# Gettesty – Automated Test Data Generation

## Overview

Overview  
gettesty is a purpose-built software tool designed to streamline and scale the generation of test datasets for test environments. Developed to simulate real-world production data patterns, gettesty can generate high-volume data that can use for ingestion app — all in a fully automated fashion.

## Key Features & Capabilities

* Key Features & Capabilities  
    
  1. High-Volume Test Dataset Generation for SOR Tables  
  the main function of gettesty is its ability to generate synthetic data for five interrelated SOR tables based on the production model schema. These tables can be seamlessly joined using a set of shared keys: ECI, ARN, Hashed\_account\_number, and acct\_nb. This ensures that relational integrity is preserved across the generated datasets, closely mimicking live production scenarios.  
    
  To validate its performance and scalability, I conducted a stress test generating 240 million records. The dataset was then fed into ingestion app. The results were impressive: the entire pipeline—from generation to ingestion—ran smoothly and without error, demonstrating both the robustness and the practical utility of gettesty.  
    
  2. Full Automation with AWS EMR & Control-M  
  We bring gettesty automation to the next level by integrating with Control-M, which triggers a daily job to spin up an AWS EMR cluster. Within this cluster, a series of EMR steps are executed to generate fresh test data for each SOR dataset. The output is saved directly to dedicated Amazon S3 buckets, organized for easy access by ingestion applications.  
    
  This automated pipeline ensures:  
  - Daily refresh of synthetic data that mirrors production structure  
  - Minimal manual intervention  
  - Seamless integration into existing big data workflows  
    
  With gettesty, teams no longer need to manually prepare test data or struggle with unrealistic mock datasets. Instead, they can rely on a stable, scalable, and fully automated pipeline that reflects production-level complexity — every single day.

## Process Diagram

