TUTORIAL

How to Configure FirewallD on CentOS 8 Step by Step

Learn how to configure FirewallD on CentOS 8 step by step: verify prerequisites, choose the right zone, open services or ports, validate rules, and confirm the firewall behaves as expected before production use.

Operating Systems - July 18, 2026

Why this configuration matters

CentOS 8 systems often need a firewall policy that allows only the services you intentionally expose while keeping everything else blocked by default. The practical problem is not just opening a port; it is making sure the rule lands in the correct zone, survives a reload, and matches the real network path into the server. If you configure the wrong zone or skip validation, the service may appear reachable in one test and fail in production, or worse, expose more than intended.

This tutorial shows how to configure FirewallD on CentOS 8 step by step so you can define the correct zone, allow specific services or ports, verify the active rules, and confirm the final state before production use. By the end, you will know how to apply a safe workflow, check the result with commands, and recognize the common failure points before they become outages.

Prerequisites and stop-here-if warnings

Before you change firewall rules, confirm that you have console or out-of-band access, or another reliable recovery path. If you are connected only through SSH over the interface you plan to modify, stop here until you have a fallback. A mistaken zone or a dropped SSH service can lock you out immediately.

You should also know which services the host must accept, which source networks are trusted, and which interface is used for management traffic. If that information is unclear, gather it first. Firewall rules are policy decisions, not guesses.

What you need

- Root access or a sudo-capable account
- A CentOS 8 host with FirewallD installed and running
- Knowledge of the required service names, ports, and trusted source ranges
- A plan for verifying SSH access before and after changes

Initial state to confirm

Check whether FirewallD is active and what zone is currently in use:

Expected output is that FirewallD reports running, and the active zones list shows which interface belongs to which zone. If FirewallD is not running, start it before continuing:

Common failure: administrators assume a rule is applied globally, but FirewallD evaluates traffic by zone. If the interface is attached to the wrong zone, the rule will not have the effect you expect.

Step 1: Identify the correct zone for the interface

The first job is to determine which zone receives traffic from the network interface you care about. This matters because a rule added to the wrong zone can look correct in configuration while still not protecting or exposing the target service.

Goal

Attach your management or service interface to the correct policy zone before opening any ports.

Action

List available zones and inspect the one currently active on the interface:

If you need to bind an interface to a specific zone, do it explicitly. For example, to place eth0 in the public zone:

To make the assignment persistent across reboots, add --permanent and then reload FirewallD:

If you are designing the policy from scratch, it can help to review the zone model before changing production rules. A practical companion explanation is CentOS FirewallD Configuration for Secure Server Hardening, which focuses on policy design before exposure.

Expected output

The interface should appear under the intended zone when you run --get-active-zones. After a reload, the zone should still own the interface.

Validation

Confirm the mapping and inspect the zone details:

You should see the correct interfaces, services, and ports for that zone.

Common failure

A frequent mistake is editing the permanent configuration but forgetting that the runtime state is separate until you reload or apply the same change to runtime. Another common issue is assigning the interface to a zone while the host still receives traffic through a different uplink than expected.

Step 2: Decide whether to allow a service or a port

FirewallD can allow traffic by predefined service name or by raw port. Use a service when the application matches a known service definition; use a port when you need a specific protocol and port combination that is not covered by a service entry.

Goal

Choose the least ambiguous rule type for the application you are exposing.

Action

List the available services to see whether one matches your application:

If the service is available, allow it in the correct zone. For example, to permit SSH in the public zone:

If you need a port instead, specify the protocol explicitly. For example, to allow TCP 8443:

Expected output

The command should return success. The rule becomes part of the permanent configuration and will appear after reload.

Validation

Check the configured zone to confirm the rule is present:

If you are allowing SSH only for a controlled network, validate that the host is still reachable from your management path before closing the session. For more on safe exposure patterns, see CentOS 7 FirewallD Rules for Secure Service Exposure, which explains how to avoid exposing services broadly when a narrower rule is enough.

Common failure

Opening a port in the wrong protocol is a classic error. A TCP service will not work if you opened only UDP, and vice versa. Another mistake is assuming a service definition includes every required dependency; some applications need additional ports or source restrictions.

Step 3: Apply rules safely and preserve access

You should distinguish between temporary runtime changes and permanent configuration. Runtime changes let you test behavior quickly; permanent rules survive reboots. For production changes, it is usually safest to test first, then commit.

Goal

Add the rule, validate it in the running firewall, then persist it if it behaves correctly.

Action

To test a rule at runtime first, omit --permanent:

Validate access from a second terminal or trusted host. If the result is correct, save the same rule permanently:

If you need to remove a mistaken rule, reverse the command:

Expected output

The runtime command should return success, and after reload the rule should still be present. The service should remain reachable only through the intended zone and path.

Validation

Use a listing command to verify both the active and permanent configuration:

The two views should match after reload.

Common failure

Administrators sometimes reload after editing the permanent configuration but forget to verify the runtime result. Another common issue is adding a rule to a zone that does not own the target interface, which makes the configuration appear correct while the traffic still fails.

Step 4: Restrict exposure by source when needed

Not every service should be open to every source. In many environments, the better control is to allow a service only from a trusted subnet or an administrative jump host. This keeps the rule targeted and reduces the blast radius if the host is scanned from elsewhere.

Goal

Allow traffic only from a trusted source network where policy requires it.

Action

Add a rich rule that permits a service from a specific source network. For example, to allow SSH only from 192.0.2.0/24 in the public zone:

You can use the same pattern with a port-based rule if needed.

Expected output

The rule should be saved, reloaded, and applied only to traffic matching the defined source network.

Validation

List the zone contents and verify the rich rule exists:

From a trusted source, confirm that access still works. From an untrusted source, confirm that the connection is blocked.

Common failure

A source-restricted rule can fail silently if the source address is not the actual address seen by the firewall, such as when NAT or a proxy changes the source path. In that case, verify the upstream network path before changing the firewall rule.

Step 5: Reload, verify, and confirm the final state

The final check is not optional. A firewall change is not complete until you confirm that the running configuration matches what you intended and that the host remains reachable where required.

Goal

Ensure the rule is active, persistent, and aligned with the correct zone.

Action

Reload FirewallD if you changed permanent settings:

Then verify the active state:

If you need a deeper check of what FirewallD believes is configured, inspect both runtime and permanent settings. This is especially useful when a change seems to ?not stick? after reload.

Expected output

- FirewallD reports running
- The intended interface is attached to the correct zone
- The required service or port appears in the zone listing
- The host remains reachable only through approved paths

Validation

From a client system, test the service itself rather than only the firewall rule. For example, use curl, nc, ssh, or the application-specific client as appropriate. Firewalls permit traffic; they do not guarantee the service is healthy.

Common failure

A service may still fail even though the firewall is correct. In that case, the issue is likely in the application listener, SELinux policy, binding address, or upstream routing. If the firewall rule looks correct but access still fails, check whether the application is actually listening on the expected interface and port. When the service is affected by mandatory access controls, Configure SELinux Booleans on CentOS to Enforce Least Privilege can help you decide whether the fix belongs in SELinux rather than in FirewallD.

Step 6: Keep the configuration operationally safe

A good firewall setup is one that can be reviewed and maintained under change control. That means knowing what was added, why it was added, and how to remove it later if the application changes.

Goal

Leave the system in a documented, predictable state that can be audited and rolled back.

Action

Capture the effective configuration after each meaningful change:

Record which zone is tied to which interface, which services or ports are allowed, and whether any rules are source-restricted. If the host will be reused for another role, remove stale permissions before handing it over.

Expected output

You should have a concise view of the final policy and a clear rollback path through the matching --remove-service, --remove-port, or --remove-rich-rule commands.

Validation

Review the policy as if you were validating it after an incident:

- Is the interface in the intended zone?
- Are only the required services or ports open?
- Are any source restrictions still appropriate?
- Does the permanent configuration match the running state?

Common failure

The most common long-term mistake is drift: temporary tests become permanent exposures, and old ports remain open after an application change. Periodic review prevents that from becoming a security problem.

What the finished state should look like

After you configure FirewallD on CentOS 8 correctly, the host should have a clear and auditable network policy:

- The intended interface belongs to the correct zone
- Only the required services or ports are allowed
- Optional source restrictions are in place where needed
- The active and permanent settings match after reload
- The service is reachable from approved clients and blocked elsewhere

If those conditions are true, the firewall is doing its job as a control layer rather than a guess-based port opener. That is the operational standard you want before production use.

Use this guidance together with CentOS 7 hardening with SELinux and Firewalld and BitLocker recovery key management to connect the workflow with related operational context already available on the site.