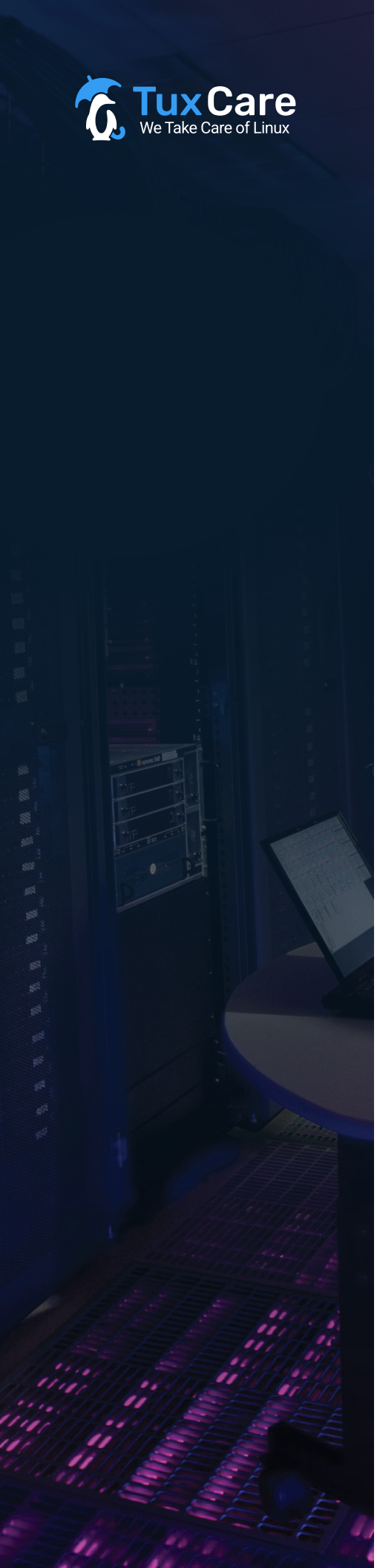

TuxCare
We Take Care of Linux

WHITE PAPER

ePortal API
Puppet Plan

Puppet provides the framework to run tasks on target systems. The following is a bash script that can run as such a task and demonstrates how to use the delete_server api to correctly remove a server from ePortal during tear down. It can be integrated with your other removal scripts or tasks to avoid manual operations.

puppet

You can create a separate ePortal account for API usage with:

```
kc.eportal -a api-user -p <password>
```

Unregister KernelCare agent through API call:

```
#!/bin/bash

EPORTAL_API_USERNAME=<your ePortal api user name>
EPORTAL_API_PASSWORD=<your ePortal api user password>
EPORTAL_URL='your ePortal URL'

#this is taken from the primary ip in the system. If awk is available, it
is used, but a fallback using other common tools is also provided
if hash awk 2>/dev/null; then
        IP_TO_UNREGISTER=`ip route get 1 | awk '{print $(NF-2);exit}'`
# using awk
else
        IP_TO_UNREGISTER=`ip route get 1 | cut -f 3 -d" "| head -1`
# simpler alternative for when awk is not available
fi

curl -kL -u "${EPORTAL_API_PASSWORD}"':'"${EPORTAL_API_PASSWORD}" -X POST
"${EPORTAL_URL}"'/admin/api/delete_server?ip='"${IP_TO_UNREGISTER}"
```

Example (unregister_server.sh):

```
#!/bin/bash

EPORTAL_API_USERNAME=admin
EPORTAL_API_PASSWORD=admin
EPORTAL_URL='http://192.168.246.110'

#this is taken from the primary ip in the system. If you want to pick a
different one, adjust the next line.
if hash awk 2>/dev/null; then
        IP_TO_UNREGISTER=`ip route get 1 | awk '{print $(NF-2);exit}'`
# using awk
else
        IP_TO_UNREGISTER=`ip route get 1 | cut -f 3 -d" "| head -1`
# simpler alternative for when awk is not available
fi

curl -kL -u "${EPORTAL_API_PASSWORD}"':'"${EPORTAL_API_PASSWORD}" -X POST
"${EPORTAL_URL}"'/admin/api/delete_server?ip='"${IP_TO_UNREGISTER}"
```

Puppet Plan

If you prefer to have a plan rather than a task, then you can create one from this script with the following steps:

- Create a new directory called “eportal_puppet”
- Inside this directory, create a bolt project:

```
bolt project init
```

- Create a scripts directory inside it
- Place the script above inside of it (call it “unregister_server.sh”)
- Create the bolt plan using:

```
bolt plan new eportal_puppet::unregister_server --script eportal_puppet/
scripts/unregister_server.sh
```

- Now your plan is ready and can be called directly with:

```
bolt plan run eportal_puppet:unregister_server -t <TARGETS>
```

Ad hoc run example with:

```
bolt plan run eportal_puppet::unregister_server -t 192.168.246.110
```

This can be called during machine tear down to properly remove the server from ePortal.

https://docs.tuxcare.com/eportal/#eportal-api