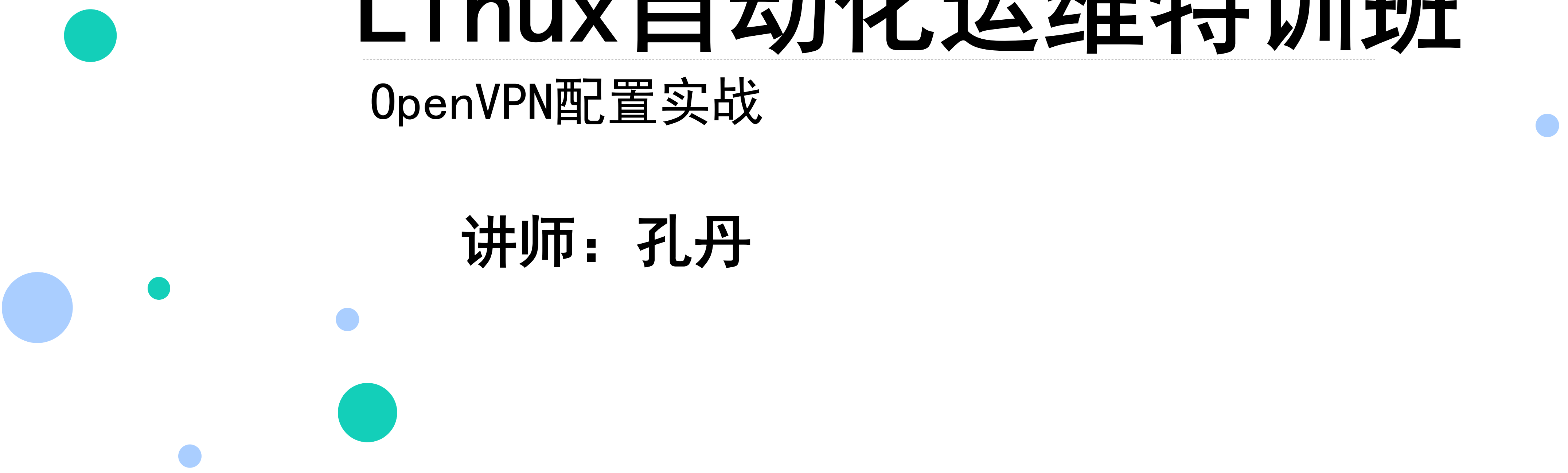


# Linux 自动化运维特训班

自动化运维基础

讲师：孔丹



# 大纲

- 自动化部署
- 自动化运维
- 自动化配置

版权所有，侵权必究

# 自动化部署

## □ DevOps 简介

1. DevOps（Development和Operations的组合词）是一种重视“软件开发人员（Dev）”和“IT运维技术人员（Ops）”之间沟通合作的文化、运动或惯例。透过自动化“软件交付”和“架构变更”的流程，来使得构建、测试、发布软件能够更加地快捷、频繁和可靠。
2. DevOps是强调产品管理，软件开发和运营专业人员之间沟通和协作的软件开发过程。DevOps还可以自动化软件集成，测试，部署和基础设施变更过程。DevOps旨在建立一套快速、频繁、稳定地进行构建，测试，发布软件的文化与环境。

版权所有，侵权必究

# 自动化部署

## □ DevOps 简介

1. DevOps一词的来自于Development和Operations的组合，突出重视软件开发人员和运维人员的沟通合作，通过自动化流程来使得软件构建、测试、发布更加快捷、频繁和可靠。DevOps概念早先升温于2009年的欧洲，因传统模式的运维之痛而生。
2. DevOps是为了填补开发端和运维端之间的信息鸿沟，改善团队之间的协作关系。不过需要澄清的一点是，从开发到运维，中间还有测试环节。DevOps其实包含了三个部分：开发、测试和运维。

# 自动化部署

## □ DevOps 好处

1. DevOps的一个巨大好处就是可以高效交付，这也正好是它的初衷。
2. DevOps另外一个好处就是会改善公司组织文化、提高员工的参与感。

版权所有，侵权必究

# 自动化部署

## □ 实现DevOps需要什么？

1. 代码管理（SCM）：GitHub、GitLab、BitBucket、SubVersion
2. 构建工具：Ant、Gradle、maven
3. 自动部署：Capistrano、CodeDeploy
4. 持续集成（CI）：Bamboo、Hudson、Jenkins
5. 配置管理：Ansible、Chef、Puppet、SaltStack、ScriptRock GuardRail
6. 容器：Docker、LXC、第三方厂商如AWS
7. 编排：Kubernetes、Core、Apache Mesos、DC/OS
8. 服务注册与发现：Zookeeper、etcd、Consul
9. 脚本语言：python、ruby、shell
10. 日志管理：ELK、Logentries
11. 系统监控：Datadog、Graphite、Icinga、Nagios
12. 性能监控：AppDynamics、New Relic、Splunk
13. 压力测试：JMeter、Blaze Meter、loader.io
14. 预警：PagerDuty、pingdom、厂商自带如AWS SNS
15. HTTP加速器：Varnish
16. 消息总线：ActiveMQ、SQS
17. 应用服务器：Tomcat、JBoss
18. Web服务器：Apache、Nginx、IIS
19. 数据库：MySQL、Oracle、PostgreSQL等关系型数据库；cassandra、mongoDB、redis等NoSQL数据库
20. 项目管理（PM）：Jira、Asana、Taiga、Trello、Basecamp、Pivotal Tracker

版权所有，侵权必究



# 自动化运维

## □ 自动化运维背景

网站业务上线，需要运维人员在短时间内完成几百台服务器部署，包括系统安装、系统初始化、软件的安装与配置、性能的监控.....

所谓运维自动化，即在最少的人工干预下，利用脚本与第三方工具或自行开发的工具，保证业务系统快速上线、7\*24小时高效稳定运行

各个阶段	运维规模（/台）	运维效率	技能要求
纯手工	小于50	低	一般
脚本与文档	50-100	一般	高
运维工具（第三方工具）	100-500	高	一般
脚本+运维工具	大于500	最高	高

# 自动化运维

## □ 自动化运维的体系结构

一个完善的运维自动化体系包括，系统预备、配置管理以及监控报警3个功能模块：

### 1. 系统预备

- 自动化安装操作系统及常用软件包
- 自动化安装与升级系统补丁
- 自动化升级相关软件
- .....

### 2. 配置管理

- 自动化部署业务系统软件包并完成配置
- 远程管理服务器
- 变更回滚.....

### 3. 监控报警

- 服务器可用性、性能状态、安全监控
- 向管理员发送报警信息等
- .....

版权所有，侵权必究



# 自动化运维

## □ 自动化运维工具

预备类工具	配置类工具	监控报警类
kickstart	Chef	Nagios
Cobbler	ControlTier	OpenNMS
OpenQRM	Func	Zabbix
Spacewalk	Puppet	Cacti

版权所有，侵权必究

# 自动化运维

## □ 自动化运维发展

运维自动化发展 运维学习和发展的一个线路：

搭建服务（部署并运行起来）

用好服务（监控、管理、优化）

自动化（服务直接的关联和协同工作）

产品设计（如何设计一个运维系统）

-----当下云计算的核心竞争力是运维！

系统架构师（偏管理）：网络 系统 数据库 开发 云计算 自动化

运维管理 服务管理 项目管理 测试 业务

-----专注于某一领域

版权所有，侵权必究

# 自动化运维

## □ 自动化运维发展

运维自动化发展 运维工作内容分类：

- 监控运维（7x24运维值班、故障处理）

- 应用运维（业务熟悉、服务部署、业务部署、版本管理、灰度发布、应用监控）

- 安全运维（整体的安全方案、规范、漏洞检测、安全防护等）

- 系统运维（架构层面的分布式缓存、分布式文件系统、日志收集、环境规划（测试、开发、生产）、架构设计、性能优化）

- 基础服务运维（包含运维开发）（内部dns、负载均衡、系统监控、资产管理、运维平台）

- 基础设施运维（系统初始化、网络维护、负责设备上下架、巡检、报修、硬件监控）

# 自动化运维

## □ 自动化运维发展

运维自动化发展——标准化

物理设备层面：

1. 服务器标签化、设备负责人、设备采购详情、设备摆放标准
2. 网络划分、远程控制卡、网卡端口
3. 服务器机型、硬盘、内存统一。根基业务分类
4. 资产命名规范、编号规范、类型规范
5. 监控标准

操作系统层面：

1. 操作系统版本
2. 系统初始化（dns、ntp、内核参数调优、rsyslog、主机名规范）
3. 基础agent配置（zabbix agent、logstash agent、saltstack minion）
4. 系统监控标准（cpu、内存、硬盘、经常）

版权所有，侵权必究

# 自动化运维

## □ 自动化运维发展

运维自动化发展——标准化

应用服务层面：

1. web服务选型（apache、nginx）
2. 进程启动用户、端口监听规范、日志收集规范（访问日志、错误日志、运行日志）
3. 配置管理（配置文件规范、脚本规范）
4. 架构规范（nginx+keepalived、lvs+keepalived）
5. 部署规范（位置、包命名等）

运维操作层面：

1. 机房巡检流程（周期、内容、报修流程）
2. 业务部署流程（先测试、后生产、回滚）
3. 故障处理流程（紧急处理、故障升级、重大故障管理）
4. 工作日志标准（如何编写工作日志）
5. 业务上线流程（1. 项目发起2. 系统安装3. 部署应用4. 解析域名5. 测试6. 加监控7. 备份）

# 自动化运维

## □ 自动化运维发展

运维自动化发展——标准化

运维操作层面：

6. 业务下线流程（谁发起，数据如何处理）

7. 运维安全规范（密码复杂度、更改周期、vpn使用规范、服务登录规范）

——目标：文档化

版权所有，侵权必究



# 自动化运维

## □ 自动化运维发展

### 运维自动化发展--工具化

- 1、shell脚本（功能性（流程）脚本、检查性、报表性）
- 2、开源工具（zabbix、elkstack、saltstack、cobbler）

目标：

- 1、促进标准化的实施
- 2、将重复的操作，简单化
- 3、将多次操作，流程化
- 4、减少人为操作的低效和降低故障率

工具化和标准化是好基友

痛点：

- 1、至少要ssh到服务器上执行，可能犯错
- 2、多个脚本有执行顺序的时候，可能犯错
- 3、权限不好管理，日志没法统计
- 4、无法避免手工操作

# 自动化运维

## □ 自动化运维发展

### 运维自动化发展--工具化

例子：比如某天我们要对一台数据库从库进行版本停机升级，那么要求进行评估

统计影响：晚上3点有定时任务连接该数据库，做数据报表统计

- 1、凌晨3点我们所有系统的定时任务有哪些 crontab
- 2、这些crontab哪些连接我们要停止的从库
- 3、哪些可以停，哪些不能停（修改到主库），哪些可以后补
- 4、这些需要后补的脚本哪个业务、谁加的、什么时候加的

版权所有，侵权必究

# 自动化运维

## □ 自动化运维发展

运维自动化发展——web化

运维平台：

例子：

- 1、做成web界面
- 2、权限控制
- 3、日志记录
- 4、弱化流程
- 5、不用ssh到服务器，减少人为操作造成的故障

dns web 管理 bind-DLZ

负载均衡web管理

job管理平台

监控平台 zabbix

操作系统安装平台

版权所有，侵权必究

# 自动化运维

## □ 自动化运维发展

运维自动化发展--服务化（api化）

dns web 管理 bind-DLZ dns-api

负载均衡web管理 slb-api

job管理平台 job-api

监控平台 zabbix zabbix-api

操作系统安装平台 cobbler-api

部署平台 deploy-api

配置管理平台 saltstack-api

1、调用cobbler-api 安装操作系统

2、调用saltstack-api进行系统初始化

3、调用dns-api 解析主机名

4、调用zabbix-api该新上线机器加上监控

5、再次调用saltstack-api部署软件（nginx-php）

6、调用deploy-api将当前版本代码部署到服务器上

7、调用test-api 测试当前服务运行是否正常

8、调用slb-api将该节点加入集群

# 自动化运维

## □ 自动化运维发展

运维自动化发展--智能化

智能化的自动化扩容、缩容、服务降级、故障自愈  
触发机制-》决策系统-》

1. zabbix触发Action

触发:

1.1 当某个集群的访问量超过最大支撑量, 比如10000

1.1.1 cpu使用率达到xx。内存使用率达到xx

1.2 当持续5分钟

1.3 不是攻击

1.4 资源池有可用资源

1.4.1 当前网络带宽使用率

1.4.2 如果公有云--钱够不够

1.5 当前后端服务支撑量是否超过阈值, 如果超过应该后端先扩容

1.6 数据库是否可用支撑当前并发

1.7 当前自动化扩展队列, 是否有正在扩容的节点

1.8 其他业务相关的

# 自动化运维

## □ 自动化运维发展

### 运维自动化发展—智能化

之前：先判断buffer是否有最近x小时，已经移除的之前创建的虚拟机，并查询软件版本是否和当前一致，如果一致跳过2、3、4步骤，如果不一致跳过2、3步骤

2. openstack 创建虚拟机
3. saltstack 配置环境-----》监控
4. 部署系统部署当前代码
5. 测试服务是否可用（注意间隔和次数）
6. 加入集群
7. 通知（短信、邮件）

### 自动化缩容：

1. 触发条件和决策
2. 从集群中移除节点----》关闭监控--》移除
3. 通知
4. 移除的节点存放在buffer里面
5. buffer里面超过一天的虚拟机，自动关闭，存放于xx区
6. xx区的虚拟机。每七天清理删除



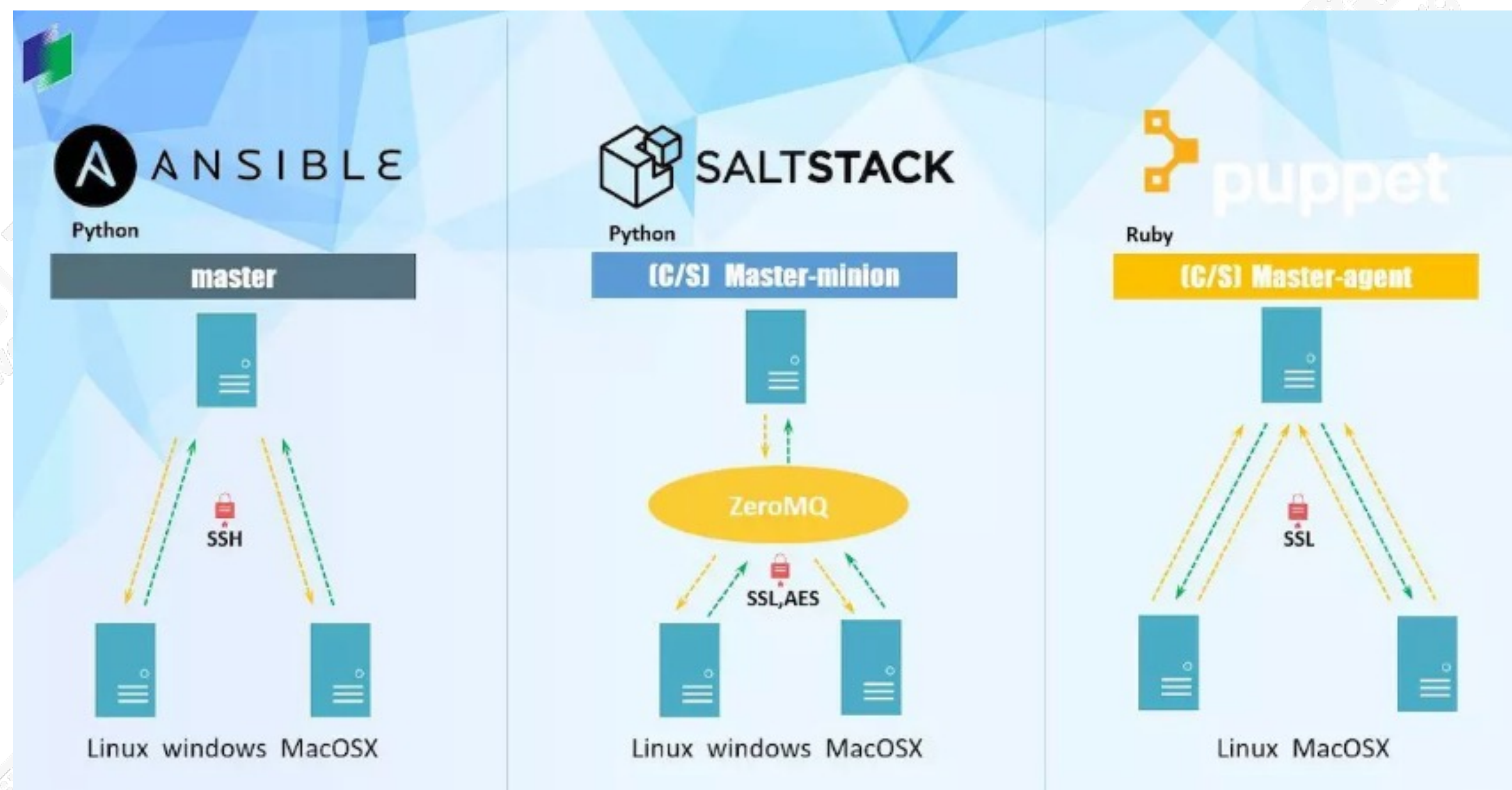
# 自动化配置

## ▣ 技术特性比较

名称	Puppet	SaltStack	Ansible
开发语言	Ruby	Python	Python
客户端	有	有	无
二次开发	不支持	支持	支持
通信验证	是	是	是
通信加密	标准SSL协议	AES加密	OpenSSH
平台支持	AIX, BSD, HP-UX, Linux, Mac OS X, Solaris, Windows	BSD, Linux, Mac OS X , Solaris, Windows	AIX, BSD, HP-UX, Linux, Mac OS X, Solaris
配置文件格式	Ruby语法格式	YAML	YAML
Web UI	提供	提供	提供（商业版本）
命令执行	不支持（配置模块可实现）	支持	支持

# 自动化配置

## □ 架构对比



版权所有，侵权必究

# 总结

- 自动化部署
- 自动化运维
- 自动化配置

版权所有，侵权必究



# 谢谢观看

更多好课，请关注万门大学APP

