**DS644:INTRODUCTION TO BIG DATA**

**SECTION: 852**

REPORT

On

MILESTONE 2

# **LOAN - CREDIT RISK & POPULATION STABILITY DATA ANALYSIS**

By

|  |  |  |
| --- | --- | --- |
| **NAME** | **UCID** | **EMAIL** |
| Kenil Pravinbhai Avaiya | Ka683 | ka683@njit.edu |
| Neel Chimanbhai Barvaliya | nb585 | nb585@njit.edu |

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **NO** | **CONTENTS** | **PAGE NO.** |
| 1 | Introduction | 1 |
| 2 | Software Specifications | 1 |
| 3 | Dataset Specification | 3 |
| 4 | Hadoop Cluster Setup | 4 |

**INTRODUCTION**

Understanding credit risk is a crucial aspect of financial decision-making, particularly for lending institutions. The advent of peer-to-peer lending platforms, such as LendingClub, has necessitated the management and analysis of loan data with increasing significance. This project endeavors to analyze loan data from LendingClub, which encompasses approximately 1.8 million consumer loans issued between 2014 and 2018. The objective is to discern discernible patterns and trends in loan approvals, defaults, and borrower behavior.

Leveraging the capabilities of big data technologies, analyzing this dataset can reveal significant insights that are otherwise concealed amidst its vast volume. Through the utilization of tools such as Hadoop and Oozie, this project endeavors to systematically process and analyze the dataset in order to address pertinent inquiries, including:

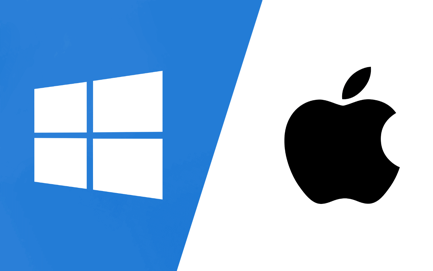
1. What factors increase the risk of loan default?
2. How do borrower characteristics (such as income or loan purpose) affect loan outcomes?
3. Are there noticeable trends in credit risk over the years?

Through this analysis, our objective is to enhance our comprehension of credit risk and streamline the loan decision-making process. Furthermore, this project demonstrates the capabilities of big data tools in managing vast datasets and facilitating the discovery of valuable insights that can support business and policy decisions.

**SOFTWARE SPECIFICATIONS**

In this section, we will dive into the software behind this project.

1. MacOS & Windows - This is the primary operating system utilized for the development and management of the project locally.



1. Amazon EC2 - Amazon Web Services (AWS) offers a cloud service that previously hosted and operated large data tools such as Hadoop and Oozie. This service provides the computing resources necessary to manage and process extensive datasets.



1. Java - Required for running Hadoop and Oozie, as both are built using Java. It also helps in writing some backend logic or scripts if needed.



1. Hadoop - A big data framework used to store and process large amounts of loan data across a distributed system. It helps in managing data that is too big for traditional systems.



1. MapReduce - Hadoop’s programming model for data processing. It breaks down big tasks into smaller ones and processes them at the same time, speeding up data analysis.



1. Oozie - A workflow scheduler system for managing Hadoop jobs. It helps automate and run data processing tasks in the correct order.



**DATASET**

**Dataset Name & Source**

In this project, we select Lone – credit risk and Population Stability.

Link: Kaggle Dataset: [Loan - Credit Risk & Population Stability](https://www.kaggle.com/datasets/beatafaron/loan-credit-risk-and-population-stability?select=loan_2019_20.csv)

**Dataset Description**

This selected dataset contains details of loan applications and information data from LendingClub across the number of years( 2019-2020 ), with a focus or primary attention on credit risk, the probability that someone who borrowed money, does not repay their debt, and population stability. In this dataset, every entry represents a unique loan issued to a borrower and includes various fields related to:

* **Loan Information:** loan amount, funded amount, term, interest rate, installment
* **Borrower Profile:** employment length, annual income, home ownership, purpose of the loan
* **Credit History:** credit score (FICO), delinquency records, number of credit lines, inquiries
* **Loan Performance:** loan status (fully paid, charged off, default), payment progress, recovery
* **Demographics:** state, zip code, issue date

The entire dataset presented is tabular in nature. Each row represents a distinct loan number, while each column serves as a feature or attribute.

**Size & Format:**

Format: This file is in CVS form.

size: 510MB

**Reason for Selection:**

This dataset is ideal for MapReduce processing due to its large volume of structured data, enabling parallel processing for tasks such as:

* Aggregation of loan performance across states or income groups
* Filtering and transformation of high-risk loan profiles
* Statistical analysis on credit score distribution or default trends
* Creating derived features for risk modeling using distributed computation

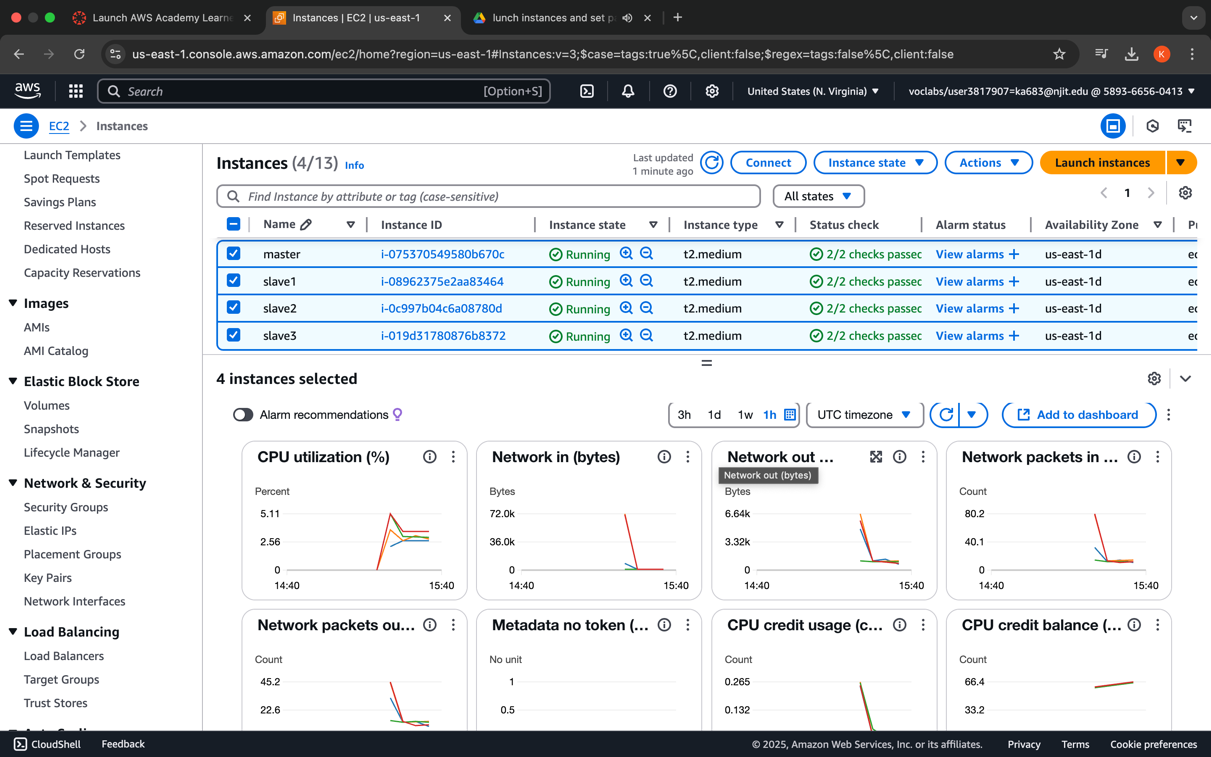
It provides real-world, finance-related data that is relevant, diverse, and complex enough to

showcase the benefits of big data tools like Hadoop and Spark.

**HADOOP CLUSTER SETUP**

1. **A screenshot of the Amazon instance** management web interface displaying the current state of each of your virtual machine (VM) instances.

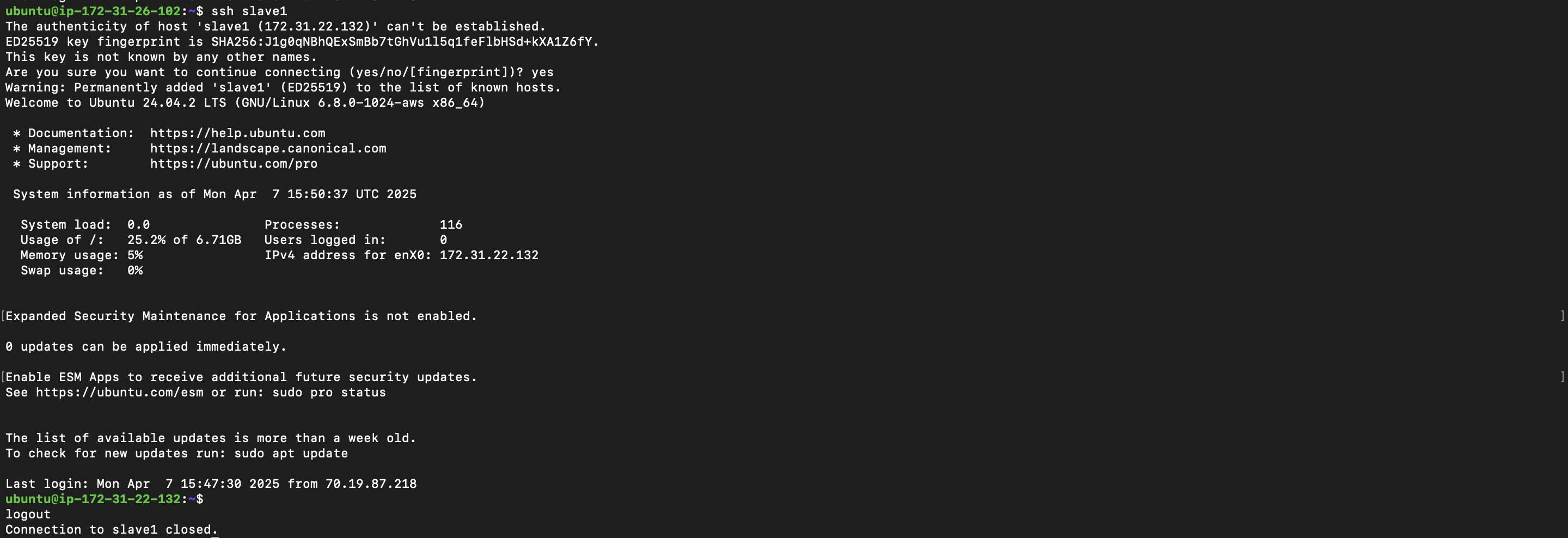
* All master and slave nodes are in the **“running”** state and have completed **“2/2” status checks**.

****

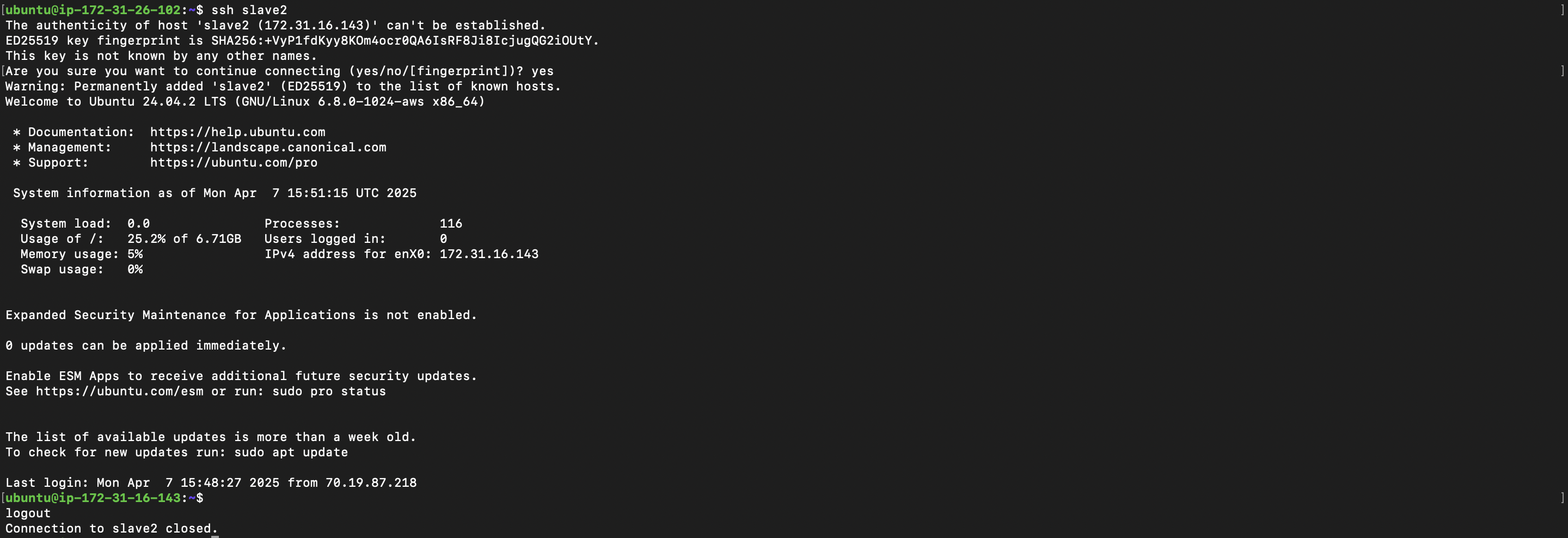
Screenshot of running instances.

1. Capture a screenshot **of a passphrase-free SSH login** from my **Master node instance to one of all Slave nodes.**

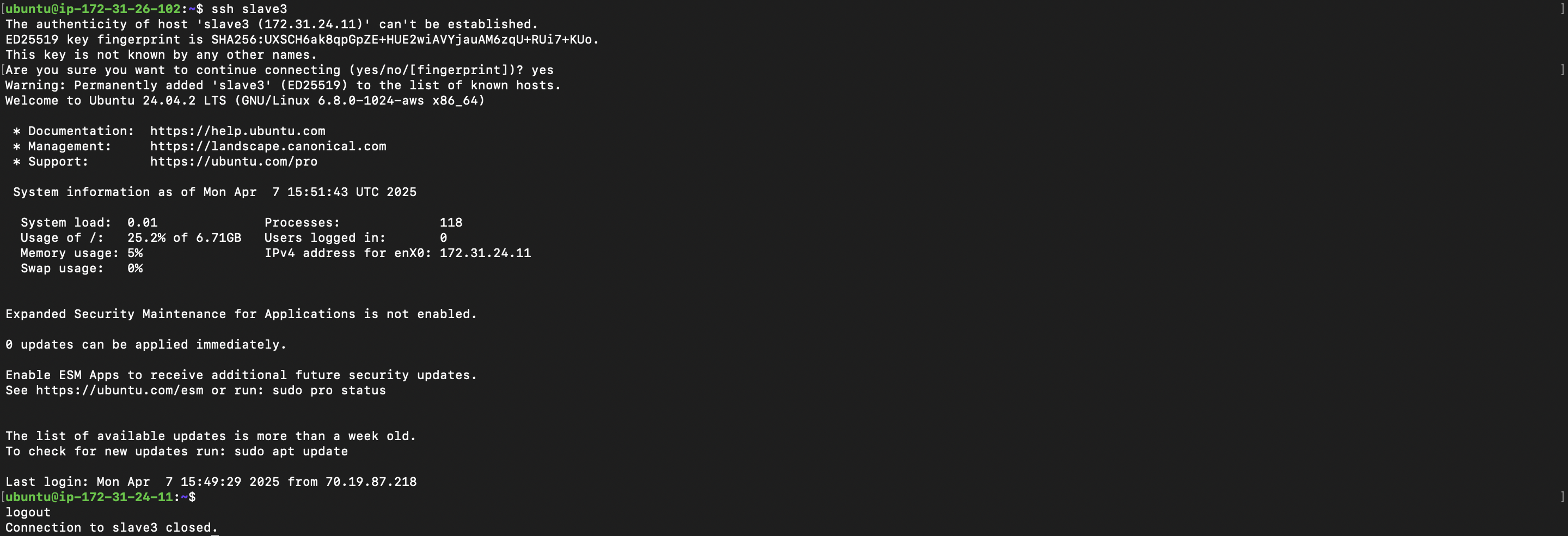
* The screenshot can show a successful login message after you give the SSH login command.

****

Passphrase-less SSH login - slave1.

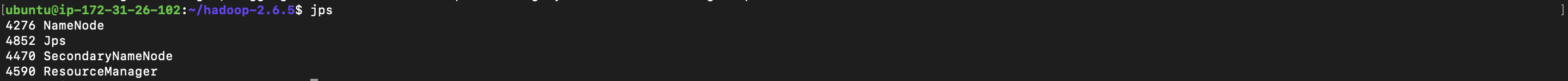
****

Passphrase-less SSH login – slave2

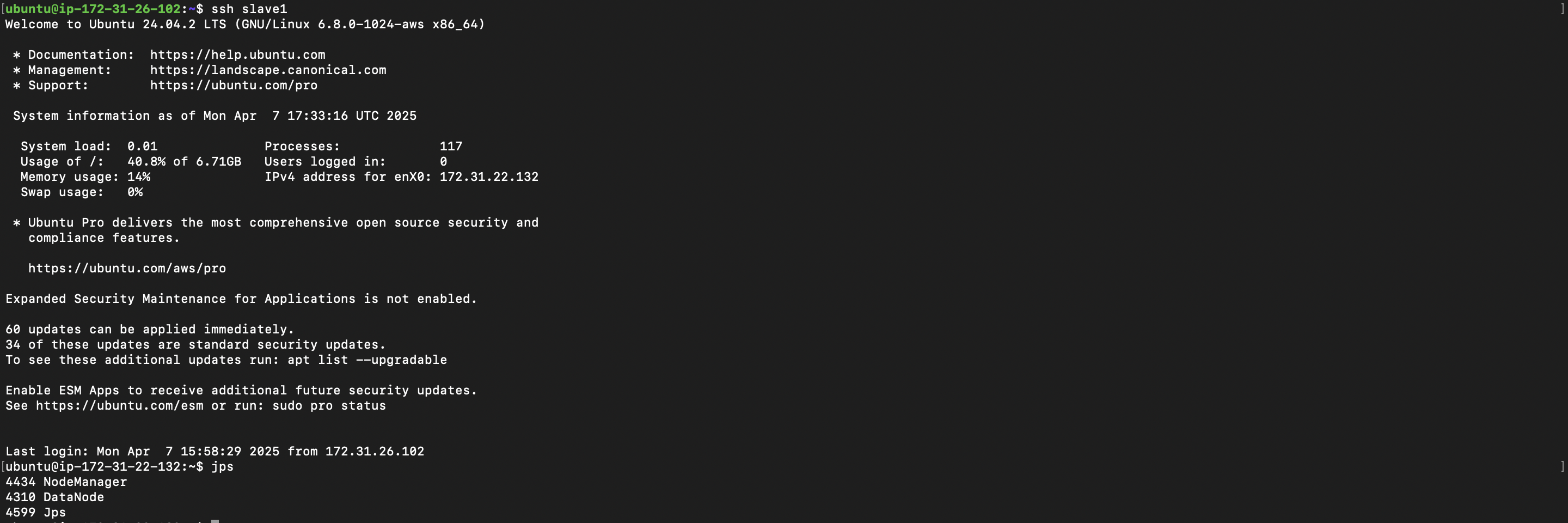
****

Passphrase less SSH login – slave3

1. Screenshots showing the output of the **"jps"** command on both the Master node and all Slave nodes after **formatting the namenode and starting Hadoop.**

****

The output of the "jps" command - master

****

The output of the "jps" command – slave1

****

The output of the "jps" command – slave2

****

The output of the "jps" command – slave3