Script to add a node license

ONTAP Select

David Peterson
November 21, 2019

This PDF was generated from https://docs.netapp.com/us-en/ontap-select/reference_api_script_anl.html on December 09, 2020. Always check docs.netapp.com for the latest.

Table of Contents

Script to add a node license 1

Script to add a node license

You can use the following script to add a license for an ONTAP Select node.

```
#!/usr/bin/env python
##------------------------------------------------------------------
#
# File: add_license.py
#
# (C) Copyright 2019 NetApp, Inc.
#
# This sample code is provided AS IS, with no support or warranties of
# any kind, including but not limited for warranties of merchantability
# or fitness of any kind, expressed or implied. Permission to use,
# reproduce, modify and create derivatives of the sample code is granted
# solely for the purpose of researching, designing, developing and
# testing a software application product for use with NetApp products,
# provided that the above copyright notice appears in all copies and
# that the software application product is distributed pursuant to terms
# no less restrictive than those set forth herein.
#
##------------------------------------------------------------------

import argparse
import logging
import json

from deploy_requests import DeployRequests


def post_new_license(deploy, license_filename):
    log_info('Posting a new license: {}'.format(license_filename))

    # Stream the file as multipart/form-data
    deploy.post('/licensing/licenses', data={},
                files={'license_file': open(license_filename, 'rb')})

    # Alternative if the NLF license data is converted to a string.
    # with open(license_filename, 'rb') as f:
    #    nlf_data = f.read()
    #    r = deploy.post('/licensing/licenses', data={},
    #                    files={'license_file': (license_filename, nlf_data)})


def put_license(deploy, serial_number, data, files):
    log_info('Adding license for serial number: {}'.format(serial_number))
```

```
43
44      deploy.put('/licensing/licenses/{}'.format(serial_number), data=data, files=
    files)
45
46
47  def put_used_license(deploy, serial_number, license_filename, ontap_username,
    ontap_password):
48      ''' If the license is used by an 'online' cluster, a username/password must be
    given. '''
49
50      data = {'ontap_username': ontap_username, 'ontap_password': ontap_password}
51      files = {'license_file': open(license_filename, 'rb')}
52
53      put_license(deploy, serial_number, data, files)
54
55
56  def put_free_license(deploy, serial_number, license_filename):
57      data = {}
58      files = {'license_file': open(license_filename, 'rb')}
59
60      put_license(deploy, serial_number, data, files)
61
62
63  def get_serial_number_from_license(license_filename):
64      ''' Read the NLF file to extract the serial number '''
65      with open(license_filename) as f:
66          data = json.load(f)
67
68          statusResp = data.get('statusResp', {})
69          serialNumber = statusResp.get('serialNumber')
70          if not serialNumber:
71              log_and_exit("The license file seems to be missing the serialNumber")
72
73          return serialNumber
74
75
76  def log_info(msg):
77      logging.getLogger('deploy').info(msg)
78
79
80  def log_and_exit(msg):
81      logging.getLogger('deploy').error(msg)
82      exit(1)
83
84
85  def configure_logging():
86      FORMAT = '%(asctime)-15s:%(levelname)s:%(name)s: %(message)s'
87      logging.basicConfig(level=logging.INFO, format=FORMAT)
```

```
    logging.getLogger('requests.packages.urllib3.connectionpool').setLevel(logging
.WARNING)


def main(args):
    configure_logging()
    serial_number = get_serial_number_from_license(args.license)

    deploy = DeployRequests(args.deploy, args.password)

    # First check if there is already a license resource for this serial-number
    if deploy.find_resource('/licensing/licenses', 'id', serial_number):

        # If the license already exists in the Deploy server, determine if its used
        if deploy.find_resource('/clusters', 'nodes.serial_number', serial_number):

            # In this case, requires ONTAP creds to push the license to the node
            if args.ontap_username and args.ontap_password:
                put_used_license(deploy, serial_number, args.license,
                                 args.ontap_username, args.ontap_password)
            else:
                print "ERROR: The serial number for this license is in use. Please
provide ONTAP credentials."
        else:
            # License exists, but its not used
            put_free_license(deploy, serial_number, args.license)
    else:
        # No license exists, so register a new one as an available license for later
use
        post_new_license(deploy, args.license)


def parseArgs():
    parser = argparse.ArgumentParser(description='Uses the ONTAP Select Deploy API to
add or update a new or used NLF license file.')
    parser.add_argument('-d', '--deploy', required=True, type=str, help='Hostname or
IP address of ONTAP Select Deploy')
    parser.add_argument('-p', '--password', required=True, type=str, help='Admin
password of Deploy server')
    parser.add_argument('-l', '--license', required=True, type=str, help='Filename of
the NLF license data')
    parser.add_argument('-u', '--ontap_username', type=str,
                        help='ONTAP Select username with privelege to add the
license. Only provide if the license is used by a Node.')
    parser.add_argument('-o', '--ontap_password', type=str,
                        help='ONTAP Select password for the ontap_username. Required
only if ontap_username is given.')
    return parser.parse_args()
```

```
127
128 if __name__ == '__main__':
129     args = parseArgs()
130     main(args)
```

Copyright Information

Copyright © 2020 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system-without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark Information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.