



华为云培训中心

学以致用 云世界大有可为

Ansible自动化部署LNMP

华为云 + 智能, 见未来

www.huaweicloud.com



前言

- 课程介绍：
 - 自动化运维工具Ansible
 - 通过华为云资源和 DevCloud 快速部署LNMP环境
- 适合人群：
 - 对自动化运维感兴趣的在职工作者
 - 想使用华为云DevCloud进行开发的高校学生/开发者
- 培训方案：
 - 课程讲解、实验操作、试题测试



目标

- 学完本课程后，您将能够：
 - 对自动化运维有基本认识
 - 熟悉如何编写 Ansible 运维脚本
 - 熟悉华为云DevCloud开发平台，代码托管和部署



目录

1. 自动化运维发展趋势
2. Ansible基础简介
3. DevCloud Ansible介绍
4. DevCloud Ansible实践



运维的定义

- 运维（O&M），从字面来看，可以分为两部分：
 - 运行 Operation
运，一般的理解是运行，将服务运转起来，以满足用户和客户的需求；进一步的含义还有运筹，即统筹安排资源，提供最优解决方案，以达到效益最大化
 - 维护 Maintenance
维，一般的理解是维护，维持并监护服务的运行过程，包括应对服务管理请求和事件；进一步的含义还有维系，或者说连接，特指其起到的承上启下和枢纽作用
- IT运维是指企业IT部门采用相关的方法、手段、技术、制度等，对IT软硬运行环境、IT业务系统和IT运维人员进行的管理



运维的范围

- IT运维应包括以下内容：
 - 环境定义：包括开发环境、测试环境、预生产环境、生产环境等
 - 部署：将包有效地部署到不同环境的能力
 - 监控：部署后能够监控系统 and 应用程序
 - 告警响应：告警发生时，对告警的响应和处理机制
 - 性能调优：包括对各种服务的调优，如Nginx、Java、PHP、数据库、网络等
 - SLA保障：一般需要与业务相关部门讨论确定



传统运维的痛点

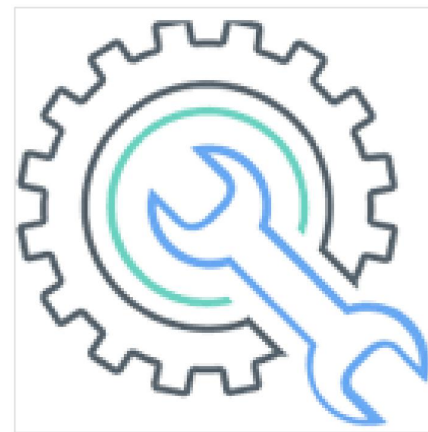
- 传统的IT运维是等到IT出现故障后，再由运维人员采取相应的补救措施。这种被动的、孤立的IT运维管理模式往往使IT部门疲惫不堪，主要表现在以下三个方面：



运维人员被动、效率低



缺乏有效的IT运维机制

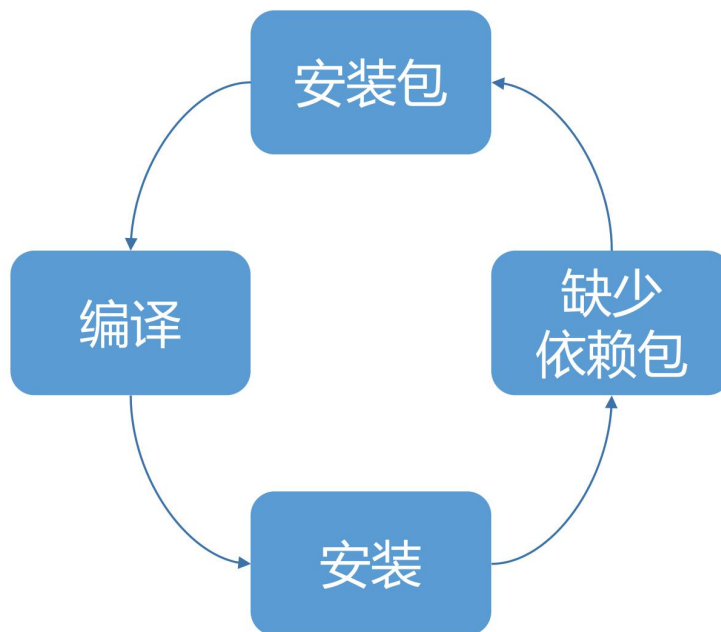


缺乏高效的IT运维工具



传统运维的缺陷

- 我们在部署应用的时候需要预先安装各种依赖、运行环境，同时由于应用通常是一系列服务的组合，需要一一安装。因此部署一个应用是一件痛苦而漫长的过程





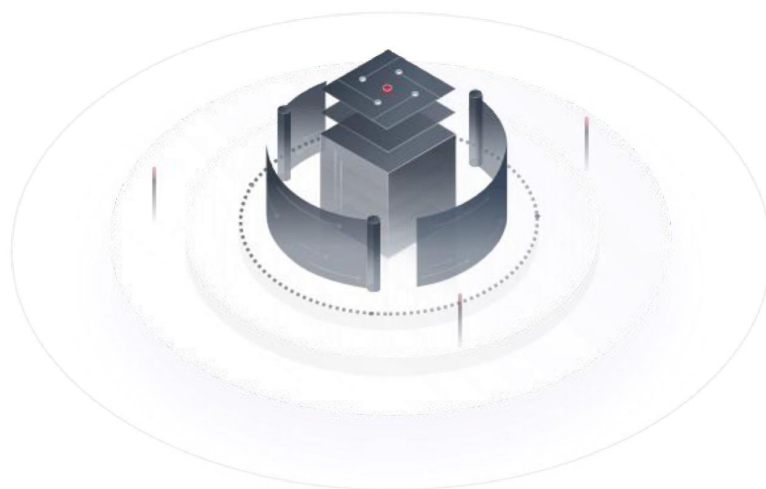
自动化运维的背景

- 您是否希望摆脱繁重的日常操作和维护服务器的工作量？您是否希望把这些无聊的东西自动化，把注意力集中在更有创意的东西上？
- 自动化运维消除了手动和重复性任务，使用工具和平台来自动化曾经是手动的流程，如打补丁、配置和部署等



自动化运维的定义

- 自动化运维是通过工具或者平台，实现IT基础设施和应用的日常任务和运维流程自动化，从而提高效率，降低风险，促进组织业务能力提升
- 主要包括：日常任务处理自动化、运维流程自动化、IT服务自动化、业务服务自动化以及整体运维运营能力升级等内容





自动化运维的三个阶段

运行自动化



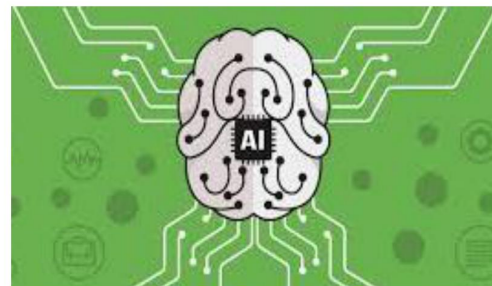
- 在这个阶段通过写一些单机脚本来进行运维工作，但是一旦运行条件发生变化，可能要改变脚本。运维压力仍然很大，容易出错

场景自动化



- 自动化工具根据外部环境判断如何运行，这些判断条件是预先定义的。这种运维系统需要各种环境数据来判断，依赖于流程引擎

人工智能运营AIOps

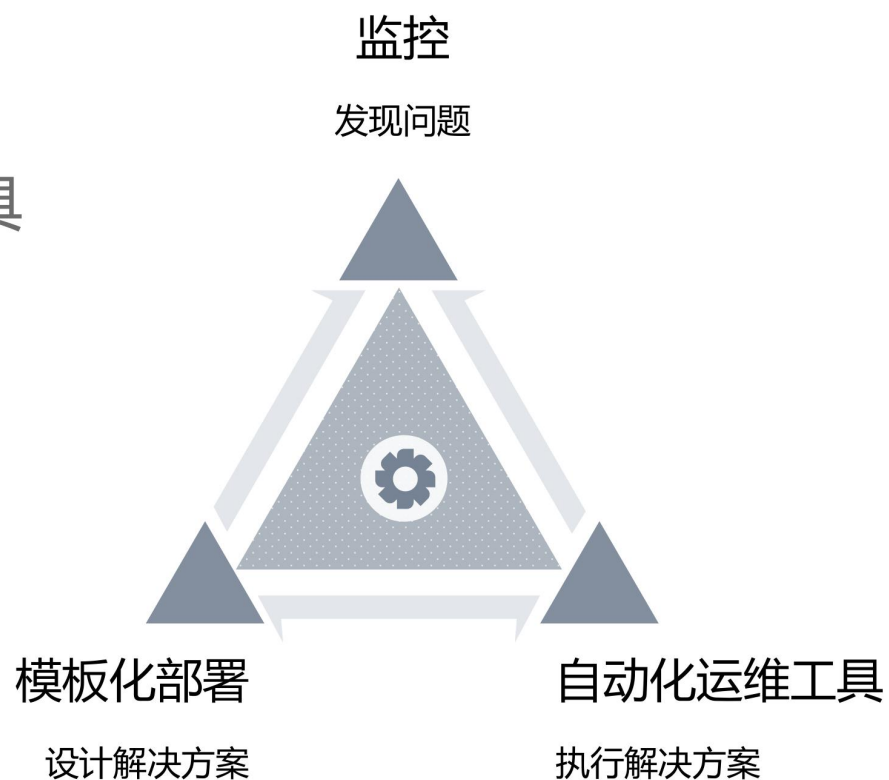


- 人工智能操作AIOps。这个层次的运维系统有一个数据核心，能够根据数据进行分析 and 判断，并能自己做出决策和执行



自动化运维技术要素

- 监控
- 模板化部署
- 自动化运维工具





自动化运维工具

- 业界主流的自动化运维工具有Ansible、SaltStack、Puppet等
 - Ansible: 是一个非常简单的IT自动化系统。它能够处理配置管理、应用程序部署、云供应、特定任务执行、网络自动化和多节点编排
 - SaltStack: 是服务器基础设施的集中管理平台, 具有配置管理、远程执行和监控功能
 - Puppet: 是Unix平台的集中配置管理系统, 它可以管理配置文件、用户、cron任务、软件包、系统服务等





小结

本章主要介绍了：

- 运维的内涵和基本职责
- 传统运维的痛点和缺陷
- 自动化运维的三个阶段



目录

1. 自动化运维发展趋势
2. Ansible基础简介
3. DevCloud Ansible介绍
4. DevCloud Ansible实践

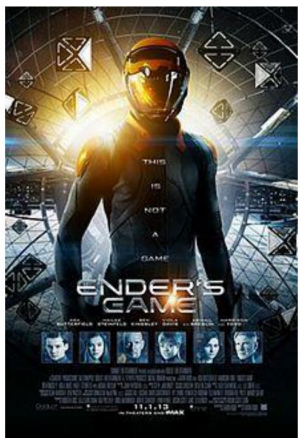


Ansible开端

- 2012年，Ansible正式发布第一个版本，原作者Michael DeHaan
- 2013年Ansible Inc公司成立，负责Ansible的商业化和捐助
- 在2015年RedHat收购Ansible



ANSIBLE



趣味知识：

Ansible名字的来源是作者Michael DeHaan最喜欢的一本小说《安德的游戏》，在小说中Ansible是一种超光速通讯装置，主角安德和他的同伴通过Ansible实时控制无数的战舰来进行战争游戏



Ansible主要功能



- DevOps加速了开发、测试和运维的循环，使过去“三年部署一次，修修补补用三年”的应用部署变成了可能一天内就要部署更新多次，手动方式渐渐变得无法满足需求
- 使用Ansible自动化运维可以减少服务中断时间、测试基础设施、降低意外风险，让开发、测试和正式环境无缝接轨，提升DevOps的整体效能

如同科幻世界中安德使用Ansible操控无数的战舰，现实中的Ansible用于操作无数的服务器



Ansible应用场景

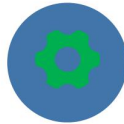
部署

自动化部署，使用简单可读的语言实现无缝的从部署到配置的流程



配置管理

批量更新配置，推送配置文件到大量节点，适用新安装场景或更新场景



应用开发

能够通过一个通用框架可靠，一致地部署多层应用程序



持续交付

满足“快速发布，频繁发布”的需求，在持续交付过程中通过滚动更新等方式减少停机



安全自动化

提前预制安全应对策略，遇到安全威胁自动开启响应



编排

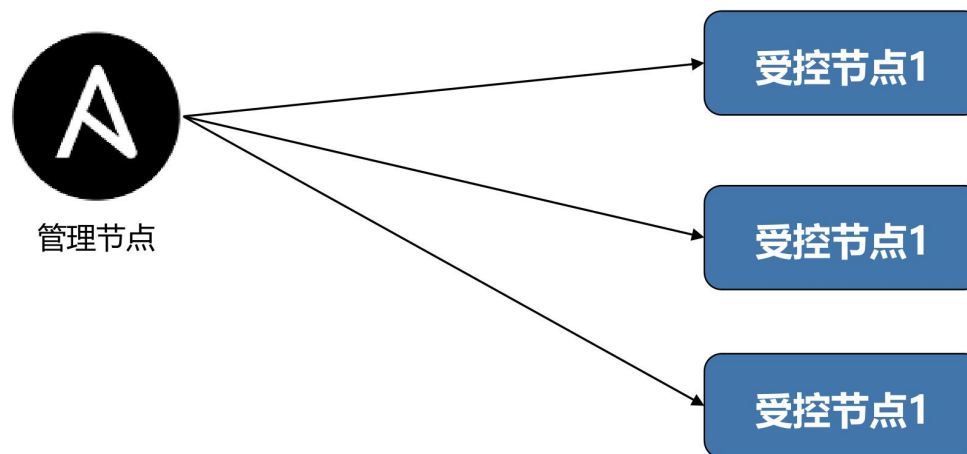
复杂应用场景可能包含多集群、多数据中心、多云，自动化编排简化复杂工作





Ansible架构

- Ansible的基础结构分为控制节点和受控节点
- 默认情况下，Ansible使用SSH协议在控制节点和受控节点之间进行通信





Control (控制) 节点

- Ansible安装在控制节点
- 在控制节点上可以运行命令以及playbook
- 你可以将任何安装了Python的节点设置为控制节点，包括笔记本电脑、远程桌面或者是服务器
- 不支持使用windows主机作为控制节点
- Ansible支持多个控制节点



Managed (受控)节点

- 受控节点指所有受控制节点管辖的节点
- 受控节点通常也会被称为 “hosts”
- 受控节点上无需安装代理等服务
- 节点可以是服务器，也可以是其他硬件设备，如网络设备，甚至是虚拟节点，例如：
 - 容器
 - SDN、无线和其他网络控制器



Inventory

- Inventory翻译为中文意为“清单”，它本质上就是一份清单——受控节点的清单
- Inventory直观呈现是一个文件，类似windows的ini文件，上面保存着各个主机的IP地址、端口号等必须的信息
- Inventory默认文件路径在/etc/Ansible/hosts中



如何让所有绿色服务器成为受控主机？
想要将操作下发到红色主机该如何实现？

Modules



=



+



+



Ansible任务：
完成软件部署安装

步骤一：
下载软件

步骤二：
安装软件

步骤三：
配置软件

- 一个完整的Ansible任务可以拆分为一系列子任务的集合，这些子任务是高度模块化的，只要稍作参数改动就可被放到各种Ansible任务里。这使得一个完整的Ansible任务就像一个积木拼起来的玩具 这个“积木”就是模块
- Module是Ansible执行代码单元，创建一个Ansible任务可以用一个或多个模块，我们日常用到的模块大多是内建模块



Playbook

- Playbook相当于一个Tasks的列表，其中的Tasks通常是有序的
- 我们也可以重复执行Playbook，结果应该是一致的（幂等性）
- Playbooks使用YAML格式语法编写，易读、易写、易懂
- 后续我们将详细介绍Playbooks的编写

```
---  
- hosts: webservers  
  vars:  
    http_port: 80  
    max_clients: 200  
    remote_user: root  
  tasks:  
    - name: ensure apache is at the latest version  
      yum: pkg=httpd state=latest  
    - name: write the apache config file  
      template: src=/srv/httpd.j2 dest=/etc/httpd.conf  
      notify:  
        - restart apache  
    - name: ensure apache is running  
      service: name=httpd state=started  
  handlers:  
    - name: restart apache  
      service: name=httpd state=restarted
```



Playbooks简介

- Playbooks与ad-hoc相比，是一种完全不同的运用Ansible的方式，是非常之强大的。简单来说，playbooks是一种简单的配置管理系统与多机器部署系统的基础，与现有的其他系统有不同之处，且非常适合于复杂应用的部署
- Playbooks的格式是YAML，语法做到最小化，意在避免playbooks成为一种编程语言或是脚本，但它也并不是一个配置模型或过程的模型



YAML介绍

- 如果学习过Kubernetes，相信对YAML不会陌生，它可以便捷的让我们发放各种基于容器的应用
- YAML语言是专门用来写配置文件的语言，非常简洁和强大
YAML在设计之初的目标，就是方便人类读写，因此在阅读YAML格式的文本时往往会感觉较为易懂 它实质上是一种通用的数据串行化格式

趣味知识:

YAML是"YAML Ain't a Markup Language" (YAML不是一种标记语言) 的递归缩写



YAML基本格式 - 通用格式

- 建议在每个YAML文件开头用三个短横表示一个文件的开始（不管是不是在Ansible中），这是YAML格式的一部分，一个小怪癖
- 在YAML格式中用缩进表示不同层级，如右边的文件中，“前菜”和“正餐”是一个层级，而“水煮毛豆”要比二者低一个层级
- 缩进需要用空格表示，不能使用tab&空格，数量随意，但为了展示美观，用两个的情况比较多

```
---  
# 今天的晚餐  
前菜：  
    水煮毛豆  
正餐：  
    - 红烧肉  
    - 炒青菜  
水果：西瓜
```



YAML基本格式 - 列表

- Ansible的playbook中，往往开头是以列表类型为开头的
- YAML对于列表的语法如右图所示
 - 用一个短横和一个空格做为开头
 - 列表中所有成员（平等的），处于相同的缩进级别

```
---  
# 一个美味水果的列表  
- Apple  
- Orange  
- Strawberry  
- Mango
```




YAML基本格式 - 字典

- 一个字典是指一个由简单的“键: 值”形式组成的
- 键和值之间有一个冒号和一个空格
- 值不一定是一个简单的元素，可能是一个列表，或者是另外的字典，如右侧示例

```
# Employee records
- martin:
  name: Martin D'vloper
  job: Developer
  skills:
    - python
    - perl
    - pascal
- tabitha:
  name: Tabitha Bitumen
  job: Developer
  skills:
    - lisp
    - fortran
    - erlang
```



Playbooks的 “Plays”

- Playbooks的语法尽量做到最小化，避免playbooks成为一种编程语言或是脚本，前面已经强调过，Ansible是避免面向过程的
- Playbooks是由一个或多个Plays组成的，它其实是一个Plays为元素的列表
- 回顾一下之前的内容：在Ansible中，play的内容，被称为tasks，即任务。在基本层次的应用中，一个任务是一个对Ansible模块的调用



Tasks

- Tasks是Playbook中基本的单元，在一个Play中会包含一系列tasks，在上一章节中我们使用过Ad-hoc命令，一个命令相当于一个Task
- 在Playbook中Task的写法如右图所示，可以看到这个任务列表中包含了三个子tasks

tasks:

- name: ensure apache is at the latest version
yum: pkg=httpd state=latest
- name: write the apache config file
template: src=/srv/httpd.j2
dest=/etc/httpd.conf
notify:
 - restart apache
- name: ensure apache is running
service: name=httpd state=started



Modules

- Module是Ansible执行代码单元，创建一个Ansible任务可以用一个或多个模块，我们日常用到的模块大多是内建模块
- 我们之前将Modules称为搭建Ansible任务的积木，那么我们继续看上一页中的代码（右图），其中用到了三个Modules：
 - Yum
 - Template
 - Service

tasks:

```
- name: ensure apache is at the
latest version
  yum: pkg=httpd state=latest
- name: write the apache config file
  template: src=/srv/httpd.j2
dest=/etc/httpd.conf
  notify:
    - restart apache
- name: ensure apache is running
  service: name=httpd state=started
```



Hosts与用户

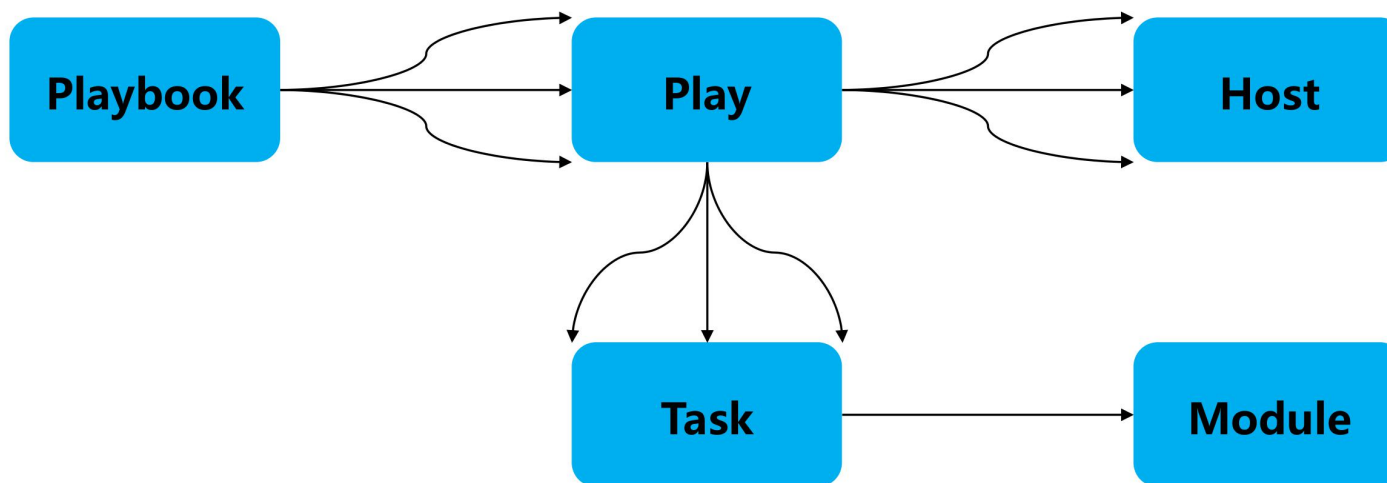
- 一个Ansible任务运行必不可少的是指定运行在何方，Inventory中我们维护了一组主机的清单列表，但任务并非需要运行在清单中所有的主机上，通过Playbook中Hosts元素的指定，我们圈定了哪些服务器上将会运行这个任务
- Hosts的写法十分简单，如下图所示，运行的主机组就是Inventory中的指定的“webservers”这一组主机 如果有多个主机组用逗号分隔
- 用户使用remote_user参数表示，含义为使用哪个用户在远程主机上操作任务，写法如下图

```
- hosts: webservers  
  remote_user: root
```



Playbook的结构

- 一个完整的Playbook包含一个或多个Play，每个Play包含一个或多个Host以及一个或多个Task，其中每个Task都使用了一个Module





一个完整的Playbook (1)

```
---  
- hosts: webservers  
  vars:  
    http_port: 80  
    max_clients: 200  
    remote_user: root  
  tasks:  
    - name: ensure apache is at the latest  
      version  
      yum: pkg=httpd state=latest  
.....
```

- 我们从一个入门的Playbook文件来看一下这写元素是如何组合的

- 1, 三个短横表示开始
- 2, 指定运行的主机组, 短横开始表示列表
- 3, vars, 一系列变量与值, 可选项, 在本文件中指定了端口号和客户端上线两个参数
- 4, 指定在hosts上使用root账号运行任务



一个完整的Playbook (2)

```
.....
tasks: 5
  - name: ensure apache is at the latest
    version
    yum: pkg=httpd state=latest 6
  - name: write the apache config file
    template: src=/srv/httpd.j2
    dest=/etc/httpd.conf
  notify: 7
  - restart apache
  - name: ensure apache is running
    service: name=httpd state=started
handlers:
  - name: restart apache 8
    service: name=httpd state=restarted
```

- 5, tasks缩进与之前的hosts, vars等一致, 之后知道handlers之前的内容都是tasks
- 6, 第一个task, 完成yum模块的使用, 包括了模块使用中一些参数的配置
- 7, 第二和第三个tasks, template和服务
- 8, handlers, 一种触发式的任务, 只有当前文的notify触发了, handler才会启动



运行Playbook

- Playbook在编写完成后是一个yaml格式的文件，使用ansible-playbook命令可以运行该任务，并且显示任务运行的结果

```
[root@ansible-server playbookdir]# ansible-playbook command.yml
PLAY [webservers]
*****
TASK [Gathering Facts]
*****
ok: [121.36.16.50]
TASK [say hello]
*****
changed: [121.36.16.50]
PLAY RECAP
*****
121.36.16.50      : ok=2   changed=1   unreachable=0   failed=0   skipped=0
rescued=0   ignored=0
```



小结

本章主要介绍了：

- Ansible 的发展历程和主要应用场景
- Ansible 的部署架构
- YAML 文档的结构特征
- Playbook 的编写方法



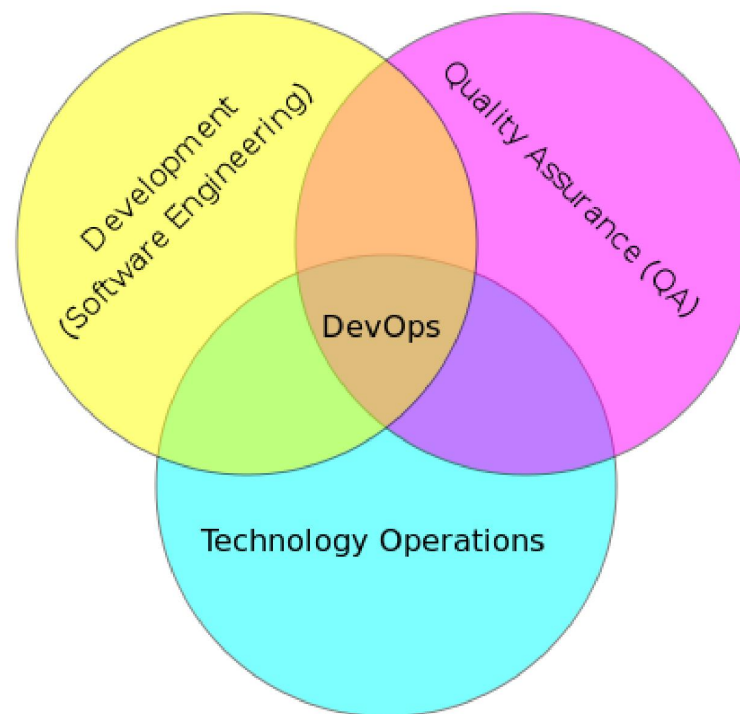
目录

1. 自动化运维发展趋势
2. Ansible基础简介
3. DevCloud Ansible介绍
4. DevCloud Ansible实践



DevOps的含义

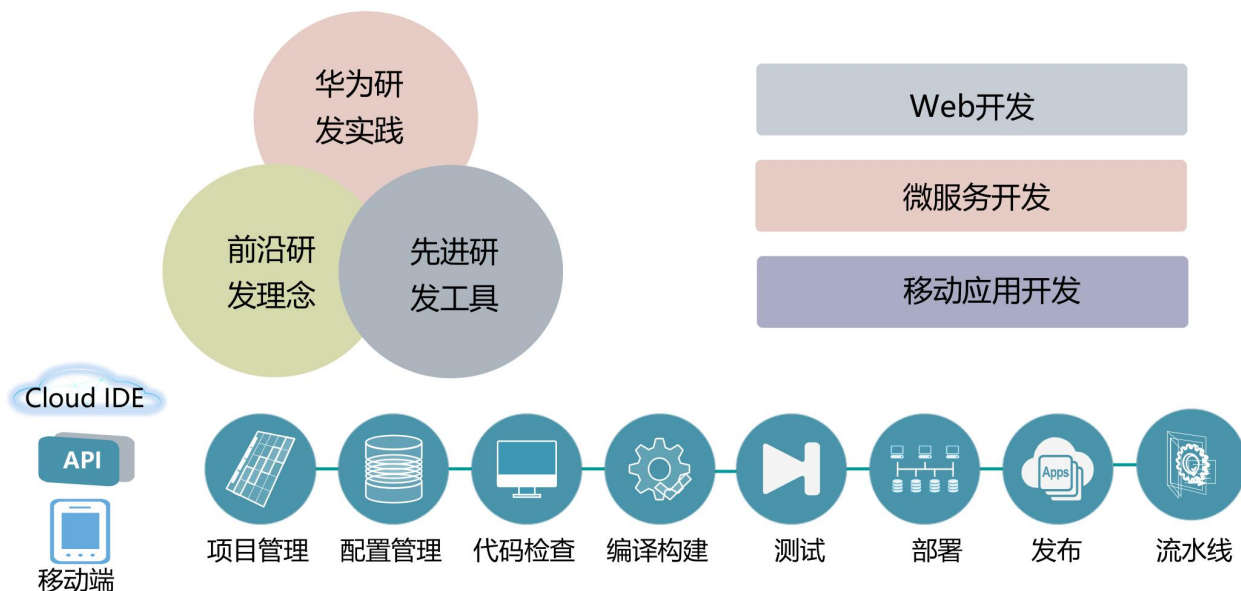
- DevOps是软件开发、运维和质量保证三个部门之间的沟通、协作和集成所采用的流程、方法和体系的一个集合，它是人们为了及时生产软件产品或服务，以满足某个业务目标，对开发与运维之间相互依存关系的一种新的理解





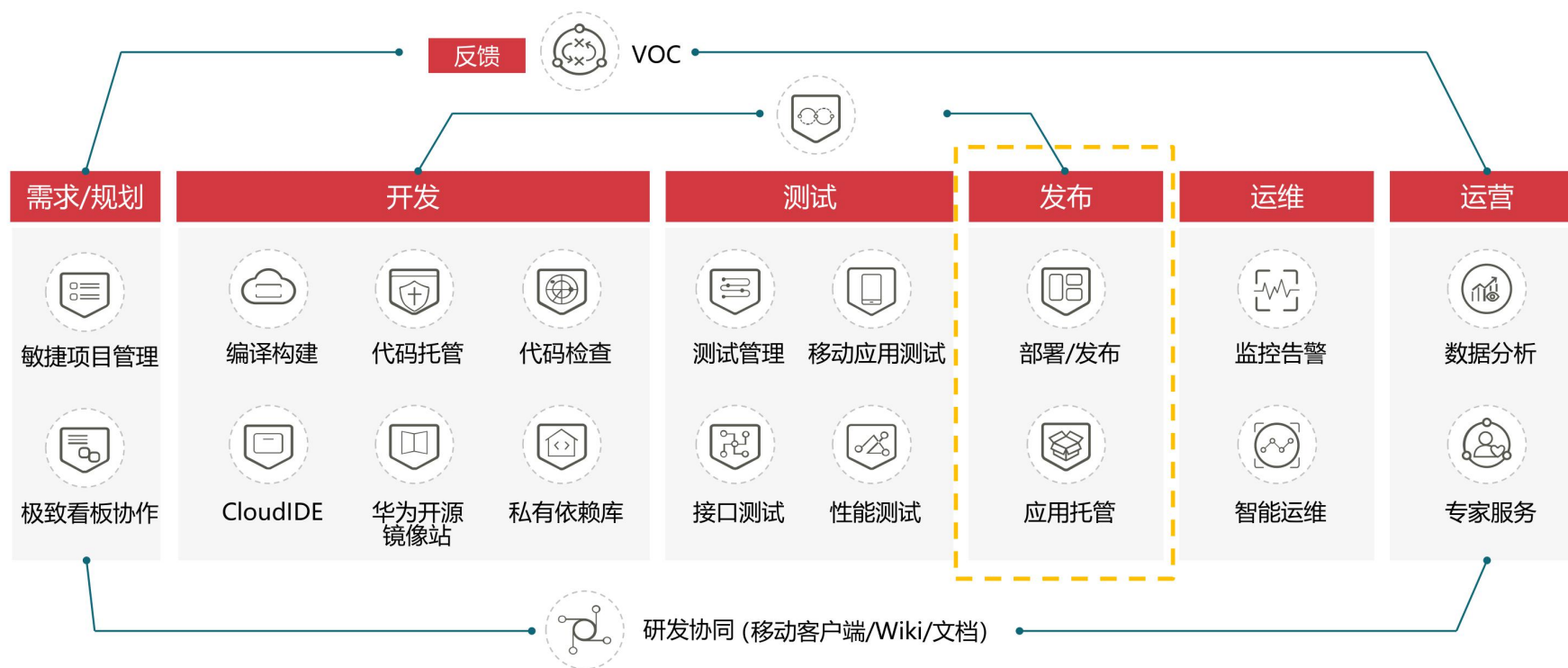
华为云DevOps平台 - DevCloud

- 软件开发平台（DevCloud）是集华为近30年研发实践、前沿研发理念、先进研发工具为一体的一站式云端DevOps平台，面向开发者提供的云服务，即开即用，让开发者快速而又轻松地开启云端开发之旅





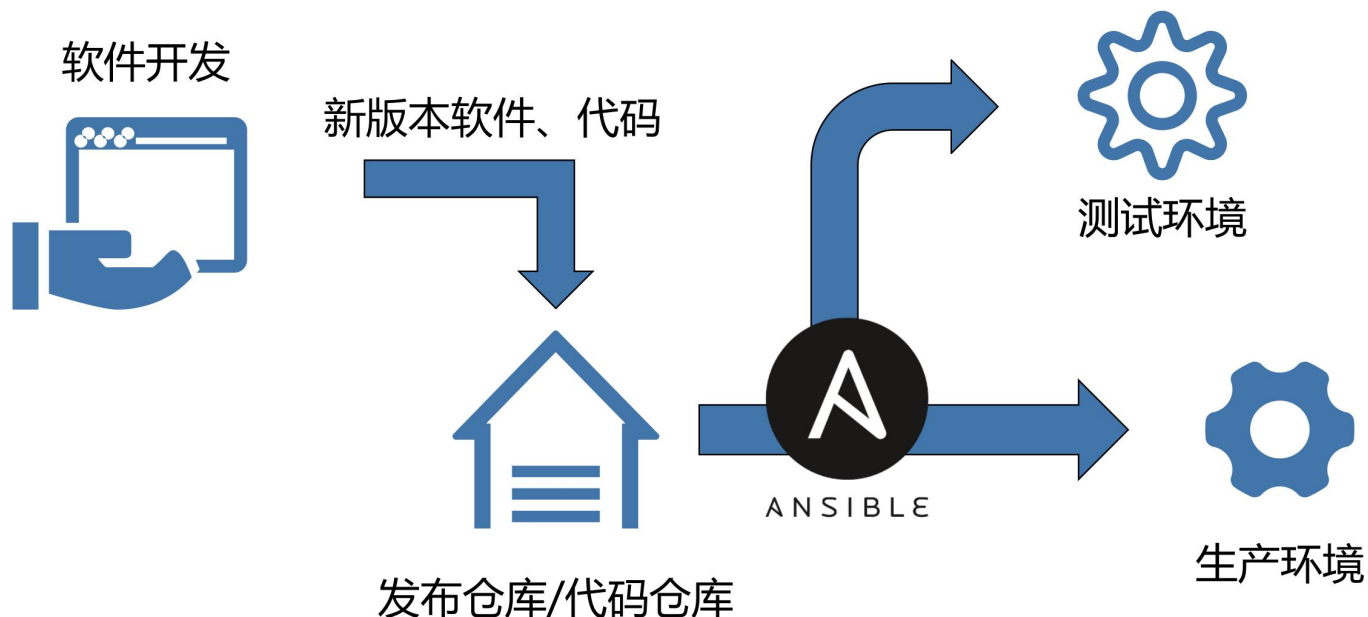
全流程DevOps平台



- Ansible是在发布阶段可以使用的一种工具



DevCloud中Ansible的使用场景

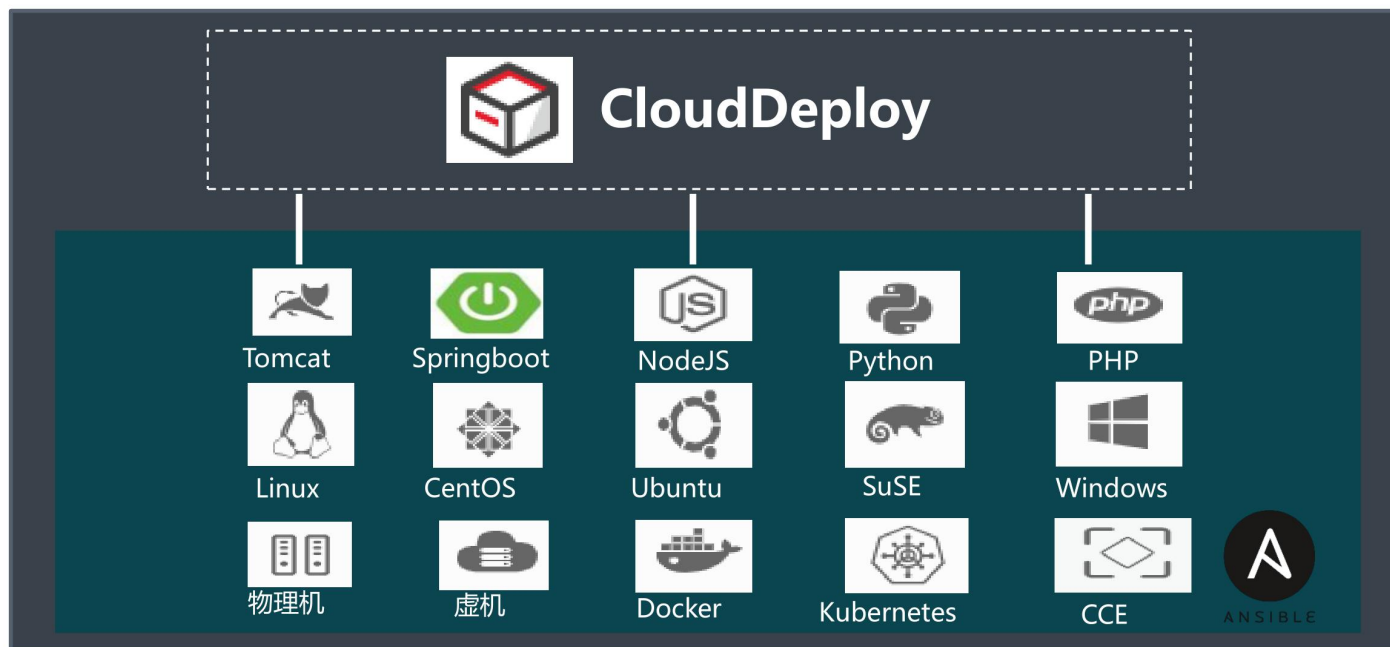


在DevCloud中 Ansible承担的工作是将发布仓库或者代码仓库中已经编写好的软件、代码等发布到环境当中。同时playbook等发布必须的脚本也保存在仓库中



云部署服务

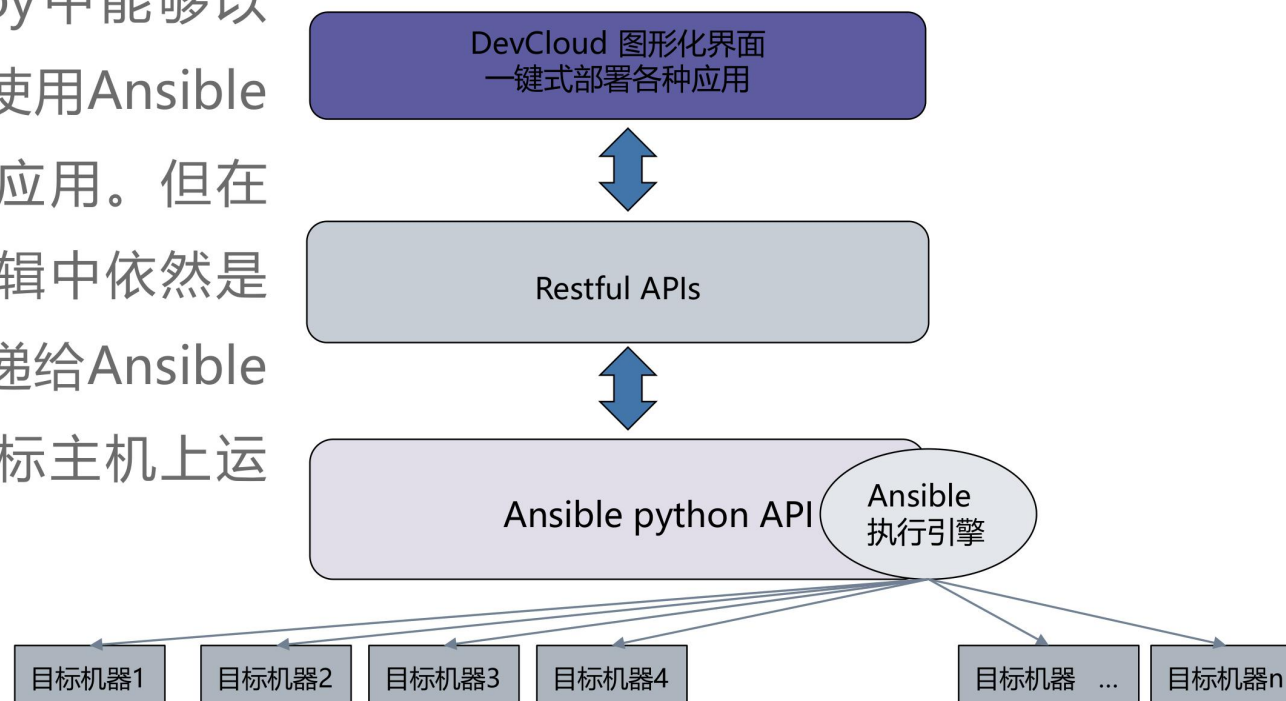
- Ansible在DevCloud中作为一种部署工具存在，隶属于CloudDeploy云服务。除了Ansible之外，CloudDeploy提供了更多功能实现丰富的云部署





CloudDeploy中使用Ansible

- 在CloudDeploy中能够以图形化的方式使用Ansible来部署和编排应用。但在底层的业务逻辑中依然是将这些任务传递给Ansible执行引擎到目标主机上运行





小结

本章主要介绍了：

- DevCloud 是什么
- 如何在 DevCloud 中使用Ansible
- DevCloud 为自动化运维带来的便利



目录

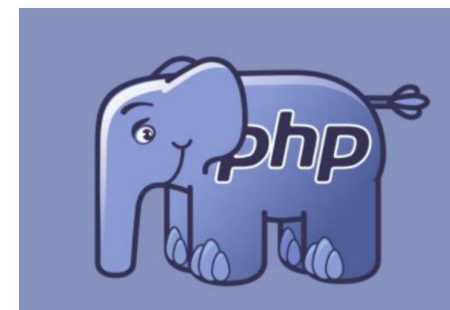
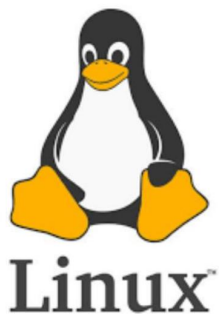
1. 自动化运维发展趋势
2. Ansible基础简介
3. DevCloud Ansible介绍
4. DevCloud Ansible实践



Ansible自动化部署LNMP

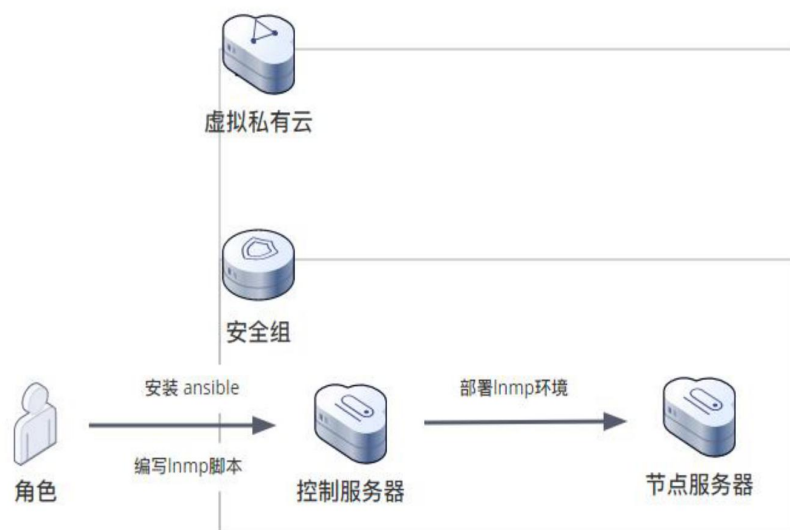
- LNMP 是什么?

LNMP=Linux + Nginx + Mysql + PHP 是广为接受的一种企业网站架构



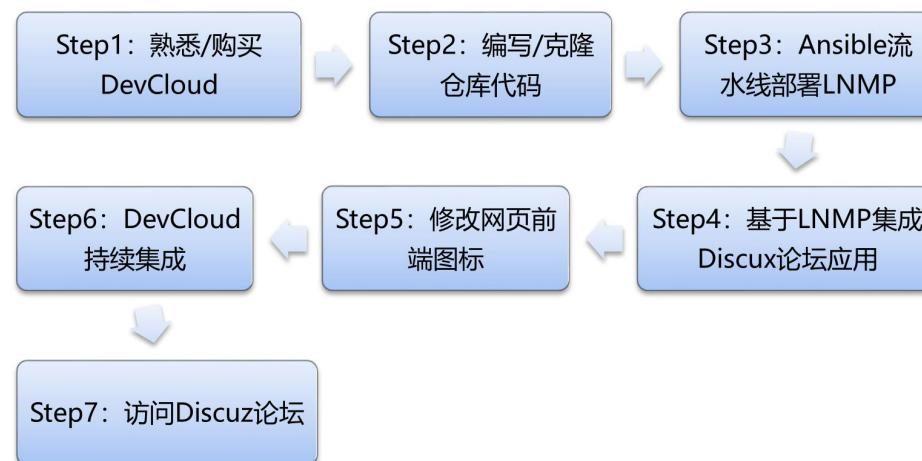
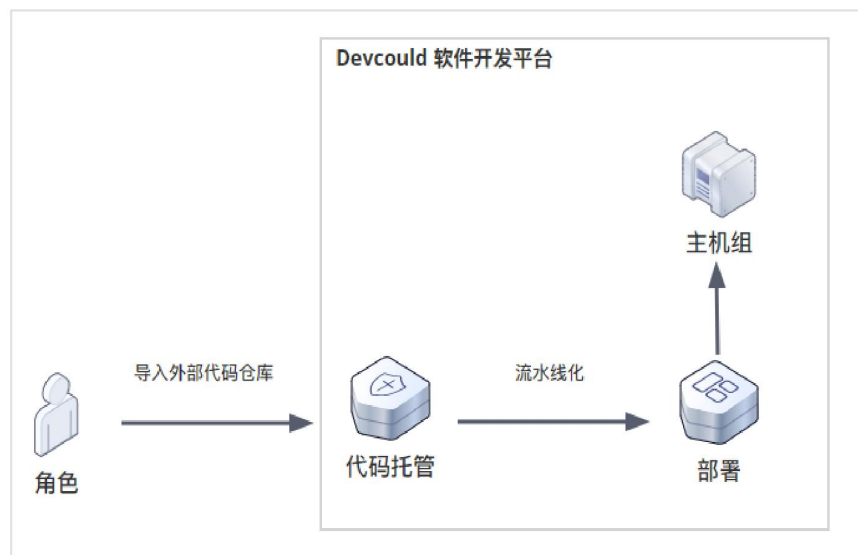


ECS中使用Ansible部署LNMP流程





DevCloud中使用Ansible部署LNMP流程





DevCloud中使用Ansible的优势

与ECS中使用Ansible相比

- DevCloud 中不需要在控制主机上安装Ansible，自身集成
- DevCloud 中可直接导入购买的ECS，无需维护hosts文档
- DevCloud 中代码托管playbook，可克隆可编辑
- DevCloud 网页一站式运维，开发和运维更近一步



学习推荐

- [华为云Ansible](#)
- [Ansible 中文指南](#)
- [Ansible 官网手册](#)
- [华为云DevCloud](#)
- [What is Ansible](#)



华为云培训中心

学以致用 云世界大有可为

Thank You.

Copyright©2021 Huawei Technologies Co., Ltd. All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

华为云 + 智能, 见未来

www.huaweicloud.com