

Sujet d'épreuves des Finales Nationales de la 47^e Compétition des Métiers

MÉTIER N°53 CLOUD COMPUTING

Soumis par :

Brandon ANCELIN, Expert WorldSkills France

1. EXPLICATION DU SUJET – JOUR 2

DUREE TOTALE DE L'ÉPREUVE	6 heures
DIFFUSION DU SUJET	Découvert le jour de la compétition

Contents

This Test Project consists of the following documentation/files:

- 47_FNAT_53_Cloud-Computing_Day2.pdf

Context

Some customers need to move services on the AWS Cloud. With a background as AWS Solution Architect and DevOps, you must implement the services needed for each customer request.

All of your deployments must be referenced with your company standard, you have to make all of your deployments using Infrastructure as Code with Terraform.

Good to know, the requests might come from someone without any technical knowledges. If so, it's your duty to implicitly determine which AWS services to use and make your deployments secure.

Do not forget good practices and AWS Well Architect Framework during your tasks.

Terraform

For each request, you must create your terraform project in a dedicated and isolated folder. Using this structure enable you and your company to run specific request without impacting another one when you are performing terraform command.

Do not deploy multiple customer request at the same time or you will get conflict.

As an AWS Consulting, you must implement the easiest process to allow your colleague to use your project and use its own AWS Account.

An example below of how you should structure your folders:

```
Document/  
├─ CR01 - Example 01 /  
│   ├── .terraform /  
│   ├── main.tf  
│   ├── config /  
│       └─ variables.tfvars  
├─ CR02 - Example 02 /  
│   ├── main.tf  
│   ├── provider.tf  
│   └─ variables.tf
```

In each Custom Request, there are 2 parts: **Summary** and **Mandatory task**.

The summary is the explanations of the context and enable you to know your tasks, explicitly or implicitly. If there is no precision, you can use your method that you want. But take care of the details of each custom request!

Mandatory task is an explicit constraint to accomplish your tasks. It can be a specific AWS Services to use, or a specific output in your Terraform root module (to show the value when executing a terraform apply command). Keep in your mind that each detail in customer requests context is important.

AWS

You will use the AWS Account given during the competition day. This account is personal and must not be shared with others.

You must use the eu-west-3 (Paris) region. You may need to use global services which is not linked to specific region.

The AWS Account must be empty when a Customer Request is destroyed. Only one Customer Request must be deployed to avoid conflicts.

Description of project and tasks

Each Customer Request has its request details in this document.

Each Customer Request has its assets or resource files in a specific folder which can be downloaded by a provided url in each Customer Request.

Customer Request 01 - Deploy a web server

Summary

We are a company with 500+ employees across the World, each of them work remotely from their home. I am Executive Marketing Manager, and our local IT explained to me that our on-premise datacenter will be turned off in 2 months. For this reason, I need your help to migrate one of our website in the Cloud.

I don't know what really is this, but below a message of the local IT has sent to me:

Hi Mathew,

I'll be off for 2 weeks. I need you to contact an external Cloud Engineer to migrate your primary website (the one where your department publish weekly news).

Here an important details to forward to the external Engineer:

I want to host the web server in EC2 instances named **marketing-webapp**. It's probably not the most cost optimized but I want to keep access to my favorite linux system, Amazon Linux 2 and manage it remotely as I want. With the most secure way as possible of course!

Create an IAM account for me named **mathew** linked to a group named **AdminRole**. The created role must have full access to the AWS console with the right AWS managed policy. Generate a password for me but force me to change it in my first log in.

Almost forgot, I want an Application Load Balancer named **marketing-web-alb** for my web app! The ALB must be the only one possible endpoint to get access the the web application, exposed to the port 80 (HTTP).

It's a lightweight app which does not require a lot of resources. Please use 1vCPU and 1GB Memory for x86 architecture or 2vCPU and 2GB Memory for arm architecture.

Do your best about security and elasticity..

In addition, I know the traffic is quite stable throughout the year.

Mandatory tasks

In your terraform project add the following outputs:

- **website-url** - Show the url website.
- **mathew-user-password** - Show the IAM User password.

The server application has two versions: x86_64 and arm. They are both statically linked and are binaries executable, ready to use. It has been tested on Amazon Linux 2. You can find the binaries here:

<http://wsfrskill53.s3.eu-west-3.amazonaws.com/47CNAT/C2/CR01/>

Customer Request 02 - Deploy a public API

Summary

Hi team,

We have a request from one of our customers. It's quite critical for them. Below the email of the request:

from: david.wald@itcorp.com

to: aws-engineer@aws.com

Object: I wanna deploy a public API

Hello,

I really need your help. It's very important and a critical service. My boss needs to reduce the cost of our very easy, but critical, API endpoint. It's currently hosted in our EC2 instance. But my boss want that the services must be serverless and highly available with no single failure. When you access to our API with your browser, the result has to be as below:

The path `/` show the string `Hello World` message.

The path `/date` show a string containing the current date and time with this specific format: `DD/MM/YYYY hh:mm:ss`.

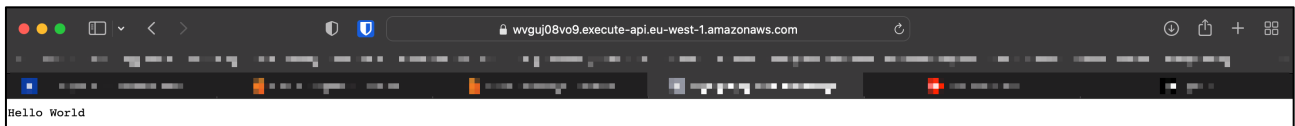
The path `/message` show the string `Please INPUT your message`. When the GET variables named "text" is set, the value must be shown in the webpage.

Because I love this language, I want you use Python 3.10.

Can you do this for me?

Take care of the API result, it's case sensitive and lot of applications using this API. We also want to see the Lambda logs in Cloudwatch, in case of error.

Below an example for you of our current API with the path `/` :



Mandatory tasks

In you terraform project, add the output as below:

- `api-arn` - The ARN of your deployed API.
- `lambda-hello-arn` - The ARN of your AWS lambda used to show the Hello World message.
- `lambda-date-arn` - The ARN of your AWS lambda used to show the date.
- `lambda-message-arn` - The ARN of your AWS lambda used to show a message.
- `api-endpoint-lambda-hello` - The direct URL to see the hello world message.
- `api-endpoint-lambda-date` - The direct URL to see the current date and time.
- `api-endpoint-lambda-message` - The direct URL to see `Worldskills`.

You must use at least these AWS services:

- API Gateway (no matter the API Type)
- AWS Lambda

Your Lambda function naming :

- `lambda-hello` - The function to return the "Hello World" message.
- `lambda-date` - The function to return the date.
- `lambda-message` - The function to return the messages.

Customer Request 03 - Deploy a static web server

Summary

Hello, I'm John, a local IT of a big company in Australia. With the website, we sponsor different brands along the year and depending on the holiday season. We notice a peak of traffic during the Summer and Christmas.

I host a static website on apache in a VM with vSphere Hypervisor. For few months now, the website becomes slower and slower and it's getting really critical for our business during a peak time.

Moreover, because we host it on-premise in Australia, all of our customers are located in Europe and North America have a high latency...

I know the Cloud AWS can help us to fix the problems, by using serverless services. I don't know really which AWS Services and how to do it, that's why I need your help. I have complete confidence in you to do your best for security and optimisation in all steps.

Mandatory tasks

For your terraform project add the following outputs:

- You must host the static website using a S3 bucket with the prefix `wsc2024-47fnat-modb-`.
- Create one distribution which contains an origin named `wsc2024-47fnat-modb`. The cache policy must be `CachingOptimized`. The Origin request policy must forward the user-agent by using the origin request named `UserAgentRefererHeaders`.
- The access to the website must be allowed only by using the Cloudfront URL endpoint.
- I want that my webserver is secure between the client and the Cloudfront endpoint, please allow only HTTPS.

Configure correctly with best practices about optimization, security, performance...For hosting a website using a S3 bucket.

You can use other AWS Services if needed.

Show these outputs in your Terraform project:

- `s3-bucket-arn` - The ARN of your s3 bucket.
- `s3-bucket-name` - The name of your s3 bucket.
- `s3-bucket-url` - Website URL of your s3 bucket.
- `website-url` - The best/preferred URL to use to get access to the website.

Règlement

En plus du règlement officiel de la compétition, s'ajoute :

Utilisation d'internet

L'accès à internet est autorisé. Vous pouvez naviguer sur les forums, documentations, etc...

Cependant, quelques restrictions s'appliquent :

- Interdiction de se connecter à des sites internet nécessitant :
 - Un nom d'utilisateur et/ou mot de passe
- Utilisation de site publique uniquement.
 - Pas de site personnel
 - Pas de bucket public
- Assistance IA
 - ChatGPT
 - Tout autre possibilité de se reposer sur une IA

Liste des Annexes

- Annexe 1 : 47_FNAT_53_Cloud-Computing_Sujet Day 2.pdf
- Annexe 2 : <http://wsfrskill53.s3-website.eu-west-3.amazonaws.com/#47CNAT/C2/>