CLOUD ARCHITECTURE DOCUMENTATION

PREPARED FOR

FakeCorp

PREPARED BY

Alvaro Garcia Bamala

SUMMARY

DOCUMENT CONTROL

Title	Cloud Architecture Documentation
Subject	Documentation and proposal of the cloud architecture for FakeCorp
Contributors	Alvaro Garcia Bamala
E-Mail	alvaro.garcia-bamala@epitech.eu
Update date	16 November 2022
Document version	1.2.0

Date	Version	Author(s)	Section(s)	Comment
16/11/2022	1.0.0	Alvaro García	*	Creation of the document
23/11/2022	1.0.1	Alvaro García	*	Rewritten misspellings and fixed format
23/11/2022	1.1.0	Alvaro García	*	Creation of the glossary
23/11/2022	1.2.0	Alvaro García	Planning	Added planning

EXECUTIVE SUMMARY

FakeCorp SA is a development company that provides customers with infrastructures and simple web applications.

They are currently using:

- An accounting software.
- A Customer Relationship Management software.

We know for sure that both are compatible with the IAM[1] caution of the chosen cloud provider.

They organically developed until they hit 5 customers and now want to move from an on-premise[2] environment to a full cloud infrastructure[3].

Their customers are mostly in Europe but one of them is in the USA.

In this context, this document contains a presentation of what will be the new full cloud architecture for this project.

Signed as accepted by client:

Name, Position, Signature

Date

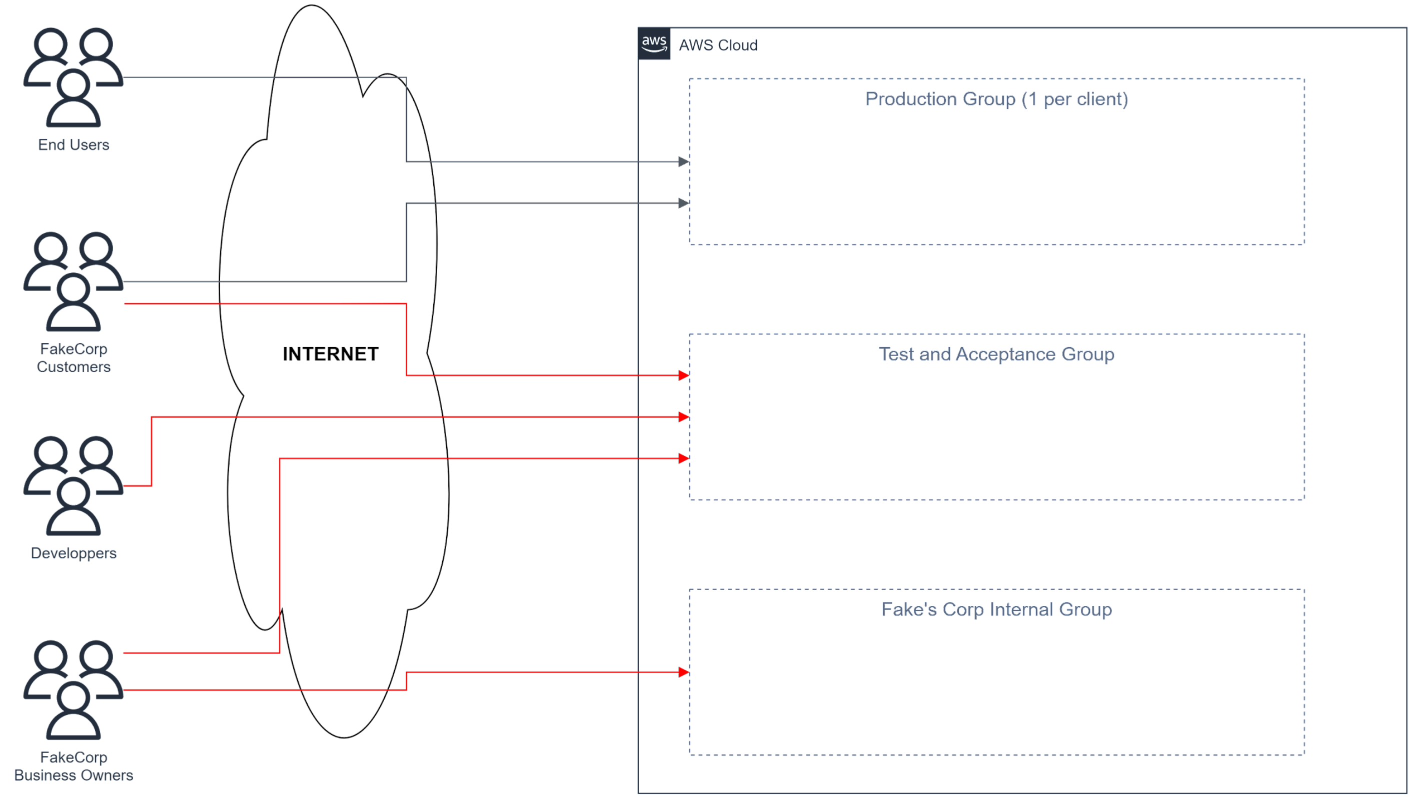
SCHEME

1. General group

Diagram specifications:

The **RED** color represents a VPN connection

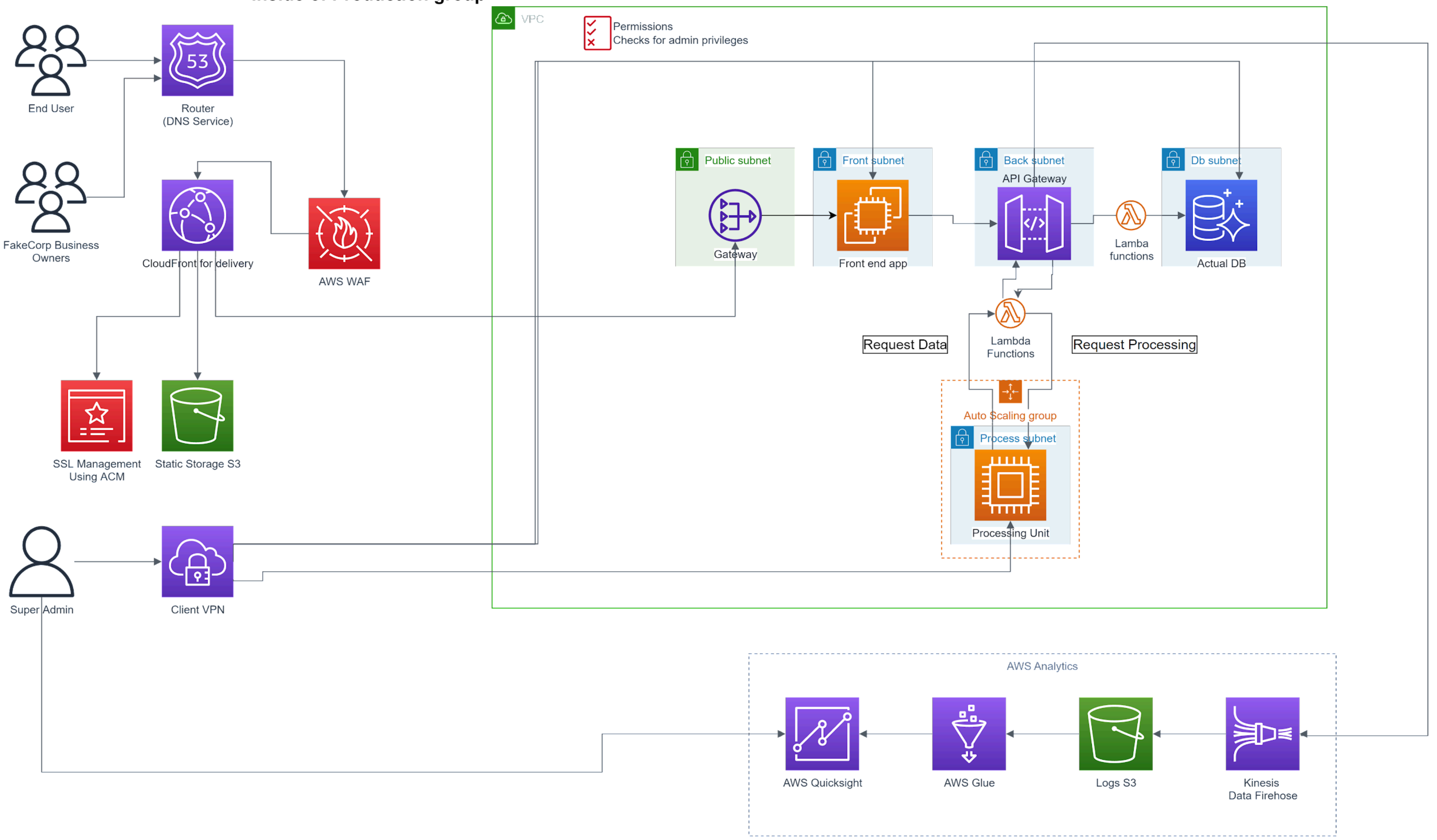
The **BLACK** color represents a simple DNS conenction


End Users
FakeCorp Customers
Developpers
FakeCorp Business Owners
INTERNET
aws
AWS Cloud
Production Group (1 per client)
Test and Acceptance Group
Fake's Corp Internal Group

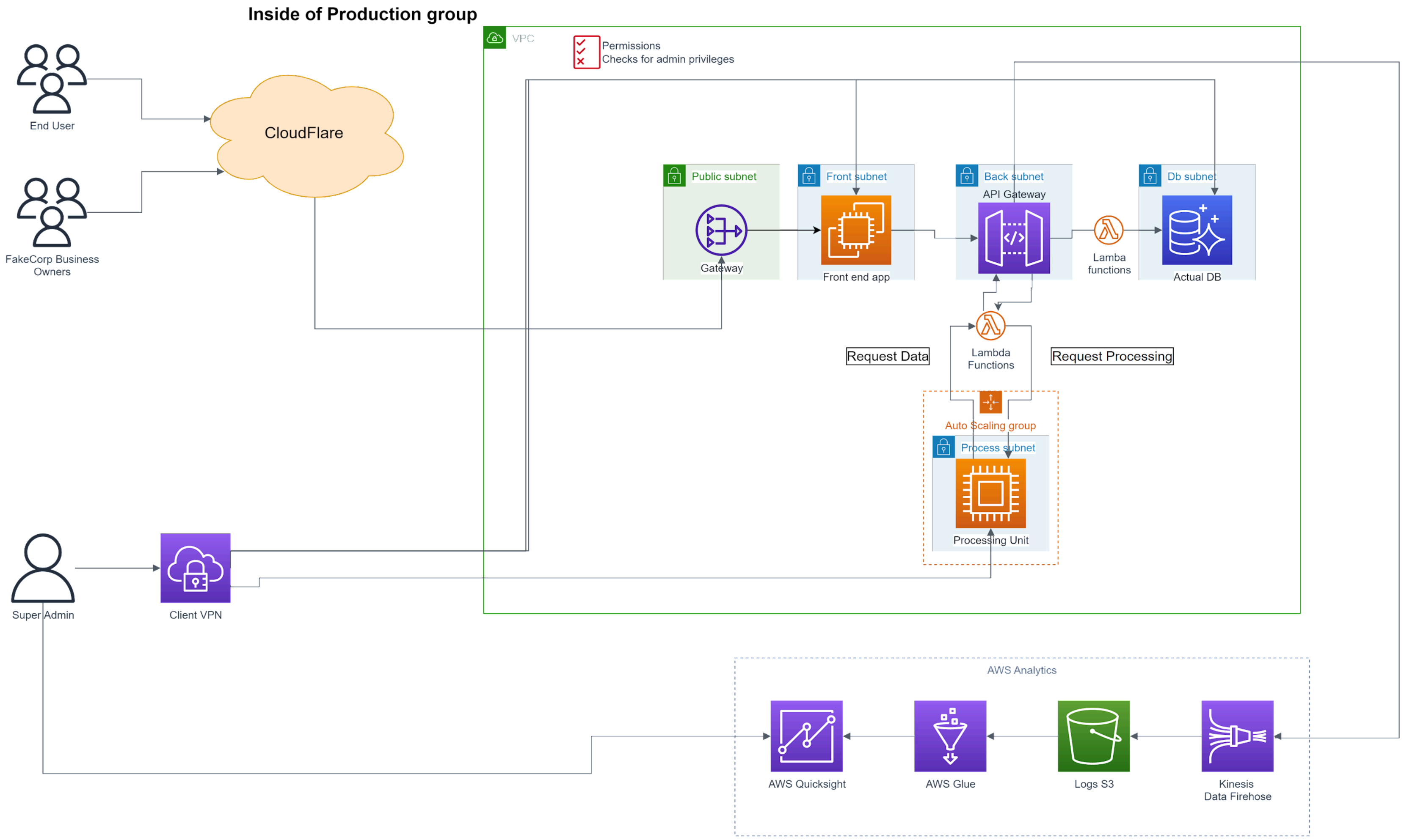
2. Production group

- Full AWS

Inside of Production group

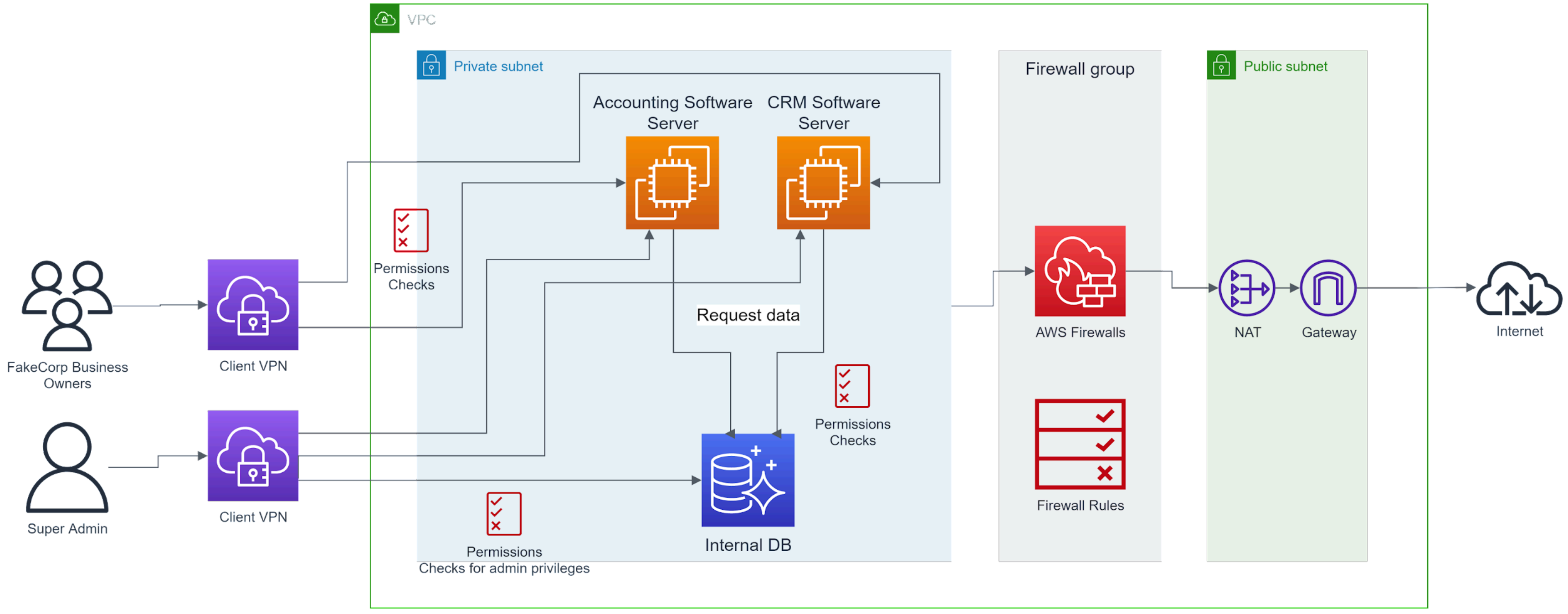


- Using third parties



3.Internal group

Inside of FakeCorp Internal group



4. Test & acceptance group

- Full AWS

Inside of Test and Acceptance group

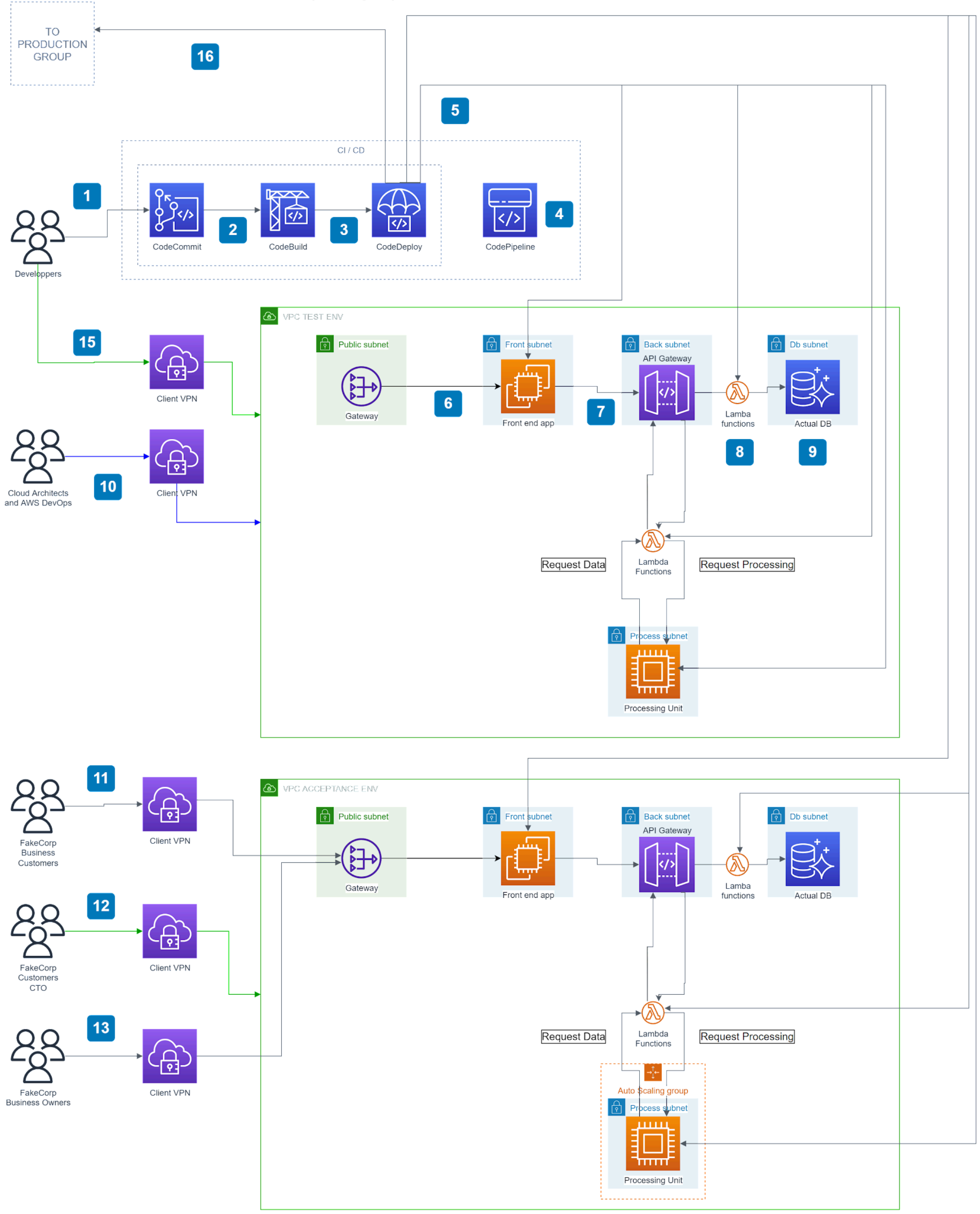


Diagram specifications:

The **BLUE** color represents a full access (read, write, and modify) to **everything** conainted inside the VPC

The **GREEN** color represents a full **READ ONLY** access to **everything** conainted inside the VPC

The **BLACK** color represents a simple access

1 Developpers push their code on **AWS CodeCommit** so that it can be controlled before going any further

2 **AWS CodeBuild** is a continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy on a dynamically-created build server. After the build is successful, the pipeline moves to the deploy stage.

3 **AWS CodeDeploy** is a fully-managed deployment service that automates software deployments to **AWS Fargate**. The deployments contain the code or application (associated to new features to be deployed based on developer changes) running CodeDeploy agents.

4 **AWS CodePipeline** is used to create an end-to-end pipeline that fetches the application code from **CodeCommit**, builds and tests using **CodeBuild**, and finally deploys using **CodeDeploy** from Amazon Elastic Container Registry (Amazon ECR) to the Fargate serverless platform for handling user traffic.

5 If the deployement is a success, **CodeDeploy** will update the content on the **EC2** Front End app, the lambdas, the scheme on the actual db, the compute unit and the **API Gateway** config

6 The **Gateway** is used to allow internet connection from outside to the inside of private subnet

7 The front end application makes call to the **API Gateway** that is in charge of routing these calls to the associated lambda functions

8 **Lambda** functions holds only one function used for the api, they can request data from the **Aurora DB**, or request processing to the **Processing Unit**

9 **Aurora DB** holds the data for this software, since this is a test environnement there is no Backup DB, but in production there is one to ensure data accessibility

10 **Cloud Architects** and the **DevOps** have access to every resources inside the test environnement in order to check that this is made in accordance of what they want. This is done through a VPN to ensure security

11 **FakeCorp Customers Business owners** have access to the delivery front end app to see if this fits their needs throught a VPN

12 **FakeCorp Customers CTO** have a **READONLY** access to the cloud infrastructure to make sure that this is in accordance to what they wanted

13 **FakeCorp Business owners** have access to the delivery front end app to see if they are satisfied with the implemented features

14 In the case of a push on the develop branch, the code and the infrastructure is deployed to the **acceptance** environnement

15 Developpers have a **READONLY** access to see if the infrastructure is working properly

16 After the validation of the "acceptance environnement" and a validation of **FakeCorp's Business Owners** and **FakeCorps Customers, CodeDeploy** can deploy the **acceptance environnement** to the production environnement

- Third parties

Inside of Test and Acceptance group

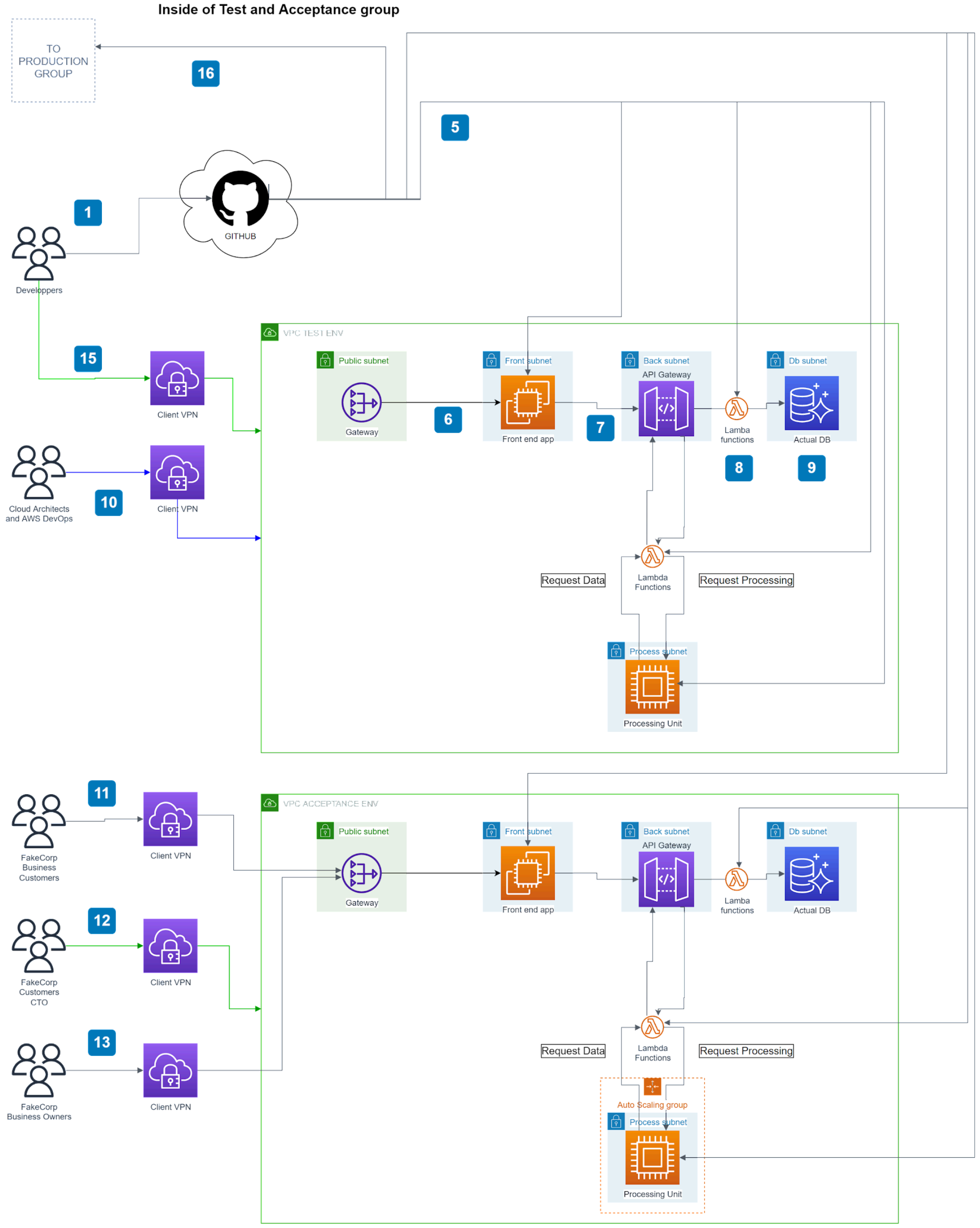


Diagram specifications:

The **BLUE** color represents a full access (read, write, and modify) to **everything** conainted inside the VPC

The **GREEN** color represents a full **READ ONLY** access to **everything** conainted inside the VPC

The **BLACK** color represents a simple access

1 Developpers push their code on **AWS CodeCommit** so that it can be controlled before going any further

5 If the deployement is a success, **CodeDeploy** will update the content on the **EC2** Front End app, the lambdas, the scheme on the actual db, the compute unit and the **API Gateway** config

6 The **Gateway** is used to allow internet connection from outside to the inside of private subnet

7 The front end application makes call to the **API Gateway** that is in charge of routing these calls to the associated lambda functions

8 **Lambda** functions holds only one function used for the api, they can request data from the **Aurora DB**, or request processing to the **Processing Unit**

9 **Aurora DB** holds the data for this software, since this is a test environnement there is no Backup DB, but in production there is one to ensure data accessibility

10 **Cloud Architects** and the **DevOps** have access to every resources inside the test environnement in order to check that this is made in accordance of what they want. This is done through a VPN to ensure security

11 **FakeCorp Customers Business owners** have access to the delivery front end app to see if this fits their needs throught a VPN

12 **FakeCorp Customers CTO** have a **READONLY** access to the cloud infrastructure to make sure that this is in accordance to what they wanted

13 **FakeCorp Business owners** have access to the delivery front end app to see if they are satisfied with the implemented features

14 In the case of a push on the develop branch, the code and the infrastructure is deployed to the **acceptance** environnement

15 Developpers have a **READONLY** access to see if the infrastructure is working properly

16 After the validation of the "acceptance environnement" and a validation of **FakeCorp's Business Owners** and **FakeCorps Customers, CodeDeploy** can deploy the **acceptance environnement** to the production environnement

PRICING

1. Production group

- **Full AWS**

Service	Price (in € / month)
Front End App server (2 vCPUs, 8Gb RAM, 30Gb Storage)	38.90
Api Gateway (1 million requests / month with 100Kb per request)[4]	1.20
Lambda function(1 million requests / month with 100ms of runtime each)[5]	0.00
Processing Unit (2 vCPUs, 16Gb RAM)	59.49
Actual Database (30Gb of storage, with 720Gb of backup storage)	117.41
Route 53 (1 million query / month)[6]	1.50
Web Application Firewall[7]	5.00
Cloudfront (1,024 Gb of traffic / month)[8]	0.00
Static Storage S3 (10 Gb / month)[9]	0.24
SSL ACM[10]	400 (Private CA)
VPN Access for the SuperAdmin	73.36
Quicksight for 1 viewer[11]	26.00
Glue[12]	0.14
Analytics S3 (10 Gb / month)[13]	0.24
Firehose[14]	416

Total estimated price of the production environment (per client): **1139.48 € / month**

- ### Third parties

Service	Price (in € / month)
Front End App server (2 vCPUs, 8Gb RAM, 30Gb Storage)	38.90
Api Gateway (1 million requests / month with 100Kb per request)	1.20
Lambda (1 million requests / month with 100ms of runtime each)	0.00
Processing Unit (2 vCPUs, 16Gb RAM)	59.49
Actual Database (30Gb of storage, with 720Gb of backup storage)	117.41
VPN Access for the SuperAdmin	73.36
Quicksight for 1 viewer	26.00
Glue	0.14
Analytics S3 (10 Gb / month)	0.24
Firehose	416
CloudFlare[15]	0.00

Total estimated price of the production environment (per client): **732.74 € / month**

2. Internal group

Service	Price (in € / month)
CRM server (2 vCPUs, 4Gb RAM, 30Gb Storage)	21.24
Accountant server (2 vCPUs, 4Gb RAM, 30Gb Storage)	21.24
Firewall (160 hours/month, 100Gb a month)	69.7
Actual Database (50Gb of storage, with 1Tb of backup storage)	114.39
VPN Accesses (5)	117.00

Total estimated price of the internal environment : **343.57 € / month**

3.Test and Acceptance group

- **Test part**

Service	Price (in € / month)
Front end server (2 vCPUs, 4Gb RAM, 30Gb Storage)	21.24
Api Gateway (1 million requests / month with 100 Kb per request)	1.20
Lambda (1 million requests / month with 100ms of runtime each)	0.00
Processing Unit (2 vCPUs, 8Gb RAM)	38.9
Actual Database (15Gb of storage)	49.66
VPN Access (10)	161.00

Total estimated price of the test environment : **272 € / month**

- ### Acceptance part

Service	Price (in € / month)
Front end server (2 vCPUs, 8Gb RAM, 30Gb Storage)	38.9
Api Gateway (1 million requests / month with 100 Kb per request)	1.20
Lambda (1 million requests / month with 100ms of runtime each)	0.00
Processing Unit (2 vCPUs, 16Gb RAM)	59.49
Actual Database (15Gb of storage)	49.66
VPN Access (10)	161.00

Total estimated price of the acceptance environment (per client) : **310.25 € / month**

CI/CD[16] Part hosted on Amazon cost this :

Service	Price (in € / month)
CodeCommit[17]	0.00
CodeBuild[18]	3.50
CodeDeploy[19] (4 deployments a day)	24.33
CodePipeline[20] (11 pipelines)	10.00

The full AWS CI/CD estimated costs is : **37.83 € / month**

Whereas Github[21] costs 4 Euros per user which would be: **40 € / month**

TECHNICAL SPECIFICATIONS

1. Cloud provider

In order to choose a correct cloud provider for your infrastructure we only have two options: Microsoft Azure and Amazon Web Services, here is a quick comparison between these two.

Criteria	Microsoft Azure	Amazon AWS
Marketshare	21%	33%
ISO 27001[21]	Yes	Yes
IAM Pricing	Free (with Premium options)	Free
Creation date	2010	2006
SLA[22]	> 99%	> 99%

Because AWS has more market share and a bigger community we decided to choose this one, since it will be easier to recruit people to maintain it in the future.

2.Operating systems

When it comes to choosing an operating system to host the solution on it we have three options : Windows, Linux or MacOS. Since MacOs doesn't offer a proper enterprise solution we will only compare Linux and MacOS.

Criteria	Microsoft Windows	Linux
Free	No	Yes
OpenSource	No	Yes
Server market share	72%	14%
Remote control out of the box	Yes	No

We know that Microsoft Windows is better for this solution but since internally we don’t have any skills related to this OS we will choose Linux to be the base of every server related to this architecture. We will use CentOs 7.0-2009 as it’s the latest LTS[23] version.

3. General group

There will be one production group per customer accessible via the Internet. Each group is independent from the other ones so that if one group gets corrupted the other are still safe.

The test and acceptance group is only available through a VPN because it is not designed to be made public to prevent unauthorized access to the test environment or the acceptance one.

The internal group is also only accessible through a VPN because it should be accessible to FakeCorp business people or to FakeCorp SuperAdmin if a problem occurs within this group.

4.Production group

End users and FakeCorp business customers go through a DNS Service to route them to the application, it also implements a firewall to prevent some common web vulnerabilities and DDoS attacks to the production VPC[24].

CloudFront is used as a Content Delivery Network[25] to deliver the front end application securely and with low latency to the end users. It uses ACM to verify and store the SSL certificate[26].

It is connected to an Amazon S3 which holds the already rendered web application.

This whole process is used to reduce latency and to improve security when users want to access the web application.

The gateway allows users to access the front end application located in a private subnet.

The front end application can then make requests to the API Gateway which will handle the request through lambda functions. So that you only pay for what's really used instead of renting another EC2[27] to hold your backend implementation.

Lambda functions hold the logic and are the only ones who can access the compute region and the database instance. The computer region is auto scaled to never make the user wait for its resources.

The database instance is saved every month and the data is kept for 2 years.

Here is the table for the machine sizes :

Machine Name	Specifications
Front End App Server	2 vCPUS, 8Gb of RAM, 30Gb of storage
Processing Unit	4 vCPUS, 16Gb of RAM, 30Gb of storage
Aurora Database	2 vCPUS, 4Gb of RAM, 30Gb of storage, 720Gb of backup

5. Internal group

FakeCorp has one group for its internal softwares, it can only be accessed by a VPN and only FakeCorp business people and one super admin have access to this group.

This is where FakeCorp softwares are securely hosted.

The VPN does permission checks to make sure that only the correct business people have access to the correct servers.

FakeCorp accountant has access to the accounting service.

FakeCorp sales manager has access to the CRM service.

FakeCorp admin has access to both.

These two servers share the same database but they only have access to certain parts of this database to protect each software data.

All of this is held in a private subnet and the only way for the accounting and the CRM software to communicate with the Internet is to go through the firewall to prevent unwanted requests to the outside environment.

A SuperAdmin has access to everything inside this environment in case of a failure of the infrastructure.

The database instance is saved every month and the data is kept for 2 years.

Here is the table for the machine sizes :

Machine Name	Specifications
Accountant Software Server	2 vCPUS, 4Gb of RAM, 30Gb of storage
CRM Software Server	2 vCPUS, 4Gb of RAM, 30Gb of storage
Aurora Database	2 vCPUS, 4Gb of RAM, 50Gb of storage, with 1Tb of backup storage

6. Test and Acceptance Group

Every technical aspect of this group is noted on the scheme.

The database does not have any backup storage since this is a not production environment.

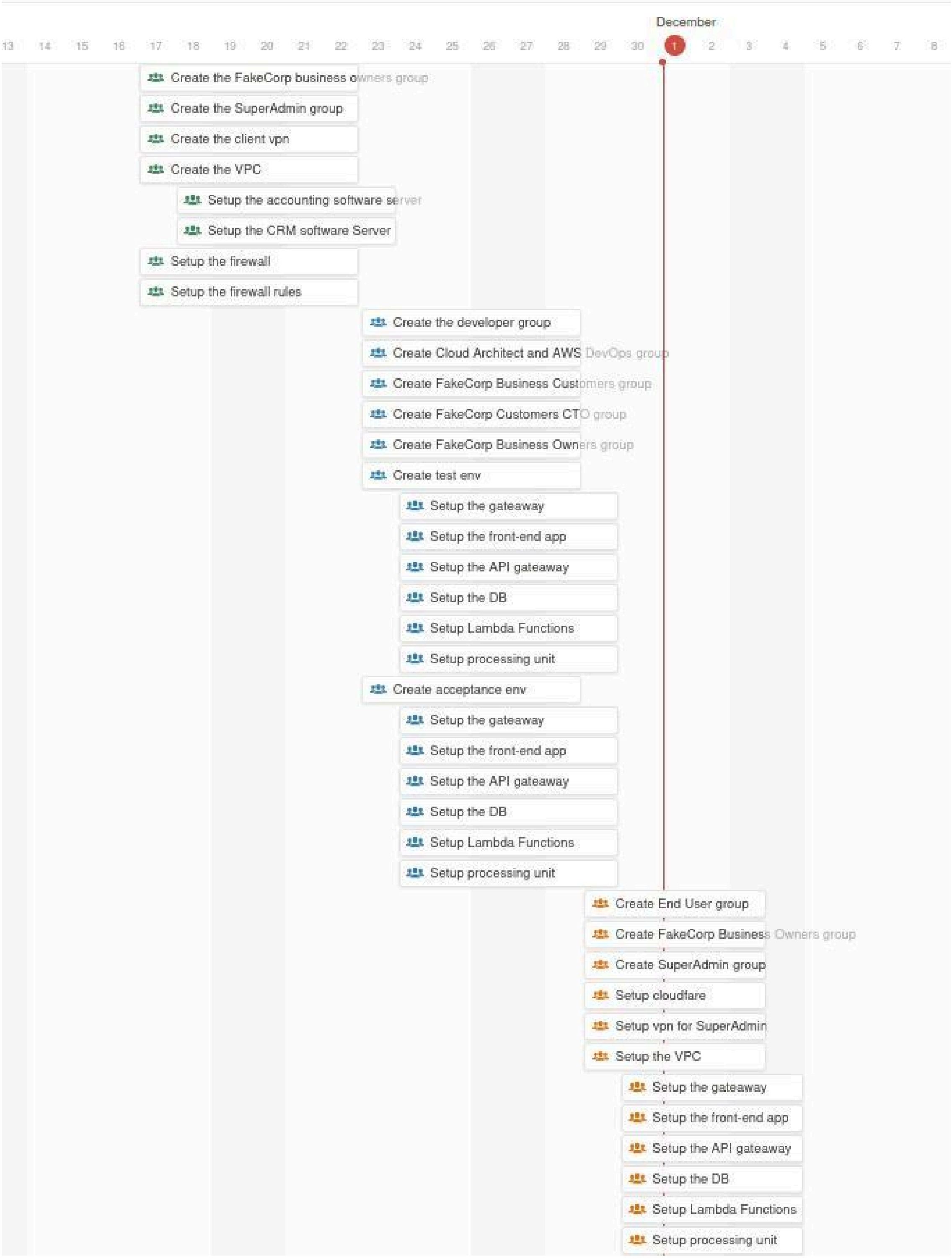
Here is the table for the machine sizes :

Machine Name	Specifications
Front end app (Test server)	2 vCPUS, 4Gb of RAM, 30Gb of storage
Front end app (Acceptance server)	2 vCPUS, 8Gb of RAM, 30Gb of storage
Aurora Database (Both servers)	2 vCPUS, 4Gb of RAM, 15Gb of storage, with no backup storage
Compute unit (Test server)	2 vCPUS, 8Gb of RAM, 30Gb of storage
Compute unit (Acceptance server)	2 vCPUS, 16Gb of RAM, 30Gb of storage

7. IAM Table

	End Users of customers	FakeCorp Business Owners	SuperAdmin	Developers	FakeCorp Customers CTO	FakeCorp Customers
Accounting		X	X			
CRM		X	X			
Production API			X			
Production Front End	X	X	X		X	X
Production DB			X			
Production Compute Unit			X			
Test Front End			X	X		
Test DB			X	X		
Test API			X	X		
Test Compute Unit			X	X		
Acceptance API			X		X	
Acceptance Front End		X	X		X	X
Acceptance DB			X		X	
Acceptance Compute Unit			X		X	
CI/CD to deploy code			X	X		

PLANNING



Create the FakeCorp business owners group

Create the SuperAdmin group

Create the client vpn

Create the VPC

Setup the accounting software server

Setup the CRM software Server

Setup the firewall

Setup the firewall rules

Create the developer group

Create Cloud Architect and AWS DevOps group

Create FakeCorp Business Customers group

Create FakeCorp Customers CTO group

Create FakeCorp Business Owners group

Create test env

Setup the gateaway

Setup the front-end app

Setup the API gateaway

Setup the DB

Setup Lambda Functions

Setup processing unit

Create acceptance env

Setup the gateaway

Setup the front-end app

Setup the API gateaway

Setup the DB

Setup Lambda Functions

Setup processing unit

Create End User group

Create FakeCorp Business Owners group

Create SuperAdmin group

Setup cloudfare

Setup vpn for SuperAdmin

Setup the VPC

Setup the gateaway

Setup the front-end app

Setup the API gateaway

Setup the DB

Setup Lambda Functions

Setup processing unit

GLOSSARY

1. **Identity and Access Management(IAM):** Framework of policies and technologies to ensure that the right users (that are part of the ecosystem connected to or within an enterprise) have the appropriate access to technology resources.
2. **On-premise environment:** On-premise software is installed and operated from the software customer's own data-center, a group of servers that you privately own and control.
3. **Cloud infrastructure:** Components needed for cloud computing, which includes hardware, abstracted resources, storage, and network resources. Cloud computing is the act of running workloads within clouds—which are IT environments that abstract, pool, and share scalable resources across a network. Neither cloud computing nor clouds are technologies unto themselves.
4. **API gateway:** Provides a unified entry point across internal APIs. It allows you to control user access, and it enables security measures, like rate limiting, and applies security policies.
5. **Lambda function:** Lambda is a compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging.
6. **Route 53:** Amazon Route 53 is a scalable and highly available Domain Name System service. It translates human readable domain names (for example, www.amazon.com) to machine readable IP addresses (for example, 192.0. 2.44).
7. **Web Application firewall:** A Firewall is a network security device that monitors and filters incoming and outgoing network traffic based on an organization's previously established security policies. At its most basic, a firewall is essentially the barrier that sits between a private internal network and the public Internet.
8. **CloudFront:** Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users. CloudFront delivers your content through a worldwide network of data centers called edge locations.
9. **Static Storage S3:** Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. You can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere.
10. **SSL ACM:** AWS Certificate Manager (ACM) is a service that lets you easily provision, manage, and deploy public and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS services and your internal connected resources.

11. **Quicksight:** Amazon QuickSight is a cloud-scale business intelligence (BI) service that you can use to deliver easy-to-understand insights to the people who you work with, wherever they are. Amazon QuickSight connects to your data in the cloud and combines data from many different sources.
12. **Glue:** AWS Glue is a serverless data integration service that makes it easier to discover, prepare, move, and integrate data from multiple sources for analytics, machine learning (ML), and application development.
13. **Analytics S3:** S3 Storage Class Analysis enables you to monitor access patterns across objects to help you decide when to transition data to the right storage class to optimize costs. You can then use this information to configure an S3 Lifecycle policy that makes the data transfer.
14. **Firehose:** Amazon Kinesis Firehose captures and loads streaming data in storage and business intelligence (BI) tools to enable near real-time analytics in the Amazon Web Services (AWS) cloud.
15. **CloudFlare:** CloudFlare is a company. Fundamentally, is a large network of servers that can improve the security, performance, and reliability of anything connected to the Internet. Cloudflare does this by serving as a reverse proxy.
16. **CI/CD:** CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development. The main concepts attributed to CI/CD are continuous integration, continuous delivery, and continuous deployment.
17. **CodeCommit:** CodeCommit is a secure, highly scalable, managed source control service that hosts private Git repositories. CodeCommit eliminates the need for you to manage your own source control system or worry about scaling its infrastructure. You can use CodeCommit to store anything from code to binaries.
18. **CodeBuild:** CodeBuild compiles your source code, runs unit tests, and produces artifacts that are ready to deploy. CodeBuild eliminates the need to provision, manage, and scale your own build servers.
19. **CodeDeploy:** AWS CodeDeploy is a service that automates code deployments to any instance, including Amazon EC2 instances and instances running on-premises.
20. **CodePipeline:** AWS CodePipeline is a continuous delivery service you can use to model, visualize, and automate the steps required to release your software. You can quickly model and configure the different stages of a software release process. CodePipeline automates the steps required to release your software changes continuously.
21. **GitHub:** GitHub is a company that offers a cloud-based Git repository (version control software) hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration.
22. **ISO 27001:** ISO 27001 is the only auditable international standard that defines the requirements of an ISMS (information security management system). An ISMS is a set of policies, procedures, processes and systems that manage information security risks, such as cyber attacks, hacks, data leaks or theft.
23. **SLA:** A service-level agreement (SLA) sets the expectations between the service provider and the customer and describes the products or services to be delivered, the single point of contact for end-user problems, and the metrics by which the effectiveness of the process is monitored and approved.

24. **VPC:** A virtual private cloud (VPC) is a secure, isolated private cloud hosted within a public cloud. VPC customers can run code, store data, host websites, and do anything else they could do in an ordinary private cloud, but the private cloud is hosted remotely by a public cloud provider.
25. **CDN (Content Delivery Network):** A CDN is a network of servers that distributes content from an “origin” server throughout the world by caching content close to where each end user is accessing the internet via a web-enabled device. The content they request is first stored on the origin server and is then replicated and stored elsewhere as needed.
26. **SSL certificate:** An SSL certificate is a bit of code on your web server that provides security for online communications. When a web browser contacts your secured website, the SSL certificate enables an encrypted connection.
27. **EC2:** Amazon Elastic Compute Cloud (EC2) is a part of AWS, that allows users to rent virtual computers on which to run their own computer applications.