In the given code:
The test provided will fail due to the line `Assert.That(true, Is.EqualTo(false), "oops!");`, which asserts a condition that is always false. This assertion is set to fail every time the test is run.

Modified the Test to Pass:
Made changes to `TimeControllerTests.cs`

```
using NUnit.Framework;
using SuperService.Controllers;

namespace SuperService.UnitTests
{
  public class TimeControllerTests
  {
    private TimeController controller;
    private static readonly System.DateTime now = new
System.DateTime(2020, 01, 01);

    [SetUp]
    public void Setup()
    {
      controller = new TimeController(new MockClock(now));
    }

    [Test]
    public void TheTimeIsNow()
    {
      var time = controller.Get();
      Assert.That(time, Is.EqualTo(now));
    }
  }
}
```

Deploy.ps1 for Automated Run: (Saved in `super-service` folder)

```
# Define variables
$AppPath = "C:\super-service\src"
$TestPath = "C:\super-service\test"
$DockerImageName = "super-service-webapi"
$DockerContainerName = "super-service-webapi-container"
$Dockerfile = "$AppPath\Dockerfile"
```

```
# Step 1: Run automated tests
Write-Host "Running automated tests..."
dotnet test $TestPath
if ($LASTEXITCODE -ne 0) {
    Write-Host "Tests failed. Continuing deployment..."
} else {
    Write-Host "Tests passed successfully."
}

# Step 2: Package the application as a Docker image
Write-Host "Building Docker image..."
docker build -t $DockerImageName $AppPath
if ($LASTEXITCODE -ne 0) {
    Write-Host "Docker build failed. Aborting deployment."
    exit 1
}
Write-Host "Docker image built successfully."

# Step 3: Deploy and run the Docker container locally
Write-Host "Running Docker container..."
docker run -d --name $DockerContainerName -p 80:80 $DockerImageName
if ($LASTEXITCODE -ne 0) {
    Write-Host "Docker run failed. Aborting deployment."
    exit 1
}
Write-Host "Docker container is running successfully."

Write-Host "Deployment completed."
```

Dockerfile: (saved in src folder)

```
# Using the official .NET Core SDK image for the base runtime
FROM mcr.microsoft.com/dotnet/aspnet:6.0 AS base
WORKDIR /app
EXPOSE 80

# Using the official .NET Core SDK image for the build
FROM mcr.microsoft.com/dotnet/sdk:6.0 AS build
WORKDIR /src
COPY ["SuperService.csproj", "."]
RUN dotnet restore "./SuperService.csproj"
COPY . .
RUN dotnet build "SuperService.csproj" -c Release -o /app/build
```

```
FROM build AS publish
RUN dotnet publish "SuperService.csproj" -c Release -o /app/publish

# Final stage/image
FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "SuperService.dll"]
```