SAP Business Technology Platform – Administration

Kapitel 3
Fortgeschrittene Identity-Authentication-Service-Themen

Die SAP BTP bietet im Zusammenspiel mit Identity Authentication sehr viele Möglichkeiten, mit denen Sie eine Vielzahl an Anforderungen im Standard abdecken können. Zentral ist dabei, dass Sie einen kundeneigenen Identity Provider für beliebig viele Ihrer Subaccounts an einer Stelle konfigurieren können, aber auch Dinge wie eine Attribut-basierte Authentifizierung (»conditional authenfication«) sind möglich.

In diesem Kapitel tauchen wir tiefer in die Welt der fortgeschrittenen Konzepte und Technologien ein, die im Bereich des Identity Authentication Service von zentraler Bedeutung sind. Unsere Reise führt uns durch verschiedene Schlüsseltechnologien und praxisnahe Beispiele, die ein umfassendes Verständnis dieser komplexen und dynamischen Domäne ermöglichen.

An dieser Stelle möchten wir nochmals *Single-Sign-On* (SSO) in Erinnerung rufen. SSO ist ein Verfahren, das es Nutzer*innen ermöglicht, sich mit einer einzigen Authentifizierung (z. B. Benutzername und Passwort) bei mehreren Anwendungen oder Diensten anzumelden. Anstatt sich für jede Anwendung separat anmelden zu müssen, bietet SSO eine zentrale Anmeldeplattform, die nach einer einmaligen Authentifizierung Zugriff auf mehrere Systeme oder Ressourcen ermöglicht. SSO bietet zahlreiche Vorteile, die es sowohl für Anwender*innen als auch für das Administrationsteam zu einer attraktiven Lösung macht.

Einer der Hauptvorteile ist die Benutzerfreundlichkeit, da nur eine einzige Kombination aus Benutzername und Passwort zur Authentifizierung benötigt wird. Dies erleichtert den Anmeldeprozess erheblich und verringert die Wahrscheinlichkeit, dass Passwörter vergessen werden. In der Folge steigert SSO die Effizienz, da Mitarbeitende weniger Zeit mit Anmeldeprozessen verbringen und sich stattdessen auf ihre eigentliche Arbeit konzentrieren können. Für das Administrationsteam vereinfacht SSO die Verwaltung von Benutzerkonten und Zugriffsrechten, da sie anstatt mehrerer separater Authentifizierungssysteme nur eine zentrale Plattform überwachen und warten müssen. Da Sie Identity Authentication hinsichtlich SSO unterstützt, gehen wir in diesem Kapitel mit SAML 2.0 und OIDC auf die technischen Grundlagen ein.

Security Assertion Markup Language (SAML 2.0) und OpenID Connect sind beides Standards für eine webbasierte Authentifizierung und Autorisierung. Sie haben jedoch unterschiedliche Eigenschaften und Anwendungsbereiche. Beide Standards spielen in Kombination mit der SAP BTP eine wichtige Rolle.

Wir beginnen in Abschnitt 3.1, »SAML 2.0«, mit einem Überblick über SAML, einer Kernkomponente für das Identitätsmanagement in vielen Unternehmensumgebungen. SAML ermöglicht es, Authentifizierungsinformationen sicher zwischen verschiedenen Parteien zu übermitteln. Wir betrachten dessen Funktionsweise, Vorteile und Herausforderungen im Detail. Als Nächstes beleuchten wir in Abschnitt 3.2, »OpenID Connect«, einen modernen Identitätslayer, der auf dem OAuth-2.0-Protokoll aufbaut. Durch praktische Beispiele erkunden wir, wie OpenID Connect eine vereinfachte und doch sichere Authentifizierung und Autorisierung im Internet ermöglicht. In Abschnitt 3.3, »Praxisbeispiel: SAP Identity Authentication Service als Proxy zu Microsoft Entra ID«, behandeln wir die praktische Anwendung und Integration von Identity Authentication als Proxy zu Azure Active Directory. Wir zeigen, wie diese Konfiguration die Verwaltung von Benutzeridentitäten und Zugriffsrechten in cloudbasierten Umgebungen effizient unterstützt. Die Zwei-Faktor-Authentifizierung ist ein entscheidendes Element der modernen Sicherheitsarchitektur. In Abschnitt 3.4, »Praxisbeispiel: Zwei-Faktor Authentifizierung/risikobasierte Authentifizierung«, untersuchen wir verschiedene Methoden und Technologien, die in der Zwei-Faktor-Authentifizierung verwendet werden, und demonstrieren deren Implementierung in realen Szenarien. Conditional Authentication ist eine fortgeschrittene Technik, die Ihnen dabei hilft, das Sicherheitslevel auf der Basis von Benutzerkontext und -verhalten zu verbessern. Wir erläutern in Abschnitt 3.5, »Praxisbeispiel: Conditional Authentication«, wie diese Methode die Sicherheit erhöht, ohne die Benutzererfahrung zu beeinträchtigen.

3.1 SAML 2.0

SAML 2.0 (Security Assertion Markup Language) ist ein Standard für den Austausch von Authentifizierungs- und Autorisierungsinformationen. SAML 2.0 wird oft in Unternehmensumgebungen eingesetzt, um SSO zu ermöglichen. Der SAML-Standard wird vom OASISopen-Konsortium verwaltet. OASIS ist eine gemeinnützige Organisation, die sich der Entwicklung von offenen Standards für die Informationstechnologie widmet. SAML wurde ursprünglich von IBM, Microsoft und Novell entwickelt. Die erste Version des Standards wurde 2002 veröffentlicht. SAML 2.0, die aktuelle Version des Standards, wurde 2005 veröffentlicht.

SAML ist ein XML-basiertes Authentifizierungsprotokoll, über das Identity Provider mit Service Providern kommunizieren. *Identity Provider* verwalten die Anmeldeda-

ten von Nutzer*innen, während *Service Provider* jene Anwendungen sind, die eine Authentifizierung erfordern.

SAML 2.0 unterstützt zwei Arten, einen Anmeldeprozesses zu beginnen: das *Service Provider Initiated SSO* und das *Identity Provider Initiated SSO*. Der Hauptunterschied zwischen Service Provider Initiated SSO und Identity Provider Initiated SSO besteht darin, an welcher Stelle der Anmeldevorgang beginnt. Bei Service Provider Initiated SSO beginnt der Anmeldevorgang bei der Anwendung, die der Benutzer öffnen möchte. Bei Identity Provider Initiated SSO beginnt der Anmeldevorgang beim Identity Provider, der die Anmeldedaten der Benutzer verwaltet.

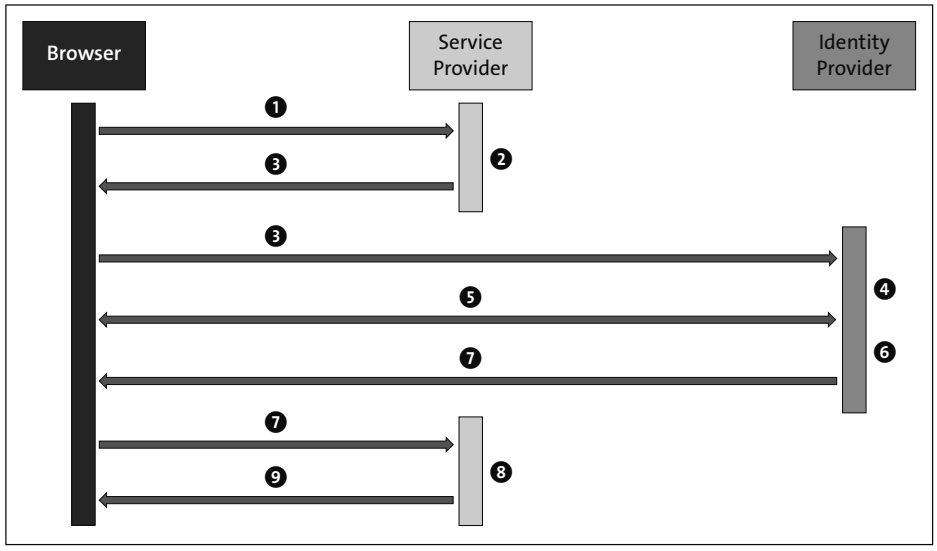

Abbildung 3.1 Ablauf von Service Provider Initiated SSO

Der Ablauf von Service Provider Initiated SSO wird in Abbildung 3.1 dargestellt. Er funktioniert wie folgt:

❶ Der Benutzer besucht die Website einer Anwendung, die Service Provider Initiated SSO unterstützt.

❷ Die Anwendung (die in diesem Kontext die Funktion des Service Providers erfüllt) erkennt, dass der User noch nicht angemeldet ist und erstellt und signiert einen SAML-Request.

❸ Die Anwendung leitet den Benutzer damit zur Identity-Provider-Login-URL weiter.

❹ Der Identity Provider prüft die Signatur der Anforderung, parst den SAML-Request und fordert den Benutzer auf, sich zu authentifizieren.

❺ Der Benutzer gibt seine Anmeldedaten ein.

❻ Der Identity Provider authentifiziert den Benutzer und erstellt eine SAML-Response.

❼ Mit der SAML-Response wird der Benutzer an den Service Provider weitergeleitet.

❽ Die Anwendung prüft die Signatur der SAML-Response, validiert diese und führt eine Berechtigungsprüfung des Benutzers durch.

❾ Der Service Provider stellt die gewünschte Ressource bereit.

Der Ablauf von Identity Provider Initiated SSO wird in Abbildung 3.2 dargestellt. Er funktioniert wie folgt:

❶ Der Benutzer besucht die Website des Identity Providers.

❷ Der Benutzer gibt seine Anmeldedaten beim Identity Provider ein.

❸ Der Identity Provider authentifiziert den Benutzer und erstellt eine SAML-Assertion.

❹ Die SAML-Assertion wird an den Browser zurückgesendet.

❺ Der Browser sendete den Benutzer mit der SAML-Response im Gepäck an die Anwendung.

❻ Der Service Provider prüft die Signatur der SAML-Response, validiert diese und führt eine Berichtigungsprüfung des Benutzers durch.

❼ Die Anwendung stellt die gewünschte Ressource bereit.

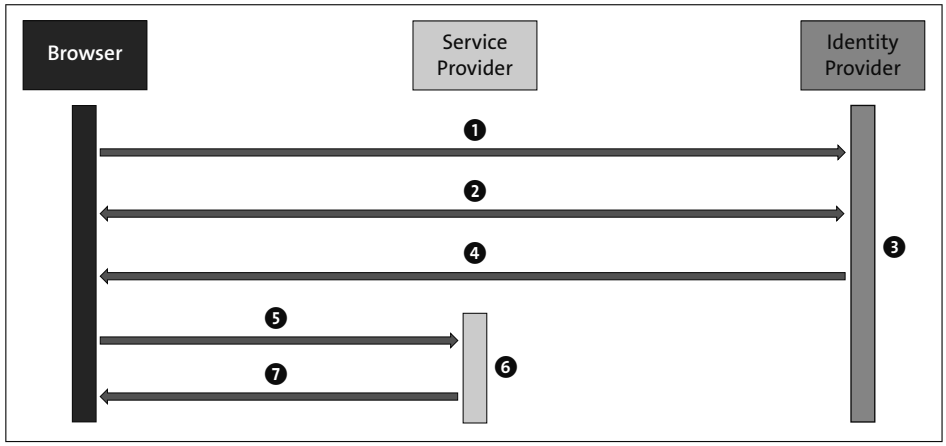

Abbildung 3.2 Ablauf von Identity Provider Initiated SSO

SAML wird auch auf der SAP BTP für verschiedene Szenarien unterstützt. SAP stellt die SAP BTP, wie bereits erwähnt, in Form von zwei Technologie-Stacks, der alten Neo-Umgebung und der neuen Multi-Cloud-Umgebung bereit. In der Neo-Umgebung wurde ausschließlich auf SAML gesetzt, da OpenID Connect erst 2014 veröffentlich worden war und daher zum Zeitpunkt der Einführung der Neo-Umgebung noch nicht zur Verfügung stand. SAML 2.0 war zu diesem Zeitpunkt der De-Facto-Standard, auf den auch SAP gesetzt hat. SAML 2.0 kommt deshalb in der Neo-Umgebung sowohl für

die Integration von Identity Authentication in der Rolle als Platform Identity Provider und als Application Identity Provider zum Einsatz.

Zusätzlich können in der Neo-Umgebung auch Identity Provider von Drittanbietern, beispielsweise Microsoft Entra ID, als Application Identity Provider über SAML 2.0 integriert werden. Details und Praxisbeispiele dazu finden Sie in Kapitel 5, »Subaccounts administrieren«, dieses Buches.

In der Cloud-Foundry-Umgebung können Sie SAML 2.0 für die Integration von Identity Providern von Drittanbietern als Application Identity Provider nutzen. Der Service Identity Authentication wird in der Cloud-Foundry-Umgebung sowohl als Platform Identity Provider als auch als Application Identity Provider ausschließlich über OpenID Connect angebunden.

3.2 OpenID Connect

OpenID Connect (OIDC) ist ein Authentifizierungsprotokoll, das die Benutzerauthentifizierung und -autorisierung über Identity Provider von Drittanbietern ermöglicht. Es baut auf dem OAuth-2.0-Protokoll auf und bietet zusätzliche Funktionen für die sichere Verwaltung von Benutzeridentitäten und Zugriffstokens. Daher erklären wir Ihnen zunächst OAuth 2.0.

OAuth 2.0 ist ein Protokoll zur Autorisierung von Webanwendungen und APIs. Es ermöglicht einer Anwendung, auf Ressourcen einer anderen Anwendung zuzugreifen, ohne dass die Anwendung die Anmeldedaten des Benutzers kennen müsste. OAuth 2.0 basiert auf dem Prinzip der Zustimmung. Der Benutzer erteilt einer Anwendung die Erlaubnis, auf seine Ressourcen zuzugreifen. Die Anwendung kann dann einen Token von einem Autorisierungsserver anfordern, der die Erlaubnis des Benutzers bestätigt.

OAuth 2.0 vereinfacht die Implementierung der Autorisierung für Client-Entwickler*innen und bietet gleichzeitig spezifische Autorisierungsabläufe für unterschiedliche Anwendungstypen. Der OAuth-2.0-Standard und dessen Erweiterungen werden von der Internet Engineering Task Force (IETF) definiert. Das Framework wird im RFC 6749 spezifiziert. RFC steht hier für »Request for Comments«, die Sammlung der Internetspezifikationen.

OAuth 2.0 definiert vier Rollen innerhalb des Autorisierungsprozesses:

- **Ressourcenbesitzer**
  Der *Ressourcenbesitzer* ist eine Einheit, die in der Lage ist, Zugang zu einer geschützten Ressource zu gewähren. Wenn der Ressourcenbesitzer eine Person ist, wird er als Benutzer bezeichnet.

- **Ressourcenserver**
  Der Server, auf dem die geschützten Ressourcen gehostet werden, ist in der Lage, unter der Verwendung von Zugriffstokens Anfragen nach geschützten Ressourcen anzunehmen und zu beantworten. Er heißt deshalb *Ressourcenserver*.
- **Client**
  Der *Client* ist eine Anwendung, die im Namen des Ressourcenbesitzers und mit dessen Genehmigung geschützte Ressourcen anfragt. Der Begriff »Client« impliziert keine besonderen Implementierungsmerkmale (z. B. einen Formfaktor oder die Bauart des Geräts).
- **Autorisierungsserver**
  Der Server, der nach erfolgreicher Authentifizierung des Ressourcenbesitzers und dessen Autorisierung Zugriffstokens an den Client ausgibt, wird *Autorisierungsserver* genannt.

Um einen Zugriffstoken anzufordern, erhält der Client die Autorisierung vom Ressourcenbesitzer. Die Autorisierung wird in Form einer Berechtigungserteilung (auf Englisch: Authorization Grant) ausgedrückt. Mit dieser Berechtigung fordert der Client dann das Zugriffstoken an (siehe Abbildung 3.3).

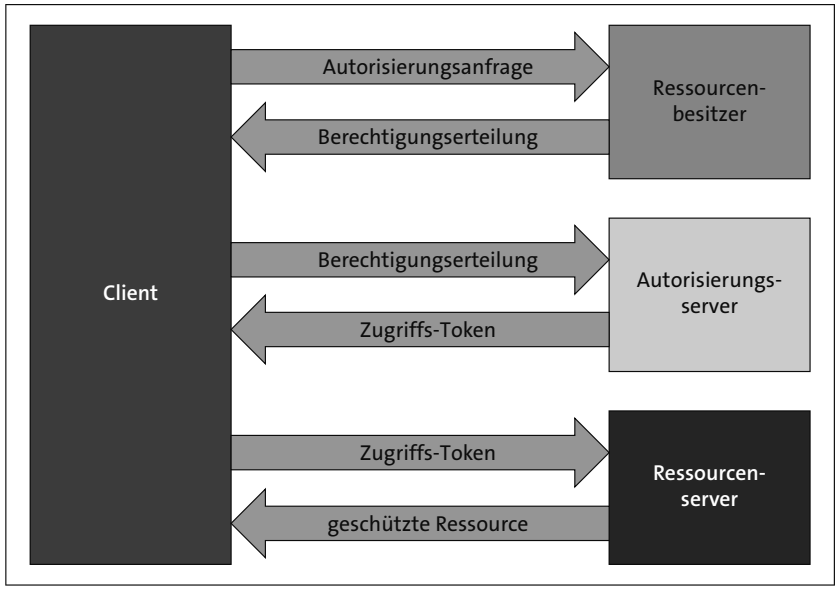

Abbildung 3.3 Ablauf des Autorisierungsprozesses mit OAuth 2.0

OAuth 2.0 definiert vier Arten der Berechtigungserteilung (englisch: Grant Types):

- über einen Autorisierungscode
- über die implizite Autorisierung

- über die Anmeldedaten (das Passwort) des Ressourcenbesitzers
- über die Anmeldedaten des Clients (die Client Credentials)

Zusätzlich sieht OAuth 2.0 einen Erweiterungsmechanismus für die Definition zusätzlicher Arten der Berechtigungserteilung vor.

OAuth 2.0 bietet eine Vielzahl an Anwendungsfällen, darunter:

- **Social Login**
  Benutzer können sich mit ihren Social-Media-Konten bei Anwendungen anmelden.
- **SSO**
  Benutzer können sich mit einem einzigen Satz von Anmeldedaten bei mehreren Anwendungen anmelden.
- **API-Authentifizierung**
  Anwendungen können auf geschützte Ressourcen und Dienste zugreifen, ohne dass die Benutzer ihre Anmeldedaten eingeben müssten.

Da Sie nun eine genauerer Vorstellung von OAuth 2.0 haben, können wir wieder auf OpenID Connect zurückkommen und beleuchten, wie beide Protokolle zusammenspielen. Sowohl OAuth 2.0 als auch OIDC sind offene Standardprotokolle, die für die Authentifizierung und Autorisierung von Benutzern verwendet werden. OAuth 2.0 ist ein Autorisierungsprotokoll, das es einer Anwendung ermöglicht, auf Ressourcen einer anderen Anwendung zuzugreifen, ohne dass die Anwendung die Anmeldedaten des Benutzers kennen müsste. OIDC ist ein Authentifizierungsprotokoll, das es Benutzern ermöglicht, sich bei einer Anwendung zu authentifizieren, indem sich diese bei einem Identity Provider authentifizieren.

OIDC basiert auf OAuth 2.0 und fügt zusätzliche Funktionen hinzu, um die Authentifizierung zu vereinfachen und zu verbessern. Dazu gehören:

- **JSON-Web-Token-basierte Authentifizierung**
  OIDC verwendet *JSON Web Tokens* (JWTs), um die Identität und Autorisierung des Benutzers zu übermitteln. JWTs sind kompakte, signierte Tokens, die sicher über das Netzwerk übertragen werden können.
- **Zentralisierte Authentifizierung**
  OIDC ermöglicht es Benutzern, sich bei einer Anwendung zu authentifizieren, indem sie sich bei einem Identity Provider authentifizieren. Dies vereinfacht den Authentifizierungsprozess für Benutzer und erleichtert die Einhaltung von Sicherheitsrichtlinien.
- **Dezentrale Identitätsverwaltung**
  OIDC ermöglicht es Identity Providern, die Identitätsdaten von Benutzern zu verwalten. Dies vereinfacht die Verwaltung von Benutzerdaten und verbessert die Sicherheit.

An dieser Stelle erklären wir Ihnen die JWTs genauer, da sie das zentrale Element in Zusammenhang mit OIDC sind. Ein JWT besteht aus drei Teilen, die durch Punkte getrennt sind:

- **Header**
  Der Header enthält Informationen über das Token, wie z. B. das verwendete Signaturverfahren.
- **Payload**
  Der Payload enthält die eigentlichen Daten des Tokens, wie beispielsweise die Identität und Autorisierung des Benutzers.
- **Signature**
  Die Signature ist eine digitale Signatur, die die Integrität des Tokens gewährleistet.

In Abbildung 3.4 sehen Sie ein Beispiel für einen JWT sowie dessen decodierte Darstellung.

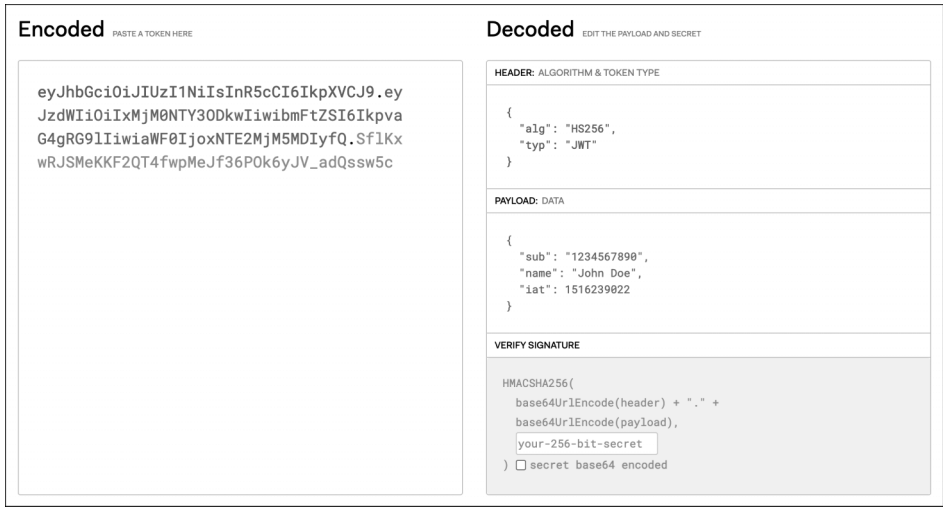

Abbildung 3.4 Beispiel für einen JWT

OAuth 2.0 und OIDC arbeiten zusammen, um eine sichere und einfache Authentifizierung und Autorisierung für Benutzer zu ermöglichen. Der allgemeine Ablauf folgt den folgenden Schritten:

1. Der Benutzer möchte sich bei einer Anwendung anmelden.
2. Die Anwendung fordert den Benutzer auf, sich bei einem Identity Provider zu authentifizieren.
3. Der Benutzer gibt seine Anmeldedaten beim Identity Provider ein.
4. Der Identity Provider authentifiziert den Benutzer und stellt ein JWT aus, das die Identität und Autorisierung des Benutzers enthält.

5. Die Anwendung erhält das JWT vom Identity Provider.
6. Die Anwendung verwendet das JWT, um zu überprüfen, ob der Benutzer berechtigt ist, auf die entsprechenden Ressourcen zuzugreifen.

In diesem Ablauf übernimmt OAuth 2.0 die Verantwortung für die Autorisierung der Anwendung, die für den Zugriff auf die Ressourcen des Ressourcenbesitzers erforderlich ist. OIDC übernimmt die Verantwortung für die Authentifizierung des Benutzers, für die Übermittlung der Identität und für die Autorisierung des Benutzers an die Anwendung.

Auf der SAP BTP kommt OIDC an verschiedenen Stellen der Multi-Cloud-Umgebung zum Einsatz. Auf der Ebene des Global Account kann Identity Authentication mithilfe von OIDC als Identity Provider für die Global-Account-Benutzer angebunden werden. Zusätzlich erfolgt in der Multi-Cloud-Umgebung auf der Ebene der Subaccounts die Integration von Identity Authentication sowohl als Platform Identity Provider als auch als Application Identity Provider mithilfe von OIDC. Falls Identity Authentication als Proxy zu einem Identity Provider eines Drittanbieters verwendet wird, kann für die Verbindung OIDC verwendet werden.

3.3 Praxisbeispiel: SAP Identity Authentication Service als Proxy zu Microsoft Entra ID

Ein in der Praxis häufig zu findendes Beispiel ist die Verwendung von Identity Authentication als Proxy zu anderen Identity Providern. Damit haben Sie die Möglichkeit, dass Sie beispielsweise Microsoft Entra ID (ehemals Microsoft Azure Active Directory) im Proxy-Modus verwenden können. Der Vorteil dieses Ansatzes liegt darin, dass Sie die Verbindung zwischen Identity Authentication und Identity Provider eines Drittanbieters nur einmal herstellen müssen, unabhängig davon, in wie vielen Subaccounts dieser später verwendet wird. Damit lassen sich der initiale Konfigurationsaufwand und der Aufwand der Administration im laufenden Betrieb verringern. Hierzu bietet Identity Authentication zwei Möglichkeiten: Die erste Möglichkeit ist die Integration des Identity Providers mithilfe von SAML 2.0, und die zweite Option ist die Integration mithilfe von OIDC.

In den folgenden beiden Abschnitten zeigen wir Ihnen anhand von Schritt-für-Schritt-Anleitungen, wie die Konfiguration in Identity Authentication und in Microsoft Entra ID erfolgt. In Abschnitt 3.3.1, »Mit SAML 2.0 Microsoft Entra ID in Identity Authentication integrieren«, nutzen wir für die Integration SAML 2.0, und in Abschnitt 3.3.2, »Mit OpenID Connect Microsoft Entra ID in Identity Authentication integrieren«, nutzen wir OIDC.

Wenn Sie unserer Anleitung folgen möchten, benötigen Sie einen Benutzer, der sowohl in Identity Authentication als auch auf dem Identity Provider des Drittanbieters, in unserem Fall Microsoft Entra ID, über ausreichende Berechtigungen verfügt.

3.3.1 Mit SAML 2.0 Microsoft Entra ID in Identity Authentication integrieren

Im Kontext von SAML nimmt der Service Identity Authentication nun die Rolle des Service Providers und Microsoft Entra ID die Rolle des Identity Providers ein. Wie in der SAML-Konfiguration üblich, müssen die Metadaten zwischen Service Provider und Identity Provider ausgetauscht werden. Daher müssen Sie zuerst die Metadaten Ihrer Identity-Authentication-Instanz finden und sie auf Ihren Rechner herunterladen.

Melden Sie sich zunächst bei Identity Authentication an. Dort sehen Sie das Startbild (siehe Abbildung 3.5). Klicken Sie im Bereich **Applications & Resources** auf die Kachel **Tenant Settings**.

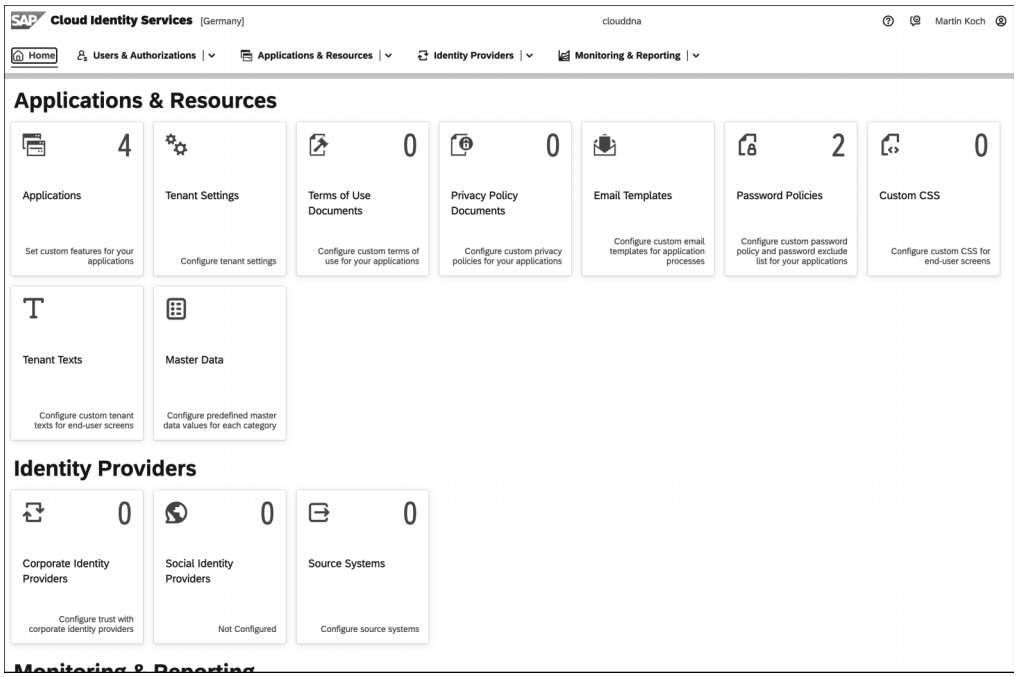

Abbildung 3.5 Startbild von Identity Authentication

Navigieren Sie nun auf die Registerkarte **Applications & Resources**, und wählen Sie in der Liste die gewünschte Applikation (in unserem Beispiel **clouddna**) aus. Klicken Sie in der sich nun öffnenden Ansicht im Bereich **Single Sign-On** auf den Eintrag **SAML 2.0 Configuration** (siehe Abbildung 3.6).

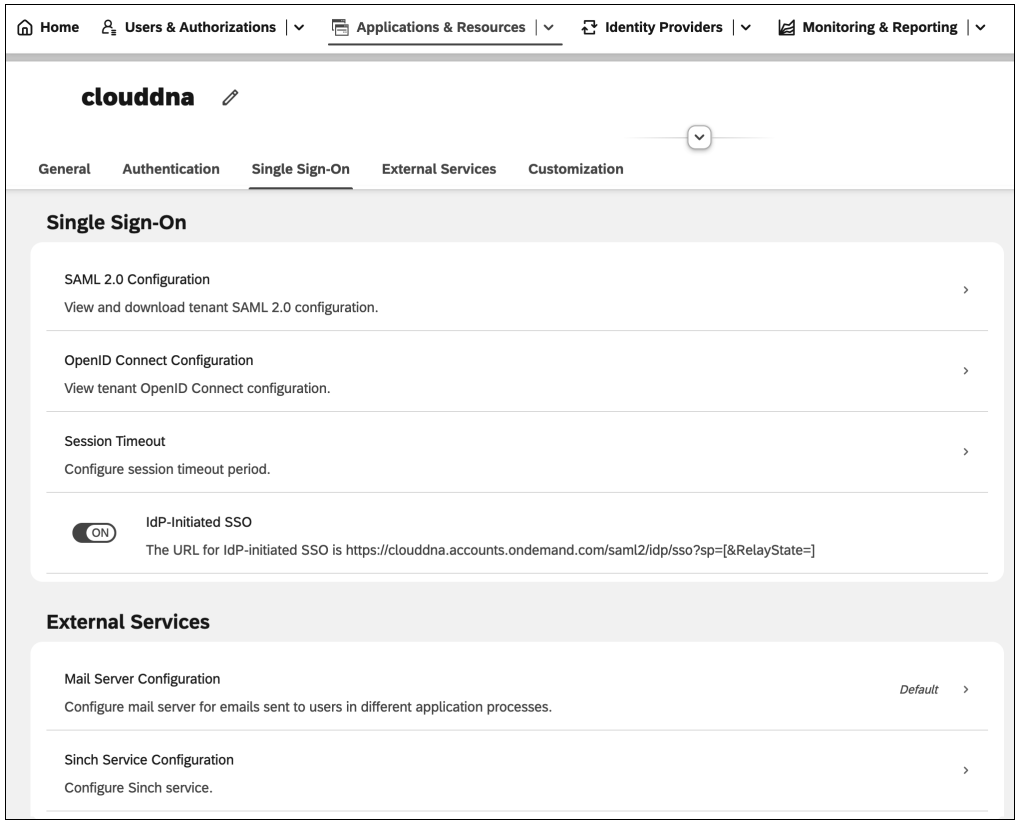

Abbildung 3.6 SAML-2.0-Konfiguration öffnen

Klicken Sie in den Details der SAML-2.0-Konfiguration auf den Button **Download Metadata File** (siehe Abbildung 3.7). Dadurch werden die Service-Provider-Metadaten auf Ihren lokalen Rechner geladen.

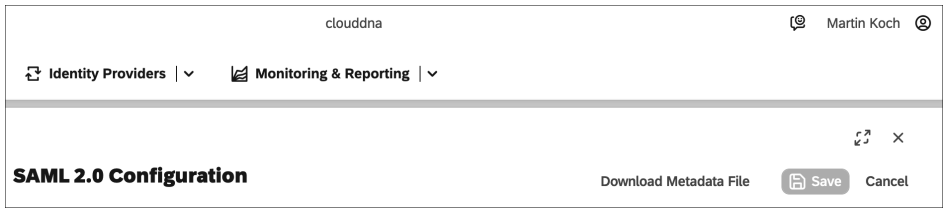

Abbildung 3.7 Service-Provider-Metadaten herunterladen

Sie können nun mit der Konfiguration in Microsoft Entra ID beginnen. Melden Sie sich dazu mit den entsprechenden Berechtigungen über *www.s-prs.de/v100200015* an. Klicken Sie im Startbild auf den Eintrag **Microsoft Entra ID** (siehe Abbildung 3.8).

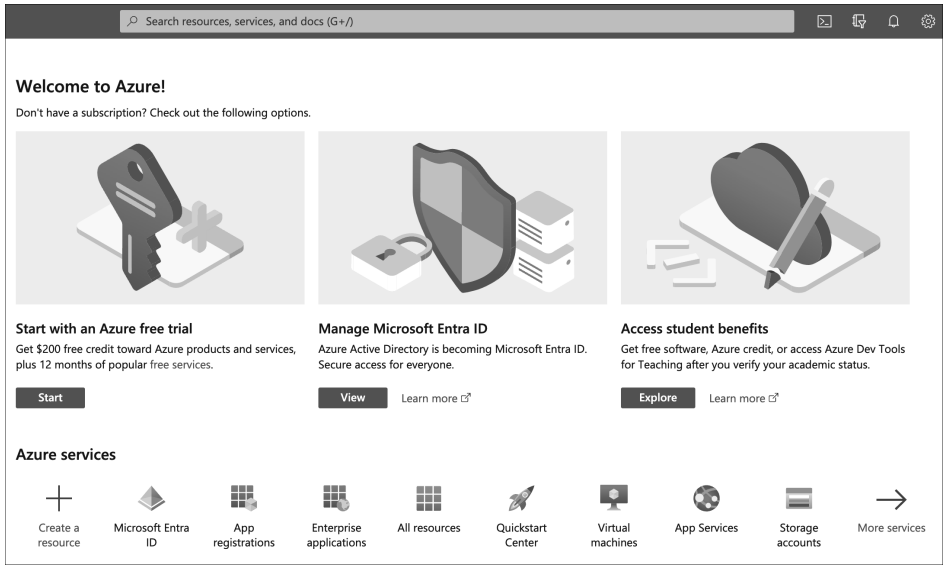

Abbildung 3.8 Startbildschirm des Azure-Portals

Sie müssen nun den Service Provider anlegen. Klicken Sie dazu im Seitenmenü auf den Eintrag **Enterprise applications** (siehe Abbildung 3.9).

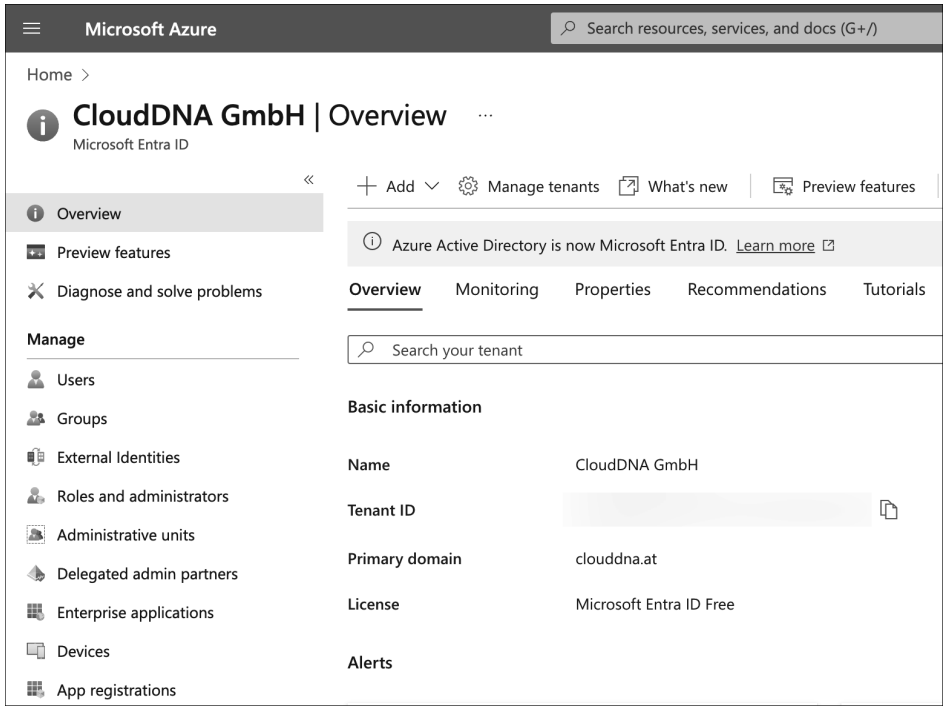

Abbildung 3.9 Enterprise Application registrieren

Klicken Sie anschließend auf den Button **New application** (siehe Abbildung 3.10), um eine neue Enterprise Application anzulegen.

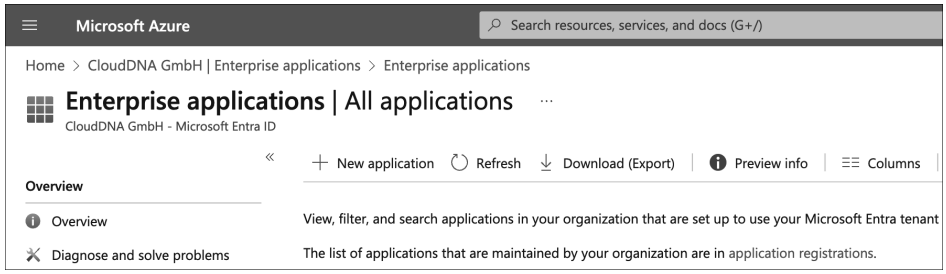

Abbildung 3.10 Neue Enterprise Application anlegen

Ihnen stehen nun verschiedene Cloud-Plattformen zur Wahl (siehe Abbildung 3.11). Dies ist dadurch bedingt, dass Microsoft alle gängigen Cloud-Anbieter unterstützt. Klicken Sie auf die Kachel **SAP**.

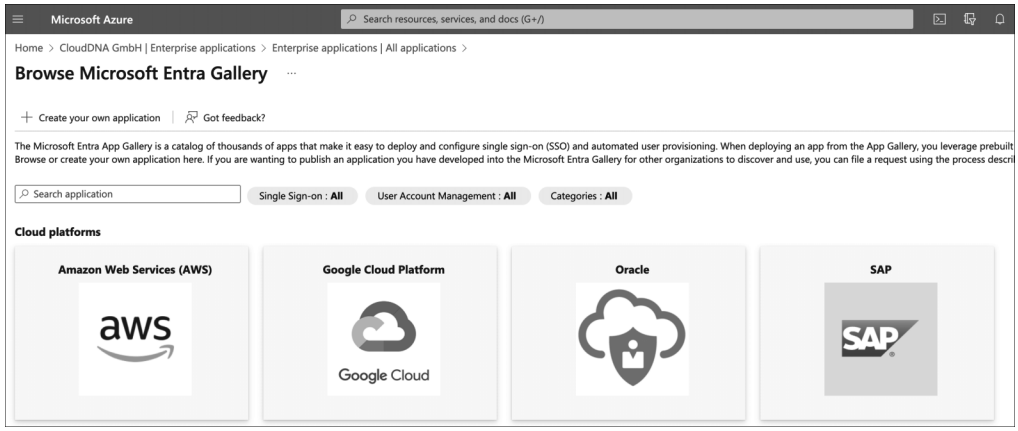

Abbildung 3.11 Cloud-Plattform auswählen

Sie können nun zwischen verschiedenen SAP-Anwendungen auswählen (siehe Abbildung 3.12). Klicken Sie auf **SAP Cloud Identity Services**. An dieser Stelle könnten Sie beispielsweise auch eine Konfiguration für SAP Analytics Cloud oder für SAP Fieldglass durchführen.

Sie müssen nun, wie in Abbildung 3.13 dargestellt, einen Namen für den Service Provider vergeben. Grundsätzlich sind Ihrer Kreativität an dieser Stelle keine Grenzen gesetzt. Aus der praktischen Erfahrung heraus empfehlen wir Ihnen jedoch, einen sprechenden Namen zu wählen. Tragen Sie den gewünschten Namen in das Feld **Name** ein, und klicken Sie anschließend auf den Button **Create**.

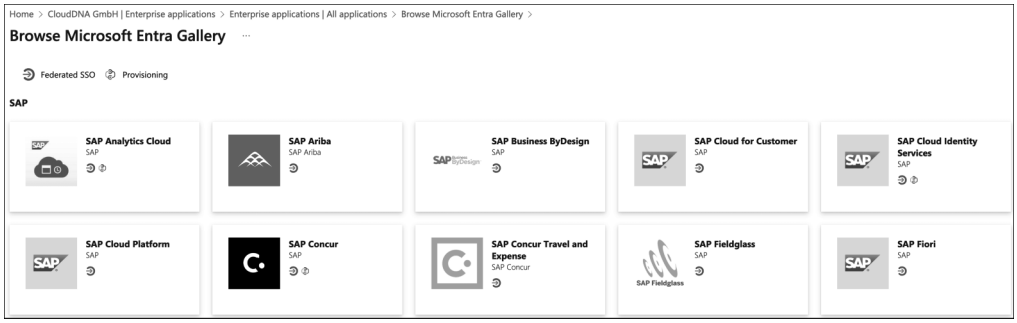

Abbildung 3.12 Übersicht der SAP Services zur Konfiguration

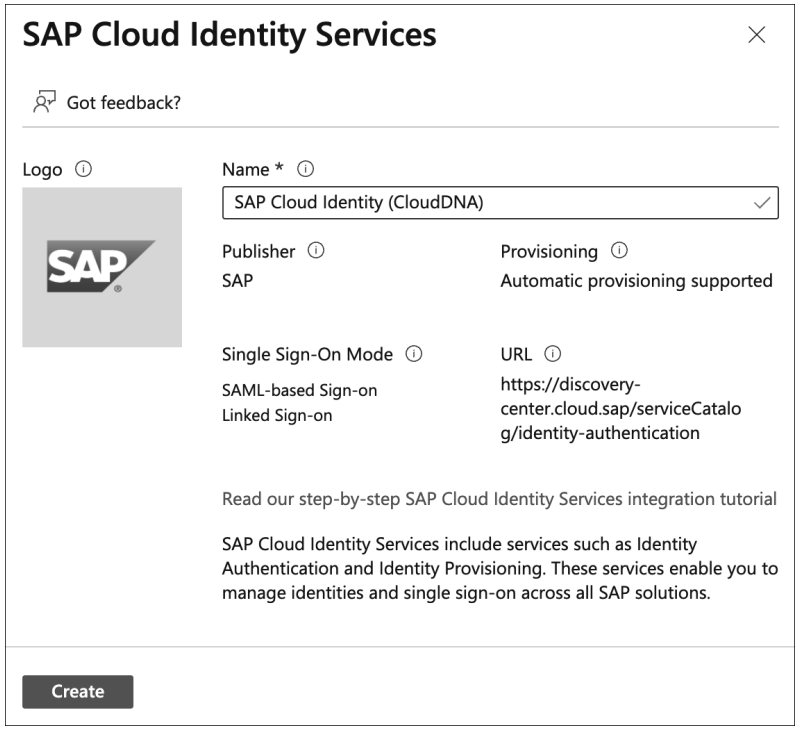

Abbildung 3.13 Service-Provider-Namen vergeben

Es dauert einige Sekunden, bis die Enterprise Application angelegt wird. Danach werden Sie, wie in Abbildung 3.14 zu sehen, direkt zu den Details der Enterprise Application weitergeleitet. Klicken Sie dort auf die Kachel **Set up single sign on**.

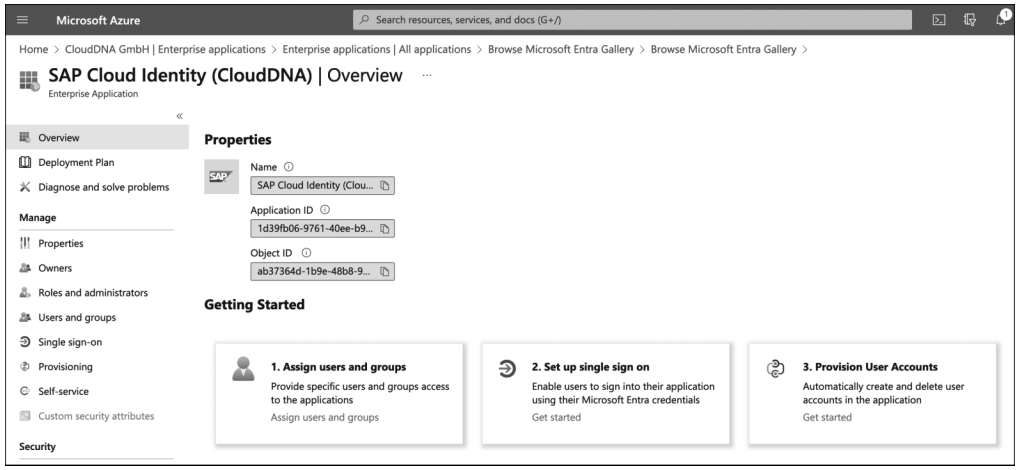

Abbildung 3.14 Übersicht der Enterprise Application

Wählen Sie nun als Single-Sign-On-Methode SAML aus (siehe Abbildung 3.15). Klicken Sie dazu auf die entsprechende Kachel **SAML**.

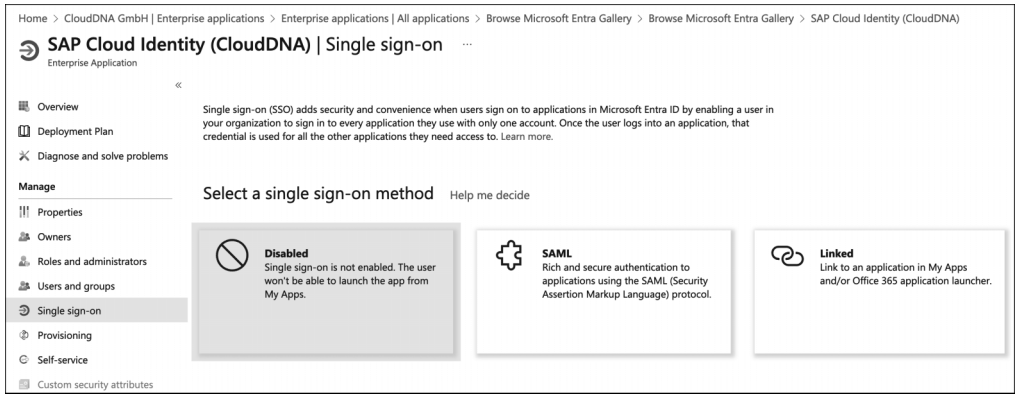

Abbildung 3.15 Single-Sign-On-Methode auswählen

Sie müssen nun die Metadaten hochladen. Klicken Sie dazu auf den Button **Upload metadata file** (siehe Abbildung 3.16).

Wählen Sie nun die zuvor auf ihren Rechner geladene Metadatendatei des Service Providers (Identity Authentication) aus. Klicken Sie anschließend auf den Button **Add**, um die Metadaten hochzuladen (siehe Abbildung 3.17).

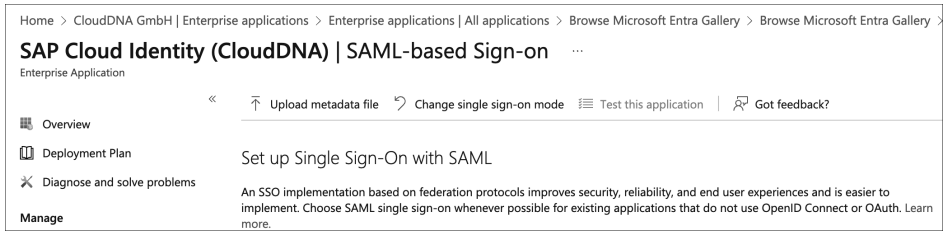

Abbildung 3.16 Metadaten hochladen

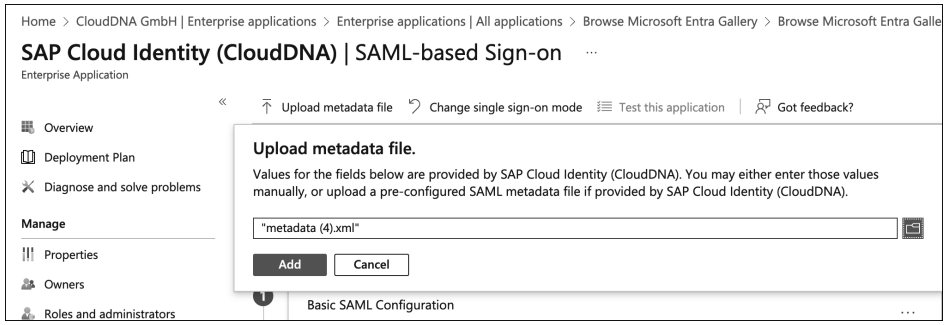

Abbildung 3.17 Metadaten auswählen

Sie sehen anschließend die aus den Metadaten ermittelte SAML-Konfiguration (siehe Abbildung 3.18). An dieser Stelle sind im ersten Schritt keine Anpassungen erforderlich. Klicken Sie daher auf den Button **Save**.

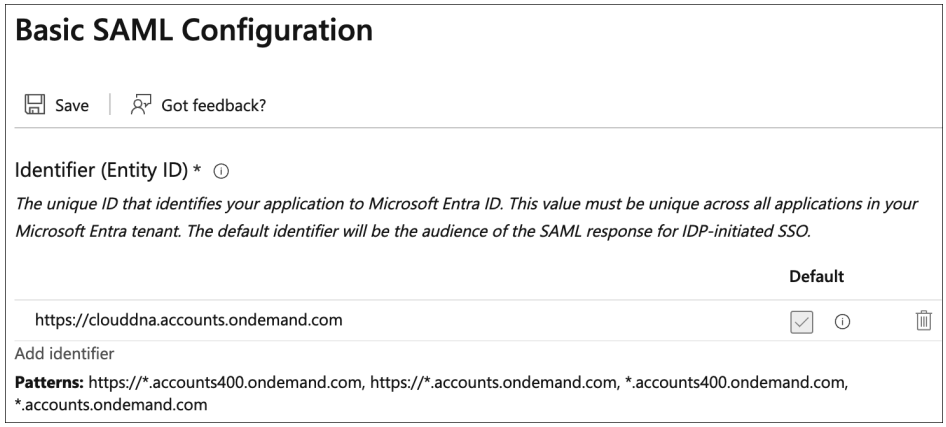

Abbildung 3.18 SAML-Konfiguration speichern

Damit Sie die Anmeldung über Microsoft Entra ID testen können, ist es erforderlich, dass Sie der Enterprise Application entweder einen Benutzer oder eine Gruppe hinzufügen. Klicken Sie dazu im Seitenmenü der Enterprise Application auf den Eintrag

Users and groups, und fügen Sie den gewünschten User mit einem Klick auf **Add user/group** hinzu (siehe Abbildung 3.19).

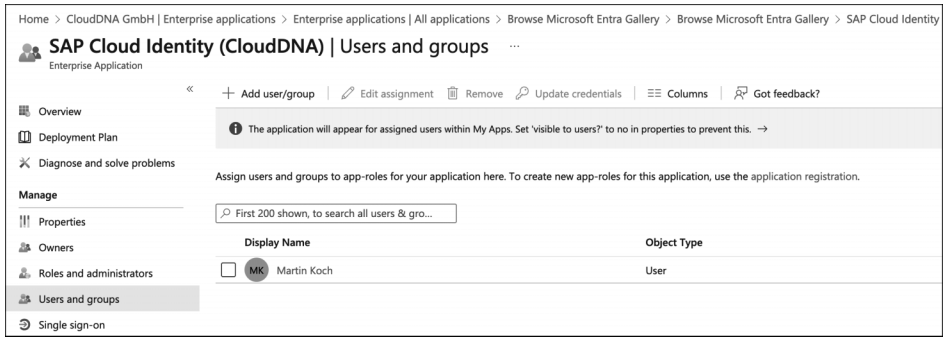

Abbildung 3.19 User hinzufügen

Sie können nun die Metadaten des Identity Providers downloaden. Navigieren Sie dazu im Seitenmenü der Enterprise Application in den Bereich **Single sign-on**. Klicken Sie im Bereich **SAML Certificates** neben dem Eintrag **Federation Metadata XML** auf den Link **Download**. Dadurch werden die Metadaten auf Ihrem lokalen Rechner gespeichert (siehe Abbildung 3.20).

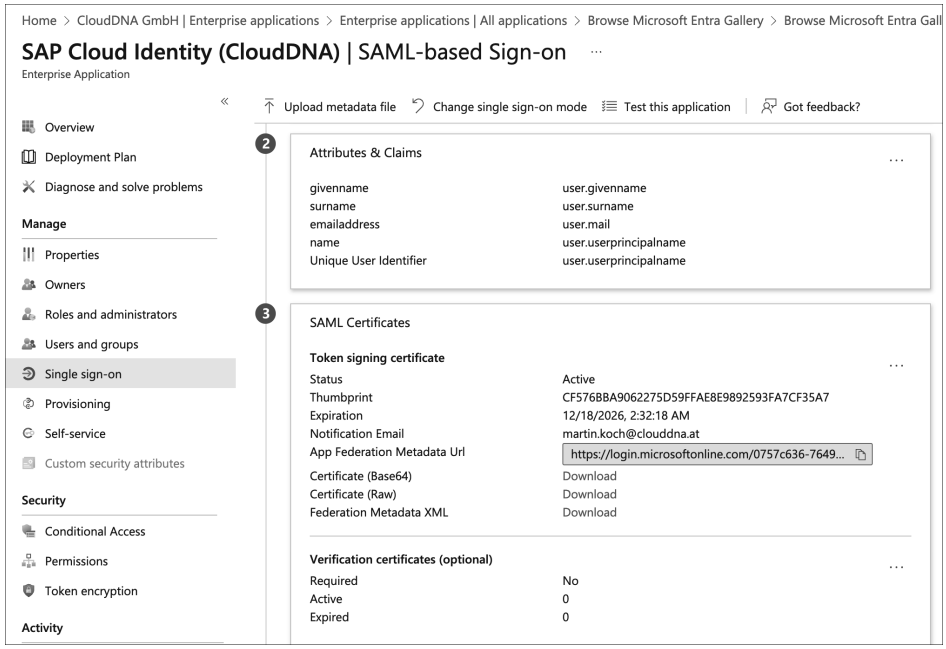

Abbildung 3.20 Metadaten des Identity Providers downloaden

Sie müssen nun die soeben heruntergeladenen Metadaten des Identity Providers in Identity Authentication importieren. Dazu ist es erforderlich, dass ein sogenannter Corporate Identity Provider angelegt wird. Klappen Sie das Menü **Identity Providers** aus, und klicken Sie auf **Corporate Identity Providers** (siehe Abbildung 3.21).

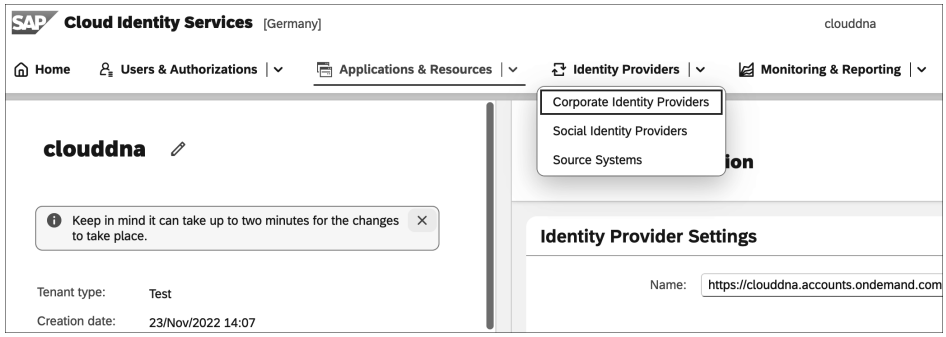

Abbildung 3.21 Zu »Corporate Identity Providers« navigieren

Klicken Sie anschließend auf den Button **Create**, um einen neuen Identity Provider anzulegen (siehe Abbildung 3.22).

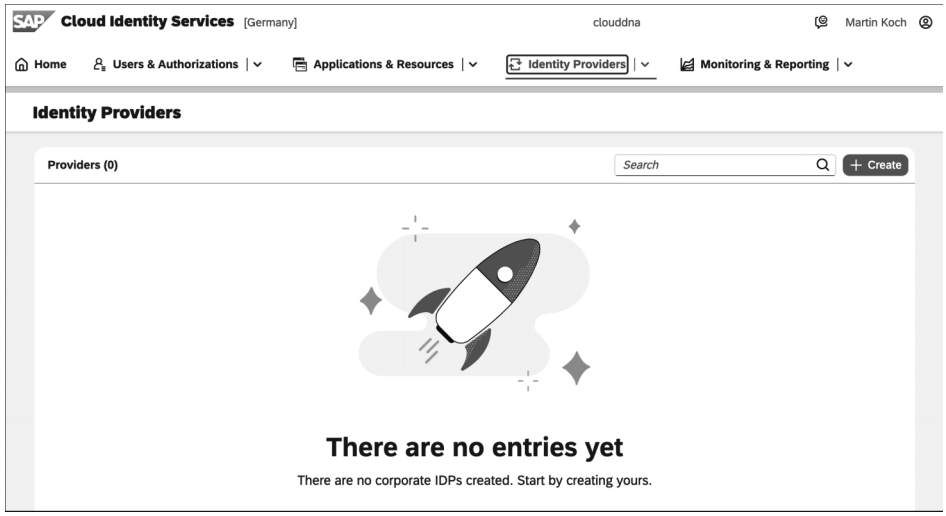

Abbildung 3.22 Neuen Corporate Identity Provider anlegen

Die Konfiguration erfolgt in einem eigenen Dialog (siehe Abbildung 3.23). Vergeben Sie einen sprechenden **Display Name**. Diesen weisen Sie später dem Subaccount zu. Wählen Sie als **Identity Provider Type** den Eintrag **Microsoft ADFS / Azure AD (SAML 2.0)**. Klicken Sie anschließend auf **Save**.

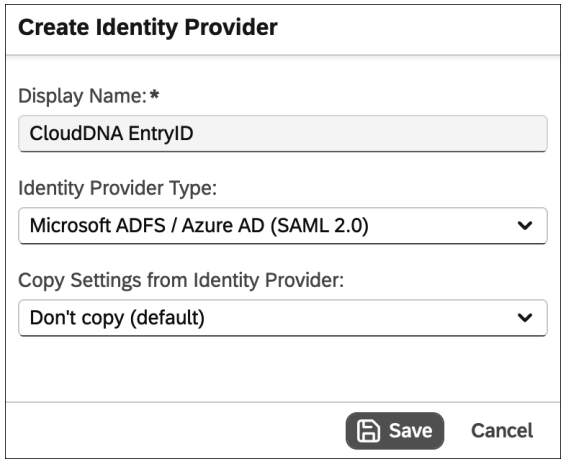

Abbildung 3.23 Corporate Identity Provider konfigurieren

Sie werden danach zu den Details des Identity Providers weitergeleitet. Nun können die Metadaten von Microsoft Entra ID hochladen. Klicken Sie dazu auf den Eintrag **SAML 2.0 Configuration** (siehe Abbildung 3.24).

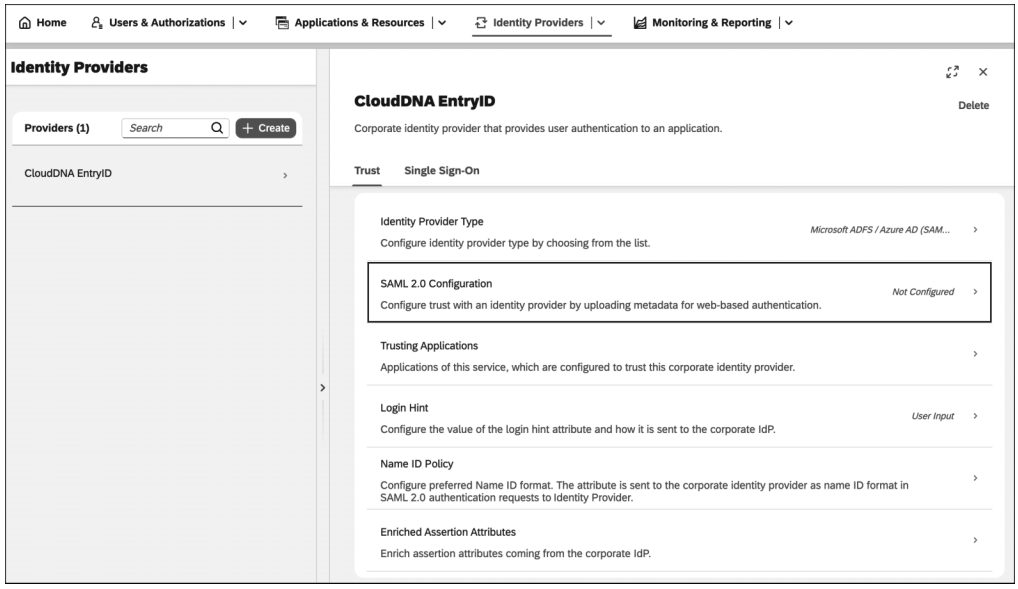

Abbildung 3.24 SAML-2.0-Konfiguration durchführen

Klicken Sie nun im Bereich **Define from Metadata** neben dem Feld **Metadata File** auf den Button **Browse** (siehe Abbildung 3.25).

Wählen Sie die zuvor heruntergeladenen Metadaten aus, und klicken Sie anschließend auf **Save** (siehe Abbildung 3.26).

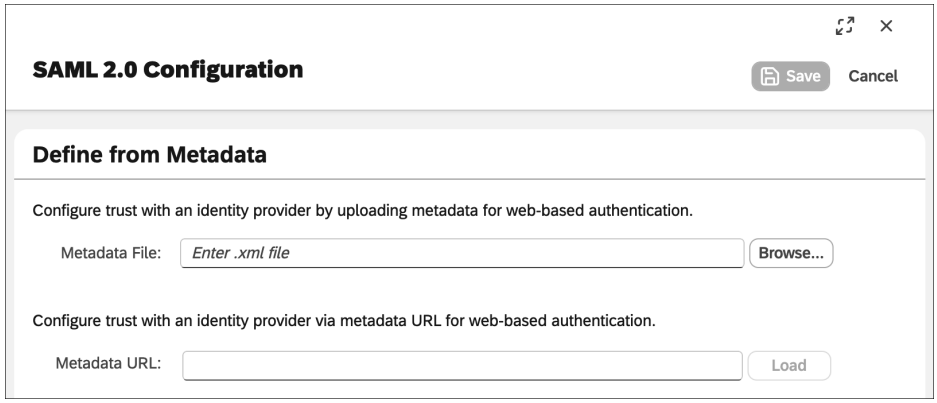

Abbildung 3.25 Upload der Metadaten von Microsoft Entra ID starten

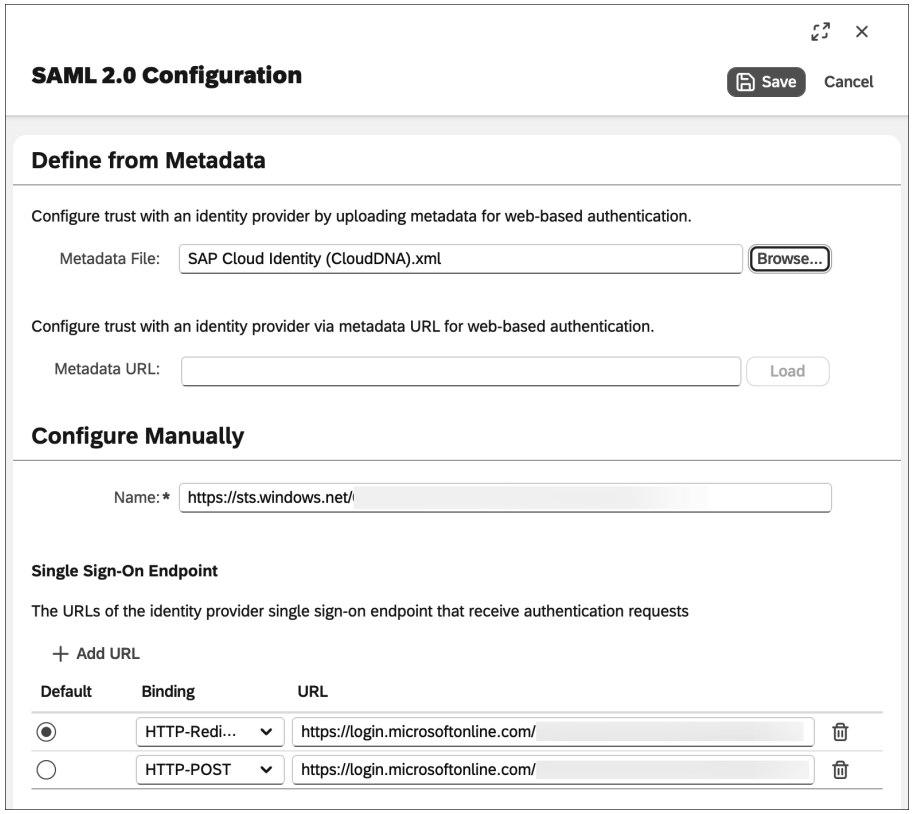

Abbildung 3.26 SAML-2.0-Konfiguration speichern

Die einfachste Art, um die Konfiguration zu testen, ist deren Hinterlegung in der sogenannten Conditional Authentication eines Subaccounts. Navigieren Sie dazu über den Pfad **Applications & Resources • Applications** (siehe Abbildung 3.27).

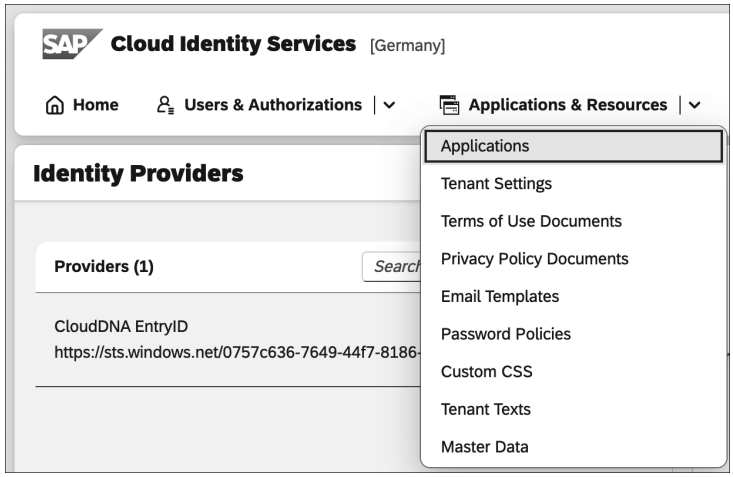

Abbildung 3.27 Applikationen öffnen

Wählen Sie nun im Seitenmenü einen Subaccount aus, der in Identity Authentication registriert ist. Wir haben dazu vorab einen Subaccount mit dem Namen »IAS Proxy Demo« angelegt und einen Trust zu Identity Authentication hergestellt. Klicken Sie in den Details im Bereich **Conditional Authentication** auf den Eintrag **Conditional Authentication** (siehe Abbildung 3.28).

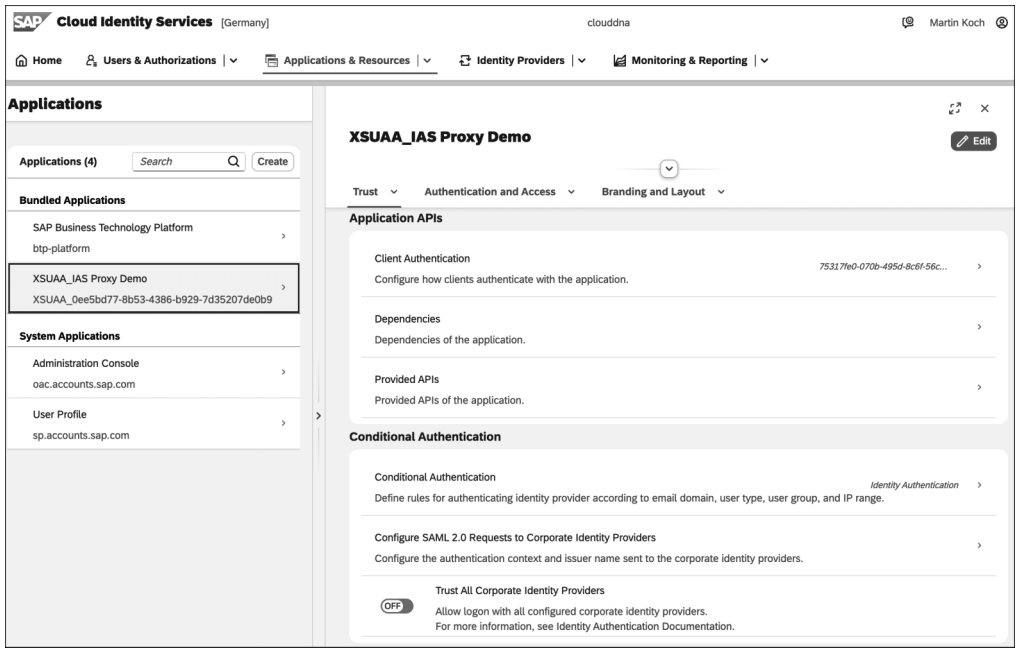

Abbildung 3.28 In die Conditional Authentication eines Subaccounts abspringen