



GUIDA AMMINISTRATORE

Azura-Relay

AzuraRelay è "relè in una scatola" contenente un'applicazione Web leggera e server Icecast in grado di connettersi e trasmettere un'istanza madre di AzuraRelay.

Se vuoi offrire i tuoi flussi agli utenti di tutto il mondo o accettare un grande volume di traffico in arrivo, AzuraRelay è il compagno perfetto per AzuraCast.

Requisiti di installazione dei genitori

Prima di installare AzuraRelay, assicurati che l'installazione "genitore" AzuraCast:

- è aggiornato all'ultima versione,
- ha le porte radio dirette (cioè 8000, 8010) esposte (cioè non è dietro un proxy CloudFlare)

Requisiti di sistema

- Una CPU x86 (x86_64) a 64 bit (dispositivi ARM, come il Raspberry Pi 3/4, non sono supportati.)
- 512MB o superiore di RAM
- 20GB o più di spazio sul disco rigido

Per gli host Linux, il `sudo`, `curl` e `git` pacchetti devono essere installati prima dell'installazione. La maggior parte delle distribuzioni Linux include già questi pacchetti.

Installazione

AzuraRelay è alimentato da Docker e utilizza immagini pre-costruite che contengono ogni componente del software. Non preoccuparti se non hai molta familiarità con Docker; i nostri facili strumenti di installazione gestiranno l'installazione di Docker e Docker Compose per te e gli aggiornamenti sono molto semplici.

Connettiti al server o al computer su cui vuoi installare AzuraRelay tramite un terminale SSH. Dovresti essere un utente amministratore con accesso root o la possibilità di utilizzare il `sudo` comando.

Scegli una directory di base sul tuo computer host che AzuraRelay può utilizzare. Se sei su Linux, puoi seguire i passaggi riportati di seguito per utilizzare la directory consigliata:

Finestra terminale

```
mkdir -p /var/azurarelay
cd /var/azurarelay
```

Utilizza questi comandi per scaricare il nostro Docker Utility Script, impostarlo come eseguibile e quindi eseguire il processo di installazione di Docker:

Finestra terminale

```
curl -L https://raw.githubusercontent.com/AzuraCast/AzuraRelay/main/docker.sh > docker.sh
```

```
chmod a+x docker.sh  
./docker.sh install
```

I suggerimenti sullo schermo ti mostreranno come sta procedendo l'installazione.

Aggiornamento

Utilizzando lo script di utilità Docker incluso, l'aggiornamento è semplice come eseguire:

Finestra terminale

```
./docker.sh update-self  
./docker.sh update
```

Backup & Ripristino

AzuraCast include un'utilità integrata per il backup dell'installazione, incluso il database, tutte le stazioni e le loro impostazioni, e facoltativamente i supporti locali legati con la stazione (s).

Modi per fare il backup

Tramite l'interfaccia Web

La sezione “Amministrazione di sistema” della tua interfaccia web AzuraCast ha una pagina dedicata “Backup”, in cui puoi configurare backup automatizzati periodici, eseguire un backup una tantum e scaricare file di backup esistenti.

I backup vengono eseguiti tramite il pannello Web non interrompono le trasmissioni o scollegano gli ascoltatori.

Tramite la linea di comando

Se vuoi un maggiore controllo su dove stanno andando i tuoi file di backup o vuoi sfruttare strumenti preesistenti (come attività cron) per gestire il backup, puoi utilizzare anche lo strumento di utilità Docker per generare backup.

La versione più elementare di questo comando è:

Finestra terminale

```
cd /var/azuracast
./docker.sh backup path-to-backup.zip
```

Puoi anche passare il flag `esclude media` per eseguire il backup solo del database e delle statistiche, ma non i supporti stessi, il che riduce significativamente la dimensione del file di backup (ma dovresti assicurarti di eseguire il backup dei tuoi supporti altrove):

Finestra terminale

```
cd /var/azuracast
./docker.sh backup path-to-backup.zip --exclude-media
```

Sia i formati `.zip` che `.tar.gz` sono supportati per i backup. Il formato corretto sarà determinato automaticamente dall'estensione del nome del file specificato per il file di backup.

Ripristino di un backup

Questi sono i passaggi generali per ripristinare un backup di AzuraCast:

Questo potrebbe richiedere un po' di tempo...

I backup di grandi dimensioni (100GB +) possono richiedere molto tempo per ripristinare completamente sul server e possono utilizzare grandi quantità di CPU e o memoria.

1. **Installare AzuraCast ([Vedi la nostra guida](#))**
 - Se si dispone di un file di backup più vecchio e si riscontrano problemi con il ripristino, installare AzuraCast su `stabler` rispondendo Y alla domanda “Preferire versioni di rilascio stabili di AzuraCast?”
2. **Dopo aver terminato l'installazione, eseguire il comando di ripristino:**
 - `./docker.sh restore path-to-backup.zip`
3. **Se avete usato il `stable` installazione a causa di un vecchio file di backup si dovrebbe ora tornare al `rolling-release` tramite il seguente comando e risposta N alla domanda se si vuole utilizzare la versione stabile:**
 - `./docker.sh setup-release`
4. **Infine eseguire il seguente comando per aggiornare l'installazione alla versione corrente di rilascio di `rolling`:**
 - `./docker.sh update`

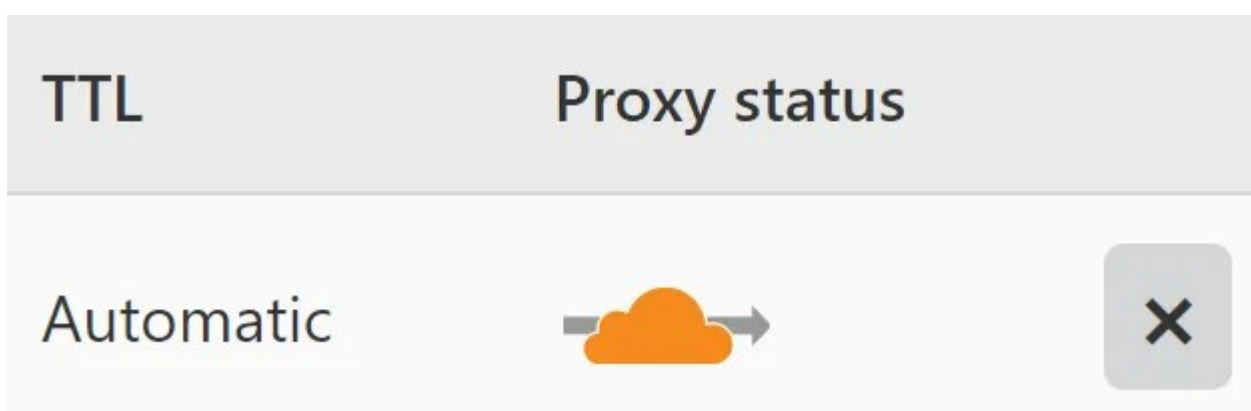
Cloudflare

Cloudflare è un fornitore leader di servizi di proxying inverso e CDN per il web. Il loro livello gratuito offre enormi vantaggi nelle prestazioni e nella protezione che puoi sfruttare mentre utilizzi AzuraCast, con alcuni importanti avvertimenti. Questo documento descrive come utilizzare Cloudflare con AzuraCast.

Abilitare Cloudflare con AzuraCast

AzuraCast ha il pieno supporto per l'utilizzo della protezione di Cloudflare di fronte al server radio.

L'abilitazione del supporto Cloudflare dal pannello di controllo Cloudflare è semplice quanto garantire che la piccola "cloud" accanto al dominio del server radio (o sottodominio) sia arancione, indicando che la protezione è abilitata.



Le impostazioni predefinite per Cloudflare saranno sufficienti, anche se puoi anche passare dalla modalità SSL "Flessibile" a "Full" se vuoi (dal momento che AzuraCast stesso supporta HTTPS). La modalità SSL "rigorosa" non è raccomandata, in quanto ciò richiede di mantenere un certificato SSL aggiornato su AzuraCast stesso (che è possibile, ma non necessario).

Una volta abilitato il supporto Cloudflare per il tuo dominio (o sottodominio della stazione radio) dal pannello di controllo Cloudflare, devi solo abilitare una funzionalità all'interno di AzuraCast per consentire agli ascoltatori di connettersi:

Abilitare il Web Proxy per le trasmissioni radio

Una limitazione principale imposta da Cloudflare è che non inoltrano le connessioni in entrata al server che non provengono dalle tradizionali porte web (cioè 80 e 443). Per impostazione predefinita, AzuraCast serve ogni stazione radio sulla propria porta distinta in un range da 8000 a 9000. Ciò significa che i tuoi ascoltatori normalmente non sarebbero in grado di connettersi.

Use Web Proxy for Radio

Use Web Proxy for Radio

By default, radio stations broadcast on their own ports (i.e. 8000). CloudFlare or accessing your radio station by SSL, you should enable all radio through the web ports (80 and 443).

Per fortuna abbiamo già costruito una soluzione a questo problema! Nell'amministrazione del sistema di AzuraCast, nella pagina "Impostazioni di sistema", abbiamo una casella di controllo etichettata "Usa proxy Web per Radio". Abilitare questa casella di controllo e tutti gli URL di riproduzione della stazione in tutto il sistema verranno aggiornati per utilizzare automaticamente i collegamenti proxy della porta web, che sono completamente accessibili anche quando la protezione Cloudflare è abilitata.

Note Importanti

Non Mettere In Cache I Contenuti Radiofonici

Cloudflare non consente di servire i contenuti della radio web tramite il CDN memorizzato nella cache, poiché il contenuto è un file audio in streaming che non è memorizzabile tramite i loro servizi tradizionali. Il modo più semplice per incaricare Cloudflare di non memorizzare nella cache le trasmissioni radiofoniche è tramite la funzione Regole di pagina.

Per creare una nuova regola di pagina:

1. Visita il pannello di controllo Cloudflare, poi "Regole", quindi "Regole di pagina".
2. Fare clic su "Crea regola di pagina".
3. Inserisci l'URL dell'installazione di AzuraCast, seguito da uno dei modelli riportati di seguito. Il modello finito dovrebbe assomigliare `azuracast.example.com/radio/*`
4. Scegli l'impostazione "Livello di cache", impostala su "Bypass".
5. Fare clic su "Salva e implementa la regola della pagina" nella parte inferiore della pagina.

Ripeti questi passaggi per tutti i seguenti modelli:

- `/radio/*`
- `/listen/*`
- `/hls/*`

Disabilitare sempre Rocket Loader

AzuraCast non è compatibile con l'ottimizzazione Rocket Loader offerta da Cloudflare. Se Rocket Loader è abilitato, incontrerai errori imprevisti con AzuraCast.

Per risolvere questo errore, è necessario accedere alla dashboard di Cloudflare e disattivare le impostazioni di Rocket Launcher.

Informazioni su Incoming DJ Connections

Poiché Cloudflare blocca tutte le connessioni in arrivo che non si trovano nelle porte web standard, blocca anche le connessioni in arrivo che i tuoi streamer / DJ utilizzerebbero per trasmettere alla tua stazione. A differenza dell'esperienza che offriamo agli ascoltatori attraverso il nostro proxy radio, non possiamo proxyare la trasmissione in arrivo allo stesso modo per motivi tecnici.

Si consiglia di istruire gli streamer/DJ a connettersi al server utilizzando il suo indirizzo IP piuttosto che il suo nome di dominio protetto da Cloudflare. Questo permetterà loro di connettersi senza alcun problema, e senza esporre il vostro IP di origine alla vostra base di ascoltatore regolare.

Informazioni su AzuraRelay Instances

Se si utilizzano istanze AzuraRelay che dovrebbero trasmettere un'installazione protetta da Cloudflare, è necessario utilizzare l'indirizzo IP dell'installazione come URL di base per il relè (in un formato come <http://127.0.0.1>), piuttosto che l'indirizzo pubblico protetto da Cloudflare. In caso contrario, non sono necessarie modifiche.

Interfaccia a riga di comando

AzuraCast dispone di un potente strumento di interfaccia a riga di comando (CLI) che consente di eseguire attività di manutenzione e risoluzione dei problemi "sotto il cofano".

Per utilizzare la CLI di AzuraCast, è necessario disporre dell'accesso al terminale di shell al server che esegue AzuraCast (alla macchina host, se si esegue Docker).

Alcuni di questi comandi sono destinati solo agli sviluppatori per eseguire il debug o per risolvere i problemi, usare cautela quando si eseguono questi comandi e farli solo quando viene detto, o sai cosa stai facendo.

Invocare il CLI

Utilizzo dello script di utilità Docker:

Se si utilizza il metodo di installazione Docker, si consiglia di scaricare il Docker Utility Script, che vi permetterà di utilizzare questa sintassi più breve:

Finestra terminale

```
./docker.sh cli [command]
```

Da un'installazione Docker:

Finestra terminale

```
docker-compose run --rm cli azuracast_cli [command]
```

Installazione Ansibile:

Finestra terminale

```
php /var/azuracast/www/bin/console [command]
```

Comandi CLI correlati all'account

Questa sezione coprirà tutti i comandi CLI correlati all'account e la risposta che la CLI fornirà.

Elenca tutti gli account

Finestra terminale

```
(cli_command) azuracast:account:list
```

AzuraCast User Accounts

=====

E-mail Address	Name	Roles	Created
name@domain.com		Super Administrator	YYYY-MM-DD HH:MM

Crea un URL di recupero di login univoco per l'account specificato

Finestra terminale

```
(cli_command) azuracast:account:login-token [example]
```

```
Generate Account Login Recovery URL
```

```
=====
```

```
The account recovery URL is:
```

```
https://domain/recover/hash:hash
```

```
Log in using this temporary URL and set a new password using the web interface.
```

Reimpostare la password dell'account specificato

Finestra terminale

```
(cli_command) azuracast:account:reset-password [example]
```

```
Reset Account Password
```

```
=====
```

```
The account password has been reset. The new temporary password is:
```

```
[example]
```

```
Log in using this temporary password and set a new password using the web interface.
```

Impostare l'account specificato come amministratore globale

Finestra terminale

```
(cli_command) azuracast:account:set-administrator [example]
```

```
Set Administrator
```

```
=====
```

```
The account associated with e-mail address "[example]" has been set as an administrator
```

Comandi CLI correlati API

Questa sezione coprirà i nostri comandi (s) relativi alle API, questo sarà correlato all'API del tuo AzuraCast.

Rigenerare la documentazione API di AzuraCast

Finestra terminale

```
(cli_command) azuracast:api:docs
```

```
API documentation updated!
```

Questo comando dovrebbe essere utilizzato quando si è su una macchina di sviluppo, produzione o test e si stanno apportando modifiche all'API, questo aggiornerà la tua API di conseguenza (dominio / api)

Comandi CLI interni

Questa sezione coprirà i nostri processi interni, come l'autenticazione Liquidsoap, SSL e SFTP.

Ottieni l'indirizzo IP esterno per questa istanza

Finestra terminale

```
(cli_command) azuracast:internal:ip
```

[example]

L'esecuzione di questo comando restituirà l'IP di questa macchina.

Maniglia chiamate API Liquidsoap

Finestra terminale

```
(cli_command) azuracast:internal:liquidsoap [example] [example]
```

Questo comando può eseguire varie chiamate API Liquidsoap, questo viene utilizzato principalmente per scopi di debug o sviluppo.

Ricarica i frontend di trasmissione quando cambia un certificato SSL

Finestra terminale

```
(cli_command) azuracast:internal:on-ssl-renewal
```

Tentativo di autenticazione SFTP

Finestra terminale

```
(cli_command) azuracast:internal:sftp-auth
```

Invia il feedback della canzone da AutoDJ di nuovo a AzuraCast

Finestra terminale

```
(cli_command) azuracast:internal:sftp-event
```

AzuraCast Impostazioni CLI Comdanti

Questa sezione coprirà le nostre impostazioni interne interne che vanno dall'URL di base al smtp. Questo può essere utilizzato se si è bloccati fuori dalle impostazioni o non è possibile accedere a AzuraCast direttamente dal browser.

Elencare tutte le impostazioni per AzuraCast

Finestra terminale

```
(cli_command) azuracast:settings:list
```

Questo comando restituirà tutte le impostazioni che è possibile modificare tramite l'interfaccia della riga di comando, che verrà visualizzata nel campo successivo.

Modificare le impostazioni per AzuraCast

Finestra terminale

```
(cli_command) azuracast:settings:set [example] [example]
```

Il `setting-key` parametro può essere raccolto dal `settings:list` Comando CLI, questo comando ti permetterà di modificare quasi tutte le parti delle impostazioni di AzuraCast, questo è utile se hai problemi con le impostazioni configurate male.

Eseguire Manualmente AzuraCast Setup

Finestra terminale

```
(cli_command) azuracast:setup [--update]
```

Esegue tutti gli aggiornamenti del database necessari per portare l'installazione di AzuraCast all'ultima versione. Questo viene normalmente eseguito automaticamente come parte dei processi di installazione e aggiornamento, ma può essere eseguito manualmente per la risoluzione dei problemi o lo sviluppo locale.

Questo comando scollega gli ascoltatori!

L'esecuzione di questo comando collegherà tutti gli ascoltatori attivi attuali nelle stazioni radio.

Cancella Tutte le cache

Finestra terminale

```
(cli_command) cache:clear
```

Cancella tutte le cache utilizzate internamente da AzuraCast. Questo può essere utilizzato come passaggio di risoluzione dei problemi se si riscontrano problemi con informazioni non aggiornate che appaiono sulle pagine del dashboard. Si noti che alcune pagine potrebbero richiedere un po' più di tempo per il caricamento dopo che tutte le cache sono cancellate.

Eseguire Attività di sincronizzazione

Finestra terminale

```
(cli_command) sync:run [nowplaying|short|medium|long]
```

Invoca manualmente le attività sincronizzate (“bron jobs”) che normalmente vengono eseguite automaticamente dietro le quinte per mantenere aggiornato AzuraCast.

- ora la riproduzione corrisponde al controllo di ogni 15 secondi di tutte le stazioni che attualmente riproducono metriche di canzoni e ascoltatori

- breve corrisponde all'attività di sincronizzazione di ogni minuto
- medium corrisponde all'attività di sincronizzazione di ogni 5 minuti
- lungo corrisponde all'attività di sincronizzazione di ogni ora

Comandi specifici della stazione

Rielaborare Manualmente Tutti I Supporti

Finestra terminale

```
(cli_command) azuracast:media:reprocess
```

Itera attraverso le directory multimediali di tutte le stazioni e ricarica manualmente le informazioni sui metadati memorizzate all'interno di AzuraCast con i dati più recenti sui file stessi. Questo è utile per la risoluzione dei problemi dei brani che sono bloccati nello stato di "Elaborazione" o se hai recentemente caricato più canzoni tramite SFTP.

Riavviare Tutte Le Stazioni Radio

Finestra terminale

```
(cli_command) azuracast:radio:restart
```

Spegne sia i frontend (Icecast, Shoutcast, ecc) che i backend (Liquidsoap) di tutte le stazioni radio, riscrive i loro file di configurazione, quindi li rilancia. Questo è identico al comando "Restart Broadcasting" all'interno dell'interfaccia web.

Questo comando scollega gli ascoltatori!

L'esecuzione di questo comando collegherà tutti gli ascoltatori attivi attuali nelle stazioni radio.

Queste attività possono anche essere invocate direttamente dall'interfaccia web tramite la homepage dell'amministrazione.

Altri Comandi

L'interfaccia di AzureCast CLI espone anche una serie di altri comandi avanzati. Questi comandi sono destinati agli sviluppatori da utilizzare quando si costruisce l'applicazione, e spesso non dovrebbero essere gestiti dai proprietari di stazioni su installazioni di produzione.

Per un elenco completo dei comandi di AzureCast CLI, visualizzali [Qui](#) .

Per ulteriori informazioni sugli strumenti aggiuntivi della riga di comando disponibili, vedere le rispettive pagine di documentazione qui sotto:

- [Riferimento della riga di comando della dottrina](#)
- [Migrazioni Dottrina](#)

Utilizzo di CSS personalizzati

Un cambiamento importante nelle versioni più recenti...

Nella versione 0.19.0, siamo passati da un vecchio sistema di stile a una versione più recente di Bootstrap e, nel processo, il prefisso sui comandi sottostanti è cambiato da `[data-theme]` a `[data-bs-theme]`. È necessario aggiornare qualsiasi CSS personalizzato che si sta utilizzando di conseguenza.

AzuraCast ti offre la possibilità di personalizzare il CSS predefinito tramite il **Custom Branding** pagina a `/admin/branding`.

Lì è possibile modificare lo stile delle pagine interne e pubbliche tramite il proprio CSS con gli editor per **Custom CSS for Public Pages** e **Custom CSS for Internal Pages**.

Questa pagina ti fornirà una semplice panoramica delle classi CSS e degli id utilizzati nell'HTML di AzuraCast che puoi quindi utilizzare nel tuo CSS per personalizzare tali elementi.

Se non hai alcuna esperienza precedente con CSS dai un'occhiata al [MDN Web Docs](#) a proposito di CSS.

Nota sulla rilevazione automatica del tema

AzuraCast ha ora una funzione integrata in cui rileverà automaticamente la combinazione di colori preferita dall'utente dal proprio browser tramite una query multimediale CSS. Si consiglia che se si intende personalizzare le pagine pubbliche, si disabilita questo visitando il **Custom Branding** pagina e impostazione di un tema specifico per le pagine pubbliche che non è "Preferisci sistema predefinito".

Le istruzioni riportate di seguito presumono che questa modifica sia già stata apportata.

CSS personalizzato per le pagine pubbliche

Le pagine pubbliche di ogni stazione hanno una classe applicata al `<body>` elemento che segue il formato `station-<station_short_name>`.

Quindi se hai una stazione chiamata `My Radio`, questa classe sarebbe `station-my_radio`. È possibile utilizzare questo per definire gli stili CSS che dovrebbero applicarsi solo a una singola stazione prefissando i selettori CSS come questo:

```
# Targeting either dark or light themed pages
[data-bs-theme] body.station-my_radio {
  ...
}
```

```
# Targeting only light themed pages
[data-bs-theme="light"] body.station-my_radio {
    ...
}

# Targeting only dark themed pages
[data-bs-theme="dark"] body.station-my_radio {
    ...
}
```

Modifica dell'immagine di sfondo

Il CSS per l'immagine di sfondo sulle pagine pubbliche è definito così:

```
# See above for theme-specific prefixes
[data-bs-theme] body.page-minimal {
    background: #edecec url(../img/hexbg.png);
    background-size: cover;
    background-attachment: fixed;
}
```

È possibile modificare l'immagine di sfondo utilizzando lo stesso selettore CSS e quindi semplicemente sovrascrivere il **background** attributo come questo:

```
[data-bs-theme] body.page-minimal {
    background: url(https://example.com/my_custom_background_image.png);
}
```

Styling il lettore radio

È possibile modificare il colore di sfondo del lettore radio come questo:

```
[data-bs-theme] body.page-minimal .public-page .card {
    background-color: #5e0f16;
}
```

Il selettore CSS per il titolo del lettore è il seguente:

```
[data-bs-theme] body.page-minimal .public-page .card .card-body .card-title {
    ...
}
```

I pulsanti nella parte inferiore del lettore possono essere stilizzati con il seguente selettore CSS:

```
[data-bs-theme] body.page-minimal .public-page .card .card-actions .btn {
    ...
}
```

Gli elementi del giocatore stesso si trovano nel `radio-player-widget` classe e sono separati più sotto `now-playing-detail` e `radio-controls`.

Come esempio su come personalizzare quegli elementi questa pagina fornisce un breve frammento per un elemento di ogni parte del widget del lettore. Per informazioni più

dettagliate si consiglia di guardare direttamente l'HTML della pagina tramite l'ispettore HTML del browser.

Questo esempio mostra come cambiare il colore del testo del titolo e dell'artista:

```
[data-bs-theme] body.page-minimal .public-page .card .now-playing-title {
  color: #fff;
}

[data-bs-theme] body.page-minimal .public-page .card .now-playing-artist {
  color: #fefefe;
}
```

Questo esempio mostra come modificare il colore del pulsante di riproduzione:

```
[data-bs-theme] body.page-minimal .public-page .card .radio-control-play-button
a {
  color: #000;
}
```

Styling i modali per Song History & Song Requests

Per modificare il colore di sfondo dei modali è possibile utilizzare il seguente selettore CSS:

```
[data-bs-theme] body.page-minimal .modal .modal-content {
  background-color: #5e0f16;
}
```

Per modificare il colore del testo del titolo utilizzare il seguente selettore CSS:

```
[data-bs-theme] body.page-minimal .modal .modal-content .modal-title {
  color: #fff;
}
```

Per modificare il colore di sfondo della paginazione utilizzare il seguente selettore CSS:

```
[data-bs-theme] body.page-minimal .modal .modal-content .pagination {
  background: #3d3d3d;
}
```

Il colore del testo dell'intestazione e delle righe della tabella delle richieste di canzoni può essere modificato in questo modo:

```
[data-bs-theme] body.page-minimal .modal .modal-content .table thead {
  color: #fff;
}

[data-bs-theme] body.page-minimal .modal .modal-content .table tbody {
  color: #fff;
}
```

CSS personalizzato per le pagine interne

Le pagine interne possono anche essere condizionalmente acconciate per il tema oscuro e chiaro come si fa per le pagine pubbliche:

```
[data-bs-theme="light"] body {  
    ...  
}
```

```
[data-bs-theme="dark"] body {  
    ...  
}
```

Cambiare il colore della navigazione superiore

Se si desidera modificare il colore della navigazione superiore è possibile utilizzare il seguente selettore CSS per utilizzare il proprio colore e rimuovere l'immagine di sfondo:

```
[data-bs-theme] header.navbar {  
    background-image: none;  
    background: #3d3d3d;  
}
```

Cambiare il colore dell'intestazione della carta

La maggior parte delle carte e alcuni altri elementi nelle pagine interne hanno un colore di sfondo blu. Per cambiare questo colore è possibile utilizzare il seguente selettore CSS:

```
[data-bs-theme] .bg-primary-dark {  
    background-color: #5e0f16 !important;  
}
```

Utilizzo di JavaScript personalizzato

JavaScript personalizzato per le pagine pubbliche

AzuraCast ti offre la possibilità di aggiungere JS personalizzati tramite il Custom Branding pagina a /admin/branding..

Lì puoi aggiungere i tuoi JavaScript alle pagine pubbliche con l'editor per Custom JS for Public Pages..

Puoi allegare a eventi del giocatore pubblico in questo modo:

```
$(document).on('now-playing', function(np_new) {  
    // custom code with np_new  
});
```

Esempio: Video Riproduzione per Pagina Pubblica

Potete usare Custom JS for Public Pages per aggiungere un elemento video alla pagina e stile tramite la Custom CSS for Public Pages usando il nostro esempio qui sotto.

CSS personalizzato per le pagine pubbliche

```
[data-theme] body.page-minimal .background-video {  
    width: 100vw;  
    height: 100vh;  
    object-fit: cover;  
    position: fixed;  
    left: 0;  
    right: 0;  
    top: 0;  
    bottom: 0;  
    z-index: -1;  
}
```

JS personalizzati per le pagine pubbliche

```
let videoBackgroundElement = document.createElement('video');  
videoBackgroundElement.autoplay = true;  
videoBackgroundElement.loop = true;  
videoBackgroundElement.muted = true;  
videoBackgroundElement.poster = 'Enter.URL.Here';  
videoBackgroundElement.className = 'background-video';
```

```
let videoBackgroundSource = document.createElement('source');  
videoBackgroundSource.src = 'Enter.URL.Here';  
videoBackgroundSource.type = 'video/mp4';
```

```
videoBackgroundElement.appendChild(videoBackgroundSource);
```

```
document.body.append(videoBackgroundElement);
```

Docker

Le installazioni di Docker sono dotate di quattro file per impostazione predefinita:

- `docker.sh`, lo script di utilità Docker;
- `.env`, che contiene variabili d'ambiente utilizzate da Docker Compose stesso;
- `azuracast.env`, che contiene variabili d'ambiente personalizzabili inviate a AzuraCast e servizi correlati; e
- `docker-compose.yml`, un file di grandi dimensioni che definisce tutti i servizi utilizzati da AzuraCast e come interagiscono.

Script di utilità Docker

Se stai utilizzando il metodo di installazione Docker per eseguire AzuraCast, abbiamo creato uno script di utilità utile per eseguire funzioni comuni senza dover digitare nomi di comandi lunghi.

Scarica lo script Utility

Se hai seguito di recente le [istruzioni di installazione](#) di [Docker](#), hai già installato Docker Utility Script. Il file è denominato `docker.sh`.

Se si dispone di un'installazione precedente, è possibile utilizzare Docker Utility Script eseguendo questi comandi all'interno della directory AzuraCast sul computer host:

Finestra terminale

```
curl -L https://raw.githubusercontent.com/AzuraCast/AzuraCast/main/docker.sh > docker.sh
chmod a+x docker.sh
```

Eeguire strumenti di riga di comando

Finestra terminale

```
./docker.sh cli [command_name]
```

Esegue qualsiasi comando esposto dagli strumenti di interfaccia della riga di comando.

Comandi disponibili

Installare AzuraCast

Finestra terminale

```
./docker.sh install
```

Estrae l'ultima versione di tutte le immagini di Docker e imposta il database AzuraCast. Una volta completata, la tua istanza AzuraCast dovrebbe essere attiva e funzionante.

Aggiornamento AzuraCast

Finestra terminale

```
./docker.sh update-self  
./docker.sh update
```

Nota: È sempre necessario eseguire l'aggiornamento-self prima per aggiornare lo script updater.

Tira automaticamente giù tutte le immagini Docker aggiornate e applica qualsiasi modifica del database e della configurazione dall'ultimo aggiornamento di AzuraCast.

Disinstalla AzuraCast

Finestra terminale

```
./docker.sh uninstall
```

Disattiva e elimina in modo permanente sia i contenitori AzuraCast Docker che i volumi permanenti che memorizzano il database e i supporti delle stazioni di AzuraCast.

Questo comando si traduce in perdita di dati!

Questo comando rimuoverà completamente qualsiasi supporto di stazione, statistiche e metriche e l'intero database associato all'istanza di AzuraCast.

Backup di file e impostazioni

Finestra terminale

```
./docker.sh backup [/path/to/backup.tar.gz]
```

Crea una copia di backup .tar.gz dei media, delle statistiche e delle metriche di ogni stazione, insieme a una copia del database completo di AzuraCast. È possibile ripristinare successivamente da questo stesso file in caso di perdita di dati o corruzione.

Ripristinare file e impostazioni

Finestra terminale

```
./docker.sh restore /path/to/backup.tar.gz
```

Estrae un file .tar.gz precedentemente creato dal comando di backup di questo stesso script, copiando i supporti, le statistiche e le metriche per ogni stazione in AzuraCast e importando la versione del database contenuta nel backup.

Questo comando sostituisce la tua attuale installazione!

Il ripristino da un backup rimuoverà qualsiasi database o supporto AzuraCast esistente che esiste all'interno dei volumi Docker.

Personalizzazione delle installazioni di Docker

Per gli utenti esperti che desiderano personalizzare o espandere la loro configurazione Docker, è necessario seguire queste best practice:

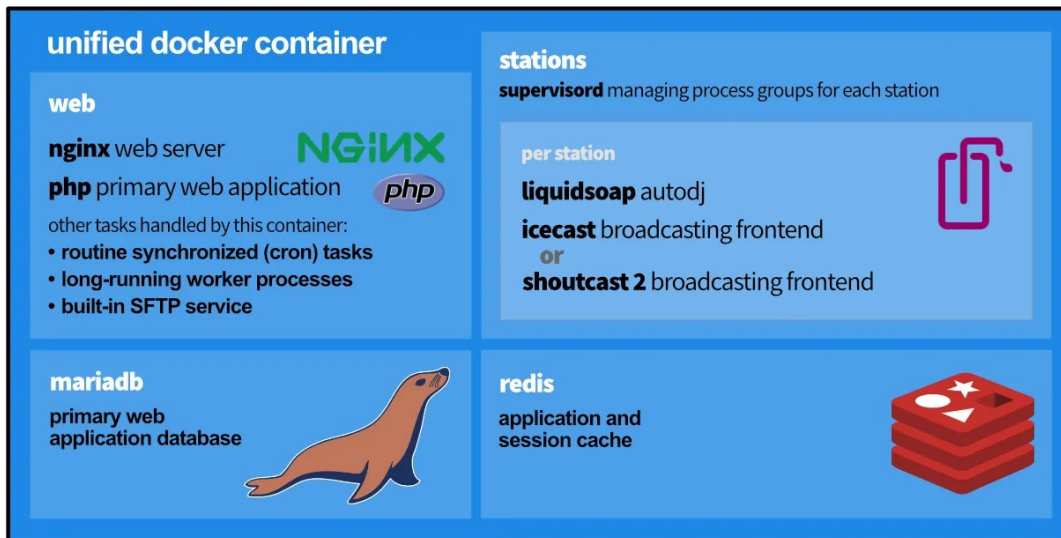
- Non modificare o sostituire il `docker . shscript` di utilità.
- Quando si aggiorna (utilizzando il `docker . shscript` di utilità), si consiglia di eseguire `./docker . sh update-self` prima di correre `./docker . sh update`, per garantire che lo script Docker Utility stesso sia aggiornato prima di aggiornare l'installazione di Docker.
- Variabili ambientali impostate in `. env` sono utilizzati solo da Docker Compose stessa, e non vengono passati direttamente nei contenitori AzuraCast. È necessario modificare questo file solo per modificare le mappature delle porte HTTP e HTTPS utilizzate da Nginx (vedere la sezione "Usa porte non standard" sopra).
- Il `azuracast . envfile` è specifico per il vostro ambiente e può essere personalizzato come vi piace. Non sarà sostituito durante nessun aggiornamento. Una volta creato il database, tuttavia, la modifica della password elencata in questo file causerà il fallimento del sistema. Se vuoi cancellare in modo distruttivo il tuo database esistente e altri file e configurarne uno nuovo con la password aggiornata, aggiungi il `-vflag` alla fine di ``docker-comporre` verso il basso per rimuovere tutti i volumi esistenti, incluso il database.
- Se possibile, non si deve modificare direttamente `docker -compose . yml`, poiché alcuni aggiornamenti possono modificare il modo in cui viene definito per risolvere i bug o aggiungere nuove funzionalità. Quando si aggiorna, ti verrà sempre chiesto se vuoi aggiornare questo file; se non l'hai modificato, dovresti sempre farlo.
- Invece di modificare `docker -compose . yml`, è possibile creare un file denominato `docker -compose . override . yml` con le tue personalizzazioni. La struttura di questo file è la stessa della principale `docker -compose . yml` file, e viene automaticamente analizzato da Docker Compose per sovrascrivere qualsiasi definizione nel file principale. Gli aggiornamenti non sostituiranno questo file.

Modificare Docker

L'installazione di AzureCast Docker è costruita per soddisfare le esigenze della stragrande maggioranza delle installazioni fuori dagli schemi, ma potrebbero esserci momenti in cui è necessario personalizzare il modo in cui Docker serve la tua installazione, pur mantenendo i vantaggi della compatibilità e del facile aggiornamento che Docker offre.

L'infrastruttura complessiva

Nelle versioni recenti, abbiamo unito la nostra infrastruttura Docker in un unico contenitore che contiene l'intera applicazione, i suoi database e cache e i controlli di trasmissione.



Docker Compose

Per rendere la nostra installazione Docker semplice da mantenere, ci affidiamo a un software secondario chiamato [Docker Compose](#). . Piuttosto che utilizzare lunghi comandi a riga di comando, Docker Compose consente di scrivere semplici file YAML per configurare il modo in cui i nostri contenitori Docker si avviano, che portano in avanti all'host, volumi di dischi montati, ecc.

Se segui le nostre istruzioni di installazione (o usi una delle nostre immagini precostruite), l'installazione di Docker si trova sul tuo computer host all'indirizzo `/var/azuracast.` . All'interno di questa cartella, vedrai quattro file per impostazione predefinita:

- `docker-compose.yml`, il file di configurazione primario Docker Compose. L'aggiornamento di AzuraCast richiede automaticamente di mantenere aggiornato questo file, che è fortemente raccomandato;
- `docker.sh`, il Docker Utility Script che fornisce facili alias per le attività Docker comuni, come l'aggiornamento, l'esecuzione dei comandi all'interno dei contenitori e altro ancora;
- `azuracast.env`, un elenco di variabili di ambiente (configurazione) che vengono inviate all'applicazione AzuraCast stessa, in esecuzione all'interno del contenitore web; e
- `.env`, un file ambiente separato che influenza il modo in cui Docker Compose stesso è configurato, e viene quindi utilizzato per modifiche di configurazione di livello superiore come le mappature delle porte.

Composta Docker prevalente

Piuttosto che modificare `docker-compose.yml` direttamente, si consiglia vivamente di creare invece un nuovo file chiamato `docker-compose.override.yml` nella stessa cartella. Docker Compose cercherà automaticamente e includerà questo file se esiste, e applica la sua configurazione sopra le istruzioni del file di base.

Ci sono alcune considerazioni quando si crea la propria `docker-compose.override.yml`:

- Il file deve essere valido YAML usando lo standard [Docker formato composito](#).
- Quando si modificano porte o altri elenchi, è possibile aggiungere solo nuovi elementi, non rimuovere quelli esistenti.

Una volta modificata la configurazione Docker Compose, è necessario applicare queste modifiche eseguendo:

Finestra terminale

```
docker-compose down
docker-compose up -d
```

Si noti che questo spegnerà temporaneamente l'installazione di AzuraCast e scollegherà brevemente i tuoi ascoltatori.

Utilizzo di porte non standard

Suggerimento

Nelle versioni più recenti, non è più necessario farlo manualmente; è invece possibile utilizzare l'installatore rispondendo a "Sì" quando viene richiesto di personalizzare le impostazioni.

Potresti voler servire l'applicazione web AzuraCast stessa su una porta diversa o ospitare la tua stazione radio su una porta che non rientra nell'intervallo predefinito che AzuraCast serve (8000-8999).

È possibile utilizzare uno strumento helper nello script di utilità Docker per modificare facilmente le porte utilizzate da AzuraCast:

Finestra terminale

```
cd /var/azuracast
./docker.sh update-self
./docker.sh change-ports
```

Per sovrascrivere funzionalità più complesse nell'installazione di Docker, vedere la sezione "Docker personalizzato" qui sotto.

Ampliare la gamma di porte della stazione

Suggerimento

Le installazioni che eseguono la versione 0.16.0 o superiore possono ora utilizzare il nostro script di installazione per personalizzare le porte tramite il terminale che scriverà automaticamente le modifiche nel `docker-compose.yml` File per te.

Per motivi di prestazioni, per impostazione predefinita, le installazioni Docker aprono solo le porte radio dalla porta 8000 all'8500. Questo permette di utilizzare 50 stazioni uniche.

A seconda del tuo hardware, potrebbe essere possibile eseguire più di 50 stazioni su un'istanza di AzuraCast, ma se desideri accedere direttamente alle porte radio aggiuntive, puoi seguire questo semplice processo.

Nella stessa cartella in cui si trova l'installazione di Docker (se si utilizzano le istruzioni consigliate, questo è `/var/azuracast`), creare un nuovo file denominato `docker-compose.override.yml`.

In questo file, incollare i seguenti contenuti:

```
services:
  web:
    ports:
      - "8500-8999:8500-8999"
```

È possibile modificare l'intervallo di porte in questo file per soddisfare le vostre esigenze, come l'espansione alla porta 8999 invece di 8500.

Quando si crea una nuova stazione, AzuraCast tenterà di assegnarle automaticamente una porta disponibile dalla gamma di porte disponibile. Se cambi questo intervallo di porta, dovresti informare AzuraCast aggiungendo o aggiornando i seguenti valori nel tuo `azuracast.envfile`:

```
AUTO_ASSIGN_PORT_MIN=8500
AUTO_ASSIGN_PORT_MAX=8999
```

Dovrai riavviare i tuoi contenitori Docker utilizzando `docker-compose down`, allora `docker-compose up -d` per applicare eventuali modifiche apportate a questi file.

Montaggio di una directory in una stazione

Potresti voler aggiungere musica a una stazione da una directory sulla tua macchina host senza copiare i dati in AzuraCast. È possibile montare la directory nelle stazioni e nel contenitore web per renderle disponibili a AzuraCast creando un `docker-compose.override.yml`.

Nella stessa cartella in cui si trova l'installazione di Docker (se si utilizzano le istruzioni consigliate, questo è `/var/azuracast`), creare un nuovo file denominato `docker-compose.override.yml`.

In questo file, incollare i seguenti contenuti:

```
services:
  web:
    volumes:
```

`/path/on/host/computer:/var/azuracast/stations/<STATION_NAME>/media/`

Sostituire il `<STATION_NAME>` con il nome della directory della stazione trovata nella sezione "Amministrazione" delle impostazioni del profilo della stazione e modificare la `/path/on/host/computer` con il percorso verso la directory che si desidera montare.

Dovrai riavviare i tuoi contenitori Docker utilizzando `docker-compose down`, allora `docker-compose up -d` per applicare eventuali modifiche apportate a questo file.

Memorizzazione dei dati della stazione sulla macchina host

È possibile memorizzare tutti i dati della stazione in una directory sulla vostra macchina host. Questo può essere utile se si desidera avere AzuraCast in esecuzione su un piccolo SSD e memorizzare i dati della stazione su un grande HDD.

Nella stessa cartella in cui si trova l'installazione di Docker (se si utilizzano le istruzioni consigliate, questo è `/var/azuracast`), creare un nuovo file denominato `docker-compose.override.yml`.

In questo file, incollare i seguenti contenuti:

```
services:
  web:
    volumes:
      - /path/on/host/computer:/var/azuracast/stations
```

Modificare il `/path/on/host/computer` con il percorso verso la directory che si desidera montare.

Dovrai riavviare i tuoi contenitori Docker utilizzando `docker-compose down`, allora `docker-compose up -d` per applicare eventuali modifiche apportate a questo file.

Traccia predefinita personalizzata

Suggerimento

Nelle versioni più recenti di AzuraCast, non è più necessario caricare una traccia predefinita personalizzata in questo modo; è possibile caricare una per-stazione unica "Custom Fallback Track" utilizzando l'interfaccia web.

Quando nulla viene giocato sulla tua stazione sentirai il file predefinito `error.mp3` di AzuraCast in riproduzione. È possibile sostituire questo file montando il proprio file `.mp3` tramite un `docker-compose.override.yml`.

Nella stessa cartella in cui si trova l'installazione di Docker (se si utilizzano le istruzioni consigliate, questo è `/var/azuracast`), creare un nuovo file denominato `docker-compose.override.yml`.

È possibile posizionare il file `.mp3` ovunque sulla macchina host. Devi solo specificare il percorso verso di esso sostituendo questa parte: `/path/to/your/file.mp3`

Consigliamo di inserire quel file all'interno del `/var/azuracastdirectory` però in modo da avere tutto nello stesso posto. Assicurarsi che il formato del file specificato corrisponda esattamente al formato di streaming.

In questo file, incollare i seguenti contenuti:

```
services:
```

```
  web:
```

```
    volumes:
```

```
  /path/to/your/file.mp3:/usr/local/share/icecast/web/error.mp3
```

Dovrai riavviare i tuoi contenitori Docker utilizzando `docker-compose down`, allora `docker-compose up -d` per applicare eventuali modifiche apportate a questo file.

File introduttivo del flusso

Suggerimento

Nelle versioni più recenti di AzuraCast, è ora possibile caricare direttamente una traccia introduttiva nella pagina di modifica di Mount Point.

È possibile aggiungere un file musicale da riprodurre quando qualcuno si connette inizialmente al flusso. Ricorda quando crei file introduttivi che devono corrispondere allo stesso formato, bitrate e frequenza di campionamento del punto di montaggio per funzionare correttamente.

In primo luogo, dite al filesystem Docker dove trovare il vostro file introduttivo. All'interno della directory AzuraCast sul tuo host (per impostazione predefinita, `/var/azuracast`), creare un file denominato `docker-compose.override.yml` con i seguenti contenuti:

```
services:
```

```
  stations:
```

```
    volumes:
```

```
  /path/to/your/file.mp3:/usr/local/share/icecast/web/intro.mp3
```

Ora riavviare AzuraCast via `docker-compose down` && `docker-compose up -d`.

Torna all'interfaccia web di AzureCast, visita la pagina "Punti montati" per la tua stazione, modifica il punto di montaggio per cui desideri aggiungere un'introduzione e, all'interno del campo "Advanced: Custom Frontend Configuration", inserisci questo, in base all'ultima porzione del file montato dall'esempio precedente:

```
<intro>/intro.mp3</intro>
```

Per ulteriori informazioni, vedere il [Documentazione Icecast](#).

Favicon personalizzato

Suggerimento

Nelle versioni più recenti di AzuraCast, è possibile gestirlo tramite l'interfaccia utente web.

Per ignorare il favicon e le altre icone del browser utilizzate da AzuraCast, è necessario seguire questi passaggi:

1. Carica la tua icona su favicon-generator.org. Assicurati di lasciare le “icone generate per Web, Android, Microsoft e iOS (iPhone e iPad) Apps” controllate.
2. Una volta scaricato il set di icone, sarà in un formato di file .zip; estrarre quel file e copiarlo sul server host in cui AzuraCast è ospitato.
3. All'interno della directory AzuraCast sul tuo host (per impostazione predefinita, /var/azuracast), crea un file denominato docker-compose.override.yml con i seguenti contenuti:

```
services:
```

```
  web:
```

```
    volumes:
```

```
      - /path/to/your/favicon/folder:/var/azuracast/www/static/icons/production
```

4. Riavvia AzuraCast via `docker -compose down && docker -compose up -d..`

Aumentare la memoria consentita da PHP

Quando si verifica un errore come quello seguente sarà necessario aumentare la quantità di memoria che PHP è autorizzato a utilizzare:

```
AzuraCast.CRITICAL: Fatal error: Allowed memory size of 268435456 bytes exhausted (tried to allocate 212992 bytes) ...
```

Si può facilmente fare questo modificando il `azuracast.env` situato in `/var/azuracast/..`

Qui potete trovare la seguente voce:

```
# PHP's maximum memory limit.  
# PHP_MEMORY_LIMIT=128M
```

Cambia questo per assomigliare a questo:

```
# PHP's maximum memory limit.  
PHP_MEMORY_LIMIT=512M
```

È necessario aumentare il valore fino a quando l'errore scompare.

Il M dopo che il valore sta per Megabytes. .

Fare riferimento a [La documentazione di PHP](#) per ulteriori informazioni sui valori consentiti per questa impostazione.

Dopo aver cambiato questa impostazione è necessario riavviare AzuraCast via `docker -compose down && docker -compose up -d..`

Aggiungere configurazioni NGINX personalizzate

Se hai bisogno di aggiungere alcune configurazioni personalizzate a NGINX puoi farlo dal tuo `custom.conf` file nel web contenitore tramite il `docker-compose.override.yml` così:

```
services:
```

```
  web:
```

```
    volumes:
```

```
      - ./custom.conf:/etc/nginx/azuracast.conf.d/custom.conf
```

Tutti i file di configurazione inseriti nel `/etc/nginx/azuracast.conf.d/` sono automaticamente inclusi nella configurazione principale di AzuraCast NGINX.

Installazione multi-sito

AzuraCast può esistere su un server con altri siti web ospitati in diversi URL. Dai un'occhiata alla nostra [guida all'installazione di Docker multi-sito](#) per ulteriori informazioni.

Installazione multi-sito Docker

Se si utilizza il metodo di installazione Docker consigliato per AzuraCast, è possibile configurare l'installazione per sedersi dietro un proxy nginx in grado di gestire anche altri siti Web sullo stesso host. Questo ti consente di ospitare, ad esempio, il sito web della tua stazione sullo stesso server dell'installazione di AzuraCast.

Seguire i passaggi riportati di seguito per configurare l'installazione per il supporto multisito.

Responsabile proxy di Nginx

La maggior parte degli utenti preferisce utilizzare un proxy manager che include una GUI per una gestione più facile del proxy. Ci sono molti strumenti diversi per questo, uno dei quali è il [Responsabile proxy di Nginx](#), che utilizza NGINX e include una GUI facile da usare e automatica Let's Encrypt SSL che dovrebbe essere sufficiente per la maggior parte degli utenti.

Inserisci la directory di base di AzuraCast

Connettiti al tuo server tramite SSH e passa alla directory di base in cui si trova l'installazione di AzuraCast. Se ha seguito le istruzioni di installazione consigliate (o ha usato un'immagine precostruita) è possibile eseguire:

Finestra terminale

```
cd /var/azuracast
```

Disattivare I Servizi Esistenti

Mentre stai apportando queste modifiche di configurazione, i tuoi contenitori Docker dovrebbero essere disattivati. Disattivare i contenitori Docker esistenti eseguendo:

Finestra terminale

```
docker -compose down
```

Questo scollegherà i tuoi ascoltatori e impedirà alla tua stazione di trasmettere. Non perderai alcun dato o statistica durante l'aggiornamento.

Modifica AzuraCast Web Porte di servizio

Il primo passo è prevenire il proprio AzuraCast webcontainer dal tentativo di ascoltare sulle porte 80 e 443, come proxy ora farà quel lavoro.

C'è un file nella directory di installazione di AzuraCast denominata `.env` che ha due valori impostati per impostazione predefinita:

```
AZURACAST_HTTP_PORT=80  
AZURACAST_HTTPS_PORT=443
```

Nota il nome del file!

Questo file è in realtà chiamato `.env`, non `azuracast.env`, che è un file separato. La maggior parte degli host Linux nasconderà il `.envfile` per impostazione predefinita, ma si può ancora vedere in esecuzione `ls -lah` oppure modificarlo con un comando come `vim .env..`

Modifica il `.envfile` nel tuo editor di scelta per cambiare quei valori in una porta rivolta al pubblico inutilizzata. Una buona opzione sarebbe gli esempi seguenti:

```
AZURACAST_HTTP_PORT=10080
AZURACAST_HTTPS_PORT=10443
```

Salva le tue modifiche e torna alla shell.

Avviare i servizi Docker

Riattivare i contenitori Docker AzuraCast per riprendere la trasmissione con la nuova configurazione:

Finestra terminale

```
docker-compose up -d
```

Configurazione Nginx Proxy Manager

Creare una nuova directory per Nginx Proxy Manager e immetterla tramite i seguenti comandi:

Finestra terminale

```
mkdir /var/proxy_manager
cd /var/proxy_manager
```

Quindi creare un file con il nome `docker-compose.yml` in questa directory (ad esempio con `nano`) così:

Finestra terminale

```
nano docker-compose.yml
```

Ora segui il [Configurazione rapida](#) per il Proxy Manager di Nginx che inizia con il passaggio 2..

Crea Proxy Host per AzuraCast

Quando si è impostato con successo il Nginx Proxy Manager e si ha accesso alla GUI è possibile iniziare a creare un proxy per AzuraCast tramite il **Proxy Host** pagina.

Clicca sulla **Create Proxy Host** pulsante, immettere un dominio che punta al server nel **Domain Name** campo.

Nel campo “Schema”, selezionare https. . In “Forward Hostname/IP”, inserisci l’IP rivolto al pubblico del tuo server (non il nome di dominio). In “Porta in avanti”, inserisci la stessa porta che hai specificato sopra per AzuraCast (nel nostro esempio, 10443).

Abilita il Websockets Supportopzione.

Passa al SSLscheda e selezionare Request a new SSL Certificatenel SSL Certificatediscesa.

Abilita il Force SSLcosì come la HTTP/2 Supportopzioni, inserire il proprio indirizzo e-mail, accettare i Termini di servizio Let’s Encrypt e fare clic sul Savepulsante.

Ruoli e autorizzazioni

Autorizzazioni richieste

Per gestire ruoli e autorizzazioni, gli utenti devono essere in un ruolo che ha il permesso “Amministra autorizzazioni” a livello globale.

AzuraCast dispone di un sistema di Access Control List (ACL) altamente granulare che consente di controllare le autorizzazioni specifiche di ciascun gruppo di utenti.

In sintesi, si può pensare all'elenco di controllo degli accessi come funzionante in questo modo:

- Uno o più Utenti possono essere in un ruolo.
- Ogni Ruolo ha uno o più Permessi, che dà a tutti in quel Ruolo quel livello di accesso. Le autorizzazioni possono essere specifiche per una singola stazione o a livello globale per l'intera installazione.

La pagina Ruoli e autorizzazioni ha protezioni in atto per impedire agli utenti di negare accidentalmente le autorizzazioni alla pagina Autorizzazioni stessa.

Autorizzazioni speciali

Ci sono autorizzazioni speciali che garantiranno l'accesso ad altre sezioni del sistema, anche quando non concedi esplicitamente quelle autorizzazioni di livello inferiore a ciascun utente:

- Il permesso globale All Permissions crea un super-utente con accesso a tutta l'installazione di AzuraCast e il permesso specifico per la stazione All Permissions crea un super-utente con accesso completo a quella stazione specifica.
- Il permesso globale di Manage Stations consente inoltre all'utente di amministrare tutte le stazioni esistenti nell'installazione di AzuraCast.

Requisiti di autorizzazione specifici per funzionalità

AzuraCast ha robusti controlli di autorizzazione che impediscono l'accesso non autorizzato alle funzionalità critiche della stazione:

- Skip Current Track : richiede all'utente di avere il manage station broadcasting permesso per una stazione.
- Sonda ora / Aggiungi alla coda : richiede all'utente di avere il manage station media permesso per una stazione.
- Eliminazione dei media : mentre qualsiasi utente con il manage station media permesso può visitare l'interfaccia Media Manager della stazione, solo gli utenti con la delete station media permesso può rimuovere i file esistenti.

SSL & Let's Encrypt

AzureCast include ora il supporto integrato per la creazione e la gestione dei certificati SSL (HTTPS) tramite LetsEncrypt dal pannello Impostazioni di sistema.

LetsEncrypt è un modo semplice e gratuito per consentire connessioni sicure e protette alla tua installazione di AzuraCast. Con un certificato SSL valido, è possibile:

- Proteggi il tuo collegamento con AzureCast durante la gestione delle stazioni,
- Applicare la sicurezza per tutti gli amministratori di AzuraCast tramite HTTP Strict Transport Security (HSTS) e
- Fornire un endpoint di ascolto sicuro agli ascoltatori, evitando avvisi "Contenuti contrastanti" quando il segnale radio viene riprodotto da una pagina Web sicura.

Considerazioni Importanti

Prima di configurare LetsEncrypt, è necessario assicurarsi che siano soddisfatte le seguenti condizioni:

- AzuraCast deve essere sul proprio dominio o sottodominio. Non è possibile configurare LetsEncrypt utilizzando solo un indirizzo IP; è necessario disporre di un dominio (cioè example.com) o di un sottodominio (radio.example.com) impostato per puntare all'installazione di AzuraCast.
- Il server web di AzureCast deve essere servito sulle porte predefinite, 80 per HTTP e 443 per HTTPS. Per impostazione predefinita, AzureCast è già configurato in questo modo, ma se hai modificato le porte per servire il sito su una porta secondaria, devi passare le porte alle impostazioni predefinite quando si configura LetsEncrypt e quando si eseguono i rinnovi.

Abilitare LetsEncrypt

Per abilitare LetsEncrypt, attenersi alla seguente procedura:

- Accedi all'installazione di AzuraCast
- Fai clic sul menu a discesa in alto a destra, quindi su "Amministrazione di sistema"
- Fare clic su "Impostazioni di sistema"
- Selezionare la scheda "Servizi"
- Completare i campi della sezione LetsEncrypt
- Fare clic su "Salva modifiche" in basso

Il certificato HTTPS verrà generato automaticamente nei prossimi minuti, ma è possibile farlo manualmente facendo clic sul pulsante "Crea/Renew Certificate" sotto i campi LetsEncrypt.

Suggerimento

Se gli utenti hanno ancora problemi con l'audio non in riproduzione, assicurati di abilitare l'opzione "Usa proxy Web" nelle impostazioni di sistema.

Rinnovare un certificato Let's Encrypt

Il servizio web rinnoverà automaticamente i certificati LetsEncrypt. Se si fornisce un'e-mail nel processo di configurazione iniziale, tale e-mail verrà utilizzata per inviarti promemoria della prossima scadenza nel caso in cui il rinnovo automatico non riesca.

Utilizzare sempre HTTPS (HTTP Strict Transport Security)

Dal pannello "Impostazioni di sistema" all'interno di AzuraCast, è possibile abilitare l'impostazione "Utilizza sempre HTTPS" nella scheda "Sicurezza e privacy".

Una volta abilitata questa impostazione, non solo tutti gli utenti saranno reindirizzati alla versione sicura di AzuraCast durante la visita, ma la sicurezza del trasporto HTTP Strict (HSTS) sarà abilitata, il che richiede un certificato SSL valido per funzionare. Questo migliora significativamente la sicurezza della tua connessione con AzuraCast e dovrebbe essere abilitato quando possibile.

Disabilitare HSTS tramite CLI

Se si abilita la sicurezza del trasporto rigido HTTP e il dominio non è più disponibile tramite HTTPS, normalmente non sarà possibile accedere all'installazione tramite il browser Web.

È possibile, tuttavia, disattivare questa impostazione tramite il terminale SSH, utilizzando il codice sottostante:

Finestra terminale

```
cd /var/azuracast
bash docker.sh cli azuracast:settings:set always_use_ssl 0
```

Questo disabiliterà l'impostazione e ti consentirà di riconnetterti all'installazione.

Utilizzo di un certificato personalizzato

Se si dispone di un certificato SSL personalizzato sul tuo host, è necessario creare un `docker-compose.override.yml` file nel tuo `/var/azuracast` directory sul server host con i contenuti riportati di seguito, modificati per riflettere il tuo nome di dominio e il percorso verso il tuo certificato SSL e la chiave:

```
services:
  web:
    volumes:
      - /path/on/host/to/ssl.crt:/var/azuracast/acme/ssl.crt:ro
      - /path/on/host/to/ssl.key:/var/azuracast/acme/ssl.key:ro
```

Suggerimento

Si prega di notare che Icecast si aspetta una chiave privata RSA e un file di certificato con la catena di certificati completa. Per i certificati personalizzati nel `.pem` formato generato da qualcosa come `Certbot` dovrai convertirli così:

```
openssl rsa -in privkey.pem -out example.com.key
```

```
openssl crl2pkcs7 -nocrl -certfile fullchain.pem | openssl pkcs7  
-print_certs -out example.com.crt
```

Infine è necessario riavviare AzuraCast per poter applicare le modifiche:

Finestra terminale

```
docker-compose down  
docker-compose up -d
```

Luoghi di stoccaggio

Per ogni stazione ci sono **Storage Locations** definito in AzuraCast per **Station Media**, **Station Recording** e **Backups**..

A **Storage Location** può essere uno dei seguenti tipi:

- **Filesystem locale**
- **S3-Compatibile**
- **Dropbox**

Posizione di stoccaggio compatibile con S3

Suggerimento

Le prestazioni di elaborazione dei media variano a seconda del provider che stai utilizzando (backblaze, Digital Ocean, Linode, S3, ecc) quindi non farti prendere dal panico se ci vuole del tempo.

Se l'elaborazione ti sembra lenta, misura le velocità di download del tuo bucket S3 per determinare se si tratta di un problema con le risorse del provider S3 prima di contattarci attraverso i nostri metodi di supporto.

Questa guida è rivolta a installazioni più piccole che hanno piccole dimensioni del disco di storage. Le nostre guide ti guideranno attraverso la raccolta delle chiavi di accesso / segrete. Avrai bisogno di utilizzare il servizio compatibile con S3: **Access Key ID**, **Secret Key**, **Endpoint**, **Region**, **Bucket Name** e **API version** (La versione API dovrebbe essere sempre latest).

Per configurare una stazione per utilizzare un servizio compatibile con S3, è necessario andare a yourdomain.com/admin/storage_locations, clicca su **edite** poi spuntare il **Remote: S3 Compatible** invece di **Local Filesystem**..

Sarà così:

Add Storage Location ✕

Storage Adapter

Local Filesystem

Remote: S3 Compatible

Path/Suffix

/bucket_media

For local filesystems, this is the base path of the directory. For remote filesystems, this is the folder prefix.

Storage Quota

Set a maximum disk space that this storage location can use. Specify the size with unit, i.e. "8 GB". Units are measured in 1024 bytes. Leave blank to default to the available space on the disk.

Remote: S3 Compatible

Access Key ID

Secret Key

Endpoint

Bucket Name

Region

API Version

latest

CLOSE
SAVE CHANGES

Servizio di archiviazione semplice di Amazon (S3)

[Servizio di archiviazione semplice Amazon](#), altrimenti noto come (S3) è un servizio di archiviazione di oggetti. [Prezzi di S3](#) è più complesso degli altri, invece di offrire una tassa di 'base', addebitata per quello che si usa. Puoi usare la nostra piccola guida elencata di seguito o utilizzare la documentazione ufficiale di Amazon. Puoi usare la nostra piccola guida per creare un bucket e raccogliere le chiavi o l'uso [La documentazione di Amazon](#)

- Con il sistema S3 di Amazon, dovrai accedere al [Console S3](#) e clicca **Create Bucket**
- Sarai in grado di scegliere il nome del secchio (deve essere unico in tutti gli S3, essere 3 e 63 caratteri di lunghezza).
- Scegli una regione dopo aver fatto il nome di un secchio.
- È possibile trovare la chiave di accesso e sicurezza all'interno del **My Security Credentials**

Backblaze

[Backblaze B2 Cloud Storage](#) è un servizio compatibile con S3. Ha un piano di prezzo simile al servizio S3 di Amazon, ma ha un [calcolatrice dei prezzi](#). Puoi usare la nostra piccola guida o l'uso [L'ampia documentazione di Backblaze](#) su B2 Cloud Storage.

- Con Backblaze dovrai accedere e andare alla tua pagina **Buckets**.

- Da lì è possibile raccogliere l'endpoint S3 (Nota importante: i secchi creati prima del 4 maggio 2020 non sono compatibili con S3, se un bucket non è compatibile, creane uno nuovo)
- È possibile creare l'ID della chiave di applicazione e la chiave di applicazione nella pagina Chiavi dell'app. (Nota importante: chiave di applicazione ID = chiave di accesso | Chiave di applicazione = chiave segreta)

Spazi oceanici digitali

[Spazi oceanici digitali](#) è un servizio di archiviazione di oggetti compatibile con S3. Il [piano di base](#) costa \$ 5 per 250GB con 1 TB di trasferimento in uscita. Puoi usare la nostra guida elencata di seguito o utilizzare [Documentazione di Digital Ocean sugli spazi](#)

- Per creare il servizio di storage Spaces dovrai creare un account tramite Digital Ocean, se ne hai uno, puoi saltare questo passaggio.
- Si può creare un Spaces dal menu `Crea e ti porterà alla pagina. Dovrai scegliere una regione del datacenter e abilitare o disabilitare la CDN (opzionale).
- Infine puoi scegliere un nome unico per lo Spazio, deve essere unico in quanto servirà come il tuo Endpoint. .
- Dovrai creare una chiave di accesso/segreto seguendo [questa guida](#) da Oceano Digitale.

Archiviazione degli oggetti di linodo (S3)

[Archiviazione di oggetti di linode](#) è un servizio di stoccaggio offerto da Linode e la [piano di base](#) parte da \$ 5 per 250 GB con 1 TB di trasferimento in uscita. Puoi usare la nostra piccola guida elencata di seguito o utilizzare [Documentazione di Linode su Object Storage](#)

- Pugno di tutto è necessario afferrare l'accesso e la chiave segreta dal [Cloud Manager](#), e clicca su **Object Storage**
- Se questo è il primo bucket che hai creato con Linode, dovrai confermare l'Oggett Storage. (Salta questa parte se hai un secchio)
- Dopo di che sarai in grado di generare una chiave di accesso e segreto.

Posizione Dropbox Storage

Per creare una posizione Dropbox Storage in AzureCast dovrai prima creare un token di accesso OAuth 2 per il tuo account Dropbox.

Vai alla [Dropbox App Console](#) e clicca sul **Create app** pulsante.

Seleziona le seguenti opzioni:

- 1. Choose an API
 - Scoped access
- 2. Choose the type of access you need
 - Se stai usando il tuo Dropbox, altre cose a parte l'uso di AzuraCast App folder per creare automaticamente una cartella separata nel tuo account Dropbox per questa app
 - Altrimenti selezionare Full Dropbox
- 3. Name your app

- **Dai un nome alla tua app, si consiglia di scegliere un nome specifico per la tua radio**

Quindi clicca su `Create` appper finire questo passaggio.

Nella pagina ora generata per l'app passare al `Permissions` scheda nella parte superiore della pagina e selezionare le seguenti autorizzazioni:

- `files.metadata.write`
- `files.metadata.read`
- `files.content.write`
- `files.content.read`

Quindi clicca sul `Submit` pulsante nella parte inferiore dello schermo.

Ora torna al `Settings` scheda nella parte superiore della pagina.

Cambiare il `Access token expiration` da `Short-lived` a `No expiration` sotto `Generated access token` clicca su `Generate`.

Copia il token che ora ti viene mostrato e incollalo nel `Dropbox Auth Token` casella di testo quando si crea una nuova posizione di archiviazione di tipo `Remote: Dropbox` in `AzuraCast`.

Attività sincronizzate

AzuraCast ha alcune attività in background che vengono regolarmente attivate per eseguire lavori di elaborazione interni necessari per il funzionamento delle stazioni.

In questa pagina troverete una panoramica di tutte le Sync Tasks che AzuraCast dipende.

Aumentare il timeout di esecuzione

Se si esegue regolarmente timeout mentre un'attività di sincronizzazione è l'elaborazione è possibile aumentare manualmente il tempo in cui le attività di sincronizzazione sono autorizzate a eseguire tramite il `azuracast.env` in `/var/azuracast/..`.

Nel `azuracast.env` cercare le voci seguenti e aumentare il tempo specificato in secondi alle proprie esigenze.

```
# The maximum execution time (and lock timeout) for the 15-second,  
1-minute and 5-minute synchronization tasks.
```

```
# SYNC_SHORT_EXECUTION_TIME=600
```

```
# The maximum execution time (and lock timeout) for the 1-hour  
synchronization task.
```

```
# SYNC_LONG_EXECUTION_TIME=1800
```

Ecco come dovrebbero essere queste linee dopo il cambiamento (i valori sono solo per la dimostrazione):

```
# The maximum execution time (and lock timeout) for the 15-second,  
1-minute and 5-minute synchronization tasks.
```

```
SYNC_SHORT_EXECUTION_TIME=900
```

```
# The maximum execution time (and lock timeout) for the 1-hour  
synchronization task.
```

```
SYNC_LONG_EXECUTION_TIME=2200
```

Dopo aver cambiato queste voci riavvia l'installazione di AzuraCast tramite:

Finestra terminale

```
docker-compose down  
docker-compose up -d
```

Attività sincronizzate

Riattivare Streamer

Quando si scollega manualmente un DJ dal vivo è possibile consentire ad AzuraCast di disabilitare il proprio account per un determinato periodo di tempo. Questo compito è responsabile della riattivazione degli account DJ dopo il passare del tempo.

CheckRequests

Questo compito è responsabile dell'invio di brani richiesti dagli utenti al Liquidsoap AutoDJ da riprodurre.

RunBackup

Questo compito è responsabile del controllo se è necessario eseguire un backup programmato.

CleanupRelays

Questo compito è responsabile della pulizia automatica dei dati del relè.

CheckMedia

Questo compito è responsabile della scansione della posizione di archiviazione multimediale per i file multimediali di nuova aggiunta o modifica che AzuraCast deve elaborare.

CheckFolderLe liste di gioco

Questa attività è responsabile dell'assegnazione automatica del supporto di una cartella alla playlist a cui viene assegnata la cartella.

CheckUpdate

Questa attività è responsabile di verificare se ci sono aggiornamenti disponibili per l'installazione di AzuraCast

RunAnalytics

Questo compito è responsabile dell'aggiornamento dei dati giornalieri degli ascoltatori e dell'analisi che può richiedere tempo per essere calcolato.

CleanupStoria

Questo compito è responsabile della pulizia dei vecchi dati di listener e cronologia delle canzoni in base alle impostazioni di sistema.

CleanupStorage

Questa attività è responsabile della cancellazione della directory temporanea dei dati e del controllo delle posizioni di archiviazione multimediali per i file che devono essere rimossi.

RotateLogs

Questo compito è responsabile dei registri rotanti e del backup dell'installazione di AzuraCast.

AggiornamentoGeoLite

Questa attività è responsabile per l'aggiornamento automatico del database GeoLite per la geolocalizzazione dell'indirizzo IP se hai fornito una chiave API.

Utenti

Autorizzazioni richieste

Per gestire gli utenti, gli utenti devono essere in un ruolo che ha il permesso “Amministra utenti” a livello globale.

Dal pannello di amministrazione “Utenti”, è possibile modificare eventuali utenti esistenti o crearne di nuovi. È possibile modificare i ruoli di cui un determinato utente è parte e forzare una modifica della password nel caso in cui l'utente abbia dimenticato la propria password.

Reimpostazione di una password dell'account

Se hai perso la password per accedere a un account, ma hai comunque accesso al terminale SSH per il server, puoi eseguire il seguente comando per generare una nuova password casuale per un account nel sistema.

Sostituire `YOUREMAILADDRESS` con l'indirizzo e-mail la cui password si intende reimpostare.

Finestra terminale

```
cd /var/azuracast  
./docker.sh cli azuracast:account:reset-password YOUREMAILADDRESS
```