icinga2 Meldung via Pushover

Vorarbeiten:

- Anmeldung bei Pushover
- Device registrieren (7 Tage kostenlos, danach 4,..€ je Platform)
- erstellung eines Application/API Token

Scripte

Die Scripte können 1zu1 kopiert und eingefügt werden.
Bei der Ausarbeitung habe ich versucht, so nahe wie möglich an der Konfiguration für die Email-Benachrichtigung zu bleiben.
Grund: Man kann die Unterschiede zum Original besser sehen und weitere Notifications für andere Dienste (SMS, Pushbullet, Signal, Telegram, ...) erstellen

/etc/icinga2/scripts/pushover-service-notifiaction.sh

```
#!/bin/bash

################################################################################
#############################################
#
# full dokumentation on https://pushover.net/api
#
# POST an HTTPS request to https://api.pushover.net/1/messages.json
with the following parameters:
# token (required) - your application's API token
# user (required) - the user/group key (not e-mail address) of your
user (or you),
#     viewable when logged into our dashboard (often referred to as
USER_KEY in our documentation and code examples)
# message (required) - your message
#
#    Some optional parameters may be included:
# device - your user's device name to send the message directly to that
device,
#     rather than all of the user's devices (multiple devices may be
separated by a comma)
# title - your message's title, otherwise your app's name is used
# url - a supplementary URL to show with your message
# url_title - a title for your supplementary URL, otherwise just the
URL is shown
# priority - send as
#     -2 to generate no notification/alert,
#     -1 to always send as a quiet notification,
```

```
#      0 (default) to send notification with sound, vibration and
display(not in quiet hours)
#      1 to display as high-priority and bypass the user's quiet hours,
or
#      2 to also require confirmation from the user
#         retry
#         expire
# timestamp - a Unix timestamp of your message's date and time to
display to the user, rather than the time your message is received by
our API
# sound - the name of one of the sounds supported by device clients to
override the user's default sound choice
#
# That's it. Make sure your application is friendly to our API servers
and you're all set.
# For more information on each parameter, keep reading or jump to a
section at the left.
#
# Need help using our API or found an error in the documentation? Drop
us a line.
#
################################################################################
#############################################

logpath="/var/log/pushover_icinga.txt"
ICINGA2HOST="$(hostname)"
CURLPROXY=""
debug="0"

#########################################################
#Übergebene Parameter
#
#    PUSHOVERUSER = "$user.vars.pushover_user$"
#    PUSHOVERTOKEN = "$user.vars.pushover_token$"
#    PUSHOVERDEVICE = "$user.vars.pushover_device$"
#
#    PUSHOVERPRIORITY = "$host.vars.pushover_priority$"
#    PUSHOVERRETRY = "$host.vars.pushover_retry$"
#    PUSHOVEREXPIRE = "$host.vars.pushover_expire$"
#
#    NOTIFICATIONTYPE = "$notification.type$"
#    NOTIFICATIONCOMMENT = "$notification.comment$"
#    NOTIFICATIONAUTHOR = "$notification.author$"
#
#    ICINGALONGDATETIME = "$icinga.long_date_time$"
#
#    HOSTNAME = "$host.name$"
#    HOSTDISPLAYNAME = "$host.display_name$"
#    HOSTSTATE = "$host.state$"
#    HOSTOUTPUT = "$host.output$"
```

```
#
#####################################################

#***** Host Monitoring on $ICINGA2HOST *****
PUSHOVERMESSAGE=$(cat << EOF
***** Host Monitoring on icinga *****
$HOSTDISPLAYNAME is $HOSTSTATE!
Info:    $HOSTOUTPUT
When:    $ICINGALONGDATETIME
Host:    $HOSTNAME
EOF
)

#Wenn ein Kommentar eingetragen wurde (Downtimekommentar,
Benachrichtigungskommentar), wird dieser angehangen
if [ -n "$NOTIFICATIONCOMMENT" ]
then
PUSHOVERMESSAGE=$(cat << EOF
$PUSHOVERMESSAGE
Comment: $NOTIFICATIONCOMMENT
Author:  $NOTIFICATIONAUTHOR
EOF
)
fi

PUSHOVERTITLE="$NOTIFICATIONTYPE - Host $HOSTDISPLAYNAME is
$HOSTSTATE!"

#Wenn die Priorität 2 vergeben wurde, ist ein retry zwingend
erforderlich
#Sollte retry nicht gesetzt sein, wird er auf 30 gesetzt
if [ "$PUSHOVERPRIORITY" = "2" ] && [ "$PUSHOVERRETRY" = "" ]
then
    PUSHOVERRETRY="30"
fi

#Wenn die Priorität 2 vergeben wurde, ist ein expire zwingend
erforderlich
#Sollte expire nicht gesetzt sein, wird er auf 300 gesetzt
if [ "$PUSHOVERPRIORITY" = "2" ] && [ "$PUSHOVEREXPIRE" = "" ]
then
    PUSHOVEREXPIRE="300"
fi


#Kommando, um per curl die Pushover-message zu verschicken
failstate=$(curl \
  --silent \
  --insecure --proxy "$CURLPROXY" \
  --form-string "token=$PUSHOVERTOKEN" \
  --form-string "user=$PUSHOVERUSER" \
```

```
  --form-string "message=$PUSHOVERMESSAGE" \
  --form-string "title=$PUSHOVERTITLE" \
  --form-string "priority=$PUSHOVERPRIORITY" \
  --form-string "retry=$PUSHOVERRETRY" \
  --form-string "expire=$PUSHOVEREXPIRE" \
  --form-string "device=$PUSHOVERDEVICE" \
  --location https://api.pushover.net/1/messages.json)


#Wenn das debugging eingeschaltet ist, wird die folgende Meldung
ausgegeben
#$logpath sollte vorhanden sein und auf nagios:nagios gesetzt sein
if [ "$debug" = "1" ]
then
cat << EOF >> "$logpath"
###############################################
Debugging-Tool
###############################################
DatumZeit:      $(date)
PUSHOVERTOKEN:      $PUSHOVERTOKEN
PUSHOVERUSER:       $PUSHOVERUSER
PUSHOVERTITLE:      $PUSHOVERTITLE
PUSHOVERDEVICE:     $PUSHOVERDEVICE
PUSHOVERPRIORITY:   $PUSHOVERPRIORITY
PUSHOVERRETRY:      $PUSHOVERRETRY
PUSHOVEREXPIRE:     $PUSHOVEREXPIRE
HOSTDISPLAYNAME:    $HOSTDISPLAYNAME
ICINGALONGDATETIME: $ICINGALONGDATETIME
NOTIFICATIONTYPE:   $NOTIFICATIONTYPE
NOTIFICATIONCOMMENT:    $NOTIFICATIONCOMMENT
NOTIFICATIONAUTHOR: $NOTIFICATIONAUTHOR
ICINGA2HOST:        $ICINGA2HOST
HOSTNAME:       $HOSTNAME
HOSTSTATE:      $HOSTSTATE
HOSTOUTPUT:     $HOSTOUTPUT
pushover json output:   $failstate

EOF
fi
```

/etc/icinga2/scripts/pushover-host-notification.sh

```
#!/bin/bash

#############################################################################
###############################################
#
# full dokumentation on https://pushover.net/api
#
```

```
# POST an HTTPS request to https://api.pushover.net/1/messages.json
with the following parameters:
# token (required) - your application's API token
# user (required) - the user/group key (not e-mail address) of your
user (or you),
#     viewable when logged into our dashboard (often referred to as
USER_KEY in our documentation and code examples)
# message (required) - your message
#
#    Some optional parameters may be included:
# device - your user's device name to send the message directly to that
device,
#     rather than all of the user's devices (multiple devices may be
separated by a comma)
# title - your message's title, otherwise your app's name is used
# url - a supplementary URL to show with your message
# url_title - a title for your supplementary URL, otherwise just the
URL is shown
# priority - send as
#     -2 to generate no notification/alert,
#     -1 to always send as a quiet notification,
#      0 (default) to send notification with sound, vibration and
display(not in quiet hours)
#      1 to display as high-priority and bypass the user's quiet hours,
or
#      2 to also require confirmation from the user
#         retry
#         expire
# timestamp - a Unix timestamp of your message's date and time to
display to the user, rather than the time your message is received by
our API
# sound - the name of one of the sounds supported by device clients to
override the user's default sound choice
#
# That's it. Make sure your application is friendly to our API servers
and you're all set.
# For more information on each parameter, keep reading or jump to a
section at the left.
#
##############################################################################
#############################################

logpath="/var/log/pushover_icinga.txt"
ICINGA2HOST="$(hostname)"
CURLPROXY=""
debug="1"

#########################################################
```

```
#Übergebene Parameter
#
#    PUSHOVERUSER = "$user.vars.pushover_user$"
#    PUSHOVERTOKEN = "$user.vars.pushover_token$"
#    PUSHOVERDEVICE = "$user.vars.pushover_device$"
#
#    PUSHOVERPRIORITY = "$service.vars.pushover_priority$"
#    PUSHOVERRETRY = "$service.vars.pushover_retry$"
#    PUSHOVEREXPIRE = "$service.vars.pushover_expire$"
#
#    NOTIFICATIONTYPE = "$notification.type$"
#    NOTIFICATIONCOMMENT = "$notification.comment$"
#    NOTIFICATIONAUTHOR = "$notification.author$"
#
#    ICINGALONGDATETIME = "$icinga.long_date_time$"
#
#    HOSTNAME = "$host.name$"
#    SERVICENAME = "$service.name$"
#    HOSTDISPLAYNAME = "$host.display_name$"
#    SERVICESTATE = "$service.state$"
#    SERVICEOUTPUT = "$service.output$"
#
########################################################

#***** Service Monitoring on $ICINGA2HOST *****
PUSHOVERMESSAGE=$(cat << EOF
***** Service Monitoring on icinga *****
$SERVICEDISPLAYNAME on $HOSTDISPLAYNAME is ${SERVICESTATE}!
Info:    $SERVICEOUTPUT
When:    $ICINGALONGDATETIME
Service: $SERVICENAME
Host:    $HOSTNAME
EOF
)

#Wenn ein Kommentar eingetragen wurde (Downtimekommentar,
Benachrichtigungskommentar), wird dieser angehangen
if [ -n "$NOTIFICATIONCOMMENT" ]
then
PUSHOVERMESSAGE=$(cat << EOF
$PUSHOVERMESSAGE
Comment: $NOTIFICATIONCOMMENT
Author:  $NOTIFICATIONAUTHOR
EOF
)
fi

PUSHOVERTITLE="$NOTIFICATIONTYPE - $HOSTDISPLAYNAME -
$SERVICEDISPLAYNAME is $SERVICESTATE"
```

```
#Wenn die Priorität 2 vergeben wurde, ist ein retry zwingend
erforderlich
#Sollte retry nicht gesetzt sein, wird er auf 30 gesetzt
if [ "$PUSHOVERPRIORITY" = "2" ] && [ "$PUSHOVERRETRY" = "" ]
then
    PUSHOVERRETRY="30"
fi

#Wenn die Priorität 2 vergeben wurde, ist ein expire zwingend
erforderlich
#Sollte expire nicht gesetzt sein, wird er auf 300 gesetzt
if [ "$PUSHOVERPRIORITY" = "2" ] && [ "$PUSHOVEREXPIRE" = "" ]
then
    PUSHOVEREXPIRE="300"
fi


#Kommando, um per curl die Pushover-message zu verschicken
failstate=$(curl \
  --silent \
  --insecure --proxy "$CURLPROXY" \
  --form-string "token=$PUSHOVERTOKEN" \
  --form-string "user=$PUSHOVERUSER" \
  --form-string "message=$PUSHOVERMESSAGE" \
  --form-string "title=$PUSHOVERTITLE" \
  --form-string "priority=$PUSHOVERPRIORITY" \
  --form-string "retry=$PUSHOVERRETRY" \
  --form-string "expire=$PUSHOVEREXPIRE" \
  --form-string "device=$PUSHOVERDEVICE" \
  --location https://api.pushover.net/1/messages.json)


#Wenn das debugging eingeschaltet ist, wird die folgende Meldung
ausgegeben
#$logpath sollte vorhanden sein und auf nagios:nagios gesetzt sein
if [ "$debug" = "1" ]
then
cat << EOF >> "$logpath"
###############################################
Debugging-Tool
###############################################
DatumZeit:      $(date)
PUSHOVERTOKEN:      $PUSHOVERTOKEN
PUSHOVERUSER:       $PUSHOVERUSER
PUSHOVERTITLE:      $PUSHOVERTITLE
PUSHOVERDEVICE:     $PUSHOVERDEVICE
PUSHOVERPRIORITY:   $PUSHOVERPRIORITY
PUSHOVERRETRY:      $PUSHOVERRETRY
PUSHOVEREXPIRE:     $PUSHOVEREXPIRE
NOTIFICATIONTYPE:   $NOTIFICATIONTYPE
NOTIFICATIONCOMMENT:    $NOTIFICATIONCOMMENT
```

```
NOTIFICATIONAUTHOR: $NOTIFICATIONAUTHOR
HOSTDISPLAYNAME:    $HOSTDISPLAYNAME
SERVICEDISPLAYNAME: $SERVICEDISPLAYNAME
SERVICESTATE:       $SERVICESTATE
ICINGALONGDATETIME: $ICINGALONGDATETIME
ICINGA2HOST:        $ICINGA2HOST
SERVICEOUTPUT:      $SERVICEOUTPUT
SERVICENAME:        $SERVICENAME
pushover json output:   $failstate

EOF
fi
```

/etc/icinga2/conf.d/commands/pushover-notification.conf

```
object NotificationCommand "pushover-host-notification" {
import "plugin-notification-command"

command = [ SysconfDir + "/icinga2/scripts/pushover-host-
notification.sh" ]

env = {
    PUSHOVERUSER = "$user.vars.pushover_user$"
    PUSHOVERTOKEN = "$user.vars.pushover_token$"
    PUSHOVERDEVICE = "$user.vars.pushover_device$"

    PUSHOVERPRIORITY = "$host.vars.pushover_priority$"
    PUSHOVERRETRY = "$host.vars.pushover_retry$"
    PUSHOVEREXPIRE = "$host.vars.pushover_expire$"

    NOTIFICATIONTYPE = "$notification.type$"
    NOTIFICATIONCOMMENT = "$notification.comment$"
    NOTIFICATIONAUTHOR = "$notification.author$"

    ICINGALONGDATETIME = "$icinga.long_date_time$"

    HOSTNAME = "$host.name$"
    HOSTDISPLAYNAME = "$host.display_name$"
    HOSTSTATE = "$host.state$"
    HOSTOUTPUT = "$host.output$"
  }
}

object NotificationCommand "pushover-service-notification" {
  import "plugin-notification-command"

  command = [ SysconfDir + "/icinga2/scripts/pushover-service-
notification.sh" ]
```

```
  env = {
    PUSHOVERUSER = "$user.vars.pushover_user$"
    PUSHOVERTOKEN = "$user.vars.pushover_token$"
    PUSHOVERDEVICE = "$user.vars.pushover_device$"

    PUSHOVERPRIORITY = "$service.vars.pushover_priority$"
    PUSHOVERRETRY = "$service.vars.pushover_retry$"
    PUSHOVEREXPIRE = "$service.vars.pushover_expire$"

    NOTIFICATIONTYPE = "$notification.type$"
    NOTIFICATIONCOMMENT = "$notification.comment$"
    NOTIFICATIONAUTHOR = "$notification.author$"

    ICINGALONGDATETIME = "$icinga.long_date_time$"

    HOSTNAME = "$host.name$"
    SERVICENAME = "$service.name$"
    HOSTDISPLAYNAME = "$host.display_name$"
    SERVICESTATE = "$service.state$"
    SERVICEOUTPUT = "$service.output$"
  }
}
```

/etc/icinga2/conf.d/templates/pushover.conf

```
template Notification "pushover-host-notification" {
  command = "pushover-host-notification"

  states = [ Up, Down ]
  types = [ Problem, Acknowledgement, Recovery, Custom,
        FlappingStart, FlappingEnd,
        DowntimeStart, DowntimeEnd, DowntimeRemoved ]

  period = "24x7"
}

template Notification "pushover-service-notification" {
  command = "pushover-service-notification"

  states = [ OK, Warning, Critical, Unknown ]
  types = [ Problem, Acknowledgement, Recovery, Custom, FlappingStart,
FlappingEnd, DowntimeStart, DowntimeEnd, DowntimeRemoved ]

  period = "24x7"
}
```

/etc/icinga2/conf.d/notifications/pushover.conf

```
apply Notification "pushover-icingaadmin" to Host {
```

```
  import "pushover-host-notification"

  user_groups = host.vars.notification.pushover.groups
  users = host.vars.notification.pushover.users

  assign where host.vars.notification.pushover
  interval = 0 // disable re-notification
}


apply Notification "pushover-icingaadmin" to Service {
  import "pushover-service-notification"

  user_groups = host.vars.notification.pushover.groups
  users = host.vars.notification.pushover.users

  assign where host.vars.notification.pushover
  interval = 0 // disable re-notification
}
```

/etc/icinga2/conf.d/users/alle.conf

```
object User "patrick" {
  import "generic-user"

  display_name = "Patrick"
  #  groups = [ "icingaadmins" , "icingaadmins-pushover" ]
  groups = [ "icingaadmins-pushover" ]

  email = "root@localhost"

  #Pushover Messages
  vars.pushover_user = "user-Token"
  vars.pushover_token = "API-Token"
  #Devices
  #(mehrere devices = [ "device1", "device2" ]
  vars.pushover_device = [ "device" ]
  # vars.pushover_priority = "2"
  # retry >= 30
  # vars.pushover_retry = "30"
  # expire <= 10800
  # vars.pushover_expire = "300"

}
```

Kurze Beschreibung:

- device:

    - Devices können einzeln angegeben werden ([„device1“ , „device2“])
    - Wenn keine Device gesetzt ist, wird die Benachrichtigung an alle Devices gesendet
- Standard-Priority = 0
- Wenn Priorität = 2, dann muss retry und expire gesetzt sein. Sollte nichts gesetzt sein, wird via Script retry auf 30 und expire auf 300 gesetzt

/etc/icinga2/conf.d/usergroups/alle.conf

```
object UserGroup "icingaadmins-pushover" {
  display_name = "Icinga 2 Admin Group - Pushover"
}
```

/etc/icinga2/conf.d/hosts/ein_host.conf

```
object Host "foobar" {
  import "generic-host"

  display_name = "Foo bar"
  address = "<IP-Adresse>"

  #Benachrichtigung via email
  vars.notification["mail"] = {
    groups = [ "icingaadmins" ]
  }

  #Benachrichtigung via Pushover
  vars.notification["pushover"] = {
    groups = [ "icingaadmins-pushover" ]
  }

}
```

From:
https://wiki.da-checka.de/ - **PSwiki**

Permanent link:
https://wiki.da-checka.de/doku.php/wiki/programme/icinga2/pushover?rev=1570024160

Last update: **2019/10/02 15:49**