DevOps Crash Course

by Oleksii Yakivchik

softserve

Vagrant

- Vagrant is a tool for building and managing virtual machine environments in a single workflow. With an easy-to-use workflow and focus on automation, Vagrant lowers development environment setup time, increases production parity, and makes the "works on my machine" excuse a relic of the past.
- Vagrant provides easy to configure, reproducible, and portable work environments built on top of industry-standard technology and controlled by a single consistent workflow to help maximize the productivity and flexibility of you and your team.

Vagrant

- To achieve its magic, Vagrant stands on the shoulders of giants. Machines are provisioned on top of VirtualBox, VMware, AWS, or any other provider. Then, industry-standard provisioning tools such as shell scripts, Chef, or Puppet, can automatically install and configure software on the virtual machine.

Vagrant

- If you are a developer, Vagrant will isolate dependencies and their configuration within a single disposable, consistent environment, without sacrificing any of the tools you are used to working with (editors, browsers, debuggers, etc.). Once you or someone else creates a single Vagrantfile, you just need to vagrant up and everything is installed and configured for you to work. Other members of your team create their development environments from the same configuration, so whether you are working on Linux, Mac OS X, or Windows, all your team members are running code in the same environment, against the same dependencies, all configured the same way.

Vagrant

- If you are an operations engineer or DevOps engineer, Vagrant gives you a disposable environment and consistent workflow for developing and testing infrastructure management scripts. You can quickly test things like shell scripts, Chef cookbooks, Puppet modules, and more using local virtualization such as VirtualBox or VMware. Then, with the same configuration, you can test these scripts on remote clouds such as AWS or RackSpace with the same workflow.

Start work with Vagrant

- To start working with Vagrant you should:
- Install the latest version of Vagrant.
- Install VirtualBox

Try Vagrant

- Initialize Vagrant: vagrant init hashicorp/bionic64
- Start virtual machine: vagrant up
- Destroy virtual machine: vagrant destroy

Vagrantfile

- After Vagrant initialization it will create a Vagrantfile in your current directory
- Vagrantfile is the file which describes state of your machine, including amount of RAM, CPU cores, networks, provisioning details, etc

Vagrantfile basics

- Vagrant.configure("2") do |config|
- #Description of machine…
- end

softserve

Vagrantfile basics

- Vagrant.configure("2") do |config|
- config.vm.box = "centos/7"
- end

softserve

Vagrantfile basics

- Networking
- config.vm.network "forwarded_port", guest: 80, host: 8080
- config.vm.network "private_network", ip: "192.168.33.30"
- config.vm.network "public_network"

Vagrantfile basics

- Synced folders
- Share an additional folder to the guest VM. The first argument is
- the path on the host to the actual folder. The second argument is
- the path on the guest to mount the folder. And the optional third
- argument is a set of non-required options.

- config.vm.synced_folder "../data", "/vagrant_data"

Vagrantfile basics

- config.vm.provider "virtualbox" do |vb|
- vb.memory = "4096"
- vb.cpus = "2"
- end

softserve

Vagrantfile basics

- Provisioning
- config.vm.provision "shell" do |s|
- s.path = "script.sh"
- s.args = ["testuser", "passw0rd"]
- end

softserve