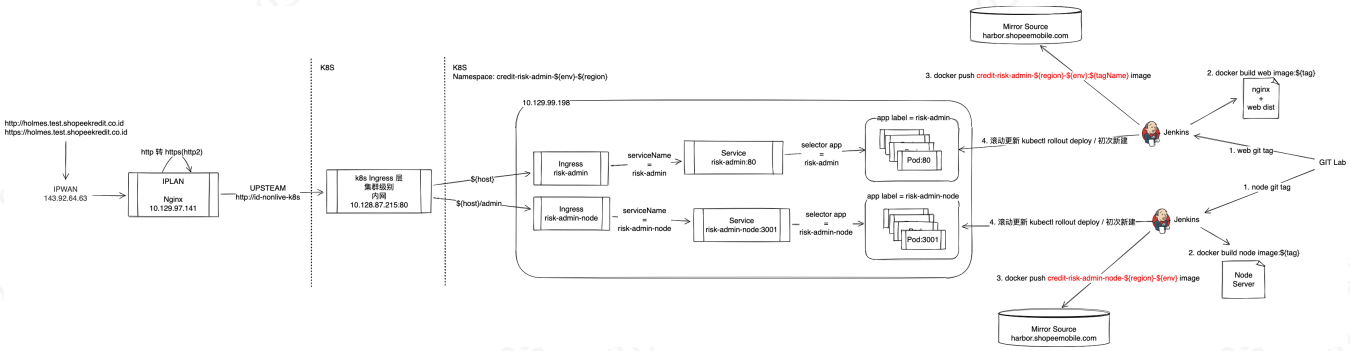
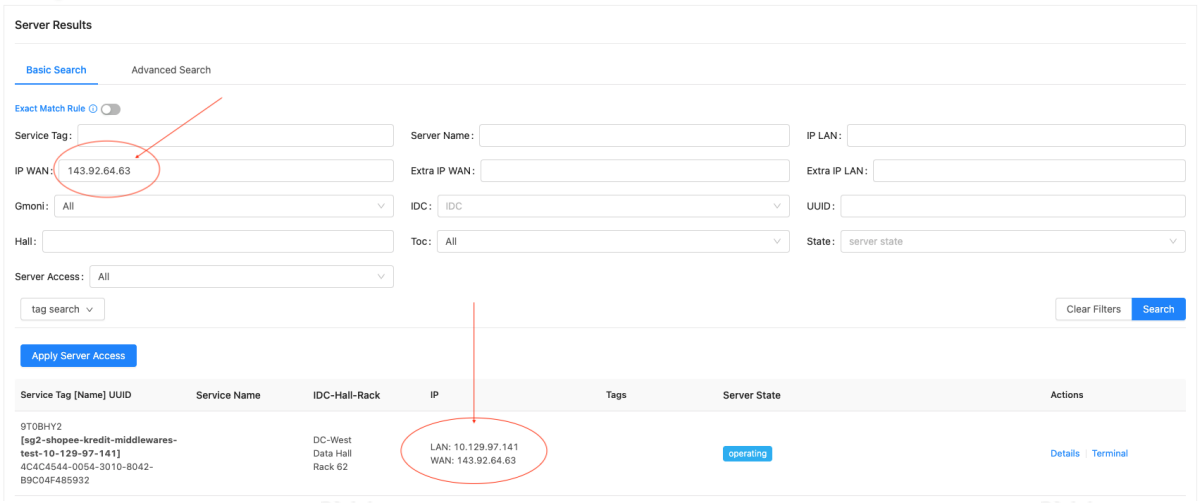
[Holmes] 服务部署 & 链路指北



- 1. 本地 ping 域名，达到 143.92.64.63，可在 TOC 查看 对应 LAN IP，即 10.129.97.141



- 2. 对应域名 nginx 设置 /etc/nginx/${regin}-conf.d/，upstream 设置 /etc/nginx/id-upstream/ 或 /etc/nginx/upstream，通过 upstream 设置可查看转发 IP，转发至【credit 集群 k8s ingress 层】

nginx

```
# http  https
server {
    listen              80;
    server_name         holmes.test.shopeekredit.co.id;
    client_max_body_size 120M;
    location /{
        return 307 https://holmes.test.shopeekredit.co.id$request_uri;
    }
}

# http2
server {
    listen              443 ssl http2 ;
    server_name         holmes.test.shopeekredit.co.id;
    client_max_body_size 120M;
    keepalive_timeout   70;

    access_log /var/log/nginx/holmes.test.shopeekredit.co.id.access.log main buffer=16k;
    error_log /var/log/nginx/holmes.test.shopeekredit.co.id.error.log;
    ssl_certificate     /etc/nginx/ssl/wildcard.test.shopeekredit.co.id/ssl.chain.crt;
    ssl_certificate_key /etc/nginx/ssl/wildcard.test.shopeekredit.co.id/server.key;
    ssl_protocols        TLSv1.1 TLSv1.2;
    ssl_ciphers         HIGH:!aNULL:!MD5;

    location / {
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
        proxy_pass http://id-nonlive-k8s; #
    }
}
```

nginx upstream

```
upstream id-nonlive-k8s {
    keepalive 100;
    server 10.128.87.215:80;
}
```

- **3. 10.129.99.198 为部署 IP，查看对应域名 ingress：kubectl get ing -A | grep holmes.test.shopeekredit.co.id，查看 yaml 配置详情 kubectl get ing risk-admin -n credit-risk-admin-test-id -oyaml ：**

namespace：credit-risk-admin-test-id（namespace 规则 credit-risk-admin-${env}-${region}），根据 namespace 可查看关联的 ingress / services / pods 等

ingress 分别对应（risk-admin）前端项目一份，（risk-admin-node）后端项目一份，具体域名对应路径是怎么转发的，根据 rules 内 paths 字段可看到配置：

risk-admin ingress yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/proxy-body-size: 120m
    nginx.ingress.kubernetes.io/use-regex: "true"
  creationTimestamp: "2021-04-22T07:07:37Z"
  generation: 7
  name: risk-admin
  namespace: credit-risk-admin-test-id
  resourceVersion: "228119579"
  selfLink: /apis/extensions/v1beta1/namespaces/credit-risk-admin-test-id/ingresses/risk-admin
  uid: 6b4f85dc-a339-11eb-9514-8069336b8bc8
spec:
  rules:
  - host: holmes.test.shopeekredit.co.id
    http:
      paths:
      - backend:
          serviceName: risk-admin
          servicePort: http
  - host: risk-admin.credit-risk-admin-test-id.api.ingress
    http:
      paths:
      - backend:
          serviceName: risk-admin
          servicePort: http
status:
  loadBalancer:
    ingress:
    - ip: 100.127.189.217
```

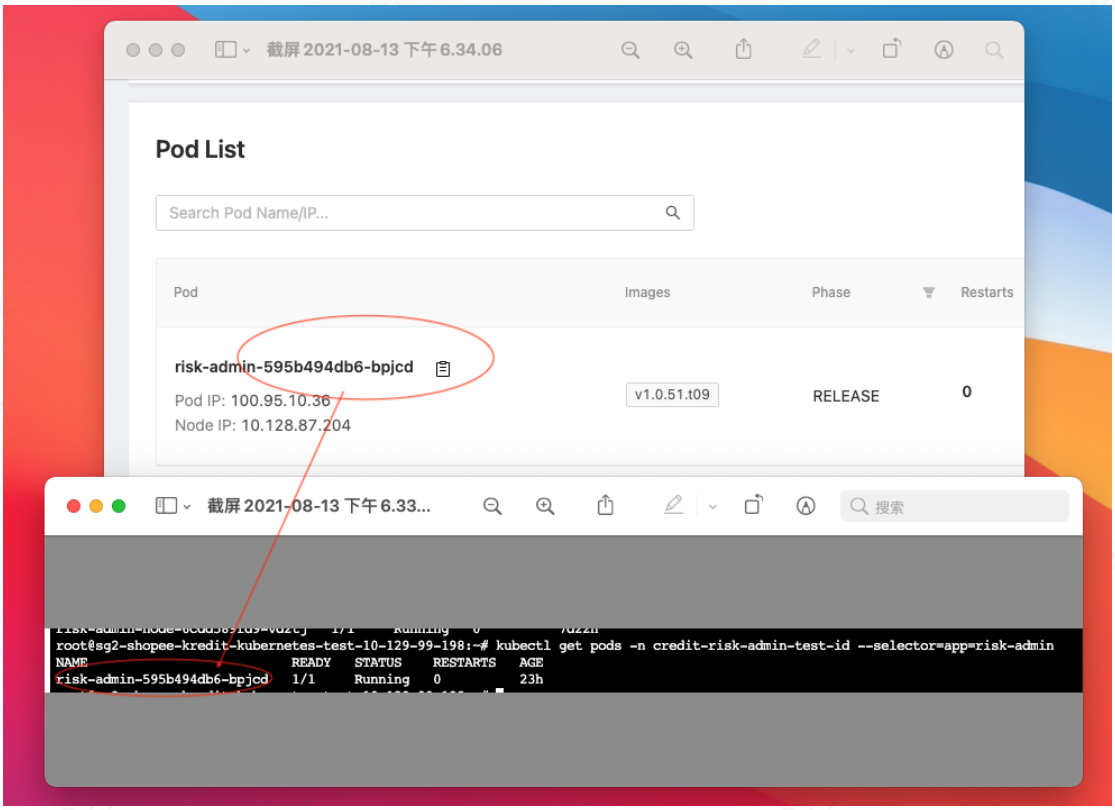
risk-admin-node ingress yaml 配置内 path 路径为 /admin：

risk-admin-node ingress yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/proxy-body-size: 120m
    nginx.ingress.kubernetes.io/use-regex: "true"
  creationTimestamp: "2021-04-21T09:37:52Z"
  generation: 12
  name: risk-admin-node
  namespace: credit-risk-admin-test-id
  resourceVersion: "227760189"
  selfLink: /apis/extensions/v1beta1/namespaces/credit-risk-admin-test-id/ingresses/risk-admin-node
  uid: 7cd310c4-2cc4-4e7c-a91b-e30cb718b112
spec:
  rules:
  - host: holmes.test.shopeekredit.co.id
    http:
      paths:
      - backend:
          serviceName: risk-admin-node
          servicePort: http
        path: /admin/
  - host: risk-admin-node.credit-risk-admin-test-id.api.ingress
    http:
      paths:
      - backend:
          serviceName: risk-admin-node
          servicePort: http
        path: /admin/
status:
  loadBalancer:
    ingress:
    - ip: 100.127.189.217
```

- **4. ingress 配置内，rules 内通过 serviceName 访问对应的 serivce，可通过 kubectl get services -A | grep credit-risk-admin-test-id 查看对应的 service 服务（grep 对应的 namespace），通过 kubectl get services risk-admin -n credit-risk-admin-test-id -oyaml 查看对应 yaml 配置:**

通过 selector app = risk-admin 获取到对应部署 pods（多个 pods 指定同一 label name），node 后端也是同样的道理，查看关联这个 selector app name 的 pods 可使用 kubectl get pods -n credit-risk-admin-test-id --selector=app=risk-admin（指定对应 namespace 与 selector app name），与 K8S 平台是能对应上的:



risk-admin（前端）service yaml 配置：

risk-admin service yaml

```
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: "2021-04-22T07:07:37Z"
  labels:
    app: risk-admin-test-id
    application: risk-admin-test-id
    cid: id
    env: test
    group: SeaMoney-Credit
    project: credit-risk-admin
  name: risk-admin
  namespace: credit-risk-admin-test-id
  resourceVersion: "211666784"
  selfLink: /api/v1/namespaces/credit-risk-admin-test-id/services/risk-admin
  uid: c2718b8d-8c6a-438f-a168-23a2628495ec
spec:
  clusterIP: 100.127.184.204
  ports:
  - name: http
    port: 80
    protocol: TCP
    targetPort: 80
  - name: metric-monitor
    port: 30009
    protocol: TCP
    targetPort: 30009
  selector:
    app: risk-admin
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
```

risk-admin-node service 指定 selector app = risk-admin-node：

risk-admin-node service yaml

```
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: "2021-04-21T09:37:52Z"
  labels:
    app: risk-admin-node-test-id
    application: risk-admin-node-test-id
    cid: id
    env: test
    group: SeaMoney-Credit
    project: credit-risk-admin
  name: risk-admin-node
  namespace: credit-risk-admin-test-id
  resourceVersion: "211666403"
  selfLink: /api/v1/namespaces/credit-risk-admin-test-id/services/risk-admin-node
  uid: 7d357839-38a5-46b1-8e7d-4d7503161fc1
spec:
  clusterIP: 100.127.190.186
  ports:
  - name: http
    port: 3001
    protocol: TCP
    targetPort: 3001
  - name: metric-monitor
    port: 30009
    protocol: TCP
    targetPort: 30009
  selector:
    app: risk-admin-node
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
```

- **5. pods 为一个个可部署最小单元，一个个应用容器，可根据 kubectl get pods -n credit-risk-admin-test-id --selector=app=risk-admin -oyaml 获取到对应的 yaml 配置文件:**

1. metadata.labels.app 制定了 label 名称，services 是通过此方式关联到对应 pods 是哪些
2. namespace 为指定命名空间
3. image 指定当前部署镜像是什么（比如 harbor.shopeemobile.com/airpay/credit-risk-admin-id-test:v1.0.51.t09，harbor.shopeemobile.com 是镜像仓库地址，credit-risk-admin-${region}-${env}:${tagName}，这个 tagName 对应 jenkins 流水线的发布 git tag name）
4. 其他配置细节可自行慢慢查看~

risk-admin pods yaml

```
apiVersion: v1
items:
- apiVersion: v1
  kind: Pod
  metadata:
    annotations:
      cni.projectcalico.org/podIP: 100.95.10.36/32
      kube-platform.shopee.io/application: risk-admin
      kube-platform.shopee.io/cid: id
      kube-platform.shopee.io/env: test
      kube-platform.shopee.io/group: SeaMoney Credit
      kube-platform.shopee.io/project: credit-risk-admin
```

```
    prometheus.io/scrape: "true"
  creationTimestamp: "2021-08-12T11:10:19Z"
  generateName: risk-admin-595b494db6-
  labels:
    app: risk-admin
    application: risk-admin
    cid: id
    env: test
    group: SeaMoney-Credit
    metrics: "true"
    phase: release
    pod-template-hash: 595b494db6
    project: credit-risk-admin
  name: risk-admin-595b494db6-bpjcd
  namespace: credit-risk-admin-test-id
  ownerReferences:
  - apiVersion: apps/v1
    blockOwnerDeletion: true
    controller: true
    kind: ReplicaSet
    name: risk-admin-595b494db6
    uid: 2e0aeb3a-7a27-4731-a0ab-4b4840da8c7c
  resourceVersion: "233711841"
  selfLink: /api/v1/namespaces/credit-risk-admin-test-id/pods/risk-admin-595b494db6-bpjcd
  uid: 373f9b9b-c171-42a4-b384-0678e4a9458d
spec:
  containers:
  - env:
    - name: HOST
      value: holmes.test.shopeekredit.co.id
    - name: CPU
      value: "0.2"
    - name: MEM
      value: 0.4Gi
    - name: ENV
      value: test
    - name: REGION
      value: id
    - name: POD_PORT
      value: "80"
    - name: REDIS_HOSTS
      value: $REDIS_HOSTS
    - name: ETCD_HOSTS
      value: $ETCD_HOSTS
    - name: APP_NAME
      value: risk-admin
    - name: APP_TYPE
      value: credit
    - name: MONITOR_PORT
      value: "30009"
    - name: POD_NAME
      valueFrom:
        fieldRef:
          apiVersion: v1
          fieldPath: metadata.name
    - name: HOST_IP
      valueFrom:
        fieldRef:
          apiVersion: v1
          fieldPath: status.hostIP
    image: harbor.shopeemobile.com/airpay/credit-risk-admin-id-test:v1.0.51.t09
    imagePullPolicy: IfNotPresent
    lifecycle:
      preStop:
        exec:
          command:
          - sh
          - -c
          - sleep 5
    livenessProbe:
      failureThreshold: 3
```

```
      initialDelaySeconds: 15
      periodSeconds: 20
      successThreshold: 1
      tcpSocket:
        port: 80
      timeoutSeconds: 1
    name: risk-admin
    ports:
    - containerPort: 80
      name: http
      protocol: TCP
    - containerPort: 30009
      name: metrics
      protocol: TCP
    readinessProbe:
      failureThreshold: 3
      initialDelaySeconds: 5
      periodSeconds: 10
      successThreshold: 1
      tcpSocket:
        port: 80
      timeoutSeconds: 1
    resources:
      limits:
        cpu: 200m
        memory: 429496729600m
      requests:
        cpu: 200m
        memory: 429496729600m
    terminationMessagePath: /dev/termination-log
    terminationMessagePolicy: File
    volumeMounts:
    - mountPath: /data/server/logs/
      name: node-logs
    - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
      name: default-token-h78zs
      readOnly: true
  dnsPolicy: ClusterFirst
  enableServiceLinks: true
  nodeName: sg2-shopee-kredit-kubernetes-test-10-128-87-204
  priority: 0
  restartPolicy: Always
  schedulerName: default-scheduler
  securityContext: {}
  serviceAccount: default
  serviceAccountName: default
  terminationGracePeriodSeconds: 30
  tolerations:
  - effect: NoExecute
    key: node.kubernetes.io/not-ready
    operator: Exists
    tolerationSeconds: 300
  - effect: NoExecute
    key: node.kubernetes.io/unreachable
    operator: Exists
    tolerationSeconds: 300
  volumes:
  - hostPath:
      path: /data/log/credit-risk-admin-test-id
      type: ""
    name: node-logs
  - name: default-token-h78zs
    secret:
      defaultMode: 420
      secretName: default-token-h78zs
status:
  conditions:
  - lastProbeTime: null
    lastTransitionTime: "2021-08-12T11:10:19Z"
    status: "True"
    type: Initialized
```

```
    - lastProbeTime: null
      lastTransitionTime: "2021-08-12T11:10:35Z"
      status: "True"
      type: Ready
    - lastProbeTime: null
      lastTransitionTime: "2021-08-12T11:10:35Z"
      status: "True"
      type: ContainersReady
    - lastProbeTime: null
      lastTransitionTime: "2021-08-12T11:10:19Z"
      status: "True"
      type: PodScheduled
    containerStatuses:
    - containerID: docker://750d5c526eec60311092d80efc5aa25b1aff2360dec203f92de15c3df9ef534e
      image: harbor.shopeemobile.com/airpay/credit-risk-admin-id-test:v1.0.51.t09
      imageID: docker-pullable://harbor.shopeemobile.com/airpay/credit-risk-admin-id-test@sha256:
e70552e6422ab22743238f97789841fd4e62b0c0e4caaf77045b77cc0ffdbaa6
      lastState: {}
      name: risk-admin
      ready: true
      restartCount: 0
      state:
        running:
          startedAt: "2021-08-12T11:10:23Z"
    hostIP: 10.128.87.204
    phase: Running
    podIP: 100.95.10.36
    qosClass: Guaranteed
    startTime: "2021-08-12T11:10:19Z"
kind: List
metadata:
  resourceVersion: ""
  selfLink: ""
```

risk-admin-node pods yaml

```
apiVersion: v1
items:
- apiVersion: v1
  kind: Pod
  metadata:
    annotations:
      cni.projectcalico.org/podIP: 100.95.10.26/32
      kube-platform.shopee.io/application: risk-admin-node
      kube-platform.shopee.io/cid: id
      kube-platform.shopee.io/env: test
      kube-platform.shopee.io/group: SeaMoney Credit
      kube-platform.shopee.io/project: credit-risk-admin
      prometheus.io/scrape: "true"
    creationTimestamp: "2021-08-05T11:34:07Z"
    generateName: risk-admin-node-6cdd589fd9-
    labels:
      app: risk-admin-node
      application: risk-admin-node
      cid: id
      env: test
      group: SeaMoney-Credit
      metrics: "true"
      phase: release
      pod-template-hash: 6cdd589fd9
      project: credit-risk-admin
    name: risk-admin-node-6cdd589fd9-vd2tj
    namespace: credit-risk-admin-test-id
    ownerReferences:
    - apiVersion: apps/v1
      blockOwnerDeletion: true
      controller: true
      kind: ReplicaSet
      name: risk-admin-node-6cdd589fd9
```

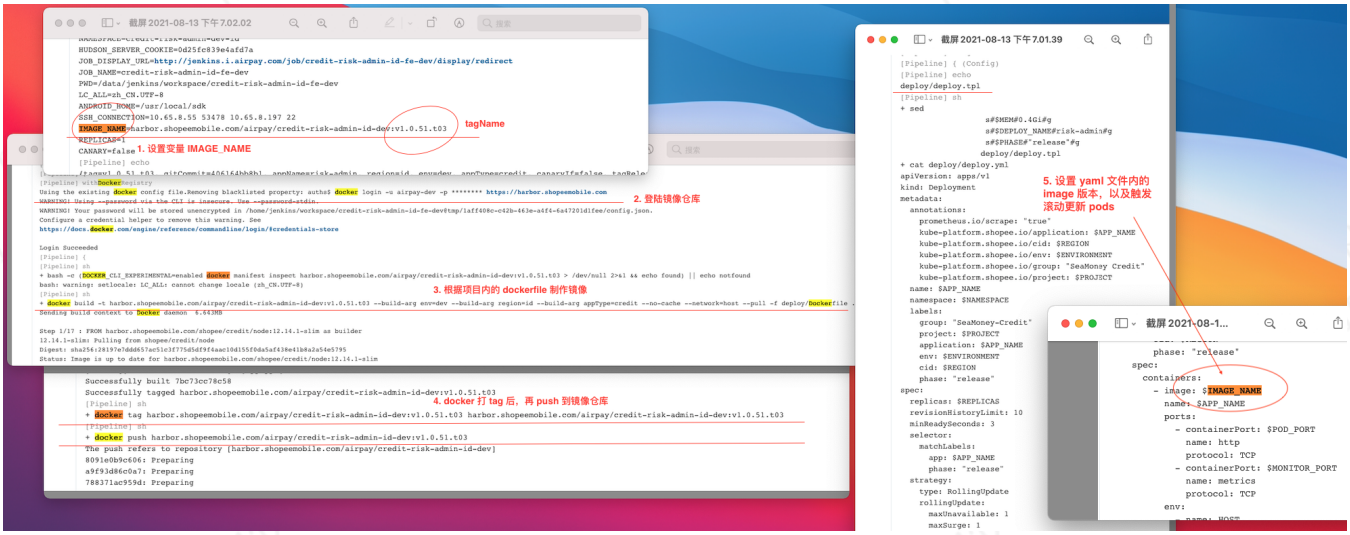
```
    uid: 0b91747e-594e-4b51-ae51-d0055a65fd55
  resourceVersion: "230900248"
  selfLink: /api/v1/namespaces/credit-risk-admin-test-id/pods/risk-admin-node-6cdd589fd9-vd2tj
  uid: 00b0cae2-5f48-4ba4-a3b1-41de58007739
spec:
  containers:
  - env:
    - name: HOST
      value: holmes.test.shopeekredit.co.id
    - name: CPU
      value: "2"
    - name: MEM
      value: 4Gi
    - name: ENV
      value: test
    - name: REGION
      value: id
    - name: POD_PORT
      value: "3001"
    - name: REDIS_HOSTS
      value: $REDIS_HOSTS
    - name: ETCD_HOSTS
      value: $ETCD_HOSTS
    - name: APP_NAME
      value: risk-admin-node
    - name: APP_TYPE
      value: credit
    - name: MONITOR_PORT
      value: "30009"
    - name: POD_NAME
      valueFrom:
        fieldRef:
          apiVersion: v1
          fieldPath: metadata.name
    - name: HOST_IP
      valueFrom:
        fieldRef:
          apiVersion: v1
          fieldPath: status.hostIP
    image: harbor.shopeemobile.com/airpay/credit-risk-admin-node-id-test:v1.0.51.t04
    imagePullPolicy: IfNotPresent
    lifecycle:
      preStop:
        exec:
          command:
          - sh
          - -c
          - sleep 5
    livenessProbe:
      failureThreshold: 3
      initialDelaySeconds: 15
      periodSeconds: 20
      successThreshold: 1
      tcpSocket:
        port: 3001
      timeoutSeconds: 1
    name: risk-admin-node
    ports:
    - containerPort: 3001
      name: http
      protocol: TCP
    - containerPort: 30009
      name: metrics
      protocol: TCP
    readinessProbe:
      failureThreshold: 3
      initialDelaySeconds: 5
      periodSeconds: 10
      successThreshold: 1
      tcpSocket:
        port: 3001
```

```
      timeoutSeconds: 1
    resources:
      limits:
        cpu: "2"
        memory: 4Gi
      requests:
        cpu: "2"
        memory: 4Gi
    terminationMessagePath: /dev/termination-log
    terminationMessagePolicy: File
    volumeMounts:
    - mountPath: /data/server/logs/
      name: node-logs
    - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
      name: default-token-h78zs
      readOnly: true
  dnsPolicy: ClusterFirst
  enableServiceLinks: true
  nodeName: sg2-shopee-kredit-kubernetes-test-10-128-87-204
  priority: 0
  restartPolicy: Always
  schedulerName: default-scheduler
  securityContext: {}
  serviceAccount: default
  serviceAccountName: default
  terminationGracePeriodSeconds: 30
  tolerations:
  - effect: NoExecute
    key: node.kubernetes.io/not-ready
    operator: Exists
    tolerationSeconds: 300
  - effect: NoExecute
    key: node.kubernetes.io/unreachable
    operator: Exists
    tolerationSeconds: 300
  volumes:
  - hostPath:
      path: /data/log/credit-risk-admin-node-test-id
      type: ""
    name: node-logs
  - name: default-token-h78zs
    secret:
      defaultMode: 420
      secretName: default-token-h78zs
status:
  conditions:
  - lastProbeTime: null
    lastTransitionTime: "2021-08-05T11:34:07Z"
    status: "True"
    type: Initialized
  - lastProbeTime: null
    lastTransitionTime: "2021-08-05T11:34:37Z"
    status: "True"
    type: Ready
  - lastProbeTime: null
    lastTransitionTime: "2021-08-05T11:34:37Z"
    status: "True"
    type: ContainersReady
  - lastProbeTime: null
    lastTransitionTime: "2021-08-05T11:34:07Z"
    status: "True"
    type: PodScheduled
  containerStatuses:
  - containerID: docker://7cd7c70edc316799c5b679662dd96ec34e4ac3c870f9433987220e47912b6dd2
    image: harbor.shopeemobile.com/airpay/credit-risk-admin-node-id-test:v1.0.51.t04
    imageID: docker-pullable://harbor.shopeemobile.com/airpay/credit-risk-admin-node-id-test@sha256:
4c5dd3f091a951641f2ed76d34810ba93286b6fca34f765339eb595523ccbb46
    lastState: {}
    name: risk-admin-node
    ready: true
    restartCount: 0
```

```
      state:
        running:
          startedAt: "2021-08-05T11:34:23Z"
    hostIP: 10.128.87.204
    phase: Running
    podIP: 100.95.10.26
    qosClass: Guaranteed
    startTime: "2021-08-05T11:34:07Z"
kind: List
metadata:
  resourceVersion: ""
  selfLink: ""
```

- **6. Jenkins 根据 git 仓库打的 tag，制作对应 tag 版本的镜像包，推送到 镜像仓库【https://harbor.shopeemobile.com】（运维内部使用，不开放给外部），推送完后，再触发 k8s pods 滚动更新新的 image 版本，具体可在 Jenkins 平台查看对应 log:**

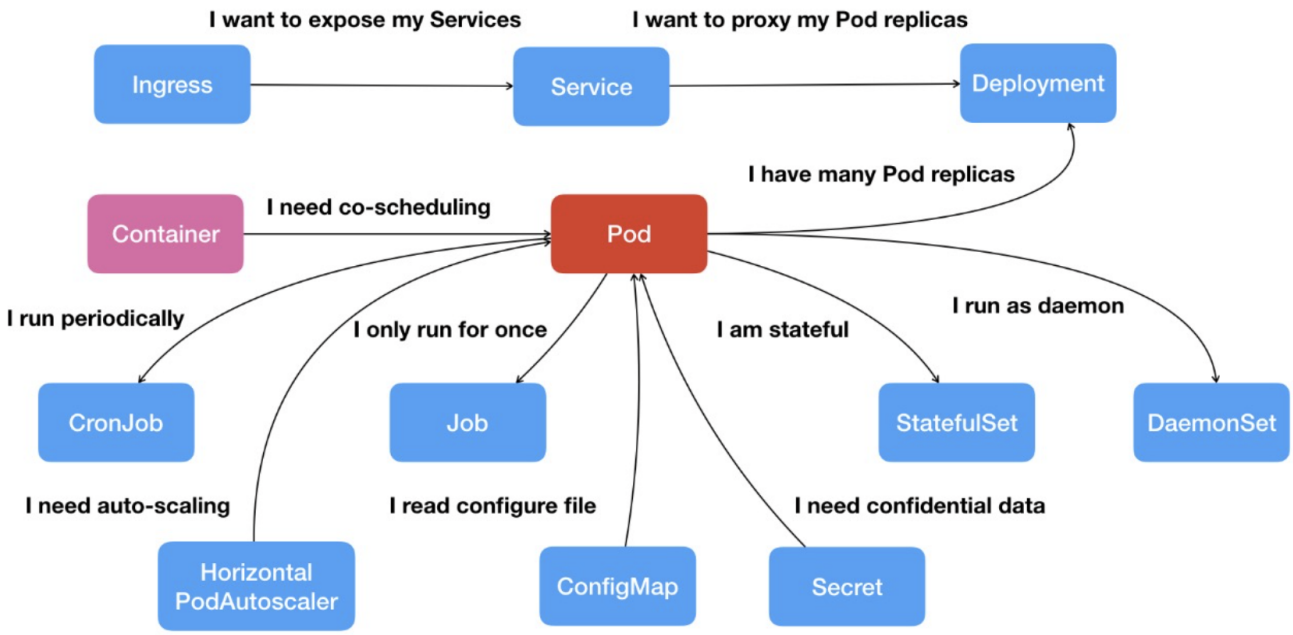
随便找个部署 log 举个例子（https://jenkins.i.airpay.com/job/credit-risk-admin-id-fe-dev/lastBuild/console）：



具体执行流水线步骤（https://jenkins.i.airpay.com/job/credit-risk-admin-id-fe-dev/lastBuild/flowGraphTable/），也可查看对应 GIT 仓库流水线代码（https://git.garena.com/shopee/sz-devops/finance-sre/credit-pipeline-shared-lib-fe/-/tree/credit）

相关资料（搭配理解效果更佳）：

- **K8S 总览图**

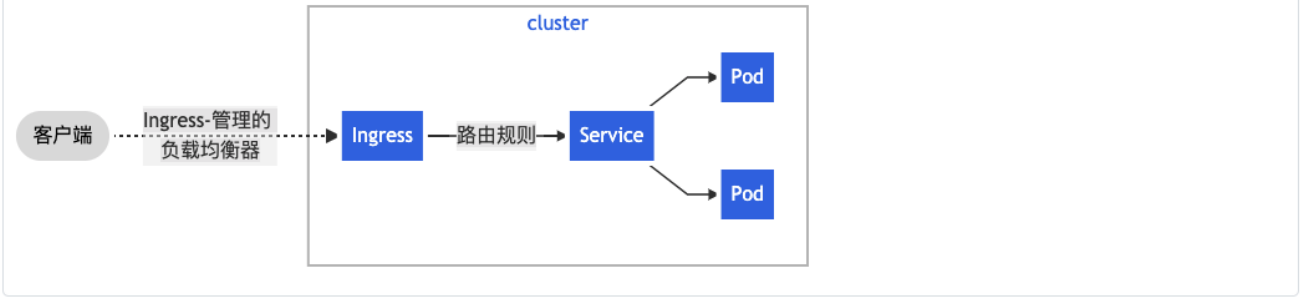


- **什么是 Ingress？**（https://kubernetes.io/zh/docs/concepts/services-networking/ingress/#ingress-%E6%98%AF%E4%BB%80%E4%B9%88）

Ingress 是什么?

Ingress 公开了从集群外部到集群内服务的 HTTP 和 HTTPS 路由。 流量路由由 Ingress 资源上定义的规则控制。

下面是一个将所有流量都发送到同一 Service 的简单 Ingress 示例：



可以将 Ingress 配置为服务提供外部可访问的 URL、负载均衡流量、终止 SSL/TLS，以及提供基于名称的虚拟主机等能力。 Ingress 控制器 通常负责通过负载均衡器来实现 Ingress，尽管它也可以配置边缘路由器或其他前端来帮助处理流量。

Ingress 不会公开任意端口或协议。 将 HTTP 和 HTTPS 以外的服务公开到 Internet 时，通常使用 Service.Type=NodePort 或 Service.Type=LoadBalancer 类型的服务。

- **为什么需要 Service？**（https://kubernetes.io/zh/docs/concepts/services-networking/connect-applications-service/#%E5%88%9B%E5%BB%BA-service）

创建 Service

我们有 Pod 在一个扁平的、集群范围的地址空间中运行 Nginx 服务，可以直接连接到这些 Pod，但如果某个节点死掉了会发生什么呢？ Pod 会终止，Deployment 将创建新的 Pod，且使用不同的 IP。这正是 Service 要解决的问题。

Kubernetes Service 从逻辑上定义了运行在集群中的一组 Pod，这些 Pod 提供了相同的功能。 当每个 Service 创建时，会被分配一个唯一的 IP 地址（也称为 clusterIP）。 这个 IP 地址与一个 Service 的生命周期绑定在一起，当 Service 存在的时候它也不会改变。 可以配置 Pod 使它与 Service 进行通信，Pod 知道与 Service 通信将被自动地负载均衡到该 Service 中的某些 Pod 上。

- **Pod 是什么？**（https://kubernetes.io/zh/docs/concepts/workloads/ ）

工作负载

工作负载是在 Kubernetes 上运行的应用程序。
无论你的负载是单一组件还是由多个一同工作的组件构成，在 Kubernetes 中你 可以在一组 Pods 中运行它。 在 Kubernetes 中，Pod 代表的是集群上处于运行状态的一组 容器。

Kubernetes Pods 有确定的生命周期。 例如，当某 Pod 在你的集群中运行时，Pod 运行所在的 节点 出现致命错误时， 所有该节点上的 Pods 都会失败。Kubernetes 将这类失败视为最终状态： 即使该节点后来恢复正常运行，你也需要创建新的 Pod 来恢复应用。

不过，为了让用户的日子略微好过一些，你并不需要直接管理每个 Pod。 相反，你可以使用 *负载资源* 来替你管理一组 Pods。 这些资源配置 控制器 来确保合适类型的、处于运行状态的 Pod 个数是正确的，与你所指定的状态相一致。

Kubernetes 提供若干种内置的工作负载资源：

目前咋门的服务用的是 deployment，可以查看对应的 pod yaml 文件定义

- Deployment 和 ReplicaSet （替换原来的资源 ReplicationController）。 `Deployment` 很适合用来管理你的集群上的无状态应用，`Deployment` 中的所有 `Pod` 都是相互等价的，并且在需要的时候被换掉。
- StatefulSet 让你能够运行一个或者多个以某种方式跟踪应用状态的 Pods。 例如，如果你的负载会将数据作持久存储，你可以运行一个 `StatefulSet`，将每个 `Pod` 与某个 `PersistentVolume` 对应起来。你在 `StatefulSet` 中各个 `Pod` 内运行的代码可以将数据复制到同一 `StatefulSet` 中的其它 `Pod` 中以提高整体的服务可靠性。
- DaemonSet 定义提供节点本地支撑设施的 `Pods`。这些 Pods 可能对于你的集群的运维是 非常重要的，例如作为网络链接的辅助工具或者作为网络 插件 的一部分等等。每次你向集群中添加一个新节点时，如果该节点与某 `DaemonSet` 的规约匹配，则控制面会为该 `DaemonSet` 调度一个 `Pod` 到该新节点上运行。
- Job 和 CronJob。 定义一些一直运行到结束并停止的任务。`Job` 用来表达的是一次性的任务，而 `CronJob` 会根据其时间规划反复运行。

在庞大的 Kubernetes 生态系统中，你还可以找到一些提供额外操作的第三方 工作负载资源。通过使用 定制资源定义（CRD）， 你可以添加第三方工作负载资源，以完成原本不是 Kubernetes 核心功能的工作。 例如，如果你希望运行一组 `Pods`，但要求所有 Pods 都可用时才执行操作 （比如针对某种高吞吐量的分布式任务），你可以实现一个能够满足这一需求 的扩展，并将其安装到集群中运行。