IFUSION  
To create a simple Hello World ASP.NET MVC application and deploy it on IIS, follow these steps:

**Gitlab ultimate**:192.168.3.170  
**windows runner**:192.168.3.177 nutanix

**IIS Server**: 192.168.3.68

**Step-by-Step Guide**

**1. Create a New ASP.NET MVC Project**

1. **Open Command Prompt or PowerShell**.
2. **Create a new project**:

#dotnet new mvc -o HelloWorldWebApp

#cd HelloWorldWebApp

1. **Modify the HomeController**:

Open Controllers/HomeController.cs and modify the Index action to return a simple "Hello, World!" message.

#using Microsoft.AspNetCore.Mvc;

namespace HelloWorldWebApp.Controllers

{

public class HomeController : Controller

{

public IActionResult Index()

{

return Content("Hello, World!");

}

}

}

2. Build the Project

**Restore dependencies and build the project:**

#dotnet restore

#dotnet build

1. **Publish the project**:

#dotnet publish -c Release -o ./publish

This command will create a publish directory containing all necessary files to run your application.

**3. Deploy to IIS**

1. **Install IIS** (if not already installed):
   * Go to Control Panel > Programs and Features > Turn Windows features on or off.
   * Check "Internet Information Services" and install it.
2. **Add a new Application Pool**:
   * Open IIS Manager.
   * Right-click on "Application Pools" and select "Add Application Pool".
   * Name it HelloWorldAppPool, and set the .NET CLR version to No Managed Code.
3. **Create a new Site**:
   * In IIS Manager, right-click on "Sites" and select "Add Website".
   * Set the Site name to HelloWorldWebApp.
   * Set the Physical path to the publish directory created earlier.
   * Set the Application Pool to HelloWorldAppPool.
   * Set the Port to a desired port (e.g., 8080).
4. **Set Permissions**:
   * Ensure the IIS App Pool identity (e.g., IIS AppPool\HelloWorldAppPool) has read and execute permissions on the publish directory.
5. **Browse the Website**:
   * Open a web browser and navigate to http://localhost:8080 (or the port you specified).
6. The web.config file is automatically generated during the publish step. Here is an example:

#<?xml version="1.0" encoding="utf-8"?>

<configuration>

<system.webServer>

<handlers>

<add name="aspNetCore" path="\*" verb="\*" modules="AspNetCoreModuleV2" resourceType="Unspecified"/>

</handlers>

<aspNetCore processPath="dotnet" arguments=".\HelloWorldWebApp.dll" stdoutLogEnabled="false" stdoutLogFile=".\logs\stdout" hostingModel="inprocess"/>

</system.webServer>

</configuration>

1. Push the code to gitlab
2. Create .gitlab-ci.yml

#.gitlab-ci.yml

stages:

- build

- publish

build:

stage: build

tags:

- ASP.NET

script:

- dotnet restore

- dotnet build --configuration Release

publish:

stage: publish

tags:

- ASP.NET

script:

- dotnet publish --configuration Release --output publish

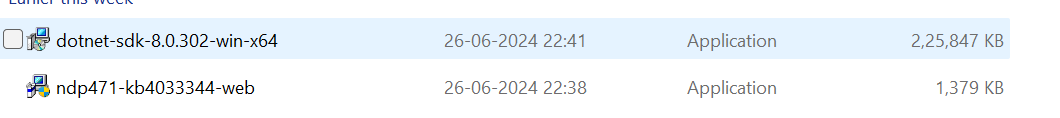
artifacts:

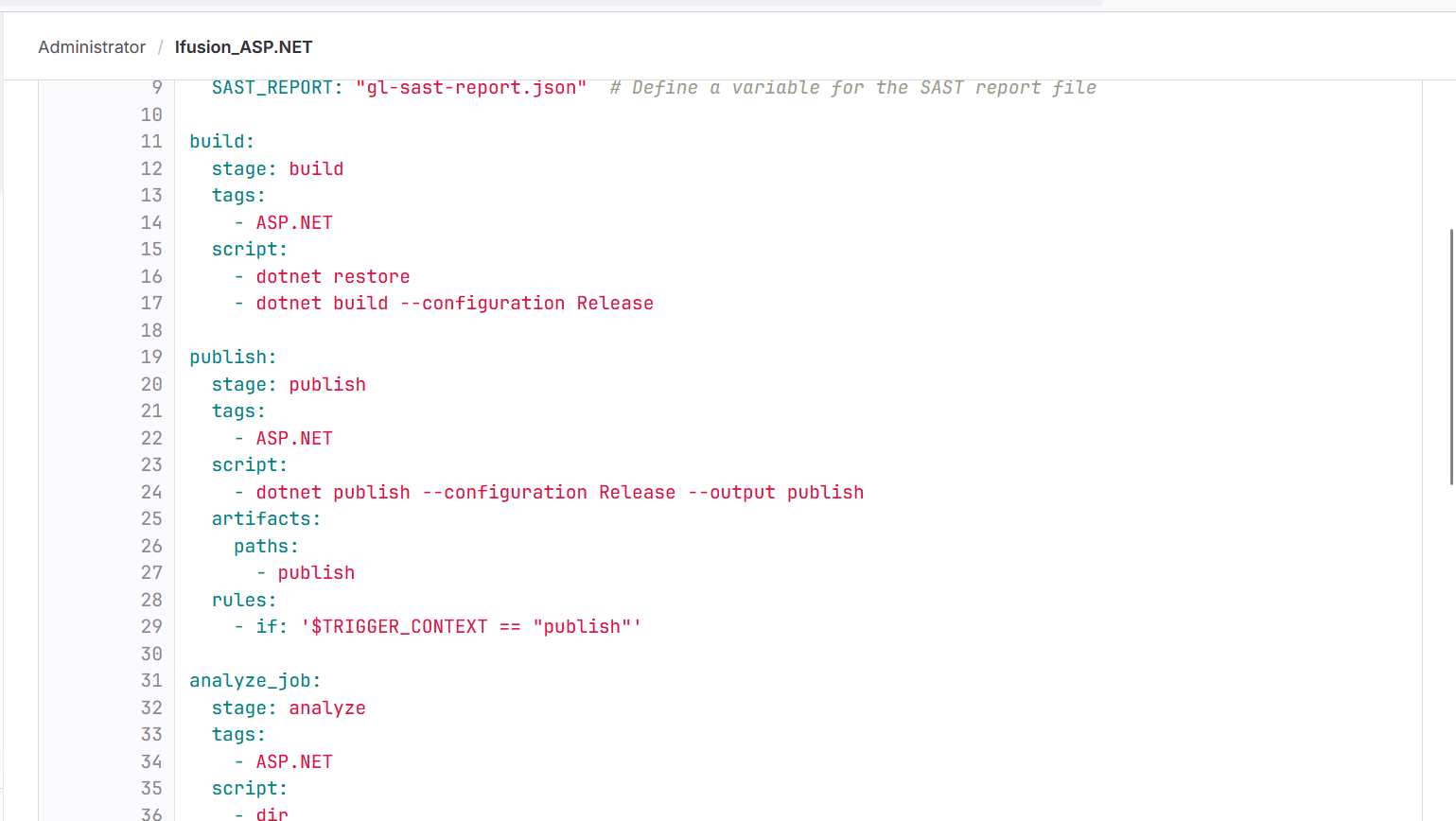
paths:

- publish

9. To build the ASP.NET required framework4.7.1 and SDK8.0 on windows runner.  
 <https://dotnet.microsoft.com/download/dotnet-framework/net471>

<https://dotnet.microsoft.com/download/dotnet/8.0>



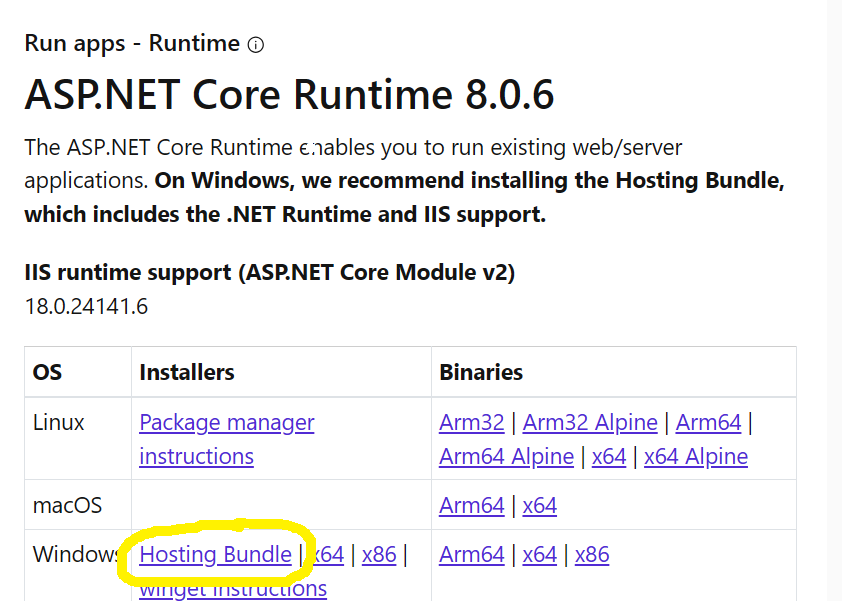




**4.NET Core Hosting Bundle Installation on target windows server**:

<https://dotnet.microsoft.com/en-us/download/dotnet/8.0>

dotnet-hosting-8.0.6-win.exe.



#dotnet –version

Restart the iis manager using the following command.

#iisreset

**On Ansible machine need to install module plugin**

**#** ansible-galaxy collection install --force community.windows  
**#** ansible-galaxy collection install --force ansible.windows

**Copy ansible playbook in gitlab.**

Ansible playbook  
#cat dep\_ifusion.yml  
---  
- name: Deploy iFusion application on Windows IIS Server  
  hosts: ifusion  
  tasks:  
    - name: Create application directory  
      win\_file:  
        path: C:\inetpub\ifusion  
        state: directory  
      # This task creates the directory where the application files will be stored.

    - name: Copy application files to the server  
      win\_copy:  
        src: /etc/ansible/playbooks2/files/  
        dest: C:\inetpub\ifusion\  
        recurse: yes  
      # This task copies the application files from the Ansible control node to the Windows server.

PATH to deploy on IIS: C:\inetpub\ifusion\  
  
  
  
  
**gitlab.yml deploy stage**  
stages:

- build

- publish

- deploy

build:

stage: build

tags:

- ASP.NET

script:

- dotnet restore

- dotnet build --configuration Release

publish:

stage: publish

tags:

- ASP.NET

script:

- dotnet publish --configuration Release --output publish

artifacts:

paths:

- publish

Deploy:

stage: deploy

tags:

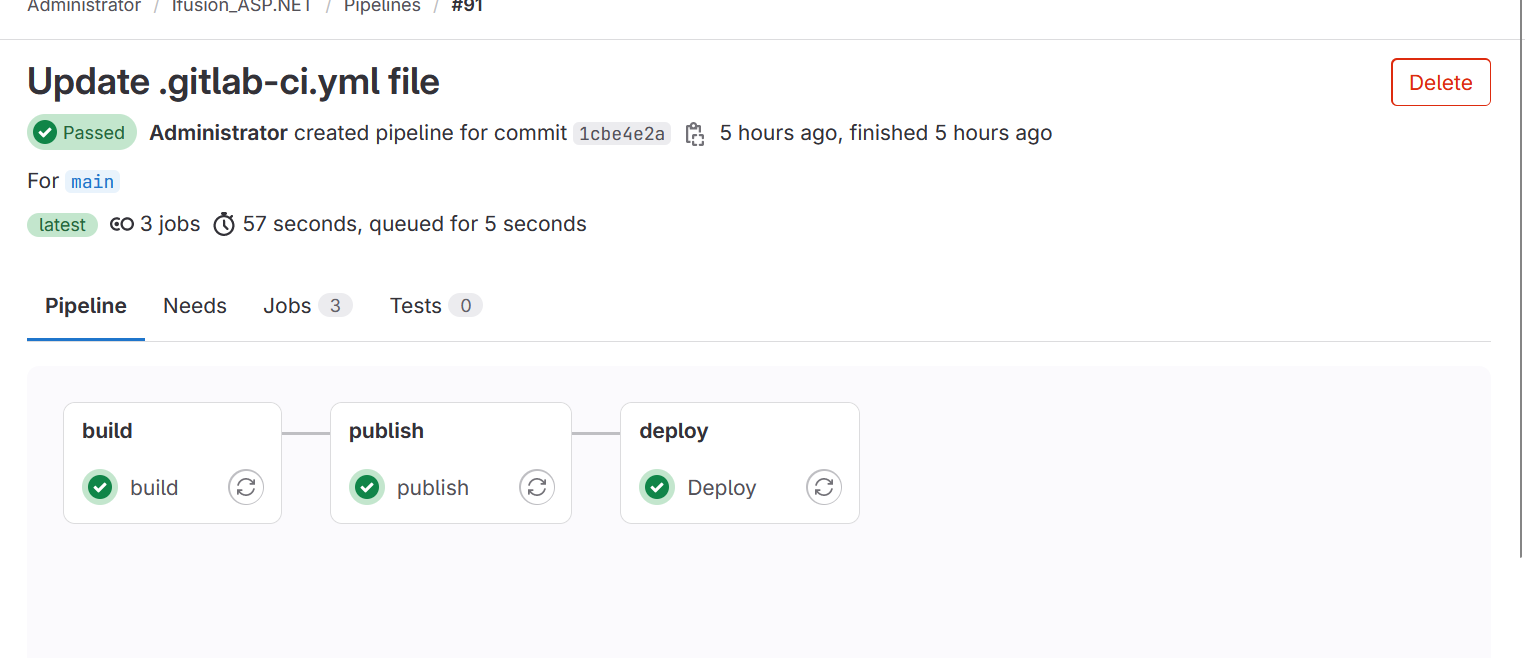
- ASP.NET

script:

- echo yes | scp -v -i $SSH\_PRIVATE\_KEY -o StrictHostKeyChecking=no -r ./publish/\* root@192.168.3.200:/etc/ansible/playbooks2

- echo yes | scp -v -i $SSH\_PRIVATE\_KEY -o StrictHostKeyChecking=no ./dep\_ifusion.yml root@192.168.3.200:/etc/ansible/playbooks2

- ssh -i $SSH\_PRIVATE\_KEY -o StrictHostKeyChecking=no root@192.168.3.200 "cd /etc/ansible/playbooks2 && ansible-playbook -vvv dep\_ifusion.yml"



A screenshot of a computer

Description automatically generated

Prerequisites:  
1. .NET SDK

2..NET Framework

3.Hosting bundle

4.WINRM

5.SSH keys

6.two collections required on ansible machine

7.powershell executor on runner