Audit Log (Revisionsprotokoll)

Interaktionen und Ergebnisse der Verwendung von Ressourcen werden protokolliert, um sie später auswerten zu können - zum Beispiel im Fehlerfall, in der Schadensaufklärung oder für die Abrechnung der Nutzung (Nutzungsgebühren).

Über das Audit Log File

Das Audit-Log leitet alle Änderungen an Ressourcen vom Server aus weiter. Über selbst geschriebene Plugins kann man diese Informationen nutzen, um z.B. Abrechnungen zu erstellen oder Maschinenzeiten grafisch auszuwerten. Diese Funktion bietet auch Möglichkeiten für eine detaillierte Analyse und Optimierung der Maschinennutzung.

Als zusätzliche Option können eigene Prozesse an die API gebunden werden. Diese ermöglichen einen direkten Zugriff auf die Funktionen des Servers über die API und eröffnen die Möglichkeit für Massenänderungen oder die zeitgesteuerte Zuweisung von Rollen.

Beim Erzeugen von Vorratsdaten ist stets dabei darauf zu achten, nur die minimal notwendigen Daten zu speichern und auszuwerten, um die Privatsphäre der Nutzer zu schützen und das Vertrauen nicht zu verletzen. FabAccess ist hier sehr sparsam und schreibt generisch je Ereignis eine Zeile im JSON-Format. Der Zeitstempel ist dabei im Unix-Format:

```
{"timestamp":<UNIX Zeitstempel>,"machine":"Ressourcen-ID","state":"<Status> <Benutzer>"}
```

Bffh protokolliert Zustandsänderungen in der Audit-Log-Datei (der Pfad wird über `auditlog_path` in der Konfigurationsdatei (*.dhall) festgelegt). Ausschnitt einer Beispiel Log File:

```
{"timestamp":1726239904,"machine":"Fokoos","state":"inuse local_lab_admin"}
{"timestamp":1726239905,"machine":"Fokoos","state":"free"}
{"timestamp":1726239932,"machine":"Ender","state":"inuse local_lab_admin"}
{"timestamp":1726239933,"machine":"Ender","state":"free"}
{"timestamp":1726240081,"machine":"Ender","state":"inuse local_lab_admin"}
{"timestamp":1726240088,"machine":"Ender","state":"free"}
{"timestamp":1726240112,"machine":"Mjolnir","state":"inuse local_lab_admin"}
{"timestamp":1726240122,"machine":"Mjolnir","state":"disabled"}
{"timestamp":1726240125,"machine":"Mjolnir","state":"disabled"}
```

{"timestamp":1726240126,"machine":"Mjolnir","state":"blocked local_lab_admin"}
{"timestamp":1726240128,"machine":"Mjolnir","state":"free"}
{"timestamp":1726240132,"machine":"Mjolnir","state":"inuse local_lab_admin"}
{"timestamp":1726240134,"machine":"Mjolnir","state":"free"}
{"timestamp":1726240139,"machine":"BashMachine","state":"inuse local_lab_admin"}
{"timestamp":1726240141,"machine":"BashMachine","state":"free"}

Parsing mit JQ

```
sudo apt install jq
```

Allgemeines Parsen

```
jq . /var/log/bffh-audit.json
```


```
{
  "timestamp": 1727110784,
  "machine": "Mjolnir",
  "state": "inuse local_lab_admin"
}
{
  "timestamp": 1729189184,
  "machine": "Mjolnir",
  "state": "free"
}
{
  "timestamp": 1729189186,
  "machine": "Mjolnir",
  "state": "inuse local_lab_admin"
}
```


UTC Timestamps

Parsen mit allgemeingültiger UTC-Zeitstempelformatierung:


```
{
  "timestamp": "2024-09-23T18:59:44 CET",
  "machine": "Mjolnir",
```

```
  "state": "inuse local_lab_admin"
}
{
  "timestamp": "2024-10-17T20:19:44 CET",
  "machine": "Mjolnir",
  "state": "free"
}
{
  "timestamp": "2024-10-17T20:19:46 CET",
  "machine": "Mjolnir",
  "state": "inuse local_lab_admin"
}
```

Mit lokaler Zeitzone

Oder zum Beispiel mit der Zeitzone `Europe/Berlin`:

```
TZ=Europe/Berlin jq '.|.timestamp |= strflocaltime("%Y-%m-%dT%H:%M:%S %Z")' /var/log/bffh-audit.json
```

```
{
  "timestamp": "2024-09-23T18:59:44 CET",
  "machine": "Mjolnir",
  "state": "inuse local_lab_admin"
}
{
  "timestamp": "2024-10-17T20:19:44 CET",
  "machine": "Mjolnir",
  "state": "free"
}
{
  "timestamp": "2024-10-17T20:19:46 CET",
  "machine": "Mjolnir",
  "state": "inuse local_lab_admin"
}
```

oder noch lesbarer durch Umstellen des Formats:

```
TZ=Europe/Berlin jq '.|.timestamp |= strflocaltime("%d.%m.%Y %H:%M:%S Uhr")' /var/log/bffh-audit.json
```

```
{
  "timestamp": "08.12.2024 00:57:27 Uhr",
  "machine": "zam-raum1-ecke8-macgyver",
  "state": "free"
}
{
  "timestamp": "08.12.2024 00:57:29 Uhr",
  "machine": "zam-raum1-ecke8-macgyver",
  "state": "inuse Admin"
}
{
  "timestamp": "08.12.2024 00:57:30 Uhr",
  "machine": "zam-raum1-ecke8-macgyver",
  "state": "free"
}
```


> Zum Parsen und Patterns testen empfehlen wir das Online-Tool https://jqplay.org

Grafische Darstellung des Audits und Auswertung

Für die Dashboard-Anzeige in Grafana können die Werkzeuge Alloy und Loki verwendet werden.

Achtung: Das Parsen des Audit Logs ist nur so lange zuverlässig, wie das Log nicht gelöscht oder rotiert wird. Für eine Langzeitarchivierung und eine tiefergehende Analyse von Statistiken sollte u.U. in Betracht gezogen werden den Audit in einer Datenbank zu speichern, z.B. PostgreSQL oder MariaDB. Mit Hilfe einer Datenbank könnten dann tiefergreifende Auswertungsabfragen programmiert werden, die sich dann wiederrum z.B. in Grafana grafisch auswerten und exportieren ließen, oder aber für die Abrechnung genutzt werden könnten. Es sollte auch beachtet werden, dass immer größer werdende json-Dateien sich nicht mehr performant parsen lassen, um darüber akkumulierte Gesamtstatistiken zu erzeugen.

Statistiken aus dem Audit gewinnen

Statistikabfragen könnten z.B. sein:

- Wer nutzt den Space bzw. bestimmte Ressourcen am meisten oder wenigsten?
- Welche Ressourcen werden generell am häufigsten genutzt?
- Welche Ressourcen werden generell am längsten genutzt?
- wer nutzt am meisten Dinge im Space gleichzeitig?
- Wann ist der Space in Nutzung und wann ruht er?
- wieviele Maschinenstunden kommen pro Tag, Woche, Monat, Jahr zusammen?

Log Rotation

Sind Log Files zu groß, werden sie in der Regel "rotiert" und in Archiven komprimiert, um Speicher zu sparen. Folgende Konfiguration kann angewendet werden. Wir heben maximal 10 als *.gz komprimierte Log-Archive auf. Die Logs werden automatisch rotiert, wenn sie 1 Megabyte überschreiten.

```
sudo vim /etc/logrotate.d/bffhd
```

```
/var/log/bffh/audit.json
{
	rotate 10
	size 1M
	copytruncate
	missingok
	notifempty
	compress
}
```

Logs manuell rotieren (testen)

```
sudo logrotate /etc/logrotate.d/bffhd -f
```

Weitere Links

Siehe auch Aktor: Logger (CSVlog)

Version #28
Erstellt: 2024-10-18 09:40:26 CEST von Mario Voigt (Stadtfabrikanten e.V.)
Zuletzt aktualisiert: 2025-03-16 00:42:12 CET von Mario Voigt (Stadtfabrikanten e.V.)