blem Fix Playbook

ALL PROBLEMS COVERED

1. ■ RAM full / swap pressure high	**2.** ■ System restart required
3. ■■ Next.js Server Action error	**4.** ■ Container is down
5. ■ Disk full	**6.** ■ Nginx is down
7. ■ Container crash loop	**8.** ■■ High CPU usage
9. ■■ Site returning 502 Bad Gateway	**10.** ■ Docker daemon is down
11. ■■ MongoDB connection error in logs	**12.** ■■ SSL certificate expired

PROBLEMS & FIX STEPS

■ CRITICAL

Problem 1 — RAM full / swap pressure high

Signs: free RAM < 100Mi · swap > 400Mi · iowait > 15%

1 **Find which container eats most RAM**

```
docker stats --no-stream
```

2 **Restart the heavy one**

```
docker restart
```

3 **Clear Docker cache**

```
docker system prune -f
```

4 **Drop Linux page cache**

```
sudo sync && sudo sh -c 'echo 3 > /proc/sys/vm/drop_caches'
```

5 **Verify RAM improved**

```
free -h
```

SIDE EFFECTS

✓ No downtime, no data loss

■ prune removes unused images only — running containers are safe

■ CRITICAL

Problem 2 — System restart required

*Signs: *** System restart required *** shown at login*

1 **Confirm all containers have restart policy**

```
docker inspect --format '{{.Name}} -> {{.HostConfig.RestartPolicy.Name}}' $(docker ps -q)
```

All must show 'always' before rebooting

2 **Reboot — do at late night Nepal time**

```
sudo reboot
```

3 **After 3 min — SSH back and verify**

```
docker ps
free -h
sudo systemctl status nginx
```

SIDE EFFECTS

✓ Swap clears, RAM feels fresh after reboot

■ ~2-3 min downtime for all sites — plan ahead

■■ WARNING Problem 3 — Next.js Server Action error

Signs: "Failed to find Server Action x" in container logs

1. **Pull latest image and redeploy**

```
docker pull seamicshq/seamics-website:latest
docker restart seamics-website-frontend
```

2. **Check logs after fix**

```
docker logs --tail=20 seamics-website-frontend
```

If error still appears — tell developer to rebuild image cleanly

SIDE EFFECTS

✓ Only website container restarts (~30 sec), others unaffected

■ If persists after redeploy — it is a code issue, not server issue

■ CRITICAL Problem 4 — A container is down

Signs: container missing from docker ps list

1. **See all containers including stopped**

```
docker ps -a
```

2. **Check why it stopped**

```
docker logs --tail=50
```

3. **Start it again**

```
docker start
```

4. **Verify**

```
docker ps
```

SIDE EFFECTS

✓ Safe, no data loss

■ Check logs to find root cause so it does not happen again

■ CRITICAL Problem 5 — Disk full

Signs: df -h shows /dev/root at 90%+ · containers may crash

1. **Check disk usage**

```
df -h
```

2. **Remove unused Docker images and containers**

```
docker system prune -af
```

3. **Check what is taking space**

```
du -sh /var/lib/docker/*
```

4. **Clear old logs**

```
sudo journalctl --vacuum-time=3d
```

5. **Verify disk freed**

```
df -h
```

SIDE EFFECTS

✓ No downtime

■ prune -af removes ALL unused images — old versions gone (can re-pull anytime)

■ CRITICAL

Problem 6 — Nginx is down

Signs: sites not loading · systemctl status nginx shows inactive/failed

1. **Check status and error**
```
sudo systemctl status nginx
sudo nginx -t
```
2. **If config is OK — restart nginx**
```
sudo systemctl restart nginx
```
3. **If config has error — check logs**
```
sudo tail -20 /var/log/nginx/error.log
```
Fix the config error shown, then restart
4. **Verify**
```
sudo systemctl status nginx
```

SIDE EFFECTS

✓ Restart takes less than 5 seconds

■ All sites are down while nginx is down — fix ASAP

■ CRITICAL

Problem 7 — Container keeps restarting (crash loop)

Signs: STATUS shows 'Restarting' in docker ps · uptime keeps resetting

1. **Check logs to find the error**
```
docker logs --tail=50
```
2. **Common causes to check in logs**
Missing env variable (e.g. MONGO_URI not set) · Port already in use · OOM killed
3. **Check if port is already taken**
```
sudo lsof -i :
```
4. **After fixing root cause — restart**
```
docker restart
```

SIDE EFFECTS

✓ Other containers not affected

■ Do not restart blindly — read logs first to find root cause

■■ WARNING

Problem 8 — High CPU usage

Signs: top shows CPU idle < 20% for a long time

1. **Find which container uses most CPU**
```
docker stats --no-stream
```
2. **Check its logs for errors or loops**
```
docker logs --tail=50
```
3. **Restart that container**
```
docker restart
```
4. **Verify CPU drops**
```
top -bn1 | head -5
```

SIDE EFFECTS

✓ Restart takes ~10 seconds

■ If CPU stays high after restart — may be a traffic spike or code bug, tell developer

■■ WARNING

Problem 9 — Site returning 502 Bad Gateway

Signs: browser shows 502 · nginx is running but site not loading

1. **Check if container for that site is running**
   ```
   docker ps
   ```
2. **If container is down — start it**
   ```
   docker start
   ```
3. **If container is up — check its logs**
   ```
   docker logs --tail=30
   ```
4. **Check nginx error log**
   ```
   sudo tail -20 /var/log/nginx/error.log
   ```

SIDE EFFECTS

✓ Starting container is safe, no data loss

■ 502 means nginx is working but backend is not — always check container first

■ CRITICAL

Problem 10 — Docker daemon is down

Signs: 'docker: command not found' or 'Cannot connect to Docker daemon'

1. **Check Docker service status**
   ```
   sudo systemctl status docker
   ```
2. **Start Docker**
   ```
   sudo systemctl start docker
   ```
3. **Start all containers**
   ```
   docker start seamics-backend seamics-website-frontend seamics-admindashboard
   seamics-tracking nivix-backend principal-backend healthcare-backend
   ```
4. **Verify all running**
   ```
   docker ps
   ```

SIDE EFFECTS

✓ No data loss

■ All sites are down until Docker and containers are back up — fix immediately

■■ WARNING

Problem 11 — MongoDB connection error in logs

Signs: MongoNetworkError or ECONNREFUSED in container logs

1. **Check which container has the error**
   ```
   docker logs --tail=30
   ```
2. **Check MongoDB Atlas dashboard**
   Go to Atlas dashboard and confirm cluster status is green
3. **Check env variables are correct**
   ```
   docker inspect | grep MONGO
   ```
4. **Restart the container**
   ```
   docker restart
   ```

SIDE EFFECTS

✓ Restart is safe

■ If Atlas cluster is down — wait for Atlas to recover, nothing to do on server side

■■ WARNING

Problem 12 — SSL certificate expired

Signs: browser shows 'Your connection is not private' · HTTPS not working

1. **Check certificate expiry**
   ```
   sudo certbot certificates
   ```
2. **Renew certificate**
   ```
   sudo certbot renew
   ```
3. **Reload nginx to apply new cert**
   ```
   sudo systemctl reload nginx
   ```
4. **Verify expiry date**
   ```
   sudo certbot certificates
   ```

SIDE EFFECTS

✓ No downtime, renewal is safe

✓ Certbot auto-renews every 90 days if set up — test with: sudo certbot renew --dry-run