PRIVIOUS SPRINTS

1. **SSH Access / Users / Keys:2**

As an engineer, I would like to have every team member have access to every server in all environments so that I can be confident that I will be able to resolve an issue with a specific server if nobody else is available.

Master key has been created and added to a private github repo only accessible by team leads.

Acceptance Criteria:

* Implementation plan documented for properly using keypair to ensure access to all servers via master key.

Puppet responsible for creating users, adding keys, and configuring sudo access

**2) Puppet - Autocleanup Reports 2**

As an engineer, I would like the Report files generated by Puppet to be cleaned up regularly by the Puppet process so that they don't fill up the system desk.

Acceptance Criteria:

* Puppet configured to purge old report files after 2 weeks

Old puppet reports removed from server

**3) Investigate Cassandra Puppetization 3**

As a project owner, Investigate how we can Puppetize Cassandra, and deploy a new server to the existing cluster using puppet.

Acceptance criteria

* Investigate the best practices
* Document the procedures or steps

Get Reviewed by PO

**4) Sensu/Uchiwa - Autostart on Reboot 1**

As an engineer, I would like Sensu and Uchiwa to start correctly upon boot so that I can be confident that services will continue to function after recovery from an interruption.

Acceptance Criteria:

Sensu and Uchiwa start correctly after a server reboot

**5) POC of Cassandra Puppetization 3**

As a product owner, i would like to see a POC of Puppetizaton of the Cassandra Instances. When a new node is added, it should be completely configured through puppet and added to the cluster.

Acceptance Criterion

* POC for Cassandra Puppetization

Demo a new node added to the cluster, gets completely configured by puppet.

**6) Review Vulnerability Report 1**

As a technical owner, I would like the vulnerability report of the Usage Meter Desktop App servers reviewed for outstanding work so that I can feel confident that vulnerabilities are addressed

https://wiki.io.comcast.net/pages/viewpage.action?pageId=435392409

Acceptance Criteria:

* vulnerability report reviewed for outstanding work

user stories filed for work to be completed

**7) Sensu - Design/Integrate Email 3**

As an Engineer, I want Sensu to send meaningful email alerts according to our guidelines for alerting.

Acceptance Criteria:

* Email integration with Sensu complete according to guidelines for alerting (e.g. project owners get emails, not whole team)
* Use a mail or mailer handler instead of a pipe handler followed by calling a script.

Email alerts are well formatted and easily readable by engineers

**8) Investigate S3 Integration 3**

As technical owner I would like the Usage Meter App platforms included in the S3 system so they can be tested for configuration and system vulnerabilities.

Acceptance Criteria:

Usage Meter App servers are configured in S3 and reporting correctly

**9) IPDR automated file delivery 3**

As an engineer, I would like to have the IPDR Vertica Rollup files for 1 CMTS delivered automatically and placed on the ubbaadb-wc-1p server so that they can be ingested into the reporting datastore.

Acceptance Criteria:

* Work with IPDR team to have daily vertica file delivered to UBB AA server
* Create cron job to clean up rollup files after 7 days

The following CMTSs should have data delivered:

|  |
| --- |
| acr01.a2atlanta.ga.atlanta.comcast.net |
| acr02.a2atlanta.ga.atlanta.comcast.net |
| acr03.a2atlanta.ga.atlanta.comcast.net |

**10) DPI automated file delivery 3**

As an engineer, I would like to have the DPI files for 1 CMTS delivered automatically and placed on the ubbaadb-wc-1p server so that they can be ingested into the reporting datastore.

Acceptance Criteria:

* Work with DPI team to have daily file delivered to UBB AA server

Create cron job to clean up rollup files after 7 days

**11) Upgrade Apache and Tomcat 3**

As a Project Owner, need to update Apache and Tomcat of USMD servers to the latest version so that they are safe from any Vulnerabilities.

Acceptance Criterion:

* Update the Apache version from 2.2.15 to 2.2.31
* Update the Tomcat Version from 7.0.53 to 7.0.59

New version needs to be validated in QA and planned for production

**12) Magneto Puppet Project Created 2**

As an engineer, I would like a project created in Magneto to manage the Puppet infrastructure for the Accuracy Assurance platform so that I can have a centralized puppet infrastructure for integration.

Acceptance Criteria:

* Magneto project requested and created for Accuracy Assurance platform
* Access given to primary team members (Adam, Koushik, Libo)
* Built with 3 branch configuration (master = prod)
* Part of CPAP project team
* Configured to use librarian-puppet-simple, configured per https://wiki.io.comcast.net/display/ISE/Local+Puppet+Development+with+Test+Kitchen

Demo to PO

**13) Puppetize DB Nodes 3**

As an engineer, I would like to have the two DB nodes puppetized so that their configuration can be managed automatically.

Acceptance Criteria:

* Puppet installed on ubbaadb-wc-1p.sys.comcast.net and ubbaadb-wc-2p.sys.comcast.net
* Nodes configured to communicate with Magneto (firewall requests, etc)

Demo to PO

**14) Puppetize Cassandra instance 2**

As an engineer, I would like to have the Cassandra cluster configuration puppetized so that I can easily add and remove nodes from the cluster.

Acceptance Criteria:

* Cassandra cluster configuration created in UBB AA Puppet repo
* Nodes automatically join cluster after Puppet is executed

Documentation created for adding/removing nodes

**15) Jenkins/Puppetize Web UI 3**

As an engineer, I would like to have the web UI built using Jenkins and deployed using Puppetized so that new software can be deployed to a web server.

Acceptance Criteria:

* Web UI packages built out of Git using Jenkins
* New packages built on every git commit
* Puppet configuration created to manage deployments of UI packages to Web UI server
* DNS entry requested for UI server (ubbaa.comcast.net)
* Documentation updated to include build information and UI access info

Demo to technical owner

**16) Puppetize Apache Spark 3**

As an engineer, I would like the Apache Spark configuration for the cluster maintained by Puppet so that I can add and remove nodes to/from the cluster and have them automatically configured to join with minimal manual intervention.

Acceptance Criteria:

* Apache Spark installed via puppet
* Spark configured in cluster mode using standalone cluster mode for management
* New nodes automatically join cluster

Demo to TO

**17) Puppetize SNMP tools 3**

As an engineer, I would like to have the SNMP tools puppetized so that they are automatically deployed to and executed on the SNMP nodes so they can be integrated with the platform and start collecting information.

https://github.comcast.com/akaufm200/ubbaa/tree/master/Poller

Acceptance Criteria:

* SNMP tools automatically deployed to SNMP servers from puppet
* 1 poller service per node executed on startup
* 1 listener service executed per node on startup
* 1 schedule service executed on startup on master node
* Process documented in Confluence
* Demo to TO

**18) SNMP Node -> DB Node Firewall Access 1**

As an engineer, I would like to validate that the SNMP node in the Agile network is able to access the DB node in the black zone so that I can write SNMP data to the DB.

Acceptance Criteria:

* Able to communicate with Cassandra from UBB SNMP nodes (96.119.80.137 and 96.119.80.159) via SSH and Cassandra (Ports 23 and 9160)

Demo to TO

**20) Build/Package Spark + Cassandra Tools 5**

As an engineer, I would like to have the Spark and Cassandra tools built and packaged through Jenkins and made available to be deployed via Puppet so that I can automatically deploy software updates to my cluster.

Acceptance Criteria:

* Build project created for Spark and Cassandra tools
* Builds run automatically on git changes
* Packages pushed to repository for installation via puppet
* Package version number updates based on change #
* Documentation updated to detail process for building packages

Demo to technical owner

**21) Production Plan for Tomcat, Apache Upgrades 1**

As a Project Owner, come up with a plan to update Apache and Tomcat of USMD servers to the latest version so that they are safe from any Vulnerabilities.

Acceptance Criterion

* To Document the steps and procedures

To get the date and time setup for the upgrades.

**22) Puppetize RabbitMQ on SNMP nodes 3**

As an engineer, I would like to have RabbitMQ configured on the SNMP nodes so that I can reliably use the service.

Acceptance Criteria:

* RabbitMQ running on SNMP nodes
* Rabbit starts automatically on reboot
* Rabbit clustered between both SNMP nodes
* Rabbit configuration managed by puppet
* Usage documented in Confluence
* Demo to TO

Note: Need to look into firewall request.

**23) Puppetize Rundeck 5**

As an engineer I would like to puppetize Rundeck so that it can be deployed consistently in the production environment.

Acceptance Criteria:

* One Rundeck node deployed in each datacenter (WC and HO)
* Configuration deployed using best practices for availability (Primary/Secondary, Active-Active, etc)
* Rundeck able to communicate with Spark, Cassandra & SNMP nodes
* Validate Firewall requests
* Rundeck able to execute test Spark jobs
* Rundeck able to execute test shell jobs
* GLSB created for rundeck service
* DNS entry created for cuantos-rundeck.comcast.net

Demo to TO

**24) Configure Sensu Monitoring 3**

As an engineer, I would like the services on the UBBAA systems to be monitored using Sensu so that I can be confident that all of the expected services are running and that the systems are behaving as desired.

Acceptance Criteria:

* CPU/Memory/Disk (/, /log, /app partitions) status monitored
* Standard processes monitored by node type
  1. SNMP Services (Poller, Listener)-Check for Green zone connection "SMOJO"
  2. Cassandra
  3. Spark
  4. Play UI
  5. Rundeck
* Alerts posted to #UBBAA\_ALERTS slack channel

Alerts emailed to UBB team

**25) Puppet - Configure Green Zone DSCP Markings on SNMP Nodes 1**

As an engineer, I would like the automatic configuration of SNMP nodes to include the attached script to ensure the configuration of Green zone DSCP markings on the SNMP nodes (Agile) so that I can use services like SMOJO on the green zone.

Acceptance Criteria:

* Attached script included in SNMP node deployments
* Executed automatically on startup

Able to curl http://smojo.cable.comcast.com/query from SNMP node

**26) Puppetize Atlas RPMS for Cassandra and Spark Tools 2**

As an engineer, I would like to have the new Cassandra and Spark RPMS deployed automatically to the DB and PRC nodes so that I can get updates to RPMs as available automatically.

Acceptance Criteria:

* Atlas repo configured on DB and PRC nodes as YUM repo
* New RPMS able to be pushed for DB and PRC nodes
* Template files updated to deploy to correct application paths

Demo to TO

**27) Create Web UI RPM 2**

As an engineer, I would like the Web UI package to be deployed as an RPM so that I can better control packages and updates to the servers.

Acceptance Criteria:

* RPM contains JARs and scripts required for production deployment, as well as sample configuration files
* Environment configuration maintained in Puppet
* Able to be installed on a test VM
* Build process updated to automatically increment version # based on git commit (see SparkTools project and ask Adam)

Demo to TO

**28) Add monitoring for incomplete ingestion 2**

As an engineer, I would like to be alerted when ingestion fails for a specific file so that I can quickly investigate and resolve the failure.

Acceptance Criteria:

* Sensu monitor configured to monitor for .csv and .dat files in /app/aupm\_service/work that have been in place for more than 1 hour
* Alerts sent to UBB team, including file name that is waiting (depending on functionality of sensu)

Demo to TO

**29) Monitoring for daily rollup 2**

As an engineer, I would like to monitor for daily rollup data so that I can validate that all expected data is available.

Acceptance Criteria:

* Sensu sends alarm when data for SNMP, DPI, or IPDR is not available in daily\_rollup table for previous day
* Checks at 9AM ET every day
* Alert indicates which datasets are missing data
* Demo to TO

Dependancy on  US674360: Add aggregate totals to daily usage report

**30) Puppetize SNMP Rabbitmq monitoring 2**

As an engineer, I would like the Rabbitmq service moniotred so that I can know when the process is down or the cluster status is bad.

Acceptance Criteria:

* Sensu to install Rabbitmq plugins

Monitoring enabled for Rabbitmq service and cluster

**31) Stand up QA Environment 3**

As an engineer, I would like to have a full QA environment stood up to replicate the functionality of the production environment so that I can test changes without impacting production.

Acceptance Criteria:

* New QA nodes stood up for all node types
  1. Database (Cassandra)
  2. Processing (Spark)
  3. Web UI
  4. SNMP Poller
  5. Job Scheduling (Rundeck)
* All applications configured to run with qa-specific configurations (isolated environments, \_test table names, etc)

Systems added to Sensu Monitoring

**32) Include @here in slack alerts 1**

As an engineer, I would like to be alerted in the slack channel on any sensu alerts investigate and resolve the failure.

Acceptance Criteria:

Slack Alerts to include @here to notify all the users in the channel.

**33) Magneto Manifest Cleanup / Organization 3**

As an engineer, I would like to have the Magneto Puppet repository cleaned up so that it makes more logical sense and is easy to understand.

Acceptance Criteria:

* Manifests and resources named with descriptive terms that are easy to understand
* Repository layout is consistent across all components/node types

Configuration files dynamically configured to allow for environment-based specification of values

**34) Configure Jenkins to push artifacts to QA and Prod 2**

As an engineer, I would like to have the new artifacts build by jenkins to be pushed without changing the post build scripts between both QA and Prod Environment, so that both the QA and Prod has same generalized scripts.

Acceptance Criteria:

Post build scripts to be configured in a way that both QA and Prod has the same scripts, but the post build artifacts should be deployed to the respective atlas repository locations.

**This space has moved to** [**https://wiki.sys.comcast.net/display/ISE/**](https://wiki.sys.comcast.net/display/ISE/)**.**

**Everything here is now read-only.**

**You can** [**click here**](https://confluence.atlassian.com/doc/confluence-5-9-release-notes-785333580.html) **to access the Confluence 5.9.5 documentation.**

**You can** [**click here**](https://confluence.atlassian.com/doc/confluence-documentation-home-135922.html) **to access the Confluence 5.9.5 release notes to learn about new features and system improvments in the updated Confluence software.**

Deprecated since 3.4. To be removed in a future version of Confluence; use AJS.Confluence.getContextPath() include system css resources [if IE]> <link type="text/css" rel="stylesheet" href="/s/2036/44/2/\_/download/superbatch/css/batch.css?ieonly=true" media="all"> <![endif] [if IE]> <link type="text/css" rel="stylesheet" href="/s/2036/44/1.0/\_/download/batch/confluence.web.resources:view-comment/confluence.web.resources:view-comment.css?ieonly=true" media="all"> <![endif] [if IE]> <link type="text/css" rel="stylesheet" href="/s/2036/44/1.0/\_/download/batch/confluence.web.resources:aui-forms/confluence.web.resources:aui-forms.css?ieonly=true" media="all"> <![endif] end system css resources Deprecated since 3.4. To be removed in a future version of Confluence; use atl.header include system javascript resources end system javascript resources

* Skip to content
* Skip to breadcrumbs
* Skip to header menu
* Skip to action menu
* Skip to quick search

Make sure to use one of following id's with banner to avoid custom CSS in VIDSS site. https://wiki.io.comcast.net/display/VIDSS/National+Dashboards+Style+Sheet infdev-announce infdev-announce-warning infdev-announce-note infdev-announce <div class="infoMessage" id="infdev-announce-info"><span style=”font-weight:bolder;”>Confluence will have database maintenance performed starting Tuesday October 8, 2013 7:00 PM Eastern. The maintenance will run through Wednesday October 9, 2013 7:00 AM. The service may be interrupted or in read only mode during this time..</span> <a href="http://sams.cable.comcast.com/cgi-bin/arnie.cgi?OPT\_M=CM\_PRINT&TICKET=CM000863298">CM000863298</a> has been created for this maintenance. This maintenance is part of our stabilization and upgrade effort. </div>

[DEAA Team Now Supports Confluence!](http://deaa.sys.comcast.net/pages/viewpage.action?pageId=1933998) All requests for Confluence support can be made by [submitting a ticket here](http://tkts.sys.comcast.net/servicedesk/customer/portal/19/group/386).

Quick nav appears here

* Browse
  + [Pages](https://wiki.io.comcast.net/pages/listpages.action?key=ISE)
  + [Blog](https://wiki.io.comcast.net/pages/viewrecentblogposts.action?key=ISE)
  + [Labels](https://wiki.io.comcast.net/labels/listlabels-heatmap.action?key=ISE)
  + [Attachments](https://wiki.io.comcast.net/spaces/listattachmentsforspace.action?key=ISE)
  + [Bookmarks](https://wiki.io.comcast.net/spaces/space-bookmarks.action?spaceKey=ISE)
  + [Mail](https://wiki.io.comcast.net/spaces/viewmailarchive.action?key=ISE)
  + [Forums](https://wiki.io.comcast.net/spaces/forums/spaceForums.action?key=ISE)
  + [Advanced](https://wiki.io.comcast.net/spaces/viewspacesummary.action?key=ISE)
  + [People Directory](https://wiki.io.comcast.net/peopledirectory.action)
  + Keyboard Shortcuts
  + Confluence Gadgets
* Surya Maddala
  + [Update Status…](https://wiki.io.comcast.net/display/~smadda001c)
  + [Create Personal Space](https://wiki.io.comcast.net/spaces/createpersonalspace.action)
  + [Recently Viewed](https://wiki.io.comcast.net/users/viewuserhistory.action)
  + [My Communities](https://wiki.io.comcast.net/users/bubbles/user-community-profile.action)
  + [Profile](https://wiki.io.comcast.net/users/viewmyprofile.action)
  + [Network](https://wiki.io.comcast.net/users/viewfollow.action?username=smadda001c)
  + [Status Updates](https://wiki.io.comcast.net/display/status/smadda001c/list)
  + [Labels](https://wiki.io.comcast.net/users/viewmylabels.action)
  + [Watches](https://wiki.io.comcast.net/users/viewnotifications.action)
  + [Drafts](https://wiki.io.comcast.net/users/viewmydrafts.action)
  + [Settings](https://wiki.io.comcast.net/users/viewmysettings.action)
  + [Log Out](https://wiki.io.comcast.net/logout.action)
* [Dashboard](https://wiki.io.comcast.net/dashboard.action)
* [Internet Systems Engineering](https://wiki.io.comcast.net/display/ISE)
* [Internet Systems Engineering - Home](https://wiki.io.comcast.net/display/ISE/Internet+Systems+Engineering+-+Home)
* [Customer Protection & Anti Piracy](https://wiki.io.comcast.net/pages/viewpage.action?pageId=90803226)
* Local Puppet Development with Test Kitchen

\#header

* Tools
  + [Attachments (0)](https://wiki.io.comcast.net/pages/viewpageattachments.action?pageId=454574588)
  + [Page History](https://wiki.io.comcast.net/pages/viewpreviousversions.action?pageId=454574588)
  + [](https://wiki.io.comcast.net/pages/mailpage/mailpage.action?pageId=454574588)
  + [E-mail](https://wiki.io.comcast.net/pages/mailpage/mailpage.action?pageId=454574588)
  + [Restrictions](https://wiki.io.comcast.net/pages/viewinfo.action?pageId=454574588)
  + [Favourite](https://wiki.io.comcast.net/labels/addfavourite.action?entityId=454574588&atl_token=ccdcbb2139c1badc18702475cef55562d5ce1099)
  + [Watch](https://wiki.io.comcast.net/pages/addpagenotification.action?pageId=454574588&atl_token=ccdcbb2139c1badc18702475cef55562d5ce1099)
  + [Stop Watching](https://wiki.io.comcast.net/pages/removepagenotification.action?pageId=454574588&atl_token=ccdcbb2139c1badc18702475cef55562d5ce1099)
  + [](https://wiki.io.comcast.net/notificationpreferences/addpagefamilynotification.action?pageId=454574588)
  + [Watch Page Family](https://wiki.io.comcast.net/notificationpreferences/addpagefamilynotification.action?pageId=454574588)
  + [Info](https://wiki.io.comcast.net/pages/viewinfo.action?pageId=454574588)
  + [Link to this Page…](https://wiki.io.comcast.net/pages/viewinfo.action?pageId=454574588)
  + [View in Hierarchy](https://wiki.io.comcast.net/pages/listpages-dirview.action?key=ISE&openId=454574588#selectedPageInHierarchy)
  + [View Wiki Markup](https://wiki.io.comcast.net/pages/viewpagesrc.action?pageId=454574588)
  + [Watchers](https://wiki.io.comcast.net/pages/watchers/pagewatchers.action?pageId=454574588)
  + [Favourites](https://wiki.io.comcast.net/pages/favourites/pagefavourites.action?pageId=454574588)

[Local Puppet Development with Test Kitchen](https://wiki.io.comcast.net/display/ISE/Local+Puppet+Development+with+Test+Kitchen)

Start restrictions section End restrictions section

Skip to end of metadata

* Added by [Ryan Van Antwerp](https://wiki.io.comcast.net/display/~rvanan200), last edited by [Ryan Van Antwerp](https://wiki.io.comcast.net/display/~rvanan200) on Jan 06, 2016  ([view change](https://wiki.io.comcast.net/pages/diffpages.action?pageId=454574588&originalId=454575862))

Go to start of metadata

wiki content

Executive Summary

This tutorial will outline how to perform local Puppet manifest development using a number of virtualization and automation tools. This will allow Puppet manifests to be tested locally before pushing them to MagNETO to be tested on a QA cluster. This tutorial will utilize the following tools:

* [VirtualBox](https://www.virtualbox.org/)
* [Vagrant](https://www.vagrantup.com/)
* [librarian-puppet-simple](https://github.com/bodepd/librarian-puppet-simple)
* [test-kitchen](http://kitchen.ci/)
* [bundler](http://bundler.io/)

Tool Installation

Download and install VirtualBox: <https://www.virtualbox.org/wiki/Downloads>

Download and install Vagrant: <https://www.vagrantup.com/downloads.html>

Install bundler

sudo gem install bundler

Environment Setup

In order to begin development, we need to setup our directory structure and environment. In an attempt to make MagNETO integration as straightforward as possible, the environment will be built to match the expected structure for a MagNETO project.

First, make a directory in order to begin your development and navigate to it.

Initialize a Gemfile via bundler in order to handle necessary dependencies:

bundle init

Add librarian-puppet-simple, test-kitchen, and kitchen-puppet to this file (as in, we're specifying that we need the gems necessary to support our testing and development environment).

echo 'gem "librarian-puppet-simple"' >> Gemfile

echo 'gem "test-kitchen"' >> Gemfile

echo 'gem "kitchen-puppet"' >> Gemfile

Install our dependencies via bundler

bundle install

Initialize our test-kitchen environment

kitchen init

Do as kitchen says, and run bundle install again

bundle install

Edit the .kitchen.yml file in order to specify which boxes you're testing against, as well as general configuration options:

---

driver:

name: vagrant

provisioner:

name: puppet\_apply

manifests\_path: manifests

modules\_path: modules

hiera\_data\_path: hiera

files\_path: files

hiera\_config\_path: hiera.yaml

require\_chef\_for\_busser: false

resolve\_with\_librarian\_puppet: false

platforms:

- name: ubuntu-14.04-64-nocm

driver\_plugin: vagrant

driver\_config:

box: puppetlabs/ubuntu-14.04-64-nocm

box\_url: puppetlabs/ubuntu-14.04-64-nocm

verifier:

ruby\_bindr: '/usr/bin'

- name: centos-7.0-64-nocm

driver\_plugin: vagrant

driver\_config:

box: puppetlabs/centos-7.0-64-nocm

box\_url: puppetlabs/centos-7.0-64-nocm

verifier:

ruby\_bindr: '/usr/bin'

- name: centos-6.6-64-nocm

driver\_plugin: vagrant

driver\_config:

box: puppetlabs/centos-6.6-64-nocm

box\_url: puppetlabs/centos-6.6-64-nocm

verifier:

ruby\_bindr: '/usr/bin'

suites:

- name: default

manifest: site.pp

**Note**: Testing across multiple VMs can be very time consuming. I suggest focusing on a single OS that you plan on using in production, unless you plan on making a module (in which case it should be tested across multiple environments). Comment out entire 'platform' blocks in order to only test against your targeted environment. For our purposes, we'll be testing on just centos-6.6, so I'll comment out the other platforms, like so:

...

platforms:

# - name: ubuntu-14.04-64-nocm

# driver\_plugin: vagrant

# driver\_config:

# box: puppetlabs/ubuntu-14.04-64-nocm

# box\_url: puppetlabs/ubuntu-14.04-64-nocm

# verifier:

# ruby\_bindr: '/usr/bin'

# - name: centos-7.0-64-nocm

# driver\_plugin: vagrant

# driver\_config:

# box: puppetlabs/centos-7.0-64-nocm

# box\_url: puppetlabs/centos-7.0-64-nocm

# verifier:

# ruby\_bindr: '/usr/bin'

- name: centos-6.6-64-nocm

driver\_plugin: vagrant

driver\_config:

box: puppetlabs/centos-6.6-64-nocm

box\_url: puppetlabs/centos-6.6-64-nocm

verifier:

ruby\_bindr: '/usr/bin'

...

Now let's ensure test-kitchen can properly spawn the VMs we need - type the following and grab a coffee:

kitchen create

Test-kitchen (and Vagrant) should download the necessary images the first time you do this and spawn them. Subsequent calls will use the locally cached version of the instance.

**Note**: If you run into SSL certificate errors, make sure you have the latest Websense/Bluecoat proxy certificates installed on your machine. Unfortunately, since these changes so frequently and there doesn't appear to be a source of truth for these, you're on your own tracking down the latest version. The Self Service app is hit or miss. You may have luck with [this github repo](https://github.comcast.com/cats-misc/proxy-ca-fix-recipes)

If all went well, you should have one or more running VMs. You can verify this by running the following:

kitchen list

Since I had all 3 instances enabled in my .kitchen.yml, kitchen list returned the following:

Instance Driver Provisioner Verifier Transport Last Action

default-ubuntu-1404-64-nocm Vagrant PuppetApply Busser Ssh Created

default-centos-70-64-nocm Vagrant PuppetApply Busser Ssh Created

default-centos-66-64-nocm Vagrant PuppetApply Busser Ssh Created

When you're done testing and want to destroy your instances, use the following:

kitchen destroy

Puppet Integration

Create a directory to hold manifests to begin Puppet integration:

mkdir manifests

There is one more housekeeping task we need to do in order to more closely align with MagNETO - we need to create a hiera configuration file. This file designates the order (or hierarchy) that values are determined when a hiera call is used in a manifest. You can read more about hiera [here](http://docs.puppetlabs.com/hiera/1/) but for our purposes, just understand that we're creating this file as it matches the file on the MagNETO puppet masters. This way, if we utilize hiera in our test-kitchen setup, it'll work identically on a deployed server managed by MagNETO. Create a file called hiera.yaml and populate it with the following:

---

:backends:

- yaml

:hierarchy:

- "%{::hostname}"

- "%{::environment}"

- "%{::system\_role}/%{::clientcert}"

- "%{::system\_role}/%{::location}/common"

- "%{::system\_role}/common"

- common

With all of that out of the way, we can actually start writing Puppet code!

Inside the manifests folder, create a file site.pp - this file is the manifest that will get applied on our Vagrant VMs. Insert the following into the site.pp file.

notify {"Hello World!":}

Now we can actually apply the manifests. To do this, we'll call the converge method of test-kitchen. This will spawn instances (if they are not already running), install puppet, and apply the manifest.

kitchen converge

This should produce something similar to the following output:

Notice: Compiled catalog for default-centos-66-64-nocm.cable.comcast.com in environment production in 0.03 seconds

Notice: Hello World!

Notice: /Stage[main]/Main/Notify[Hello World!]/message: defined 'message' as 'Hello World!'

Notice: Finished catalog run in 0.02 seconds

Managing Puppet Modules - librarian-puppet-simple

In order to include puppet modules for our testing, we're going to use [librarian-puppet-simple](https://github.com/bodepd/librarian-puppet-simple) for the following reasons:

* MagNETO supports it
* It's lightweight, simple, and straightforward
* You can manage all of your modules using a single file

As an example, let's pull down the puppetlabs NTP module - in order to find modules, [you can search the puppetforge](https://forge.puppetlabs.com/).

If you search NTP, you'll see the puppetlabs NTP module - once you find a module, you'll need two things - the dependencies (since librarian-puppet-simple doesn't manage dependency resolution) and the github url (often found on the "Project URL" link on the top of the page). For the NTP module, the dependency is just on the puppetlabs stdlib - find this module and record the github URL as well as any dependencies.

Once at the github URL for a module, click 'releases' on the top and note the release number you want to use. For the puppetlabs/stdlib module, let's use 4.10.0 and for the puppetlabs/ntp module, let's use 4.1.2. Now that we have all the necessary information, let's populate our Puppetfile.

Create a file called Puppetfile in your development directory and add the following contents:

mod "puppetlabs/stdlib",

:git => "https://github.com/puppetlabs/puppetlabs-stdlib.git",

:ref => "4.10.0"

mod "puppetlabs/ntp",

:git => "https://github.com/puppetlabs/puppetlabs-ntp.git",

:ref => "4.1.2"

Now let's have puppet-librarian-simple handle the rest - type the following to download your modules:

librarian-puppet install

You should get output similar to the following:

Cloning into 'stdlib'...

remote: Counting objects: 7715, done.

remote: Total 7715 (delta 0), reused 0 (delta 0), pack-reused 7714

Receiving objects: 100% (7715/7715), 1.54 MiB | 2.27 MiB/s, done.

Resolving deltas: 100% (3636/3636), done.

Checking connectivity... done.

Note: checking out '4.10.0'.

You are in 'detached HEAD' state. You can look around, make experimental

changes and commit them, and you can discard any commits you make in this

state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may

do so (now or later) by using -b with the checkout command again. Example:

git checkout -b <new-branch-name>

HEAD is now at 0b4822b... Merge pull request #557 from jbondpdx/4.10.x

Cloning into 'ntp'...

remote: Counting objects: 2139, done.

remote: Total 2139 (delta 0), reused 0 (delta 0), pack-reused 2139

Receiving objects: 100% (2139/2139), 449.33 KiB | 0 bytes/s, done.

Resolving deltas: 100% (1085/1085), done.

Checking connectivity... done.

Note: checking out '4.1.2'.

You are in 'detached HEAD' state. You can look around, make experimental

changes and commit them, and you can discard any commits you make in this

state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may

do so (now or later) by using -b with the checkout command again. Example:

git checkout -b <new-branch-name>

HEAD is now at 4426906... Merge pull request #299 from tphoney/release\_4.1.1

In addition, you should see a modules directory created and two new directories inside that "modules" directory:

HQSML-146983 :: ~/Development/puppetdev » ls modules

ntp stdlib

Now let's modify our manifest to install NTP - edit your site.pp to contain the following:

notify {'Hello World!':}

class { '::ntp': }

Rerun your converge command:

kitchen converge

And you should see output similar to the following:

Notice: Compiled catalog for default-centos-66-64-nocm.cable.comcast.com in environment production in 0.42 seconds

Notice: Hello World!

Notice: /Stage[main]/Main/Notify[Hello World!]/message: defined 'message' as 'Hello World!'

Notice: /Stage[main]/Ntp::Config/File[/etc/ntp.conf]/content: content changed '{md5}7fda24f62b1c7ae951db0f746dc6e0cc' to '{md5}c1d0e073779a9102773754cf972486be'

Notice: /Stage[main]/Ntp::Service/Service[ntp]/ensure: ensure changed 'stopped' to 'running'

Notice: Finished catalog run in 0.45 seconds

Since puppet is [idempotent](https://docs.puppetlabs.com/guides/introduction.html#idempotency), it won't apply any changes that are not needed. Since ntp is already installed (as it is with this VM), all this manifest does is modify the configuration file and start the ntp service. If ntp wasn't installed, it would be installed as part of the process.

Development Best Practices - Roles and Profiles Pattern

There are many ways to write Puppet manifests and modules, some of which are more readable/maintainable than others. In my (limited) experience, I've found the "Roles and Profiles" pattern to work best. [Here's a great video on it](https://www.youtube.com/watch?v=ZpHtOnlSGNY) (and I believe the source of the pattern itself). What this means for us is that in order to abstract out node classifications, we create profiles (manifests that get applied to multiple nodes) and roles (manifests that only apply to **one** node). This allows us to reuse profiles on multiple nodes (e.g. common package installation, security and authentication) and define roles for a single node that consists only of other profiles.

To do a trivial example of this, let's create some directories to house our roles and profiles manifests.

mkdir modules/site

mkdir modules/site/manifests

mkdir modules/site/manifests/roles

mkdir modules/site/manifests/profiles

Let's assume we have a web server that depends on a database to form a full web application. We can define two roles here:

* web (Customer facing aspect)
* database (Data store that supports the frontend)

In our design, let's assume we've decided on apache as the frontend and mysql as the backend. Let's also assume there are some modules that are needed on both servers (ntp and vim). We can divide this up into profiles:

* base (common packages, services, etc that are needed on all servers)
* mysql (includes the mysql module and any other database specific configuration)
* apache (includes the apache module and any other web server specific configuration)

**Note**: Refer to this blog post on how to appropriately name roles and profiles (though there are no strict rules here - do what works best for you): <http://garylarizza.com/blog/2014/02/17/puppet-workflow-part-2/>

We'll need the apache and mysql modules (and their dependencies), so let's update the Puppetfile as such:

mod "puppetlabs/stdlib",

:git => "https://github.com/puppetlabs/puppetlabs-stdlib.git",

:ref => "4.10.0"

mod "puppetlabs/ntp",

:git => "https://github.com/puppetlabs/puppetlabs-ntp.git",

:ref => "4.1.2"

mod "puppetlabs/concat",

:git => "https://github.com/puppetlabs/puppetlabs-concat.git",

:ref => "1.2.5"

mod "nanliu/staging",

:git => "https://github.com/nanliu/puppet-staging.git",

:ref => "1.0.4"

mod "puppetlabs/apache",

:git => "https://github.com/puppetlabs/puppetlabs-apache.git",

:ref => "1.7.1"

mod "puppetlabs/mysql",

:git => "https://github.com/puppetlabs/puppetlabs-mysql.git",

:ref => "3.6.2"

Since we've updated the Puppetfile, we need to refresh our modules using librarian-puppet:

librarian-puppet install

We should have some new modules in our "modules" folder (namely apache, and mysql).

Let's start with the base class - create the file modules/site/manifests/profiles/base.pp and add the following:

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:trackback="http://madskills.com/public/xml/rss/module/trackback/"> <rdf:Description rdf:about="https://wiki.io.comcast.net/display/ISE/Local+Puppet+Development+with+Test+Kitchen" dc:identifier="https://wiki.io.comcast.net/display/ISE/Local+Puppet+Development+with+Test+Kitchen" dc:title="Local Puppet Development with Test Kitchen" trackback:ping="https://wiki.io.comcast.net/rpc/trackback/454574588"/> </rdf:RDF>

Add Labels

\#main

* Powered by [Atlassian Confluence](http://www.atlassian.com/software/confluence) 3.4.6, the [Enterprise Wiki](http://www.atlassian.com/software/confluence/tour/enterprise-wiki.jsp)   |  [Report a bug](http://jira.atlassian.com/secure/BrowseProject.jspa?id=10470)  |  [Atlassian News](http://www.atlassian.com/about/connected.jsp?s_kwcid=Confluence-stayintouch)

© Comcast, 2011. All contents confidential and proprietary to Comcast.