

- **你的CDN中，后台更新了数据，怎么刷新？**

- 最笨的办法就是等，CDN会有缓存时间，会周期性的回源，缓存时间到了就会自动更新。

- 可以在程序的链接中加入不同版本号，CDN会认为这是一个新的链接就会去源站取了

- 还可以手动提交链接到CDN后台接口，直接刷新

- **iptables有哪些策略？**

- iptables有四表五链八策略。

- 四表是filter表，nat表，mangle表，raw表。其中mangle表和raw表不常用，filter和nat用的比较多。filter表是用来过滤数据包的，nat表是用来进行网络地址转换的。

- 五链是prerouting，input，forward，output，postrouting，所有的数据包进来时首先经过prerouting的处理，如果是转发数据包的话就再经过forward的处理，进来的数据包交给input处理，出去的数据包交给output处理，对所有的数据包做完处理后交给postrouting处理。

- 策略指的是处理动作，accept表示允许通过，drop表示拒绝通过，reject也表示拒绝通过，但是它会返回拒绝信息，dnat表示目标地址转换，snat表示源地址转换，redirect表示端口重定向，masquerand表示地址伪装，log表示记录日志。

- **http1.0和1.1、2.0的区别？**

- http实际上是从0.9版本开始的，它的功能有限，只有get，没有post，甚至头信息都没有，状态码也没有，那个时代只传输文本，所以能满足需求。

- http1.0在0.9的基础上增加了post方法，引入了header，也引入了状态码，能够传输文本以外的东西，支持了keepalive长连接，但需要参数配置

- http1.1在1.0的基础上增加了host头信息，同时默认设置了keepalive长连接

- http2.0采用了多路复用的技术，同一个连接同时并发做多个请求的处理，而1.1中一个连接只能处理一个请求。2.0还支持header头信息压缩，传输速度更快，还有服务推送等等，这个我也只是了解，现在主流用的还是http1.1版本。

- **http的各种状态码**

- 200是ok，表示一切正常，对get和post请求的应答文档跟在后面。

- 201是create，表示创建了文档，location的头给出了url。

- 300是多种选择，表示请求的文档可以在多个位置找到，返回了这些位置，需要指定location选择。

- 301是永久重定向，请求的文档在别的地方，已经永久分配了新的url，location已经给出新的url

- 302表示临时重定向，请求的文档临时改变了位置并分配了新的url。

- 304使用缓存
- 400请求出现语法错误
- 403forbidden资源不可用，一般是权限问题
- 404not found无法找到指定位置的资源
- 500是服务器发生错误，无法完成请求
- 501服务器不支持请求所需功能
- 502网关错误
- 503服务不可用，一般是数据库出现问题
- 504gateway timeout远程服务器不可用
- 505版本不支持，不支持请求中指明的http版本
- **DR模式能不能跨网段？**
- 不能跨网段，也不能跨广播域。因为dr直接路由模式是通过修改目标的mac地址来进行数据转发的，mac地址的解析需要用到arp协议，而arp协议必须在同一个广播域中才生效。
- **什么是mysql事务？**
- mysql只有innodb引擎支持事务处理，它是逻辑上的一组操作，组成这组操作的各个单位，要不全成功要不全部失败，这就是事务处理的特性。它主要用于处理操作量大，复杂度高的数据。它有四个性质也算是条件。不可分割性，一致性，独立性，持久性。不可分割性是指事务里的所有操作是一个整体，不会结束在某一个环节。一致性指的是事务开始到事务结束，数据库的完整性没有被破坏。独立性指的是数据库允许多个事务处理同时进行数据读写，而不会导致数据的不一致。持久性指的是事务处理完毕之后的修改是永久的，系统故障或者重启都不会丢失数据。
- **CDN的缺点是什么？**
- CDN可以帮助我们拦下95%甚至更多的并发请求，但是它还是有缺点的。我们在源站进行更新时，CDN节点并不会同步更新，这时候就算用户使用ctrl+F5刷新浏览器端的缓存也不会得到最新的数据。这个时候我们可以等待一个CDN缓存失效时间得到新数据，也可以在程序的链接中加入一个版本号，这样在访问时CDN就会认为这是一个新的请求，去源站拉取数据，还可以直接刷新CDN后台缓存，也可以解决这个问题。
- **iptables只允许访问8002端口访问**
- iptables -A INPUT ! -p tcp --dport 8002 -j DROP
- iptables -A INPUT -p tcp --dport 8002 -j ACCEPT + iptables -A INPUT -j DROP
- **postrouting和prerouting的区别？**

- 所有的进来的数据包都会经过prerouting，一般用来修改目标地址。所有出去的数据包都会经过postrouting，一般用来修改源地址。
- **iptables工作在第几层？**
- iptables主要工作在osi七层模型的第二数据链路层、第三网络层、第四传输层。