线上故障应急手册

版本: v1.0
适用对象: 运维、SRE、应用 Owner、值班工程师
核心原则: **先恢复，后定位；先止血，后根治**

一、故障分级标准

级别	定义	响应时间	升级机制
P0	全站不可用 / 资损 / 数据丢失	立即响应	5min 内升级至总监级
P1	核心功能不可用 / 大面积降级	5min 内响应	15min 内升级至 Leader
P2	部分功能异常 / 性能严重下降	15min 内响应	30min 内同步值班长
P3	非核心功能异常 / 体验问题	1h 内响应	走工单流程

二、应急响应 5 步法（黄金 30 分钟）

Step 1：发现与确认（0~3 分钟）

- [ ] 接收告警（Zabbix/Prometheus/夜莺/自研监控）
- [ ] 多渠道交叉验证（监控大盘 + 用户反馈 + 日志抽样）
- [ ] **避免误判**：先看大盘全站指标，再下结论
- [ ] 在应急群内同步："【疑似 P?】现象 + 时间 + 影响面"

Step 2：定级与拉群（3~5 分钟）

- [ ] 按上述标准定级
- [ ] P0/P1 立即拉应急群（IM 群 + 电话会议）
- [ ] 角色分工：
    - **Incident Commander（指挥官）**：1 人决策，不参与具体操作
    - **Ops 处置**：执行回滚/切流/扩容
    - **研发排查**：定位代码/中间件问题
    - **沟通官**：对外 PR、对内同步进展

Step 3：止血 / 恢复（5~15 分钟）

> **第一原则：恢复服务优先，不要边查边修！**

常用止血手段优先级排序：

优先级	手段	适用场景	操作示例
P0-1	**流量切换/降级**	单机房/单集群故障	DNS 切流、LVS/Nginx 下线故障节点
P0-2	**服务回滚**	刚发布导致	`kubectl rollout undo` / 蓝绿切换
P0-3	**熔断/限流**	依赖方超时雪崩	Sentinel/Hystrix 触发降级规则
P0-4	**关闭非核心功能**	资源耗尽	关闭推荐、关闭日志采集、关闭定时任务
P0-5	**扩容**	容量不足	临时扩容 ECS/Redis 分片/数据库连接池
P0-6	**重启服务**	进程僵死/内存泄漏	滚动重启、强制 kill -9 后拉起

Step 4：根因定位（15~30 分钟）

- ☐ **5 Why 分析**：从现象倒推至少 5 层
- ☐ 查看时间线：监控告警 → 发布记录 → 流量变化 → 依赖方状态
- ☐ 工具箱：
  - **APM**：Skywalking/Cat/Pinpoint 看慢调用链路
  - **日志**：ELK/Loki 关键字检索（traceId）
  - **JVM**：jstat、jmap、arthas 在线诊断
  - **DB**：慢 SQL、锁等待、连接池
  - **网络**：tcpdump、mtr、iftop

Step 5：收尾与复盘（24h 内）

- ☐ 故障恢复后**持续观察 30 分钟**再撤离
- ☐ 1 个工作日内完成 **RCA（Root Cause Analysis）报告**
- ☐ 3 个工作日内输出**改进 Action** 并跟踪闭环
- ☐ 5 个工作日内组织**复盘会**（无指责文化，聚焦流程改进）

三、典型故障场景速查表

🔴 场景 1：服务 502/504 飙高

```
1. Nginx upstream 健康检查是否正常
2. 后端进程是否存在（ps -ef | grep java）
3. 端口是否监听（netstat -tlnp | grep port）
4. 是否有 OOM（dmesg | grep -i oom）
5. 连接池是否打满（DB/Redis 客户端监控）
```

🔴 场景 2：数据库慢查询

```
1. show processlist 看当前会话
2. information_schema.innodb_trx 看长事务
3. 慢日志定位 SQL
4. 紧急 kill 阻塞会话：kill <id>
5. 必要时切换只读副本
```

🔴 场景 3：Redis 集群故障

```
1. 哨兵/Cluster 状态：redis-cli cluster info
2. 是否触发脑裂：检查多数派
3. 内存是否打满：info memory
4. 是否有 bigkey 阻塞：redis-cli --bigkeys
5. 紧急 LRU 调整：CONFIG SET maxmemory-policy allkeys-lru
```

🔴 场景 4：K8s 节点 NotReady

```
1. kubectl describe node <node>
2. kubelet 状态：systemctl status kubelet
3. 磁盘：df -h（默认会因 disk pressure 驱逐）
4. 网络：calico/flannel 组件状态
5. 临时解封：kubectl uncordon <node>
```

🔴 场景 5：磁盘写满

```
1. du -sh /* 定位大目录
2. 清理日志：journalctl --vacuum-time=3d
3. 清理 Docker：docker system prune -a
4. 扩容云盘（生产慎用，需先备份）
```

四、沟通模板

4.1 故障公告（内部）

```
【P1 故障通报】
时间：2024-XX-XX HH:MM
现象：xxx 接口成功率下降至 XX%
影响：约 XX% 用户
当前措施：已切流至备用集群
负责人：@张三 @李四
下一同步：15 分钟后
```

4.2 用户公告（外部）

> 【服务异常通知】
> 亲爱的用户，我们当前监测到 xxx 功能出现异常，
> 技术团队正在紧急处理中，预计 XX 分钟内恢复。
> 给您带来不便，敬请谅解。

五、不可触碰的红线 🚫

1. ❌ **生产环境禁止 rm -rf /**（哪怕加通配符）
2. ❌ **禁止在排查时直接修改线上数据**（必须先备份）
3. ❌ **禁止单人操作核心数据库 DDL**（需双人复核）
4. ❌ **禁止跳过灰度直接全量发布**
5. ❌ **禁止故障期间发表定性结论**（未定位完成前只说现象）
6. ❌ **禁止隐瞒故障不上报**（哪怕只影响了 1 个用户）

六、值班与交接规范

- 值班表**提前 2 周**排定，**不接受口头换班**
- 交接必须**书面化**：遗留告警、进行中变更、风险点
- P0/P1 故障**必须通宵**直至恢复稳定
- 重大变更（发布、迁移、割接）**禁止安排在周五/节假日前一天**

七、附录：常用应急命令速查

```
# 查看系统整体状态
top -c | head -20
vmstat 1 5
iostat -x 1 5
ss -s

# 查看网络连接
ss -tan state established | wc -l    # ESTABLISH 连接数
netstat -an | grep ESTABLISHED | awk '{print $5}' | awk -F: '{print $1}' | sort | uniq -c

# 进程级定位
ps aux --sort=-%cpu | head -10
ls /proc/<pid>/fd | wc -l            # 进程打开 fd 数

# 日志实时跟踪
tail -F /var/log/messages
journalctl -u <service> -f --since "10 min ago"

# K8s 应急
```

```
kubectl rollout undo deployment/<name>
kubectl scale deployment/<name> --replicas=0  # 紧急下线
kubectl get pods -A | grep -v Running

# 抓包
tcpdump -i eth0 -nn -s 0 -w /tmp/cap.pcap host <ip> and port <port>
```

手册维护：本手册应由 SRE 团队每季度 review 一次，结合实际故障案例持续迭代。
反馈渠道：[内部 Wiki 链接] / 邮件至 sre@company.com