Truffle Suite

Archived: This tutorial has been archived and may not work as expected; versions are out of date, methods and workflows may have changed. We leave these up for historical context and for any universally useful information contained. Use at your own risk!

Update : Since this tutorial was published, we have released Ganache , a personal blockchain and a replacement to the TestRPC. We have left this tutorial unaltered, but we highly recommend checking out our Ganache Documentation This post was originally published by David Burela on his blog Burela's House-o-blog . Big thanks to David for allowing us publish it here!

I have been working on automating the compilation and testing of Ethereum solidity contracts, via the use of Truffle Suite . I've got the test results being published back into the portal, allowing me to see on each commit if my code still compiles and passes my tests.

I'm assuming you already have a Truffle project locally that you want to automate the continuous builds & testing on. Follow the tutorial on installing Truffle & TestRPC on Windows .

My final system will allow you to run "truffle test" locally to see standard test output, but will modify the test runner on the server to output it as JUnit format.

The Build system¶

The system uses the Visual Studio Team Services (VSTS) build engine to automate this. You can sign up for free, and get unlimited private Git repos.

You can have the code hosted on any Git provider. So either within VSTS itself, or GitHub, BitBucket, etc.

Prepare truffle.js¶

A pre-step is to define the test section in the truffle.js file

mocha :

{

reporter :

" spec " ,

reporterOptions :

{

mochaFile :

' junitresults . xml '

} }

Create a build agent¶

VSTS does provide hosted build agents, which are generic and can build standard .Net projects, Xamarin, etc. But because we are going to use npm packages installed globally on the box to handle the Truffle builds

- Create a new Windows VM (Can be your own hosted server, or Azure). e.g Windows Server 2016 Datacentre edition on Azure
- Install the VSTS build agent. Instructions at https://www.visualstudio.com/en-us/docs/build/admin/agents/v2-windows
- Note: DON'T select to run service as NT AUTHORITY\NETWORK, this will not work with TestRPC (needs to open ports). Run the service as another user, or NT AUTHORITY\SYSTEM
- Install chocolatey https://chocolatey.org/install
- Install these chocolatey packages

choco install git -y

choco install nodejs.install –y * Install npm packages (make sure you open a new PowerShell window so that node is in your path)

npm install -g npm

npm install -g –production windows-build-tools

npm install -g ethereumjs-testrpc

npm install -g truffle

npm install -g mocha

npm install -g mocha-junit-reporter * Restart the build agent so that all new paths are working

Configure VSTS build¶

Create a new variable with the path to where the npm global path is, for the user you installed the npm packages on above:

- variable name:npm.path
- variable value: path to npm packages e.g.C:\Users\AppData\Roaming\npm

Add 7 PowerShell tasks, and configure them like this¶

- Name: System version information
- Script:

Setting environment paths

ENV:Path

ENV:Path + “ ; ” + env:npm_path npm config set prefix env:npm_path

only needs to be set once, will update for user

DEBUG

env:path

npm list -g –depth=0

Display system information

Write-Host “ System version information ” Write-Host -nonewline “ node version : ” ; node -v Write-Host -nonewline “ npm version : “ ; npm -v Write-Host -nonewline “ npm prefix : “ ; npm prefix -g Write-Host -nonewline “ truffle : ” ; truffle version

- Name: Config transform & test clean
- Script:

remove old test results

rm .\ junitresults . xml -ea SilentlyContinue

Modify the Truffle test runner to use the JUnit reporter

Rename-Item .\ truffle . js .\ truffle_temp . js cat .\ truffle_temp . js | % { _ -replace ‘ reporter : “ spec ”‘ , ‘ reporter : “ mocha-junit-reporter ”‘ } | Out-File -Encoding ASCII .\ truffle . js rm .\ truffle_temp . js

- Name: Truffle build
- Script:

Setting environment paths

ENV:Path

ENV:Path + “ ; ” + env:npm_path

Truffle build

truffle compile

- Name: Launch TestRPC
- Script:

Setting environment paths

ENV:Path

ENV:Path + “ ; ” + env:npm_path

launch the process

echo “ launching TestRPC ” testrpcProcess = Start-Process testrpc -passthru

persist the PID to disk and display in logs

testrpcProcess . Id | Export-CliXml testrpcPID . xml cat testrpcPID . xml * Name: Run Truffle tests * Script:

Setting environment paths

ENV:Path

ENV:Path + “ ; ” + env:npm_path

Run the tests

truffle test

- Name: Shutdown TestRPC
- Other Settings: Enable “Always Run” (to make sure it is shutdown if there is an error)
- Script:

Setting environment paths

ENV:Path

ENV:Path + “ ; ” + env:npm_path

retrieve the PID and kill the entire processs tree

cat testrpcPID . xml testrpcPID = Import-CliXml testrpcPID . xml taskkill / pid testrpcPID / F / T

- Add a new Publish test result
- Test Result Format: JUnit
- Test Result Files:junitresults.xml

Future work¶

Things that I would like to add in the future:

- Figure out how to automate this on a Linux build agent (VSTS supports both Windows & Linux based build agents)
- Automate Release Management to runtruffle migrate
- to push to aBletchley test environment