

K8s管理系统项目实战【前端开发】

讲师：阿杜

一、项目概述

- 本节课程是k8s管理系统项目实战的前端开发部分，在完成API接口的整体开发后，我们开始着手于前端部分，构建一个功能页面，将管理系统平台化。
- 前端部分使用vue3框架以及ant-design组件库完成，开发过程中，我们还会使用到以下依赖：
 - xterm 命令行终端模拟器
 - xterm-addon-fit xterm的终端尺寸插件
 - nprogress 浏览器顶部的进度条
 - jsonwebtoken token的生成与校验组件
 - codemirror-editor-vue3 代码编辑器，用于编辑k8s资源YAML
 - js-yaml 用于yaml转对象
 - json2yaml 用于json转yaml
 - moment 获取当前时间的组件
 - axios http请求组件

二、项目初始化

2.1 启动配置

- `vue.config.js`

```
module.exports = {
  devServer: {
    host: '0.0.0.0', // 监听地址
    port: 7070, // 启动端口号
    open: true // 启动后是否自动打开网页
  },
  // 关闭语法检测
  lintOnSave: false
}
```

2.2 封装Axios

- 封装axios请求，添加自定义配置，如超时、重试、header等等
- `src/request/index.js`

```
import axios from 'axios';

// 新建个axios对象
```

```

const httpClient = axios.create({
  validateStatus(status) {
    return status ≥ 200 && status < 504 // 设置默认的合法的状态
  },
  timeout: 20000 // 超时时间10秒
});

httpClient.defaults.retry = 3 // 请求重试次数
httpClient.defaults.retryDelay = 1000 // 请求重试时间间隔
httpClient.defaults.shouldRetry = true // 是否重试

// 添加请求拦截器
httpClient.interceptors.request.use(
  config => {
    // 添加header
    // config.headers['Content-Type'] = 'application/json'
    config.headers['Accept-Language'] = 'zh-CN'
    config.headers['Authorization'] = localStorage.getItem('token') // 可以全局设置
    // 接口请求header中带token

    if (config.method === 'post') {
      if (!config.data) { // 没有参数时, config.data为null, 需要转下类型
        config.data = {}
      }
      // config.data = JSON.stringify(config.data) // qs序列化参数
    }
    return config
  },
  err => {
    // Promise.reject()方法返回一个带有拒绝原因的Promise对象, 在F12的console中显示报错
    Promise.reject(err)
  }
);

// 添加响应拦截器
httpClient.interceptors.response.use(
  response => {
    if (response.status !== 200) {
      if (!response.data.msg) {
        let res = {
          data: response.data,
          msg: "服务端异常: " + response.status + " " + response.statusText
        }
        return Promise.reject(res)
      } else {
        return Promise.reject(response.data)
      }
    } else {
      return response.data
    }
  }
);

```

```

    }
  },
  err => {
    return Promise.reject(err)
  }
};

export default httpClient;

```

2.3 配置后端路由

- 前端所有配置统一管理和维护。
- `src/config/index.js`

```

const baseHost = 'http://localhost:9090'
export default {
  // 登录
  // 所有资源数据
  k8sAllRes: baseHost + '/api/k8s/allres',
  loginAuth: baseHost + '/api/login',
  // cluster
  k8sClusterList: baseHost + '/api/k8s/clusters',
  // event
  k8sEventList: baseHost + '/api/k8s/events',
  // namespace
  k8sNamespaceList: baseHost + '/api/k8s/namespaces',
  k8sNamespaceDetail: baseHost + '/api/k8s/namespace/detail',
  k8sNamespaceDel: baseHost + '/api/k8s/namespace/del',
  // deployment
  k8sDeploymentList: baseHost + '/api/k8s/deployments',
  k8sDeploymentDetail: baseHost + '/api/k8s/deployment/detail',
  k8sDeploymentUpdate: baseHost + '/api/k8s/deployment/update',
  k8sDeploymentScale: baseHost + '/api/k8s/deployment/scale',
  k8sDeploymentRestart: baseHost + '/api/k8s/deployment/restart',
  k8sDeploymentDel: baseHost + '/api/k8s/deployment/del',
  k8sDeploymentCreate: baseHost + '/api/k8s/deployment/create',
  k8sDeploymentNumNp: baseHost + '/api/k8s/deployment/numnp',
  // pod
  k8sPodList: baseHost + '/api/k8s/pods',
  k8sPodDetail: baseHost + '/api/k8s/pod/detail',
  k8sPodUpdate: baseHost + '/api/k8s/pod/update',
  k8sPodDel: baseHost + '/api/k8s/pod/del',
  k8sPodContainer: baseHost + '/api/k8s/pod/container',
  k8sPodLog: baseHost + '/api/k8s/pod/log',
  k8sPodNumNp: baseHost + '/api/k8s/pod/numnp',
  k8sTerminalWs: 'ws://localhost:8081/ws',
  // ingress
  k8sIngressList: baseHost + '/api/k8s/ingresses',
  k8sIngressDetail: baseHost + '/api/k8s/ingress/detail',

```

```
k8sIngressUpdate: baseHost + '/api/k8s/ingress/update',
k8sIngressDel: baseHost + '/api/k8s/ingress/del',
k8sIngressCreate: baseHost + '/api/k8s/ingress/create',
//service
k8sSvcList: baseHost + '/api/k8s/services',
k8sSvcDetail: baseHost + '/api/k8s/service/detail',
k8sSvcUpdate: baseHost + '/api/k8s/service/update',
k8sSvcDel: baseHost + '/api/k8s/service/del',
k8sSvcCreate: baseHost + '/api/k8s/service/create',
//daemonset
k8sDaemonSetList: baseHost + '/api/k8s/daemonsets',
k8sDaemonSetDetail: baseHost + '/api/k8s/daemonset/detail',
k8sDaemonSetUpdate: baseHost + '/api/k8s/daemonset/update',
k8sDaemonSetDel: baseHost + '/api/k8s/daemonset/del',
//statefulset
k8sStatefulSetList: baseHost + '/api/k8s/statefulsets',
k8sStatefulSetDetail: baseHost + '/api/k8s/statefulset/detail',
k8sStatefulSetUpdate: baseHost + '/api/k8s/statefulset/update',
k8sStatefulSetDel: baseHost + '/api/k8s/statefulset/del',
//node
k8sNodeList: baseHost + '/api/k8s/nodes',
k8sNodeDetail: baseHost + '/api/k8s/node/detail',
//pv
k8sPvList: baseHost + '/api/k8s/pvs',
k8sPvDetail: baseHost + '/api/k8s/pv/detail',
k8sPvDel: baseHost + '/api/k8s/pv/del',
//configmap
k8sConfigmapList: baseHost + '/api/k8s/configmaps',
k8sConfigmapDetail: baseHost + '/api/k8s/configmap/detail',
k8sConfigmapUpdate: baseHost + '/api/k8s/configmap/update',
k8sConfigmapDel: baseHost + '/api/k8s/configmap/del',
//secret
k8sSecretList: baseHost + '/api/k8s/secrets',
k8sSecretDetail: baseHost + '/api/k8s/secret/detail',
k8sSecretUpdate: baseHost + '/api/k8s/secret/update',
k8sSecretDel: baseHost + '/api/k8s/secret/del',
//pvc
k8sPvctList: baseHost + '/api/k8s/pvcs',
k8sPvcDetail: baseHost + '/api/k8s/pvc/detail',
k8sPvcUpdate: baseHost + '/api/k8s/pvc/update',
k8sPvcDel: baseHost + '/api/k8s/pvc/del',
//release
helmReleaseList: baseHost + '/api/helmstore/releases',
helmReleaseDetail: baseHost + '/api/helmstore/release/detail',
helmReleaseInstall: baseHost + '/api/helmstore/release/install',
helmReleaseUninstall: baseHost + '/api/helmstore/release/uninstall',
//chart
helmChartList: baseHost + '/api/helmstore/charts',
helmChartAdd: baseHost + '/api/helmstore/chart/add',
helmChartUpdate: baseHost + '/api/helmstore/chart/update',
```

```

helmChartDel: baseHost + '/api/helmstore/chart/del',
helmChartFileUpload: baseHost + '/api/helmstore/chartfile/upload',
helmChartFileDel: baseHost + '/api/helmstore/chartfile/del',
}

```

三、页面开发

3.1 整体布局-集群动态数据



- `src/layout/Layout.vue`
- `template`, 绑定`change`事件

```

<a-menu-item v-for="(item) in clusterList" :key="item" @click="clusterChange(item)">
  {{ item }}</a-menu-item>

```

- `javascript`

```

// 集群列表
const clusterList = ref([])
const clusterListData = reactive({
  url: common.k8sClusterList,
})
// 获取列表
const getClusterList = () => {
  appLoading.value = true
  httpClient.get(clusterListData.url)
    .then(res => {
      // 响应成功, 获取deployment列表和total
      clusterList.value = res.data
      localStorage.setItem('cluster_num', clusterList.value.length)
      if (!selectedKeys1.value.length) {
        selectedKeys1.value[0] = clusterList.value[0]
        localStorage.setItem('k8s_cluster', clusterList.value[0])
      }
    })
    .catch(res => {
      message.error(res.msg)
    })
    .finally(() => {
      appLoading.value = false
    })
}
// 处理集群变更
const clusterChange = (val) => {
  if (selectedKeys1.value[0] = val) {

```

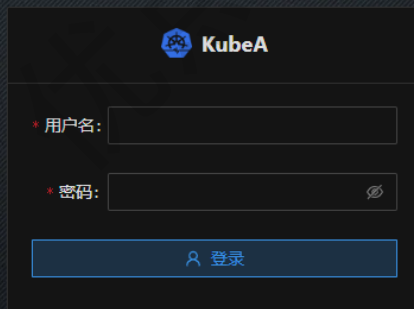
```

        return
    } else {
        selectedKeys1.value[0] = val
        localStorage.setItem('k8s_cluster', val)
        //router.push(router.currentRoute.value.path)
        location.replace(router.currentRoute.value.path)
    }
}

//生命周期钩子
onBeforeMount(() => {
    getClusterList()
    if (localStorage.getItem('k8s_cluster')) {
        selectedKeys1.value[0] = localStorage.getItem('k8s_cluster')
    }
})

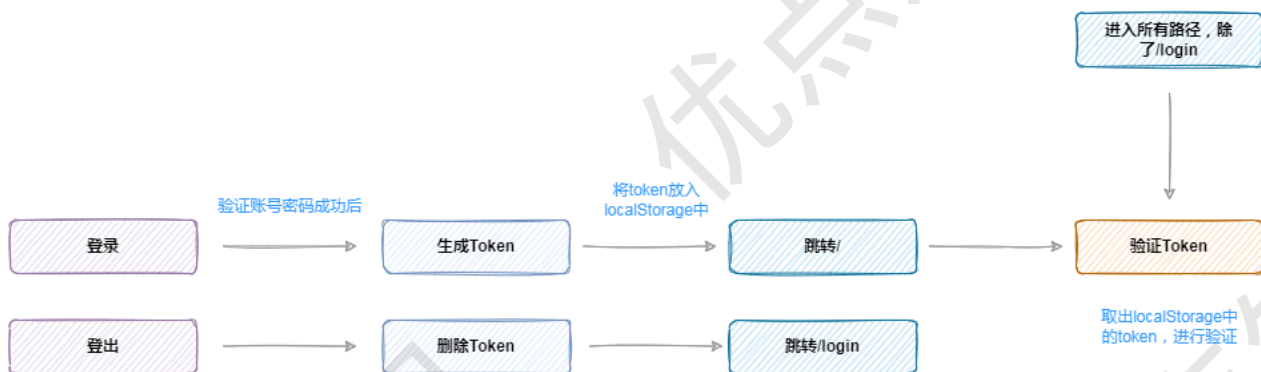
```

3.2 登录页



The image shows a login page for KubeA. It features a dark background with a central white login form. The form has a header with the KubeA logo and name. Below the header, there are two input fields: one for the username (labeled '用户名:') and one for the password (labeled '密码:'). The password field has a toggle icon on the right. At the bottom of the form is a blue button with a user icon and the text '登录' (Login).

(1) 登录/登出逻辑



(2) 添加路由，创建文件 `src/views/common/Login.vue`

```

{
  path: '/login', //url路径
  component: () => import('@views/common/Login.vue'), //视图组件
  meta: {title: "登录", requireAuth: false}, //meta元信息
},

```

(3) 登录页代码

- `src/views/common/Login.vue`

```

<template>
  <div class="login">
    <a-card class="login-card" :headStyle="{padding:'0px'}" :bodyStyle="{padding:'20px'}">
      <template #title>
        <div class="login-card-header">
          
          <span>KubeA</span>
        </div>
      </template>
      <a-form ref="formRef" :model="loginData" :label-col="{ span: 5 }">
        <a-form-item
          label="用户名"
          name="username"
          :rules="[{ required: true, message: '请输入账号' }]">
          <a-input v-model:value="loginData.username" />
        </a-form-item>
        <a-form-item
          label="密码"
          name="password"
          :rules="[{ required: true, message: '请输入密码' }]">
          <a-input-password v-model:value="loginData.password" />
        </a-form-item>
        <a-form-item style="text-align:center;margin-bottom:10px;">

```

```

        <c-button style="width:100%;" type="primary" size="normal"
icon="user-outlined" @click="onCheck()">登录</c-button>
      </a-form-item>
    </a-form>
  </a-card>
</div>
</template>

```

```

<script>
import kubeLogo from '@assets/k8s-metrics.png';
import { reactive, ref } from 'vue';
import { message } from 'ant-design-vue';
import { useRouter } from 'vue-router'
import httpClient from '@request';
import common from "@config";
import jwt from 'jsonwebtoken';
import moment from 'moment';
export default({
  setup() {
    const formRef = ref()
    const loginData = reactive({
      username: '',
      password: ''
    })
    const loginUrl = common.loginAuth

    const router = useRouter()

    const onCheck = async () => {
      try {
        await formRef.value.validateFields();
        //console.log('Success:', values);
        handleLogin()
      } catch (errorInfo) {
        console.log('Failed:', errorInfo);
      }
    }

    const handleLogin = () => {
      httpClient.post(loginUrl, loginData)
        .then(res => {
          localStorage.setItem('username', loginData.username);
          localStorage.setItem('loginDate', moment().format('YYYY-MM-DD
HH:mm:ss'));

          let token = jwt.sign(loginData, 'adoodevops', { expiresIn: '10h' });
          localStorage.setItem('token', token);
          router.push('/');
          message.success(res.msg)
        })
        .catch(res => {
          message.error(res.msg)
        })
    }
  }
})

```



```

    })
  }
  return {
    kubeLogo,
    loginData,
    formRef,
    handleLogin,
    onCheck
  }
},
})
</script>

<style scoped>
  .login {
    height: 100vh;
    background-image: url(../assets/login3.webp);
    background-size: 100%;
    display: grid;
    place-items: center;
  }
  .login-card {
    width: 350px;
  }
  .login-card-header {
    font-size: 18px;
    font-weight: bold;
    text-align: center;
  }
</style>

```

(4) 添加登录验证的路由守卫

- `src/request/index.js`

```

//导入jwt token组件
import jwt from 'jsonwebtoken'

//使用钩子函数对路由进行权限跳转
router.beforeEach((to, from, next) => {
  //验证jwt token是否合法
  jwt.verify(localStorage.getItem('token'), 'adoodevops', function (err) {
    if (to.path === '/login') {
      next()
    } else if (err) {
      next('/login');
    } else {
      next();
    }
  })
})

```

```
});  
});
```

3.3 组件开发

1. Header组件

命名空间: kube-system

请输入

搜索

刷新

- 知识点: 父传子属性使用 `props`, 子传父事件使用 `emit`
- `src/components/mainHeader/index.vue`

```
<template>  
  <div>  
    <a-card :bodyStyle="{padding: '10px'}">  
      <a-row>  
        <a-col :span="20">  
          <div style="text-align:left;">  
            <span v-if="namespace" style="font-size:14px;">命名空间:  
          </span>  
            <a-select v-if="namespace" @change="nsChange" show-search  
style="width:140px;margin-right:20px;" size="small" v-model:value="namespaceValue"  
placeholder="请选择">  
              <a-select-option  
                v-for="(item, index) in namespaceList"  
                :key="index"  
                :value="item.metadata.name">  
                {{ item.metadata.name }}  
              </a-select-option>  
            </a-select>  
            <a-input allow-clear @change="searchChange"  
style="width:200px;margin-right:10px;" v-model:value="searchValue" size="small"  
clearable :placeholder="searchDescribe">  
            </a-input>  
            <a-button size="small" type="primary" ghost  
@click="$emit('dataList')">  
              <template #icon><SearchOutlined /></template>  
              搜索  
            </a-button>  
          </div>  
        </a-col>  
        <a-col :span="4">  
          <div style="text-align:right;">  
            <a-button v-if="add" style="margin-right:10px;" size="small"  
type="primary" ghost @click="$emit('addFunc')">  
              <template #icon><PlusOutlined /></template>  
              新增
```

```

        </a-button>
        <a-button size="small" @click="$emit('dataList')">
            <template #icon><UndoOutlined /></template>
            刷新
        </a-button>
    </div>
</a-col>
</a-row>
</a-card>
</div>
</template>

<script>
import { onMounted, reactive, ref } from 'vue';
import common from "@/@config";
import httpClient from '@/@request';
import { message } from 'ant-design-vue';
export default ({
    name: 'MainHead',
    // 声明父传子属性
    props: {
        namespace: { type: Boolean, default: false },
        searchDescribe: { type: String, default: '' },
        add: { type: Boolean, default: false },
    },
    // 声明子传父事件
    emits: ['searchChange', 'namespaceChange', 'namespaceList', 'dataList',
    'addFunc'],
    setup(props, ctx) {
        const searchValue = ref('')
        const namespaceValue = ref('default')
        // 命名空间列表
        const namespaceList = ref([])
        const namespaceTotal = ref(0)
        const namespaceListData = reactive({
            url: common.k8sNamespaceList,
            params: {
                cluster: ''
            }
        })
        // 【监听】
        // watch(namespaceValue, (newVal) => {
        //     localStorage.setItem("namespace", newVal)
        //     ctx.emit('namespaceChange', newVal)
        //     ctx.emit('dataList')
        //     console.log(localStorage.getItem("namespace"), 456)
        // }, {immediate:true})
        // 【方法】
        const nsChange = (val) => {
            namespaceValue.value = val

```

```

        localStorage.setItem('namespace', val)
        ctx.emit('namespaceChange', val)
        ctx.emit('dataList')
    }
    const searchChange = () => {
        ctx.emit('searchChange', searchValue.value)
    }
    // 获取命名空间列表
    const getNamespaceList = () => {
        namespaceListData.params.cluster = localStorage.getItem('k8s_cluster')
        httpClient.get(namespaceListData.url, {params: namespaceListData.params})
            .then(res => {
                namespaceList.value = res.data.items
                namespaceTotal.value = res.data.total
                ctx.emit('namespaceList', namespaceList.value)
            })
            .catch(res => {
                message.error(res.msg)
            })
    }
    onMounted(() => {
        if (localStorage.getItem('namespace')) {
            nsChange(localStorage.getItem('namespace'))
        } else {
            nsChange(namespaceValue.value)
        }
        getNamespaceList()
    })
    return {
        searchValue,
        namespaceValue,
        namespaceList,
        nsChange,
        searchChange
    },
    {}
})
</script>

<style scoped>
    .ant-btn {
        border-radius: 1px;
    }
</style>

```

2. Button组件



- `src/components/CButton/index.vue`

```
<template>
  <a-button
    :type="customType"
    :class="customClass"
    :size="customSize"
    :disabled="disabled"
  >
    <template v-if="iconType" #icon>
      <component :is="iconType" />
    </template>
    <slot />
  </a-button>
</template>
<script>
import { defineComponent, ref, watch } from 'vue'
export default defineComponent({
  name: 'CButtonIndex',
  props: {
    type: { type: String, default: '' },
    size: { type: String, default: '' },
    icon: { type: String, default: '' },
    disabled: { type: Boolean, default: false },
    permission: { type: [String, Boolean], default: true }
  },
  setup (props) {
    const customClass = ref('c-button-primary')
    const customType = ref('')
    const customSize = ref('small')
    const iconType = ref('')
    watch(() => props.type, (v) => {
      switch (v) {
        case 'warning':
          customClass.value = 'c-button-warning'
          customType.value = 'default'
          break
        case 'error':
          customClass.value = 'c-button-error'
          customType.value = 'default'
          break
        case 'success':
          customClass.value = 'c-button-success'

```

```

        customType.value = 'default'
        break
      case 'primary':
        customClass.value = 'c-button-primary'
        customType.value = 'primary'
        break
      case 'cyan':
        customClass.value = 'c-button-cyan'
        customType.value = 'default'
        break
      case 'black':
        customClass.value = 'c-button-black'
        customType.value = 'default'
        break
      case 'purple':
        customClass.value = 'c-button-purple'
        customType.value = 'default'
        break
      case 'text':
        customClass.value = ''
        customType.value = 'text'
        break
      case 'link':
        customClass.value = ''
        customType.value = 'link'
        break
      default:
        customClass.value = ''
        customType.value = 'default'
        break
    }
  }, { immediate: true })
  watch(() => props.size, v => {
    customSize.value = !v ? 'small' : v
  }, { immediate: true })
  watch(() => props.icon, v => {
    iconType.value = v
  }, { immediate: true })
  watch(() => props, () => {}, { immediate: true })
  return {
    customClass,
    customType,
    customSize,
    iconType
  }
}
})
</script>
<style scoped>
.c-button-primary {

```



```
color: #3991e3;
background-color: #3991e33a;
border-color: #3991e3;
}
.c-button-primary:hover {
color: #fff;
background-color: #3991e389;
border-color: #3991e389;
}
.c-button-primary[disabled], .c-button-primary[disabled]:hover, .c-button-
primary[disabled]:focus, .c-button-primary[disabled]:active {
color: rgba(0, 0, 0, 0.25);
background: #f5f5f5;
border-color: #d9d9d9;
text-shadow: none;
box-shadow: none;
}

.c-button-warning {
color: #10c376;
background-color: #12bce225;
border-color: #10c376;
}
.c-button-warning:hover {
color: #fff;
background-color: #10c37576;
border-color: #10c37576;
}
.c-button-warning[disabled], .c-button-warning[disabled]:hover, .c-button-
warning[disabled]:focus, .c-button-warning[disabled]:active {
color: rgba(0, 0, 0, 0.25);
background: #f5f5f5;
border-color: #d9d9d9;
text-shadow: none;
box-shadow: none;
}

.c-button-error {
color: #a3a2a3;
background-color: #84808238;
border-color: #a3a2a3;
}
.c-button-error:hover {
color: #fff;
background-color: #8480826d;
border-color: #8480826d;
}
.c-button-error[disabled], .c-button-error[disabled]:hover, .c-button-
error[disabled]:focus, .c-button-error[disabled]:active {
color: rgba(0, 0, 0, 0.25);
```

```
background: #f5f5f5;
border-color: #d9d9d9;
text-shadow: none;
box-shadow: none;
}

.c-button-success {
  color: #fff;
  background-color: #00cc66;
  border-color: #00cc66;
}
.c-button-success:hover {
  color: #fff;
  background-color: #03e071;
  border-color: #00cc66;
}
.c-button-success[disabled], .c-button-success[disabled]:hover, .c-button-
success[disabled]:focus, .c-button-success[disabled]:active {
  color: rgba(0, 0, 0, 0.25);
  background: #f5f5f5;
  border-color: #d9d9d9;
  text-shadow: none;
  box-shadow: none;
}

.c-button-cyan {
  color: #fff;
  background-color: #04c1e1;
  border-color: #04c1e1;
}
.c-button-cyan:hover {
  color: #fff;
  background-color: #0ad5f8;
  border-color: #04c1e1;
}
.c-button-cyan[disabled], .c-button-cyan[disabled]:hover, .c-button-
cyan[disabled]:focus, .c-button-cyan[disabled]:active {
  color: rgba(0, 0, 0, 0.25);
  background: #f5f5f5;
  border-color: #d9d9d9;
  text-shadow: none;
  box-shadow: none;
}

.c-button-black {
  color: #fff;
  background-color: #131313;
  border-color: #131313;
}
.c-button-black:hover {
```

```
color: #fff;
background-color: #313131;
border-color: #131313;
}
.c-button-black[disabled], .c-button-black[disabled]:hover, .c-button-
black[disabled]:focus, .c-button-black[disabled]:active {
color: rgba(0, 0, 0, 0.25);
background: #f5f5f5;
border-color: #d9d9d9;
text-shadow: none;
box-shadow: none;
}

.c-button-purple {
color: #c34bf2;
background-color: #c589dd54;
border-color: #c34bf2;
}
.c-button-purple:hover {
color: #fff;
background-color: #c951fa;
border-color: #B500FE;
}
.c-button-purple[disabled], .c-button-purple[disabled]:hover, .c-button-
purple[disabled]:focus, .c-button-purple[disabled]:active {
color: rgba(0, 0, 0, 0.25);
background: #f5f5f5;
border-color: #d9d9d9;
text-shadow: none;
box-shadow: none;
}
.ant-btn {
border-radius: 1px;
}
</style>
```

3.4 概览

1. 集群信息



- `src/views/home/Home.vue`
- `template`

```
<template>
  <div>
    <a-spin :spinning="appLoading">
      <a-collapse v-model:activeKey="activeKey" ghost>
        <a-collapse-panel style="font-size:18px;" key="1" header="概览">
          <a-row class="home-row1">
            <a-col :span="7">
              <div style="font-size:13px;">名称</div>
              <div style="text-align:center;font-size:30px;font-weight:bold;">KubeA</div>
            </a-col>
            <a-divider type="vertical" style="height: 70px; background-color: rgb(89, 104, 173)" />
            <a-col :span="7">
              <div style="font-size:13px;">集群数</div>
              <div style="text-align:center;font-size:30px;font-weight:bold;">{{ clusterNum }}</div>
            </a-col>
            <a-divider type="vertical" style="height: 70px; background-color: rgb(89, 104, 173)" />
            <a-col :span="7">
              <div style="font-size:13px;">版本</div>
              <div style="text-align:center;font-size:30px;font-weight:bold;">v1.0.0</div>
            </a-col>
          </a-row>
          <a-row :gutter="10">
            <template v-for="val,key in resourceList" :key="key">
              <a-col :span="4" style="margin-bottom:10px;">
                <a-card :bordered="false" style="background-color:rgb(33, 46, 64)" :bodyStyle="{padding: '10px'}">
                  <div style="float:left;margin:15px 10px 0px">
                    <a-progress
```

```

        :width="40"
        :strokeWidth="18"
        :stroke-color="color"
        :percent="100"
        :showInfo="false"
        type="circle"
      />
    </div>
    <div style="text-align:center;">
      <span style="font-size:14px;">{{ key }}

      <br/>
      <span style="font-size:30px;font-weight:bold;">{{ resourceList[key] }}</span>
    </div>
  </a-card>
</a-col>
</template>
</a-row>
</a-collapse-panel>
<a-collapse-panel style="font-size:18px;" key="2" header="事件">
</a-collapse-panel>
</a-collapse>
</a-spin>
</div>
</template>

```

- javascript

```

<script>
import { onBeforeUnmount, onMounted, reactive, ref } from 'vue';
import httpClient from '@/request';
import common from "@/config";
import { message } from 'ant-design-vue';
export default ({
  setup() {
    const appLoading = ref(false)
    const activeKey = ref(['1', '2'])

    const color = reactive({
      '0%': '#73b9bc',
      '100%': '#109ee9',
    })
    const clusterNum = ref()
    // 定时器
    let timer = null
    // 资源数据
    const resourceList = ref()
    const resourceListData = reactive({
      url: common.k8sAllRes,

```

```

        params: {
            cluster: ''
        }
    })
    function getAllRes() {
        appLoading.value = true
        resourceListData.params.cluster = localStorage.getItem('k8s_cluster')
        httpClient.get(resourceListData.url, {params: resourceListData.params})
            .then(res => {
                resourceList.value = res.data
            })
            .catch(res => {
                message.error(res.msg)
            })
            .finally (() => {
                appLoading.value = false
            })
    }
    //生命周期钩子
    onMounted(() => {
        timer = setInterval(() => {
            if (localStorage.getItem('cluster_num') &&
                localStorage.getItem('k8s_cluster')) {
                clusterNum.value = localStorage.getItem('cluster_num')
                getAllRes()
            }
        }, 100)
    })
    onBeforeUnmount(() => {
        if (timer) {
            clearInterval(timer)
            timer = null
        }
    })

    return {
        appLoading,
        activeKey,
        clusterNum,
        color,
        resourceList,
    },
    },
    })
</script>

```

- CSS


```

<style scoped>
  .home-row1 {
    background-color: rgb(40, 46, 58) !important;
    padding: 20px;
    margin-bottom: 15px;
  }
</style>

```

2. Event表格

事件

资源名	资源类型	命名空间	状态	原因	描述	事件时间	集群
kube-scheduler	Lease	kube-system	Normal	LeaderElection	master1_fe310823-4860-46f8-9475-9c35d92f6098 became leader	2023-01-01 14:56:31	TST-1
kube-proxy	DaemonSet	kube-system	Normal	SuccessfulCreate	Created pod: kube-proxy-5jjs9	2023-01-01 14:58:16	TST-1
kube-proxy	DaemonSet	kube-system	Normal	SuccessfulCreate	Created pod: kube-proxy-mlc89	2023-01-01 14:57:45	TST-1
kube-proxy	DaemonSet	kube-system	Normal	SuccessfulCreate	Created pod: kube-proxy-55rrv	2023-01-01 14:56:44	TST-1

- `src/views/home/Home.vue`
- `template`

```

<a-collapse-panel style="font-size:18px;" key="2" header="事件">
  <div style="text-align:right;margin-bottom:10px;">
    <a-input-search
      allow-clear
      v-model:value="searchValue"
      placeholder="请输入资源名"
      style="width: 200px"
      @search="getEventList"
    />
  </div>
  <a-card :bodyStyle="{padding: '10px'}">
    <a-table
      style="font-size:12px;"
      :loading="appLoading"
      :columns="columns"
      :data-source="eventList"
      :pagination="pagination"
      @change="handleTableChange">

```

```

        <template #bodyCell="{ column, record }">
            <template v-if="column.dataIndex === 'name'">
                <a style="color: rgb(84, 138, 238);font-weight:
bold;">{{ record.name }}</a>
            </template>
            <template v-if="column.dataIndex === 'event_time'">
                <span>{{ timeTrans(record.event_time) }}</span>
            </template>
        </template>
    </a-table>
</a-card>
</a-collapse-panel>

```

- javascript

```

const searchValue = ref()
const columns = ref([
  {
    title: '资源名',
    dataIndex: 'name',
  },
  {
    title: '资源类型',
    dataIndex: 'kind',
    width: 150,
  },
  {
    title: '命名空间',
    dataIndex: 'namespace',
  },
  {
    title: '状态',
    dataIndex: 'rtype',
  },
  {
    title: '原因',
    dataIndex: 'reason',
  },
  {
    title: '描述',
    dataIndex: 'message',
    width: 500,
  },
  {
    title: '事件时间',
    dataIndex: 'event_time',
  },
  {
    title: '集群',
    dataIndex: 'cluster',
  },

```

```

    },
  })
  //分页
  const pagination = reactive({
    showSizeChanger: true,
    showQuickJumper: true,
    total: 0,
    currentPage: 1,
    pageSize: 10,
    pageSizeOptions: ["10", "20", "50", "100"],
    showTotal: total ⇒ `共 ${total} 条`
  })
  //事件列表
  const eventList = ref([])
  const eventListData = reactive({
    url: common.k8sEventList,
    params: {
      name: '',
      cluster: '',
      page: 1,
      limit: 10
    }
  })
  //【方法】
  function handleTableChange(val) {
    pagination.currentPage = val.current
    pagination.pageSize = val.pageSize
    getEventList()
  }
  function timeTrans(date) {
    date = date.substring(0, 19).replace('T', ' ')
    date = date.substring(0, 19).replace('+08:00', '')
    return date
  }
  function getEventList () {
    //关闭定时器
    clearInterval(timer)
    timer = null
    if (searchValue.value) {
      pagination.currentPage = 1
    }
    eventListData.params.name = searchValue.value
    eventListData.params.cluster = localStorage.getItem('k8s_cluster')
    eventListData.params.page = pagination.currentPage
    eventListData.params.limit = pagination.pageSize
    httpClient.get(eventListData.url, {params: eventListData.params})
    .then(res ⇒ {
      eventList.value = res.data.items
      pagination.total = res.data.total
    })
  }

```

```

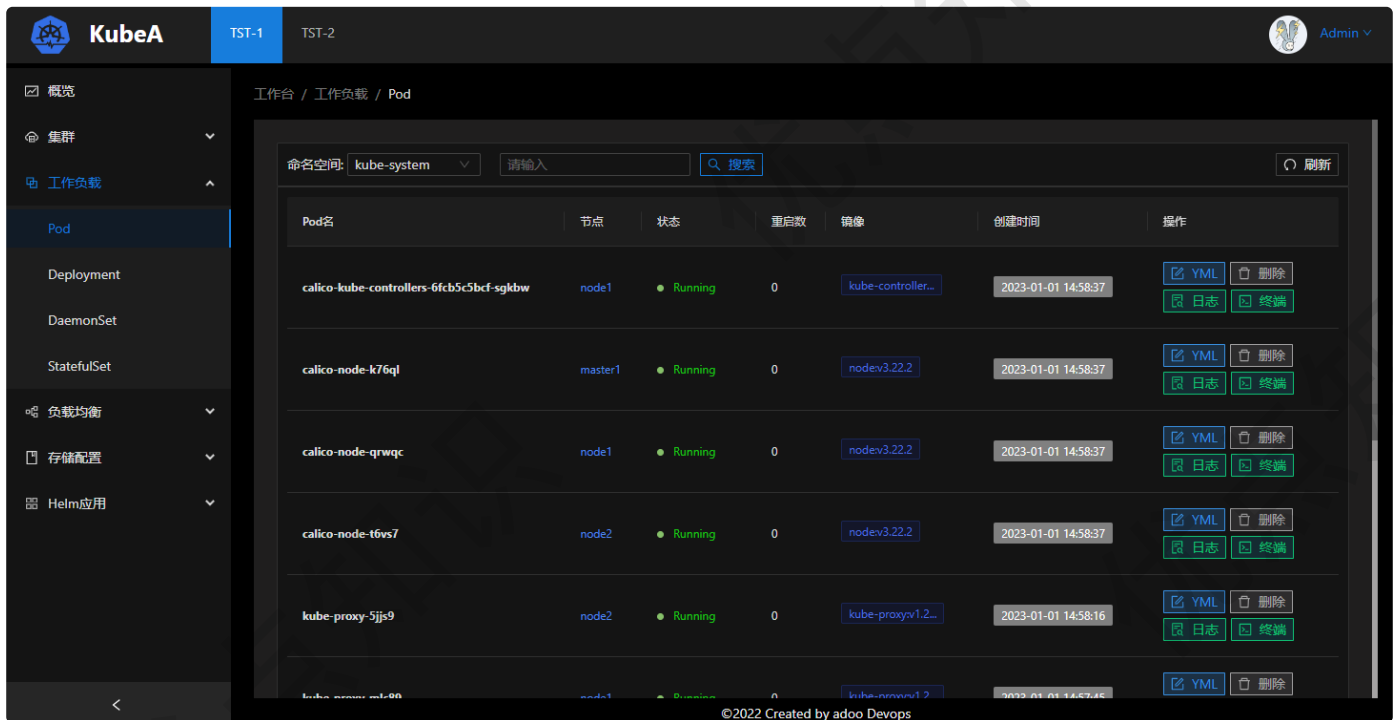
    })
    .catch(res => {
      message.error(res.msg)
    })
  }
  // 生命周期钩子
  onMounted(() => {
    timer = setInterval(() => {
      if (localStorage.getItem('cluster_num') &&
localStorage.getItem('k8s_cluster')) {
        clusterNum.value = localStorage.getItem('cluster_num')
        getEventList()
        getAllRes()
      }
    }, 100)
  })

  return {
    searchValue,
    eventList,
    columns,
    pagination,
    handleTableChange,
    timeTrans,
    getEventList
  }
}

```

3.5 工作负载

1. Pod



- 功能: [列表](#) [详情](#) [编辑](#) [删除](#) [终端](#) [日志](#)
- 数据表格(用于列表展示)

字段	说明
name	Pod名
node	节点
state	状态
restarts	重启数
image	镜像
creationTimestamp	创建时间
action	操作: YML、删除、终端、日志

- `src/views/workLoad/Pod.vue`

(1) 头部工具栏

- `template`

```

<template>
  <div>
    <!-- main部分的header -->
    <MainHead
      searchDescribe="请输入"
      @searchChange="getSearchValue"
      namespace
      @namespaceChange="getNamespaceValue"
      @dataList="getPodList" />
    </div>
  </template>

```

- javascript

```

<script>
import { createVNode, reactive, ref } from 'vue';
import { useRouter } from 'vue-router'
import MainHead from '@components/MainHead';
import httpClient from '@request';
import common from "@config";
import { message } from 'ant-design-vue';
import yaml2obj from 'js-yaml';
import json2yaml from 'json2yaml';
import { ExclamationCircleOutlined } from '@ant-design/icons-vue';
import { Modal } from 'ant-design-vue';
export default({
  components: {
    MainHead,
  },
  // 常用项
  const searchValue = ref('')
  const namespaceValue = ref('')
  function getSearchValue(val) {
    searchValue.value = val
  }
  function getNamespaceValue(val) {
    namespaceValue.value = val
  }
  return {
    getSearchValue,
    getNamespaceValue
  }
})
</script>

```


(2) 列表

- template

```

<!-- 数据表格 -->
<a-card :bodyStyle="{padding: '10px'}">
  <a-table
    style="font-size:12px;"
    :loading="appLoading"
    :columns="columns"
    :dataSource="podList"
    :pagination="pagination"
    @change="handleTableChange">
    <template #bodyCell="{ column, record }">
      <template v-if="column.dataIndex === 'name'">
        <span style="font-weight: bold;">{{ record.metadata.name }}
      </span>
      </template>
      <template v-if="column.dataIndex === 'node'">
        <span style="color: rgb(84, 138, 238);">{{
record.spec.nodeName }}</span>
      </template>
      <template v-if="column.dataIndex === 'state'">
        <div :class="{ 'succ-dot':record.status.phase === 'Running',
'warn-dot':record.status.phase === 'Pending', 'err-dot':record.status.phase !==
'Running' && record.status.phase !== 'Pending'}"></div>
        <span :class="{ 'succ-state':record.status.phase === 'Running',
'warn-state':record.status.phase === 'Pending', 'err-status':record.status.phase !==
'Running' && record.status.phase !== 'Pending'}">{{ record.status.phase }} </span>
      </template>
      <template v-if="column.dataIndex === 'restarts'">
        <span>{{ restartTotal(record) }} </span>
      </template>
      <template v-if="column.dataIndex === 'labels'">
        <div v-for="(val, key) in record.metadata.labels" :key="key">
          <a-popover>
            <template #content>
              <span>{{ key + ":" +val }}</span>
            </template>
            <a-tag style="margin-bottom:5px;cursor:pointer;"
color="blue">{{ ellipsis(key + ":" +val, 15) }}</a-tag>
          </a-popover>
        </div>
      </template>
      <template v-if="column.dataIndex === 'image'">
        <div v-for="(val, key) in record.spec.containers" :key="key">
          <a-popover>
            <template #content>
              <span>{{ val.image }}</span>
            </template>
          </a-popover>
        </div>
      </template>
    </template>
  </a-table>
</a-card>

```

```

<a-tag style="margin-bottom:5px;cursor:pointer;"
color="geekblue">{{ ellipsis(val.image.split('/').pop() ? val.image.split('/').pop()
: val.image, 15 ) }}</a-tag>
</a-popover>
</div>
</template>
<template v-if="column.dataIndex === 'creationTimestamp'">
  <a-tag color="gray">{{
timeTrans(record.metadata.creationTimestamp) }}</a-tag>
</template>
<template v-if="column.key === 'action'">
  <c-button class="pod-button" type="primary" icon="form-
outlined" @click="getPodDetail(record)">YML</c-button>
  <c-button style="margin-bottom:5px;" class="pod-button"
type="error" icon="delete-outlined" @click="showConfirm('删除', record.metadata.name,
delPod)">删除</c-button>
  <c-button class="pod-button" type="warning" icon="file-
search-outlined" @click="gotoLog(record)">日志</c-button>
  <c-button class="pod-button" type="warning" icon="code-
outlined" @click="gotoTerminal(record)">终端</c-button>
</template>
</template>
</a-table>
</a-card>

```

- javascript

```

const columns = ref([
  {
    title: 'Pod名',
    dataIndex: 'name'
  },
  {
    title: '节点',
    dataIndex: 'node'
  },
  {
    title: '状态',
    dataIndex: 'state',
    width: 120
  },
  {
    title: '重启数',
    dataIndex: 'restarts'
  },
  {
    title: '镜像',
    dataIndex: 'image'
  },
])

```

```

        title: '创建时间',
        dataIndex: 'creationTimestamp'
    },
    {
        title: '操作',
        key: 'action',
        fixed: 'right',
        width: 200
    }
])
const appLoading = ref(false)
//分页
const pagination = reactive({
    showSizeChanger: true,
    showQuickJumper: true,
    total: 0,
    currentPage: 1,
    pagesize: 10,
    pageSizeOptions: ["10", "20", "50", "100"],
    showTotal: total => `共 ${total} 条`
})
//列表
const podList = ref([])
const podListData = reactive({
    url : common.k8sPodList,
    params: {
        filter_name: '',
        namespace: '',
        cluster: '',
        page: 1,
        limit: 10
    }
})
//处理时间格式
function timeTrans(timestamp) {
    let date = new Date(new Date(timestamp).getTime() + 8 * 3600 * 1000)
    date = date.toJSON();
    date = date.substring(0, 19).replace('T', ' ')
    return date
}
//处理截断
function ellipsis(val, len) {
    return val.length > len ? val.substring(0, len) + '...' : val
}
//处理重启次数
function restartTotal(e) {
    let index, sum = 0
    let containerStatuses = e.status.containerStatuses
    for (index in containerStatuses) {
        sum = sum + containerStatuses[index].restartCount
    }
}

```

```

    }
    return sum
  }
  //处理翻页
  function handleTableChange(val) {
    pagination.currentPage = val.current
    pagination.pageSize = val.pageSize
    getPodList()
  }
  //列表
  function getPodList() {
    appLoading.value = true
    if (searchValue.value) {
      pagination.currentPage = 1
    }
    podListData.params.filter_name = searchValue.value
    podListData.params.namespace = namespaceValue.value
    podListData.params.cluster = localStorage.getItem('k8s_cluster')
    podListData.params.page = pagination.currentPage
    podListData.params.limit = pagination.pageSize
    httpClient.get(podListData.url, {params: podListData.params})
    .then(res => {
      //响应成功，获取pod列表和total
      podList.value = res.data.items
      pagination.total = res.data.total
    })
    .catch(res => {
      message.error(res.msg)
    })
    .finally(() => {
      appLoading.value = false
    })
  }
  return {
    appLoading,
    pagination,
    columns,
    podList,
    timeTrans,
    ellipsis,
    restartTotal,
    handleTableChange,
    getPodList,
  }
}

```

- css

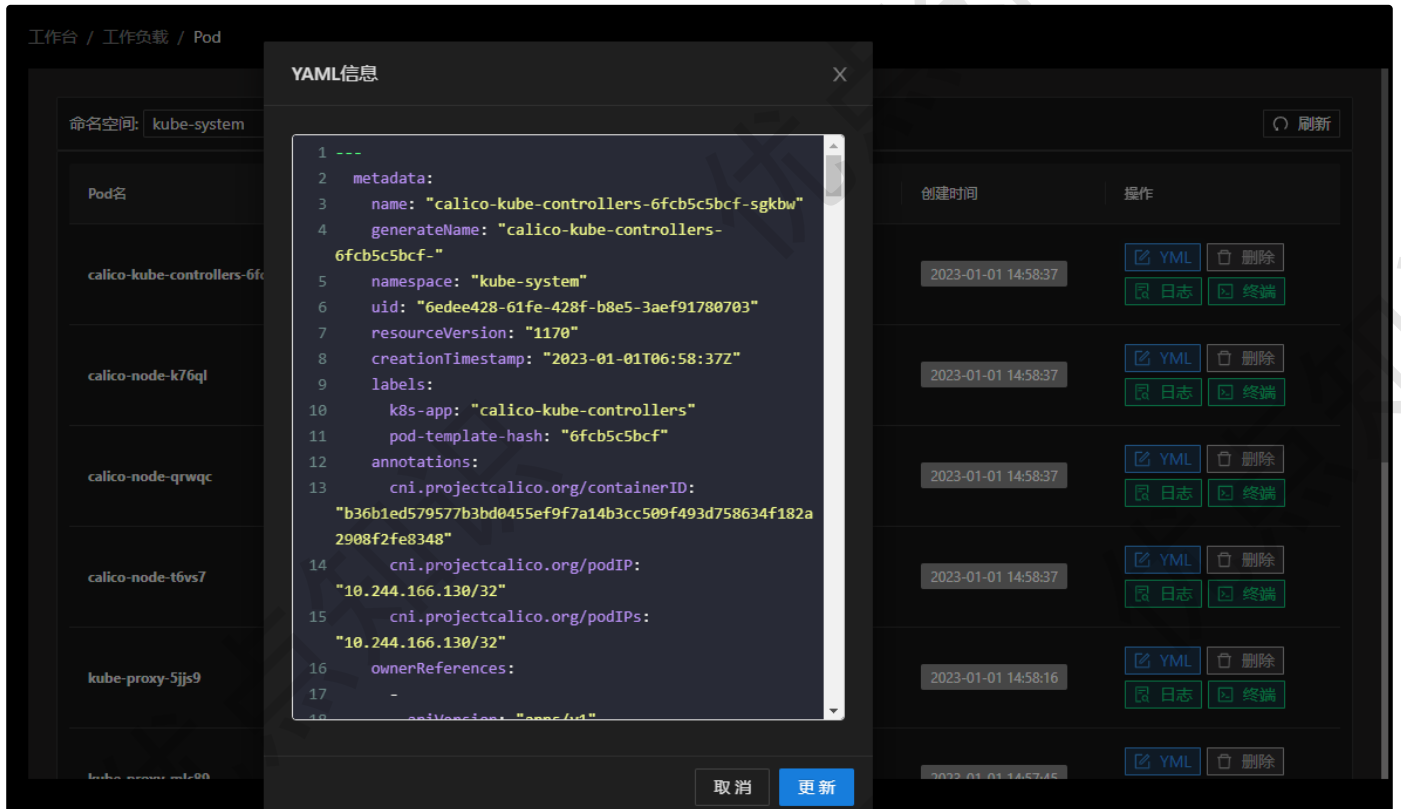
```

<style scoped>
.pod-button {
  margin-right: 5px;
}

```

```
}
.succ-dot{
  display:inline-block;
  width: 7px;
  height:7px;
  background:#86c169;
  border-radius:50%;
  border:1px solid #88c06c;
  margin-right: 10px;
}
.warn-dot{
  display:inline-block;
  width: 7px;
  height:7px;
  background: rgb(233, 200, 16);
  border-radius:50%;
  border:1px solid rgb(233, 200, 16);
  margin-right: 10px;
}
.err-dot{
  display:inline-block;
  width: 7px;
  height:7px;
  background:rgb(199, 85, 85);
  border-radius:50%;
  border:1px solid rgb(207, 94, 94);
  margin-right: 10px;
}
.succ-state {
  color: rgb(27, 202, 21);
}
.warn-state {
  color: rgb(233, 200, 16);
}
.err-state {
  color: rgb(226, 23, 23);
}
</style>
```

(3) 详情



- 添加codemirror配置, `src/config/index.js`

//编辑器配置

```
cmOptions: {
  // 语言及语法模式
  mode: 'text/yaml',
  // 主题
  theme: 'dracula',
  // 显示行数
  lineNumbers: true,
  smartIndent: true, //智能缩进
  indentUnit: 4, // 智能缩进单元长度为 4 个空格
  styleActiveLine: true, // 显示选中行的样式
  matchBrackets: true, //每当光标位于匹配的方括号旁边时, 都会使其高亮显示
  readOnly: false,
  lineWrapping: true //自动换行
},
```

- `src/views/workload/Pod.vue`
- `template`

←!— 展示YAML信息的弹框 —→

```
<a-modal
  v-model:visible="yamlModal"
  title="YAML信息"
  :confirm-loading="appLoading"
  cancelText="取消"
  okText="更新"
```



```

@ok="updatePod">
<!-- codemirror编辑器 -->
<!-- border 带边框 -->
<!-- options 编辑器配置 -->
<!-- change 编辑器中的内容变化时触发 -->
<codemirror
  :value="contentYaml"
  border
  :options="cmOptions"
  height="500"
  style="font-size:14px;"
  @change="onChange"
></codemirror>
</a-modal>

```

- javascript

```

//详情
const contentYaml = ref('')
const yamlModal = ref(false)
const cmOptions = common.cmOptions
const podDetailData = reactive({
  url: common.k8sPodDetail,
  params: {
    pod_name: '',
    namespace: '',
    cluster: ''
  }
})
//json转yaml方法
function transYaml(content) {
  return json2yaml.stringify(content)
}
//详情
function getPodDetail(e) {
  appLoading.value = true
  podDetailData.params.pod_name = e.metadata.name
  podDetailData.params.namespace = namespaceValue.value
  podDetailData.params.cluster = localStorage.getItem('k8s_cluster')
  httpClient.get(podDetailData.url, {params: podDetailData.params})
    .then(res => {
      contentYaml.value = transYaml(res.data)
      yamlModal.value = true
    })
    .catch(res => {
      message.error(res.msg)
    })
    .finally(() => {
      appLoading.value = false
    })
}

```

```

}
//编辑器内容变化时触发的方式,用于将更新的内容复制到变量中
function onChange(val) {
  contentYaml.value = val
}
return {
  contentYaml,
  yamlModal,
  cmOptions,
  getPodDetail,
  onChange,
}

```

(5) 编辑

- javascript

```

//yaml更新
const updatePodData = reactive({
  url: common.k8sPodUpdate,
  params: {
    namespace: '',
    content: '',
    cluster: ''
  }
})
//yaml转对象
function transObj(content) {
  return yaml2obj.load(content)
}
//更新
function updatePod() {
  appLoading.value = true
  //将yaml格式的pod对象转为json
  let content = JSON.stringify(transObj(contentYaml.value))
  updatePodData.params.namespace = namespaceValue.value
  updatePodData.params.content = content
  updatePodData.params.cluster = localStorage.getItem('k8s_cluster')
  httpClient.put(updatePodData.url, updatePodData.params)
    .then(res => {
      message.success(res.msg)
    })
    .catch(res => {
      message.error(res.msg)
    })
    .finally(() => {
      getPodList()
      yamlModal.value = false
    })
}

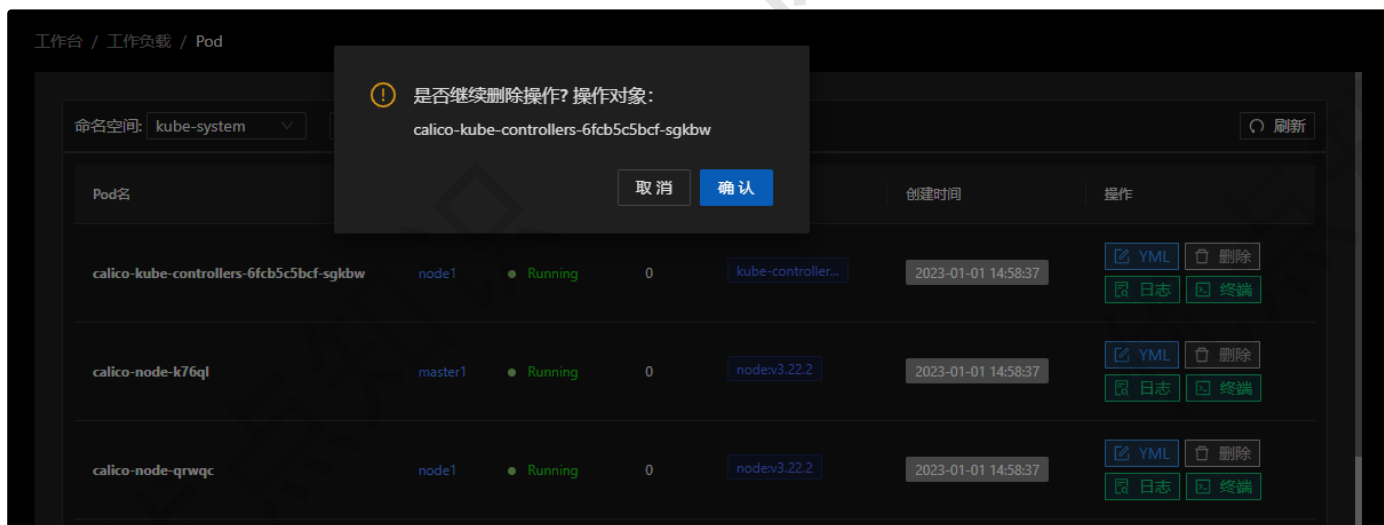
```

```

return {
  updatePod,
}

```

(6) 删除



• javascript

```

// 删除
const delPodData = reactive({
  url: common.k8sPodDel,
  params: {
    pod_name: '',
    namespace: '',
    cluster: ''
  }
})
// 删除
function delPod(name) {
  appLoading.value = true
  delPodData.params.pod_name = name
  delPodData.params.namespace = namespaceValue.value
  delPodData.params.cluster = localStorage.getItem('k8s_cluster')
  httpClient.delete(delPodData.url, {data: delPodData.params})
  .then(res => {
    message.success(res.msg)
  })
  .catch(res => {
    message.error(res.msg)
  })
  .finally(() => {
    getPodList()
  })
}
// 确认框
function showConfirm(action, res, fn) {

```

```

Modal.confirm({
  title: '是否继续' + action + "操作? 操作对象: ",
  icon: createVNode(ExclamationCircleOutlined),
  content: createVNode('div', {
    //style: 'color:red;',
  }, res),
  cancelText: '取消',
  okText: '确认',
  onOk() {
    fn(res)
  },
  // onCancel() {
  //   console.log('Cancel');
  // }
})
}
return {
  delPod,
  showConfirm
}

```

(7) 日志

容器: calico-kube-co... 刷新

```

2023-01-01 07:00:51.992 [INFO][1] main.go 97: Loaded configuration from environment config=&config.Config{LogLevel:"info", WorkloadEndpointWorkers:1, ProfileWorkers:1, PolicyWorkers:1, NodeWorkers:1, Kubeconfig:"",
DatastoreType:"kubernetes"}
W0101 07:00:51.994334 1 client_config.go:615] Neither --kubeconfig nor --master was specified. Using the inClusterConfig. This might not work.
2023-01-01 07:00:51.994 [INFO][1] main.go 118: Ensuring Calico datastore is initialized
2023-01-01 07:00:52.003 [INFO][1] main.go 159: Getting initial config snapshot from datastore
2023-01-01 07:00:52.019 [INFO][1] main.go 162: Got initial config snapshot
2023-01-01 07:00:52.020 [INFO][1] watchersyncer.go 89: Start called
2023-01-01 07:00:52.020 [INFO][1] main.go 179: Starting status report routine
2023-01-01 07:00:52.020 [INFO][1] main.go 188: Starting Prometheus metrics server on port 9094
2023-01-01 07:00:52.020 [INFO][1] main.go 463: Starting informer informer=&cache.sharedIndexInformer{indexer(*cache.cache)(0xc00046a138), controllercache.Controller(nil), processor(*cache.sharedProcessor)(0xc000139a40),
cacheMutationDetector:cache.dummyMutationDetector(), listerWatcher(*cache.ListWatch)(0xc00046a120), objectType(*v1.Pod)(0xc000071c00), resyncCheckPeriod:0, defaultEventHandlerResyncPeriod:0, clock(*clock.RealClock)
(0x2e335b0), started:false, stopped:false, startedLocksync.Mutex{state:0, sema:0x0}, blockDeltaSync.Mutex{state:0, sema:0x0}, watchErrorHandler:(cache.WatchErrorHandler)(nil)}
2023-01-01 07:00:52.020 [INFO][1] main.go 463: Starting informer informer=&cache.sharedIndexInformer{indexer(*cache.cache)(0xc00046a198), controllercache.Controller(nil), processor(*cache.sharedProcessor)(0xc000139ab0),
cacheMutationDetector:cache.dummyMutationDetector(), listerWatcher(*cache.ListWatch)(0xc00046a168), objectType(*v1.Node)(0xc0001c5500), resyncCheckPeriod:0, defaultEventHandlerResyncPeriod:0, clock(*clock.RealClock)
(0x2e335b0), started:false, stopped:false, startedLocksync.Mutex{state:0, sema:0x0}, blockDeltaSync.Mutex{state:0, sema:0x0}, watchErrorHandler:(cache.WatchErrorHandler)(nil)}
2023-01-01 07:00:52.020 [INFO][1] main.go 469: Starting controller ControllerType="Node"
2023-01-01 07:00:52.020 [INFO][1] controller.go 189: Starting Node controller
2023-01-01 07:00:52.020 [INFO][1] watchersyncer.go 127: Sending status update Status=wait-for-ready
2023-01-01 07:00:52.020 [INFO][1] syncer.go 78: Node controller syncer status updated: wait-for-ready
2023-01-01 07:00:52.020 [INFO][1] watchersyncer.go 147: Starting main event processing loop
2023-01-01 07:00:52.020 [INFO][1] watchercache.go 175: Full resync is required ListRoot="/calico/ipam/v2/assignment/"
2023-01-01 07:00:52.021 [INFO][1] watchercache.go 175: Full resync is required ListRoot="/calico/resources/v3/projectcalico.org/nodes"
2023-01-01 07:00:52.022 [INFO][1] resources.go 350: Main client watcher loop
2023-01-01 07:00:52.025 [INFO][1] watchercache.go 273: Sending synced update ListRoot="/calico/ipam/v2/assignment/"
2023-01-01 07:00:52.025 [INFO][1] watchersyncer.go 127: Sending status update Status=resync
2023-01-01 07:00:52.025 [INFO][1] syncer.go 78: Node controller syncer status updated: resync
2023-01-01 07:00:52.025 [INFO][1] watchersyncer.go 209: Received InSync event from one of the watcher caches
2023-01-01 07:00:52.043 [INFO][1] watchercache.go 273: Sending synced update ListRoot="/calico/resources/v3/projectcalico.org/nodes"
2023-01-01 07:00:52.044 [INFO][1] watchersyncer.go 209: Received InSync event from one of the watcher caches

```

- `src/views/workload/Pod.vue`
- `javascript`

```

//router实例
const router = useRouter()
//跳转日志页
function gotoLog(record) {
  let routeUrl = router.resolve({

```

```

        path: "/workload/pod/log",
        query: {
          pod_name: record.metadata.name,
          namespace: record.metadata.namespace,
          cluster: localStorage.getItem('k8s_cluster')
        }
      })
      window.open(routeUrl.href, '_blank')
    }
    return {
      gotoLog
    }
  }

```

- `src/views/workload/PodLog.vue`
- `template`

```

<template>
  <div>
    <div style="height:10px;"></div>
    <a-card :bodyStyle="{padding: '10px', height: '45px'}">
      <span style="font-size:14px;">容器: </span>
      <a-select @change="getContainerLog()" style="width:140px;" size="small"
        v-model:value="containerValue" placeholder="请选择">
        <a-select-option
          v-for="(item, index) in containerList"
          :key="index"
          :value="item">
          {{ item }}
        </a-select-option>
      </a-select>
      <a-button style="margin-left: 20px;" size="small" type="primary" ghost
        @click="getContainerLog()">
        <template #icon><UndoOutlined/></template>
        刷新
      </a-button>
    </a-card>
    <a-card
      class="log-card"
      :loading="appLoading"
      :bodyStyle="{padding: '10px'}">
      {{ logContent }}
    </a-card>
  </div>
</template>

```

- `javascript`

```

<script>
import { onMounted, reactive, ref } from 'vue';
import { useRouter } from 'vue-router'

```

```

import httpClient from '@request';
import common from "@config";
import { message } from 'ant-design-vue';
export default ({
  setup() {
    const appLoading = ref(false)
    const router = useRouter()
    // 容器列表
    const containerValue = ref()
    const containerList = ref([])
    const containerListData = reactive({
      url: common.k8sPodContainer,
      params: {
        pod_name: '',
        namespace: '',
        cluster: ''
      }
    })
    // 容器日志
    const logContent = ref()
    const containerLogData = reactive({
      url: common.k8sPodLog,
      params: {
        container_name: '',
        pod_name: '',
        namespace: '',
        cluster: ''
      }
    })
    // 【方法】
    // 容器列表
    const getContainerList = () => {
      let query = router.currentRoute.value.query
      containerListData.params.pod_name = query.pod_name
      containerListData.params.namespace = query.namespace
      containerListData.params.cluster = query.cluster
      httpClient.get(containerListData.url, {params: containerListData.params})
        .then(res => {
          containerList.value = res.data
          containerValue.value = res.data[0]
          getContainerLog()
        })
        .catch(res => {
          message.error(res.msg)
        })
    }
    // 日志
    const getContainerLog = () => {
      appLoading.value = true
      let query = router.currentRoute.value.query

```

```

        containerLogData.params.container_name = containerValue
        containerLogData.params.pod_name = query.pod_name
        containerLogData.params.namespace = query.namespace
        containerLogData.params.cluster = query.cluster
        httpClient.get(containerLogData.url, {params: containerLogData.params})
        .then(res => {
            logContent.value = res.data
        })
        .catch(res => {
            message.error(res.msg)
        })
        .finally(() => {
            appLoading.value = false
        })
    }
    // 生命周期钩子
    onMounted( () => {
        getContainerList()
    })

    // 返回出去
    return {
        appLoading,
        containerValue,
        containerList,
        logContent,
        getContainerLog
    }
}
})
</script>

```

- CSS

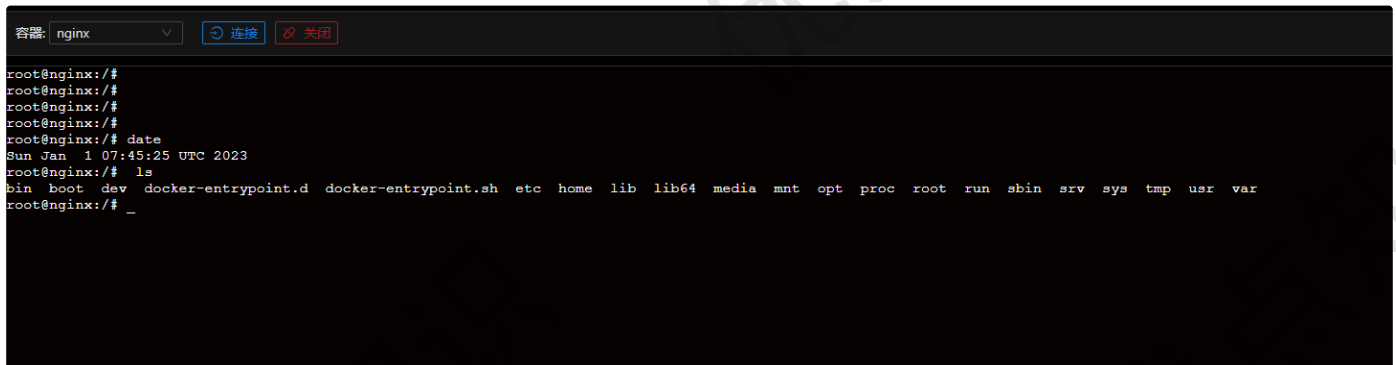
```

<style scoped>
    .log-card {
        margin-top: 10px;
        height: calc(100vh - 67px);
        white-space: pre-wrap;
        overflow-y: auto;
    }
    .log-card::-webkit-scrollbar {
        width: 10px;
    }
    .log-card::-webkit-scrollbar-thumb {
        background-color: #666;
    }
    .log-card::-webkit-scrollbar-track {
        background-color: rgb(164, 162, 162);
    }

```

</style>

(8) 终端



```
容器: nginx 连接 关闭
root@nginx:/#
root@nginx:/#
root@nginx:/#
root@nginx:/#
root@nginx:/# date
Sun Jan 1 07:45:25 UTC 2023
root@nginx:/# ls
bin boot dev docker-entrypoint.d docker-entrypoint.sh etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@nginx:/# _
```

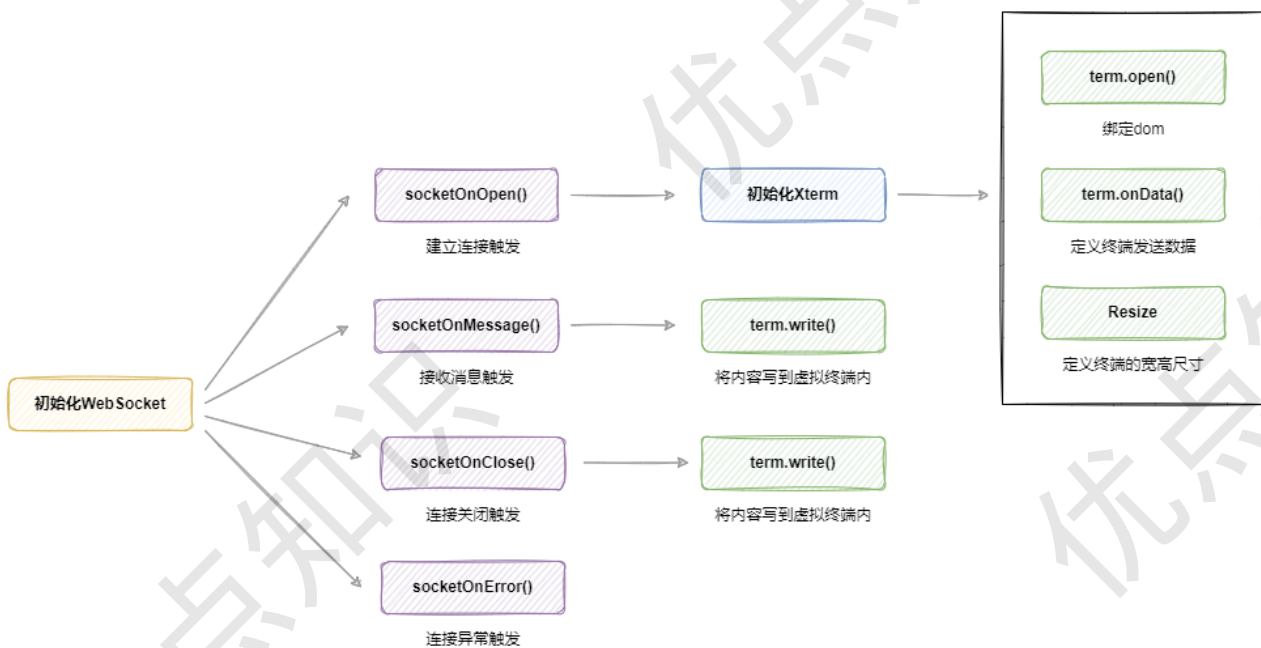
- `src/views/workload/Pod.vue`
- `javascript`

```
// 跳转终端页
function gotoTerminal(record) {
  let routeUrl = router.resolve({
    path: "/workload/pod/terminal",
    query: {
      pod_name: record.metadata.name,
      namespace: record.metadata.namespace,
      cluster: localStorage.getItem('k8s_cluster')
    }
  })
  window.open(routeUrl.href, '_blank')
}
```

- 添加终端配置, `src/config/index.js`

```
// xterm终端配置
termOptions: {
  rendererType: 'canvas', // 渲染类型
  rows: 40, // 行数
  cols: 100,
  convertEol: false, // 启用时, 光标将设置为下一行的开头
  scrollbar: 10, // 终端中的回滚量
  disableStdin: false, // 是否应禁用输入
  cursorStyle: 'underline', // 光标样式
  cursorBlink: true, // 光标闪烁
  theme: {
    foreground: 'white', // 字体
    background: '#060101', // 背景色
    cursor: 'help' // 设置光标
  }
}
```

- 终端实现图解



- `src/views/workload/PodTerminal.vue`
- `template`

```

<template>
  <div>
    <div style="height:10px;"></div>
    <a-card :bodyStyle="{padding: '10px', height: '45px'}">
      <span style="font-size:14px;">容器: </span>
      <a-select style="width:140px;" size="small" v-
model:value="containerValue" placeholder="请选择">
        <a-select-option
          v-for="(item, index) in containerList"
          :key="index"
          :value="item">
          {{ item }}
        </a-select-option>
      </a-select>
      <a-button style="margin-left: 20px;" size="small" type="primary" ghost
@click="initSocket()">
        <template #icon><LoginOutlined /></template>
        连接
      </a-button>
      <a-button style="margin-left: 10px;" size="small" type="danger" ghost
@click="closeSocket()">
        <template #icon><DisconnectOutlined /></template>
        关闭
      </a-button>
    </a-card>
  </a-card>

```

```

        class="term-card"
        :loading="appLoading"
        :bodyStyle="{padding: '0px', height: '100%'}">
        <div class="xterm-div" id="xterm"></div>
    </a-card>
</div>
</template>

```

- javascript

```

<script>
import { onBeforeUnmount, onMounted, reactive, ref } from 'vue';
import { useRouter } from 'vue-router';
import httpClient from '@/request';
import common from "@/config";
import { message } from 'ant-design-vue';
import { Terminal } from 'xterm';
import { FitAddon } from 'xterm-addon-fit';
import 'xterm/css/xterm.css';
import 'xterm/lib/xterm.js';
export default ({
  setup() {
    const appLoading = ref(false)
    const router = useRouter()
    // 容器列表
    const containerValue = ref()
    const containerList = ref([])
    const containerListData = reactive({
      url: common.k8sPodContainer,
      params: {
        pod_name: '',
        namespace: '',
        cluster: ''
      }
    })
    // 终端
    const termOptions = common.termOptions
    var term = null
    var socket = null
    // 【方法】
    // 容器列表
    const getContainerList = () => {
      let query = router.currentRoute.value.query
      containerListData.params.pod_name = query.pod_name
      containerListData.params.namespace = query.namespace
      containerListData.params.cluster = query.cluster
      httpClient.get(containerListData.url, {params: containerListData.params})
        .then(res => {
          containerList.value = res.data
          containerValue.value = res.data[0]

```

```

        //getContainerLog()
    })
    .catch(res => {
        message.error(res.msg)
    })
}
//初始化虚拟终端
const initTerm = () => {
    term = new Terminal(termOptions)
    term.open(document.getElementById('xterm'))
    const fitAddon = new FitAddon()
    term.loadAddon(fitAddon)
    fitAddon.fit()
    term.focus();
    term.onData(function (key) {
        let msgOrder = {
            operation: 'stdin',
            data: key,
        }
        socket.send(JSON.stringify(msgOrder))
    })
    let msgOrder2 = {
        operation: 'resize',
        cols: term.cols,
        rows: term.rows,
    };
    socket.send(JSON.stringify(msgOrder2))
}
//初始化websocket
const initSocket = () => {
    if (term) {
        term.dispose()
    }
    if (socket) {
        socket.close()
    }
    let query = router.currentRoute.value.query
    let terminalWsUrl = common.k8sTerminalWs + "?pod_name=" + query.pod_name
    + "&container_name=" + containerValue.value + "&namespace=" + query.namespace +
    "&cluster=" + query.cluster
    socket = new WebSocket(terminalWsUrl)
    socket.onopen = () => {
        // 链接成功后
        initTerm()
    }
    socket.onmessage = (msg) => {
        let content = JSON.parse(msg.data)
        term.write(content.data)
    }
    socket.onsend = (msgOrder) => {

```

```

        let order = []
        order.push(msgOrder)
        socket.send(JSON.stringify(order));
    }
    socket.onclose = () => {
        console.log('close socket')
    }
    socket.onerror = () => {
        console.log('socket 连接失败')
    }
}
const closeSocket = () => {
    if (!socket) {
        return
    }
    term.write("连接关闭")
    socket.close()
}
//生命周期钩子
onMounted( () => {
    getContainerList()
})
onBeforeUnmount( () => {
    if (socket) {
        socket.close()
    }
})
//返回出去
return {
    appLoading,
    containerValue,
    containerList,
    initSocket,
    closeSocket
}
}
})
</script>

```

- css

```

<style scoped>
.term-card {
    margin-top: 10px;
    height: calc(100vh - 67px);
}
.term-card::-webkit-scrollbar {
    width: 10px;
}
.term-card::-webkit-scrollbar-thumb {

```

```

        background-color: #666;
    }
    .term-card::-webkit-scrollbar-track {
        background-color: rgb(164, 162, 162);
    }
    .xterm-div {
        height: 100%;
        background-color: black;
    }
    /deep/.xterm-viewport::-webkit-scrollbar {
        width: 10px;
    }
    /deep/.xterm-viewport::-webkit-scrollbar-thumb {
        background-color: #666;
    }
    /deep/.xterm-viewport::-webkit-scrollbar-track {
        background-color: rgb(164, 162, 162);
    }
}
</style>

```

2. Deployment

The screenshot shows the KubeA web interface. The top navigation bar includes the KubeA logo and tabs for TST-1 and TST-2. The sidebar on the left contains a menu with options: 概览 (Overview), 集群 (Clusters), 工作负载 (Workloads), Pod, Deployment (selected), DaemonSet, StatefulSet, 负载均衡 (Load Balancers), 存储配置 (Storage), and Helm应用 (Helm Applications). The main content area is titled '工作台 / 工作负载 / Deployment' (Dashboard / Workloads / Deployment). It features a search bar with '命名空间: kube-system' (Namespace: kube-system) and a '请输入' (Please enter) field. Below the search bar is a table of Deployments:

Deployment名	标签	容器组	镜像	创建时间	操作
calico-kube-controllers	k8s-app:calico-...	1 / 1	kube-controller...	2023-01-01 14:58:36	[YML] [删除] [扩容] [重启]
coredns	k8s-app:kube-dn...	2 / 2	coredns:1.7.0	2023-01-01 14:56:30	[YML] [删除] [扩容] [重启]

At the bottom of the table, it shows '共 2 条' (Total 2 items) and pagination controls for page 1 of 10.

- 功能: [创建](#) [列表](#) [详情](#) [编辑](#) [删除](#) [扩容](#) [重启](#)
- 数据表格(用于列表展示)

字段	说明
name	Deployment名
labels	标签
containers	容器组
image	镜像
creationTimestamp	创建时间
action	操作: YML、删除、重启、扩容

- `src/views/workload/Deployment.vue`

(1) 头部工具栏

- `template`

```
<template>
  <div>
    <MainHead
      searchDescribe="请输入"
      @searchChange="getSearchValue"
      namespace
      @namespaceChange="getNamespaceValue"
      @namespaceList="getNamespaceList"
      @dataList="getDeploymentList"
      add
      @addFunc="handleAdd" />
    </div>
  </template>
```

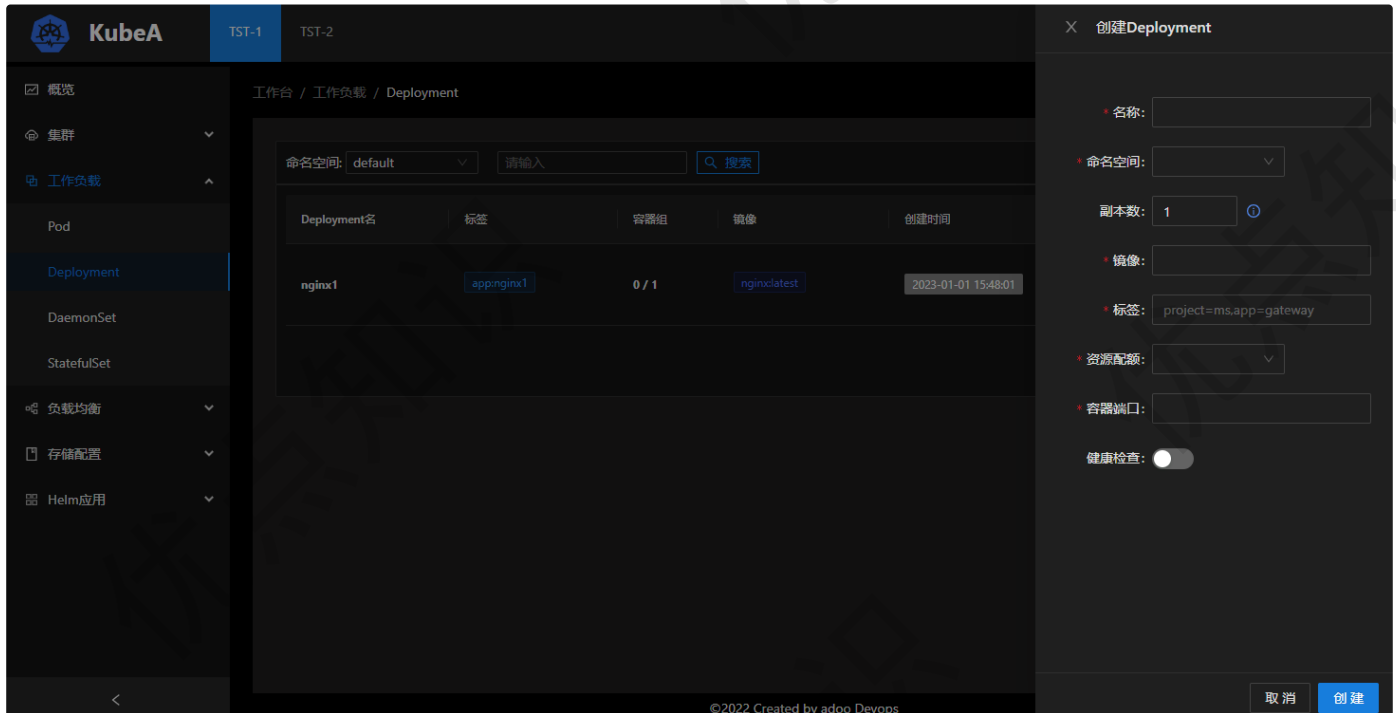
- `javascript`

```
// 常用项
const searchValue = ref('')
const namespaceValue = ref('')
const namespaceList = ref([])
function getSearchValue(val) {
  searchValue.value = val
}
function getNamespaceValue(val) {
  namespaceValue.value = val
}
function getNamespaceList(val) {
  namespaceList.value = val
}
return {
  namespaceList,
  getSearchValue,
  getNamespaceValue,
```

```
getNamespaceList
```

```
}
```

(2) 创建



- template

←!— 创建deployment的抽屉弹框 —→

```
<a-drawer
  v-model:visible="createDrawer"
  title="创建Deployment"
  :footer-style="{ textAlign: 'right' }"
  @close="onClose">
  <br>
  <a-form ref="formRef" :model="createDeployment" :labelCol="{style:
{width: '30%'}}">
    <a-form-item
      label="名称"
      name="createName"
      :rules="[{ required: true, message: '请输入Deployment名称' }]">
      <a-input v-model:value="createName" />
    </a-form-item>
    <a-form-item
      label="命名空间"
      name="createNamespace"
      :rules="[{ required: true, message: '请选择命名空间' }]">
      <a-select show-search style="width:140px;" v-
model:value="createNamespace" placeholder="请选择">
        <a-select-option
          v-for="(item, index) in namespaceList"
          :key="index"
```

```

        :value="item.metadata.name">
        {{ item.metadata.name }}
      </a-select-option>
    </a-select>
  </a-form-item>
  <a-form-item
    label="副本数"
    name="createReplicas">
    <a-input-number v-model:value="createReplicas" :min="1"
:~max="30"></a-input-number>
    <a-popover>
      <template #content>
        <span>Deployment副本数最小1, 最大30</span>
      </template>
      <info-circle-outlined style="margin-left:10px;color:rgb(84,
138, 238);" />
    </a-popover>
  </a-form-item>
  <a-form-item
    label="镜像"
    name="createImage"
    :rules="[{ required: true, message: '请输入镜像名' }]">
    <a-input v-model:value="createImage" />
  </a-form-item>
  <a-form-item
    label="标签"
    name="createLabelStr"
    :rules="[{ required: true, message: '请输入标签' }]">
    <a-input v-model:value="createLabelStr"
placeholder="project=ms,app=gateway" />
  </a-form-item>
  <a-form-item
    label="资源配额"
    name="createResource"
    :rules="[{ required: true, message: '请选择资源规格' }]">
    <a-select show-search style="width:140px;" v-
model:value="createResource" placeholder="请选择">
      <a-select-option value="0.5/1">0.5核/1G</a-select-option>
      <a-select-option value="1/2">1核/2G</a-select-option>
      <a-select-option value="2/4">2核/4G</a-select-option>
      <a-select-option value="4/8">4核/8G</a-select-option>
      <a-select-option value="8/16">8核/16G</a-select-option>
    </a-select>
  </a-form-item>
  <a-form-item
    label="容器端口"
    name="createContainerPort"
    :rules="[{ required: true, message: '请输入端口号' }]">
    <a-input v-model:value="createContainerPort" />
  </a-form-item>

```



```

    <a-form-item
      label="健康检查"
      name="createHealthCheck">
      <a-switch v-model:checked="createHealthCheck" />
    </a-form-item>
    <a-form-item
      v-if="createHealthCheck"
      label="检测路径"
      name="createHealthPath"
      :rules="[ { required: true, message: '请输入健康检测路径' } ]">
      <a-input v-model:value="createHealthPath" />
    </a-form-item>
  </a-form>
</template #footer>
  <a-button style="margin-right: 8px" @click="onClose()">取消</a-button>
  <a-button type="primary" @click="formSubmit()">创建</a-button>
</template>
</a-drawer>

```

- javascript

```

// 创建
const formRef = ref()
const createDrawer = ref(false)
const createDeployment = reactive({
  createName: '',
  createNamespace: '',
  createReplicas: 1,
  createImage: '',
  createResource: '',
  createHealthCheck: false,
  createHealthPath: '',
  createLabelStr: '',
  createContainerPort: ''
})
const createDeploymentData = reactive({
  url: common.k8sDeploymentCreate,
  params: {
    name: '',
    namespace: '',
    replicas: 1,
    image: '',
    cpu: '',
    memory: '',
    health_check: false,
    health_path: '',
    label: {},
    container_port: '',
    cluster: ''
  }
})

```

```

})
//处理新增
function handleAdd() {
  createDrawer.value = true
}
//验证表单
async function formSubmit() {
  try {
    await formRef.value.validateFields();
    //console.log('Success:', values);
    createDeploymentFunc()
  } catch (errorInfo) {
    console.log('Failed:', errorInfo);
  }
}
function resetForm() {
  formRef.value.resetFields();
}
//创建deployment
function createDeploymentFunc() {
  //正则匹配, 验证label的合法性
  let reg = new RegExp("(^[A-Za-z]+[A-Za-z0-9]+).*$")
  if (!reg.test(createDeployment.createLabelStr)) {
    message.warning("标签填写异常, 请确认后重新填写")
    return
  }
  //加载loading动画
  appLoading.value = true
  //定义label、cpu和memory变量
  let label = new Map()
  let cpu, memory
  //将label字符串转成数组
  let a = (createDeployment.createLabelStr).split(",")
  //将数组转成map
  a.forEach(item => {
    let b = item.split("=")
    label[b[0]] = b[1]
  })
  //将deployment的规格转成cpu和memory
  let resourceList = createDeployment.createResource.split("/")
  cpu = resourceList[0]
  memory = resourceList[1] + "Gi"
  //赋值
  createDeploymentData.params.label = label
  createDeploymentData.params.cpu = cpu
  createDeploymentData.params.memory = memory
  createDeploymentData.params.name = createDeployment.createName
  createDeploymentData.params.namespace = createDeployment.createNamespace
  createDeploymentData.params.replicas = createDeployment.createReplicas
  createDeploymentData.params.image = createDeployment.createImage

```

```

        createDeploymentData.params.health_check =
createDeployment.createHealthCheck
        createDeploymentData.params.health_path =
createDeployment.createHealthPath
        createDeploymentData.params.container_port =
parseInt(createDeployment.createContainerPort)
        createDeploymentData.params.cluster = localStorage.getItem('k8s_cluster')
        httpClient.post(createDeploymentData.url, createDeploymentData.params)
        .then(res => {
            message.success(res.msg)
        })
        .catch(res => {
            message.error(res.msg)
        })
        .finally(() => {
            //重置表单
            resetForm()
            getDeploymentList()
            //关闭抽屉
            createDrawer.value = false
        })
    }
    //关闭抽屉
    function onClose () {
        Modal.confirm({
            title: "是否确认关闭操作?",
            icon: createVNode(ExclamationCircleOutlined),
            content: createVNode('div', {
                //style: 'color:red;',
            }),
            cancelText: '取消',
            okText: '确认',
            onOk() {
                createDrawer.value = false
                resetForm()
            },
            onCancel() {
                createDrawer.value = true
            }
        })
    }
    return {
        createDrawer,
        createDeployment,
        formRef,
        ...toRefs(createDeployment),
        handleAdd,
        onClose,
        formSubmit
    }
}

```

(2) 重启



• javascript

```
// 重启
const restartDeploymentData = reactive({
  url: common.k8sDeploymentRestart,
  params: {
    deployment_name: '',
    namespace: '',
    cluster: ''
  }
})

// 重启deployment
function restartDeployment(name) {
  appLoading.value = true
  restartDeploymentData.params.deployment_name = name
  restartDeploymentData.params.namespace = namespaceValue.value
  restartDeploymentData.params.cluster =
    localStorage.getItem('k8s_cluster')
  httpClient.put(restartDeploymentData.url, restartDeploymentData.params)
    .then(res => {
      message.success(res.msg)
    })
    .catch(res => {
      message.error(res.msg)
    })
    .finally(() => {
      getDeploymentList()
    })
}

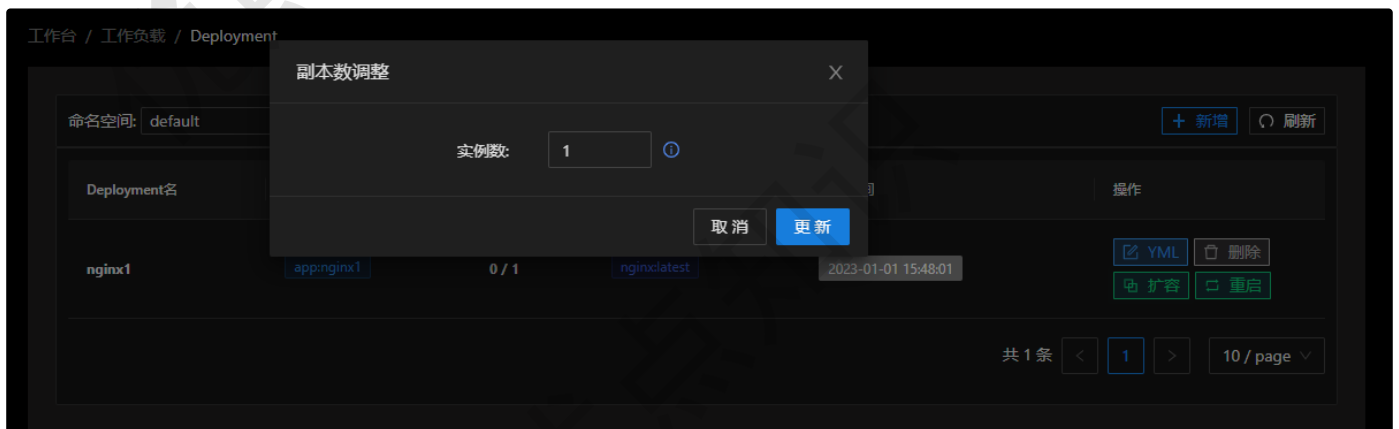
// 确认框
function showConfirm(action, res, fn) {
  Modal.confirm({
    title: '是否继续' + action + "操作? 操作对象: ",
    icon: createVNode(ExclamationCircleOutlined),
    content: createVNode('div', {
      // style: 'color:red;',
    })
  })
}
```

```

    }, res),
    cancelText: '取消',
    okText: '确认',
    onOk() {
      fn(res)
    },
    // onCancel() {
    //   console.log('Cancel');
    // }
  })
}
return {
  restartDeployment,
  showConfirm
}

```

(3) 扩容



- template

```

<!-- 调整副本数的弹框 -->
<a-modal
  v-model:visible="scaleModal"
  title="副本数调整"
  :confirm-loading="appLoading"
  cancelText="取消"
  okText="更新"
  @ok="scaleDeployment">
  <div style="text-align:center">
    <span style="margin-right:30px;">实例数: </span>
    <a-input-number v-model:value="scaleNum" :min="0" :max="30" label="描述文字"></a-input-number>
    <a-popover>
      <template #content>
        <span>Deployment实例数最小0, 最大30</span>
      </template>
      <info-circle-outlined style="margin-left:10px;color:rgb(84, 138, 238);" />
    </a-popover>
  </div>
</a-modal>

```

```

        </a-popover>
      </div>
    </a-modal>

```

- javascript

```

//扩缩容
const scaleNum = ref(0)
const scaleModal = ref(false)
const scaleDeploymentData = reactive({
  url: common.k8sDeploymentScale,
  params: {
    deployment_name: '',
    namespace: '',
    scale_num: '',
    cluster: ''
  }
})
//扩缩容的中间方法，用于赋值及打开弹出框
function handleScale(e) {
  scaleModal.value = true
  deploymentDetail.metadata = e.metadata
  scaleNum.value = e.spec.replicas
}
//扩缩容
function scaleDeployment() {
  appLoading.value = true
  scaleDeploymentData.params.deployment_name =
deploymentDetail.metadata.name
  scaleDeploymentData.params.namespace = namespaceValue.value
  scaleDeploymentData.params.scale_num = scaleNum.value
  scaleDeploymentData.params.cluster = localStorage.getItem('k8s_cluster')
  httpClient.put(scaleDeploymentData.url, scaleDeploymentData.params)
    .then(res => {
      message.success(res.msg)
    })
    .catch(res => {
      message.error(res.msg)
    })
    .finally(() => {
      getDeploymentList()
      scaleModal.value = false
    })
  }
  return {
    scaleModal,
    scaleNum,
    handleScale,
    scaleDeployment,
  }
}

```

3. DaemonSet

- 功能: 列表 详情 编辑 删除
- 数据表格(用于列表展示)

字段	说明
name	DaemonSet名
labels	标签
containers	容器组
image	镜像
creationTimestamp	创建时间
action	操作: YML、删除

- `src/views/workload/DaemonSet.vue`

4. StatefulSet

- 功能: 列表 详情 编辑 删除
- 数据表格(用于列表展示)

字段	说明
name	StatefulSet名
labels	标签
containers	容器组
image	镜像
creationTimestamp	创建时间
action	操作: YML、删除

- `src/views/workload/StatefulSet.vue`

3.6 集群

1. Node

- 功能: 列表 详情
- 数据表格(用于列表展示)

字段	说明
name	Node名
standard	规格
podCidr	Pod网段
version	k8s版本
creationTimestamp	创建时间
action	操作: YML

- [src/views/cluster/Node.vue](#)

2. Namespace

- 功能: 列表 详情 删除
- 数据表格(用于列表展示)

字段	说明
name	Namespace名
labels	标签
status	状态
creationTimestamp	创建时间
action	操作: YML、删除

- [src/views/cluster/Namespace.vue](#)

3. PV

- 功能: 列表 详情 删除
- 数据表格(用于列表展示)

字段	说明
name	Pv名
labels	标签
status	状态
storage	容量
accessMode	访问模式
pvc	PVC
creationTimestamp	创建时间
action	操作: YML、删除

- [src/views/cluster/PV.vue](#)

3.7 负载均衡

1. Ingress

- 功能：创建 列表 详情 编辑 删除
- 数据表格(用于列表展示)

字段	说明
name	Ingress名
labels	标签
host	绑定域名
path	Path
external-ip	外部IP
tls	TLS
creationTimestamp	创建时间
action	操作：YML、删除

- [src/views/loadbalance/Ingress.vue](#)

2. Service

- 功能：创建 列表 详情 编辑 删除
- 数据表格(用于列表展示)

字段	说明
name	Svc名
labels	标签
type	类型
cluster-ip	CLUSTER-IP
external-ip	外部IP
port	端口
creationTimestamp	创建时间
action	操作：YML、删除

- [src/views/loadbalance/Service.vue](#)

3.8 存储配置

1. ConfigMap

- 功能：列表 详情 编辑 删除
- 数据表格（用于列表展示）

字段	说明
name	Configmap名
labels	标签
data	数据
creationTimestamp	创建时间
action	操作：YML、删除

- `src/views/storage/Configmap.vue`

2. Secret

- 功能：列表 详情 编辑 删除
- 数据表格（用于列表展示）

字段	说明
name	Secret名
labels	标签
data	数据
type	类型
creationTimestamp	创建时间
action	操作：YML、删除

- `src/views/storage/Secret.vue`

3. PVC

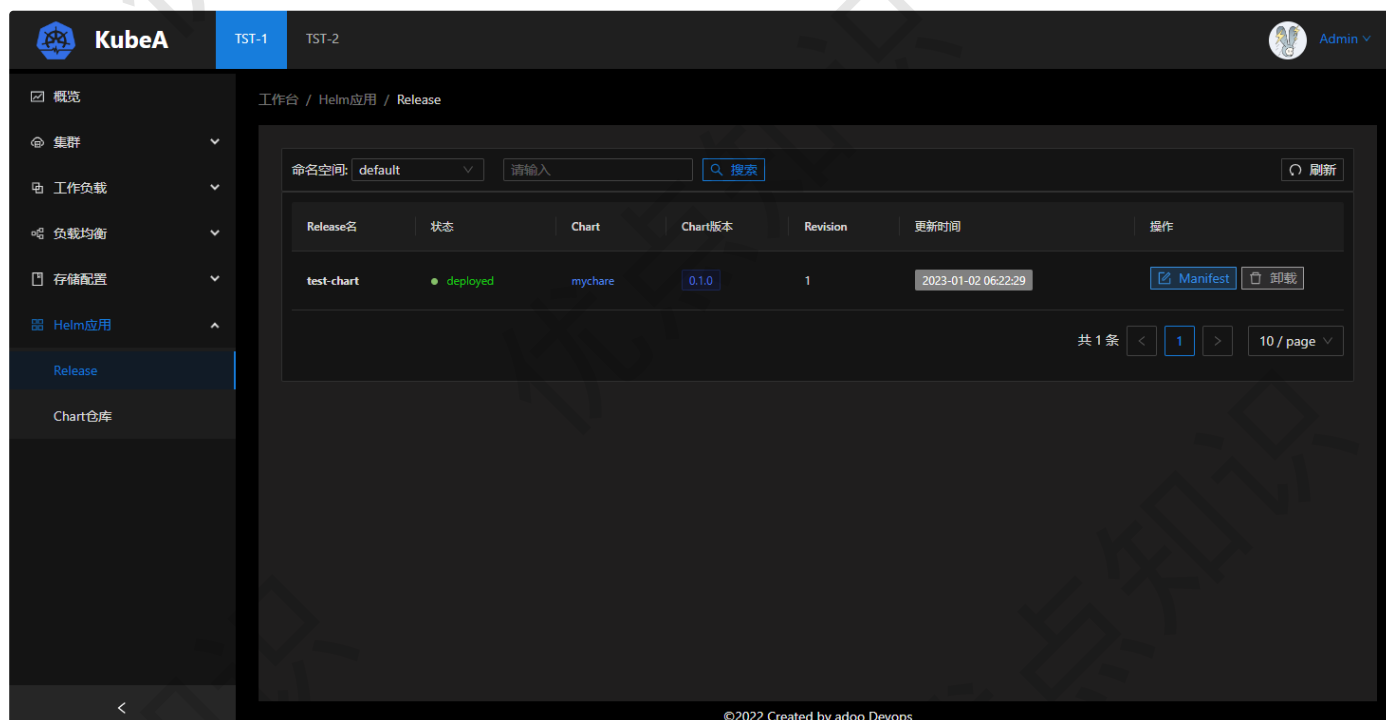
- 功能：列表 详情 编辑 删除
- 数据表格（用于列表展示）

字段	说明
name	Pvc名
labels	标签
status	状态
storage	容量
accessMode	访问模式
storageclass	StorageClass
creationTimestamp	创建时间
action	操作: YML、删除

- `src/views/storage/PVC.vue`

3.9 Helm应用

1. Release



- 功能: 列表 Manifest 卸载
- 数据表格(用于列表展示)

字段	说明
name	Release名
status	状态
chart	所属Chart
chart_version	Chart版本
revision	Revision
creationTimestamp	创建时间
action	操作：Manifest、卸载

- `src/views/helmstore/Release.vue`

2. Chart仓库

- 功能：创建 列表 安装 编辑 删除
- `src/views/helmstore/ChartRepo.vue`

(1) 头部工具栏



- `template`

```
<template>
  <div>
    <!-- 头部工具栏 -->
    <div style="text-align:right;margin-bottom:15px;">
      <a-space>
        <a-input-search
          allow-clear
          v-model:value="searchValue"
          placeholder="请输入"
          style="width: 200px"
          @search="getChartList"
        />
        <a-button size="small" type="primary" ghost @click="handleAdd()">
          <template #icon><PlusOutlined /></template>
          新增
        </a-button>
        <a-button size="small" @click="getChartList()">
          <template #icon><UndoOutlined /></template>
          刷新
        </a-button>
      </a-space>
    </div>
  </div>
</template>
```

```
</template>
```

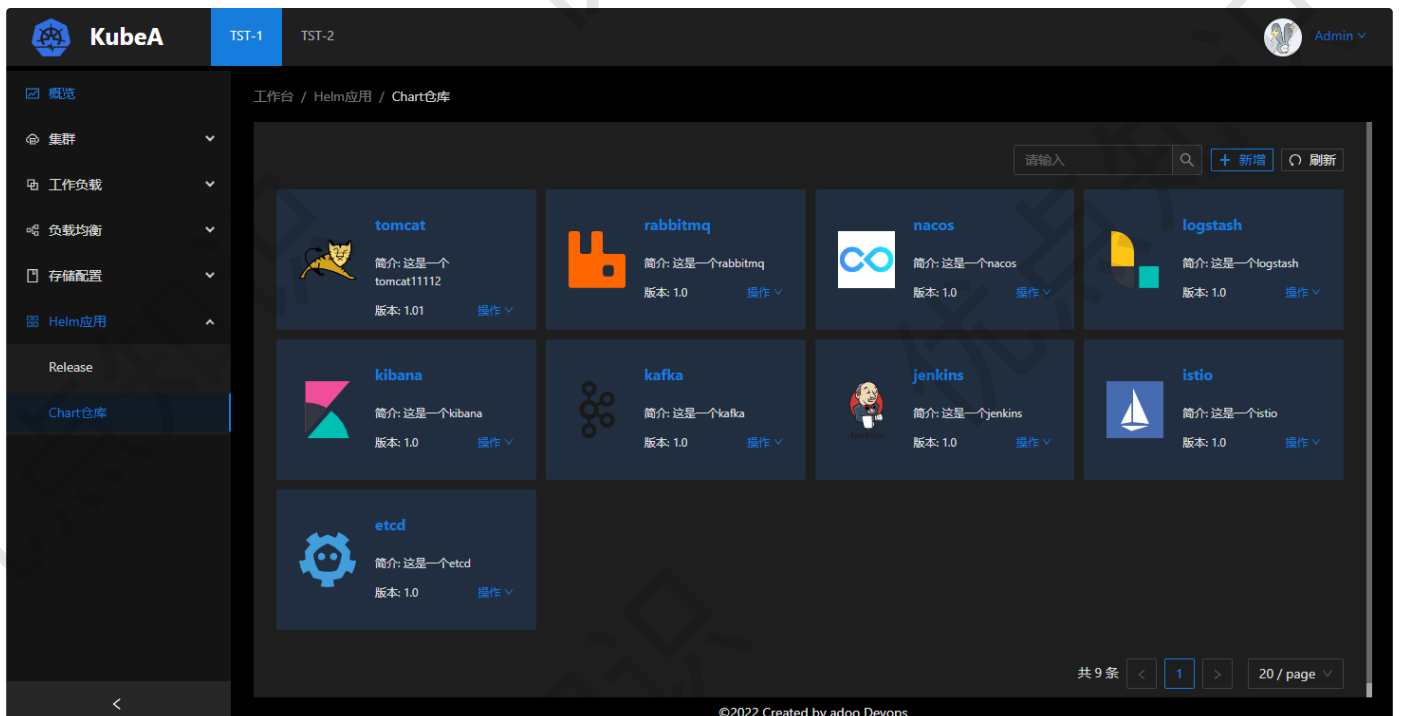
- javascript

```
<script>
import { ref, toRefs, reactive, onMounted, createVNode } from 'vue'
import httpClient from '@/request';
import common from "@/config";
import { message } from 'ant-design-vue';
import { ExclamationCircleOutlined } from '@ant-design/icons-vue';
import { Modal } from 'ant-design-vue';
export default({
  setup() {
    const searchValue = ref('')
    return {
      searchValue,
    },
  },
})
</script>
```

- CSS

```
<style scoped>
  .ant-btn {
    border-radius: 1px;
  }
</style>
```

(2) 卡片列表



- template

```
<!-- Chart卡片 -->
<a-spin :spinning="appLoading">
  <a-row :gutter="10">
    <template v-for="item, index in chartList" :key="index">
      <a-col :span="6" style="margin-bottom:10px;">
        <a-card :bordered="false" class="chart-card">
          <div style="float:left;margin-right:20px;padding: 20px
0;">
            
          </div>
          <div>
            <a style="font-size:16px;font-weight:bold;">{{
item.name }}</a><br><br>
            <template v-if="item.describe.length ≥ 20">
              <a-popover>
                <template #content>
                  <div style="width:200px;">{{
item.describe }}</div>
                  </template>
                  <p class="chart-describe">简介: {{
item.describe }}</p>
                </a-popover>
              </template>
              <p v-else>简介: {{ item.describe }}</p>
            </div>
            <div>
              <span>版本: {{ item.version }}</span>
              <a-dropdown :overlayStyle="{paddingTop: '10px'}">
                <a style="float:right;" @click.prevent>
                  操作
                  <DownOutlined />
                </a>
                <template #overlay>
                  <a-menu>
                    <a-menu-item>
                      <a @click="handleInstall(item)">安装</a>
                    </a-menu-item>
                    <a-menu-item>
                      <a @click="handleUpdate(item)">编辑</a>
                    </a-menu-item>
                    <a-menu-item>
                      <a @click="showConfirm('删除', item,
delChartFunc)">删除</a>
                    </a-menu-item>
                  </a-menu>
                </template>
              </a-dropdown>
            </div>
          </div>
        </a-card>
      </template>
    </a-row>
  </a-spin>
</div>
```

```

        </div>
      </a-card>
    </a-col>
  </template>
</a-row>
</a-spin>
<a-pagination
  style="text-align:right;margin-top:20px;"
  v-model:current="pagination.currentPage"
  v-model:page-size="pagination.pageSize"
  :page-size-options="pagination.pageSizeOptions"
  :total="pagination.total"
  :show-total="pagination.showTotal"
  show-quick-jumper
  show-less-items
  show-size-changer
  @change="handleTableChange"
/>

```

- javascript

```

const appLoading = ref(false)
const pagination = reactive({
  total: 0,
  currentPage: 1,
  pageSize: 12,
  pageSizeOptions: ["12", "24", "36", "48"],
  showTotal: total => `共 ${total} 条`
})
//列表
const chartList = ref([])
const chartListData = reactive({
  url : common.helmChartList,
  params: {
    name: '',
    page: 1,
    limit: 12
  }
})
function handleTableChange(current, pageSize) {
  pagination.currentPage = current
  pagination.pageSize = pageSize
  getChartList()
}
//列表
function getChartList() {
  appLoading.value = true
  if (searchValue.value) {pagination.currentPage = 1}
  chartListData.params.name = searchValue.value
  chartListData.params.page = pagination.currentPage
}

```

```

chartListData.params.limit = pagination.pagesize
httpClient.get(chartListData.url, {params: chartListData.params})
.then(res => {
  //响应成功, 获取chart列表和total
  chartList.value = res.data.items
  pagination.total = res.data.total
})
.catch(res => {
  message.error(res.msg)
})
.finally(() => {
  appLoading.value = false
})
}
onMounted(() => {
  getChartList()
})
return {
  appLoading,
  chartList,
  pagination,
  getChartList,
}

```

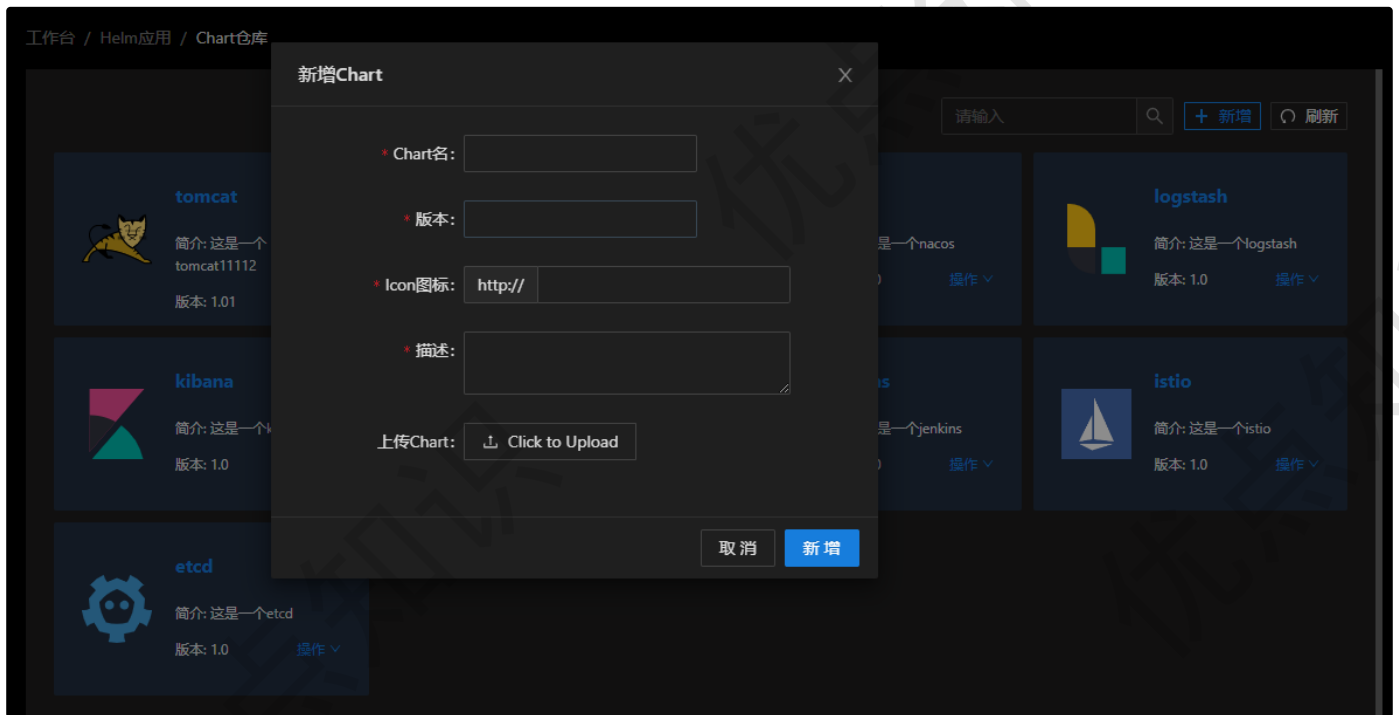
- CSS

```

.chart-card {
  background-color: rgb(33, 46, 64);
  font-size: 12px;
  height: 148px;
}
.chart-describe {
  text-overflow: ellipsis;
  overflow: hidden;
  white-space: nowrap;
}

```

(3) 创建



- template

```

<!-- 新增Chart -->
<a-modal
  v-model:visible="addModal"
  title="新增Chart"
  :confirm-loading="appLoading"
  cancelText="取消"
  okText="新增"
  :afterClose="handleClose"
  @ok="addSubmit">
  <a-form ref="addRef" :model="addChart" :labelCol="{style: {width:
'30%'}}">
    <a-form-item
      label="Chart名"
      name="name"
      :rules="[{ required: true, message: '请输入Chart名' }]">
      <a-input v-model:value="addChart.name" style="width: 200px"/>
    </a-form-item>
    <a-form-item
      label="版本"
      name="version"
      :rules="[{ required: true, message: '请输入版本' }]">
      <a-input v-model:value="addChart.version" style="width: 200px"/>
    </a-form-item>
    <a-form-item
      label="Icon图标"
      name="icon_url"
      :rules="[{ required: true, message: '请输入图标地址' }]">
      <a-input v-model:value="addChart.icon_url" addon-before="http://"
style="width: 280px"/>
  </a-form>
</a-modal>

```

```

    </a-form-item>
    <a-form-item
      label="描述"
      name="describe"
      :rules="[{ required: true, message: '请填写Chart描述' }]">
      <a-textarea v-model:value="addChart.describe" style="width:
280px" />
    </a-form-item>
    <a-form-item
      label="上传Chart"
      name="chartFile">
      <!-- 上传组件
      max-count 最大上传数为2
      remove 删除文件 -->
      <a-upload
        v-model:file-list="fileList"
        :max-count="2"
        name="chart"
        :customRequest="handleUpload"
        @remove="handleFileRemove">
        <a-button>
          <upload-outlined />
          Click to Upload
        </a-button>
      </a-upload>
    </a-form-item>
  </a-form>
</a-modal>

```

- javascript

```

// 新增
const addModal = ref(false)
const addRef = ref()
const addChart = reactive({
  name: '',
  icon_url: '',
  version: '',
  describe: '',
  file_name: ''
})
const addChartData = reactive({
  url: common.helmChartAdd,
  params: {
    name: '',
    file_name: '',
    icon_url: '',
    version: '',
    describe: ''
  }
})

```

```

    })
    // 上传文件
    const fileList = ref([])
    const fileUploadUrl = common.helmChartFileUpload
    // 删除文件
    const fileDelData = reactive({
      url: common.helmChartFileDel,
      params: {
        chart: ''
      }
    })
  })
  function handleAdd() {
    addModal.value = true
  }
  // 新增
  function addChartFunc() {
    // 检查是否有文件上传
    if (!fileList.value.length) {
      message.warning("未上传Chart文件,请上传")
      return
    }
    // 添加
    appLoading.value = true
    addChartData.params = addChart
    httpClient.post(addChartData.url, addChartData.params)
      .then(res => {
        message.success(res.msg)
      })
      .catch(res => {
        message.error(res.msg)
      })
      .finally(() => {
        getChartList()
        // 重置表单
        addRef.value.resetFields()
        // 关闭抽屉
        addModal.value = false
        // 删除文件
        fileList.value.pop()
      })
  }
  async function addSubmit() {
    try {
      await addRef.value.validateFields();
      addChartFunc()
    } catch (errorInfo) {
      console.log('Failed:', errorInfo);
    }
  }
}

// 上传文件

```

```

function handleUpload(data) {
  if(fileList.value.length > 1) {
    fileList.value.pop()
    message.warning("最大上传文件数为1,请删除后重新上传")
    return
  }
  let formData = new FormData()
  formData.append('chart', data.file)
  httpClient.post(fileUploadUrl, formData)
  .then(res => {
    message.success(res.msg)
    if (addModal.value) {
      addChart.file_name = data.file.name
    }
    if (updateModal.value) {
      updateChart.file_name = data.file.name
    }
    data.onSuccess()
  })
  .catch(res => {
    message.error(res.msg)
    data.onError()
    //用于重复上传文件的清空前端展示
    fileList.value.pop()
  })
}
//删除文件
function fileDelFunc(fileName) {
  fileDelData.params.chart = fileName
  httpClient.delete(fileDelData.url, {data: fileDelData.params})
  .then(res => {
    message.success(res.msg)
  })
  .catch(res => {
    message.error(res.msg)
  })
}
//用于所有关闭modal的动作,删除已上传的文件
function handleClose() {
  //删除文件
  if (fileList.value.length) {
    fileDelFunc(fileList.value[0].name)
    fileList.value.pop()
  }
}
//用于前端小删除按钮的文件删除
function handleFileRemove(file) {
  fileDelFunc(file.name)
}
return {

```

```

    addModal,
    addRef,
    addChart,
    fileList,
    handleAdd,
    addSubmit,
    handleUpload,
    handleClose,
    handleFileRemove,
  }

```

(4) 编辑

- template

```

<!-- 编辑Chart -->
<a-modal
  v-model:visible="updateModal"
  title="编辑Chart"
  :confirm-loading="appLoading"
  cancelText="取消"
  okText="更新"
  :afterClose="handleClose"
  @ok="updateSubmit">
  <a-form ref="updateRef" :model="updateChart" :labelCol="{style: {width:
'30%'}}">
    <a-form-item
      label="Chart名"
      name="name"
      :rules="[{ required: true, message: '请输入Chart名' }]">
      <a-input disabled v-model:value="updateChart.name" style="width:
200px" />
    </a-form-item>
    <a-form-item
      label="版本"
      name="version"
      :rules="[{ required: true, message: '请输入版本' }]">
      <a-input v-model:value="updateChart.version" style="width:
200px" />
    </a-form-item>
    <a-form-item
      label="Icon图标"
      name="icon_url"
      :rules="[{ required: true, message: '请输入图标地址' }]">
      <a-input v-model:value="updateChart.icon_url" addon-
before="http://" style="width: 280px" />
    </a-form-item>
    <a-form-item
      label="描述"
      name="describe"

```

```

        :rules="[ { required: true, message: '请填写Chart描述' } ]">
        <a-textarea v-model:value="updateChart.describe" style="width:
280px" />
    </a-form-item>
    <a-form-item
      label="上传Chart"
      name="chartFile">
      <a-upload
        v-model:file-list="fileList"
        :max-count="2"
        name="chart"
        :customRequest="handleUpload"
        @remove="handleFileRemove">
        <a-button>
          <upload-outlined />
          Click to Upload
        </a-button>
      </a-upload>
    </a-form-item>
  </a-form>
</a-modal>

```

- javascript

```

//编辑
const updateModal = ref(false)
const updateRef = ref()
const updateChart = reactive({
  id: 0,
  name: '',
  icon_url: '',
  version: '',
  describe: '',
  file_name: ''
})
const updateChartData = reactive({
  url: common.helmChartUpdate,
  params: {
    id: 0,
    name: '',
    file_name: '',
    icon_url: '',
    version: '',
    describe: ''
  }
})
function handleUpdate(val) {
  updateChart.id = val.id
  updateChart.name = val.name
  updateChart.icon_url = val.icon_url

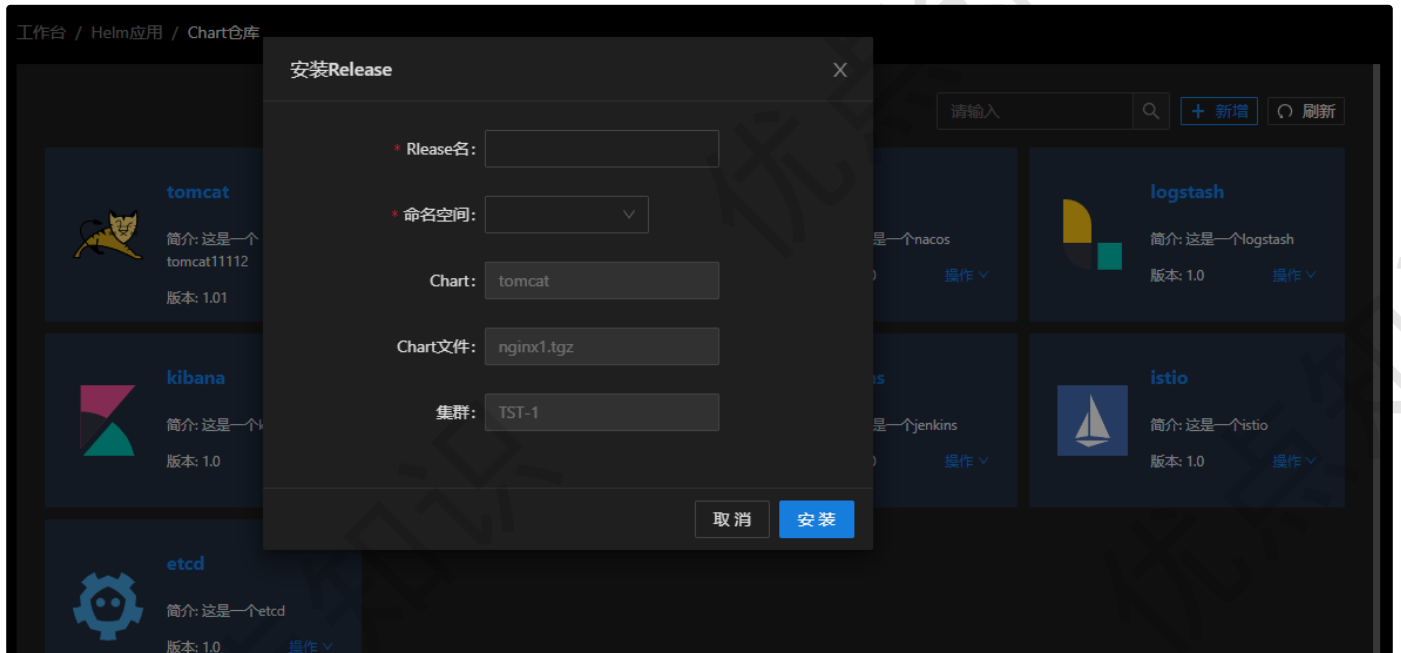
```

```

    updateChart.version = val.version
    updateChart.describe = val.describe
    updateChart.file_name = val.file_name
    updateModal.value = true
  }
  //编辑
  function updateChartFunc() {
    //更新
    appLoading.value = true
    updateChartData.params = updateChart
    httpClient.put(updateChartData.url, updateChartData.params)
    .then(res => {
      message.success(res.msg)
    })
    .catch(res => {
      message.error(res.msg)
    })
    .finally(() => {
      getChartList()
      //重置表单
      updateRef.value.resetFields()
      //关闭抽屉
      updateModal.value = false
      fileList.value.pop()
    })
  }
}
async function updateSubmit() {
  try {
    await updateRef.value.validateFields();
    updateChartFunc()
  } catch (errorInfo) {
    console.log('Failed:', errorInfo);
  }
}
}
return {
  updateModal,
  updateRef,
  updateChart,
  handleUpdate,
  updateSubmit,
}

```

(5) 安装



- template

```

<!-- 安装release -->
<a-modal
  v-model:visible="installModal"
  title="安装Release"
  :confirm-loading="appLoading"
  cancelText="取消"
  okText="安装"
  @ok="installSubmit">
  <a-form ref="installRef" :model="installRelease" :labelCol="{style:
{width: '35%'}}">
    <a-form-item
      label="Release名"
      name="release"
      :rules="[{ required: true, message: '请输入Release名' }]">
      <a-input v-model:value="release" style="width: 200px"/>
    </a-form-item>
    <a-form-item
      label="命名空间"
      name="namespace"
      :rules="[{ required: true, message: '请选择命名空间' }]">
      <a-select show-search style="width:140px;" v-
model:value="namespace" placeholder="请选择">
        <a-select-option
          v-for="(item, index) in namespaceList"
          :key="index"
          :value="item.metadata.name">
          {{ item.metadata.name }}
        </a-select-option>
      </a-select>
    </a-form-item>
  </a-form-item>
</a-modal>

```



```

        label="Chart"
        name="chartName">
        <a-input disabled v-model:value="chartName" style="width:
200px"/>

    </a-form-item>
    <a-form-item
        label="Chart文件"
        name="chartFile">
        <a-input disabled v-model:value="chartFile" style="width:
200px"/>

    </a-form-item>
    <a-form-item
        label="集群"
        name="cluster">
        <a-input disabled v-model:value="cluster" style="width: 200px"/>
    </a-form-item>
    </a-form>
</a-modal>

```

- javascript

```

// 命名空间列表
const namespaceList = ref([])
const namespaceListData = reactive({
  url: common.k8sNamespaceList,
  params: {
    cluster: ''
  }
})
// 安装
const installModal = ref(false)
const installRef = ref()
const installRelease = reactive({
  release: '',
  namespace: '',
  cluster: '',
  chartName: '',
  chartFile: ''
})
const installReleaseData = reactive({
  url: common.helmReleaseInstall,
  params: {
    release: '',
    namespace: '',
    cluster: '',
    chart: ''
  }
})
function handleInstall(val) {
  installRelease.cluster = localStorage.getItem('k8s_cluster')

```

```
installRelease.chartName = val.name
installRelease.chartFile = val.file_name
getNamespaceList()
installModal.value = true
}
// 安装
function installReleaseFunc() {
  appLoading.value = true
  installReleaseData.params.release = installRelease.release
  installReleaseData.params.chart = installRelease.chartFile
  installReleaseData.params.namespace = installRelease.namespace
  installReleaseData.params.cluster = installRelease.cluster
  httpClient.post(installReleaseData.url, installReleaseData.params)
    .then(res => {
      message.success(res.msg)
    })
    .catch(res => {
      message.error(res.msg)
    })
    .finally(() => {
      // 重置表单
      installRef.value.resetFields()
      // 关闭抽屉
      installModal.value = false
      appLoading.value = false
    })
}
// 验证表单
async function installSubmit() {
  try {
    await installRef.value.validateFields();
    installReleaseFunc()
  } catch (errorInfo) {
    console.log('Failed:', errorInfo);
  }
}
return {
  namespaceList,
  installModal,
  installRef,
  installRelease,
  installReleaseFunc,
  handleInstall,
  installSubmit,
}
```

(6) 删除

- javascript

```
// 删除
const delChartData = reactive({
  url: common.helmChartDel,
  params: {
    id: 0,
    file_name: ''
  }
})
// 删除chart
function delChartFunc(val) {
  appLoading.value = true
  delChartData.params.id = val.id
  delChartData.params.file_name = val.file_name
  httpClient.delete(delChartData.url, {data: delChartData.params})
  .then(res => {
    message.success(res.msg)
  })
  .catch(res => {
    message.error(res.msg)
  })
  .finally(() => {
    getChartList()
  })
}
// 确认框
function showConfirm(action, res, fn) {
  Modal.confirm({
    title: '是否继续' + action + "操作? 操作对象: ",
    icon: createVNode(ExclamationCircleOutlined),
    content: createVNode('div', {
      // style: 'color:red;',
    }, res.name),
    cancelText: '取消',
    okText: '确认',
    onOk() {
      fn(res)
    },
    // onCancel() {
    //   console.log('Cancel');
    // }
  })
}
return {
  delChartFunc,
  showConfirm
}
```

