This guide describes how to troubleshoot configuration issues for a Google Cloud internal TCP/UDP load balancer.

The types of issues discussed in this guide include the following:

- General connectivity issues
- Backend failover issues (beta (/products/?hl=EN#product-launch-stages))
- Load balancer as next-hop issues

Before investigating issues, familiarize yourself with the following pages.

For general connectivity:

- Internal TCP/UDP Load Balancing concepts (/load-balancing/docs/internal/index)
- Setting Up Internal TCP/UDP Load Balancing (/load-balancing/docs/internal/setting-up-failover)

For failover:

- Failover concepts for Internal TCP/UDP Load Balancing (/load-balancing/docs/internal/failover-overview)
- Configuring failover for Internal TCP/UDP Load Balancing (/load-balancing/docs/internal/setting-up-failover)

For next hop:

- Next-hop concepts for Internal TCP/UDP Load Balancing (/load-balancing/docs/internal/ilb-next-hop-overview)
- Setting up Internal TCP/UDP Load Balancing for third-party appliances (/load-balancing/docs/internal/setting-up-ilb-next-hop)

- **Symptom:** I can't connect to my internal TCP/UDP load balancer from a VM client in another region.
- **Reason:** internal TCP/UDP load balancers is regional. They're only accessible from their own region.

If you can't connect to an internal TCP/UDP load balancer, check for the following issues:

- Ensure that ingress allow firewall rules (/load-balancing/docs/health-check-concepts) are defined to permit health checks to backend VMs.
- Ensure that ingress allow firewall rules (/vpc/docs/firewalls) allow traffic to the backend VMs from clients.
- Make sure that the client connecting to the load balancer is in the same region as the load balancer.

If you are using Shared VPC and you cannot create a new internal TCP/UDP load balancer in a particular subnet, an organization policy might be the cause. In the organization policy, add the subnet to the list of allowed subnets or contact your organization administrator. For more information, refer to the `constraints/compute.restrictSharedVpcSubnetworks` (/resource-manager/docs/organization-policy/org-policy-constraints) constraint.

If you've configured failover for an internal TCP/UDP load balancer, the following sections describe the issues that can occur.

- Make sure that you've designated at least one failover backend.
- Verify your failover policy settings:
    - Failover ratio

- Dropping traffic when all backend VMs are unhealthy
- Disabling connection draining on failover

- **Symptom:** The active pool is changing back and forth (flapping) between the primary and failover backends.
- **Possible reason:** Using managed instance groups with autoscaling and failover might cause the active pool to repeatedly failover and failback between the primary and failover backends. GCP doesn't prevent you from configuring failover with managed instance groups, because your deployment might benefit from this setup.

Disabling connection draining only works if the backend service is set up with protocol TCP.

The following error message appears if you create backend service with UDP while connection draining is disabled:

Currently, failover options are only available in the Beta API. If creation of a backend service fails with an error saying that `failover options` is not a valid field, make sure that you've created the backend service using the correct API (`gcloud beta compute backend-services...`).

First check the following: If the client VM is **also** a backend VM of the load balancer, it's expected behavior that connections sent to the IP address of the load balancer's forwarding rule are always answered by the backend VM itself. For more information, refer to testing connections from a single client (/load-balancing/docs/internal/index#single_client_tests) and sending requests from load balanced VMs (/load-balancing/docs/internal/setting-up-internal#test-from-backend-vms).

If the client VM is **not** a backend VM of the load balancer:

- For requests from a single client, refer to testing connections from a single client (/load-balancing/docs/internal/index#single_client_tests) so that you understand the limitations of this method.
- Ensure that you have configured ingress allow firewall rules to allow health checks (/load-balancing/docs/health-check-concepts).
- For a failover configuration, make sure that you understand how membership in the active pool (/load-balancing/docs/internal/failover-overview#active_pool) works, and when Google Cloud performs failover and failback (/load-balancing/docs/internal/failover-overview#failover_failback). Inspect your load balancer's configuration:
    - Use the Cloud Console to check for the number of healthy backend VMs in each backend instance group. The Cloud Console also shows you which VMs are in the active pool.
    - Make sure that your load balancer's failover ratio is set appropriately. For example, if you have ten primary VMs and a failover ratio set to `0.2`, this means Google Cloud performs a failover when **fewer than** two (`10 × 0.2 = 2`) primary VMs are healthy. A failover ratio of `0.0` has a special meaning: Google Cloud performs a failover when no primary VMs are healthy.

Edit your backend service's failover policy (#failover_policy). Ensure that connection draining on failover is enabled.

When you set an internal TCP/UDP load balancer to be a next hop (/load-balancing/docs/internal/setting-up-ilb-next-hop) of a custom static route, the following issues might occur:

- If you can't ping your backend, keep in mind that a route with the internal TCP/UDP load balancer set to be the next hop is only supported for TCP and UDP traffic. Other protocol packets, such as ICMP, are dropped.
- When using an internal TCP/UDP load balancer as a next hop for a custom static route, all TCP and UDP traffic is delivered to the load balancer's healthy backend VMs, regardless of the protocol configured for the load balancer's internal backend service, and regardless of the port or ports configured on the load balancer's internal forwarding rule.
- Ensure that you have created ingress allow firewall rules that correctly identify sources of traffic that should be delivered to backend VMs via the custom static route's next hop. Packets that arrive on backend VMs preserve their source IP addresses, even when delivered by way of a custom static route.

The destination range of a custom static route can't be more specific than any subnet route in your VPC network. If you receive the following error message when creating a custom static route:

- You cannot create a custom static route with a destination that exactly matches or is more specific (with a longer mask) than a subnet route (/vpc/docs/routes#subnet-routes). Refer to applicability and order (/vpc/docs/routes#instancerouting) for further information.
- If packets go to an unexpected destination, remove other routes in your VPC network with more specific destinations. Review the routing order (/vpc/docs/routes#routeselection) to understand Google Cloud route selection.

You cannot assign a network tag to a custom static route when the next hop is an internal TCP/UDP load balancer. For example, the following `gcloud` command produces the error message listed below:

- See Internal TCP/UDP Load Balancing Concepts (/load-balancing/docs/internal/index) for important fundamentals.
- See Failover concepts for Internal TCP/UDP Load Balancing (/load-balancing/docs/internal/failover-overview) for important information about failover.
- See Internal Load Balancing and DNS Names (/load-balancing/docs/dns-names) for available DNS name options your load balancer can use.
- See Setting Up Internal TCP/UDP Load Balancing (/load-balancing/docs/internal/setting-up-internal) for an example internal TCP/UDP load balancer configuration.
- See Configuring failover for Internal TCP/UDP Load Balancing (/load-balancing/docs/internal/setting-up-failover) for configuration steps and an example internal TCP/UDP load balancer failover configuration.
- See Internal TCP/UDP Load Balancing Logging and Monitoring (/load-balancing/docs/internal/internal-logging-monitoring) for information on configuring Stackdriver logging and monitoring for Internal TCP/UDP Load Balancing.
- See Internal TCP/UDP Load Balancing and Connected Networks (/load-balancing/docs/internal/internal-lb-and-other-networks) for information about accessing internal TCP/UDP load balancers from peer networks connected to your VPC network.