网站建设 Jira Helm 资产文档

一、资产基本介绍

- **资产简介**

> Jira 是Atlassian公司出品的项目与事务跟踪工具，被广泛应用于缺陷跟踪、客户服务、需求收集、流程审批、任务跟踪、项目跟踪和敏捷管理等工作领域，其配置灵活、功能全面、部署简单、扩展丰富。
>
> 基于 Jira 平台构建的产品和应用可以帮助团队规划、指派、跟踪、报告和管理工作。Jira 平台能够将各个团队集合起来完成一切工作，从敏捷软件开发和客户支持，一直到初创公司和大型企业（或者到婚礼筹备和住宅装修）。
>
> 基于 Jira 平台构建的产品为 Jira Software、Jira Service Management 和 Jira Work Management。Jira Align 是衔接规模化工作的企业级敏捷开发规划平台。
>
> Jira Software、Jira Service Management 和 Jira Work Management 均内置适用于不同用例的模板并可无缝集成。因此，组织的各个团队能够更加协调、智能地开展工作。
>
> Jira Helm 模版可以在 Kubernetes 平台上一键部署一个可扩展的Jira系统，同时集成时速云公有云 PaaS 平台的运维功能，实现对 Jira 系统的自动化运维。

- **资产依赖**
  - Jira Helm包：192.168.1.52:/root/helm/jira-2.1.0.tgz Md5值: 3cf8b54fa124bc60383f48c66134f227
  - Jira 镜像：dev-registry.tenxcloud.com/system_containers/jira-software:8.16.0 镜像ID: cb8c072de47b

二、资产购买流程

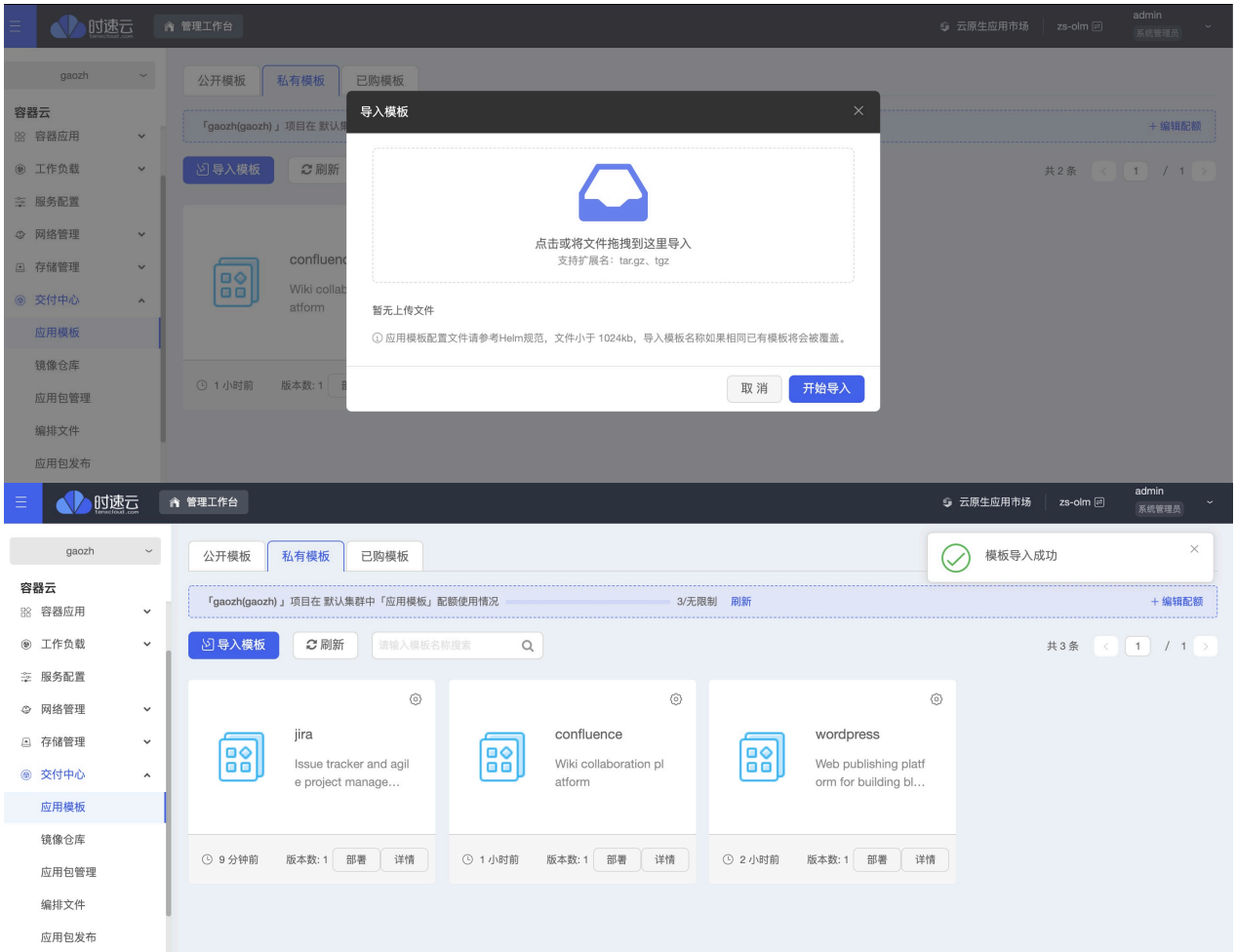
- **在“云市场”中查询需要购买的资产**

- **查看资产详情**
- **购买资产**：在资产详情中点击“购买”
  - 阅读《云原生应用市场用户协议》，同意后勾选“我已阅读并同意...”确认
  - 点击“支付”
- **查看已购资产**：购买资产后会自动跳转到“已购资产”页面显示被购买的资产

三、资产部署指南

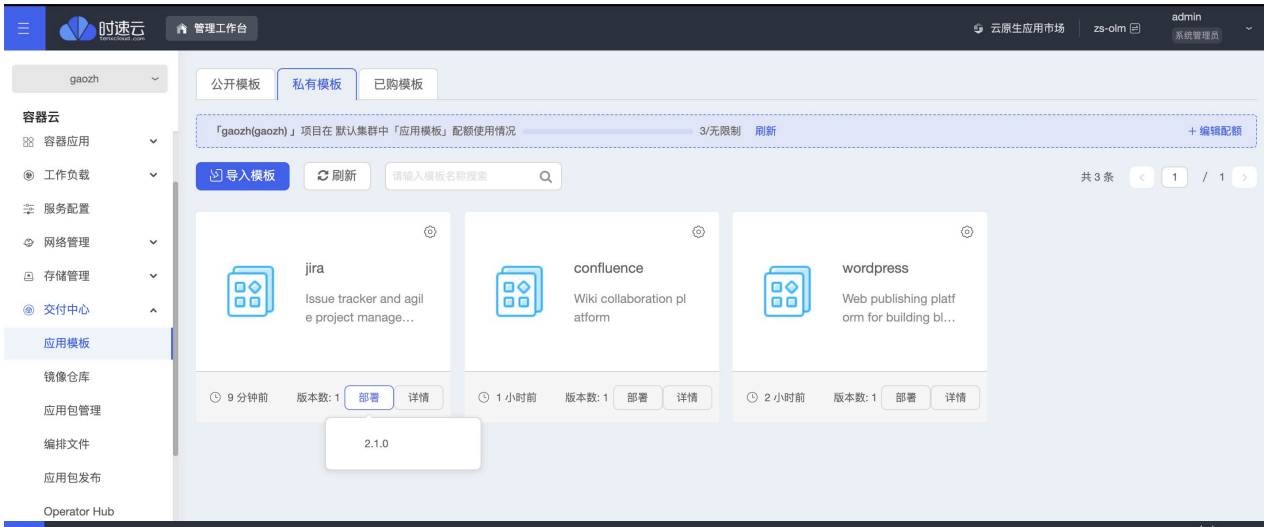
- **查询购买(或导入)Jira模版**
  - 在容器云--交付中心--应用模版 里搜索 " jira ”

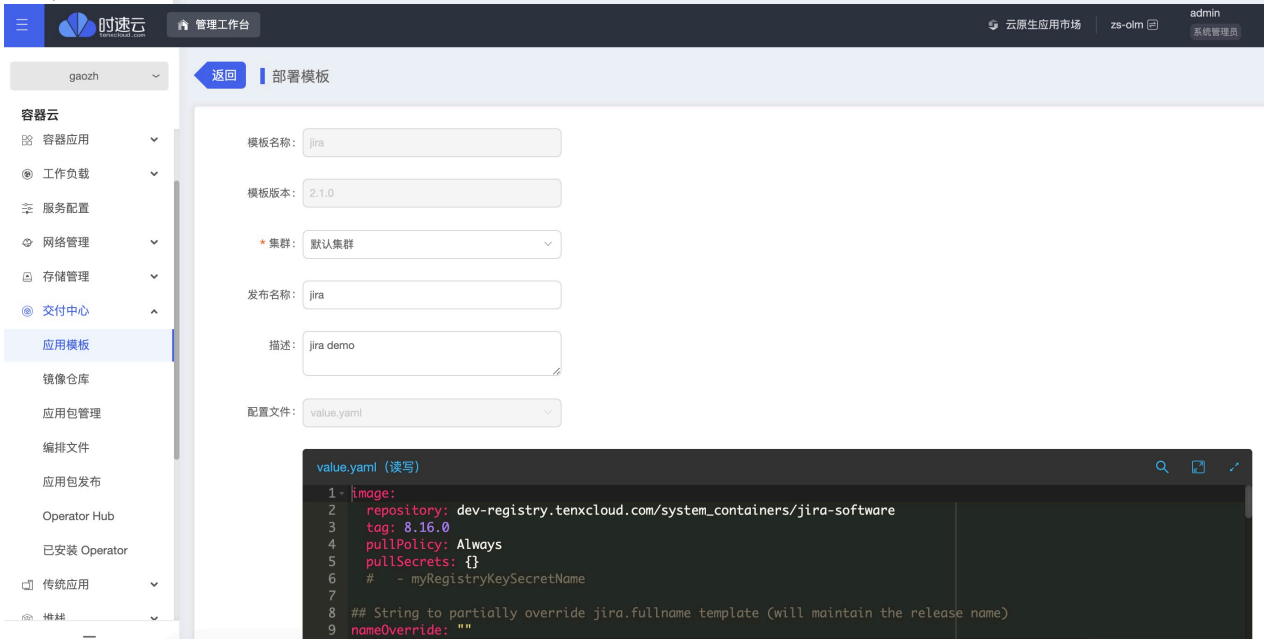
- 如果没有搜到Jira模版，也可以在容器云--交付中心--应用模版--私有模版 里导入私有Jira模版



- **安装 Jira 模版**

- 点击“部署”, 选择版本开始安装





  - 集群: 选择项目所授权的集群
  - 发布名称：指定模版应用的名称，如: " jira "
  - 描述：模版应用的描述信息, 如: " jira demo "
  - 配置文件：

```
image:
  repository: dev-registry.tenxcloud.com/system_containers/confluence-server
  tag: 7.12.0
  pullPolicy: Always
  pullSecrets: {}
  #   - myRegistryKeySecretName

  ## String to partially override wiki.fullname template (will maintain the release name)
  nameOverride: ""

## String to fully override wiki.fullname template
```

```
fullnameOverride: ""

## ref: https://kubernetes.io/docs/tasks/configure-pod-
container/configure-service-account/
serviceAccount:
  # Specifies whether a service account should be created
  create: false
  # Annotations to add to the service account
  annotations: {}
  # The name of the service account to use.
  # If not set and create is true, a name is generated using the fullname
template
  name: ""

## ref: https://kubernetes.io/docs/reference/generated/kubernetes-
api/v1.17/#podsecuritycontext-v1-core
podSecurityContext:
  fsGroup: 2002

## Security context
## ref: https://kubernetes.io/docs/tasks/configure-pod-container/security-
context/
securityContext: {}
  # capabilities:
  #   drop:
  #   - ALL
  # readOnlyRootFilesystem: true
  # runAsNonRoot: true
  # runAsUser: 1000

## Service/Networking
## ref: https://kubernetes.io/docs/concepts/services-networking/service/
service:
  ## For minikube, set this to NodePort, elsewhere use LoadBalancer
  type: ClusterIP
  ## Use serviceLoadBalancerIP to request a specific static IP, otherwise
leave blank
  ##
  ## Avoid removing the http connector, as the Synchrony proxy health
check, still requires HTTP
  ## HTTP Port, must be the same as ATL_TOMCAT_PORT (default: 8090)
  port: 8090
  ## HTTPS Port, in case ATL_TOMCAT_SCHEME is set to 'https'
  httpsPort:
  loadBalancerIP:
  ## Use nodePorts to requets some specific ports when usin NodePort
  ## nodePorts:
  ##   http: <to set explicitly, choose port between 30000-32767>
  ##   https: <to set explicitly, choose port between 30000-32767>
```

```
  ##
  nodePorts:
    http:
    https:

## Configure the ingress resource that allows you to access the
## Confluence installation. Set up the URL
## ref: http://kubernetes.io/docs/user-guide/ingress/
ingress:
  ## Set to true to enable ingress record generation
  enabled: false
  annotations: {}
  # kubernetes.io/ingress.class: nginx
  # kubernetes.io/tls-acme: "true"
  hosts:
    - host: confluence-server.local
      paths: []
  tls: []
  # - secretName: confluence-server.local-tls
  #   hosts:
  #     - confluence-server.local

## ref: https://kubernetes.io/docs/concepts/configuration/manage-compute-
resources-container/
resources:
  requests:
    cpu: 500m
    memory: 4Gi
  limits:
    cpu: 4
    memory: 8Gi

## Replication (without ReplicaSet)
## ref:
https://kubernetes.io/docs/concepts/workloads/controllers/deployment/
replicaCount: 1

## Node labels for pod assignment
## ref: https://kubernetes.io/docs/user-guide/node-selection/
nodeSelector: {}

## Tolerations for pod assignment
## ref: https://kubernetes.io/docs/concepts/configuration/taint-and-
toleration/
tolerations: []

## Affinity for pod assignment
## ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-
node/#affinity-and-anti-affinity
```

```
affinity: {}

## Pod annotations
## ref: https://kubernetes.io/docs/concepts/overview/working-with-
objects/annotations/
podAnnotations: {}

## Persistent Volume Claim
## Confluence Home directory
## https://kubernetes.io/docs/concepts/storage/persistent-volumes/
persistence:
  enabled: true
  annotations: {}
  ## existingClaim needs the existing PVC name
  existingClaim: ""
  accessMode: ReadWriteOnce
  size: 1Gi

  ## If defined, storageClassName: <storageClass>
  ## If set to "-", storageClassName: "", which disables dynamic
provisioning
  ## If undefined (the default) or set to nil, no storageClassName spec is
  ##   set, choosing the default provisioner.  (gp2 on AWS, standard on
  ##   GKE, AWS & OpenStack)
  ##
  storageClass: nfs-94

# Additional volume mounts
extraVolumeMounts: []
  ## Example: Mount CA file
  # - name: ca-cert
  #   subPath: ca_cert
  #   mountPath: /path/to/ca_cert

# Additional volumes
extraVolumes: []
  ## Example: Add secret volume
  # - name: ca-cert
  #   secret:
  #     secretName: ca-cert
  #     items:
  #     - key: ca-cert
  #       path: ca_cert

## Use an alternate scheduler, e.g. "stork".
## ref: https://kubernetes.io/docs/tasks/administer-cluster/configure-
multiple-schedulers/
schedulerName: ""
```

```
## Container Probes
## ref: https://kubernetes.io/docs/concepts/workloads/pods/pod-
lifecycle/#container-probes
## ref: https://kubernetes.io/docs/tasks/configure-pod-
container/configure-liveness-readiness-probes/#configure-probes
## Depending what values we give, Confluence won't be reachable. In doubt,
leave it as it is.
        readinessProbe: {}
#  httpGet:
#    path: /status
#    port: http
#  initialDelaySeconds: 300
#  periodSeconds: 30
#  failureThreshold: 6
#  timeoutSeconds: 10

livenessProbe: {}
#  httpGet:
#    path: /status
#    port: http
#  initialDelaySeconds: 480
#  periodSeconds: 30
#  failureThreshold: 6
#  timeoutSeconds: 10

## Environment Variables that will be injected in the ConfigMap
## Default values unless otherwise stated
envVars:
  ## Memory / Heap Size (JVM_MINIMUM_MEMORY) Mandatory, see @Notes above
  ## default: 1024m
  JVM_MINIMUM_MEMORY: 2048m
  ## Memory / Heap Size (JVM_MAXIMUM_MEMORY) Mandatory, see @Notes above
  ## default: 1024m
  JVM_MAXIMUM_MEMORY: 2048m
  #
  ## Tomcat and Reverse Proxy Settings
  ## Confluence running behind a reverse proxy server options
  ## Note - When ingress is enabled:
  ## These values are set automatically. Do not uncomment these proxy
settings.
  # ATL_PROXY_NAME: ""
  # ATL_PROXY_PORT: ""
  # ATL_TOMCAT_PORT: 8090
  # ATL_TOMCAT_SCHEME: http
  # ATL_TOMCAT_SECURE: false
  # ATL_TOMCAT_CONTEXTPATH: ""
  #
  ## Tomcat/Catalina options
  ## ref: https://tomcat.apache.org/tomcat-7.0-doc/config/index.html
```

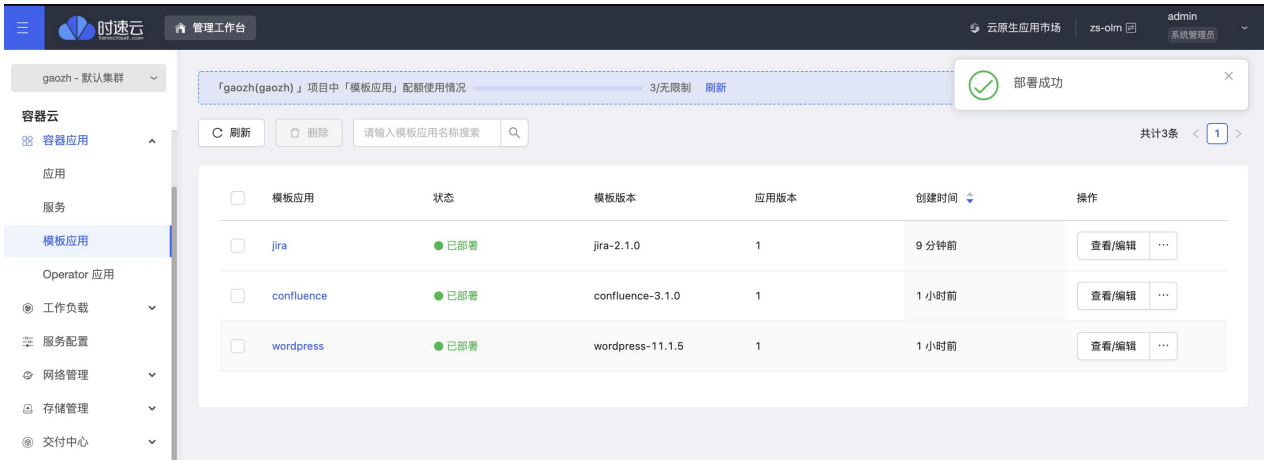
```
  # ATL_TOMCAT_MGMT_PORT: 8000
  # ATL_TOMCAT_MAXTHREADS: 100
  # ATL_TOMCAT_MINSPARETHREADS: 10
  # ATL_TOMCAT_CONNECTIONTIMEOUT: 20000
  # ATL_TOMCAT_ENABLELOOKUPS: false
  # ATL_TOMCAT_PROTOCOL: "HTTP/1.1"
  # ATL_TOMCAT_ACCEPTCOUNT: 10
  #
  ## Cookie age (Remember Me maximum time remain logged-in)
  # ATL_AUTOLOGIN_COOKIE_AGE: 1209600
  #
  ## Home directory. This may be on a mounted volume; if so it
  ## should be writable by the user confluence. See note below about UID
mappings.
  # CONFLUENCE_HOME: ""
  #
  ## Optional connection pool database settings
  # ATL_DB_POOLMINSIZE: 20
  # ATL_DB_POOLMAXSIZE: 100
  # ATL_DB_TIMEOUT: 30
  # ATL_DB_IDLETESTPERIOD: 100
  # ATL_DB_MAXSTATEMENTS: 0
  # ATL_DB_VALIDATE: false
  # ATL_DB_ACQUIREINCREMENT: 1
  # ATL_DB_VALIDATIONQUERY: "select 1"
## End of Environment Variables (envVars)

## JVM_SUPPORT_RECOMMENDED_ARGS
## Additional container environment variables
# extraEnv: "-XX:MaxMetaspaceSize=512m -XX:MaxDirectMemorySize=10m -
Dsynchrony.memory.max=0m"
```

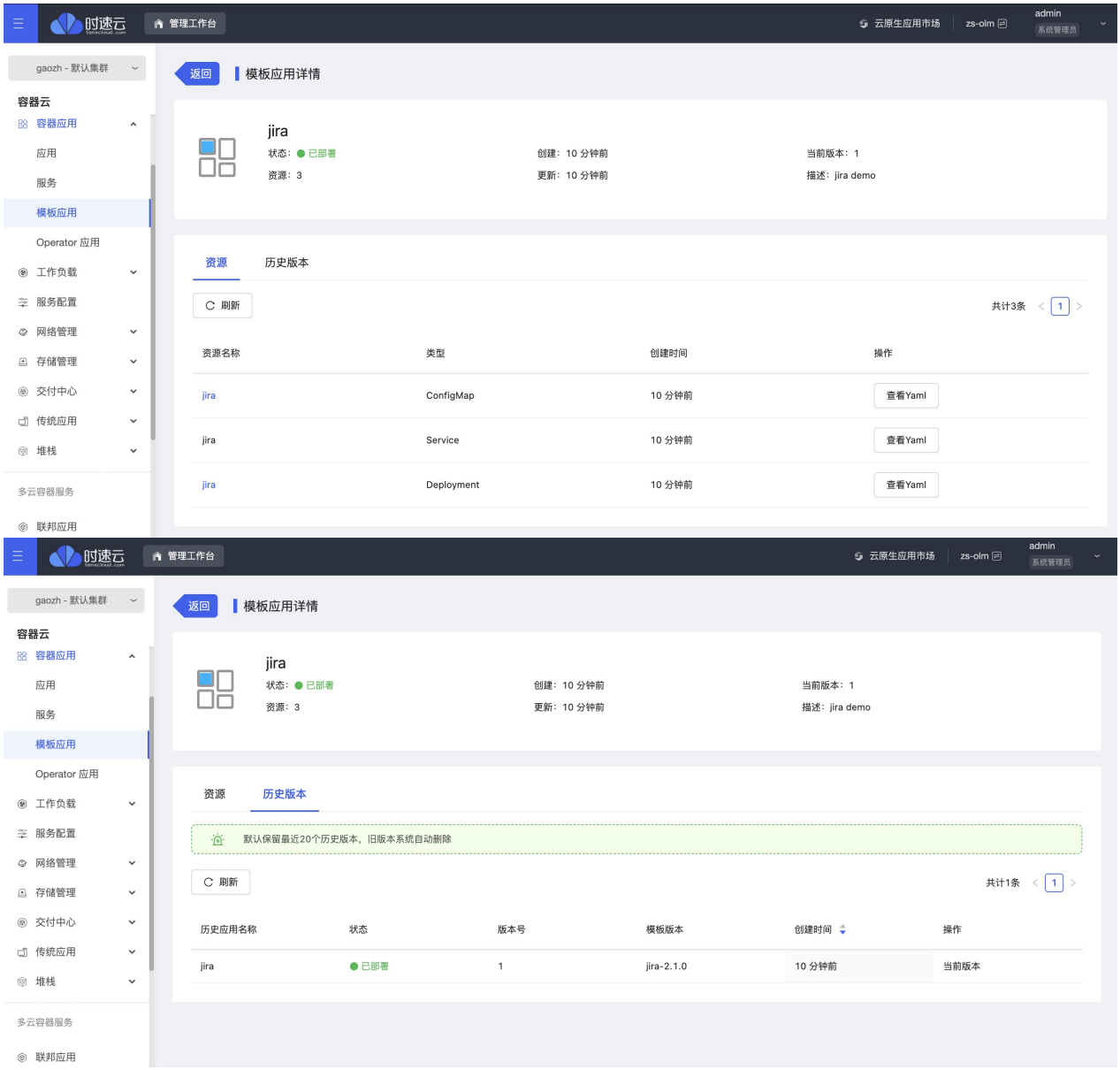
- 基础配置说明： * image.repository: 镜像地址，指定具体的jira-software地址, 如： "dev-registry.tenxcloud.com/system_containers/jira-software" * image.tag: 镜像tag，如："8.16.0" * resources.requests: 每个Pod 请求的 CPU、内存资源大小，推荐使用 2C/4G 配置 * resources.limits: 每个Pod 请求的 CPU、内存资源大小，推荐使用 4C/8G 配置, 如果资源充足可以适当设置大一些 * persistence.storageClass: 集群使用的存储类名称，从“ 容器云--存储管理--存储卷--创建存储卷--存储类下拉列表” 中可以查看到可以使用的存储类

* persistence.size: 存储大小，可根据存储的资源情况进行设置

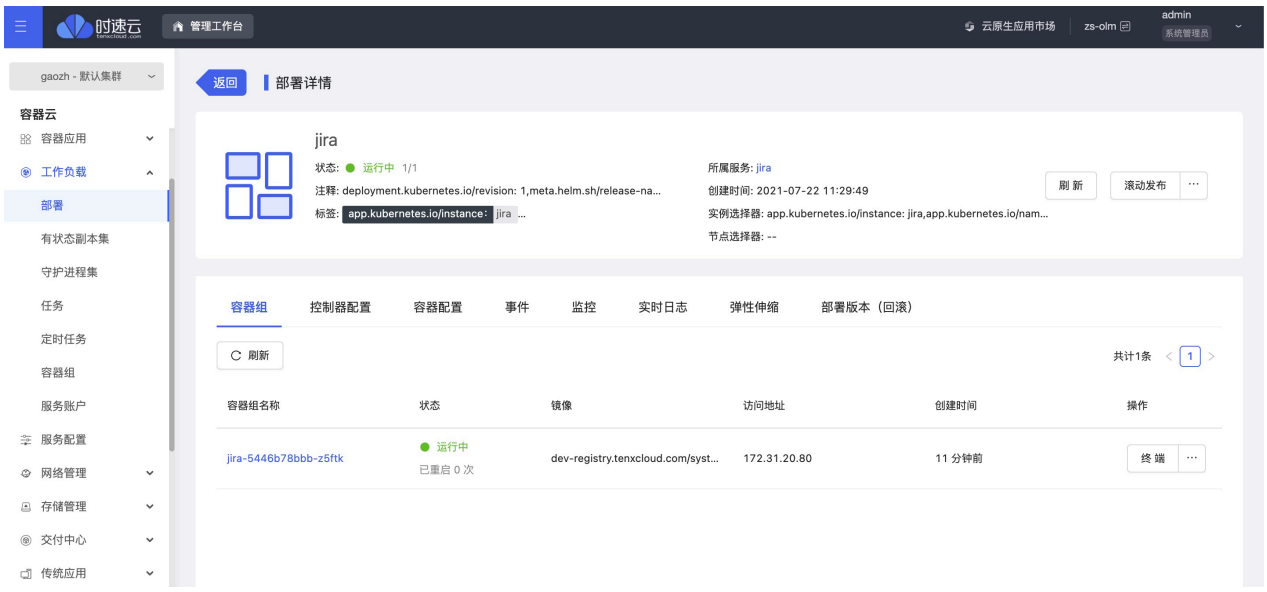
- 点击“确定”：自动跳转到 模版应用 菜单



- 点击 模版应用 名称，进入 模版应用 详情

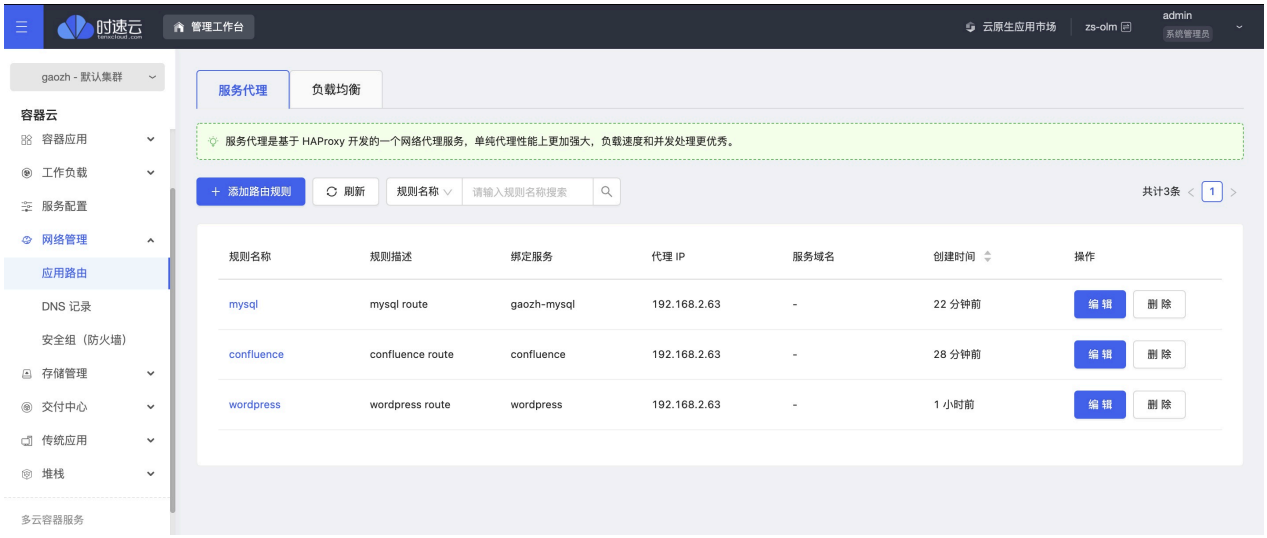


- 查看 实例运行 状态

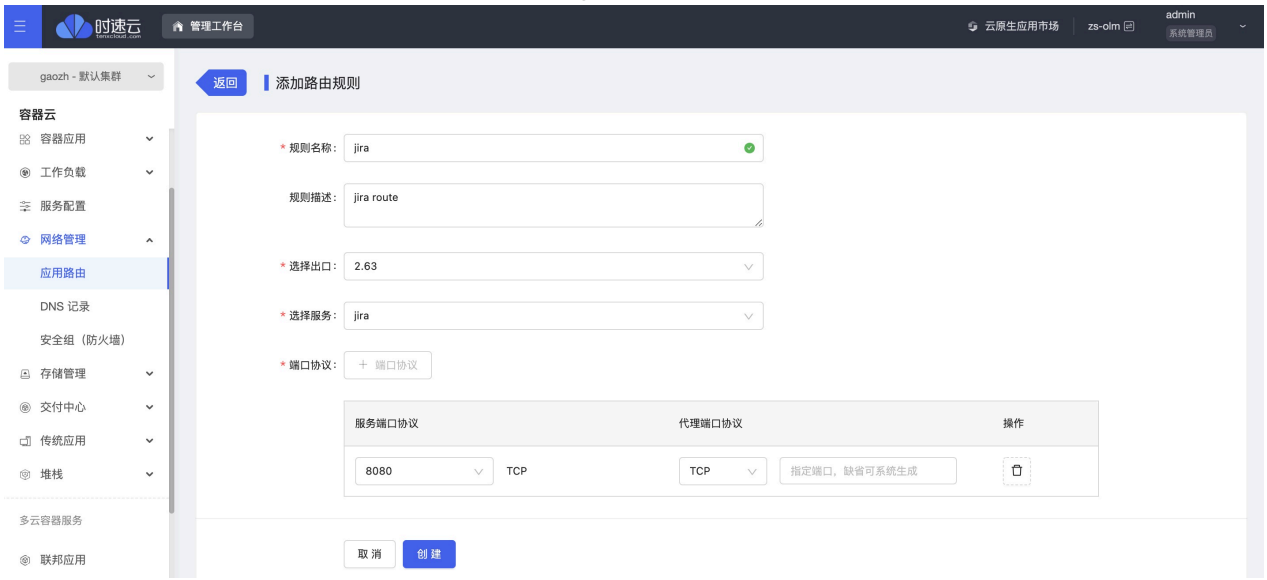


- **配置 Jira 集群外访问**

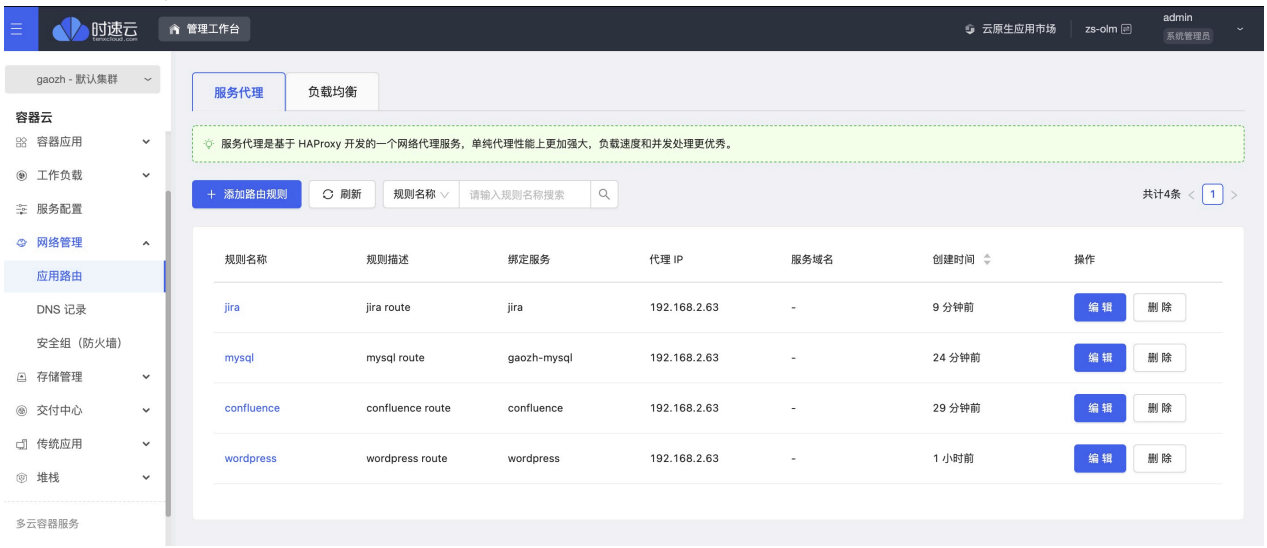
- 在“容器云--网络管理--应用路由”页面，点击“添加路由规则”



- 在“添加路由规则”页面，选择一个服务出口代理 Jira 服务

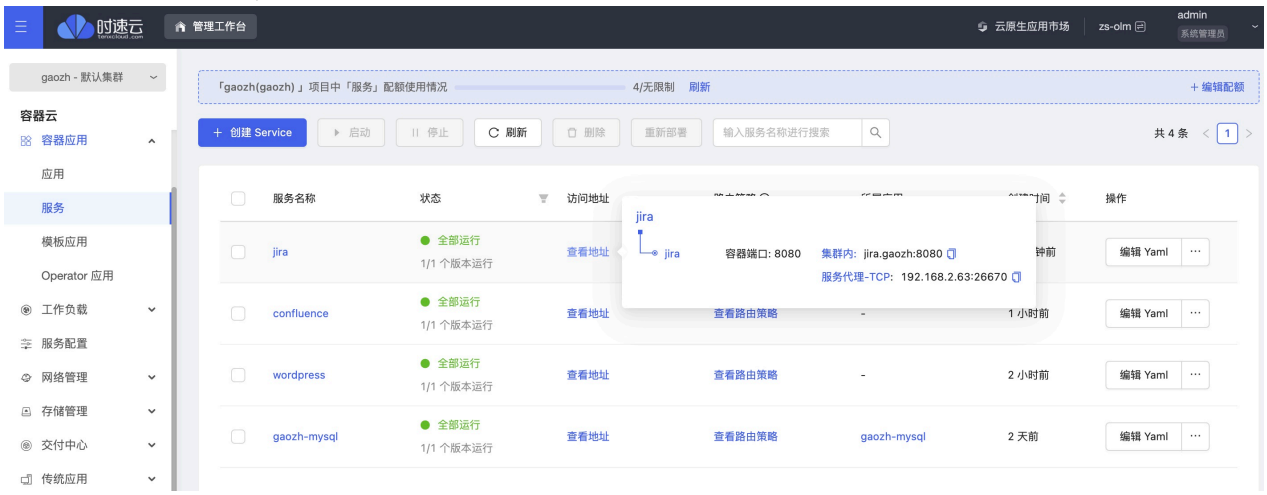


  - 规则名称：这条路由规则的名称, 如："jira"
  - 选择出口：选择一个平台的服务访问出口
  - 选择服务：Jira 服务, 如："jira"
- 点击“创建”后，可以看到生成的路由规则



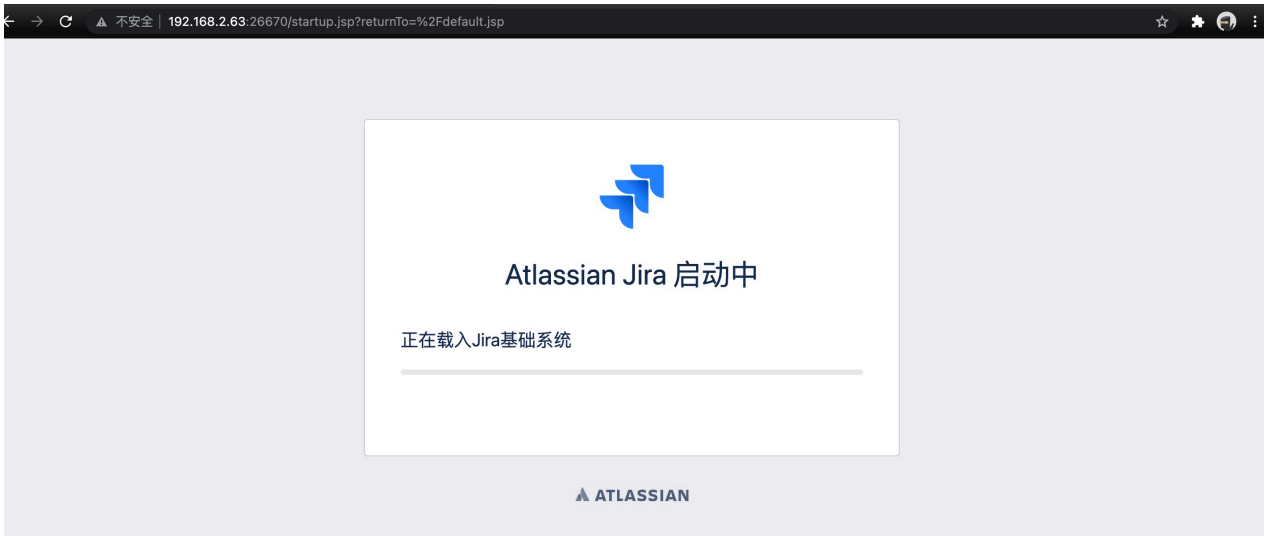
- **验证 Jira 服务状态**

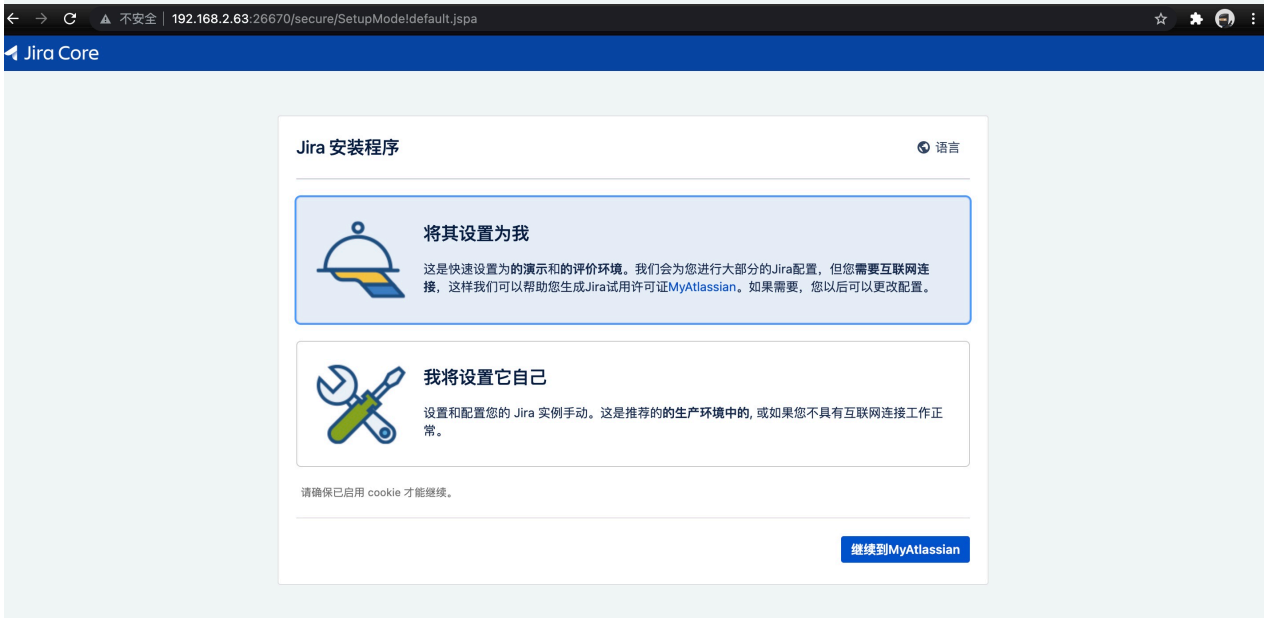
- 在“容器云--容器应用--容器服务”列表中，找到被代理的 jira 服务，点击“查看地址”，点击地址旁边的拷贝图标保存地址信息，用于后面访问验证。



  - 集群内：在 Kuberenetes 集群内访问 jira 服务，使用这个地址
  - 服务代理-TCP：在 Kuberenetes 集群外访问 jira 服务，使用这个地址

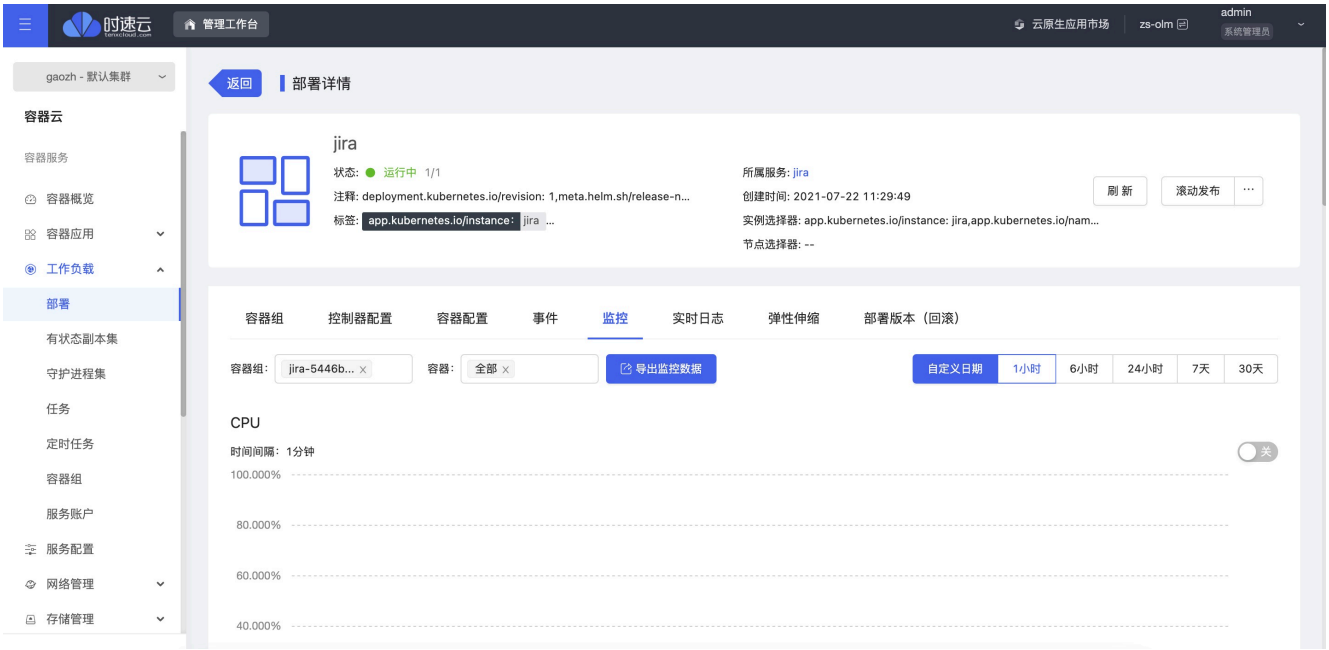
- 打开浏览器输入集群外访问地址，如"192.168.2.63:26670"



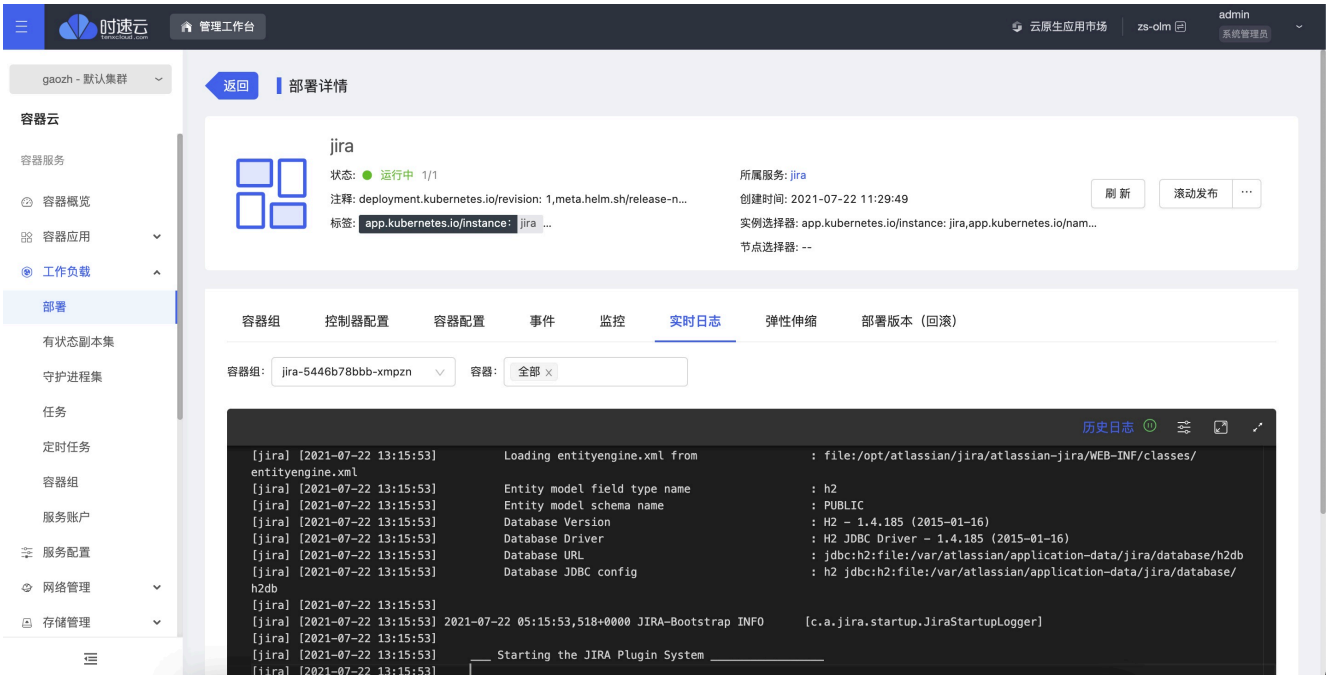


四、应用运维指南；（补充界面部署方式）

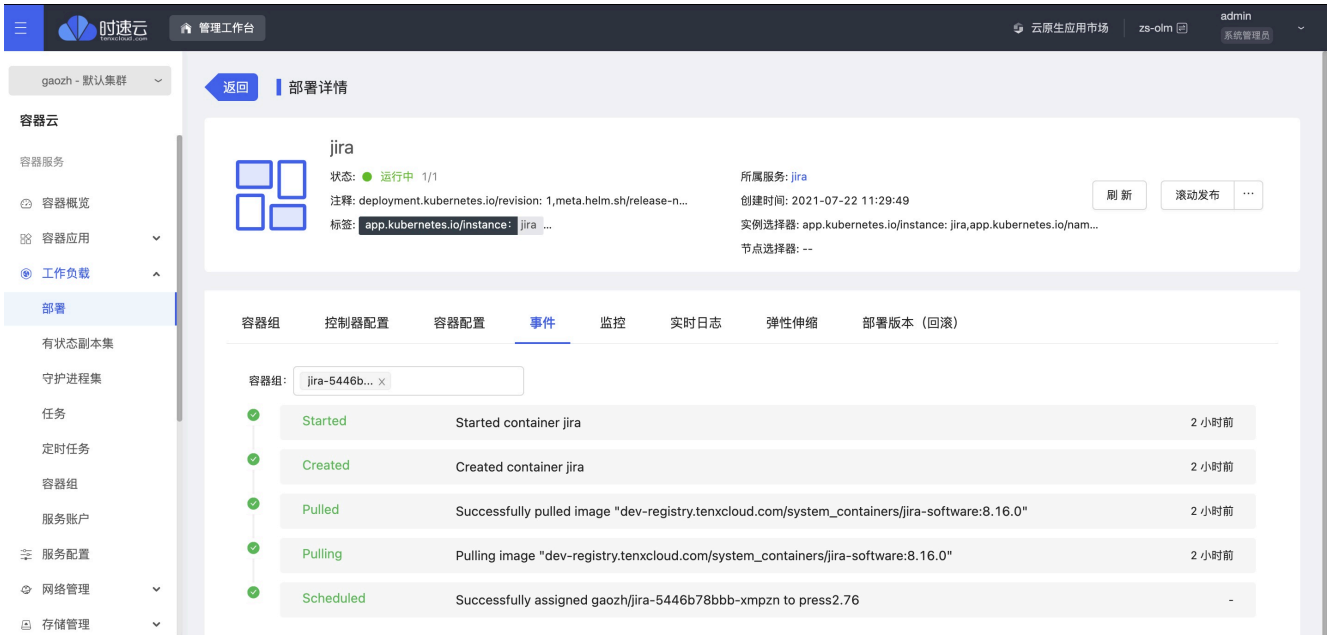
- **监控信息查看**：在“容器云--工作负载--部署”，点击进入 Jira 副本集



- **日志信息查看**：在“容器云--工作负载--部署”，点击进入 Jira 副本集



- **事件信息查看**：在“容器云--工作负载--部署”，点击进入 Jira 副本集



- **审计信息查看**：在“安全和运维--平台运维--操作审计--审计记录”，选择“容器云/容器应用/模版应用”、相应租户、项目后点击“立即查询”