Overview & Repository Layout

Purpose

This book documents PBR's Ansible-based configuration management for Linux infrastructure. It covers the `ssh-baseline` role, supporting playbooks, design rationale, deployment procedure, and operational reference.

The `ssh-baseline` role establishes a hardened, AD-integrated SSH access baseline on Ubuntu servers. It joins each host to Active Directory via SSSD, retrieves SSH public keys from AD (via the `sshPublicKey` schema extension), enforces Duo MFA on both SSH login and sudo, applies CIS-aligned sshd hardening, and configures fail2ban.

Source Repository

GitHub: `git@github.com:Puffing-Billy-Railway/pbr-infra.git`

Branch: `main` — all production-ready changes commit here. There are no other long-lived branches.

Tags: Semantic version tags mark each baseline release (`v2.3`, `v2.4`, `v2.4.1`, `v2.4.2`). The current production release is **v2.4.2**.

Cloning the repo

```
git clone git@github.com:Puffing-Billy-Railway/pbr-infra.git
cd pbr-infra
```

Vault

The repo contains an encrypted Ansible Vault file at `inventory/group_vars/all/vault.yml`. The vault password lives at `~/.ansible_vault_pass` on the control node (mode 0600). Vault contents include:

- `vault_ad_join_user` — AD service account UPN for realm join
- `vault_ad_join_password` — that account's password
- `vault_duo_ikey`, `vault_duo_skey`, `vault_duo_api_host` — Duo Auth API credentials

The vault is never decrypted to disk; `ansible-playbook` reads `--vault-password-file ~/.ansible_vault_pass` at runtime.

Current Deployment State

All hosts run **ssh-baseline v2.4.2**:

Host	IP	Virtualization	auditd	Notes
`pbr-uisp-kl1`	10.1.8.23	KVM	Managed	Canary — deploy and verify here first
`pbr-docker-kl1`	10.1.8.55	KVM (Ubuntu 24.04)	Managed	Docker host
`pbr-graylog-kl1`	10.1.8.26	LXC	Skipped	auditd auto-skipped on LXC (see Known Limitations)
`pbr-lme-kl1`	10.1.8.35	KVM	Managed	Logging Made Easy
`pbr-thingsboard-kl1`	10.1.8.25	LXC	Skipped	ThingsBoard for level crossing telemetry

---

Control Node

Hostname: `pbr-ansible-kl1`

Working directory: `~/pbr-infra` (under `pbr_admin`)

The control node is explicitly excluded from inventory targets — playbooks reference `hosts: targets` rather than `all`, so the control node cannot be accidentally hit by a baseline run. The relevant comment in `inventory/hosts.yml`:

```
# Control node - excluded from automation.
# Uncomment only if you intentionally need ansible-kl1 in inventory
# (e.g., for monitoring or facts gathering) - never as an ssh-baseline target.
```

```
# pbr-ansible-kl1:
#   ansible_host: 127.0.0.1
```

The ansible service account on the control node uses an ed25519 private key (`~/.ssh/ansible_svc`). Public key:

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBMc7IDlr/IZ5M/2HcXU7cGCKZ03SLjpr5cbmiHnokdP ansible-svc@pbr-ansible-kl1
```

This public key is installed on every target host by the bootstrap script (see Deployment Runbook).

Repository Layout

```
pbr-infra/
├── ansible.cfg                     # Inventory path, become config, vault password file
├── requirements.yml                # Collection dependencies
├── inventory/
│   ├── hosts.yml                   # Host definitions and `targets` group
│   └── group_vars/all/
│       ├── main.yml                # AD domain config (non-secret)
│       └── vault.yml               # Encrypted secrets (vault)
├── playbooks/
│   ├── preflight.yml               # Verification only (no changes)
│   ├── ssh-baseline.yml            # Preflight + apply baseline
│   ├── verify.yml                  # Post-deployment validation
│   └── teardown.yml                # Reverse the role (testing)
├── roles/
│   ├── preflight/                  # Preflight checks (separate role)
│   │   ├── defaults/main.yml
│   │   ├── meta/main.yml
│   │   └── tasks/
│   │       ├── main.yml
│   │       ├── local.yml           # OS, hostname, NTP, users, sudoers
│   │       ├── ad.yml              # AD DC reachability
│   │       ├── scepman.yml         # SCEPman CA reachability
│   │       ├── schema.yml          # sshPublicKey schema check
│   │       └── control-node.yml    # Vault password file, collections
│   └── ssh-baseline/               # Main role
```

```
│       ├── defaults/main.yml           # All tunable variables
│       ├── handlers/main.yml           # sshd, sssd, fail2ban, ca-cert restarts
│       ├── meta/main.yml
│       ├── tasks/
│       │   ├── main.yml                # Task orchestration
│       │   ├── preconditions.yml       # Ansible account local sudo group
│       │   ├── ca-trust.yml            # SCEPman root CA installation
│       │   ├── packages.yml            # apt installs, auditd auto-detect
│       │   ├── timezone.yml            # Australia/Melbourne
│       │   ├── ad-join.yml             # realm join, SSSD config
│       │   ├── sudo.yml                # AD sudo + pbr_admin sudoers drop-ins
│       │   ├── duo.yml                 # duo-unix install, PAM stacks
│       │   ├── sshd.yml                # Hardening drop-in, banner, validate
│       │   └── fail2ban.yml            # jail.local
│       └── templates/
│           ├── krb5.conf.j2            # Minimal client config; SRV discovery
│           ├── sssd.conf.j2            # AD provider, GPO disabled, access filter
│           ├── sshd_hardening.conf.j2  # 10-pbr-hardening.conf
│           ├── pam_sshd.j2             # /etc/pam.d/sshd with Duo + break-glass
│           ├── pam_sudo.j2             # /etc/pam.d/sudo with Duo + carve-outs
│           └── pam_duo.conf.j2         # ikey/skey/host, group restriction
└── scripts/
    └── bootstrap-ansible-user.sh       # Idempotent ansible-account bootstrap
```

Version Tags Overview

See the Known Limitations & Version History page for the full changelog. Quick reference:

Tag	Description
`v2.4.2`	Current release. Auto-skip auditd on LXC containers.
`v2.4.1`	Ensure ansible automation account is in local `sudo` group (post-hardening connectivity fix).
`v2.4`	Duo MFA on sudo for AD sudo group with carve-outs.
`v2.3`	Duo MFA on SSH via `duo-unix` from Duo's official repo (replacing Ubuntu universe `libpam-duo`).

Quick Reference: Standard Workflow

1. Bootstrap the ansible automation account on a fresh host (`scripts/bootstrap-ansible-user.sh`).
2. Pre-clean any stale AD computer object in AD Users & Computers.
3. Add the host to `inventory/hosts.yml` (both the `linux` children and the `targets` group).
4. Run preflight: `ansible-playbook playbooks/preflight.yml -l <host>`
5. Run baseline: `ansible-playbook playbooks/ssh-baseline.yml -l <host> --vault-password-file ~/.ansible_vault_pass`
6. Run verify: `ansible-playbook playbooks/verify.yml -l <host> -e verify_test_user=a.mfraser --vault-password-file ~/.ansible_vault_pass`
7. Manual SSH test from workstation as AD user and as `pbr_admin`.

See the **Deployment Runbook** page for the full procedure including known retry behaviour.

Where to Read Next

- **Architecture & Design Decisions** — the "why" behind each major choice in the role
- **Deployment Runbook — New Host** — step-by-step for adding a new host to the baseline
- **Configuration Reference** — every variable in `defaults/main.yml` explained

---

Revision #1
Created 2026-05-13 05:21:34 UTC by PBR_Documentation
Updated 2026-05-13 05:21:34 UTC by PBR_Documentation