尽管CockroachDB有多种[保障集群可用性的设计](https://www.cockroachlabs.com/docs/stable/high-availability.html)，对于集群健康和性能的持续监控仍然是有必要的，预先针对需要调查和干预的事件创建报警规则，以便出现问题的时候能够第一时间获知并及时处理。

本小节将介绍CockroachDB提供的监控工具、以及能够报警的严重事件和指标列表。

监控工具

Admin界面

内置的Admin界面向用户提供了有关集群健康的基本指标，例如存活（live）、死亡（dead）和可疑（suspect）节点的数量，不可用Range的数量，以及每秒的查询量和集群中的服务延迟。默认情况下，可以通过集群的任意节点访问http://<host>:<http-port>或http://<host>:8080来登陆Admin界面。

**WARNING:** 考虑到Admin界面内置于CockroachDB当中，如果集群不可用，则Admin界面无法访问，因此有必要设置一些其他的手段去监控集群健康。

Prometheus端点

CockroachDB集群的每个节点都在http://<host>:<http-port>/\_status/vars中获取粒化时间序列度量标准（granular timeseries metrics）。 这些指标有特定格式，便于集成到[Prometheus](https://prometheus.io/)。后者是一种用于存储、聚合和查询时间序列数据的开源工具。该格式易于解析，也可以录入到其他第三方监控系统，例如[Sysdig](https://sysdig.atlassian.net/wiki/plugins/servlet/mobile?contentId=64946336" \l "content/view/64946336)和[Stackdriver](https://github.com/GoogleCloudPlatform/k8s-stackdriver/tree/master/prometheus-to-sd)。

参考[Monitor CockroachDB with Prometheus](https://www.cockroachlabs.com/docs/stable/monitor-cockroachdb-with-prometheus.html)

curl http:*//localhost:8080/\_status/vars*

**# HELP gossip\_infos\_received Number of received gossip Info objects**

**# TYPE gossip\_infos\_received counter**

gossip\_infos\_received 0

**# HELP sys\_cgocalls Total number of cgo calls**

**# TYPE sys\_cgocalls gauge**

sys\_cgocalls 3501

**# HELP sys\_cpu\_sys\_percent Current system cpu percentage**

**# TYPE sys\_cpu\_sys\_percent gauge**

sys\_cpu\_sys\_percent 1.098855319644276e-10

**# HELP replicas\_quiescent Number of quiesced replicas**

**# TYPE replicas\_quiescent gauge**

replicas\_quiescent{store="1"} 20

...

**NOTE:** 除了使用导出时间序列数据到外部系统来监视集群以外，用户还可以针对性地指定报警规则，以确保将可能的需要干预或调查的重大事件或问题及时通知给用户，详见[Events to Alert On](http://doc.cockroachchina.baidu.com/" \l "deploy/monitoring-and-alerting/overview/#events-to-alert-on)。

健康端点

CockroachDB提供了两个HTTP端点用于单独一个检查节点的健康状况。

/health

如果一个节点故障，则http://<host>:<http-port>/health端口返回Connnection refused错误。

curl http:*//localhost:8080/health*

curl: (7) Failed **to** connect **to** localhost port 8080: Connection refused

正常情况下则是返回HTTP200 OK状态响应码：

{

"nodeId": 1,

"address": {

"networkField": "tcp",

"addressField": "JESSEs-MBP:26257"

},

"buildInfo": {

"goVersion": "go1.9",

"tag": "v2.0-alpha.20180212-629-gf1271b232-dirty",

"time": "2018/02/21 04:09:53",

"revision": "f1271b2322a4a1060461707bdccd77b6d5a1843e",

"cgoCompiler": "4.2.1 Compatible Apple LLVM 9.0.0 (clang-900.0.39.2)",

"platform": "darwin amd64",

"distribution": "CCL",

"type": "development",

"dependencies": null

}

}

/health?ready=1

(New in v2.0) 在以下发生错误的场景访问http://<node-host>:<http-port>/health?ready=1端点，返回HTTP503 Service Unavailable状态响应码。

* 节点处于正在[退役(decommissioned)](http://doc.cockroachchina.baidu.com/" \l "maintain/remove-nodes/)或正在[关闭](https://www.cockroachlabs.com/docs/stable/stop-a-node.html)的状态，此时无法接受SQL连接和执行查询。该端口能够确保负载均衡器不会将流量路由到处于存活状态但未“就绪”的节点，尤其是在[滚动升级（rolling upgrades）](http://doc.cockroachchina.baidu.com/" \l "maintain/upgrade-cockroach-version/)的时候特别重要。

**TIPS:** 用户如果发现负载均衡器在健康检查的时候无法识别一个节点在关闭之前一段时间处于未就绪的状态，则需要提高集群配置server.shutdown.drain\_wait使得节点在执行关闭操作之前能够正确地返回503 Service Unavailable。

* 节点无法与集群中其他大多数节点通信，且可能原因是因为集群很多节点故障而导致集群不可用。

curl http://localhost:8080/health?ready=1

{

"error": "node is not ready",

"code": 14

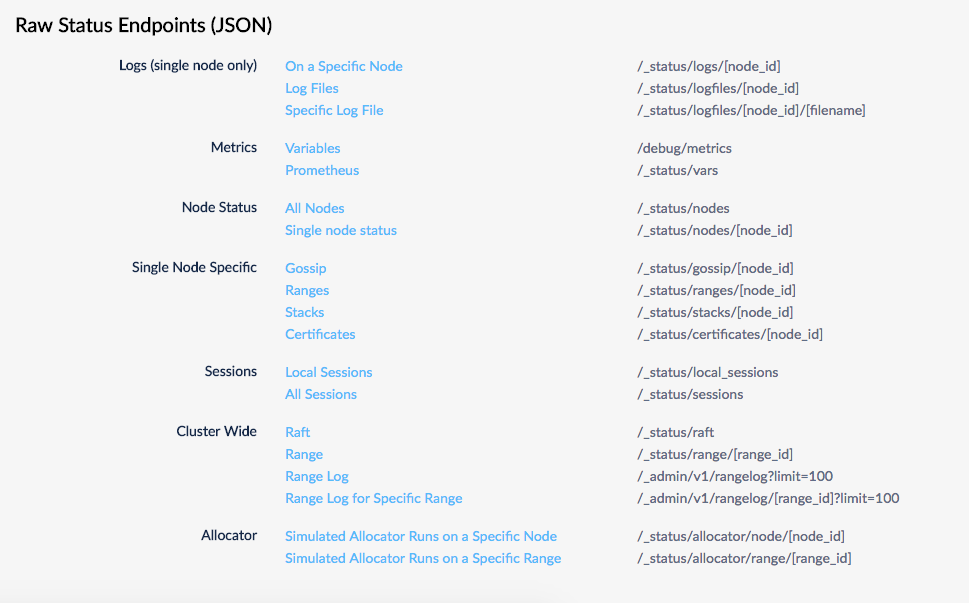
}

正常情况下，该端口返回HTTP200 OK状态响应码，内容为空：

{

}

未格式化状态端点

http://<host>:<http-port>/#/debug中可以查看到部分端点能够返回未格式化状态指标（raw status metrics），需要注意这些端点可能会随版本有所变化： 

节点状态命令

cockroach node status命令可以获取每个节点的健康状态信息。

* 使用--ranges标签，可以获取粒度range和冗余的信息，包括不可用和不可复制；
* 使用--stats标签，可以获得粒度磁盘使用细节；
* 使用--decommission标签，可以获得关于[node decommissioning](http://doc.cockroachchina.baidu.com/" \l "maintain/remove-nodes/)进程的信息；
* 使用--all标签，可以获得以上的所有信息；

可报警事件

积极的监控可以帮助用户及时发现问题。此外，通过创建报警规则使得用户在需要调查和干预的事件发生的时候能够及时地获知并处理，也是十分重要的。本小节讲展示一些重要的事件并配合Prometheus端点针对性地创建报警规则。

**TIPS:** 用户若使用Prometheus进行监控，可以在Alertmanager使用自定义的[报警规则](https://github.com/cockroachdb/cockroach/blob/master/monitoring/rules/alerts.rules.yml)，详见[Monitor CockroachDB with Prometheus](https://www.cockroachlabs.com/docs/stable/monitor-cockroachdb-with-prometheus.html)

节点故障

**规则**：节点故障超过5分钟，则报警

**检测方式**：当节点故障时，\_status/vars端点会返回Connection refused错误，正常情况下，返回内容中liveness\_livenodes指标为集群中存活节点的总数。

节点频繁重启

**规则**：节点10分钟内重启超过5次，则报警

**检测方式**：节点重启后，\_status/vars端点返回内容中的sys\_uptime指标（代表cockroach进程已经运行的时间，单位为秒）会重置为零，因此可以统计一段时间内该指标重置为零的次数。

节点硬盘空间不足

**规则**：硬盘空间不足15%，则报警

**检测方式**：通过\_status/vars端点返回内容中的capacity和capacity\_available计算得到硬盘空间使用情况。

节点不执行SQL

**规则**：节点即使有连接页不执行SQL语句，则报警

**检测方式：**通过\_status/vars端点返回内容中的sql\_conns和sql\_query\_count指标，判断前者大于零时后者都是等于0。用户也可以通过sql\_select\_count、sql\_insert\_count、sql\_update\_count、sql\_delete\_count指标细化SQL类型。

CA证书即将过期

**规则**：CA证书有效期少于1年，则报警

**检测方式**：通过\_status/vars端口返回内容中的security\_certificate\_expiration\_ca指标进行判断。

节点证书即将过期

**规则**：节点证书有效期少于1年，则报警

**检测方式**：通过\_status/vars端口返回内容中的security\_certificate\_expiration\_node指标进行判断。