everis

an NTT DATA Company

POCs de migración de modelos analíticos a la Nube

Arquitectura Big Data

POC

Enero 2019

aws

Contenido

1. Everis: Soluciones Cloud Big Data
2. Capacidades Analíticas Cloud
3. POC AWS
4. Experiencias en soluciones de AI

1. Everis: Soluciones Cloud Big Data

- Cómo lo hacemos?
- Por qué Cloud?
- Herramientas AWS.

Un paseo por la nube

Principios básicos arquitectura Cloud

everis
an NTT DATA Company

La arquitectura será implementada o proporcionada en base a los siguientes principios rectores para garantizar la robustez de la solución:

Escalabilidad

Arquitectura e infraestructura que permita la escalabilidad del modelo pudiendo empezar con entornos o componentes con alcance reducido y pudiendo crecer y escalar minimizando el impacto en los sistemas productivos.

Disponibilidad

Compromiso de disponibilidad de la solución. Se debe tener en cuenta la criticidad de la misma para adaptar la infraestructura y arquitectura a las necesidades de alta disponibilidad, disaster recovery, etc...

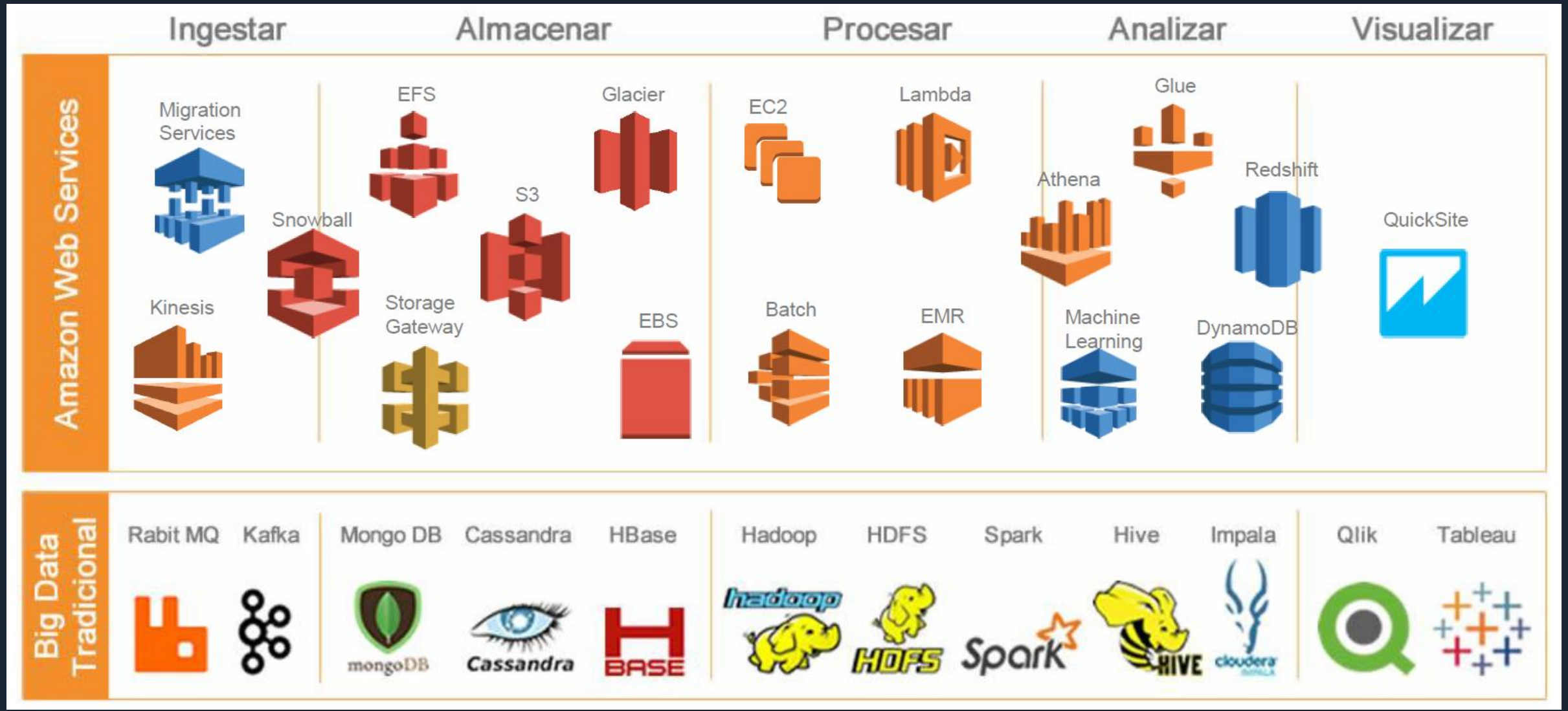
Rendimiento

A nivel de rendimiento se debe definir una arquitectura que permita optimizar el rendimiento en función de los casos de uso y las necesidades de explotación.

Fiabilidad

Los componentes de la solución a todos los niveles deben ser fiables, robustos y maduros para garantizar la fiabilidad de la misma.

Amazon Web Services (AWS)



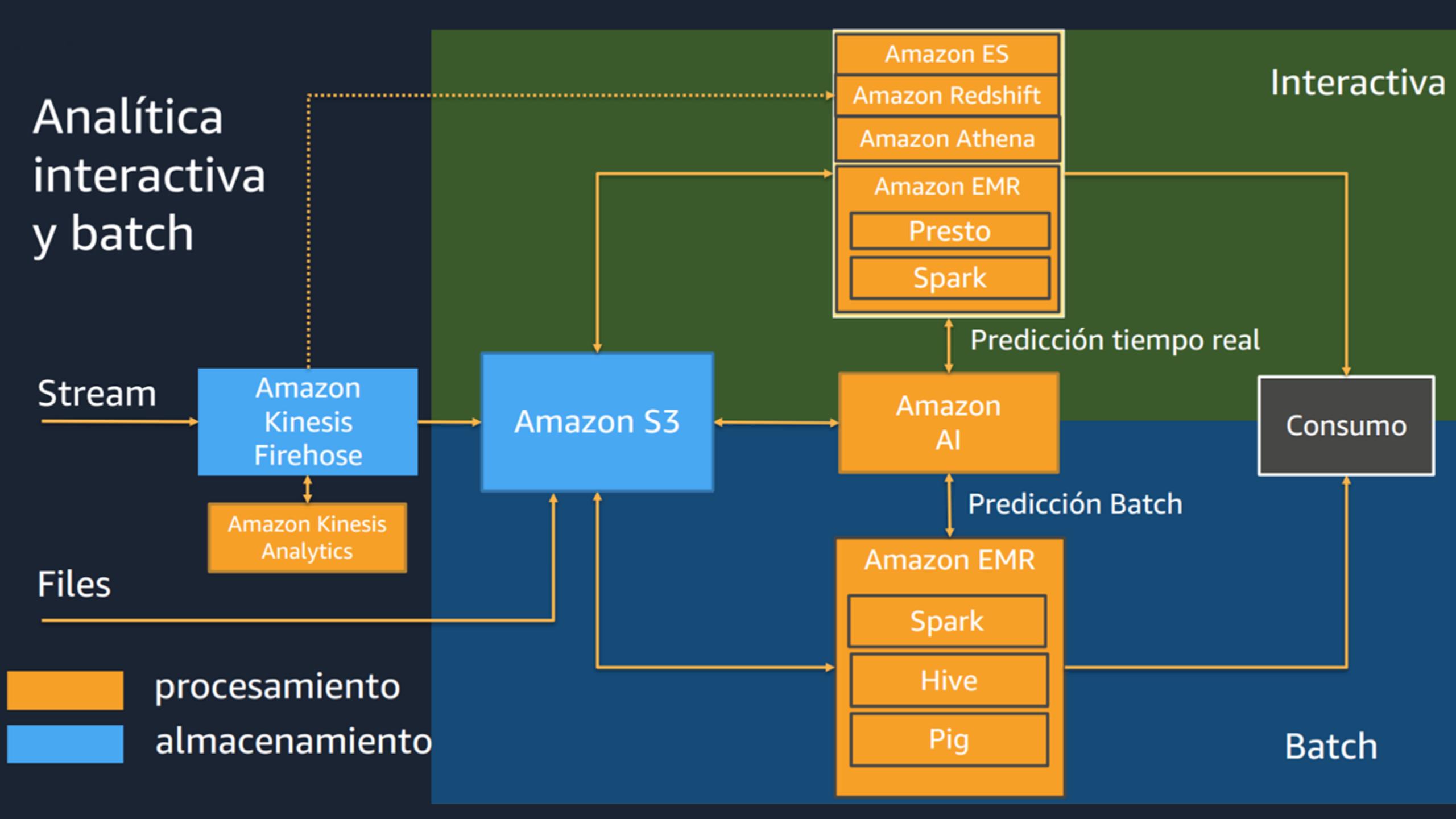
2. Capacidades Analíticas Cloud

- **Tecnológicas.**
- **Analíticas.**
- **Producción.**

Capacidades Tecnológicas

everis
an NTT DATA Company

Analítica en tiempo real


Analítica interactiva y batch
Amazon ES
Amazon Redshift
Amazon Athena
Amazon EMR
Presto
Spark
Interactiva
Predicción tiempo real
Stream
Amazon Kinesis Firehose
Amazon S3
Amazon AI
Consumo
Amazon Kinesis Analytics
Predicción Batch
Files
Amazon EMR
Spark
Hive
Pig
procesamiento
almacenamiento
Batch

Capacidades Analíticas

- Desarrollo de Procesos Analíticos.
  - Batch.
  - Real Time.
- Ambientes de desarrollo colaborativo.
- Procesos Analíticos sobre Big Data.
- Frameworks optimizados para Machine Learning.
- Modelos pre-entrenados de deep learning.

- Servicios cognitivos de alto nivel.
  - Speech Analytics y NLP.
  - Traducción Automática.
  - Reconocimiento de Objetos.
  - Sintonización de Voz.
  - Transcripción de Voz a Texto.

Capacidades Analíticas

Ground Truth

Set up and manage labeling jobs for highly accurate training datasets using active learning and human labeling.

- Labeling jobs

Notebook

Availability of AWS and SageMaker SDKs and sample notebooks to create training Jobs and deploy models.

- Notebook instances

Training

Train and tune models at any scale. Leverage high performance AWS algorithms or bring your own.

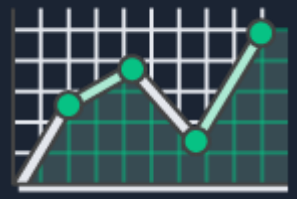
- Training jobs
- Hyperparameter tuning jobs

Inference

Create models from training jobs or import external models for hosting to run inferences on new data.

- Models
- Endpoints
- Batch transform jobs

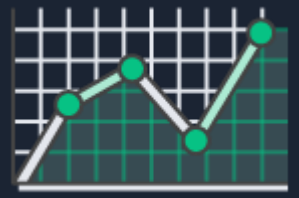
Capacidades de Producción



- Velocidad

- Colaboración Mejorada

- Entrega Inmediata



- Escalado

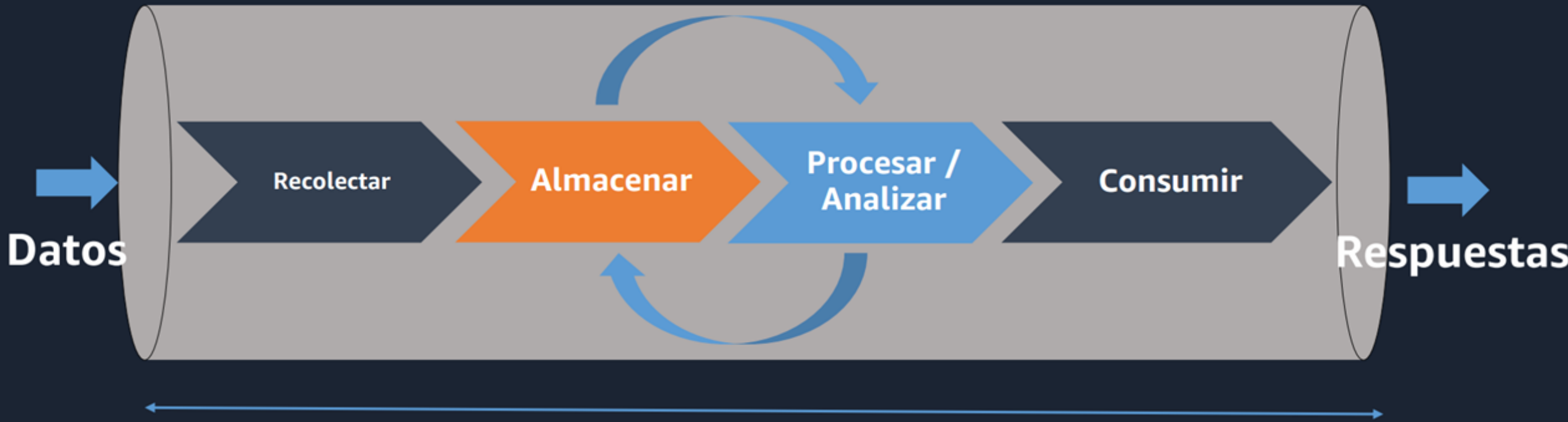
- Fiabilidad

- Seguridad

3. POC – Cloud AWS

- Proceso Big Data
- POC Cloud Deployment
- POC Big Data & Analytics
- POC Real Time

Proceso Big Data



Tiempo para responder (Latencia)

Desempeño

Costo

POC Cloud Deployment

- **Objetivo:**

  Ejecución de un proceso analítico en la nube para evaluar su despliegue y rendimiento.

- **Objetivo especifico:**

  Ejecutar en la nube de Amazon scripts en R, que aplican una serie de modelos pre-entrenados sobre un conjunto de registros almacenados en archivos con formato CSV. Con esto se desea probar:

  - Facilidad del despliegue
  - Uso de herramientas de desarrollo en línea
  - Evaluación y comparación de rendimientos

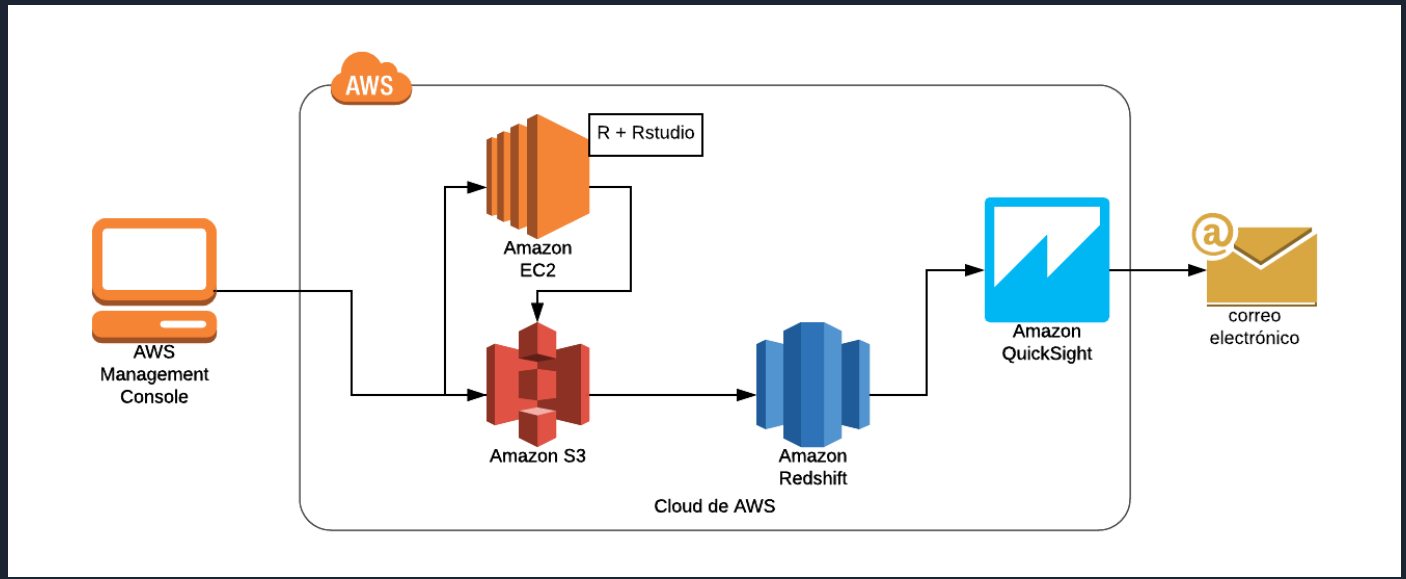
- **Información suministrada:**

  - Dos scripts en R.
  - Seis modelos de arboles de decisión (C50) para predecir niveles de riesgo en la asignación de tarjetas de crédito, almacenados en formato RDS.
  - 13 archivos de datos de personas en formato CSV, 10 archivos con datos de “No clientes” y 4 archivos con datos de “Clientes”.

Flujo – POC Cloud Deployment

La Arquitectura propuesta para la POC de. Consiste en un Patrón Fan-Out. De manera que se pueda:

- Creación de script (CloudFormation) que crea todas las instancias necesarias en la nube
- Migrar todos scripts modelos y datos locales a un Bucket S3
- Crear el Proceso ETL en Lambda que pasa los scripts a una instancia de EC2 con R+Rstudio que ejecuta los scripts.
- Persistir los resultados del Proceso analítico en el S3.
- Crear un ETL con Lambda para pasar los resultados del S3 a Redshift para poder realizar la visualización de forma óptima.
- Utilizar QuickSite para consultar los resultados en el Redshift realizar la visualización.



POC – Paralelización de Código

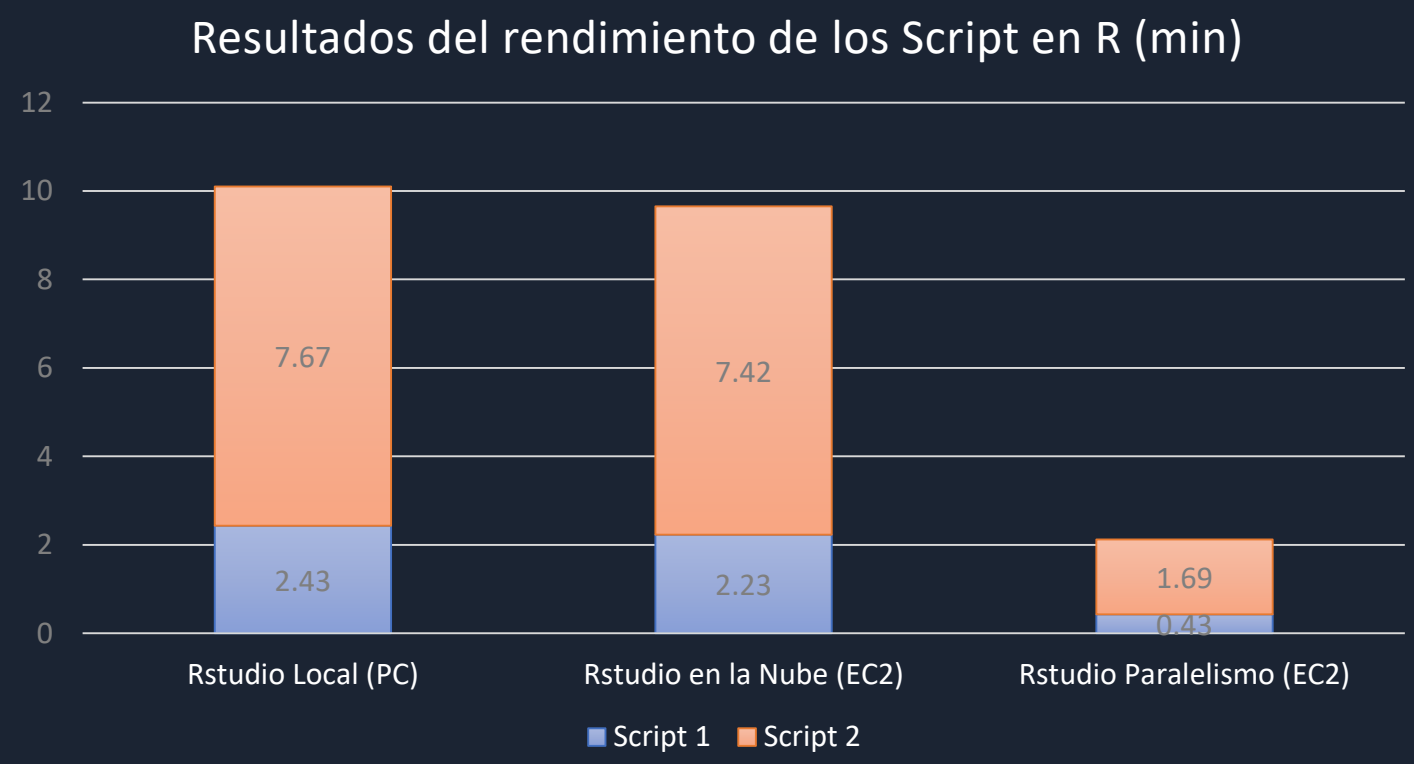
- Recibimos un par de scripts en R con la ejecución serial de tres modelos sobre un conjunto de archivos de datos.

```
################################################################################
# Market Predictive Model
################################################################################
#library(C50)
# Upload Model
#credit_model_tc_c <- readRDS("../../Modelo R/credit_model_tc_c.rds")
#credit_model_tc_c <- s3readRDS(object = "s3://everis-ibk-poc-bigdata/IBK-POC/Modelo R/credit_model_tc_c.rds")

#summary(credit_model_tc_c)
# Create a factor vector of predictions on test data
credit_pred_tc_c_p <- predict(credit_model_tc_c,data_tc_c,type="prob")
# Add the first probability (Market probability)
prob_market_c<-credit_pred_tc_c_p[,2]
# Consolidate matrix with score
data_tc_c_2_0<-cbind(data_tc_c,prob_market_c)
data_tc_c_2_0<-data_tc_c_2_0[,-180]
```

- Para mejorar el desempeño de los procesos y evidenciar el efecto del **escalamiento vertical** se modificó el código para que soporte el paralelismo que provee la plataforma AWS.

```
# Parallelism
library(doParallel)
no_cores <- detectCores() - 1
cl <- makeCluster(no_cores, type="FORK")
registerDoParallel(cl)

foreach(i=1:4) %dopar% {
  tmp <- data_tc_list[i]
  data_tc_c <- data.frame(tmp)

  ################################################################################
  # Market Predictive Model
  ################################################################################
  credit_pred_tc_c_p <- predict(credit_model_tc_c,data_tc_c,type="prob")
  # Add the first probability (Market probability)
  prob_market_c<-credit_pred_tc_c_p[,2]
  # Consolidate matrix with score
  data_tc_c_2_0<-cbind(data_tc_c,prob_market_c)
  data_tc_c_2_0<-data_tc_c_2_0[,-180]
...
  ################################################################################
  # Export Predictive Model
  ################################################################################
  #write.table(data_tc_c_2_4,file="../Resultado/MES_201608CLI_1.csv", sep="|")

  s3write_using(data_tc_c_2_4, write.table, sep ="|", row.names = FALSE, object = outfile[i])
}

stopCluster(cl)
```

POC – Escalamiento Vertical



	Rstudio Local (PC)	Rstudio en la Nube (EC2)	Rstudio - Paralelismo (EC2)
	(4 cores, 8Gb RAM)	(16 cores, 16 Gb)	(16 cores, 16 Gb)
Script 1 (Clientes)	2.43	2.23	0.43
Script 2 (No Clientes)	7.67	7.42	1.69

POC Big Data & Analytics

- **Objetivo:**

  Ejecución de un proceso analítico en batch sobre un volumen grande de datos para evaluar las capacidades y el efecto del escalamiento horizontal.

- **Objetivo especifico:**

  Ejecutar en la nube de Amazon scripts en Python, que aplique un proceso analítico de reducción de dimensionalidad sobre un conjunto de datos grande en formato CSV. Con esto se desea probar:

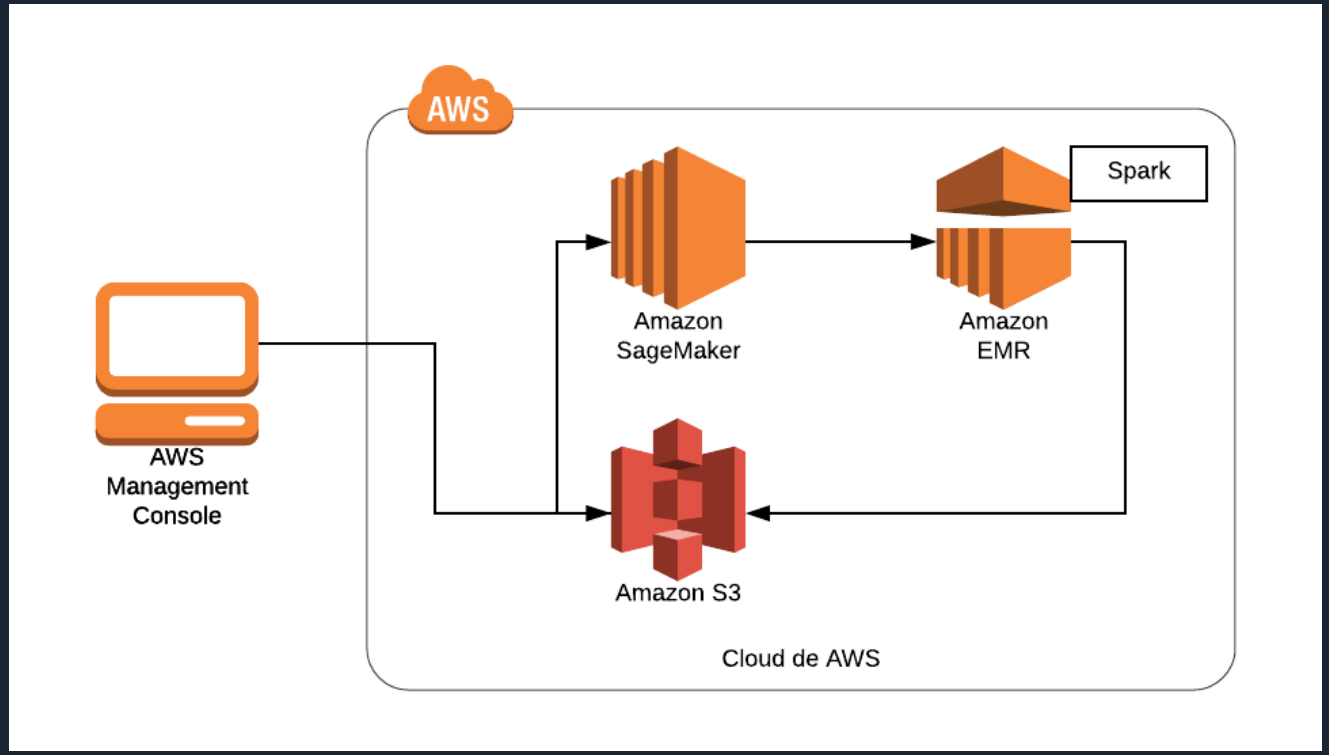
  - Capacidad de manejo de grandes volúmenes de datos
  - Uso de plataformas de procesamiento distribuido (EMR)
  - Uso de frameworks de machine learning (Amazon ML) y
  - Uso de frameworks de procesamiento distribuido (Spark)

- **Información suministrada:**

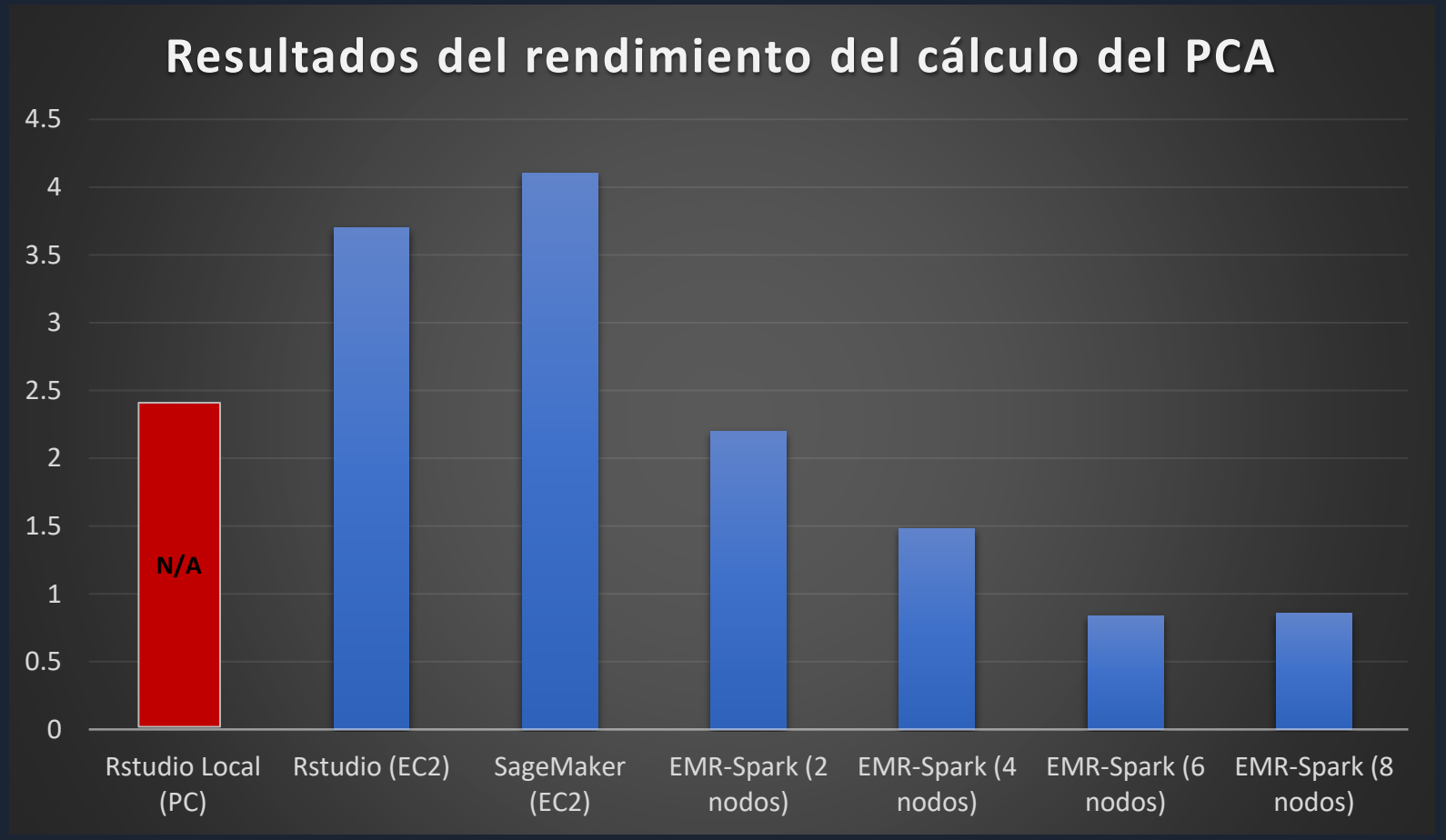
  Un archivo con información de personas y un volumen de 6Gb, con 10.180.084 registros y 109 variables.

Flujo – POC Big Data & Analytics

- Creación de script (CloudFormation) que crea todas las instancias necesarias en la nube.
- Migrar el archivo de datos locales a un Bucket S3.
- Crear el Proceso ETL en Lambda que pasa los scripts a una instancia de SageMaker con Python y Spark que ejecuta una función de PCA sobre los datos.
- Persiste los resultados del Proceso analítico en el S3.



POC – Escalamiento Horizontal



Rstudio Local (PC)	Rstudio (EC2)	SageMaker (EC2)	EMR -Spark (2 nodos)	EMR -Spark (4 nodos)	EMR -Spark (6 nodos)	EMR -Spark (8 nodos)
(4 cores, 8 Gb Ram)	(16 Vcpu, 64 Gb Ram)	(16 Vcpu, 64 Gb Ram)	(16 cores, 16 Gb) x 2	(16 cores, 16 Gb) x 4	(16 cores, 16 Gb) x 6	(16 cores, 16 Gb) x 8
0 min	3.7 min	4.1 min	2.2 min	1.48 min	50.16 seg	51.47 seg

POC – Analítica en Real Time

- **Objetivo:**

  Ejecución de un proceso de desarrollo analítico para tiempo real.

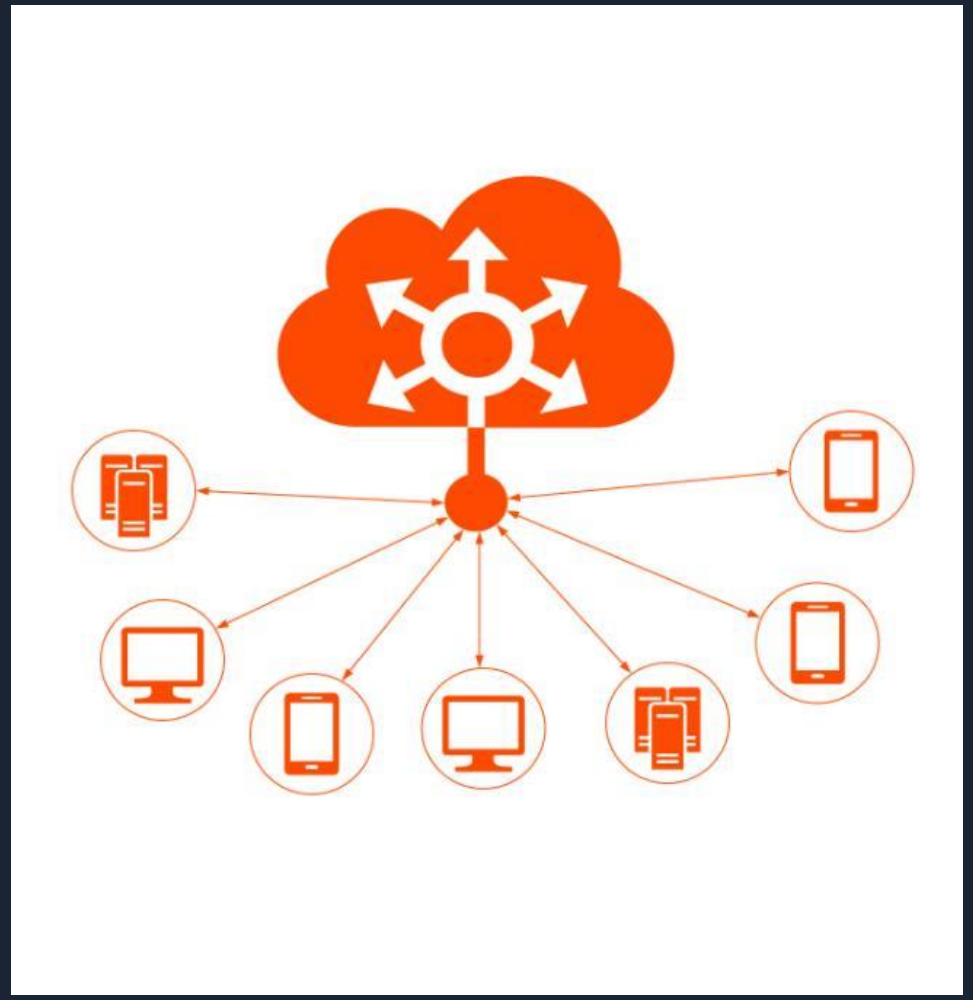
- **Objetivo especifico:**

  Ejecutar en la nube de Amazon un proceso de construcción de un modelo analítico predictivo y disponibilizarlo a través de una plataforma de consulta del modelo en tiempo real. Con esto se desea probar:

  - Uso de Amazon ML y verificar sus ventajas en el entrenamiento de modelos de machine learning.
  - Disponibilización de modelos predictivos para consultas de baja latencia.

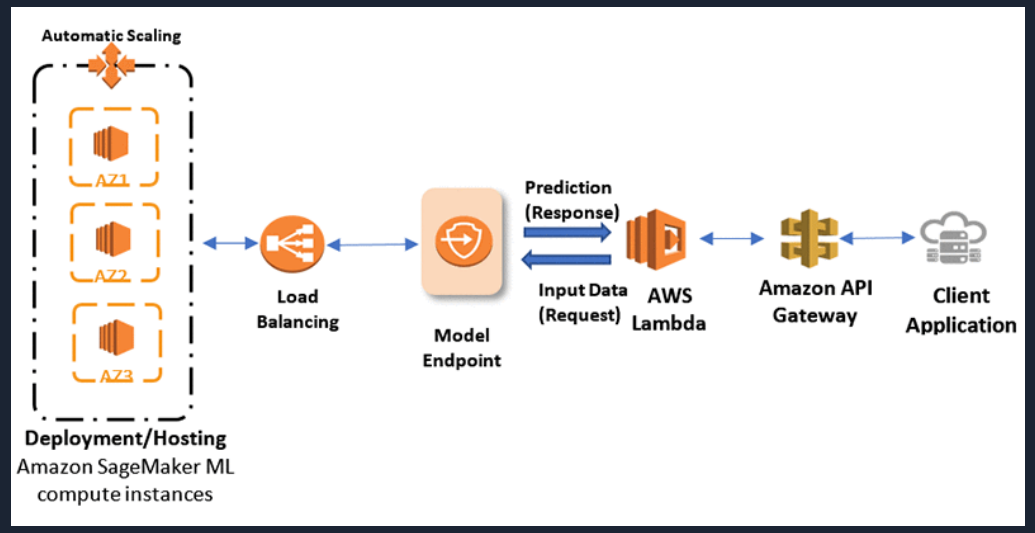
- **Información suministrada:**

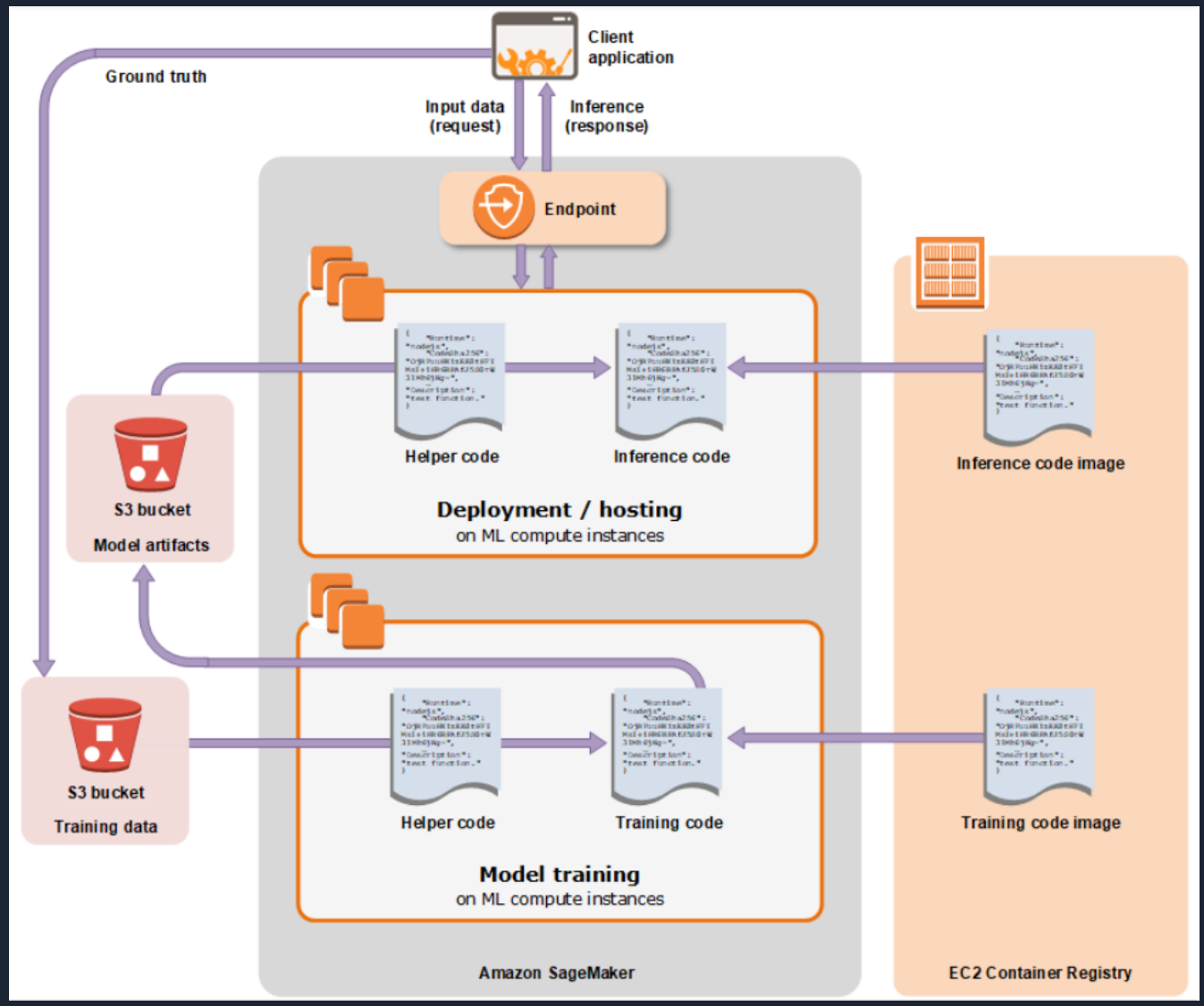
  Un archivo con información de crímenes ocurridos en la ciudad de San Francisco en el año 2014 – 2016 con un volumen de 1Gb, con 6.779.002 registros y 22 variables.



Flujo – POC Analítica en Real Time

- Migrar el archivo de datos locales a un Bucket S3
- Entrenar una modelo predictivo en el **Sagemaker** con el **Amazon ML.**
- Persistir los resultados del Proceso analítico en el S3.
- Disponibilizar el modelo en la plataforma de consulta de modelos analíticos en tiempo real.





Conclusiones

- Aprovisionamiento acelerado de infraestructura y plataformas.
- Capacidades de escalamiento (horizontal y vertical) mejoran el rendimiento del proceso de entrenamiento.
- Las arquitecturas de Big Data disponibles permiten rabajar con data masiva para el entrenamiento de modelos.
- El uso de platformas de microservicios permite disponibilizar modelos con baja latencia y facilitar su integración en todo tipo de sistemas.
- Se requiere definir una arquitectura unificada para el desarrollo y disponibilización de modelos analíticos y de IA.

everis
an NTT DATA Company

Muchas Gracias!

- E-mail: josersosa@gmail.com
- Twitter: @JoserSosaB
- LinkedIn: https://ve.linkedin.com/in/josersosa
- GitHub: https://github.com/josersosa