

从 .NET Framework 到云原生的那些坑

主讲：刘海峰 / MVP | 51Aspx创始人



Agenda

.NET Conf China
2022
开源 · 安全 · 赋能



- 为什么要.NET云原生？
- 传统的Framework如何云原生？
- .NET云原生如何避坑？

https://dotnet.microsoft.com/en-us/

Microsoft | .NET Why .NET Features Learn Docs Downloads Community LIVE TV All Microsoft

Free. Cross-platform. Open source.

Build any app with .NET

Create beautiful apps and scalable cloud services, faster and easier with the free, open-source platform that's loved by developers and trusted by organizations.

[Download](#) [Get started](#)

Supported on Windows, Linux, and macOS



About

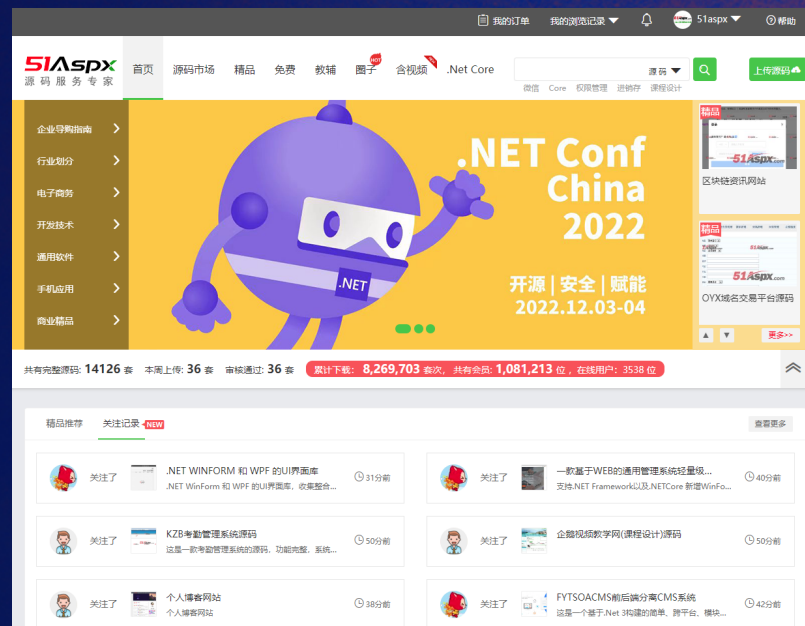


2007年开始，坚持守护.NET这方净土，助力.NET生态发展



About 51Aspx

- 专注于.NET源码技术服务
- 创建于2007年，2009年商业化运营
- 现有注册会员108万
- 完整项目源码14000套
- 最高在线人数10000人
- 微信公众号粉丝10万人
- 2014年获得微软云创意大赛企业组冠军

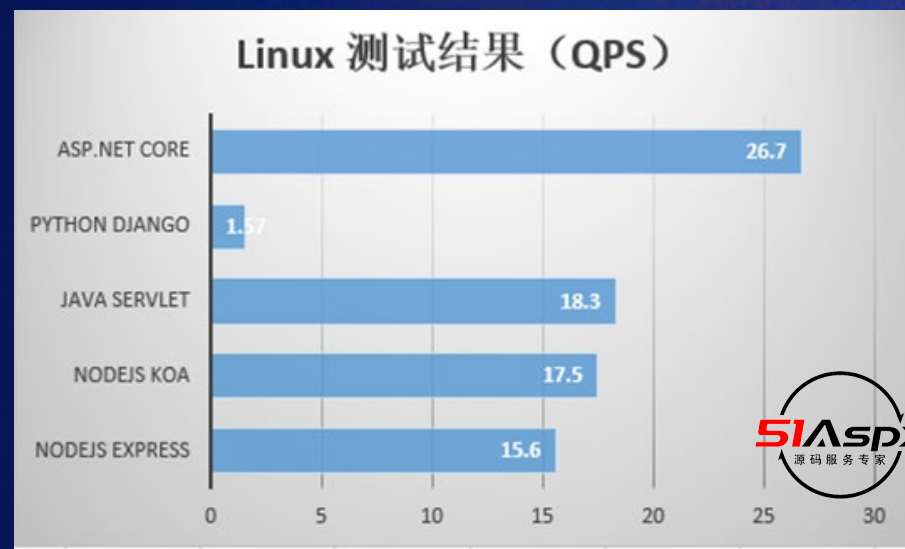


让.NET开发者的成果变现

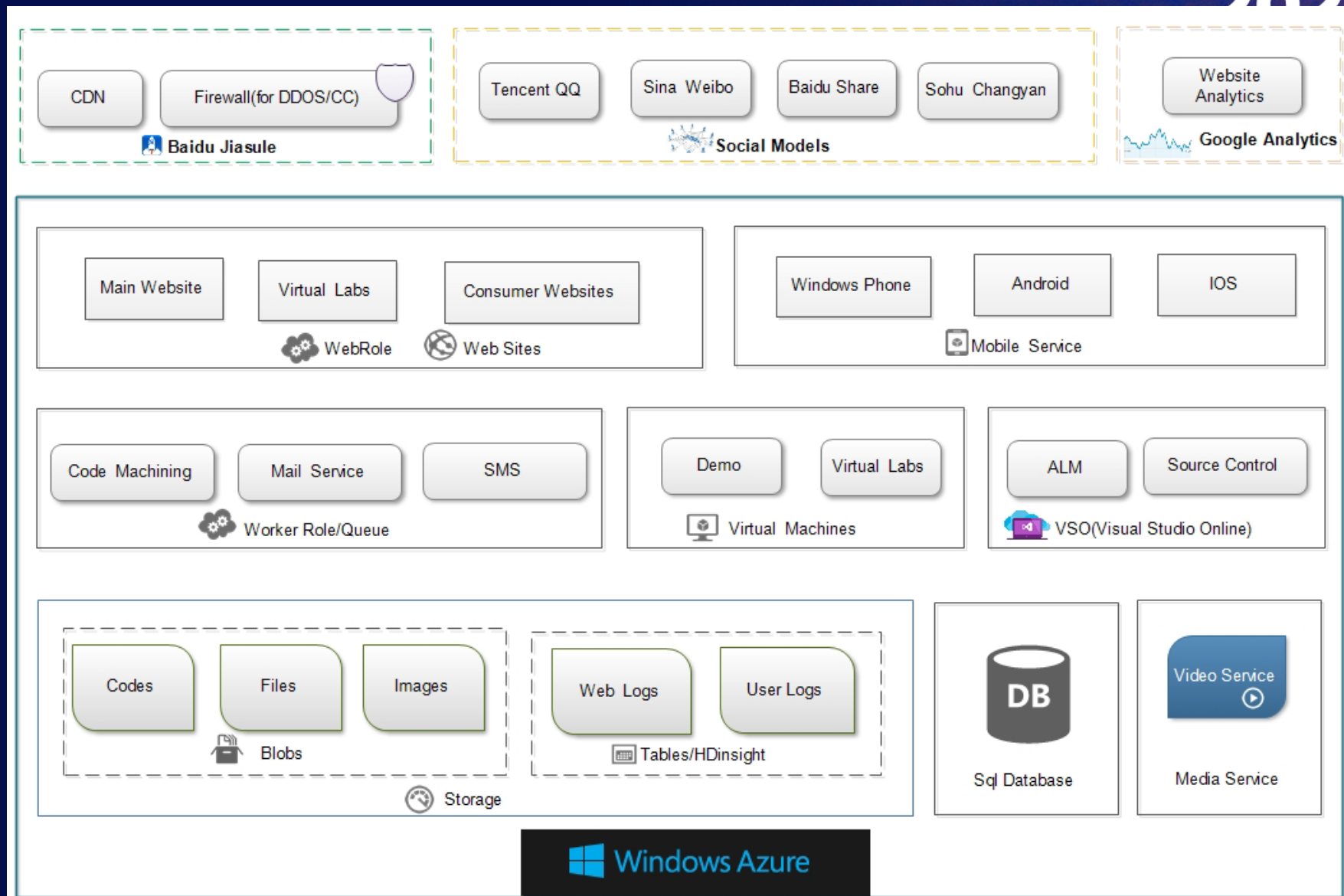
.NET Conf China

为什么要升级?

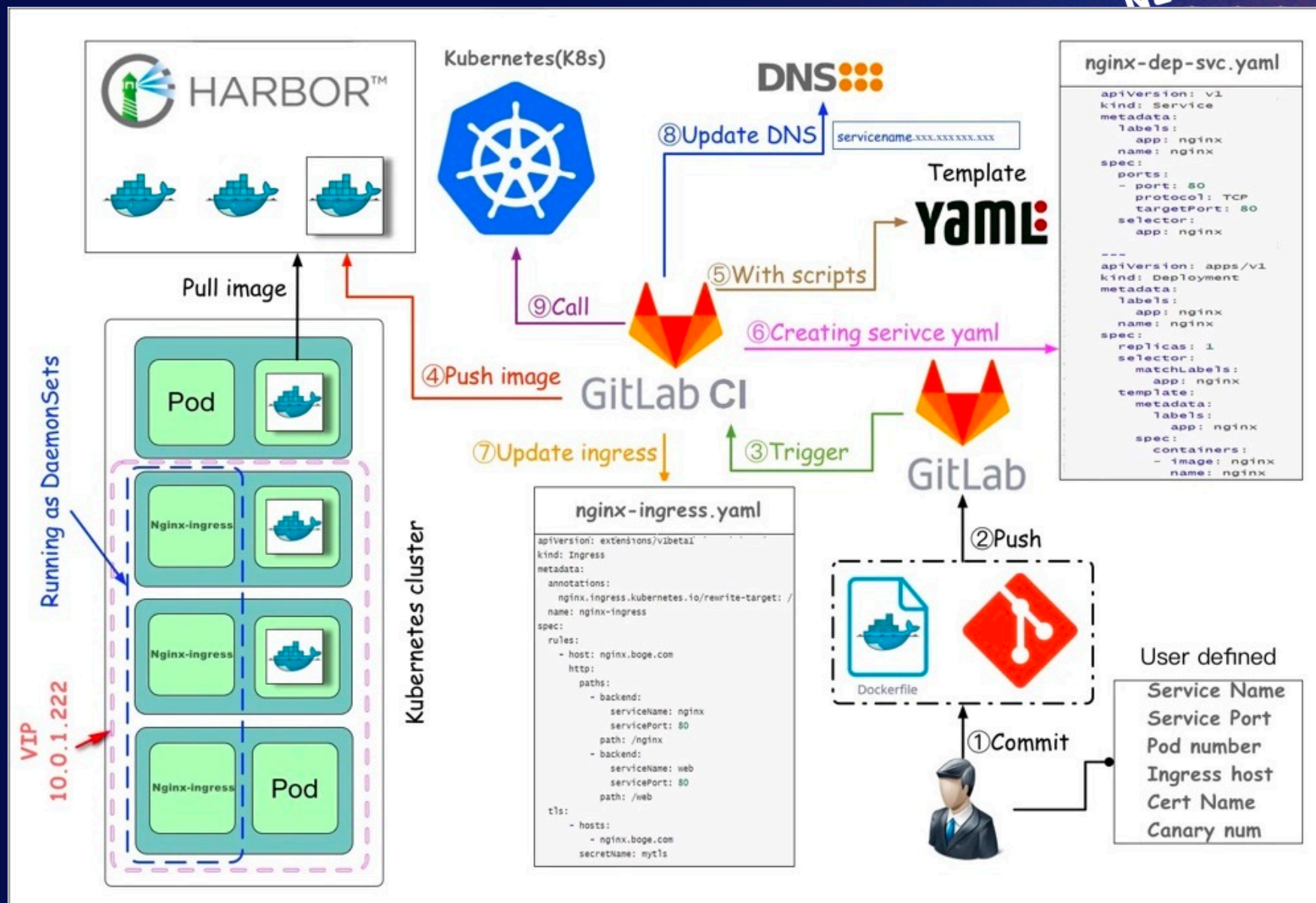
- 平台本身
 - 服务器压力逐渐增大
 - 升级扩展很难与新技术兼容
 - 架构越来越复杂, 运维难度增大
- .NET技术的先进性
 - 可以跨平台运行, 启动快, 资源消耗低
 - 更轻量: 体积小 (.NET6+Alpine镜像)
 - 微服务、容器化, 更方便CI/CD
 - .NET的开源协议更宽松, 完全社区化
 - 为云原生而生, 可以更灵活的调配资源



历史架构: Azure+.NET Framework

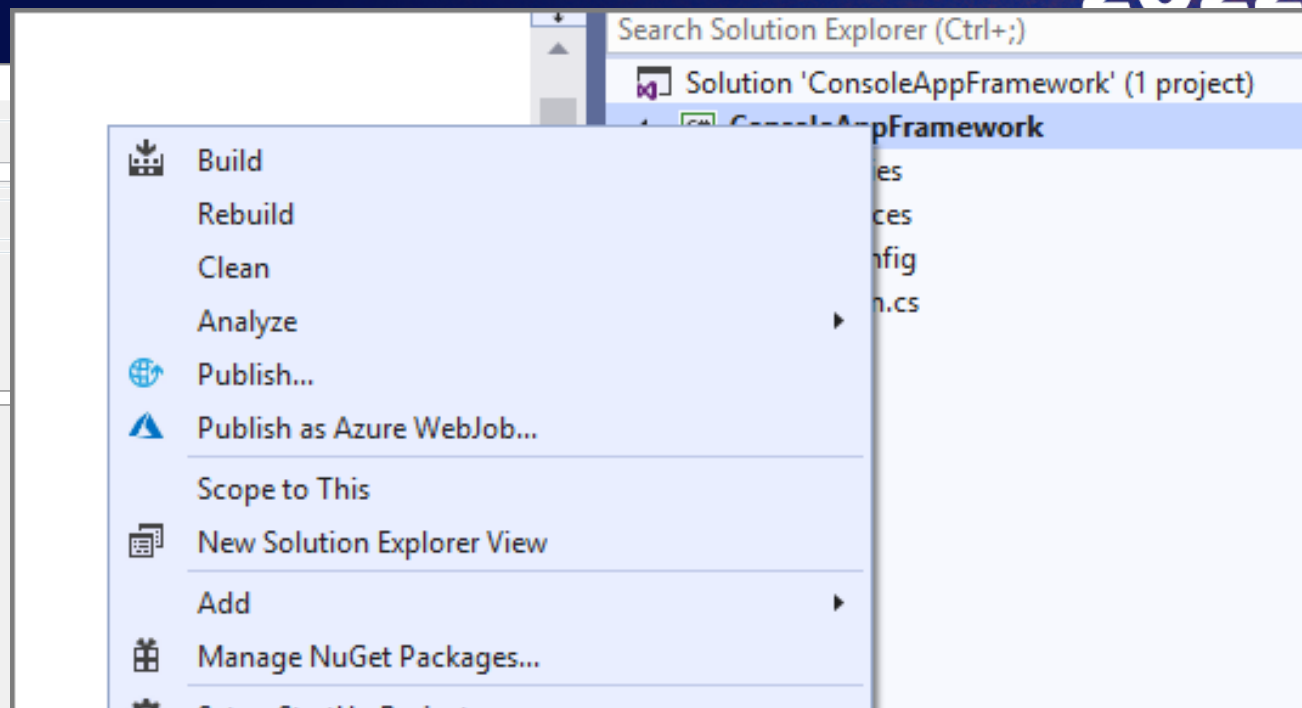
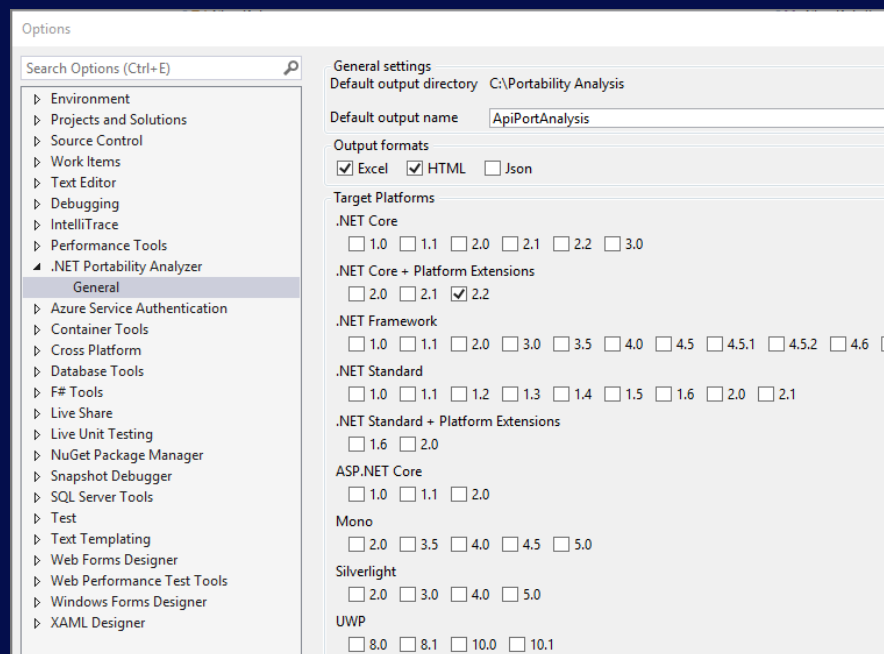


目标架构: 云原生 + .NET



升级前的准备? .Net可移植性分析

.NET Conf China
2022



Submission Id	5552b6d0-b7ed-4fa8-9a3c-e3d8099c7ab1				
Description					
Targets	.NET Core + Platform Extensions,.NET Core,.NET Framework,.NET Standard + Platform Extensions				
Assembly Name	Target Framework	.NET Core + Platform	.NET Core	.NET Framework	.NET Standard + Platform Extensions
svcutil	.NETFramework,Version=v4.5	71.24	71.48	100	71.24
API Catalog last updated on	Tuesday, March 5, 2019				
See 'http://go.microsoft.com/fwlink/?LinkId=397652' to learn how to read this table					
Portability Summary Details					

<https://learn.microsoft.com/zh-cn/dotnet/standard/analyzers/portability-analyzer>

```

Current Project: D:\Source\AspNetServer\Valley.Mvc.Template\src\Valley.Mvc.Template\Valley.Mvc.Template.csproj

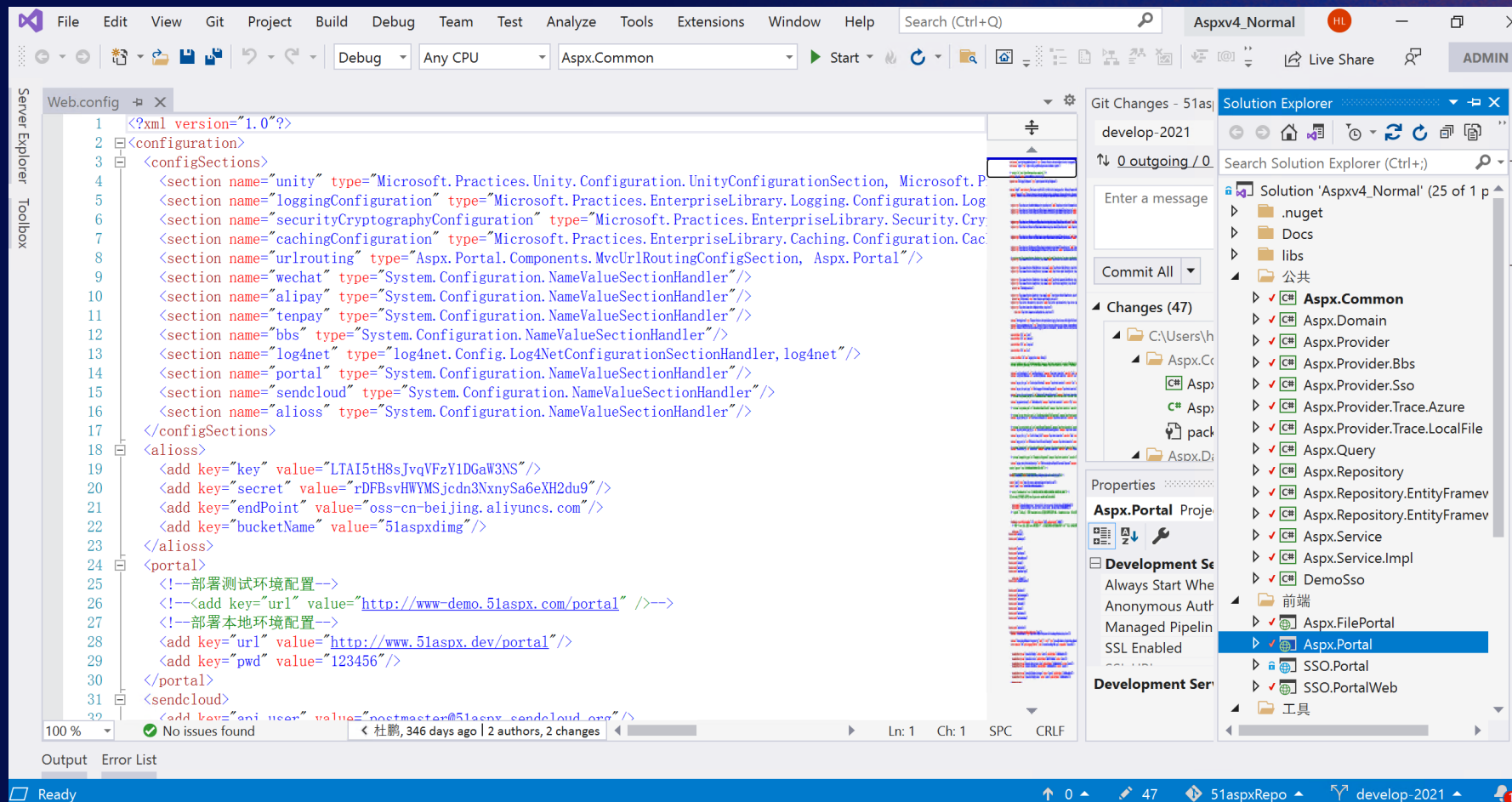
1. [Template] Backup project
2. [Template] Convert project file to SDK style
3. [New step] Update TFM
4. Update NuGet packages
5. Add template files
6. Migrate any unific files
7. Migrate appSettings
8. Disable unsupported configuration sections
9. Migrate System.Web.WebPages.Razor.PageNamespace
10. Update Cs Source
    1. Apply fix for ASPNET: ASP.NET Core projects should not reference ASP.NET project
    2. Apply fix for ASPNET: MvcFilterTypes should be replaced with Microsoft.AspNetCore.Mvc.Filters
    3. Apply fix for ASPNET: AccountResult types should move from the Microsoft.AspNetCore.Mvc namespace
    4. Apply fix for ASPNET: HttpContext filter system should be used from the Microsoft.AspNetCore.Mvc.Filters namespace
    5. Apply fix for ASPNET: do not use HttpContext.Current
    6. Apply fix for ASPNET: HttpContext.Diagnostics.DebuggerEnabled should be replaced with System.Diagnostics.Debugger.IsAttached
    7. Apply fix for ASPNET: HttpContext should be replaced with HttpContext
    8. Apply fix for ASPNET: HttpContext should be replaced with HttpContext
    9. Apply fix for ASPNET: HttpContext should be replaced with HttpContext
    10. Apply fix for ASPNET: HttpContext should be replaced with HttpContext
11. Move to next project
12. Done

```

<https://dotnet.microsoft.com/zh-cn/platform/upgrade-assistant/tutorial/intro>

历史代码结构: .NET Framework4.8+MVC

.NET Conf China
2022
开源 · 安全 · 赋能



.NET Conf China

当前：.NET+MVC+Container



```
1 FROM mcr.microsoft.com/dotnet/aspnet:6.0 AS base
2 WORKDIR /app
3 EXPOSE 80
4 EXPOSE 443
5 EXPOSE 9000
6
7 # ENV ASPNETCORE_ENVIRONMENT=Development
8
9 FROM mcr.microsoft.com/dotnet/sdk:6.0 AS build
10 WORKDIR /src
11 COPY ["Aspx.Portal/Aspx.Portal.csproj", "Aspx.Portal/"]
12 RUN dotnet restore "Aspx.Portal/Aspx.Portal.csproj"
13 COPY . .
14 WORKDIR "/src/Aspx.Portal"
15 RUN dotnet build "Aspx.Portal.csproj" -c Release -o /app/build
16
17 FROM build AS publish
18 RUN dotnet publish "Aspx.Portal.csproj" -c Release -o /app/publish
19
20 FROM base AS final
21 WORKDIR /app
22 COPY --from=publish /app/publish .
23 ENTRYPOINT ["dotnet", "Aspx.Portal.dll"]
24
```


目标架构：云原生 + .NET



ruanjimu

Only User Namespaces

Starred

Cluster

Workload

- CronJobs 0
- DaemonSets 0
- Deployments 4
- Jobs 0
- StatefulSets 0
- Pods 3

Apps

Service Discovery

Storage

Monitoring

More Resources

Deployment: aspxwww (Active)

Namespace: default Age: 3 days Pod Restarts: 0

Namespace * default

Name * aspxwww

Description
Any text you want that better describes this resource

Replicas * 3

Deployment Pod container-0 + Add Container

General

Health Check

Resources

Security Context

Storage

General

Container Name
container-0

Init Container

Standard Container

Image

Container Image
registry.cn-xx-xx.aliyuncs.com/aspxwww:1.1.5

Pull Policy
IfNotPresent

Pull Secrets
g x

Ports

Service Type
Node Port

Name
web

Private Container Port
80

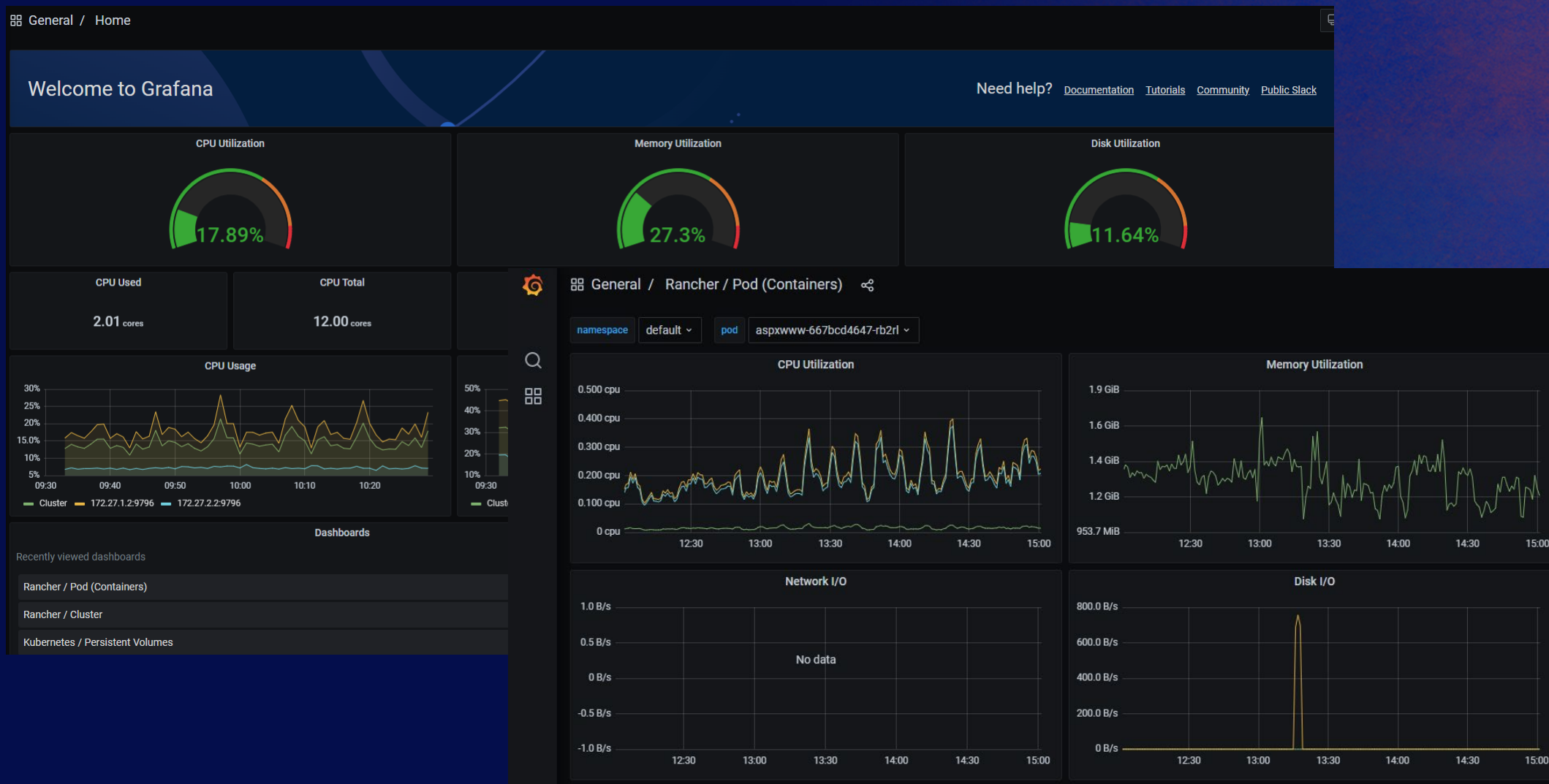
Protocol
TCP

Listening Port
32080

Add Port

Cluster Tools

目标架构：云原生 + .NET



遇到的一些挑战和经验分享

- Windows的组件和API都失效
- 第三方组件的兼容
- Cookie不兼容
- 图形验证码不显示
- 文件名区分大小写导致很多路径无效
- 不支持属性注入
- SEO配置全部失效
- ...



- 数据库PaaS化
- 数据持久化
- 无状态化
- 容器化





从PaaS+IaaS-》容器化



.NET+云原生

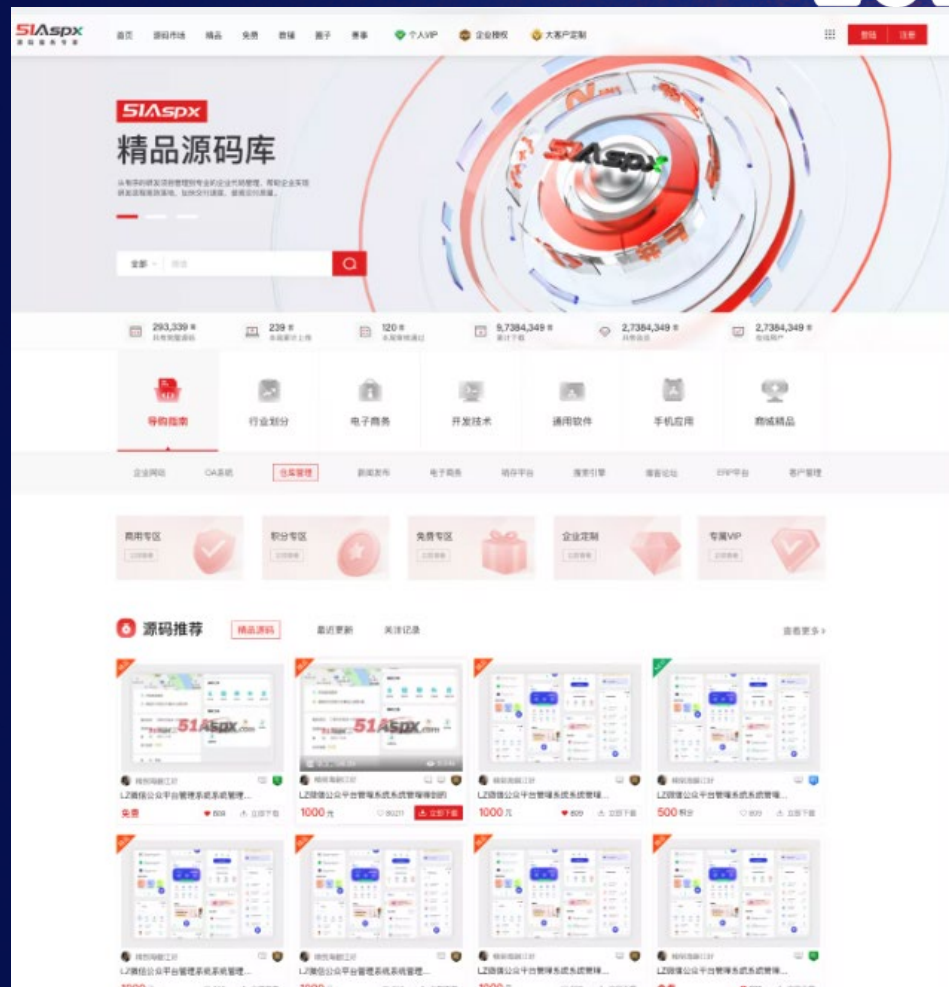
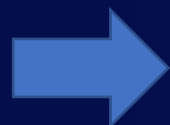
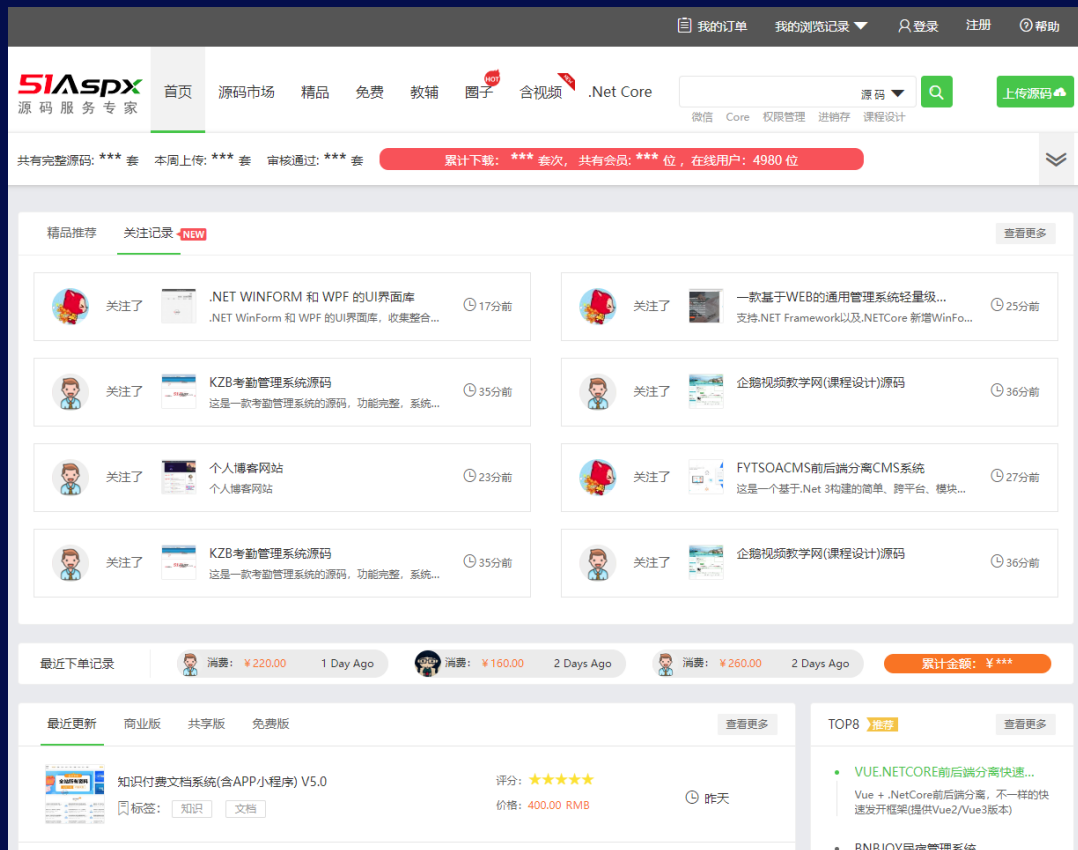
更标准、更灵活，跨云更随意

.NET Conf China



.NET Conf China
2022

赋能



更标准、更灵活，跨云更随意

《.NET云原生开发实战》

Jump to file

chapter_01

img_01

01_CloudNativeDefinition-v2.md

chapter_02

chapter_03

chapter_04

chapter_05

chapter_06

chapter_07

chapter_08

chapter_09

chapter_10

416 lines (230 loc) · 37.9 KB

CodeBlameRaw

第1章：云原生应用的设计与运行

弹指一挥间，应用的架构设计走过了单体应用架构和微服务架构，来到了云原生架构的时代。架构演进的背后，是软件系统面对应用规模不断扩大，资源使用越来越灵活，扩展性越来越强的挑战作出的反应和应对，所谓云原生，它不是一个产品，而是一套技术体系和一套方法论，云原生是思想先行，从内到外的整体变革。更确切地说，它是一种文化，更是一种潮流，是云计算的一个必然导向。

1.1 日益普及的云原生应用

某一天，你的公司需要与领先的电子商务网站竞争，作为系统架构师，你需要为公司设计一个先进的电子商务应用，应该如何构建它呢？以你 15 年以上的经验，很可能你会设计出如图 1-1 所示的系统。

Client app

Monolithic server process

Database



.NET Conf China
2022
开源 · 安全 · 赋能



51Aspx.com



.NET Conf China
2022
开源 · 安全 · 赋能

THANK!

感谢聆听