

解决方案实践

埃林哲云时通供销一体化中台解决方案 实践

文档版本 1.1
发布日期 2024-04-28



版权所有 © 华为技术有限公司 2024。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。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目录

1 方案概述.....	1
2 资源和成本规划.....	4
3 基于 CCE 搭建云时通业务中台流程.....	6
4 实施步骤.....	8
4.1 搭建 VPC 网络环境.....	8
4.2 搭建云时通中台系统运行环境.....	9
4.3 配置 ELB 及验证操作中台系统.....	23
5 修订记录.....	26

1 方案概述

应用场景

营销中台：

1. 实现企业对下游B端客户、经销商、门店的多层次多渠道销售管控。
2. 实现C端客户拉新、促活、留存、转化、分享的管理闭环以及精准化营销，通过商城实现线上下单和O2O销售。
3. 实现销售团队对业务员拜访、巡店销售、团队业绩的精细化管理。
4. 实现企业对各类营销费用的精细化管理，C端各类促销费用、经销商渠道侧的各类费补、货补、门店活动等管理。

供应链中台：

搭建端到端的供应链计划平台，拉通采销上下游信息，通过中央库存&库存可视化，优供应链资源，实现企业全渠道一盘货管理，提升供应链敏捷度，降低库存成本，降低业务执行风险。

财务中台：

实现业务系统与财务系统有机统一，业务系统与总账系统、ERP、财务系统等高效对接，实现企业业财一体化。通过全面预算管理、成本核算管理，结合费控管理TPM系统，营销管理系统、数据中台等，赋予财务人员提前融入业务的能力，将财务分析结果前移，降低运营风险，实现企业业财有机融合。

方案架构

图 1-1 架构图



云时通中台以可扩展性的能力中心作为底座基础，通过能力中心构建组合赋能架构之上的企业业务场景，灵活构建，快速响应前端业务需求。同时融合业务中台和数据中台，帮助企业实现业务数据化和数据业务化，打造真正的数字化优良企业。

云时通采用微服务、智能化、组件式的技术架构，为数字化业务提供动力引擎，以弹性伸缩的技术服务，实现业务快速部署，支撑业务快速响应用户和市场变化。

方案优势

- 市场侧竞争力
 - a. 行业咨询能力：埃林哲面向大型企业提供信息化整体建设规划化和CIO伴随咨询服务，通过行业及技术咨询引导，整合从ERP大后端到企业中台产品及其他套件产品，为企业提供数字化升级端到端整体解决方案。
 - b. 全方位数字化落地能力：埃林哲具备营销+供应链业务中台、数据中台的自研中台产品和SAP&Oracle ERP的一体化数字平台实施落地能力。在供应链协同与计划侧产品具有优良优势：时通中台在OMS与供应链协同与计划侧建立了较为强大的规则引擎，满足订单接入审核、仓库路由等重点环节的灵活配置需求。
- 技术侧竞争力
 - a. 多触点营销：具备从前端（小程序，APP，B2C商城，B2B商城，经销商平台、第三方平台）到业务中台、后端ERP的一体化数字平台的打造能力。
 - b. 深度接口配置：云时通中台与SAP、Oracle等ERP接口做了深度配置，可大大减少多系统间系统集成工作量。
 - c. 高效的低代码开发平台：帮助企业实现业务的研发的低成本与快速迭代。微服务、智能化、组件式的技术架构，为数字化业务提供动力引擎，以弹性伸缩的技术服务，实现业务快速部署，支撑业务快速响应用户和市场变化。
- 云服务竞争力

- a. 灵活弹性的高性能云服务器：部署HPC头节点和工作节点，该计算实例类型是H系列和M系列针对HPC业务的增强型实例，承载该实例的物理服务器提供最为强劲的计算能力。
- b. 优良性能的裸金属服务器：采用2路18核服务器，无虚拟化损耗，具备最佳的性能优势，用户可通过管理控制台实现裸金属服务器的自动化发放，满足HPC业务按需灵活弹性的需求。
- c. Lustre高性能并行文件系统：使用经过优化的专用ECS、EVS规格部署Lustre文件系统，单OSS可提供最大4.5GBps吞吐量，最大提供2Tbps吞吐量，满足HPC文件系统高性能要求。

2 资源和成本规划

表 2-1 资源和成本规划

序号	云服务	规格	数量	参考价格/月
1	云容器引擎	产品分类： CCE容器集群 混合集群 50节点 是	1	1262.40
2	ECS Node1	规格： X86计算 通用计算型 s6.2xlarge.4 8核 32GB 镜像： CentOS CentOS 7.9 64bit 系统盘： 高IO 40GB 弹性公网IP： 全动态BGP 独享 按带宽计费 5Mbit/s	1	991.40
3	ECS Node2	规格： X86计算 通用计算型 s6.2xlarge.4 8核 32GB 镜像： CentOS CentOS 7.9 64bit 系统盘： 高IO 40GB 弹性公网IP： 全动态BGP 独享 按带宽计费 5Mbit/s	1	991.40
4	ECS Node3	规格： X86计算 通用计算型 s6.2xlarge.4 8核 32GB 镜像： CentOS CentOS 7.9 64bit 系统盘： 高IO 40GB 弹性公网IP： 全动态BGP 独享 按带宽计费 5Mbit/s	1	991.40
5	弹性公网IP	产品类型： 共享 按带宽计费 5Mbit/s 1个	1	400.00
6	云堡垒机	性能规格： 100资产标准版	1	3780.00

序号	云服务	规格	数量	参考价格/月
7	弹性负载均衡	实例规格类型：共享型负载均衡 IP费用：1LCU 全动态BGP 带宽：全动态BGP 带宽 3Mbit/s	1	0.53
8	云硬盘	磁盘容量：高IO 100GB	1	35.00
9	分布式缓存服务 Redis (集群)	版本号：5.0 Proxy 集群 X86 DRAM 4 GB	1	420.00
10	云数据库 RDS(for MySQL) 主备	云数据库RDS for MySQL 主备 x86通用型 8 vCPUs 32 GB 200G , 建议连接数：10000, TPS/QPS: 3382 67644"	1	2460.00
11	云LTS日志服务	原始日志大小：10g/天 日志存储时长(天) 7天	1	150.00
12	企业主机安全	规格：企业版	3	270.00
13	云备份	存储库类型：云服务器备份存储库 1000GB	1	200.00
14	分布式消息服务 RocketMQ版	规格：rocketmq.4u8g.cluster.small 代理存储空间：超高IO：300GB 实例类型：集群	1	4080.00
15	对象存储服务	产品类型：对象存储 标准存储单AZ存储包 100GB	1	9.00
16	云硬盘	磁盘容量：高IO 100GB	1	35.00
17	云硬盘	磁盘容量：高IO 100GB	1	35.00
18	Anti-DDoS流量清洗	免费版	1	0.00

3 基于 CCE 搭建云时通业务中台流程

本文介绍云时通业务中台的云原生部署方式，实现vpc局域网互通及中台系统接入互联网；云时通业务中台系统涉及到基础服务组件包括redis服务、RDS For Mysql、MQ；包括基础设施ECS等；云时通业务中台系统部署流程如图：

图 3-1 部署流程图

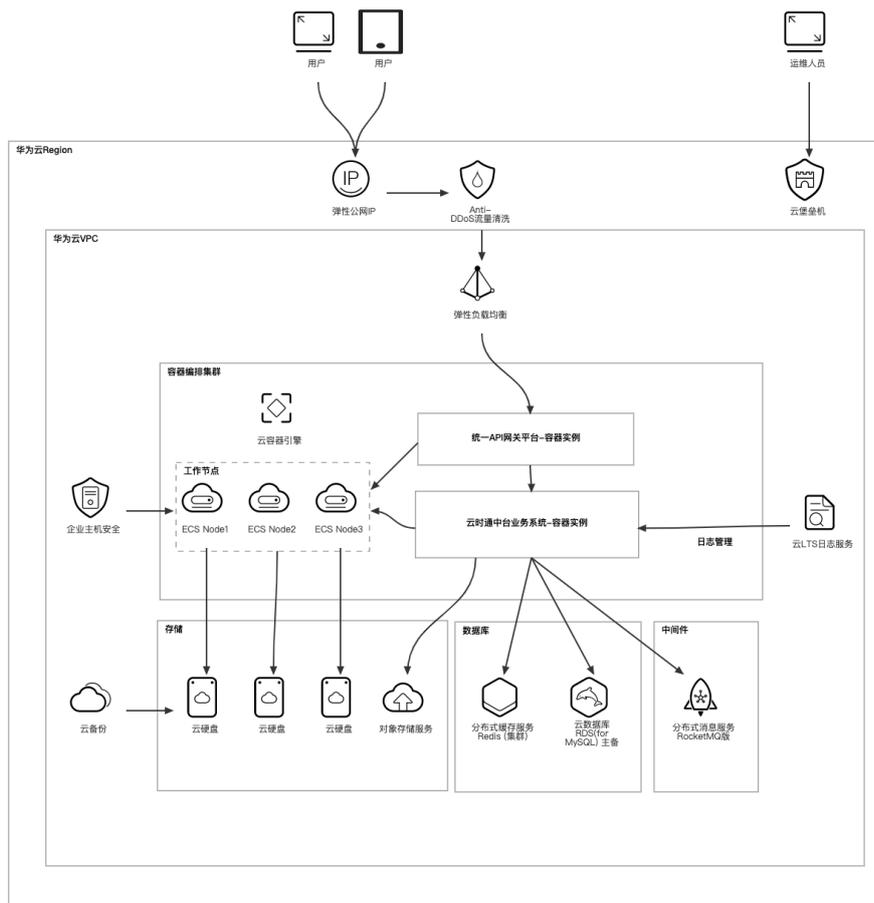


表 3-1 云时通业务中台系统依赖环境及服务说明

序号	步骤	说明
1	Region	华为云物理机房的所在地理区域，如北京一、北京四、上海一
2	Vpc	华为云上隔离的、私密的虚拟网络环境，可以自由配置VPC内的IP地址段、子网、安全组等子服务，VPC中云资源默认无法访问公网；企业构建混合云时，注意Vpc网段不要IDC机房网段冲突。
3	ELB	负载均衡器，提供云时通中台系统的互联网访问入口
4	CCE	容器治理引擎，快速创建kubernetes集群，实现云上轻松部署、管理和扩展容器化应用程序
5	ECS	基础设施计算资源，可随时自由获取、可弹性伸缩；
6	RDS For Mysql	基于云计算平台的即开即用、稳定可靠、弹性伸缩、便捷管理的在线云数据库服务。
7	分布式缓存服务	提供即开即用、安全可靠、弹性扩容、便捷管理的在线分布式缓存能力，兼容redis、Memcached
8	RocketMQ	分布式消息服务RocketMQ版是一个低延迟、弹性高可靠、高吞吐、动态扩展、便捷多样的消息中间件服务。
9	TLS	提供日志收集、分析、存储等服务。用户可以通过云日志服务快速高效地进行设备运维管理、用户业务趋势分析、安全监控审计等操作
10	DDoS	华为云提供的安全防护方案，包括基础防护、高级防护等

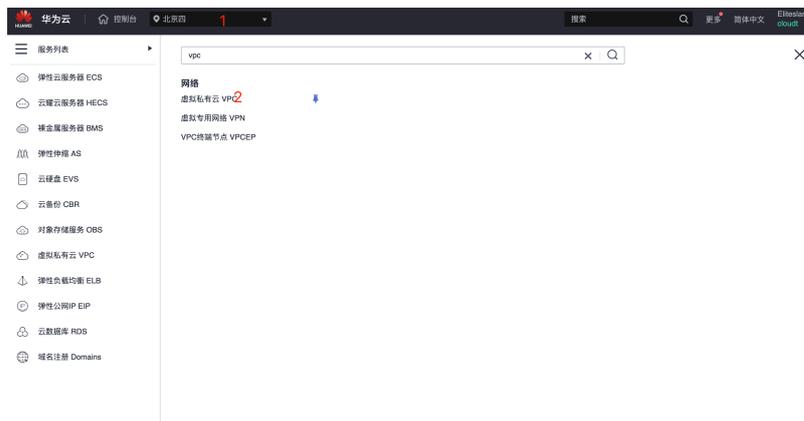
4 实施步骤

- 4.1 搭建VPC网络环境
- 4.2 搭建云时通中台系统运行环境
- 4.3 配置ELB及验证操作中台系统

4.1 搭建 VPC 网络环境

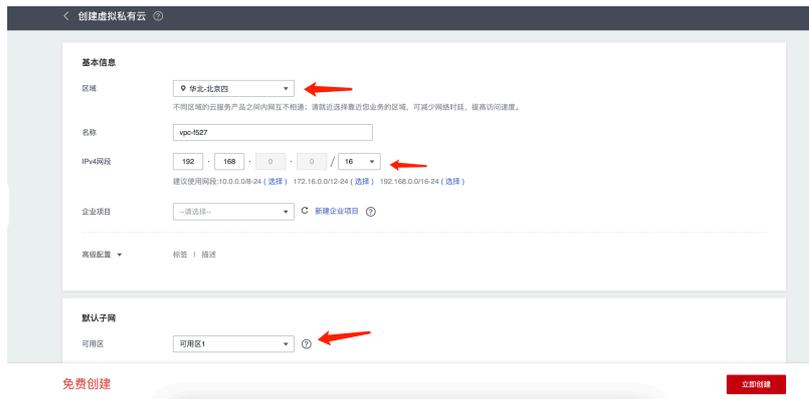
1. 规划vpc网段，如 192.168.0.0/16
2. 创建vpc网络：
在产品列表里搜索vpc，进入vpc控制台

图 4-1 进入 vpc 控制台



创建vpc，选择区域，填写提前规划的网段、选择可用区

图 4-2 创建 vpc

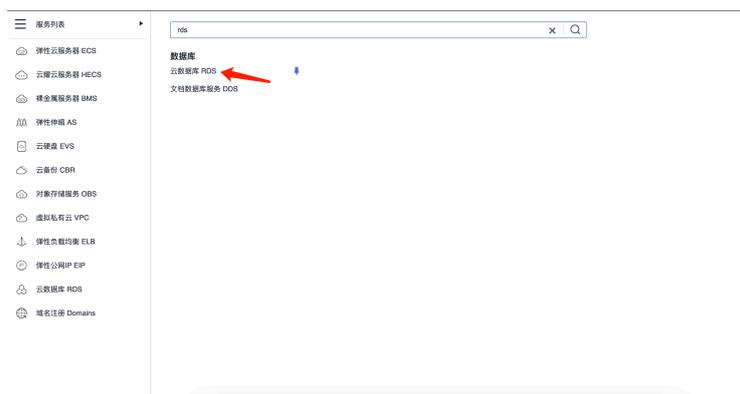


4.2 搭建云时通中台系统运行环境

搭建 RDSForMysql 环境

- 服务列表里搜索RDS，进入RDS控制台

图 4-3 进入 RDS 控制台



- 选择区域、选配规格及vpc网络

图 4-4 选择 1

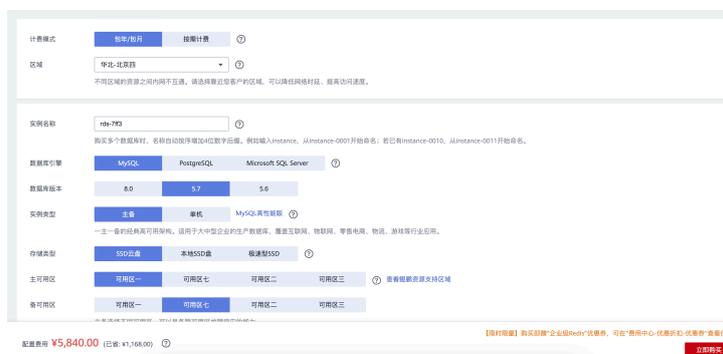
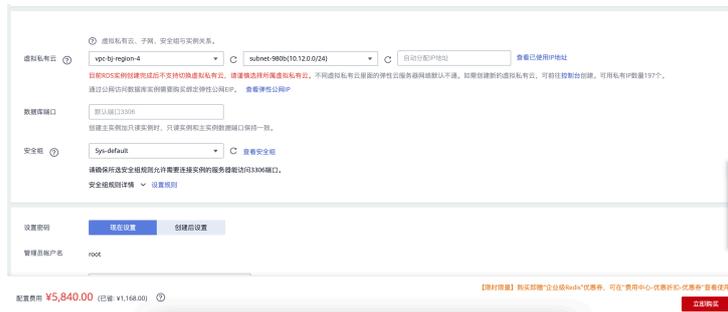
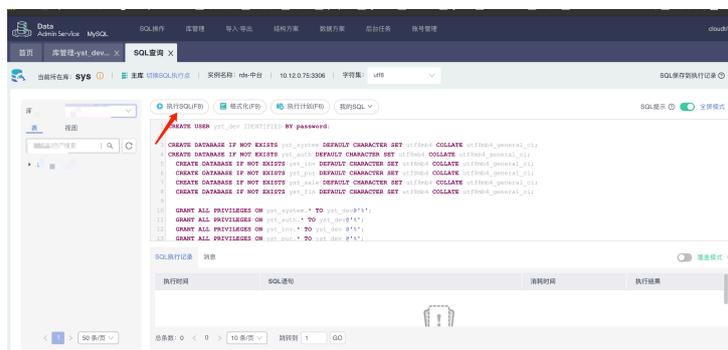


图 4-5 选择 2



- 创建数据库用户及schema

图 4-6 创建数据库

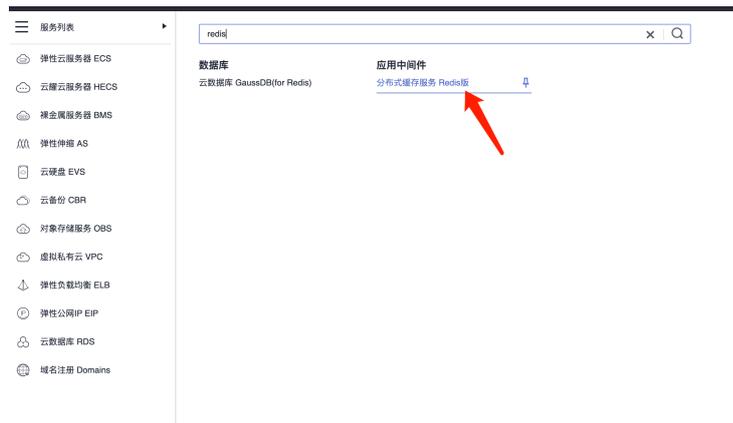


```
CREATE USER `yst_dev`@`%` IDENTIFIED WITH mysql_native_password BY password
CREATE DATABASE IF NOT EXISTS yst_system DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;
CREATE DATABASE IF NOT EXISTS yst_auth DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;
CREATE DATABASE IF NOT EXISTS yst_inv DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;
CREATE DATABASE IF NOT EXISTS yst_pur DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;
CREATE DATABASE IF NOT EXISTS yst_sale DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;
CREATE DATABASE IF NOT EXISTS yst_fin DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;
GRANT ALL PRIVILEGES ON yst_system.* TO yst_dev@'%';
GRANT ALL PRIVILEGES ON yst_auth.* TO yst_dev@'%';
GRANT ALL PRIVILEGES ON yst_inv.* TO yst_dev @'%';
GRANT ALL PRIVILEGES ON yst_pur.* TO yst_dev @'%';
GRANT ALL PRIVILEGES ON yst_sale.* TO yst_dev @'%';
GRANT ALL PRIVILEGES ON yst_fin.* TO yst_dev @'%';
FLUSH PRIVILEGES;
```

搭建 Redis 环境

服务列表里搜索Redis，进入分布式缓存控制台

图 4-7 应用中间件



选择区域、选配规格及vpc网络

图 4-8 选择 1



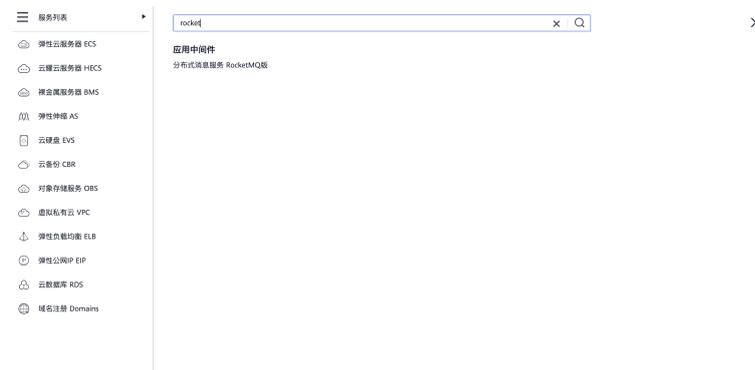
图 4-9 选择 2



搭建 RocketMQ 环境

服务列表里搜索rocketmq，进入分布式消息服务控制台

图 4-10 进入控制台

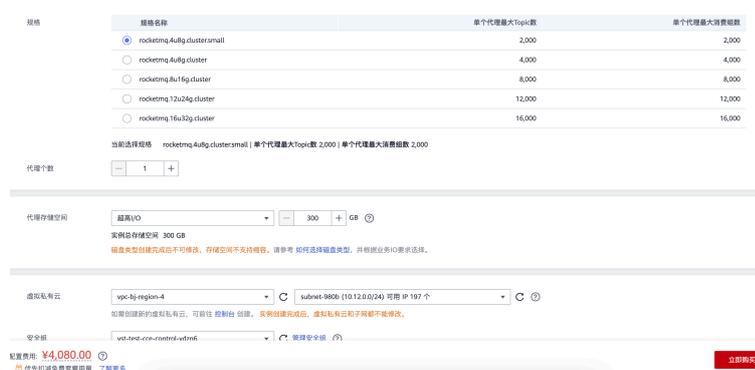


选择区域、选配规格及vpc网络

图 4-11 选择 1



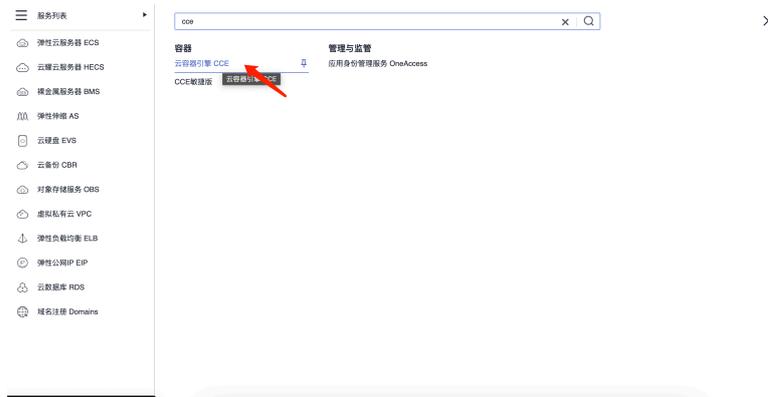
图 4-12 选择 2



搭建 CCE 集群环境

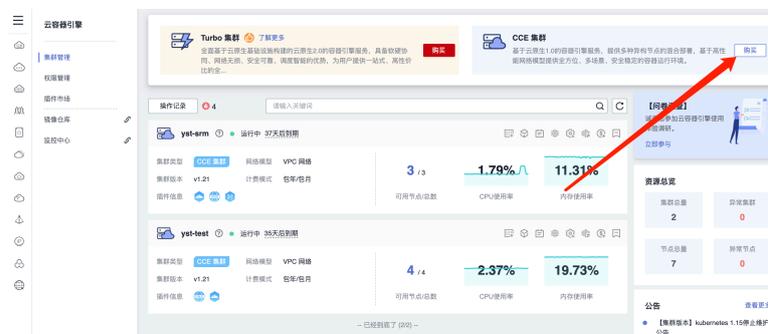
- 服务列表里搜索cce，进入分布式消息服务控制台

图 4-13 进入控制台



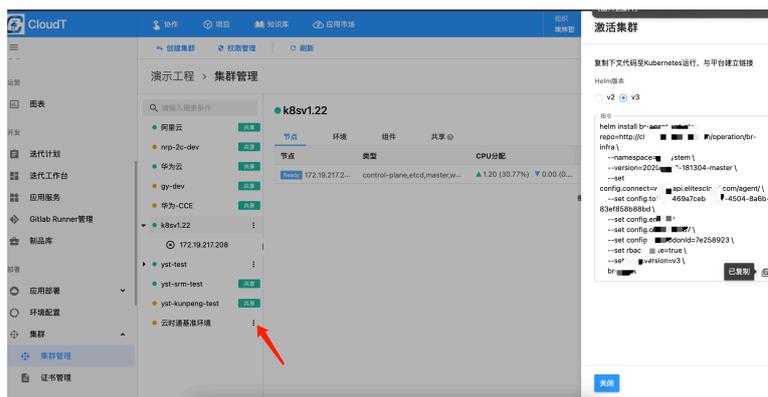
- 单击购买cce集群

图 4-14 购买



- 托管到云梯-DevOps平台
云梯-DevOps平台创建集群，copy激活脚本

图 4-15 托管 1




```

`data_id` varchar(255) NOT NULL COMMENT 'data_id',
`group_id` varchar(255) DEFAULT NULL,
`content` longtext NOT NULL COMMENT 'content',
`md5` varchar(32) DEFAULT NULL COMMENT 'md5',
`gmt_create` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
`gmt_modified` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '修改时间',
`src_user` text COMMENT 'source user',
`src_ip` varchar(50) DEFAULT NULL COMMENT 'source ip',
`app_name` varchar(128) DEFAULT NULL,
`tenant_id` varchar(128) DEFAULT '' COMMENT '租户字段',
`c_desc` varchar(256) DEFAULT NULL,
`c_use` varchar(64) DEFAULT NULL,
`effect` varchar(64) DEFAULT NULL,
`type` varchar(64) DEFAULT NULL,
`c_schema` text,
PRIMARY KEY (`id`),
UNIQUE KEY `uk_configinfo_datagrouptenant` (`data_id`,`group_id`,`tenant_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='config_info';
/*****
/* 数据库全名 = nacos_config */
/* 表名称 = config_info_aggr */
*****/
CREATE TABLE `config_info_aggr` (
`id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT 'id',
`data_id` varchar(255) NOT NULL COMMENT 'data_id',
`group_id` varchar(255) NOT NULL COMMENT 'group_id',
`datum_id` varchar(255) NOT NULL COMMENT 'datum_id',
`content` longtext NOT NULL COMMENT '内容',
`gmt_modified` datetime NOT NULL COMMENT '修改时间',
`app_name` varchar(128) DEFAULT NULL,
`tenant_id` varchar(128) DEFAULT '' COMMENT '租户字段',
PRIMARY KEY (`id`),
UNIQUE KEY `uk_configinfoaggr_datagrouptenantdatum` (`data_id`,`group_id`,`tenant_id`,`datum_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='增加租户字段';
/*****
/* 数据库全名 = nacos_config */
/* 表名称 = config_info_beta */
*****/
CREATE TABLE `config_info_beta` (
`id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT 'id',
`data_id` varchar(255) NOT NULL COMMENT 'data_id',
`group_id` varchar(128) NOT NULL COMMENT 'group_id',
`app_name` varchar(128) DEFAULT NULL COMMENT 'app_name',
`content` longtext NOT NULL COMMENT 'content',
`beta_ips` varchar(1024) DEFAULT NULL COMMENT 'betalps',
`md5` varchar(32) DEFAULT NULL COMMENT 'md5',
`gmt_create` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
`gmt_modified` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '修改时间',
`src_user` text COMMENT 'source user',
`src_ip` varchar(50) DEFAULT NULL COMMENT 'source ip',
`tenant_id` varchar(128) DEFAULT '' COMMENT '租户字段',
PRIMARY KEY (`id`),
UNIQUE KEY `uk_configinfobeta_datagrouptenant` (`data_id`,`group_id`,`tenant_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='config_info_beta';
/*****
/* 数据库全名 = nacos_config */
/* 表名称 = config_info_tag */
*****/
CREATE TABLE `config_info_tag` (
`id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT 'id',
`data_id` varchar(255) NOT NULL COMMENT 'data_id',
`group_id` varchar(128) NOT NULL COMMENT 'group_id',
`tenant_id` varchar(128) DEFAULT '' COMMENT 'tenant_id',
`tag_id` varchar(128) NOT NULL COMMENT 'tag_id',
`app_name` varchar(128) DEFAULT NULL COMMENT 'app_name',
`content` longtext NOT NULL COMMENT 'content',
`md5` varchar(32) DEFAULT NULL COMMENT 'md5',
`gmt_create` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
`gmt_modified` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '修改时间',

```

```
`src_user` text COMMENT 'source user',
`src_ip` varchar(50) DEFAULT NULL COMMENT 'source ip',
PRIMARY KEY (`id`),
UNIQUE KEY `uk_configinfotag_datagrouptenanttag` (`data_id`,`group_id`,`tenant_id`,`tag_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='config_info_tag';
/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = config_tags_relation */
/*****/
CREATE TABLE `config_tags_relation` (
`id` bigint(20) NOT NULL COMMENT 'id',
`tag_name` varchar(128) NOT NULL COMMENT 'tag_name',
`tag_type` varchar(64) DEFAULT NULL COMMENT 'tag_type',
`data_id` varchar(255) NOT NULL COMMENT 'data_id',
`group_id` varchar(128) NOT NULL COMMENT 'group_id',
`tenant_id` varchar(128) DEFAULT '' COMMENT 'tenant_id',
`nid` bigint(20) NOT NULL AUTO_INCREMENT,
PRIMARY KEY (`nid`),
UNIQUE KEY `uk_configtagrelation_configidtag` (`id`,`tag_name`,`tag_type`),
KEY `idx_tenant_id` (`tenant_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='config_tag_relation';
/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = group_capacity */
/*****/
CREATE TABLE `group_capacity` (
`id` bigint(20) unsigned NOT NULL AUTO_INCREMENT COMMENT '主键ID',
`group_id` varchar(128) NOT NULL DEFAULT '' COMMENT 'Group ID,空字符表示整个集群',
`quota` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '配额,0表示使用默认值',
`usage` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '使用量',
`max_size` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '单个配置大小上限,单位为字节,0表示使用默认值',
`max_aggr_count` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '聚合子配置最大个数,0表示使用默认值',
`max_aggr_size` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '单个聚合数据的子配置大小上限,单位为字节,0表示使用默认值',
`max_history_count` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '最大变更历史数量',
`gmt_create` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
`gmt_modified` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '修改时间',
PRIMARY KEY (`id`),
UNIQUE KEY `uk_group_id` (`group_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='集群、各Group容量信息表';
/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = his_config_info */
/*****/
CREATE TABLE `his_config_info` (
`id` bigint(64) unsigned NOT NULL,
`nid` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
`data_id` varchar(255) NOT NULL,
`group_id` varchar(128) NOT NULL,
`app_name` varchar(128) DEFAULT NULL COMMENT 'app_name',
`content` longtext NOT NULL,
`md5` varchar(32) DEFAULT NULL,
`gmt_create` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
`gmt_modified` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
`src_user` text,
`src_ip` varchar(50) DEFAULT NULL,
`op_type` char(10) DEFAULT NULL,
`tenant_id` varchar(128) DEFAULT '' COMMENT '租户字段',
PRIMARY KEY (`nid`),
KEY `idx_gmt_create` (`gmt_create`),
KEY `idx_gmt_modified` (`gmt_modified`),
KEY `idx_data_id` (`data_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='多租户改造';
/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = tenant_capacity */
```

```
/*-----*/
CREATE TABLE `tenant_capacity` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT COMMENT '主键ID',
  `tenant_id` varchar(128) NOT NULL DEFAULT '' COMMENT 'Tenant ID',
  `quota` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '配额,0表示使用默认值',
  `usage` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '使用量',
  `max_size` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '单个配置大小上限,单位为字节,0表示使用默认值',
  `max_aggr_count` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '聚合子配置最大个数',
  `max_aggr_size` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '单个聚合数据的子配置大小上限,单位为字节,0表示使用默认值',
  `max_history_count` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '最大变更历史数量',
  `gmt_create` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  `gmt_modified` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '修改时间',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_tenant_id` (`tenant_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='租户容量信息表';

CREATE TABLE `tenant_info` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT 'id',
  `kp` varchar(128) NOT NULL COMMENT 'kp',
  `tenant_id` varchar(128) default '' COMMENT 'tenant_id',
  `tenant_name` varchar(128) default '' COMMENT 'tenant_name',
  `tenant_desc` varchar(256) DEFAULT NULL COMMENT 'tenant_desc',
  `create_source` varchar(32) DEFAULT NULL COMMENT 'create_source',
  `gmt_create` bigint(20) NOT NULL COMMENT '创建时间',
  `gmt_modified` bigint(20) NOT NULL COMMENT '修改时间',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_tenant_info_kptenantid` (`kp`,`tenant_id`),
  KEY `idx_tenant_id` (`tenant_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='tenant_info';

CREATE TABLE `users` (
  `username` varchar(50) NOT NULL PRIMARY KEY,
  `password` varchar(500) NOT NULL,
  `enabled` boolean NOT NULL
);

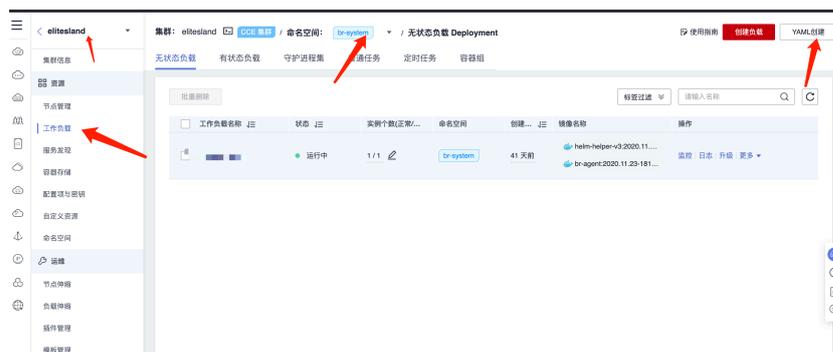
CREATE TABLE `roles` (
  `username` varchar(50) NOT NULL,
  `role` varchar(50) NOT NULL,
  UNIQUE INDEX `idx_user_role` (`username` ASC, `role` ASC) USING BTREE
);

CREATE TABLE `permissions` (
  `role` varchar(50) NOT NULL,
  `resource` varchar(255) NOT NULL,
  `action` varchar(8) NOT NULL,
  UNIQUE INDEX `uk_role_permission` (`role`,`resource`,`action`) USING BTREE
);

INSERT INTO users (username, password, enabled) VALUES ('nacos', '$2a$10$EuWpzHzz32dJN7jexM34MOeYirDdFAZm2kuWj7VEOJhhZkDrxfvUu', TRUE);
INSERT INTO roles (username, role) VALUES ('nacos', 'ROLE_ADMIN');
```

- 创建nacos服务
进入cce集群，单击工作负载，选择命名空间，单击“YAML创建”

图 4-19 创建 nacos 服务 1

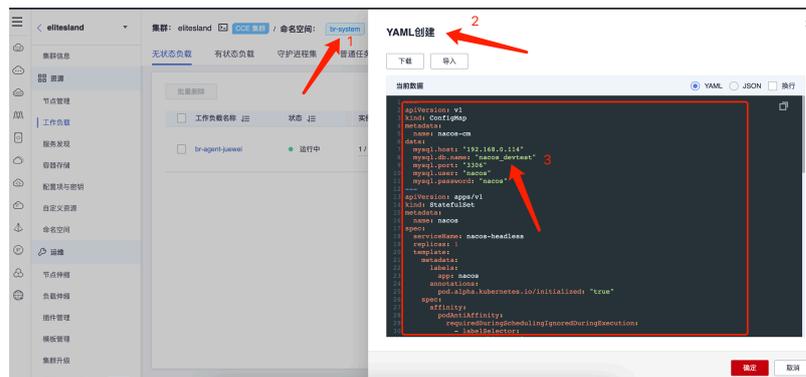


将内容粘贴到编辑面板里：

```
apiVersion: v1
kind: Service
metadata:
  name: nacos-headless
labels:
  app: nacos-headless
spec:
  type: ClusterIP
  clusterIP: None
  ports:
    - port: 8848
      name: server
      targetPort: 8848
    - port: 9848
      name: client-rpc
      targetPort: 9848
    - port: 9849
      name: raft-rpc
      targetPort: 9849
    ## 兼容1.4.x版本的选举端口
    - port: 7848
      name: old-raft-rpc
      targetPort: 7848
  selector:
    app: nacos
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: nacos-cm
data:
  mysql.host: "192.168.0.114"
  mysql.db.name: "nacos_devtest"
  mysql.port: "3306"
  mysql.user: "nacos"
  mysql.password: "nacos"
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: nacos
spec:
  serviceName: nacos-headless
  replicas: 1
  template:
    metadata:
      labels:
        app: nacos
    annotations:
      pod.alpha.kubernetes.io/initialized: "true"
    spec:
      affinity:
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions:
                  - key: "app"
                    operator: In
                    values:
                      - nacos
              topologyKey: "kubernetes.io/hostname"
      containers:
        - name: k8snacos
          imagePullPolicy: Always
          image: nacos/nacos-server:latest
      resources:
        requests:
          memory: "2Gi"
```

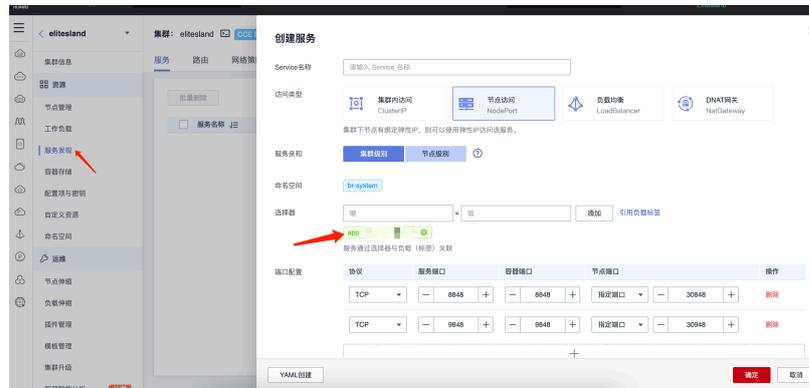
```
cpu: "500m"
ports:
- containerPort: 8848
name: client
- containerPort: 9848
name: client-rpc
- containerPort: 9849
name: raft-rpc
- containerPort: 7848
name: old-raft-rpc
env:
- name: NACOS_REPLICAS
value: "1"
- name: MYSQL_SERVICE_HOST
valueFrom:
configMapKeyRef:
name: nacos-cm
key: mysql.host
- name: MYSQL_SERVICE_DB_NAME
valueFrom:
configMapKeyRef:
name: nacos-cm
key: mysql.db.name
- name: MYSQL_SERVICE_PORT
valueFrom:
configMapKeyRef:
name: nacos-cm
key: mysql.port
- name: MYSQL_SERVICE_USER
valueFrom:
configMapKeyRef:
name: nacos-cm
key: mysql.user
- name: MYSQL_SERVICE_PASSWORD
valueFrom:
configMapKeyRef:
name: nacos-cm
key: mysql.password
- name: MODE
value: "cluster"
- name: NACOS_SERVER_PORT
value: "8848"
- name: PREFER_HOST_MODE
value: "hostname"
- name: MODE
value: "standalone"
- name: NACOS_SERVERS
value: "nacos-0.nacos-headless.a-b-test.svc.cluster.local:8848"
selector:
matchLabels:
app: nacos
```

图 4-20 创建 nacos 服务 1



创建nacos-service

图 4-21 创建 nacos-service



- 上传配置文件“nacos_config_yst.zip”
- 修改MQ、Redis、Mysql相关配置

图 4-22 修改 1

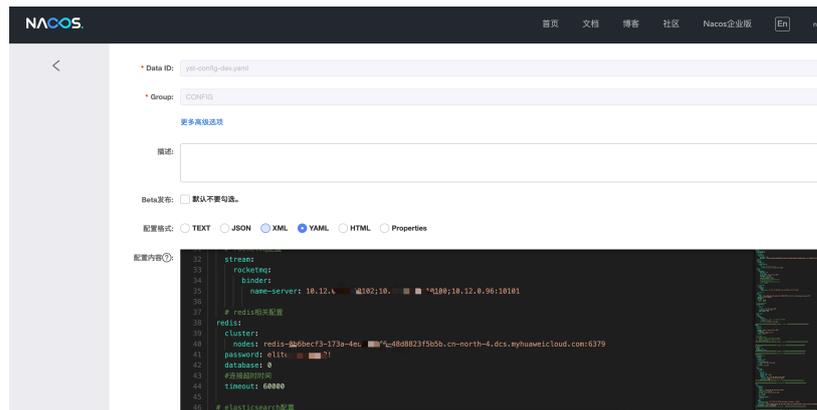
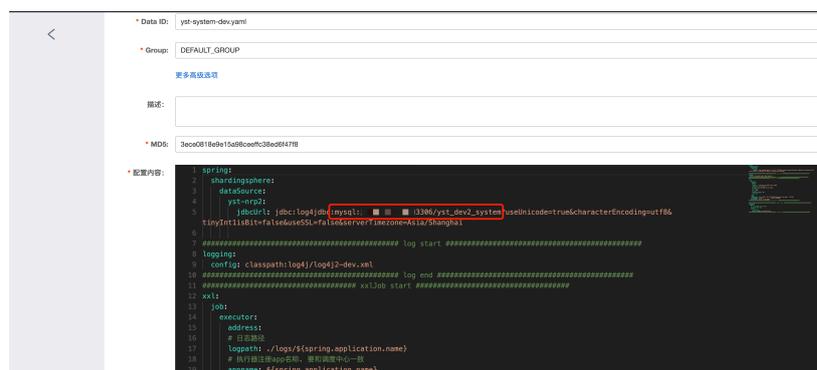


图 4-23 修改 2



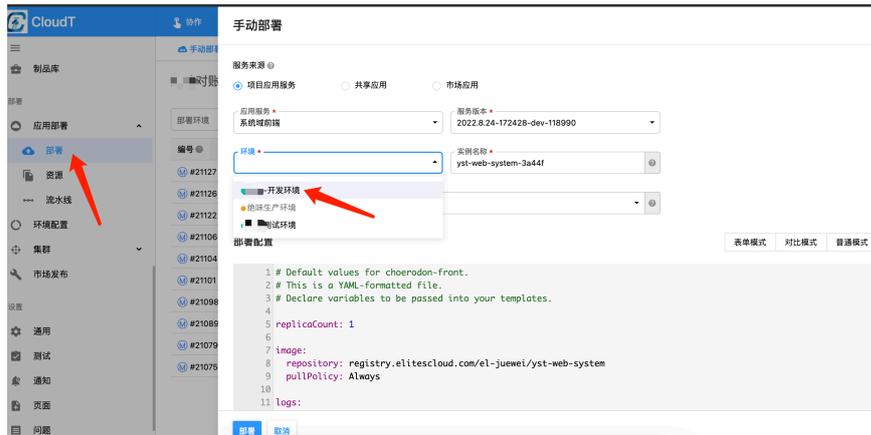
部署云时通中台系统-微服务应用

表 4-1 微服务应用列表

序号	名称/应用	应用描述	集群数量	内存 limit/G
1	yst-workflow	工作流	2	5
2	yst-sale	销售中心	2	5
3	yst-order	订单中心	2	5
4	yst-inv	库存中心	2	5
5	yst-pur	采购中心	2	5
6	yst-support-domain	支撑域中心	2	5
7	yst-tms	物流中心	2	5
8	yst-fin	财务结算中心	2	5
9	yst-base-gateway	网关	2	5
10	yst-auth	鉴权中心	2	5
11	yst-system	系统中心	2	5
12	coordinator- message	消息中心	1	3
13	coordinator-fsm	附件中心	1	3
14	coordinator-job	调度中心	1	3
15	yst-web	云时通总前端	1	1
16	yst-web-b2b	2b2中心前端	1	1
17	yst-web-oms	oms中心前端	1	1
18	yst-web-fin	结算中心前端	1	1
19	yst-web-supply	供应链前端	1	1
20	yst-web-support	支撑域前端	1	1
21	workflow-web	工作流前端	1	1

- 部署前端服务(eg: yst-system-web)

图 4-24 部署前端服务



- 部署后端服务(eg: yst-system)
选择应用服务、选择版本、并修改nacos连接信息

图 4-25 部署后端服务 1

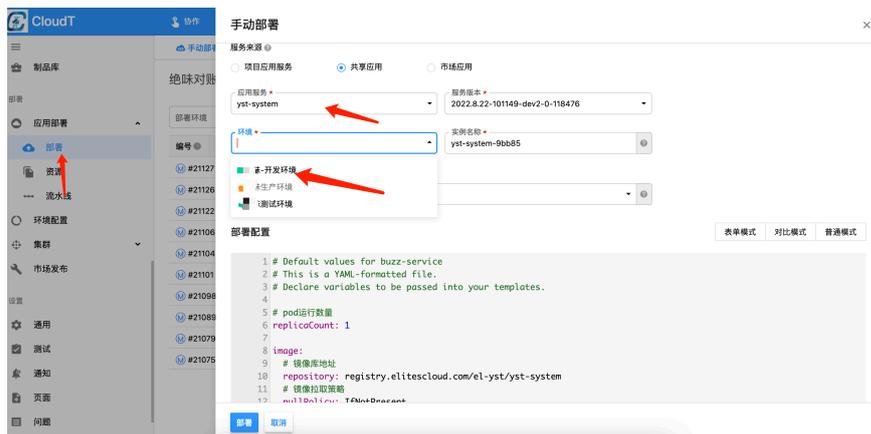


图 4-26 部署后端服务 2

```
41 spring:
42   application:
43     # application.name 请在此处进行显式的指定
44     name: yst-system
45   # 项目的配置文件交由配置中心来进行统一的管理
46   # 项目启动时会加载指定配置中心的命名空间 (默认为 public) 和分组 (默认为 DEFAULT_GROUP) 下的
47   # {application.name}-{active}.{file-extension} 配置文件
48   # 请部署项目时, 确保配种中心中存在对应的配置文件
49   cloud:
50     nacos:
51       config:
52         server:
53           addr: 172.26.1.172:8848
54           enabled: true
55           file-extension: yaml
56           namespace: dev
57           group: DEFAULT_GROUP
58       discovery:
59         enabled: true
60         server:
61           addr: 172.26.1.172:8848
62           register-enabled: true
63           namespace: dev
64           group: DEFAULT_GROUP
```

- 查看应用服务运行情况, 实例事件、Pod详情 变为绿色后, 服务启动完成

图 4-27 查看



- 其他前端、后端应用部署方式一致

4.3 配置 ELB 及验证操作中台系统

配置 ELB

- 购买ELB负载均衡器

图 4-28 选择可用区及子网 VPC

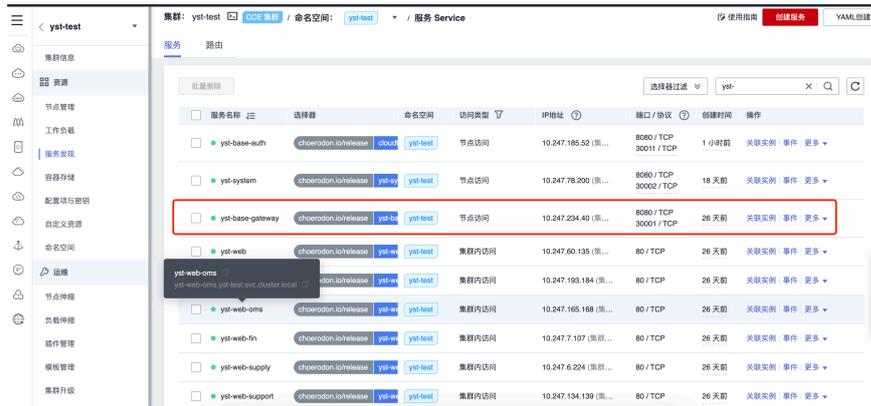


图 4-29 选择规格



- 创建服务发现
base-gateway NodePort 类型的service，前端服务ClusterIp类型的service

图 4-30 创建服务发现



- 创建Ingress 路由
为base-gateway service 配置ingress

图 4-31 创建路由 1

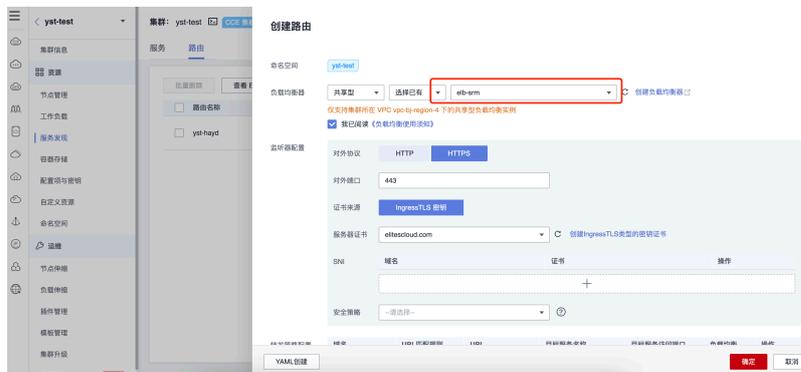


图 4-32 创建路由 2



配置域名解析，将域名解析道ELB公网IP或内网IP

验证操作中台系统

登录平台进行验证

图 4-33 登录系统



5 修订记录

发布日期	修订记录
2024-04-28	规范词、敏感词专项处理，章节优化
2022-09-19	第一次正式发布。