Aurora Deployment TDD

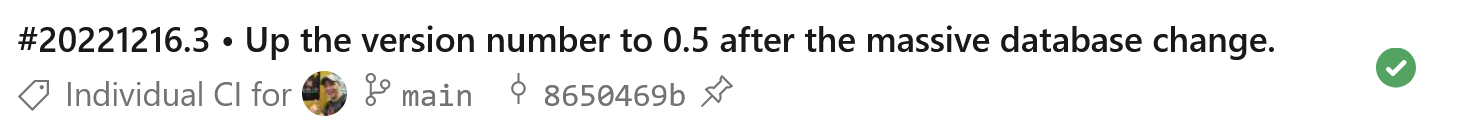
by Ward Childress for Aurora version 0.5.

There are hardly any modern MUDs, Multi-User Dungeons. This era flourished in a pre-MMO world, when access to a terminal and the imagination necessary to explore a massive RPG world using just words and crude inputs were still in supply. But these games were innovative and brilliant. What would it take to build one with today’s tools? This document will capture, at least, the Microsoft Azure-side of deploying and running a MUD in the cloud.

fruitofwisdom.ddns.net, port 6006 – the live version of Aurora, from the “main” branch in GitHub

A telnet client like [PuTTY](https://www.putty.org/) or, even better, a MUD client like [Mudlet](https://www.mudlet.org/) is required.

# Azure Pipeline



The bulk of the work for building and deploying happens in Azure DevOps. Since this project is mainly hosted in GitHub, an Azure Pipeline is used to build and deploy the build artifacts to our virtual machine. For each commit in the Git repository, a build is automatically started and, if successful, a copy of the built Aurora executable is downloaded to the VM ready to be executed. The steps are:

1. On a commit trigger, a ‘BuildAndDeploy’ stage is ran.
2. A release build is started with MSBuild targeting a Windows environment.
3. If the build is successful, the output is copied to the artifact staging directory and published to a container.
4. On deployment, the artifacts are downloaded to the build directory.

At present, the executable is not started at the end of deployment, due to needing a way to cleanly shut down the server and because starting an executable (or batch file or PowerShell script) can’t happen asynchronously with a pipeline.

# Azure Environment



There is a virtual machine environment that is the deployment target for the pipeline and it is on this VM that the current build is running. This is managed by hand so it can be maintained without being affected by every commit and because there’s no easy way to restart automatically from the pipeline.

Release builds are copied out of the build directory on the Windows VM and placed on the desktop in a folder named after the version number. The Aurora server is then launched by hand and loaded with the game that is hosted at the live environment with port 6006 open to public telnet access.

# Other Azure Resources

In addition to the Azure DevOps services, other Azure resources are needed including:

* auroravm – A Windows 10 Pro virtual machine (2 vcpus, 8 GiB memory).
* auroravm-ip – The public IP address for the VM.
* auroravm-nsg – The network security group for the VM with port 6006 open.