Networking Troubleshooting Case Studies for CloudOps Engineers

10 Real-world Networking Scenarios with Logs, Commands, and Explanations

Case 1: DNS Resolution Failure

Problem: Application servers failed to resolve internal hostnames after reboot.

Investigation:

```
$ ping api.internal.company.com
ping: unknown host api.internal.company.com
$ journalctl -u systemd-resolved | tail -n 5
Failed to send hostname reply: cache corrupted
$ resolvectl status
Link 2 (eth0): Current DNS Server: 10.0.0.2
```

Root Cause: systemd-resolved cache corruption prevented DNS resolution.

Resolution: Restarted the resolver: `systemctl restart systemd-resolved` and flushed caches.

Prevention: Implement post-boot DNS health checks and monitor /etc/resolv.conf consistency.

Case 2: Default Gateway Misconfiguration

Problem: Instances in private subnet unable to reach Internet despite NAT gateway.

Investigation:

```
$ ip route
default via 172.16.0.1 dev eth0 proto static
$ ping 8.8.8.8
Network unreachable
$ ip route del default && ip route add default via 172.16.1.1
```

Root Cause: Default route pointed to incorrect gateway IP after DHCP renewal.

Resolution: Updated default gateway and configured DHCP static routes.

Prevention: Use cloud-init to enforce correct routes on startup.

Case 3: FirewallD Blocking Service Port

Problem: Application not reachable on TCP port 8080 from load balancer.

Investigation:

```
$ firewall-cmd --list-all
services: ssh dhcpv6-client
$ firewall-cmd --add-port=8080/tcp --permanent
$ firewall-cmd --reload
```

Root Cause: FirewallD missing port rule for the service.

Resolution: Opened port 8080/tcp and reloaded FirewallD configuration.

Prevention: Maintain firewall rule baseline with Ansible automation.

Case 4: Proxy Misconfiguration

Problem: YUM and curl commands failed while wget worked fine.

Investigation:

```
$ cat /etc/yum.conf | grep proxy
proxy=http://proxy.company.local:8080
$ curl https://repo.company.com
curl: (56) Received HTTP code 403 from proxy
```

Root Cause: Proxy authentication misconfigured for non-interactive services.

Resolution: Updated YUM and system proxy configuration with credentials.

Prevention: Validate proxy settings in systemd environment files.

Case 5: NTP Time Drift Causing SSL Errors

Problem: SSL handshake failures observed intermittently on app servers.

Investigation:

```
$ timedatectl status
System clock synchronized: no
$ grep 'SSL3_GET_SERVER_CERTIFICATE' /var/log/httpd/error_log
[error] certificate verify failed: certificate not yet valid
```

Root Cause: NTP service stopped, causing system clock to drift >5 minutes.

Resolution: Restarted chronyd and forced time sync: `chronyc makestep`.

Prevention: Enable chronyd service and configure monitoring for clock offset.

Case 6: Network Bonding Misconfiguration

Problem: Bonded interface intermittently dropped packets under load.

Investigation:

```
$ cat /proc/net/bonding/bond0
Bonding Mode: active-backup
MII Status: down for eth1
$ dmesg | grep bond0
bond0: link status down for slave eth1
```

Root Cause: Mismatch between bonding mode and switch configuration.

Resolution: Reconfigured switch to support LACP (802.3ad) mode.

Prevention: Standardize bonding configuration and document switch compatibility.

Case 7: Interface Naming Mismatch After Kernel Upgrade

Problem: After kernel update, network interfaces renamed from eth0 to ens33.

Investigation:

```
$ dmesg | grep eth0
Device not found
$ ip link
ens33: mtu 1500
```

Root Cause: Predictable network interface naming policy changed post-upgrade.

Resolution: Updated network configuration files to use new interface names.

Prevention: Use consistent naming via udev rules or GRUB parameter `net.ifnames=0`.

Case 8: MTU Mismatch Causing Packet Loss

Problem: Intermittent connection issues between VPC and on-prem VPN.

Investigation:

```
$ ping -M do -s 1472 10.20.0.1
Frag needed and DF set (mtu = 1450)
$ ip link set dev eth0 mtu 1450
```

Root Cause: MTU mismatch between VPN tunnel and instance interface.

Resolution: Adjusted MTU to 1450 for tunnel compatibility.

Prevention: Document and enforce MTU settings for hybrid network connections.

Case 9: IPv6 Misrouting in Hybrid Environment

Problem: Instances with dual-stack enabled intermittently unreachable.

Investigation:

```
$ ping6 google.com
connect: Network is unreachable
$ ip -6 route
default via fe80::1 dev eth0 metric 1024
```

Root Cause: IPv6 default route pointing to invalid gateway.

Resolution: Removed incorrect IPv6 route and disabled IPv6 autoconf temporarily.

Prevention: Define static IPv6 routes and use RA suppression in hybrid environments.

Case 10: IPtables Persistence Issue After Reboot

Problem: Custom NAT rules disappeared after instance reboot.

Investigation:

```
$ iptables -t nat -L
Chain POSTROUTING (policy ACCEPT) ... no rules
$ iptables-save > /etc/iptables/rules.v4
$ systemctl enable netfilter-persistent
```

Root Cause: IPTables rules not saved to persistent configuration.

Resolution: Saved rules and enabled netfilter-persistent service.

Prevention: Automate iptables persistence with configuration management tools.