Configuration Reference

Variable Source Hierarchy

Variables resolve in standard Ansible precedence order. The role uses three layers:

1. **Role defaults** — `roles/ssh-baseline/defaults/main.yml` (lowest precedence; the safe baseline)
2. **Group vars** — `inventory/group_vars/all/main.yml` (organisation-wide overrides, including vault-sourced secrets)
3. **Host vars** — `inventory/host_vars/<hostname>.yml` (per-host overrides; not currently used in this repo but supported)

The `group_vars/all/main.yml` file overrides the most security-sensitive defaults (AD domain, OUs, groups, SCEPman URL) so they cannot drift even if a role default is accidentally edited.

Group Vars (Organisation-Wide)

File: `inventory/group_vars/all/main.yml`

```
---
# AD join credentials - sourced from vault.yml (encrypted)
ad_join_user: "{{ vault_ad_join_user }}"
ad_join_password: "{{ vault_ad_join_password }}"

# Domain configuration
ad_domain: "pbr.org.au"
ad_computer_ou: "OU=Linux,OU=Servers,OU=Computers,OU=PBR,DC=pbr,DC=org,DC=au"

# Access control via AD security groups (must exist in AD)
ad_server_access_group: "SG_ServerAccess"
ad_sudo_group: "SG_Sudo"
```

```
# SCEPman PKI - root CA distribution endpoint
scepman_ca_url: "https://pki.pbr.org.au/ca"
```

Vault-Sourced Variables

Group var	Vault key	Purpose
`ad_join_user`	`vault_ad_join_user`	UPN of the AD service account used by `realm join`. Must have create-computer rights in the target OU.
`ad_join_password`	`vault_ad_join_password`	Password for the join service account.

The Duo credentials are also vault-sourced and referenced in `roles/ssh-baseline/templates/pam_duo.conf.j2`:

Template var	Vault key	Purpose
`duo_ikey`	`vault_duo_ikey`	Duo Auth API integration key
`duo_skey`	`vault_duo_skey`	Duo Auth API secret key
`duo_api_host`	`vault_duo_api_host`	Duo API hostname (e.g. `api-XXXXXXXX.duosecurity.com`)

To edit the vault:

```
cd ~/pbr-infra
ansible-vault edit inventory/group_vars/all/vault.yml \
    --vault-password-file ~/.ansible_vault_pass
```

Role Defaults: AD & Access

File: `roles/ssh-baseline/defaults/main.yml` (referenced; group_vars override these)

Variable	Default	Purpose
`ad_domain`	`pbr.org.au`	AD DNS domain. Used for realm membership, krb5.conf, SSSD.
`ad_computer_ou`	Linux servers OU	OU where computer objects are created by `realm join`.
`ad_server_access_group`	`SG_ServerAccess`	AD security group for read-only SSH access (no sudo).

Variable	Default	Purpose
`ad_sudo_group`	`SG_Sudo`	AD security group for sudo-enabled users. Members trigger Duo on sudo.
`pbr_admin_allowed_sources`	`10.0.0.0/8,192.168.0.0/16`	Source-IP allow-list (CIDR, comma-separated, no spaces) for the `pbr_admin` break-glass `Match` block.
`ad_access_filter`	*See below*	LDAP filter applied by SSSD for access control. Default is `memberOf=<ServerAccess DN>` OR `memberOf=<Sudo DN>`, both fully qualified.

`ad_access_filter` default:

```
(|(memberOf=CN=SG_ServerAccess,OU=Security,OU=Groups,DC=pbr,DC=org,DC=au)(memberOf=CN=SG_Sudo,OU=Security,OU=Groups,DC=pbr,DC=org,DC=au))
```

Role Defaults: PKI (SCEPman)

Variable	Default	Purpose
`scepman_ca_url`	`https://pki.pbr.org.au/ca`	Endpoint that returns the SCEPman root CA in DER format.
`scepman_ca_cert`	`/usr/local/share/ca-certificates/pbr-root-ca.crt`	PEM-format location of the trusted root CA (added to system trust store).
`scepman_ca_der`	`/etc/ssl/certs/pbr-root-ca.der`	DER-format location of the root CA (kept for reference; PEM is what's trusted).

Role Defaults: System

Variable	Default	Purpose
`timezone`	`Australia/Melbourne`	System timezone applied via `community.general.timezone`.
`manage_auditd`	`auto`	Whether to enable auditd. `auto` = skip on LXC (kernel audit netlink isolated), enable elsewhere. Accepts `true`, `false`, or `auto`.

Role Defaults: SSH Hardening

These map directly to `sshd_config` directives in `10-pbr-hardening.conf`.

Variable	Default	sshd_config directive	Notes
`ssh_port`	`22`	`Port`	Changing this requires systemd ssh.socket overrides on Ubuntu 22.10+.
`ssh_banner`	`/etc/issue.net`	`Banner`	Path to legal banner file.
`ssh_log_level`	`VERBOSE`	`LogLevel`	CIS Ubuntu 22.04 recommendation.
`ssh_login_grace_time`	`60`	`LoginGraceTime`	Seconds before unauthenticated connection drops.
`ssh_max_auth_tries`	`3`	`MaxAuthTries`	Per-connection auth attempt cap.
`ssh_max_sessions`	`4`	`MaxSessions`	Concurrent multiplexed sessions per connection.
`ssh_max_startups`	`10:30:60`	`MaxStartups`	Concurrent unauthenticated connections (start:rate:full).
`ssh_client_alive_interval`	`300`	`ClientAliveInterval`	Seconds between keepalive probes.
`ssh_client_alive_count_max`	`2`	`ClientAliveCountMax`	Idle connections drop after `interval × count_max` seconds.
`ssh_permit_root_login`	`no`	`PermitRootLogin`	Hard no.
`ssh_password_authentication`	`no`	`PasswordAuthentication`	Disabled globally; re-enabled for `pbr_admin` via `Match` block.
`ssh_pubkey_authentication`	`yes`	`PubkeyAuthentication`	Required by all flows.
`ssh_kbdint`	`yes`	`KbdInteractiveAuthentication`	Required for Duo PAM keyboard-interactive.
`ssh_allow_tcp_forwarding`	`no`	`AllowTcpForwarding`	Disabled.
`ssh_x11_forwarding`	`no`	`X11Forwarding`	Disabled.
`ssh_allow_agent_forwarding`	`no`	`AllowAgentForwarding`	Disabled.

Variable	Default	sshd_config directive	Notes
`ssh_compression`	`no`	`Compression`	Defence against compression-side-channel attacks.
`ssh_tcp_keep_alive`	`no`	`TCPKeepAlive`	Use SSH-level keep-alive instead.
`ssh_authentication_methods`	`publickey,keyboard-interactive`	`AuthenticationMethods`	Both required; keyboard-interactive is Duo via PAM.

Modern Crypto

Algorithm lists prepended with the post-quantum hybrid KEX where available:

Variable	Default
`ssh_ciphers`	`chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr`
`ssh_macs`	`hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,umac-128-etm@openssh.com`
`ssh_kex_algorithms`	`sntrup761x25519-sha512@openssh.com,curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp521,ecdh-sha2-nistp384,ecdh-sha2-nistp256`

Role Defaults: fail2ban

Variable	Default	Purpose
`fail2ban_bantime_default`	`3600`	Default ban duration in seconds (1 hour) for non-sshd jails.
`fail2ban_findtime`	`600`	Window in seconds during which `maxretry` failures trigger a ban.
`fail2ban_maxretry_default`	`5`	Failures within `findtime` before ban (default for non-sshd jails).
`fail2ban_sshd_maxretry`	`3`	Tighter setting for the sshd jail.
`fail2ban_sshd_bantime`	`86400`	24-hour ban for sshd failures.
`fail2ban_ignoreip`	*list, see below*	CIDRs exempt from banning.

Default `fail2ban_ignoreip`:

```
fail2ban_ignoreip:
  - "127.0.0.1/8"
  - "::1"
  - "10.1.0.0/16"        # PBR server LAN
  - "10.1.8.80/32"       # pbr-ansible-kl1 control node (explicit)
  - "192.168.0.0/16"     # Admin workstation VLANs supernet (TEMPORARY)
```

The 192.168.0.0/16 entry is annotated **TEMPORARY** in the role — intended to be removed when VLAN segmentation completes and admin workstations land on a single, well-defined CIDR.

Role Defaults: Duo MFA

Variable	Default	Purpose
`duo_failmode`	`safe`	`safe` = allow login if Duo cloud unreachable (fall through to single-factor publickey); `secure` = deny login during outage.
`duo_pushinfo`	`yes`	Include hostname and command in the Duo push notification.
`duo_prompts`	`3`	Max retries at the Duo prompt before failure.
`duo_autopush`	`yes`	Auto-send push to user's primary device.
`break_glass_user`	`pbr_admin`	Username carved out of the Duo PAM flow.
`duo_sudo_enabled`	`true`	Toggle Duo MFA on sudo (v2.4+).
`sudo_timestamp_timeout`	`30`	Minutes the sudo credential cache lasts; reduces Duo prompts during a session.

Preflight Role Defaults

File: `roles/preflight/defaults/main.yml`

Variable	Default	Purpose
`preflight_min_ubuntu_major`	`22`	Minimum Ubuntu major version. 22.04 LTS is the floor.

Variable	Default	Purpose
`preflight_required_users`	`[ansible, pbr_admin]`	Local accounts that must exist before baseline.
`preflight_ad_ports`	`[88, 389]`	Ports tested for AD DC reachability. 88 = Kerberos, 389 = LDAP.
`preflight_skip_schema_check`	`false`	Set true to bypass the AD schema check if `python3-ldap` is unavailable on the control node and you've verified schema manually.

Override Patterns

Per-host override via host_vars

Create `inventory/host_vars/<hostname>.yml`. Example: a host that requires a tighter source-IP allow-list:

```
---
# inventory/host_vars/pbr-pos-belgrave.yml
pbr_admin_allowed_sources: "10.1.8.0/24"  # POS LAN only
fail2ban_sshd_bantime: 604800              # 7 days for POS hosts
```

Forcing auditd on/off per host

```
---
# inventory/host_vars/pbr-graylog-kl1.yml
# Force-skip auditd even if the host migrates from LXC to KVM
manage_auditd: false
```

Adding a CIDR to fail2ban ignoreip

Override the full list (Ansible doesn't merge list defaults by default):

```
fail2ban_ignoreip:
  - "127.0.0.1/8"
  - "::1"
```

```
  - "10.1.0.0/16"
  - "10.1.8.80/32"
  - "192.168.0.0/16"
  - "203.0.113.42/32"   # NEW: external admin static IP
```

---

ansible.cfg Settings

The runtime configuration on `pbr-ansible-kl1` is fixed by `ansible.cfg` in the repo root:

```
[defaults]
inventory           = inventory/hosts.yml
remote_user         = ansible
private_key_file    = ~/.ssh/ansible_svc
host_key_checking   = True
retry_files_enabled = False
stdout_callback     = yaml
interpreter_python  = auto_silent
vault_password_file = ~/.ansible_vault_pass
roles_path          = roles
collections_path    = collections
forks               = 5

[privilege_escalation]
become              = True
become_method       = sudo
become_user         = root
become_ask_pass     = False

[ssh_connection]
pipelining          = True
ssh_args            = -o ControlMaster=auto -o ControlPersist=60s
```

Notable settings:

- `host_key_checking = True` — rejects connection to hosts with unknown SSH host keys. Adding a new host requires accepting its host key once (the bootstrap step naturally surfaces this).
- `vault_password_file` is set in `ansible.cfg`, so the `--vault-password-file` flag is technically redundant on the command line. It's included explicitly in this book's runbooks for

portability if the config changes.
- `forks = 5` caps concurrency. Combined with `serial: 1` in playbooks, the effective concurrency is 1 host at a time.
- `pipelining = True` reduces task overhead by skipping the SCP/SFTP transfer of small modules.

Collection Requirements

File: `requirements.yml`

```
---
collections:
  - name: ansible.posix
    version: ">=2.1.0"
  - name: community.general
    version: ">=12.0.0"
  - name: paloaltonetworks.panos
    version: ">=2.20"
  - name: arubanetworks.aoscx
    version: ">=10.0"
```

Used by ssh-baseline: `ansible.posix` (assorted modules), `community.general` (`timezone` module, `ldap_search` for schema check).

Other collections: `paloaltonetworks.panos` and `arubanetworks.aoscx` are listed for future use cases (Palo Alto NGFW automation, AOS-CX switch config) but are not used by the ssh-baseline role.

Install/update collections:

```
cd ~/pbr-infra
ansible-galaxy collection install -r requirements.yml --upgrade
```

Where to Read Next

- **AD Integration & SSSD** — how the AD variables map to SSSD config
- **Duo MFA Integration** — how the Duo variables map to `pam_duo.conf` and the PAM stacks
- **SSH Hardening Reference** — how each SSH variable lands in the deployed config

---

Revision #2
Created 2026-05-13 05:26:12 UTC by PBR_Documentation
Updated 2026-06-17 07:58:45 UTC by PBR_Documentation