vagrant-ark

Vagrant is a tool that allows easily configurable, reproducible and portable development environments. It uses configuration files to manage a Virtual Machine (VM), and integrates with tools such as Puppet (<https://puppetlabs.com/)> or Ansible (<http://www.ansible.com/)> to provision the VM. We have used Vagrant and Puppet to create a VM on an Ubuntu 14.04 LTS base image, with Tomcat7, SLAPD and MySQL configured to create a ready-to-go Ark server.

These instructions have been tested on OSX, Ubuntu Linux and Windows. Note that on Windows, the files under the puppet directory may appear to be formatted strangely. This is due to these files requiring Unix line endings

Before attempting to using vagrant-ark, you should be comfortable with the following topics:

* Using a command prompt
* Virtual Machines and Virtualization
* Version Control Systems, specifically git

The vagrant-ark repository contains a submodule of The Ark Github repository (<https://github.com/The-Ark-Informatics/ark)>, and is pulled down with git clone --recursive. If you didn’t clone using the recursive flag, you will need to run git submodule update --init --recursive to pull The Ark repository.

# Prerequisite Software

* Virtualbox - <https://www.virtualbox.org/>
* Vagrant - <https://www.vagrantup.com/>
* Git - <http://git-scm.com/>

# Setup

1. Recursively clone the vagrant-ark repository
   1. git clone --recursive <https://github.com/The-Ark-Informatics/vagrant-ark.git>
2. Run the Virtual Machine
   1. vagrant up

The first time vagrant up is run, Vagrant will automatically provision the VM, installing and configuring all dependencies needed to run The Ark.

# Usage

Once the VM is running, you can SSH into it via the vagrant ssh command from within the project directory. To access The Ark, you can go to <http://192.168.33.10:8080/ark> in your browser.

For development use, there are two approaches to working with vagrant-ark.

1. You can work directly on the code in the submodule
2. or you can work on copy of The Ark outside of vagrant-ark and copy the compiled WAR into the shared webapps directory.

For approach 1, the code in src/ark will be recompiled and deployed if the image is provisioned again (i.e with vagrant up --provision), however database changes will persist throughout provisions.

With approach 2, Tomcat will automatically detect when the WAR file in the shared webapps directory has been modified, and will automatically redeploy it.

You should note that new or modified database patches will **NOT** be applied using either of the methods above, and should be managed independently.

# Users and Passwords

The username and passwords for the services configured in vagrant-ark are listed below.

|  |  |  |
| --- | --- | --- |
| Service | Account/Username | Password |
| The Ark | [arksuperuser@ark.org.au](mailto:arksuperuser@ark.org.au) | Password |
| Linux User | vagrant | vagrant |
| Slapd | cn=admin,dc=the-ark,dc=org,dc=au | password |
| MySQL | root | password |