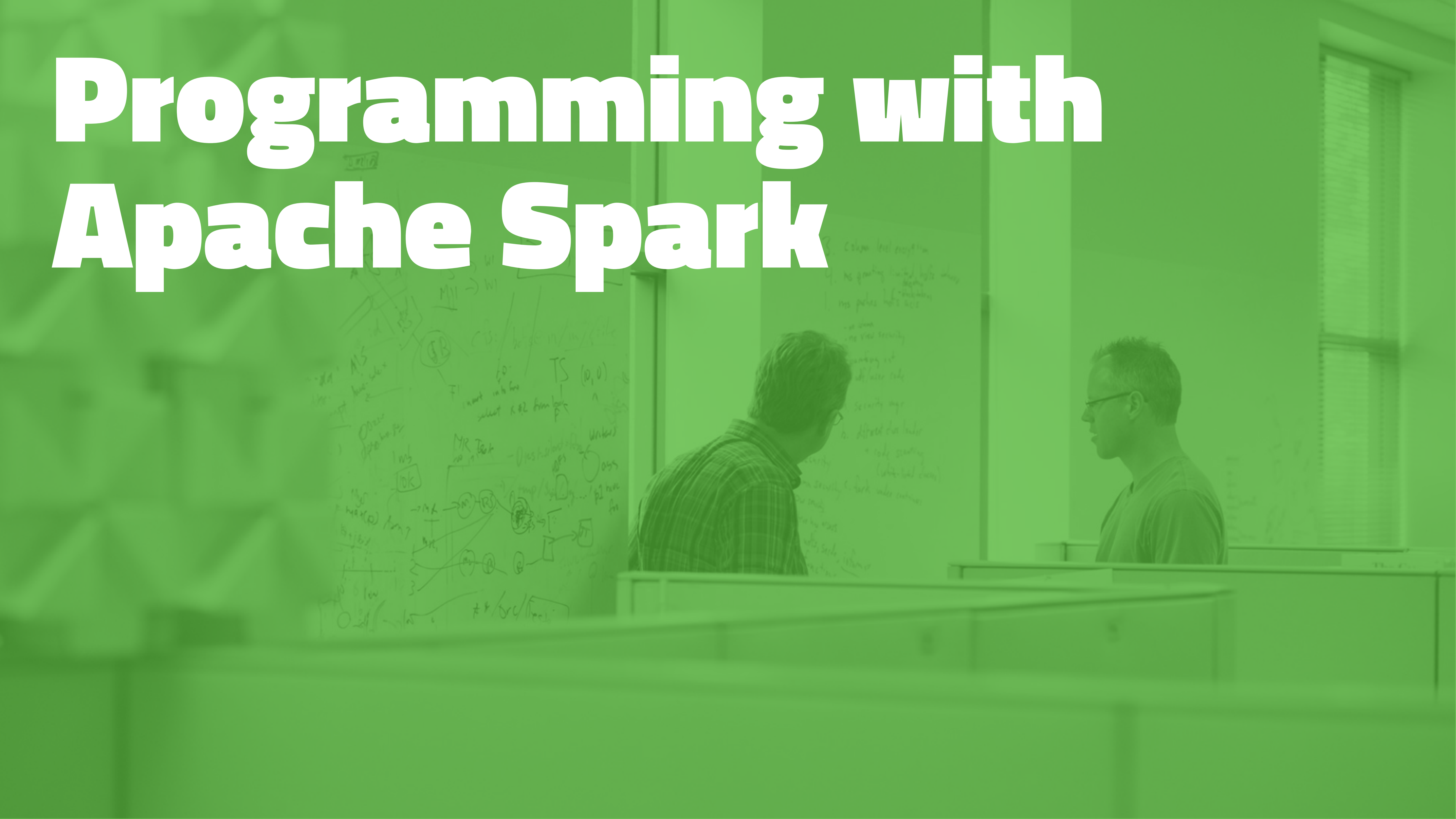
Programming with Apache Spark

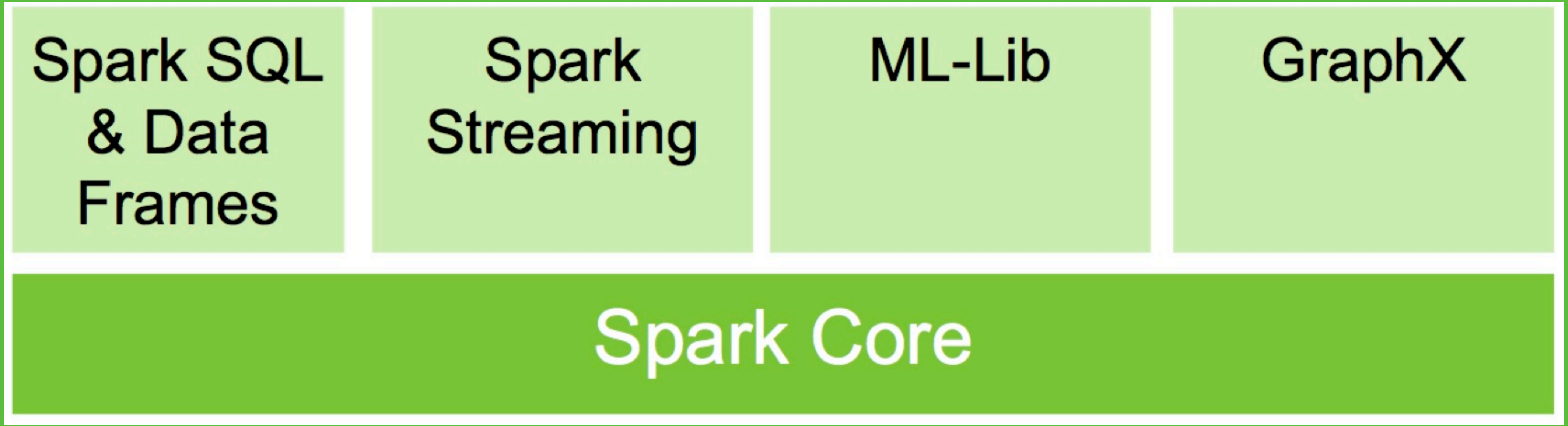


Learning Objectives

After you complete this lesson you should be able to:

- Start the Spark shell
- Understand what an RDD is
- Load data from HDFS and perform a word count
- Know the differences between Transformation and Action
- Explain Lazy Evaluation

The Spark Ecosystem



Spark SQL & Data Frames

Spark Streaming

ML-Lib

GraphX

Spark Core

The Resilient Distributed Dataset

An *Immutable* collection of objects (or records) that can be operated in parallel

- **Resilient**: can be created from parent RDDs - An RDD keeps its lineage information
- **Distributed**: partitions of data are distributed across nodes in the cluster
- **Dataset**: a set of data that can be accessed

Each RDD is composed of 1 or more partitions - The user can control the number of partitions - More partitions => More parallelism

What Does "Lazy Execution" Mean?

```
file = sc.textFile("hdfs://some-text-file")
counts = file.flatMap(lambda line: line.split(" ")) \
    .map(lambda word: (word,1)) \
    .reduceByKey(lambda a,b: a+b)
```

DAG of transformations is built by Spark on driver side

```
counts.saveAsTextFile("hdfs://wordcount-out")
```

Action triggers execution of whole DAG

Transformation: filter()

Keep some elements based on a predicate

```
rdd = sc.parallelize([1, 2, 3, 4, 5])

rdd.filter(lambda x: x%2 == 0).collect()
[2, 4]

rdd.filter(lambda x: x<3).collect()
[1, 2]
```

Creating a DataFrame: from a table in Hive

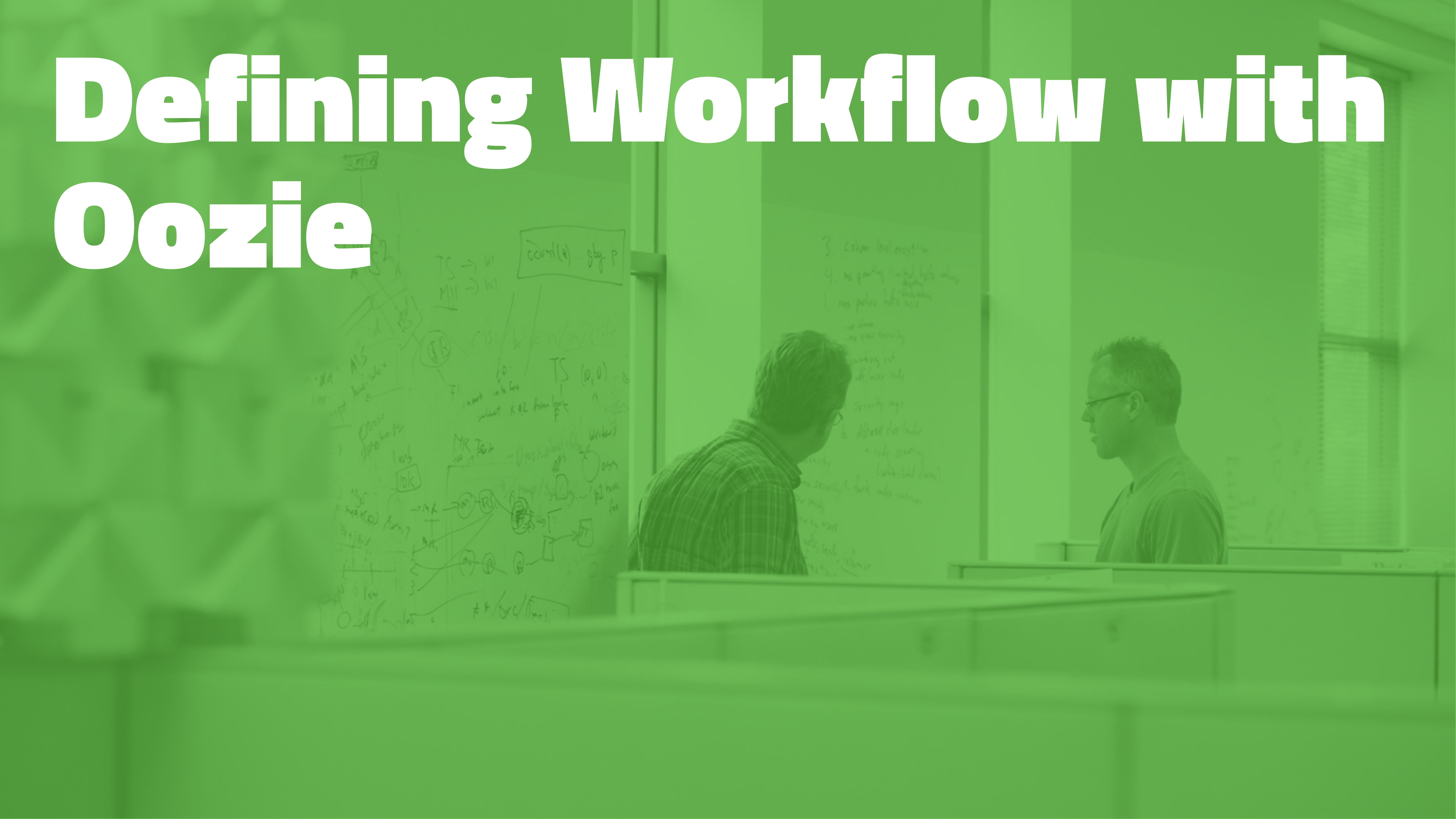
Load the entire table

```
df = hc.table("patients")
```

Load using a SQL query

```
df1 = hc.sql("SELECT * FROM patients WHERE age>50")
df2 = hc.sql("""
    SELECT col1 AS timestamp, SUBSTR(date,1,4) AS year, event
      FROM events WHERE year>2014""")
```

Defining Workflow with Oozie

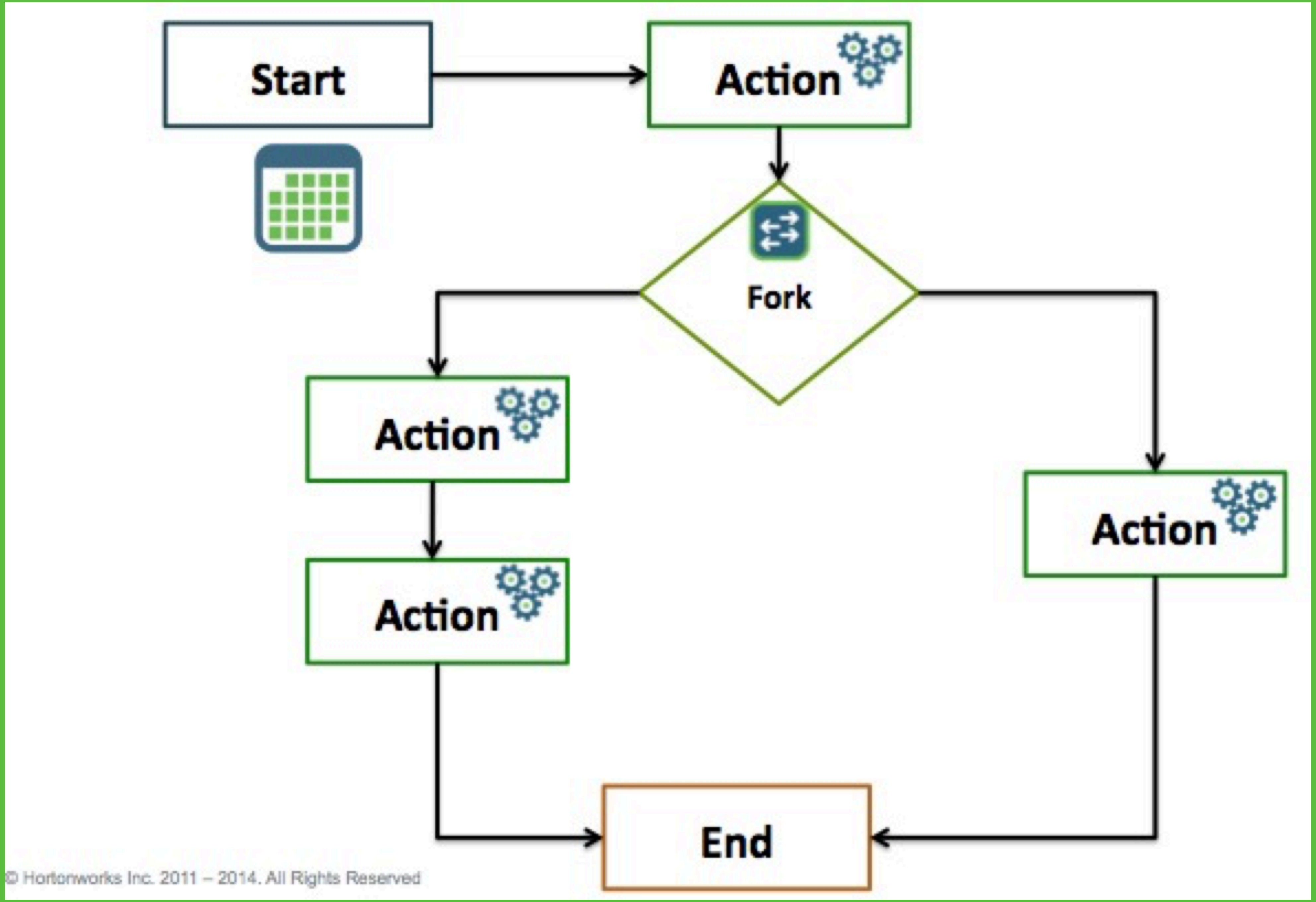


Overview of Oozie

Oozie has two main capabilities:

- **Oozie Workflow**: a collection of actions
- **Oozie Coordinator**: a recurring workflow

Defining an Oozie Workflow



Pig Actions

```
<workflow-app xmlns="uri:oozie:workflow:0.2"
          name="whitehouse-workflow">
    <start to="transform_whitehouse_visitors"/>
    <action name="transform_whitehouse_visitors">
        <pig>
            <job-tracker>${resourceManager}</job-tracker>
            <name-node>${nameNode}</name-node>
            <prepare>
                <delete path="wh_visits"/>
            </prepare>
           <script>whitehouse.pig</script>
        </pig>
        <ok to="end"/>
        <error to="fail"/>
    </action>
    <kill name="fail">
        <message>Job failed, error
           message[${wf:errorMessage(wf:lastErrorNode())}]
        </message>
    </kill>
    <end name="end"/>
</workflow-app>
```