Übungsideen: Nagios/check_mk

- Erstellen Sie ein Arbeitsverzeichnis, das Sie unter Versionsverwaltung durch git stellen.
- Erstellen Sie zwei Unterverzeichnisse: `check_mk` für die lokalen Erweiterungen von check_mk und `mrpe` für Nagios-kompatible Plugins, die über mrpe in check_mk/OMD eingebunden werden können.
- Übernehmen Sie die Makefiles aus den bereitgestellten Materialien. Damit können Sie im jeweiligen Verzeichnis „entwickeln“, Ihren Arbeitsfortgang mit git verwalten und zu Testzwecken mit `make install` die lokale check_mk Erweiterung bzw. das Nagios Plugin installieren. Im Falle des Nagios Plugins muss die Datei `/etc/check_mk/mrpe.cfg` stimmig angepasst werden.

Workshop: Steigen Sie mit dem Beispiel zu den bash-Prozessen ein und Experimentieren Sie selbständig weiter – gerne auch auf Ihrem eigenen Server.

Prozessanzahl

Die Befehlssequenz `ps -C bash | wc -l` zählt die Anzahl der Prozesse mit dem Namen „bash“. Erstellen Sie einen Check für die Anzahl der Bash Prozesse auf der git-Desktop Maschine – sowohl als nagios-Plugin als auch als lokale check_mk Erweiterung.

- Beginnen Sie mit fest codierten Leveln für Warning/Critical
- Erweitern Sie den Nagios Check um Kommandozeilen-Optionen für die Warning/Critical-Level.
- Wie kann man die Schwellwerte für den Lokalen Check beeinflussen?Weiter

Zu Testzwecken können Sie auf der git-Desktop Maschine weitere Terminalfenster öffnen, um die Zahl der bash Prozesse zu vergrößern.

- Erweitern Sie das Skript um eine Option, den Prozessnamen auf der Kommandozeile zu übergeben. Funktioniert das für beide Verfahren?

Weitere Ideen – evtl auch für den eigenen Server

- Zahl der angemeldeten Benutzer? Ist ein Benutzer mit root-Rechnen eingeloggt?
- (Komplex) Zeitpunkt der letzten Synchronisation (linbo/rembo)
- (Je nach Strategie) Backupkontrolle?
- (Nagios Plugins vorhanden) USV Checks?