Getting Started / Onboarding

Nachdem ihr Was ist FabAccess? gelesen habt und hier gelandet seid, freuen wir uns über eure Entscheidung, FabAccess in eurer Umgebung zu implementieren und möchten euch mit hier einen reibungslosen Einstieg in die Nutzung ermöglichen.

Man fragt sich vielleicht, mit welchen Ressourcen begonnen werden und ob am Ende alles wie geplant funktionieren wird. Mit diesem **Getting Started** möchten wir euch die Angst nehmen und eine Anleitung geben, wie ihr Schritt für Schritt FabAccess bei euch einführen könnt.

Wir empfehlen euch

- sich vorher in die FabAccess Basics etwas einzulesen. Keine Angst! Am Anfang ist es nicht so einfach zu überschauen. Anhaltspunkte über die grundlegende Funktionsweise zu FabAccess findest du bei den Grundkonzepten.
- einen Überblick (Liste) über euren Space zu erhalten und zu klären, wo und wie genau ihr FabAccess einsetzen möchtet. Das bezieht sich auf die Ressourcen (z.B. Maschinen, Gerätschaften, ...) und etwaige Berechtigungen (Rollen), die später einmal Anwendung finden sollen.

Gründe für die Nutzung von FabAccess (Vorteile)

Warum sollten wir FabAccess nutzen? Die Gründe dafür können vielseitig sein. In der Regel sind es Ziele wie

- Ressourcen, die gefährlich sind, viel Strom verbrauchen oder in Anschaffung und Reparatur kostenintensiv sind, nur geschulten / eingewiesenen Personen überlassen
- Eine Übersicht über die Gerätschaften und ggf. deren Nutzung (Zeiten) und/oder Verbräuche (Strom) erfassen

Was wird für FabAccess allgemein benötigt?

- ein oder mehrere Personen, die einen FabAccess Server aufsetzen und betreuen können/wollen - hierzu gibt es verschiedene Rollen, z.B. IT-Administrator, Lab Manager (Schichtleiter, Werstattverantwortlicher, Bereichsleiter oder sonstige

Synonyme), Mitglieder, Studenten, sonstige Nutzer
- Zeit zur Integration von Hardware und Software
- Wille, sich Dokumentation durchzulesen und zu verstehen
- Motivation
- ein stabiles Wifi-Netzwerk (Mesh) am Standort - denn Aktoren sind häufig per 2,4 GHz IoT eingebunden

Was sollten wir integrieren?

In FabAccess lässt sich grundsätzlich jeder Wunsch integrieren. Allerdings ist es nicht immer sinnvoll, da es entweder zu komplex ist, zu viel Zeit kosten würde oder sozial nicht verträglich ist.

Für einen guten Einstieg empfehlen wir, mit den Ressourcen beginnen, deren Nutzung eingeschränkt werden muss. Sobald die Nutzer das System verstanden haben, kann das System auf andere Ressourcen ausgeweitet werden.

Und wo sollen wir anfangen?

Einen guten Start bieten 3D-Drucker, da sich diese einfach mit 230V-Zwischensteckern freischalten lassen. Zudem gibt es dadurch eine direkte Referenz, von welchem Nutzer der letzte Druck stammt.

Testen, Downloaden, Installieren, Konfigurieren

Da am Anfang nicht abzuschätzen ist, ob FabAccess euren Wünschen entspricht, empfehlen wir den Client zunächst auszuprobieren, um ein Gefühl dafür zu bekommen. Dafür wird ein Server benötigt. Hierzu gibt es verschiedene Möglichkeiten:

- den Testserver unter test.fab-access.org ausprobieren
- ein Debian-Paket installieren
- ein Docker Image installieren
- selber kompilieren
- ein Demo-Setup ausleihen
- eine Werkstatt besuchen, die FabAccess schon benutzt

Welche Hardware sollte gekauft bzw. vorgehalten werden?

FabAccess ist frei skalierbar und der Bedarf hängt von der Werkstatt und den Nutzern ab. Je mehr Räume, Ressourcen und Nutzer existieren, desto höher ist der Anspruch an das System.

Ein simples und kostengünstiges Setup von FabAccess lässt sich bereits mit einem Raspberry Pi, einer FRITZ!Box und ein ein paar 230V-Schaltsteckdosen ("Zwischenstecker") mit MQTT-Schnittstelle realisieren. Mit diesen können bereits die ersten Ressourcen in ein funktionsfähiges FabAccess Setup integriert werden. Alles andere an Hardware kann dann nachgerüstet werden, sobald das System läuft und die entsprechenden Berechtigungen vorhanden sind.

Nach oben hin gibt es bekanntlich keine Grenzen. Hinsichtlich Performance, Sicherheit und Administrierbarkeit ist für viele User durchaus auch ein Serveraufbau empfehlenswert. Hier gibt es verschiedenste Aufbauten.

Struktur-Beispiel Sternenlabor Plauen

- getrennte Wifi SSIDs in der Werkstatt: eine SSID für das "normale" Benutzernetz zum Surfen und eines für Geräte (IoT-Netz)
- Proxmox Server mit
  - einer virtuellen Maschine, in der wiederrum mehrere Docker-Container laufen:
    - Docker Container für FabAccess BFFH
    - Docker Container für einzelne Prometheus Exporter (z.B. FabAccess Exporter, MQTT Exporter)
    - Docker Container für Prometheus Service (Metriken sammeln)
    - Docker Container für Grafana (Monitoring / Alerting)
    - Docker Container für Watchtower (Docker Images automatisch aktualisieren, um alles auf den neuesten Stand zu halten)
    - sonstige Container für gekapselte Dienste
  - einer weiteren virtuellen Maschine, auf der Portainer läuft. Mit Portainer lassen sich die obigen Container über eine Web-Oberfläche gut administrieren/orchestrieren
  - Systemübersicht mit Glances (eine Art `htop`-Variante mit Web Interface)

- eine weitere virtuelle Maschine für OPNsense - eine Open Source Firewall, die es erlaubt, dass Nutzer aus dem normalen Benutzer-Wifi auf die FabAccess-Instanz zugreifen können, welche letztlich in einem anderen Netz agiert.

Ressourcennerfassung für die Integration mit FabAccess

Um einen detaillierten Überblick über eure Ressourcen, also z.B. Maschinen oder auch Dinge wie Außenleuchten oder Türschlösser, zu erhalten und eine reibungslose Integration mit FabAccess zu gewährleisten, haben wir eine gedankliche Tabellenstruktur vorbereitet, in der ihr die erforderlichen Informationen zu euren Dingen eintragen könnt.

Da FabAccess nicht nur für stationäre Maschinen konzipiert ist, könnt ihr auch andere Geräte in der Tabelle erfassen, deren Nutzung ihr nachverfolgen möchtet. Ressourcen können überdies auch virtuell oder menschlich sein (aka "Human Resources"). D.h. es lässt sich so auch Support-Personal (ein guter Name ist z.B. "Joris hilft") über FabAccess abbilden: Schaltet ein Nutzer z.B. einen Thekendienstmitarbeiter frei, dann wird dessen Arbeitszeit für die anfragende Person reserviert und kann danach beispielsweise zum Abrechnen verwendet werden (z.B. minütlich, viertelstündlich, stündlich, ...). Hierdurch lassen weiterführend Fallbeispiele konstruieren, die z.B. den Support innerhalb und außerhalb allgemeiner (ggf. kostenloser) Öffnungszeiten unterscheiden und ggf. ebenso Relevanz für die etwaige Berechnung haben.

Welche Informationen sollten erfasst werden? Grundsätzlich lassen sich Dinge nach verschiedenen Stichworten einordnen, z.B.

- nach Kategorie/Geräteart/Funktionsprinzip (Drucker, Computer, Laser, Nähmaschine, Werkzeugmaschine, Handgerät, CNC, etc., ...)
- nach Tätigkeit (z.B. Schleifen, Bohren, Reinigen, Schweißen, Heizen, Nähen, Messen, ...=
- nach Medium (Druckluft, Strom, Gas, Flüssigkeit, ...)
- nach Ort/Werkstattbereich (Gebäude, Raum, Zone, Bereich, Werkstatt, Gewerk, ...)
- nach Betriebsmittelart (beweglich, ortsfest, stationär, ...)
- nach Name oder eindeutiger Kennzeichnung (ID, Inventarnummer, Seriennummer, ...)
- nach Schutzklasse (ohe Schutz, Schutzerdung, Schutzisolierung, Schutzkleinspannung, ...)
- nach Ansprechpartner (z.B. Werkstattleiter, Bereichsleiter, Abteilung, ...)

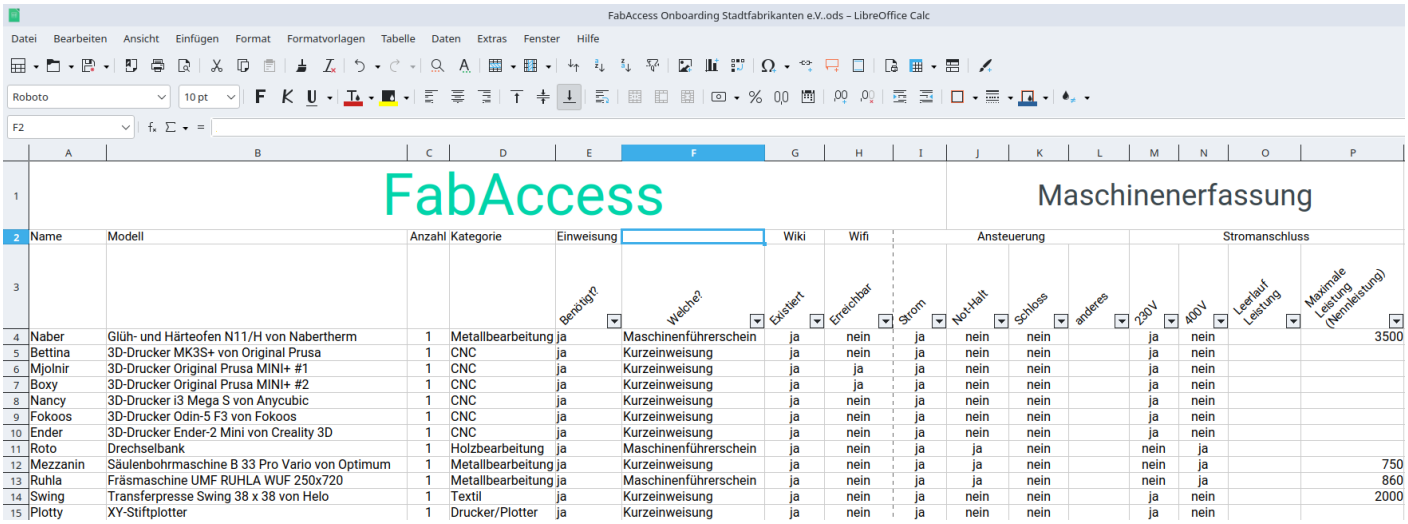
- nach Nutzungsberechtigung (ohne Einweisung, mit Kurzeinweisung, mit Maschinenführerschein, mit bestandenem Kurs, ...)

Diese Metainformationen können an verschiedenen Stellen eingearbeitet werden, z.B. in Name, Beschreibung und/oder Kategorie.

1. **Name, Modell und Anzahl**
   - Name: Gebt jeder Ressource einen eindeutigen Namen
   - Modell: Erfasst das genaue Modell der Ressource
   - Anzahl: Falls mehrere Ressourcen dasselbe Modell haben, notiert die Anzahl
2. **Beschreibung**
   - eine kurze Beschreibung der Sache - hilfreich, wenn die Ressource z.B. einen Namen trägt, aus dem sich der Zweck nicht sofort erschließen lässt (z.B. falls die Schleifmaschine "Hildegard" heißt)
3. **Kategorie**
   - Weist jeder Ressource einer Kategorie zu für eine bessere Übersicht. Ein Schema könnte z.B. sein: <Werkstatt>/<Geräteart>, also z.B. Holzwerkstatt/Bandschleifer
4. **Einweisung**
   - Klärt, ob eine Einweisung für die Nutzung der Ressource erforderlich ist
   - Notiert, wenn es spezielle Einweisungen gibt. Für viele Ressourcen ist keine Einweisung notwendig. Dennoch ist es hilfreich zu wissen, wer die Ressource nutzt und wie häufig sie in Gebrauch ist. Diese Informationen könnt ihr in FabAccess hinterlegen, um eine detaillierte Übersicht über die Nutzung zu erhalten.
5. **Wiki**
   - Falls bereits ein Wiki-Eintrag für eine Ressource existiert, fügt den entsprechenden Link in FabAccess hinzu
6. **Wifi**
   - Prüfe bei der Ressource, ob eine stabile Wifi Verbindung verfügbar ist. Ressourcen müssen nicht über Wifi freigeschaltet werden, jedoch ist das der einfachste Einstieg in FabAccess. Wenn ihr FabAccess längerfristig einsetzen möchtet oder schon einen Umbau in eurem Space geplant habt, bietet es sich an auch eine Ethernet-Anbindung an der Ressource zur Verfügung zu haben. So kann der Betrieb mit FabAccess noch reibungsloser verlaufen.
7. **Freischaltung**
   - Gebt an, wie die Ressource freigeschaltet wird (z.B., smarte Steckdosen für 230 V, Schütze für 400 V). Um eine Ressource freizuschalten, benötigt FabAccess die Möglichkeit, einen Freischaltbefehl an die Ressource zu senden. Die Ansteuerung kann je nach Ressourcentyp unterschiedlich erfolgen:

- Für einfache Ressourcen mit einem 230 V Anschluss: Hierfür können smarte Steckdosen verwendet werden
- Für Ressourcen mit 400 V Anschluss: Die Ansteuerung erfolgt über ein Schütz, um den Strom zuzuschalten
- Für Ressourcen, die nicht hart ausgeschaltet werden sollten: Die Ansteuerung kann evtl. über den Not-Aus-Kreis erfolgen. Es ist wichtig zu beachten, dass bei der Ansteuerung über die Stromzufuhr darauf geachtet werden muss, dass die Ressource eine Abschaltung des Stroms während des Betriebs verkraften kann. Für mobile Geräte, die nicht über den Strom geschaltet werden können, empfiehlt sich die Verwendung eines elektromechanischen Schlosses zur Freigabe einer Aufbewahrungsbox.

8. **Stromanschluss/Leistungsaufnahme**
   - Notiert den benötigten Stromanschluss (z.B., 230 V, 400 V)
   - Notiert auch die Leistungsaufnahme der Ressource. Die verbrauchte Leistung der Ressource kann entweder durch eigene Messgeräte oder spezielle Zwischenstecker gemessen werden. Diese Informationen sind nicht nur wichtig, um die Nutzungsdauer der Ressource zu verstehen, sondern ermöglichen auch eine Abschätzung der Betriebskosten oder ein automatisiertes Abschalten, wenn die Ressource länger als eine einstellbare Zeit nicht aktiv ist. Mit diesen Informationen könnt ihr nun mit der Modellierung der Berechtigungen fortfahren.



FabAccess Maschinenerfassung

Name	Modell	Anzahl	Kategorie	Einweisung		Wiki	Wifi	Ansteuerung				Stromanschluss			
				Benötigt?	Welche?	Existiert	Erreichbar	Strom	Not-Halt	Schloss	anderes	230V	400V	Leerlauf Leistung	Maximale Leistung (Nennleistung)
Naber	Glüh- und Härteofen N11/H von Nabertherm	1	Metallbearbeitung	ja	Maschinenführerschein	ja	nein	ja	nein	nein		ja	nein		3500
Bettina	3D-Drucker MK3S+ von Original Prusa	1	CNC	ja	Kurzeinweisung	ja	nein	ja	nein	nein		ja	nein		
Mjolnir	3D-Drucker Original Prusa MINI+ #1	1	CNC	ja	Kurzeinweisung	ja	ja	ja	nein	nein		ja	nein		
Boxy	3D-Drucker Original Prusa MINI+ #2	1	CNC	ja	Kurzeinweisung	ja	ja	ja	nein	nein		ja	nein		
Nancy	3D-Drucker i3 Mega S von Anycubic	1	CNC	ja	Kurzeinweisung	ja	nein	ja	nein	nein		ja	nein		
Fokoos	3D-Drucker Odin-5 F3 von Fokoos	1	CNC	ja	Kurzeinweisung	ja	nein	ja	nein	nein		ja	nein		
Ender	3D-Drucker Ender-2 Mini von Creality 3D	1	CNC	ja	Kurzeinweisung	ja	nein	ja	nein	nein		ja	nein		
Roto	Drechselbank	1	Holzbearbeitung	ja	Maschinenführerschein	ja	nein	ja	ja	nein		nein	ja		
Mezzanin	Säulenbohrmaschine B 33 Pro Vario von Optimum	1	Metallbearbeitung	ja	Kurzeinweisung	ja	nein	ja	ja	nein		nein	ja		750
Ruhla	Fräsmaschine UMF RUHLA WUF 250x720	1	Metallbearbeitung	ja	Maschinenführerschein	ja	nein	ja	ja	nein		nein	ja		860
Swing	Transferpresse Swing 38 x 38 von Helo	1	Textil	ja	Kurzeinweisung	ja	nein	ja	nein	nein		ja	nein		2000
Plotty	XY-Stiftplotter	1	Drucker/Plotter	ja	Kurzeinweisung	ja	nein	ja	nein	nein		ja	nein		

Beispiel einer manuell ausgefüllten Tabelle (nicht ganz vollständig)

> Diese Infos lassen sich zum Beispiel auch im **FabAccess Config Generator** erfassen, der viel Arbeit von obigen Konzepten generalisiert abnimmt!

Modellierung der Berechtigungen

FabAccess ermöglicht eine präzise Kontrolle über die Nutzung von Ressourcen. Das System basiert auf einem Rollensystem, das die Berechtigungen für Ressourcen in einer Rolle zusammenfasst. Nutzer können diese Rollen erhalten und erhalten dadurch die entsprechenden Ressourcenberechtigungen. Ressourcen verfügen über vier Berechtigungsfelder (**read**, **write**, **disclose**, **manage**). Diese Details finden sich unter RBAC (Benutzerrollen und Berechtigungen).

Für einen einfachen Einstieg in dieses System empfiehlt es sich, zwei Standardrollen zu verwenden:

1. **"Offene Rolle":**
   Jeder Nutzer, der sich im Space anmeldet, erhält diese Rolle. Mit dieser Rolle können alle frei zugänglichen Ressourcen verwendet werden, und es besteht die Möglichkeit, alle öffentlichen Ressourcen über die Ressourcenliste zu entdecken und Informationen abzurufen.
2. **Administrator-Rolle:**
   Diese Rolle ermöglicht das Anlegen von Nutzern, das Ansehen aller Ressourcen und das Verwalten von Ressourcen. Wichtig ist dabei, auch Berechtigungen für das Nutzermanagement hinzuzufügen. Sollten im Space weitere Unterstützer aktiv sein, die jedoch keine Nutzer und Rollen verwalten sollen, kann eine eigene Rolle mit Manage-Berechtigungen erstellt werden.

Für alle weiteren Rollen können diese nach Bedarf entsprechend den erforderlichen Einweisungen für die Ressourcen erstellt werden. Als Beispiel: Wenn für 3D-Drucker eine Einweisung erforderlich ist, kann eine Rolle "3D-Drucken" erstellt werden. Wenn jedoch zwischen verschiedenen 3D-Drucker-Typen unterschieden werden soll, können zwei separate Rollen erstellt werden, wie "3D-Drucken FDM" und "3D-Drucken SLA". Wenn Nutzer alle 3D-Drucker verwenden können sollen, kann entweder jedem Nutzer alle relevanten Rollen zugewiesen oder eine vererbte Rolle wie "3D-Drucken Gesamt" erstellt werden.

Es wird empfohlen, zu Beginn mit feinen Rollen zu arbeiten, da das Entfernen von Rollen für Nutzer im Nachhinein mehr Aufwand bedeutet.

Mit der Berechtigungstabelle könnt ihr erste Entscheidungen darüber treffen, wer auf welche Ressource zugreifen darf und welche Ressourcen gemeinsam in einer Berechtigung genutzt werden können. Ebenso könnt ihr festlegen, welche Ressourcen für alle zugänglich sein sollen und welche ausschließlich nach einer Freischaltung verwendet werden dürfen.

Die Installation von FabAccess **Difluoroborane** (BFFH) ist auf jedem Server möglich, der das Ausführen einer kompilierten Rust-Binary erlaubt. Das kann ein Bare Metal Linux Server, ein Proxmox, ein Docker-Container oder sonstige Systeme sein.

Neben der reinen Installation sind weitere Schritte der Vor- und Nachkonfiguration notwendig oder empfehlenswert, um ein sinnvolles und praxistaugliches System zu erhalten.

Vorüberlegungen und Best Practices

Die Installation von FabAccess **Difluoroborane** (BFFH) ist auf jedem Server möglich, der das Ausführen einer kompilierten Rust-Binary erlaubt. Das kann ein Bare Metal Linux Server, ein Proxmox, ein Docker-Container oder sonstiges System sein.

Neben der reinen Installation sind weitere Schritte der Vor- und Nachkonfiguration notwendig oder empfehlenswert, um ein sinnvolles und praxistaugliches System zu erhalten.

Separierte Netzwerke

Für ein funktionierendes FabAccess-System sollten die Geräte, die Ressourcen freischalten, in einem separaten Netzwerk betrieben werden - also zum Beispiel Tasmota Schaltsteckdosen oder FabReader. Dadurch wird das Risiko minimiert, dass unerwartete Freischaltungen stattfinden.

Wir separieren sozusagen Notebooks und Smartphones der Nutzer (und Gäste), die sich meistens im regulären Netz befinden, von den IoT-Geräten, die an den Ressourcen anhängen und die die FabAccess-Struktur bilden.

Die Netzwerkseparation kann adminstrativ durch verschiedene Wege erfolgen - zum Beispiel durch VLANs oder eigens für die Geräte aufgespannte Wifi-SSIDs.

Wifi SSIDs können auf 2,4 GHz oder 5 GHz ausgestrahlt werden und unterschiedlich anhand ihrer Frequenz benannt werden. Natürlich kann auch für beide Frequenzbänder die gleiche SSID vergeben werden. Je nach Setup sollte hier überlegt werden, was besser ins eigene Space-Konzept passt. Prinzipiell ist es häufig so, dass einige MQTT-fähige Geräte nur im 2,4 GHz Band arbeiten können. Für eine leichtere Unterscheidung kann eine passende Benennung von SSIDs und Geräten hilfreich sein, um Fehler schneller zu finden.

Nutzer anlernen und aktiv einbeziehen

- Erklärt das Konzept von FabAccess
- Erklärt, wie der Space FabAccess verwendet

Überautomatisierung vermeiden

Nicht alles, was automasiert werden kann, sollte auch automatisiert werden. Nicht jedes Werkzeug oder jede Ressource sollte weggeschlossen bzw. verriegelt werden. Das nimmt u.U. zu viele Grundrechte oder übertechnisiert den Ausstattungsbedarf und die Wartung des Spaces.

FabAccess sicher machen (physisch + digital)

- Vermeide den physischen Zugriff auf Schalter: Zum Schutz vor Manipulation können z.B. Tasmota Stecker oder Shellys außer Reichweite platziert werden - zum Beispiel unter dem Tisch oder über oder hinter dem Regal
- Etabliere eine sinnvolle Ausgabe von DESFire Karten zum Zugriff auf Ressourcen, um die unerlaubte Weitergabe durch Nutzer zu unterbinden
- Verwende sichere Verschlüsselung für die Funkverbindung im MQTT-Netzwerk und anderen Stellen, wo es möglich ist (TLS Verschlüsselung, LetsEncrypt Zertifikate)
- Verwende aktuelle Standardsicherheitswerkzeuge wie Firewalls, fail2ban und Co.
- sichere (komplexe) Passwörter für Wifi SSIDs verwenden
- Falls FabAccess BFFH öffentlich gehosted wird, also der Standardport 59961 nach außen hin geöffnet ist, sollte trotz TLS-gesicherter Client-Kommunikation bedacht werden, dass Angriffe von außen prinzipiell möglich sind. Im Falle, dass ein Server auf irgendeine Weise erfolgreich gehacked wird, sollten Sicherheitsmaßnahmen in der Konfiguration von Anfang an getroffen werden:
  - dedizierten Benutzer `bffh` anlegen, der den Service verwaltet und startet/stoppt. BFFH sollte nie als `root` Nutzer laufen, sondern stets beschränkte Rechte haben. Insbesondere dann, wenn BFFH z.B. Zugriff auf Aktoren-/Initiatoren-Plugins hat, die z.B. weitere Prozesse ausführen (z.B. Python, Perl, Bash).
  - bei Hacking besteht in der Regel immer das Risiko des Verlusts oder Offenlegung persönlicher Daten wie Benutzernamen oder Passwörtern. Glücklicherweise ist FabAccess so datensparsam, dass hier maximal Benutzernamen und Rollenzuweisungen (an denen ein Mitgliedsstatus ggf. genauer ausgelesen werden kann) verloren gehen können (die Argon2 Passwort-Hashes sind sehr sicher im Diebstahlfall!)
  - Insbesondere, weil FabAccess auf den Zugriff von physischer Hardware ausgelegt ist, wären etwaige Maschinenbestromungen, z.B. nachts unter fehlender Anwesenheit, durch illegale Fremdeinwirkung u.U. fatal. Hier

könnte z.B. auch entgegengewirkt werden, indem der FabAccess Server oder das verwendete IoT-Wifi-Netz nur zu bestimmten Uhrzeiten für Nutzer erreichbar ist.

Verwende einen Test bzw. Staging Server

Neben dem Produktivserver wird empfohlen eine Staging Instanz zu installieren, die baugleich zur Produktivinstanz ist, jedoch verwendet werden kann, um etwaige Änderungen vorher auszutesten. Insbesondere beim Integrieren von neuen Ressourcen oder beim Ändern von Benutzerrollen- und berechtigungen in einem lebendigen Werktattumfeld kann sonst u.U. turbulent werden. Es ist mit geringem Aufwand möglich mehrere BFFH Instanzen (auch ohne Docker) auf dem gleichen Server laufen zu lassen.

Erstelle regelmäßig Backups

Auch bei FabAccess ist es wichtig, regelmäßig Backups der Benutzerdatenbank und der Konfiguration zu erstellen. Nichts wirft einen mehr zurück als der Verlust aller zugewiesenen Rollen durch einen Ausfall.

Es gibt verschiedene Backup-Strategien, je nach vorhandener IT-Infrastruktur, eigenem Geschmack und natürlich auch der Art und Weise, wie FabAccess installiert wurde.

Backups könnten sein:

- kompletter System-Dump (Kopie der Festplatte mitsamt Betriebssystem)
- inkrementelle Backups des Systems oder der FabAccess-relevanten Daten
- Docker-Container Backup
- ein git-Repository, in das wir Konfigurationsdateien pushen und versionieren

Häufig lohnt es sich 1-2 Backup-Strategien zu kombinieren.

> Tipp: Backups erstellen ist gut! Noch besser: **Testen**, ob die Backups im Ernstfall dann auch funktionieren, also ob ein Restore auch problemlos möglich ist!

Siehe auch Backup einrichten

Audit und Zeitsynchronisation

Für Abrechnungsvorgänge oder zum Nachvollziehen von Ereignissen in der Werkstatt sind genaue Zeitstempel wichtig. Es ist deshalb ratsam den Server, auf dem FabAccess BFFH läuft, mit einem Zeitserver zu synchronisieren. Generell ist es empfehlenswert, wenn sich

alle Serverdienste und Clients in der Werkstatt auf den gleichen Zeitserver beziehen.

Bei Verwendung von kleinen Geräten wie einem Raspberry Pi ist es zudem ggf. hilreich ein RTC (Echtzeituhr) einzubauen, da diese meistens keine eigene Uhrzeit haben. Im Falle eines Internetausfalls hat der Server so trotzdem eine valide Uhrzeitquelle.

Deine Fragen sind nicht beantwortet?

Schau unsere unsere FAQs. Dort finden sich viele häufige Fragen und Antworten!

Version #52
Erstellt: 2024-10-16 12:42:34 CEST von Mario Voigt (Stadtfabrikanten e.V.)
Zuletzt aktualisiert: 2025-03-17 17:05:23 CET von Mario Voigt (Stadtfabrikanten e.V.)