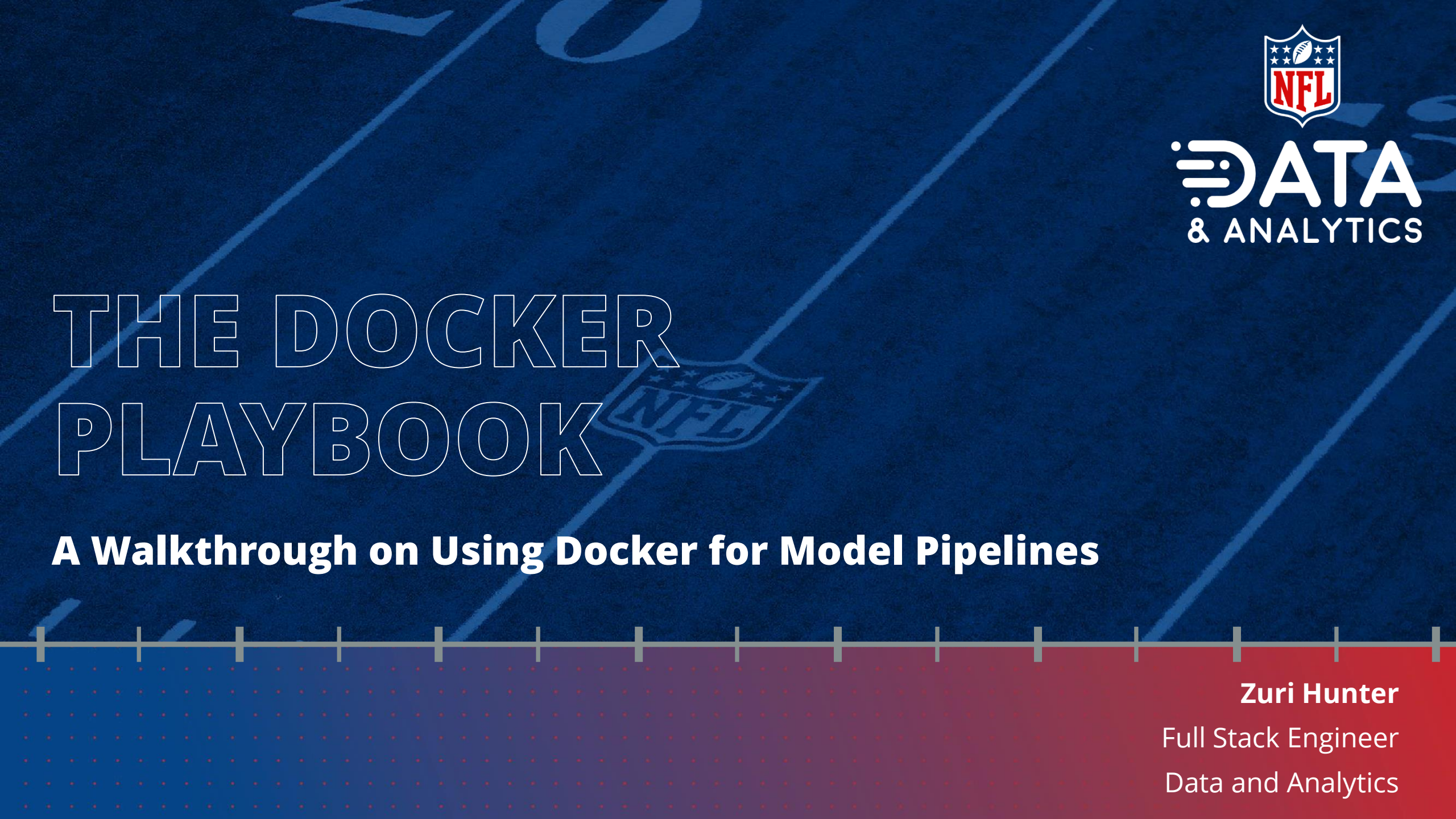


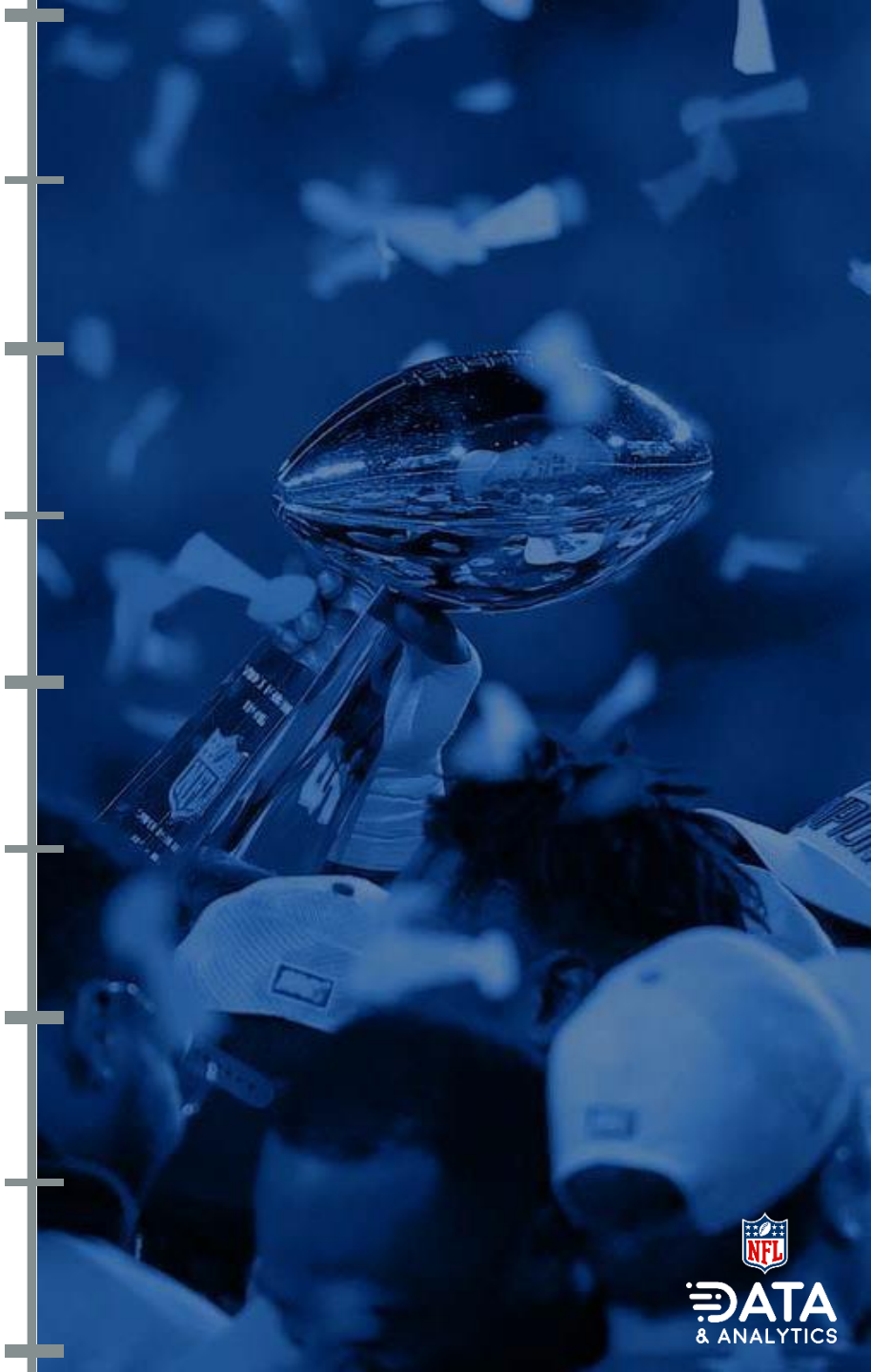
Zuri Hunter
Full Stack Engineer
Data and Analytics

Zuri Hunter

Full Stack Engineer
Data Analytics

WHAT'S IN THE PLAYBOOK?

- Deployment Dilemma
- Docker Fundamentals
- Containerize the Model Pipeline
- Cloud and Terraform Fundamentals
- Code Walkthrough
- Q&A



The Deployment Dilemma

Dependency Hell

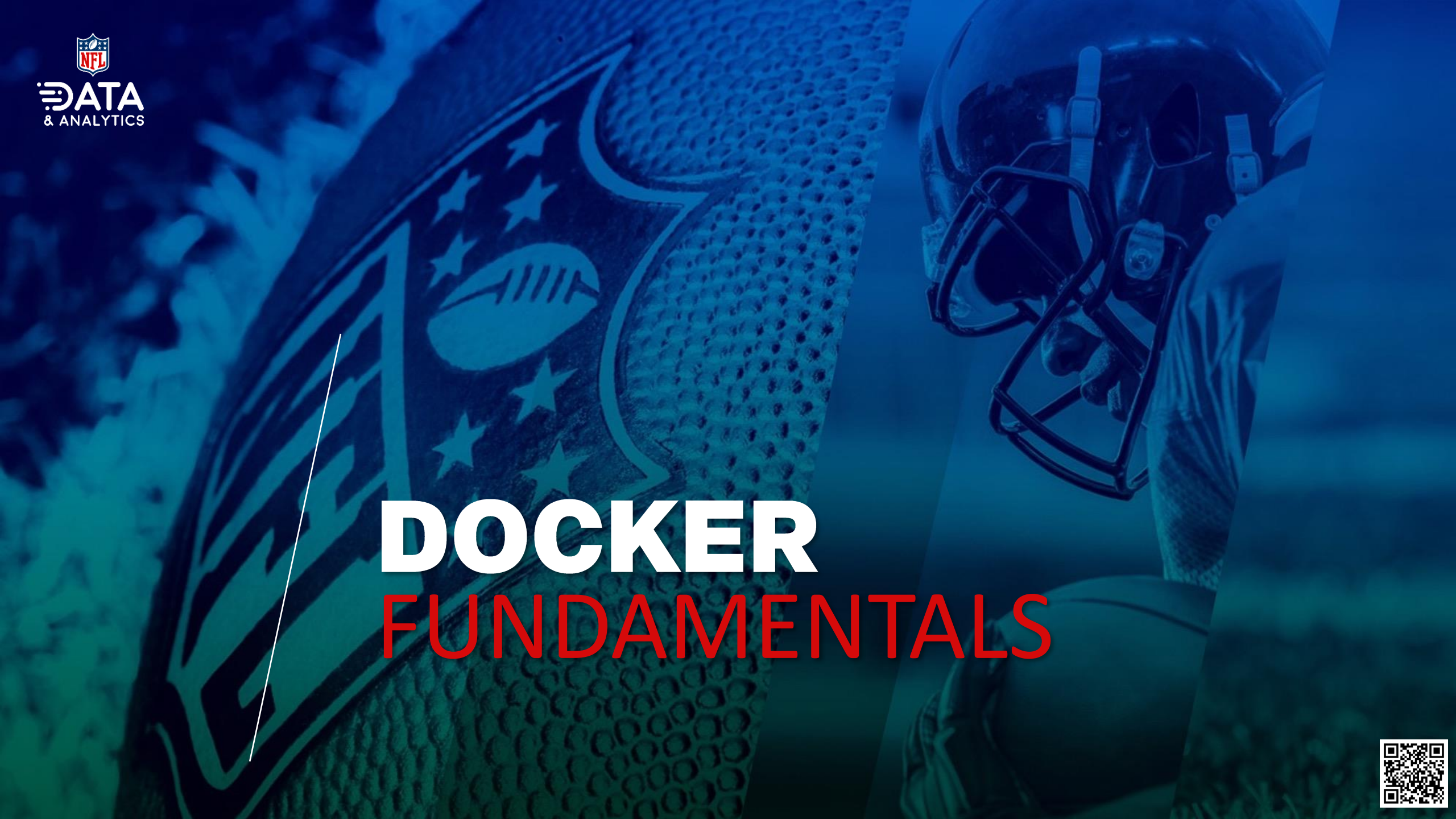
Inconsistent environments and incompatible packages stalls reproducibility.

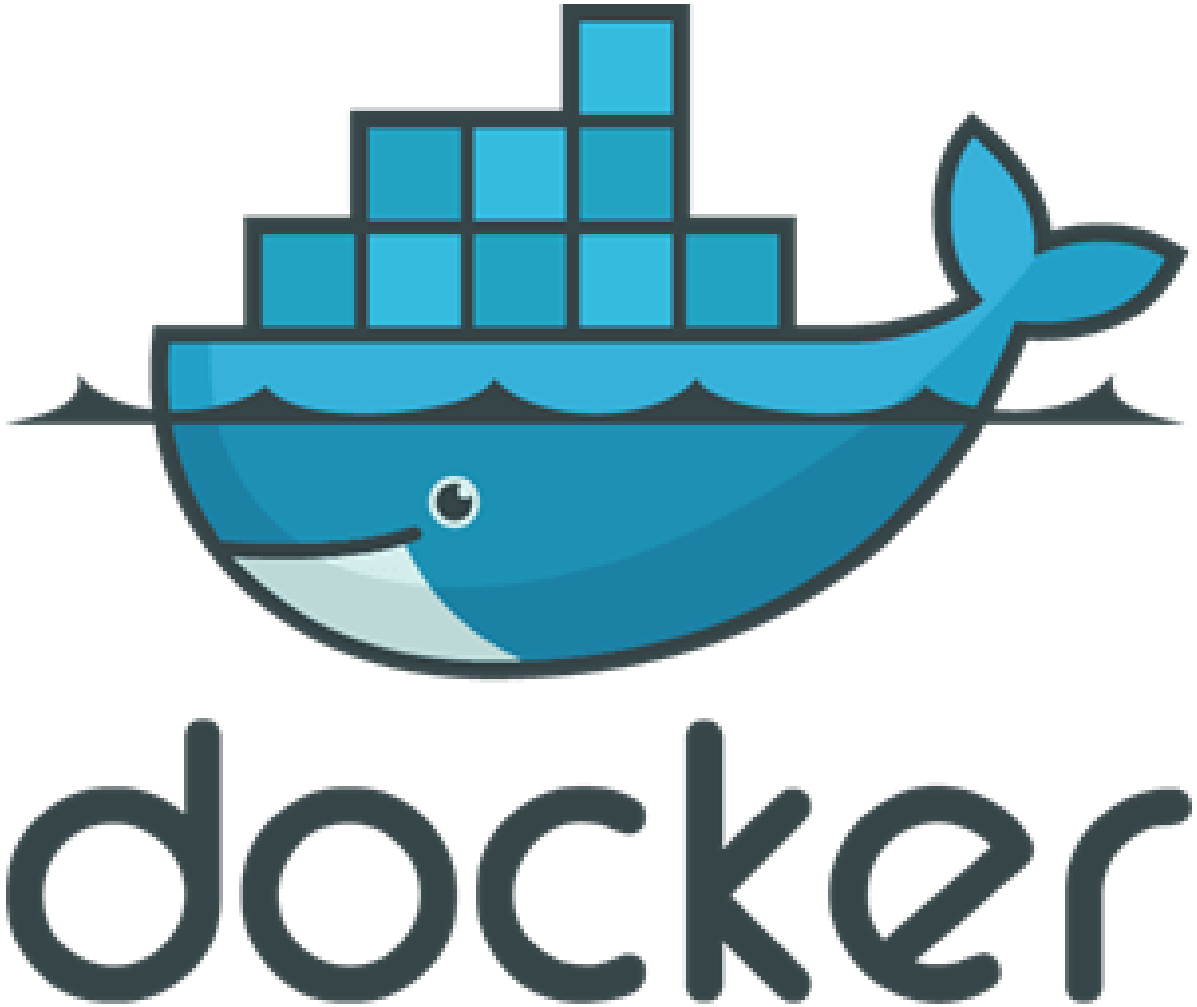
Scaling Issues

Training locally is easy but iterating and scaling to the cloud is a different game.

Manual Overhead

Manual steps is laborious and allows for multiple errors in the process.


NFL
DATA
& ANALYTICS
DOCKER
FUNDAMENTALS



Is a software platform that empowers everyone to build, share and run applications on any platform using containers.

Docker Fundamentals

Containerization

Is a one of the many methods of computer virtualization.

It allows for applications to run in isolated environments.

Think of it as a computer cloning itself.

```
FROM rocker/r-ver:4.4.0

WORKDIR /app

COPY model.R /app/

RUN R -e 'pak::local_install("/app", dependencies = FALSE)'

ENTRYPOINT ["Rscript"]

CMD ["e", "source(/app/model.R)"]

```

Dockerfile

Is a script that defines the steps to create a *blueprint* of your container.

Docker Image

Is a blueprint on how to build and run your code.

Essential Dockerfile Commands

- **FROM** is an instruction that specifies the base image that lays the groundwork. Used in multi-stage builds.
- **RUN** executes specific commands within the container during the build process.
- **COPY** transfers files from the host to the container file system.
- **CMD** defines the default actions within the container when it runs.
- **ENTRYPOINT** permits the container to run an executable.

Containerize the Model Pipeline

Current Script (Python)

1. Reads tracking data from a data source.
2. Trains the model.
3. Predicts the results on a series of variables.
4. Send the results to another source.

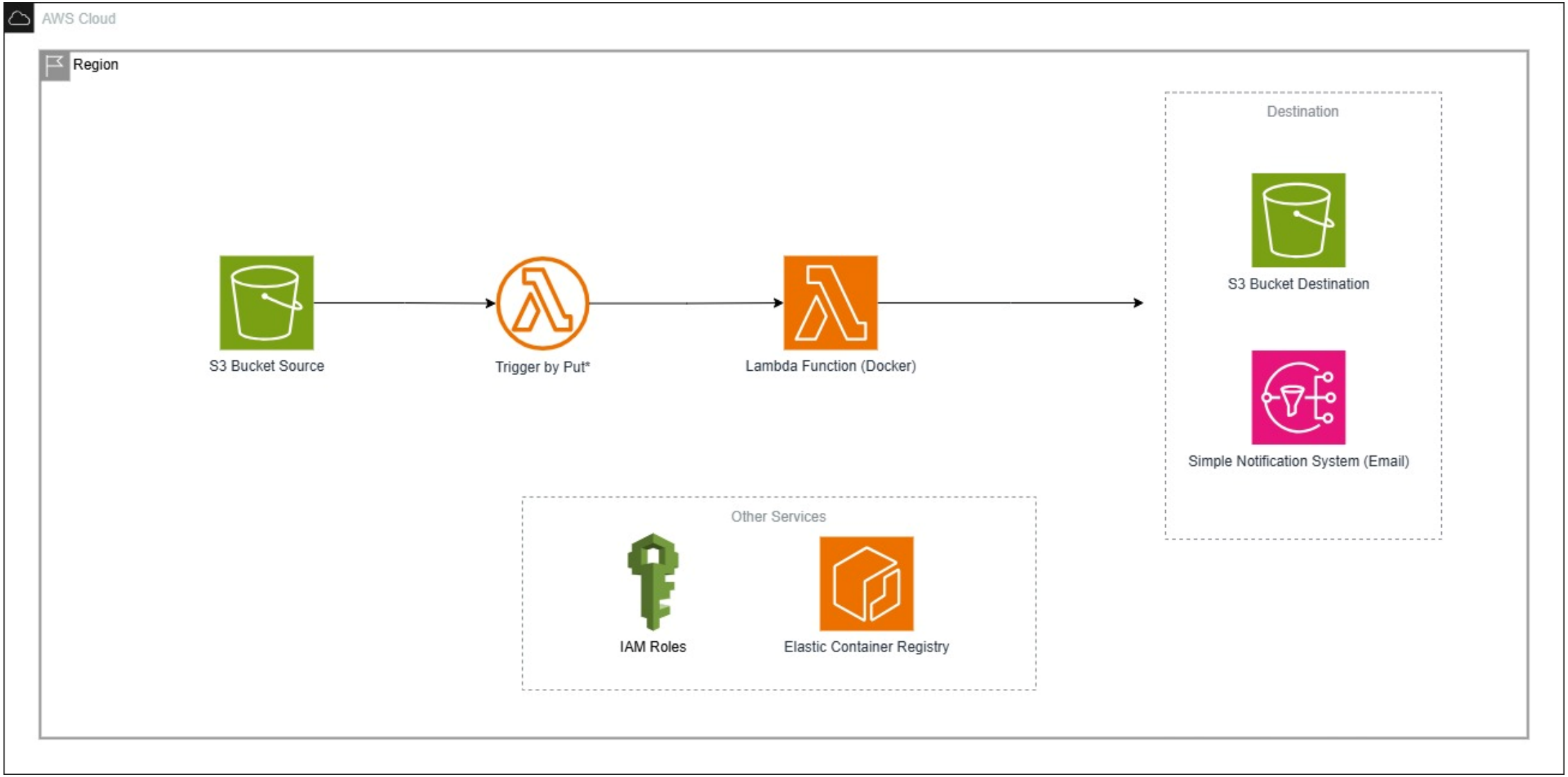
Containerize the Model Pipeline

1. Create a file called **`Dockerfile`** at the root of the model.
2. Use AWS Lambda Python as a base image.
3. Copy the content of the script into the directory including the dependency file.
4. Install script dependencies.
5. Run the script.

```
FROM public.ecr.aws/lambda/python:3.9

WORKDIR ${LAMBDA_TASK_ROOT}

COPY model.py ${LAMBDA_TASK_ROOT}

COPY requirements.txt ${LAMBDA_TASK_ROOT}

RUN pip install --no-cache-dir -r requirements.txt

RUN chmod +x model.py

CMD ["model.lambda_handler"]

```


NFL
DATA
& ANALYTICS
CLOUD AND
TERRAFORM
FUNDAMENTALS

Cloud and Terraform Fundamentals



HashiCorp

Terraform

Is an open source infrastructure as code tool used to programmatically provision and manage both on premise and cloud resources.

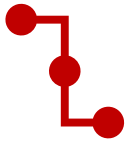
Cloud and Terraform Fundamentals

Declarative Language

Allows for you to describe infrastructure in *.tf files and manages state.

Platform Agnostic

Works with most cloud providers and on-premise servers.



Version Control

Allows for versioning, review and rollback infrastructure.

Each resource starts with the *name of the resource* and then fill in the attribute of the resource.

```
"resource_type" "custom_name_of_resource"  {

   attribute_name = ""
   attribute_name = ""

}
```

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type Info

- General purpose
  Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Bucket name Info

myawsbucket

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or nu

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Obje

- ACLs disabled (recommended)
  All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

Object Ownership

Bucket owner enforced

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use v
recover from both unintended user actions and application failures. Learn more

Bucket Versioning

- Disable
- Enable

Tags - *optional* (0)

You can use bucket tags to track storage costs and organize buckets. Learn more

No tags associated with this bucket.

Add tag

Default encryption Info

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info

- Server-side encryption with Amazon S3 managed keys (SSE-S3)
- Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
  Secure your objects with two separate layers of encryption. For details on pricing, see **DSSE-KMS pricing** o

Bucket Key

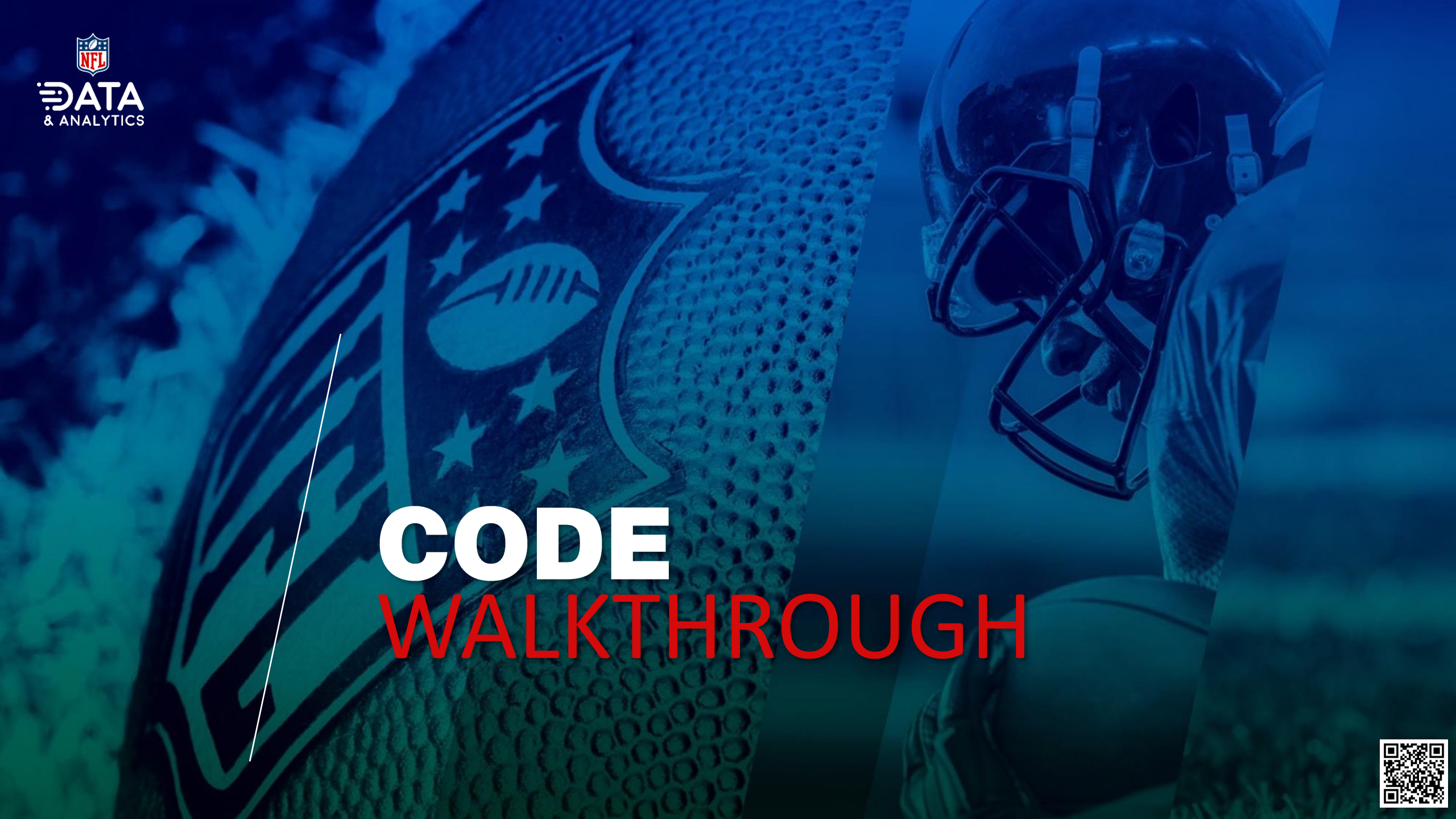
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys ar

- Disable
- Enable


```
resource "aws_s3_bucket" "my_test_s3_bucket" {
    bucket = "random_number_bucket_name"
    versioning = ""
    apply_server_side_encryption_by_default =""

    tags = {
      Name        = ""
      Environment = ""
    }
}
```

Cloud and Terraform Fundamentals

- Configure you AWS account role/user to be able to create the following resources:
  - IAM
  - Elastic Container Registry
  - Lambda
  - S3
  - Simple Notification System
  - Cloudwatch


NFL
DATA
& ANALYTICS
CODE
WALKTHROUGH

QUESTIONS

Zuri Hunter
Full Stack Engineer
Data and Analytics