**SIRUM instructions to setup and restore AWS Instances**

**Recent Changes**

Mount /dev/sdg to /dscsa

(this contains the keys folder too, there is not a separate drive e.g, /dev/sdf for the keys)

Ensure there is a symlink from /var/www/sirum to /dscsa/sirum (sudo ln -s /dscsa/sirum /var/www/sirum)

Checkout with ‘sudo git pull’

**Restore Server from an EBS Backup**

1. ***Important! If downgrading server, comment out ephemeral storage, skip to step 7. Otherwise the micro server may not boot!***

*sudo nano /etc/fstab*

* comment out sda2, sda3…

1. *If restoring, create a volume “filesystem” from the backup snapshot*
2. *Create new instance*

* *regular quick start Linux AMI*
* *make sure it is in region us-west-1b*

1. *Stop the instance*
2. *Detach and delete its EBS volume (delete is optional(*
3. *Attach the new “filesystem” volume to /dev/sda1*
4. *Create a 1Gb volume named “credentials”*

*- Attach the credential volume to /dev/sdf*

1. *Restart the instance (reattach elastic IP if needed)*
2. *SSH into server and create secure folder*

* *Create secure folder*

sudo cryptsetup -c aes-cbc-essiv:sha256 --key-size 256 luksFormat /dev/sdf

* Answer YES (in UPPERCASE)

sudo cryptsetup luksOpen /dev/sdf key

sudo mkfs.ext3 /dev/mapper/key

sudo mount /dev/mapper/key /var/www/sirum/key

(sudo mount device\_name folder, e.g sudo mount /dev/xvdg/ /dscsa)

* *Upload Keys to Server*

sudo cp /var/www/sirum/key-/\* /var/www/sirum/key

sudo mv /var/www/sirum/key/development.php /var/www/sirum/key/{environment}.php

sudo nano /var/www/sirum/key/{environment}.php

* add the correct keys to the file

sudo mv /var/www/sirum/keys-/swekey.txt /var/www/sirum/keys/swekey.conf

Note: if the server has changed you will need to create a new CSR and receive a new SSL certificate since the CSR is server specific

*10) Start the server services*

~~cd /var/www/sirum~~ cd /dscsa

~~sudo svn update~~ sudo git pull

sudo service httpd start

sudo service mysqld start

sudo service vsftpd start

ping google.com

(if can’t resolve DNS, “Unknown Host”, then add Google DNS)

/etc/resolv.conf

nameserver 8.8.8.8

nameserver 8.8.4.4

(you may need to do httpd restart after this. We had SOAP failing but it started working again after I restarted apache like this). You will most likely also have to take the steps in **Email** Issues as sending email is also affected by DNS.

USEFUL COMMANDS

Watch disk usage (mysql running out of space)

sudo watch –n1 df –h mysqld

du -ah /var | sort -nr | head -n 10

du -xh / |grep '^\S\*[0-9\.]\+G'|sort -rn

**increase size of EBS in AWS then sudo resize2fs /dev/xvda1**

**Email Issues**

1. *Sudo service sendmail restart*
2. *Sudo nano /var/spool/root or /var/spool/mqueue/*

*(look for errors, last date sent deferred = DNS Issue)*

1. *Sudo mailq (are they sitting in the queue?)*
2. *Sudo sendmail –q –v (try resending ones in the queue)*

*(Might need to be super user i.e., sudo su)*

**Create New Instance From Scratch**

1. *Launch new Instance*

Username info@sirum.org

Secure SIRUM password

Launch Instance in West 1b

*2) Security Group:*

Name => Firewall

Open Ports

22 SSH 0.0.0.0/0

80 HTTP 0.0.0.0/0

443 HTTPS 0.0.0.0/0

*3) Other Settings*

Termination Protection Enabled

Name => SIRUM

Key Name => sirum\_ec2\_key

Download Key to Desktop

*4) Start Server*

Start Instance

Name current volume “filesystem”

Create a new 1GB volume named “credentials”

Attach new volume to instance at /dev/sdf

Assign an Elastic IP

**Login to Server**

*1) Open Terminal and Change Directory to Key File:*

cd ~/Desktop

*2) Change permissions of Key File:*

chmod 400 <Key Name>

*3) Copy keys\*.php to key.php and fill with credentials*

*4) Create an EC2 folder and place both key file and keys.php inside*

*5) Go to application/utilities/disk utility*

File > New > New Image from Folder > 256 Bit Encryption

6) Mount the image and enter your password

*7) Login through SSH with this command*

ssh -v -i /Volumes/EC2/sirum\_ec2\_key.pem ec2-user@<Public IP>

**Install APACHE/PHP**

*Note: since you are ec2-user rather than root you must use the sudo tag in front of most commands*

*1) Install PHP*

*sud*

sudo yum update

sudo yum remove php54 php54-mcrypt php54-gd php54-mysql php54-mbstring php54-soap php54-pecl-apc php54-pecl-memcache php54-xml

sudo yum install php55 php55-mcrypt php55-gd php55-mysqlnd php55-mbstring php55-soap php55-pecl-apc php55-pecl-memcache php55-xml

sudo nano /etc/php.ini

(control W to search, control O to save, and control X to exit)

- change the following settings

- expose\_php = Off

- max\_execution\_time = 300

- memory\_limit = 900M

- post\_max\_size = 300M

- upload\_max\_filesize = 300M

- date.timezone = 'UTC'

- display\_errors = On

*2) Install Apache and SSL*

sudo yum install httpd24 openssl mod24\_ssl mod24\_security mod\_security\_crs

*3) Configure Mod Security to not mess up phpmyadmin*

sudo nano /etc/httpd/conf.d/mod\_security.conf

* change

SecRequestBodyLimit to 131072000

SecRule TX:/^MSC\_/ "!@streq 0" \ from deny to log,allow

SecAuditLog /var/www/sirum/log/modsec\_audit.log

SecDebugLog /var/www/sirum/log/modsec\_debug.log

sudo nano modsecurity\_crs\_10\_config.conf

SecDefaultAction "phase:2,pass,log"

sudo nano /etc/httpd/modsecurity.d/activated\_rules/modsecurity\_localrules.conf

* add

SecRule REQUEST\_URI ^/sirum phase:1,allow,ctl:ruleEngine=off

SecRule REQUEST\_URI ^/cms phase:1,allow,ctl:ruleEngine=off

sudo nano /etc/httpd/modsecurity.d/activated\_rules/modsecurity\_crs\_49\_inbound\_blocking.conf

- add

SecAction "sanitiseArg:password, sanitiseArg:confirm\_password, sanitiseArg:first\_name, sanitiseArg:last\_name, sanitiseArg:existing\_patient, sanitiseArg:rx"

- change

SecRule TX:ANOMALY\_SCORE "@ge 200" \

SecRule TX:ANOMALY\_SCORE "@ge 200" \

*4) Install PageSpeed*

sudo nano /etc/yum.repos.d/mod-pagespeed.repo

-add

[mod-pagespeed]

name=mod-pagespeed

baseurl=http://dl.google.com/linux/mod-pagespeed/rpm/stable/i386

enabled=1

gpgcheck=0

sudo yum --enablerepo=mod-pagespeed install mod-pagespeed

sudo nano /etc/httpd/conf.d/pagespeed.conf

* Uncomment

ModPagespeedEnableFilters rewrite\_javascript,rewrite\_css

ModPagespeedEnableFilters collapse\_whitespace

ModPagespeedEnableFilters add\_instrumentation

* Add

ModPagespeedDisallow \*/cms/\*

ModPagespeedEnableFilters combine\_javascript,sprite\_images,extend\_cache

ModPagespeedDisableFilters rewrite\_images,trim\_urls

- trim\_urls messes up tooltip links

ModPagespeedBeaconUrl "/ajax/beacon?"

(elide\_attributes breaks jquery validation plugin).

*5) Install Other Extensions*

sudo yum install logwatch

sudo nano /etc/logwatch/conf/logwatch.conf

- Add

MailTo = adam@sirum.org

Detail = Medium

**Configure Apache**

*1) Configure Apache*

sudo nano /etc/httpd/conf/httpd.conf

- remove /etc/httpd/conf.d/welcome.conf

- change

DocumentRoot to /var/www/sirum/url

<Directory from: /var/www/html to: /var/www/sirum/url

Within <Directory>, AllowOverride from None to All

From: <Files ".ht\*"> To: <Files "ht\*">

<IfModule dir\_module>

DirectoryIndex index.html index.php

</IfModule>

* add

KeepAlive On

KeepAliveTimeout 80

StartServers 5

MinSpareServers 3

MaxSpareServers 7

MaxRequestsPerChild 4000

ExpiresActive On

# Add Proper MIME-Type for Favicon

AddType image/x-icon .ico

ExpiresByType image/x-icon "access plus 1 month"

ExpiresByType image/gif "access plus 1 month"

ExpiresByType image/png " access plus 1 month "

ExpiresByType image/jpeg " access plus 1 month "

ExpiresByType text/javascript " access plus 1 day "

ExpiresByType text/css " access plus 1 day "

AddOutputFilterByType DEFLATE text/javascript text/css text/html

FileETag None

<LocationMatch ".\*\.svn.\*">

Order allow,deny

Deny from all

</LocationMatch>

AccessFilename htaccess.txt

(control O to save control W to search and control X to exit)

**Configure SSL**

1. *Create Public Key Credentials*

sudo openssl genrsa 2048 > *private-key.pem*

sudo openssl req -new -key *private-key.pem* -out *csr.pem*

nano csr.pem

*2) Copy and paste CSR into an SSL provider (rapidssl)*

Save/Rename Web Server CERTIFICATE to sirum\_org.crt

Save/Rename INTERMEDIATE CA (chain) to sirum\_org.ca-bundle

Exit server ssh

To:

scp -v -r -i /Volumes/EC2/sirum\_ec2\_key.pem /Users/adam/Desktop/sirum\_org\* ec2-user@donate.sirum.org:

From:

scp -v -r -i /Volumes/EC2/sirum\_ec2\_key.pem ec2-user@donate.sirum.org:/var/www/sirum/log/log-2014-09-17.php ~

Label Example:

scp -v -r -i /Volumes/EC2/sirum\_ec2\_key.pem /Users/adam/Desktop/D710R719T4439\_label.pdf ec2-user@donate.sirum.org:/var/www/sirum/url/label/

*3) Move SSL Certificates*

Enter server ssh again

sudo mv csr.pem /etc/pki/tls/private

sudo mv private-key.pem /etc/pki/tls/private

sudo mv sirum\_org.crt /etc/pki/tls/certs

sudo mv sirum\_org.ca-bundle /etc/pki/tls/certs

1. *Configure SSL*

sudo nano /etc/httpd/conf.d/ssl.conf

* change the following

SSLProtocol -ALL +SSLv3 +TLSv1

SSLCipherSuite ALL:!aNULL:!ADH:!eNULL:!LOW:!MEDIUM:!EXP:RC4+RSA:+HIGH

SSLCertificateFile /etc/pki/tls/certs/sirum\_org.crt

SSLCertificateKeyFile /etc/pki/tls/private/private-key.pem

SSLCertificateChainFile /etc/pki/tls/certs/sirum\_org.ca-bundle

SSLCACertificateFile /etc/pki/tls/certs/sirum\_org.ca-bundle

- change these lines

*# SetEnvIf User-Agent ".\*MSIE.\*" \*

*# nokeepalive ssl-unclean-shutdown \*

*# downgrade-1.0 force-response-1.0*

*SetEnvIf User-Agent ".\*MSIE [1-5].\*" \*

*nokeepalive ssl-unclean-shutdown \*

*downgrade-1.0 force-response-1.0*

*SetEnvIf User-Agent ".\*MSIE [6-9].\*" \*

*ssl-unclean-shutdown \*

(control O to save and control X to exit)

**Install Subversion**

*1) Install subversion*

sudo yum install subversion

*2) Download code*

cd /var/www

sudo rm -R \*

sudo svn checkout --username=sirum http://svn.christianyang.com/sirum

*3) Change file permissions*

sudo nano /var/www/sirum/url/index.php

- change environment to {environment}

(control O to save and control X to exit)

*sudo chmod 777 /var/www/sirum/url/manifest*

*sudo chmod 777 /var/www/sirum/url/label*

*sudo chmod 777 /var/www/sirum/url/pic/org*

*sudo chmod 777 /var/www/sirum/log*

*sudo chmod -R 777 /var/www/sirum/data/upload*

**Secure Credentials**

*1) Install filesystem encryption*

sudo yum install cryptsetup

sudo modprobe sha256

sudo modprobe dm-crypt

1. *Create secure folder*

sudo cryptsetup -c aes-cbc-essiv:sha256 --key-size 256 luksFormat /dev/sdf

Answer YES (in UPPERCASE)

sudo cryptsetup luksOpen /dev/sdf key

sudo mkfs.ext3 /dev/mapper/key

sudo mkdir –p /var/www/sirum/key

sudo mount /dev/mapper/key /var/www/sirum/key

(sudo mount device\_name folder, e.g sudo mount /dev/xvdg/ /dscsa)

1. *Upload Keys to Server*

sudo cp /var/www/sirum/key-/\* /var/www/sirum/key

sudo mv /var/www/sirum/key/development.php /var/www/sirum/key/{environment}.php

sudo nano /var/www/sirum/key/{environment}.php

* add the correct keys to the file

sudo tr '\r' '\n' < /var/www/sirum/key-/swekey.txt > /var/www/sirum/key/swekey.conf

- convert mac newlines into unix newlines while we move

1. *Start Apache and test server*

sudo service httpd start

goto the public IP address

Site should be running although codeigniter does

not display database errors in production mode

**Install MYSQL-SERVER**

*1) Install and start mysql server*

sudo yum install mysql-server

sudo service mysqld start

sudo nano /etc/my.cnf

--add lines under [mysqld] not [mysqld\_safe]

#this makes admin metric.csv downloads work otherwise outfile commands fail

*secure-file-priv=""*

*2) Secure the installation*

sudo /usr/bin/mysql\_secure\_installation

(yes to everything except no to disabling remote login)

*3) Upload large unzipped sql files into database*

sudo mysql -u root -p

CREATE DATABASE `sirum-{environment}`;

USE `sirum-{environment}`;

SOURCE /var/www/sirum/data/script/sirum-structure.sql;

SOURCE /var/www/sirum/data/script/sirum-item.sql;

CREATE DATABASE `sirumCMS`;

USE `sirumCMS`;

SOURCE /var/www/sirum/data/scripts/sirumCMS.sql;

**Install PhpMyAdmin**

*1) Download and unzip latest version*

find latest version of phpmyadmin on

http://www.phpmyadmin.net/home\_page/downloads.php

cd /var/www/sirum/url

sudo wget http://sourceforge.net/projects/phpmyadmin/files/phpMyAdmin/<version>/phpMyAdmin-<version>-all-languages.tar.bz2/download

sudo bunzip2 phpMyAdmin-<version>-all-languages.tar.bz2

sudo tar xf phpMyAdmin-<version>-all-languages.tar

sudo rm phpMyAdmin-<version>-all-languages.tar

*2) Create phpmyadmin folders*

sudo mv phpMyAdmin-<version>-all-languages sirum

cd /var/www/sirum/url/sirum

sudo mkdir config

sudo chmod o+rw config

sudo cp config.sample.inc.php config/config.inc.php

sudo chmod o+w config/config.inc.php

*4) Setup phpmyadmin*

goto <PUBLIC IP>/sirum/setup/index.php

Features > Security > Force SSL option

Overview > New Server

verbose => SIRUM

host => localhost

user SSL -> TRUE

compress connection => true

set SweKey config file to /var/www/sirum/key/swekey.txt

save the page

save the overall settings

*5) Secure the installation*

sudo nano /var/www/sirum/url/sirum/config/config.inc.php

- add these lines

$cfg['Servers'][$i]['AllowDeny']['order'] = 'explicit';

$cfg['Servers'][$i]['AllowDeny']['rules'] = array('allow % from 98.248.35.137', 'allow % from 171.66.[00-99].[000-999]');

$cfg['Servers'][$i]['only\_db'] = array('sirum-production', 'sirum-testing', 'sirumCMS');

$cfg['Servers'][$i]['auth\_swekey\_config'] = '/var/www/sirum/keys/swekey.txt';

(control O to save and control X to exit)

sudo mv /var/www/sirum/url/sirum/config/config.inc.php /var/www/sirum/url/sirum

sudo chmod 705 /var/www/sirum/url/sirum/config.inc.php

sudo rm -r /var/www/sirum/url/sirum/setup

sudo rm -r /var/www/sirum/url/sirum/config

*6) Change Database As Necessary*

goto <PUBLIC IP>/sirum

Login into phypmyadmin with

user name: root

password: same as entered during mysql\_secure\_installation

Run any applicable scripts located in data/scripts/dbupdates.rtf

**Setup Cron Jobs**

1. *Change to the correct volume-id*

sudo nano /var/www/sirum/app/config/{environment}/config.php

- change aws\_volume to volume-id of the new instance

1. *Create Single Hourly Cron Job*

sudo crontab -e

i (command to insert in vi editor)

0 \* \* \* \* pgrep wget > /dev/null || (wget --spider http://localhost/bkg/{Cron Password}) >/dev/null

* spider avoids downloading the page, pgrep skips job if already running. All standard output is nulled.

1. *Create throttle alert*

*\* \* \* \* \* /var/www/sirum/data/scripts/ThrottleAlert.sh >/dev/null*

esc (command to stop inserting)

ZZ (UPPERCASE command to save and quit)

**Finish Installation**

1. *If production, turn off Error Reporting*

sudo nano /etc/php.ini

- change

display\_errors = Off

1. *Cleanup extra processes*

sudo top

- get the PID for any processes that should not be running

- sudo kill <PID> to stop those processes

1. *Take snapshot of volume and name it “initial setup”*
2. *Test site speed on webpagetest.org*

**Update EDI**

[*http://stackoverflow.com/questions/7052875/setting-up-ftp-on-amazon-cloud-server*](http://stackoverflow.com/questions/7052875/setting-up-ftp-on-amazon-cloud-server)

1. sudo yum install vsftpd
2. *In AWS Security Group enable ports 20-21 & 1024-1048. Note the public IP of the address.*
3. *sudo nano /etc/vsftpd/vsftpd.conf*

* *change anonymous\_enable to “=NO”*
* *Uncomment chroot\_local\_user=YES*
* *add pasv\_enable=YES*
* *add pasv\_min\_port=1024*

*-add pasv\_max\_port=1048*

*-add pasv\_address=<Public IP>*

1. *sudo service vsftpd start*
2. *Add User*

* *Mkdir /var/www/sirum/data/edi/<org\_id>*
* *sudo usermod --home /var/www/sirum/data/edi/<org\_id> <username>*