

资料整理自网络，仅作免费交流分享，侵权删！

更多面试资料请关注微信公众号：程序员面试吧



简述Ansible及其优势？

Ansible 是一款极其简单的开源的自动化运维工具，基于Python开发，集合了众多运维工具(puppet, cfengine, chef, func, fabric)的优点。实现了批量系统配置，批量程序部署，批量运行命令等功能。同时 Ansible是基于模块工作，其实现批量部署的是ansible所运行的模块。

Ansible其他重要的优势：

- 跨平台支持：Ansible在物理、虚拟、云和容器环境中为Linux、Windows、UNIX和网络设备提供无代理支持。
- 人类可读的自动化：Ansible playbook以YAML文本文件的形式编写，易于阅读，有助于确保每个人都理解他们将要做的事情。
- 对应用程序的完美描述：Ansible playbook可以进行任何更改，并且可以描述和记录应用程序环境的每个细节。
- 易于管理的版本控制：Ansible剧本和项目是纯文本。它们可以像源代码一样处理，并放在现有的版本控制系统中。
- 支持动态库存：Ansible管理的机器列表可以从外部资源动态更新，以便随时捕获所有受管服务器的正确的当前列表，无论基础设施或位置如何。
- 易于与其他系统集成的编排：HP SA、Puppet、Jenkins、Red Hat Satellite，以及存在于环境中的其他系统，都可以被利用并集成到Ansible工作流中。

简述Ansible工作机制及其特性？

Ansible是一款自动化运维工具，基于Python开发，具有批量系统配置, 批量程序部署, 批量运行命令等功能。

其工作机制如下：

- 1、用户使用Ansible或Playbook，在服务器中断输入Ansible的Ad-Hoc命令集或Playbook；
- 2、Ansible遵循预先编排的规则将Playbooks逐条拆解为Play；
- 3、Play组织成Ansible可识别的任务（Task）；
- 4、Task会调用任务所涉及的所有模块（Module）和插件（Plugin）；
- 5、读取Inventory中定义的主机列表；

- 6、通过SSH认证（默认）将任务集以临时文件或命令的形式传输到远程客户端执行并返回执行结果。其特性如下：
- 7、no agents：不需要在被管控主机上安装任何客户端，只需SSH、Python即可，建议Python版本为2.6.6以上；
- 8、no server：无服务器端，使用时直接运行命令即可；
- 9、modules in any languages：基于模块工作，丰富的内置模块，可使用任意语言开发模块；
- 10、yaml, not code：使用yaml语言定制剧本playbook，易于管理，API简单明了；
- 11、ssh by default：基于SSH工作，整个过程简单、方便、安全，建议使用公钥方式认证；
- 12、strong multi-tier solution：可实现多级指挥。

简述Ansible中如何保存敏感数据？

在ansible内容中保留秘密数据并仍然公开共享，那么可以在playbooks中使用Vault。Ansible Vault，它包含在Ansible中，可以加密和解密Ansible使用的任何结构化数据文件。

简述Ansible适合的场景？

Ansible将编排与配置管理、供应和应用程序部署结合并统一在一个易于使用的平台上。Ansible的一些主要场景包括：

- 配置管理：集中配置文件管理和部署是Ansible的一个常见场景。
- 应用程序部署：当使用Ansible定义应用程序，并使用Ansible Tower管理部署时，团队可以有效地管理从开发到生产的整个应用程序生命周期。
- 部署：当在系统上部署或安装应用程序时，Ansible和Ansible Tower可以帮助简化供应系统的流程，无论是PXE启动的裸金属服务器或虚拟机，还是从模板创建虚拟机或云实例。
- 持续交付：创建CI/CD管道需要许多团队的协调和参与。如果没有一个简单的自动化平台，团队协作很难完成。而Ansible playbook在应用程序的整个生命周期中可以保持适当的部署(和管理)
- 安全性和审计：当安全策略在Ansible中定义时，可以将站点范围的安全策略的扫描和修复集成到其他自动化流程中。安全性是部署的所有内容中不可或缺的一部分。
- 编排：配置本身不能定义环境，需要定义多个配置如何交互，并确保可以将不同的部分作为一个整体来管理。

简述Ansible Inventory？

Ansible中受管主机列在主机清单（inventory）文本文件中，清单还将这些系统组织成group，以便更容易地进行批量管理。一个Inventory定义了Ansible将管理的主机集合。这些主机还可以分配至组，可以对组进行批量管理。组可以包含子组，主机可以是多个组的成员。Inventory根据类型可分为静态清单和动态清单：

- 静态主机Inventory可以由文本文件定义。
- 动态主机Inventory可以由脚本或其他程序根据需要使用外部信息提供者生成。

简述Ansible配置文件优先级？

Ansible 只使用最高优先级配置文件中的设置，其它配置文件中的设置将被忽略。即使存在其他优先级较低的文件，它们的设置也将被忽略，并且不会与所选配置文件中的设置相结合。

\$ANSIBLE_CONFIG环境变量指定的任何文件都将覆盖所有其他配置文件。如果没有设置该变量，接下来将检查运行ansible命令的目录以查找ansible.cfg文件。如果该文件不存在，则检查用户的主目录以查找.ansible.cfg文件。如上配置文件都不存在时，才使用全局/etc/ansible/ansible.cfg文件。

简述Ansible ad-hoc命令？

Ad-Hoc命令是一种快速执行单个Ansible任务的方法，适合于不需要永久保存该任务，临时执行的场景。Ad-Hoc是简单的控制台操作，无需编写剧本就可以运行。它们对于快速测试和更改非常有用。

简述Ansible ad-hoc和playbook的区别？

- Ad-Hoc 命令可以作为一次性命令对一组目标主机运行单个、简单的任务。
- Ad-Hoc 不适合复杂配置管理或编配场景，Ad-Hoc 一次只能调用一个模块和一组参数。当需要多个操作时，必须使用多个 Ad-Hoc 来执行。
- playbook可以实现以一种简易重复的方式对一组目标主机运行多个复杂的任务。
- Playbook 是描述要在受管主机上实施的必要配置或程序性步骤的文件。
- Playbook 为配置管理和部署提供了强大而灵活的解决方案。
- Playbook 可以将冗长而复杂的管理任务转变为可轻松重复的历程，并且可预测成果然而。
- playbook 是一个文本文件，其中包含一个或多个按顺序运行的play的列表。
- playbook中，可以将playbook中的tasks保存为人类可读且可立即运行的形式。
- play 是一组有序的任务，应该对从目录中选择的主机运行。

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如何实现任务的循环？

简单循环：

- Ansible支持使用loop在一组item上迭代任务；
- loop可以使用列表中的每个项、列表中每个文件的内容、生成的数字序列或使用更复杂的结构来重复任务。
- 使用loop使管理员不必编写使用相同模块的多个任务。

复杂（嵌套）循环：

- with_nested键用于嵌套循环，循环在循环中运行。它需要一个包含两个或多个列表的列表。例如，将一个列表划分为两个列表，任务将迭代第一个列表中的每一项与第二个列表中的每一项。

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handler程序。handler程序是响应由其他任务组成的通知的任务。每个handler程序都有一个全局唯一的名称，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。

简述Ansible Block?

在 playbook 中， blocks 是囊括了任务的子句；

blocks 允许对任务进行逻辑分组，并可用于控制任务的执行方式，例如，管理员可以定义一组主要任务和一组附加任务，附加任务仅在第一组失败时执行。为此，可利用三个关键字在 playbook 中使用块：

- block：定义要运行的主要任务；
- rescue：定义将在 block 子句中定义的任务失败时运行的任务；
- always：定义始终都独立运行的任务，不论 block 和 rescue 子句中定义的任务是成功还是失败。

简述Ansible如何处理play错误的？

Ansible审查每个任务的返回代码，以确定任务是否成功或失败。默认情况下，当一个任务失败时，Ansible会立即中止该主机上的其他操作，并跳过所有后续任务。

实际生产中，若希望即使任务失败也能继续执行play，Ansible也包含了多种特性用于管理任务错误：

忽略任务失败：在任务中使用ignore_errors关键字忽略错误，即使任务失败，也继续在主机上执行playbook。

简述Ansible角色？

数据中心有各种不同类型的主机。如web服务器、数据库服务器，基于开发环境的服务器。随着时间的推移，具有处理所有这些情况的任务和人员的Ansible playbook将变得庞大而复杂。

- 角色允许将复杂的剧本组织成独立的、更小的剧本和文件。
- 角色提供了一种从外部文件加载任务、处理程序和变量的方法。
- 角色也可关联和引用静态的文件和模板。
- 角色可以编写成满足普通用途需求，并且能被重复利用。
- 定义角色的文件具有特定的名称，并以严格的目录结构进行组织。

简述Ansible Galaxy？

Ansible Galaxy是一个由各种Ansible管理员和用户编写的Ansible角色的公共库。它是一个包含数千个Ansible角色的归档文件，并且有一个可搜索的数据库，帮助Ansible用户识别可能帮助他们完成管理任务的角色。Ansible Galaxy包括指向新用户和角色开发人员的文档和视频的链接。

简述Ansible如何控制任务的并行执行？

通过在所有主机上并行运行任务，Ansible可以对剧本的执行进行更多的控制。默认情况下，Ansible默认最多并行5个，因此它将同时在5台不同的机器上运行一个特定的任务。Ansible可以通过配置forks来设置并行执行任务数量。

同时Ansible也可以通过serial来减少fork数量所指定的并行数，serial关键字主要用于控制滚动更新，避免一次性更新过多的节点。

简述Ansible故障后的排查思路？

- **日志判断**：默认情况下，Ansible没有配置为将其输出，记录到任何日志文件中。可通过ansible.cfg配置文件default部分中的log_path参数或\$ANSIBLE_LOG环境变量进行配置。然后通过日志进行定位。
- **Debug模块**：调试模块是Ansible可用的模块之一，它可以更好地了解控制节点上正在进行的操作。这个模块可以在playbook执行时为某个变量提供值。
- **syntax-check**：通过ansible-playbook 命令的 --syntax-check选项检查剧本的YAML语法。

- `diff`: Ansible还提供了`--diff`选项。此选项报告对受管主机上的模板文件所做的更改。如果与`--check`选项一起使用，这些更改将显示出来，而不是实际执行。从而判断Ansible整个过程需要做何种更改。