

.NET Conf China

2022

# 基于YARP打造扩展性极强的云原生网关 Ingress Controller

肖伟宇

FireUG技术社区组织者之一



.NET Conf China

# 什么是YARP



# 什么是YARP (Yet Another Reverse Proxy)



A **toolkit** for developing **high-performance** HTTP reverse proxy applications.

YARP (Yet Another Reverse Proxy) is a **highly customizable** reverse proxy built using .NET.

The biggest differentiator between YARP and other reverse proxies is how it is built and packaged – YARP is supplied as a library and samples showing how to create a proxy that is customized to the needs of your specific scenarios.



# YARP的特性



基于 MIT 协议

基于新一代的跨平台的 .NET

提供了完善的文档和样例代码

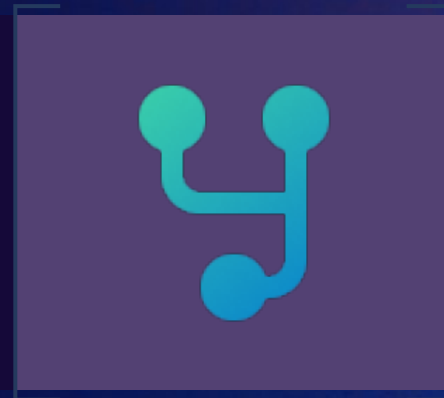
贡献代码 or 自行扩展

# 开源



# .NET Conf China

高性能



# 性能测试

## 主机配置

处理器 AMD Ryzen 7 3700X 8-Core Processor 3.60 GHz  
机带 RAM 32.0 GB  
系统类型 64 位操作系统, 基于 x64 的处理器

## 软件

Docker Desktop 4.14.1 (91661) WSL 2 based engine  
Kubeneretes v1.25.2  
Jmeter 5.5

## 测试说明

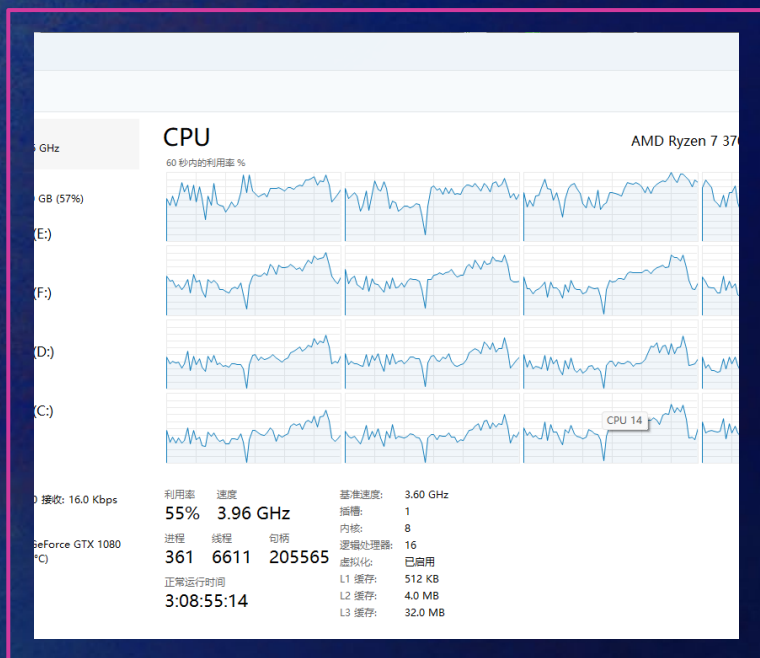
ASPNET Core 作为后端应用, 不限定资源  
用例: short-text (响应 “Hello World!” )、long-text (12.2kb)  
NetCorePal Ingress Controller 作为网关, 不同资源情况下测试  
后端应用、网关、压测工具运行在同一台主机

KUBERNETES

### CLUSTERS

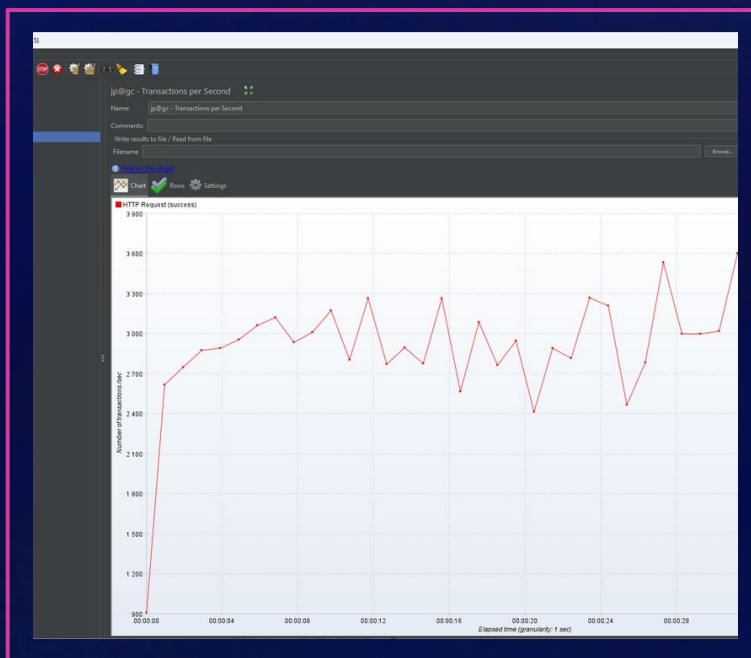
- docker-desktop
  - Namespaces
  - Nodes
  - Workloads
    - Deployments
      - backend-helm-charts-backend
        - backend-helm-charts-backend-6c95487d84-6nn5v
      - my-netcorepal-ingress-controller
        - my-netcorepal-ingress-controller-69d8f67bd-dvrkx
    - StatefulSets
    - DaemonSets
    - Jobs
    - CronJobs
    - Pods
  - Network
    - Services
    - Endpoints
    - Ingress
      - backend-helm-charts-backend
  - Storage
  - Configuration
  - Custom Resources
  - Helm Releases

# 限定资源 1C1G-short-text



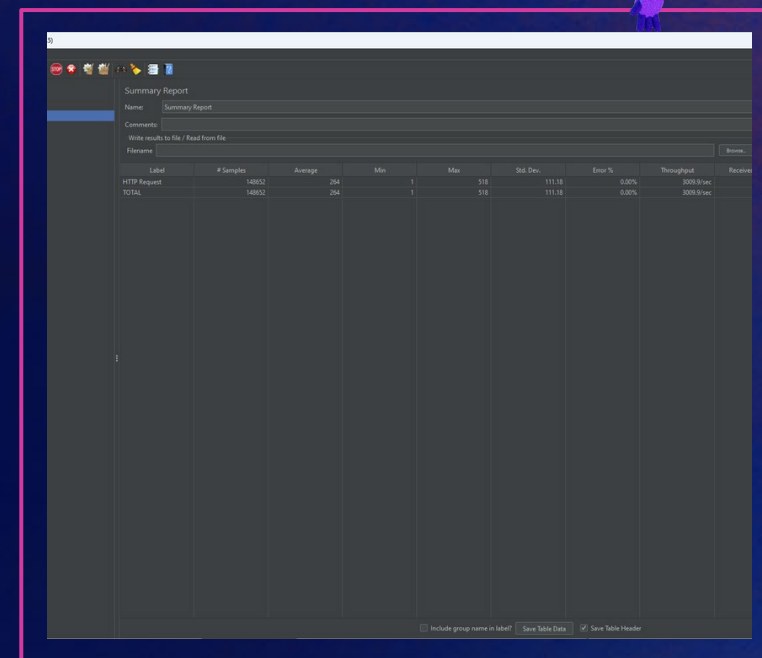
## CPU 使用率

55%



## TPS

峰值: 3600 Transactions per Second



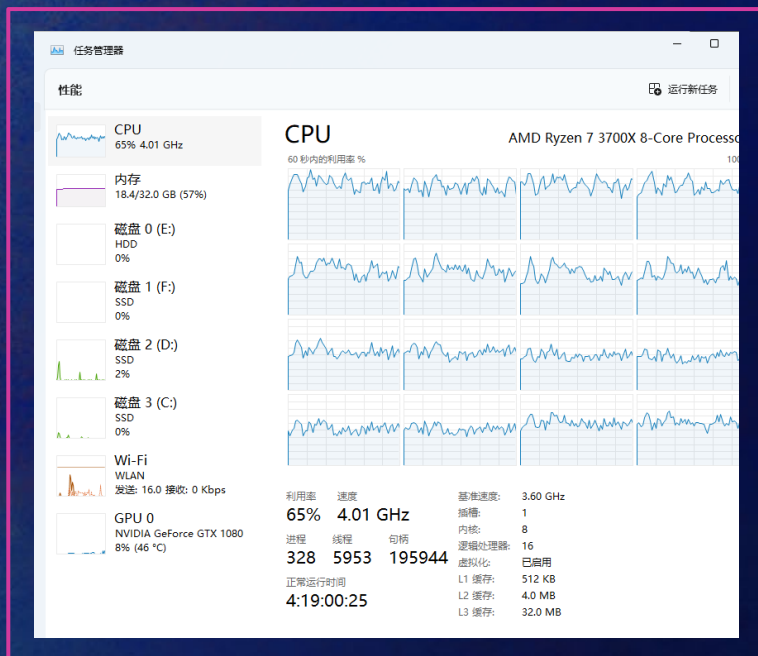
## 响应时间

Min: 1 ms  
Max: 518 ms  
Average: 264 ms



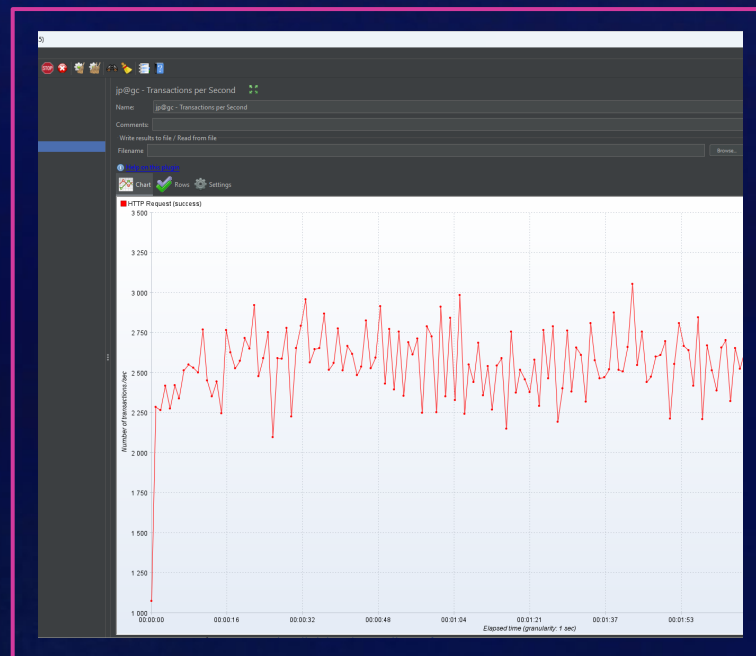


# 限定资源 1C1G-long-text



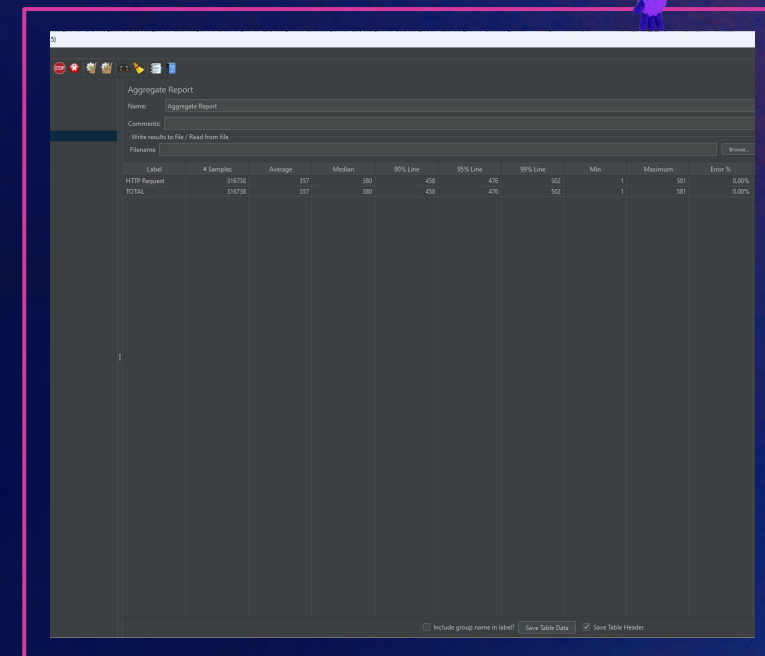
## CPU 使用率

65%



## TPS

峰值: 3050 Transactions per Second

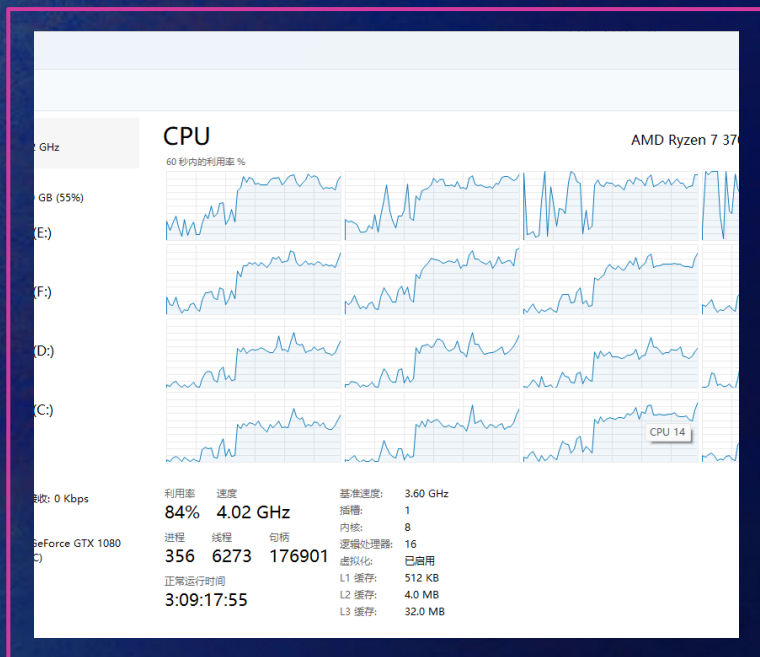


## 响应时间

Min: 1 ms  
Max: 581 ms  
Average: 357 ms

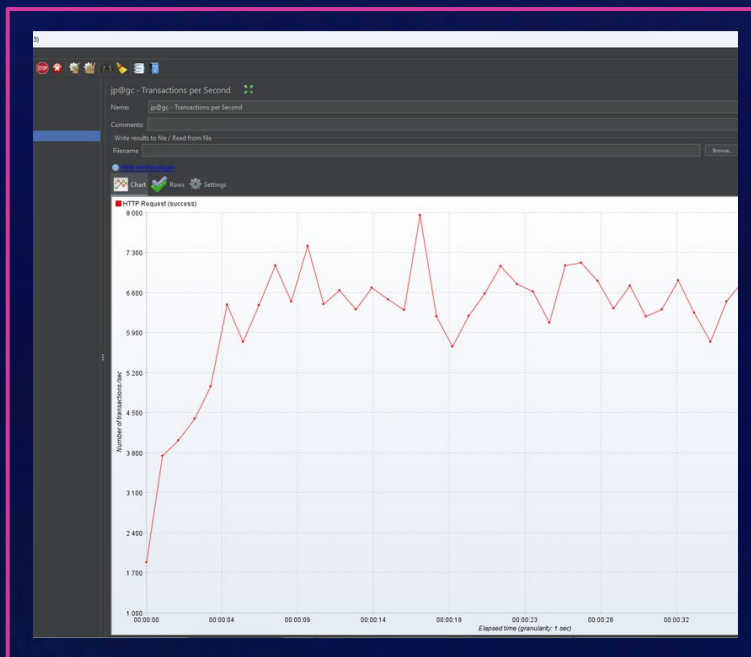


# 限制资源 2C1G-short-text



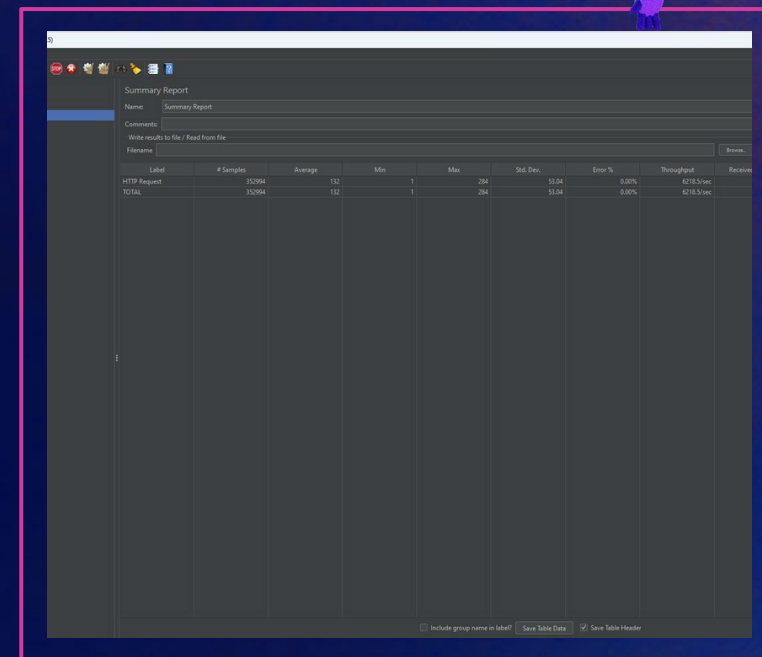
## CPU 使用率

84%



## TPS

峰值: 8000 Transactions per Second

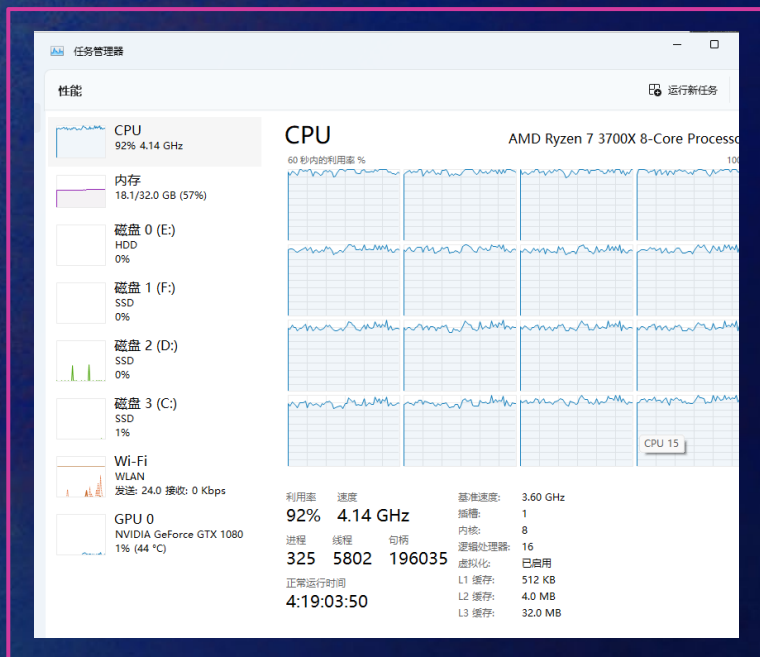


## 响应时间

Min: 1 ms  
Max: 284 ms  
Average: 132 ms



# 限制资源 2C1G-long-text



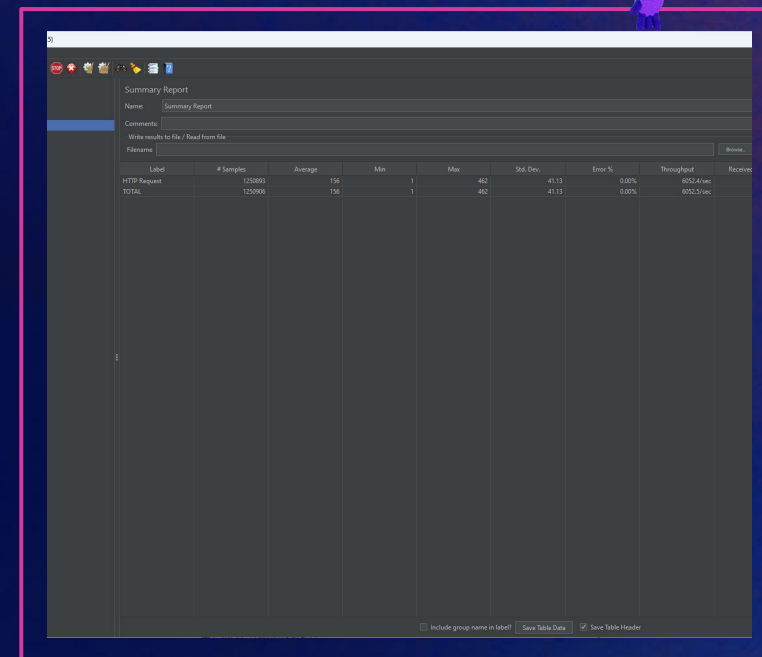
## CPU 使用率

92%



## TPS

峰值: 7600 Transactions per Second

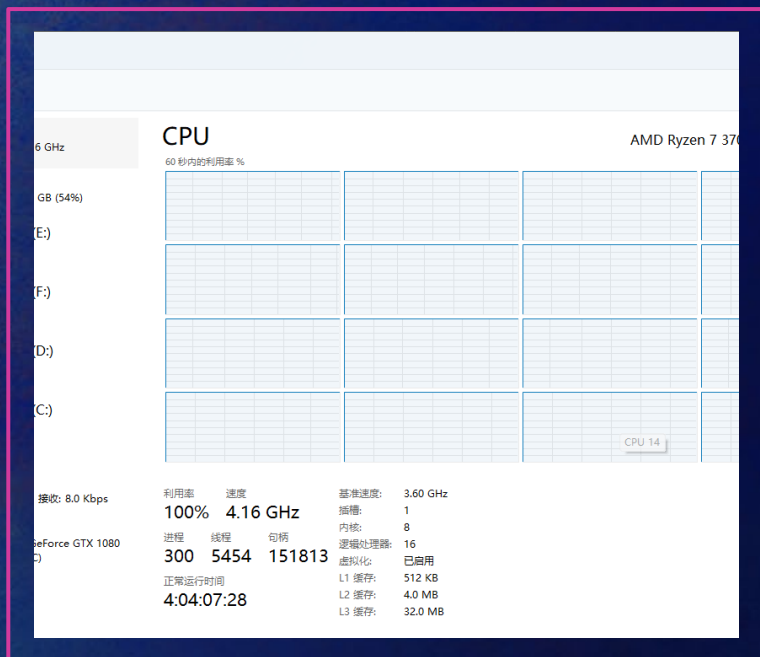


## 响应时间

Min: 1 ms  
Max: 462 ms  
Average: 156 ms

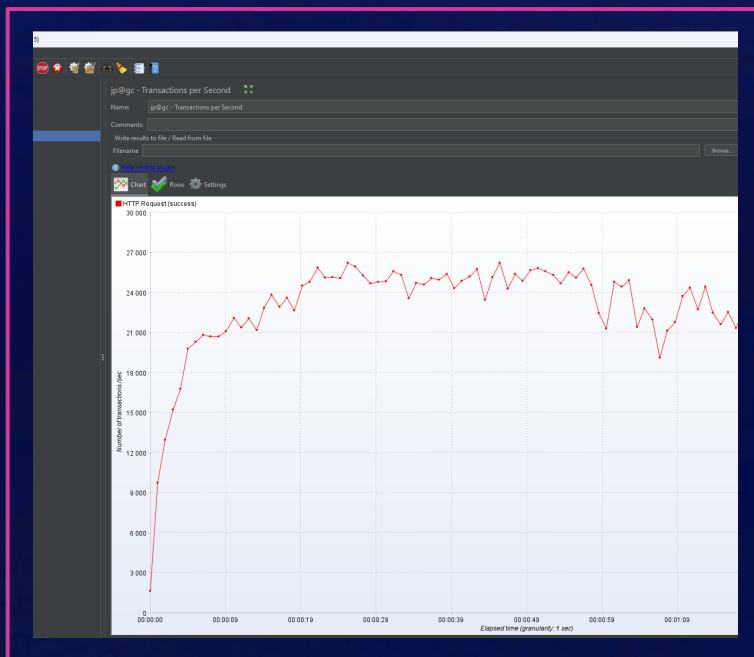


# nolimit-short-text



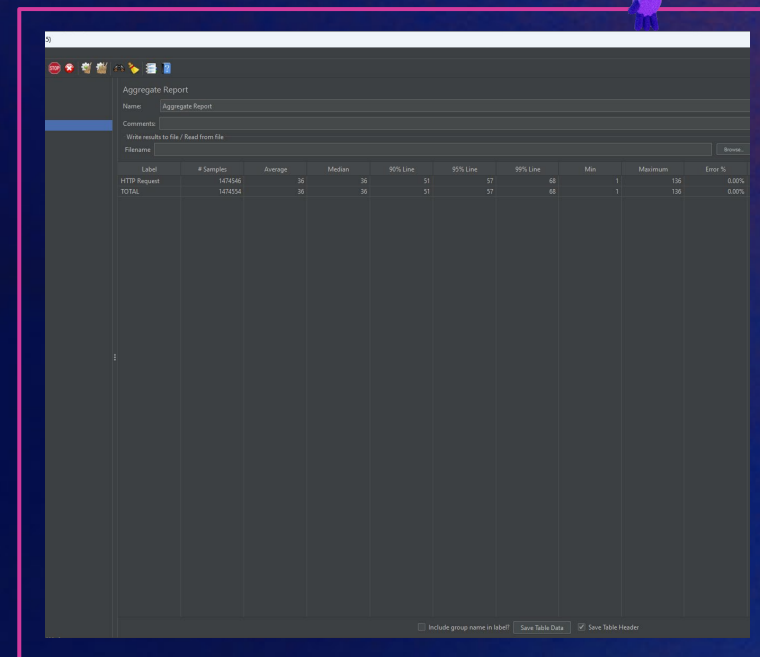
## CPU 使用率

100%



## TPS

峰值: 26000 Transactions per Second

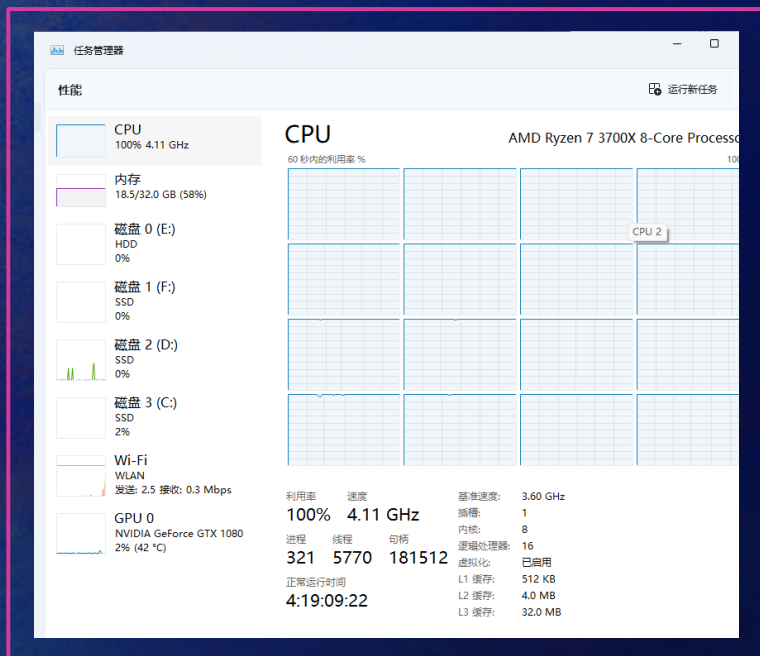


## 响应时间

Min: 1 ms  
Max: 136 ms  
Average: 36 ms



# nolimit-long-text



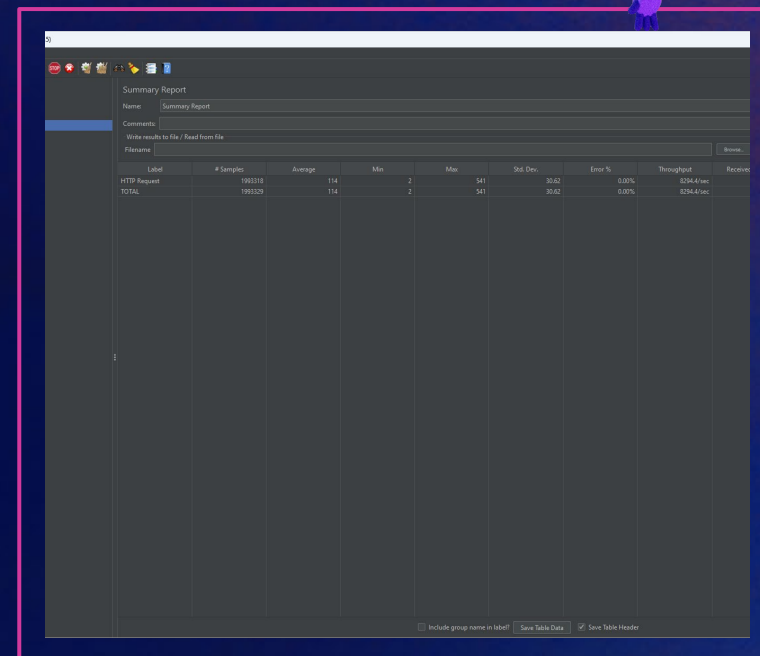
## CPU 使用率

100%



## TPS

峰值: 9050 Transactions per Second



## 响应时间

Min: 2 ms  
Max: 541 ms  
Average: 114 ms



# 测试结果

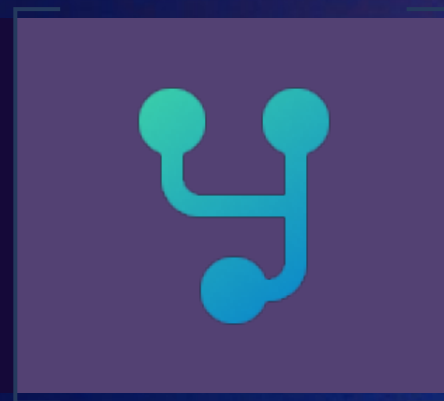


用例	CPU	TPS	Min(ms)	Max(ms)	Avg(ms)
1C1G-short-text	55%	3600	1	518	264
1C1G-long-text	65%	3050	1	581	357
2C1G-short-text	84%	8000	1	284	132
2C1G-long-text	92%	7600	1	462	156
nolimit-short-text	100%	26000	1	136	36
nolimit-long-text	100%	9050	2	541	114





扩展性



Configuration Files  
Configuration Providers  
Configuration Filters  
Direct Forwarding  
HTTP client configuration  
HTTPS & TLS  
Header Guidelines  
Header Routing  
Authentication and Authorization  
Cross-Origin Requests (CORS)  
Session Affinity  
Load Balancing

Middleware  
Transforms  
Destinations Health Checks  
Distributed Tracing  
gRPC  
WebSockets and SPDY  
Service Fabric Integration  
Http.sys Delegation  
Diagnosing proxy issues  
A/B Testing  
HTTP/3  
Lets Encrypt





# 关键扩展点



## Configuration Providers

- Configuration Files
- InMemory
- Ingress Resource

## LoadBalancingPolicy

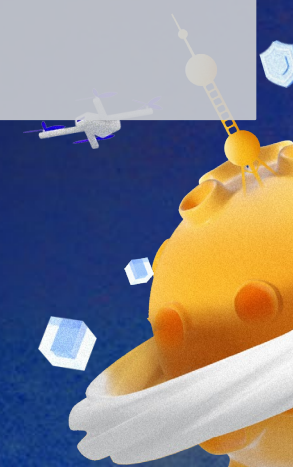
- FirstAlphabetical
- Random
- PowerOfTwoChoices
- RoundRobin
- LeastRequests

## TransformFactory

- Path\*
- Query\*Parameter
- HttpMethodChange
- RequestHeader\*
- X-Forwarded

## Middleware

- Any Middleware





什么时候要打造自己的 Ingress Controller?



单体架构向服务化架构演进时

系统运行环境向k8s迁移时

期望在系统网关上做个性化扩展时

场景



.NET Conf China

# 自定义 Ingress Controller vs Nginx Ingress Controller



- 基于 .NET，有效利用团队能力
- 相较于Nginx具备更灵活的扩展能力
- Ingress Controller 即网关，更短的调用链路
- 基于Ingress Resource 的路由配置，无需额外服务注册发现中间件





# 如何打造自己的 Ingress Controller



# 打造自定义 Ingress Controller



官方样例

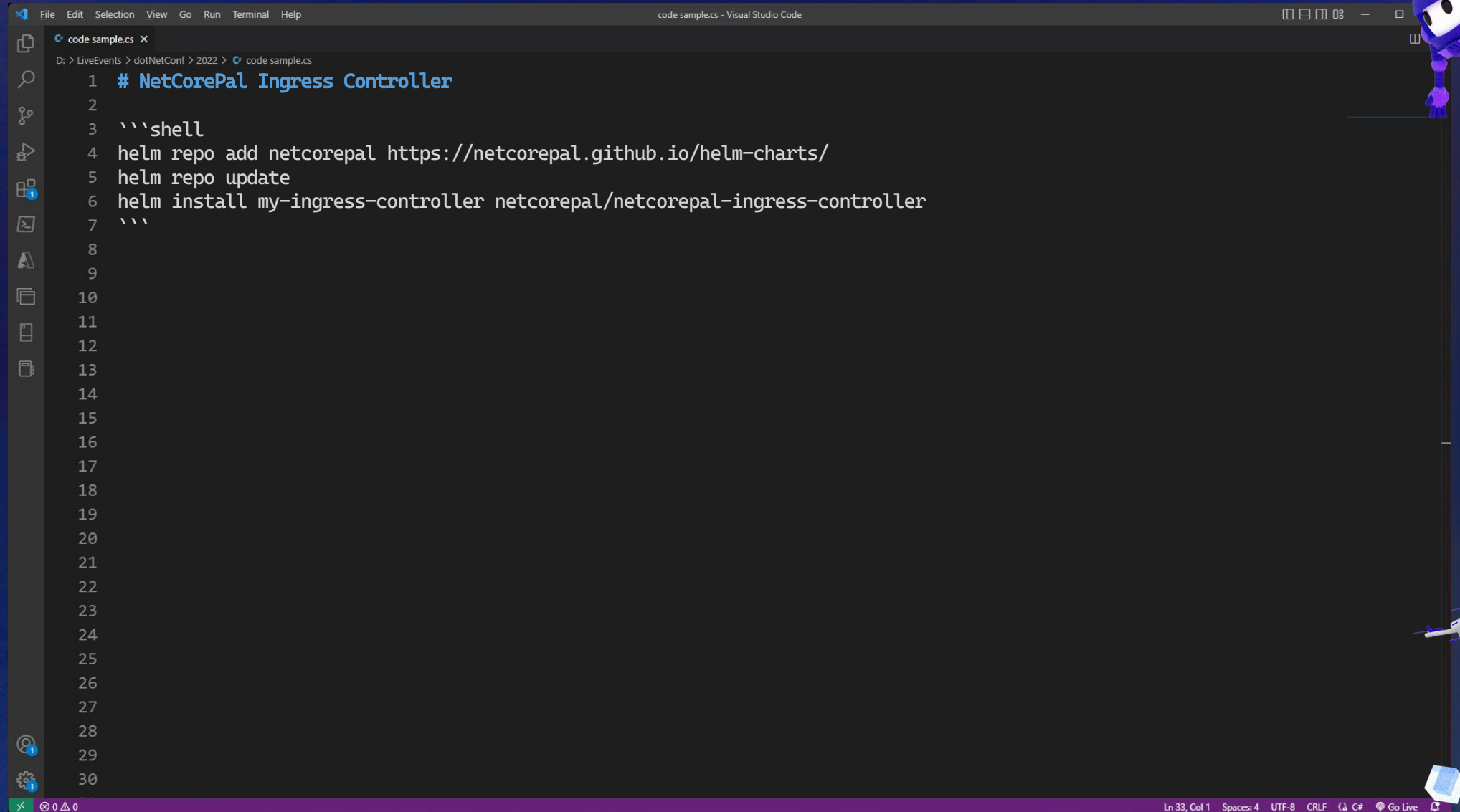
NetCorePal  
Helm Charts

NetCorePal  
YARP Template

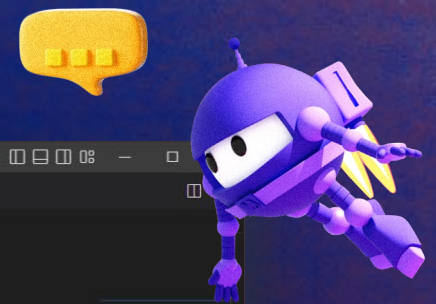


helm repo add netcorepal https://netcorepal.github.io/helm-charts/ helm repo update helm install my-ingress-controller netcorepal/netcorepal-ingress-controller

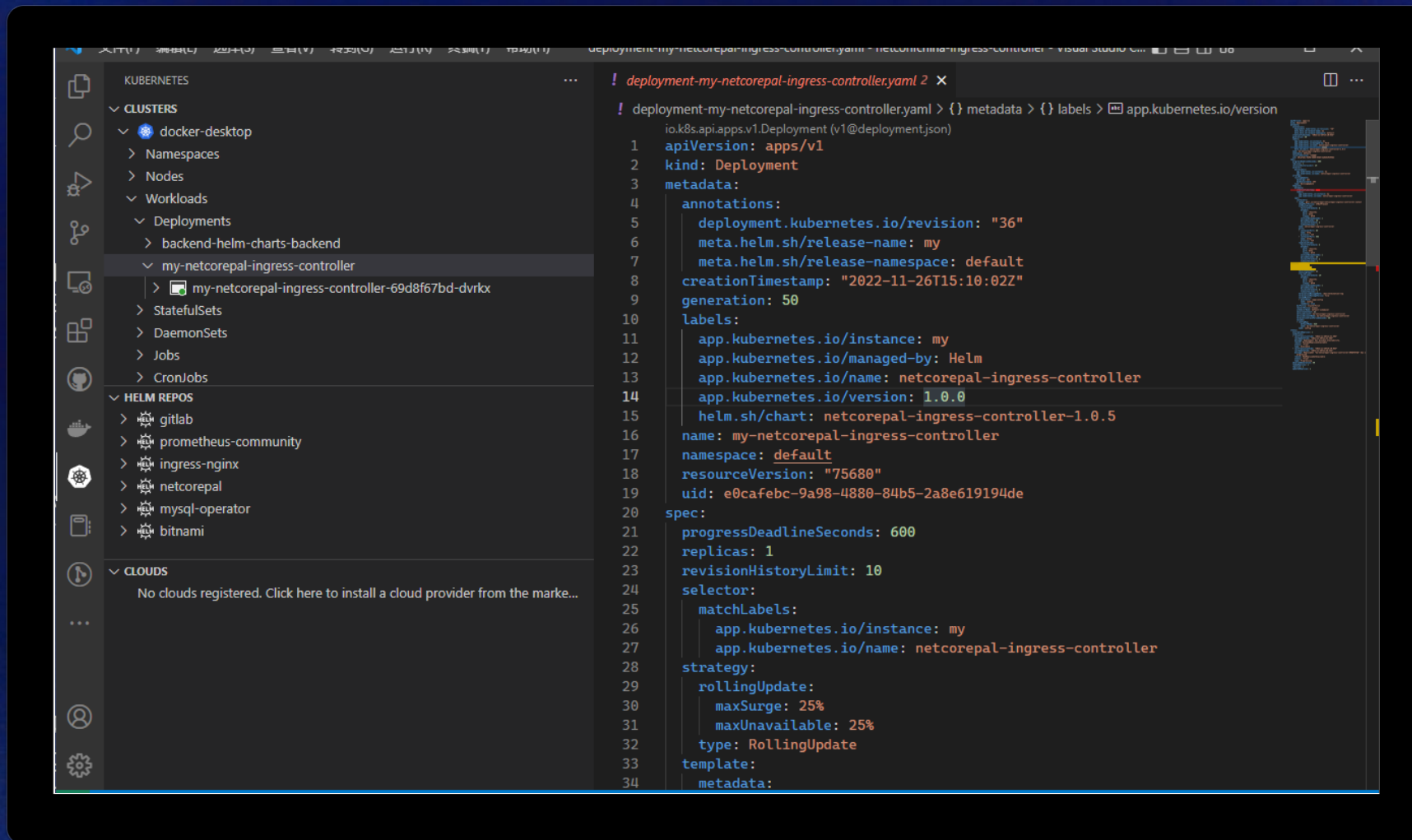
# NetCorePal Helm Charts



```
1 # NetCorePal Ingress Controller
2
3 ```shell
4 helm repo add netcorepal https://netcorepal.github.io/helm-charts/
5 helm repo update
6 helm install my-ingress-controller netcorepal/netcorepal-ingress-controller
7 ```
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
```



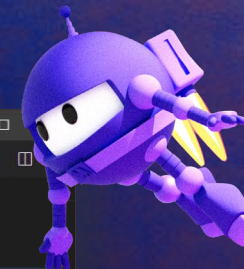
# NetCorePal Helm Charts 部署效果





dotnet new install NetCorePal.YARP.Template dotnet new yarpingresscontroller -n yourname  
dotnet new install NetCorePal.YARP.Template dotnet new yarpingresscontroller -n yourname  
dotnet new install NetCorePal.YARP.Template dotnet new yarpingresscontroller -n yourname

# NetCorePal YARP Template



```
code sample.cs - Visual Studio Code
code sample.cs x
D: > LiveEvents > dotNetConf > 2022 > code sample.cs
1  ## How to use
2
3  ```shell
4  dotnet new install NetCorePal.YARP.Template
5  dotnet new yarpingresscontroller -n yourname
6  ```
7
8
9  dotnet new yarpingresscontroller -n NetConfChina
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
```



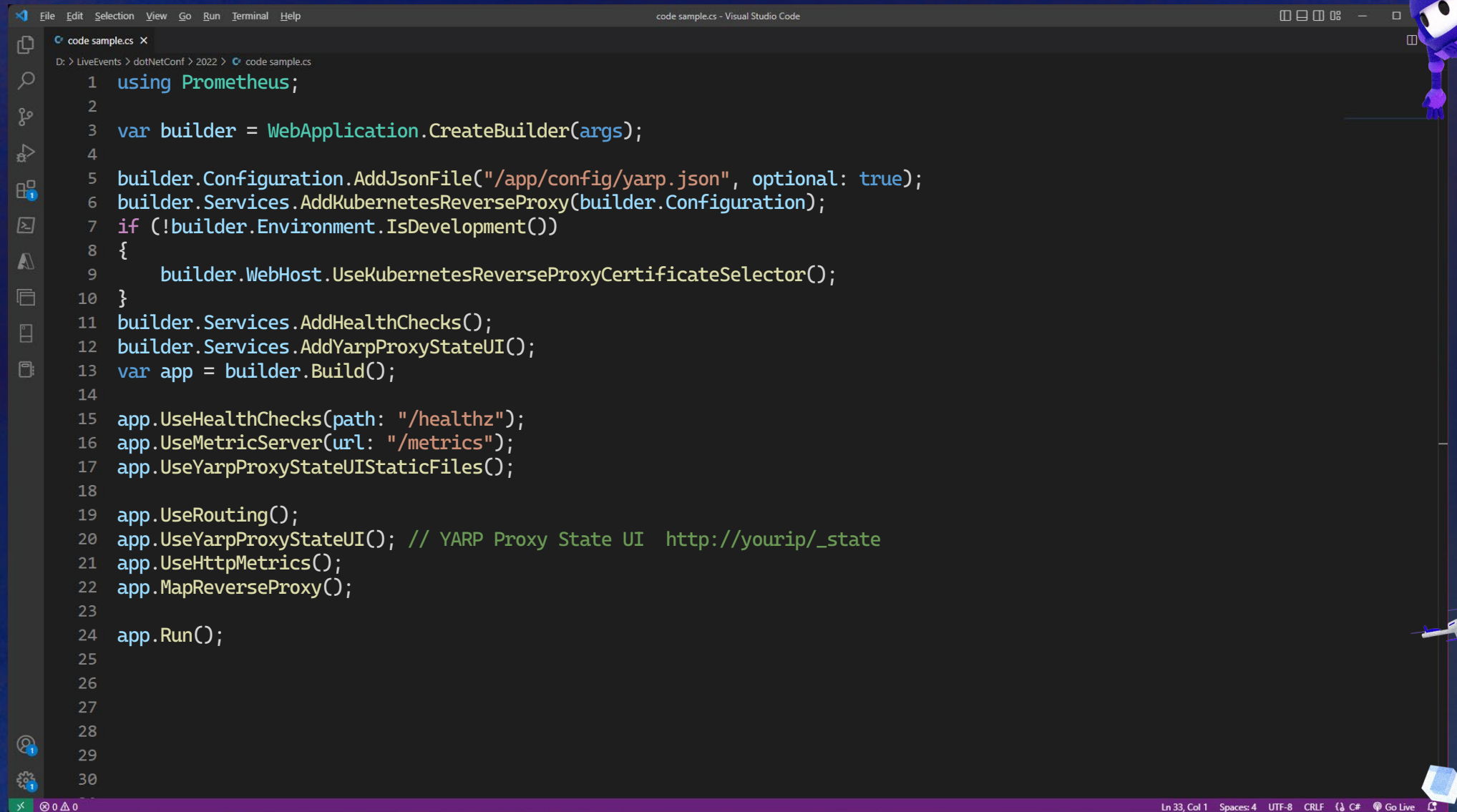
# 工程目录

> 此电脑 > 本地磁盘 (D:) > code > netconfchina-ingress-controller >

名称	修改日期	类型	大小
helm-chart	2022/11/30 22:51	文件夹	
src	2022/11/30 22:51	文件夹	
.dockerignore	2022/11/30 22:51	DOCKERIGNOR...	1 KB
.gitignore	2022/11/30 22:51	Git Ignore 源文件	6 KB
NetConfChina.IngressController.sln	2022/11/30 22:51	Visual Studio Sol...	2 KB
nuget.config	2022/11/30 22:51	Configuration 源...	1 KB
README.md	2022/11/30 22:51	Markdown 源文件	1 KB
values.yaml	2022/11/30 22:51	Yaml 源文件	2 KB

dotnet new install NetCorePal.YARP.Template dotnet new yarpingresscontroller -n yourname  
dotnet new install NetCorePal.YARP.Template dotnet new yarpingresscontroller -n yourname  
dotnet new install NetCorePal.YARP.Template dotnet new yarpingresscontroller -n yourname

# Program.cs

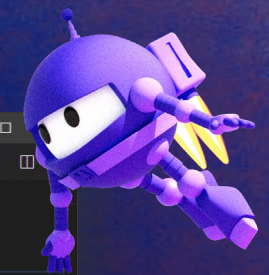


```
1 using Prometheus;
2
3 var builder = WebApplication.CreateBuilder(args);
4
5 builder.Configuration.AddJsonFile("/app/config/yarp.json", optional: true);
6 builder.Services.AddKubernetesReverseProxy(builder.Configuration);
7 if (!builder.Environment.IsDevelopment())
8 {
9     builder.WebHost.UseKubernetesReverseProxyCertificateSelector();
10 }
11 builder.Services.AddHealthChecks();
12 builder.Services.AddYarpProxyStateUI();
13 var app = builder.Build();
14
15 app.UseHealthChecks(path: "/healthz");
16 app.UseMetricServer(url: "/metrics");
17 app.UseYarpProxyStateUIStaticFiles();
18
19 app.UseRouting();
20 app.UseYarpProxyStateUI(); // YARP Proxy State UI http://yourip/_state
21 app.UseHttpMetrics();
22 app.MapReverseProxy();
23
24 app.Run();
25
26
27
28
29
30
```

dotnet new install NetCorePal.YARP.Template dotnet new yarpingresscontroller -n yourname  
dotnet new install NetCorePal.YARP.Template dotnet new yarpingresscontroller -n yourname  
dotnet new install NetCorePal.YARP.Template dotnet new yarpingresscontroller -n yourname

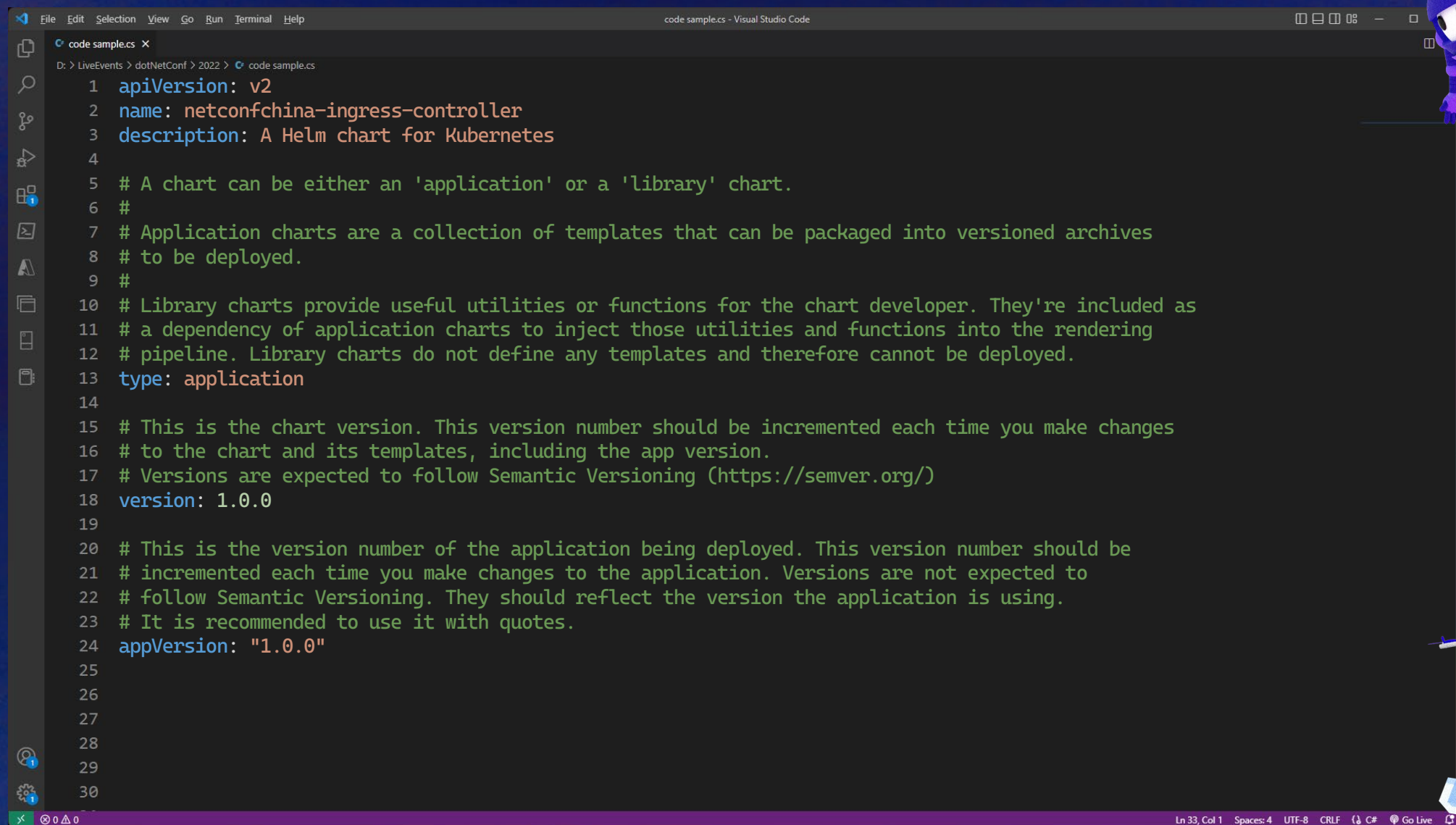
# values.yaml

```
code sample.cs - Visual Studio Code
code sample.cs X
D: > LiveEvents > dotNetConf > 2022 > code sample.cs
1 # Default values for netconfchina-ingress-controller.
2 # This is a YAML-formatted file.
3 # Declare variables to be passed into your templates.
4
5 replicaCount: 1
6
7 image:
8   repository: netconfchina
9   pullPolicy: IfNotPresent
10  # Overrides the image tag whose default is the chart appVersion.
11  tag: "latest"
12
13 imagePullSecrets: []
14 nameOverride: ""
15 fullnameOverride: ""
16
17 yarp:
18   # default use .Release.Name
19   controllerClass: ""
20   isDefaultClass: false
21   serverCertificates: false
22   # default use .Release.Namespace
23   defaultSslCertificateSecretNamespace: ""
24   defaultSslCertificateSecretName: ""
25
26
27
28
29
30
```

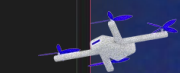


dotnet new install NetCorePal.YARP.Template dotnet new yarpingresscontroller -n yourname  
dotnet new install NetCorePal.YARP.Template dotnet new yarpingresscontroller -n yourname  
dotnet new install NetCorePal.YARP.Template dotnet new yarpingresscontroller -n yourname

# Chart.yaml

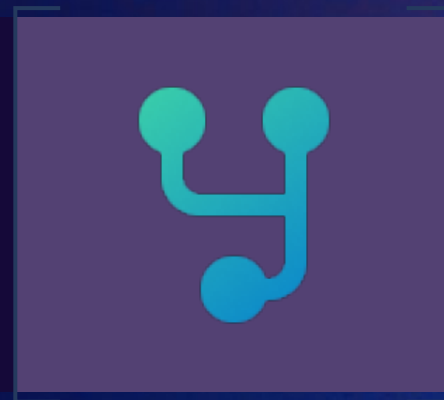


```
code sample.cs - Visual Studio Code
D:\> LiveEvents > dotNetConf > 2022 > code sample.cs
1 apiVersion: v2
2 name: netconfchina-ingress-controller
3 description: A Helm chart for Kubernetes
4
5 # A chart can be either an 'application' or a 'library' chart.
6 #
7 # Application charts are a collection of templates that can be packaged into versioned archives
8 # to be deployed.
9 #
10 # Library charts provide useful utilities or functions for the chart developer. They're included as
11 # a dependency of application charts to inject those utilities and functions into the rendering
12 # pipeline. Library charts do not define any templates and therefore cannot be deployed.
13 type: application
14
15 # This is the chart version. This version number should be incremented each time you make changes
16 # to the chart and its templates, including the app version.
17 # Versions are expected to follow Semantic Versioning (https://semver.org/)
18 version: 1.0.0
19
20 # This is the version number of the application being deployed. This version number should be
21 # incremented each time you make changes to the application. Versions are not expected to
22 # follow Semantic Versioning. They should reflect the version the application is using.
23 # It is recommended to use it with quotes.
24 appVersion: "1.0.0"
25
26
27
28
29
30
Ln 33, Col 1  Spaces: 4  UTF-8  CRLF  C#  Go Live
```





# 相关资料



YARP GitHub: <https://github.com/microsoft/reverse-proxy>

Docs: <https://microsoft.github.io/reverse-proxy/articles/getting-started.html>

NetCorePal Ingress Controller: <https://github.com/netcorepal/netcorepal-ingress-controller>

NetCorePal Helm Charts: <https://github.com/netcorepal/helm-charts>

NetCorePal YARP Template: <https://github.com/netcorepal/netcorepal-yarp-template>

Performance Testing: <https://github.com/netcorepal/yarp-performance-testing>

## 相关资料



.NET Conf China

# Thank you!

Wechat: xiaoweiyu2020

