Enhancing Team Productivity with Local Development Environments: A Guide

Garvin LeClaire
garvin.leclaire@gmail.com

Background

- *Developer of Apps, embedded systems*
- *Developer of HW and SW test tools/ systems*
- *Worked for Corporate, Startups and Consulting*

Agenda

- *Goals*
- *Problems/Challenges to Solve*
- *History*
- *References*

Goals

- *Reliable and repeatable development environment.*
- *Simple and easy for others to understand.*
- *Solve the works on my machine story.*
- *Ease of provisioning environment(s) for others on a team.*
- *Deliver to multiple processor platforms (i.e. amd64, arm64, GPU).*

Problems/Challenges to Solve

- *Linux is a preferred/economical OS.*
- *Windows and MacOS are preferred development environments.*
- *Not all tools are easy to coexist with other versions of itself or other tools and dependencies.*
- *Memory issues for development.*

Problems/Challenges to Solve

- *Data for reliable testing and provisioning.*

History

- *Develop on Mac/Windows.*
- *Build Shared development Linux machines.*
- *Virtual Machines (i.e. VirtualBox).*
- *Vagrant to provision environments.*
- *Docker/Containers.*

Tools for Development

- *HomeBrew (Scoop, Chocolatey).*
- *Shell (i.e. bash, zsh, PowerShell, WSL2).*
- *Terminal (iTerm2, Windows Terminal).*
- *IDE (i.e. JetBrains, Visual Studio Code).*
- *Command line build/dev tools (i.e. Maven, Yarn, NPM).*

Tools for Development

- *draw.io*
- *Diagrams (Diagram as Code)*
-

References

- *Docker*
- *Vagrant*
- *VirtualBox*
- *KubeVirt*
- *Draw.io*
- *Diagrams*