How-To: Use the SMACK Toolchain

- SMACK PRINT -

WINTER 2016

Compiled By: Kyle Derby MacInnis

Contents

## Introduction

This document outlines the various utilities that form the SMACK toolchain. As more tools are added, this document will be updated to reflect them. They are broken down into categories of usage and will be listed with as much description as should be sufficient to utilize them.

The toolchain is a collection of scripts and utilities designed to interact with the cluster backend and the SMACK framework and provide a simpler more user-friendly method of interacting with it, without having to worry about the nitty-gritty details such as setting up environment variables, passing many parameters, and dealing with overly complicated authentication processes. In short, it is designed to provide a simpler-yet-effective form of interaction with OpenStack and and provides both a streamlined guided process (where possible) as well as a manual process for the more complex operations.

The guided methods provided by the toolchain are not designed to be used for large operations – they are designed to be easy to follow and to be used to help gain a better understanding of what each operation does without requiring the users to be fluent with cluster management or OpenStack itself. They prompt for information and where possible list the available options such that it is easy to jump in and begin working with the system.

The manual methods provided by the toolchain are designed for the development team and for users who require more direct access to some of the OpenStack functionality without requiring the users to be familiar with it. It provides a Command-line interface with parameters that can be used to specify the information without required prompts and user input. These manual tools are also incorporated into many other parts of the SMACK project and will be found to be used in many of the scheduling, initialization, and cluster automation that is part of this project.

## Cluster Management Control

The following utilities are designed to focus on managing instances within the cluster, deployment of virtual machines, and managing them as well as configuring their IP addresses and properties.

# Logging into the Cluster

There are two options for logging into the cluster: Guided and Manual. For security purposes, make sure that you always logout when you are done working with the cluster.

Guided Usage:
Just run the program and follow the prompts:

Example:

```
      $ :> smack-login
```

Manual Usage:
When manually using the program, the following parameters can be used:

Parameters:

```
      -h    :     Help Message

      -u    :     Username

      -x    :     Password

      -p    :     Project Name
```

Example:

```
      $ :> smack-login –h

      $ :> smack-login –u <username> -x <password> -p <project>
```

# Logging out of the Cluster

There is only one way to use the logout command – and that is just to type it and run. It will clear all login credentials from memory and will require the user to login prior to the toolchain functioning. Make sure you logout after each session.

Manual Usage:
To Log out of the OpenStack Cluster use the following program:

Example:

```
      $ :> smack-logout
```

# Listing Node Instances

If you wish to list the current nodes that are initialized within a project, please use the following program. This will list information related to the project you are working on and all the information regarding the instances – including size, type, name, ip, and networking.

Manual Usage:

To list the nodes that are currently in the cluster and to read their properties, the following command can be run.

Example:

```
$ :> smack-lsnode
```

# Making a New Instance

If you wish to launch a new instance you can either use the guided or the manual usage methods. The guided usage is simple, but it requires that data be input each and every time whereas the manual method can be scripted and done programmatically.

Guided Usage:

Just run the program and follow the prompts to launch a new instance.

Example:

```
$ :> smack-mknode
```

Manual Usage (*not implemented):

To run the program manually, use the following parameters to pass the required data through the command line. Any data that is missing will be prompted for prior to creation of the instance.

Parameters:

-h	:	**Help Message**
-d	:	**Use Default Values**
-n	:	**Instance Name**
-f	:	**Flavour Type**
-i	:	**VM Image**
-k	:	**Access Key**
-x	:	**Deployment Script**

Examples:

```
$ :> smack-mknode -h

$ :> smack-mknode -d -n <name> -f <flavour>

$ :> smack-mknode -d -i <image> -k <accesskey> -x <script>
```

# Deleting an Instance (*Not Implemented)

The following commands can be used to delete instances from the cluster. There will be two methods used to delete instances: guided and manual. The guided method will prompt and display the available nodes whilst the manual method will be used for programmatically shutting down many or specific instances.

Guided Usage (*not implemented):

The guided method can be used to delete a single instance per run. The program will prompt for all necessary information and will display the available nodes that can be deleted. It will also confirm deletion prior to doing so.

```
$ :> smack-terminate
```

Manual Usage (*not implemented):

The manual method will be used when more control is required or scripting and programming is necessary or desired. The following parameters can be passed and used on the command line to select and perform the deletion. There will not be any confirmation so use caution when deleting instances manually.

```
$ :> smack-terminate
```

# Suspending an Instance (*Not Implemented)

When needing to suspend an instance – either for resource purposes, or for whatever reason, the following commands can be used. There are two methods: guided and manual.

Guided Usage (*not implemented):

The guided method can be used to select and suspend an instance that is currently active through an interactive prompt. All required information will be asked for and where possible it will list the available options.

```
$ :> smack-suspend
```

Manual Usage (*not implemented):

The manual method will be used where multiple instances need to be suspended concurrently or in cases where you know exactly which instance needs to be and you wish to do it in a single command.

```
$ :> smack-suspend
```

# Configuring the IP (*Not Implemented)

For tasks related to configuring the IP for the cluster and associating any IPs to various VM instances, the following tool has been developed. There are two methods: guided and manual.

Guided Usage (*not implemented):

The guided method will prompt for the required information and can be used to associate, disassociate, and change the floating IP (if applicable).

```
$ :> smack-setip
```

Manual Usage (*not implemented):

The manual method will be useful for automating network related tasks as well as configuring multiple instances programmatically. There will be parameters that can be used to configure the IP. Some examples are included below.

```
$ :> smack-setip
```

## Object Database Control

The following utilities are designed to interact with the Swift object storage database and will be used to upload, download, update, copy, and modify the contents of the Swift Storage for the cluster. Where applicable the utilities will be described for both guided and manual usage.

# Database Listing

If you wish to explore the objects stored within the database, the following program can be used to do so. The guided method allows you to explore and view the containers within your project. The manual method can be used to explore with more accuracy and more detail.

Guided Usage:

To list the various containers and to interactively check through them, just run the program and follow the prompts.

Example:

```
$ :> smack-lsdb
```

Manual Usage:

To list details about the various containers and their objects, you can use the manual parameters to gain a little more control over the listings. There is also an option to view the object and container statistics.

Parameters:

-c	:	**Container**
-s	:	**Show Statistics instead of listing**
-o	:	**Object**
-h	:	**Help Message**
-l	:	**List Root Containers**

Examples:

```
$ :> smack-lsdb –c <container>

$ :> smack-lsdb –c <container> -o <object>

$ :> smack-lsdb –s -c <container>

$ :> smack-lsdb –l

$ :> smack-lsdb –h
```

# Uploading Objects

If you wish to upload objects to swift storage, you can use the following program to do so. There is both a guided and a manual method for using it. The guided is useful if you are beginning, but it is limited to one object per run. If you wish to upload many objects, you can do so using the manual method and some scripting.

Guided Usage:

To upload a single file to the swift object storage, use the following command and follow the prompts. This can be used to upload only one file per run.

Example:

```
$ :> smack-upload
```

Manual Usage:

If you wish to have a little more control over your file uploads or you want to programmatically run and upload many files, the following parameters can be used to pass information from the command line.

```
Parameters:

      -a    :     Upload all files within directory

      -e    :     Upload all files within directory with extension

      -f    :     Local File to Upload

      -o    :     Remote Object Name to Store as

      -c    :     Container to Store Object in

      -h    :     Display Help Message

Examples:

      $ :> smack-upload –h

      $ :> smack-upload –a –c <container>

      $ :> smack-upload –e <extension> -c <container>

      $ :> smack-upload –f <local file> -o <remote file> -c <container>

      $ :> smack-upload –f <local file>

      $ :> smack-upload –h
```

# Downloading Objects

Similarly to uploading, if you wish to download an object from the swift storage to a local instance, use the following commands. There are two options: guided and manual. Guided is useful if you are looking for a particular file and only need one. Manual is good for programmatically retrieving multiple files.

Guided Usage:

For guided usage, just enter the following command and follow the prompts.

```
      $ :> smack-download
```

Manual Usage:

For manual usage, the following parameters can be passed to the command line for more specific control.

```
Parameters:

      -a    :     Download all objects within container

      -f    :     Local File to Save as
```

```
    -o    :    Remote Object to Download

    -c    :    Container to Download from

    -h    :    Display Help Message
```

Examples:

```
    $ :> smack-download -h

    $ :> smack-download -a -c <container>

    $ :> smack-download -c <container> -o <object> -f <local file>
```

# Deleting Objects (*Not Implemented)

The following commands can be used to delete objects from within the swift storage system. There will be two methods to deleting objects: guided and manual. The guided will prompt while the manual method can be used programmatically.

Guided Usage (*Not Implemented):

The guided method will be used to delete single items and containers through an interactive process. It can only be used to delete one object per run.

```
    $ :> smack-rmdb
```

Manual Usage (*Not Implemented):

The manual method will be used to delete multiple objects and containers, and be used in conjunction with scripts to help automate more complex tasks. There will be parameters that can be used to pass information through the command line and any missing information will be prompted for prior to running.

```
    $ :> smack-rmdb
```

# Updating Objects (*Not Implemented)

The following commands can be used to update objects that have been stored within swift object storage. There will be two main methods: guided and manual. The guided method will allow users to search and update objects within the database whereas the manual method will be used to programmatically update specified objects. This tool will also perform many of the related tasks, such as copying, moving, and renaming objects as well as the containers.

Guided Usage (*Not Implemented):

The guided method will be used to prompt for the information required, and will be as user friendly as possible. It will however only be used to perform one update at a time.

```
      $ :> smack-updb
```

Manual Usage (*Not Implemented):
The manual method will be used to perform more complex tasks relating to updating the objects within the database. There will be parameters setup to pass all the necessary information via the command line and programmatically for scripting.

```
      $ :> smack-updb
```

# Container Creation (*Not Implemented)

This program will be used to create containers within the swift storage system. The containers will be the main locations for the data and this will be used to perform tasks related to the containers specifically. Creation and duplication will be the main purposes of this tool – for other required actions see the above mentioned tools.

Guided Usage (*Not Implemented):
The guided method will be used to perform single actions in an interactive way. It will prompt for all required information and will be user friendly, but it will only be capable of performing one action per run.

```
      $ :> smack-mkdb
```

Manual Usage (*Not Implemented):
The manual method will be used for more control when dealing with the database. This would be used in cases where programmatically, containers must be made, and additionally in cases where error checking and redundancies can be easily implemented.

```
      $ :> smack-mkdb
```