stonebranch

Documentation

Universal Automation Center support for scheduling Docker Container Scheduling

script_dockerfile-centos_linux

Associated Activities:

Date: 27 April 2018

Author: Nils Buer

Revision: 01

CONFIDENTIALITY INFORMATION

Distribution list: Stonebranch Marketplace

Revision	Date	Author	Changes
00	20180427	Nils Buer	Initial Document (WIP)

Abstract:

This document provides all steps how to create a centos7 container with a fully configured Universal Agent inside. After performing all steps, you can schedule any application installed in your container using the Universal Controller Web-Client or REST API or any other Scheduler.

Contents

1 Disclaimer

No support and no warranty are provided by Stonebranch GmbH for this document and the related Universal Task. The use of this document and the related Universal Task is on your own risk.

Before using this task in a production system, please perform extensive testing.

Stonebranch GmbH assumes no liability for damage caused by the performance of the Universal Tasks

2 Scope

This document provides a documentation how to create a centos7 container with a fully configured Universal Agent inside.

It describes the steps to build the image using a predefined Dockerfile and how to build and start containers from that image.

3 Introduction

Docker Container are an effortless way to deploy applications because containers that contain all the necessary packages are easy to transport and install as files. Containers ensure the separation and management of resources used on a computer. This includes: code, runtime module, system tools, system libraries - everything that can be installed on one computer.

Due to this concept container are used in many C:D DevOps processes.

From a Scheduling point of view an application in a container needs to be scheduled like any other application. Universal Automation Center automates the entire life cycle for scheduling an application in a container:

1. Providing a Dockerfile to build an OS image e.g. Centos, SUSE, .. with a Universal Agent
2. Downloading a container image from an image store like the open or private docker hub
3. Creating a container based on the selected image
4. Starting a container
5. Scheduling the and Launching the application in the container
6. Stopping a container
7. Removing a container
8. Removing the local image of the container

This document covers Step 1 and 2 describing the steps to build the image using a predefined Dockerfile and how to build and start containers from that image.

4 Installation

To install a Universal Agent in a Docker image the following Prerequisites are required:

Docker Engine installed on your docker host

- Client: Version: 17.09.1-ce
- Server: Version: 17.09.1-ce

For Information how to install the docker engine refer to:

https://docs.docker.com/install/

Software requirements:

Universal Agent

A Universal Agent installation file (This file is available for download from the Stonebranch Customer Portal: https://stonebranch.zendesk.com/hc/en-us) of Version 6.4.x is required (Note earlier Version 5.2. or later might also work)

The examples in this document have been done with the Agent Install file:

- ***sb-6.4.2.2-linux-2.6-x86_64.tar.Z***

Universal Controller

Universal Controller Version 6.x is needed for the Agent in the Container to connect to the Universal Controller (Note earlier Controller Version 5.2.x or later might also work).

Note: Any Other Scheduler can also connect to the Universal Agent in the Container.

5 DOCKER Build Context

The first step to create a Docker image is to create the **build context**. The build context contains all files required to build the image.

To build context to create a CentOS container with a Universal Agent contains:

- Dockerfile – to build the base Linux Image containing a Universal Agent
- sb-6.4.2.2-linux-2.6-x86_64.tar.Z – universal agent application
- env_agent_file – universal agent configuration parameters, which are used when initiating the container from the image
- ucmd – pam configuration for linux

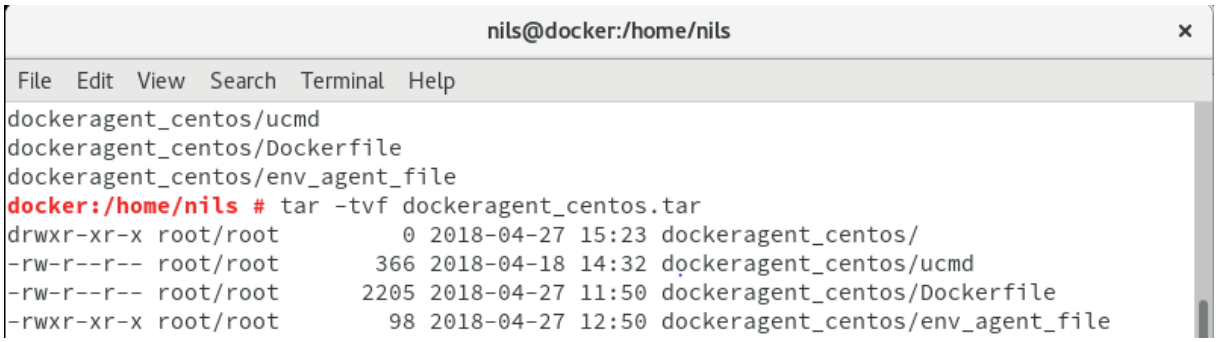
Installation of the build context:

1. **Download and extract the build context**

   Download and extract the build context tar file "*dockeragent_centos.tar*" to a directory on your docker host:

   tar -xvf dockeragent_centos.tar

   Tar-file content:



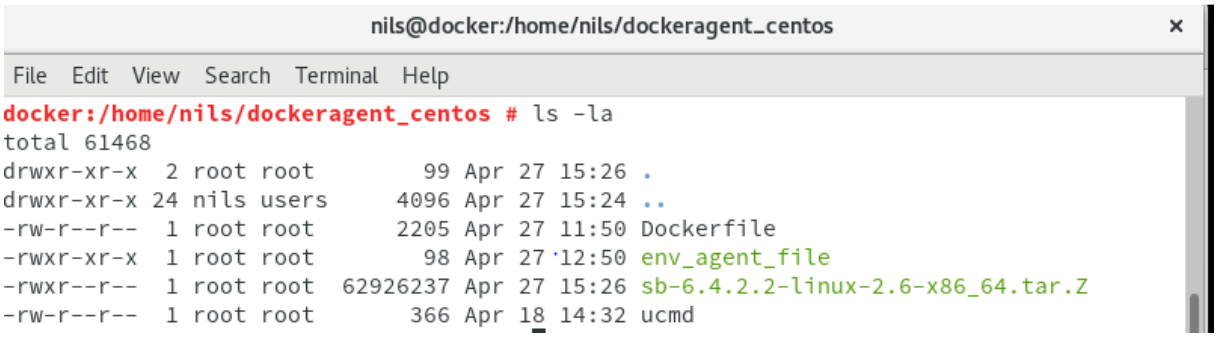
2. Copy the Agent Install file to the build context folder

Download from the Stonebranch Customer Portal an Agent Install file of Version 6.4.x (Note earlier Version 5.2.x or later might also work) and copy it to the build context.

The examples in this document have been done with the Agent Install file:

- ***sb-6.4.2.2-linux-2.6-x86_64.tar.Z***

To start building the docker image your build context folder should contain the following files:



6 Docker Image

To create a container with a universal agent you first need to build the docker image as a base to build a container. This is a onetime action.

The docker image containing a Universal Agent is build using the following command:

docker build --build-arg agentfile=sb-6.4.2.2-linux-2.6-x86_64.tar.Z -t centos7_img .

Description:

Command	Description
docker build	Docker CLI command to build a new docker image
--build-arg agentfile	This option sets a Variable which is valid during the docker build process. You need to set the value to the Universal Agent install file, which you want to install. In the example it is the Agent: *sb-6.4.2.2-linux-2.6-x86_64.tar.Z* The latest install file can be downloaded from the Stonebranch customer portal
-t	This Parameter set the name of the image. In the example the image will be called *centos7_img*

Check:

If the build command was successful you should see the following images:

```
docker:/home/nils # docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
centos7_img         latest              d555c48f7ad7        2 hours ago         313MB
centos              7                   e934aafc2206        2 weeks ago         199MB
```

Build Image Process:

The following described what is happing during the build process:

The docker build command reads the Dockerfile and performs the following steps:

- ⇨ Download the based centos 7 image from the internet (FROM centos:7)
- ⇨ Create a working directory in that image (WORKDIR "/root/stonebranch")
- ⇨ Add from the local file system the agent installer file (ADD ${agentfile}/root/stonebranch/)
- ⇨ Add from the local file system the PAM configuration file umd (ADD ucmd /etc/pam.d/)
- ⇨ Extract the Agent installer file in the image (RUN tar -xvzf /root/stonebranch/${agentfile} -C /root/Stonebranch)

Note: After these steps you have a centos 7 images (centos_img:latest) with the extracted agent installer file and the PAM configuration. **The agent is not yet installed.**

7 Docker Container

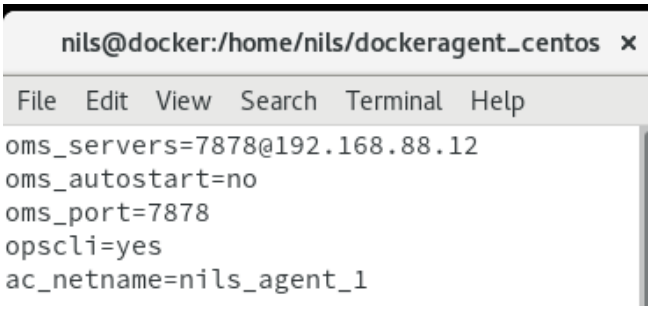
Once the image has been built you can create one or more containers from that images. Each container is like an own VM with own IP, hostname etc.

To build a container two steps need to be done:

1. **Configure the Universal Agent Installation settings**

In the file *env_agent_file* you need to configure the Agent settings:

Example *env_agent_file:*



Command	Description
oms_servers	Enter here the port and IP of the OMS this agent should connect to. Format: PORT@IP. If you have multiple OMS you can also provide here a comma separated list.
oms_autostart	Set if this agent should act as an OMS Server. (yes|no)
oms_port	Port of the OMS in case this agent should be used as OMS
Opscli	Decside if the agent should be reachable via cli.
ac_netname	**This value must be unique** Prefix name of the agent. The agent will appear in the controller with the name hostname – ac_netname. If you set the value to ac_netname=OPSAUTOCONF autonaming will be done.

Note: The Universal Agent has many more configuration Parameters. If you need additional Parameters, the Dockerfile needs to be adjusted accordingly.

2. Build the docker container

To build the container execute the following command:

docker run -it -d --hostname=centos01 --env-file=env_agent_file --name centos7_container centos7_img:latest

Command	Description
docker run	Docker CLI command to build a new docker container
--env-file	Parameter file with the agent installation parameters (needs to be set-up before starting the docker run command)
--hostname	Optional parameter. This Parameter sets the hostname in the container. If you omit it a hostname will be automatically assigned. The Agent appear in the Universal Controller in the format: <***hostname> – <ac_netname>.***
--name	Name of the docker container to build
<centos7_img:latest>	Last Parameter of the command defines the image that is used as a base to create the container. In the example it takes the image centos7_img with the tagged as latest

Check:

If the container build was successful you should see the following running container:

```
docker:/home/nils/dockeragent_centos # docker run -it -d --hostname=centosDEV --env-file=env_agent_file --name centos7_container1 centos7_img:late
6bf72f26803a32df113a6daab075346094300c1908b3843b1ef06ff452e7f80d
docker:/home/nils/dockeragent_centos # docker ps -a
CONTAINER ID        IMAGE                 COMMAND                  CREATED             STATUS              PORTS               NAMES
6bf72f26803a        centos7_img:latest    "/bin/sh -c 'sh /r..."   9 seconds ago       Up 8 seconds                            centos7_container1
docker:/home/nils/dockeragent_centos #
```

Additionally, you should check the container log file:

docker logs centos7_container1

Build Container process:

The following described what is happing during the build container process.

This process is executed each time you create a new container from the image.

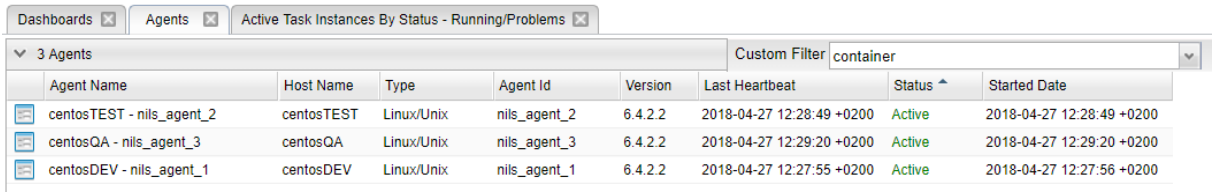
The docker run commands executed the **CMD command** in the **Dockerfile**.

When the container run command is executed the image *centos7_img:latest* is taken as a base to build the container. The environment variables are set, as defined in the *env_agent_file* (OMS_SERVERS, ac_netname, ..) and the command defined the docker file starting with CMD (line 52) is executed. Note: Only the Command CMD is executed when building the container. The CMD does the installation of the Universal Agent.

Note: After executing the CMD command the Docker container is created and can be started and stop as required.

Example:

The following screenshot shows three containers connect to the universal controller:



Agent Name	Host Name	Type	Agent Id	Version	Last Heartbeat	Status	Started Date
centosTEST - nils_agent_2	centosTEST	Linux/Unix	nils_agent_2	6.4.2.2	2018-04-27 12:28:49 +0200	Active	2018-04-27 12:28:49 +0200
centosQA - nils_agent_3	centosQA	Linux/Unix	nils_agent_3	6.4.2.2	2018-04-27 12:29:20 +0200	Active	2018-04-27 12:29:20 +0200
centosDEV - nils_agent_1	centosDEV	Linux/Unix	nils_agent_1	6.4.2.2	2018-04-27 12:27:55 +0200	Active	2018-04-27 12:27:56 +0200

These are the commands, which have been executed to build the three containers:

First Container:

1. Changed in the env_agent_file then name to nils_agent_1
2. docker run -it -d --hostname=centosDEV --env-file=env_agent_file --name centos7_container1 centos7_img:latest

Second Container:

1. Changed in the env_agent_file then name to nils_agent_2
2. docker run -it -d --hostname=centosTEST --env-file=env_agent_file --name centos7_container2 centos7_img:latest

Third Container:

1. Changed in the env_agent_file then name to nils_agent_3
2. docker run -it -d --hostname=centosQA --env-file=env_agent_file --name centos7_container3 centos7_img:latest

8 Docker Commands

The following provides some useful docker cli commands:

 # General

docker version

docker version

docker cli command to list images

docker images

docker cli to list container

docker ps -a

docker cli commands to start/stop a container

docker start centos7_container

docker stop centos7_container

container log file:

docker logs centos7_container1

Open a shell in a running container

docker exec -it centos7_container /bin/bash

 # Agent commands in the container

Start/Stop/restart/Status/Restart Universal Agent

/opt/universal/ubroker/ubrokerd start

/opt/universal/ubroker/ubrokerd stop

/opt/universal/ubroker/ubrokerd restart

/opt/universal/ubroker/ubrokerd status

Agent log file

tail -f /var/opt/universal/log/unv.log

9 Dockerfile

The following provides a description of the Dockerfile:

```
#####################################################################
# Dockerfile to build Universal Agent container images
# Based on Centos Linux
#####################################################################
# Set the base image to centos version 7
FROM centos:7

# File Author / Maintainer
MAINTAINER "Nils Buer <nils.buer@stonebranch.com>"

################### BEGIN INSTALLATION ######################
# Install Universal Agent
# Ref: https://www.stonebranch.com/confluence/display/UA64/Installation+Information

# Set Variables for agent installation file
ARG agentfile

# update all installed packages with newer available versions and install tar
# on centos:7 base image tar is already available
#####################################
# RUN yum -y update && yum -y install tar
#####
# change to workdir
WORKDIR "/root/stonebranch"

# Usage: ADD agent installer files to the image
ADD ${agentfile} /root/stonebranch/

# Usage: ADD pam configuration for centos Linux
ADD ucmd /etc/pam.d/

# Extract Agent Installer files
RUN tar -xvzf /root/stonebranch/${agentfile} -C /root/stonebranch

#################### INSTALLATION END ####################

# Run agent Install command during container creation
# and keep container running

CMD sh /root/stonebranch/unvinst --network_provider oms \
                                 --oms_servers $oms_servers \
                                 --oms_port $oms_port \
                                 --oms_autostart $oms_autostart \
                                 --ac_netname $ac_netname \
                                 --opscli $opscli \
                                 --user ubroker \
                                 --create_user yes \
        && tail -f /dev/null
```

Description:

Line	Command	Description
	All Lines starting with a # are ignored	
6	FROM centos:7	Builds an image using the centos:7 base image from the docker repository in the internet
9	MAINTAINER "Nils Buer <nils.buer@stonebranch.com>"	Meta data Information on who maintains the docker file. This info can be used to query for e.g. all images build by nils buer
16	ARG agentfile	Defines a new Variable agentfile
24	WORKDIR "/root/stonebranch"	Create a workdir in the image
27	ADD ${agentfile}/root/stonebranch/	ADD agent installer files to the image. Source is the local file system. agentfile is the parameter taken during the docker build command
30	ADD ucmd /etc/pam.d/	ADD pam configuration for centos Linux
33	RUN tar -xvzf /root/stonebranch/${agentfile} -C /root/Stonebranch	Extract agent installer file
40	CMD sh /root/stonebranch/unvinst --network_provider oms \ --oms_servers $oms_servers \ --oms_port $oms_port \ --oms_autostart $oms_autostart \ --ac_netname $ac_netname \ --opscli $opscli \ --user ubroker \ --create_user yes \ && tail -f /dev/null	Installs the Universal Agent and keeps the container alive (&& tail -f /dev/null)

10 Test Cases

The following basic test cases has been performed:

Case#	Assumed behavior	Result
docker run ..	After the docker run command is executed the agent should appear in the UC with status Active	Correct
docker stop ..	After stopping the container, the agent should go to status offline.	Correct
Docker start ..	After stopping the container and then restarting it the agent should go to status Active again	Correct
- Start the container - Reboot docker host - restart container	When shutting down the agent goes offline After the reboot the agent stays offline When starting the container the agent goes Active again.	Correct

11 Problems

The following provides a list of known problems:

#N	Problem	Description
1	Agent does not appear in the Universal Controller	a) Check firewall from the docker host is open to the OMS IP and Port configured in the env_agent_file b) Check in the env_agent_file that the ac_netname parameter is unique for all agents or set it to OPSAUTOCONF c) Check that your license key supports the number of agents you are connecting to the Controller

12 Document References

There are no document references.