Ubuntu Landscape - Final Demo Presentation

Slide 1: Title

Title: Ubuntu Landscape Centralized Ubuntu Systems Management

Subtitle: Monitor, patch, and automate fleet operations at scale

Speaker Notes: Today I'll walk you through how Landscape helps us manage Ubuntu systems centrally from patching and compliance to reporting and automation.

Slide 2: What is Landscape?

- Canonicals systems management tool for Ubuntu

- Web-based dashboard for managing thousands of systems

- Includes patching, package control, reporting, user/group management

Speaker Notes: Think of it as Red Hat Satellite for Ubuntu tailored for Ubuntu/Debian systems.

Slide 3: Core Components What Powers Landscape Behind the Scenes

Heres what makes Landscape tick just like any web app, it has a front, a brain, a messenger, and a storage room:

1. Web Interface (apache2)

This is the front door its what you see when you log in.

- Hosts the dashboard in your browser

- Handles secure HTTPS access

2. The Brain (landscape-server)

This decides what happens when you click buttons in the UI.

- Receives your actions (e.g., update this system)

- Passes those jobs to worker services

3. The Worker (landscape-job-handler)

This is the task-runner it installs updates, runs scripts, etc.

- Does the actual job on the backend

- Think of it as the robot arm that carries out instructions

4. The Messenger (landscape-message-server)

Keeps the conversation going between the server and your machines.

- Constantly listens for check-ins

- Tells systems what to do, and gets their responses

5. The Filing Cabinet (PostgreSQL)

Where everything gets saved hosts, jobs, users, history.

- Stores metadata, job results, and login sessions

6. The Agent (landscape-client)

Installed on every Ubuntu system it talks to the server.

- Checks in every few minutes

- Takes orders like: Run update, Send logs, Execute script

Speaker Notes: These six parts are what make Landscape work. The UI is just the entry point. Under the hood, youve got a smart backend, messengers, workers, and a client on every machine doing the actual job.

Slide 4: Key Features

- Inventory of all registered Ubuntu machines
- Live system metrics: memory, disk, CPU usage
- Package update management & automation
- Custom script execution across systems
- Role-based access & audit logs

Slide 5: Architecture Overview

- Landscape Server (self-hosted or cloud)
- Landscape Client on each managed machine
- PostgreSQL backend
- Uses HTTPS for secure communication

Slide 6: Demo Flow Checklist

Objective: Show end-to-end management of Ubuntu hosts

Live Steps:

1. Log in to Landscape UI
2. Show dashboard with system metrics
3. Filter machines by tags (e.g., prod, dev)
4. Register a new host:
   a. SSH into a new Ubuntu system
   b. Copy the Landscape server certificate:

      scp deploy@pln-landscape01.caal.dev:/etc/landscape/landscape.pem /etc/landscape/landscape.pem

   c. Install the Landscape client:

sudo apt install landscape-client

d. Configure /etc/landscape/client.conf

e. Register the system:

```
sudo landscape-config --computer-title "test-host" \
  --account-name standalone \
  --url https://pln-landscape01.caal.dev/message-system \
  --ping-url https://pln-landscape01.caal.dev/ping \
  --ssl-public-key /etc/landscape/landscape.pem \
  --silent
```

f. Refresh dashboard to confirm registration

5. Pick a machine and:

- Show system status and available updates
- Trigger a package update or run a script

6. Go to Script Execution run a command like `uptime`

7. Show scheduled updates or auto-approve feature

8. Optional: Add a tag or manage user/group access

Speaker Notes: Well copy the servers certificate, register a new system, and then demo basic management tasks like updates and scripts.

Slide 7: Use Cases

- Patch automation across 1000s of servers
- Audit and compliance checks
- Custom health-check scripts
- Host grouping by environment or team

Slide 8: Closing

- Landscape = visibility + control for Ubuntu systems

- Easy integration with CI/CD and automation tools

- Scalable for hybrid or air-gapped environments