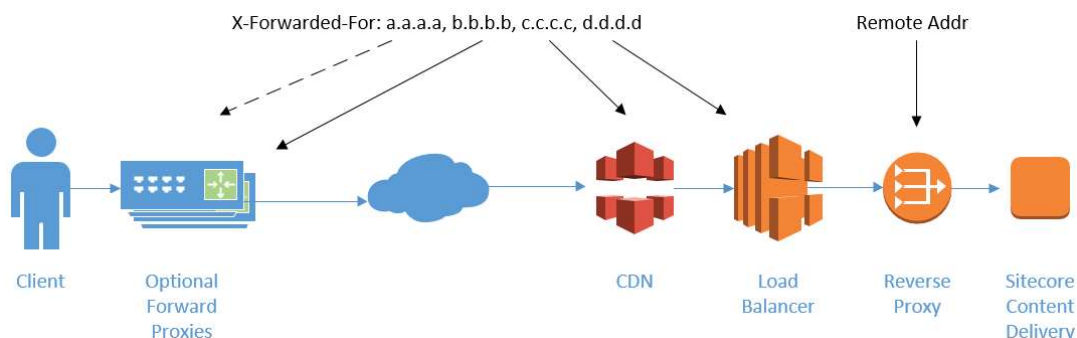


项目发布相关配置

2018年11月20日 16:18

代理转接头设置



代理和负载均衡转接头配置: X-Forwarded-For、X-Forwarded-Proto 和 X-Forwarded-Host

HttpContext.Connection.RemoteIpAddress

HttpContext.Request.Scheme

HttpContext.Request.Host

代理设置: X-Forwarded-Proto 和 X-Forwarded-For

Install-package Microsoft.AspNetCore.HttpOverrides

```
app.UseForwardedHeaders(new ForwardedHeadersOptions
{
    ForwardedHeaders = ForwardedHeaders.XForwardedFor | ForwardedHeaders.XForwardedProto
});
```

默认信任 localhost (127.0.0.1, [::1])转接过来的请求, 要信任其它来源, 需要配置白名单:

```
services.Configure<ForwardedHeadersOptions>(options =>
{
    options.KnownProxies.Add(IPAddress.Parse("10.0.0.100"));
});
```

自动添加转接头

在IIS中: 当应用在 IIS 和 ASP.NET Core 模块后方运行时, 转接头中间件由 IIS 集成中间件默认启用。

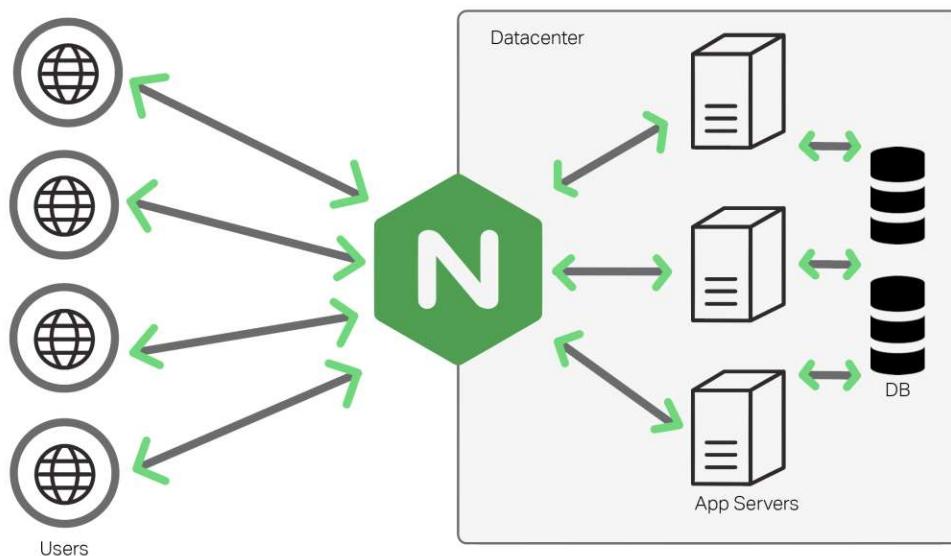
NGINX 中的代理头配置

来自 <<https://www.nginx.com/resources/wiki/start/topics/examples/forwarded/>>

Apache 使用 mod_proxy 模块设置

来自 <https://httpd.apache.org/docs/2.4/mod/mod_proxy.html#x-headers>

■ 负载均衡设置



- 1、转发功能：根据配置算法，将客户端请求转发到不同服务器（或容器）上，减轻单服务器压力，提高并发量。
- 2、故障移除：心跳检测，判断服务器是否正常工作，如果宕掉，自动将请求发送到其他服务器。
- 3、恢复添加：检测到发生故障服务器恢复工作，自动添加到处理用户请求队伍中。

Nginx算法：轮询、IP哈希、URL哈希、权重。

■ 项目生成与发布过程

确保生成目标：

```
<Project Sdk="Microsoft.NET.Sdk.Web">
```

计算项目项：

判断内容类型和处理方式

命令行发布：

```
dotnet new mvc
```

```
dotnet publish D:\Webs\Web1
```

发布配置文件：

```
dotnet publish WebApplication.csproj /p:PublishProfile=<FolderProfileName>
```

排除文件：

以下 <Content>元素将从 wwwroot/content 文件夹及其所有子文件夹中排除所有文本 (.txt) 文件：

```
<ItemGroup>
  <Content Update="wwwroot/content/**/*.txt" CopyToPublishDirectory="Never" />
</ItemGroup>
```

注意：可以添加到项目文件或发布配置文件，如果添加到项目文件，则影响所有发布配置。

以下 <MsDeploySkipRules> 元素排除 wwwroot/content 文件夹中的所有文件：

```
<ItemGroup>
  <MsDeploySkipRules Include="CustomSkipFolder">
    <ObjectName>dirPath</ObjectName>
    <AbsolutePath>wwwroot\content</AbsolutePath>
  </MsDeploySkipRules>
</ItemGroup>
```

包含文件：

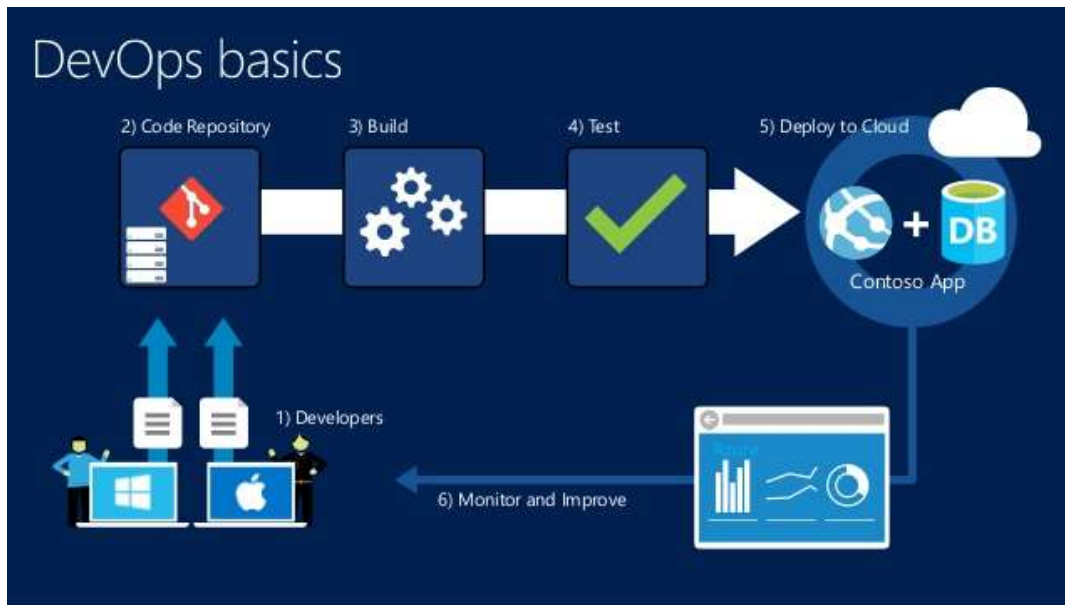
以下标记将项目目录外的 images 文件夹包含到发布站点的 wwwroot/images 文件夹中：

```
<ItemGroup>
  <_CustomFiles Include="$(MSBuildProjectDirectory)/../images/**/*.*" />
  <DotnetPublishFiles Include="@(_CustomFiles)">
    <DestinationRelativePath>wwwroot/images/%(RecursiveDir)%(Filename)%(Extension)</DestinationRelativePath>
  </DotnetPublishFiles>
</ItemGroup>
```

发布前后运行目标：

```
<Target Name="CustomActionsBeforePublish" BeforeTargets="BeforePublish">
  <Message Text="Inside BeforePublish" Importance="high" />
</Target>
<Target Name="CustomActionsAfterPublish" AfterTargets="AfterPublish">
  <Message Text="Inside AfterPublish" Importance="high" />
</Target>
```

■ 谈谈持续集成与发布



你所在的公司是如何实施DevOps的？

来自 <<https://www.zhihu.com/question/24413538>>