

Git によるバージョン管理入門

田中 健策（株式会社ぺあのしすてむ）

第五回（最終回）

前回までのあらすじと今回のあらすじ

前回までのあらすじ

- プルリクエストを使ったブランチのレビューとマージの自動化と記録
- Github Actions を使ったコンパイルとリリースの自動化

今日は git と直接的な関係はないものの、Github Actions に絡めて、仮想化、コンテナ化技術を少し話します。

仮想化技術とは何か

仮想化 (**virtualization**) とは、実際には存在しないものを、まるで存在しているかのように見せかけることである。

コンピュータ技術における主要な仮想化としては、仮想マシンなどがある。

仮想マシンとはコンピュータの中に、メモリを持ち、ハードディスクを持ち、画面やキーボードやマウスを持つ「コンピュータ」を丸ごとシミュレートする。

Mac マシンの中に Windows や Linux など別の OS がインストールされたマシンをシミュレートできる。

Virtualbox などをインストールすれば誰でも使える。

仮想マシンは何が嬉しいか

- 簡単に作れる（本物のコンピュータは簡単に作れない）
- 簡単にコピーできる（本物のコンピュータは簡単に以下同文）
- 簡単に捨てられる（以下同文）

なので

- たくさん作れる
- 実験が簡単にできる
- 設定などを自動化できる
- 必要な時に作って終わったら捨てられる

動かしたい環境と同じ環境を手元で作るためなどに使われる。

仮想マシンの弱点

仮想マシンはコンピュータを丸ごと使うので、色々と重い。

- 立ち上がりが遅い
- メモリをたくさん使う
- 何十 GB もの大きなファイルが必要になる
- 動き自体も遅い

これでは例えば本番では使えない。つまり、「実際に動かすときの環境を手元に作ること」ではなく、「作ったときと同じ環境で、別の場所で動かすこと」はできない。

もっと気軽に作っては壊して、実験し、そのまま持ち運びたい。そうすれば、作った時と同じ環境で、別の場所でも動かせるし、複数同時に動かすのも簡単になる。

そこで登場するのがコンテナ化技術である。

コンテナ化技術とは何か

コンテナ化とは、アプリケーションを動かすのに必要な OS の機能や環境だけをうまく仮想化する技術である。

コンピュータを丸ごと仮想化しているわけではないので、軽い。特に有名なのは **Docker** というソフトウェアで、Docker コンテナというものに必要な環境を入れて動かす。

これでアプリケーションは、どこでもほぼ同じ環境の中で動く。また既存の Docker コンテナが多数公開されており、例えば「テスト用のメールサーバが欲しい」が必要なら「それ用の Docker コンテナ」をダウンロードして起動するだけで、すぐに使える。

また、新しい Docker コンテナを作るための元となるコンテナも多数公開されている。

Github Actions で Docker を動かす

Github Actions では、Dockerfile というファイルを作成することで、既存の Docker コンテナをダウンロードし、調整し、起動することができる。

それによって、複数の action が同時に起動してもお互いに、まるで違うコンピュータの上で動いているように、干渉し合わず動かすことができる。

また、何回作動させても、同じ環境から始まるので、1 回目と 2 回目で結果が変わらない。これをこの操作はべき等性 (**idempotent**) を持つ、という。これにより、テストがしやすく、問題も解決しやすい。

なぜ仮想化技術が重要なのか

Git と無関係な仮想化技術の話をなぜしたかというと、これを習得すると、気軽に計算機環境が手に入るからである。

紙と鉛筆が数学の最強の道具であるのは、気軽に書いては捨てられるから。コンピュータはまだそこまで柔軟ではない。

まだまだ様々な技術の開発が必要ではあるが、仮想化技術が一つの重要な役割を果たすことは確実である。

今回 Github actions を皆に使ってもらい、Dockerfile を皆に書いてもらうことはできなかったが、「数学用の精算機を気楽に作っては捨てて実験でき、別の場所で同じ動きを再現できたら便利」という「お話」は覚えてもらいたい。

というところで、この講義は終わり。

道具こそ人間を人間たらしめてるものの一つなので、道具をうまく使って人生と数学を生きていってください。