



最终汇报

董云鹏 (组长) 张子谦 冯逸飞

2023.06.02

CONTENTS



技术栈与软件开发



项目架构与组件功能



功能视频展示



QA



1

技术栈与软件开发



1.1 选用技术栈

开发语言: Go

github.com/fatih/color minik8s的分级日志系统
github.com/klauspost/pgzip 用户文件的zip压缩
github.com/gin-gonic/gin APIServer框架

容器运行时: docker

github.com/mholt/archiver Docker
镜像打包时用到的tar压缩

分布式存储: ETCD

go.etcd.io/etcd/client/v3 和Etcd存储交互操作的客户端

持续性集成: CI/CD

测试: 单元测试: go-test

[docker/login-action](https://github.com/docker/login-action) CICD自动推送镜像到dockerHub
[docker/setup-qemu-action](https://github.com/docker/setup-qemu-action) CICD交叉编译平台
[gotest.tools/v3](https://github.com/gotest.tools/v3) 项目测试框架


支持应用

github.com/melbahja/goph GPU Job的SSH的客户端
github.com/melbahja/goph GPU Job的SSH的客户端
github.com/pallets/flask Serveless容器内的运行的程序

命令行

gopkg.in/yaml.v3 go的yaml文件解析
[spf13/cobra](https://github.com/spf13/cobra) 命令行解析
github.com/spf13/cobra Kubectl的命令行工具
github.com/jedib0t/go-pretty/table Kubectl美化

1.2 CI/CD git actions进行持续性集成



[Pull requests](#) [Issues](#) [Codespaces](#) [Marketplace](#) [Explore](#)

Musicminion / minik8s Private

[Watch](#) [Fork](#) [Starred](#)

[Code](#) [Issues](#) [Pull requests](#) [Discussions](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

All workflows

Showing runs from all workflows

469 workflow runs

	Event	Status	Branch	Actor
<div>Update API Server Readme</div> <div>minik8s #310: Commit d1c7734 pushed by Musicminion</div>	feature/deployment	4 days ago 20s		...
<div>Merge pull request #41 from Musicminion/development</div> <div>minik8s #71: Commit 60e6ff pushed by Musicminion</div>	master	4 days ago 1m 10s		...
<div>Merge pull request #41 from Musicminion/development</div> <div>minik8s #309: Commit 60e6ff pushed by Musicminion</div>	master	4 days ago 50s		...
<div>Merge pull request #41 from Musicminion/development</div> <div>image-build #16: Commit 90e6ff pushed by Musicminion</div>	master	4 days ago 2m 18s		...
<div>[2023/5/27] Merge Development into Master</div> <div>minik8s #70: Pull request #41 opened by Musicminion</div>	development	4 days ago 35s		...
<div>[2023/5/27] Merge Development into Master</div> <div>minik8s #308: Pull request #41 opened by Musicminion</div>	development	4 days ago 27s		...
<div>feat: update func, workflow node check, branch workflow</div> <div>minik8s #307: Commit 38cf431 pushed by every-breaking-wave</div>	feature/function	4 days ago 27s		...
<div>Merge branch 'feature/function' of github.com:Musicminion/minik8s into...</div> <div>minik8s #306: Commit e734308 pushed by every-breaking-wave</div>	feature/function	4 days ago 27s		...
<div>feat: kubectrl get for workflow</div> <div>minik8s #305: Commit 31c7a0f pushed by every-breaking-wave</div>	feature/function	4 days ago 29s		...
<div>refactor: remove redundant func of apiobject</div> <div>minik8s #304: Commit 9777110 pushed by every-breaking-wave</div>	feature/function	4 days ago 41s		...
<div>fix: parse result of medium result of workflow</div> <div>minik8s #303: Commit 1ea30cd pushed by every-breaking-wave</div>	feature/function	4 days ago 30s		...
<div>feat: add kubectrl execute</div> <div>minik8s #302: Commit 42f06e7 pushed by every-breaking-wave</div>	feature/function	4 days ago 36s		...
<div>fix: function pod scale bug, func request early return bug</div> <div>minik8s #301: Commit 7c71bd5 pushed by every-breaking-wave</div>	feature/function	4 days ago 27s		...
<div>fix: some bug</div> <div>minik8s #300: Commit 1573cd8 pushed by Musicminion</div>	feature/function	4 days ago 26s		...

Actions

New workflow

All workflows

image-build

minik8s

minik8s

minik8s

Management

Caches

1.3 版本控制

1 代码量较大

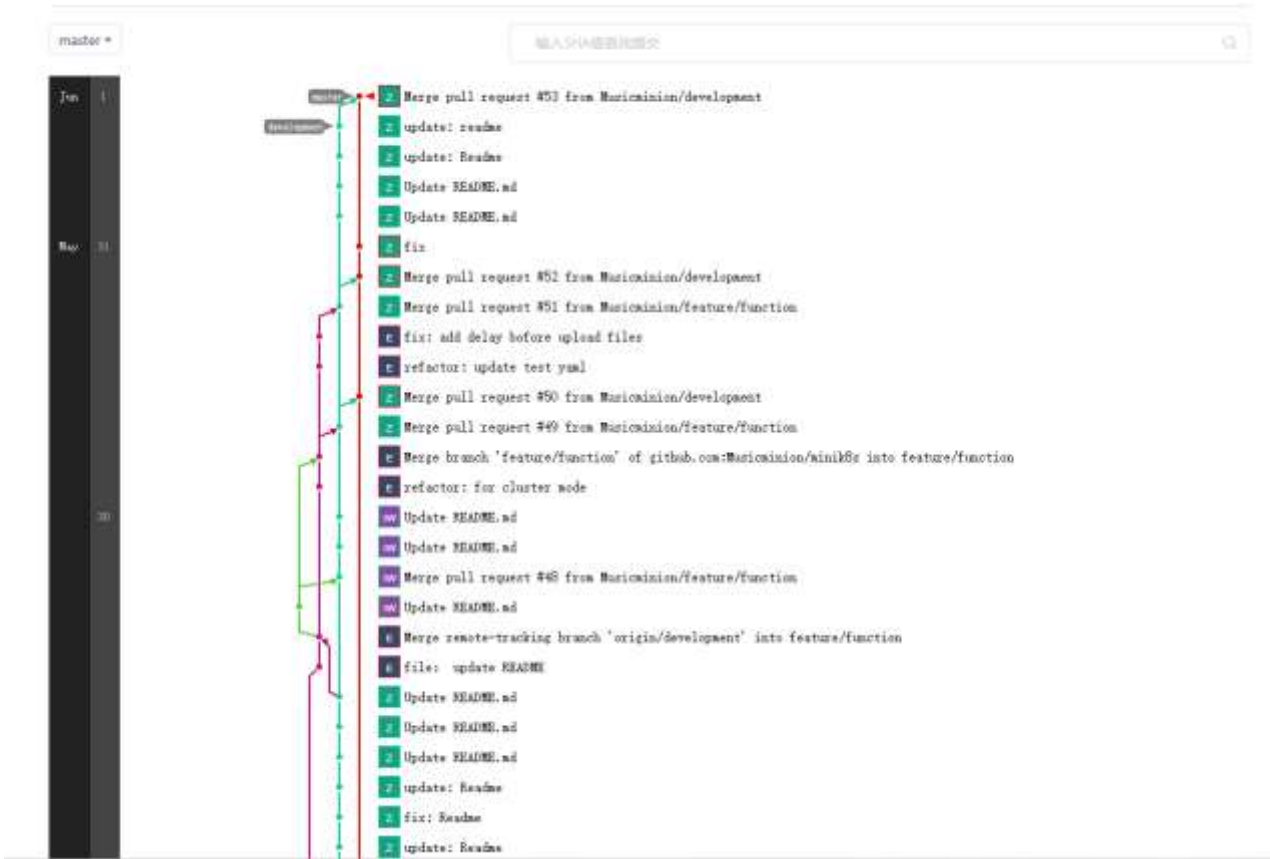
Languages

language	files	code	comment	blank	total
Go	172	19,287	0	4,063	23,350
Go Checksum File	1	426	0	1	427
Go Module File	1	102	0	4	106

2 每日审查与合并



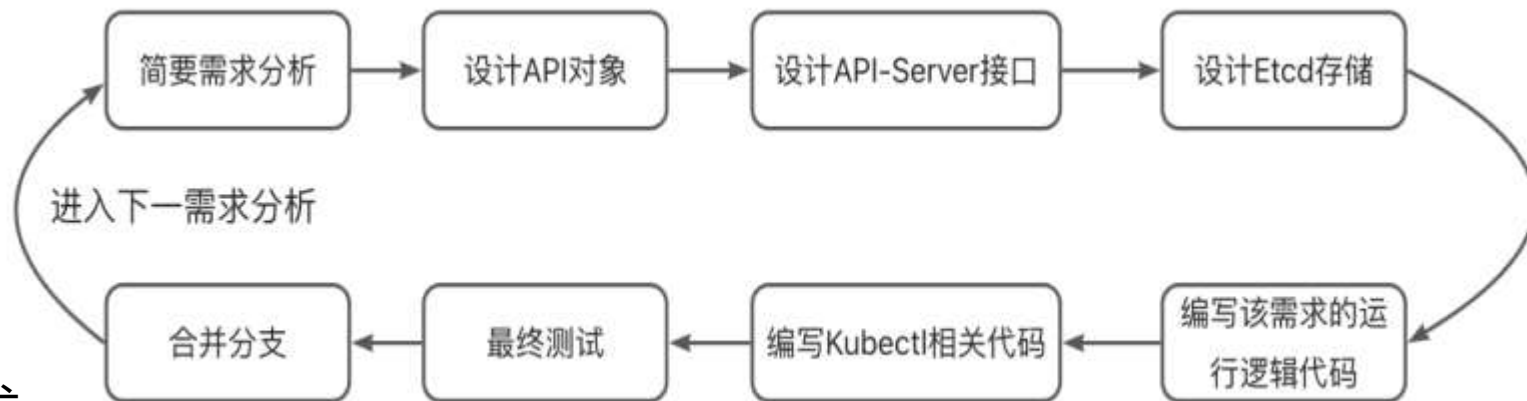
3 分支管理



1.4 Other Attempts

1

规范项目开发流程



•我们的软件开发基于迭代开发、敏捷开发。

•小组成员每天晚上在软件学院大楼实验室集中进行开发新功能，减少沟通障碍，做到有问题及时解决、沟通，有困难相互请教，这也大大的提高了我们小组的效率。截止15周周末，我们已经完成了所有的功能的开发。基本符合预期进度。

•对于新功能开发，我们采用"动态分配"方法，根据进度灵活分配成员的任务。项目框架搭建好之后，基本上在任何时间点小组同时在开发两个或者两个以上的需求。一人开发完成之后，交给另外一个组员完成代码的审查和测试，测试通过之后合并到Master

•功能开发的过程主要是：简要的需求分析->设计API对象->设计API-Server的接口->设计Etcd存储情况->编写该需求的运行逻辑代码->编写KubectI相关代码->最终测试

1.4 Other Attempts

2

规范PR提交

Filters		Labels 10		Milestones 0		New pull request	
[X] Clear current search query, filters, and sorts							
0 Open ✓ 53 Closed		Author		Label		Projects	
[2023/6/1] Merge Development into Master		bug-fix		Daily Merge		documentation	
#53 by Muscimoron was merged 10 hours ago							
[2023/5/29] Merge Development into Master							
#52 by Muscimoron was merged yesterday							
[2023/5/31] Merge daily work into Development							
#51 by Muscimoron was merged yesterday							
[2023/5/29] Merge Development into Master		bug-fix		Daily Merge		MasterUpdate	
#50 by Muscimoron was merged yesterday							
[2023/5/31] Merge daily work into Development		bug-fix		Daily Merge		DevUpdate	
#49 by Muscimoron was merged yesterday							
Feature/function							
#48 by every-breaking-wave was merged 2 days ago							
[2023/5/30] Merge Development into Master		bug-fix		Daily Merge		MasterUpdate	
#47 by Muscimoron was merged 3 days ago							
[2023/5/30] Merge daily work into Development		bug-fix		Daily Merge		DevUpdate	
#46 by Muscimoron was merged 3 days ago							
[2023/5/29] Merge Development into Master		bug-fix		Daily Merge		MasterUpdate	
#45 by Muscimoron was merged 4 days ago							
[2023/5/29] Merge daily work into Development		bug-fix		Daily Merge		DevUpdate	
#44 by Muscimoron was merged 4 days ago							
[2023/5/28] Merge Development into Master		Daily Merge		documentation		MasterUpdate	
#43 by Muscimoron was merged 4 days ago							
Update API Server Readme		documentation					
#42 by Muscimoron was merged 4 days ago							
[2023/5/27] Merge Development into Master		Daily Merge		MasterUpdate			
#41 by Muscimoron was merged 4 days ago							
[2023/5/28] Merge daily work into Development		Daily Merge		DevUpdate			
#40 by Muscimoron was merged 4 days ago							

1.4 Other Attempts

3

使用Git Actions

master分支收到推送之后,测试后会构建跟项目有关的image, 比如func的server, 并推送到dockerhub

The screenshot shows the GitHub Actions interface for a workflow named 'image-build' in the repository 'Musicminion / minik8s'. The workflow is currently in a 'fix #22' state. The left sidebar shows the workflow's structure, including a 'Summary' section and a 'Jobs' section with two jobs: 'build-and-push-1' and 'build-and-push-2'. The main area displays the details of the 'build-and-push-1' job, which succeeded yesterday in 7m 41s. The job steps are listed as follows:

- Set up job (3s)
- Checkout code (3s)
- Set up QEMU (7s)
- Set up Docker Buildx (0s)
- Login to Docker Registry (1s)
- Build and push Docker image (7m 26s)
- Post Build and push Docker image (0s)
- Post Login to Docker Registry (0s)
- Post Set up Docker Buildx (3s)
- Post Checkout code (0s)
- Complete job (0s)

1.5 接口文档

云OS

公共页面

K8s Restful API接口表格

虚拟机密码

1-设置环境并开始 demo

2-Standard Go Project Layout

3-工具参考

4-依赖库

5-测试yaml文件

6-container

7-containerd里面的 with 函数

8-docker 安装

9-API-Server

10-RabbitMQ消息队列

11-kubeproxy

12-CNI-weave

13-现有 Bug

最终答辩文档

第二次进度报告

常用文档链接

免费块已用 16%

K8s Restful API接口表格

共3人

K8s Restful API 接口表格

表格视图

API 接口表

资源类型	请求类型	URI 字段	URI	参数说明	描述	期望返回值
Node	GET	NodesURL	/api/v1/nodes	暂无	获取所有的 Node	200 OK
Node	POST	NodesURL	/api/v1/nodes	暂无	创建一个 Node	201 Created
Node	GET	NodeSpecURL	/api/v1/nodes/:name	name 是节点的名字, 必须	获取一个 Node 的信息	200 OK
Node	PUT	NodeSpecURL	/api/v1/nodes/:name	name 是节点的名字, 必须	更新一个 Node 的信息	200 OK
Node	DEL	NodeSpecURL	/api/v1/nodes/:name	name 是节点的名字, 必须	从集群删除 Node	204 DEL
Node	GET	NodeSpecStatusURL	/api/v1/nodes/:name/status	name 是节点的名字, 必须	获取 Node 的状态	200 OK
Node	PUT	NodeSpecStatusURL	/api/v1/nodes/:name/status	name 是节点的名字, 必须	更新 Node 的状态	200 OK
Pod	GET	PodsURL	/api/v1/namespaces/:namespace/pods	namespace 名字空间	获取所有的 Pod	200 OK
Pod	POST	PodsURL	/api/v1/namespaces/:namespace/pods	namespace 名字空间	创建一个 Pod	201 Created
Pod	GET	PodSpecURL	/api/v1/namespaces/:namespace/pods/:name	同上, name 是 pod 名字	获取某个特定的 Pod	200 OK
Pod	PUT	PodSpecURL	/api/v1/namespaces/:namespace/pods/:name	同上, name 是 pod 名字	更新某个特定的 Pod	200 OK
Pod	DEL	PodSpecURL	/api/v1/namespaces/:namespace/pods/:name	同上, name 是 pod 名字	删除某个 Pod	204 DEL
Pod	GET	PodSpecStatusURL	/api/v1/namespaces/:namespace/pods/:name/status	同上, name 是 pod 名字	获取某个 Pod 的状态	200 OK
Pod	POST	PodSpecStatusURL	/api/v1/namespaces/:namespace/pods/:name/status	同上, name 是 pod 名字	更新某个 Pod 的状态	200 OK
Service	POST	ServiceURL	/api/v1/namespaces/:namespace/services	namespace 名字空间	创建一个 Service	201 Create ?

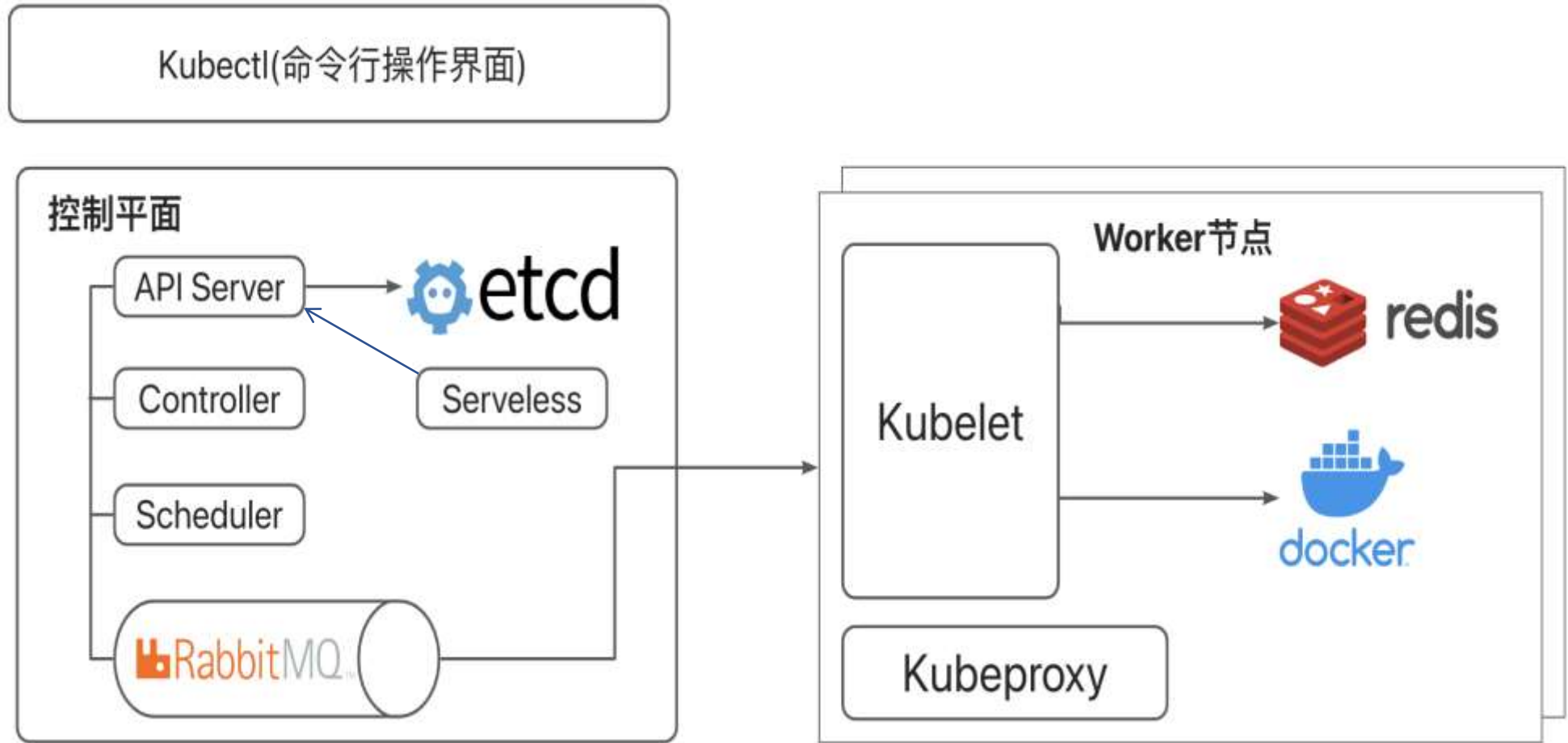


2

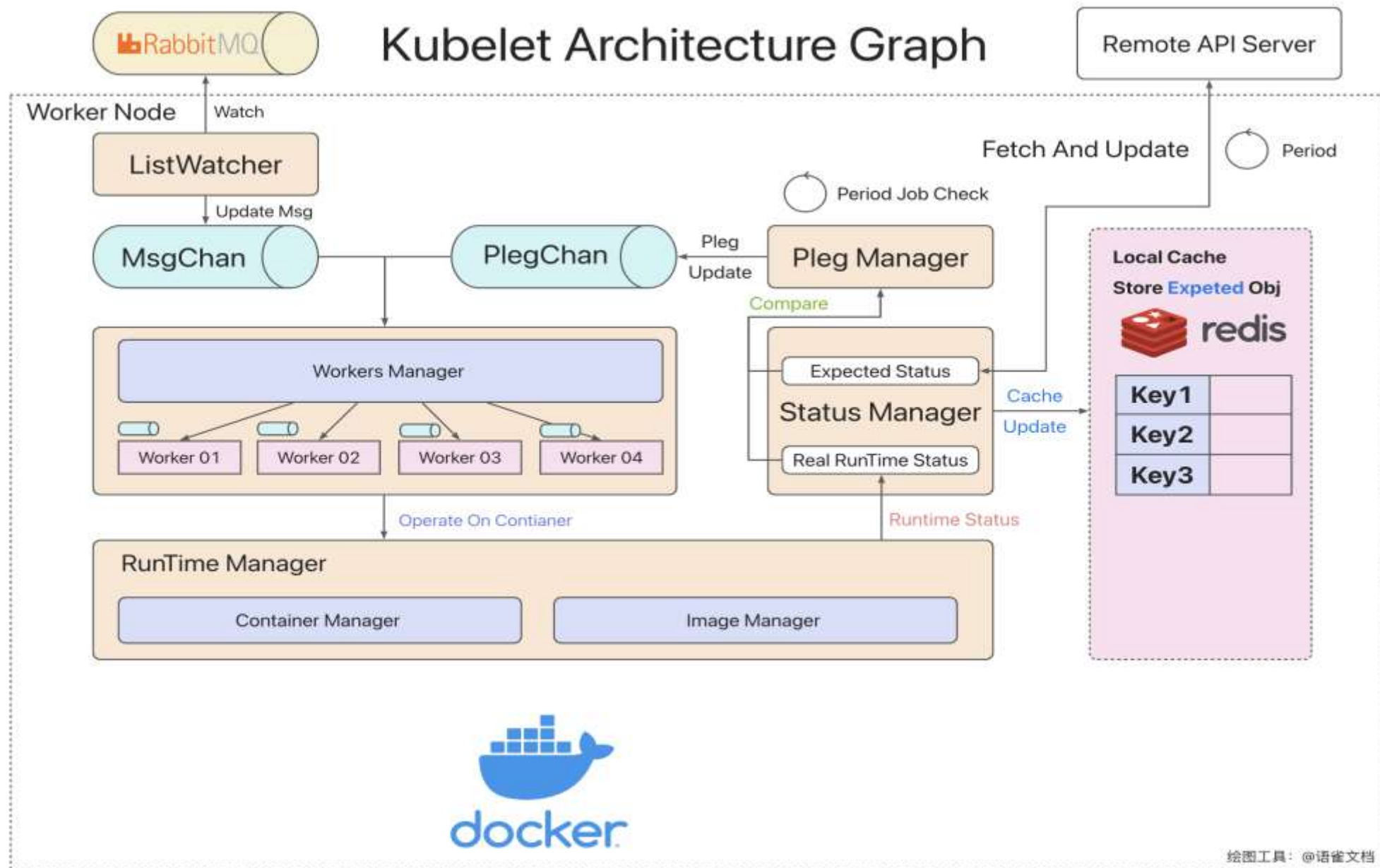
系统结构及组件功能



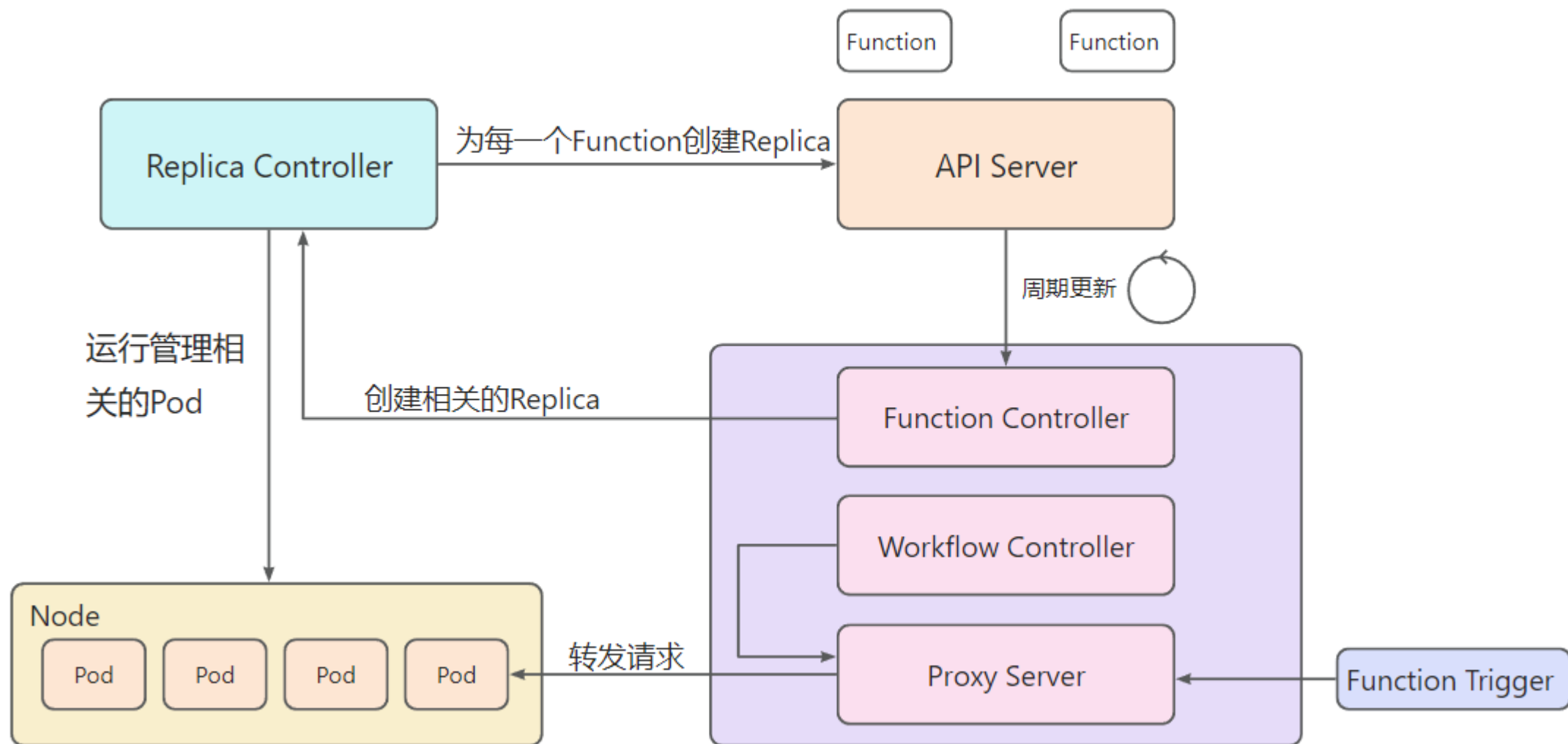
2.1 系统结构



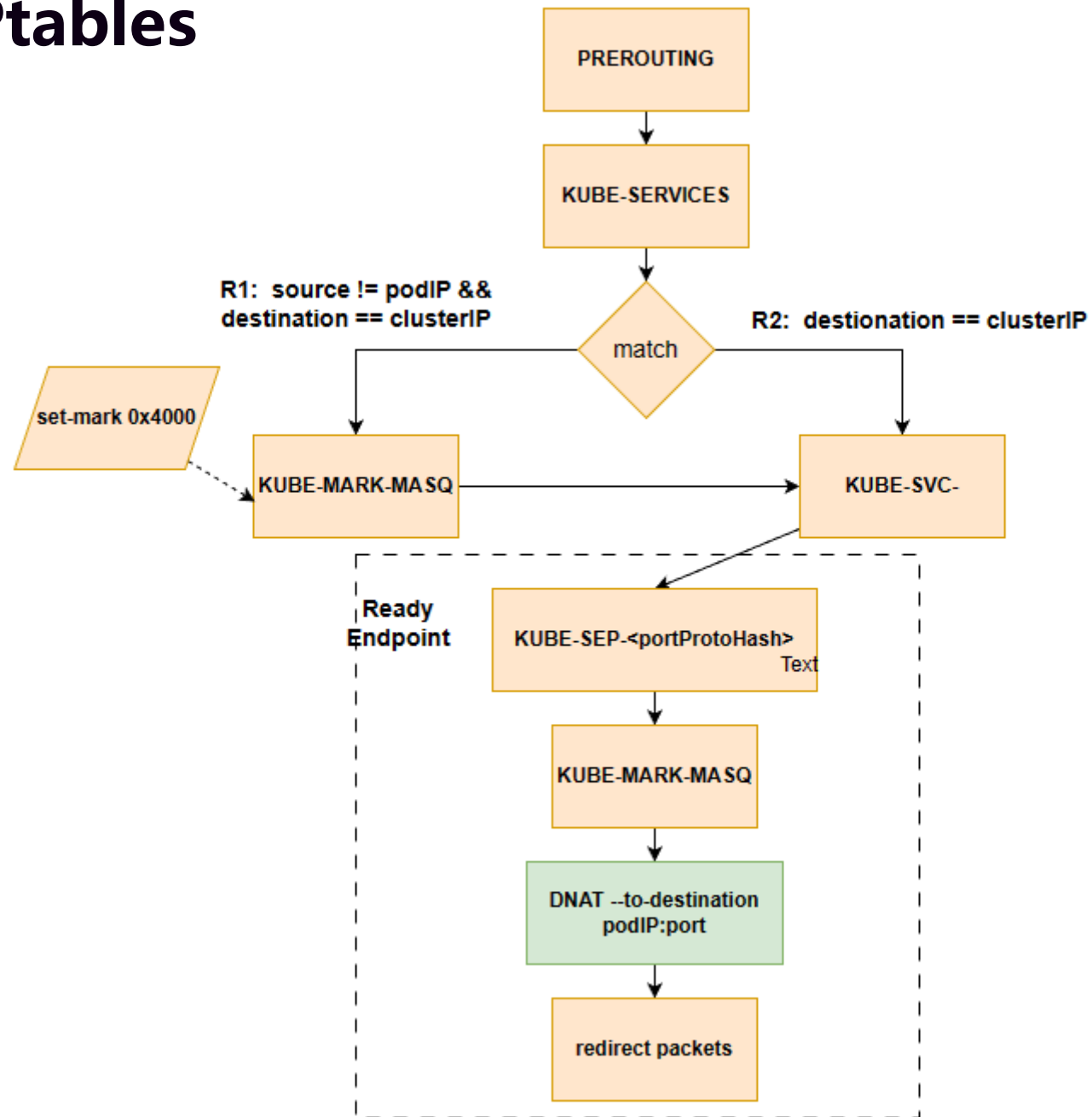
2.2 kubelet



2.2 Serverless 架构



2.3 KubeProxy IPtables





3

演示视频



3.1node

minik8s [SSH: cloud1]

getgo ApiObject.go X main.go kubelet/... apply.go channel.go workflow.go config kubletConfig.go node.go apiserverUtil.go

pkg > apiObject > ApiObject.go > KindToStructType

```
1 package apiObject
2
3 import (
4     "reflect"
5     "strings"
6 )
7
```

当前执行命令: kubectl get node

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
root@node-1:/home/minik8s# kubectl get node
+-----+-----+-----+-----+-----+
| KIND | NAME | STATUS | IP | CPU | MEMORY |
+-----+-----+-----+-----+-----+
root@node-1:/home/minik8s#
```

```
root@node-1:/home/minik8s#
```

bash minik8s

- bash minik8s
- node1-boot script
- boot minik8s
- node2 minik8s
- node2-boot minik8s
- apiserver main

SSH: cloud1 feature/function* Go 1.20.3 0 0 0

Ln 41, Col 56 Tab Size: 4 UTF-8 LF Go

3.2pod

Pod的配置yaml文件

```
testFile > ! pod-1.yaml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4  labels:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

CONTAINER ID	IMAGE	COMMAND	CREATED	ST
ATUS	PORTS	NAMES		
c06baf26b21c	nginx	"/docker-entrypoint...."	16 minutes ago	Up
16 minutes	nginx			
69565fc9478a	registry.aliyuncs.com/google_containers/pause:3.6	"/pause"	16 minutes ago	Up
16 minutes	127.0.0.1:39993->80/tcp	pause-08335c26-f8c7-46e2-927c-8b857f2b7be0		
6c9d3efc7854	registry:2.8.2	"/entrypoint.sh /etc...."	16 minutes ago	Up
16 minutes	0.0.0.0:5000->5000/tcp	minik8s-registry		
7b1b70b2a64b	weaveworks/weave:latest	"/home/weave/weaver ..."	20 hours ago	Up
20 hours	weave			
813cad81810e	weaveworks/weaveexec:latest	"data-only"	2 days ago	Cr
eated	weavevolumes-latest			
64d7fd640dff	weaveworks/weavedb:latest	"data-only"	2 days ago	Cr
eated	weavedb			

```
root@node-1:/home/minik8s# kubectl get pod
```

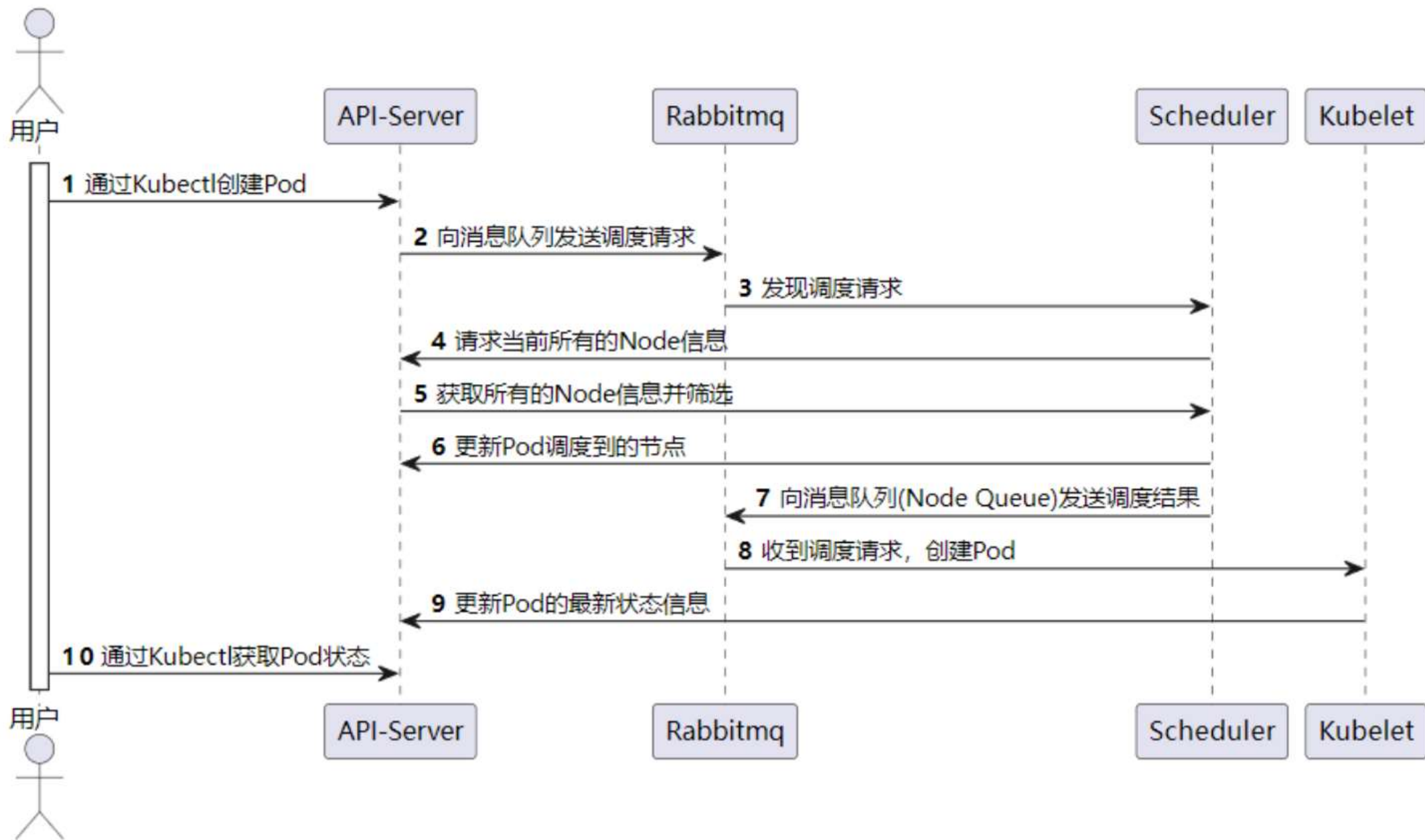
KIND	NAMESPACE	NAME	STATUS	IP	RUNTIME	NODE
pod	default	dns-nginx-pod-dK40D	Running	10.32.0.2	16m21s	node-1

```
root@node-1:/home/minik8s#
```

bash minik8s

- node1-boot script
- boot minik8s
- node2 minik8s
- node2-boot minik8s

3.2pod



3.2pod

minik8s [SSH: cloud1]

kubeproxy.log pod-for-nettest.yaml pod-1.yaml M pod-for-volume.yaml

testFile > ! pod-1.yaml

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   labels:
```

testFile > ! pod-for-volume.yaml

```
13
14 ports:
15   - containerPort: 10180
16   - image: docker.io/library/nginx
17   name: volumeContainer
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

KIND	NAMESPACE	NAME	STATUS	IP	RUNTIME	NODE
pod	default	dns-nginx-pod-EScsB	Pending		Not Created Yet	node-2
pod	default	dns-nginx-pod-diY4b	Running	10.32.0.2	27s	node-1

```
root@node-1:/home/minik8s# kubectl get pod
```

KIND	NAMESPACE	NAME	STATUS	IP	RUNTIME	NODE
pod	default	dns-nginx-pod-EScsB	Running	10.44.0.0	6s	node-2
pod	default	dns-nginx-pod-diY4b	Running	10.32.0.2	27s	node-1

```
root@node-1:/home/minik8s# kubectl get pod
```

KIND	NAMESPACE	NAME	STATUS	IP	RUNTIME	NODE
pod	default	dns-nginx-pod-EScsB	Running	10.44.0.0	7s	node-2
pod	default	dns-nginx-pod-diY4b	Running	10.32.0.2	28s	node-1

```
root@node-1:/home/minik8s#
```

root@node-1:/home/minik8s

nik8s#

bash minik8s

bash minik8s

node1-boot script

boot minik8s

node2 minik8s

node2-boot min...

SSH: cloud1 feature/function* Go 1.20.3 0 0 0 Ln 12, Col 13 Spaces: 2 UTF-8 LF YAML

3.3service

展示pod的yaml文件

minik8s [SSH: cloud1]

! pod-for-nettest.yaml ! pod-1.yaml M ! pod-2.yaml X ! service.yaml X

testFile > ! pod-2.yaml

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4 labels:
```

testFile > ! service.yaml

```
1 kind: Service
2 apiVersion: v1
3 metadata:
4 # Unique key of the Service instance
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

-> 192.168.1.7:6783 3e:81:8f:ac:83:e6(node-2) established

root@node-1:/home/minik8s# kubectl delete testFile/service.yaml
code: 204

KIND	RESULT	INFO	REASON(MSG)
Service	Success	delete obj success	

root@node-1:/home/minik8s# kubectl get service

KIND	NAMESPACE	NAME	CLUSTERIP
service	default	dns-nginx-service	192.168.174.20

NAMESPACE/NAME	CLUSTERIP	PORT	ENDPOINTIP/PORT	PROTOCOL
default/dns-nginx-service	192.168.174.20	80	10.32.0.2/80 10.44.0.0/80	tcp

root@node-1:/home/minik8s#

bash minik8s

- node1-boot script
- boot minik8s
- node2 minik8s
- node2-boot minik8s

SSH: cloud1 feature/function* Go 1.20.3 0 0 0 Ln 11, Col 20 Spaces: 2 UTF-8 LF YAML

3.4DNS

handler.gofunc.py! pod-for-hpa.yaml M! hpa.yamlcontainer.godns.go Mget.go M! dns.yaml M Xruntime.go

testFile > ! dns.yaml
1 apiVersion: v1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

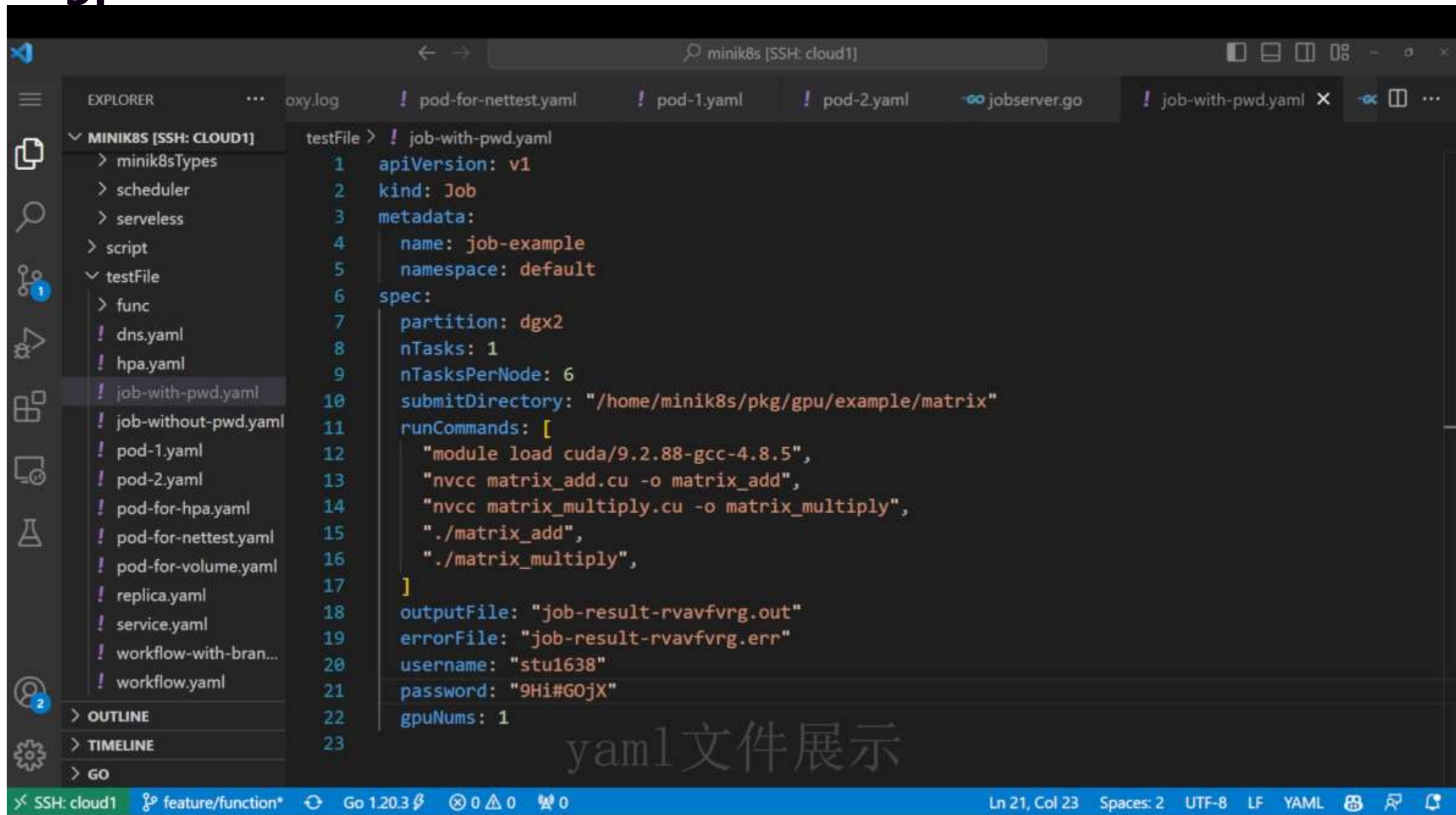
root@node-1:/home/minik8s#

node2-boot minik8s
node3-boot minik8s
boot minik8s
bash minik8s

SSH: cloud1feature/function*Go 1.20.30 0 0 0Ln 14, Col 1Spaces: 2UTF-8LFYAML15:002023/8/1

展示已有的Pod和Service可以可以访问

3.5gpu



minik8s [SSH: cloud1]

EXPLORER

- MINIK8S [SSH: CLOUD1]
 - minik8sTypes
 - scheduler
 - serveless
 - script
 - testFile
 - func
 - dns.yaml
 - hpa.yaml
 - job-with-pwd.yaml
 - job-without-pwd.yaml
 - pod-1.yaml
 - pod-2.yaml
 - pod-for-hpa.yaml
 - pod-for-nettest.yaml
 - pod-for-volume.yaml
 - replica.yaml
 - service.yaml
 - workflow-with-bran...
 - workflow.yaml
 - OUTLINE
 - TIMELINE
 - GO

testFile > ! job-with-pwd.yaml

```
1  apiVersion: v1
2  kind: Job
3  metadata:
4    name: job-example
5    namespace: default
6  spec:
7    partition: dgx2
8    nTasks: 1
9    nTasksPerNode: 6
10   submitDirectory: "/home/minik8s/pkg/gpu/example/matrix"
11   runCommands: [
12     "module load cuda/9.2.88-gcc-4.8.5",
13     "nvcc matrix_add.cu -o matrix_add",
14     "nvcc matrix_multiply.cu -o matrix_multiply",
15     "./matrix_add",
16     "./matrix_multiply",
17   ]
18   outputFile: "job-result-rvavfvrg.out"
19   errorFile: "job-result-rvavfvrg.err"
20   username: "stu1638"
21   password: "9Hi#60jX"
22   gpuNums: 1
23
```

yaml文件展示

SSH: cloud1 feature/function* Go 1.20.3 0 0 0 Ln 21, Col 23 Spaces: 2 UTF-8 LF YAML

3.5gpu

cudaError_t cudaMalloc (void devPtr, size_t size);**

CUDA 中的一个内存分配函数，用于分配 device 内存。

cudaMalloc 函数可以分配指定大小的连续内存块，并返回指向此内存块的指针给 devPtr。注意这里的 devPtr 应该是存放在 host 中的。

```
//allocate GPU mem
cudaMalloc((void **)&dev_A, sizeof(int *) * M);
cudaMalloc((void **)&dev_B, sizeof(int *) * M);
cudaMalloc((void **)&dev_C, sizeof(int *) * M);
```

C++

我们编写了简易的并行矩阵加法和乘法函数。

使用CUDA编程的情况下，我们首先定义了matrix_add和matrix_multiply两个使用__global__标记的核函数。

CUDA的风格类C，所以在处理矩阵这样的二维数组时需要两重指针，在进行host内存和device内存之间的数据转移，这里的cudaMemcpy函数作用是拷贝一段连续的内存，所以无法处理二重指针，需要辅助指针dev_A和dev_B，也就是说我们共需要四类指针，host上的二重和一重指针，device上的二重和一重指针。

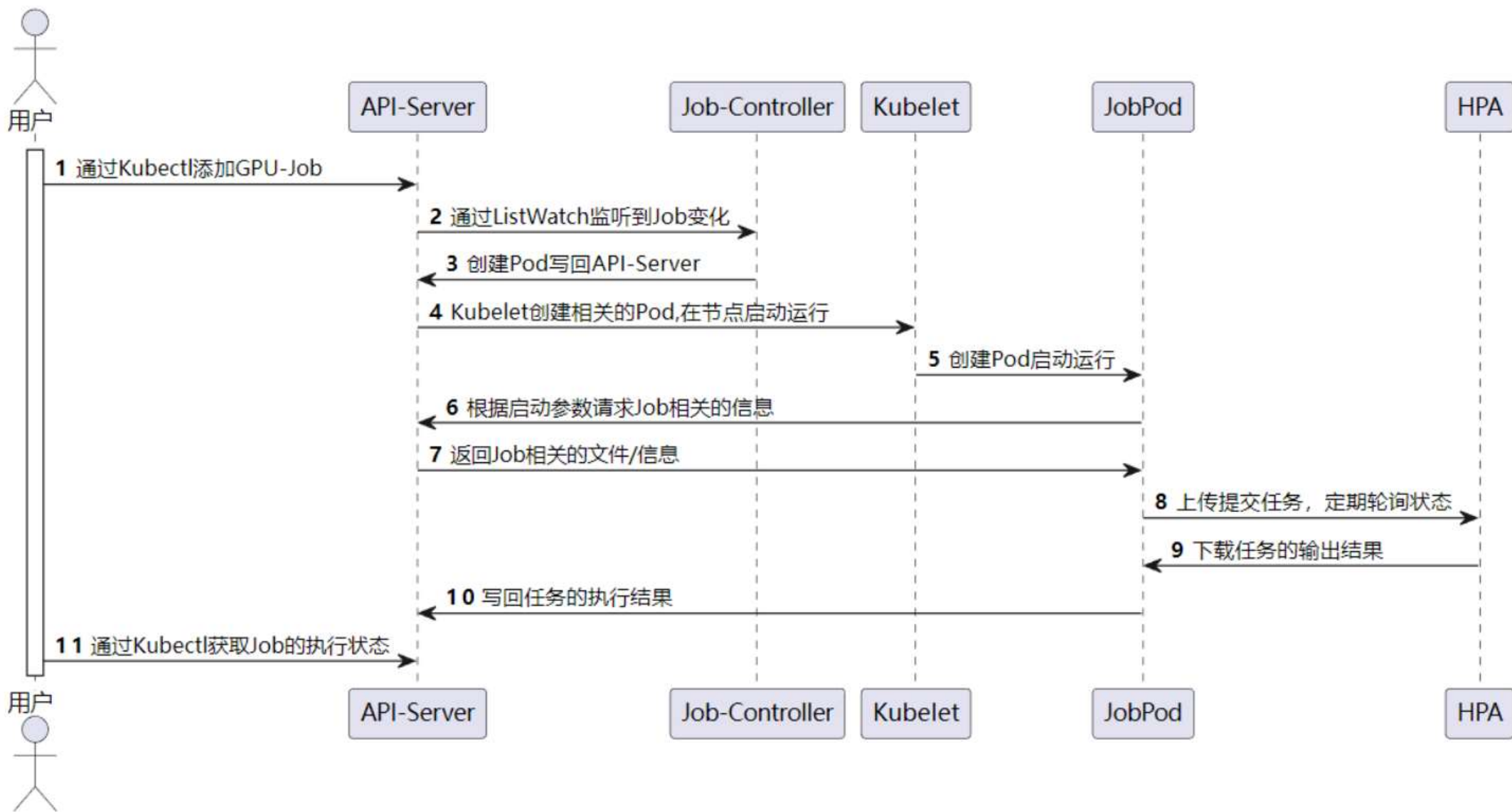
3.5gpu

```
问题 输出 测试结果 终端 调试控制台 窗口 GITLENS

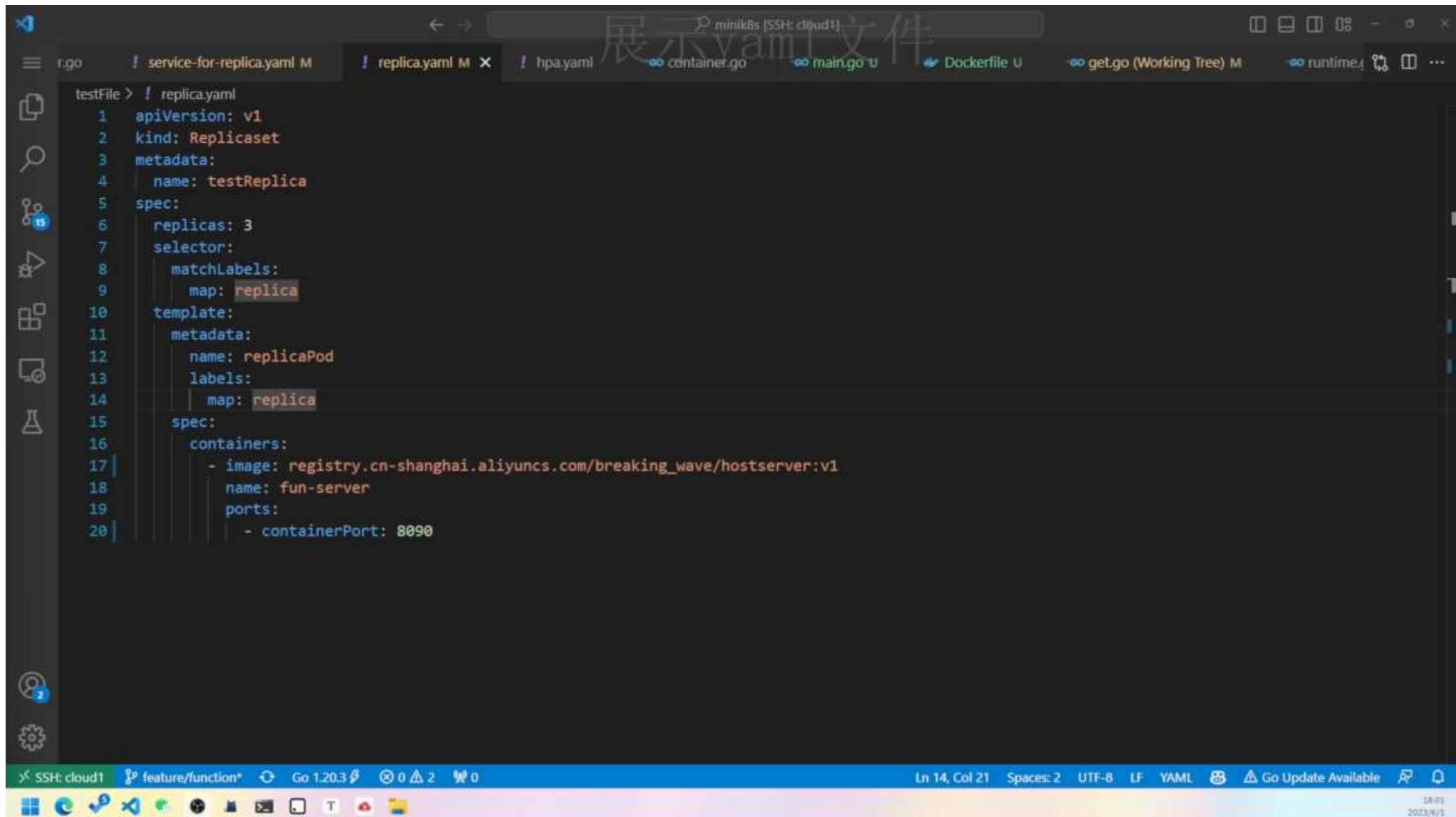
+-----+
| minik8s@ubuntu:~/minik8s/pkg/kubect1$ go run ./main/ get job test-job-namespace/job-example1 |
+-----+
| KIND | NAMESPACE | NAME | STATUS |
+-----+
| job | job-example1 | test-job-namespace | COMPLETED |
+-----+

+-----+
| OUTTYPE | NAMESPACE/NAME | CONTENT |
+-----+
| output | test-job-namespace/job-example1 | host上面的矩阵A:
| 0 1 2 3 4 5 6 7
| 8 9 10 11 12 13 14 15
| 16 17 18 19 20 21 22 23
| 24 25 26 27 28 29 30 31
| 32 33 34 35 36 37 38 39
| 40 41 42 43 44 45 46 47
| 48 49 50 51 52 53 54 55
| 56 57 58 59 60 61 62 63
| host上面的矩阵B:
| 0 1 2 3 4 5 6 7
| 8 9 10 11 12 13 14 15
| 16 17 18 19 20 21 22 23
| 24 25 26 27 28 29 30 31
| 32 33 34 35 36 37 38 39
| 40 41 42 43 44 45 46 47
| 48 49 50 51 52 53 54 55
| 56 57 58 59 60 61 62 63
| 矩阵加法的结果:
| 0 2 4 6 8 10 12 14
| 16 18 20 22 24 26 28 30
| 32 34 36 38 40 42 44 46
| 48 50 52 54 56 58 60 62
| 64 66 68 70 72 74 76 78
| 80 82 84 86 88 90 92 94
| 96 98 100 102 104 106 108 110
| 112 114 116 118 120 122 124 126
| host上面的矩阵A:
| 0 1 2 3 4 5 6 7
| 8 9 10 11 12 13 14 15
| 16 17 18 19 20 21 22 23
```

3.5gpu



3.6 replicaset



The screenshot shows a code editor with a dark theme. The main editor area displays a YAML file named `replica.yaml`. The file content is as follows:

```
1 apiVersion: v1
2 kind: Replicaset
3 metadata:
4   name: testReplica
5 spec:
6   replicas: 3
7   selector:
8     matchLabels:
9       map: replica
10  template:
11    metadata:
12      name: replicaPod
13      labels:
14        map: replica
15    spec:
16      containers:
17      - image: registry.cn-shanghai.aliyuncs.com/breaking_wave/hostserver:v1
18        name: fun-server
19        ports:
20        - containerPort: 8090
```

The editor's interface includes a sidebar on the left with icons for Explorer, Search, Source Control, Run and Debug, Extensions, Testing, and Settings. The top of the editor shows several open files: `r.go`, `! service-for-replica.yaml M`, `! replica.yaml M` (the active file), `! hpa.yaml`, `container.go`, `main.go U`, `Dockerfile U`, `get.go (Working Tree) M`, and `runtime.go`. The status bar at the bottom indicates the current file is `SSH: cloud1`, the editor is using `Go 1.20.3`, and the file is in `UTF-8` encoding with `LF` line endings. The cursor is positioned at line 14, column 21.

3.7auto scale

minik8s (SSH: cloud1)

! pod-for-hpa.yaml M X ! service-for-hpa.yaml U ! hpa.yaml Dockerfile .../autoscale U container.go main.go U Dockerfile .../server_echo_host U

testFile > ! pod-for-hpa.yaml

1 apiVersion: v1

2 kind: Pod

3 metadata:

4 name: pod-for-hpa

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

root@node-1:/home/minik8s#

root@node-1:/home/minik8s#

root@node-1:/home/minik8s#

root@node-1:/home/minik8s#

CONTAINER ID NAME CPU % MEM % NET I/O BLOCK I/O PIDS

4ec41fa0f591 weave 0.29% 26

.75MiB / 3.826GiB 0.68% 0B / 0B 21MB / 8.4MB 16

7afd80e82aa7 pause-845a202e-78e3-45c1-a780-8de9187a561d 0.00% 1.277MiB / 3.826GiB 0.03% 1.09kB / 0B 0B / 0B 1

f6ecf493501b minik8s-registry 0.00% 5.102MiB / 3.826GiB 0.13% 976B / 0B 0B / 0B 5

068e16667384 nginx 0.00% 3.629MiB / 3.826GiB 0.09% 1.09kB / 0B 160kB / 16.4kB 3

node2-boot...

node3-boot...

boot minik8s

bash minik8s

docker min...

bash minik8s

bash minik8s

SSH: cloud1 feature/function* Go 1.20.3 0 2 0

Ln 14, Col 30 (19 selected) Spaces: 2 UTF-8 LF YAML Go Update Available

17:43 2023/6/1

3.8serverless: function

The screenshot displays a VS Code editor window with a Go project for serverless functions. The Explorer on the left shows a directory structure with files like `func.py`, `func-1.yaml`, and `execute_function.sh`. The Terminal on the right shows multiple `kubectl get pod` commands and their output, indicating that pods are running on node-1 and node-2.

The Explorer shows the following files and directories:

- MINIK8S [SSH: CLOUD1]
 - .github
 - .VSCodeCounter
 - build
 - ce1ab139-2513...
 - cmd
 - doc
 - log
 - apiserver.log
 - controller.log
 - kubelet.log
 - kubeproxy.log
 - scheduler.log
 - serverless.log
 - object
 - pkg
 - apiObject
 - apiserver
 - config
 - controller
 - entity
 - etcd
 - gpu
 - k8log
 - kubectl
 - cmd
 - apply_test.go
 - apply.go
 - delete.go
 - describe.go
 - execute.go 1 M
 - get_test.go
 - OUTLINE
 - TIMELINE
 - GO

The Terminal shows the following commands and output:

```
root@node-1:/home/minik8s# kubectl get pod
KIND   NAME           STATUS    IP           RUNTIME   NODE
pod     dns-nginx-pod-c1Zb9   Running   10.32.0.2    16s       node-1

root@node-1:/home/minik8s# kubectl get pod
KIND   NAME           STATUS    IP           RUNTIME   NODE
pod     dns-nginx-pod-c1Zb9   Running   10.32.0.2    17s       node-1

root@node-1:/home/minik8s# kubectl get pod
KIND   NAME           STATUS    IP           RUNTIME   NODE
pod     dns-nginx-pod-c1Zb9   Running   10.32.0.2    17s       node-1

root@node-1:/home/minik8s# kubectl get pod
KIND   NAME           STATUS    IP           RUNTIME   NODE
pod     dns-nginx-pod-c1Zb9   Running   10.32.0.2    17s       node-1

root@node-1:/home/minik8s# kubectl get pod
KIND   NAME           STATUS    IP           RUNTIME   NODE
pod     dns-nginx-pod-H3EVO   Running   10.44.0.0    10s       node-2
pod     dns-nginx-pod-c1Zb9   Running   10.32.0.2    32s       node-1

root@node-1:/home/minik8s#
```


3.8serverless: workflow

The screenshot shows a VS Code editor with a workflow configuration file named `workflow-with-branch.yaml` open. The file is located in the `testFile` directory under the `func` subdirectory. The configuration defines a workflow with three nodes: `node1` (a function), `node2` (a choice), and `node3` (a function). The workflow is named `workflow-example` and is in the `default` namespace.

```
1 # Unique key of the Service instance
2 name: workflow-example
3 namespace: default
4
5 spec:
6   entryParams: '{"x": 1, "y": 2}'
7   entryNodeName: node1
8   workflowNodes:
9     - name: node1
10       type: func
11       funcData:
12         funcName: func2 # x = x + y, y = x - y
13         funcNamespace: default
14         nextNodeName: node2
15     - name: node2
16       type: choice
17       choiceData:
18         trueNextNodeName: node3
```

The terminal shows the output of the `kubectl` commands. The first command, `kubectl get workflow`, shows the workflow is in the `default` namespace and is in the `Running` phase. The second command, `kubectl get function`, shows three functions: `func1`, `func2`, and `func3`, all in the `default` namespace and in the `Running` phase.

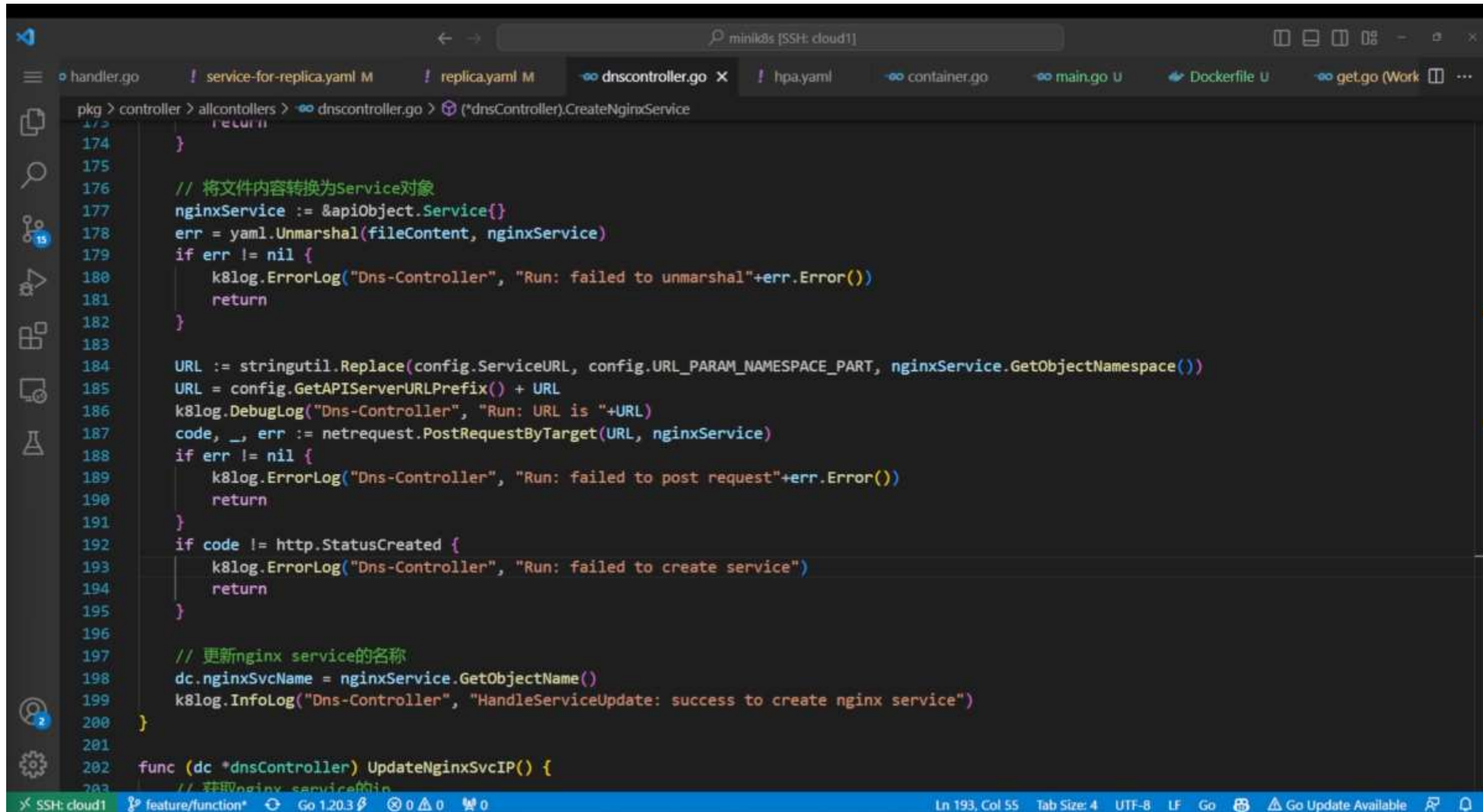
KIND	NAMESPACE	NAME	FILEPATH
function	default	func1	/home/minik8s/pkg/serveless/test-data/example-1
function	default	func2	/home/minik8s/pkg/serveless/test-data/example-2
function	default	func3	/home/minik8s/pkg/serveless/test-data/example-3

The terminal also shows the output of the `kubectl` commands. The first command, `kubectl get workflow`, shows the workflow is in the `default` namespace and is in the `Running` phase. The second command, `kubectl get function`, shows three functions: `func1`, `func2`, and `func3`, all in the `default` namespace and in the `Running` phase.

KIND	NAMESPACE	NAME	PHASE	RESULT
function	default	func1	Running	
function	default	func2	Running	
function	default	func3	Running	

The right sidebar shows the list of services running in the `minik8s` environment. The services are: `node2-boot`, `node3-boot`, `boot`, `bash`, `apiserver`, `kubelet`, `controller`, `scheduler`, `kubeproxy`, and `serverless`.

3.9 fault tolerance



```
pkg > controller > allcontrollers > dnscontroller.go > (*dnsController).CreateNginxService
173     return
174 }
175
176 // 将文件内容转换为Service对象
177 nginxService := &apiObject.Service{}
178 err = yaml.Unmarshal(fileContent, nginxService)
179 if err != nil {
180     k8log.ErrorLog("Dns-Controller", "Run: failed to unmarshal"+err.Error())
181     return
182 }
183
184 URL := stringutil.Replace(config.ServiceURL, config.URL_PARAM_NAMESPACE_PART, nginxService.GetObjectNamespace())
185 URL = config.GetAPIServerURLPrefix() + URL
186 k8log.DebugLog("Dns-Controller", "Run: URL is "+URL)
187 code, _, err := netrequest.PostRequestByTarget(URL, nginxService)
188 if err != nil {
189     k8log.ErrorLog("Dns-Controller", "Run: failed to post request"+err.Error())
190     return
191 }
192 if code != http.StatusCreated {
193     k8log.ErrorLog("Dns-Controller", "Run: failed to create service")
194     return
195 }
196
197 // 更新nginx service的名称
198 dc.nginxSvcName = nginxService.GetObjectName()
199 k8log.InfoLog("Dns-Controller", "HandleServiceUpdate: success to create nginx service")
200 }
201
202 func (dc *dnsController) UpdateNginxSvcIP() {
203     // 更新nginx service的ip
```



Thanks & QA

