

解决方案实践

数字政通城市运行管理服务平台解决方案

文档版本 1.0
发布日期 2023-09-15



版权所有 © 华为技术有限公司 2023。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

1 方案概述	1
2 资源和成本规划	5
3 实施步骤	7
3.1 云服务资源准备	7
3.1.1 VPC&安全组	7
3.1.2 OBS	9
3.1.3 RDS	10
3.1.4 DCS-Redis	12
3.1.5 CCE	14
3.1.6 EIP	18
3.1.7 ELB	19
3.1.8 ECS	21
3.2 应用服务部署及配置	24
4 附录	42
5 修订记录	44

1 方案概述

华为云与北京数字政通科技股份有限公司伙伴联合打造的城市运行管理服务平台，本交付指导主要针对综合执法、智慧环卫、市政通用和园林绿化四大功能模块，为城市管理者决策提供技术支撑。

本课程介绍城市运行管理服务平台（web应用）使用华为云服务资源进行上云部署的流程，包含云服务资源以及应用部署配置的详细流程。

整体架构

图 1-1 整体架构



方案描述：以城市运行管理“一网统管”为主要目标，以城市运行、管理、服务为主要内容，以物联网、大数据、人工智能等前沿技术为支撑，具有统筹协调、指挥调度、监测预警、监督考核和综合评价等功能的信息化平台。

方案竞争力：

- 市场占有率高：拥有18项目发明专利和300多项软件著作权，在全国580多个城市、1600多个县成功建设超过5000个数字化平台项目，服务人口数量超过5.5亿人，服务国内城市建成区面积超过70%。
- 行业影响力大：城市运管服平台技术标准主编单位，参与编写数据标准、运行标准、管理标准及建设指南等，对客户项目验收技术解释话语权大。

- 技术领先性强：拥有对占道经营、无证游商等数十种城管违法问题的智能识别算法；物联网云平台支持十万台终端，可高效处理千万级数据传输；基于游戏引擎的晶石CIM平台可快速实现城市的数字化模拟；L4级“棋骥”无人驾驶网格车可24小时对城市运行状况进行巡查。
- 方案覆盖面全：整合市政公用、市容环卫、园林绿化、城管执法等行业应用系统，汇聚行业应用数据，通过全周期、动态化的综合评价，发现城市规划缺项、建设漏项、管理弱项，实现源头治理，实现大城管领域的全业务综合，推动实现城市管理领域一网统管的转型。

部署架构

图 1-2 部署架构

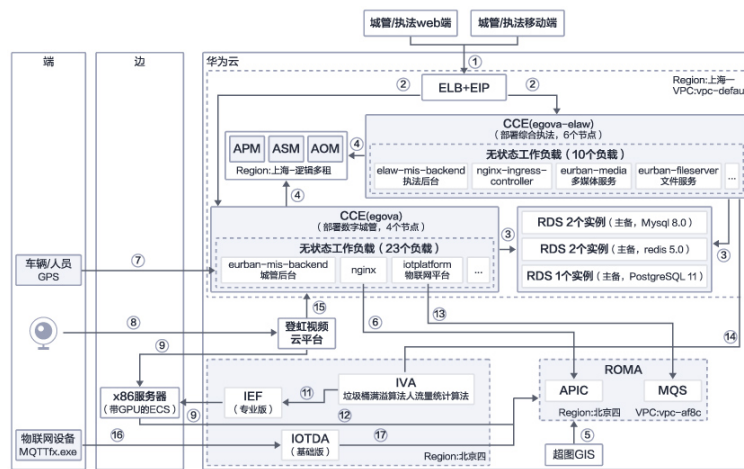


表 1-1 路径说明

路径	说明
①	城管/执法应用web端以及移动端通过ELB访问城管和执法应用，将数据写入RDS（设施数据写入PG库，其他数据都写入RDS（MySQL）），GPS数据写入DCS，通过ASM、AOM、APM进行应用服务的监控及运维
②	
③	
④	
⑤	将超图 GIS接口在ROMA APIC中注册，应用通过nginx访问注册在ROMA中的GIS接口
⑥	
⑦	车辆及人员GPS数据，通过GPS对接子系统写入DCS
⑧	摄像头接入登虹视频云平台
⑨	IEF纳管边缘节点(带GPU的ECS)
⑩	登虹视频云的RTMP视频流接入边缘设备
⑪	IVA将边缘垃圾桶满溢算法和边缘人流量统计算法通过IEF下发至边缘节点，分析接入边缘设备的RTMP视频流

路径	说明
⑫ ⑬	客流统计结果通过ROMA APIC编排的回调地址写入MQS，由政通物联网平台进行消费
⑭	垃圾桶异常监测结果通过视频智能分析平台提供的回调地址发送至视频智能分析平台
⑮	视频中台通过登虹视频云的接口，获取hls直播视频流，用于环卫和园林场景的视频直播
⑯ ⑰ ⑱	使用MQTT客户端模拟物联网设备，对接IoTDA，通过ROMA APIC封装的回调地址，将结果写入到ROMA MQS中，由政通物联网平台进行消费

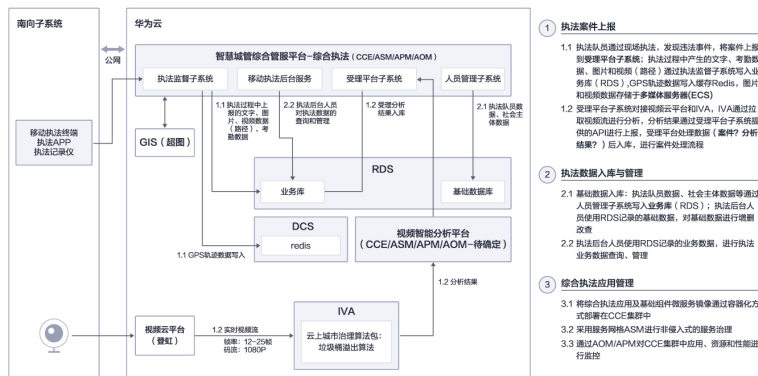
说明

视频云平台部署于IVA测试账号下；GIS部署于ROMA Lab测试账号下。

场景集成架构

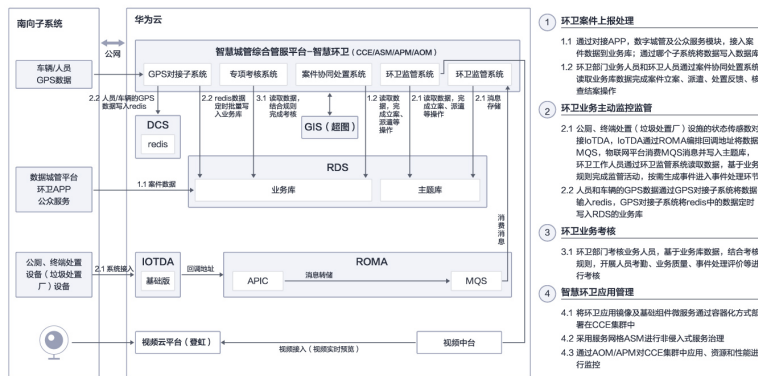
场景一：智慧城管运管服平台-综合执法

图 1-3 场景一



场景二：智慧城管运管服平台-智慧环卫

图 1-4 场景二



2 资源和成本规划

部署资源规格说明：根据上云部署所开通的资源进行罗列（项目中需要根据项目规模进行配置）。

表 2-1 资源和成本规划

序号	云服务	云服务配置	区域	数量
1	云数据库 RDS 1	MySQL 8.0 主备 通用增强型 8核32GB;超高IO 40GB;	华北-北京四	4
2	弹性云服务器 ECS 1	X86计算 通用计算增强型 c3ne.2xlarge.4 8核 32GB;CentOS CentOS 8.0 64bit;通用型SSD 100GB;	华北-北京四	10
3	云数据库 RDS 3	PostgreSQL 12 主备 通用增强型 8核32GB;超高IO 500GB;	华北-北京四	1
4	弹性云服务器 ECS 2	X86计算 通用计算增强型 c3ne.4xlarge.2 16核 32GB;CentOS CentOS 8.0 64bit;高IO 1000GB;	华北-北京四	2
5	对象存储服务 OBS 1	对象存储 标准存储单AZ存储包 2TB;	华北-北京四	1
6	对象存储服务 OBS 2	对象存储 公网流出流量包 10TB;	华北-北京四	1
7	对象存储服务 OBS 3	对象存储 回源流量包 5TB;	华北-北京四	1
8	云容器引擎 CCE 1	CCE容器集群 混合集群 50节点 否;	华北-北京四	1

序号	云服务	云服务配置	区域	数量
9	弹性公网IP 1	独享 全动态BGP 按带宽计费 50Mbit/s x3;3个;	华北- 北京 四	1
10	NAT网关 1	中型;	华北- 北京 四	1
11	企业主机 安全 1	基础版;	华北- 北京 四	3 0
12	Web应用 防火墙 WAF 1	专业版;	华北- 北京 四	1
13	数据库安 全服务 DBSS 1	数据库安全审计 专业版;	华北- 北京 四	2
14	容器安全 服务 1	容器安全企业版;	华北- 北京 四	3
15	IOC视频分 析算法1	云上通用算法包	华北- 北京 四	3

3 实施步骤

- 3.1 云服务资源准备
- 3.2 应用服务部署及配置

3.1 云服务资源准备

3.1.1 VPC&安全组

虚拟私有云（Virtual Private Cloud，简称VPC），为云服务器、云容器、云数据库等资源构建隔离的、用户自主配置和管理的虚拟网络环境，提升用户云上资源的安全性，简化用户的网络部署。详情请参考[帮助文档](#)。后续所有涉及到VPC的资源，都选择本章节开通的VPC，以便资源间内网互通。

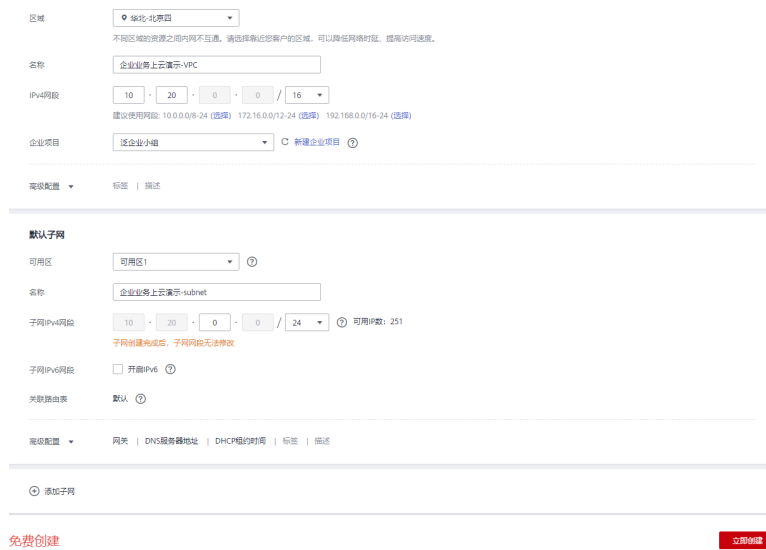
1. 登录华为云，进入网络控制台
2. 在左侧菜单栏选择“虚拟私有云”，点击“创建虚拟私有云”

图 3-1 创建虚拟私有云



- 自定义VPC名称，如“vpc-default”（记住VPC名称，后面创建其他云服务资源时选择该VPC）
- 自定义网段，如192.168.0.0/16（即VPC的地址范围，VPC内的子网地址必须在VPC的地址范围内）
- 选择自己的企业项目（自定义）。具体请参考[企业管理介绍](#)
- 自定义子网名称，如“xxxx- subnet”（记住子网名称，后面创建其他云服务资源时选择该子网）
- 子网网段默认即可（也可以自定义，如10.20.0.0/24，注意子网网段需要在VPC的地址范围内，并且后续资源尽量在同一子网）
- 其他默认，点击“立即创建”，完成VPC的创建。

图 3-2 创建虚拟私有云 2

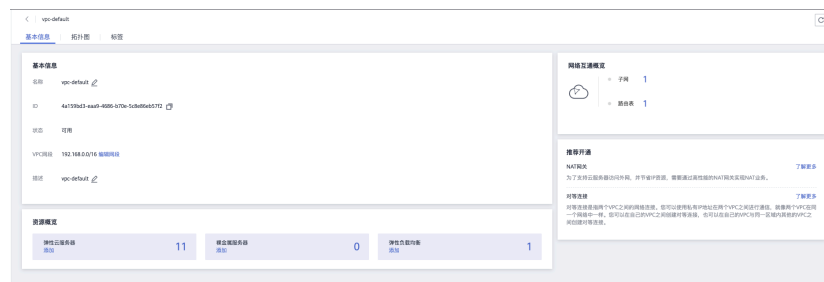


3. 点击vpc名称可查看创建好的VPC详情

图 3-3 VPC 详情 1



图 3-4 VPC 详情 1



4. 在网络控制台左侧菜单栏选择“访问控制”-“安全组”，点击“创建安全组”
- 名称自定义，如：xxxx-sg，
 - 选择自己的企业项目（自定义）。具体请参考[企业管理介绍](#)
 - 模板：选择“通用Web服务器”

图 3-5 创建安全组 1



图 3-6 创建安全组 2



5. 点击安全组名称，可查看基本信息

图 3-7 基本信息 1

名称	安全组规则	关联实例	描述	企业项目	操作
企业业务上云演示-sg	0	0	通用Web服务器，默认放通22、3389、80、443端口和ICMP协议。适用于需要远程登录、公网ping及用于网站服务的云服务器场景。	泛企业小组	配置规则 管理实例 更多

图 3-8 基本信息 2



3.1.2 OBS

对象存储服务（Object Storage Service，OBS）是一个基于对象的海量存储服务，为客户提供海量、安全、高可靠、低成本的数据存储能力。详情参考[帮助文档](#)。本次实践，使用OBS来存储xxxx服务平台上传的图片、文件或者通过接口调用生成的json文件。

1. 登录[华为云OBS管理控制台](#)
2. 点击“创建桶”
 - 桶名：自定义，如“xxxx-obs”（后续章节会用到）
 - 存储类别：标准存储
 - 其他选择默认即可

图 3-9 创建桶 1



图 3-10 创建桶 2

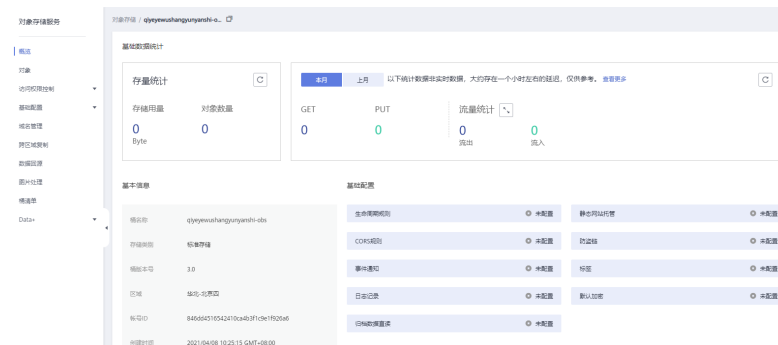


3. 点击创建好的桶名称，可查看桶详情

图 3-11 查看桶详情 1



图 3-12 查看桶详情 2



3.1.3 RDS

一种基于云计算平台的可即开即用、稳定可靠、弹性伸缩、便捷管理的在线关系型数据库服务。RDS 具有完善的性能监控体系和多重安全防护措施，并提供专业的数据库管理平台。详情参考[帮助文档](#)。本次使用RDS（MySQL）数据库作为xxxx服务平台的主数据库。

1. 登录华为云[RDS管理控制台](#)。
2. 单击右上角的“购买数据库实例”按钮。按以下配置进行购买。并设置密码。
 - 计费模式：按需计费
 - 实例名称：“rds-egova”
 - 数据库引擎：MySQL
 - 数据库版本：8.0.20
 - 实例类型：单机

- 存储类型：本地SSD盘
- 性能规格：通用增强 II 型，2 vCPUs | 4 GB
- 存储空间：超高IO 40G
- vpc、子网和安全组选择3.1.1章节创建的
- 密码：xxxx（注意记录好密码，后续章节会用到）

图 3-13 配置 1



图 3-14 配置 2



图 3-15 配置 3

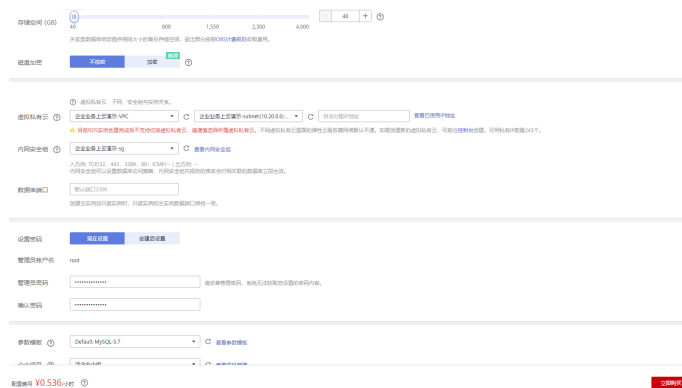


图 3-16 配置 4



图 3-17 配置 5

实例名称	实例备注	实例规格	数据库引擎版本	运行状态	计费模式	内网地址	企业项目	操作
qiyewohangyunguohi-rds		规格 2 vCPUs 4 GB	MySQL 5.7.31	正常	按量计费	10.20...	还企业小账	续费 重置密码 更多

3. 点击RDS实例名称，可查看实例详情

图 3-18 查看实例详情 1

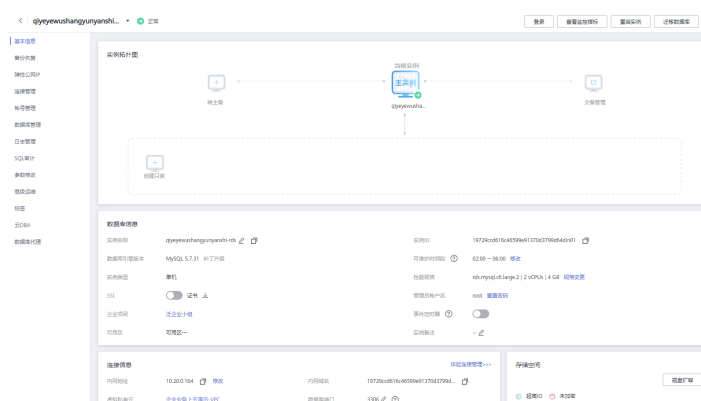


图 3-19 查看实例详情 2

实例名称	实例备注	实例规格	数据库引擎版本	运行状态	计费模式	内网地址	企业项目	操作
qiyewohangyunguohi-rds		规格 2 vCPUs 4 GB	MySQL 5.7.31	正常	按量计费	10.20...	还企业小账	续费 重置密码 更多

3.1.4 DCS-Redis

基于云计算平台的可即开即用、稳定可靠、在线伸缩、便捷管理的分布式Key-Value缓存服务。详情参考[帮助文档](#)。本次使用Redis来缓存xxxx服务平台的用户登录token、往来单位信息等。

1. 登录华为云**DCS管理控制台**。
2. 进入页面后，单击右上角的“购买缓存实例”按钮。按以下配置进行购买。并设置密码。

图 3-20 设置密码



- 计费模式：按需计费
- 缓存类型：Redis
- 产品类型：社区版
- 版本号：5.0
- 实例类型：Cluster集群（由于后续要部署的应用定义好了连接配置，因此必须选择应用定义好的Cluster集群）
- 副本数：1
- 实例规格：redis.cluster.xu1.large.r2.4
- vpc、子网和安全组选择3.1.1章节创建的
- 密码：xxxx（注意记录密码，后续章节会用到）
- 名称：自定义，如“xxxx-dcs”

图 3-21 配置 1

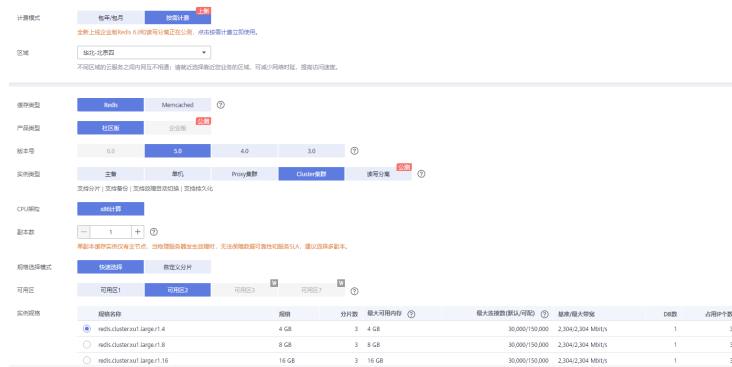


图 3-22 配置 2

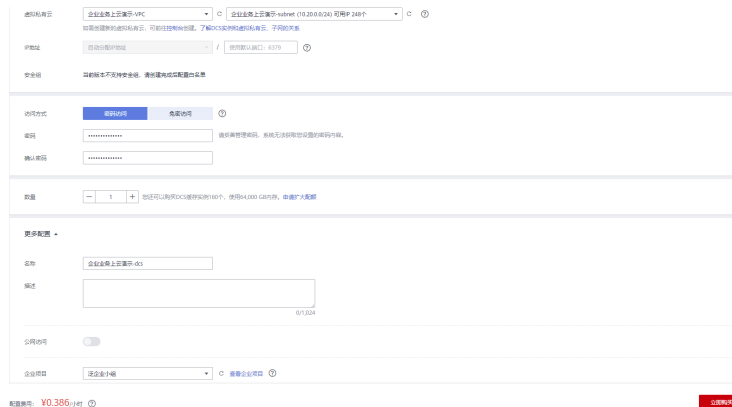


图 3-23 配置 3



图 3-24 配置 4

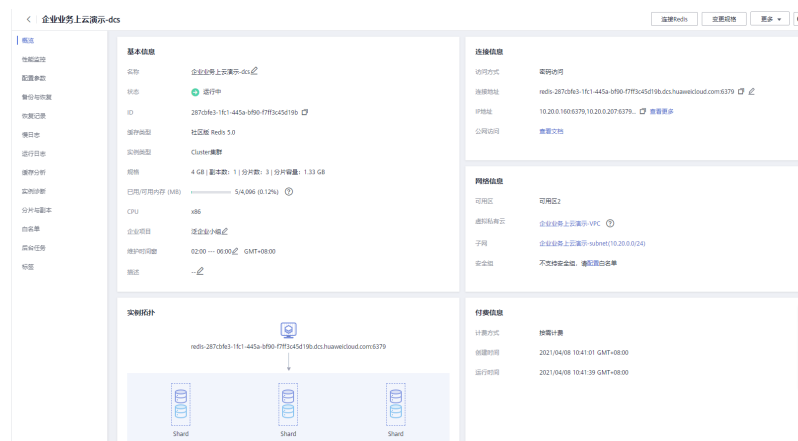


3. 点击Redis名称，可查看Redis详情

图 3-25 查看 Redis 详情 1



图 3-26 查看 Redis 详情 2



3.1.5 CCE

CCE是一种托管的Kubernetes产品/服务，可进一步简化基于容器的应用程序部署和管理，可以在CCE中方便的创建Kubernetes集群、部署容器化应用，以及方便的管理和维护。详情参考[帮助文档](#)。本次使用CCE来部署xxxx服务的后端应用。

1. 登录华为云[云容器引擎CCE 控制台](#)
2. 点击购买Kubernetes集群->购买CCE集群

图 3-27 购买 CCE 集群 1



图 3-28 购买 CCE 集群 2



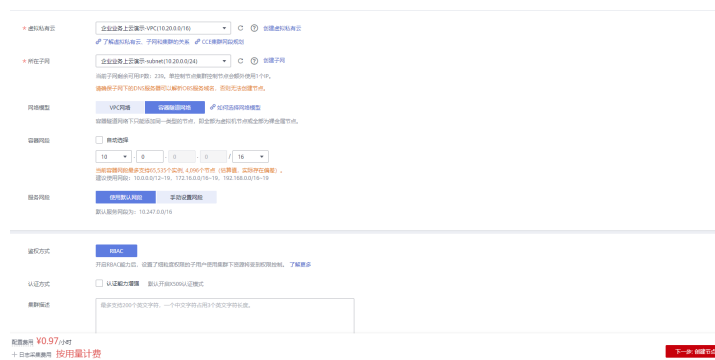
a. 服务选型

- 计费模式：按需计费
- 集群名称：自定义，如“xxxx-cce”
- 版本：v1.17.17
- 集群管理规模：80节点
- 控制节点数：1
- 网络模型：容器隧道网络
- vpc、子网选择3.1.1章节创建的

图 3-29 服务选型 1



图 3-30 服务选型 2



b. 创建节点

- 计费模式：按需计费
- 节点名称：默认
- 节点规格：通用型 sn3.xlarge.2 | 4核 | 8 GB
- 操作系统：CentOS 7.6
- 系统盘：高IO 40G
- 数据盘：高IO 100G
- 子网选择3.1.1章节创建的
- 密码：xxxx（注意记录密码，后续章节会用到）

图 3-31 创建节点 1

The screenshot shows the 'Create Node' configuration page. At the top, the 'Billing Mode' is set to 'Pay-as-you-go' (按需计费). The 'Region' is 'China North-2' (华北-2). Under 'Availability Zone', 'Zone 1' (可用区1) is selected. The 'Node Type' is 'General Purpose' (通用型). The 'Node Name' is 'sjyewshangzongshibh-oc-11297'. The 'vCPUs' are set to 'All' (全部) and 'Memory' to 'All' (全部). A table lists various instance specifications:

规格名称	CPU/内存	网络/最大带宽	内网收发包
sn3.large.2	2核 4GB	0.35/1.5 Gbit/s	150,000 pps
sn3.large.4	2核 8GB	0.35/1.5 Gbit/s	150,000 pps
sn3.xlarge.2	4核 8GB	0.7/2 Gbit/s	250,000 pps
sn3.xlarge.4	4核 16GB	0.7/2 Gbit/s	250,000 pps
sn3.2xlarge.2	8核 16GB	1.3/3 Gbit/s	500,000 pps
sn3.2xlarge.4	8核 32GB	1.3/3 Gbit/s	500,000 pps

图 3-32 创建节点 2

The screenshot shows the 'Create Node' configuration page for OS and disks. The 'Operating System' is 'CentOS 7.6'. A warning message states: 'k8s软件依赖系统内核版本，请不要使用yum update更新节点的操作系统内核'. The 'System Disk' is 'High IO (5,000 IOPS)' with a size of 40 GB. The 'Data Disk' is 'High IO (5,000 IOPS)' with a size of 100 GB and 'LVM Management' (LVM管理) is checked. The 'Disk Space Allocation' section shows: Docker占90%, Kubelet占10%, 用户空间占0%. A warning message states: '开启LVM管理的数据盘将按照设置的比例进行统一分配'. The 'Virtual Private Cloud' is 'Enterprise Cloud on VPC'. The 'Subnet' is 'Enterprise Cloud on VPC - subnet (集群子网) (10.20...)' with a 'Create Subnet' button. A warning message states: '温馨提示：子网网段不能与容器网段(10.0.0.0/16)/Kubernetes默认服务网段(10.247.0.0/16)冲突，请确保子网下的DNS服务器可以解析OBS服务域名，否则无法创建节点。'

图 3-33 创建节点 3



c. 插件默认即可

图 3-34 插件默认即可 1

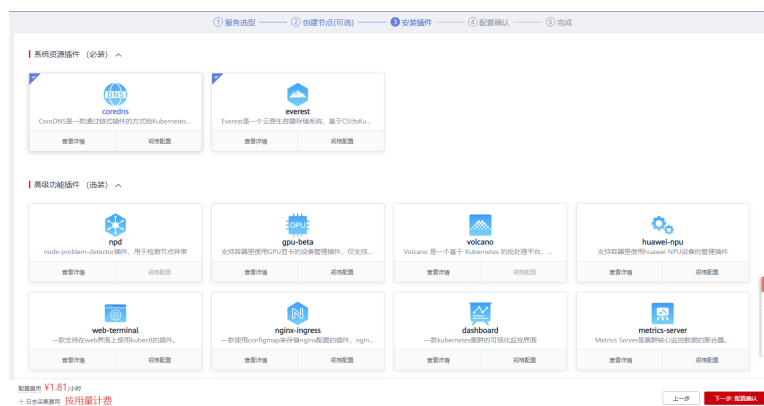
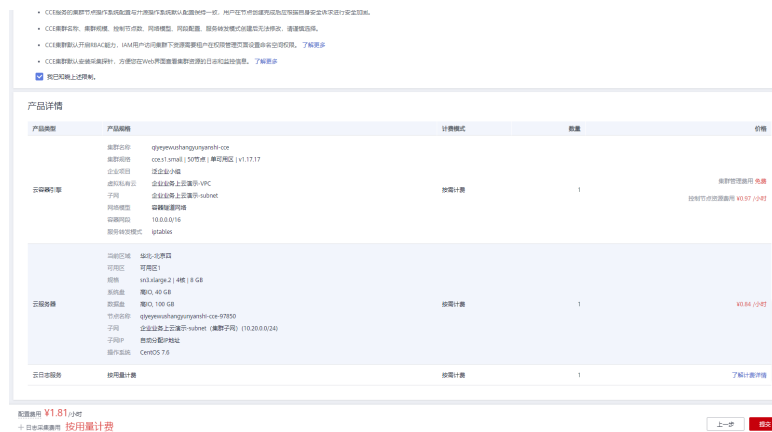
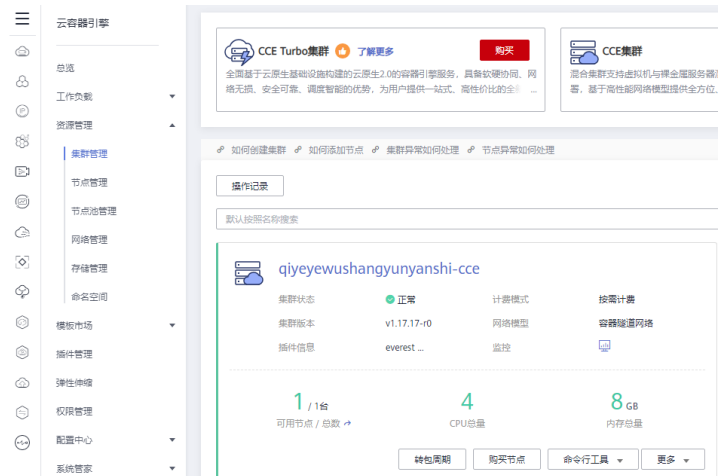


图 3-35 插件默认即可 2



3. 点击资源管理->集群管理，可查看集群详情

图 3-36 查看集群详情 1



4. 点击资源管理->节点管理，选择当前集群，可查看节点详情

图 3-37 查看集群详情 2



3.1.6 EIP

弹性公网IP（Elastic IP，简称EIP）提供独立的公网IP资源，包括公网IP地址与公网出口带宽服务。可以与弹性云服务器、裸金属服务器、虚拟IP、弹性负载均衡、NAT网关等资源灵活地绑定及解绑。详情参考[帮助文档](#)。

1. 登录华为云[网络控制台](#)
2. 点击“弹性公网IP和带宽”->“弹性公网IP”，点击右上角的“购买弹性公网IP”

图 3-38 购买弹性公网 IP1



- 计费模式：按需计费
- 线路：全动态BGP
- 公网带宽：按流量计费
- 带宽大小：100
- 购买量：2

图 3-39 购买弹性公网 IP2



图 3-40 购买弹性公网 IP3



图 3-41 购买弹性公网 IP4



图 3-42 购买弹性公网 IP5



3.1.7 ELB

将访问流量根据分配策略分发到后端多台服务器的流量分发控制服务。弹性负载均衡可以通过流量分发扩展应用系统对外的服务能力，同时通过消除单点故障提升应用系统的可用性。详情参考[帮助文档](#)。本次使用ELB作为智慧城管运营服务平台所有请求的流量入口，通过ELB上的监听器来转发请求到相应的服务做处理。

1. 登录华为云**网络控制台**
2. 在左侧菜单栏选择“弹性负载均衡”-“负载均衡器”，点击购买弹性负载均衡按如下配置购买即可。

图 3-43 购买 1



- vpc、子网和安全组选择**3.1.1章节**创建的
- 弹性公网IP选择**3.1.6 EIP**创建的其中一个
- 名称：自定义，如“xxxx-ELB”

图 3-44 购买 2

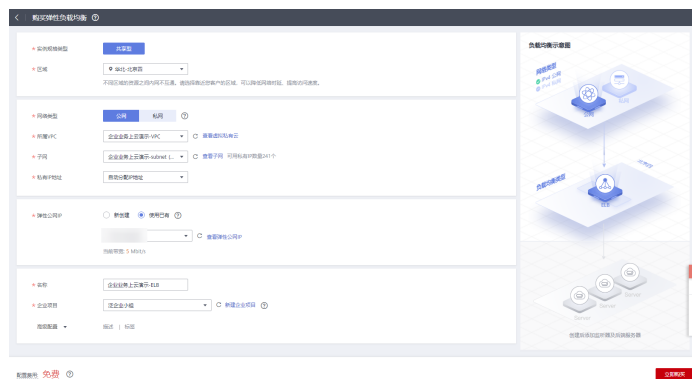


图 3-45 购买 3

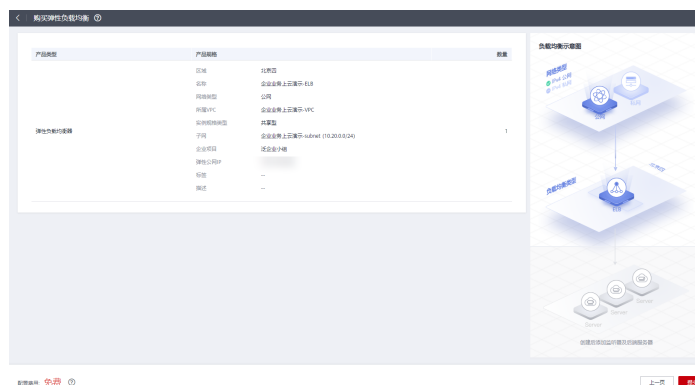


图 3-46 购买 4

名称	状态	实例规格类型	服务地址与所属网络	监听器 (前端协议/端口)	EIP计费规格	计费模式	企业项目	操作
企业云上-	运行中	共享型	19.206.0.9 (EIP私有IP) 企业云上-子网-VPC (企业云上-)	点我开始配置	--	--	企业云上-小	查看实例 删除 更多

3. 点击名称，可查看详情

图 3-47 查看详情 1



图 3-48 查看详情 2

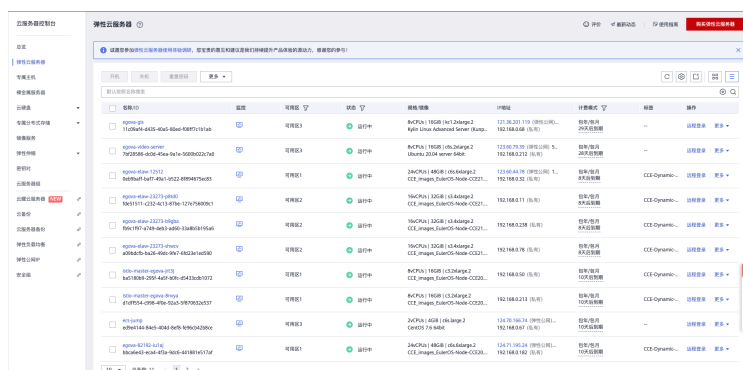


3.1.8 ECS

弹性云服务器（Elastic Cloud Server, ECS）是由CPU、内存、操作系统、云硬盘组成的基础的计算组件。弹性云服务器创建成功后，您就可以像使用自己的本地PC或物理服务器一样，在云上使用弹性云服务器。详情参考[帮助文档](#)。本次使用ECS来安装部署智慧城管管服服务平台的静态Nginx，也就是前端服务部署。

1. 登录华为云[ECS管理控制台](#)，点击“购买弹性云服务器”

图 3-49 购买弹性云服务器



2. 基础配置

- 计费模式：按需计费
- 规格：通用计算增强型 c6.xlarge.2 | 2核 | 4GB
- 镜像：centos7.6
- 系统盘：高IO 40GB
- 购买量：1

图 3-50 基础配置 1

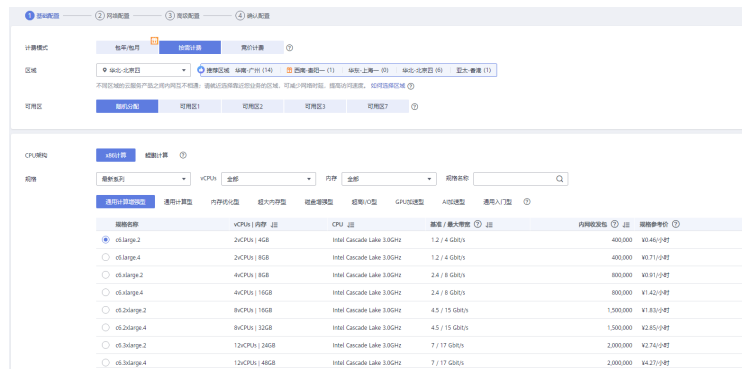


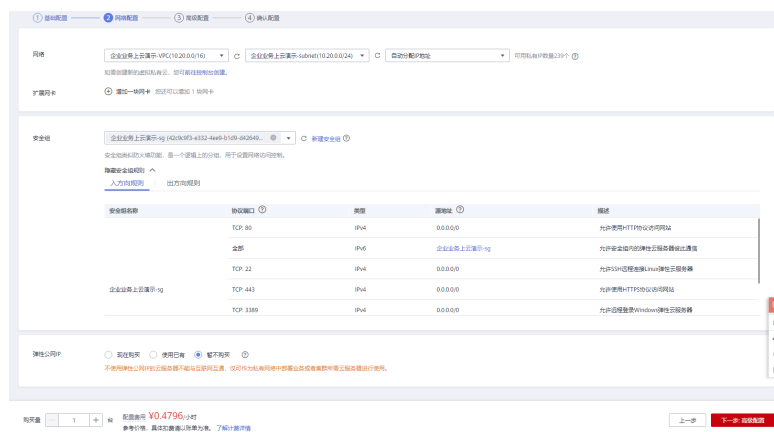
图 3-51 基础配置 2



3. 网络配置

- 网络：选择VPC&安全组中创建的VPC和子网
- 安全组：选择VPC&安全组中创建的安全组

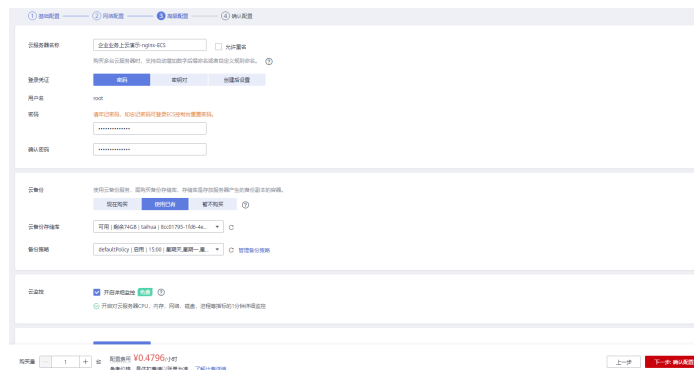
图 3-52 网络配置



4. 高级配置

- 云服务器名称：自定义，如“xxxx-nginx-ECS”
- 密码：xxxx（注意记录密码，后续章节会用到）
- 购买量：1
- 其他都保持默认即可

图 3-53 高级配置



5. 确认配置

图 3-54 确认配置 1

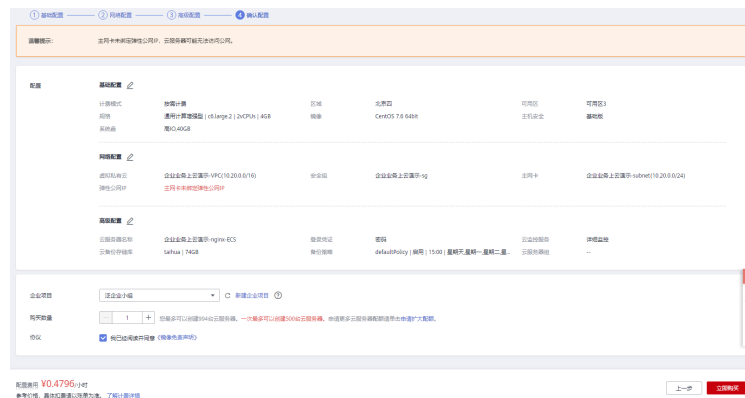


图 3-55 确认配置 2

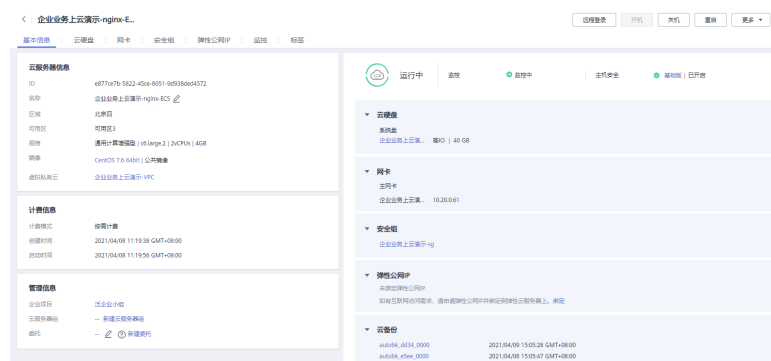
名称/ID	监控	可用区	状态	规格/镜像	IP地址	计费模式	企业项目	标签	操作
企业云上云演示-nginx-ECS e877e7b7-5822-45ce-8051-9d...		可用区3	运行中	2vCPU 4GB CentOS 7.6 64bit	10.20.0.61 (5.1)	按量计费 2021/04/08 11:...	企业云上云...	--	查看详情 >

6. 点击名称，可查看详情

图 3-56 查看详情 1

名称/ID	监控	可用区	状态	规格/镜像	IP地址	计费模式	企业项目	标签	操作
企业云上云演示-nginx-ECS e877e7b7-5822-45ce-8051-9d...		可用区3	运行中	2vCPU 4GB CentOS 7.6 64bit	10.20.0.61 (5.1)	按量计费 2021/04/08 11:...	企业云上云...	--	查看详情 >

图 3-57 查看详情 2



3.2 应用服务部署及配置

镜像上传

1. 登录华为云**SWR管理控制台**，进入组织管理，点击创建组织

图 3-58 创建组织



组织名称: xxxx

图 3-59 组织名称 1

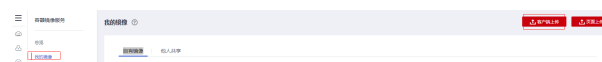


图 3-60 组织名称 2



2. 登录华为云**SWR管理控制台**，进入我的镜像->自有镜像，点击客户端上传

图 3-61 上传



点击“生成临时登录指令”

图 3-62 生成临时登录指令



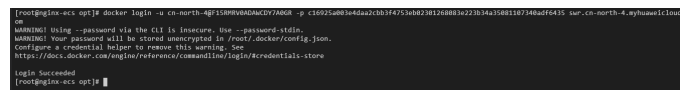
复制登录命令

图 3-63 复制登录命令



3. 在安装容器引擎的机器中执行上一步复制的登录指令。登录成功会显示“login succeeded”。

图 3-64 登录



4. 在安装容器引擎的机器给镜像打标签
docker pull [镜像名称]:[版本名称]
docker tag [镜像名称]:[版本名称] swr.cn-east-3.myhuaweicloud.com/{组织名称}/{镜像名称}:[版本名称]

样例如下:

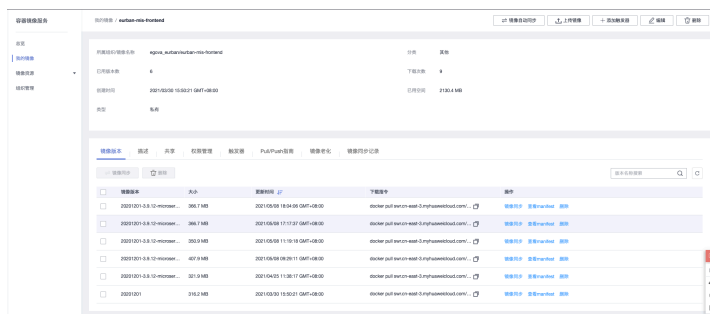
```
docker pull hub.egova.com.cn/microservice/urban-mis-frontend-library:20201201-3.9.12-microservice-t04251036  
docker tag hub.egova.com.cn/microservice/urban-mis-frontend-library:20201201-3.9.12-microservice-t04251036 swr.cn-east-3.myhuaweicloud.com/egova/urban/urban-mis-frontend-library:20201201-3.9.12-microservice-t04251036
```

图 3-65 打标签

```
[root@ecs-jump ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED            SIZE
swr.cn-east-3-myhuaweicloud.com/egova_eurban/eurban-mis-frontend    20201201-3.9.12-microservice-t04251036    70882426671a    12 days ago    390MB
hub.egova.com.cn/microservice/eurban-mis-frontend-library          20201201-3.9.12-microservice-t04251036    70882426671a    12 days ago    390MB
hub.egova.com.cn/microservice/eurban-mis-frontend-library          20201201-3.9.12-microservice-t04251036    70882426671a    12 days ago    390MB
swr.cn-east-3-myhuaweicloud.com/egova_eurban/eurban-mis-backend    20201201-3.9.12-microservice-t04251036    658243400834    12 days ago    288MB
swr.cn-east-3-myhuaweicloud.com/egova_eurban/eurban-mis-backend    20201201-3.9.12-microservice-t04251036    658243400834    12 days ago    288MB
hub.egova.com.cn/microservice/eurban-mis-backend                  20201201-3.9.12-microservice-t04251036    658243400834    12 days ago    288MB
hub.egova.com.cn/microservice/eurban-mis-backend                  20201201-3.9.12-microservice-t04251036    658243400834    12 days ago    288MB
swr.cn-east-3-myhuaweicloud.com/egova_eurban/eurban-mis-frontend-library    20201201-3.9.12-microservice-t04251036    8879ed7a5bc9    13 days ago    99.2MB
hub.egova.com.cn/microservice/eurban-mis-frontend-library        20201201-3.9.12-microservice-t04251036    8879ed7a5bc9    13 days ago    99.2MB
hub.egova.com.cn/microservice/eurban-mis-frontend-library        20201201-3.9.12-microservice-t04251036    8879ed7a5bc9    13 days ago    99.2MB
swr.cn-east-3-myhuaweicloud.com/egova_eurban/eurban-mis-backend    20201201-3.9.12-microservice-t04251036    336304e29c7c    13 days ago    288MB
swr.cn-east-3-myhuaweicloud.com/egova_eurban/eurban-mis-backend    20201201-3.9.12-microservice-t04251036    336304e29c7c    13 days ago    288MB
hub.egova.com.cn/microservice/eurban-mis-backend                  20201201-3.9.12-microservice-t04251036    336304e29c7c    13 days ago    288MB
hub.egova.com.cn/microservice/eurban-mis-backend                  20201201-3.9.12-microservice-t04251036    336304e29c7c    13 days ago    288MB
swr.cn-east-3-myhuaweicloud.com/egova_eurban/eurban-mis-frontend-library    20201201-3.9.12-microservice-t04251036    345089c139e    13 days ago    379MB
hub.egova.com.cn/microservice/eurban-mis-frontend-library        20201201-3.9.12-microservice-t04251036    345089c139e    13 days ago    379MB
hub.egova.com.cn/microservice/eurban-mis-frontend-library        20201201-3.9.12-microservice-t04251036    345089c139e    13 days ago    379MB
swr.cn-east-3-myhuaweicloud.com/egova_eurban/eurban-mis-frontend-library    20201201-3.9.12-microservice-t04251036    12e6823d3591    13 days ago    99.1MB
swr.cn-east-3-myhuaweicloud.com/egova_eurban/eurban-mis-frontend-library    20201201-3.9.12-microservice-t04251036    12e6823d3591    13 days ago    99.1MB
hub.egova.com.cn/microservice/eurban-mis-frontend-library        20201201-3.9.12-microservice-t04251036    12e6823d3591    13 days ago    99.1MB
hub.egova.com.cn/microservice/eurban-mis-frontend-library        20201201-3.9.12-microservice-t04251036    12e6823d3591    13 days ago    99.1MB
```

5. 上传镜像至镜像仓库
docker push [镜像仓库地址]/[组织名称]/[镜像名称:版本名称]
样例如下:
docker push swr.cn-east-3-myhuaweicloud.com/egova_eurban/eurban-mis-frontend-library:20201201-3.9.12-microservice-t04251036
6. 登录华为云SWR管理控制台，进入我的镜像->自有镜像，查看镜像已上传

图 3-66 查看镜像



安装集群

以下安装步骤仅供参考。
建议：安装前对默认配置进行确认，配置文件位于/etc/ansible/hosts
主要确认的配置有：



集群网络参数：如果现场服务器用到了10.88以及172.88这两个网段，请将此处修改为其他不重复的网段

图 3-67 确认配置 1

```
# 服务网段 (Service CIDR)
SERVICE_CIDR="10.88.0.0/16"

# POD 网段 (Cluster CIDR)
CLUSTER_CIDR="172.88.0.0/16"

# 服务端口范围 (NodePort Range)
NODE_PORT_RANGE="5000-40000"

# kubernetes 服务 IP (预分配，一般是 SERVICE_CIDR 中第一个 IP)
CLUSTER_KUBERNETES_SVC_IP="10.88.0.1"

# 集群 DNS 服务 IP (从 SERVICE_CIDR 中预分配)
CLUSTER_DNS_SVC_IP="10.88.0.2"

# 集群 DNS 域名
CLUSTER_DNS_DOMAIN="cluster.local."
```

服务器部署app参数：如果服务器数量较多(10台左右，如下三个参数请使用默认值no)

图 3-68 确认配置 2

```
# -----数据库节点是否可部署应用(建议不部署)-----  
DB_NODE_DEPLOY_APP=no  
  
##---测试节点是否允许部署正式服务(建议不部署)-----##  
TEST_NODE_DEPLOY_PROD=no  
  
# -----master节点是否可部署应用(建议不部署)-----  
MASTER_NODE_DEPLOY_APP=no
```

日志采集与监控相关参数（请参考后续章节，设置smtp邮件服务参数）。另外，如果服务器资源比较紧张，可以不启用日志采集服务和监控服务（这两个服务耗费内存较多）

图 3-69 确认配置 3

```
##-----监控服务-----##  
ENABLE_PROMETHEUS_MONITOR=1  
SMTP_HOST=smtp.mxhichina.com:465  
SMTP_FROM=dev@egova.com.cn  
SMTP_USERNAME=dev@egova.com.cn  
SMTP_PASSWORD="your_password"  
SMTP_TLS=false  
SMTP_RECIEVER=fuzhanli@egova.com.cn  
SMTP_PROJECT_NAME="your_project_name"  
  
##---核心服务启用控制---##  
#是否启动日志搜集服务  
ENABLE_APP_LOG_COLLECTION=yes  
#是否启动内置监控服务  
ENABLE_APP_PROMETHEUS=yes
```

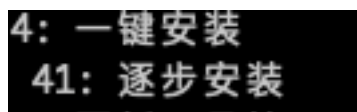
服务器较多(10+)，且资源充足（cpu,内存足够）时，可以设置多主+多etcd方式，保证集群关键服务的稳定性。其中etcd需要为奇数个

图 3-70 确认配置 4

```
# etcd集群需提供如下NODE_NAME，请注意etcd集群必须是1,3,5,7...奇数个节点  
[etcd]  
192.168.101.99 NODE_NAME=etcd1  
192.168.101.98 NODE_NAME=etcd2  
192.168.101.18 NODE_NAME=etcd3  
  
[kubernetes-master]  
192.168.101.99  
192.168.101.98
```

安装集群有两种方式，一种是运行4，直接一键全部安装。另一种是运行41，逐步进行安装。

图 3-71 安装方式



以41逐步安装进行说明，安装时需要注意哪些细节

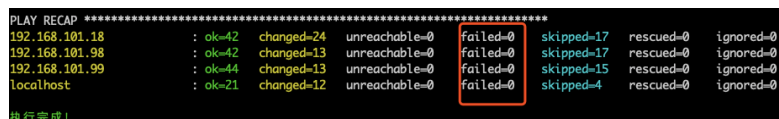
图 3-72 41



- a. 所有分步步骤均可重复运行
- b. 每步运行后，不允许出现failed的服务器（failed且非ignored的步骤，需要对安装日志进行分析，解决后，再次运行该步骤）

正常结果如下：

图 3-73 结果



异常结果如下：

图 3-74 异常结果



- c. 离线镜像约有6GB，导入耗时较多（10分钟级别），请耐心等待

图 3-75 离线镜像



集群安装成功后：
通过kubectl get node可以查看集群所有节点

图 3-76 查看集群所有节点

```
[root@cg58 ~]# kubectl get node
NAME                 STATUS    ROLES    AGE   VERSION
192.168.0.50         Ready    node     78d   v1.15.0
192.168.0.51         Ready    node     78d   v1.15.0
192.168.0.52         Ready    node     78d   v1.16.2
192.168.0.53         Ready    node     78d   v1.15.0
192.168.0.54         Ready    node     78d   v1.15.0
192.168.0.55         Ready    node     78d   v1.15.0
192.168.0.56         Ready    node     78d   v1.15.0
192.168.0.57         Ready    node     78d   v1.16.2
192.168.0.58         Ready    master   78d   v1.15.0
192.168.0.70         Ready    node     78d   v1.15.0
```

Ready代表节点已加入集群
从节点
主节点
主节点默认不允许部署城管服务

城管服务及集群维护服务都是以pod容器组方式运行在集群中，通过get pod -A可查看所有pod

图 3-77 查看所有的 pod

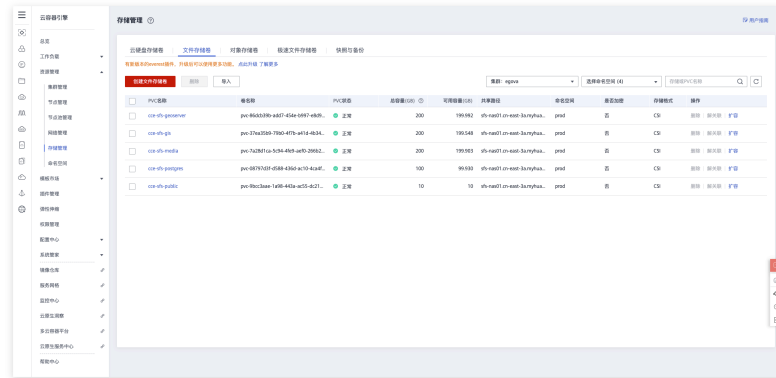
```
[root@cg58 ~]# kubectl get pod -A
NAMESPACE   NAME                                     READY   STATUS    RESTARTS   AGE
egova-log   filebeat-4fp7z                         1/1     Running   1           65d
egova-log   filebeat-54fhd                         1/1     Running   6           65d
egova-log   filebeat-8hcfh                         1/1     Running   12          65d
egova-log   filebeat-fhh4h                         1/1     Running   317         13d
egova-log   filebeat-hw5w4                         1/1     Running   5           65d
egova-log   filebeat-jv82g                         1/1     Running   5           65d
egova-log   filebeat-pwdjt                         1/1     Running   2           65d
egova-log   filebeat-ppq3m                         1/1     Running   7           65d
egova-log   filebeat-s9wc5                         1/1     Running   8           65d
egova-log   filebeat-vqw8b                         1/1     Running   12          44d
egova-log   log-es-0                                1/1     Running   1           6d14h
egova-log   log-kibana-5c6c99575b-msr7n            1/1     Running   0           3d16h
egova-monitor grafana-6d7fb6877c-6lnkc              0/1     MatchNodeSelector 0           22d
egova-monitor grafana-6d7fb6877c-ndzbr              1/1     Running   0           3d23h
egova-monitor node-exporter-46lbg                   1/1     Running   12          71d
egova-monitor node-exporter-8j5cm                   1/1     Running   1           71d
egova-monitor node-exporter-cv57m                   1/1     Running   23          71d
egova-monitor node-exporter-fhgn8                   1/1     Running   5           71d
egova-monitor node-exporter-jmx6j                   1/1     Running   7           71d
egova-monitor node-exporter-npdg7                   1/1     Running   3           71d
egova-monitor node-exporter-pmwqv                   1/1     Running   36          71d
egova-monitor node-exporter-rlcz5                   1/1     Running   5           71d
egova-monitor node-exporter-wprtq                   1/1     Running   8           71d
egova-monitor node-exporter-z4ld2                   1/1     Running   6           71d
ingress-nginx nginx-ingress-controller-6fd7dcd96d-rf7m2 1/1     Running   0           126m
ingress-nginx nginx-ingress-controller-6fd7dcd96d-z79rd 1/1     Running   1           22d
kube-system calico-kube-controllers-6554b7d9c-xlqg2 1/1     Running   2           10d
kube-system calico-node-5f79x                   1/1     Running   1           13d
kube-system calico-node-68kcx                   1/1     Running   2           13d
kube-system calico-node-76hwh                   1/1     Running   1           13d
```

应用服务部署

1. 创建存储卷

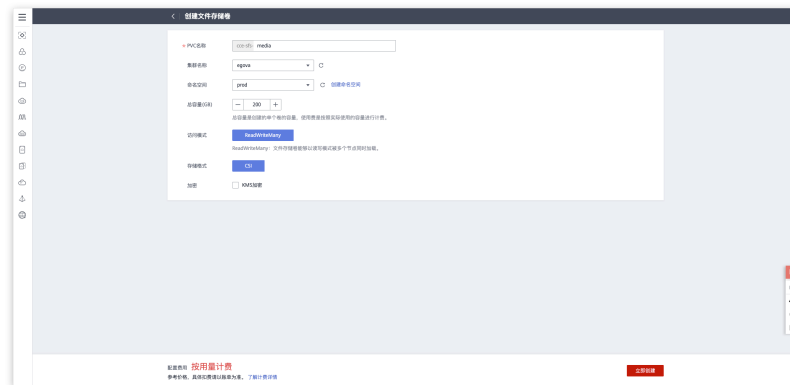
进入云容器引擎下的“资源管理 > 存储管理”，根据应用服务需求创建所需存储卷。

图 3-78 创建存储卷



点击“创建文件存储卷”，按需申请容量大小

图 3-79 创建存储卷



需要申请的存储卷如下：

- geoserver：用于存储geoserver数据，主要是data目录的数据，多个geoserver服务统一共享使用同一个pv数据目录，容量大小200G即可。
- gis：用于存储eUrbanGIS数据，主要是data/data3d/tilemap三个目录下的数据，多个gis服务统一共享使用同一个pv数据目录，容量大小200G即可。
- media：用于存储MediaRoot数据，可同时给urban-fileserver服务（多媒体上传接口）和urban-media服务（多媒体发布服务）使用，容量大小200G即可。
- public：主要用于挂在微信证书文件，约定目录为/etc/public/data，请确保wechat.properties使用此目录，容量大小为10G即可。
- postgres：用于postgre数据库物理存储，容量大小100G即可。

2. Service定义

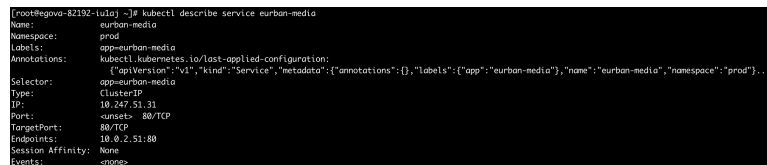
拿urban-mis-backend服务为例，定义如下：

```
Name: urban-mis-backend
Namespace: prod
Labels: app=urban-mis-backend
Annotations: asm.huaweicloud.com/put:
{"kind":"Service","apiVersion":"v1","metadata":{"name":"urban-mis-backend","namespace":"prod","selfLink":"/api/v1/namespaces/prod/service...
asm.huaweicloud.com/updateTimestamp: 2021-04-25T09:56:38Z
kubectl.kubernetes.io/last-applied-configuration:
{"apiVersion":"v1","kind":"Service","metadata":{"annotations":{"prometheus.io/jvm-path":"/
```

```
metrics","prometheus.io/jvm-port":"1234","promet...
prometheus.io/jvm-path: /metrics
prometheus.io/jvm-port: 1234
prometheus.io/jvm-scrape: true
prometheus.io/scrape: true
Selector:      app=urban-mis-backend
Type:         ClusterIP
IP:           10.247.194.115
Port:         http-tomcat 8080/TCP
TargetPort:   8080/TCP
Endpoints:    10.0.2.18:8080
Port:         http-prometheus 1234/TCP
TargetPort:   1234/TCP
Endpoints:    10.0.2.18:1234
Session Affinity: None
Events:       <none>
```

其他Service定义可根据以下命令进行查看

图 3-80 查看



```
[root@egova-82192-lu1n] ~# kubectl describe service urban-media
Name:         urban-media
Namespace:    prod
Labels:       app=urban-media
Annotations:  kubectl.kubernetes.io/last-applied-configuration:
              {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"app":"urban-media"},"name":"urban-media","namespace":"prod"}...
Selector:    app=urban-media
Type:        ClusterIP
IP:          10.247.51.31
Port:        <unset> 80/TCP
TargetPort:  80/TCP
Endpoints:   10.0.2.51:80
Session Affinity: None
Events:      <none>
```

3. Pod定义

拿urban-mis-backend为例，定义如下：

```
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "3"
    kubectl.kubernetes.io/last-applied-configuration: |
      creationTimestamp: "2021-05-08T06:22:15Z"
  generation: 3
  labels:
    app: urban-mis-backend
    name: urban-mis-backend
  namespace: prod
  resourceVersion: "31294043"
  selfLink: /apis/apps/v1/namespaces/prod/deployments/urban-mis-backend
  uid: 34de909e-1187-4faf-b7cd-5156eb505737
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: urban-mis-backend
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
      type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
    labels:
      app: urban-mis-backend
      app-group: urban-mis-backend
      version: v1
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
```

```
nodeSelectorTerms:
- matchExpressions:
- key: app.egova.com.cn/common
operator: In
values:
- "true"
- key: namespace.egova.com.cn/prod
operator: In
values:
- "true"
- key: app.egova.com.cn/tomcat-large
operator: NotIn
values:
- "false"
podAntiAffinity:
preferredDuringSchedulingIgnoredDuringExecution:
- podAffinityTerm:
labelSelector:
matchExpressions:
- key: app-group
operator: In
values:
- urban-mis-backend
topologyKey: kubernetes.io/hostname
weight: 10
containers:
- env:
- name: __ENV__MYSQL_SERVER_BIZ_HOST
valueFrom:
configMapKeyRef:
key: db_host_biz
name: global-configmap
- name: __ENV__MYSQL_SERVER_BIZ_PORT
valueFrom:
configMapKeyRef:
key: db_port_biz
name: global-configmap
- name: __ENV__DB_BIZ_USER
valueFrom:
secretKeyRef:
key: db_user_biz
name: global-secret
- name: __ENV__DB_BIZ_PASSWORD
valueFrom:
secretKeyRef:
key: db_passwd_biz
name: global-secret
- name: __ENV__DB_BIZ_NAME
valueFrom:
configMapKeyRef:
key: db_name_biz
name: global-configmap
- name: __ENV__MYSQL_SERVER_STAT_HOST
valueFrom:
configMapKeyRef:
key: db_host_stat
name: global-configmap
- name: __ENV__MYSQL_SERVER_STAT_PORT
valueFrom:
configMapKeyRef:
key: db_port_stat
name: global-configmap
- name: __ENV__DB_STAT_USER
valueFrom:
secretKeyRef:
key: db_user_stat
name: global-secret
- name: __ENV__DB_STAT_PASSWORD
valueFrom:
```

```
secretKeyRef:
key: db_passwd_stat
name: global-secret
- name: __ENV__DB_STAT_NAME
valueFrom:
configMapKeyRef:
key: db_name_stat
name: global-configmap
- name: __ENV__ZOOKEEPER_HOST
value: zookeeper.prod
- name: __ENV__ZOOKEEPER_PORT
value: "2181"
- name: __ENV__REDIS_SERVER_HOST
valueFrom:
configMapKeyRef:
key: redis_host
name: global-configmap
- name: __ENV__REDIS_SERVER_PORT
valueFrom:
configMapKeyRef:
key: redis_port
name: global-configmap
- name: __ENV__REDIS_SERVER_PASSWORD
valueFrom:
secretKeyRef:
key: redis_passwd
name: global-secret
- name: __ENV__REDIS_NAMESPACE
value: egova
- name: __ENV__POSTGRES_SERVICE_HOST
valueFrom:
configMapKeyRef:
key: pg_host
name: global-configmap
- name: __ENV__POSTGRES_SERVICE_PORT
valueFrom:
configMapKeyRef:
key: pg_port
name: global-configmap
- name: __ENV__POSTGRES_SERVICE_USER
valueFrom:
secretKeyRef:
key: pg_user
name: global-secret
valueFrom:
configMapKeyRef:
key: pg_name
name: global-configmap
- name: USE_MYSQL_CONNECTOR_PATCH
value: "1"
- name: REDIS_SESSION_ENABLED
value: "1"
- name: __ENV__WITH_JOB_MANAGER
value: "0"
- name: __ENV__CONTEXT_PATH
value: eUrbanMIS
- name: POD_NAME
valueFrom:
fieldRef:
apiVersion: v1
fieldPath: metadata.name
- name: JAVA_OPTS
image: swr.cn-east-3.myhuaweicloud.com/egova_eurban/tomcat-redis:2.3
imagePullPolicy: IfNotPresent
lifecycle:
postStart:
exec:
command:
- bash
```

```
- -c
- |
set -ex
echo $(date +%s) > /etc/inited
preStop:
exec:
command:
- bash
- -c
- |
set -ex
rm -f /etc/started
livenessProbe:
failureThreshold: 12
httpGet:
path: /eUrbanMIS/main.htm
port: 8080
scheme: HTTP
initialDelaySeconds: 900
periodSeconds: 10
successThreshold: 1
timeoutSeconds: 30
name: tomcat
ports:
- containerPort: 8080
protocol: TCP
- containerPort: 1234
protocol: TCP
readinessProbe:
failureThreshold: 12
httpGet:
path: /eUrbanMIS/main.htm
port: 8080
scheme: HTTP
initialDelaySeconds: 100
periodSeconds: 10
successThreshold: 1
timeoutSeconds: 10
resources:
limits:
cpu: "3"
ephemeral-storage: 2Gi
memory: 4200Mi
requests:
cpu: 200m
ephemeral-storage: 1Gi
memory: 2096Mi
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
volumeMounts:
- mountPath: /usr/local/tomcat/logs
name: egova-log-tomcat
subPathExpr: $(POD_NAME)
- mountPath: /etc/localtime
name: date-config
- mountPath: /usr/local/tomcat/webapps/eUrbanMIS/WEB-INF/log
name: egova-log-web
subPathExpr: $(POD_NAME)
- mountPath: /prometheus
name: prom
- mountPath: /docker-entrypoint-init.d/after-prepare-configmaps/
name: code-init-configmap
- mountPath: /usr/local/tomcat/webapps
name: code-volume
- mountPath: /etc/config
name: mis-configmap-volume
- mountPath: /etc/sidecar/
name: sidecar-volume
- image: swr.cn-east-3.myhuaweicloud.com/egova_urban/tool-inotify:1.0
```

```
imagePullPolicy: IfNotPresent
limits:
cpu: 100m
memory: 50Mi
requests:
cpu: 10m
memory: 10Mi
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
volumeMounts:
- mountPath: /etc/config
name: mis-configmap-volume
- mountPath: /etc/sidecar/
name: sidecar-volume
- mountPath: /usr/local/tomcat/webapps
name: code-volume
dnsPolicy: ClusterFirst
initContainers:
- command:
- bash
- -c
- |
set -ex
echo "数据库检查通过！"
env:
- name: DB_HOST_BIZ
valueFrom:
configMapKeyRef:
key: db_host_biz
name: global-configmap
- name: DB_PORT_BIZ
valueFrom:
configMapKeyRef:
key: db_port_biz
name: global-configmap
- name: DB_NAME_BIZ
valueFrom:
configMapKeyRef:
key: db_name_biz
name: global-configmap
- name: DB_USER_BIZ
valueFrom:
secretKeyRef:
key: db_user_biz
valueFrom:
secretKeyRef:
key: db_passwd_biz
name: global-secret
image: swr.cn-east-3.myhuaweicloud.com/egova_urban/mysql-client:1.1
imagePullPolicy: IfNotPresent
name: check
resources:
limits:
cpu: 100m
memory: 50Mi
requests:
cpu: 10m
memory: 10Mi
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
- command:
- bash
- -c
- |
set -ex
imagePullPolicy: IfNotPresent
name: init-monitor
resources:
limits:
```

```
cpu: 100m
memory: 50Mi
requests:
cpu: 10m
memory: 10Mi
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
volumeMounts:
- mountPath: /etc/prometheus
  name: prom
- command:
  - bash
  - -c
  - |
  set -ex
  mkdir -p /code/webapps
  cp -r /usr/local/tomcat/webapps/* /code/webapps/
  imagePullPolicy: IfNotPresent
  name: main
resources:
limits:
cpu: 100m
memory: 50Mi
requests:
cpu: 10m
memory: 10Mi
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
volumeMounts:
- mountPath: /code/webapps
  name: code-volume
  priorityClassName: priority-b
  restartPolicy: Always
  schedulerName: default-scheduler
  securityContext: {}
  terminationGracePeriodSeconds: 30
volumes:
- configMap:
  defaultMode: 420
  name: eurban-mis-init
  name: code-init-configmap
- emptyDir: {}
  name: code-volume
- emptyDir: {}
  name: sidecar-volume
- emptyDir: {}
  name: prom
- hostPath:
  path: /egova/log/tomcat
  type: DirectoryOrCreate
  path: /etc/localtime
  type: ""
  name: date-config
- hostPath:
  path: /egova/log/web
  type: DirectoryOrCreate
  name: egova-log-web
- configMap:
  defaultMode: 420
  name: eurban-mis-config
  name: mis-configmap-volume
status:
availableReplicas: 1
conditions:
- lastTransitionTime: "2021-05-08T09:25:30Z"
  lastUpdateTime: "2021-05-08T09:25:30Z"
  message: Deployment has minimum availability.
  reason: MinimumReplicasAvailable
  status: "True"
```

```
type: Available
- lastTransitionTime: "2021-05-08T06:22:15Z"
lastUpdateTime: "2021-06-04T06:01:41Z"
message: ReplicaSet "urban-mis-backend-7db7d7574" has successfully progressed.
reason: NewReplicaSetAvailable
status: "True"
type: Progressing
observedGeneration: 3
readyReplicas: 1
replicas: 1
updatedReplicas: 1
```

其他的pod定义可根据以下命令进行查看：

图 3-81 查看

```
[root@egova-82192-iu1aj ~]# kubectl get deployments.apps -A | grep prod
prod      egova-service-912b0aa      0/3      1      0      83d
prod      egova-service-demo         2/2      2      2      82d
prod      egova-service-export       0/3      1      0      83d
prod      egova-service-form         0/3      1      0      83d
prod      egova-service-ia           0/3      1      0      83d
prod      egova-service-media        2/2      2      2      83d
prod      egova-service-message      0/3      1      0      83d
prod      egova-service-recognition  0/3      1      0      83d
prod      eurban-fileserver          1/1      1      1      83d
prod      eurban-gis                  1/1      1      1      54d
prod      eurban-ia                   1/1      1      1      54d
prod      eurban-media                1/1      1      1      83d
prod      eurban-mis-backend         1/1      1      1      83d
prod      eurban-mis-frontend        1/1      1      1      83d
prod      eurban-mis-frontend-library 1/1      1      1      83d
prod      eurban-mis-job              1/1      1      1      83d
prod      eurban-mis-stat            1/1      1      1      83d
prod      eurban-moble-vue           1/1      1      1      83d
prod      eurban-public              1/1      1      1      83d
prod      eurban-public-h5           1/1      1      1      83d
prod      eurban-statgather          1/1      1      1      83d
prod      goosekeeper                 1/1      1      1      83d
prod      zookeeper                   1/1      1      1      83d

[root@egova-82192-iu1aj ~]# kubectl describe deployments.apps eurban-media
Name:          eurban-media
Namespace:    prod
CreationTime: Tue, 30 Mar 2021 16:15:34 -0800
Labels:       app=eurban-media
Annotations:  deployment.kubernetes.io/revision: 2
              kubectl.kubernetes.io/last-applied-configuration:
                [{"apiVersion": "apps/v1", "kind": "Deployment", "metadata": {"annotations": {}, "labels": {"app": "eurban-media"}, "name": "eurban-media", "namespace": ...
Selector:     app=eurban-media
Replicas:    1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=eurban-media
  Containers:
    media:
      Image:   ser.ch-east-3.myhuaweicloud.com/egova_eurban/eurban-media:20210201
      Port:   80/TCP
      Host Port:  80/TCP
      Limits:
        cpu:          1500m
        ephemeral-storage: 2000Mi
        memory:       500Mi
      Requests:
        cpu:          100m
        ephemeral-storage: 500Mi
        memory:       200Mi
  Liveness:   http-get http://:80/MediaBot/midiaroot.proxy delay=20s timeout=30s period=10s #success=1 #failure=3
  Readiness:  http-get http://:80/MediaBot/midiaroot.proxy delay=10s timeout=10s period=10s #success=1 #failure=3
```

- 4. 部署
定义好Service和Pod之后，可根据以下命令进行部署

图 3-82 部署

```
[root@egova-82192-iu1aj eurban-mis-backend]# pwd
/egova/k8s/common/eurban/prod/eurban-mis-backend
[root@egova-82192-iu1aj eurban-mis-backend]# kubectl apply -f web-mis-deployment.yaml
[root@egova-82192-iu1aj eurban-mis-backend]#
```

根据以下命令查看服务启动情况

图 3-83 查看服务启动情况

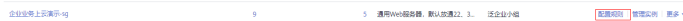
```
[root@egova-82192-lu1aj ~]# kubectl -n prod get pods
NAME                                READY   STATUS    RESTARTS   AGE
deploy-data-geoserver-pvc-gmirs      0/1    Completed 0           83d
deploy-data-gis-c5flc                1/1    Running   0           83d
egova-service-bizbase-5656995f47-n7gph 0/1    ImagePullBackOff 0           83d
egova-service-bizbase-678f498b84-7xh8p 0/1    Init:ImagePullBackOff 0           83d
egova-service-bizbase-678f498b84-szzgk 0/1    Init:ImagePullBackOff 0           83d
egova-service-bizbase-678f498b84-vjdhz 0/1    Init:ImagePullBackOff 0           83d
egova-service-demo-5446db46db-fzt6q  2/2    Running   0           82d
egova-service-demo-5446db46db-txxpl  2/2    Running   0           82d
egova-service-export-6b4c8d799f-qg85b 0/1    CrashLoopBackOff 20736       83d
egova-service-export-9b4bcf58d-dzrhd  0/1    Running   20147       83d
egova-service-export-9b4bcf58d-mwxpx  0/1    Running   20145       83d
egova-service-export-9b4bcf58d-z4j8p  0/1    Running   20147       83d
egova-service-form-74ccf58876-fv8gn  0/1    Init:ImagePullBackOff 0           83d
egova-service-form-74ccf58876-q7769  0/1    Init:ImagePullBackOff 0           83d
egova-service-form-74ccf58876-t876b  0/1    Init:ImagePullBackOff 0           83d
egova-service-lm-69c9cdb488-8lsh6    0/1    Init:ImagePullBackOff 0           83d
egova-service-lm-69c9cdb488-g798h    0/1    Init:ImagePullBackOff 0           83d
egova-service-lm-69c9cdb488-x86pm    0/1    Init:ImagePullBackOff 0           83d
egova-service-lm-869d4986f9-rzmgd    0/1    ImagePullBackOff 0           83d
egova-service-media-cff479bfg-982z4  2/2    Running   0           82d
egova-service-media-cff479bfg-n7hcd  2/2    Running   0           82d
egova-service-message-6b6c4df95-2xpqg 0/1    Init:ImagePullBackOff 0           83d
egova-service-message-6b6c4df95-jl596 0/1    Init:ImagePullBackOff 0           83d
egova-service-message-6b6c4df95-nbhm8 0/1    Init:ImagePullBackOff 0           83d
egova-service-message-6d6664d65b-jdaxm 0/1    ImagePullBackOff 0           83d
egova-service-recognition-785cb4576c-c5z5k 0/1    Init:ImagePullBackOff 0           83d
egova-service-recognition-785cb4576c-lkbjn 0/1    Init:ImagePullBackOff 0           83d
egova-service-recognition-785cb4576c-ml6tw 0/1    Init:ImagePullBackOff 0           83d
egova-service-recognition-78f6b8fd6c-kbxwr 0/1    ImagePullBackOff 0           83d
eurban-fileserver-7594f68c89-pkja5    2/2    Running   0           83d
eurban-gis-bdd9c6579-8548h            2/2    Running   0           5d2h
eurban-lm-6659cc85dd-gx49n            2/2    Running   0           11d
eurban-media-5cc74c2b05-7sc7w         1/1    Running   2           83d
eurban-mis-backend-7db7d7274-f8vjg    2/2    Running   0           18d
eurban-mis-frontend-6d87748f56-shmz2  1/1    Running   2           18d
eurban-mis-frontend-library-6d4df5847b-ngsxx 1/1    Running   0           18d
eurban-mis-job-b4fd95999-zmmsv        2/2    Running   0           18d
eurban-mis-stat-657b7f449-k6sg5       2/2    Running   0           18d
eurban-mobile-vue-7b4b778d6-9jqft     1/1    Running   2           17d
eurban-public-5c774d97fc-zrkp6        2/2    Running   0           11d
eurban-public-h5-5484578686-7hdck     1/1    Running   0           12d
eurban-statgather-79d868489c-sw7z1    1/1    Running   0           83d
geoserver-d7c555cf-m9dhk              2/2    Running   1           21d
iae-zookeeper-0                       1/1    Running   0           6d20h
iotplatfom-0                          1/1    Running   19          81d
kafka-0                                1/1    Running   0           6d20h
logrotate-5snlp                       1/1    Running   0           83d
logrotate-kn6rs                       1/1    Running   0           83d
postgis-0                              1/1    Running   0           83d
tdengine-0                             1/1    Running   0           82d
zookeeper-cdc455b7-9rmkh              1/1    Running   0           83d
```

网络配置

1. 安全组规则配置

- 登录华为云，进入网控制台
- 在网络控制台左侧菜单栏选择“访问控制”-“安全组”，点击安全组的配置规则

图 3-84 规则配置



- 点击入方向规则，点击添加规则，设置允许内网互通（3.1.1章节创建的子网），目的是让该子网下的云服务资源能够访问到被该安全组规则保护的资源。

图 3-85 添加规则



说明

上述源地址10.20.0.0/24用地址在这里查看（注意看自己在VPC&安全组中创建的VPC和子网是什么网段的，要复制自己的子网网段）：

图 3-86 添加规则 2



- d. 点击入方向规则，点击添加规则，设置放通100.125.0.0/16网段（用于ELB后端服务器健康检查）

图 3-87 进入方向规则



- e. 检查一下规则是否都配置好了

图 3-88 检查



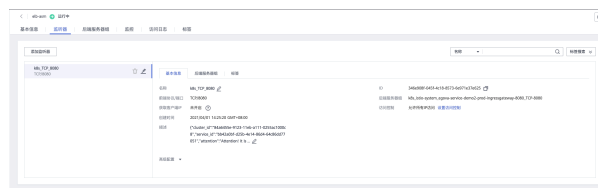
2. ELB配置

监听器配置

本章节的目的是配置静态nginx服务的监听器，实现：通过ELB公网IP+监听的端口能进入静态nginx前端界面。

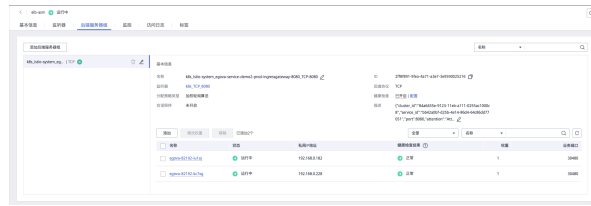
- a. 配置监听器

图 3-89 配置监听器



- b. 配置后端服务器组

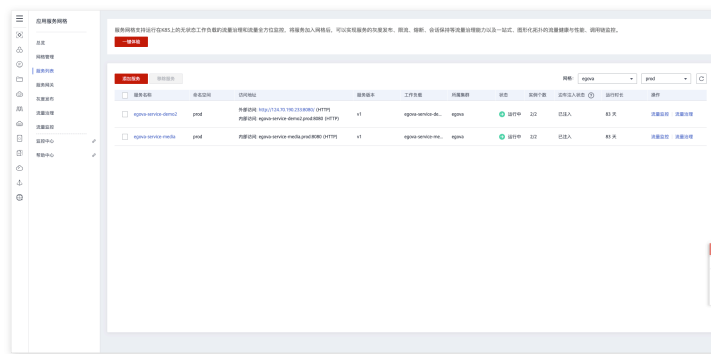
图 3-90 配置后端服务器组



应用服务网络配置

进入到应用服务网络“服务列表”下

图 3-91 服务列表



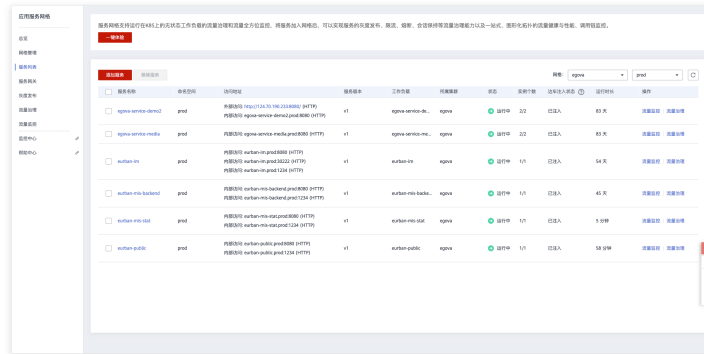
点击添加服务，选择命名空间，选择要添加的服务

图 3-92 添加服务



点击确定，添加成功

图 3-93 确定



数据初始化

由eurban-mis-job服务初始化数据。

图 3-94 初始化数据

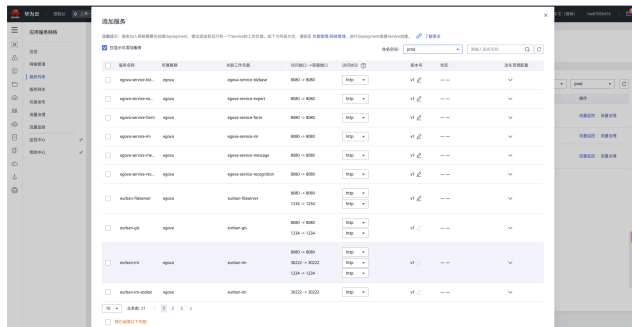
```
[root@egova-82192-iu1aj ~]# kubectl -nprod get svcignep job
eurban-mis-job      ClusterIP  10.247.163.48  <none>      8080/TCP,1234/TCP
                    51d
[root@egova-82192-iu1aj ~]#
```

4 附录

应用服务网格添加问题

添加服务时会出现异常，这时需手动调整参数。

图 4-1 调整参数



如下图所示，需添加版本号信息，否则无法添加
kubectl edit deployments.apps eurban-mis-backend

图 4-2 调整参数

```
creationTimestamp: "2021-09-08T06:22:15Z"
generation: 2
labels:
  app: eurban-mis-backend
  name: eurban-mis-backend
  namespace: prod
resourceVersion: "18578188"
selfLink: /apis/apps/v1/namespaces/prod/deployments/eurban-mis-backend
uid: 34d909e-1187-4faf-b7cd-5156eb585737
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: eurban-mis-backend
      version: v1
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
      type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: eurban-mis-backend
        name: eurban-mis-backend
        version: v1
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
              - matchExpressions:
                  - key: app.egova.com.cn/common
                    operator: In
                    values:
                      - "true"
                  - key: namespace.egova.com.cn/prod
                    operator: In
                    values:
                      - "true"
                  - key: app.egova.com.cn/tomcat-large
                    operator: NotIn
```

服务更新流程

1. 由测试打包好镜像并提供镜像版本号
2. 从数字政通镜像仓库拉去镜像到部署服务器上，并上传至华为云服务，操作如下
docker pull hub.egova.com.cn/microservice/urban-mis-frontend-library:20201201-3.9.12-microservice-t04251036
docker tag hub.egova.com.cn/microservice/urban-mis-frontend-library:20201201-3.9.12-microservice-t04251036 swr.cn-east-3.myhuaweicloud.com/egova_urban/urban-mis-frontend-library:20201201-3.9.12-microservice-t04251036
docker push swr.cn-east-3.myhuaweicloud.com/egova_urban/urban-mis-frontend-library:20201201-3.9.12-microservice-t04251036
3. 手动更新镜像版本号，保存即自动更新

图 4-3 保存

```
- command:
- bash
- -c
- |
- set -ex
cp /prantheus/* /etc/prantheus/
image: swr.cn-east-3.myhuaweicloud.com/egova_urban/jmx-exporter:0.12.0-1
imagePullPolicy: IfNotPresent
name: link-monitor
resources:
limits:
cpu: 100m
memory: 50Mi
requests:
cpu: 10m
memory: 10Mi
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
volumeMounts:
- mountPath: /etc/prantheus
name: prn
- command:
- bash
- -c
- |
- set -ex
mkdir -p /code/webapps
cp -r /usr/local/tomcat/webapps/* /code/webapps/
image: swr.cn-east-3.myhuaweicloud.com/egova_urban/urban-mis-backend:20201201-3.9.12-microservice-t05001727
imagePullPolicy: IfNotPresent
name: main
resources:
limits:
cpu: 100m
memory: 50Mi
requests:
cpu: 10m
memory: 10Mi
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
volumeMounts:
- mountPath: /code/webapps
name: code-volume
priorityClassName: priority-b
schedulerName: default-scheduler
securityContext: {}
terminationGracePeriodSeconds: 30
volumes:
- configMap:
defaultMode: 420
```

5 修订记录

表 5-1 修订记录

发布日期	修订记录
2023-10-27	删除敏感词。
2023-03-31	第一次正式发布。