CHECKLIST

Ubuntu Server SSH and Firewall Hardening Checklist

A practical production-readiness checklist for tightening SSH access and enforcing a restrictive firewall on Ubuntu Server.

Ubuntu Server SSH and Firewall Hardening Checklist

Use this checklist before moving a hardened Ubuntu Server into production. It focuses on the two controls that most directly reduce exposure: SSH and the firewall.

Goal

- Reduce attack surface without breaking legitimate administration or application access.
- Keep only required services reachable.
- Validate access paths before relying on the configuration in production.

1) Inventory what the server actually needs

- [ ] Identify the server?s role and expected inbound services.
- [ ] List every port that must be reachable from outside the host.
- [ ] Identify who needs SSH access.
- [ ] Identify from where SSH access should be allowed.
- [ ] Confirm whether any automation, monitoring, or backup jobs depend on SSH.
- [ ] Document any temporary exceptions that are still required during rollout.

2) Verify a recovery path before changing access controls

- [ ] Open a second administrative session before making changes.
- [ ] Confirm you have console, out-of-band, or break-glass access.
- [ ] Record a rollback plan.
- [ ] Confirm you can revert firewall changes if lockout occurs.
- [ ] Make sure the change window allows time for validation.

3) Harden SSH access

- [ ] Prefer key-based authentication for administrative users.
- [ ] Disable password authentication where operationally feasible.
- [ ] Restrict or disable direct root login.
- [ ] Limit SSH access to approved users or groups.
- [ ] Confirm only the required authentication methods remain enabled.
- [ ] Remove stale accounts or unused keys.
- [ ] Verify that any automation still authenticates successfully.
- [ ] Check that login banners, host key handling, and access logs are in place as required by policy.

4) Restrict network exposure with the firewall

- [ ] Set the default inbound posture to deny unless explicitly allowed.
- [ ] Allow SSH only from trusted admin sources or a jump host.
- [ ] Allow application ports only from the client networks that need them.
- [ ] Do not expose local-only services to the network.

- [ ] Review both inbound and outbound rules for unnecessary exceptions.
- [ ] Align host firewall rules with cloud security groups, network ACLs, or upstream zone policy.
- [ ] Remove test or temporary rules before production use.

5) Validate that access still works

- [ ] Test SSH from the approved source network.
- [ ] Test that SSH is blocked from unapproved source networks.
- [ ] Verify that the intended application ports are reachable only from approved sources.
- [ ] Confirm that no unexpected services are exposed.
- [ ] Disconnect the original session only after the new access path is confirmed.
- [ ] Verify administrative tasks can still be completed after the change.

6) Confirm production readiness

- [ ] SSH is reachable only from intended management sources.
- [ ] Password login has been removed or justified and documented.
- [ ] Root login is restricted or disabled.
- [ ] Firewall rules match the server?s role and service inventory.
- [ ] Only required ports are open.
- [ ] Recovery access is documented and tested.
- [ ] Rollback steps are recorded.
- [ ] Change approval and validation evidence are saved.

7) Common trade-offs to review before final approval

- [ ] Have you validated all legitimate access paths before disabling password login?

- [ ] Have you documented how new SSH keys are issued and revoked?
- [ ] Have you confirmed firewall rules will stay aligned as services change?
- [ ] Have you accounted for both host firewall policy and upstream network controls?
- [ ] Have you checked for shared accounts or ad hoc source networks that would weaken the design?

Quick pass/fail summary

- [ ] SSH is limited to trusted sources.
- [ ] SSH authentication is as strong as the environment allows.
- [ ] The firewall only allows required ports.
- [ ] A rollback path exists.
- [ ] Validation was completed from both allowed and disallowed sources.

Notes

- If the server is managed through a bastion, orchestration layer, or private management network, make sure the allowlist reflects that architecture.
- If you are transitioning from password-based access, stage the change and verify every legitimate path before removing the old method.
- Treat firewall policy as configuration that must be maintained as the server?s role evolves.