These are the instructions for building a new instance of the Listener UI: All examples use new customer "Acme".

## Assumptions

* Single-server UI for now. We can introduce load-balancing soon if it becomes necessary, but my understanding of the use model for the products suggests that it won’t.

## TBD

The application as described in this document contains significant SPOFs (Single Points of Failure):

* A single Mongo instance containing all of the data for the customer instance
* A single web server, serving the UI
* A single app server, serving the back-end API
* A single MySQL instance, providing persistence for the back end
* A single jboss instance providing messaging between the UI and the service

These should all be addressed in a later version of this document.

Note that SPOFs in this case don’t represent a risk to the customer’s continuity – all data is stored on Amazon EBS – a redundant storage solution with high reliability. Should either of the servers described in this document crash, no data should be lost.

## Spin up the New Instance

Click on the following link to create a new instance from the AWS console:

- https://console.aws.amazon.com/ec2/home?region=us-east-1#launchAmi=ami-90a05ef9

- Credentials ngenera.dev@gmail.com : r3db@rchetta

In the launch wizard, make the following selections:

1. Select availability zone us-east-1d

2. Select Instance Type Large (m1.large, 7.5 GB)

3. Select "Prevention against accidental termination"

4. Set at least the following 3 keys (feel free to add more meta-data if you find it useful):

Name: <Customer Name>

Owner: <Moxie Point-of-Contact for Listener instance>

Role: Listener - Web

7. Choose existing key pair: DEV

8. Select the existing security group titled "ListenerUI"

9. Launch

## Connect to the New Instance

Once instantiation is complete, connect to the new box by right-clicking on the instance and selecting ‘Connect’. Follow the directions in the dialog box to open an ssh session with the new instance.

## IT Request

Select a url for the new client (i.e. acme-listener.moxiesoft.com) and make a request to itsupport for a new CNAME that maps the new subdomain to the public DNS of the new instance, which can be obtained from the AWS console. Example:

Hello IT Guy,

I’d like to request a CNAME mapping acme-listener.moxiesoft.com to ec2-204-236-241-19.compute-1.amazonaws.com.

Regards,

Operations Guy

## Modify the instance

# Set the hostname to make monitoring and console apps more friendly

Hostname acme-listener-ui

echo "HOSTNAME=acme-listener-ui" >> /etc/sysconfig/network

reboot

# re-connect to the new instance

# Prepare the data volume

mkfs.ext3 /dev/sdf

mkdir -p /mount/mongo

echo "/dev/sdf /mount/mongo ext3 defaults 1 1" >> /etc/fstab

mount -a

# configure Apache

vi /etc/httpd/conf/sites-available/listenerui.vhost

# Change ServerName to reflect the desired URL (i.e. acme-listener.moxiesoft.com)

# configure the application – In this step, you will set the COMMUNITY\_URL to be the same as the CNAME you requested from IT, and the same as the ServerName you set in the apache config. The BSGRA\_GUID can be set to anything you like (it needn’t be hex) as long as you stick to the general format shown below and make sure it’s unique among all listener clients.

COMMUNITY\_URL: <http://acme-listener.moxiesoft.com>

BSGRA\_GUID: efb2fc58-ngen-acme-list-62886865list

Monit restart apache

## Test the New Instance

Go to your browser and type in the URL of the new instance:

<http://acme-listener.moxiesoft.com>

WARNING: CNAMEs can take a little while to propagate, so if you just got confirmation back from IT, it may be a couple of hours before this url resolves correctly.