**自动化运维与监控类面试题**

**Ansible 是什么？**

Ansible 是一个自动化运维工具，基于 Python 开发，集合了众多运维工具的优点，可以实现 批量系统配置、批量程序部署、批量运行命令等功能。并且它是基于模块工作的，本身没有批量部署的能力，真正批量部署的是 ansible 所运行的模块，而 ansible 只是提供一种框架。

**在Ansible中解释基本术语或概念？**

控制器计算机：控制器计算机用于配置服务器, 该服务器是受管理的。这是安装Ansible的机器。

清单：清单是一个初始化文件,其中包含有关你管理的不同服务器的详细信息。

剧本：这是一个以YAML格式编写的代码文件。剧本包含需要自动化或执行的任务。

任务：每个任务代表一个需要执行的过程, 例如, 安装一个库。

模块：模块是可以执行的一组任务。 Ansible有100多个内置模块,你也可以创建自定义模块。

角色：角色是用于组织剧本和其他文件的预定义方式, 以促进共享和重用部分配置。

播放：从头到尾执行的任务或执行剧本的过程称为播放。

事实：事实是全局变量, 用于存储有关系统的详细信息, 例如网络接口或操作系统。

处理程序：处理程序用于触发服务的状态, 例如重新启动或停止服务。

**Ansible 常用模块（至少 6 个）？**

command ping yum copy service shell file replace user group

**什么是 Ansible 模块？**

模块被认为是 Ansible 的工作单元。每个模块大多是独立的，可以用标准的脚本语言编写，

如 Python、Perl、Ruby、bash 等。模块的一个重要属性是幂等性，意味着一个操作执行多 次不会产生副作用。

**ansible中shell模块和command模块以及raw的区别**

Ansible中有一个很重要的功能就是可以执行ad-hoc命令，表示即时的意思，或者说随意的意思。

ad-hoc命令中有两个模块：command, shell。

command模块不能调用shell指令，所以没有bash环境变量，也不能使用一些shell的操作方法，其他的和shell区别不大。

shell模块可以调用/bin/sh指令

raw模块： 与shell类似，但是是老版本Python的产物，比如客户端是路由器，没有Python，那就可以使用raw,其他情况下建议使用shell

**什么是 Ansible 的 playbooks ？**

Playbooks 是 Ansible 的配置、部署和编排语言，它是基于 YAML 语言编写的。他们可以描述您希望远程系统实施的策略，或者描述一般 IT 流程中的一系列步骤。

**描述 Ansible 是如何工作的？**

Ansible 由节点和控制机器组成。控制机器是安装 Ansible 的地方，节点由这些机器通过 SSH 管理。 借助 SSH 协议，控制机器可以部署临时存储在远程节点上的模块。

控制机器使用 ansible 或者 ansible-playbooks 在服务器终端输入的 Ansible 命令集或者 playbook 后，Ansible 会遵循预先编排的规则将 PLAYbook 逐条拆解为 Play，再将 Play 组织成 Ansible 可以识别的任务 tasks，随后调用任务涉及到的所有 MODULES 及 PLUGINS，根据主机 清单 INVENTORY 中定义的主机列表通过 SSH 协议将任务集以临时文件或者命令的形式传输 到远程节点并返回结果，如果是临时文件则执行完毕后自动删除。

**简述Ansible变量？**

Ansible 利用变量存储整个 Ansible 项目文件中可重复使用的值，从而可以简化项目的创建和维护，并减少错误的发生率。在定义Ansible变量时，通常有如下三种范围的变量：

global范围：从命令行或Ansible配置中设置的变量；

play范围：在 play 和相关结构中设置的变量；

host范围：inventory、facts 或 register 的变量，在主机组和个别主机上设置的变量。

**简述Ansible hanlder？**

Ansible模块被设计成幂等的，即在一个适当编写的剧本中，剧本及其任务可以在不更改受管主机的情况下多次运行，除非它们需要进行更改以使受管主机达到所需的状态。

然而，有时当一个任务对系统进行了更改后同时需要运行另一个任务。例如，对服务的配置文件的更改可能需要重新加载服务，以便更改后的配置生效。此时就需要使用hanlder程序。handler程序是响应由其他任务组成的通知的任务。每个handler程序都有一个全局惟一的名称，并在剧本中任务块的末尾触发。

如果没有任务通过名称调用handler程序，它将不会运行。

如果一个或多个任务都调用handler程序，它将在剧中的所有其他任务完成后仅运行一次。

因为handler程序是任务，所以可以在handler程序中使用与处理任何其他任务相同的模块。通常，handler程序用于重新启动主机和重新启动服务。

handler程序可以视为非活动任务，只有在使用notify语句显式调用时才会触发这些任务。

**什么是Ansible?它的优势是什么?**

答案：Ansible是一种开源的自动化工具，用于IT基础设施的配置管理、应用程序的部署、云环境的管理等任务。它的优势在于使用简单、易于学习、跨平台、扩展性强。能够实现批量系统配置、批量程序部署、批量运行命令等功能。减少运维人员工作量。

**Ansible的基本组件是什么?**

答案：Ansible的基本组件包括:控制节点、主机、变量、任务、角色、模块等

**描述Ansible是如何工作的?**

答案：Ansible由节点和控制机器组成。 控制机器是安装Ansibles的地方，节点由这些机器通过SSH管理。  
借助SSH协议，控制机器可以部署临时存储在远程节点上的模块。控制机器使用ansible或者ansible-playbooks在服务器终端输入的Ansible命令集或者playbook后Ansible会遵循预先编排的规则将PLAYbook逐条拆解为Play，再将Play组织成Ansible可以识别的任务tasks，随后调用任务涉及到的所有MODULES及PLUGINS，根据主机清单INVENTORY中定义的主机列表通过SSH协议将任务集以临时文件或者命令的形式传输到远程节点并返回结果，如果是临时文件则执行完毕后自动删除。

**如何在Ansible中定义变量?**

答案：在Ansible中，可以通过在yaml文件中定义变量来使用。变量的定义格式为:变量名:变量值。

**如何在Ansible中定义任务?**

答案：在Ansible中，可以通过在yaml文件中定义任务来执行操作。任务的定义格式为:-name:任务名，action:操作其它参数。

**Ansible中的模块是什么?**

答案：Ansible模块是一些封装好的功能单元，用于执行各种操作，比如文件操作、用户管理、软件安装等

**Ansible的角色是什么?**

答案：Ansible的角色是一种可重用的组件，用于管理主机的配置，可以包含任务、变量、处理器等

**如何在Ansible中执行shell命令?**

答案：在Ansible中，可以使用shell模块来执行命令，也可以使用command模块来执行命令

**如何在Ansible中安装软件包?**

答案：在Ansible中，可以使用yum、apt等模块来安装软件包

**如何在Ansible中管理用户?**

答案：在Ansible中，可以使用user模块来管理用户，包括创建用户、修改密码、删除用户等

**如何获取ansible 模块的帮助？**

答案：通过ansible-doc 后接模块名查看模块的帮助

**现有100台服务器，你如何管理？**

答案：通过ansible批量管理100台服务器  
要使用Ansible管理100台服务器，您需要做以下几步配置：  
确保Ansible控制节点安装并配置好Ansible：  
Yum -y install ansible  
创建Ansible inventory文件：通常是一个名为/etc/ansible/hosts的文件，或者可以使用一个单独的文件来组织服务器组。  
在inventory文件中列出所有服务器,如服务器ip：  
[webservers]  
Server ip  
Server ip  
...  
Server ip  
确保服务器的SSH访问是通过密钥认证而非密码：  
在控制节点生成SSH密钥对（如果还没有的话）。  
将控制节点的公钥复制到所有100台服务器的~/.ssh/authorized\_keys文件中。  
测试连接  
Ansible all -m ping

**Ansible 控制节点认证有几种？**

答案：两种  
① 主机认证登录  
② 公私钥匙认证  
控制节点生成密钥对  
# ssh-keygen  
拷贝公钥匙到被控制节点  
# ssh-copy-id 服务器ip

**什么是 Ansible Inventory？**

答案：Ansible 在/etc/ansible/hosts上有一个默认文件，其中包含目标系统的 IP 地址或主机名。您可以使用命令行上的-i <path>选项指定不同的清单文件。Ansible 从这个文件中选择一部分主机来处理。文件格式可以是 INI 或 YAML，默认类似于 INI。在 INI 格式中，括号中的标题是组名，用于对系统进行分类并决定您在什么时间和出于什么目的控制哪些系统。

-------------模块类面试题--------

**shell模块的功能？**

答案：shell模块在远程主机上调用shell解释器运行命令，支持shell的各种功能，例如管道等。

**copy模块的功能？**

答案：复制文件到远程主机，可以个修改权限等。

**Setup模块功能？**

答案：查看远程主机的基本信息  
收集可用的facts,收集每个节点的相关信息, 如: 架构信息，IP,时间，域名，网卡，MAC，主机名，CPU等信息。

**Mount模块功能？**

答案：Mount 远程挂载

**Yum模块功能？**

答案：使用yum管理软件包，安装软件包和删除软件包

**Service模块功能？**

答案：对远程主机进行服务管理，服务启动、停止、重启、重载

**File模块功能？**

答案：设置文件、符号链接和目录的属性，或删除文件、符号链接、目录

**User模块功能？**

答案：对远程主机的用户管理， 用户创建 删除

**Group模块功能？**

答案：对远和主机的用户组管理，创建组 删除组

**Cron模块功能？**

答案：对远程主机设置计划任务，相当于在远程主机上编写crontab

**script模块功能？**

答案：将控制节点脚本传输到远程执行

**Geturl模块功能？**

答案：Get\_url 将文件或软件从http、https或ftp下载到本地节点上

**unarchive模块功能？**

答案：传输打包好的文件夹,在远程节点上自动解压  
特点：对于压缩文件，不仅仅是将压缩文件传输过去，而是直接将解压后的目录文件呈现出来。

----------------监控模块--------------

**Zabbix监控的优缺点有哪些**

答案：优点：开源、无软件成本、server端对设备性能要求低、支持的设备多，自带多种监控模板、能实现自动化监控、开放式接口，扩展性强。缺点:需在被监控主机上安装agent， 所有数据都存在数据库里，产生的数据量很大，瓶颈主要在数据库。UI图形展示（grafana）

**讲讲你对监控的理解，监控的目的是什么**

答案：保证服务器系统的安全运营；并提供灵活的通知机制以让系统管理员快速定位/解决存在的各种问题。

**Zabbix监控模式有几种、有什么区别**

答案：主动模式：Agent向Server发起连接，主动模式可以大大减少zabbix server的压力。被动模式：Server向Agent发起连接，被动模式zabbix server的压力大。

**Zabbix自动发现怎么做的？**

创建自动发现规则,设置键值,填写类型和zabbix端口,发现后的动作可以关联模板,发送邮件

**Zabbix分布式监控有什么优点？**

答案：缓解服务器压力，提高可用性和灵活性，减少网络负载，提高监控系统的性能和效率，更好的处理大量的监控数据。

**zabbix的分布式监控有什么作用？**

答案：部署Proxy是可选的，，但是可能非常有益于分散单个Zabbix sever的负载。 如果只有proxy收集数据，sever上的进程就会减少CPU消耗和磁盘I / O负载。Zabbix proxy是完成远程区域、分支机构、没有本地管理员的网络的集中监控的理想解决方案。

**zabbix监控遇到过什么问题**

答案：1）监控项不可用: 这通常是由于Zabbix代理未正确安装或配置所导致的。要解决此问题，您可以检查代理程序是否正在运行以及其配置文件是否正确设置。2）数据不准确: 如果您发现Zabbix监控数据不准确，请首先检查监控项是否正确配置。例如，您是否使用了正确的关键字、单位和阈值等。另外，还可以检查Zabbix服务器和被监控主机之间的网络连接是否不稳定3）无法连接到被监控主机: 如果您无法连接到被监控主机，则可能是由于防火墙或网络配置问题所致。您可以通过检查防火墙规则或调整网络配置来解决此问题

**zabbix的缺点是什么？**

如何解决性能问题: 在监控大规模复杂系统时，Zabbix的性能可能会受到影响，特别是在处理大量数据时，可能会导致系统响应变慢或者出现延迟解决方案:可以通过优化数据库、增加硬件资源、减少监控项等方法来提高Zabbix的性能UI设计不够友好: Zabbix的ul设计相对来说不够友好，使用起来可能会比较困难，特别是对于非技术人员来说解决方案: 可以考虑使用Zabbix的第三方插件或者自定义脚本来定制化UI界面，以满足不同用户的需求。

**zabbix结合grafana怎么做？**

答案：首先需要在服务器上安装Grafana和Zabbix，在数据源中填写Zabbix的APIURL、用户名和密码等信 息。使用Grafana的面板编辑功能，创建需要监控的指标和图表，比如CPU利用率、内存使用率、网络流量等等。

**Zabbix电话报警怎么做？**

答案：首先需要注册一个睿象云的账号，在睿象云设置分配策略和通知测率，配置Zabbix媒介类型，创建Zabbix报警动作，最后进行测试。首先需要选用一个报警平台,我们公司选用的是睿象云,可以在平台上面选择zabbix监控工具,然后按照提示记录appkey以及下载Cloud Alert Agent,完成安装后便可在zabbix上面会自动生成一个的动作媒介用户以及群组,只需要创建动作添加相应的触发器即可

**你们公司都接收到什么报警消息，如何处理的？**

答案：服务器报警: 当服务器的CPU利用率、内存使用率、磁盘空间等超过阈值时，会触发服务器报警。此时，我们会通过邮件或者微信等方式通知运维人员，并尽快处理问题，以避免对业务的影响。（CPU太高，内存太高，挖矿病毒）应用程序报警:当应用程序出现异常、访问量过大或者请求超时等问题时，会触发应用程序报警。我们会通过邮件或者钉钉等方式通知开发人员或者运维人员，协调解决问题，以确保业务的正常运行。（MySQL主从失效）（Tomcat堆栈吃内存）网络报警:当网络出现故障、流量异常、连接数过大等问题时，会触发网络报警。我们会通过邮件或者微信等方式通知网络运维人员，协调解决问题，以保证网络的稳定和可靠格 （nginx并发量）

**使用zabbix都监控过什么？**

答案：硬件监控：通过 SNMP 来进行路由器交换机的监控。系统监控：如 CPU 的负载，上下文切换、内存使用率、磁盘读写、磁盘使用率、磁盘 inode 节点。服务监控：比如公司用 LNMP nginx 自带 Status 模块、 PHP 也有相关的 Status 、 MySQL 的话可以通过 ODBC协议 来进行监控。网络监控：如果是云主机又不是跨机房，那么可以选择不监控网络。安全监控：如果是云主机可以考虑使用自带的安全防护。当然也可以Zabbix监控 iptables 。如果是硬件，那么推荐以Zabbix监控硬件防火墙。Web 监控：web 监控的话题其实还是很多。比如可以使用自带的 web 监控来监控页面相关的延迟、 js 响应时间、下载时间、等等。日志监控：如果是 web 的话可以使用监控 Nginx 的 500x 日志。PHP 的 ERROR 日志。流量分析：平时我们分析日志都是拿 awk sed xxx 一堆工具来实现。这样对我们统计 ip 、 pv 、 uv 不是很方便。那么可以使用百度统计、 google 统计、商业，让开发嵌入代码即可 （不想关建议去掉）。可视化：通过 screen 以及引入一 些第三方的库来美化界面，同时我们也需要知道、订单量突然增加、 突然减少。或者说突然来了一大波流量，这流量从哪儿来，是不是推广了，还是被攻击了。可以结合监控平台来梳理各个系统之间的业务关系。自动化监控：如上我们做了那么多的工作，当然不能是一台一台的来加 key 实现。可以通过Zabbix 的主动模式以及被动模式来实现。当然最好还是通过 API 来实现。

**Zabbix遇到过什么问题**

答案：问题1：主机有30多个图形，但是查看时只显示了20个图形。原因：是因为zabbix的php默认值为20。解决：修改zabbix前端默认配置文件 defines.inc.php中的 ZBX\_MAX\_GRAPHS\_P ER\_PAGE变量值，重启服务即可。问题2: 内存溢出导致zabbix\_server服务关闭？解决：修改/etc/zabbix/zabbix\_server.conf配置文件里添加CacheSize=1024M ，重启服务问题3：Zabbix给新机器添加监控，按正常操作完成后，发现主机那一栏最后的灯不亮解决：检查防火墙和SELinux是否关闭，查zabbix\_server日志，查zabbox是否启动，来回检查了好几遍，没发现问题，删除后重新加了两遍，还是灯不亮，后来在网上找相关的解决方法，试了发现不是自己遇到的问题，等过了一段时间发现灯正常亮了。

Zabbix是怎么企业微信提醒报警的 （钉钉，QQ邮箱，163邮箱，短信都可以）

答案：1）首先，需要有一个钉钉号。（一个实名认证的[微信号]一个可以使用的[手机号]一个可以登录的[邮箱号]2）下载并配置webhook接口，在网上找一个脚本运行连接即可3）配置Zabbix告警，（增加示警媒介类型，添加用户报警媒介，添加报警动作）。

**zabbix监控mysql的四大性能指标？**

答案：1）查询吞吐量2）查询执行性能3）连接情况4）缓冲池使用情况

**如果发现监控没有数据，请排查如下问题：**

答案：1）zabbix客户端是否重启1）脚本是否有执行权限2）数据库是否有用户授权3）环境变量是否有问题

**Zabbix监控了多少客户端，客户端是怎么进行批量安装的？根据实际公司台数回答。**

答案：使用命令生成密钥。将公钥发送到所有安装zabbix客户端的主机。安装 ansible 软件，（修改配置文件，将zabbix 客户机添加进组）。创建一个安装zabbix客户端的脚本。执行该脚本。验证。

**Zabbix怎么开启自定义监控？**

答案：写一个脚本用于获取待监控服务的一些状态信息。在zabbix客户端的配置文件zabbix\_agentd.conf中添加上自定义的“UserParameter”，目的是方便zabbix调用我们上面写的那个脚本去获取待监控服务的信息。在zabbix服务端使用zabbix\_get测试是否能够通过第二步定义的参数去获取zabbix客户端收集的数据。在zabbix服务端的web界面中新建模板，同时第一步的脚本能够获取什么信息就添加上什么监控项，“键值”设置成前面配置的“UserParameter”的值。数据显示图表，直接新建图形并选择上一步的监控项来生成动态图表即可。

**zabbix如何修改其中监控的一台服务器中内存阈值信息,比如正常内存使用到了 80%报警,我想修改为 60%报警?**

答案：正常来说 , 一般会把监控的服务器统一加入到一个模板中 , 修改模板的其中某一项的监控项参数和告警阈值后 , 加入模板中的所有主机都会同步。如果单独想修改其中某一台服务器内存告警阈值 ,需要模板中设定阈值使用用户自定义宏代替，这样可以在单独修改某一台主机的宏，达到单个主机内存报警，阈值修改为60%。

**讲解一下告警屏蔽？**

答案：- 处于维护中的主机，告警信息还是会报出来，但是不会触发动作；- 对于可预知的临时维护或紧急维护，可以提前设置告警屏蔽；- 可针对组、单台或多台主机，进行一次性、每日定时、每月定时、每周定时屏蔽；- 前端可以将维护中的主机不进行显示，Dashboard右上角有个图标可以进行筛选是否显示维护中的主机；- 告警屏蔽时，可以设置采集数据或不采集数据两种方式

**报警类型和故障处理：**

答案：报警类型：1)E-mail 邮件2)微信报警等故障处理：如监控服务器down机不报警1)先定位问题：zabbix服务器采用轮询的方式从监控目标机的agent获取数据。但如果目标机down了，目标机的agent通常也无法工作，所以检查被监控的主机服务2)增加一个目标主机进行是否存活的检测。

**简述配置zabbix自定义监控流程?**

答案：配置Zabbix自定义监控的流程通常包括以下步骤：1）确定监控需求：首先，你需要明确你的监控需求，即你想要监控什么内容。这可能包括服务器性能、网络状态、应用程序行为等。2）编写自定义监控脚本：Zabbix通过执行脚本或命令来获取监控数据。因此，你需要编写自定义的监控脚本，以获取你需要的监控数据。这些脚本应该能够返回Zabbix可以解析的数据格式，通常是文本或JSON格式。3）创建监控项：在Zabbix的Web界面中，你需要创建新的监控项。在创建监控项时，你需要指定监控项的名称、类型（如Zabbix Agent、SNMP等）、键值（Key）等。键值是你自定义的，用于标识这个监控项，并且与你在监控脚本中使用的名称相对应。4）配置触发器：触发器是定义当监控项的值达到某个条件时，执行特定动作（如发送警报）的规则。你需要根据监控需求配置相应的触发器。5） 配置动作：当触发器被触发时，Zabbix会执行预定义的动作。你可以配置动作来发送电子邮件、短信、推送消息等通知。此外，还可以配置其他类型的动作，如执行远程命令、更新主机状态等。6）测试与验证：完成上述配置后，你需要测试自定义监控是否按预期工作。这可以通过模拟触发条件并检查Zabbix是否发送了正确的警报或执行了相应的动作来完成。7）优化与调整：根据测试结果和实际需求，你可能需要对监控脚本、监控项、触发器和动作进行优化和调整，以确保监控的准确性和有效性。

**简述zabbix的服务架构?**

答案：Zabbix的服务架构主要包括几个核心组件，这些组件协同工作以提供全面的监控解决方案。以下是Zabbix服务架构的简述：1）Zabbix Server- Zabbix Server是监控系统的核心组件，它负责接收来自Agent的数据、处理这些数据、触发告警以及执行预定义的动作。- Server通过SNMP、Zabbix Agent、ping、端口监视等方法收集远程服务器或网络状态的数据。- Server可以运行在多种平台之上，如Linux、Solaris、HP-UX等。2）Zabbix Agent- Zabbix Agent是一个部署在被监控设备上的组件，负责收集设备的数据并发送给Zabbix Server。- Agent能够收集各种系统指标，如CPU使用率、内存使用情况、磁盘I/O等，并将这些数据发送到Server进行进一步处理。- Agent的部署是可选的，但对于某些监控需求来说，使用Agent可以提供更详细和准确的数据。3）Zabbix Proxy- Zabbix Proxy是一个可选的组件，用于在大规模环境中分担Zabbix Server的负载。- Proxy从被监控设备收集数据，并将这些数据暂时存放在本地，然后统一转发给Zabbix Server。- 通过部署Proxy，可以减少Server与Agent之间的直接通信量，提高整个监控系统的性能和可扩展性。

**请简述Zabbix模板的作用?**

答案：1）定义监控项和触发器的集合：模板允许用户将一组相关的监控项和触发器打包在一起，形成一个可复用的集合。这些监控项和触发器可能涉及主机性能、网络状态、应用程序行为等多个方面。2）简化配置和监控过程：通过使用模板，用户可以更加高效地配置和管理监控项，而不需要一个个手动添加。这大大简化了监控系统的管理和维护工作，提高了工作效率。3）统一管理：模板提供了一种统一的方式来管理一类设备的监控项和触发器。当需要对监控项进行修改时，只需要修改模板即可，所有应用了该模板的主机都会同步更新，从而实现了监控配置的一致性。4）降低出错概率：通过使用模板，可以减少手动配置带来的错误，提高了监控系统的准确性和稳定性。模板中的监控项和触发器已经经过预先定义和测试，因此可以确保配置的正确性。5）提高监控效率和可靠性：模板的使用可以使监控配置更加方便和灵活，通过继承模板来快速部署监控配置。这不仅可以节省用户在配置监控时的时间和精力，还可以提高监控配置的效率和可靠性。总的来说，Zabbix模板是监控系统中的重要工具，它简化了配置和管理过程，提高了监控的一致性和效率，降低了出错概率，为用户提供了一个强大而灵活的监控解决方案。

**实现Zabbix报警功能的规则简述?**

答案：1）定义监控项：首先，你需要确定你想要监控的内容和指标。这可以包括服务器性能、网络状态、应用程序行为等各种方面。对于每个监控项，你需要定义其数据收集的方式和频率。2）创建触发器：触发器是报警功能的核心，它定义了何时触发报警的条件。你需要为每个监控项创建一个或多个触发器，这些触发器将基于监控项的数据值来判断是否满足报警条件。例如，你可以设置一个触发器，当CPU使用率超过80%时触发报警。3）配置动作：当触发器被触发时，你需要定义执行的动作。这些动作可以包括发送电子邮件、短信通知、执行远程命令等。你可以根据实际需求配置相应的动作，以确保在触发报警时能够及时通知相关人员并采取相应的措施。4）测试与验证：完成上述配置后，你需要进行测试以确保报警功能正常工作。你可以模拟触发条件，检查是否成功接收到了报警通知，并验证执行的动作是否符合预期。在配置报警功能时，还需要注意以下几点：- 报警升级机制：Zabbix支持独特的报警升级机制，可以根据不同的时间阶段和条件，将报警信息发送给不同的用户。这样可以根据紧急程度和重要性来定制报警策略。- 用户权限控制：在Zabbix系统中，你可以设置用户权限来控制谁可以接收报警通知以及执行相应的动作。这有助于确保报警信息的准确性和安全性。综上所述，实现Zabbix报警功能的规则包括定义监控项、创建触发器、配置动作以及进行测试与验证。通过合理配置这些规则，你可以确保在发生异常或问题时能够及时接收报警通知并采取相应措施。

**什么是Prometheus**

答案：Prometheus是一款开源的监控系统，由SoundCloud开发，用于收集、存储和查询时间序列数据，其数据模型是基于键-值对的时序数据。

**Prometheus的工作流程**

答案：Prometheus server 定期从配置的targets中拉取metrics数据，并存储在本地的时间序列数据库中。Prometheus会根据用户定义的规则对这些时间序列进行计算、聚合，并触发警报通知用户。Prometheus收集的数据回在其内置的web UI中或grafana中提供可视化功能。

**Prometheus的组件有哪些**

答案：Prometheus Server：Prometheus服务器Client Libraries：检测应用程序代码的客户端库Push Gateway：用于支持短期临时或批量计划任务工作的数据汇聚节点Alertmanager：用于告警Exporter：收集监控样本数据

**Prometheus有什么特点**

答案：通过PromQL实现多维度数据模型的灵活查询。定义了开放指标数据的标准，自定义探针（如Exporter等），编写简单方便。PushGateway组件让这款监控系统可以接收监控数据。提供了VM和容器化的版本。

**Metric的几种类型？分别是什么？**

答案：Counter：其值从0开始只能增加，不会减少；重启进程后，会被重置；适用于跟踪事件的数量或大小，可以用来记录服务的请求次数、任务完成数、错误发生次数。Gauge：是一种常规指标，反映系统当前状态的快照，是对一个值的瞬时测量。Histogram：用于表示在一段时间范围内对数据进行采样，对指定区间以及总数进行分组统计。Summary：常用于跟踪与时间相关的数据。

**Prometheus的优点和缺点**

答案：优点：支持多语言客户端；支持本地和第三方远程存储，单机性能强劲；高效存储，平均一个采样数据占用3.5B左右；可扩展性强；精确告警；支持静态文件配置和动态发现等自动发现机制等缺点：不适合存储事件或者日志；不会针对大量的历史数据进行存储等

**Prometheus怎么采集数据**

答案：Prometheus调用HTTP GET请求或metrics数据接口获取数据。

**Prometheus怎么获取采集对象**

答案：通过静态和动态两种方式；静态配置是一种传统的服务发现方式，适用于固定的监控环境、IP地址和统一服务接口场景；动态发现方式适用于云环境使用。

**Prometheus的配置文件组成**

答案：Prometheus配置文件由以下几部分组成：Global：用于定义全局变量和设置相关参数Rule：用于定义报警规则和报警条件Alertmanager：用于定义报警器的配置信息Scrape：用于定义收集数据的配置信息

**Prometheus怎么监控k8s**

答案：通过kubernetes API来监控k8s集群，可以监控k8s组件和应用程序运行状态，收集k8s集群指标存储在prometheus服务器中。

**alertmanager是什么**

答案：Alertmanager是一款开源的告警管理工具，可以聚合来自Prometheus服务器的告警，并将其发送到用户指定的接收者。

**alertmanager处理告警的流程**

答案：从源接收告警然后将其聚合为单个通知，接着对告警进行路由，将其发送到配置的通知目标，还可以对告警进行处理，例如抑制和延迟。

**exporter的类型**

答案：直接采集型：这类Exporter直接内置了相应的应用程序，用于向Prometheus直接提供Target数据支持，例如cAdvisor、Kubernetes等间接采集型：原始监控目标并不直接支持Prometheus，需要我们使用Prometheus提供的Client Library编写该监控目标的监控采集程序，如：Node exporter

**什么是白盒监控**

答案：需要把对应的Exporter程序安装到被监控的目标主机上，从而实现对主机各种资源及其状态的数据采集工作的方式叫做白盒监控。

**什么是黑盒监控**

答案：无法直接通过安装exporter监控到目标，Blackbox Exporter无须安装在被监控的目标环境中，用户只需要将其安装在与Prometheus和被监控目标互通的环境中，通过HTTP、HTTPS、DNS、TCP、ICMP等方式对网络进行探测监控，还可以探测SSL证书过期时间。

**pushgateway是什么**

答案：Pushgateway是一个独立的服务组件，可在HTTP REST API上接收Prometheus指标，位于发送指标的应用服务程序和Prometheus服务器之间。Pushgateway接收指标，然后将其作为目标进行抓取，以便将指标提供给Prometheus服务器。

**pushgateway能做什么**

答案：用于处理Prometheus无法直接去拉取各个目标的监控数据，目标资源的生命周期太短，目标资源没有可以抓取的端点等情况

**pushgateway有什么缺点**

答案：单点故障，可能出现性能瓶颈会丧失Prometheus通过UP监控指标检查实例健康状况的功能Pushgateway会对推送给它的所有监控数据进行持久化，即使被监控服务已经下线销毁了，Prometheus还是会获取到过期的历史监控数据

**简述Prometheus使用场景?**

答案：Prometheus是一个开源的监控系统和时序数据库，适用于各种不同的场景，包括但不限于：1）云原生环境：Prometheus 可以在云原生环境中轻松部署和管理，支持 Kubernetes 等容器化平台。2）微服务架构：Prometheus 可以对分布式架构中的各种微服务进行监控，帮助用户了解整个系统的运行状态。3）系统监控：Prometheus 可以监控服务器的 CPU、内存、磁盘等资源使用情况，及时发现系统中的问题。4）应用监控：Prometheus 可以监控应用程序的性能指标，帮助用户优化应用程序的性能。5）网络监控：Prometheus 可以监控网络设备和流量数据，帮助用户分析网络性能和故障原因

**Prometheus 数据流向**

答案：1）Prometheus server 定期从配置好的 jobs 或者 exporters 中拉取 metrics，或者接收来自 Pushgateway 发送过来的metrics，或者从其它的Prometheus server中拉取 metrics。2）Prometheus server在本地存储收集到的 metrics，并运行定义好的 alerts.rules，记录新的时间序列或者向Alert manager推送警报。3）Alertmanager 根据配置文件，对接收到的警报进行处理，发出告警。4）在图形界面中，可视化采集数据。

**Zabbix和Prometheus区别**

总结：如果监控的是物理机，优先选 Zabbix，Zabbix在传统监控系统中，尤其是在服务器相关监控方面，占据绝对优势，甚至环境变动不会很频繁的情况下，Zabbix 也会比 Prometheus 好使；但如果是云环境的话，除非是 Zabbix 玩的非常溜，可以做各种定制，否则还是选择 Prometheus 。Prometheus开始成为主导及容器监控方面的标配，并且在未来可见的时间内被广泛应用。

**prometheus工作模式**

答案：1）Prometheus Server 基于服务发现（Service Discovery）机制或静态配置获取要监视的目标（Target），并通过每个目标上的指标 exporter来采集（Scrape）指标数据；2）Prometheus Server 内置了一个基于文件的时间序列存储来持久存储指标数据，用户可使用PromQL接口来检索数据，也能够按需将告警需求发往Altermanager完成告警内容发送；3）一些短期运行的作业的生命周期过短，难以有效地将必要的指标数据供给到Server端，它们一般会采用推送（Push）方式输出指标数据，Prometheus借助于Pushgateway 接收这些推送的数据，进而由server端进行抓取

**prometheus工作流程**

答案：1）Prometheus server可定期从活跃的（up）目标主机上（target）拉取监控指标数据，目标主机的监控数据可通过配置静态job或者服务发现的方式被prometheus server采集到，这种方式默认的pull方式拉取指标；也可通过pushgateway把采集的数据上报到prometheus server中；还可通过一些组件自带的exporter采集相应组件的数据；2）Prometheus server把采集到的监控指标数据保存到本地磁盘或者数据库；3）Prometheus采集的监控指标数据按时间序列存储，通过配置报警规则，把触发的报警发送到alertmanager；4）Alertmanager通过配置报警接收方，发送报警到邮件，微信或者钉钉等5）Prometheus 自带的web ui界面提供PromQL查询语言，可查询监控数据6）Grafana可接入prometheus数据源，把监控数据以图形化形式展示出

**简述Prometheus的四种数据类型?**

答案：- 瞬时向量（Instant vector）- 区间向量（Range vector）- 标量数据 (Scalar): 一个浮点数- 字符串 (String): 一个字符串，暂时未用

**prometheus存储数据的原理是什么**

答案：Prometheus是一个开源的监控系统，用于收集、存储和查询时间序列数据。它的存储数据的原理是基于一种称为TSDB（Time Series Database）的数据库引擎，该引擎专门用于存储时间序列数据。在Prometheus中，时间序列数据以键值对的形式存储，其中键是一个标识符，通常表示监控指标的名称以及一组标签，用于区分不同实例或维度。值则是时间戳和相应的测量值。Prometheus使用内部的时间序列数据库来存储这些键值对数据。在存储数据时，Prometheus会将数据按照时间戳进行排序并进行压缩，以便节省存储空间。此外，Prometheus还支持数据的分片存储和自动数据清理，以确保系统的性能和稳定性。通过存储时间序列数据并提供查询和分析功能，Prometheus可以帮助用户监控系统的性能指标、警报和分析数据。其存储数据的原理和设计使得Prometheus在处理时间序列数据时具有高效性能和可扩展性

**简述什么是PromQL?**

答案：PromQL（Prometheus Query Language）是 Prometheus 内置的数据查询语言，它能实现对事件序列数据的查询、聚合、逻辑运算等。它并且被广泛应用在 Prometheus 的日常应用当中，包括对数据查询、可视化、告警处理当中。简单地说，PromQL 广泛存在于以 Prometheus 为核心的监控体系中。所以需要用到数据筛选的地方，就会用到PromQL。例如：监控指标的设置、报警指标的设置等等。