Zwift Deployment

William Kelly and Ron Pedde

April 2014

Agenda

Goals

At the end of the session, we're hoping that you will be able to:

- Use ansible proficiently
- Deploy a zwift cluster on Rackspace public cloud or physical infrastructure using the ansible-zwift configuration tool.
- Operate a swift cluster effectively with a good understanding of available operational tools.

Zwift

- ZeroVM allows users to run abitrary code safely using SFI from google's native client.
- Swift is an object storage system used by Rackspace's cloud files and similar to Amazon's S3.
- Zwift combines Swift and Zerovm. Now we can run user provided arbitrary code directly on the object servers using a REST api!

Ansible

- Ansible is a configuration management system.
- Agentless (remote configuration via ssh)
- YAML based configuration language with jinja2 templating.
- Servers are assigned composable roles along with role specific variables to describe their ideal configuration.

Exercise 1: Installing Ansible

[Installing Ansible](https://github.com/ludditry/zwift-training/blob/master/operational/installing-ansible.md)

This exercise explains how to install and configure ansible.

Ansible Terminology

- *Modules* provide the core functionality of ansible – they perform actions based on yaml configuration. The core list of ansible modules is avaiexercisele on the ansible website.
- *Roles* are user created. They are a collection of yaml files that describe how to configure a server to perform a particular function.
- *Variables* can be specified in roles, through ansible's automatic discovery mechanism, in the inventory as global defaults or against a specific host, in playbooks, or at the command line at runtime, and the value of a variable specified in multiple places is resolved in the listed order, with the command line winning..
- *Playbooks* contain specific descriptions of actions to take to accomplish a particular task (configure a zwift cluster, for example).

Ansible directory layout

- The *inventory* directory contains information on the devices that ansible will be configuring.
- The aptly named *roles* directory contains roles.
- The *library* directory contains extra ansible modules to enable additional functionality.

Exercise 2: Getting Familiar with Ansible

Getting Familiar with Ansible

In this exercise, we add an inventory file to ansible and run some simple ansible commands against the servers in the inventory file.

Exercise 3: Installing Zwift

Installing Zwift

In this exercise, we install zwift.

Mechanisms for using Zwift

- API extensions to the swift api
- The zwift-ui web user interface
- Command line client

Methods of execution

- Zerovm can run objects already uploaded to swift or be used as a standalone execution environment.
- Servlet descriptors are used to indicate which objects should be invoked, which objects should be provided as input, and where the output of the job should go.
- Jobs can be initiated by posting a servlet description or a tarball that includes data as well as a servlet description.
- Applications can be registered for mime types to provide default “openers”, at which point GET requests can be used against objects of that mime type, resulting in the execution of the provided handler.

Exercise 4: Using zwift-ui

Using zwift-ui

In this exercise, we will log in to the zwift ui, load some sample data, modify the manifests appropriately for your environment, and run some things!