Benutzerrollen und -fähigkeiten unter WordPress

Der ultimative Leitfaden für WordPress-Benutzerrollen und -fähigkeiten

Salman Ravoof, August 13, 2020



Mal geteilt

Die **Rollen** und **Fähigkeiten** der WordPress-Benutzer geben dir die Möglichkeit zu kontrollieren, was andere Benutzer auf deiner Webseite tun oder nicht tun können. Du kannst sie benutzen, um Benutzeraktionen wie das Schreiben und Bearbeiten von Beiträgen, das Erstellen neuer Seiten, das Moderieren von Kommentaren, das Installieren von Plugins, das Hinzufügen neuer Benutzer und vieles mehr zu verwalten.Testen Sie die kostenlose Demo

Um eine WordPress Webseite zu verwalten, ist es wichtig, die

Rollen und Rechte der Benutzer zu verstehen. Wenn du zum Beispiel eine Webseite für einen Kunden erstellst, möchtest du nicht, dass er das installierte Theme bearbeitet oder ändert. Genauso ist es unklug, Autoren eines Multi-Autoren-Blogs Plugins installieren oder entfernen zu lassen.

Wenn du lernst, wie man WordPress-Benutzerrollen intelligent verwaltet, kannst du deinen Arbeitsablauf rationalisieren, deine Webseite sicher halten und die ultimative Kontrolle über deine Webseite erlangen.

In dieser ausführlichen Anleitung erfährst du mehr über WordPress-Benutzerrollen, die verschiedenen Möglichkeiten, die WordPress bietet, wie du bestehende Benutzerrollen bearbeiten kannst, wie du Benutzer auf einer Multisite verwalten kannst und wie du neue Rollen mit neuen Funktionen erstellen kannst.

Aufgeregt? Lasst uns loslegen!

Inhaltsverzeichnis

1. Was sind die Rollen und Fähigkeiten der WordPress-Benutzer?
2. Die sechs Standard-WordPress-Benutzerrollen
3. Wie man WordPress-Benutzerrollen effektiv verwaltet
4. Benutzer im WordPress Multisite-Netzwerk verwalten
5. Wie man existierende WordPress-Benutzerrollen anpasst
6. Wie man benutzerdefinierte Benutzerrollen in WordPress erstellt
7. Wie man Benutzerrollen aus WordPress entfernt
8. Benutzerdefinierte Fähigkeiten in WordPress erstellen
9. Die besten WordPress User Rollen und Fähigkeiten Plugins
10. MyKinsta Benutzer-Rollen

Was sind die Rollen und Fähigkeiten

der WordPress-Benutzer?

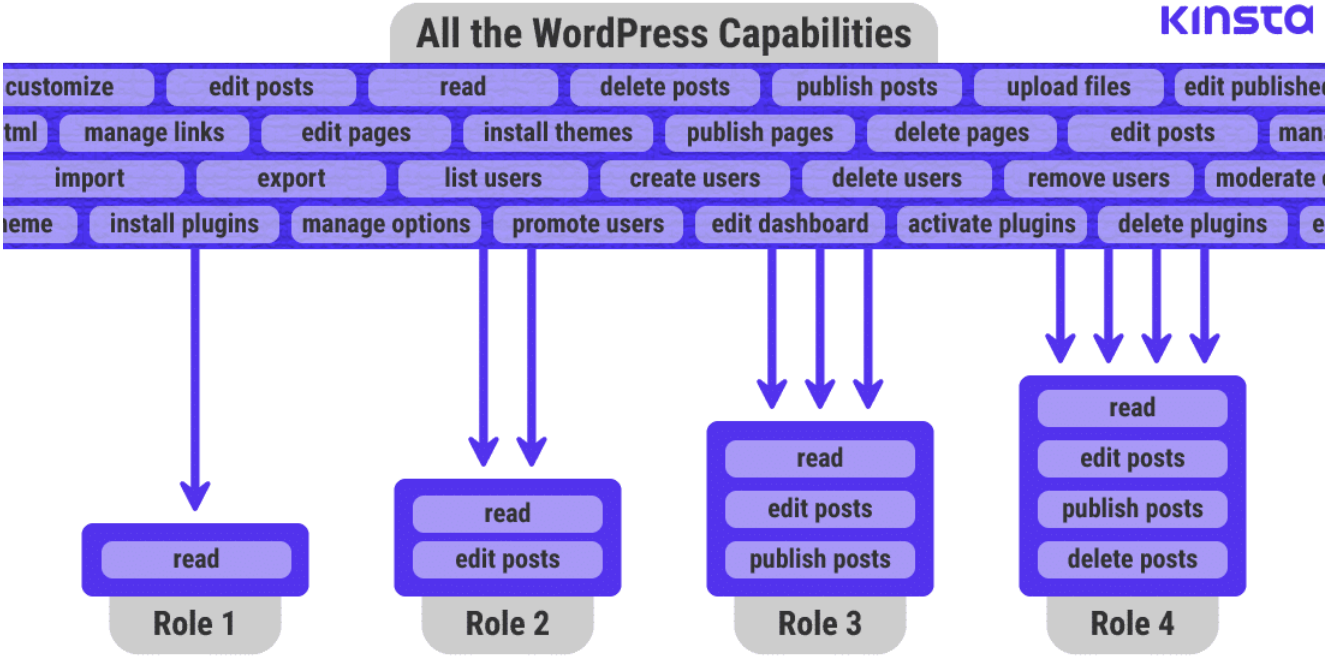
Rollen und Fähigkeiten sind das A und O der Benutzerzugangsverwaltung in WordPress. Um zu verstehen, was Benutzerrollen in WordPress sind, musst du zuerst wissen, was Fähigkeiten sind.

WordPress definiert jede Aktion, die ein Benutzer ausführen kann, als **Fähigkeit** (Capability). Hier sind ein paar Beispiele von Fähigkeiten, die in WordPress verfügbar sind und wie sie im Code von WordPress referenziert werden:

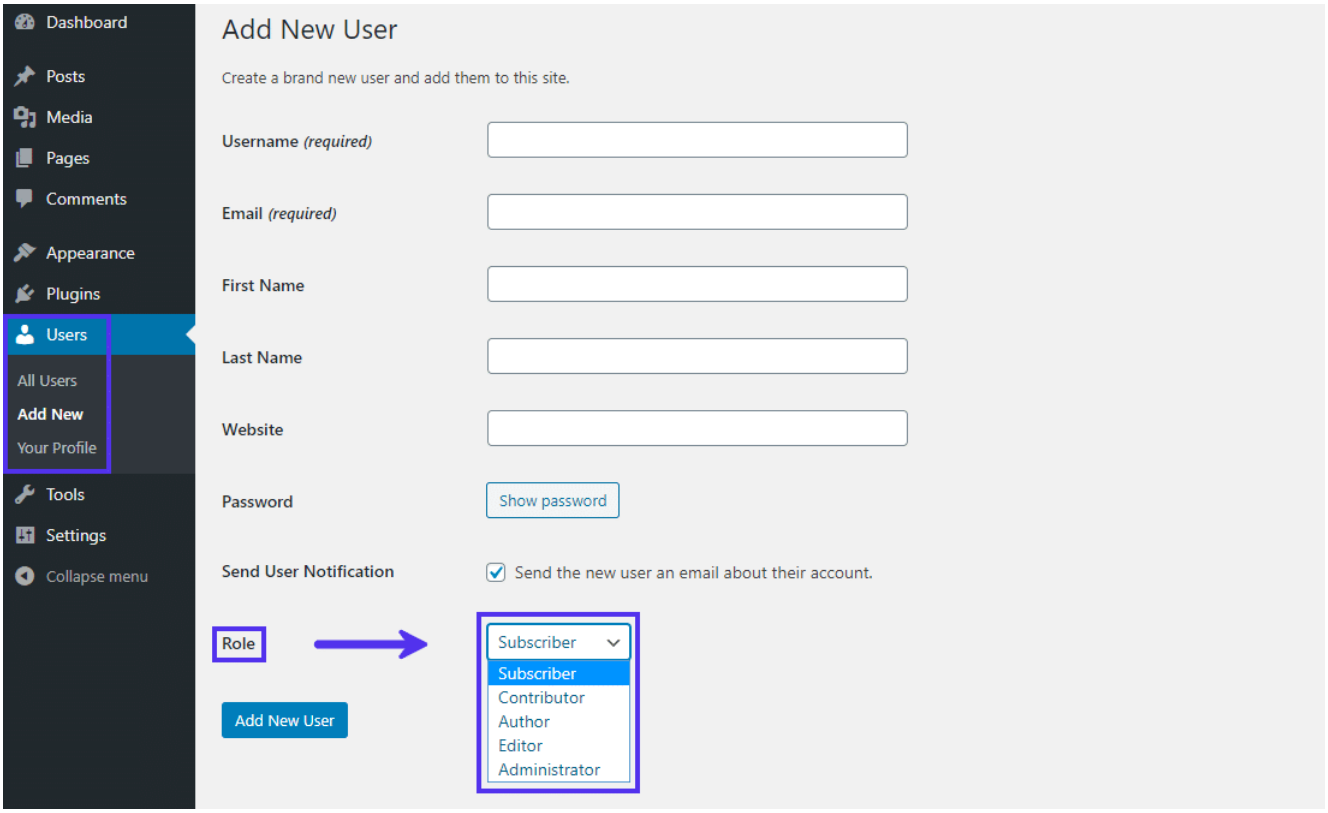
- Beiträge lesen (read)
- Beiträge schreiben und bearbeiten (edit_posts)
- Beiträge veröffentlichen (publish_posts)
- Plugins installieren (install_plugins)
- Themes löschen (delete_themes)
- Benutzer erstellen (create_users)
- Moderate Kommentare (moderate_comments)

Die meisten Fähigkeiten sind selbsterklärend durch ihren Namen. WordPress hat über 70 hardcodierte Fähigkeiten in seinem Core eingebaut.

Eine **Rolle** ist eine Sammlung von Fähigkeiten, die du einem Benutzer zuweisen kannst. Jedem WordPress-Benutzer muss eine Rolle zugewiesen werden. Ein Benutzer kann nur Aktionen ausführen, die ihm seine Rolle zuweist.

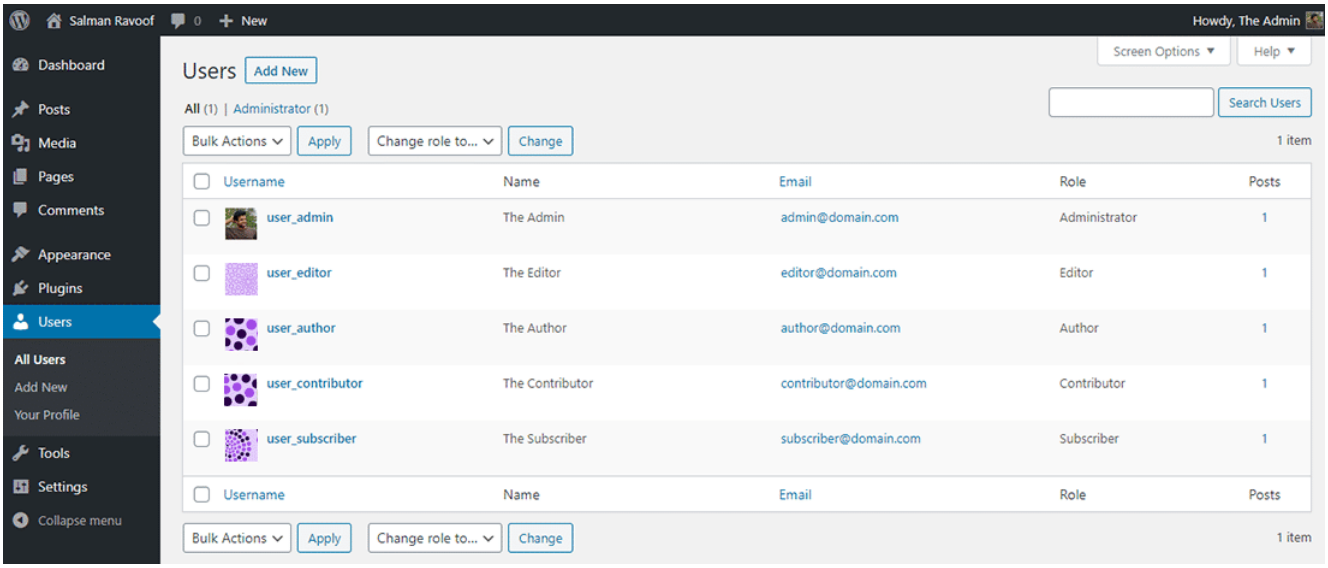


Eine ‚Rolle' ist eine Sammlung von ‚Fähigkeiten'.
In der obigen Grafik kann jeder Benutzer mit der **Rolle 1** Beiträge lesen, aber sie können keine Beiträge bearbeiten. Benutzer mit der **Rolle 2** können sowohl Beiträge lesen als auch bearbeiten, aber sie können keine Beiträge veröffentlichen. Jeder Benutzer mit der **Rolle 3** kann Beiträge lesen, bearbeiten und veröffentlichen, aber sie können keine Beiträge löschen, im Gegensatz zu Benutzern mit der **Rolle 4**, die Beiträge löschen können.



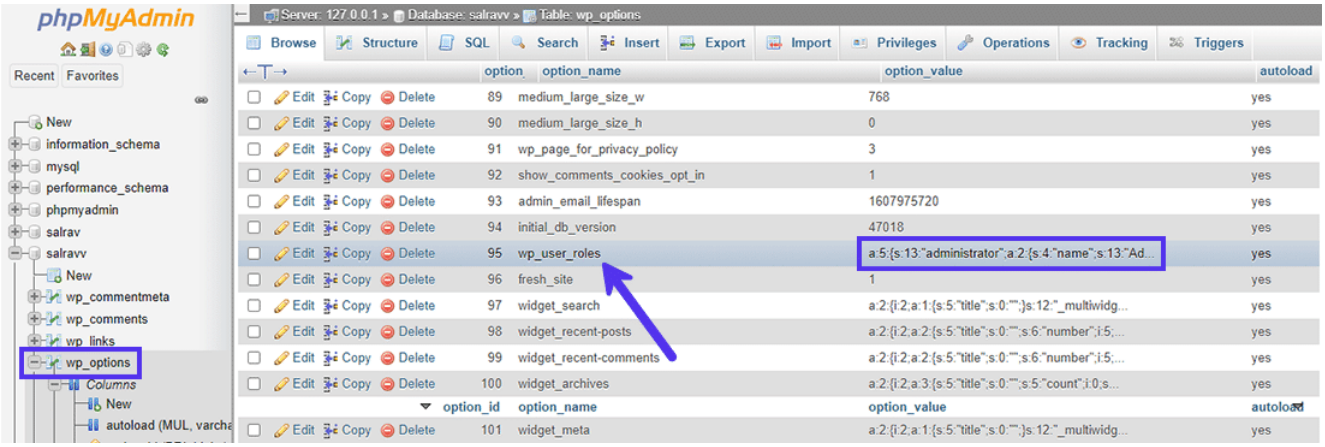
Das ‚Neuen Benutzer hinzufügen'-Panel im WordPress-Dashboard

WordPress nutzt viele seiner nativen Fähigkeiten, um seine Standardbenutzerrollen zu definieren. Zum Beispiel gewährt es Administratoren und Editoren die Fähigkeit **publish_pages**, aber es weist sie nicht den Abonnenten und Beitragenden zu.



Das ‚Benutzer'-Panel im WordPress-Dashboard

Als absolutes Minimum hat jeder WordPress-Benutzer einen Benutzernamen, ein Passwort, eine E-Mail-Adresse und eine Rolle.



phpMyAdmin zeigt, wo die WP-Datenbank die Fähigkeiten speichert

WordPress speichert all seine rollenbasierten Fähigkeiten in seiner Datenbank in der **wp_options**-Tabelle unter der serialisierten **wp_user_roles**-Option. Die **WP_Roles**-Core-Class wird verwendet, um zu definieren, wie Rollen und Fähigkeiten in der Datenbank gespeichert werden sollen.

Die WP_Roles-Class

WordPress implementiert Rollen und Fähigkeiten mit der User Roles API, von denen die meisten auf der WP_Roles Core Class basieren. Du kannst ihren Quellcode in der **wp-includes/class-wp-roles.php**-Datei finden.

Wenn du in die Datenbank schaust, wirst du feststellen, dass die Rollen in einem Array mit ihren definierten Rollennamen liegen. Der **rolename** key speichert den Namen der Benutzerrolle als Wert des **name** Key und alle Fähigkeiten in einem separaten Array als Wert des **capability** Key.

```
array (
     'rolename' => array (
          'name' => 'rolename',
          'capabilities' => array()
     )
)
```

Die WP_Roles Klasse definiert eine Menge Methoden. Du kannst sie überall in deinem Code aufrufen, um mit der User Roles API

zu interagieren.

Hinweis: WordPress enthält eine weitere Core Class namens WP_Role (beachte den Singular ‚Role'). Sie wird verwendet, um die User Roles API zu erweitern.

Wenn du den Schlüsselwert von **wp_user_roles** deinialisierst, wird es ungefähr so aussehen:

```
array (
  'administrator' =>
  array (
    'name' => 'Administrator',
    'capabilities' =>
    array (
      'switch_themes' => true,
      'edit_themes' => true,
      'activate_plugins' => true,
      // [...rest of the lines cut off for brevity...]
    ),
  ),
  'editor' =>
  array (
    'name' => 'Editor',
    'capabilities' =>
    array (
      'moderate_comments' => true,
      'manage_categories' => true,
      'manage_links' => true,
      // [...rest of the lines cut off for brevity...]
    ),
  ),
  'author' =>
  array (
    'name' => 'Author',
    'capabilities' =>
    array (
      'upload_files' => true,
      'edit_posts' => true,
      'edit_published_posts' => true,
      // [...rest of the lines cut off for brevity...]
```

```
    ),
  ),
  'contributor' =>
  array (
    'name' => 'Contributor',
    'capabilities' =>
    array (
      'edit_posts' => true,
      'read' => true,
      // [...rest of the lines cut off for brevity...]
    ),
  ),
  'subscriber' =>
  array (
    'name' => 'Subscriber',
    'capabilities' =>
    array (
      'read' => true,
      'level_0' => true,
    ),
  ),
)
```

Es ist ein multidimensionales Array, bei dem jede Rolle einen Rollennamen und eine Reihe von Fähigkeiten erhält. Ähnlich speichert WordPress die benutzerbasierten Fähigkeiten in der **wp_usermeta**-Tabelle mit dem **wp_capabilities**-Metaschlüsselnamen.

Hinweis: Das **wp_**-Präfix kann in deinem Setup unterschiedlich sein. Es hängt vom Wert der globalen Variablen **$table_prefix** in der **wp-config.php**-Datei deiner Webseite ab.

Rollen vs. Fähigkeiten-Tabelle

Capability	Super Admin	Administrator	Editor	Author	Contributor	Subscriber
install_plugins	Y	Y (single site)				
install_themes	Y	Y (single site)				
list_users	Y	Y				
manage_options	Y	Y				
promote_users	Y	Y				
remove_users	Y	Y				
switch_themes	Y	Y				
update_core	Y	Y (single site)				
update_plugins	Y	Y (single site)				
update_themes	Y	Y (single site)				
edit_dashboard	Y	Y				
customize	Y	Y				
delete_site	Y	Y				

Capability	Super Admin	Administrator	Editor	Author	Contributor	Subscriber
moderate_comments	Y	Y	Y			
manage_categories	Y	Y	Y			

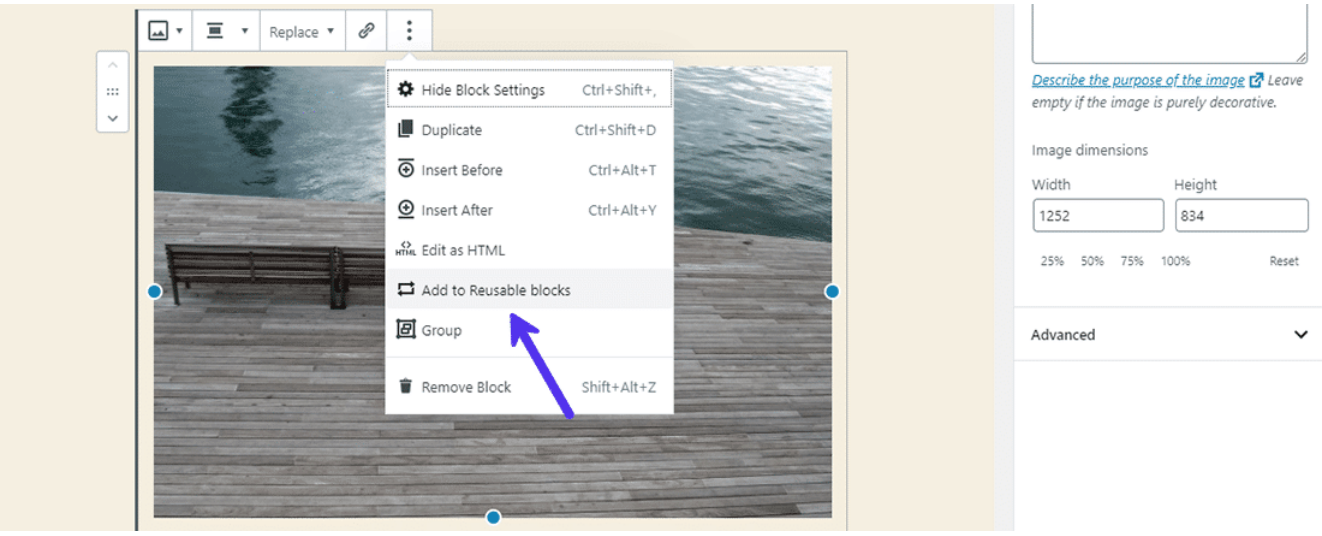
Die Tabelle ‚Rollen vs Fähigkeiten' im WordPress-Codex

Der WordPress Codex enthält eine einfache Fähigkeit-vs-Rolle-Tabelle, obwohl sie nicht so intuitiv ist. Er fasst alle Aktionen zusammen, die Standardbenutzerrollen sowohl in Einzel-Webseiten als auch in Multisite-WordPress-Setups ausführen können. Es gibt eine Pause nach einer bestimmten Anzahl von Fähigkeiten, um es dir leicht zu machen, zwischen hoch- und niedrigstufigen Fähigkeiten zu unterscheiden.

Für eine bessere Darstellung aller WordPress Rollen und Fähigkeiten auf einen Blick, könnt ihr euch diese ausgezeichnete Tabelle von Exygy ansehen.

Fähigkeiten im Zusammenhang mit wiederverwendbaren Blöcken von Gutenberg

WordPress' Gutenberg-Block-Editor hat eine erstaunliche Funktion namens **Wiederverwendbare Blöcke** eingeführt. Sie erlaubt es dir, einen ganzen Block (oder mehrere Blöcke) als Vorlage zu speichern und sie überall auf deiner Webseite zu verwenden.



Hinzufügen von ‚Wiederverwendbaren Blöcken' in WordPress' neuer Gutenberg-Block-Editor

Dementsprechend hat WordPress auch die folgenden neuen Fähigkeiten in Bezug auf wiederverwendbare Blöcke eingeführt:

- Wiederverwendbarer Blöcke erstellen
- Wiederverwendbare Blöcke bearbeiten
- Wiederverwendbare Blöcke lesen
- Wiederverwendbare Blöcke löschen

Die oben aufgeführten Fähigkeiten funktionieren ähnlich wie die Post-bezogenen Fähigkeiten. Ein Admin oder Editor hat Zugriff auf alle wiederverwendbaren Blöcke, während ein Autor nur wiederverwendbare Blöcke bearbeiten oder löschen kann, die von ihm erstellt wurden. Contributors können nur wiederverwendbare Blöcke lesen.

Besondere Fähigkeit: Ungefilterter Upload

Ungefilterter Upload ist eine besondere Fähigkeit, die standardmäßig keiner Benutzerrolle zugeordnet ist, einschließlich Administrator oder Super Admin. Sie erlaubt es einem Benutzer, Dateien mit beliebigen Erweiterungen (z.B. SVG oder PSD) hochzuladen, nicht nur die, die von WordPress auf der Whitelist stehen.

Hinweis: Du kannst eine Liste der von WordPress unterstützten Mime-Typen und Dateierweiterungen mit der Funktion wp_get_mime_types() erhalten.

Um diese Funktion zu aktivieren, musst du den unten programmierten Codeschnipsel zu deiner **wp-config.php** Datei hinzufügen. Definiere die Konstante vor der Zeile, die dich auffordert, mit dem Editieren aufzuhören.

```
define( 'ALLOW_UNFILTERED_UPLOADS', true );
```

Nachdem du diese Konstante definiert hast, kannst du jeder Benutzerrolle auf einer WordPress Single-Webseite die **Unfiltered Upload** Fähigkeit geben. Auf einer Multisite-Installation kann jedoch nur ein Super Admin diese Fähigkeit haben.

Wenn du zum Beispiel einem Editor die **Unfiltered_Upload**-Fähigkeit zuweisen willst, kannst du den folgenden Code an beliebiger Stelle in deinen WordPress-Code einfügen (idealerweise nur bei der Aktivierung von Themes/Plugins ausführen):

```
<?php

  $role = get_role( 'editor' );
  $role->add_cap( 'unfiltered_upload' );

?>
```

Wir werden später in diesem Beitrag mehr darüber diskutieren, wie man die Fähigkeiten aller Benutzerrollen oder bestimmter Benutzer hinzufügt oder anpasst.

Primitive Fähigkeiten vs. Meta-Fähigkeiten

Es gibt hauptsächlich zwei Arten von Fähigkeiten in WordPress:

- **Primitive Fähigkeiten:** Diese Fähigkeiten werden für bestimmte Rollen gewährt. Benutzer mit diesen Rollen erben die primitiven Fähigkeiten automatisch.
- **Meta-Fähigkeiten:** Diese Fähigkeiten werden standardmäßig keiner Rolle zugewiesen. WordPress sucht nach einem

bestimmten Objekt in seinem Code und seiner Datenbank, z.B. einem Beitrag, einer Seite, einem Benutzer oder einer Taxonomie, und wenn die Logik stimmt, „mappt“ es eine Meta-Fähigkeit auf eine oder mehrere primitive Fähigkeiten.

Beispielsweise gewährt WordPress den Autoren die **edit_posts**-Fähigkeit für ihre eigenen Beiträge, damit sie diese bearbeiten können. Diese Fähigkeit erlaubt es ihnen jedoch nicht, die Beiträge anderer Benutzer zu bearbeiten. Hier helfen die Meta-Fähigkeiten.

WordPress benutzt die map_meta_cap()-Funktion, um eine Reihe von primitiven Fähigkeiten zurückzugeben, die an ein bestimmtes Objekt gebunden sind. Dann vergleicht es sie mit dem Benutzerobjekt, um zu prüfen, ob der Benutzer den Beitrag bearbeiten kann.

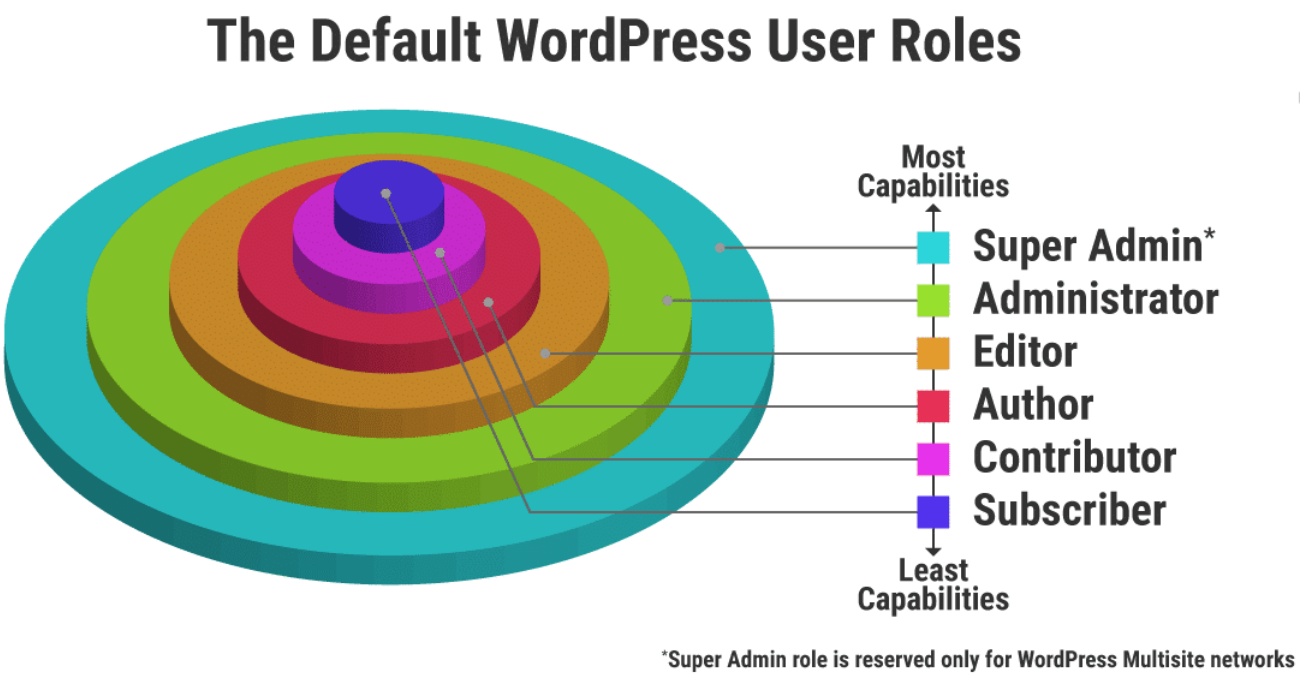
Ein paar andere Beispiele für Meta-Fähigkeiten sind **read_post**, **delete_post**, **remove_user** und **read_post**. Wir werden uns diese im Abschnitt über benutzerdefinierte Fähigkeiten weiter unten genauer ansehen. WordPress-Benutzerrollen und -Fähigkeiten verstehen = eine sicherere Webseite + Seelenfrieden für dich und deine Kunden. Mehr erfahren Click to Tweet

Die sechs Standard-WordPress-Benutzerrollen

WordPress beinhaltet sechs vordefinierte Benutzerrollen. Der erste Benutzer einer WordPress-Installation erhält standardmäßig die **Administrator**-Rolle (oder die **Super-Admin**-Rolle bei WordPress-Multisite-Installationen).

Da WordPress als Blogging-Plattform begann, bevor es sich zu einem vollwertigen CMS entwickelte, definiert es die meisten seiner Benutzerrollen, um Inhalte im Web zu veröffentlichen. Die anderen vordefinierten Benutzerrollen sind **Editor**, **Autor**,

Mitwirkender und **Abonnent**.



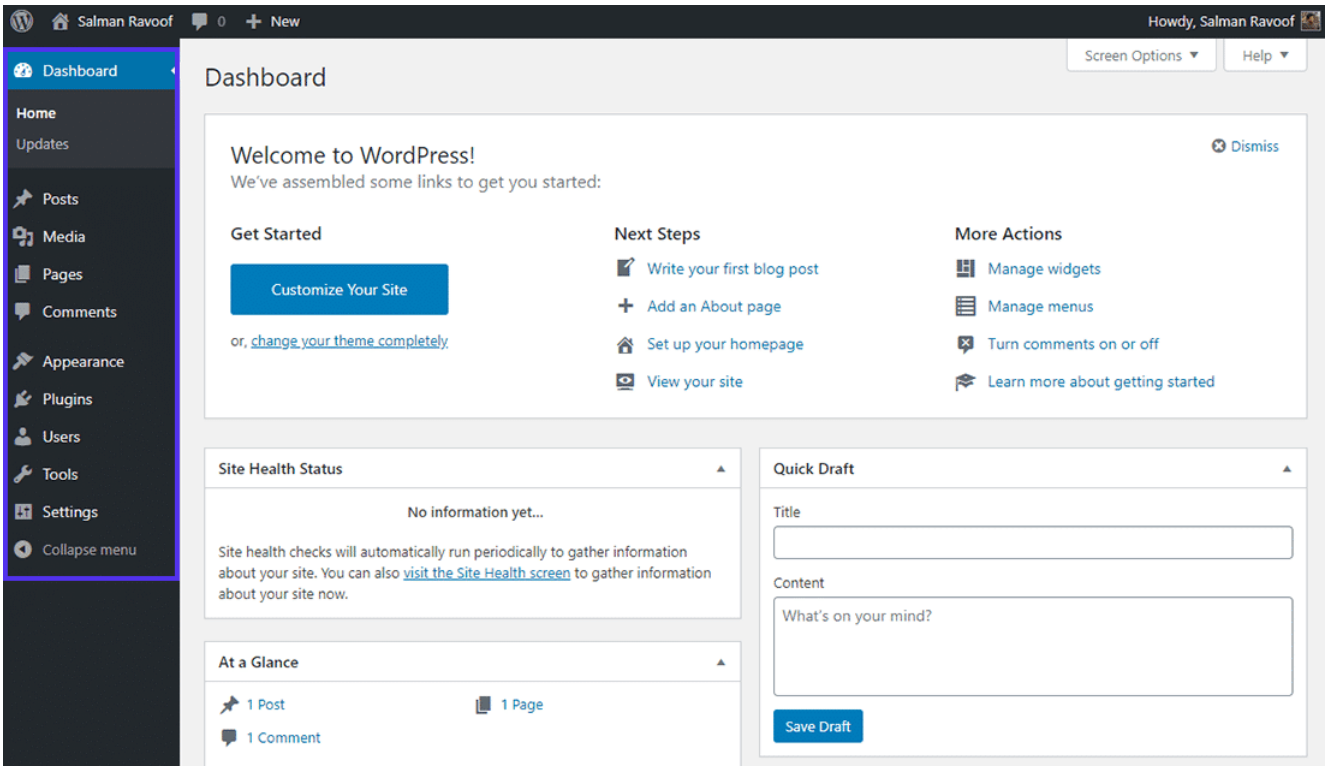
Die standardmäßigen WordPress-Benutzerrollen, gestapelt in der Reihenfolge ihrer Fähigkeiten
Stellt euch die Standardrollen der WordPress-Benutzer als eine Ansammlung gestapelter Zylinder vor, die verschiedene Fähigkeiten repräsentieren. Der größte Zylinder hat die meisten Fähigkeiten, der zweitgrößte hat die zweithöchsten Fähigkeiten und der kleinste Zylinder hat die wenigsten Fähigkeiten.

Du solltest eine Rolle nicht als einer anderen überlegen ansehen. Vielmehr solltest du dir Rollen so vorstellen, dass sie die Verantwortlichkeiten eines Benutzers innerhalb der Webseite festlegen.

> *Eine Benutzerrolle ist niemals übergeordnet oder untergeordnet, sie definiert genau das, wofür sie bestimmt ist.*

Schauen wir uns nun alle vordefinierten WordPress-Benutzerrollen im Detail an.

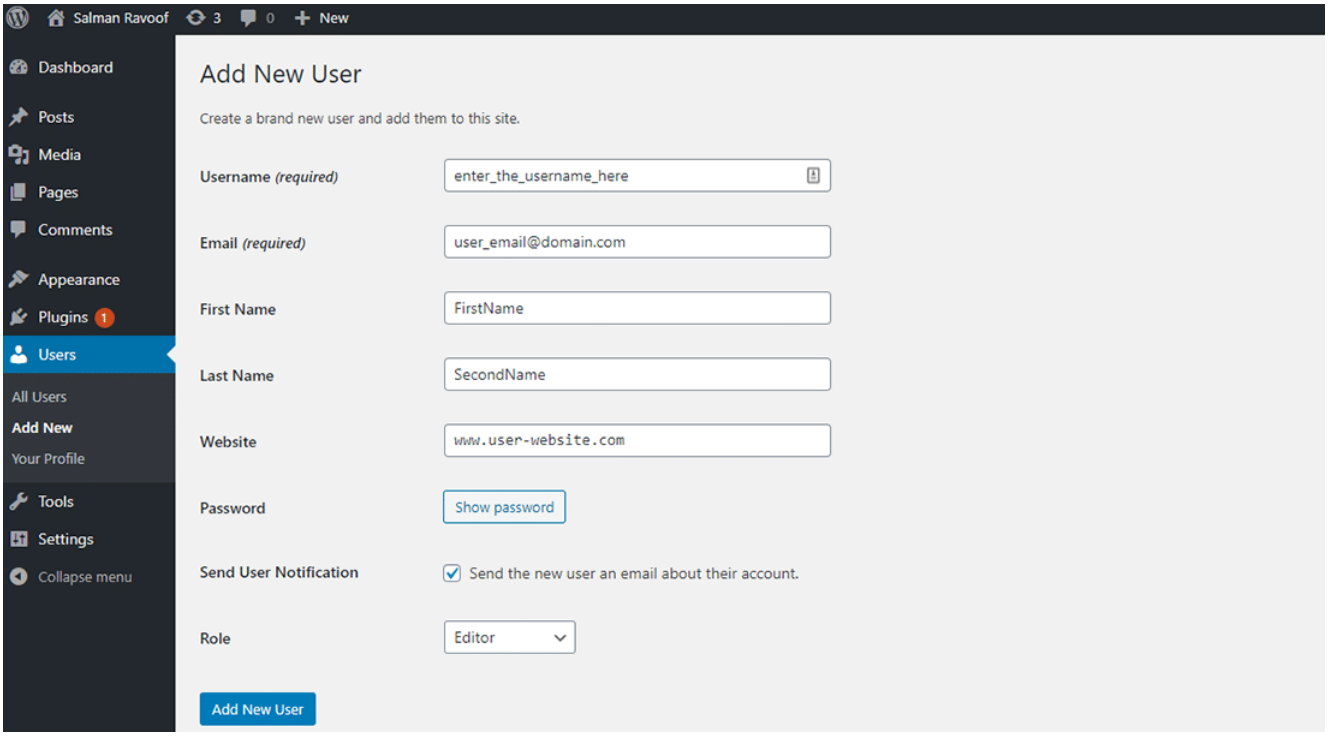
Administrator



Das ‚Administrator'-Rollen-Dashboard in WordPress

WordPress weist dem ersten Benutzer einer Webseite die Rolle des Administrators zu. Er sitzt an der Spitze aller anderen Benutzerrollen und hat Zugriff auf alle von WordPress definierten Fähigkeiten. Benutzer mit der Administrator-Rolle können zum Beispiel folgende Aktionen ausführen

- Benutzer erstellen und löschen
- Plugins und Themes installieren und verwalten
- Plugins, Themes, Dateien und programmierten Code bearbeiten

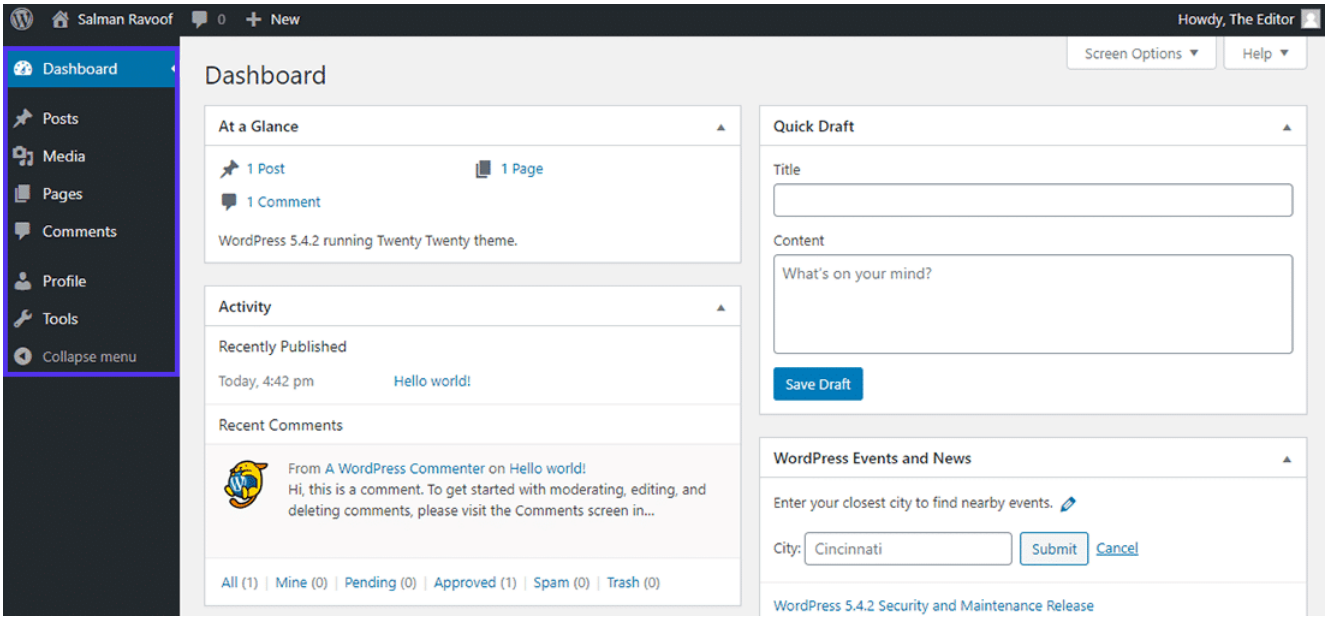


Nur Administratoren können neue Benutzer in WordPress hinzufügen

Da ein Administrator die mächtigste Rolle ist, solltest du sie nur denjenigen zuweisen, denen du vertraust. Im Idealfall sollte es nur einen Administrator pro Webseite geben.

Die Rolle des Administrators in einem WordPress Multisite-Netzwerk ist etwas anders definiert, obwohl sie gleich genannt wird. In einem Multisite-Netzwerk genießt die Rolle des Administrators nicht die Fähigkeiten, die sie in einer einzelnen WP-Webseite hat, wie das Installieren von Themes und Plugins. WordPress behält sich diese Fähigkeiten für die Super Admin Rolle vor.

Redakteur



Das Rollen-Dashboard ‚Redakteur' in WordPress

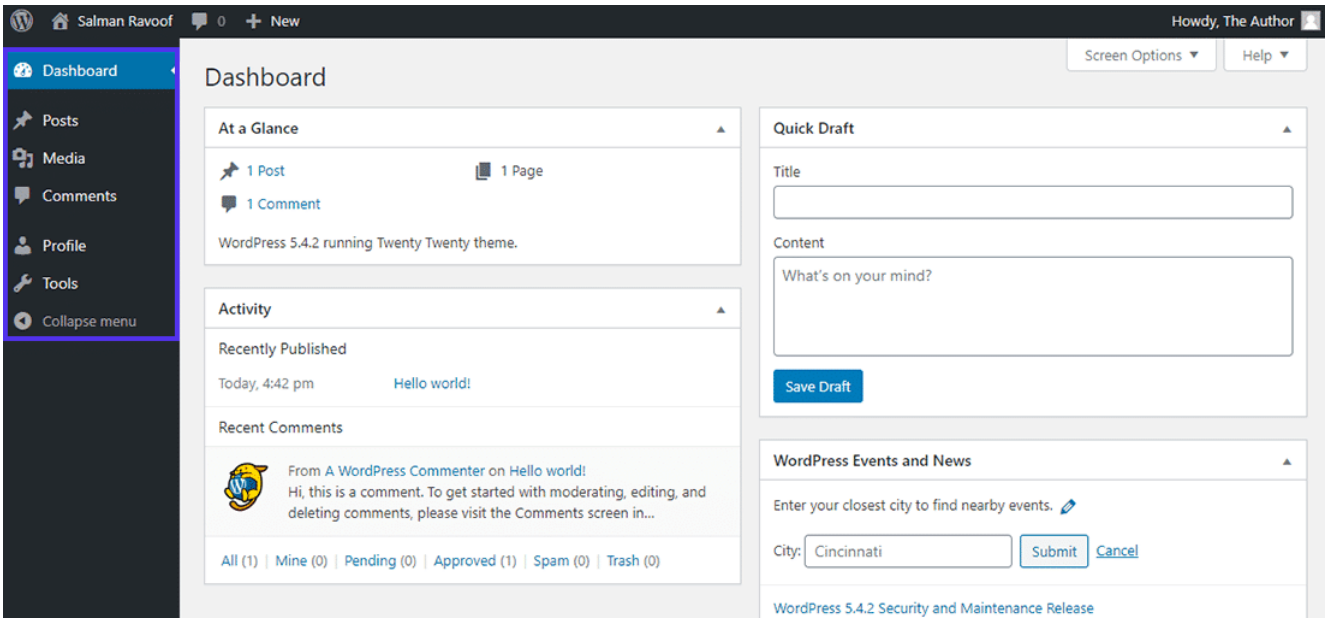
Ein Redakteur kümmert sich um die Verwaltung von Inhalten in einer WordPress Webseite. Er kann Beiträge und Seiten erstellen, ändern, veröffentlichen oder löschen, auch solche, die von anderen Benutzern erstellt wurden. Einige ihrer Fähigkeiten beinhalten:

- Veröffentlichte Beiträge und Seiten löschen
- Kommentare moderieren
- Links und Kategorien verwalten
- Beiträge und Seiten anderer Benutzer bearbeiten

Redakteure können keine Maßnahmen zur Verwaltung der Webseite ergreifen, wie zum Beispiel die Installation von Plugins und Themes. Ihre Hauptverantwortung besteht darin, die Arbeit anderer Autoren und Mitwirkender zu beaufsichtigen oder ein Ein-Personen-Inhaltsteam zu sein.

Tipp: Wenn du eine WordPress Webseite selbst verwaltest, kannst du dir einen alternativen Benutzer mit der Rolle eines Redakteurs anlegen. Auf diese Weise kannst du deine Verwaltungs- und Veröffentlichungsaufgaben getrennt halten. Dein Admin-Konto ist vor Hackern sicher, auch wenn dein Redakteurskonto kompromittiert wird.

Autor



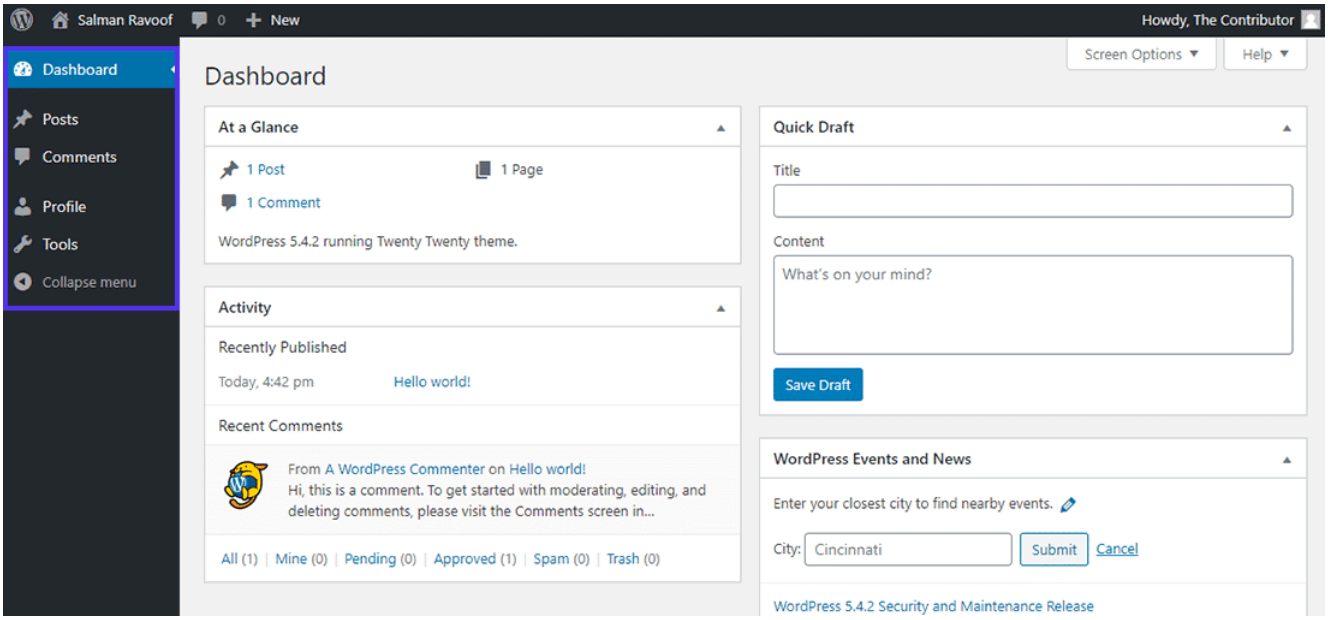
Das ‚Autor'-Rollen-Dashboard in WordPress

Wie der Name schon sagt, kann jeder Benutzer mit der Autorenrolle Beiträge erstellen, bearbeiten und veröffentlichen. Sie können auch Mediendateien hochladen und ihre eigenen Beiträge löschen, aber sie können keine Seiten erstellen oder die Beiträge anderer Benutzer bearbeiten.

Autoren können Tags zu ihren Beiträgen hinzufügen und ihre Beiträge bestehenden Kategorien zuweisen, aber sie können keine neuen Kategorien erstellen. Wie bei Redakteuren haben sie keinen Zugang zu administrativen Aufgaben wie Einstellungen, Plugins und Themes.

Hinweis: Ein Autor kann seine Beiträge auch löschen, nachdem sie veröffentlicht wurden. Wenn du jemandem die Rolle eines Autors zuweist, stelle sicher, dass es für dich in Ordnung ist, wenn er die vollständige Kontrolle über seine Beiträge hat, einschließlich des Löschens.

Mitwirkender

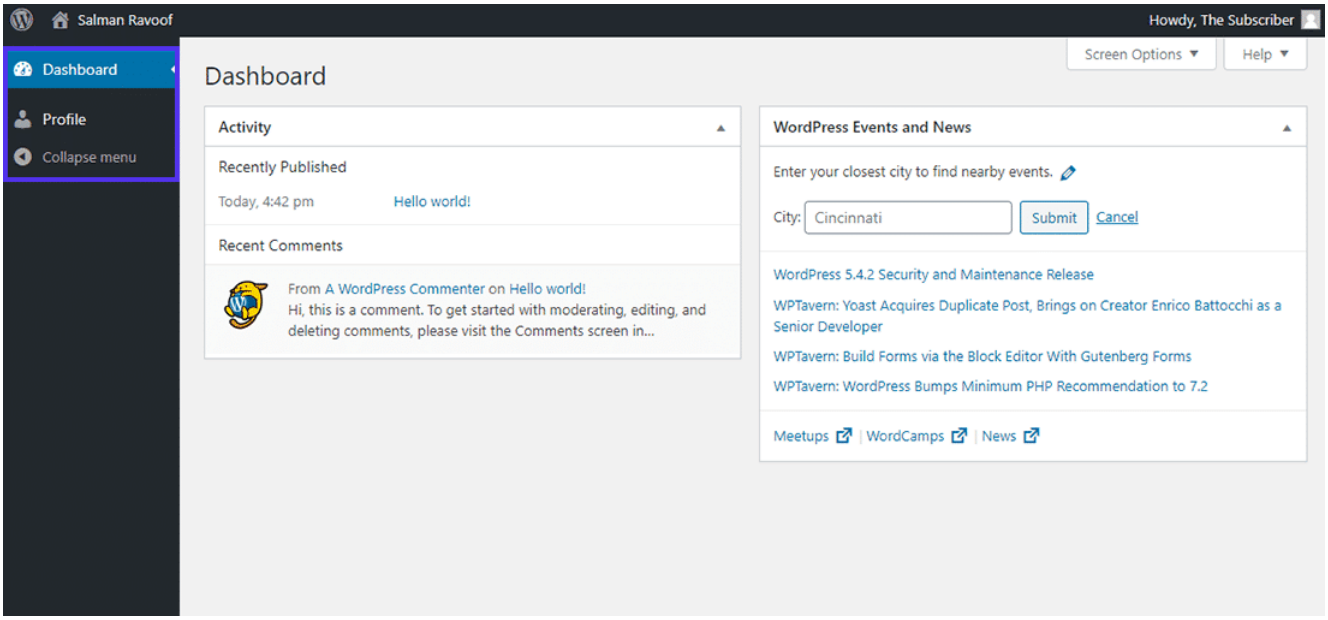


Das ‚Mitwirkender'-Rollen-Dashboard in WordPress
Die Rolle des Mitwirkenden ist eine Lite-Version der Rolle des Autors. Ein Benutzer mit der Mitwirkenden Rolle kann seine eigenen Beiträge erstellen, Entwürfe seiner Beiträge löschen, aber er kann die Beiträge nicht veröffentlichen.

Er kann die Entwürfe seiner Beiträge speichern oder sie an einen Redakteur oder Administrator zur Überprüfung und Veröffentlichung schicken. Und sobald er den Beitrag veröffentlicht hat, kann ein Beitragender seine Beiträge nicht mehr löschen. Im Gegensatz dazu können Autoren ihre veröffentlichten Beiträge löschen.

Die Rolle des Beitragenden ist ideal für neue Autoren und Gastbeitragende.

Abonnent



The ‚Subscriber' role dashboard in WordPress

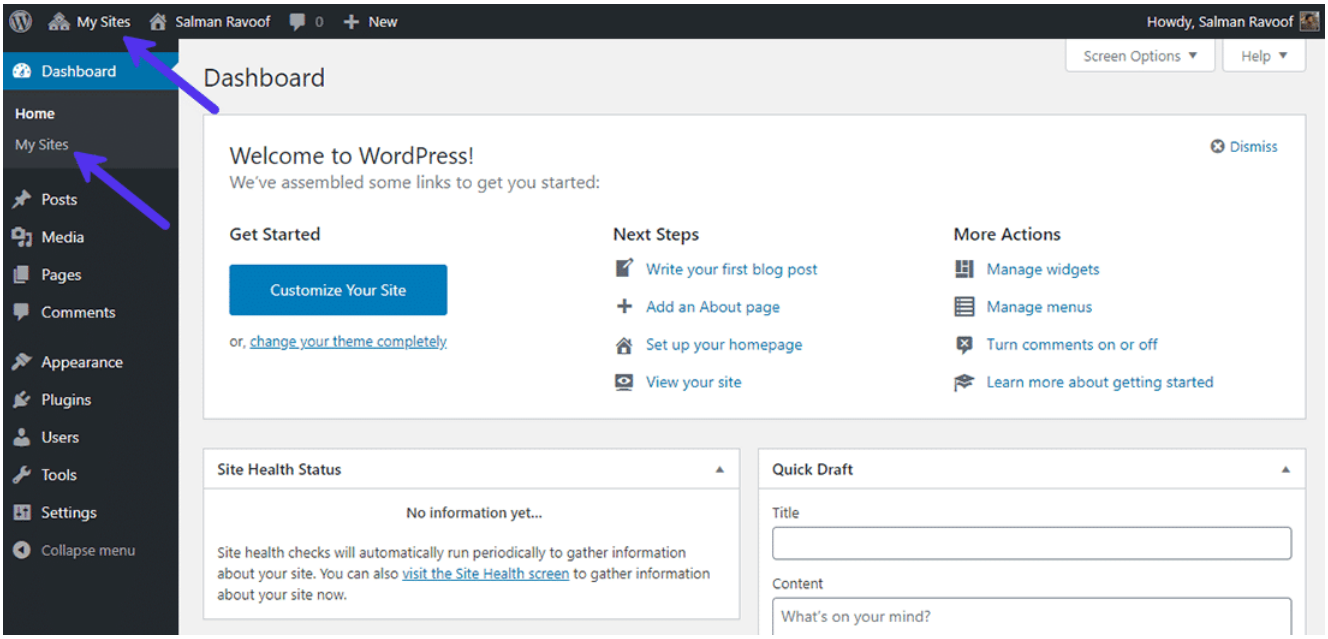
Die Rolle des Abonnenten befindet sich auf der untersten Stufe der Rangliste der Fähigkeiten. Ein Benutzer mit der Rolle „Abonnent" kann sein Profil verwalten und hat Zugang zu allen Beiträgen auf der Webseite. Das ist so ziemlich alles!



Du kannst den Inhalt auf nur eingeloggte Benutzer beschränken, einschließlich Abonnenten

Normalerweise hat jeder Zugang zum Lesen von Inhalten auf einer WordPress Webseite. Auf abonnierten oder mitgliedschaftlichen Webseiten können jedoch nur eingeloggte Benutzer den Inhalt sehen. Ein Benutzer mit der Rolle eines Abonnenten kann in diesen Fällen die Beiträge lesen.

Super Admin

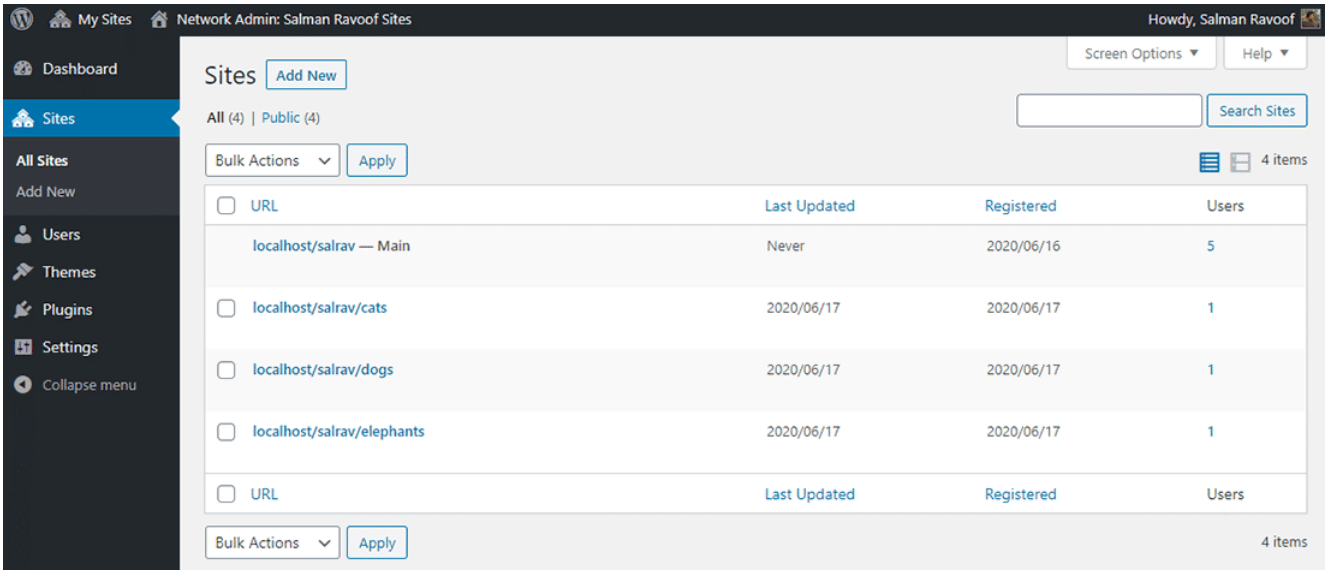


Das ‚Super Admin' Rollen-Dashboard im WordPress Multisite-Netzwerk

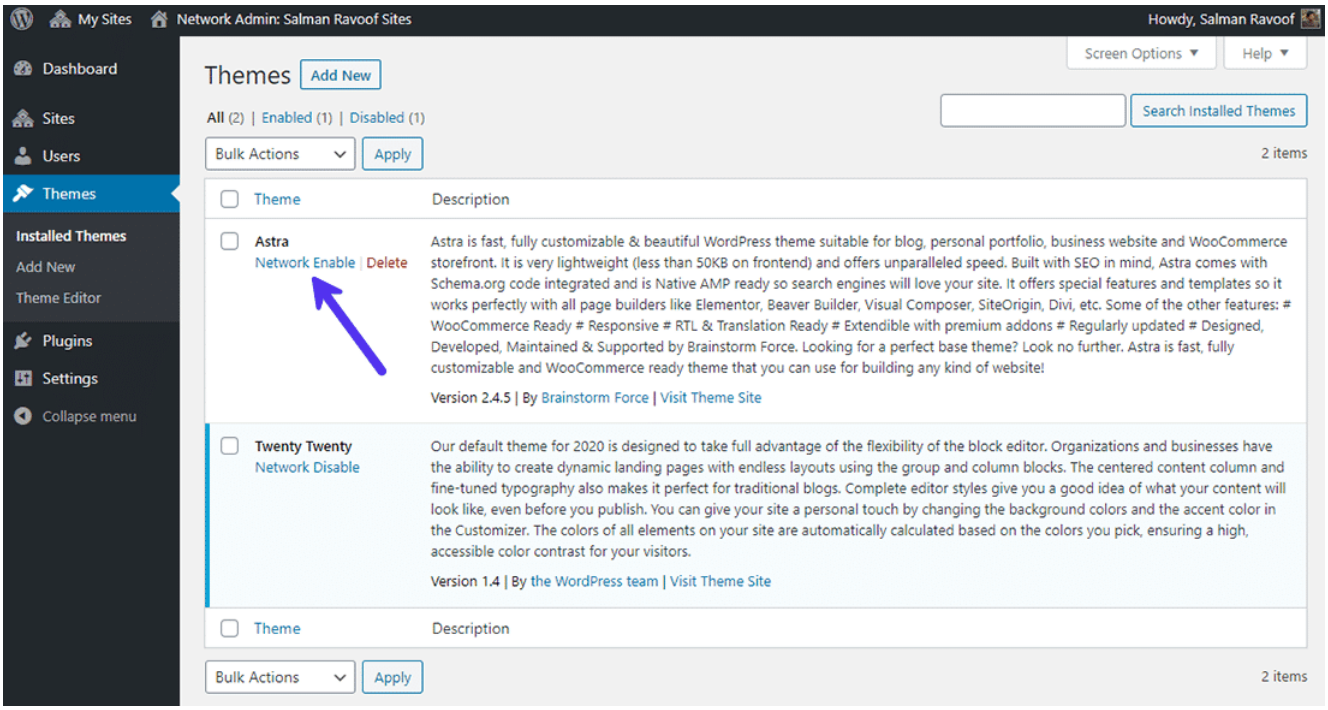
Die Super Admin Rolle ist nur bei WordPress Multisite Installationen verfügbar. Diese Rolle ersetzt die Single-Site-Admins innerhalb des Multisite-Netzwerks und gibt Zugang zu allen High-Level-Admin-Fähigkeiten.

Einige Multisite-Fähigkeiten sind nur für Super Admins verfügbar:

- Erstellen, Verwalten und Löschen von Webseiten im Netzwerk
- Verwalte Netzwerkbenutzer, Plugins, Themes und Optionen
- Alle Webseiten des Multisite-Netzwerks aktualisieren
- Ein Multisite-Netzwerk einrichten
- Weisen Sie den einzelnen Webseiten des Netzwerks Admins zu



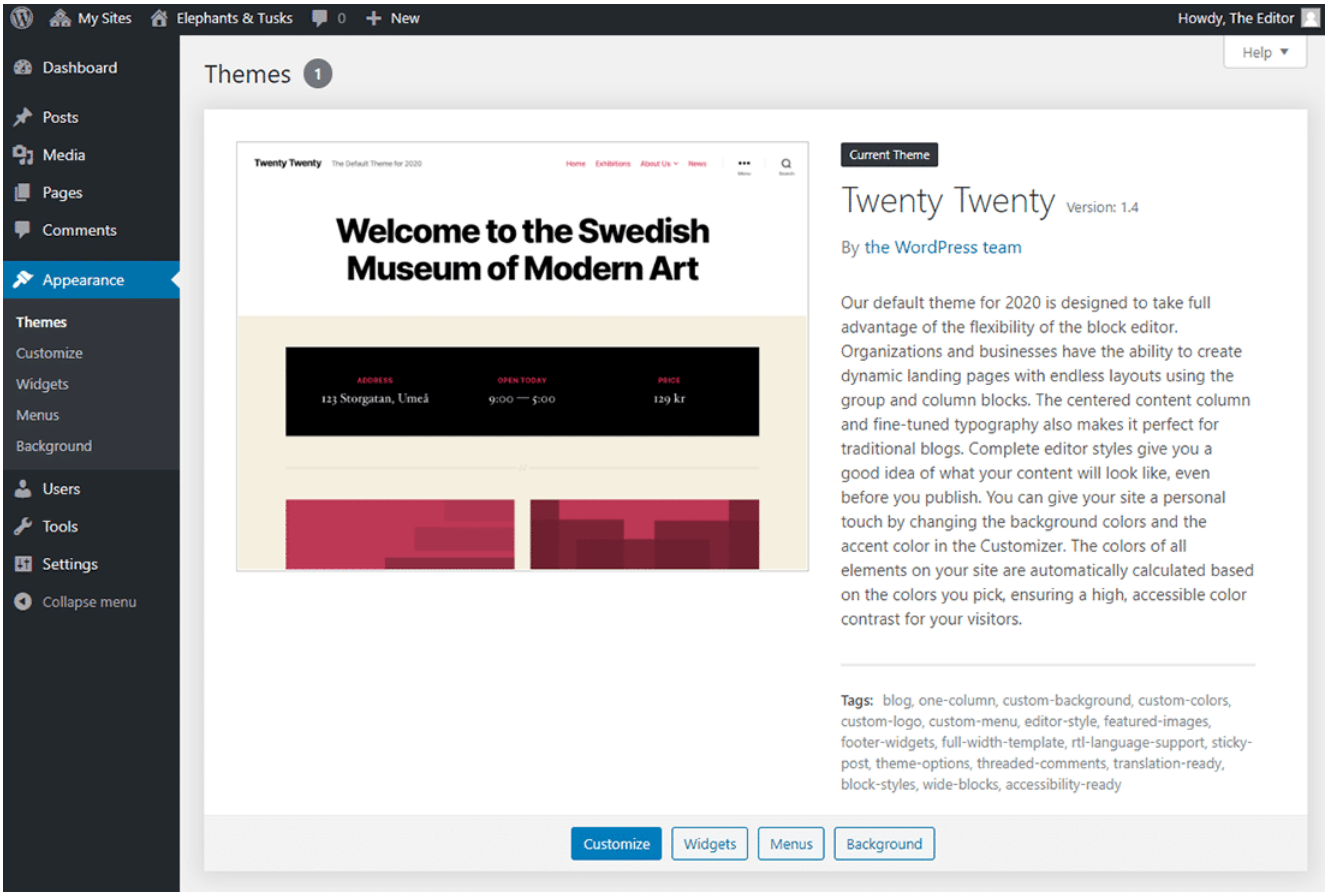
Das Panel ‚Webseiten' in einem WordPress Multisite-Netzwerk



Das ‚Themes'-Panel im Super Admin Dashboard

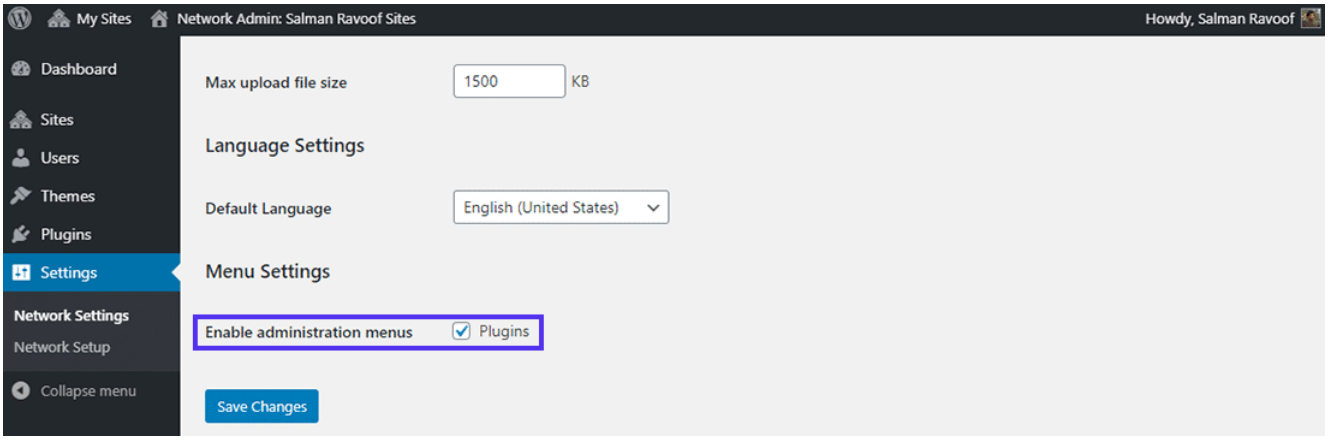
In einem Multisite-Netzwerk kann nur der Super Admin Themes installieren und sie im gesamten Netzwerk aktivieren. Admins von einzelnen Webseiten im Netzwerk können nur Themes ansehen und aktivieren, die bereits vom Super Admin installiert wurden.

Ich habe zum Beispiel das kostenlose Astra Theme in meinem Netzwerk installiert, aber nicht für das Netzwerk aktiviert. Daher können Administratoren einzelner Unterseiten in einem Netzwerk es nicht unter ihrem **Themes**-Panel aufgelistet sehen.

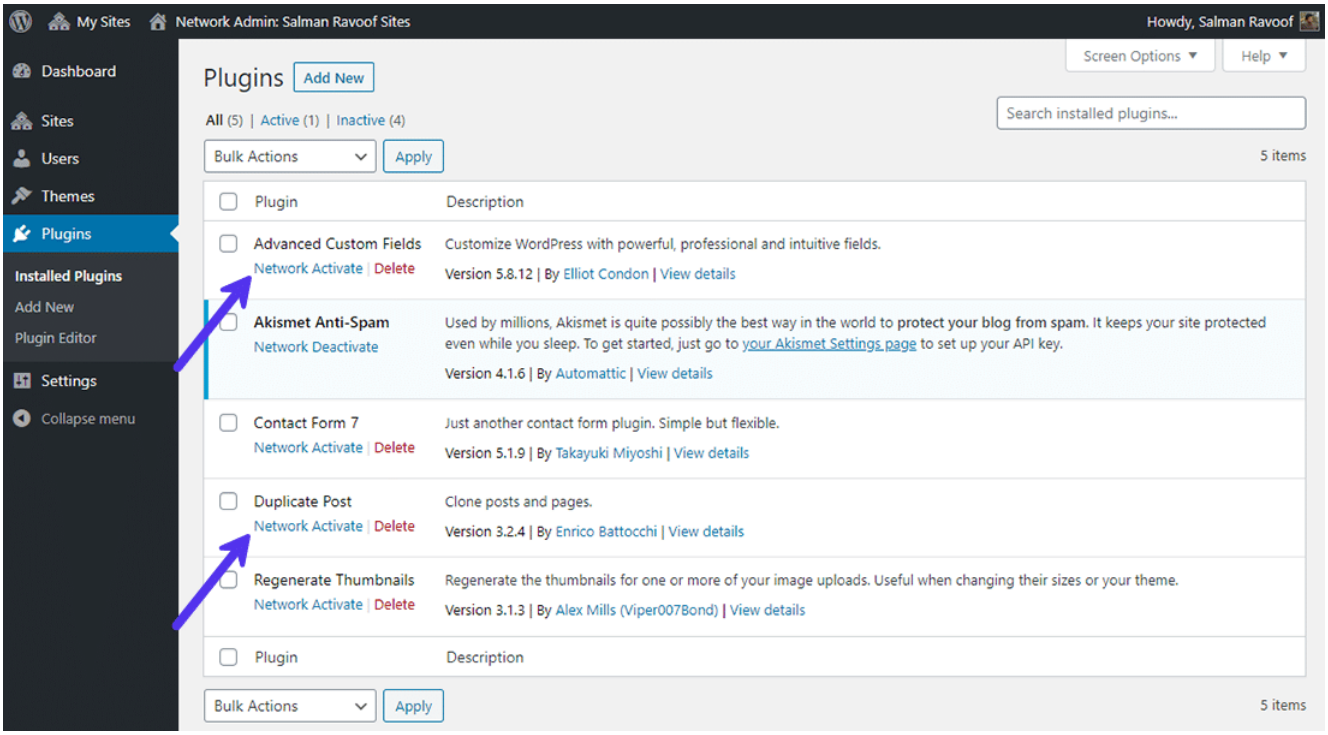


Administratoren von Netzwerkunterseiten können keine neuen Themes installieren

Auf dem obigen Screenshot kannst du auch sehen, dass das **Plugins**-Menü für Administratoren von Webseiten in einem Netzwerk nicht zugänglich ist. Anders als bei Themes kann ein Super Admin die Netzwerkeinstellungen ändern, damit die Admins die Plugins auf ihren Webseiten installieren und aktivieren können.



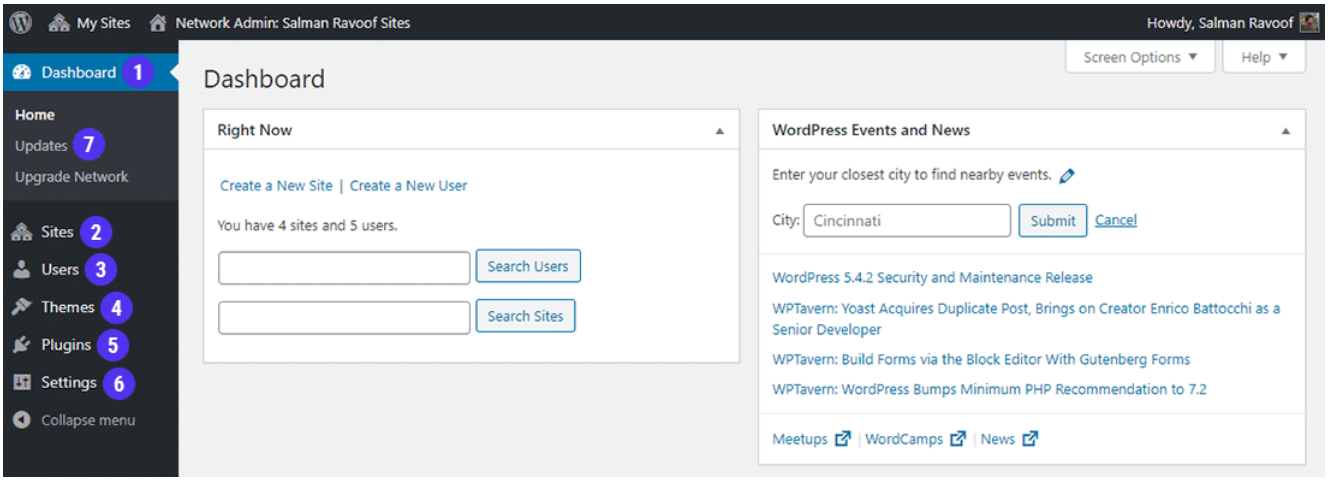
Der Super Admin kann Administratoren die Möglichkeit geben, Plugins zu verwalten



Der Super Admin kann auch Plugins im ‚Netzwerk aktivieren'. Ein Super Admin kann auch Plugins über das **Netzwerk aktivieren**, um sicherzustellen, dass sie über alle Webseiten im Netzwerk erzwungen werden. Webseiten Administratoren können netzwerkaktivierte Plugins nicht deaktivieren. Diese Einstellung ist ideal, um wichtige Plugins im gesamten Netzwerk zu erzwingen.

Der Netzwerk Admin Bildschirm

Das **Network Admin** Dashboard dient als zentraler Knotenpunkt für den Super Admin, um die Netzwerkfähigkeiten von WordPress Multisite zu verwalten. Es ist nur für Benutzer mit der Rolle des Super Admin zugänglich, nachdem sie ein Netzwerk erstellt haben.

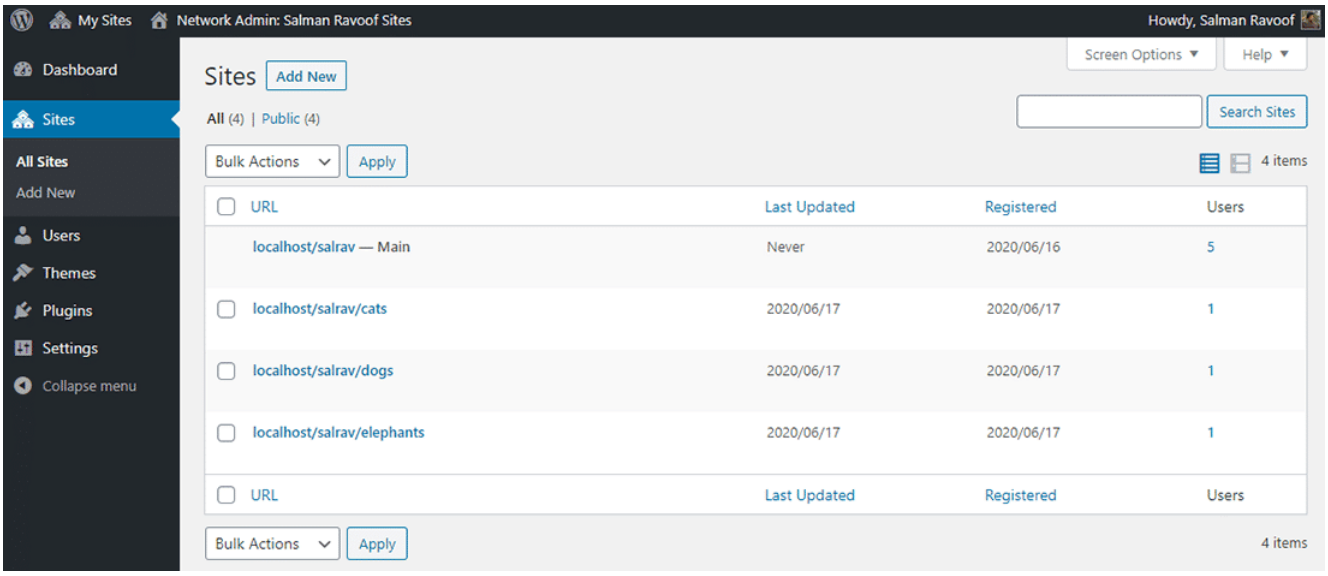


Das Netzwerk Admin Dashboard enthält einzigartige Optionen, um das Netzwerk zu verwalten

1. Dashboard

Das Netzwerk Admin Dashboard ist der zentrale Knotenpunkt für detaillierte Informationen über deine Webseiten im Netzwerk. Es gibt dir Zugriff auf alle deine Netzwerkeinstellungen.

2. Webseiten

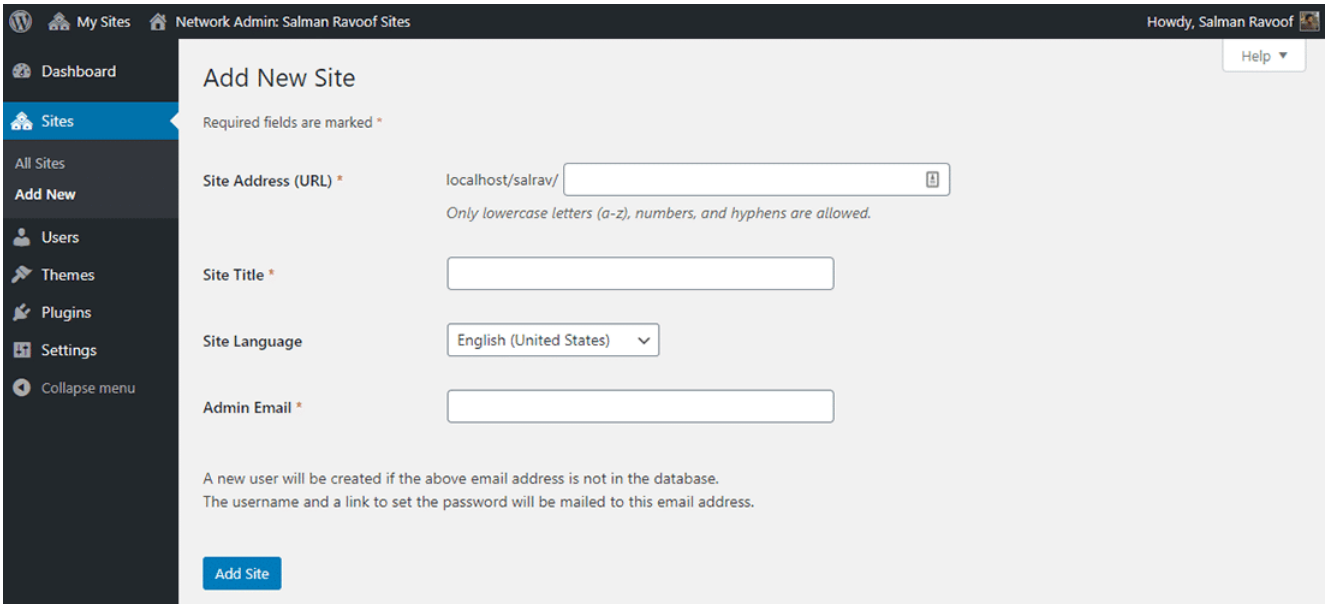


Das ‚Webseiten'-Panel in einem Network Admin Dashboard
Du kannst das Webseiten Panel benutzen, um die verschiedenen Webseiten zu verwalten, die Teil des Multisite-Netzwerks sind. Die hier aufgelisteten Webseiten werden entweder ein Unterverzeichnis oder eine Subdomain sein, je nachdem, wie du dein WordPress Multisite-Netzwerk konfiguriert hast.

Von hier aus kannst du entweder neue Webseiten zum Netzwerk

hinzufügen oder bestehende Webseiten aus dem Netzwerk entfernen.

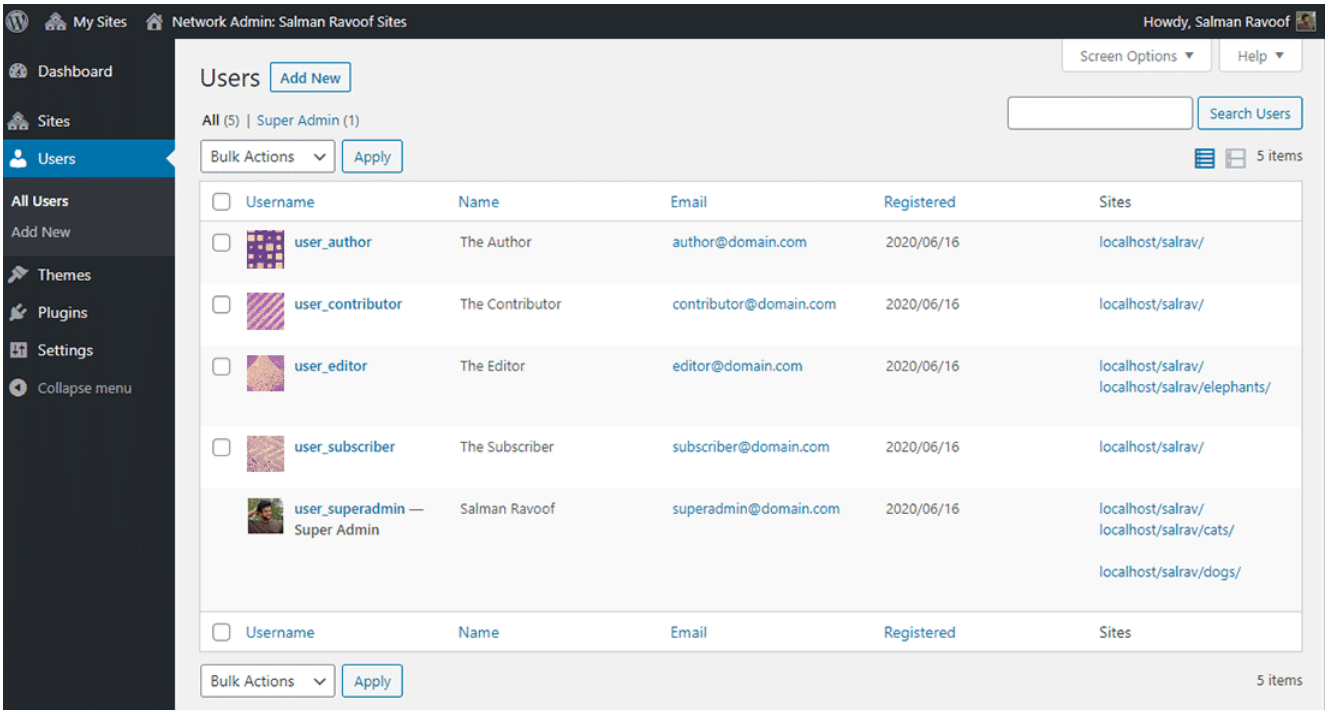
Von hier aus kannst du auch Informationen über Webseiten, Benutzer, Themes und allgemeine Netzwerkeinstellungen abrufen. Die erste Webseite, die du erstellst, ist die primäre Webseite im Netzwerk. Das Netzwerk erbt alle seine Einstellungen von den Optionen der ersten Webseite.



Neue Webseiten zum WordPress Multisite-Netzwerk hinzufügen

Wenn du auf den Link oder die Schaltfläche „Neue Webseite hinzufügen“ klickst, gelangst du zum obigen Bildschirm, wo du eine neue Webseite zu deinem Multisite-Netzwerk hinzufügen kannst. Wenn du niemanden als Admin der neuen Webseite im Sinn hast, kannst du dich auch selbst als Admin bestimmen.

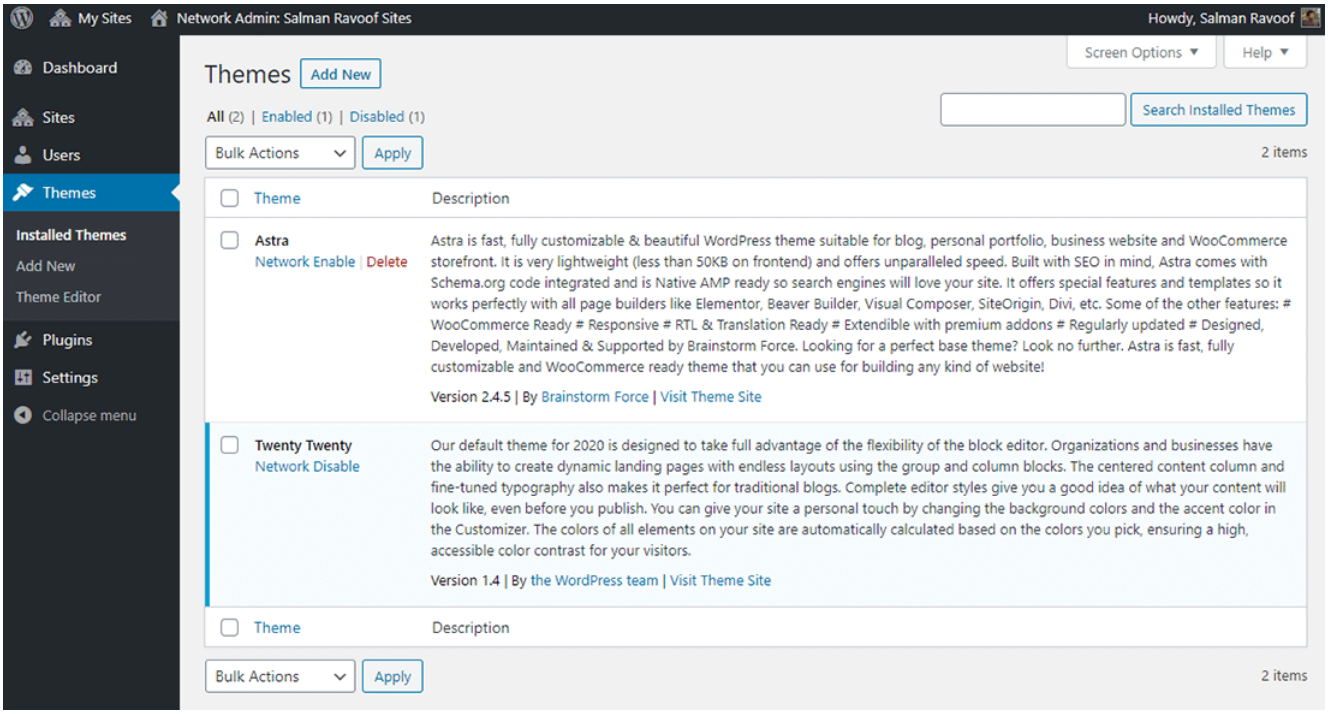
3. Benutzer



Das ‚Benutzer'-Panel im Netzwerk-Admin-Dashboard.

Der Benutzer-Bildschirm im Netzwerk-Admin-Dashboard ermöglicht es dir, Benutzer zu verwalten und neue Benutzer zu deinem Multisite-Netzwerk hinzuzufügen. Nur der Super Admin kann Benutzer zum Netzwerk hinzufügen, aber ein Super Admin kann die Netzwerkeinstellungen modifizieren, um es den Seiten-Administratoren zu ermöglichen, neue Benutzer nur zu ihren eigenen Webseiten hinzuzufügen.

4. Themes

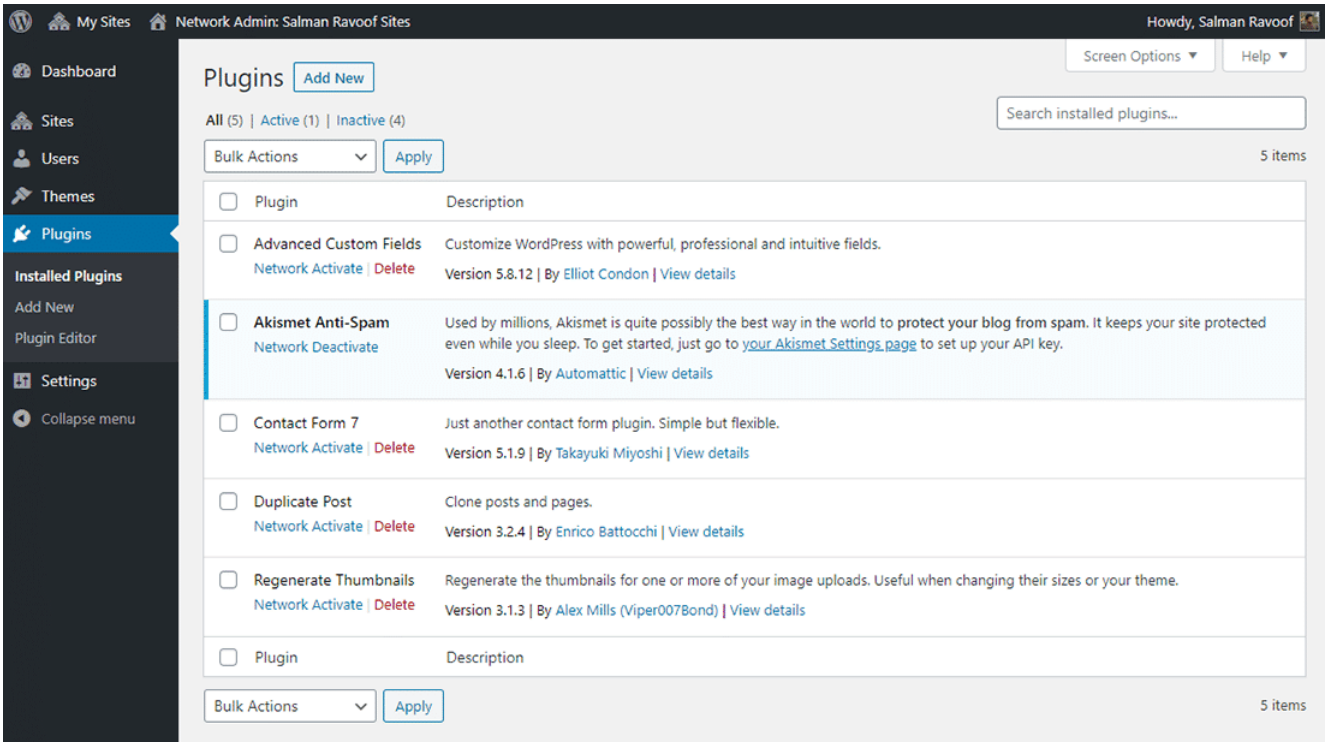


Das ‚Themes'-Panel im Network Admin Dashboard
Der Themes-Bildschirm ermöglicht es dir, Themes zu verwalten, die für die Administratoren der Webseite zugänglich sind. Es erlaubt dir nicht, Themes zu aktivieren oder zu deaktivieren, die jede Webseite benutzt, sondern nur die Themes einzustellen, die jede Webseite benutzen kann.

Wenn du ein Theme deaktivierst, das überall im Netzwerk verwendet wird, bleibt es auf dieser Webseite aktiv, auch wenn du es deaktiviert hast. Wenn die Webseiten jedoch ein anderes Theme verwenden, wird das deaktivierte Theme nicht im Themes-Panel der Netzwerkseite angezeigt.

In Kinsta's WordPress Multisite Artikel erfährst du, wie du mit Themes und Plugins in deinem Netzwerk arbeiten kannst. Du kannst auch den Theme-Editor benutzen, um deine Theme-Dateien im Dashboard selbst zu bearbeiten.

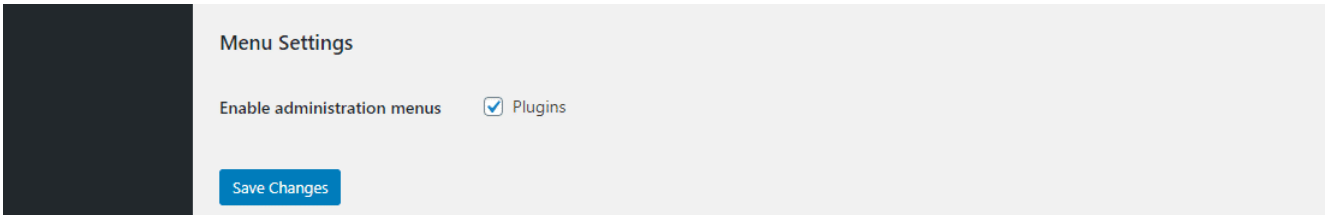
5. Plugins



Das ‚Plugins'-Panel im Netzwerk-Admin-Dashboard.
Der Plugins-Bildschirm erlaubt es Benutzern, Plugins im Netzwerk hinzuzufügen oder zu löschen. Einmal hinzugefügt, kannst du die Plugins über das Dashboard einer Webseite

aktivieren. Du kannst die Plugins auch von hier aus im **Netzwerk aktivieren**, um die Verwendung des Plugins auf allen Webseiten im Netzwerk zu erzwingen.

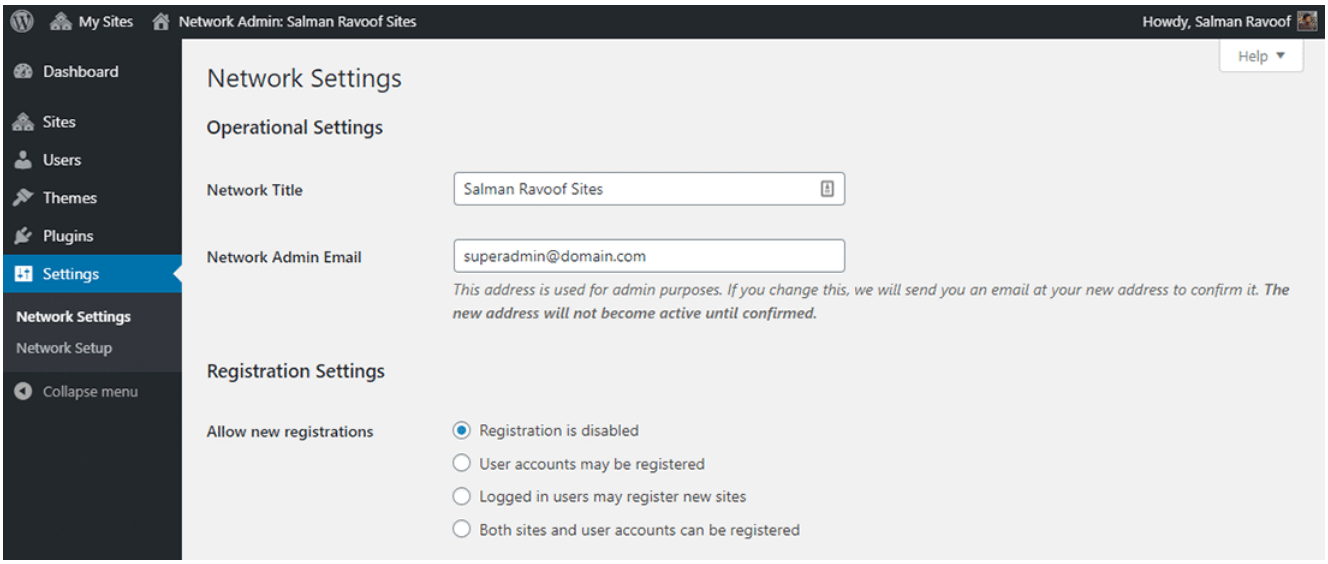
Standardmäßig hat der Administrator einer Webseite keinen Zugriff auf das Plugin-Menü in seinem Dashboard. Ein Super Admin kann dies für sie aktivieren, indem er die Netzwerkeinstellungen modifiziert.



Plugin-Verwaltung für alle Administratoren der Unterseite aktivieren

Hinweis: Nicht alle WordPress-Plugins unterstützen Multisite-Netzwerke. Du musst die Dokumentation des Plugins lesen, um zu bestätigen, ob sie in Multisite-Netzwerken funktionieren.

6. Einstellungen



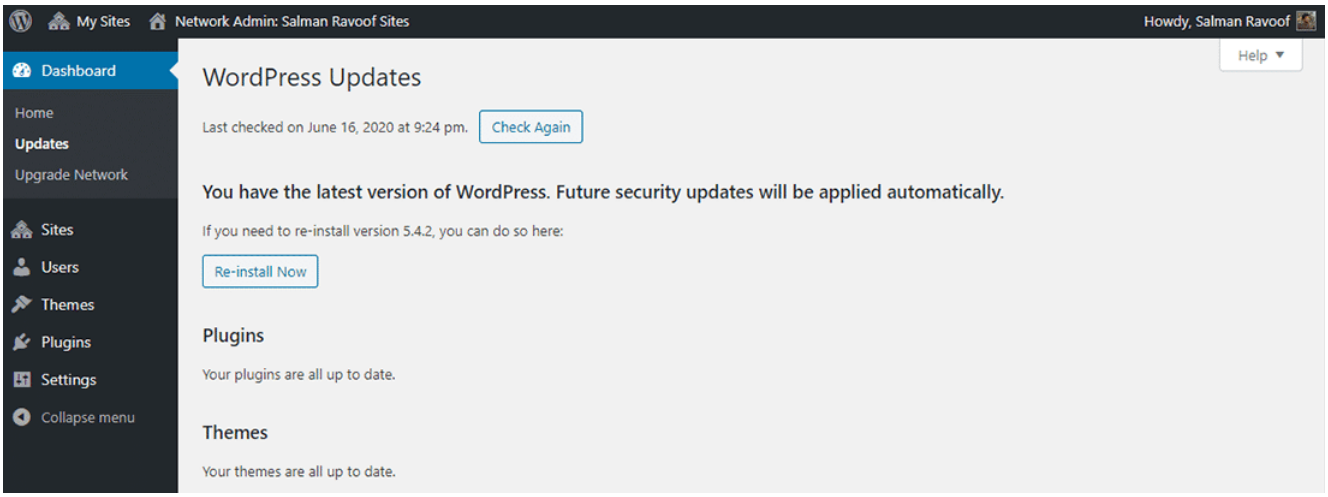
Das ‚Netzwerk-Einstellungen'-Panel im Netzwerk-Admin-Dashboard.

Du kannst die Einstellungen für das gesamte Netzwerk auf dem Bildschirm Netzwerkeinstellungen vornehmen und ändern. Die Standardeinstellungen des Netzwerks basieren auf der ersten Webseite, die du beim Einrichten des Netzwerks erstellt hast. Einige Netzwerkeinstellungen, die du hier ändern kannst, sind

- Operative Einstellungen
- Einstellungen zur Registrierung
- Neue Webseite Einstellungen
- Einstellungen hochladen
- Spracheinstellungen
- Menü Einstellungen

Hier kannst du auch auf die Netzwerk-Setup-Informationen zugreifen, die du bei der Erstellung des Netzwerks verwendet hast. Du kannst den Netzwerk Admin Einstellungsbildschirm im WordPress Codex aufrufen, um einen detaillierten Überblick über alle verfügbaren Einstellungsmöglichkeiten zu erhalten.

7. Updates



The ‚Updates' panel in Network Admin dashboard

Du kannst den Aktualisierungsprozess sowohl für das Netzwerk als auch für die einzelnen Webseiten über den Aktualisierungsbildschirm steuern. Der **Aktualisierungsbildschirm** zeigt dir alle verfügbaren Updates für den WordPress Core, Themes und Plugins an. Sobald du die neueste Version von WordPress installiert hast, kannst du sie auf alle Webseiten im Netzwerk über den Bildschirm „Netzwerk aktualisieren" anwenden.



The ‚Upgrade Network' panel in Network Admin dashboard

Hinweis: Bei einer WordPress Single-Site-Installation ist der Administrator im Wesentlichen ein Super Admin, da er Zugriff auf alle Admin-Fähigkeiten hat.

Du kannst Benutzerrollen anpassen und auch deine eigenen Rollen erstellen, indem du die vordefinierten Fähigkeiten von WordPress nutzt.

Vorteile der Benutzerrollen und -fähigkeiten

Das System der Rollen und Fähigkeiten ist das Rückgrat der Benutzerverwaltung in WordPress. Hier sind einige seiner vielen Vorteile:

- Benutzerrollen helfen dir dabei, alle Benutzer auf deiner Webseite effizienter zu verwalten. Selbst wenn du Dutzende von Benutzern auf deiner Webseite hast, die aus verschiedenen Teilen der Welt arbeiten, kannst du sie einfach überwachen, indem du jedem von ihnen die richtigen Rollen zuweist.
- Indem du die Benutzer auf bestimmte Fähigkeiten beschränkst, hilft es dir, deine Webseite sicherer zu machen. Beispielsweise können Autoren nicht die Beiträge anderer löschen, Redakteure können keine Themes ändern oder Plugins installieren und Abonnenten können nur auf ihre eigenen Profile zugreifen.
- WordPress-Plugins können überprüfen, ob ein Benutzer

bestimmte Fähigkeiten hat, und darauf basierend eine bestimmte Aktion ausführen. Die Funktion current_user_can() WordPress hilft bei dieser Überprüfung. Zum Beispiel kann ein Sicherheits-Plugin sein Optionen-Panel nur dem Admin zeigen, aber trotzdem Sicherheitswarnungen für alle Benutzer anzeigen.

- Du kannst Benutzerrollen bearbeiten, um einige deiner Rollenverantwortlichkeiten an andere Benutzer zu delegieren, damit du mehr Zeit hast. Angenommen, deine Webseite zieht eine Menge Kommentare an. In solchen Fällen kannst du einem vertrauenswürdigen Autor erlauben, moderierende Kommentare aufzunehmen. Als Admin behältst du immer noch die letzte Macht, aber du kannst einige deiner Verantwortlichkeiten je nach Bedarf teilen.
- Du kannst Fähigkeitsprüfungen verwenden, um private Beiträge und Seiten anzuzeigen, die nur von bestimmten Benutzerrollen eingesehen werden können. Dies bildet die Grundlage der Webseiten für Mitglieder.
- Du kannst Frontend-Elemente auf deiner Webseite ein- oder ausblenden (z.B. Menüpunkte, Widgets), abhängig von der Benutzerrolle.
- Du kannst benutzerdefinierte Beitragstypen mit benutzerdefinierten Fähigkeiten erstellen und diese Fähigkeiten für jede Benutzerrolle gewähren oder verweigern. Auf ähnliche Weise kannst du auch benutzerdefinierte Fähigkeiten definieren, um nur bestimmten Rollen den Zugriff auf deine Plugin- oder Theme-Einstellungen zu erlauben.

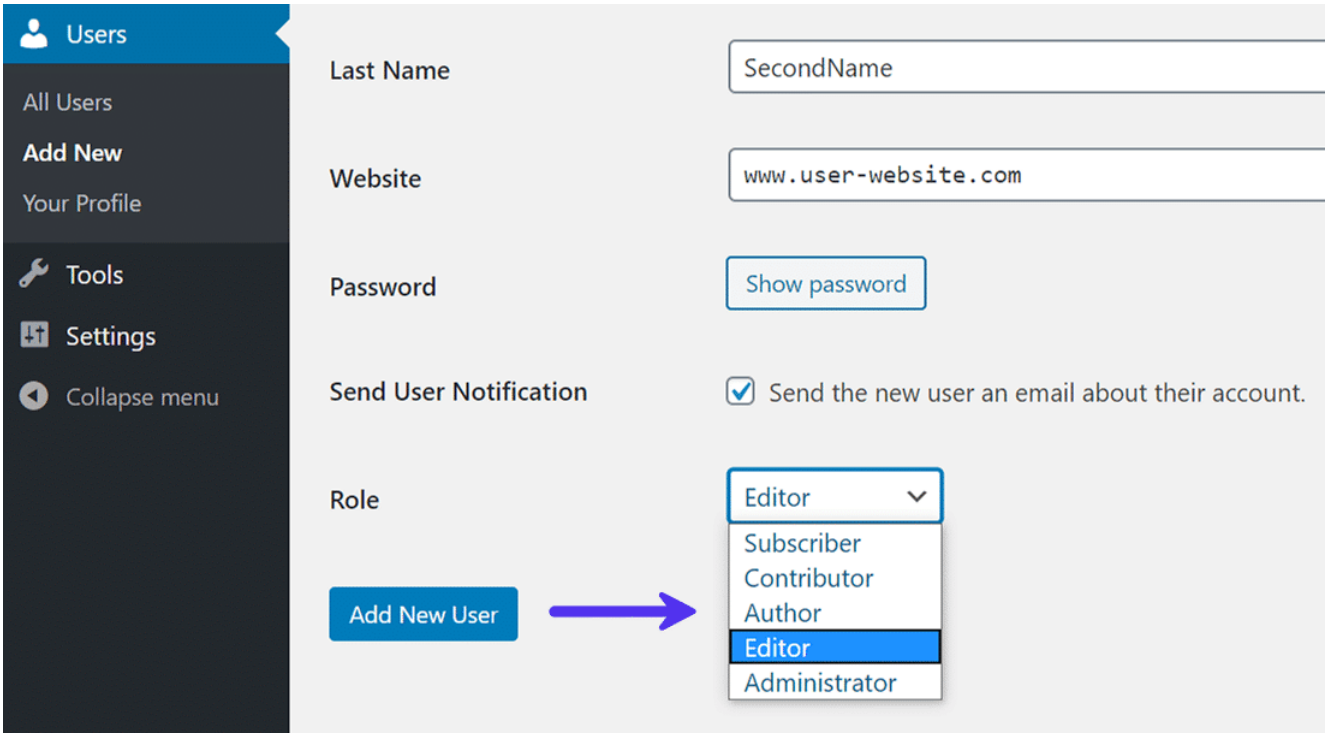
Wie man WordPress-Benutzerrollen effektiv verwaltet

Es ist wichtig, alle Benutzerrollen und Fähigkeiten zu kennen, aber du musst auch verstehen, wie du sie auf deinen Webseiten effizient verwalten kannst. Auch wenn keine zwei WordPress

Webseiten exakt gleich sind, gibt es einige Grundregeln, die du befolgen kannst, um die Benutzerrollen und Fähigkeiten von WordPress optimal zu nutzen.

Jedem Benutzer minimalen Zugang gewähren

Weise jedem Benutzer auf deiner Webseite nur die Zugangsstufe zu, die er benötigt. Es ist immer besser, weniger Berechtigungen zu erteilen als zu viele. Die Sicherung der WordPress-Benutzerrollen ist entscheidend für die Sicherheit deiner Webseite und ihrer Inhalte.



Weise jedem Benutzer sorgfältig Benutzerrollen zu

Begrenzt die Anzahl der Administratoren und Redakteure

Als allgemeine Regel gilt, dass jede Webseite nur einen Administrator haben sollte und dieser sollte nur Kern-Änderungen an der Webseite vornehmen. WordPress empfiehlt, sich an den „Grundsatz der geringsten Privilegien“ zu halten, der besagt, dass man einem Benutzer nur die Privilegien geben sollte, die für die Ausführung seiner gewünschten Arbeit notwendig sind.

Zum Beispiel ist es besser, einen Redakteur zu benutzen, um die Inhalte der Webseite zu verwalten, als einen Administrator zu benutzen. Wenn du mehr als einen Redakteur auf deiner Webseite hast, dann stelle sicher, dass du ihnen mit ihren umfangreichen Fähigkeiten vertrauen kannst.

Weise den Verfassern von Inhalten die Autorenrolle zu, denen du vertrauen kannst, da sie ihre eigenen Beiträge veröffentlichen und löschen können. Die Autorenrolle eignet sich besser für Autoren neuer Inhalte und Gastbeiträge.

Benutzerrollen je nach Bedarf anpassen

Die standardmäßigen WordPress-Benutzerrollen sind nützlich, aber sie sind vielleicht nicht für jeden Anwendungsfall geeignet. Zum Beispiel, um deinen Autoren die Möglichkeit zu geben, Kommentare zu moderieren.

Dankenswerterweise gibt uns WordPress die Möglichkeit, Benutzerrollen anzupassen oder neue Rollen zu erstellen, je nachdem, was wir benötigen. Du kannst dies manuell durch Programmcode oder mit Hilfe von WordPress Benutzerrollen-Plugins tun. Wir werden beide Methoden in diesem Artikel behandeln.

Benutzer im WordPress Multisite-Netzwerk verwalten

WordPress Multisite enthält einzigartige Einstellungen für die Benutzerverwaltung. Einige davon sind einfach zu verstehen, andere wiederum nicht.

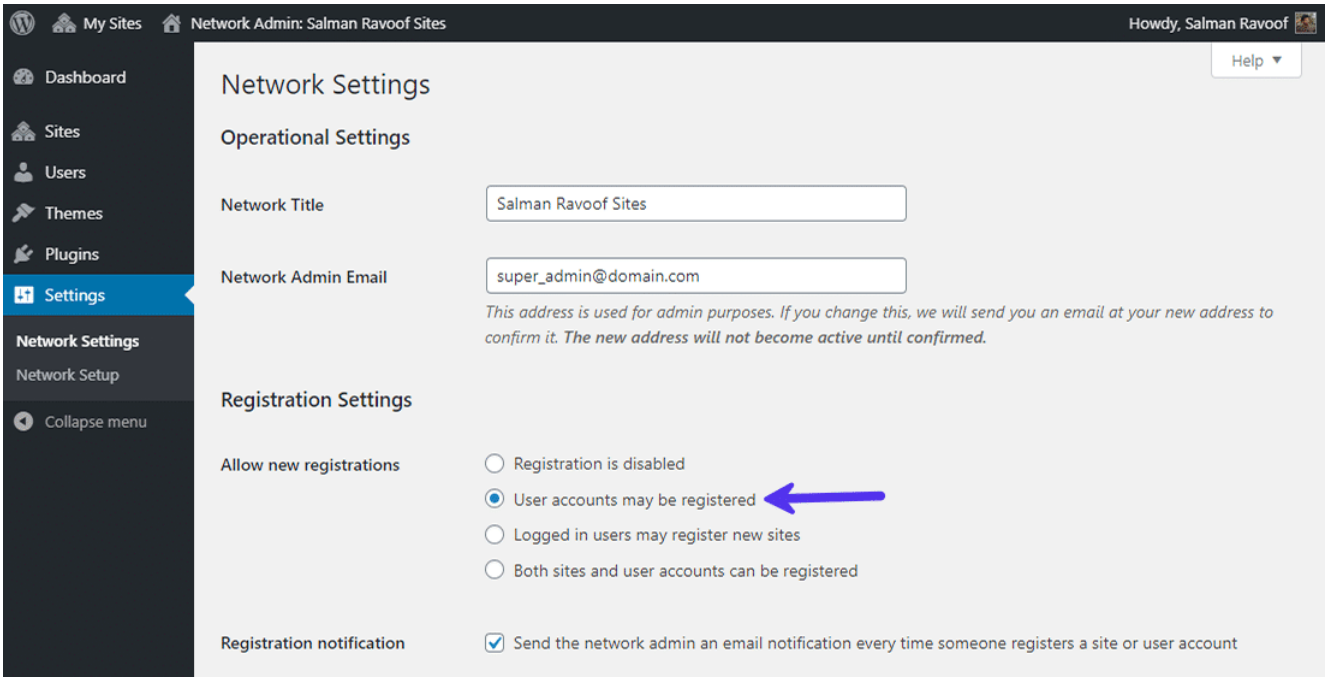
Lass sie uns alle im Detail erforschen.

Multisite-Netzwerk-

Registrierungseinstellungen

Sofort nach dem Auspacken kann nur ein Super Admin neue Benutzer und Webseiten im Netzwerk erstellen. Er kann Benutzern jedoch erlauben, Konten im Netzwerk als Abonnenten der Unterseiten zu registrieren.

Um dies zu aktivieren, gehe zu **Netzwerk-Admin > Netzwerk-Einstellungen > Registrierungseinstellungen > Neue Registrierungen erlauben** und aktiviere die Option „Benutzerkonten können registriert werden“.



Erlaube Benutzern, einen Account in deinem Netzwerk zu registrieren
Hier kannst du auch eingeloggten Benutzern erlauben, neue Webseiten in deinem Netzwerk zu erstellen. Du kannst diese Option ankreuzen, wenn du die Möglichkeit, Webseiten nur für von dir angemeldete Benutzer zu erstellen, einschränken möchtest.

Mit der letzten Option können Benutzer sowohl ein Konto registrieren als auch eine Webseite in deinem Netzwerk erstellen. Benutzer, die eine Webseite in deinem Netzwerk erstellen, erhalten die Rolle des Administrators für ihre Unterwebsite.

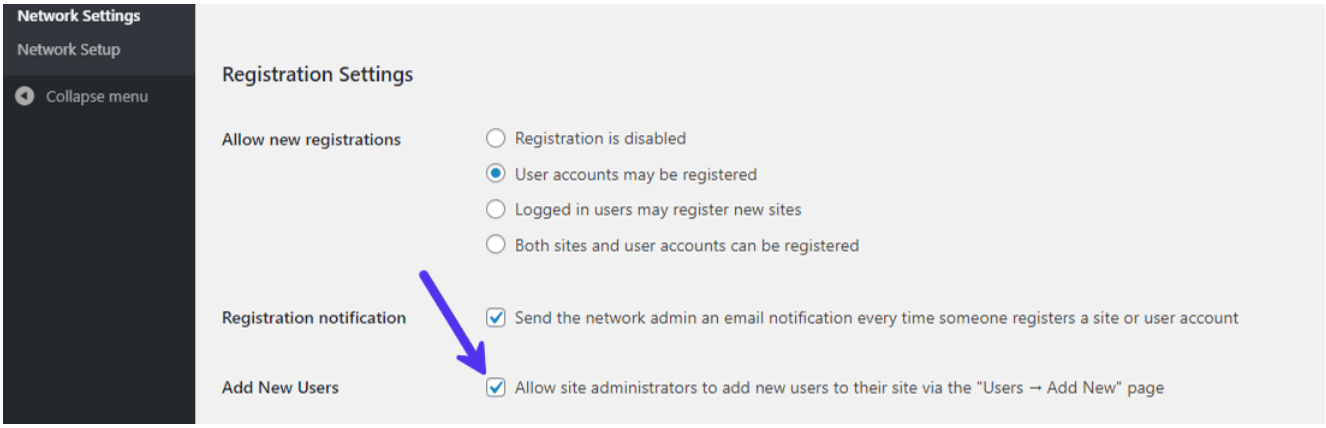
Ein Benutzerkonto für den Zugriff auf das gesamte Netzwerk

Wenn du ein Benutzerkonto in deinem Netzwerk erstellst, oder wenn ein Benutzer ein Konto auf einer der Webseiten deines Netzwerks registriert, kann er nach dem Einloggen zu jeder Webseite innerhalb des Netzwerks navigieren. Stell dir das als ein soziales Netzwerk wie Facebook oder Reddit vor, in dem du ein Konto erstellen kannst und auf alle Gruppen oder Unterreddits mit demselben Benutzerprofil zugreifen kannst.

Dies ist einer der größten Vorteile der Verwendung von WordPress Multisite. Es erlaubt deinen Benutzern, auf alle deine Webseiten zuzugreifen, indem du nur ein Konto registrierst.

Gewähren von Extra-Privilegien für die Administratoren der Webseite

Du kannst den Administratoren der Webseite erlauben, Benutzer zu ihren eigenen Webseiten hinzuzufügen, indem du die Option **Neue Benutzer hinzufügen** anklickst.



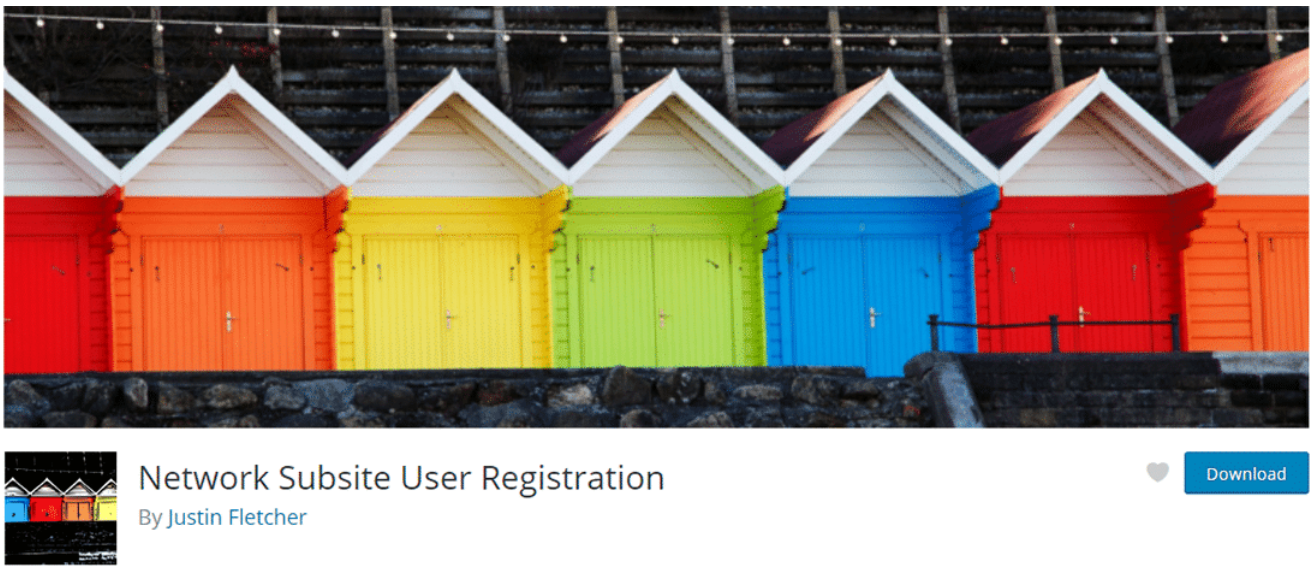
Erlaubt es den Administratoren der Webseite, neue Benutzer zu ihrer Unterwebsite hinzuzufügen

Wie bereits erwähnt, kannst du Administratoren der Webseite Zugang zur Verwaltung von Plugins auf ihren Unterseiten gewähren, indem du zu **Netzwerkeinstellungen > Menüeinstellungen** gehst und die Option **Administrationsmenüs**

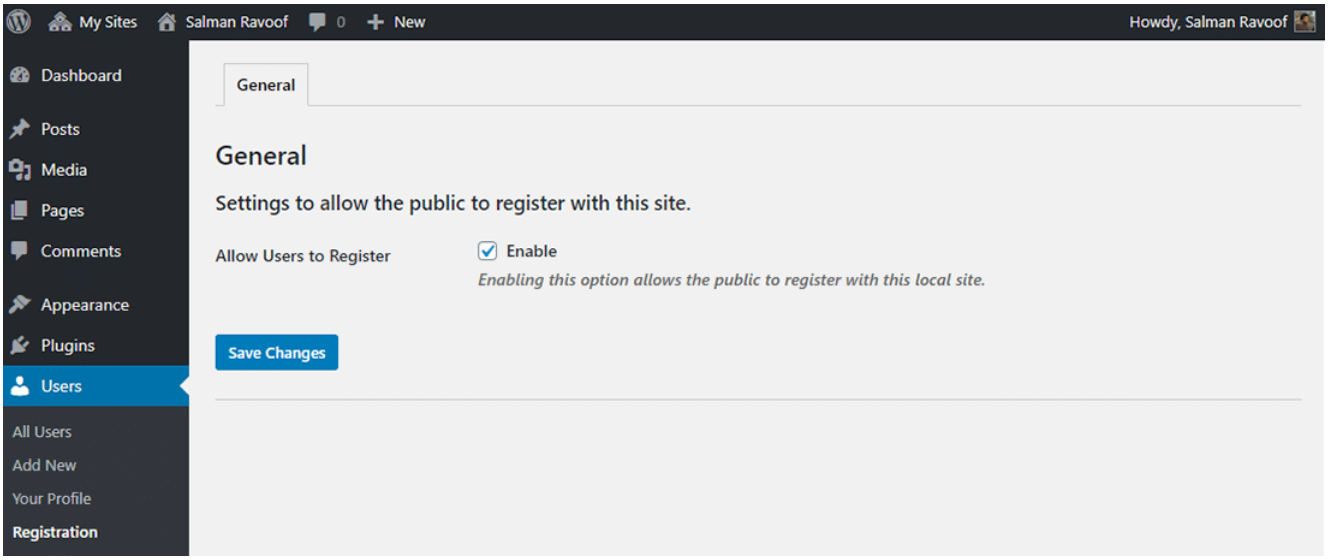
aktivieren > Plugins aktivierst.

Subsite Level Benutzer Registrierung

WordPress Multisite-Installationen erlauben standardmäßig nur Benutzerregistrierungen für das gesamte Netzwerk. Es gibt keine Option, Benutzerregistrierungen nur für eine Unterseite zu erlauben. Du kannst dieses Verhalten ändern, indem du das Plugin Network Subsite User Registration verwendest.



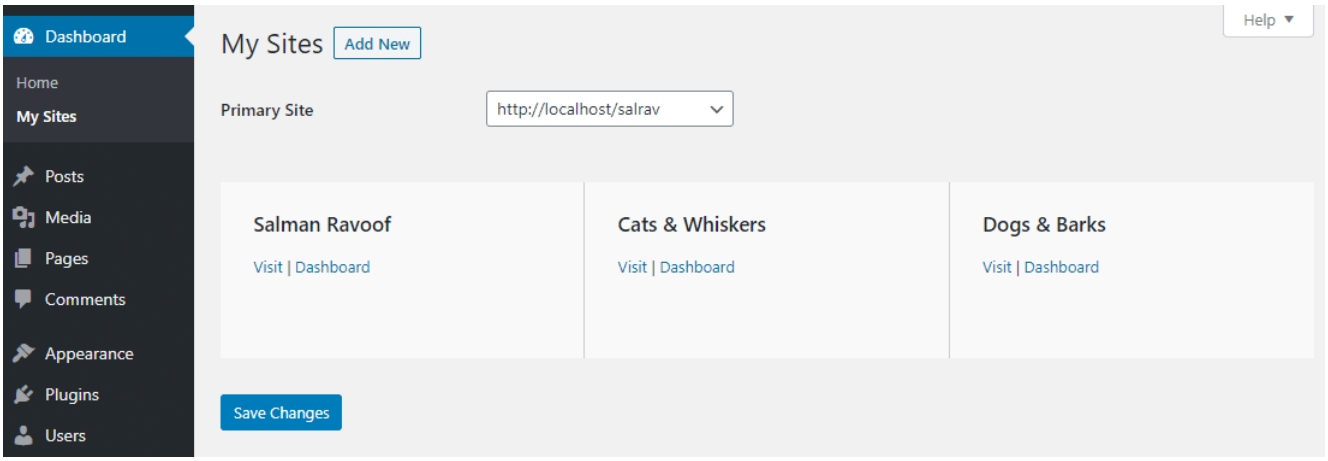
Das ‚Network Subsite User Registration'-Plugin
Dieses Plugin erlaubt es den Administratoren der Webseite, lokale Benutzerregistrierungen zu ermöglichen, wobei der Zugang nur auf ihre Webseite beschränkt ist. Neue Benutzer übernehmen standardmäßig die Rolle des Abonnenten, aber du kannst dies ändern, indem du die Einstellungen des Plugins anpasst.



Jedem ermöglichen, ein Konto nur auf deinen Unterseiten zu registrieren

Denselben Benutzer mehreren Unterseiten zuweisen

Du kannst den gleichen Benutzer mehreren Webseiten in deinem Netzwerk mit einzigartigen Rollen zuweisen. Wenn der Benutzer sich in das Dashboard seiner Webseite einloggt, kann er über den Bildschirm **Meine Webseiten** auf das Dashboard für alle seine Webseiten zugreifen.

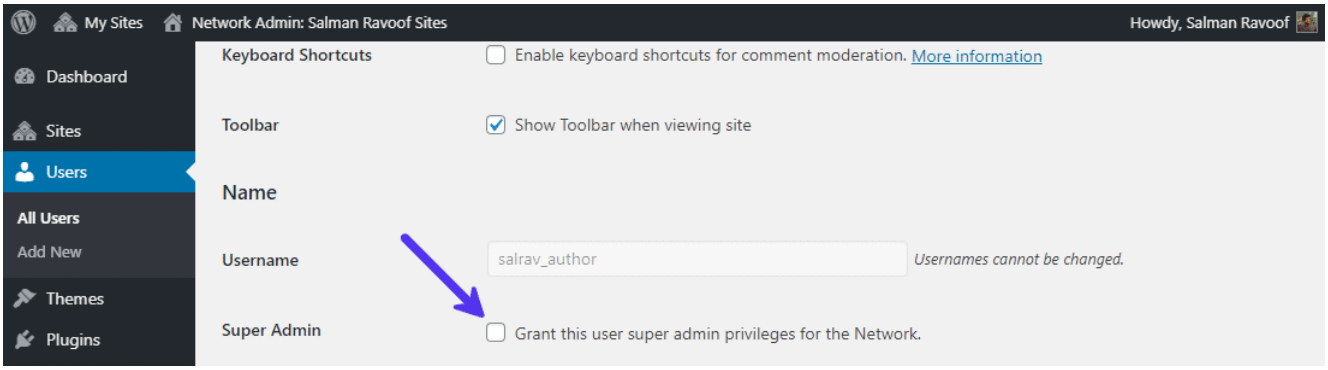


Du kannst einen Benutzer mehreren Webseiten in einem WordPress Multisite-Netzwerk zuweisen

Anderen Benutzern Super-Admin-Privilegien

gewähren

Ein Super Admin kann seine Privilegien auch mit anderen Benutzern teilen. Du solltest diese Option mit Vorsicht aktivieren und sie nur Benutzern zuweisen, denen du vertrauen kannst.



Granting other users the Super Admin privileges for the network

Wenn du alle Einstellungen der Benutzerverwaltung in WordPress Multisite verstehst, kannst du dein Netzwerk besser verwalten. Um andere hilfreiche Plugins für WordPress Multisite zu finden, kannst du sie im WordPress Repo oder in Kinsta's empfohlenem WordPress Multisite Plugins Artikel entdecken.

Wie man existierende WordPress-Benutzerrollen anpasst

Du kannst deinen bestehenden Benutzerrollen Fähigkeiten hinzufügen, um ihre Zugriffsstufe zu erhöhen. Zum Beispiel kannst du Redakteuren die Möglichkeit geben, Plugins zu verwalten. Oder vielleicht möchtest du, dass Contributors Kommentare zu ihren eigenen Beiträgen moderieren. Lasst uns lernen, wie man das macht.

Hinweis: Wenn es nicht dein Ding ist, dich mit Code zu beschäftigen, kannst du die manuelle Methode überspringen und direkt zum Abschnitt Plugins für Benutzerrollen und Fähigkeiten weiter unten gehen. Oder einfach einen WordPress-Entwickler einstellen.

Wie man Fähigkeiten zu einer Benutzerrolle hinzufügt

Du kannst eine Fähigkeit zu einer Benutzerrolle oder einem bestimmten Benutzer hinzufügen, indem du die add_cap() WordPress Funktion benutzt. Ich werde ein benutzerdefiniertes Plugin namens **Benutzerrolle anpassen** benutzen, um zu zeigen, wie man diese Funktion benutzt, um der Redakteur-Rolle die Macht zu geben, Plugins zu verwalten.

```
<?php

/*
Plugin Name:  Customize User Role
Version:  1.0
Description:  Demonstrating how to customize WordPress User Roles.
Author:  Salman Ravoof
Author URI:  https://www.salmanravoof.com/
License: GPLv2 or later
License URI: https://www.gnu.org/licenses/gpl-2.0.html
Text Domain:  customize-user-role
*/
```

WordPress empfiehlt, diese Funktion bei der Plugin- oder Theme-Aktivierung auszuführen, da die hinzugefügten Einstellungen in der Datenbank in der Tabelle **wp_options** unter **wp_user_roles** gespeichert werden. Es ist ineffizient, diese Funktion jedes Mal auszuführen, wenn du eine Seite lädst, da die Datenbanktabellen bei jedem Seitenaufruf überschrieben werden.

Da ich ein Plugin benutze, werde ich die register_activation_hook()-Funktion benutzen, um mich in die Aktion einzuloggen, die ausgeführt wird, wenn du ein Plugin aktivierst. Es gibt viele Möglichkeiten, dies zu tun, aber ich benutze eine robuste klassenbasierte Implementierung, um sicherzustellen, dass es keine Konflikte gibt.

```
// this code runs only during plugin activation and never
```

```
again
function sal_customize_user_role() {
    require_once plugin_dir_path( __FILE__ ).'includes/class-
sal-customize-user-role.php';
    Sal_Customize_User_Role::activate();
}
register_activation_hook( __FILE__, 'sal_customize_user_role'
);
```

Der oben programmierte Code läuft nur einmal während der Aktivierung des Plugins. Die gehookte Funktion **sal_customize_user_role** referenziert eine benutzerdefinierte Klasse namens **Sal_Customize_User_Role**.

Ich habe diese Klasse in einer separaten Datei namens **class-sal-customize-user-role.php** definiert und sie in meinem Plugin-Stammordner in einem Unterordner namens **includes** abgelegt, aber ihr könnt sie nennen, wie ihr wollt.

```
<?php

class Sal_Customize_User_Role {
    public static function activate() {
        // get the Editor role's object from WP_Role class
        $editor = get_role( 'editor' );

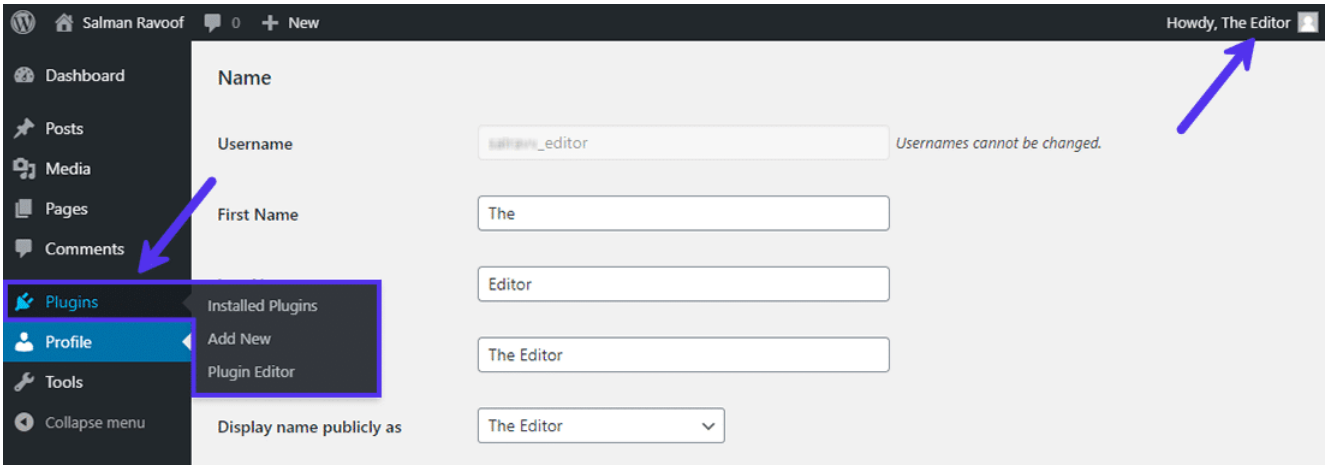
        // a list of plugin-related capabilities to add to the
Editor role
        $caps = array(
                  'install_plugins',
                  'activate_plugins',
                  'edit_plugins',
                  'delete_plugins'
        );

        // add all the capabilities by looping through them
        foreach ( $caps as $cap ) {
            $editor->add_cap( $cap );
        }
    }
}
```

Hier ist eine detaillierte Erklärung des oben programmierten Codes:

- Beginne damit, die Klasse und ihre Funktion zu definieren, die du in der Haupt-Plugin-Datei referenziert hast.
- Die get_role(‚editor‘) Funktion holt das Editor-Rollenobjekt aus der **WP_Role** Core Class und weist es der Variablen **$editor** zu.
- Das Verwalten von Plugins erfordert vier Fähigkeiten: **install_plugins**, **activate_plugins**, **edit_plugins** und **delete_plugins**. Aber die **add_cap()**-Funktion akzeptiert nur einen Parameter. Daher müssen wir alle Fähigkeiten in ein Array aufnehmen. Definiere das **$caps**-Array, das all diese Fähigkeiten enthält. Wenn du nur eine einzige Fähigkeit hinzufügst, dann musst du kein Array definieren.
- Die **add_cap($cap)** Funktion add_cap($cap) fügt alle Fähigkeiten, die im **$caps** Array definiert sind, hinzu, indem sie sie alle mit der foreach () PHP Funktion durchläuft.

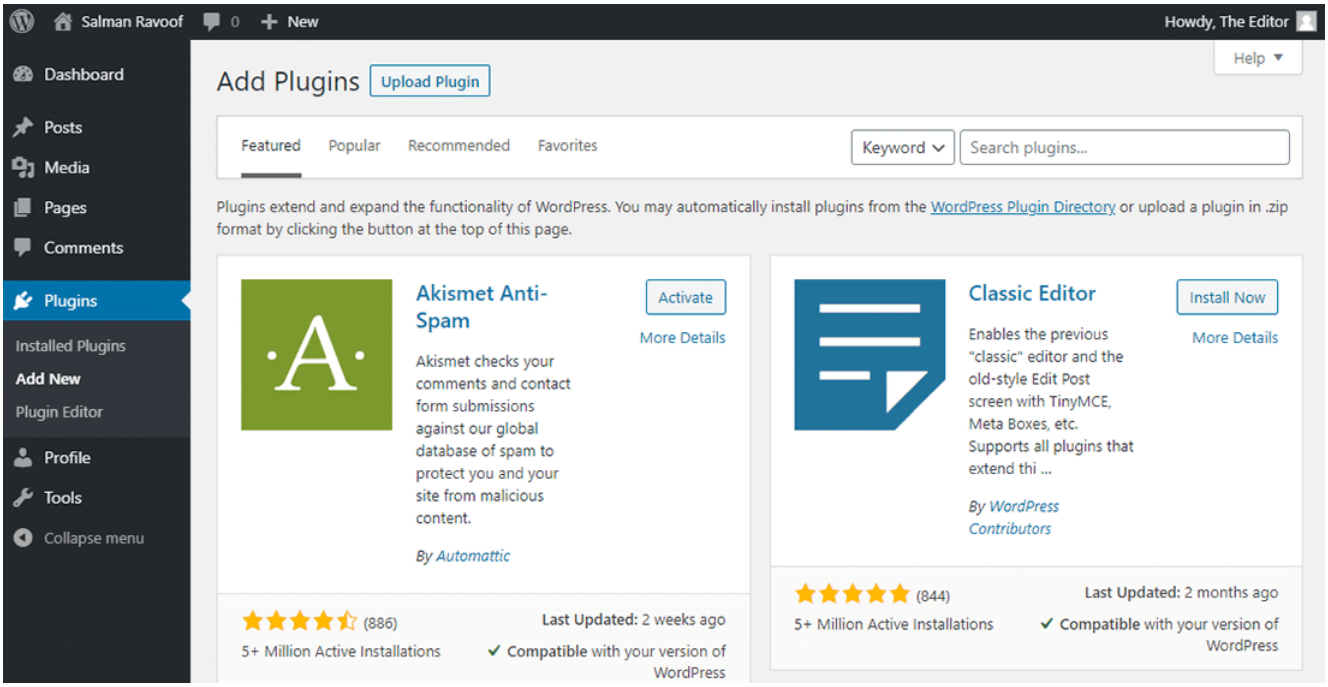
Speichere alle deine Plugin-Dateien und aktiviere das Plugin dann von deinem Administrator-Dashboard aus. Nun lass uns in das Redakteur-Dashboard einloggen, um die Änderungen zu sehen.



Redakteure können jetzt Plugins von ihrem Dashboard aus verwalten

Nachdem sie die Plugin-bezogenen Fähigkeiten zu ihrer

Benutzerrolle hinzugefügt haben, können Redakteure das Plugin-Menü in ihrem Admin-Menü sehen.



Der ‚Plugins hinzufügen'-Bildschirm im Dashboard des Redakteurs

Du kannst die Fähigkeiten, die jeder Benutzerrolle zugeordnet sind, überprüfen, indem du dir den Schlüsselwert **wp_user_roles** ansiehst, der in der **wp_options**-Tabelle der Datenbank deiner WordPress Webseite gespeichert ist.

Hier sind die Fähigkeiten, die der Redakteur-Rolle zugeordnet sind:

```
'editor' =>
  array (
    'name' => 'Editor',
    'capabilities' =>
    array (
      'moderate_comments' => true,
      'manage_categories' => true,
      // [...lines cut off for brevity...]
      'install_plugins' => true,
      'activate_plugins' => true,
      'edit_plugins' => true,
    ),
  ),
```

Beachte die letzten drei Zeilen, die den Redakteuren die Möglichkeit geben, Plugins zu verwalten.

Wenn du diese Fähigkeiten entfernen möchtest, kannst du dich in die register_deactivation_hook()-Funktion einklinken und die **remove_cap()**-Funktion verwenden, um die Fähigkeiten bei der Plugin-Deaktivierung zu entfernen, genau wie wir diese Fähigkeiten bei der Plugin-Aktivierung hinzugefügt haben.

Nachdem du nun gelernt hast, wie man einer Benutzerrolle Fähigkeiten hinzufügt, ist es an der Zeit zu lernen, wie man Fähigkeiten aus einer Benutzerrolle entfernt.

Hinweis: Du kannst dich auch in die after_switch_theme-Aktion einklinken, um diesen Code während der Aktivierung des Themes (und/oder des Child-Themes) zu feuern. Hier musst du den Code in die **functions.php**-Datei deines Themes oder des Child-Themes (empfohlen) einfügen.

Wie man Fähigkeiten aus einer Benutzerrolle entfernt

Manchmal möchtest du vielleicht eine Fähigkeit aus einer Benutzerrolle entfernen. Du kannst die remove_cap()-Funktion ausführen, um eine Fähigkeit aus einer Rolle oder einem bestimmten Benutzer zu entfernen. Zum Beispiel ist es eine ausgezeichnete Idee, die **delete_published_posts**-Funktion aus der Benutzerrolle Autor zu entfernen.

Bringen wir es hinter uns!

Ich werde ein neues benutzerdefiniertes Plugin mit dem Namen **Customize Author Role** erstellen, um anzufangen. Genau wie zuvor werde ich diesen Code nur einmal ausführen, indem ich mich in die Funktion **register_activation_hook()** einklinke.

```
<?php

/*
```

```
Plugin Name:  Customize Author Role
Version:  1.0
Description:  Demonstrating how to customize WordPress Author Role.
Author:  Salman Ravoof
Author URI:  https://www.salmanravoof.com/
License: GPLv2 or later
License URI: https://www.gnu.org/licenses/gpl-2.0.html
Text Domain:  customize-author-role
*/

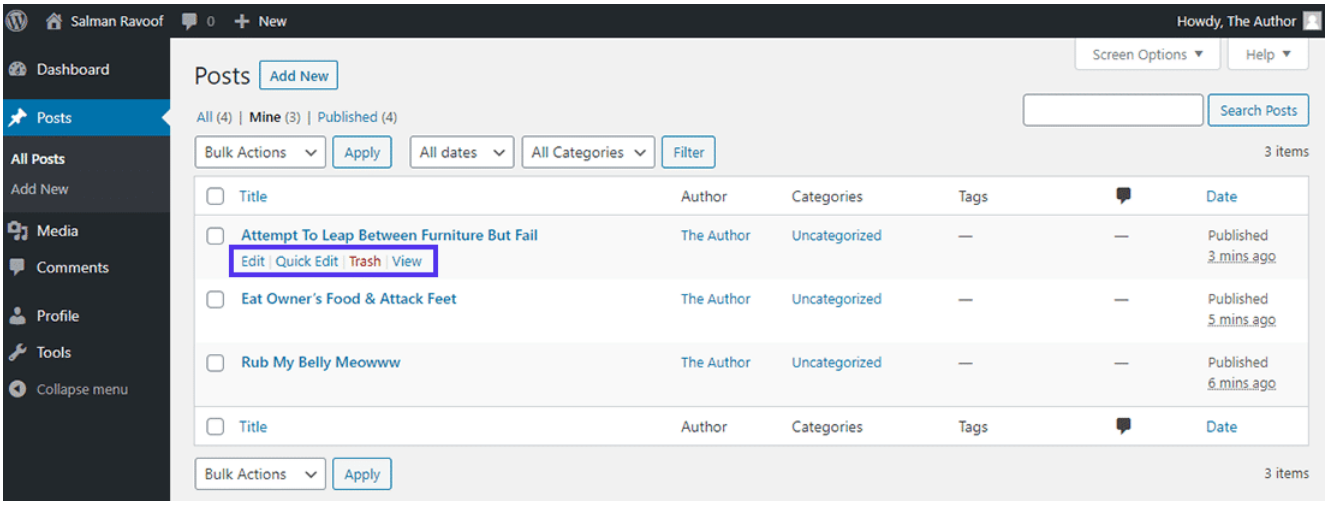
// this code runs only during plugin activation and never again
function sal_customize_author_role() {
    require_once plugin_dir_path( __FILE__ ).'includes/class-sal-customize-author-role.php';
    Sal_Customize_Author_Role::activate();
}
register_activation_hook(                    __FILE__, 'sal_customize_author_role' );
```

Als nächstes werde ich die Klasse **Sal_Customize_Author_Role** in der Datei **class-sal-customize-author-role.php** definieren. Ich habe diese beiden Ressourcen in der oben genannten Haupt-Plugin-Datei referenziert.

```
<?php
class Sal_Customize_Author_Role {
    public static function activate() {
        // get the Editor role's object from WP_Role class
        $author = get_role( 'author' );

         // remove the capability to delete published posts from an Author role
        $author->remove_cap( 'delete_published_posts' );
    }
}
```

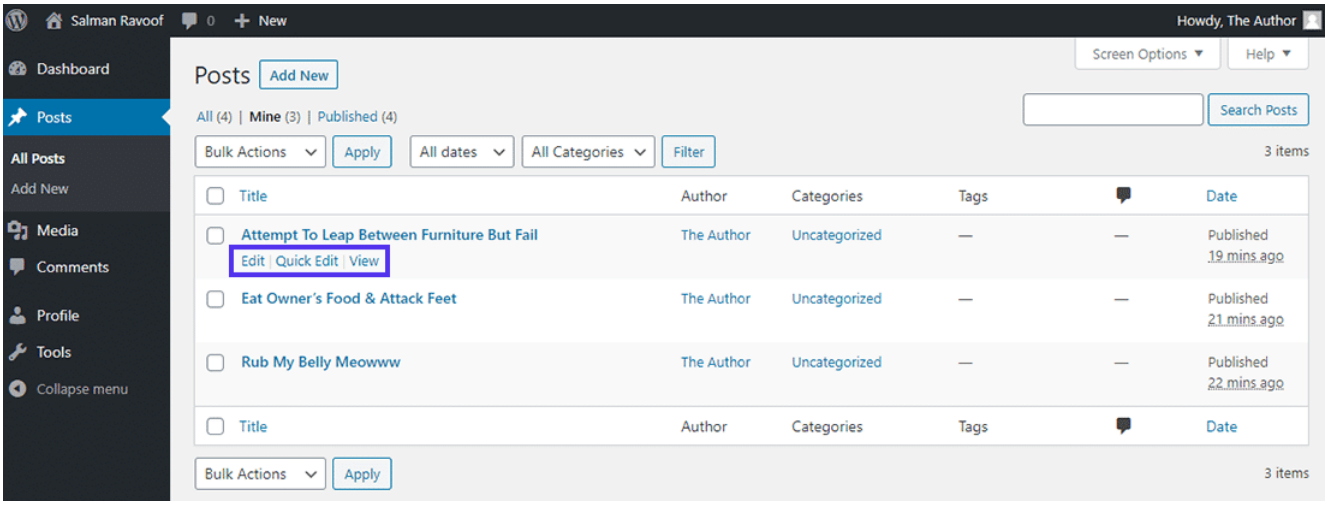
Die **remove_cap('delete_published_posts')** Funktion entfernt die Möglichkeit, veröffentlichte Beiträge aus der Autorenrolle zu löschen.



Autoren dürfen ihre veröffentlichten Beiträge standardmäßig löschen.
Es ist an der Zeit, alle Plugin-Dateien zu speichern und dann das Plugin zu aktivieren. Logge dich jetzt in das Author Dashboard ein und schau dir die Änderungen an.

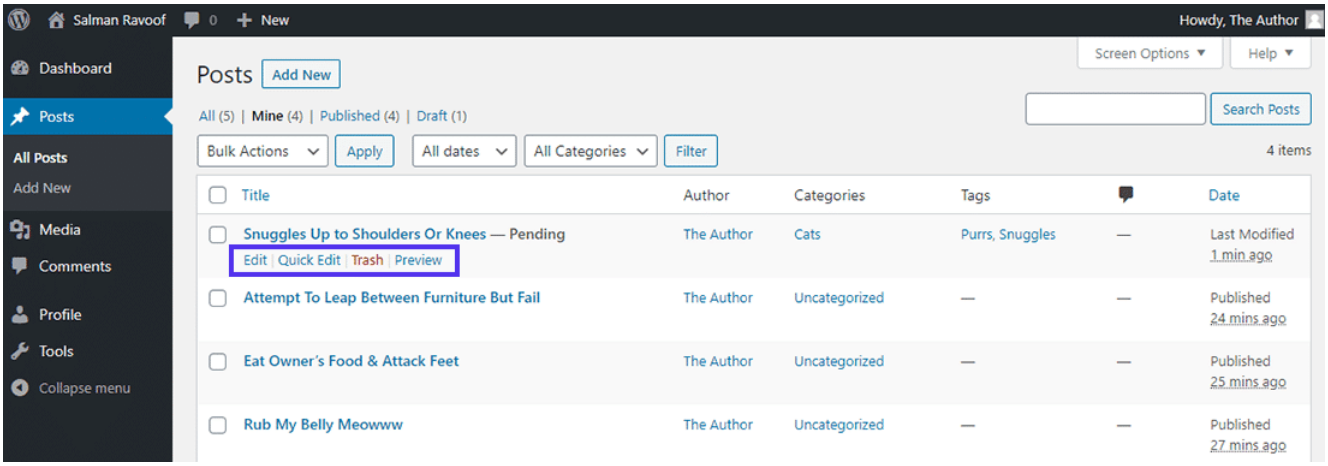
Möchten Sie wissen, wie wir unseren Traffic um über 1000% gesteigert haben?

Gehöre zu den 20 000+ Personen, die wöchentlich Newsletter mit Insider-Tipps rund um WordPress erhalten! Jetzt abonnieren



Autoren können ihre veröffentlichten Beiträge nicht mehr löschen
Die **Papierkorb-Option** ist nicht mehr verfügbar für Beiträge, die von Autoren veröffentlicht wurden. Sie können jedoch immer noch ihre unveröffentlichten Beiträge mit dem Status **Entwurf**

oder **Ausstehend** löschen.



Autoren können immer noch ihre unveröffentlichten Beiträge löschen
Wenn du sogar diese Fähigkeit deaktivieren möchtest, dann musst du auch die **delete_posts**-Fähigkeit aus der Autorenrolle entfernen.

Hinzufügen oder Entfernen von Fähigkeiten für bestimmte Benutzer

Wenn du Fähigkeiten zu einem bestimmten Benutzer hinzufügen möchtest, im Gegensatz zu einer ganzen Benutzerrolle, dann kannst du die Funktion der Klasse WP_User::add_cap() verwenden, um die Fähigkeit hinzuzufügen.

```
// get the user object by their ID
$user = new WP_User( $user_id );

// add the capability to the specific user
$user->add_cap( $cap );
```

Du kannst die get_user_by()-Funktion benutzen, um die ID eines jeden Benutzers abzurufen, indem du seine E-Mail, seinen Login-Benutzernamen oder seinen Slug benutzt.

Auf ähnliche Weise kannst du die Fähigkeiten eines bestimmten Benutzers entfernen, indem du die Klassenfunktion WP_User::remove_cap() verwendest.

```
// get the user object by their ID
$user = new WP_User( $user_id );

// add the capability to the specific user
$user->add_cap( $cap );
```

Wie zuvor, führe diese Funktionen nur bei der Aktivierung von Plugins oder Themes aus, damit dein Code optimiert bleibt.

Hinweis: Sowohl **add_cap()** als auch **remove_cap()** sind Objektmethoden der **WP_Role** Klasse. Du kannst sie nicht direkt in deinem Code aufrufen. Du musst auf sie zugreifen, indem du entweder die **get_role()**-Funktion oder die globale Variable **$wp_roles** verwendest.

Duplizieren einer Benutzer-Rolle

Du kannst eine neue Benutzerrolle erstellen, indem du alle Fähigkeiten einer bestehenden Benutzerrolle klonst. Hier siehst du, wie du das machen kannst:

```
add_role( 'clone', 'Clone', get_role( 'administrator' )->capabilities );
```

Im obigen Beispiel erstelle ich eine neue Rolle namens **Clone** mit den gleichen Fähigkeiten wie ein Administrator. Wenn ich diesen Code bei der Aktivierung des Themes oder des Plugins ausführe, wird sichergestellt, dass die geklonte Rolle nur einmal hinzugefügt wird.

Wie man benutzerdefinierte Benutzerrollen in WordPress erstellt

Die Editiermöglichkeiten der Standardbenutzerrollen sind eine schnelle Möglichkeit, sie anzupassen. Aber wenn du viele Fähigkeiten einer Rolle bearbeiten möchtest, ist es eine gute Idee, eine neue benutzerdefinierte Benutzerrolle zu erstellen.

Auf diese Weise kannst du für jede Rolle auf deiner Webseite genau die Fähigkeiten einstellen, die du willst.

Um eine benutzerdefinierte Benutzerrolle zu erstellen, musst du die add_role()-Funktion verwenden. Sie akzeptiert drei Parameter.

```
add_role(  $role, $display_name, $capabilities );
```

Die ersten beiden Parameter sollten Strings sein (und benötigt werden), damit die Funktion ausgeführt werden kann. Sie definieren den Namen der neuen benutzerdefinierten Rolle bzw. den Anzeigenamen. Der letzte Parameter ist optional und sollte ein Array sein. Du kannst ihn verwenden, um alle Fähigkeiten der neuen Rolle zuzuweisen.

Erstellen wir eine benutzerdefinierte Benutzerrolle namens **Community Manager**, die Kommentare moderieren und Beiträge auf der gesamten Webseite bearbeiten kann. Hier siehst du, wie du das machen kannst:

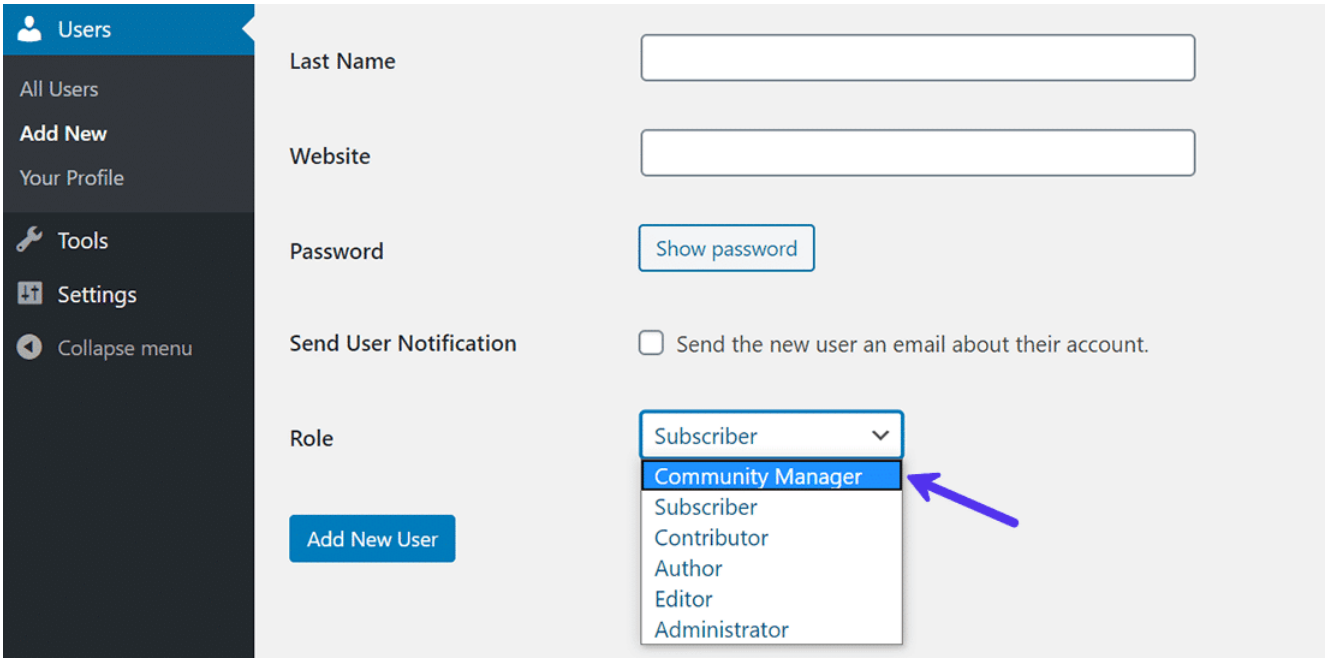
```
<?php

/*
Plugin Name:  Add Community Manager Role
Version:  1.0
Description:  Add a Custom User Role called 'Community Manager'
Author:  Salman Ravoof
Author URI:  https://www.salmanravoof.com/
License: GPLv2 or later
License URI: https://www.gnu.org/licenses/gpl-2.0.html
Text Domain:  add-community-manager-role
*/

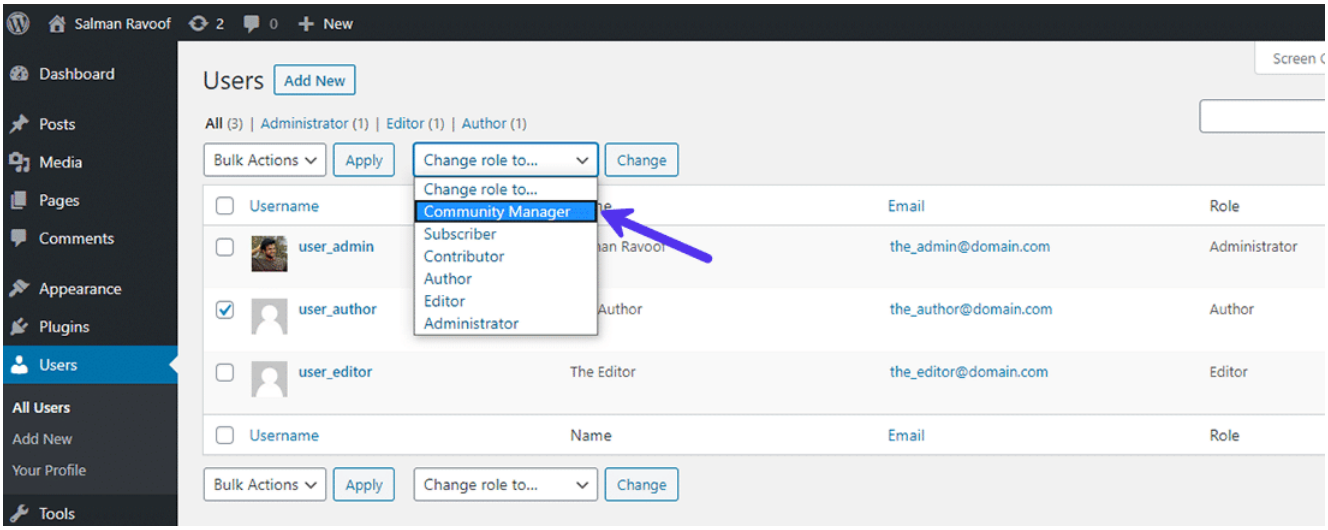
// this code will run only once on plugin activation and never again
function add_community_manager_role() {
    add_role(
        'community_manager',
          __('Community Manager', 'add-community-manager-
```

```
role'),
        array(
            'read' => true,
            'moderate_comments' => true,
            'edit_posts' => true,
            'edit_other_posts' => true,
            'edit_published_posts' => true
        )
    );
}
register_activation_hook(                    __FILE__,
'add_community_manager_role' );
```

Wie zuvor läuft die **add_role()**-Funktion nur einmal bei der Aktivierung des Plugins und nie wieder. Speichere die Datei und aktiviere das Plugin in deinem Administrator-Dashboard. Du solltest nun in der Lage sein, die Rolle des **Community Managers** sowohl neuen als auch bestehenden Usern zuzuweisen.



Neuen Benutzern die benutzerdefinierte Benutzerrolle zuweisen



Bestehenden Benutzern die benutzerdefinierte Benutzerrolle zuweisen

Du kannst auch die Fähigkeiten, die dieser neuen Rolle zugewiesen wurden, überprüfen, indem du den Wert des **wp_user_roles**-Feldes unter der **wp_options**-Tabelle in deiner Datenbank überprüfst. Hier ist, was ich in der Datenbank meiner Webseite gefunden habe:

```
array (
  'administrator' =>
    // [...]
  'editor' =>
    // [...]
  'author' =>
    // [...]
  'contributor' =>
    // [...]
  'subscriber' =>
    // [...]
  'community_manager' =>
  array (
    'name' => 'Community Manager',
    'capabilities' =>
    array (
      'read' => true,
      'moderate_comments' => true,
      'edit_posts' => true,
      'edit_other_posts' => true,
      'edit_published_posts' => true,
    ),
```

```
  ),
)
```

Ganz am Ende ist die neue Rolle aufgelistet, die wir gerade mit all ihren Fähigkeiten hinzugefügt haben. Du kannst diese Rolle weiter bearbeiten, indem du Fähigkeiten hinzufügst oder entfernst.

Bevor du die neue Benutzerrolle einem echten Benutzer zuweist, musst du unbedingt testen, ob sie wie beabsichtigt funktioniert. Hier ist eine Checkliste, die du befolgen kannst, um es zu testen:

Testen einer neuen Benutzerrolle

Bevor du die neue Benutzerrolle einem echten Benutzer zuweist, musst du unbedingt testen, ob sie wie beabsichtigt funktioniert. Hier ist eine Checkliste, die du befolgen kannst, um es zu testen:

1. Erstelle einen Test-Benutzer-Account und weise ihm die neue Benutzerrolle zu.
2. Logge dich mit dem Testbenutzer ein und stelle sicher, dass alle seine Fähigkeiten wie beabsichtigt funktionieren. Wenn du ihm zum Beispiel die Möglichkeit gegeben hast, veröffentlichte Beiträge zu bearbeiten, dann gehe zu einem beliebigen Beitrag und überprüfe, ob du ihn bearbeiten kannst. Je mehr Fähigkeiten du der Rolle zugewiesen hast, desto mehr Zeit wirst du damit verbringen, sie alle zu testen.
3. Als nächstes solltest du versuchen, einen beliebigen Admin-Link auf höherer Ebene direkt in deinem Browser zu besuchen. Ich habe dies getestet, indem ich direkt den WordPress-Einstellungsbildschirm besucht habe, und wie erwartet, hat mich WordPress nicht hereingelassen.
   Die Meldung ‚Zugang verweigert‘, die von WordPress angezeigt wird
4. Lösche den Testnutzer, nachdem du den Test beendet hast.

Das ist so ziemlich alles! Jetzt kannst du die neue Rolle den Benutzern deiner Webseite zuweisen.

Du kannst die Plugins User Switching oder View Admin As verwenden, um mit nur einem Klick zwischen verschiedenen Benutzerkonten auf deiner Webseite zu wechseln. Sie sind super praktisch, um die Fähigkeiten mehrerer Benutzer auszuprobieren. Ich habe sie beide später in diesem Artikel ausführlich behandelt.

Benutzerdefinierte Benutzerrollen in WordPress Multisite erstellen

WordPress Multisite handhabt Benutzerrollen etwas anders als WordPress Single-Site-Installationen. Während du mit der **add_role()**-Funktion eine benutzerdefinierte Benutzerrolle erstellen kannst, wie wir es zuvor getan haben, funktioniert die neue Rolle nur auf der primären Webseite des Netzwerks (die erste Webseite, die du erstellt hast). Sie wird sich nicht auf alle Unterseiten des Netzwerks ausbreiten.

Um sicherzustellen, dass der programmierte Code innerhalb deiner Callback-Funktion auf jeder Webseite deines Netzwerks läuft, musst du seine Ausführung erzwingen, indem du alle Webseiten deines Netzwerks eine nach der anderen durchschleifst. Für dieses Beispiel werde ich eine neue Benutzerrolle namens **Plugin Manager** erstellen, die alle Möglichkeiten hat, Plugins zu verwalten.

```
<?php

/*
Plugin Name:  Add Plugin Manager Role
Version:  1.0
Description:  Add a custom user role named Plugin Manager in a WordPress Multisite Installation
Author:  Salman Ravoof
Author URI:  https://www.salmanravoof.com/
License: GPLv2 or later
```

```
License URI: https://www.gnu.org/licenses/gpl-2.0.html
Text Domain:  add-plugin-manager-role
*/

/*
make the code run on every site in the network
when the plugin is Network Activated
*/
function add_plugin_manager_role( $network_wide ) {

     if ( is_multisite() && $network_wide ) {
         // run the code for all sites in a Multisite network
          foreach ( get_sites(['fields'=>'ids']) as $blog_id )
{
             switch_to_blog( $blog_id );
                 add_role(
                      'plugin_manager',
                        __('Plugin Manager', 'add-plugin-
manager-role'),
                      array(
                           'install_plugins' => true,
                           'activate_plugins' => true,
                           'edit_plugins' => true,
                           'delete_plugins' => true
                      )
                 );
             }
             restore_current_blog();
     }
     else {
         add_role(
              'plugin_manager',
              __('Plugin Manager', 'add-plugin-manager-role'),
              array(
                   'install_plugins' => true,
                   'activate_plugins' => true,
                   'edit_plugins' => true,
                   'delete_plugins' => true
              )
         );
     }
```

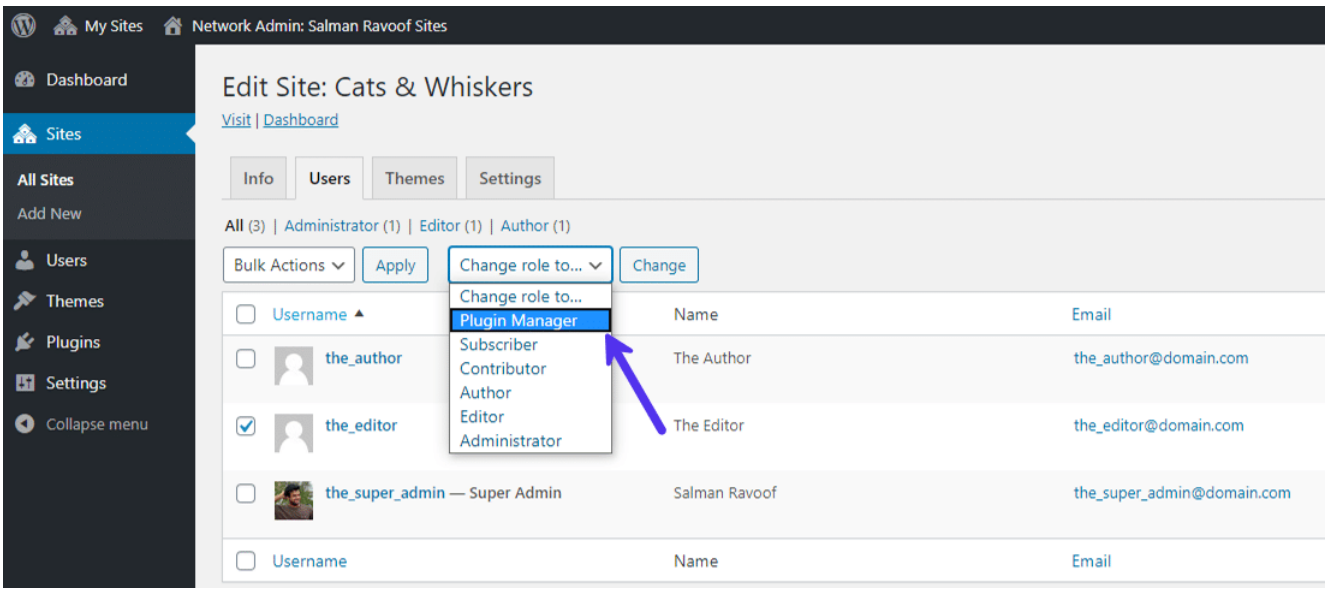
```
}
register_activation_hook( __FILE__, 'add_plugin_manager_role'
);
```

Lass uns den oben programmierten Code im Detail durchgehen:

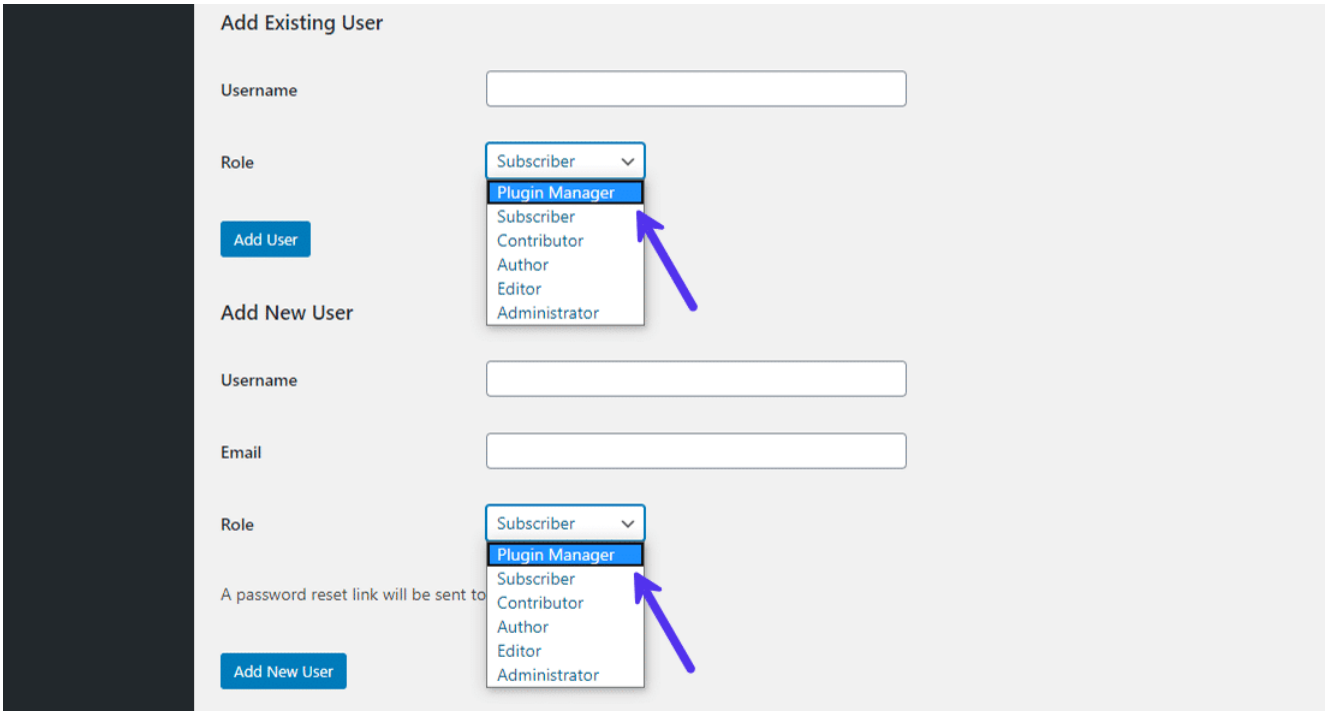
- Zuerst hookst du dich mit der **register_activation_hook()**-Funktion in die Plugin-Aktivierungsaktion ein und übergibst deine Callback-Funktion. Hier ist die Callback-Funktion **add_plugin_manager_role()**.
- Dann definierst du deine Callback-Funktion und gibst ein einziges Argument namens **$network_wide** mit ein.
- Der Parameter **$network_wide** ist ein bool, das **true** zurückgibt, wenn du das Plugin für das gesamte Netzwerk aktiviert hast. Er gibt **false** zurück, wenn du es nur für die aktuelle Webseite aktiviert hast. Außerdem ist er nur für Multisite-Installationen anwendbar und sein Standardwert ist **false**.
- Die bedingte Anweisung **is_multisite() && $network_wide** prüft, ob das Plugin bei einer Multisite-Installation ‚Network Activated' ist. Wenn es **true** ist, führt es den in der **If**-Anweisung enthaltenen Code aus. Wenn es **false** ist, wird der Code in der ‚**else**‚-Anweisung ausgeführt.
- Die Funktion **get_sites(['fields'=>'ids'])** gibt eine Liste aller Webseiten-IDs im Netzwerk zurück. Mit der PHP-Funktion **foreach()** durchläuft sie alle, um den Code auf jeder Webseite des Netzwerks einzeln auszuführen.
- Die switch_to_blog($blog_id) Funktion leitet die nächsten paar Zeilen Code zur Ausführung für die Subsite mit der **$blog_id** ID an. Da WordPress in erster Linie als Blogging-Plattform angefangen hat, kannst du das Wort ‚Blog' durch ‚Webseite' ersetzen, um die Verwendung hier besser zu verstehen.
- Als nächstes benutzt du die **add_role()**-Funktion, um die benutzerdefinierte Benutzerrolle mit ihren Fähigkeiten zu erstellen. Dies folgt den gleichen Code-Konventionen, wie zuvor in diesem Artikel erklärt.

- Bevor du die Schleife beendest, definiere die restore_current_blog()-Funktion, um sicherzustellen, dass du den Zustand der geschalteten Webseite wieder in den ursprünglichen Zustand versetzt.
- Der Code in der **else**-Anweisung ist ein Fallback, um die Kompatibilität mit Single-Site-Installationen sicherzustellen.

Speichere die Plugin-Datei und gehe auf dem **Bildschirm Network Admin > Plugins** auf ‚Netzwerk aktivieren' dein benutzerdefiniertes Plugin. Danach gehst du zum Reiter ‚**Benutzer**, unter der Seite ‚**Seite bearbeiten**, auf deiner Webseite, um zu überprüfen, ob die neue Rolle des **Plugin-Managers** verfügbar ist.

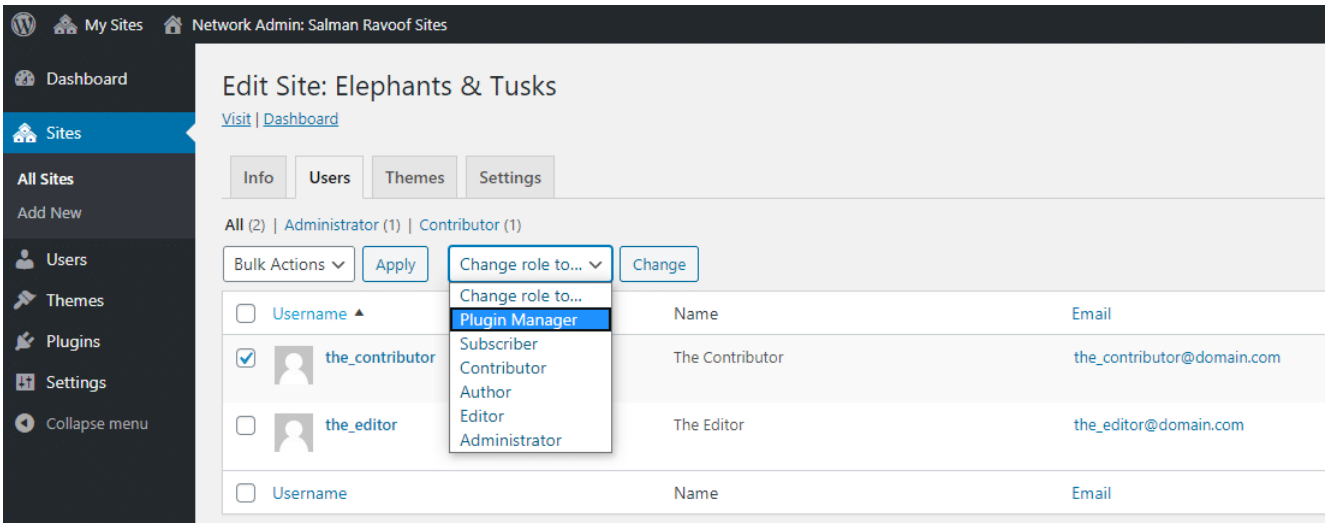


Änderung der Rolle der bestehenden Benutzer der Webseite auf die neue Benutzerrolle



Zuweisen der benutzerdefinierten Benutzerrolle an neue Benutzer für eine Subsite

Ich habe auch bestätigt, dass diese neue Benutzerrolle auf anderen Webseiten im Netzwerk verfügbar ist. Sie funktioniert einwandfrei.



Bestehenden Benutzern auf Unterseiten die neue Benutzerrolle zuweisen

Du kannst die neue benutzerdefinierte Rolle und ihre Fähigkeiten auch überprüfen, indem du einen Blick in die Datenbank deiner Webseite wirfst. Anders als bei Einzelplatz-Installationen erstellt WordPress Multisite jedoch für jede Webseite eine eigene **wp_options**-Tabelle.



Wo Benutzerrollen in einer WordPress Multisite-Datenbank gespeichert werden

Du kannst die subsite-spezifischen Tabellen als **wp_2_options**, **wp_3_options** und **wp_4_options** finden. Ebenso sind die Rollen und Fähigkeiten in ihren jeweiligen Feldern mit den Namen **wp_2_user_roles**, **wp_3_user_roles** und **wp_4_user_roles** gespeichert.

Du hast festgelegt, wie du eine benutzerdefinierte Benutzerrolle für alle Webseiten in deinem Netzwerk erstellen kannst, aber was ist mit den Webseiten, die in Zukunft erstellt werden? Um sicherzustellen, dass du diese benutzerdefinierte Benutzerrolle zu jeder neuen Webseite, die im Netzwerk erstellt wird, hinzufügst, kannst du den folgenden Code an dein Plugin anhängen:

```
// run the code once again when a new site is created
function add_custom_user_role_new_site( $blog_id ) {
    // check whether the plugin is active for the network
     if ( is_plugin_active_for_network( 'add-custom-user-role/add-custom-user-role.php' ) ) {
        switch_to_blog( $blog_id );
        add_role(
             'plugin_manager',
             __('Plugin Manager', 'add-plugin-manager-role'),
             array(
                  'install_plugins' => true,
                  'activate_plugins' => true,
```

```
                    'edit_plugins' => true,
                    'delete_plugins' => true
                )
            );
            restore_current_blog();
        }
}
add_action(  'wpmu_new_blog',  'add_custom_user_role_new_site'
);
```

- Die Aktion wpmu_new_blog wird immer dann ausgelöst, wenn jemand eine neue Webseite in einem Multisite-Netzwerk erstellt. Du kannst dich in diese Aktion mit deiner Callback-Funktion einklinken, um die benutzerdefinierte Benutzerrolle hinzuzufügen.
- Die is_plugin_active_for_network()-Funktion prüft, ob das Plugin für das gesamte Netzwerk aktiv ist und gibt einen bool-Wert zurück. Sie akzeptiert den Pfad der Plugin-Datei als Argument.
- Der Rest des programmierten Codes folgt der gleichen Logik wie vorher. Du wechselst auf die neue Webseite mit dem **$blog_id** Parameter, erstellst deine eigene Rolle mit der **add_role()** Funktion und wechselst dann zurück auf die aktuelle Webseite mit der **restore_current_blog()** Funktion.

Wie man Benutzerrollen aus WordPress entfernt

Du kannst jede Benutzerrolle aus WordPress entfernen, indem du die remove_role( ) Funktion benutzt. Sie akzeptiert nur ein Argument, nämlich den Rollennamen. Zum Beispiel kannst du die Contributor-Rolle entfernen, indem du den folgenden Code irgendwo auf deiner Webseite ausführst:

```
remove_role( 'contributor' );
```

Im Gegensatz zur **add_role()**-Funktion, die die Datenbank immer

wieder aktualisiert, wenn sie nicht bei der Plugin- oder Theme-Aktivierung ausgeführt wird, wird die **remove_role()**-Funktion nur ausgeführt, wenn die Rolle existiert. Da jede als Argument übergebene Rolle beim ersten Ausführen entfernt wird, brauchst du dir keine Gedanken darüber zu machen, wo du diese Funktion ausführst.

Um jedoch zukünftige Konflikte zu vermeiden, entferne den Code, nachdem die Rolle aus der Datenbank entfernt wurde.

Benutzerdefinierte Fähigkeiten in WordPress erstellen

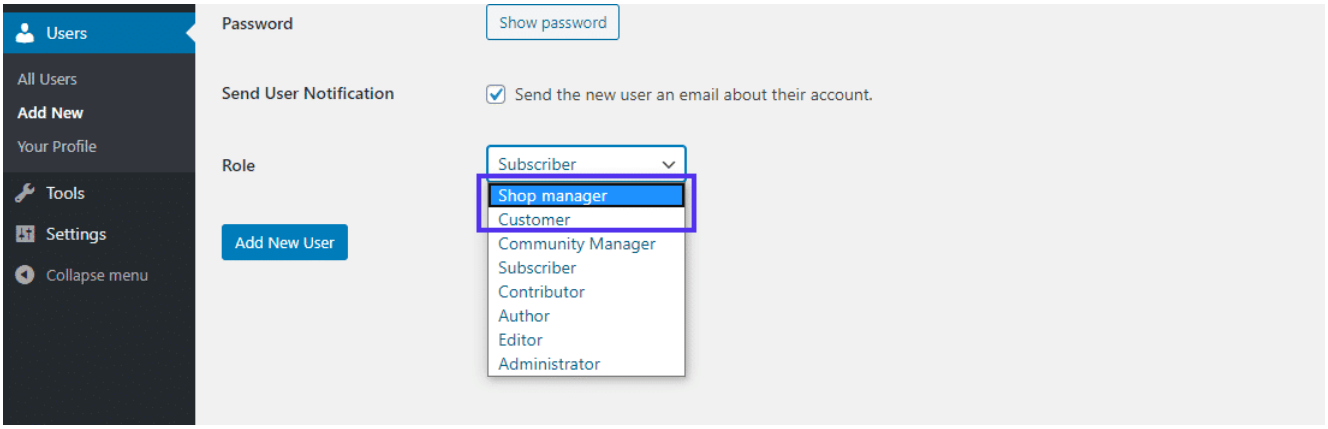
Für die meisten Anwendungsfälle reicht es aus, bestehende Benutzerrollen zu bearbeiten und neue benutzerdefinierte Rollen mit Hilfe der in WordPress eingebauten Funktionen zu erstellen, aber vielleicht möchtest du neue Funktionen für Funktionen definieren, die durch deinen benutzerdefinierten Code eingeführt wurden (mit einem Plugin oder Theme).

Du kannst dann diese benutzerdefinierten Funktionen verwenden, um neue Rollen zu definieren oder sie zu bestehenden hinzuzufügen.

WooCommerce fügt zum Beispiel zusätzliche Fähigkeiten und Rollen zusammen mit seinen umfangreichen E-Commerce-Features hinzu. Einige Fähigkeiten, die es hinzufügt, sind:

- Erlaube die Verwaltung der WooCommerce-Einstellungen
- Produkte erstellen und bearbeiten
- WooCommerce-Berichte ansehen

Mit diesen Fähigkeiten fügt es zwei neue Benutzerrollen hinzu: **Kunde** und **Shop Manager**.



WooCommerce fügt seine eigenen Benutzerrollen hinzu
Die Kundenrolle ist der Abonnentenrolle fast ähnlich, außer dass Benutzer mit der Kundenrolle ihre Kontoinformationen bearbeiten und aktuelle/vergangene Bestellungen einsehen können. Die Rolle des Shop-Managers beinhaltet alle Funktionen eines Redakteurs, außerdem stehen ihnen alle WooCommerce-Funktionen zur Verfügung.

Andere Plugins, die benutzerdefinierte Fähigkeiten und/oder Rollen einführen, sind The Events Calendar, Visual Portfolio, WPML und WP ERP.

Wenn du dich in die Dokumentation all dieser Plugins legst, wirst du feststellen, dass sie fast alle ihre benutzerdefinierten Fähigkeiten an die von ihnen definierten benutzerdefinierten Posttypen binden. Im Fall von WooCommerce sind es die benutzerdefinierten Posttypen **Produkte** und Bestellungen, während es in anderen Fällen **Events**, **Portfolios**, **Übersetzungen** und **Kunden** sind.

Lass uns lernen, wie man benutzerdefinierte Fähigkeiten für einen benutzerdefinierten Beitragstyp erstellt.

Als erstes musst du ein Plugin einrichten und den gewünschten benutzerdefinierten Beitragstyp registrieren. In meinem Beispiel registriere ich einen neuen benutzerdefinierten Beitragstyp namens **„Stories„**.

```
<?php

/*
```

```
Plugin Name:    Custom Post Type and Capabilities
Version:        1.0
Description:    Register a custom post type and define custom capabilities tied into it.
Author:         Salman Ravoof
Author URI:     https://www.salmanravoof.com/
License:        GPLv2 or later
License URI:    https://www.gnu.org/licenses/gpl-2.0.html
Text Domain:    custom-post-type-capabilities
*/

// register a custom post type, in this case it's called "story" //
function cpt_story_init() {
    $labels = array(
        'name'                  => _x( 'Stories', 'custom-post-type-capabilities' ),
        'singular_name'         => _x( 'Story', 'custom-post-type-capabilities' ),
        'menu_name'             => _x( 'Stories', 'Admin Menu text', 'custom-post-type-capabilities' ),
        'name_admin_bar'        => _x( 'Story', 'Add New on Toolbar', 'custom-post-type-capabilities' ),
        'add_new'               => __( 'Add New', 'custom-post-type-capabilities' ),
        'add_new_item'          => __( 'Add New Story', 'custom-post-type-capabilities' ),
        'new_item'              => __( 'New Story', 'custom-post-type-capabilities' ),
        'edit_item'             => __( 'Edit Story', 'custom-post-type-capabilities' ),
        'view_item'             => __( 'View Story', 'custom-post-type-capabilities' ),
        'all_items'             => __( 'All Stories', 'custom-post-type-capabilities' ),
        'search_items'          => __( 'Search Stories', 'custom-post-type-capabilities' ),
        'parent_item_colon'     => __( 'Parent Stories:', 'custom-post-type-capabilities' ),
        'not_found'             => __( 'No stories found.', 'custom-post-type-capabilities' ),
```

```
        'not_found_in_trash'    => __( 'No stories found in
Trash.', 'custom-post-type-capabilities' ),
        'featured_image'        => _x( 'Story Cover Image',
'custom-post-type-capabilities' ),
        'set_featured_image'    => _x( 'Set cover image',
'custom-post-type-capabilities' ),
        'remove_featured_image' => _x( 'Remove cover image',
'custom-post-type-capabilities' ),
        'use_featured_image'    => _x( 'Use as cover image',
'custom-post-type-capabilities' ),
        'archives'              => _x( 'Story archives',
'custom-post-type-capabilities' ),
        'insert_into_item'      => _x( 'Insert into story',
'custom-post-type-capabilities' ),
        'uploaded_to_this_item' => _x( 'Uploaded to this
story', 'custom-post-type-capabilities' ),
        'filter_items_list'     => _x( 'Filter stories list',
'custom-post-type-capabilities' ),
        'items_list_navigation' => _x( 'Stories list
navigation', 'custom-post-type-capabilities' ),
        'items_list'            => _x( 'Stories list',
'custom-post-type-capabilities' ),
    );

    $args = array(
        'labels'             => $labels,
        'public'             => true,
        'menu_icon'          => 'dashicons-book',
        'publicly_queryable' => true,
        'show_ui'            => true,
        'show_in_menu'       => true,
        'query_var'          => true,
        'rewrite'            => array( 'slug' => 'story' ),
        'capability_type'    => array ( 'story', 'stories' ),
        'map_meta_cap'       => true,
        'has_archive'        => true,
        'hierarchical'       => false,
        'menu_position'      => 6,
        'supports'           => array( 'title', 'editor',
'author', 'thumbnail', 'excerpt', 'comments' ),
        'show_in_rest'       => true,
```

```
    );

    register_post_type( 'story', $args );
}

add_action( 'init', 'cpt_story_init' );
```

Hier ist eine Aufschlüsselung des obigen Skripts:

- Benutze die register_post_type()-Funktion, um deinen benutzerdefinierten Posttyp zu registrieren. Du kannst dich in die **Init**-Aktion einklinken, um diese Funktion auszuführen.
- Die **register_post_type()**-Funktion akzeptiert zwei Argumente. Das erste ist der Name des benutzerdefinierten Beitragstyps und das zweite ist ein Array, das alle Argumente für die Registrierung des Beitragstyps enthält.
- Die Variable **$args** enthält alle Argumente, die du an die Funktion **register_post_type()** übergibst. Eines ihrer Argumente (**'labels'**) ist selbst ein Array, das separat als die Variable **$label** definiert ist.
- Beachte das **'capability_type' => 'post'**-Argument. Das ist der standardmäßige ‚capability_type', der von WordPress verwendet wird, um die Lese-, Bearbeitungs- und Löschfunktionen für den benutzerdefinierten Beitragstyp zu erstellen.
- Um deine benutzerdefinierten Fähigkeiten zu erstellen, musst du den Wert des **cability_type**-Arguments durch den bevorzugten Namen deiner benutzerdefinierten Fähigkeiten ersetzen. Es akzeptiert entweder einen String oder ein Array als Argument. Das Array ist nützlich, wenn der Plural deiner benutzerdefinierten Fähigkeiten nicht der Standard-Suffix-Syntax folgt (z.B. Buch/Bücher vs. Geschichte/Geschichten).
- Du kannst auch das **capabilities**-Argument verwenden, um die neuen Fähigkeiten anders zu benennen, als es WordPress automatisch tut.

- Du musst deine benutzerdefinierten Fähigkeiten den primitiven Fähigkeiten von WordPress zuordnen. Setze das **map_meta_cap** Argument auf **true**, damit WordPress weiß, dass es die benutzerdefinierten Fähigkeiten wie vorgeschlagen mappen muss.

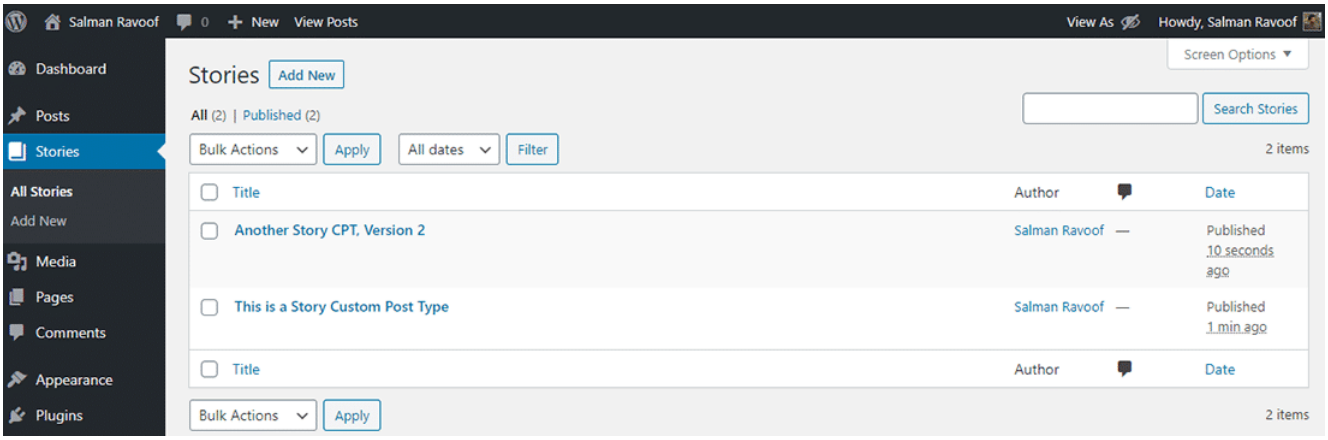
Als nächstes musst du die benutzerdefinierten Fähigkeiten zu den Rollen hinzufügen, die du für den benutzerdefinierten Beitragstyp **Stories** verwenden möchtest. In diesem Beispiel gewähre ich die Fähigkeit den Administrator- und Redakteur-Rollen.

Brauchst du ein blitzschnelles, sicheres und entwicklerfreundliches Hosting für deine Kunden-Webseiten? Kinsta wurde mit Blick auf WordPress-Entwickler entwickelt und bietet eine Vielzahl von Tools und ein leistungsstarkes Dashboard. Schau dir unsere Pläne an

```
// add the custom capabilities to the desired user roles
$roles = array( 'editor','administrator' );

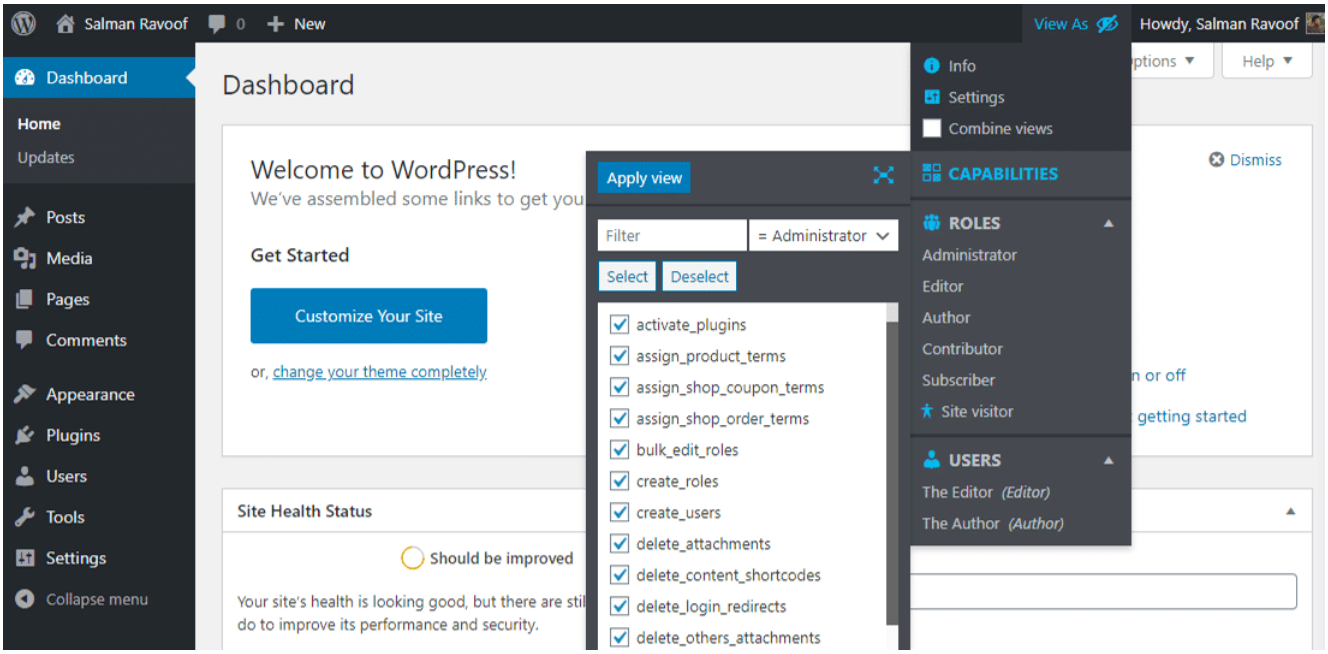
foreach( $roles as $the_role ) {
    $role = get_role($the_role);
            $role->add_cap( 'read' );
            $role->add_cap( 'read_story');
            $role->add_cap( 'read_private_stories' );
            $role->add_cap( 'edit_story' );
            $role->add_cap( 'edit_stories' );
            $role->add_cap( 'edit_others_stories' );
            $role->add_cap( 'edit_published_stories' );
            $role->add_cap( 'publish_stories' );
            $role->add_cap( 'delete_others_stories' );
            $role->add_cap( 'delete_private_stories' );
            $role->add_cap( 'delete_published_stories' );
}
```

Speichere die Datei und aktiviere dann das Plugin. Du solltest jetzt den **Stories** Link und das Panel in deinem Administrator oder Redakteur Dashboard sehen.



Das ‚Stories'-Panel im WordPress-Dashboard

Wenn du dir die auf deiner Webseite verfügbaren Funktionen ansiehst, siehst du auch alle mit den Stories zusammenhängenden Funktionen, die wir hinzugefügt haben. Hier benutze ich das Plugin View Admin As, um die Fähigkeiten zu überprüfen.



Benutzerdefinierte Fähigkeiten im Zusammenhang mit dem benutzerdefinierten Posttyp „Stories"

Du kannst eine umfangreiche Version dieses Plugins über diesen Gist herunterladen. Es registriert einen benutzerdefinierten Beitragstyp namens **Projekte** mit einer Reihe von benutzerdefinierten Fähigkeiten. Und dann weist es sie zwei benutzerdefinierten Rollen zu, die **Schüler** und **Lehrer** genannt werden, um dir beim Aufbau einer bildungsbezogenen Webseite zu helfen.

Es gibt einen Weg, benutzerdefinierte Fähigkeiten zu definieren, die den Benutzern Zugriff auf deine Plugin-Einstellungen basierend auf ihrer Rolle gewähren. Darüber zu diskutieren, wie man das macht, würde den Rahmen dieses Artikels sprengen, aber ihr könnt in diesem informativen Thread auf StackExchange nachschlagen, um weitere Informationen zu erhalten.

Die besten WordPress User Rollen und Fähigkeiten Plugins

Es ist toll zu wissen, wie man die Rollen und Fähigkeiten der Benutzer mit Hilfe von Code optimieren kann, aber es ist nicht für jeden geeignet. Es gibt so viele Dinge, die schief gehen können, wenn du dir nicht sicher bist, was du tust. Aber zu wissen, wie Rollen und Fähigkeiten in WordPress funktionieren, hilft immens, auch wenn du ein Plugin verwendest.

Schauen wir uns einige der beliebtesten WordPress-Plugins an, mit denen sich die Rollen und Fähigkeiten der WordPress-Benutzer leicht anpassen lassen. Ich werde auch ein paar hilfreiche Plugins auflisten, mit denen man die Rollen- und Fähigkeitsfunktionen schnell testen kann.

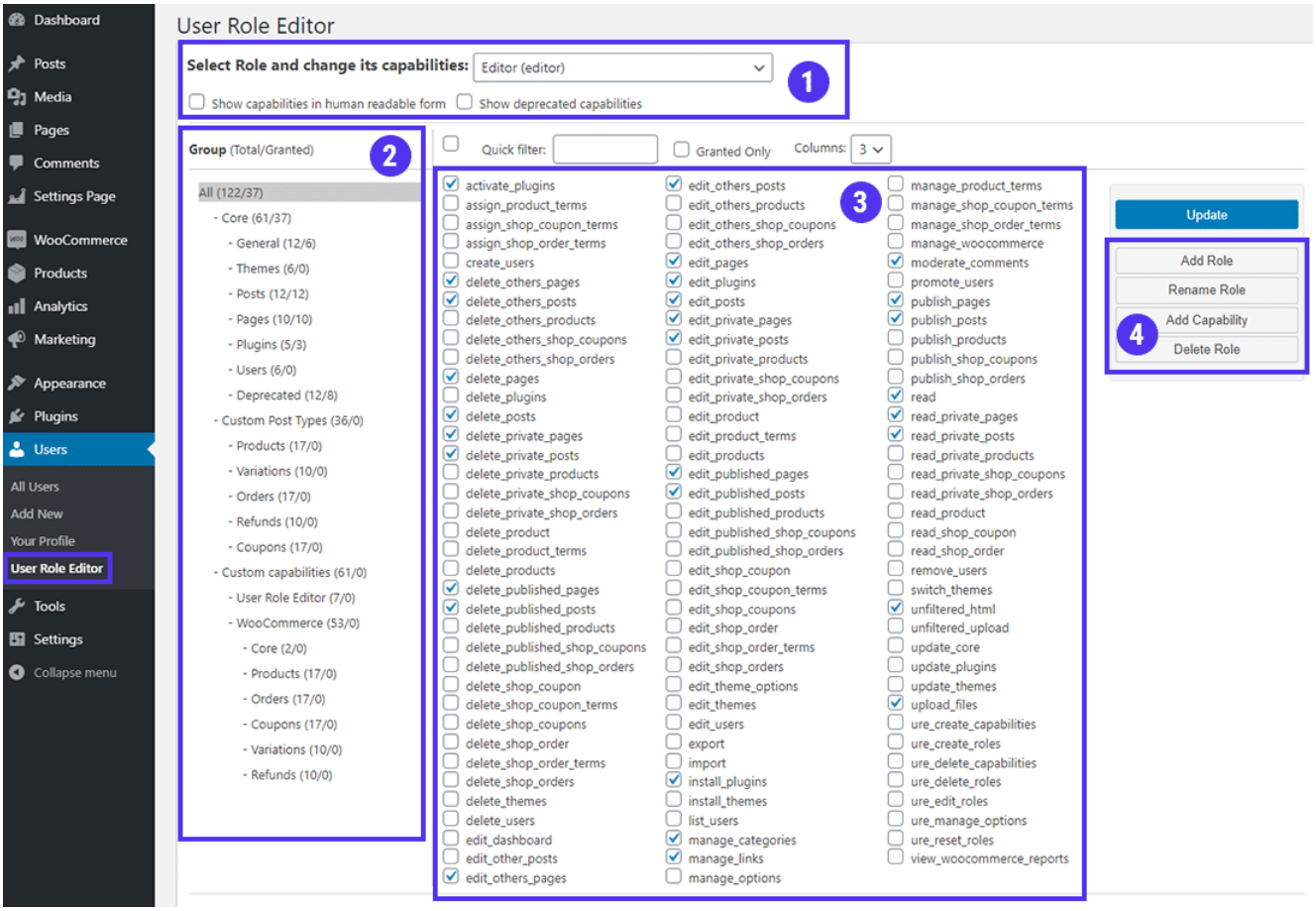
User Role Editor (von Vladimir Garagulia)



Das ‚User Role Editor' WordPress-Plugin User Role Editor ist das beliebteste Rollen- und

Fähigkeitsmanagement Plugin im WordPress Repository. Es kommt mit einem einfachen Interface, das es jedem erlaubt, Benutzerrollen und Fähigkeiten mit nur einem Klick zu bearbeiten.

Nachdem du das Plugin installiert und aktiviert hast, kannst du unter **Users > User Role Editor** in deinem Admin-Dashboard auf das primäre Interface zugreifen.



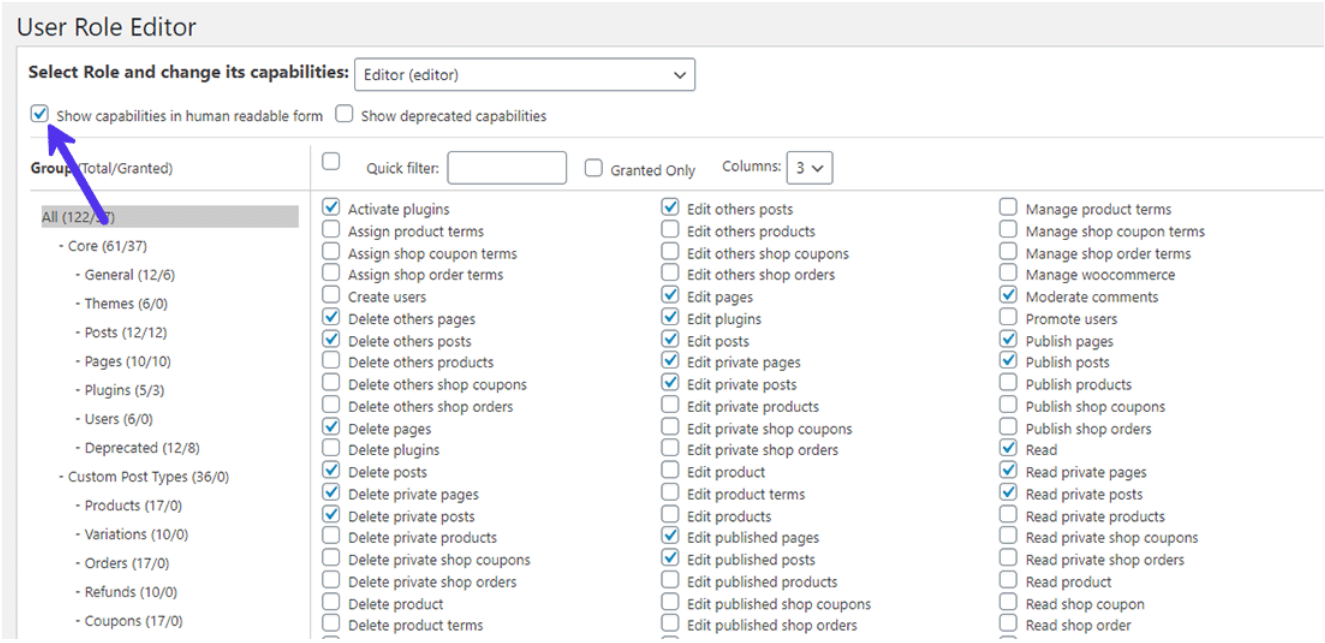
Das Dashboard des User Role Editors

Hier ist eine detaillierte Übersicht über die oben markierten Dashboard-Abschnitte:

1. Wähle die Rolle, die du anpassen möchtest, aus dem Dropdown-Menü aus. Es werden nicht nur die Standardrollen aufgelistet, sondern alle Rollen, die in deiner Datenbank vorhanden sind. Du kannst dich auch dafür entscheiden, die Fähigkeiten in menschenlesbarer Form statt ihrer Konstanten anzuzeigen. Eine weitere Option lässt dich veraltete Fähigkeiten sehen, die in

der neuesten Version von WordPress nicht mehr unterstützt werden.

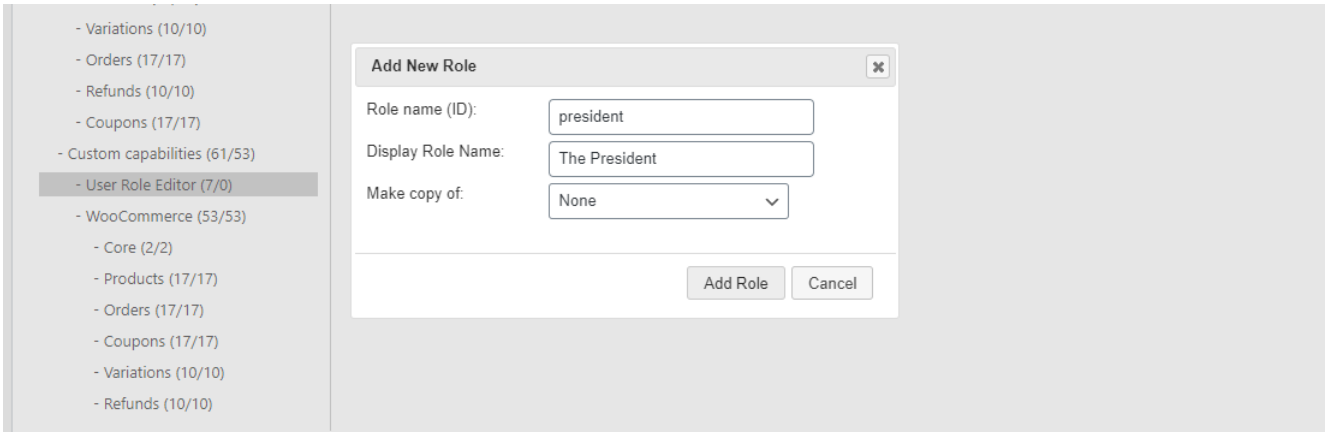
2. Der User Role Editor gruppiert alle Fähigkeiten in verschiedene Kategorien auf der linken Seite. Die Kernkategorie enthält alle eingebauten Fähigkeiten. Seitdem ich WooCommerce auf dieser Webseite installiert habe, findest du auch Fähigkeiten für seine benutzerdefinierten Posttypen. Sogar der User Role Editor Plugin fügt seinen eigenen Satz an benutzerdefinierten Fähigkeiten hinzu.
3. Auf der rechten Seite findest du alle Fähigkeiten aufgelistet. Da ich die Gruppe **Alle** ausgewählt habe, kann ich alle Fähigkeiten sehen. Du kannst sie jedoch filtern, indem du auf eine Gruppe auf der linken Seite klickst. Du kannst auch die Option **„Nur gewährte“** oben ankreuzen, um alle Fähigkeiten auszublenden, die von keiner Benutzerrolle verwendet werden.
4. Von hier aus kannst du auch eine **Rolle hinzufügen, eine Rolle umbenennen, eine Fähigkeit hinzufügen** und eine **Rolle löschen**. Ganz unten findest du eine zusätzliche Option, um die **Admin-Leiste für die Benutzerrolle auszublenden**.



Fähigkeiten in menschenlesbarer Form anzeigen

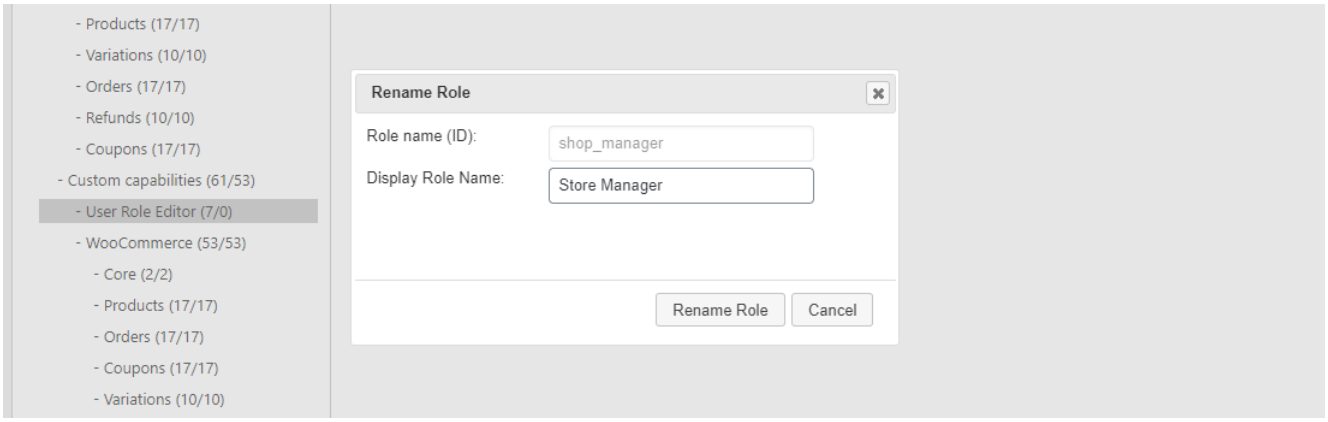
Um eine Benutzerrolle anzupassen, musst du nur die gewünschten

Fähigkeiten ankreuzen oder abwählen und auf die Schaltfläche **Aktualisieren** klicken, um deine Änderungen zu speichern. So einfach ist das.



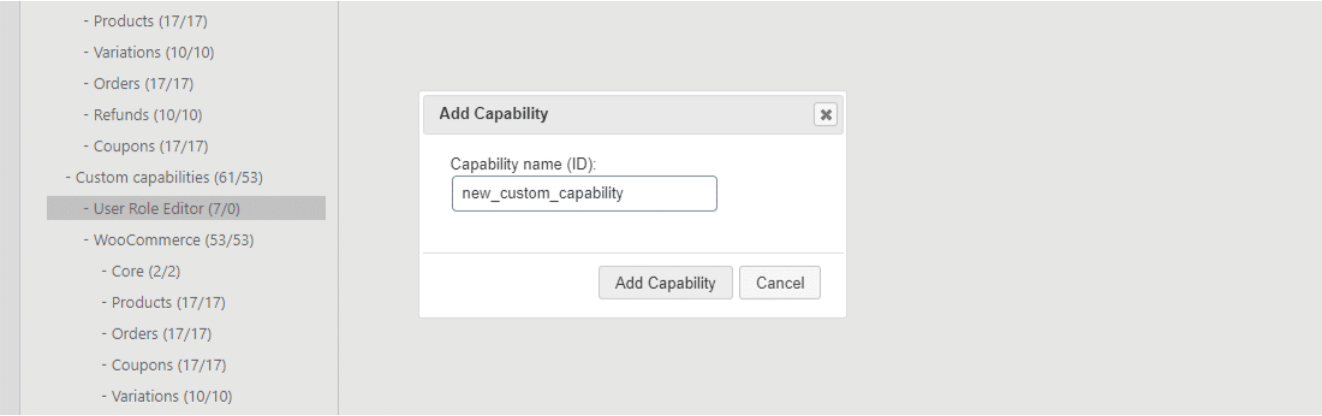
Hinzufügen einer neuen Rolle im User Role Editor

Klicke auf die Schaltfläche **Add Role**, um eine neue Rolle zu erstellen. Du kannst eine Rolle von Grund auf neu erstellen oder eine bestehende Rolle duplizieren, indem du die Dropdown-**Option Kopie von erstellen** verwendest.



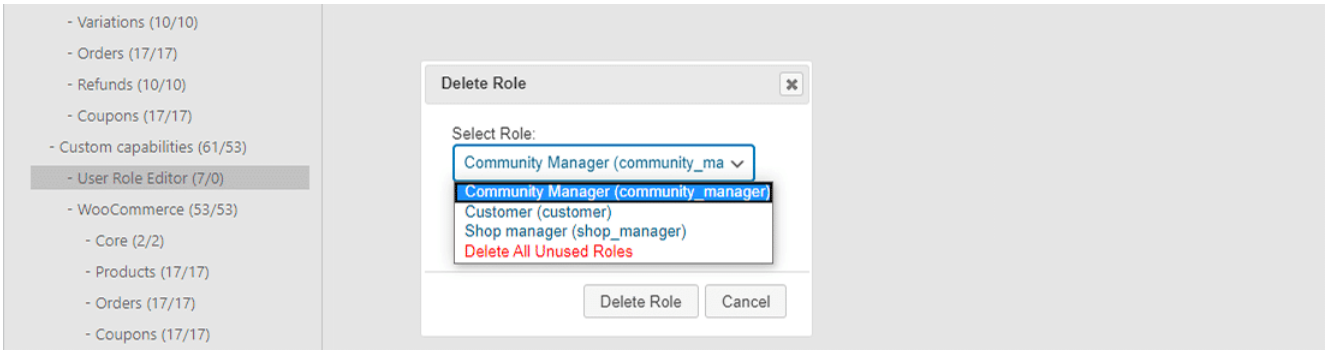
Den ‚Rollenanzeige-Namen' einfach umbenennen

Du kannst auch den Namen der **Rollenanzeige umbenennen**, indem du auf die Schaltfläche **Rolle umbenennen** klickst. Allerdings kannst du seine **Rollen-ID** (oder seinen Rollennamen) nicht ändern. Ein Workaround besteht darin, die Rolle, deren ID du ändern möchtest, zu duplizieren und dann die ursprüngliche Rolle zu löschen.



Hinzufügen einer neuen Fähigkeit im User Role Editor

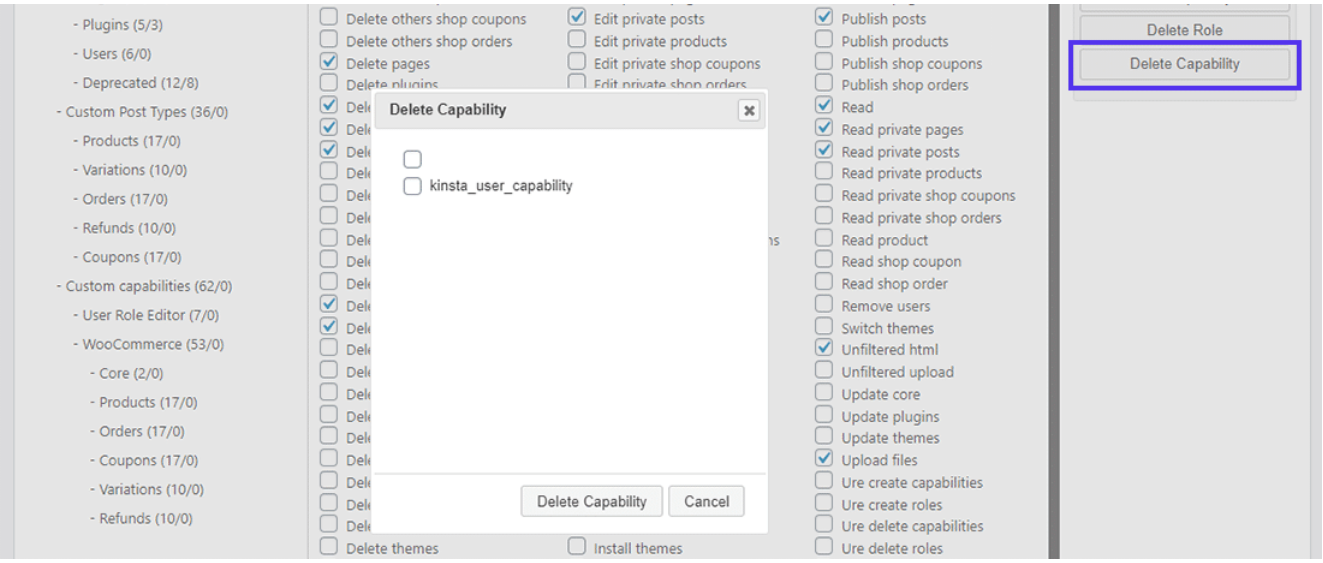
Du kannst neue Fähigkeiten hinzufügen, indem du auf die Schaltfläche **Add Capability** klickst.



Nicht zugewiesene Benutzerrollen einfach löschen

Wenn du auf die Schaltfläche „**Rollen löschen**“ klickst, kannst du benutzerdefinierte Rollen löschen, die du noch keinem Benutzer zugewiesen hast.

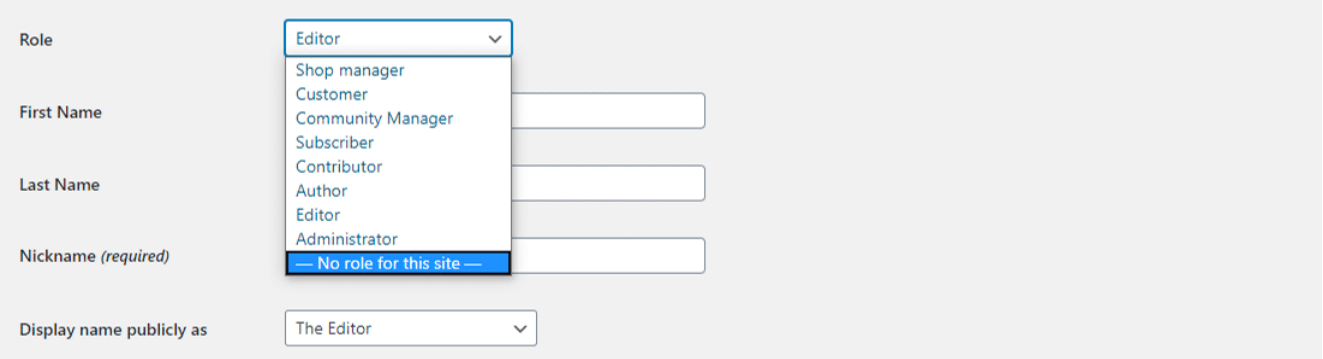
Hinweis: Mit dem User Role Editor kannst du die in WordPress eingebauten Rollen oder Fähigkeiten nicht löschen. Es ist auch nicht möglich, eine benutzerdefinierte Rolle zu löschen, wenn sie einem Benutzer zugewiesen ist, oder eine benutzerdefinierte Fähigkeit, wenn sie einer Nicht-Admin-Rolle zugewiesen ist.



Die Schaltfläche ‚Fähigkeit Löschen‘ im User Role Editor

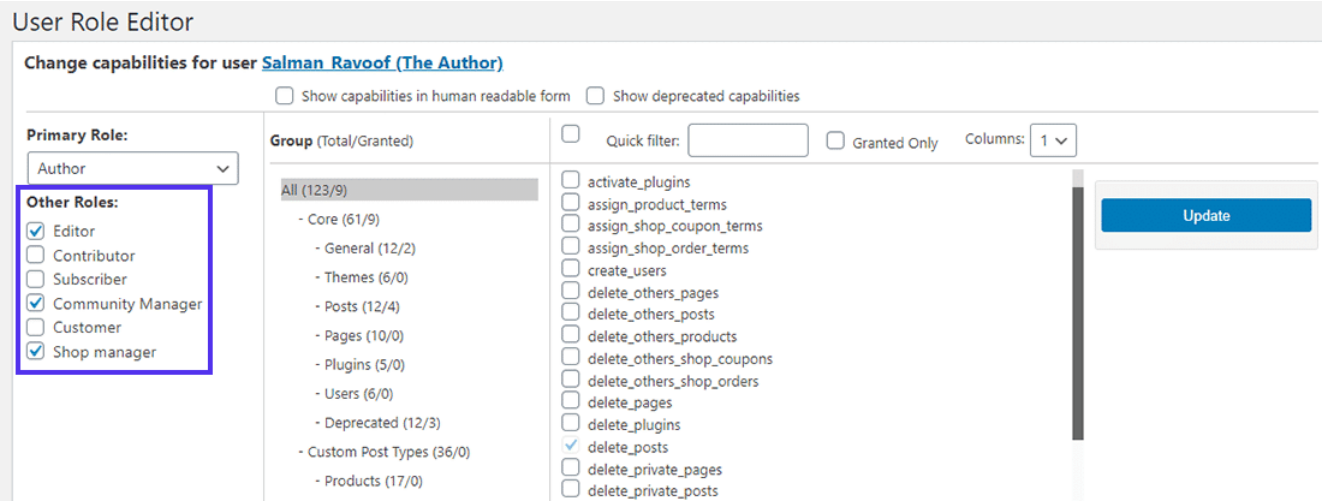
Ihr solltet beachten, dass die Schaltfläche „**Fähigkeit löschen**“ nur erscheint, wenn eine Fähigkeit nicht Nicht-Admins zugewiesen ist. Ansonsten ist sie versteckt.

Du kannst dem selben Benutzer auch mehrere Rollen zuweisen oder ihm überhaupt keine Rolle zuweisen.



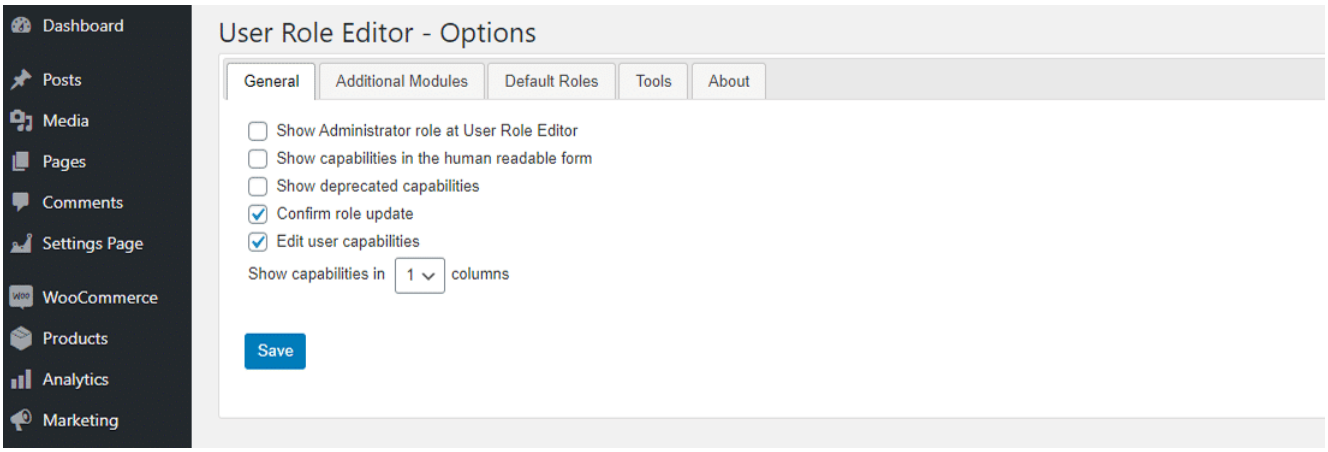
Entferne den Benutzer von jeder Rolle

Um einem Benutzer mehrere Rollen zuzuweisen, musst du zum **Benutzer**-Panel in deinem Dashboard gehen und dann auf den Link **Fähigkeiten** klicken, den du unten siehst, wenn du mit der Maus über den Benutzernamen fährst.



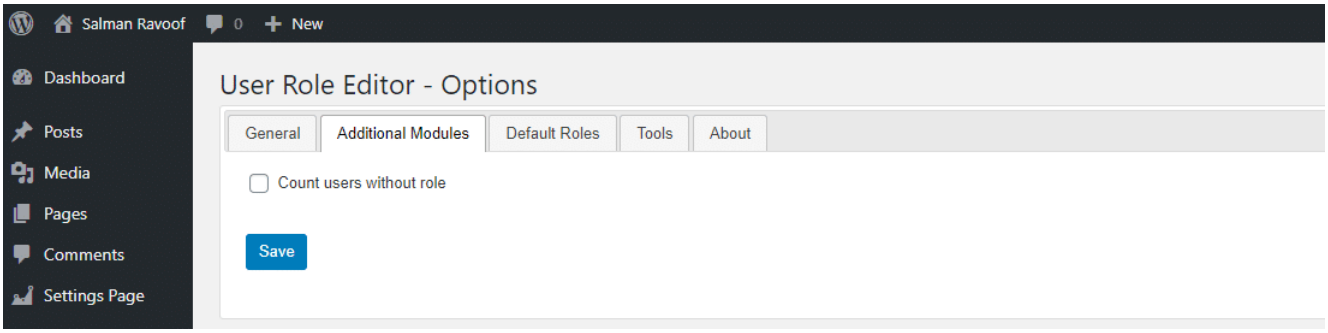
Dem gleichen Benutzer mehrere Rollen zuweisen

Wenn du in deinem Admin-Dashboard auf **Einstellungen > User Role Editor** gehst, findest du auch zusätzliche Optionen für das User Role Editor Plugin.

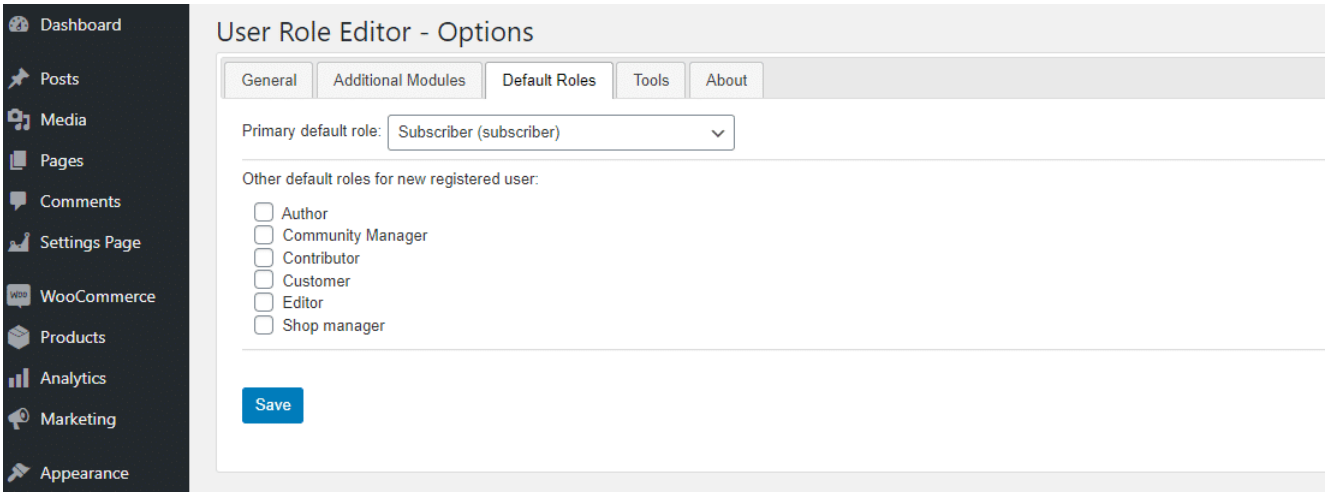


Die Registerkarte ‚Allgemein‘ für den User Role Editor

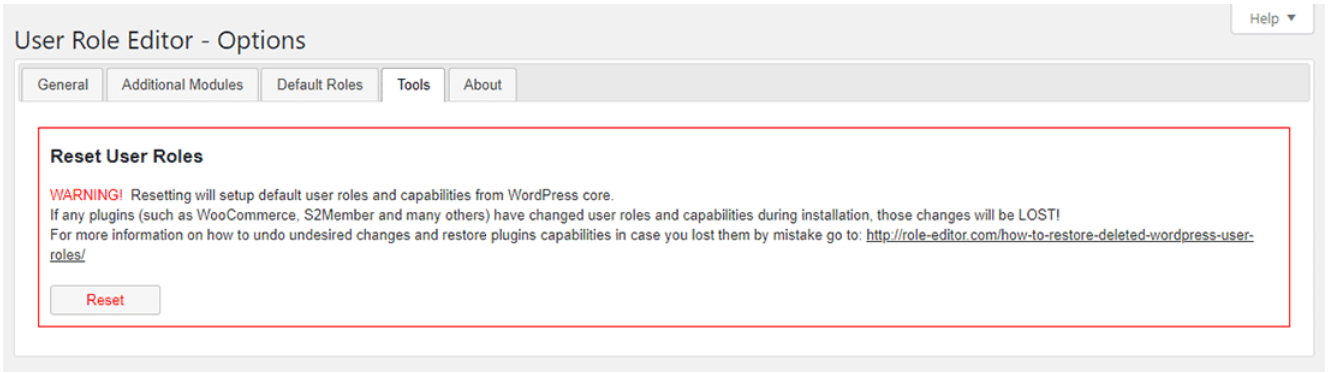
Hier kannst du die Standardeinstellungen des Plugins ändern, zusätzliche Module installieren, die Standardrolle, die neuen Benutzern zugewiesen wird, ändern und sogar Benutzerrollen und -fähigkeiten auf ihren Standardstatus zurücksetzen.



Zusätzliche Module helfen dir, die Funktionen des User Role Editors zu erweitern



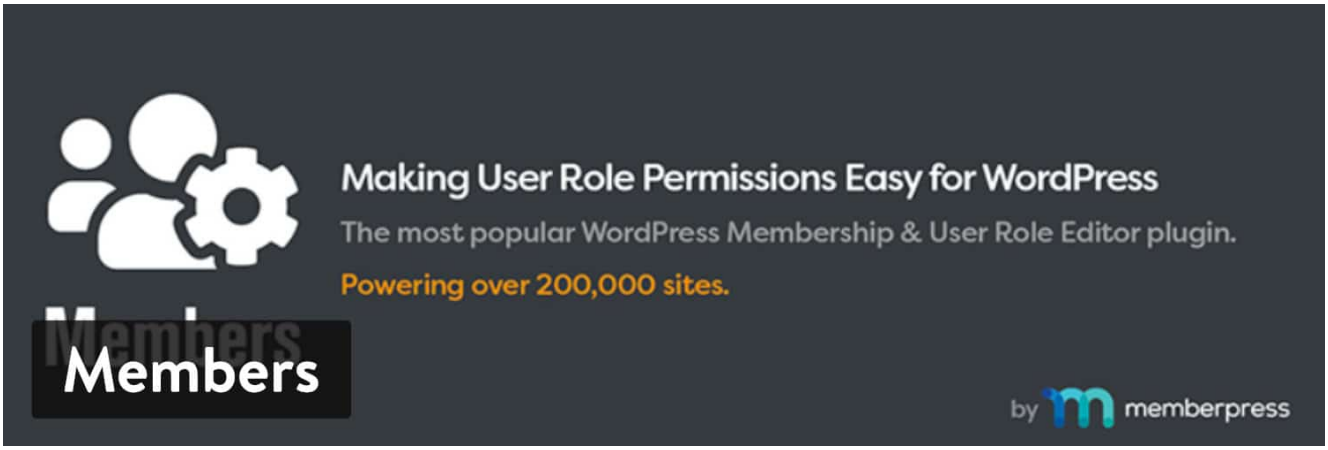
Die Standardrolle für neue Benutzer festlegen



Alle Benutzerrollen und Fähigkeiten auf ihren Standardstatus zurücksetzen

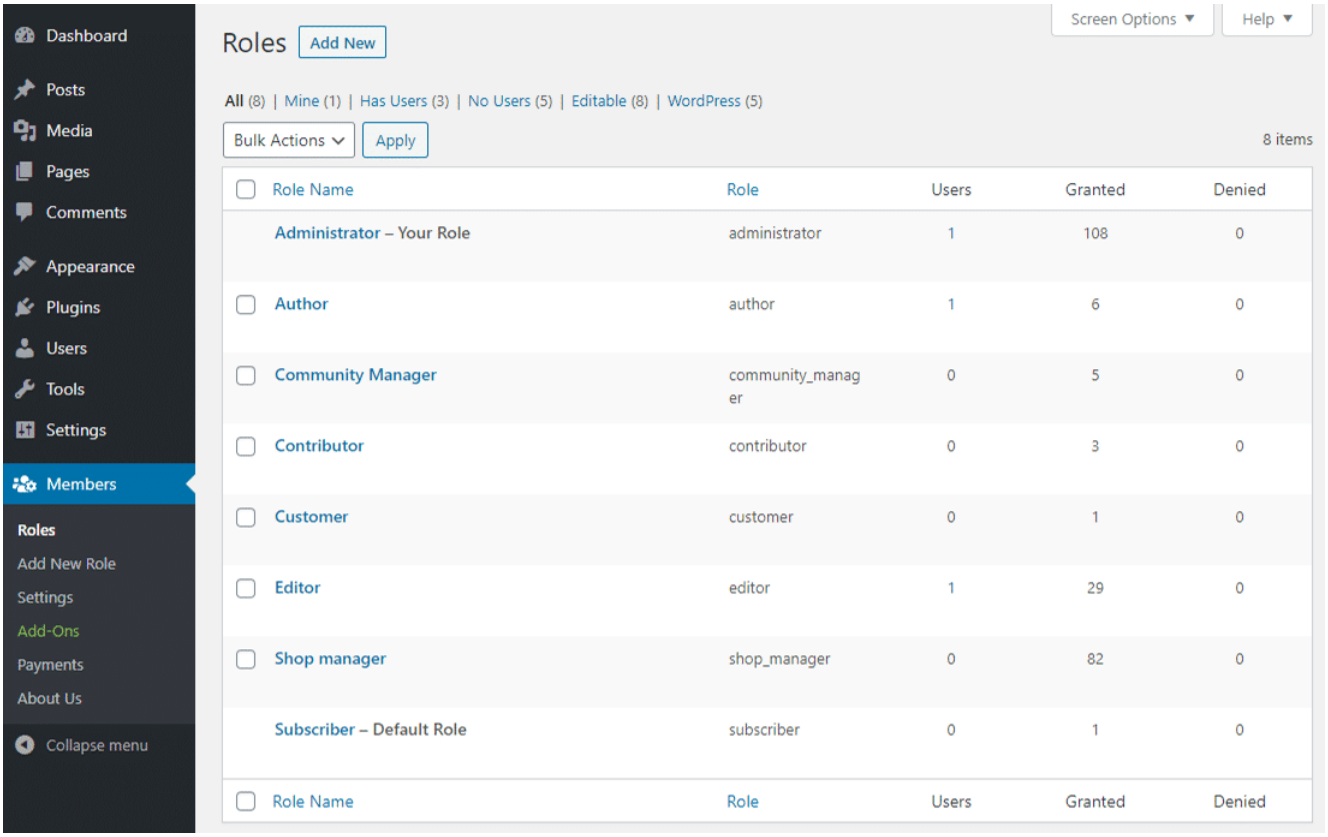
Während die kostenlose Version des User Role Editors für die meisten Anwendungsfälle mehr als genug ist, enthält die Premium-Version noch mehr Funktionen, einschließlich der Unterstützung für die Verwaltung von Rollen und Fähigkeiten in WordPress Multisite-Setups.

Members von MemberPress



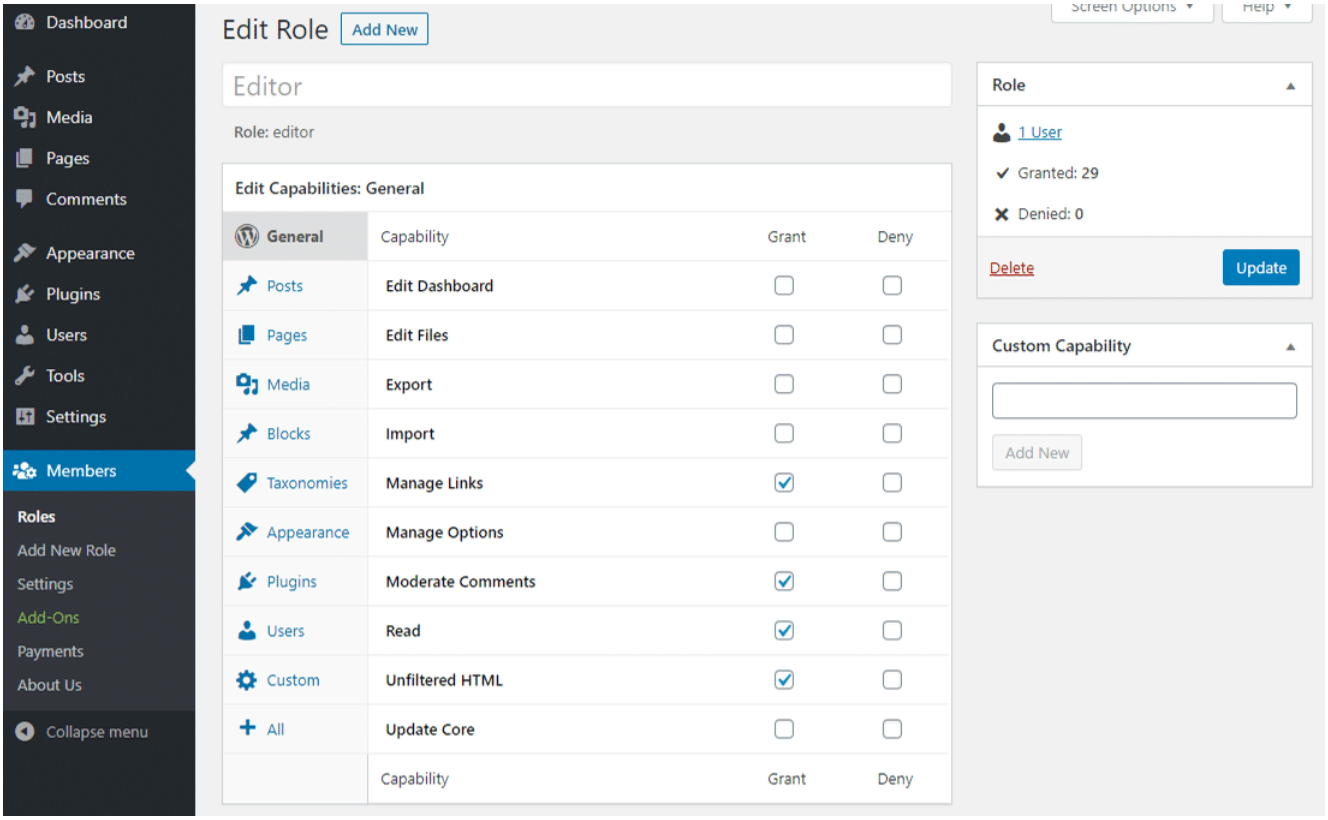
Das ‚Members‘ WordPress-Plugin von MemberPress

Members ist ein auf Mitgliedschaft fokussiertes Plugin für Benutzerrollen und -fähigkeiten für WordPress. Ursprünglich als einfaches Plugin für die Verwaltung von Benutzerrollen und Fähigkeiten gestartet, hat es sich seitdem in Richtung Mitgliedschaftsfunktionen entwickelt.



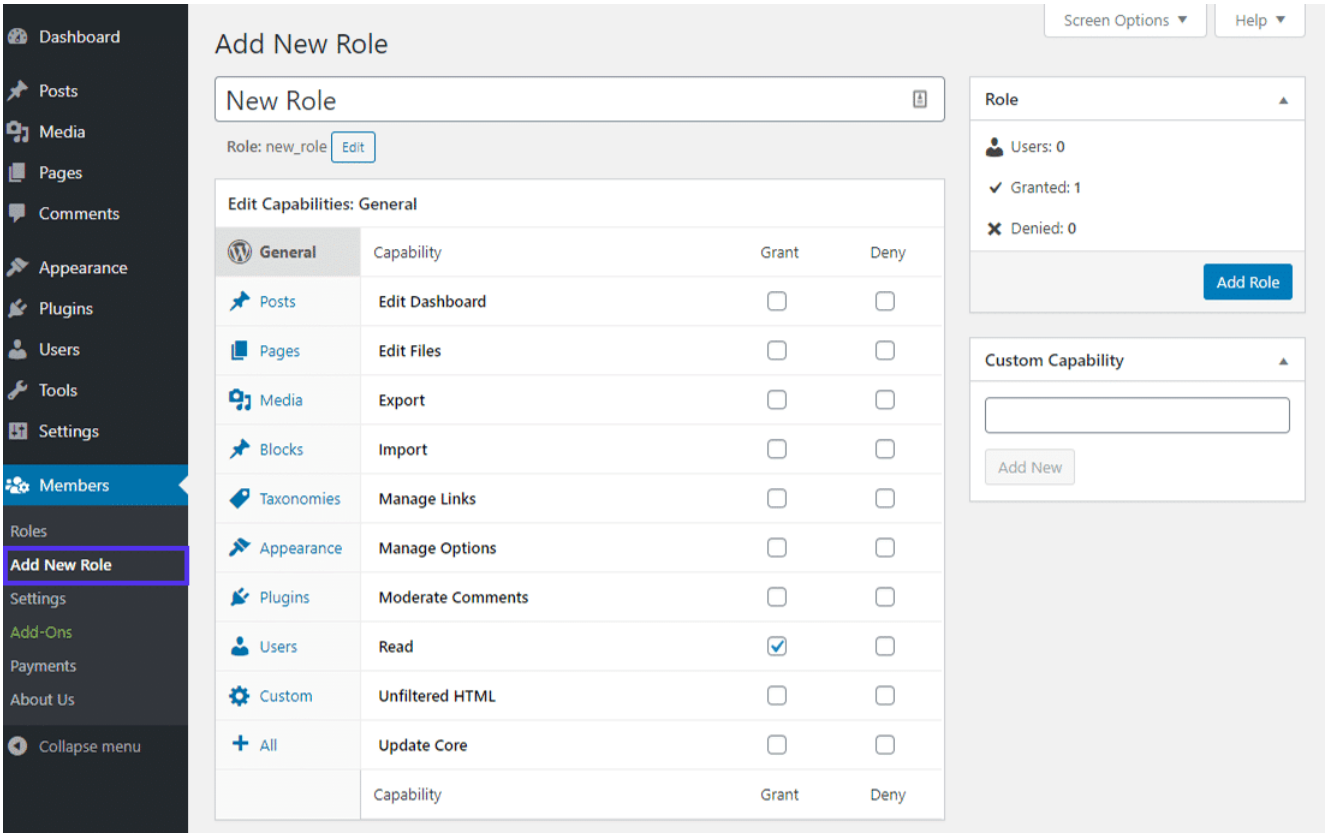
Das Panel ‚Rollen' in Members
Nachdem du das Plugin installiert und aktiviert hast, kannst du alle auf deiner Webseite verfügbaren Rollen einsehen, indem du in deinem Dashboard auf **Mitglieder > Rollen** gehst.

Mitglieder erlaubt es dir, alle Rollen zu löschen, einschließlich der eingebauten WordPress Rollen, außer der Administrator- und der Standardrolle. Du kannst auch Rollen **bearbeiten** und **klonen**, sowie alle Benutzer auflisten, die einer bestimmten Rolle zugeordnet sind.



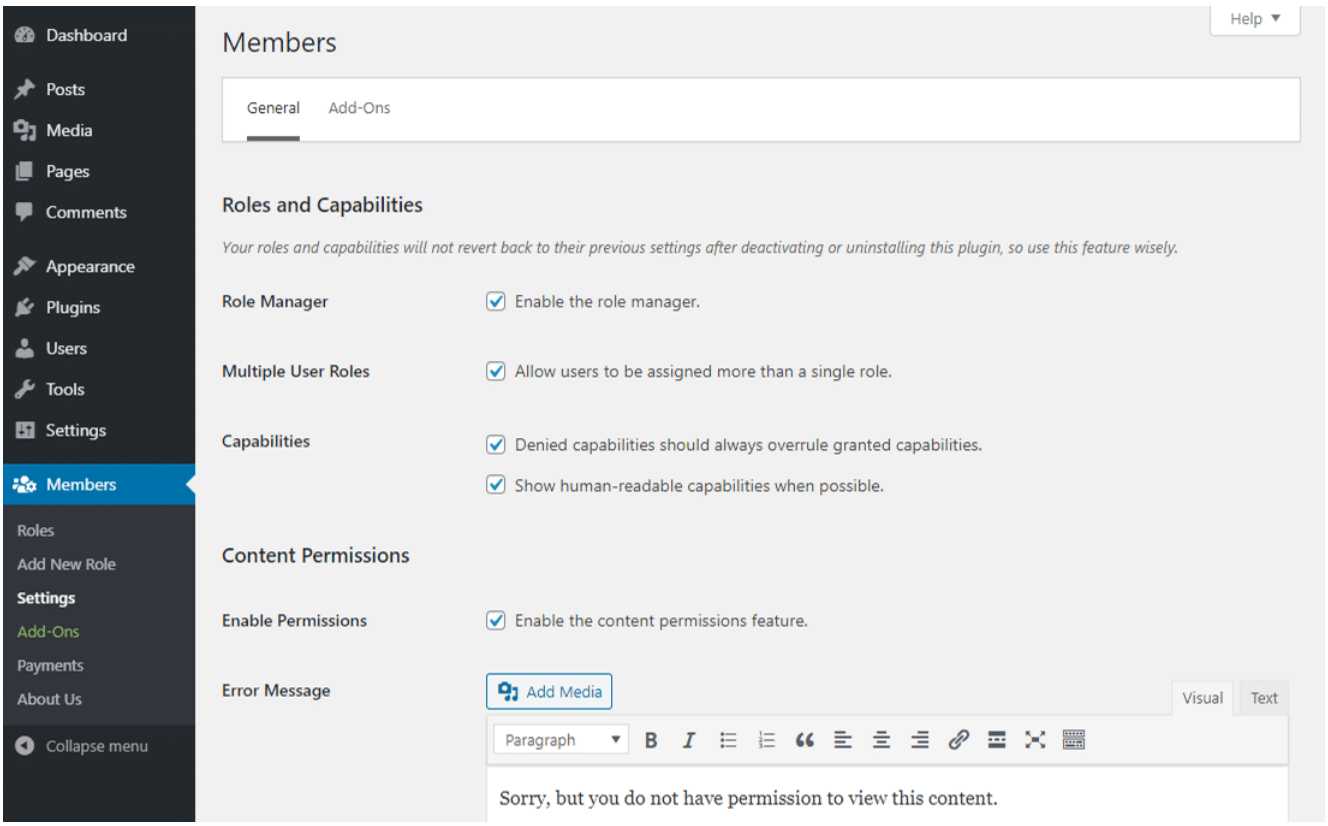
Das Panel ‚Rolle bearbeiten‘ in Members

Innerhalb des **Edit Role** Panels kannst du einer bestimmten Rolle Fähigkeiten gewähren oder explizit verweigern, indem du die entsprechenden Checkboxen an- und abwählst. Von hier aus kannst du auch eine benutzerdefinierte Fähigkeit zu der Rolle hinzufügen.

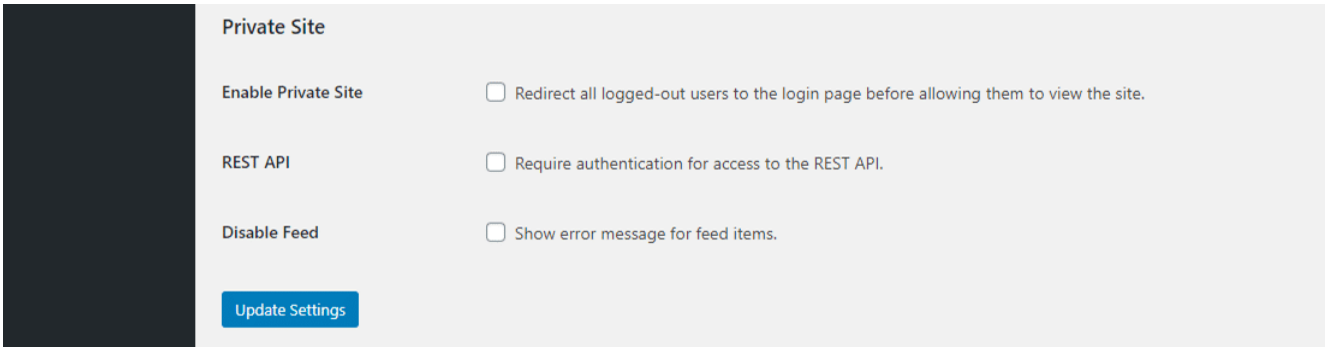


Das Panel ‚Neue Rolle hinzufügen‘ in Members

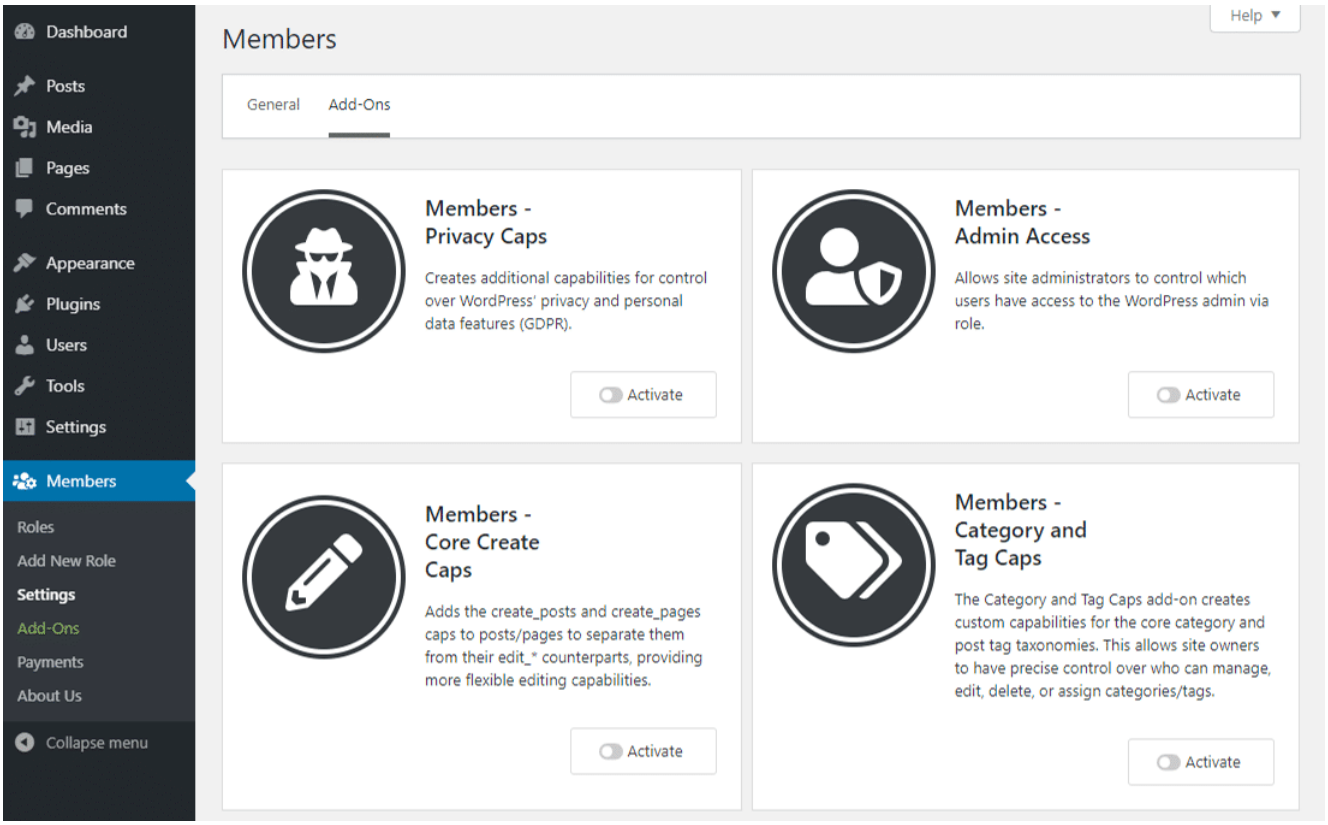
Wenn du auf den Link **Neue Rolle hinzufügen** klickst, gelangst du zu einem ähnlichen Bildschirm, auf dem du eine neue Rolle erstellen kannst, indem du ihr einen Anzeigenamen, eine ID und ihre Fähigkeiten zuweist.



Das Panel ‚Allgemeine Einstellungen‘ in Members
Genau wie im User Role Editor kannst du mit Members den Benutzern mehrere Rollen zuweisen. Du kannst auch Inhaltsberechtigungen festlegen, um Inhalte auf Benutzer mit nur einer bestimmten Rolle zu beschränken.

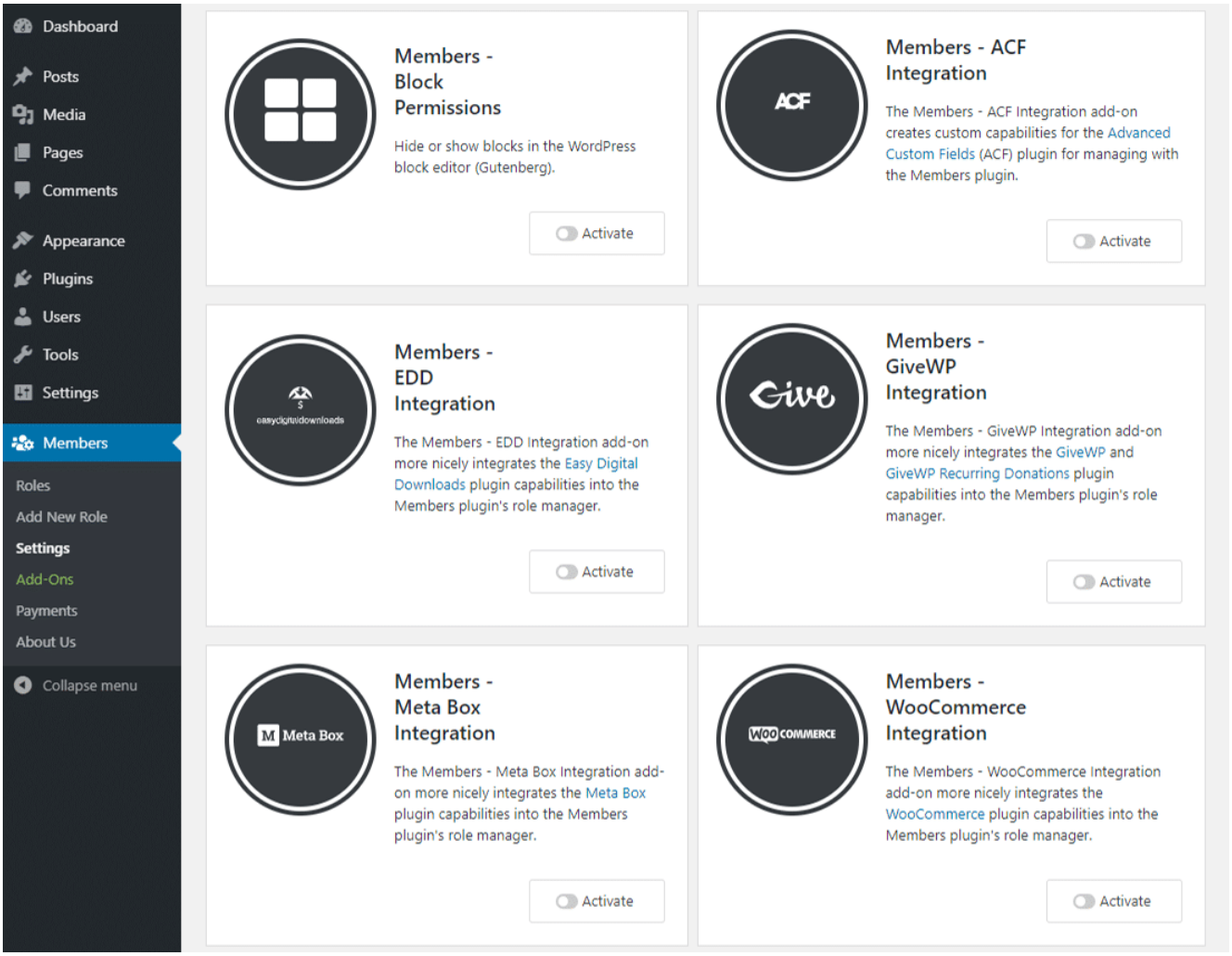


Du kannst auch den Modus ‚Private Webseite‘ unter Mitglieder aktivieren.
Du kannst deine Webseite und ihren Feed als privat einstellen. Außerdem kannst du den Zugriff auf die REST-API von WordPress auf Außenstehende beschränken, indem du die Authentifizierung erzwingst.



Verschiedene Add-ons für das Members-Plugin
Members unterscheidet sich von anderen Rollen- und Fähigkeits-Plugins durch seine erstaunlichen Add-Ons. Sie helfen dir

dabei, deiner Webseite tonnenweise zusätzliche Funktionen hinzuzufügen, wie z.B. die Verwaltung der Privatsphäre und persönlicher Daten (DSGVO), das Hinzufügen von Fähigkeiten in Bezug auf Tags und Kategorien, das Festlegen einer Rollenhierarchie und vieles mehr.



Members integriert mit beliebten WordPress-Plugins

Du kannst Members nahtlos mit vielen beliebten WordPress-Plugins integrieren. Zum Beispiel kannst du es benutzen, um benutzerdefinierte Fähigkeiten für das Advance Custom Fields (ACF) Plugin zu erstellen und zu verwalten. Einige andere Plugins, die es integriert, sind Easy Digital Downloads, GiveWP, Meta Box und WooCommerce.

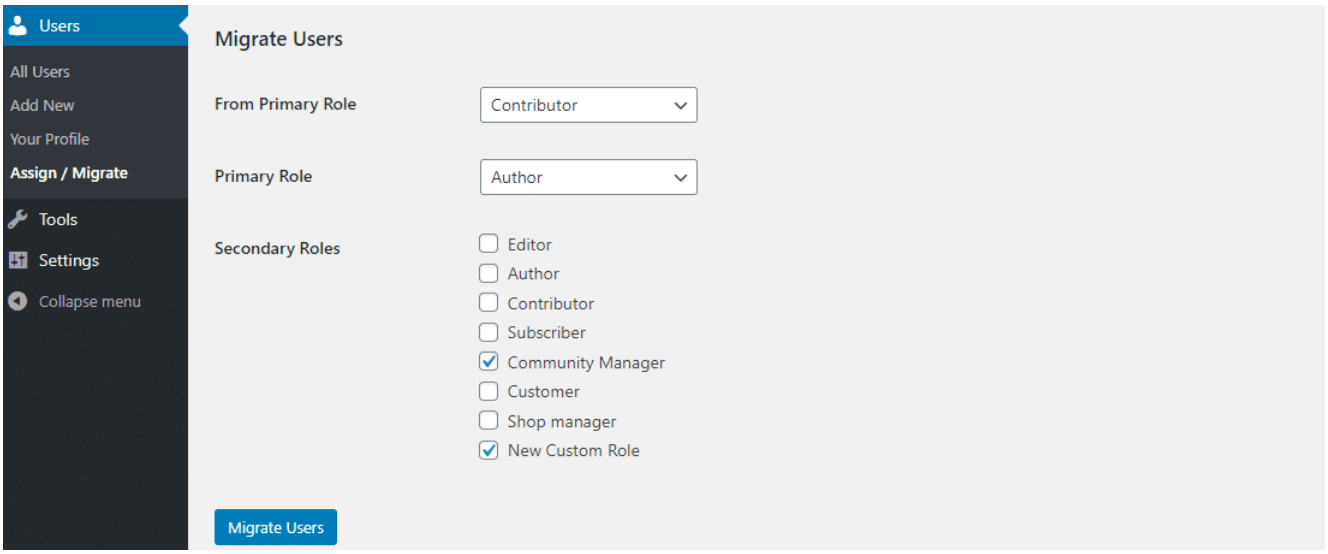
Die auf die Mitgliedschaft ausgerichteten Add-ons der Mitglieder (Zahlungen, Abonnements, E-Mail-Marketing und erweiterter Inhaltsschutz) sind nur in der Premium-Version verfügbar.

WPFront User Role Editor



Das ‚WPFront User Role Editor'-Plugin

Das [WPFront User Role Editor] hilft dir beim Erstellen, Bearbeiten oder Löschen von Benutzerrollen und -fähigkeiten auf deiner WordPress Webseite. Sein Funktionsumfang ist wie bei den zuvor besprochenen Plugins, aber er hat zwei herausragende Funktionen.

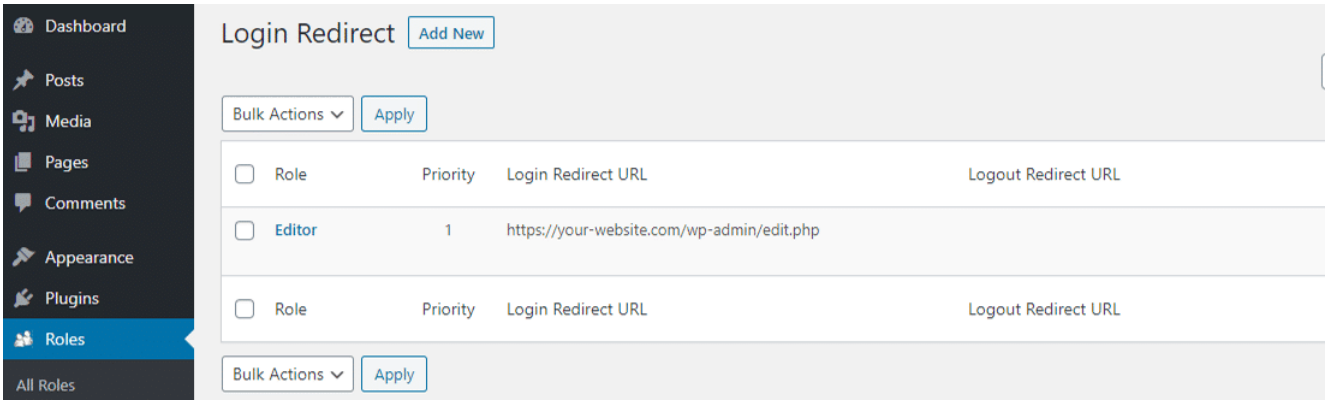


Migriere alle Benutzer von einer Rolle in eine andere

Nachdem du den WPFront User Role Editor installiert und aktiviert hast, kannst du in deinem Admin-Dashboard auf den Bildschirm **Users > Assign / Migrate** gehen und alle Benutzer, die zu einer bestimmten Benutzerrolle gehören, auf eine andere migrieren. Du kannst deinen Benutzern sogar sekundäre Rollen zuweisen.

Wenn du viele Benutzer auf deiner Webseite von einer Rolle zu einer anderen migrieren musst, ist diese Funktion sehr

nützlich.



Der Einstellungsbildschirm ‚Login Redirect‘ im WPFront User Role Editor

Eine weitere hilfreiche Funktion des WPFront User Role Editors ist der rollenbasierte **Login-Redirect**. Du kannst zum Beispiel Benutzer mit der Redakteur-Rolle nach dem Einloggen auf die **Beitragsseite** umleiten. Du hast auch die Möglichkeit, ihnen den Zugriff auf die **/wp-admin-Seite** und die Anzeige der Symbolleiste im Frontend zu verwehren.

Advanced Access Manager

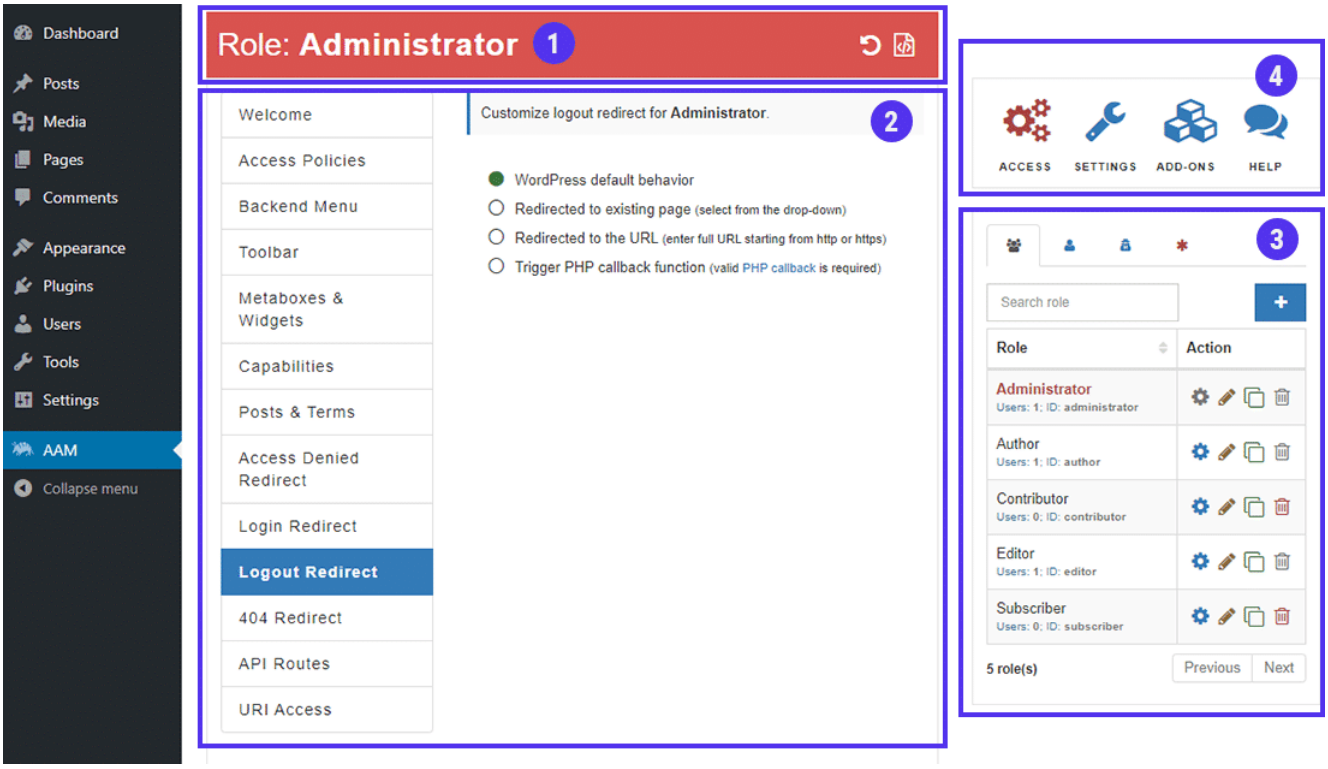


Das “Advanced Access Manager” Plugin

Advanced Access Manager (AAM) ist ein leistungsstarkes WordPress-Plugin, mit dem du fast jeden Aspekt deiner Webseite kontrollieren kannst. Es enthält über 200 verschiedene Funktionen und ist für fortgeschrittene WordPress-Benutzer gedacht, die wissen, wie Rollen und Fähigkeiten funktionieren.

Im Vergleich zu den oben aufgeführten Plugins hat AAM weit mehr Funktionen. Aber da es ein Entwickler-orientiertes Plugin

ist, ist es für Anfänger oder fortgeschrittene Benutzer nicht so einfach zu benutzen.

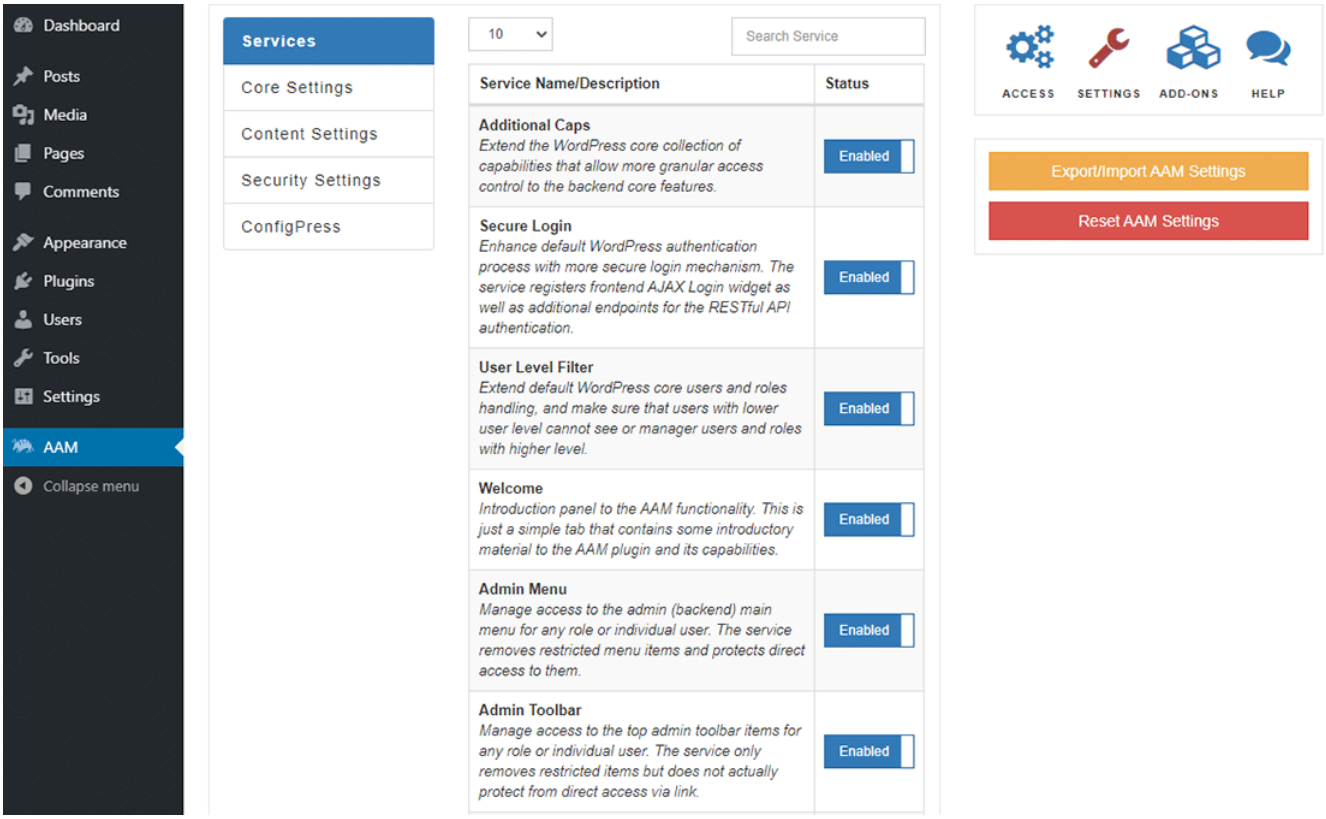


Das Haupt-Dashboard im Advanced Access Manager

Du kannst das Haupt-Dashboard des AAM in vier verschiedene Regionen aufteilen. Ich habe sie in der obigen Abbildung nummeriert und eine Übersicht darunter.

1. Der oberste Bereich erwähnt das aktuelle ‚Thema', das gerade behandelt wird. Hier ist es **‚Role': Administrator**, aber es kann ein bestimmter Benutzer, ein anonymer Besucher oder eine Standardeinstellung für alle sein.
2. Der Bereich unter dem Betreff ist das Hauptfenster, in dem du alle Einstellungen hast, um den Zugang zu verschiedenen Dingen auf deiner Webseite für den Betreff zu verwalten.
3. Der dritte Bereich ist der **Benutzer/Rollenmanager**. Über seine Registerkarten-Icons kannst du auswählen, was du verwalten möchtest. Ist es eine Benutzerrolle, ein bestimmter Benutzer, ein anonymer Besucher oder ein Standardzugriffsverhalten für alle?
4. Im vierten Bereich kannst du die Einstellungen von AAM

verwalten, seine Premium-Add-ons installieren und den Support kontaktieren.

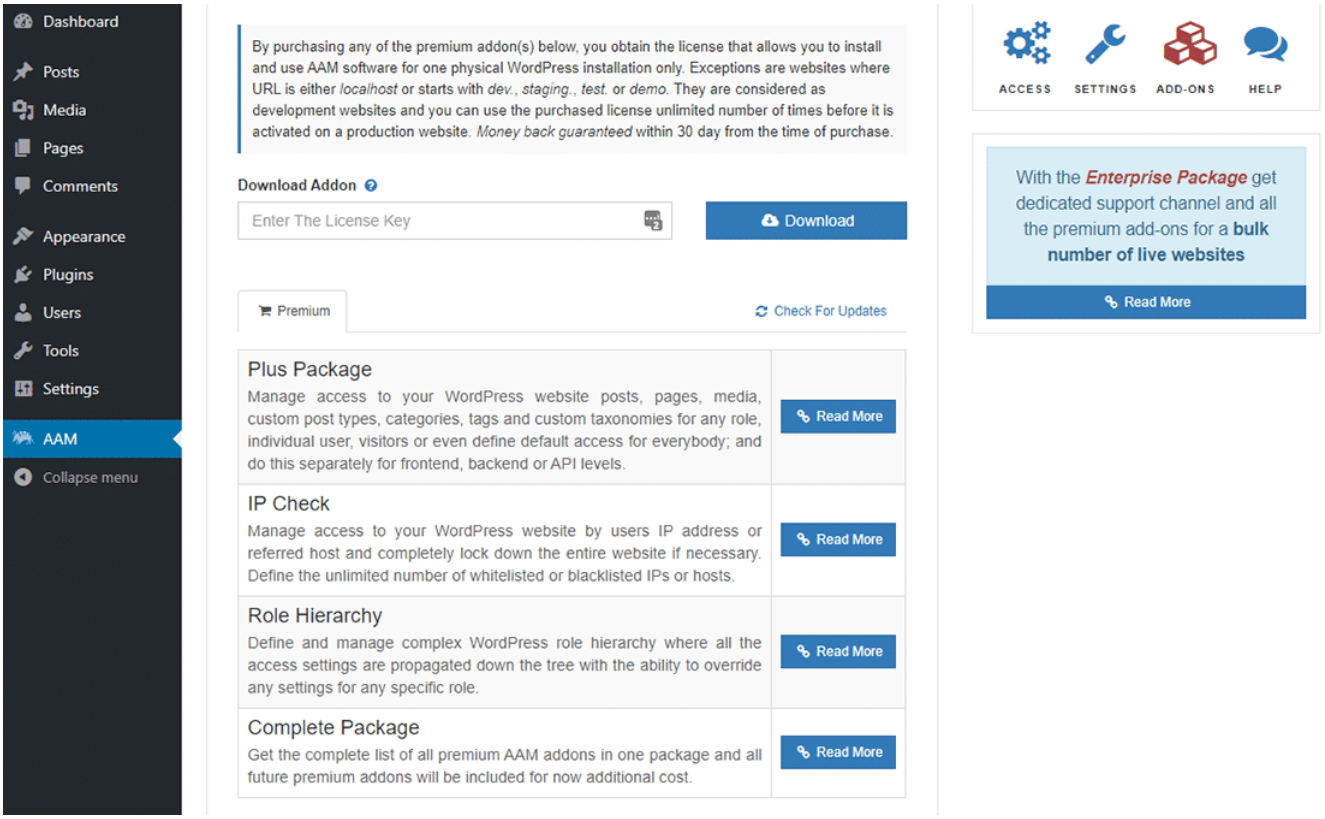


Das ‚Einstellungen'-Panel im Advanced Access Manager

AAM gliedert seine Einstellungen in 5 Gruppen, je nach ihrem Verhalten und ihrer Nutzung.

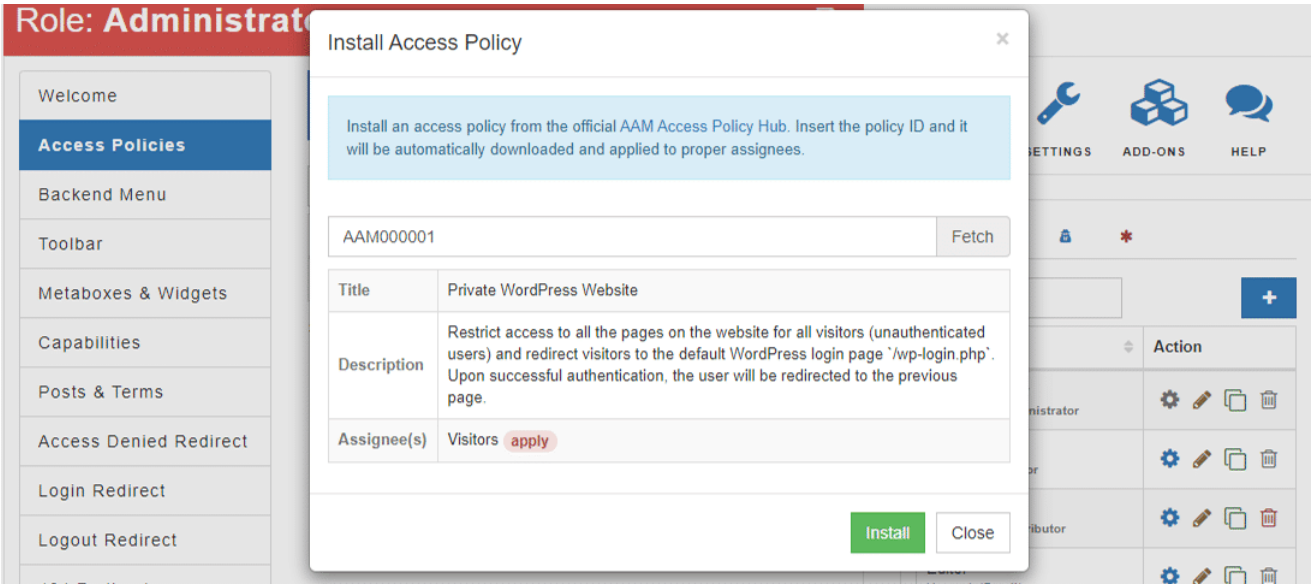
- In den Einstellungen der **Services** sind alle Module von AAM aufgelistet, die du aktivieren oder deaktivieren kannst. Wenn du die Module selektiv lädst, kannst du deine Webseite optimiert halten.
- Im Bereich **Kerneinstellungen** kannst du einige der Kernfunktionen von AAM und WordPress aktivieren oder deaktivieren.
- Die **Inhaltseinstellungen** beziehen sich auf den Inhalt der Webseite (z.B. Beiträge, Seiten, benutzerdefinierte Beitragstypen).
- Der Bereich **Sicherheitseinstellungen** enthält Einstellungen für die sichere Login-Funktion von AAM. Momentan sind nur zwei Einstellungen verfügbar: **Brute Force Lockout** und **eine Session pro Benutzer**.
- ConfigPress ist eine interessante Funktion, mit der man

[die Konfiguration des AAM-Plugins mit INI-basiertem Code ändern kann](#).



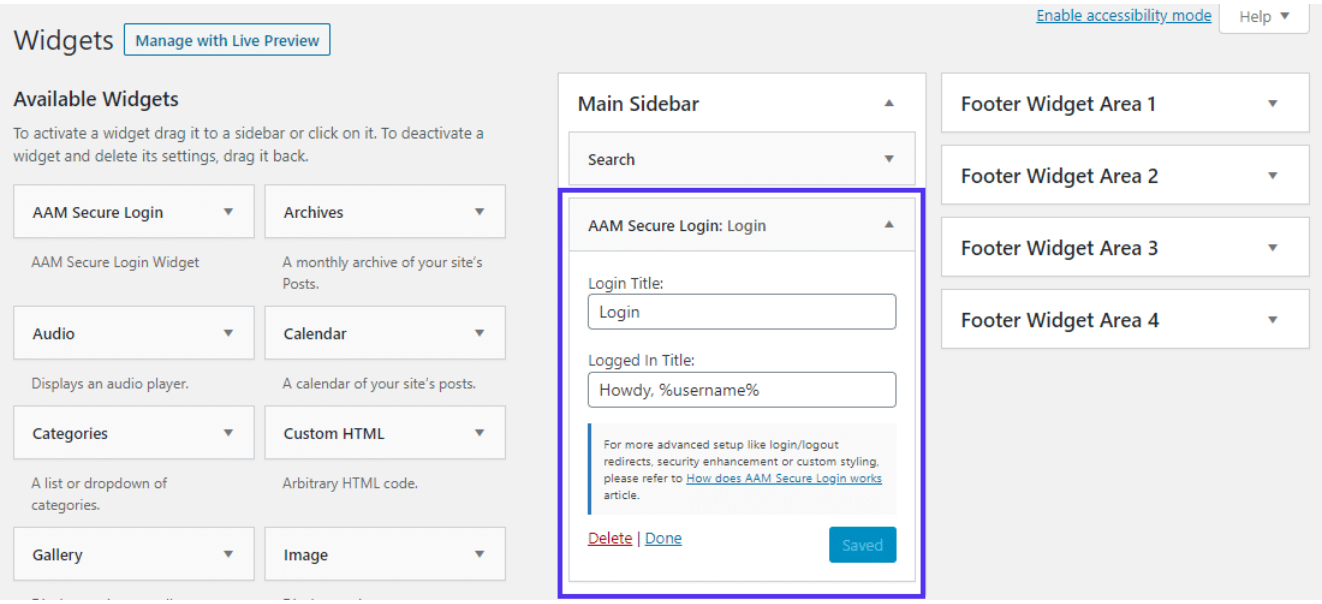
Das ‚Add-ons'-Panel im Advanced Access Manager

AAM ist ein entwicklerorientiertes Plugin, das über die reinen Benutzerrollen und Fähigkeiten hinausgeht. Es gibt Ihnen eine granulare Kontrolle darüber, was jede Rolle auf Ihrer Webseite tun oder nicht tun kann.



Installiere eine ‚Zugangsrichtlinie' für deine Webseite, um sie sicher zu halten

Du kannst AAM benutzen, um eine Zugangs- und Sicherheitsrichtlinie für deine Webseite festzulegen. Sie legt fest, welche Rolle und unter welchen Bedingungen auf verschiedene Ressourcen auf deiner Webseite zugreifen kann. Wenn du sofort loslegen möchtest, kannst du eine sofort einsatzbereite Zugriffsrichtlinie vom AAM Access Policy Hub installieren.



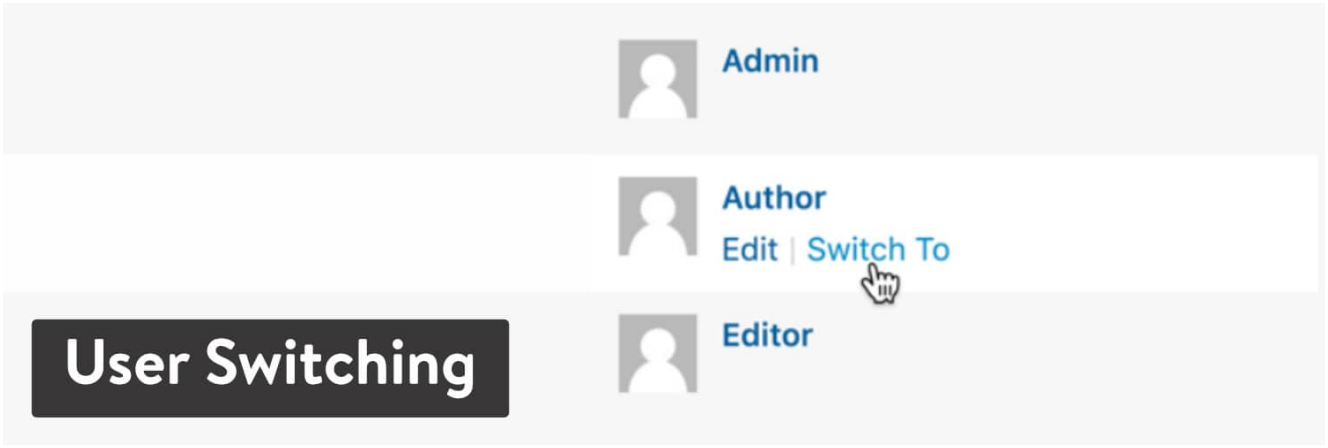
“AAM Secure Login”-Widget zum Hinzufügen eines Frontend-Login-Formulars

Mit AAM kannst du temporäre Benutzerkonten und Rollen erstellen. Es ist ein sicherer Weg, einen Account mit externen Ressourcen zu teilen. Temporäre Benutzerkonten verfallen nach dem Datum und der Uhrzeit, die du festgelegt hast. Bei temporären Rollen wird dem Benutzer nach dem festgelegten Zeitraum die Rolle entzogen.

Es liegt außerhalb der Reichweite dieses Artikels, alle Funktionen von AAM abzudecken. Du kannst in der Dokumentation des Advanced Access Managers nachschlagen, um mehr über all seine umfangreichen Funktionen zu erfahren.

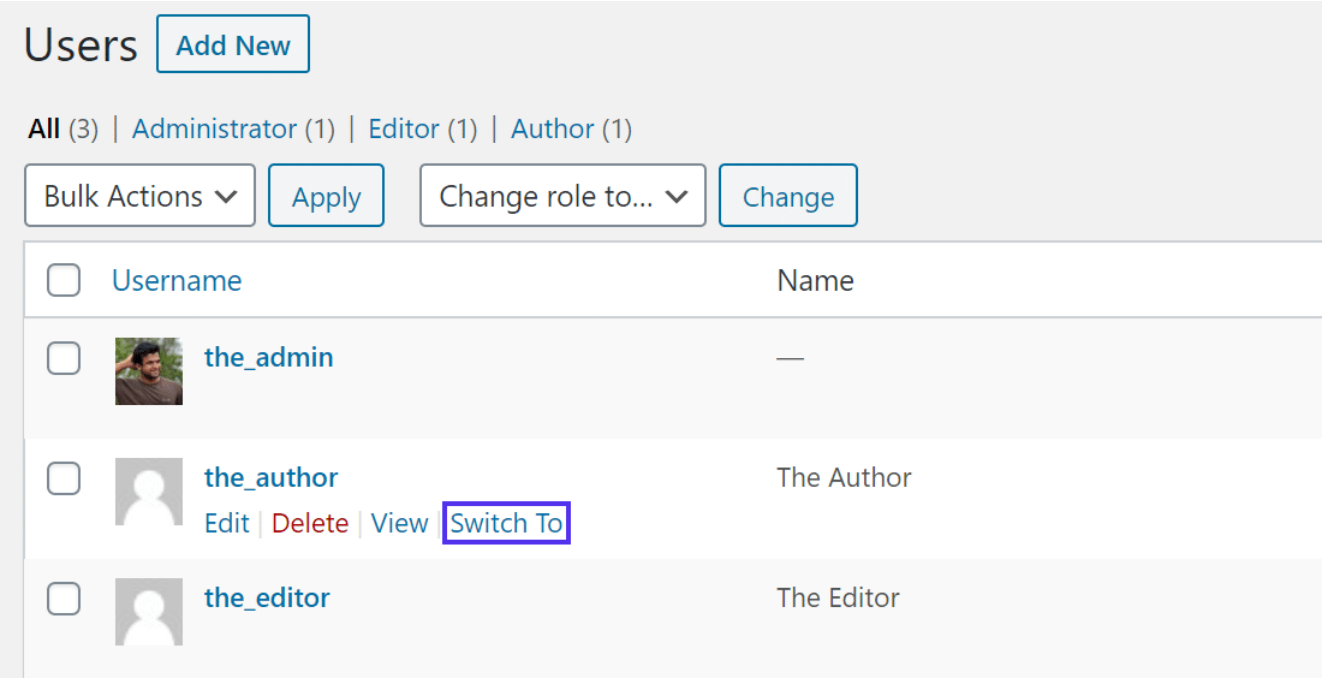
Tipp: Der User Access Manager ist eine vernünftige Alternative zum Advanced Access Manager, obwohl er weniger Features hat und nicht häufig aktualisiert wird.

User Switching



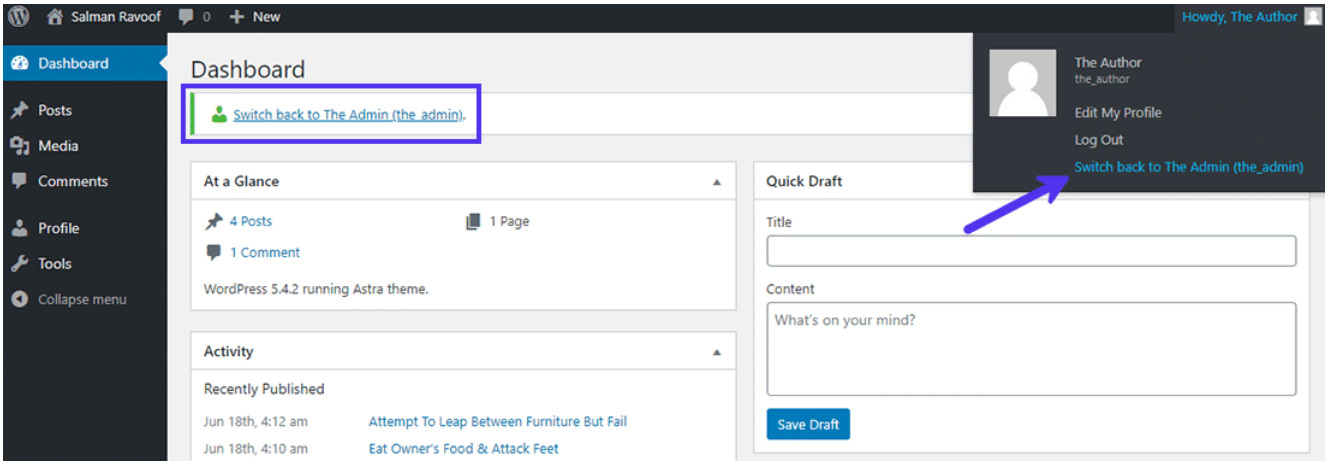
Das “User Switching” WordPress Plugin
Mit User Switching kannst du mit nur einem Klick zwischen verschiedenen WordPress-Benutzerkonten hin- und herwechseln. Wenn du viele Benutzerrollen und Fähigkeiten ausprobierst, kannst du mit diesem Plugin eine Menge Zeit sparen. User Switching verwendet das eingebaute Cookie-Authentifizierungssystem von WordPress, um sich die Accounts, von denen du gewechselt hast, zu merken, so dass du sofort wieder zu ihnen zurückwechseln kannst.

Nachdem du das Plugin installiert und aktiviert hast, besuche das **Benutzermenü** in deinem Dashboard. Du wirst einen **Switch To** Link für jeden Benutzer sehen. Wenn du darauf klickst, wechselst du zu dem gewünschten Benutzer.



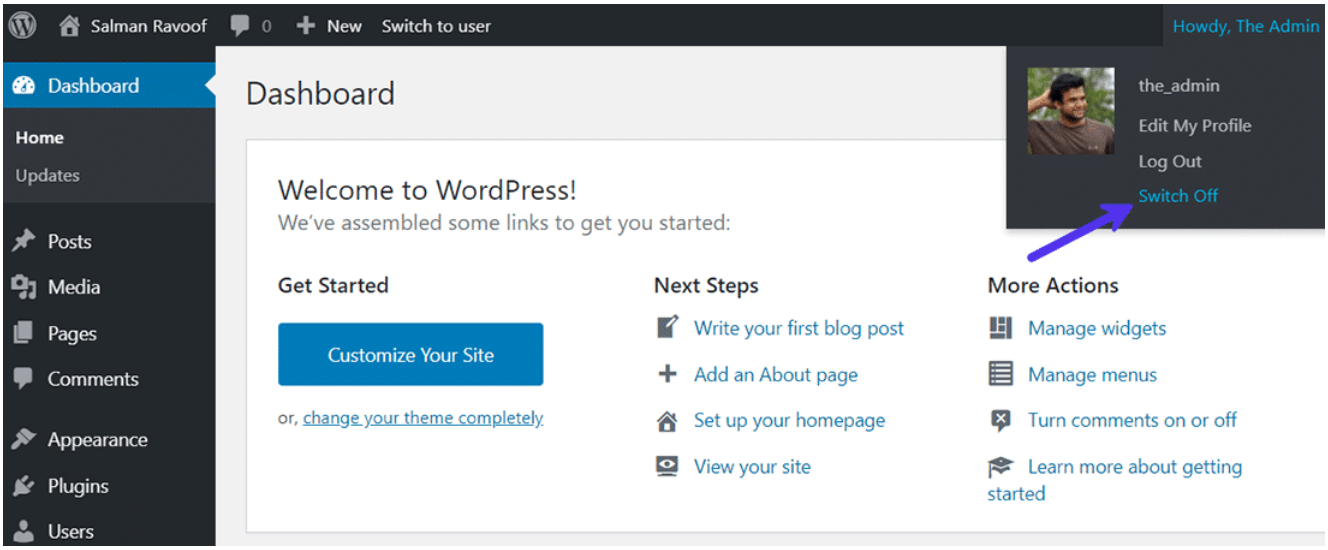
Klicke auf den Link ‚Switch To‘, um zum gewünschten Benutzer zu wechseln.

Du kannst wieder zu deinem ursprünglichen Konto wechseln, indem du auf den Link **Switch back to** im Dashboard oder in deinem Benutzerprofil-Bildschirm klickst.



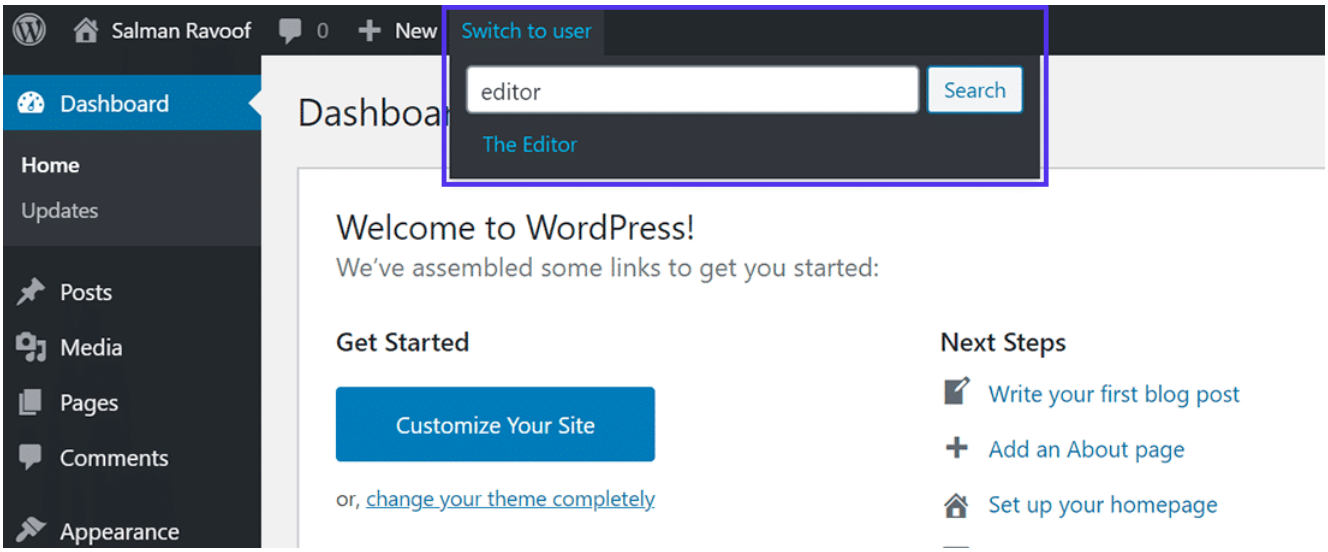
Du kannst ganz einfach zu deinem ursprünglichen Account zurückwechseln

Du kannst deinen Administrator-Account auch vorübergehend **ausschalten**, um zu sehen, wie dein Frontend den Besuchern erscheint.



Schalte deinen Account mit einem einzigen Klick ein und aus
Aus Sicherheitsgründen können nur Benutzer, die die Möglichkeit haben, Benutzer zu bearbeiten, die Benutzerkonten wechseln. Standardmäßig haben nur Administratoren diese Fähigkeit in einer WordPress Single-Site-Installation, während in einem Multisite-Netzwerk nur Super-Admins diese Fähigkeit haben.

Um den Benutzerwechsel weiter zu vereinfachen, kannst du die Erweiterung [Admin Bar User Switching] installieren, damit der Link **Switch to user** in deiner Admin Bar erscheint.



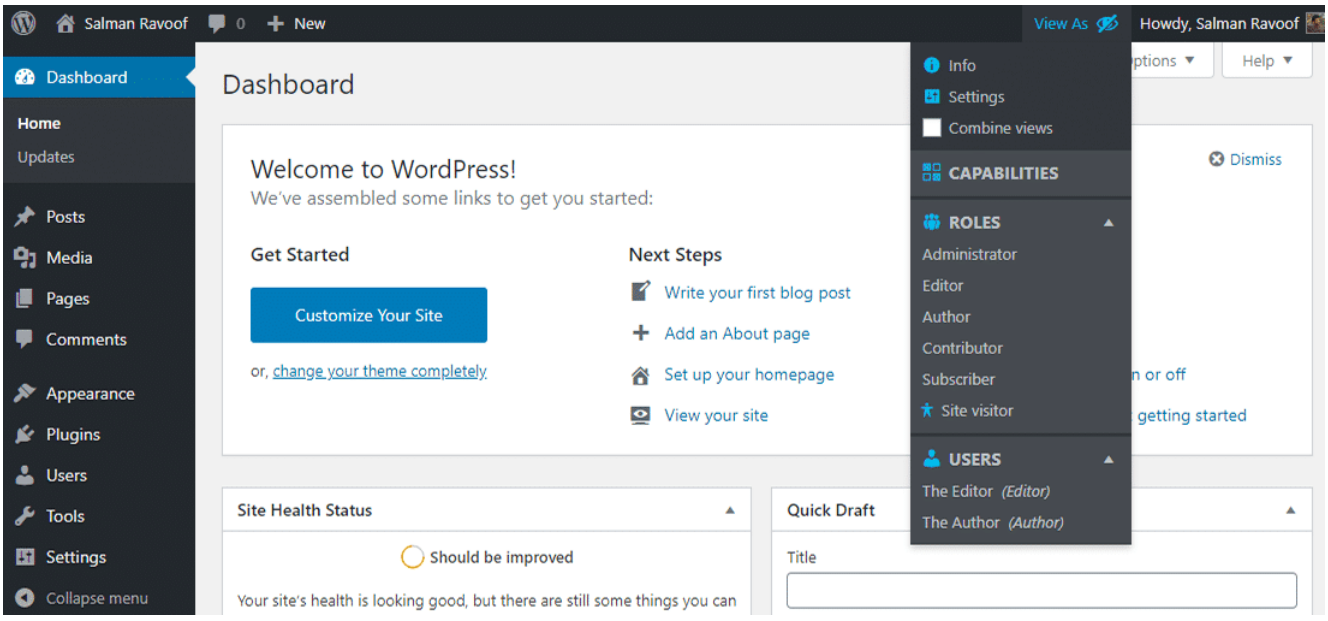
Den Link ‚Zum Benutzer wechseln' zu deiner Admin-Leiste hinzufügen

View Admin As



Das ‚View Admin As' WordPress-Plugin

[View Admin As](#) ist ein fortgeschrittenes Plugin zum Benutzerwechsel, das auch einen Rollen- und Fähigkeitsmanager enthält. Anders als beim User Switching Plugin musst du keine Erweiterung installieren, um das User Switching Menü zu deiner Admin-Leiste hinzuzufügen. View Admin As fügt standardmäßig alle seine Hauptmenüpunkte zur Admin-Leiste hinzu.

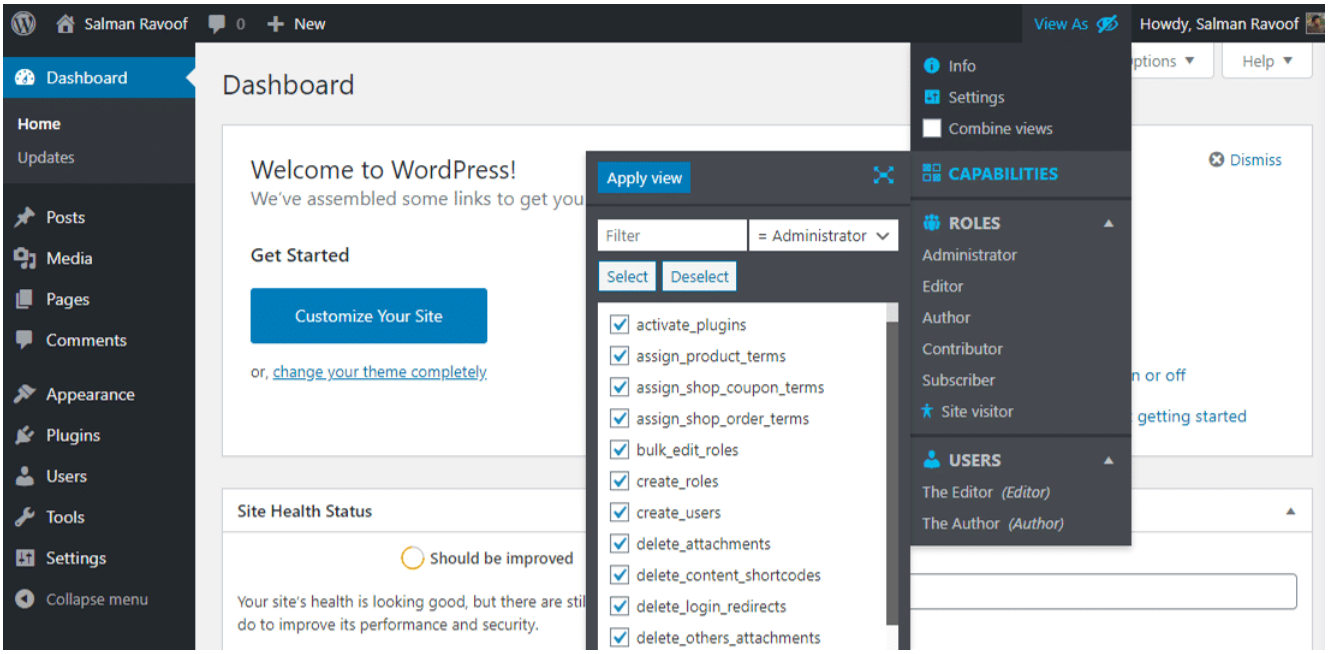


Das ‚View As'-Menü in der Verwaltungsleiste

Du kannst zwischen bestehenden Benutzern oder Rollen wechseln (indem du ihre Fähigkeiten übernimmst), auch wenn kein Benutzer mit diesen Rollen existiert. Wenn du auf den **Besucher-Link** der Webseite klickst, gelangst du zum Frontend der Webseite, wo du als normaler Benutzer die Funktionalität der Webseite testen kannst, ohne die Registerkarte deines

Browsers zu verlassen.

View Admin As erlaubt dir, deine eigenen Fähigkeiten vorübergehend zu ändern. Da dies nicht-destruktiv geschieht, verlierst du nicht den Zugang zu deinen Hauptfähigkeiten.

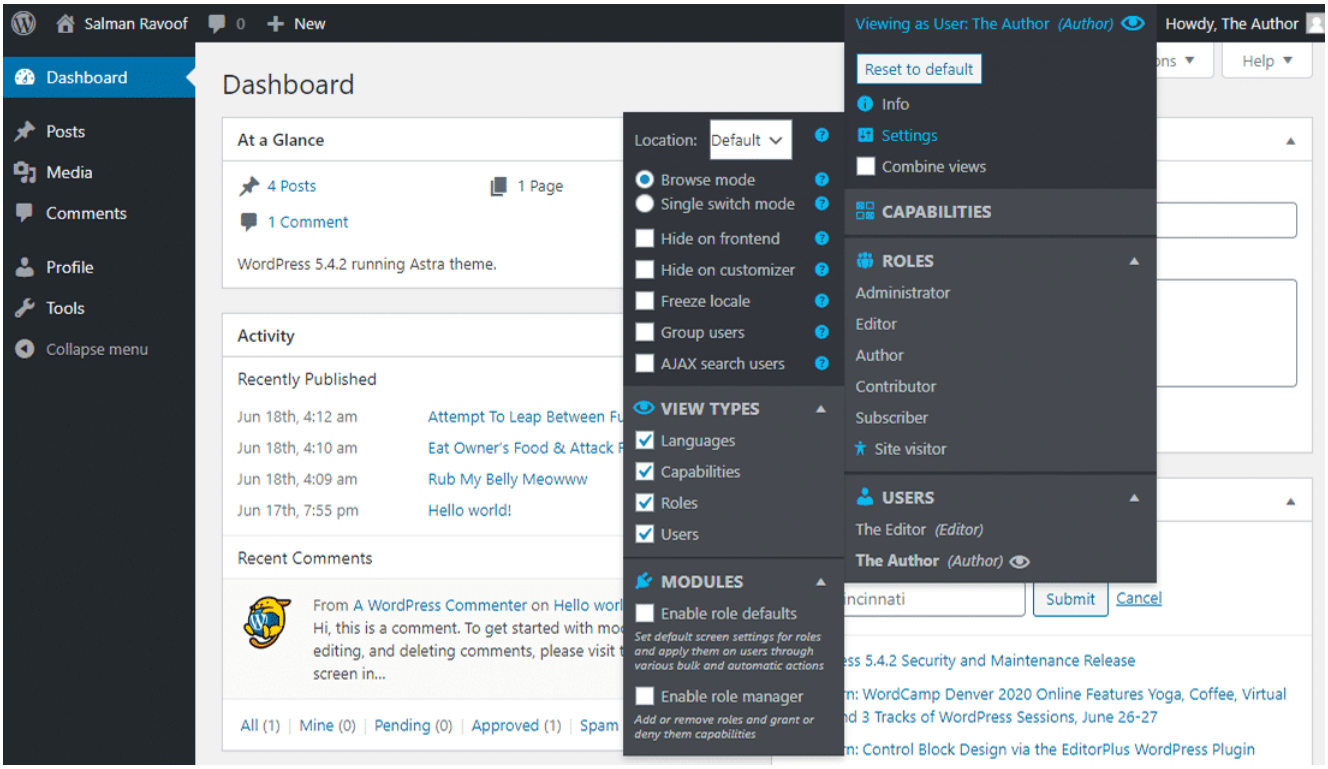


Passe die Fähigkeiten vorübergehend für deinen aktuellen Benutzer an

Nachdem du zu einem Benutzerkonto gewechselt hast, kannst du deren Bildschirmeinstellungen und Einstellungen direkt über das Menü bearbeiten. Du kannst auch die Sprache/Lokalität auf dem Frontend und dem Backend separat umschalten.

Du bist nicht auf einen Ansichtstyp beschränkt, da du die verschiedenen Optionen kombinieren und alle gleichzeitig anwenden kannst.

View Admin As kommt mit zwei optionalen Modulen, die du bei Bedarf aktivieren kannst.



View Admin As Einstellungen und optionale Module ansehen

Das erste Modul fügt die Funktion **„Role Defaults“** hinzu, mit der du Standardbildschirmeinstellungen für alle Rollen festlegen kannst. Du kannst diese Standardeinstellungen auf eine Rolle, einen einzelnen Benutzer oder zukünftige neue Benutzer anwenden.

Das zweite Modul aktiviert die **Rollenmanager**-Funktionalität. Alle Änderungen, die du mit diesem Modul an Rollen und Fähigkeiten vornimmst, sind dauerhaft. Im Gegensatz zu anderen Rolleneditor-Plugins kannst du mit diesem Modul die einem Benutzer zugewiesene Rolle löschen, indem du sie automatisch auf eine andere Rolle überträgst.

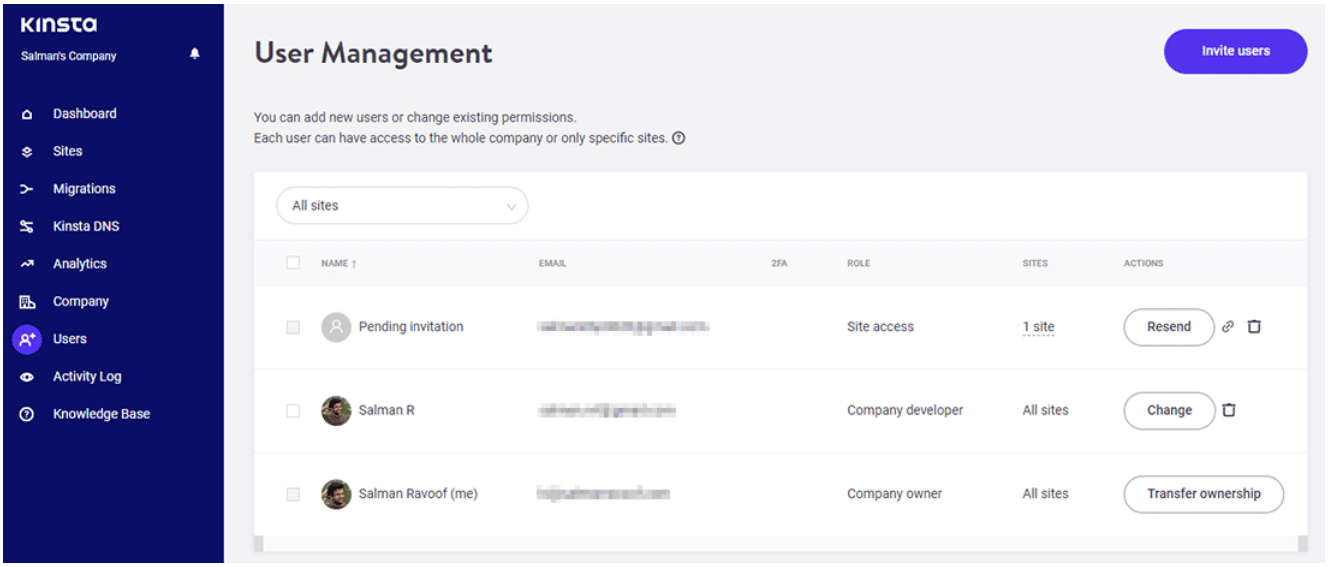
Du kannst in der Dokumentation View Admin As nachschlagen, um mehr über seine umfangreichen Funktionen zu erfahren.

MyKinsta Benutzer-Rollen

Die Multi-User-Funktion von MyKinsta ermöglicht es dir, mehrere Benutzer unter demselben Account zu erstellen und zu verwalten, indem du ihnen Zugang zu einzigartigen Aspekten deines Kinsta-Accounts oder bestimmter Webseiten, die bei

Kinsta gehostet werden, gibst.

Es gibt verschiedene Rollen, aus denen du wählen kannst, um den Benutzerzugang nach deinen Bedürfnissen anzupassen.



Der Bildschirm ‚Benutzerverwaltung' im MyKinsta-Dashboard
Der erste Benutzer erhält standardmäßig die Rolle des Firmeninhabers. Es ist die mächtigste Rolle und beinhaltet auch alle Fähigkeiten eines Firmenadministrators.

Es kann immer nur einen Firmeninhaber zur gleichen Zeit geben, aber du kannst die Rolle bei Bedarf auf einen anderen Firmenadministrator übertragen. Auf diese Weise überträgst du auch den Besitz deines Kinsta-Accounts auf den neuen Firmeninhaber.

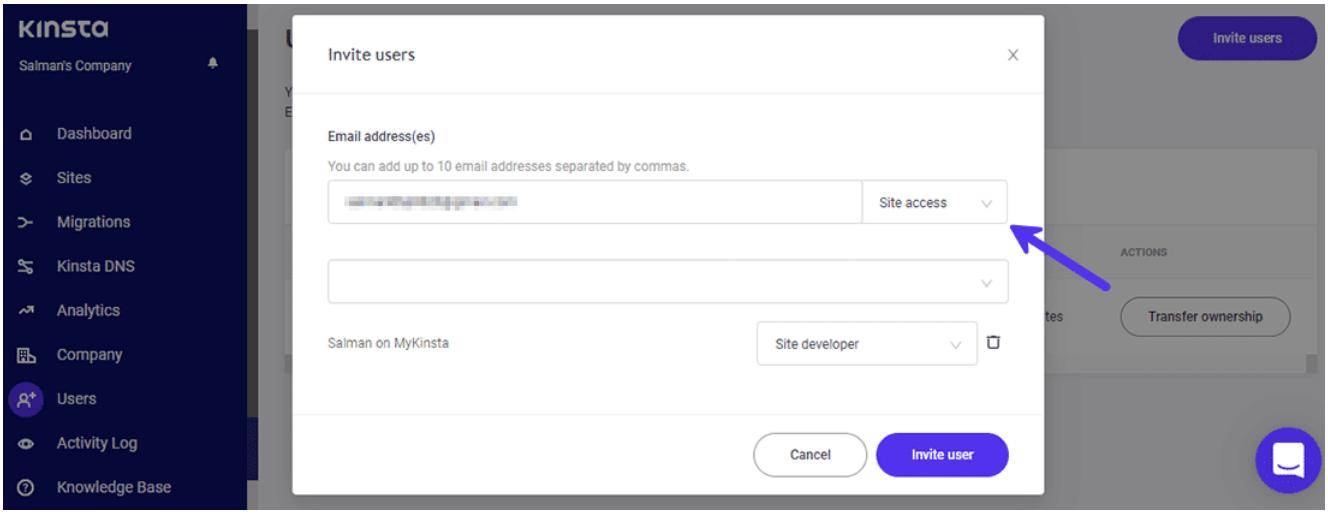
Nur der Firmeninhaber kann Kinsta auffordern, den Account zu löschen.

Du kannst die anderen Benutzerrollen in 2 Hauptrollenkategorien einteilen:

- Firmen-Level
- Webseiten-Level

Rollen auf Unternehmensebene geben den Benutzern Zugriff auf die Details des Kinsta-Accounts auf Unternehmensebene, während Rollen auf Standortebene den Benutzern nur Zugriff auf

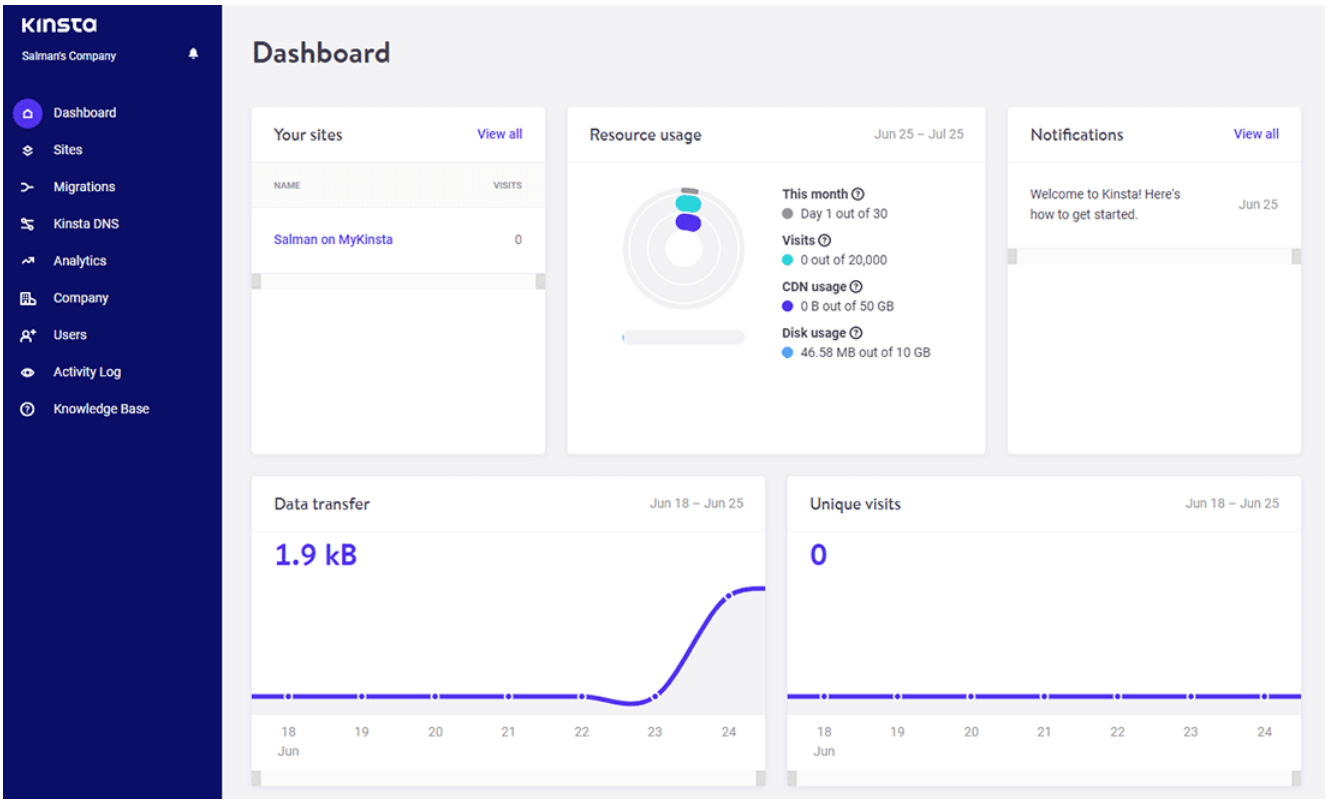
bestimmte Webseiten geben, die ihnen zugewiesen wurden. Wenn du einen neuen Benutzer einlädst oder einen bestehenden modifizierst, musst du als erstes entscheiden, ob du ihm Firmen- oder Website-Zugriff gewähren möchtest.



Auswahl des Rollentyps, wenn du einen Benutzer zu MyKinsta einlädst

Firmen-Level Rollen

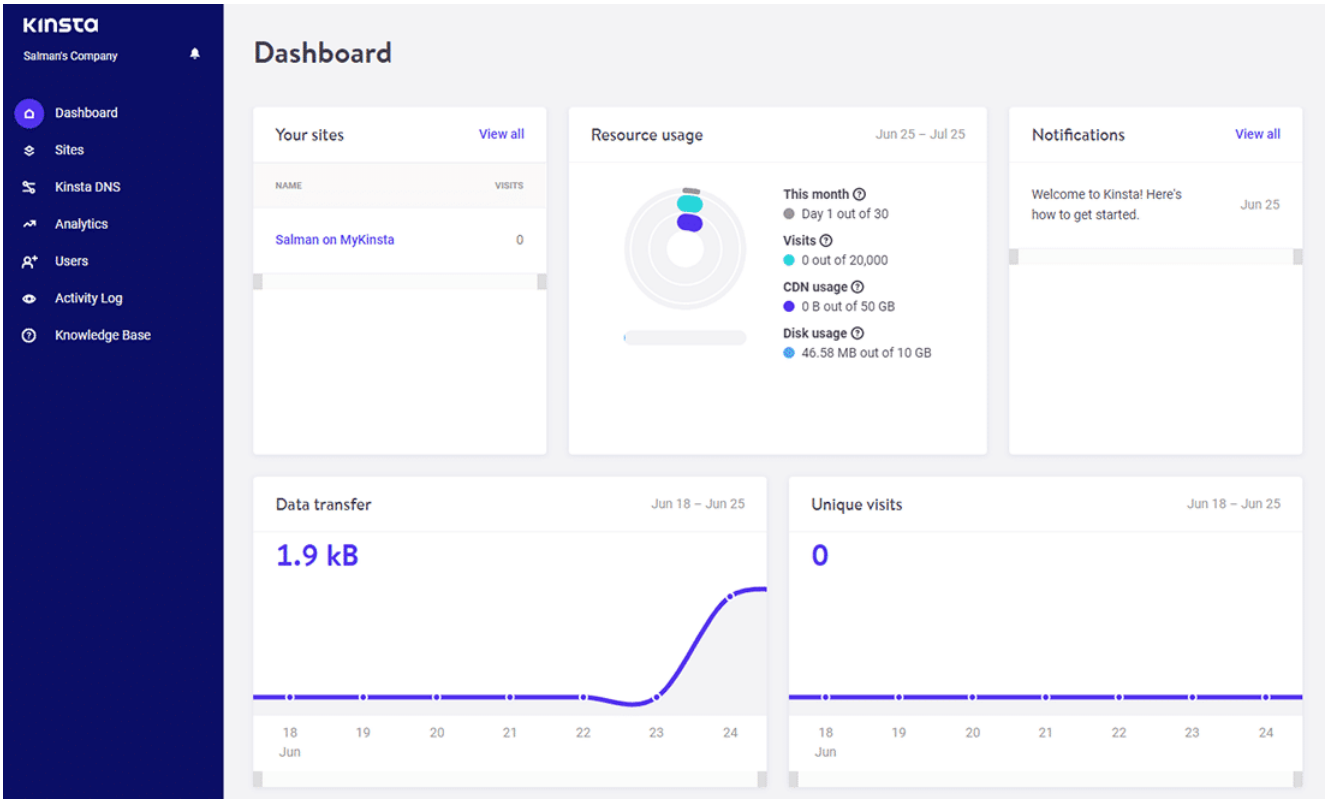
Firmenadministrator



Das ‚Firmenadministrator'-Dashboard in MyKinsta

Die Rolle des Firmenadministrators gewährt die höchste Zugangsstufe in MyKinsta. Sie gibt dem Benutzer die vollständige Kontrolle über den Kinsta-Account und all seine Webseiten. Du solltest diese Rolle nur an Benutzer vergeben, denen du vertraust.

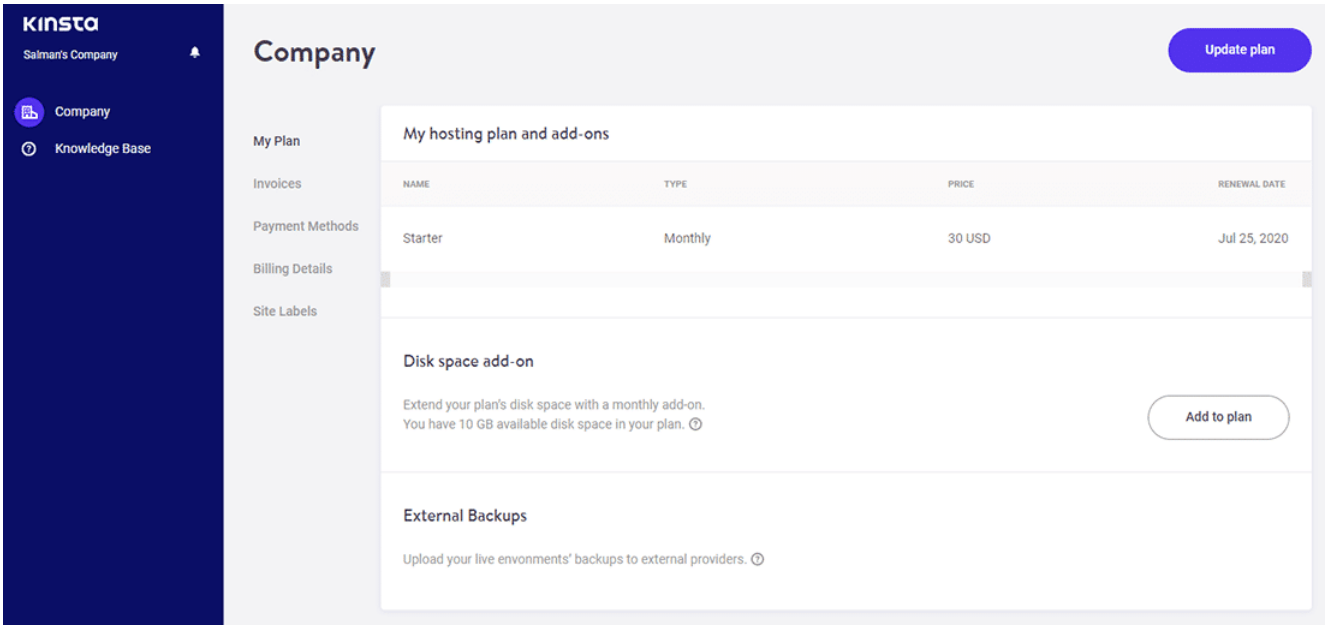
Firmenentwickler



Das ‚Firmenentwickler'-Dashboard in MyKinsta

Die Rolle des Firmenentwicklers gewährt Zugang zur Verwaltung aller Webseiten, einschließlich deren Löschung. Da die MyKinsta-Benutzerrollen hierarchisch aufgebaut sind, kann ein Firmenentwickler auch Site-Level-Benutzer verwalten. Allerdings kann ein Firmenentwickler nicht auf Firmeneinstellungen oder Rechnungsdetails zugreifen.

Firmenabrechnung

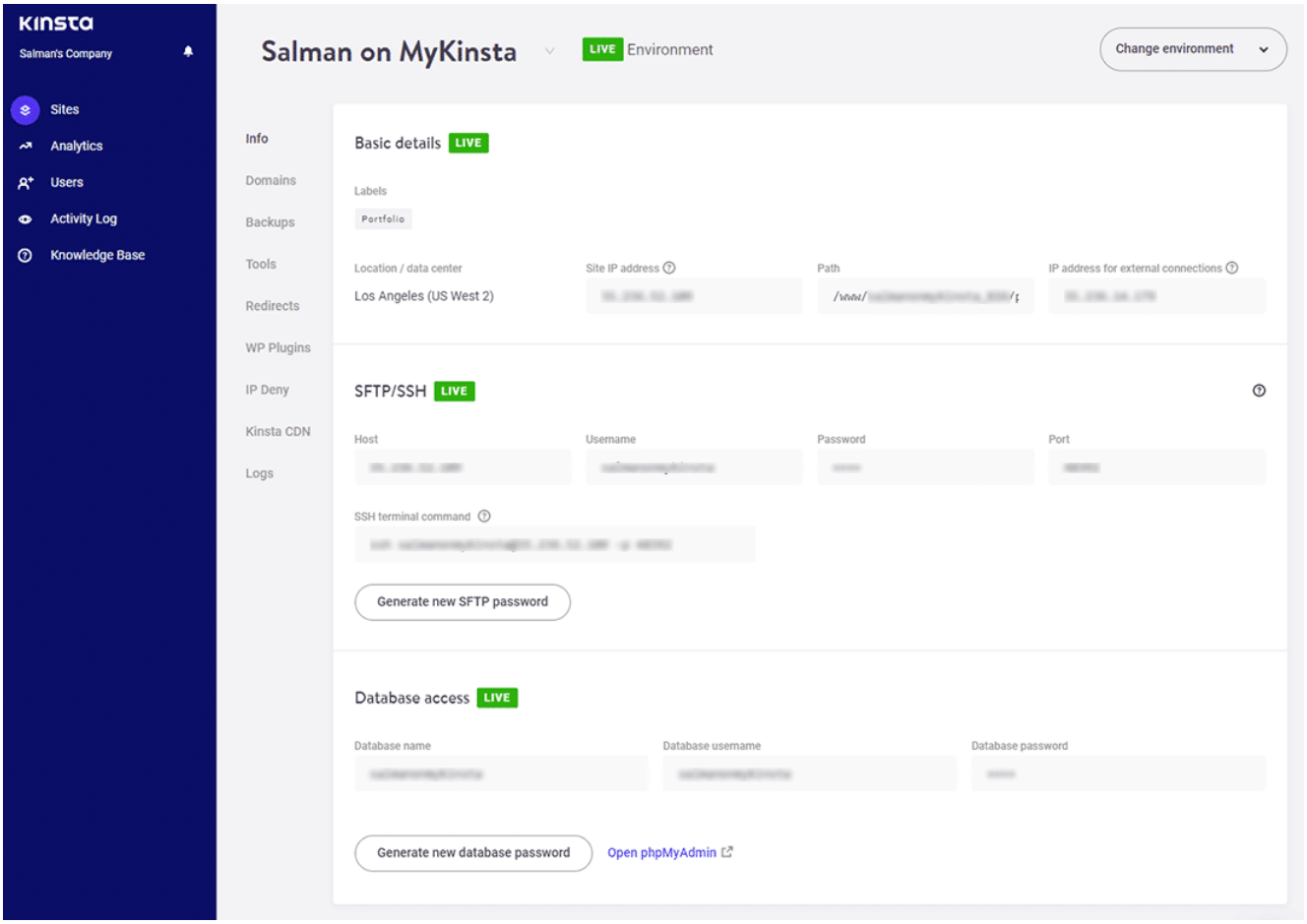


Das ‚Firmenabrechnung'-Dashboard in MyKinsta

Die Rolle „Firmenabrechnung" gewährt nur Zugang zu den Rechnungsdetails und Firmeneinstellungen. Sie haben keinen Zugang zu irgendwelchen Webseiten. Benutzer mit der Rolle „Firmenabrechnung" können Rechnungen überprüfen, automatische Rechnungs-E-Mails aktivieren und Firmendetails wie Adresse und Kontaktinformationen ändern.

Webseiten-Level Rollen

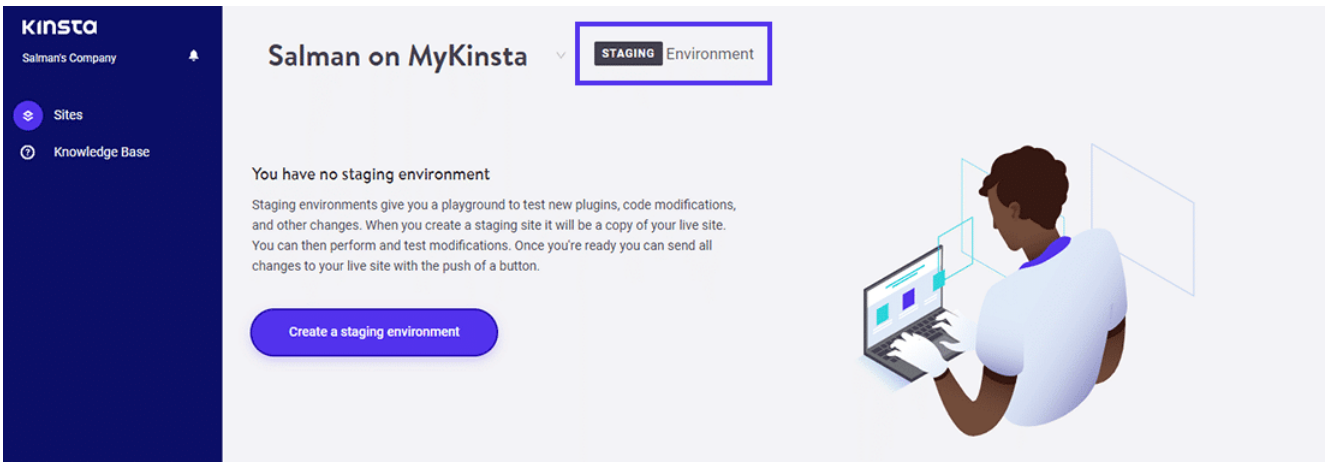
Seitenadministrator



Das ‚Seitenadministrator'-Dashboard in MyKinsta

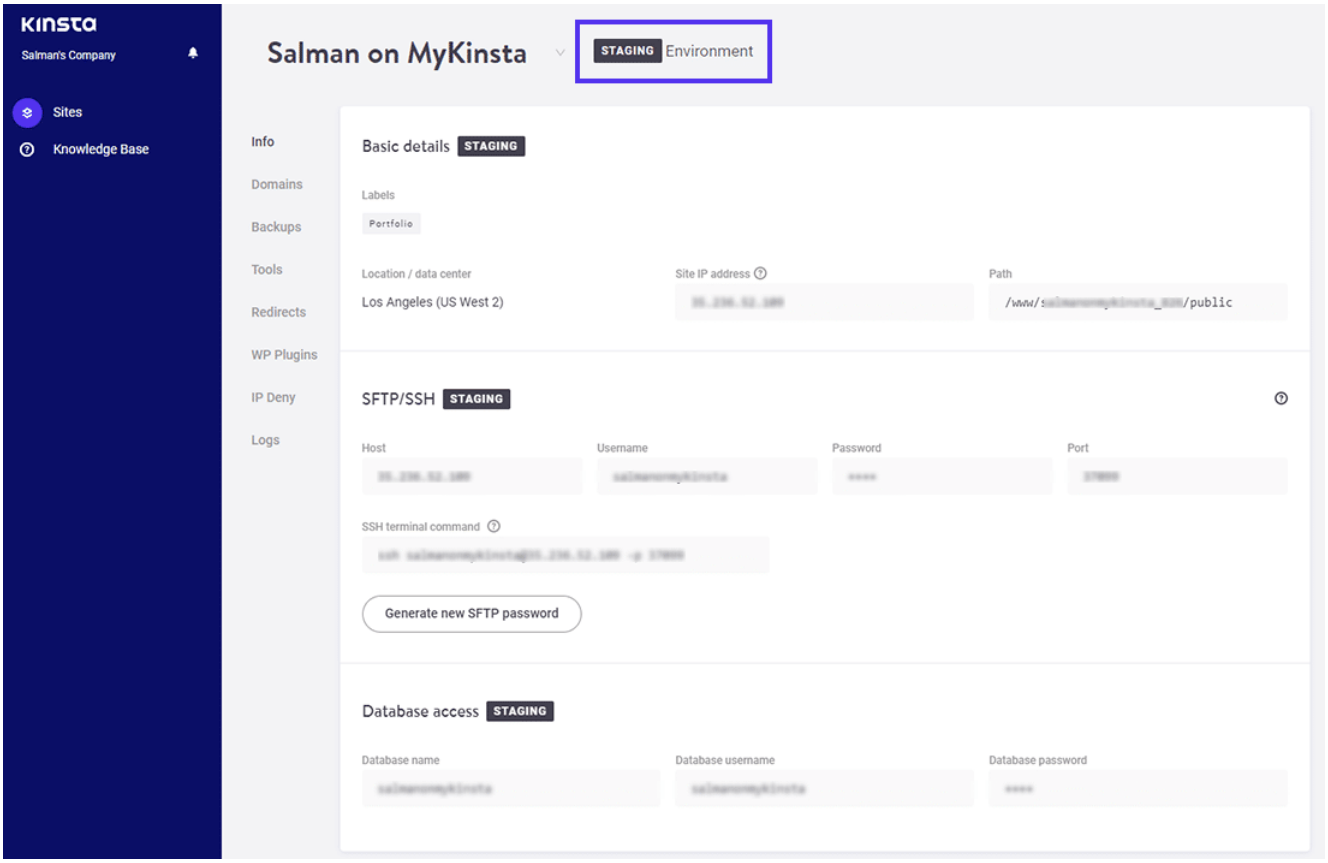
Die Rolle des Seitenadministrators hat vollständigen Zugang zu einer bestimmten Webseite, einschließlich der Kontrolle über alle Umgebungen, die mit dieser Webseite verbunden sind. Sie können eine Webseite jedoch nicht aus dem Firmenkonto entfernen. Du kannst den gleichen Benutzer als Site-Administrator für mehrere Webseiten einsetzen.

Seitenentwickler



Das ‚Seitenentwickler'-Dashboard in MyKinsta

Die Rolle des Seitenentwicklers hat nur Zugriff auf die Staging-Umgebung der ihm zugewiesenen Webseite. Sie können in der Staging-Umgebung alles tun, aber sie können die Staging-Umgebung nicht entfernen oder ihre Änderungen live übertragen. Wie bei den Seitenadministratoren kannst du denselben Benutzer als Seitenentwickler für mehrere Webseiten zuweisen.



Seitenentwickler können auf die Staging-Umgebung für die zugewiesene Webseite zugreifen

Du kannst auch sehen, dass Seitenentwickler keinen Zugang zu den Analyse-, Benutzerverwaltungs- und Aktivitätsprotokollfunktionen im MyKinsta-Dashboard haben.

MyKinsta-Benutzerrollen vs. WordPress-Benutzerrollen

Es gibt keine Überschneidungen zwischen den Benutzerrollen von MyKinsta und WordPress. Du kannst sie beide unabhängig voneinander verwenden.

Als Besitzer eines Kinsta-Accounts hilft dir die Multi-User-

Rollen-Funktion in MyKinsta, ein Team von Managern, Entwicklern und Buchhaltern einfach zu verwalten. Es macht es für Webentwicklungsagenturen super einfach, alle Webseiten ihrer Kunden von einem einzigen, leistungsstarken Dashboard aus zu verwalten.Bekomme die Kontrolle über deine WordPress Webseite und rationalisiere deinen Workflow mit klar definierten Benutzerrollen & Fähigkeiten Click to Tweet

Zusammenfassung

WordPress Rollen und Fähigkeiten sind die grundlegenden Konzepte hinter der Benutzerzugangsverwaltung. Sie helfen dir zu kontrollieren, welche Aktionen alle Benutzer deiner Webseite ausführen können. Sie werden auch von vielen Plugins und Themes verwendet, um dem Core von WordPress sehr nützliche Funktionen hinzuzufügen.

WordPress kommt mit einem eigenen Satz von Rollen und Fähigkeiten, aber wenn du mehr Flexibilität brauchst, kannst du sie anpassen oder deine eigenen Rollen und Fähigkeiten erstellen. Du kannst dies mit deinem eigenen Code oder mit einem Plugin eines Drittanbieters tun.

Zu verstehen, was Rollen und Fähigkeiten sind, und zu lernen, wie man sie verwaltet, ist ein entscheidender Schritt, um WordPress zu beherrschen. Fang noch heute mit ihnen an!

Spare Zeit, Kosten und maximiere die Leistung deiner Seite mit:

- Schnelle Hilfe von WordPress Hosting Experten, 24/7.
- Cloudflare Enterprise Integration.
- Globale Reichweite mit 29 Rechenzentren weltweit.
- Optimierung mit unserem integrierten Application Performance Monitoring.

All das und noch viel mehr, in einem Plan ohne langfristige

Verträge, betreute Migrationen und eine 30-Tage-Geld-zurück-Garantie. Schau Dir hier unsere Pakete an oder sprich mit dem Vertrieb, um den richtigen Tarif für dich zu finden.