Here is the script for workflow orchestration and automation using Apache Airflow:

```
from datetime import datetime, timedelta
from airflow import DAG
from airflow.operators.python_operator import PythonOperator
from airflow.operators.bash_operator import BashOperator

default_args = {
    'owner': 'airflow',
    'depends_on_past': False,
    'tart_date': datetime(2024, 6, 15),
    'email_on_failure': False,
    'email_on_retry': False,
    'etries': 1,
    'etry_delay': timedelta(minutes=5),
}

dag = DAG(
    'telco_data_pipeline',
    default_args=default_args,
    schedule_interval=timedelta(hours=1)
)

def ingest_data(**context):
    # Ingest data from multiple sources using AWS Glue
    glue = boto3.client('glue')
    jobs = []
```

```
    for source in ['network_traffic', 'transactions', 'product_sales', 'product_catalog', 'arketing_campaign']:
        job = glue.create_job(
            Name=source,
            Role='arn:aws:iam::123456789012:role/glue-execution-role',
            Type='Spark',
            GlueVersion='2.0',
            NumberOfWorkers=2,
            WorkerType='Standard'
        )
        jobs.append(job)
    return jobs


def transform_data(**context):
    # Transform and integrate data using Apache Spark
    spark = SparkSession.builder.appName("TelcoCorp Data Pipeline").getOrCreate()
    network_traffic_df = spark.read.format("parquet").load("s3://telcocorp-data/ingested_data/network_traffic")
    transactions_df = spark.read.format("parquet").load("s3://telcocorp-data/ingested_data/transactions")
    product_sales_df = spark.read.format("parquet").load("s3://telcocorp-data/ingested_data/product_sales")
    product_catalog_df = spark.read.format("csv").load("s3://telcocorp-data/ingested_data/product_catalog")
    marketing_campaign_df = spark.read.format("json").load("s3://telcocorp-data/ingested_data/marketing_campaign")
    transformed_df = network_traffic_df \
        .join(transactions_df, "date_id" == "date_id") \
```

```
    .join(product_sales_df, "product_id" == "product_id") \
    .join(product_catalog_df, "product_id" == "product_id") \
    .join(marketing_campaign_df, "campaign_id" == "campaign_id") \
    .select("date", "product_name", "customer_name", "traffic_volume",
"transaction_amount", "sales_amount")
  transformed_df.write.format("parquet").save("s3://telcocorp-data/transformed_data")

def load_data(**context):
  # Load transformed data into a data warehouse or data lake
  transformed_df = spark.read.format("parquet").load("s3://telcocorp-
data/transformed_data")
  transformed_df.write.format("parquet").save("s3://telcocorp-data/data_warehouse")

ingest_task = PythonOperator(
  task_id='ingest_data',
  python_callable=ingest_data,
  dag=dag
)

transform_task = PythonOperator(
  task_id='transform_data',
  python_callable=transform_data,
  dag=dag
)

load_task = PythonOperator(
```

```
    task_id='load_data',
    python_callable=load_data,
    dag=dag
)

end_task = BashOperator(
    task_id='end_task',
    bash_command='echo "Data pipeline completed successfully"',
    dag=dag
)

ingest_task >> transform_task >> load_task >> end_task
```