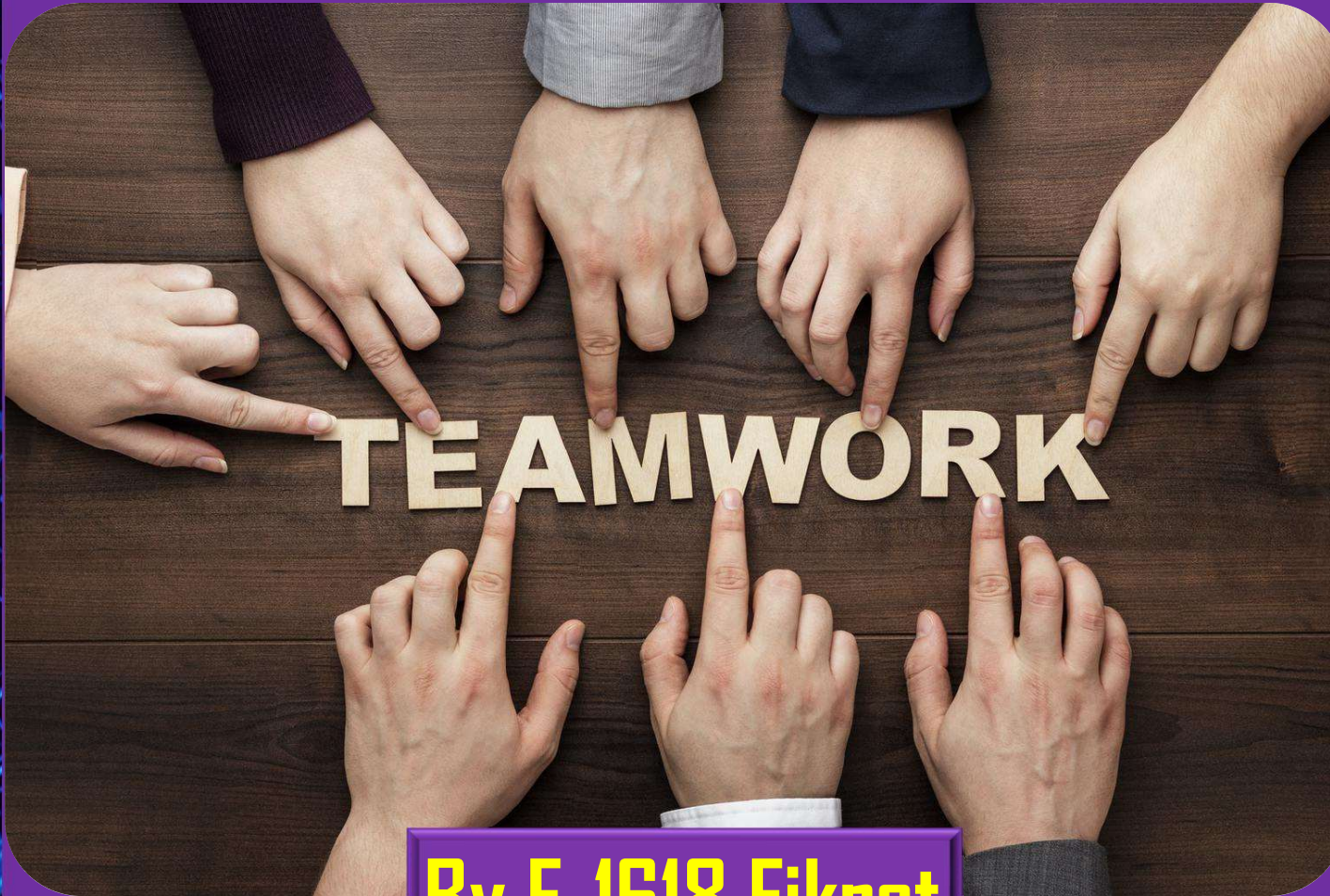


MENTORING GROUP BRAVO




By F-1618 Fikret



Objectives**AWS****Python****Have Fun****Times****21:15-22:30****TW****TW****Mentors****Josh****Q & A****Yes****Slack - Zoom Use****Use Your Mic When Needed**

1. Which Linux command gives the number of lines, words and characters in the data?

A. cat	cat	Concatenate	Allows us to create single or multiple files, view content of a file, concatenate files and redirect output in terminal or files.
B. wc	wc	Word count	It is used to find out number of lines, word count, byte and characters count
C. tee	tee		Reads the standard input and writes it to both the standard output and one or more files. The command is named after the T-splitter used in plumbing.
D. tr	tr	Translate	reads a byte stream from standard input (stdin), translates or deletes characters, then writes the result to the standard output (stdout)

«wc» stands for **word count**. As the name implies, it is mainly used for counting purpose.

- It is used to find out **number of lines, word count, byte and characters count** in the files specified in the file arguments.
- By default it displays **four-columnar output**.

•**First** column shows number of lines present in a file specified, **second** column shows number of words present in the file, **third** column shows number of characters present in file and **fourth** column itself is the file name which are given as argument.

```
wc [options] filenames
```

```
wc -l : Prints the number of lines in a file.  
wc -w : prints the number of words in a file.  
wc -c : Displays the count of bytes in a file.  
wc -m : prints the count of characters from a file.  
wc -L : prints only the length of the longest line in a file.
```

2. This command sends a request out and expects a response, indicating that both hosts are communicating.

A. **tracert**

B. **ping**

C. **nslookup**

tracert

Traceroute uses the TCP/IP suite of protocols, and sends User Datagram Protocol packets

ping

Used for checking whether any network is present and if a host is attainable.

Press **Ctrl + C** on your keyboard to stop the process.

```
test@test-VirtualBox: ~  
File Edit View Search Terminal Help  
test@test-VirtualBox:~$ ping google.com  
PING google.com (172.217.16.110) 56(84) bytes of data.  
64 bytes from prg02s12-in-f14.1e100.net (172.217.16.110): icmp_seq=1 ttl=52 time=7.68 ms  
64 bytes from prg02s12-in-f14.1e100.net (172.217.16.110): icmp_seq=2 ttl=52 time=16.0 ms  
64 bytes from prg02s12-in-f14.1e100.net (172.217.16.110): icmp_seq=3 ttl=52 time=7.77 ms  
^C  
--- google.com ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2002ms  
rtt min/avg/max/mdev = 7.684/10.498/16.039/3.918 ms  
test@test-VirtualBox:~$
```

nslookup (stands for “Name Server Lookup”) is a useful command to get information from the DNS server.

How to find an IP address

ping google.com

53

time: 200ms

1. **from**: The destination and its IP address. Note that the IP address may be different for a website depending on your geographical location.

2. **icmp_seq=1**: The sequence number of each ICMP packet. Increases by one for every subsequent echo request.

3. **ttl=52**: The Time to Live value from 1 to 255. It represents the number of network hops a packet can take before a router discards it.

4. **time=7.68 ms**: The time it took a packet to reach the destination and come back to the source.

3. Which of the following must be configured on an Elastic Load Balancing load balancer to accept incoming traffic?

- A. A port
- B. A network interface
- C. A listener**
- D. An instance

port	A port in networking is a software-defined number associated to a network protocol that receives or transmits communication for a specific service. Port 80 provides an HTTP connection
NW Interface	a protocol describing the communication rules between the operator and the device, and the format of the information that will be exchanged using that protocol.
Listener	A listener is a process that checks for connection requests, using the protocol and port that you configure. The rules that you define for a listener determine how the load balancer routes requests to the targets in one or more target groups.
Instance	An instance is a virtual server in the AWS Cloud. With Amazon EC2, you can set up and configure the operating system and applications that run on your instance.

A load balancer accepts incoming traffic from clients and routes requests to its registered targets (such as EC2 instances) in one or more Availability Zones. The load balancer also monitors the health of its registered targets and ensures that it routes traffic only to healthy targets.

You configure your load balancer to accept incoming traffic by specifying one or more listeners. A listener is a process that checks for connection requests. It is configured with a protocol and port number for connections from clients to the load balancer. Likewise, it is configured with a protocol and port number for connections from the load balancer to the targets.

4. What happens when the Elastic Load Balancing fails the health check? (Choose the best answer.)

- A. The Elastic Load Balancing fails over to a different load balancer.
- B. The Elastic Load Balancing keeps on trying until the instance comes back online.
- C. The Elastic Load Balancing cuts off the traffic to that instance and starts a new instance.
- D. The load balancer starts a bigger instance.

Health check failed

The instance will continue being monitored and if it starts failing health checks, the ELB will respond by marking it as unhealthy, **stop routing traffic to it**, and wait for the ASG to replace it.

5. Which of the following are required elements of an Auto Scaling group? (Choose 2 answers)

A. Minimum size

B. Health checks

C. Desired capacity

D. Launch configuration

What are required elements of an Auto Scaling group?

An Auto Scaling group must have a **minimum size and a **launch configuration** defined in order to be created. Health checks and a desired capacity are optional.**

1. Which of the below AWS services allows you to increase the number of resources on the demand of the application or users?

A. AWS EC2

B. AWS Autoscaling

C. AWS ELB

D. AWS Inspector

AWS Auto Scaling monitors your applications and **automatically adjusts capacity** to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it's easy to setup application **scaling for multiple resources across multiple services in minutes.** Both **S3** and **DynamoDB** automatically scale as demand dictates.

2. Which of the following components effectively facilitate a user to set up AutoScaling on EC2 instances for a web-based application? Choose 3 correct Options:

A. Launch Configuration

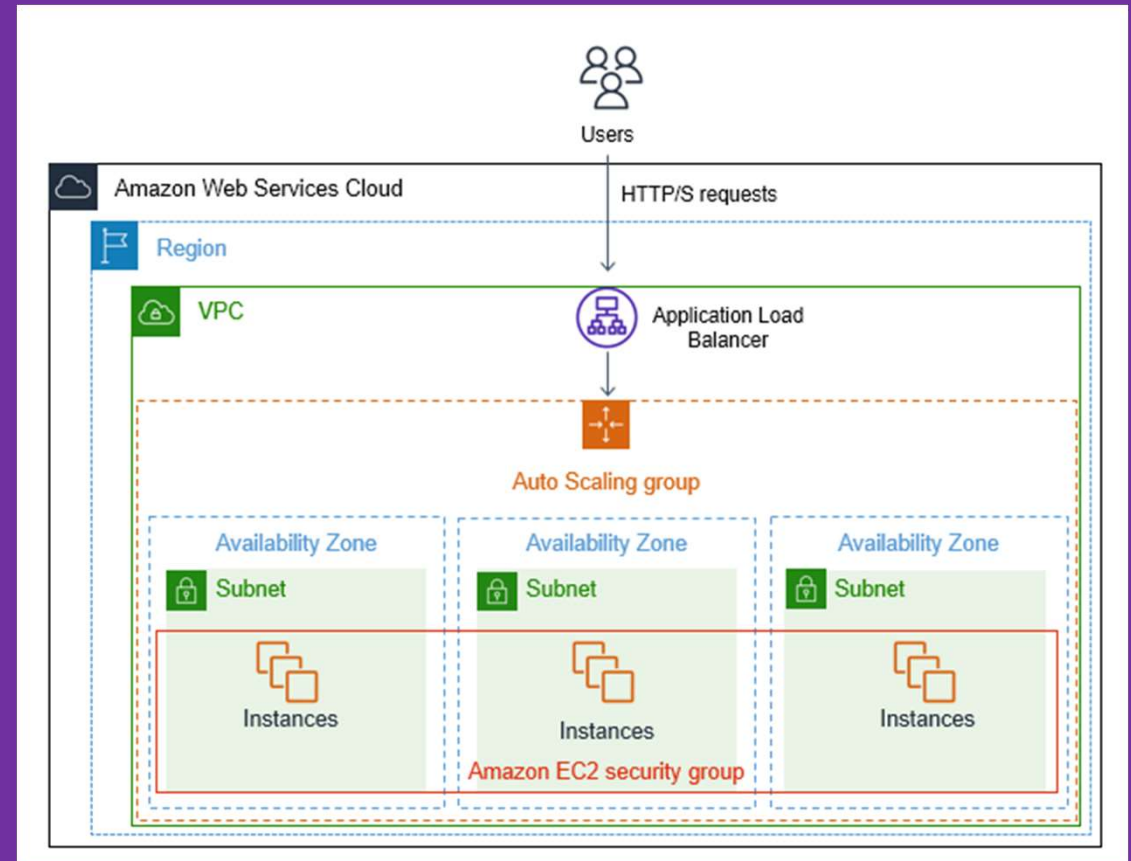
B. Elastic Load Balancer

C. Lambda

D. AutoScaling Group

E. Elastic IP

- Prerequisites
- Step 1: Set up a launch template or launch configuration
- Step 2: Create an Auto Scaling group
- Step 3: Verify that your load balancer is attached
- Step 4: Next steps
- Step 5: Clean up



3. A cell phone company is running dynamic-content television commercials for a contest. They want their **website to handle traffic spikes that come after a commercial airs**. The website is **interactive**, offering personalized content to each visitor based on location, purchase history, and the **current commercial airing**. Which architecture will configure Auto Scaling to scale out to respond to spikes of demand, while minimizing costs during quiet periods?

A. Set the minimum size of the Auto Scaling group so that it can handle high traffic volumes without needing to scale out.

B. Set the minimum size of the Auto Scaling group so that it can handle high traffic volumes without needing to scale out.

C. Configure Auto Scaling to scale out as traffic increases. Configure the launch configuration to start new instances from a preconfigured Amazon Machine Image (AMI).

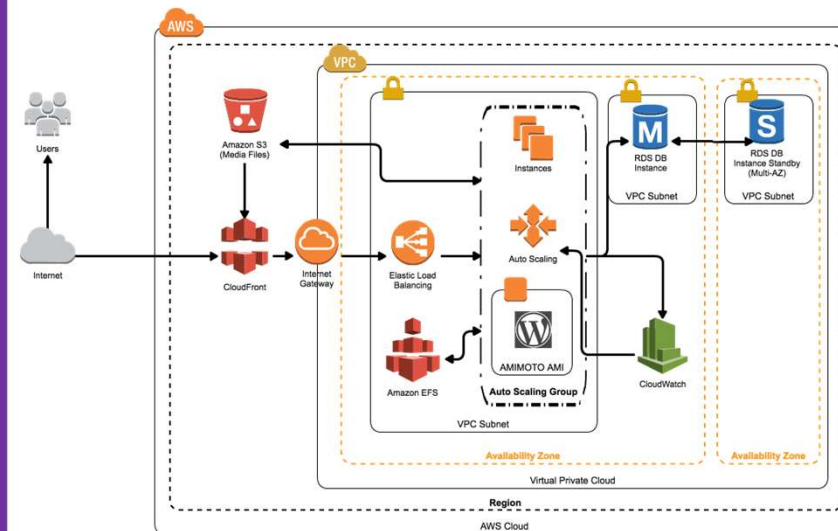
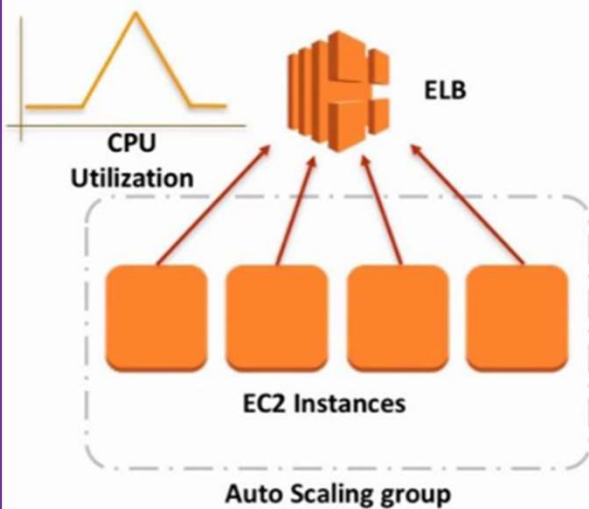
D. Use Amazon CloudFront and Amazon Simple Storage Service (Amazon S3) to cache changing content, with the Auto Scaling group set as the origin. Configure Auto Scaling to have sufficient instances necessary to initially populate CloudFront and Amazon ElastiCache, and then scale in after the cache is fully populated.

What is dynamic Auto Scaling?

Dynamic scaling **responds to changing demand** and predictive scaling automatically schedules the right number of EC2 instances based on predicted demand. Dynamic scaling and predictive scaling can be used together to scale faster.

When you **configure** dynamic scaling, you define how to **scale** the capacity of your **Auto Scaling** group in response to changing demand.

Dynamic Scaling



 **amazon**
web services
AUTO SCALING



4. Which of the following are characteristics of the Auto Scaling service on AWS? (Choose 2 answers)

A. Sends traffic to healthy instances

B. Responds to changing conditions by adding or terminating Amazon Elastic Compute Cloud (Amazon EC2) instances

C. Collects and tracks metrics and sets alarms

D. Delivers push notifications

E. Launches instances from a specified Amazon Machine Image (AMI)

AWS Auto Scaling **continually calculates the appropriate scaling adjustments and immediately adds and removes capacity as needed to keep your metrics on target.** AWS target tracking scaling policies are self-optimizing, and learn your actual load patterns to minimize fluctuations in resource capacity.

What are characteristics of Amazon EC2 Auto Scaling?




To automate fleet management for EC2 instances, EC2 Auto

Scaling **monitors the health of running instances, automatically replaces impaired instances, and balances capacity across Availability Zones.**

5. You create an Auto Scaling group in a new region that is configured with a minimum size value of 10, a maximum size value of 100, and a desired capacity value of 50. However, you notice that 30 of the Amazon Elastic Compute Cloud (Amazon EC2) instances within the Auto Scaling group fail to launch.

Which of the following is the cause of this behavior?

- A. You cannot define an Auto Scaling group larger than 20.
- B. The Auto Scaling group maximum value cannot be more than 20.
- C. You did not attach an Elastic Load Balancing load balancer to the Auto Scaling group.
- D. You have not raised your default Amazon EC2 capacity (20) for the new region.

			Ohio ▲
US East (N. Virginia)		us-east-1	
US East (Ohio)		us-east-2	
US West (N. California)		us-west-1	
US West (Oregon)		us-west-2	
Africa (Cape Town)		af-south-1	
Asia Pacific (Hong Kong)		ap-east-1	
Asia Pacific (Jakarta)		ap-southeast-3	

View your current limits

Use the **Limits** page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-Region basis.

To view your current limits

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a Region.
3. From the navigation pane, choose **Limits**.
4. Locate the resource in the list. You can use the search fields to filter the list by resource name or resource group. The **Current limit** column displays the current maximum for the resource for your account.

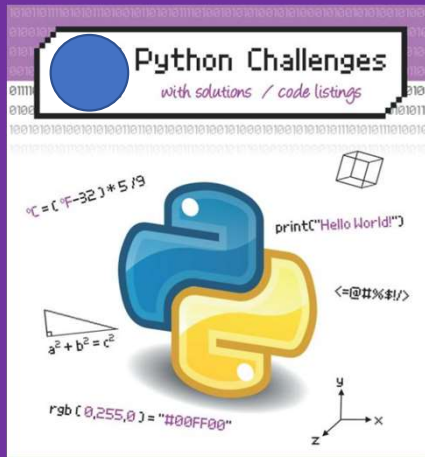


Video of the Week

- ASG Tutorial / Clarusway

<https://www.youtube.com/watch?v=ZkY28PXJIFQ&list=PL6Mbwnna00j3lt0WJpr5-l-sGxNcXeZtY&index=3>

Duration: 15 min. By Serdar



```
def MakeHTMLColor(color):
    if type(color) is type():
        color_str = "#%02x%02x%02x" % color
    else:
        color_str = str(color)
        if color_str[0] != '#':
            color_str = '#' + color
    return color_str
```

code	output
<pre>1 a = 1 2 while a < 7 : 3 if(a % 2 == 0): 4 print(a, "is even") 5 else: 6 print(a, "is odd") 7 a += 1</pre>	
<p>variables</p>	<p>www.penjee.com</p>

Coding Challenge 004 : **Validate Combination of Brackets**

Purpose of this coding challenge is to **solve a combination problem using loops.**

Learning Outcomes

At the end of this coding challenge, students will be able to;

- . understand the use of **loops**.
- . solve the **advanced and complicated** problems.
- . understand the importance of **pattern recognition**.
- . get a better understanding in **manipulating lists or strings**.

Problem Statement

Write a function that given a string containing just the characters (,), {, }, [and], determines if the input string is valid or not by using following rules.

An input string is valid if:

- Open brackets must be closed by the same type of brackets.

- Open brackets must be closed in the correct order.

Note that an empty string is also considered valid.

- **Example for user inputs and respective outputs**

Input	Output
-----	-----
"()"	true
"()[]{}"	true
"([]"	false
"([])"	false
"{[]}"	true
""	true



Solution

[https://colab.research.google.com/drive/16NA72CKW](https://colab.research.google.com/drive/16NA72CKWXJoE60EcIBXhcODspNgx15M5)
[XJoE60EcIBXhcODspNgx15M5](https://colab.research.google.com/drive/16NA72CKWXJoE60EcIBXhcODspNgx15M5)

Tutor Joseph's Solution

[https://lms.clarusway.com/mod/page/view.php?id=17](https://lms.clarusway.com/mod/page/view.php?id=17795)

795

Case study/Project 10m

Project-001 : Roman Numerals Converter Application (Python Flask) deployed on AWS EC2 with Cloudformation and AWS CLI

<https://github.com/clarusway/clarusway-aws-11-22/blob/main/aws/projects/Project-001-Roman-Numerals-Converter/README.md>

Description

The Roman Numerals Converter Application aims to convert the given number to the Roman numerals. The application is to be coded in Python and deployed as a web application with **Flask** on AWS Elastic Compute Cloud (**EC2**) Instance using AWS **Cloudformation** and **CLI** Services.

What is **Flask** Python

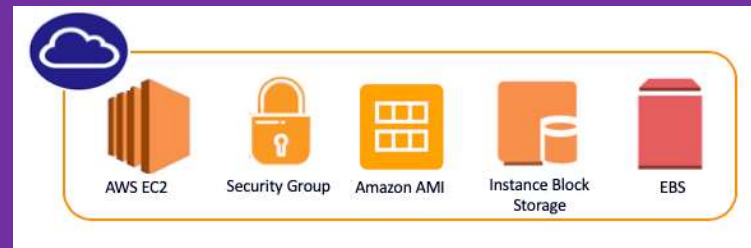
Flask is a **web framework**, it's a **Python module** that lets you **develop web applications** easily. It's has a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features. It does have many cool features like **url routing**, template engine. It is a WSGI web app framework.



EC2 stands for Amazon **Elastic Compute Cloud** is the backbone of AWS.

EC2 is a web service that provides **secure, resizable** compute capacity in the cloud. It is a service that allows you to **run application programs** in the computing environment.

It is designed to make web-scale cloud computing easier for developers



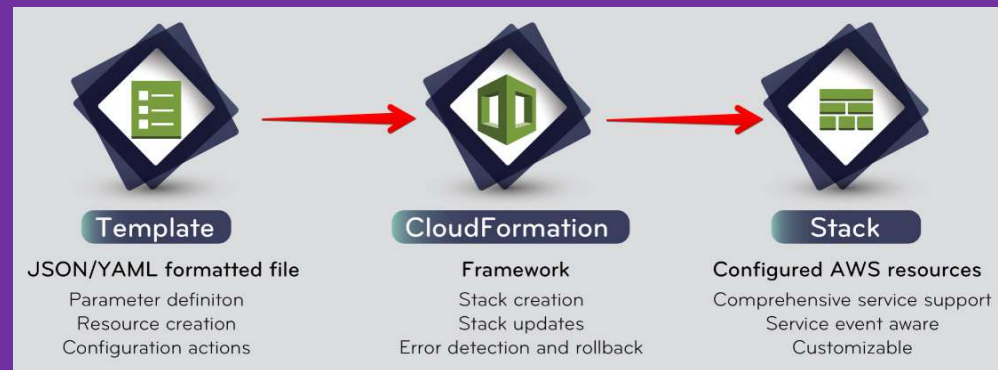
EC2 Basic Components

Features of EC2:

- EC2 provides you to **pay only for the resources** that **you** actually **use**.
- In the past, creating a server for our business sometimes takes days, weeks, but now **within 2 seconds, we can create a virtual machine** that suits our wishes.
- If you use EC2 you **don't have to buy physical servers**.

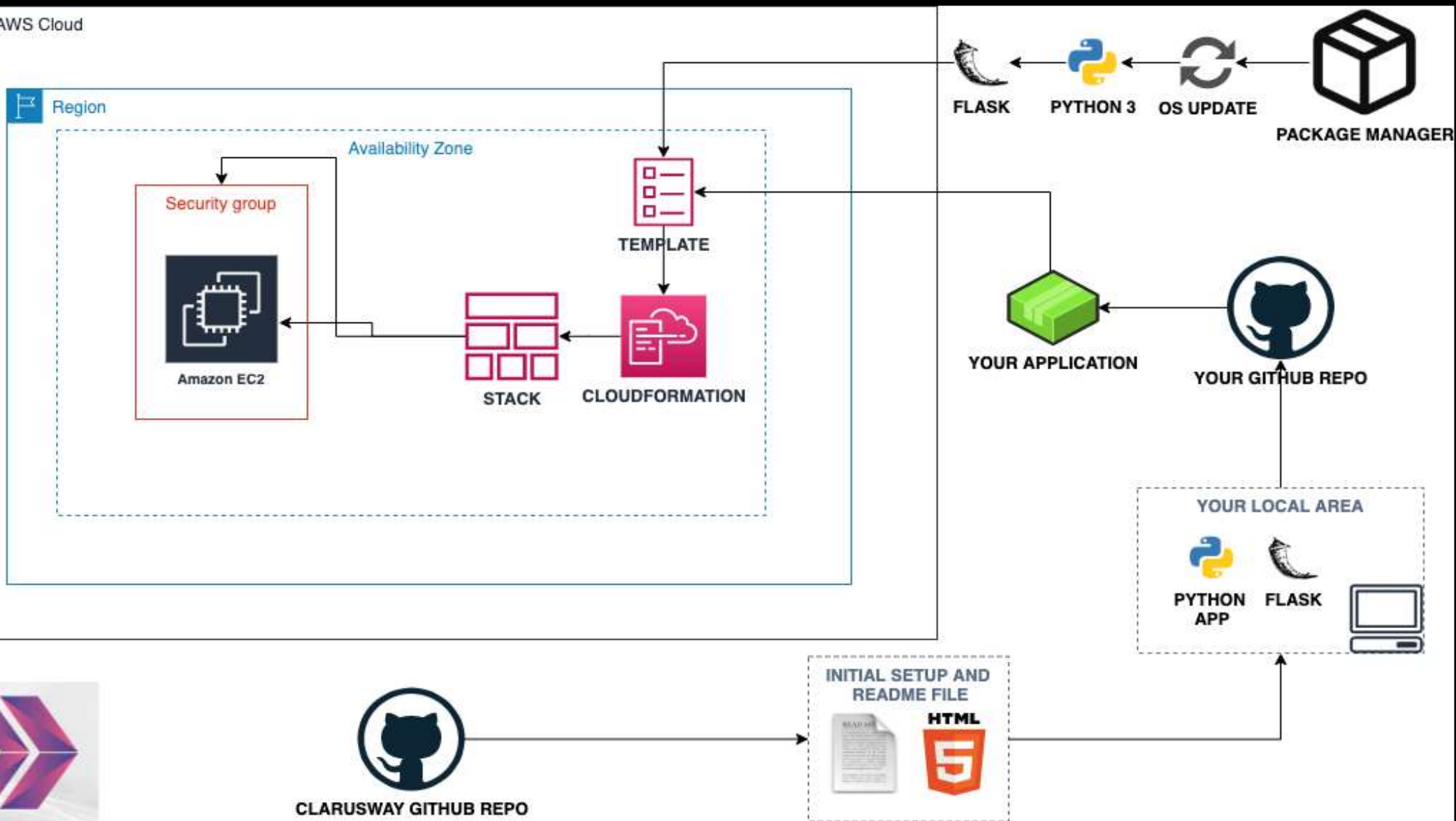
What is CloudFormation?

CloudFormation is an AWS service which enables you to create, manage, configure, replicate and delete AWS resources easily and rapidly using templates. Creating one or few resources, setting and deleting them when you are done are not so difficult and time consuming, but what if you have to deal with hundreds of them and even in different regions? Creating, setting, and managing stacks (collection of AWS resources) can be done in an optimal way, AWS CloudFormation lets you do these processes with templates, formatted text files in JSON or YAML.





AWS Cloud



Description

The Roman Numerals Converter Application aims to convert the given number to Roman numerals. The application is to be coded in Python and deployed as a web application with Flask on AWS Elastic Compute Cloud (EC2) Instance using AWS CloudFormation Service.

Açıklama

Roma Rakamları Dönüştürücü Uygulaması, verilen sayıyı Roma rakamlarına dönüştürmeyi amaçlamaktadır. Uygulama Python'da kodlanacak ve AWS CloudFormation Service kullanılarak AWS Elastik Hesaplama Bulutu (EC2) Instance'ı üzerinde Flask ile bir web uygulaması olarak deploy edilecektir.

Problem Statement

Your company has recently started on a project that aims to be one of the most used unit converters and formulas website. Roman Numerals Converter is part of the project. So you and your colleagues have started to work on the project.

As a first step of the project, you need to write a program that converts the given number (between 1 and 3999) to Roman numerals. The program should convert only from numbers to Roman numerals, not vice versa and during the conversion following notes should be taken into consideration.

You can look at the details of the program's requirements in the [README](#) file.

As a second step, after you finish the coding, you are requested to deploy your web environment using Python's Flask framework.

Problem İfadesi

Şirketiniz yakın zamanda en çok kullanılan **birim dönüştürücü ve formül web sitelerinden** biri olmayı hedefleyen bir projeye başlattı. Roma Rakamları Dönüştürücüsü projenin bir parçasıdır. Yani, siz ve meslektaşlarınız proje üzerinde çalışmaya başladınız.

Projenin ilk adımı olarak, **verilen sayıyı (1 ile 3999 arasında) Roma rakamlarına dönüştüren bir program** yazmanız gerekir. **Program sadece sayılardan Roma rakamlarına dönüştürülmelidir**, tam tersi yapılmamalı ve dönüştürme sırasında aşağıdaki notlar dikkate alınmalıdır.

Programın gereksinimlerinin ayrıntılarına [README](#) dosyasından bakabilirsiniz.

İkinci adım olarak, kodlamayı bitirdikten sonra, **Python'un Flask frame'ini kullanarak web ortamınızı deploy etmeniz** istenir.

You need to transform your program into web application using the `index.html` and `result.html` within the templates folder. Note the following for your web application.

User should **face first with index.html** when web app started.

User input should be taken via index.html using http post method

If user input is **not valid**, user should be **warned** using the index.html with template formatting.

Conversion **result should be displayed** using the result.html with template formatting.

Şablonlar klasöründeki [index.html](#) ve [result.html](#)'yi kullanarak programınızı web uygulamasına dönüştürmeniz gerekir. Web uygulamanız için aşağıdakilere dikkat edin.

- Kullanıcı, web uygulaması başladığında [index.html](#) ile ilk olarak yüz yüze gelmelidir.
- Kullanıcı girdisi [index.html](#) üzerinden http post yöntemi kullanılarak alınmalıdır.
- Kullanıcı girdisi geçerli değilse, şablon biçimlendirmesi [index.html](#) kullanılarak kullanıcı uyarılmalıdır.
- Dönüşüm sonucu, result.html şablon biçimlendirmesi kullanılarak görüntülenmelidir.

Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

- Symbol	Value
- I	1
- V	5
- X	10
- L	50
- C	100
- D	500
- M	1000

- For example, two is written as II in Roman numeral, just two one's added together. Twelve is written as, XII, which is simply X + II.

The number twenty seven is written as XXVII, which is XX + V + II.

- Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four.

The same principle applies to the number nine, which is written as IX.

There are six instances where subtraction is used:

- I can be placed before V (5) and X (10) to make 4 and 9.
- X can be placed before L (50) and C (100) to make 40 and 90.
- C can be placed before D (500) and M (1000) to make 400 and 900.

- User **input** can be either **integer or string**, thus the input is checked for the followings,
 - The input should be **a decimal number within the range of 1 to 3999**, inclusively.
 - If the input is less than 1 or greater than 3999, **program warns the user** using the given html template.
 - If the input is **string** and can not be converted to decimal number, **program warns the user using** the given html template.
- Example for user inputs and respective outputs

Input	Output
-----	-----
3	III
9	IX
58	LVIII
1994	MCMXCIV
-8	Warning with "Not Valid! Please enter a number between 1 and 3999, inclusively."
4500	Warning with "Not Valid! Please enter a number between 1 and 3999, inclusively."
Ten	Warning with "Not Valid! Please enter a number between 1 and 3999, inclusively."

- As a DevOps, developer has given you app and template folder, you are requested to deploy your web environment using Python's Flask framework.
- You are requested to push your program to the project repository on the Github and deploy your solution in the development environment on AWS EC2 Instance using AWS Cloudformation Service to showcase your project. In the development environment, you'll configure your Cloudformation template using the followings,

- The application stack should be created with [new AWS resources](#).
- The application stack should [take the name of your Key Pair as a parameter from the user](#);
- The application should [run on Amazon Linux 2](#) EC2 Instance
- EC2 Instance type can be configured as [t2.micro](#).
- [Latest AWS Linux AMI](#) should be used for template.
- Instance launched by Cloudformation should be [tagged Web Server of StackName](#)
- The Web Application should be [accessible via web browser and terminal from anywhere](#).
- The Application files should be [downloaded from Github repo](#) and [deployed on EC2 Instance](#) using [user data script](#) within [cloudformation template](#).
- Roman Numerals Converter Application Website URL should be given as [output](#) by Cloudformation Service, after the stack created.

- Uygulama **stack'i yeni AWS kaynakları ile** oluşturulmalıdır.
- Uygulama stack'i, **kullanıcıdan** bir parametre olarak **Key-Pair/Anahtar** Çiftinizin adını almalıdır;
- Uygulama **Amazon Linux 2 EC2 Instance** üzerinde çalışmalıdır.
- EC2 Instance tipi **t2.micro** olarak yapılandırılabilir.
- Cloudformation tarafından başlatılan **Instance, StackName'in Web Sunucusu** olarak etiketlenmelidir.
- Web Uygulamasına web tarayıcısı ve terminal üzerinden **her yerden erişilebilir** olmalıdır.
- Uygulama dosyaları **GitHub reposundan** indirilmeli ve **CloudFormation şablonu** içindeki kullanıcı veri komut dosyası kullanılarak **EC2 Instance üzerinde deploy** edilmeli / dağıtılmalıdır.
- Stack oluşturulduktan sonra Roma Numerals Converter Application Website **URL'si** Cloudformation Service tarafından **çıktı** olarak verilmelidir.

Lastly, try to deploy same infrastructure using AWS CLI commands to showcase your Project.

🔗 Project Skeleton

```
001-roman-numerals-converter (folder)
|----cli.sh           # To be delivered by students (CLI commands)
|----readme.md        # Given to the students (Definition of the project)
|----cfn-template.yml # To be delivered by students (Cloudformation template)
|----app.py           # Given to the students (Definition of the project)
|----templates
    |----index.html    # Given to the students (HTML template)
    |----result.html   # Given to the students (HTML template)
```

Expected Outcome



Project : Roman Numerals Converter Application

This application converts decimal numbers to Roman numerals. Only numbers from 1 to 3999 are allowed.

Please enter a number:

50

Submit

This app is developed in Python by Callahan and deployed with Flask on AWS EC2 Instance using AWS Cloudformation Service.

At the end of the project, following topics are to be covered;

- Algorithm design
- Programming with Python
- Web application programming with Python Flask Framework
- Bash scripting
- AWS EC2 Service
- AWS Security Groups Configuration
- AWS Cloudformation Service
- AWS Cloudformation Template Design
- AWS CLI Service
- AWS CLI commands, filters and queries
- Git & Github for Version Control System

At the end of the project, **students** will be able to;

Improve coding skills using iterables(dict), operators, for-loop, if statements and functions within Python

Improve web programming skills using HTTP GET/POST methods, template formatting, importing packages within Python Flask Framework

Improve bash scripting skills using user data section in Cloudformation to install and setup web application on EC2 Instance

Configure AWS EC2 Instance and Security Groups.

Configure Cloudformation template to use AWS Resources.

Use AWS Cloudformation Service to launch stacks.

Use AWS Cli to launch same stacks.

Use git commands (push, pull, commit, add etc.) and Github as Version Control System.

Steps to Solution

Step 1: Download or clone **project definition from clarusway-aws-workshop repo on Github**

Step 2: **Create project folder for local public repo on your pc**

Step 3: **Copy the Roman Numerals Converter **Application** in Python**

Step 4: Prepare a **cloudformation template to deploy your app on EC2 Instance**

Step 5: **Push your application into your own public repo **on Github****

Step 6: **Deploy your **application on AWS Cloud using Cloudformation template** to showcase your app within your team.**

Step 7: **Deploy your application on AWS Cloud using **AWS CLI** to showcase your app within your team.**

Case study/Project 10m

Project-001 : Roman Numerals Converter Application (Python Flask) deployed on AWS EC2 with Cloudformation and AWS CLI

```
from flask import Flask, render_template, request

app = Flask(__name__)

def convert(decimal_num):
    roman = {1000:'M', 900:'CM', 500:'D', 400:'CD', 100:'C', 90:'XC', 50:'L', 40:'XL', 10:'X', 9:'IX', 5:'V', 4:'IV', 1:'I'}
    num_to_roman = ''

    for i in roman.keys():
        num_to_roman += roman[i]*(decimal_num//i)
        decimal_num %= i
    return num_to_roman

@app.route('/', methods=['POST', 'GET'])
def main_post():
    if request.method == 'POST':
        alpha = request.form['number']
        if not alpha.isdecimal():
            return render_template('index.html', developer_name='Serdar', not_valid=True)
        number = int(alpha)
        if not 0 < number < 4000:
            return render_template('index.html', developer_name='Serdar', not_valid=True)
        return render_template('result.html', number_decimal = number , number_roman= convert(number), developer_name='Serdar')
    else:
        return render_template('index.html', developer_name='Serdar', not_valid=False)

if __name__ == '__main__':
    # app.run(debug=True)
    app.run(host='0.0.0.0', port=80)
```