Linux Infrastructure, Storage, and Automation Work

This body of work documents hands-on Linux systems administration performed across multiple environments, combining virtualization management, storage design, troubleshooting, and automation. The work spans VMware vSphere operations, CentOS Stream 9 system configuration, Logical Volume Manager (LVM) administration, and repeatable system reporting using Bash and Ansible.

The process begins at the virtualization layer, where Linux virtual machines are managed within a clustered VMware vSphere environment. Virtual hardware is reviewed and adjusted to align with workload requirements, including CPU, memory, networking, and multi-disk layouts. Systems are intentionally provisioned with separate disks to support clean storage segmentation and long-term operational stability.

Within the guest operating system, disk layouts are validated using standard Linux tools to confirm partitioning, logical volumes, and mount points. Application logs are isolated onto a dedicated filesystem backed by its own volume group, preventing uncontrolled log growth from impacting the root filesystem. Storage expansion is performed using LVM, including extending logical volumes and validating filesystem growth without requiring downtime.

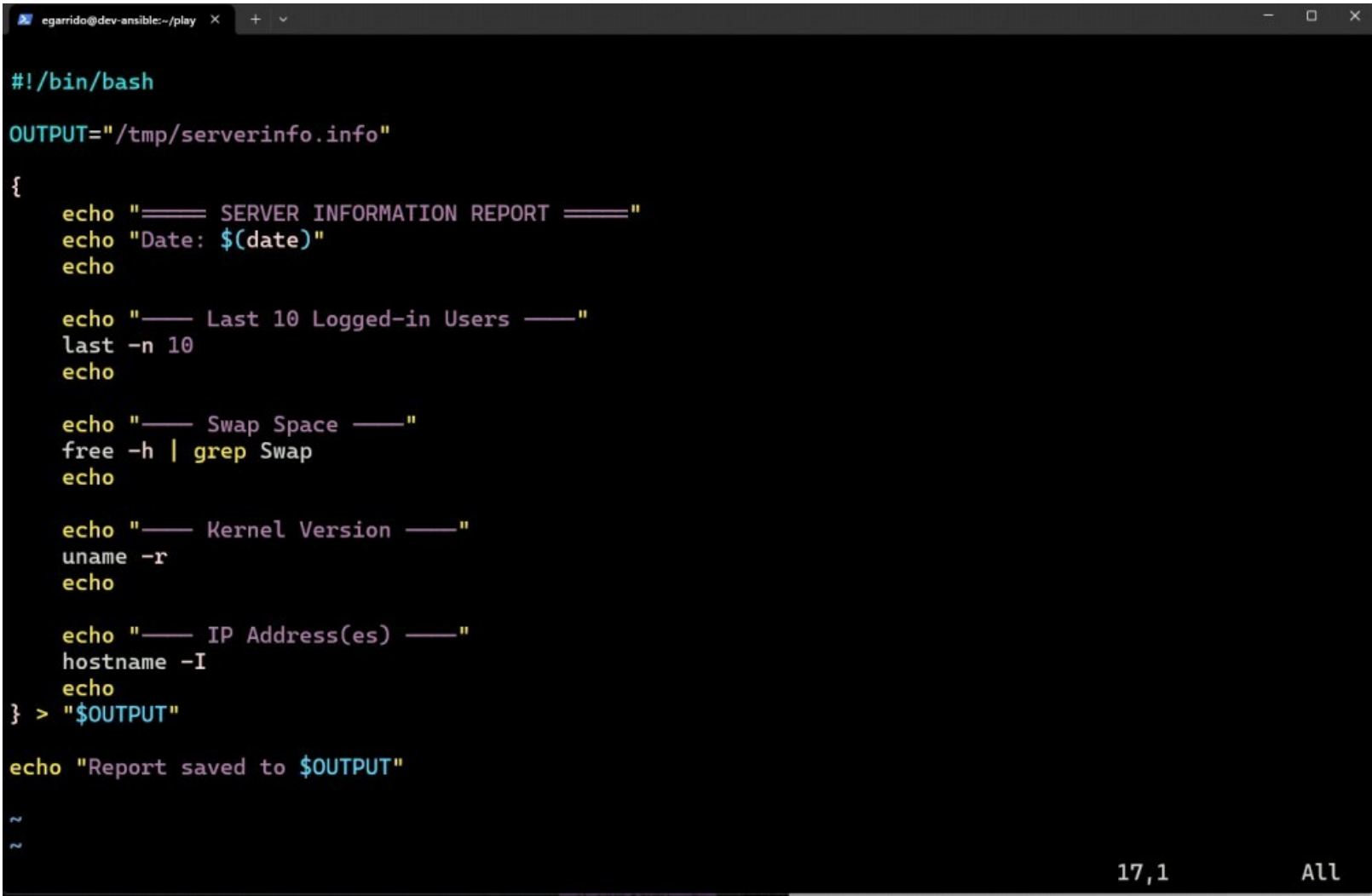
During the expansion process, permission and device-mapper issues are encountered and resolved through proper privilege escalation and corrective command usage. Changes are verified immediately through filesystem and block device inspection to confirm that the additional capacity is available and that no core system volumes are affected.

Automation is introduced to standardize system visibility across hosts. A Bash script is developed to collect key system details such as recent login activity, swap usage, kernel version, and IP addressing, writing results to a structured report file. Execution permissions and access controls are applied to ensure safe and consistent use.

To scale this process, Ansible is used to deploy and execute the reporting script across multiple servers. The automation gathers system facts, distributes the script, runs it remotely, and retrieves generated reports back to the control node. Validation confirms consistent output across development, performance, and staging environments, demonstrating reliable automation and centralized system insight.

Together, this work reflects practical experience managing Linux systems in an enterprise-style environment, designing resilient storage layouts, resolving real-world issues, and implementing automation to reduce manual effort while improving

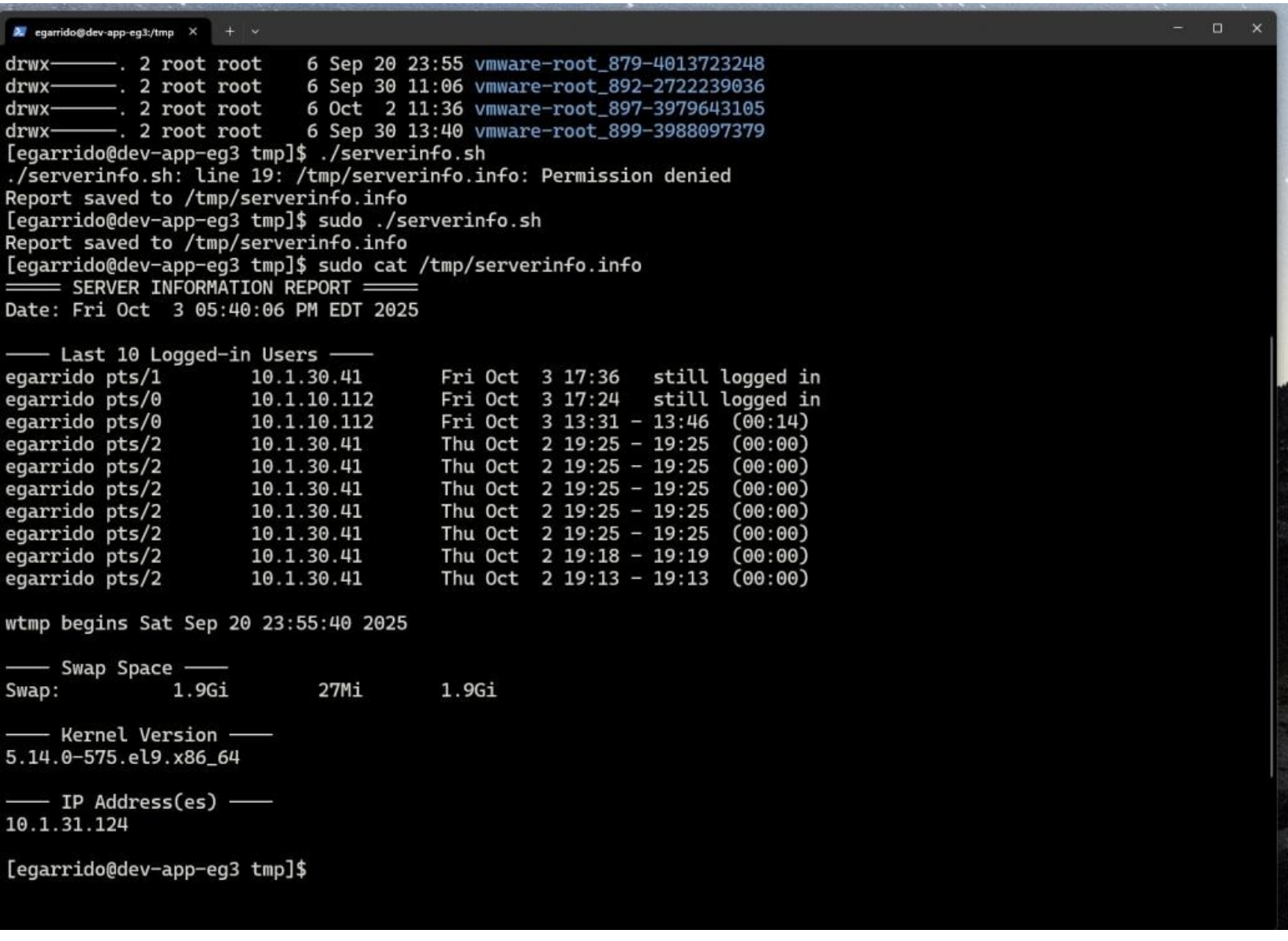
A Bash script is used to generate a consolidated server information report and write it to a file for easy review and auditing. The script captures the current date, recent user login activity, swap memory usage, kernel version, and assigned IP addresses, presenting the data in a structured and readable format. Output is redirected to a dedicated file in the /tmp directory, providing a lightweight way to collect key system details for troubleshooting, validation, or documentation without manual command execution.



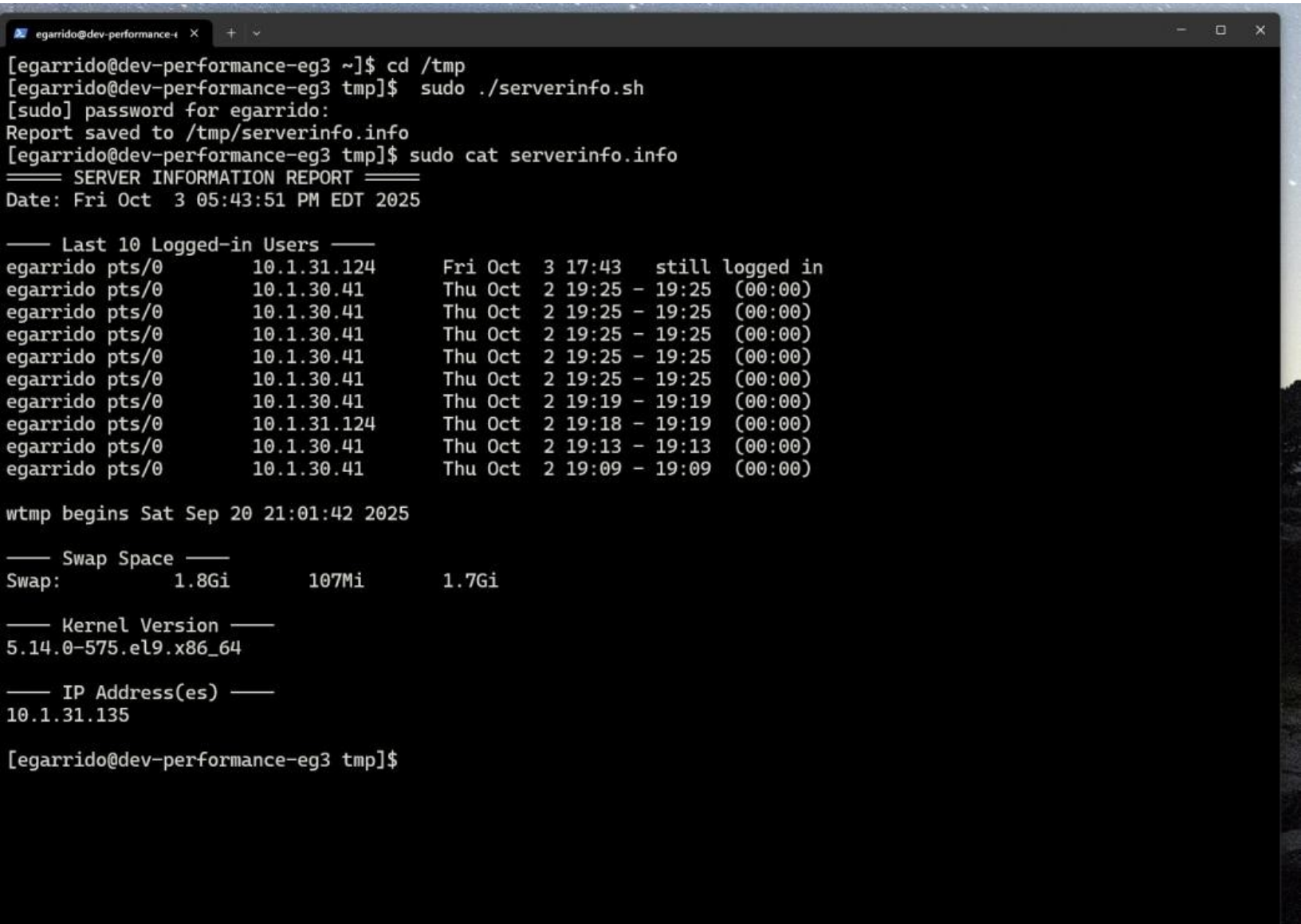
Execution permissions are applied to the server information script, allowing it to be run directly from the filesystem. File ownership and permissions are verified to confirm the script is executable by the owner and readable by others, ensuring controlled access while enabling operational use for system reporting and diagnostics.

An Ansible playbook is executed against multiple Linux hosts to deploy and run a server information script in a controlled and repeatable manner. The playbook gathers system facts, creates the script on each target host, executes it to generate a local report, and then retrieves the resulting files back to the Ansible control node for centralized review. Successful task completion across development, performance, and staging systems confirms consistent configuration, remote execution, and reliable automation across environments.

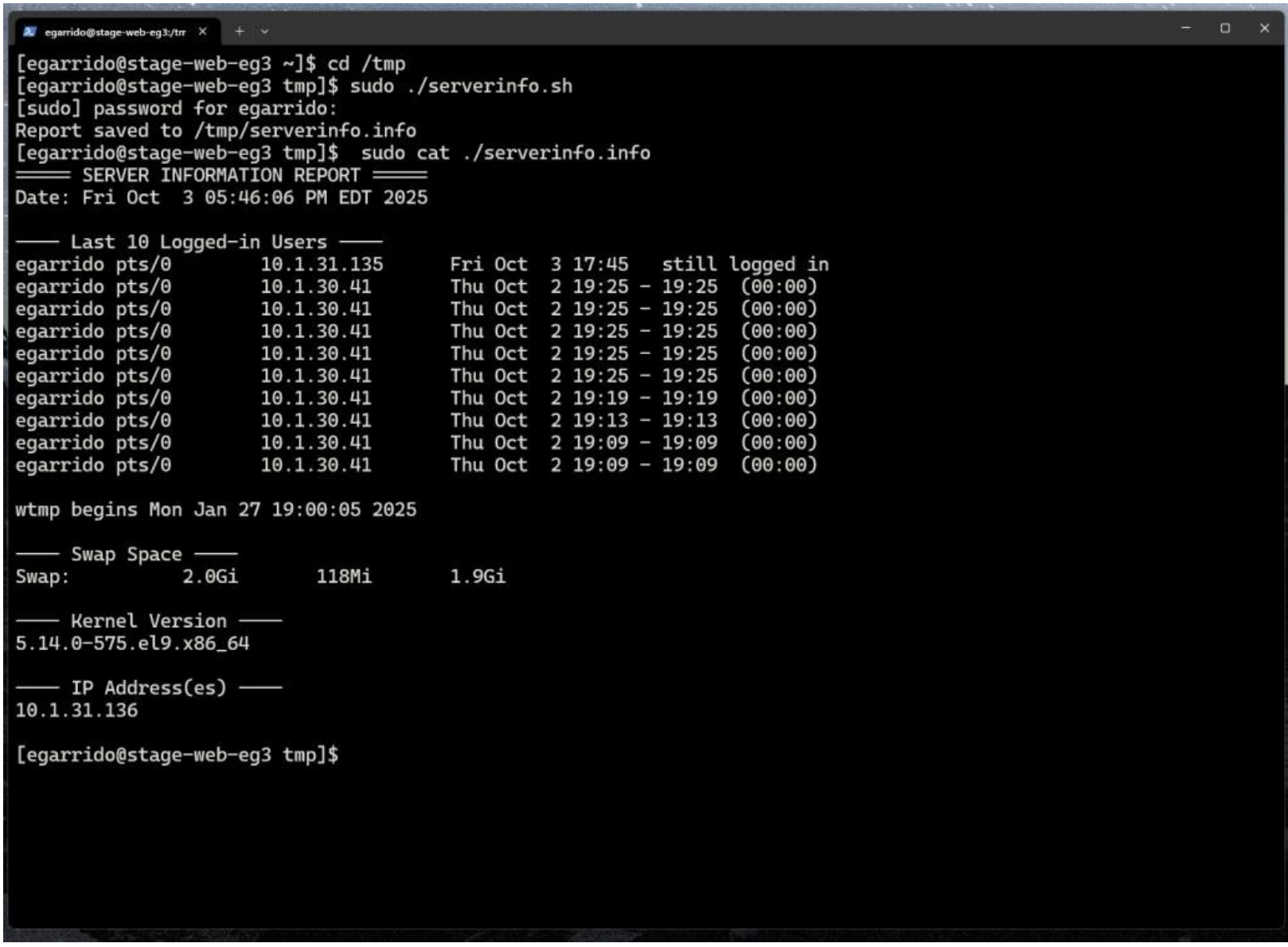
The server information script is executed locally to generate a consolidated system report. An initial permission issue occurs when writing the output file, which is resolved by running the script with elevated privileges. The resulting report successfully captures recent login activity, swap usage, kernel version, and assigned IP addresses, confirming correct script functionality and validating system state through automated data collection.



The server information script is executed on an additional host to confirm consistent behavior across environments. The report is generated successfully with elevated privileges and captures recent login activity, swap usage, kernel version, and assigned IP address for the system. Output verification confirms the script functions reliably on multiple servers, reinforcing its use as a repeatable tool for collecting baseline system information and validating host state across development and performance environments.



The server information script is run on a staging host to validate consistent execution across environments. The report is generated successfully with elevated privileges and captures recent login activity, swap usage, kernel version, and the assigned IP address for the system. Results confirm the script produces reliable and uniform output across development, performance, and staging servers, supporting its use as a repeatable method for collecting baseline system information.



This work demonstrates hands-on Linux systems administration across virtualized environments, combining VMware vSphere management, structured storage design, LVM expansion, troubleshooting, and automation. It includes validating disk layouts, isolating application logs, safely extending filesystems, and resolving permission-related issues without impacting core system volumes. Automation with Bash and Ansible is used to standardize system reporting and collect consistent host information across development, performance, and staging environments, reinforcing repeatable operations and reliable infrastructure management.