

## Why Can't We Access Our Application?

### A Simple, Real-World Guide to Ports, Firewalls, Web Servers, and Code Issues

When an application is not accessible, the first reaction is often:

"The application is down."

But in reality, the application might be running, healthy, and innocent — while something else quietly blocks access.

Let's walk through the most common real-life causes, explained simply.

#### 1. Web Server Is Inactive (Apache / Nginx / IIS)

Before an application can be accessed, the web server itself must be running.

Scenario: Server is up, but browser shows error

What You See:

- Connection refused
- Unable to connect
- Blank page

Common Causes:

- Apache/Nginx service stopped
- Web server crashed after reboot
- Web server failed to start due to config error

Troubleshooting Steps:

- Check web server status
  - Apache: systemctl status apache2
  - Nginx: systemctl status nginx
- Restart the service
- Check error logs
- Validate configuration
- Ensure web server starts on boot

Important: If the web server is down, ports 80 and 443 mean nothing.

## **2. HTTP / HTTPS Port Issues**

Ports are the entry points to your application.

HTTP → Port 80

HTTPS → Port 443

Common Causes:

- Application running on a different port
- HTTPS enabled, HTTP disabled
- SSL misconfiguration

Troubleshooting Steps:

- Check which port the app listens on
- Access using explicit port
- Verify redirects
- Confirm SSL certificate

## **3. Security Group Issues**

Common Causes:

- Port 80/443 missing in inbound rules
- Wrong source IP range
- Security group attached to wrong resource

Troubleshooting Steps:

- Allow TCP 80 and 443
- Verify security group attachment
- Confirm outbound rules

## **4. NACL Issues**

Common Causes:

- Inbound allowed, outbound blocked
- Ephemeral ports not allowed
- DENY rule overriding ALLOW

Troubleshooting Steps:

- Allow inbound 80/443
- Allow outbound 1024–65535

- Check rule priority

## 5. UFW Firewall Blocking Traffic

Common Causes:

- Port not allowed in ufw
- ufw enabled after update

Troubleshooting Steps:

- ufw status
- ufw allow 80
- ufw allow 443
- ufw reload

## 6. Application Code Missing

Common Causes:

- Code not deployed
- Wrong document root
- Deployment failure

Troubleshooting Steps:

- Verify code presence
- Check permissions
- Re-run deployment

## 7. Application Process Not Running

Common Causes:

- App crashed
- Service not started
- Missing environment variables

Troubleshooting Steps:

- Check service status
- Restart application
- Review logs

## **8. Load Balancer Issues**

Common Causes:

- Listener missing
- Wrong target port
- Health check failing

Troubleshooting Steps:

- Verify listeners
- Check target group
- Fix health checks

## **9. Internal vs External Access**

Common Causes:

- App bound to localhost
- Using private IP
- No internet gateway

Troubleshooting Steps:

- Bind to 0.0.0.0
- Use public IP/DNS
- Check routing

## **10. Golden Troubleshooting Order**

1. Is server up?
2. Is web server running?
3. Is app running?
4. Is code present?
5. Which port?
6. Is ufw open?
7. Is SG open?
8. Is NACL open?
9. Is LB configured?
10. Any recent changes?

## **Final Thoughts**

Most outages are caused by:

- Closed ports
- Inactive web servers
- Firewalls like ufw
- Missing code or stopped services

Once you know where traffic breaks, troubleshooting becomes calm and logical.