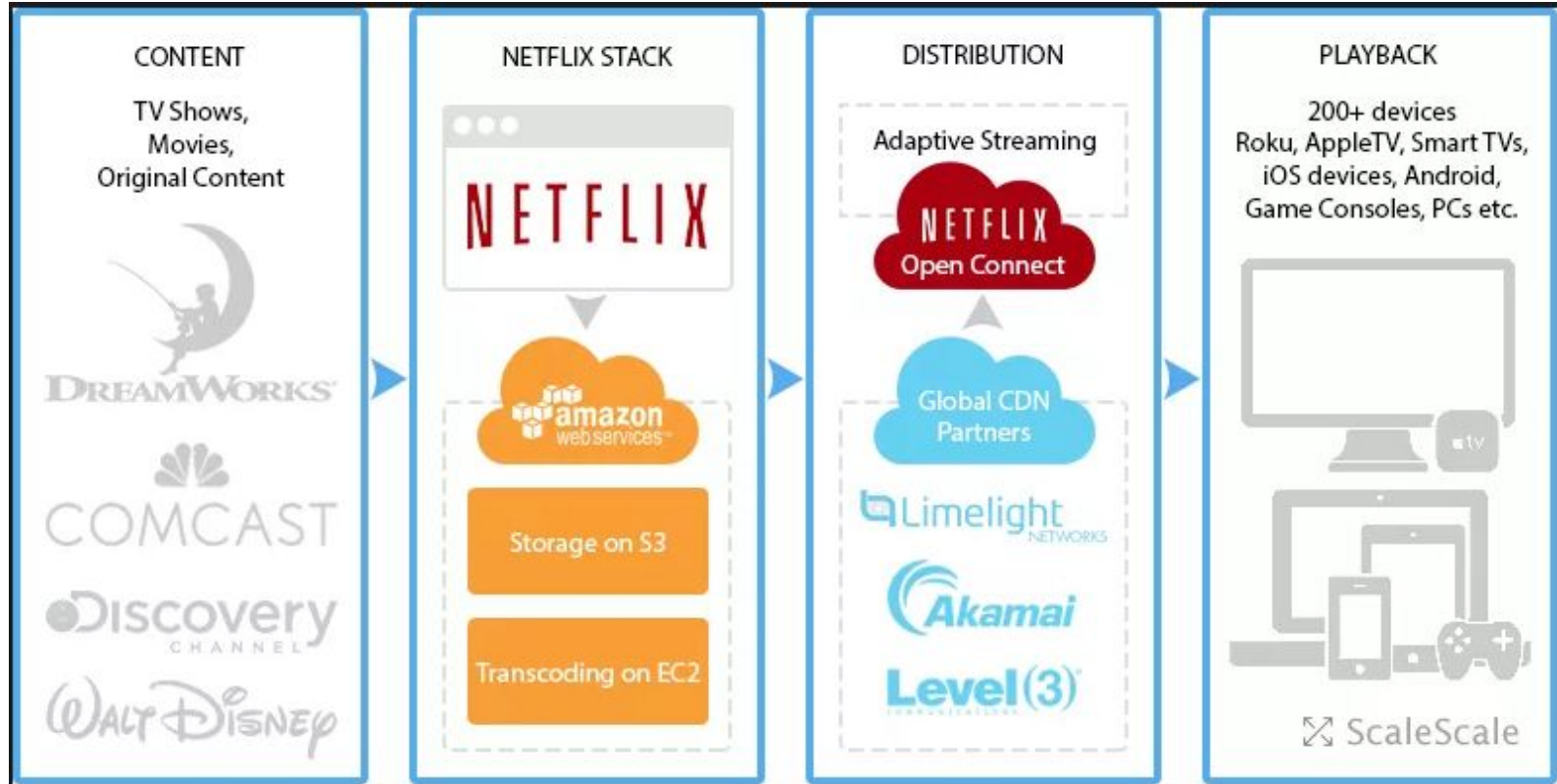
- Compute
 - Servers (EC2)
 - Serverless (Lambda)
- Network
 - Firewalls, IP addresses, DNS
- Storage
 - Database (DynamoDB)
 - Files (S3)

Slack in Amazon AWS



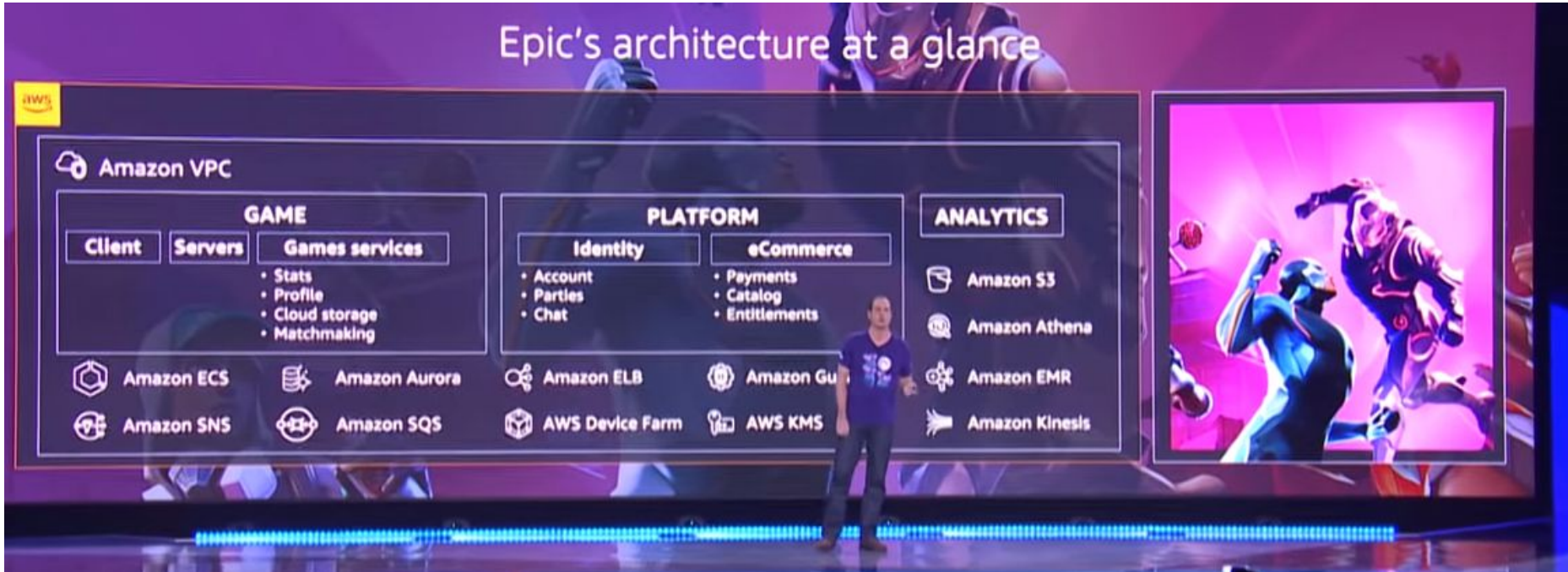
An overview of the Slack architecture on AWS

Netflix on AWS



Fortnite on AWS

Epic's architecture at a glance




Cloud Computing - AWS Lambda

- Role: GeekDojoLambdaRole
- Blueprint - HelloWorld
- Event name: test


Author from scratch ☒

Start with a simple "hello world" example.




Blueprints ☐

Choose a preconfigured template as a starting point for your Lambda function.



AWS Serverless Application Repository ☐

Find and deploy serverless applications published by AWS, AWS partners, and other developers.



Author from scratch [Info](#)

Name

jason_slack_bot

Runtime

You can select a supported AWS Lambda runtime or provide your own runtime as part of the function deployment package or Lambda layer after creating the function.

Python 3.7 ▼

Role

Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.

Choose an existing role ▼

Existing role

You can use an existing role with this function. Lambda must be able to assume this role, and the role must have Amazon CloudWatch Logs permissions.


GeekDojoLambdaRole ▼

Cloud Computing - AWS Lambda

- Role: GeekDojoLambdaRole
- Blueprint - HelloWorld
- Event name: test


Author from scratch ☒

Start with a simple "hello world" example.




Blueprints ☐

Choose a preconfigured template as a starting point for your Lambda function.



AWS Serverless Application Repository ☐

Find and deploy serverless applications published by AWS, AWS partners, and other developers.



Author from scratch [Info](#)

Name

Runtime

You can select a supported AWS Lambda runtime or provide your own runtime as part of the function deployment package or Lambda layer after creating the function.

Python 3.7

▼

Role

Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.

Choose an existing role

▼

Existing role

You can use an existing role with this function. Lambda must be able to assume this role, and the role must have Amazon CloudWatch Logs permissions.

GeekDojoLambdaRole

▼

Client and Server

Client

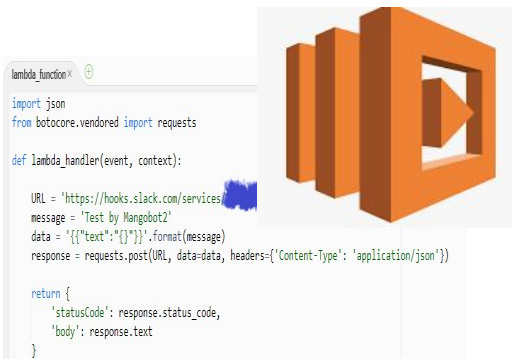
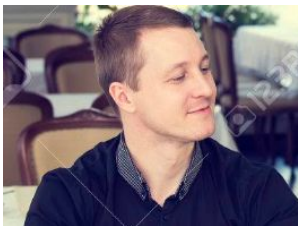
Request: Food please!

Server

Response: Here you are!



Client and Server



Http Request (POST)



Http Response (200 OK)



slack

Http Method

GET - Read record

POST - Create record

PUT - Update record

DELETE - Delete record

Http Status Code

200 OK



500 SERVER ERROR



404 NOT FOUND

Sorry, 404, Not Found!

Why JSON?

JSON Example

```
{ "employees": [
  { "firstName": "John", "lastName": "Doe" },
  { "firstName": "Anna", "lastName": "Smith" },
  { "firstName": "Peter", "lastName": "Jones" }
]}
```

XML Example

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

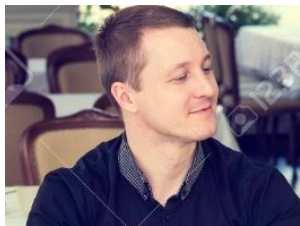
JSON - Examples

```
{ "name": "John", "age": 13 }
```

```
{ "text": "Hello, World" }
```


Client and Server

200 OK



```
POST https://hooks.slack.com...  
Content-type: application/json  
{  
  "text": "Hello, world."  
}
```

Http Request (POST)

Http Response (200 OK)



```
lambda_function  
import json  
from botocore.vendored import requests  
  
def lambda_handler(event, context):  
    URL = 'https://hooks.slack.com/services/  
    message = 'Test by Mangobot2'  
    data = '{{"text": "{}"}}'.format(message)  
    response = requests.post(URL, data=data, headers={'Content-Type': 'application/json'})  
  
    return {  
        'statusCode': response.status_code,  
        'body': response.text  
    }
```



slack

Cloud Computing - AWS Lambda

lambda_function ×



```
import json
from botocore.vendored import requests

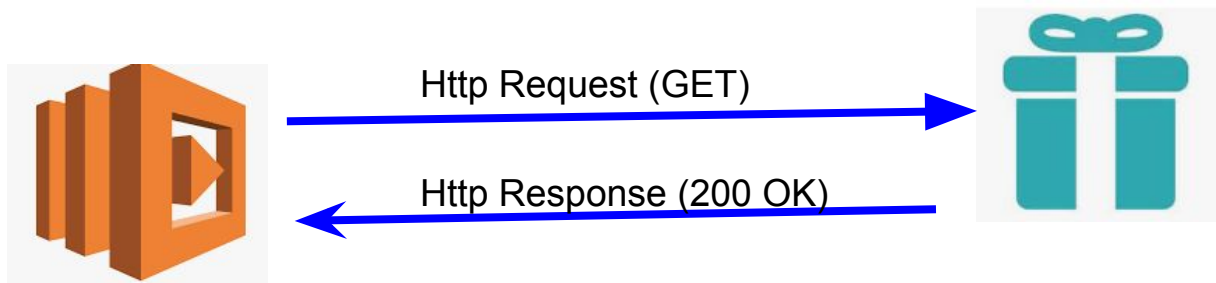
def lambda_handler(event, context):

    URL = 'https://hooks.slack.com/services/[REDACTED]'
    message = 'Test by Mangobot2'
    data = '{{"text":"{}}"}'.format(message)
    response = requests.post(URL, data=data, headers={'Content-Type': 'application/json'})

    return {
        'statusCode': response.status_code,
        'body': response.text
    }
```

Cloud Computing - AWS Lambda

- Monitoring Price

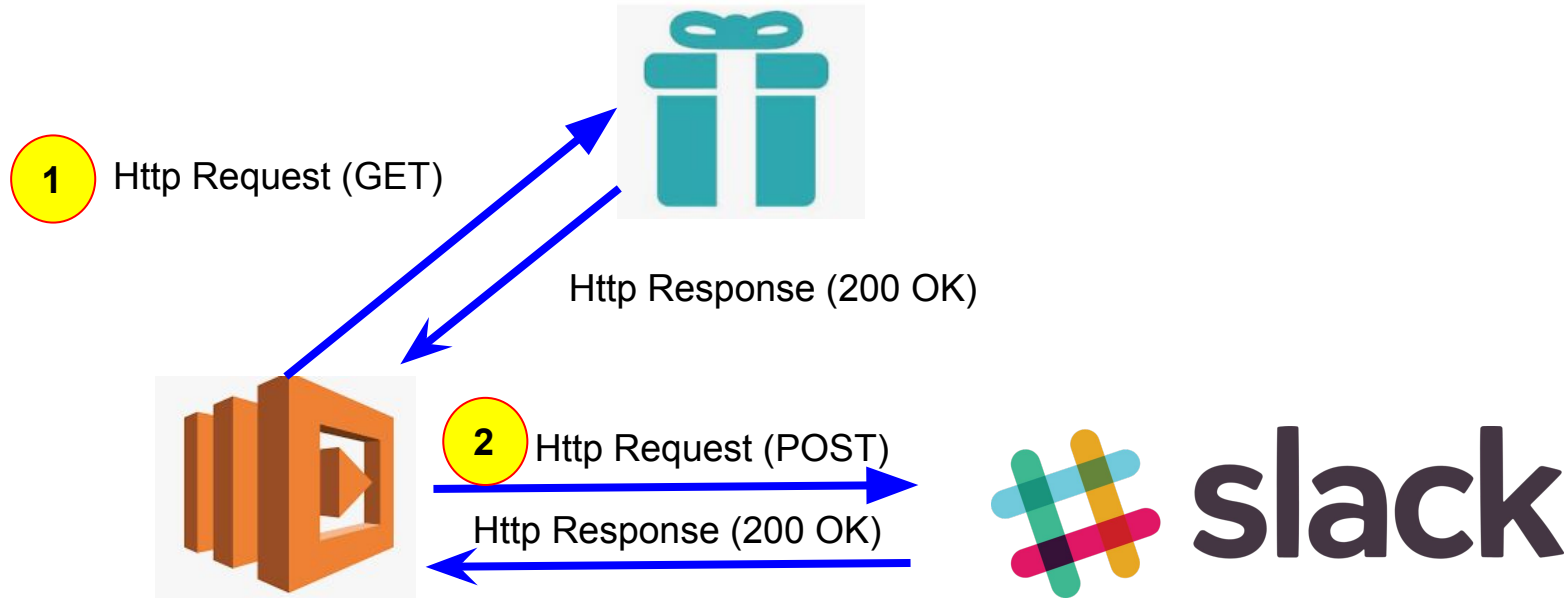




```
1 import json
2 from botocore.vendored import requests
3
4
5 def lambda_handler(event, context):
6
7     URL = 'https://s3-us-west-2.amazonaws.com/fancy-store/index.html'
8     response = requests.get(URL)
9
10    # 1
11    print(response.status_code)
12    print(response.text)
13
14    # 2
15    if '<div>Not Available</div>' in response.text:
16        print('Not Availalbe')
17
18    if '<div>Available</div>' in response.text:
19        print('Available')
20
21    # TODO implement
22    return {
23        'statusCode': 200,
24        'body': json.dumps('Hello Bob!')
25    }
```

Cloud Computing - AWS Lambda

- Send Slack after Monitoring Price



```
import json
from botocore.vendored import requests

def shouldAlert():
    URL = 'https://s3-us-west-2.amazonaws.com/fancy-store/index.html'
    response = requests.get(URL)

    print(response.status_code)
    print(response.text)

    if '<div>Not Available</div>' in response.text:
        return True
    else:
        return False

def alert(message):
    URL = 'https://hooks.slack.com/services/[REDACTED]'
    data = '{{"mrkdwn":true, "text":"{}}"}}'.format(message)
    response = requests.post(URL, data=data, headers={'Content-Type': 'application/json'})

def lambda_handler(event, context):
    if shouldAlert():
        alert('*Price Alert* \n Price has changed :smile: :brain: :face-screaming-in-fear:')

    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Lambda completed!')
    }
```