系统问题

1)所使用的语言,及其对应的版本

比如：java jdk1.8

>> 后端开发 OpenJDK 1.8.0 + java　部分服务 Python 开发

>> 前端开发 Vue + Element UI (node环境，npm 构建)

2)web容器(tomcat、jar)

>> SpringBoot 1.6 以 jar 方式启动(内嵌tomcat)

3)中间件(集群)

>> redis 目前所有环境单点

>> zookeeper (线下单点, 阿里云预发布环境单点, 阿里云生产环境5节点)

>> ElasticSearch 5.5 (线下单点, 阿里云预发布环境单点, 阿里云生产环境5节点: 5 MasterNode 3 DataNode)

>> 云存储 使用 七牛云

>> 短信 (LeanCloud, SendCloud, NetEaseCloud 网易云信)

>> 支付 BeeCloud聚合支付

4)消息队列(集群)，

比如：rocketmq(2主2从，2节点)

>> kafka (线下单点, 阿里云预发布环境单点, 阿里云生产环境5节点)

5)部署方式(ansible、saltstack、是否灰度)

ansible-playbook文件，存放在jenkins上

6)代码管理方式(git、svn)

>> git 托管于码云(gitee.com)

7)运维管理系统账号如何管理

运维有root权限，架构师有部分root权限，测试和开发只有普通权限用户

8)使用了阿里云哪些产品

阿里云服务器，云数据库RDS版，文件存储NAS，云监控，web应用防火墙

>> 阿里云服务器

>> 预发布环境有外网IP, 单点
>> 生产环境入口机有外网IP, 单点；阿里云服务器自建数据库服务内网单点；业务服务内网5台;
>> 弹性公网IP x 1

9)阿里云ecs的网络类型
一共14台阿里云服务器，其中8台式专有网络，6台式经典网络

10)业务架构使用的是spring 还是dubbo
>> 业务架构 SpringBoot + dubbo

11)配置文件如何进行区分
>> 使用基于ZK的配置中心, 项目中仅配置配置中心地址

12)业务是否启用https
>> 预发布环境http 生产环境https

13)ssl证书如何管理
存放在/etc/nginx/cert/目录下

14)nginx配置文件如何管理
生产环境的nginx配置内容主要在nginx.conf里面，部分是在/etc/nginx/conf.d里面

15)dev、test、pro环境如何进行区分
>> 开发环境和测试环境都是在内网物理机上的虚拟机上面的，共用基于zookeeper的配置中心, 私有服务注册中心隔离
>> pre预发布环境是阿里云服务器，专有网络, 与生产环境共用基于zookeeper的配置中心, 私有服务注册中心隔离
>> 生产环境是阿里云服务器, 专有网络, 与预发布环境共用基于zookeeper的配置中心, 私有服务注册中心隔离; 内网5台业务服务 弹性公网IP

16)业务的发布流程如何定义
>> 代码: 开发与测试: 功能分支 > test分支 发布: 功能分支 > pre-release分支
>> 发布: pre预发布环境(自动) > prod生产环境(手动)

17)代码仓库如何管理、权限如何控制
>> 托管于码云(gitee.com), 以项目维度进行人员管理
>> master + pre-release 分支保护, 仅管理员能提交
>> 功能分支 + test分支 + 各个项目develop 分支 未保护, 所有开发人员可提交

18)是否有构建服务器，构建服务器部署的位置
比如：jenkins
jenkins在公司内网环境和生产环境各有一台，
公司内网环境的jenkins主要是为测试和预发布准备的，主要负责后端项目，生产环境的jenkins主要是拉取前端代码

19)公司内网与阿里云如何进行通信
公司内网服务器主要通过dns解析然后直接访问阿里云服务器
>> 基于SSL证书访问 官网 + 生产环境入口机 + 预发布入口机 再以证书访问阿里云内网服务器

20)开发人员是否需要直接访问服务器
>> 目前仅通过 windows 跳板机访问线上数据库(只读账号)

安全建设方案
一、服务器安全
1.1 ssh安全策略(用户、端口、sshkey、尝试策略)
1.2 网络隔离(内外网策略、VPC网络)
1.3 用户隔离
1.4 安全审计
1.5 业务运行的最小化权限
1.6 对外最小化端口映射

二、账号管理安全
2.1 所有运维管理系统使用统一一套账号管理
2.2 根据不同的运维管理系统，使用不同的管理权限
2.3 git使用不同的分组，进行代码隔离
2.4 VPN指定人员才能访问
2.5 数据库访问必须通过数据库审计系统来实现
2.6 服务器访问根据不同人员，来指定不同机器的访问权限

2.7 业务日志查看(前期可以通过登录服务器进行查看，注意权限控制。后期可以直接上ELK)

三、网络访问权限
3.1 所有运维管理系统只能公司内部使用(VPN除外)
3.2 公司外连接相关运维系统，必须通过VPN，才能连接
3.3 公司内网与阿里云内网打通
3.4 公司内部办公网络域开发网络隔离

四、备份相关
4.1 代码备份包含本地与远程备份
4.2 运维相关脚本备份(全部在git仓库)
4.3 运维系统备份(全部在git仓库)
4.4 数据库备份(远程、本地、binlog、增量与冷备)
4.5 账号统一系统备份

五、监控相关
5.1 基本监控
5.2 监控中间件
5.3 监控应用存储状态
5.4 监控报警
5.5 监控API接口

六、使用到的系统
6.1 openldap
6.2 gitlab
6.3 jenkins
6.4 openvpn
6.5 jumpserver
6.6 archer
6.7 zabbix
6.8 ansible

风险点与建议：

1.代码建议使用自建仓库
2.构建相关依赖存放在公司内部(maven\npm)
3.pre和pro中间件分开
4.各个环境使用到的中间件结构，建议与pro保持一致
5.依赖第三方的做白名单控制
6.运维管理账号的审计未做
7.pre和pro入口做成统一的，而且要做成lb的
8.数据库如果没有dba的话，建议使用阿里云的RDS。如果一定要使用ecs的话，备份策略一定要做全。
9.业务建议所有环境都启用https，与pro保持一致
10.jenkins构建机器统一为一台，windows机器剔除

本期问题处理：
1、删除线上服务器用户和组中系统默认不使用的账号包括：lp、mail、games、ftp、nobody、postfix等。删除系统默认不使用的组，包括：mail、games、ftp、nobody、postfix等
2、启动密码策略修改/etc/login.defs来实现
PASS_MAX_DAYS 60 #密码60天过期
PASS_MIN_DAYS 1 #修改密码最小间隔为1天
PASS_MIN_LEN 8 #最短密码要求8位
PASS_WARN_AGE 7 #密码过期前7天内通知用户
3、启用证书登录
4、只是用协议版本2，禁止空口令登录和密码登录
修改/etc/ssh/sshd_config
Protocol 2 #只使用协议版本2
PermitEmptyPasswords no #禁止空口令登录
PasswordAuthentication no #不允许密码登录
5、修改默认端口，例如不使用默认的22端口
6、防火墙设置vim /etc/sysconfig/iptables
允许访问指定端口
-A INPUT -p tcp -m state --state NEW -m tcp --dport 8080 -j ACCEPT 允许访问指定端口
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT 允许访问通过指定端口

-A INPUT -p tcp -m state --state NEW -m tcp --dport 3306 -j ACCEPT 允许访问通过指定端口

7、防止暴力破解ssh密码，通过脚本将连续登陆失败达到10次的用户ip地址加入黑名单，先把始终允许的ip加入/etc/hosts.allow白名单

8、使用最小权限来运行，如果没有要求必须使用root启动的服务，都使用最小权限的用户去启动服务

9、搭建线上环境的跳板机jumpserver，限制所有服务器只允许通过这个跳板机进行访问，由跳板机对所有服务器进行统一管理，划分用户组和用户权限，按照开发、测试、运维的工作需求来分配权限