Linux Infrastructure Provisioning, Identity, and Automation Readiness

This project documents the end-to-end provisioning, configuration, and validation of a Linux virtual machine in an enterprise-style environment, from initial VM creation to full automation readiness.

The system is deployed on vSphere and configured with proper networking, hostname alignment, and operating system settings to meet infrastructure standards. Network interfaces, routing, DNS configuration, and system identity are verified to ensure reliable connectivity and compatibility with centralized services.

Centralized identity management is implemented by enrolling the host into a FreeIPA domain, enabling Kerberos authentication, SSSD-based identity resolution, and centralized sudo policy enforcement. Both local and domain user access are validated, including privilege escalation via the wheel group.

Secure access is established using SSH key-based authentication, leveraging modern cryptographic keys. SSH daemon configuration is reviewed, host key mismatches are resolved, and trusted access between the Ansible control node and managed hosts is confirmed. Static hostname resolution is applied where necessary to support controlled environments and troubleshooting scenarios.

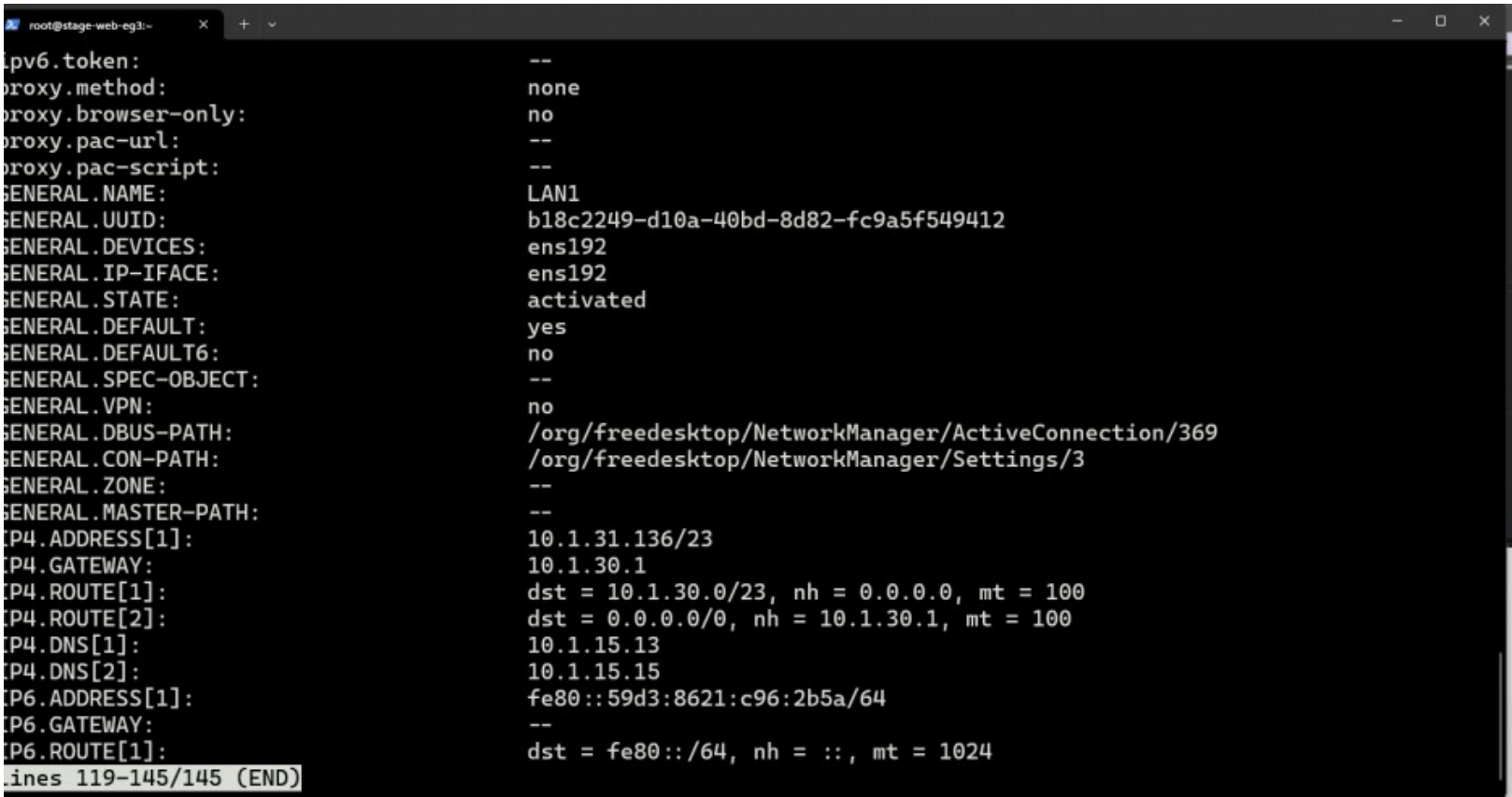
An Ansible inventory is built and refined to reflect multiple environments and roles. Initial connectivity issues are identified, corrected, and validated using the ansible -m ping module, confirming reliable, passwordless automation access across all target systems.

Finally, the virtual machine is documented in an asset management system, capturing ownership, system metadata, and lifecycle status to ensure traceability and operational visibility.

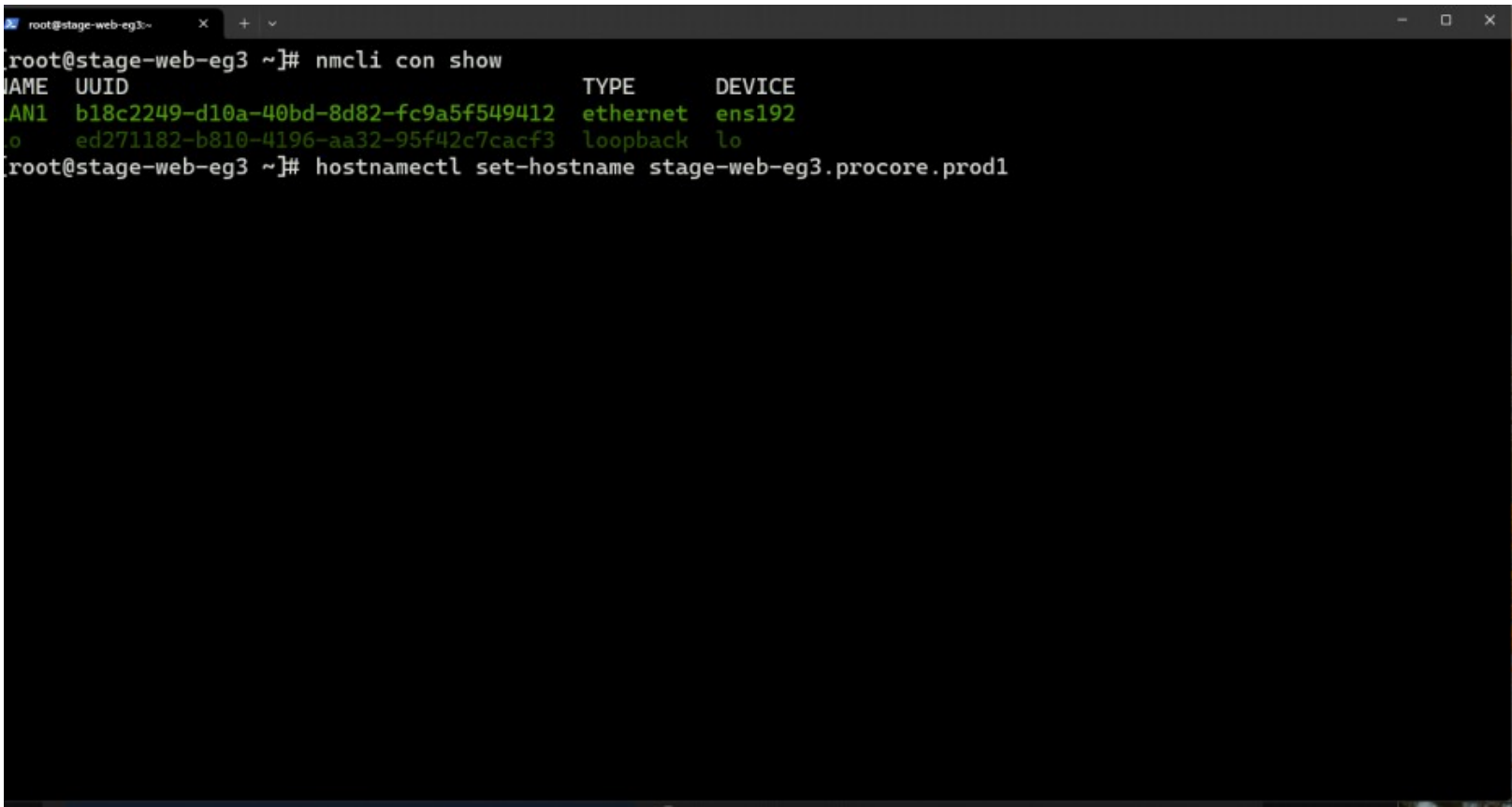
Security & Sanitization Notice

All IP addresses, hostnames, domain names, usernames, serial numbers, MAC addresses, timestamps, and environment-specific identifiers shown in screenshots, command output, and configuration files have been sanitized or obfuscated. No production credentials or sensitive infrastructure data are exposed in this repository.

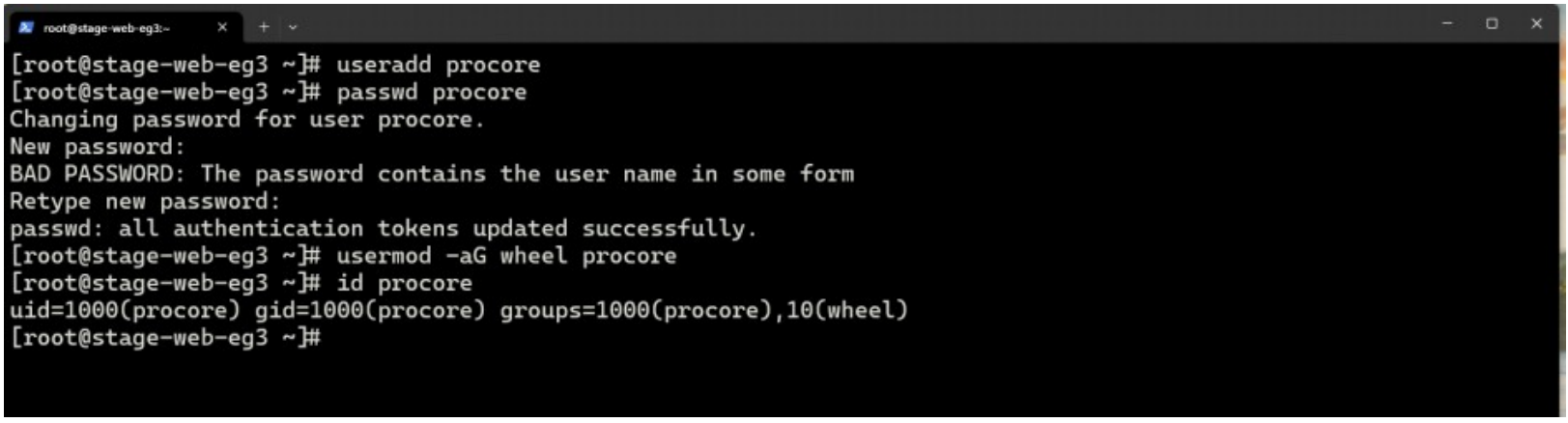
This screenshot shows detailed network interface configuration output for the LAN1 connection on stage-web-eg3.procore.prod1, likely retrieved using NetworkManager tools. It confirms the active interface, assigned IPv4 and IPv6 addresses, default gateway, routing table entries, and DNS servers. The configuration verifies proper network connectivity and routing required for domain enrollment, SSH access, and Ansible communication. All IP addresses, gateways, DNS servers, and environment-specific details shown are sanitized for security purposes.



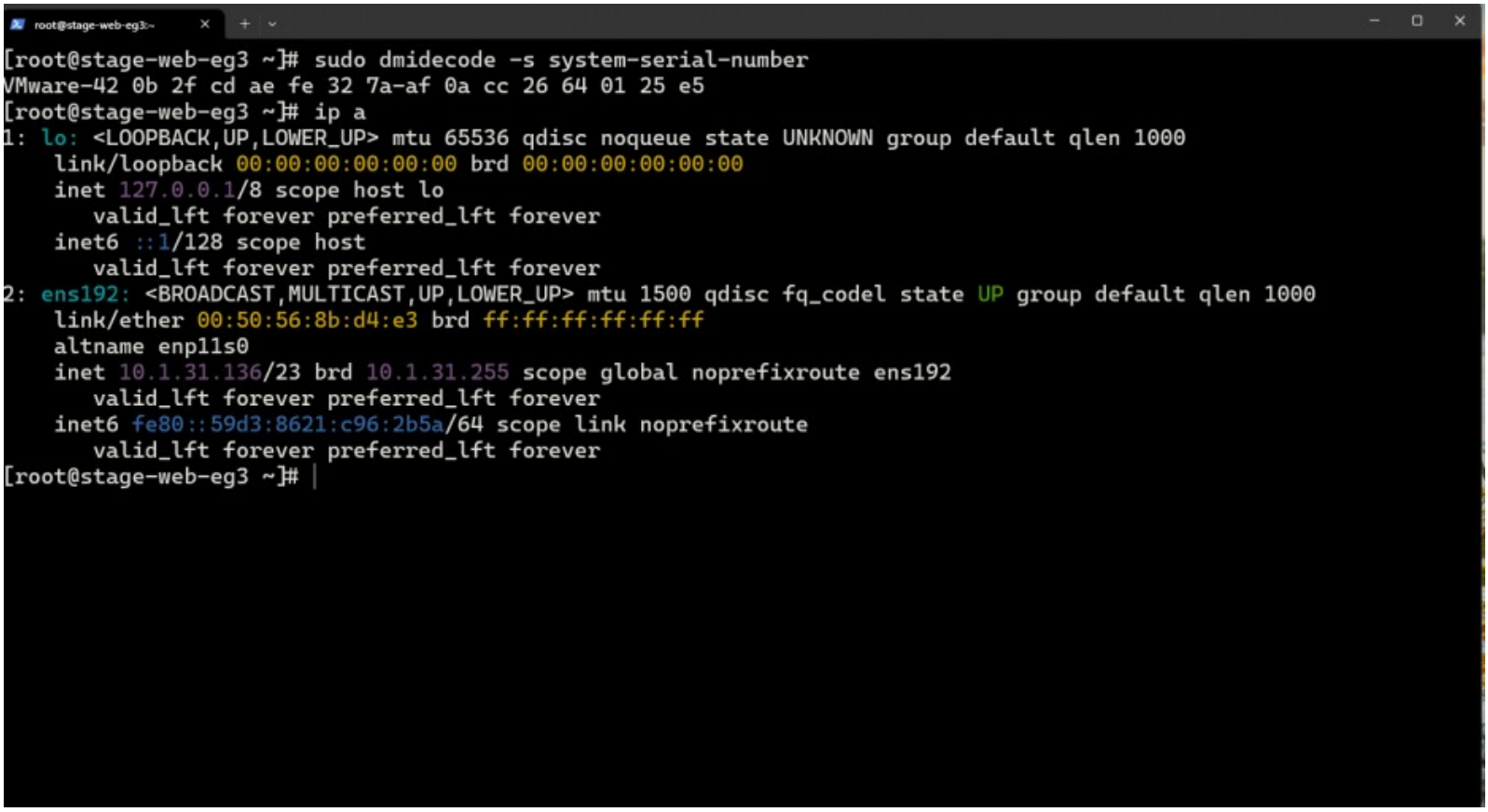
This screenshot shows NetworkManager and system identity configuration on stage-web-eg3.procore.prod1. The nmcli con show command confirms the active Ethernet connection bound to the primary network interface, while the loopback interface is also present as expected. The system hostname is then explicitly set using hostnamectl, ensuring the server’s hostname aligns with DNS, FreeIPA enrollment, and Ansible inventory requirements. All network identifiers, hostnames, and environment-specific details shown are sanitized for security purposes.



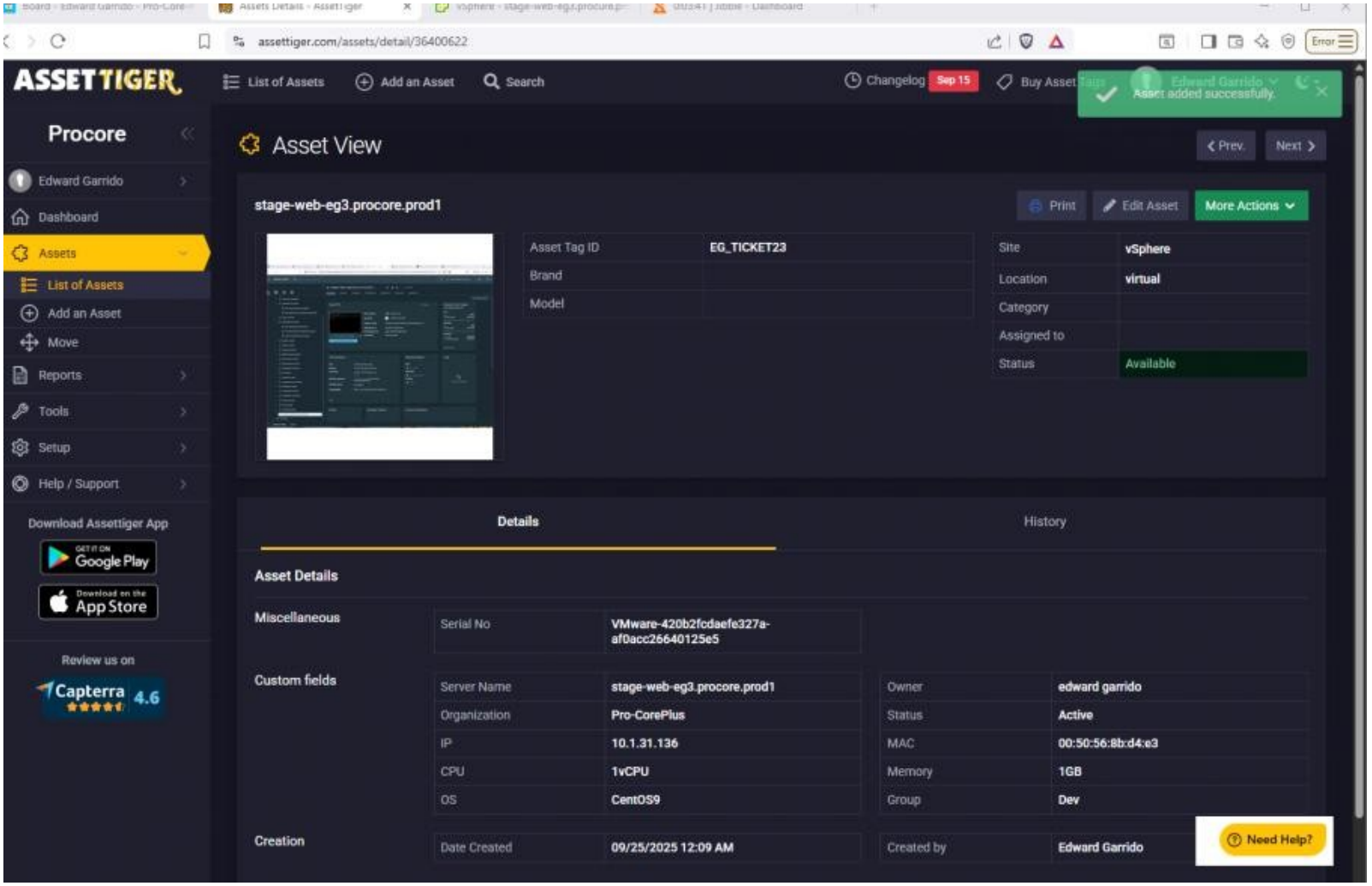
This screenshot shows local user account creation and privilege configuration on stage-web-eg3.procore.prod1. A new user is created, a password is set in compliance with system password policies, and the account is added to the wheel group to grant sudo privileges. The id command confirms the user's UID, GID, and group membership, verifying successful account setup and administrative access. All usernames, hostnames, and environment-specific details shown are sanitized for security purposes.



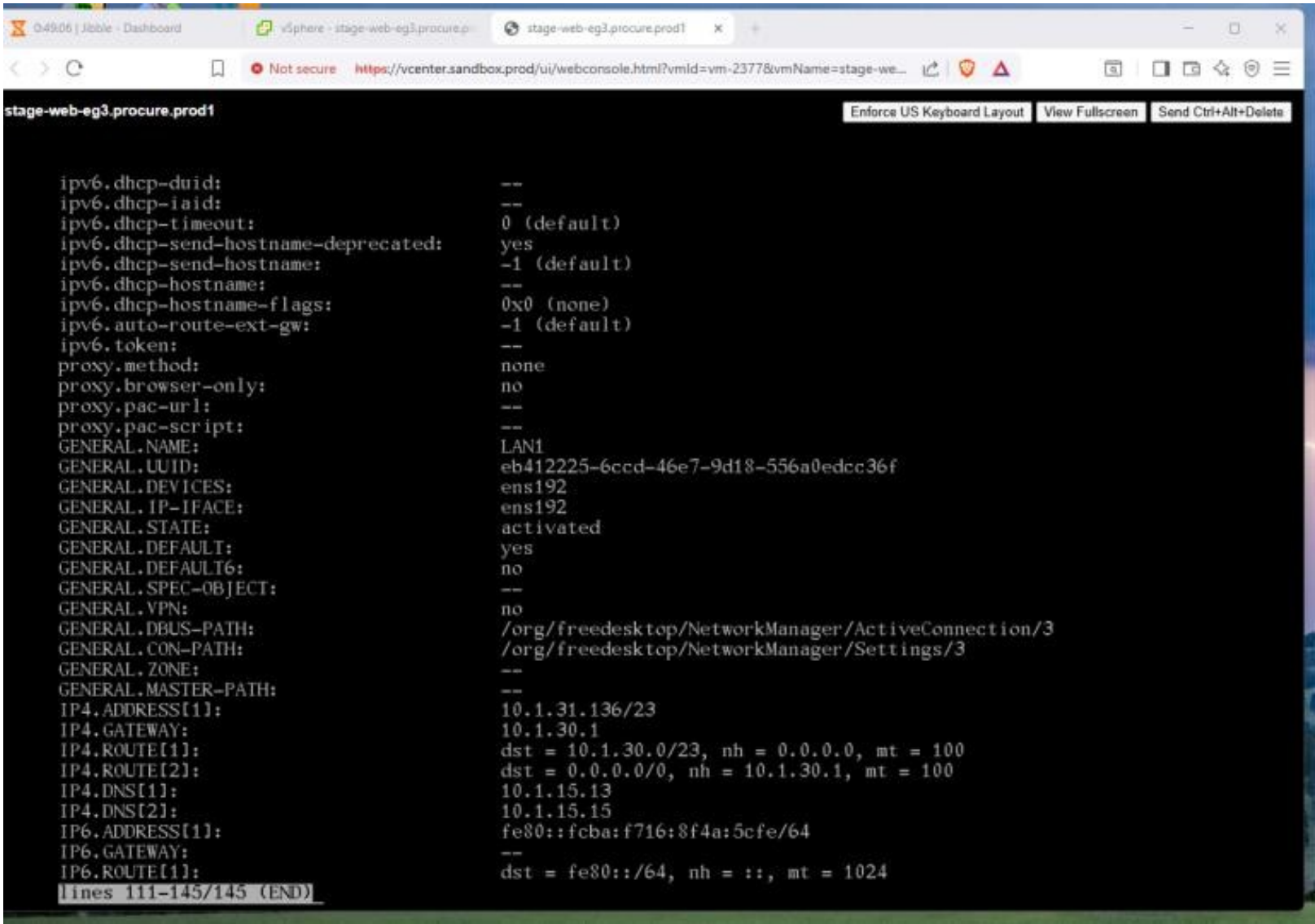
This screenshot shows system identification and network interface verification on stage-web-eg3.procore.prod1. The dmidecode command is used to retrieve the system’s serial number for inventory or asset tracking, followed by ip a to confirm active network interfaces. The output verifies the loopback interface and the primary Ethernet interface, including link status, MAC address, and assigned IPv4/IPv6 addresses. This confirms the host is properly identified and network-ready for domain services, SSH access, and automation tasks. All serial numbers, IP addresses, MAC addresses, hostnames, and environment-specific details shown are sanitized for security purposes.



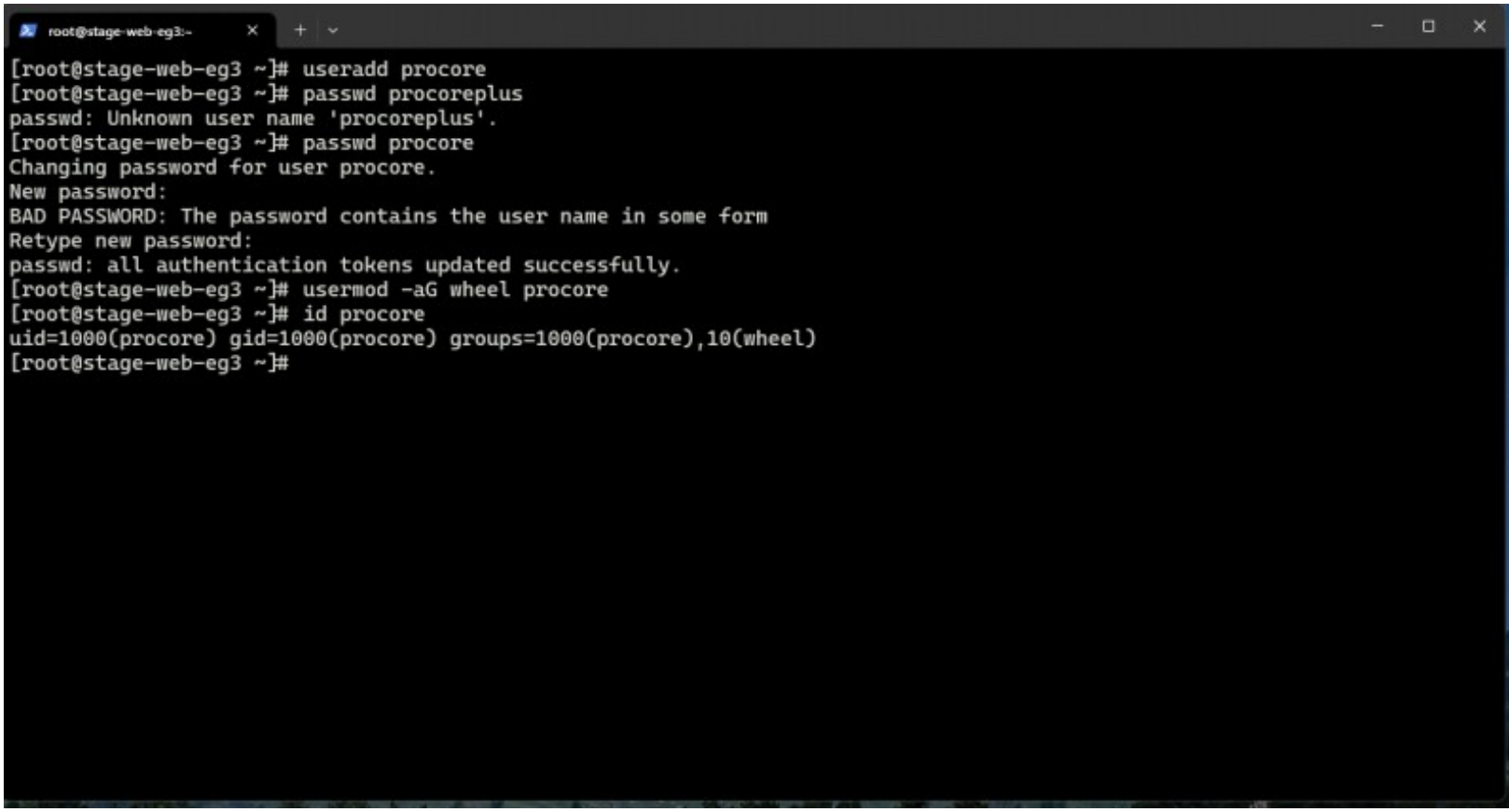
This screenshot shows the asset record creation and verification for the virtual server stage-web-eg3.procore.prod1 within an asset management system. The asset is tagged, categorized as a virtual machine hosted on vSphere, and includes key metadata such as operating system, CPU, memory, network identifiers, ownership, and status. This entry provides centralized tracking and accountability for the system as part of the infrastructure inventory. All serial numbers, IP addresses, MAC addresses, hostnames, and environment-specific details shown are sanitized for security purposes.



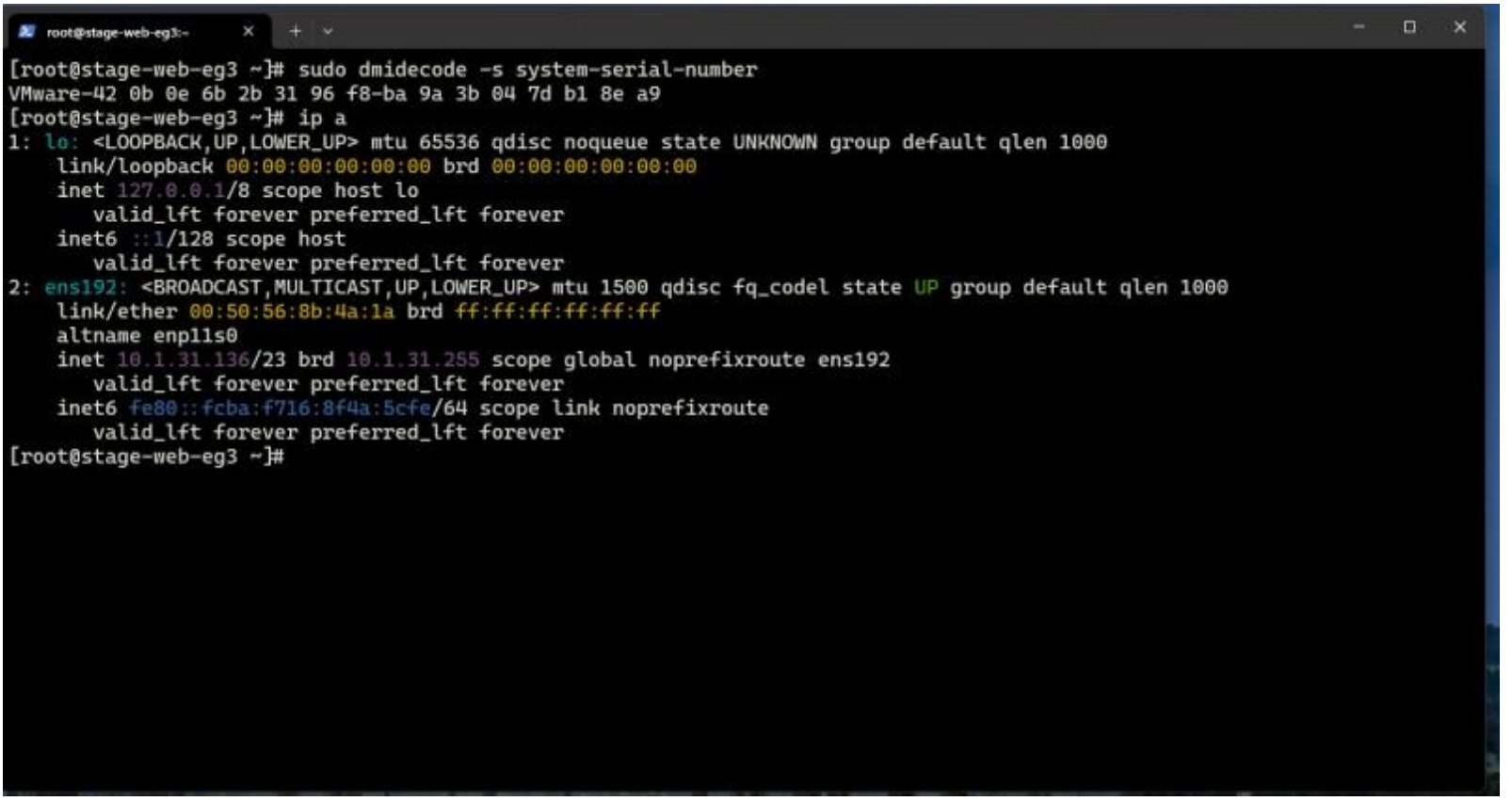
This screenshot displays NetworkManager connection details for the primary network interface on stage-web-eg3.procore.prod1. The output confirms the active Ethernet connection, associated device, UUID, interface state, assigned IPv4 and IPv6 addresses, routing entries, default gateway, and configured DNS servers. This verification ensures the system has correct network configuration and routing required for FreeIPA communication, SSH access, and Ansible automation. All IP addresses, gateways, DNS servers, UUIDs, and environment-specific details shown are sanitized for security purposes.



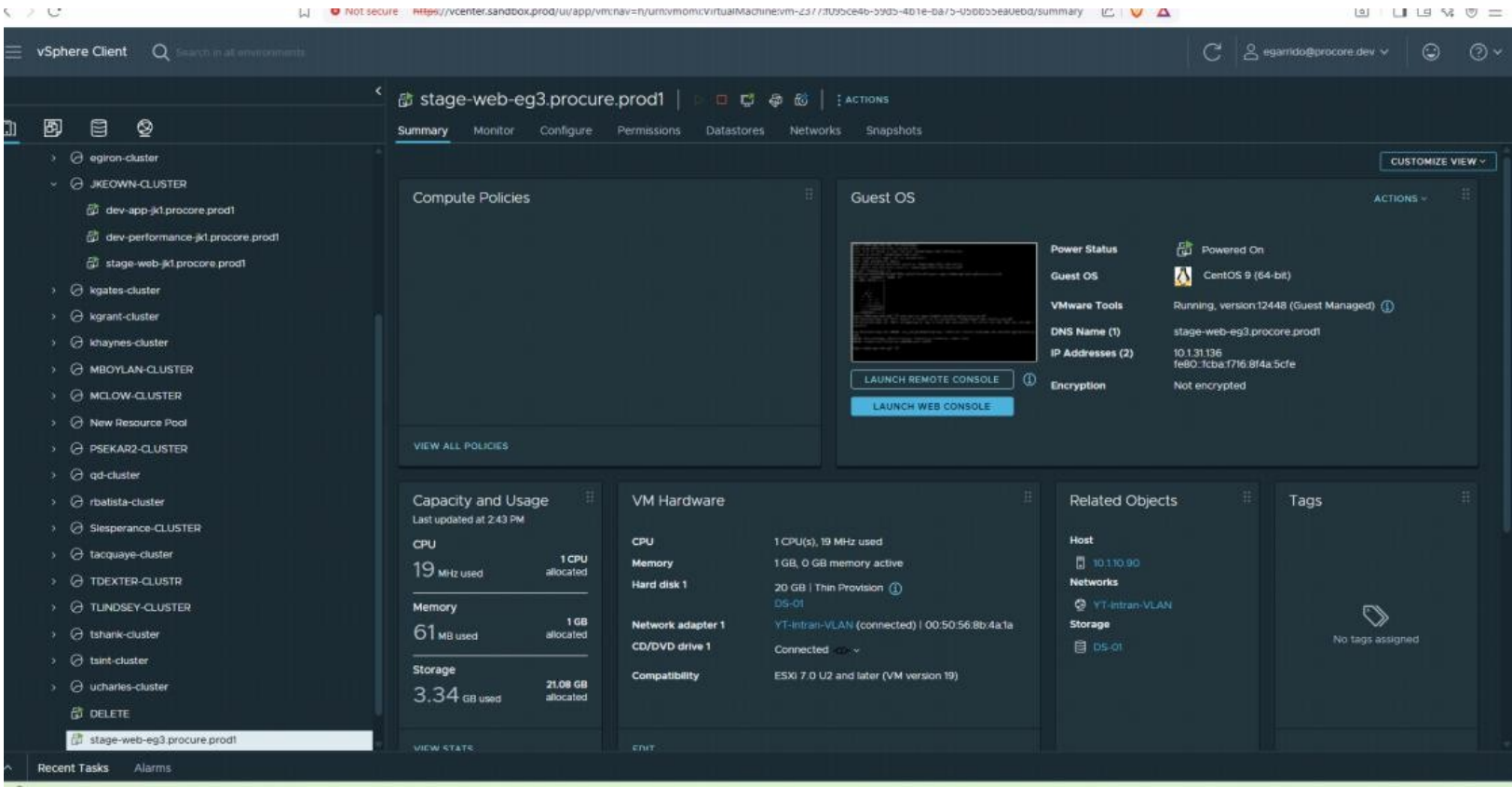
This screenshot shows local user management and privilege configuration on stage-web-eg3.procore.prod1. A local user account is verified, a password is set while enforcing system password complexity policies, and the user is added to the wheel group to grant sudo access. The id command confirms the user’s UID, GID, and group membership, validating successful administrative privilege assignment. All usernames, passwords, hostnames, and environment-specific details shown are sanitized for security purposes.



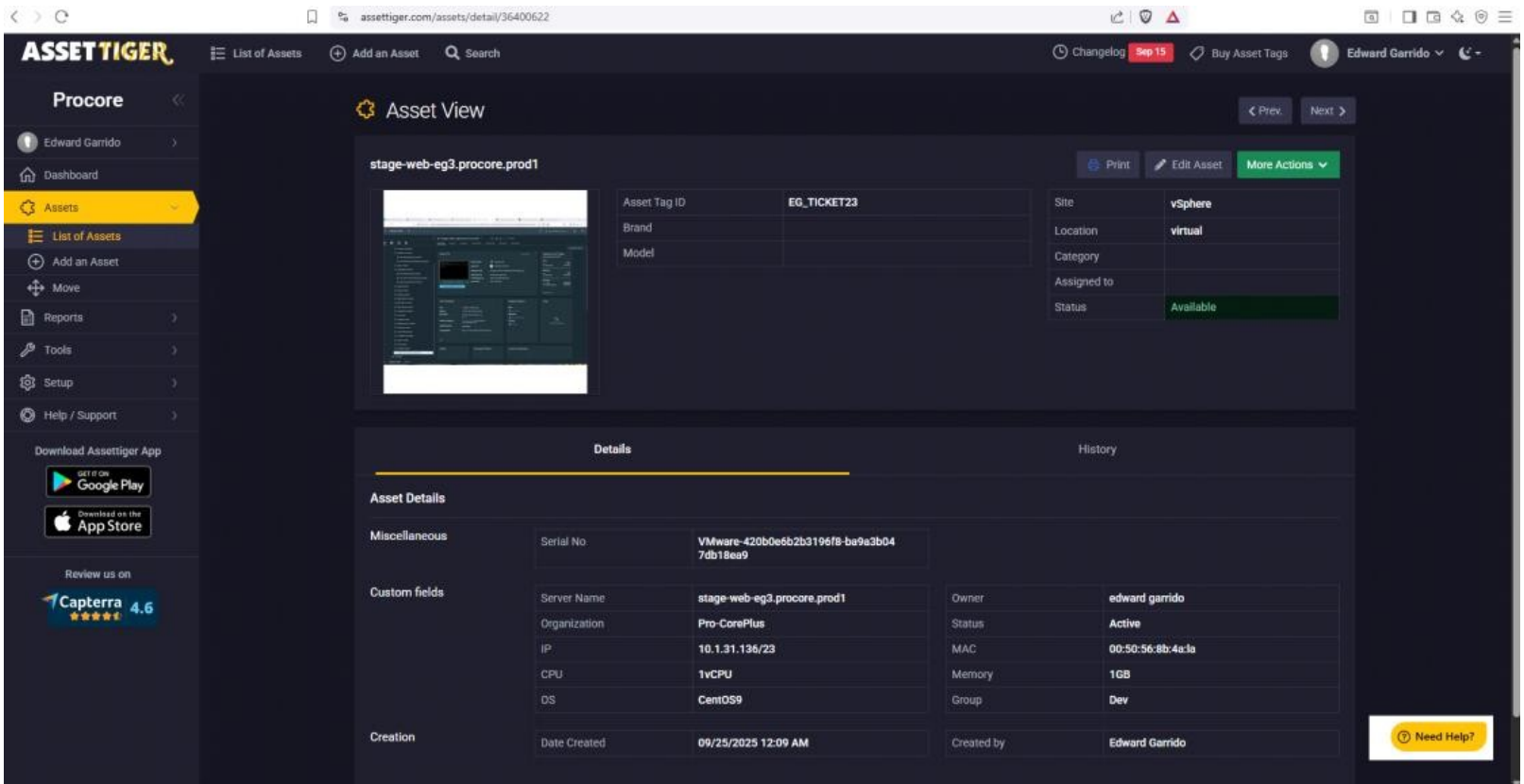
This screenshot shows system identification and network interface verification on stage-web-eg3.procore.prod1. The system serial number is retrieved using dmidecode, followed by the ip a command to confirm the loopback and primary Ethernet interface are up and properly configured. The output verifies link state, MTU, MAC address, and assigned IPv4/IPv6 addresses, confirming the host's network readiness for domain services, SSH access, and automation workflows. All serial numbers, IP addresses, MAC addresses, and environment-specific details shown are sanitized for security purposes.



This screenshot shows the vSphere Client summary view for the virtual machine stage-web-eg3.procore.prod1. It provides a consolidated overview of the VM’s status and configuration, including power state, guest operating system, VMware Tools status, DNS name, allocated CPU and memory resources, storage usage, network connectivity, and hardware compatibility. This view is typically used to confirm VM health, resource allocation, and readiness before or after system configuration tasks. All IP addresses, hostnames, identifiers, and environment-specific details shown are sanitized for security purposes.



This screenshot shows the asset inventory record for the virtual machine stage-web-eg3.procore.prod1 in the asset management system. The entry documents key metadata including asset tag, hosting platform (vSphere), virtualization status, operating system, CPU and memory allocation, network identifiers, ownership, and lifecycle status. This record provides centralized tracking, accountability, and visibility for infrastructure resources within the environment. All serial numbers, IP addresses, MAC addresses, hostnames, and environment-specific details shown are sanitized for security purposes.



Summary

This project documents the full lifecycle of provisioning and preparing a Linux virtual machine for enterprise use, including network configuration, hostname alignment, centralized identity management, secure SSH access, and Ansible automation validation. The system is deployed on vSphere, enrolled into a FreeIPA domain, configured for key-based authentication, and verified for reliable automation connectivity. Asset inventory tracking is also completed to ensure visibility and accountability. All IP addresses, hostnames, usernames, serial numbers, and environment-specific details shown are sanitized for security purposes.