



**アプリケーションの構築を通じた
OutSystems Developer Cloud の機能紹介**

2023年4月に OutSystems Developer Cloud（以下 ODC）がリリースされました。OutSystems 社の公式発表*1 によると ODC は、クラウドネイティブで動作するアプリ開発プラットフォームで、AWS のネイティブクラウドサービスを基盤として、Kubernetes、Linux コンテナ、マイクロサービスなどをサポートしており、OutSystems11（以下 O11）とは異なるアーキテクチャで動作します。

今回は、ODC と O11 は何が違うのか、O11 で構築したものは ODC では使用できなくなるのか、などのポイントについて、ODC の Early Access Program を使用して検証しました。本書では統合開発ツールとしての ODC の特徴や O11 との違いを、簡単なアプリケーション作成からデプロイ実施までの作業を通じてご紹介します*2。

*1 出典 OutSystems 社 HP :

<https://www.outsystems.com/ja-jp/low-code-platform/developer-cloud/>

*2 2023年11月時点での機能になります。

内容

ODC の特徴とその技術要素について	4
ODC の特徴	4
ODC の技術要素.....	4
コンテナ	4
Kubernetes	5
マイクロサービス	5
ODC Studio と ODC Portal	6
ODC でアプリケーションを作成する	7
ODC で作成できるアプリケーション.....	7
自動生成機能を利用したアプリケーションの作成.....	8
アプリケーションのデプロイ.....	14
O11 との違いとまとめ.....	19
アプリケーション	19
モジュールがなくなり、アプリケーションのみ作成.....	19

作成可能なアプリケーション種類.....	19
アプリケーションの利用権限の設定.....	19
公開可能なエレメントの制約.....	19
デプロイ.....	20
終わりに.....	20

ODC の特徴とその技術要素について

ODC の特徴

ODC は先にも述べた通り、クラウドネイティブのアプリ開発プラットフォームです。ODC は、クラウドネイティブのインフラ、管理、運用のベストプラクティスに基づく最新のアーキテクチャが提供されており、Amazon Elastic Kubernetes Service 上で実行されています。

*3 出典：朝日インタラクティブ ZDNET 記事：

「OutSystems ジャパン、クラウドネイティブアプリ向けのローコード開発基盤を国内提供」より

<https://japan.zdnet.com/article/35202635/>

ODC の技術要素

ODC で作成したアプリの実行環境は「Kubernetes」を利用していることからわかるように、コンテナ技術を採用しています。ODC で構築するアプリケーションは、コンテナとマイクロサービスの相性から、マイクロサービスを意識したものである必要があります。

ここで ODC の技術要素であるコンテナ、Kubernetes、マイクロサービスがどのようなものか、それぞれ一般的にどのようなメリット・デメリットがあるか、ODC でのデメリットの対応について簡単に触れておきます。

コンテナ

アプリケーションの動作環境を仮想的に構築する技術で、ホスト OS にアプリケーションやライブラリをまとめたものになります。

メリット：

- 開発環境を効率的に利用することができる
- リソースを節約することができる

デメリット：

- 管理するコンテナが増加すると、管理・運用の手間がかかり煩雑になる

ODC の対応：

- 開発環境でアプリを作成し Publish、デプロイを行うことでコンテナを自動で生成し、検証環境や本番環境へは作成されたコンテナを配置するだけなので管理・運用が煩雑になることはない

Kubernetes

オープンソースのコンテナ・オーケストレーションツールでコンテナをクラスタとして管理・実行する基盤です。

メリット：

- 宣言的なコードによる自律的なインフラとして運用することができる
- 拡張可能な構成となっている

デメリット：

- 学習コストが高い

ODC の対応：

- ODC では Kubernetes を意識することなく開発を行うことができるため、学習コストは不要

マイクロサービス

ソフトウェア開発の技法のひとつ。規模が小さいサービス同士を組み合わせることで一つの大きなアプリケーション・サイトを構築する手法です。

メリット：

- 柔軟性・拡張性・コードの再利用性が高い
- 負荷分散できる
- 開発期間を短縮できる

デメリット：

- 設計・プロジェクト統括への高いスキルが必要
- データ一貫性の担保がしづらい
- API・デバッグ・統合テストの難易度が高い

ODC の対応：

- O11 と比較すると、ODC はアプリ種類によって作成できるエレメントが制限されており、また Dependencies で参照可能なエレメントも制限されているため、アプリケーションの構成を考慮する際はマイクロサービスを意識する必要がある
- ただし、アプリケーションの構築作業・操作は O11 と大きくは変わらないので意識する必要はない

ODC Studio と ODC Portal

ODC での開発は、ODC Studio を利用します。ODC Studio は O11 の Service Studio に相当するものです。また、ODC における運用管理は、ODC Portal を使用します。ODC Portal は、O11 の Service Center、LifeTime、Users が統合されたコンソールで、アプリケーションのデプロイやバージョン管理、ユーザ管理、OutSystems が出力するログの管理ができます。

ODC Studio の画面・操作方法は、この後の「ODC でアプリケーションを作成する」章で触れますので、本章では ODC Portal がどのようなものかをご紹介します。図 1 は ODC Portal 画面の左ペインにてアプリケーションを運用するうえで必要な機能が確認できます。

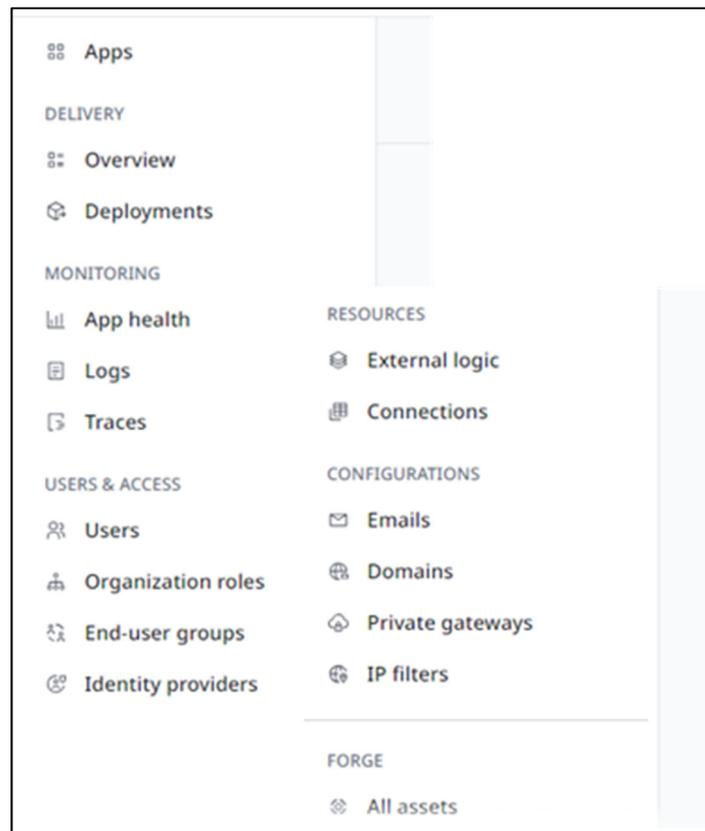


図 1 ODC Portal の左ペイン

※一部 ODC アドオンメニューが含まれています

O11 ではアプリケーション管理は Service Studio や LifeTime、エンドユーザ管理は Users など機能が分かれていましたが、ODC ではそれらアプリケーションやユーザの管理をこの ODC Portal で一元的に行うことができるようになりました。

なお、ODC Portal の詳細については以下の【参考 URL】をご参考ください。

【参考 URL】

https://success.outsystems.com/ja-jp/documentation/outsystems_developer_cloud/getting_started_with_odc/ui_overview_of_odc_portal_and_odc_studio/

ODC でアプリケーションを作成する

ODC で新規のアプリケーションを作成します。実際の ODC Studio の画面を確認しながら進めていきます。

ODC で作成できるアプリケーション

ODC は O11 と異なり、作成できるアプリケーションは、App と Library の 2 種類だけです。また、ODC ではモジュールの概念がなくなっています。新しいアプリを作成すると、以下の図 2、3、4 の順で表示されます。

※図 2 の後にアプリ動作環境を選択する画面（Web/Tablet/Phone）が表示されますが O11 と同じであるため省略しています。

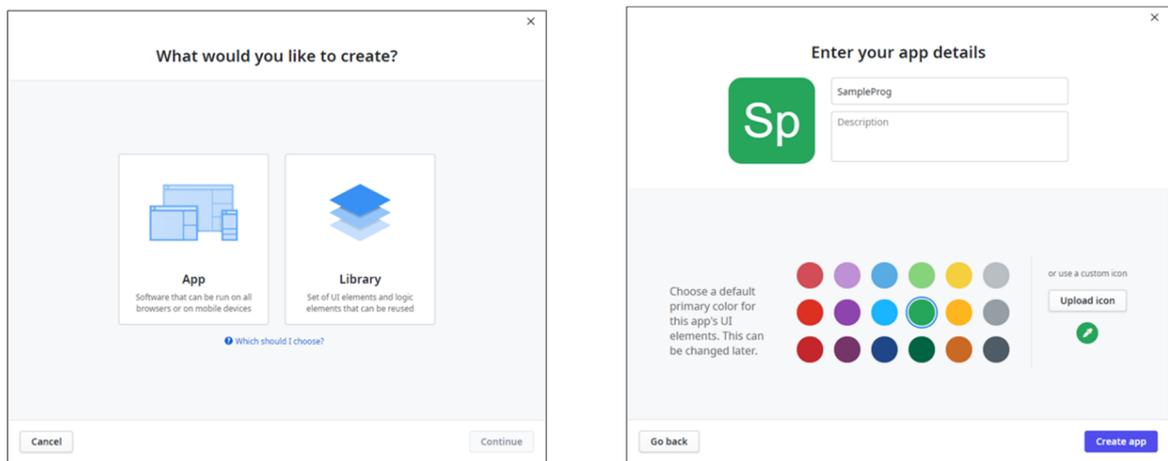


図 2 新しいアプリケーションを作成する画面

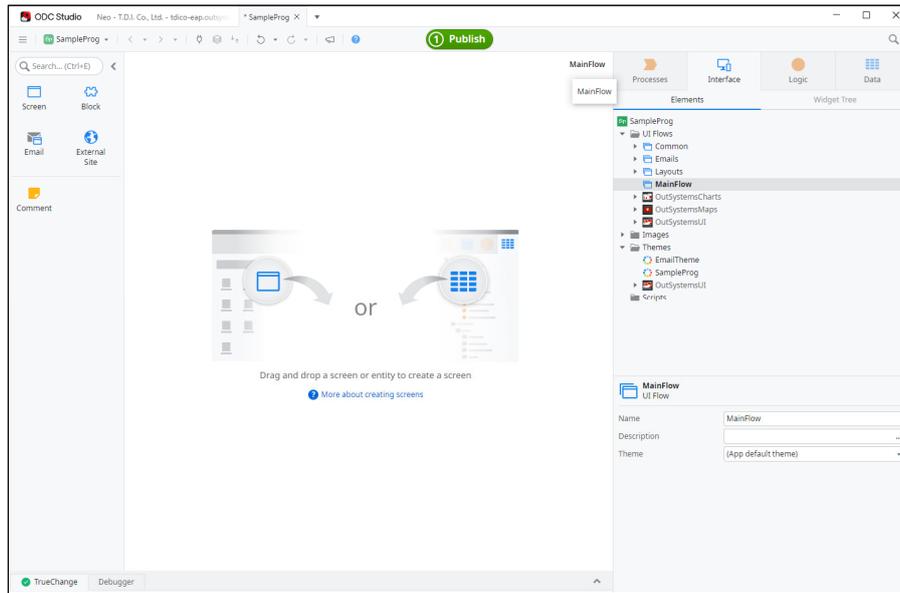


図 3 アプリケーションの初期画面

O11の場合、図2と3の間にモジュールを作成する画面がありますが、ODCではモジュール作成画面は表示されず、アプリケーションの初期画面が表示されます。このことからODCでは作成するのはアプリケーションのみとなり、モジュールの作成は必要ない、ということがわかります。

自動生成機能を利用したアプリケーションの作成

ODCでのアプリケーションを作成する手順について確認します。ここではExcelデータを取り込み、Entityを自動作成します。自動生成されたEntityからスキヤフォールディング機能を利用して画面を自動生成し、その過程においてO11と違いがあるかを見ていきます。

まずExcelデータをODC StudioにImportします。

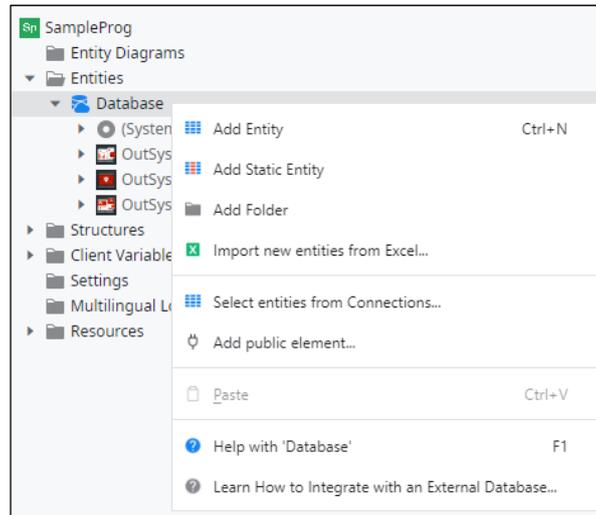


図 4 Excel データの Import

Excel データの Import 後の Entity と Attribute の状態です。

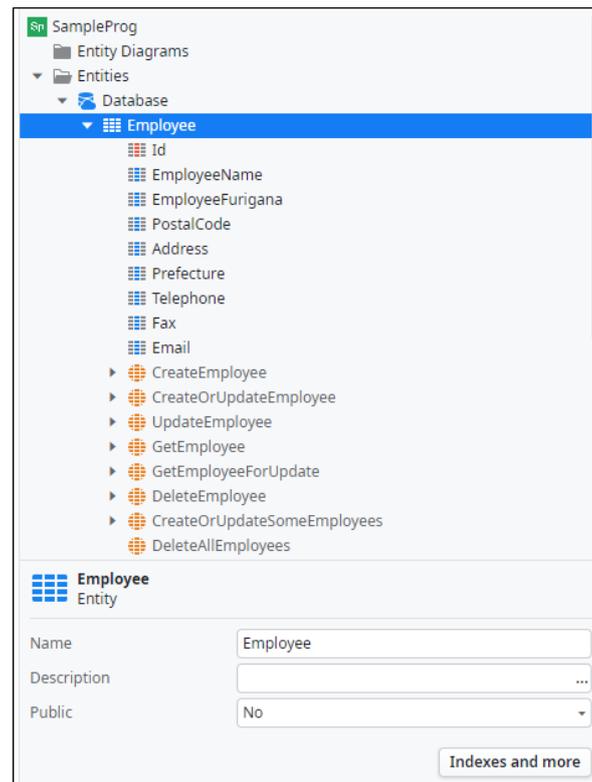


図 5 自動生成された Employee Entity

大きな違いはありませんが、Entityのプロパティに、O11にはあった「Expose Read Only」がなくなっています。これは ODC において、Entity を含め公開可能なエレメントの扱いが変わっているためです。詳細については後述の「O11 との違いとまとめ」に記載します。

続けてスキャフォールディング機能を使って画面を自動生成します。

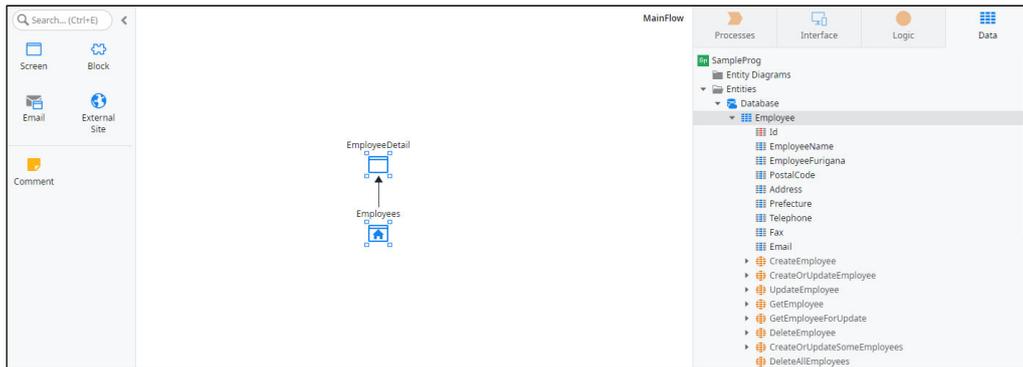


図 6 スキャフォールディング実行後の状態

O11 と同じく、一覧画面と詳細画面が自動生成されました。生成されたそれぞれの画面がどのように作成されたかを見えます。

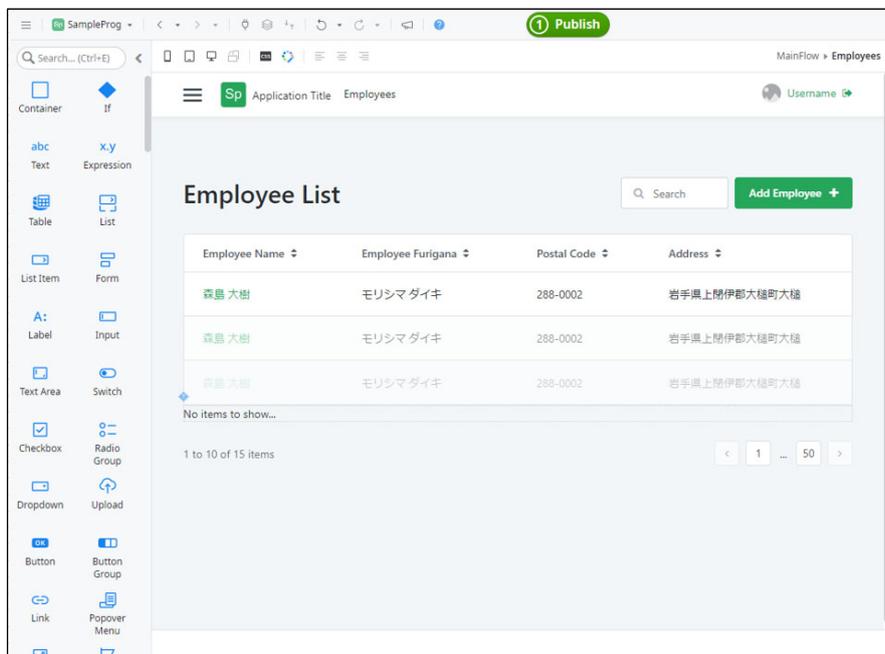


図 7 自動生成された一覧画面

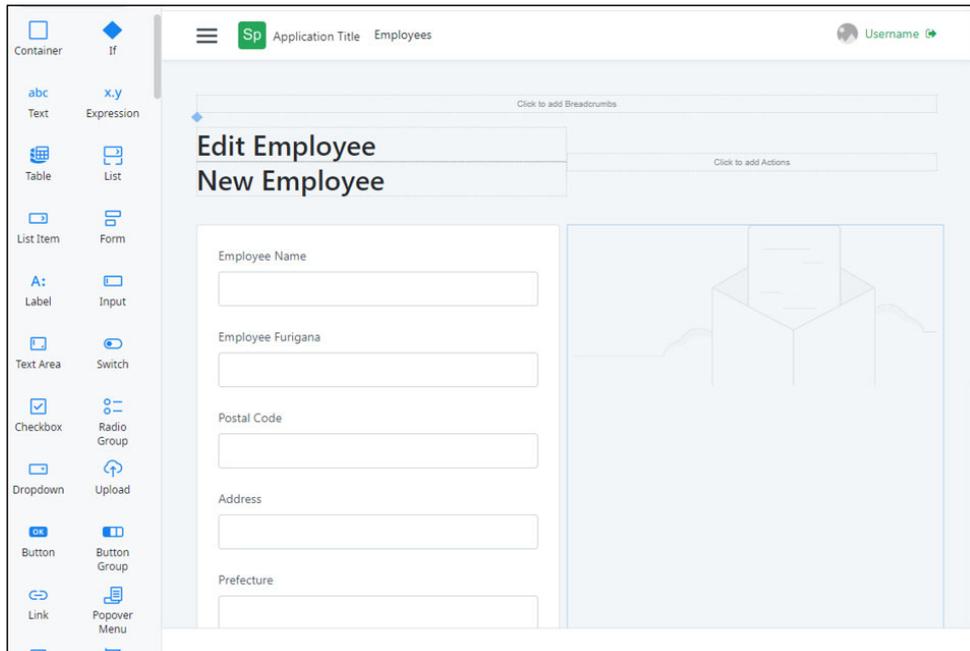


図 8 自動生成された詳細画面

自動生成された画面イメージは、O11 と大きな違いはないようです。何も修正せずに Publish してアプリケーションを実行してみます。

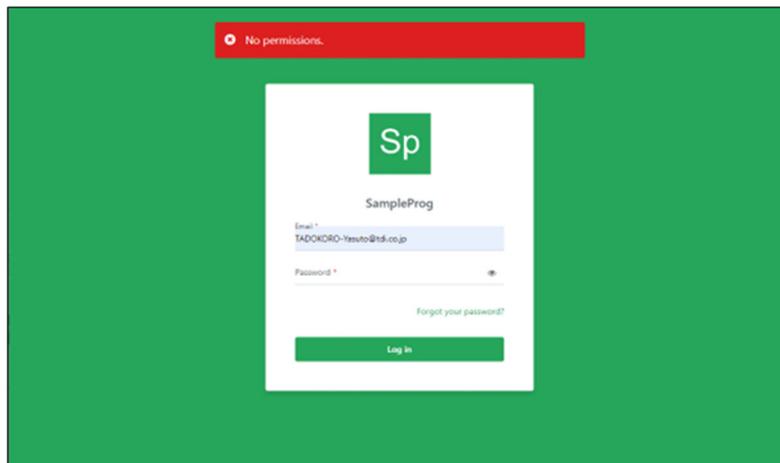
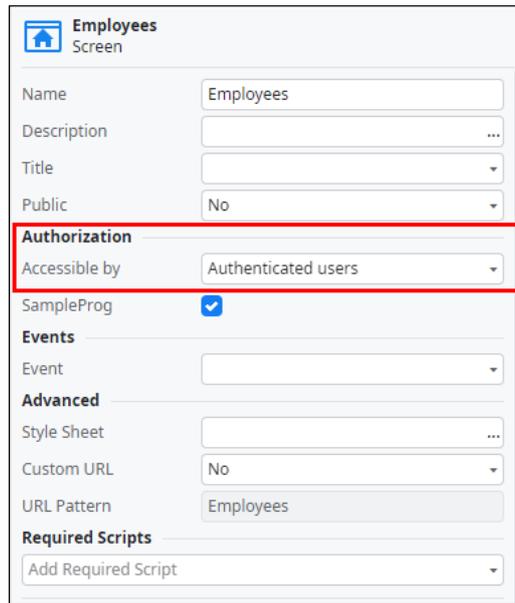


図 9 ログインエラー

ログインで Permission エラーが表示されました。表示されているエラーメッセージは「No Permissions」とありますので、権限に問題がありそうです。アプリケーションの設定と ODC Portal でユーザ権限を確認してみることにします。

まず、ODC Studio の画面プロパティでアプリケーションの権限を確認します。



The screenshot shows the 'Employees' screen properties in ODC Studio. The 'Authorization' section is highlighted with a red box, indicating the 'Accessible by' dropdown is set to 'Authenticated users'. Other visible fields include Name (Employees), Description, Title, Public (No), SampleProg (checked), Events, Advanced (Style Sheet, Custom URL, URL Pattern), and Required Scripts.

図 10 Employee 画面プロパティ

上図のとおり、O11 では「Roles」でしたが、ODC では「Authorization」と名称が変わっています。また、「Anonymous」「Registered」をチェックボックスによる ON/OFF だったものが、「Accessible by」という選択リストに変更され「Everyone/Authenticated users」のいずれかを選択する形式になっています。スキャフォールディングで自動生成される画面は、デフォルトで「Authenticated users」が選択（O11 は「Registered」のみチェック ON）された状態であるため、アプリケーションのユーザは「Authenticated（認証された user）」である必要ということがわかりました。

次にログインしたユーザの権限を確認します。アプリケーションの利用権限は ODC Portal の Users で確認できます。以下の手順で行います。

- ① ODC Portal 画面の Users を選択して、ログインしたユーザを表示。
- ② End-user access タブを押す。
- ③ 対象のアプリケーションを検索。作成したアプリケーションは「SampleProg」なので検索窓に「SampleProg」と入力。
- ④ 検索を確認。

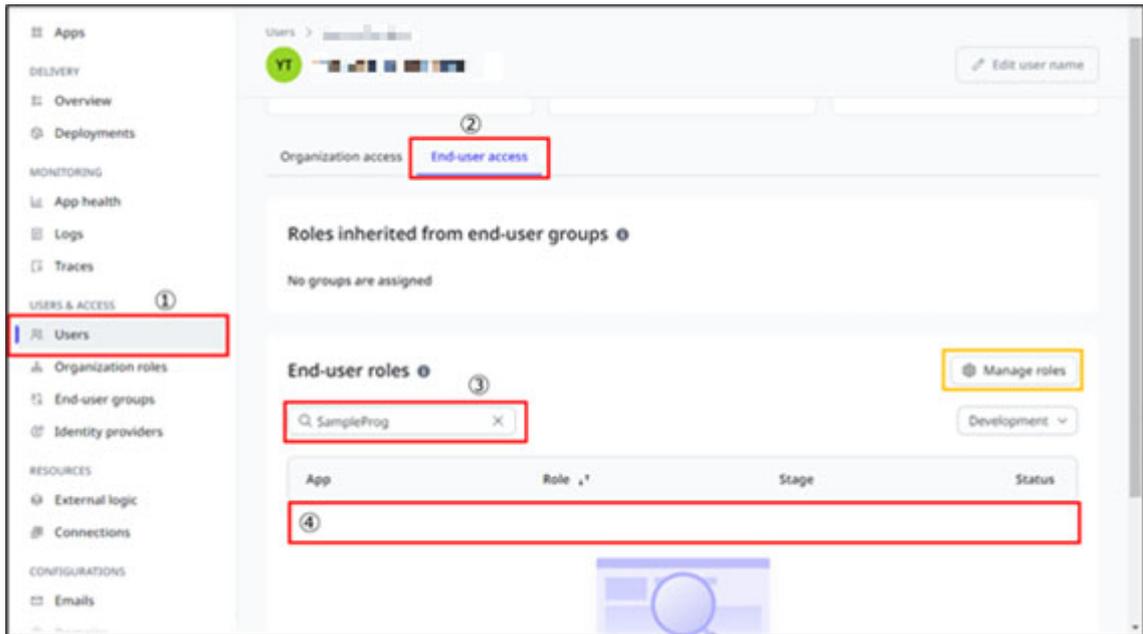


図 11 ODC Portal の Users

図 11 のとおり、検索した結果「SampleProg」が表示されなかったため、SampleProg アプリケーションにアクセス権限がないことがわかりました。そこでアプリケーションに権限を付与します。アプリケーションへの権限付与は、同じく Users で行います。同画面の「Manage roles」ボタン（図 11 の黄色枠で囲った箇所）を押すと、一覧に「SampleProg」アプリが表示されるので、チェックを入れ「Save」ボタンを押します。これで権限付与は完了です。

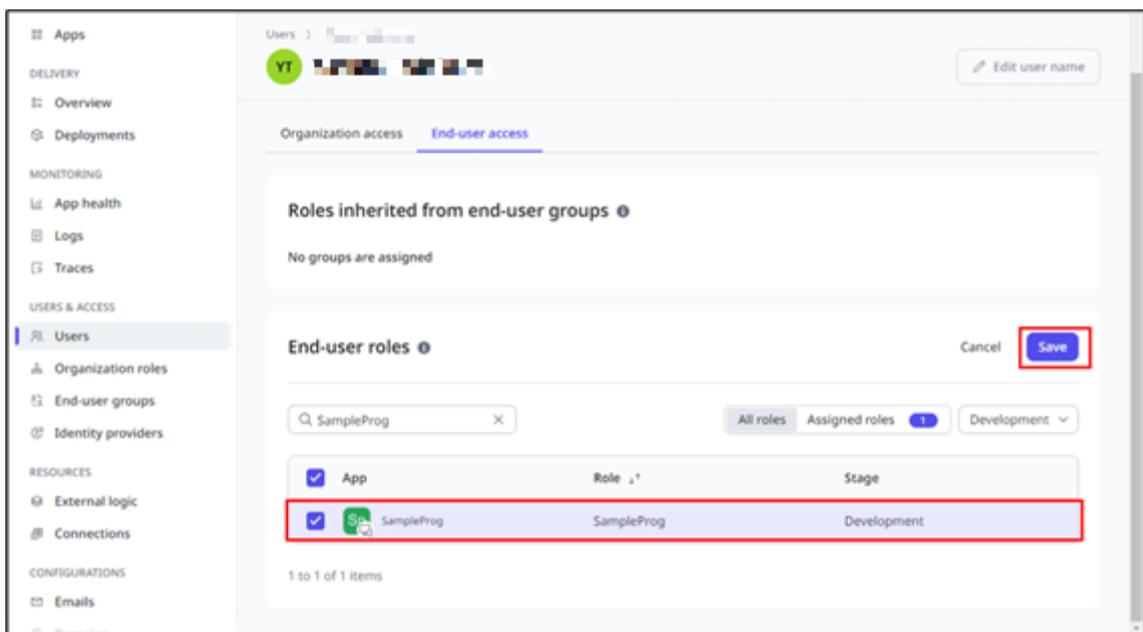


図 12 Users アプリ一覧で権限を付与

再度アプリケーションを実行し、同じユーザでログインします。

Employee Name	Employee Furigana	Postal Code	Address
情報 一部	ジョウホウ イチロウ	064-0941	北海道札幌市中央区旭ヶ丘
情報 次郎	ジョウホウ シロウ	455-0852	北海道札幌市豊平区美園二条
情報 三部	ジョウホウ サブロウ	025-0311	兵庫県朝来市和田山町久世田
情報 四郎	ジョウホウ シロウ	990-2473	大阪府八尾市荘内町
情報 五郎	ジョウホウゴロウ	891-1413	広島県東広島市西条御条町
情報 六郎	ジョウホウ ロクロウ	496-0805	新潟県糸魚川市蒲池
橋本 絵美	ハシモト エミ	920-0910	三重県四日市市榑町北五味塚
野口 楓花	ノグチ フウカ	510-0096	埼玉県行田市若小玉
瀬川 肇	セガワ ユタカ	410-2302	熊本県熊本市東区昭和町
出口 晴雄	デグチ ハルオ	300-4243	北海道網走市本町東藻琴新宮

図 13 ログイン後のアプリケーションの画面

今度は、エラーは表示されず一覧画面が表示されました。画面を自動生成する機能については O11 との違いはありませんが、画面など機能を利用するため権限設定は、操作が異なっていました。

アプリケーションのデプロイ

アプリケーションをデプロイします。ODC ではコンテナ技術を採用していることは既にお伝えしました。ODC で開発したアプリケーションを Publish してデプロイする手順が O11 の手順と変わっていないことを「ODC でアプリケーションを作成する」の章で確認できましたが、ODC ではアプリケーションを Publish ・デプロイすることでコンテナを生成しています。生成されたコンテナは、開発ステージ（O11 では環境と呼ばれていました）に配置されます。テストステージ、本番ステージへは開発ステージに配置された同じコンテナを配置するのみで、O11 のように各環境で Publish する必要がなくなっています。

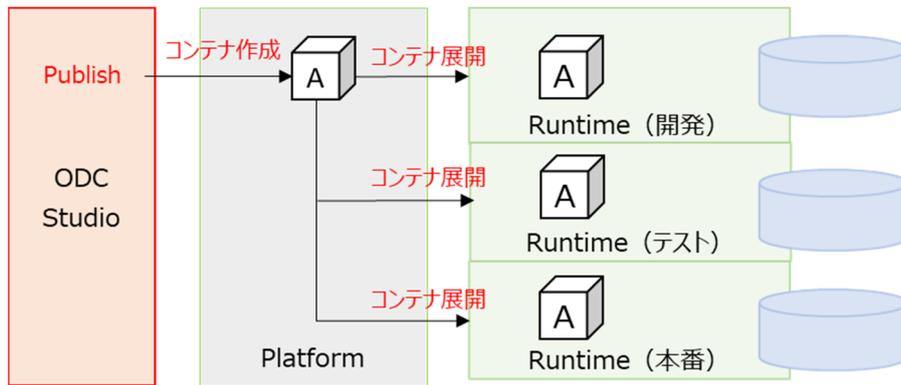


図 14 ODC のアプリケーション展開

これにより、テストステージ、本番ステージへ素早くアプリケーションをリリースすることができるようになりました。一方、テスト・本番ステージにリリースされたアプリケーションのソースコードを個別に修正することができなくなったことを意味しています。つまり、ODC のアプリケーションを保守・運用するために、ホットフィックスができないことを考慮する必要があります。

ODC のデプロイ手順を見ていきます。まずアプリケーションのリリース状況がどのような状態であるかを確認します。ODC Portal の「DELIVERY」から「Overview」を選択しデプロイ対象アプリを検索します。図 15 は SampleProg アプリを検索した結果の画面状態です。SampleProg は Development、つまり開発ステージにのみデプロイされていることがわかります。

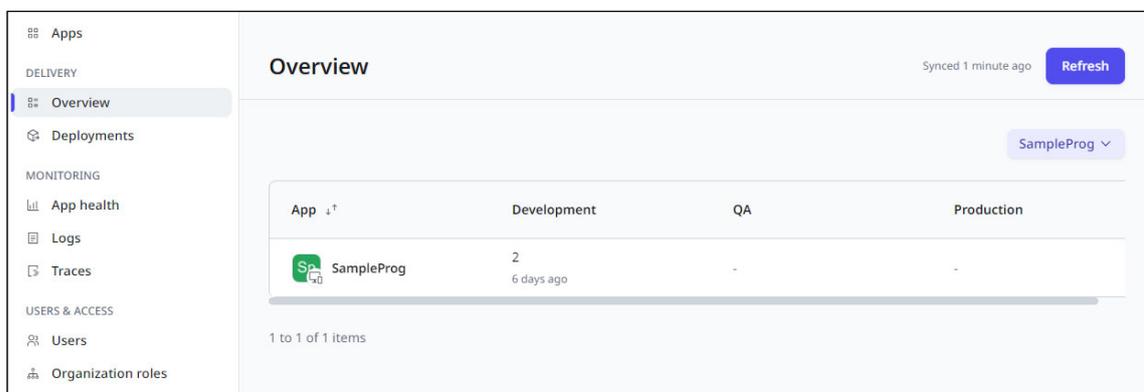


図 15 ODC Portal の Overview 画面

では、SampleProg を QA（テスト）ステージにデプロイします。

「DELIVERY」から「Deployments」を選択し Deployments 画面を表示します。

Deployments 画面では「対象のアプリ」を「どのステージ」に配置するか、を選択します。指定できる項目は図 17 にあるとおりです。①～④はアプリケーションの検索条件、⑤はアプリケーションのデプロイ先（テスト・本番）を指定して、デプロイ画面へ遷移します。

- ① 対象のアプリケーション名
- ② アプリケーションの Status : All/Running/Finished successfully/Finished with error
- ③ デプロイした日
- ④ デプロイ先 : QA/Production
- ⑤ Deploy to : QA stage/Production stage

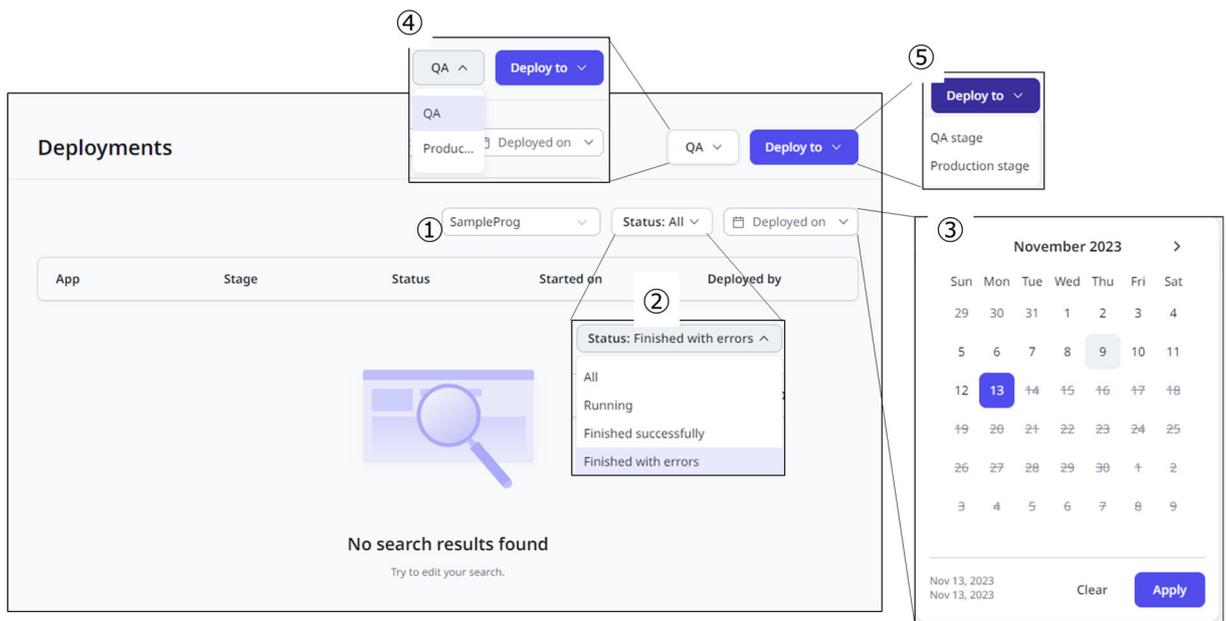


図 16 Deployments 画面

SampleProg アプリをデプロイします。Deployments 画面の「Deploy to」ボタンを押し「QA stage」を選択、Deployment to QA stage 画面に遷移して、デプロイするアプリを指定します。図 17 はデプロイする SampleProg アプリを指定した状態です。

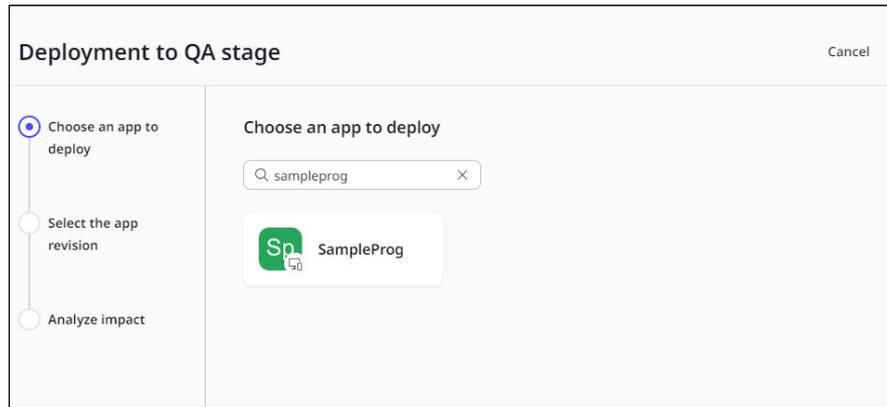


図 17 デプロイアプリを指定した状態

表示されている SampleProg を押すとデプロイするアプリケーションのバージョン一覧が表示されます（図 18）。

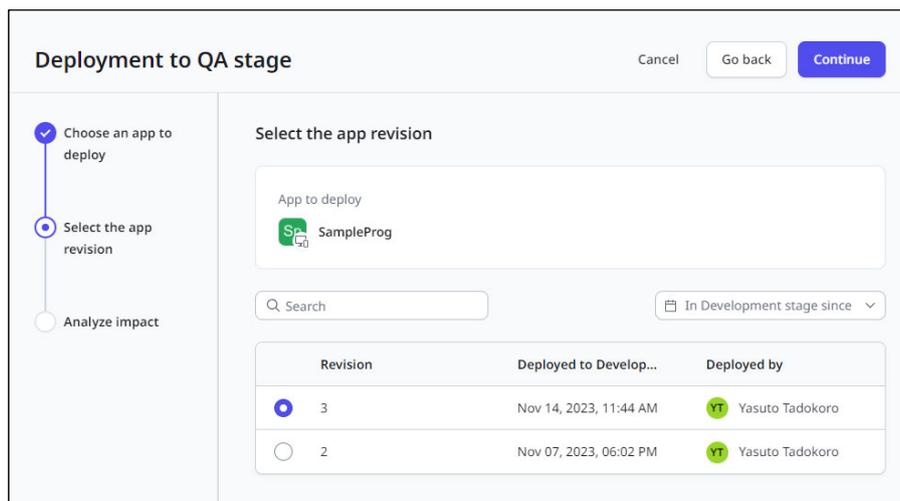


図 18 デプロイバージョンの選択

開発ステージでアプリケーションを Publish するとその回数分 Revision が作成されます。一覧にはその Revision が表示されますので、QA ステージにデプロイする Revision を選択し（図 18 は Revision 3 を選択）画面右上の Continue ボタンを押します。OutSystems はデプロイ対象のアプリをデプロイして問題がないか評価・分析を行います。問題がなければ図 19 の画面が表示されます。

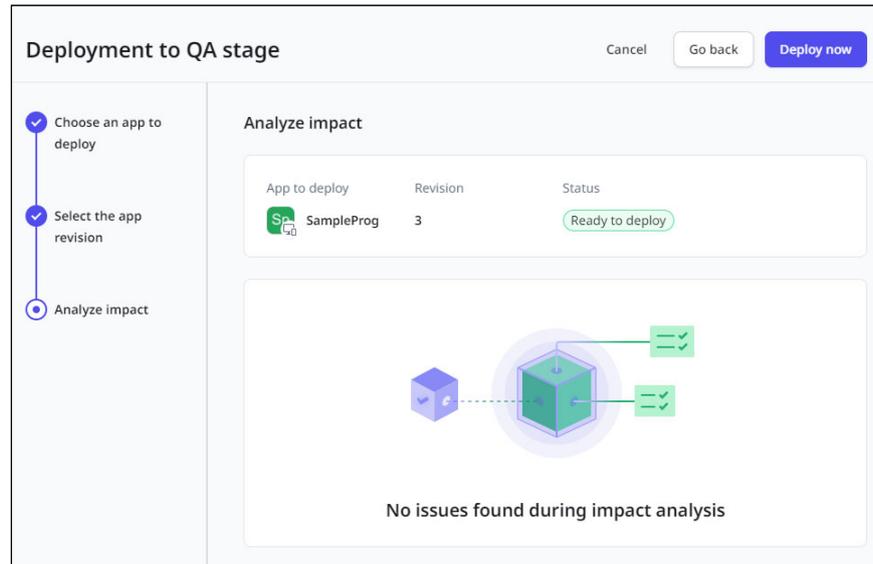


図 19 デプロイ対象アプリ評価・分析結果画面

「Deploy now」ボタンを押し、デプロイを実行します。デプロイが成功すると図 20 の画面が表示されます。

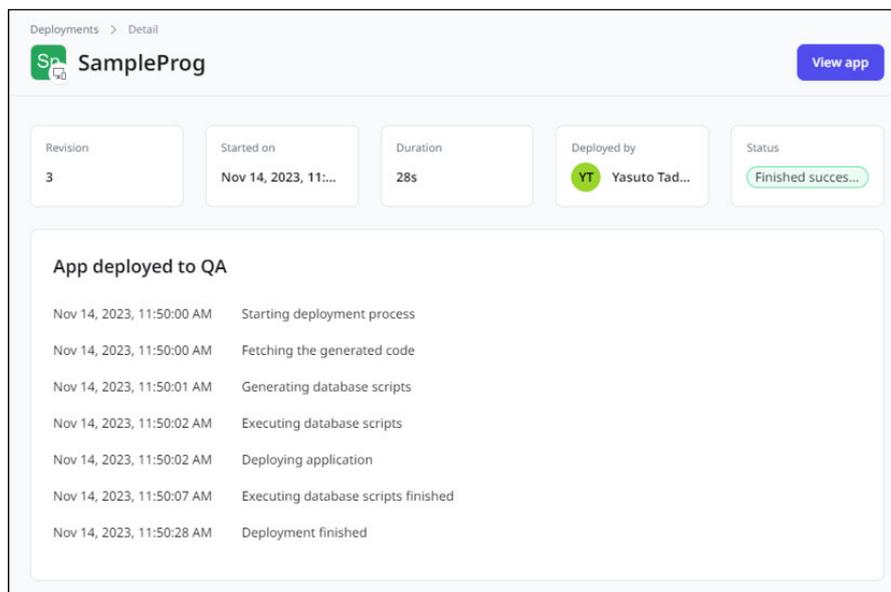


図 20 デプロイ結果

この画面の右上にある「View app」ボタンを押すとアプリケーションが起動し、動作を確認することができます。

O11 との違いとまとめ

Excel データ取込み、スキャフォールディング機能を使って簡単なアプリケーションを作成・デプロイを通じて ODC と O11 のアプリケーション作成にどのような違いがあるか見てきました。以下に違いを整理します。

アプリケーション

モジュールがなくなり、アプリケーションのみ作成

O11 ではアプリケーションには最低一つのモジュールがありましたが、ODC では「ODC で作成できるアプリケーション」で確認できた通り、アプリケーションのみとなっています。これは ODC がコンテナ、マイクロサービスへ対応したことによるもので、ODC の大きな特徴の一つです。

作成可能なアプリケーション種類

O11 では Reactive Web App や、Traditional Web（現在 O11 の新規購入では作成に制限があります）、Service などが選択できますが、ODC では「ODC で作成できるアプリケーション」の章でご覧いただいた通り、ODC で新規にアプリケーションを作成すると「App/Library」のみ選択可能となっています。

アプリケーションの利用権限の設定

「自動生成機能を利用したアプリケーションの作成」の章で確認したように、ODC では構築したアプリケーションの利用権限の付与方法が O11 とは異なり、エンドユーザに対してアプリ単位で利用権限を付与する形となっています。

公開可能なエレメントの制約

「自動生成機能を利用したアプリケーションの作成」で触れましたが、Entity の公開が O11 とは異なっています。ご紹介した通り Entity は「Expose read only」がなく、Public を Yes

で公開したとしても、公開先では読み取り専用となります。Entity 以外でも Role はアプリケーション内で閉じており、他のアプリケーションでは参照できなくなっています。

Name	Employee
Description	
Public	No
Expose Read Only	Yes

ODCではExpose Read Only がなくなっている

図 21 O11 の Entity プロパティ

デプロイ

「アプリケーションのデプロイ」の項でご紹介した通り、ODC で作成したアプリケーションは、開発ステージで Publish され、コンテナを生成しデプロイされ、QA/本番ステージへは、開発ステージにあるコンテナをデプロイします。これにより、QA/本番ステージには同じコンテナが配置されることになり、QA/本番ステージでソースコードを直接修正することができなくなるため、ホットフィックスの対応が不可となりますが、これは CI/CD のベストプラクティスを促進するための現時点（2024 年 1 月）での仕様となっています。

そのため、ODC を運用するうえで緊急バグ対応を行う場合の手順・体制などは予め検討しておく必要があります。

終わりに

本書でご紹介した ODC の機能や O11 との違いはほんの一部です。ODC の理解の一助となれば幸いです。

OutSystems 社は、現在提供している OutSystems11 製品の On Premise 版と Cloud 版は徐々に ODC へシフトしていく計画を立てております。O11 で構築したアプリケーションは、2027 年 3 月までは実行可能で、正式サポート終了の少なくとも 2 年前までには同社より連絡がある予定です。

tdi は、今後も ODC のみならず、OutSystems の機能検証を行うとともに、OutSystems の情報発信を予定しております。また、OutSystems の機能や技術について十分な知識を持った多くの技術者を有しており、資格保有者数は国内トップクラスです。ローコード開発が一般的に注目される以前（2016 年）から重ねた OutSystems 開発の実績をもとに、IT 戦略コンサルティングや OutSystems 導入から運用までをトータルサポートします。また、お客様に合わせ

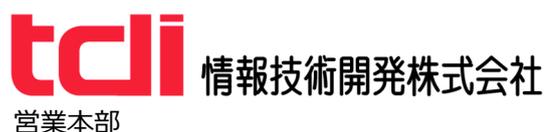
た人材育成や内製化もご支援いたします。本件を含め、何かお困りごとがございましたらどうぞお気軽にお問合せください。

【ローコード開発基盤「OutSystems」】

<https://www.tdi.co.jp/outsystems/>

【お問い合わせ】

<https://tdi.smktg.jp/public/application/add/1095>



東京:〒163-1332 東京都新宿区西新宿六丁目 5 番 1 号 新宿アイランドタワー32 階

TEL : 03-5325-4811 (代表) FAX 03-5325-4812

中部:〒451-6027 愛知県名古屋市西区牛島町 6 番 1 号 名古屋ルーセントタワー27 階

TEL 052-571-6871 (代表) FAX 052-571-3856

関西:〒530-0005 大阪府大阪市北区中之島二丁目 2 番 7 号 中之島セントラルタワー20 階

TEL.06-6201-7739(代表) FAX.06-6201-7740

九州:〒812-0013 福岡県福岡市博多区博多駅東二丁目 10 番 1 号 福岡ビル S 館 7 階

TEL.092-451-8218(代表) FAX.092-474-7379

OutSystems® は OutSystems-Software Em Rede S.A.の登録商標です。

AWS は、Amazon.com, Inc.またはその関連会社の商標です。

Azure は、米国 Microsoft Corporation の、米国およびその他の国における登録商標または商標です。