Script to resize cluster nodes

ONTAP Select

NetApp
November 21, 2019

This PDF was generated from https://docs.netapp.com/us-en/ontap-select/reference_api_script_resize.html on October 27, 2021. Always check docs.netapp.com for the latest.

Table of Contents

Script to resize cluster nodes 1

Script to resize cluster nodes

You can use the following script to resize the nodes in an ONTAP Select cluster.

```
#!/usr/bin/env python
##------------------------------------------------------------------------
#
# File: resize_nodes.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 sys

from deploy_requests import DeployRequests


def _parse_args():
    """ Parses the arguments provided on the command line when
executing this
        script and returns the resulting namespace. If all required
arguments
        are not provided, an error message indicating the mismatch is
printed and
        the script will exit.
    """

    parser = argparse.ArgumentParser(description=(
        'Uses the ONTAP Select Deploy API to resize the nodes in the
```

```
    cluster.'
 36         ' For example, you might have a small (4 CPU, 16GB RAM per
node) 2 node'
 37         ' cluster and wish to resize the cluster to medium (8 CPU,
64GB RAM per'
 38         ' node). This script will take in the cluster details and then
perform'
 39         ' the operation and wait for it to complete.'
 40     ))
 41     parser.add_argument('--deploy', required=True, help=(
 42         'Hostname or IP of the ONTAP Select Deploy VM.'
 43     ))
 44     parser.add_argument('--deploy-password', required=True, help=(
 45         'The password for the ONTAP Select Deploy admin user.'
 46     ))
 47     parser.add_argument('--cluster', required=True, help=(
 48         'Hostname or IP of the cluster management interface.'
 49     ))
 50     parser.add_argument('--instance-type', required=True, help=(
 51         'The desired instance size of the nodes after the operation is
complete.'
 52     ))
 53     parser.add_argument('--ontap-password', required=True, help=(
 54         'The password for the ONTAP administrative user account.'
 55     ))
 56     parser.add_argument('--ontap-username', default='admin', help=(
 57         'The username for the ONTAP administrative user account.
Default: admin.'
 58     ))
 59     parser.add_argument('--nodes', nargs='+', metavar='NODE_NAME',
help=(
 60         'A space separated list of node names for which the resize
operation'
 61         ' should be performed. The default is to apply the resize to
all nodes in'
 62         ' the cluster. If a list of nodes is provided, it must be
provided in HA'
 63         ' pairs. That is, in a 4 node cluster, nodes 1 and 2
(partners) must be'
 64         ' resized in the same operation.'
 65     ))
 66     return parser.parse_args()
 67
 68
 69 def _get_cluster(deploy, parsed_args):
 70     """ Locate the cluster using the arguments provided """
```

```
71
72      cluster_id = deploy.find_resource('/clusters', 'ip', parsed_args
    .cluster)
73      if not cluster_id:
74          return None
75      return deploy.get('/clusters/%s?fields=nodes' % cluster_id).
    json()['record']
76
77
78  def _get_request_body(parsed_args, cluster):
79      """ Build the request body """
80
81      changes = {'admin_password': parsed_args.ontap_password}
82
83      # if provided, use the list of nodes given, else use all the nodes
    in the cluster
84      nodes = [node for node in cluster['nodes']]
85      if parsed_args.nodes:
86          nodes = [node for node in nodes if node['name'] in
    parsed_args.nodes]
87
88      changes['nodes'] = [
89          {'instance_type': parsed_args.instance_type, 'id': node['id']}
    for node in nodes]
90
91      return changes
92
93
94  def main():
95      """ Set up the resize operation by gathering the necessary data
    and then send
96          the request to the ONTAP Select Deploy server.
97      """
98
99      logging.basicConfig(
100         format='[%(asctime)s] [%(levelname)5s] %(message)s', level
    =logging.INFO,)
101
102     logging.getLogger('requests.packages.urllib3').setLevel(logging
    .WARNING)
103
104     parsed_args = _parse_args()
105     deploy = DeployRequests(parsed_args.deploy, parsed_args
    .deploy_password)
106
107     cluster = _get_cluster(deploy, parsed_args)
```

```
108     if not cluster:
109         deploy.logger.error(
110             'Unable to find a cluster with a management IP of %s' %
    parsed_args.cluster)
111         return 1
112 
113     changes = _get_request_body(parsed_args, cluster)
114     deploy.patch('/clusters/%s' % cluster['id'], changes,
    wait_for_job=True)
115 
116 if __name__ == '__main__':
117     sys.exit(main())
```

Copyright Information

Copyright © 2021 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.