Smilart Application Manager (SAM)

User Guide v1.12

Table of Contents

Overview

SAM is based on Docker functionality (https://docs.docker.com/). You don't have to know anything about Docker to use SAM, but elementary knowledge about Docker can help you to understand deeper how SAM works.

SAM (Smilart Application Manager) is the util which helps to manage different Smilart applications. It can download, install and remove these applications. The term *application* is the main term of SAM.

Application is a docker image (local or remote) with special meta information, which tells SAM how to work with it. Each application has a version, which stores as a tag of docker image.

Applications don't have any dependencies between each other. So from the SAM side each application is absolutelly independent. It's key difference from popular linux package format such as deb or rpm.

Classic example of application is MongoDB. It's just a service which can be used by other applications, but it's independent service because it doesn't have any dependencies to run. All necessary binary and library are included in the image. From the other hand another application such as Smilart Phoenix may depend on MongoDB in *runtime*. It means Smilart Phoenix can be run without MongoDB, but need it in runtime to work correctly. SAM doesn't know anything about these type of dependency, because different services can be run on differenet servers.

> ❗ Pay attention that not any docker image is Smilart Application, but only image which has special meta information. So if you get arbitrary image from docker hub, it won't work correctly with SAM.

Application name is a name of application's docker image *without* repository part. For example if docker image name is *smilart/mongodb*, then application name is *mongodb*

Family name is one of the application properties. If the property is not defined, then *family name* equals *application name*. SAM prohibits to install applications with the same *family name*. It's necessary to protect the system from intallation several applications which use the same resources such as tcp/udp ports, config files, data directories etc.

Multiversion support is one of the application properties. If the property is not defined, it's considered that this feature is not supported. If the property is set, SAM allows to install several applications with this feature even they have the same *family name*. So you can install several version of application with this feature. The feature is usually used to install several version of the application if they don't use the same resources. Smilart daemons usually use this feature.

Installation

Install on CoreOS/SmilartOS

Earlier SmilartOS was operating system, which based on CoreOS source base with necessary

changes and adaptations to install different Smilart applications. Now it is just installer which help you to install native CoreOS including utils like SAM. So if you install CoreOS through Smilart installer, SAM util is already installed and you shouldn't do anything.

There are steps to install SAM if CoreOS was installed by official way:

- Create directory */opt/bin*
- Put SAM util into */opt/bin*
- Add line `export PATH=$PATH:/opt/bin` into */root/.bashrc* file
- Relogin

Install on Ubuntu 16.04

- Install sam deb package: `apt install <filename>.deb`
- Relogin

Subcommands

install

This command is used to install application from a remote repository or a file.

To install image from a remote repository, this repository must be added to the SAM database through repository add command. By default only official docker repository presents in SAM database.

Syntax to install image from a remote repository:

```
sam install [<repos>/]<app>:<version>
```

or

```
sam in [<repos>/]<app>:<version>
```

- <app> — image name without repository prefix.
- <version> — image tag.
- <repos> — image repository. If it's ommited, use SAM default repository.

Syntax to install image from a file:

```
sam install <filename>
```

or

```
sam in <filename>
```

- <filename> — path (absolute or relative) to application docker image file. You can create this file using docker save command.

remove

This command is used to remove application from the SAM database. Usually, the application has uninstall script which is also run by this command. This command doesn't really remove docker image from the system. To do that please use `sam purge` command.

Syntax to remove application from the system:

```
sam remove <app>[:<version>]
```

or

```
sam rm <app>[:<version>]
```

- <app> — application name to remove.
- <version> — you can ommit this parameter if only one version of the app is installed.

purge

This command is used to remove unused images from the docker local cache.

Syntax to remove unused docker images:

```
sam purge
```

Be careful with this command, **all** docker images which absent in SAM database will be deleted.

list

This command is used to get the list of installed applications.

Syntax:

```
sam list [--raw] [--label]
```

Options:

- *--raw* — output without formating. Can be used for parsing result of the command.
- *--label* — show additional information about applications (such as family and multiversion support).

search

This command is used to search application in a remote repository.

> ⛔ Please, pay attention, this command doesn't work with official docker registry (https://hub.docker.com/)

Syntax:

```
sam search [<app>]
```

or

```
sam se [<app>]
```

- <app> — application name or substring of the application name.

Command `sam search` always search application in the default repository.

Each time when you run `sam search` command, SAM create or modify a local cache of the list applications in remote repository. This cache is used in autocomplete function (when you press <TAB>).

info

This command is used to get additional info about installed application. It might have information about build time and a little bit description of the application. There aren't any guarantees about output of this command.

Syntax:

```
sam info <app>
```

- <app> — application name.

upgrade

Deprecated

export

This command is used to export installed applications. You can use this command to make snapshot of the installed applications. By default only list of installed application will be exported. If you use *--all* parameter, list applications will be exported with their images in one archive. The size of the archive depends on the current system state and can be several gigabytes.

Syntax:

```
sam export <filename> [--all]
```

- <filename> — path to exported file.
- *--all* — option to export docker images too.

import

This command is used to import applications which were exported before. This command doesn't remove existed applications if they don't conflict with imported applications.

> ⛔ All conflicted applications will be removed from your system. You will be asked to confirm these changes during import process.

Syntax:

```
sam import <filename>
```

- <filename> — name of the imported file.

If version of operating system doesn't match with OS version in imported file, you will be warned about mistmatch of OS version. In this case you can continue import process, if you are sure in success.

addrepo

This command is used to add remote repository to SAM configuration.

Syntax:

```
sam addrepo [--default] [--name <name>] [--server <hostname>] [--username <username>]
[--password <password>] [--email <email>]
```

or

```
sam ar [--default] [--name <name>] [--server <hostname>] [--username <username>] [--
password <password>] [--email <email>]
```

- *--default* — use this repository as default repository. It means that you can ommit repository name in *sam install* command.
- *--name* — repository name. Use only to manage this repository.
- *--server* — IP-address or hostname of the remote repository.
- *--username* — username to log on the remote repository.
- *--password* — password to log on the remote repository.
- *--email* — email to log on the remote repository. Obsolete since docker version 17.04.

removerepo

This command is used to remove remote repository from SAM configuration.

Syntax:

```
sam removerepo <id>|<name>
```

or

```
sam rr <id>|<name>
```

- *<id>* — id of the repository to remove.
- *<name>* — name of the repository to remove.

modifyrepo

This command is used to modify remote repository in SAM configuration.

Syntax:

```
sam modifyrepo <id> [--default] [--name <name>] [--server <hostname>] [--username
<username>] [--password <password>] [--email <email>]
```

or

```
sam mr <id> [--default] [--name <name>] [--server <hostname>] [--username <username>]
[--password <password>] [--email <email>]
```

- *<id>* — id of the modified repository.

- *--default* — mark the repository as default.
- *--name* — new name of the repository.
- *--server* — new IP-address or new hostname of the repository.
- *--username* — new username to log on the remote repository.
- *--password* — new password to log on the remote repository.
- *--email* — new email to log on the remote repository. Obsolete since docker 17.04.

listrepos

This command is used to show the list of configured remote repositories in SAM.

Syntax:

```
sam listrepos [--raw]
```

or

```
sam lr [--raw]
```

- *--raw* — show output without formating.