Catena-X PoC Developer Guide

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author** | **Description** |
| 1.0 | June 2nd, 2021 | Martin Raepple | Initial Version |

Table of Contents

[Overview 3](#_Toc73655745)

[PoC Repository Structure 3](#_Toc73655746)

[Workplace Setup 4](#_Toc73655747)

[Development Workflow 4](#_Toc73655748)

[Access Management 5](#_Toc73655749)

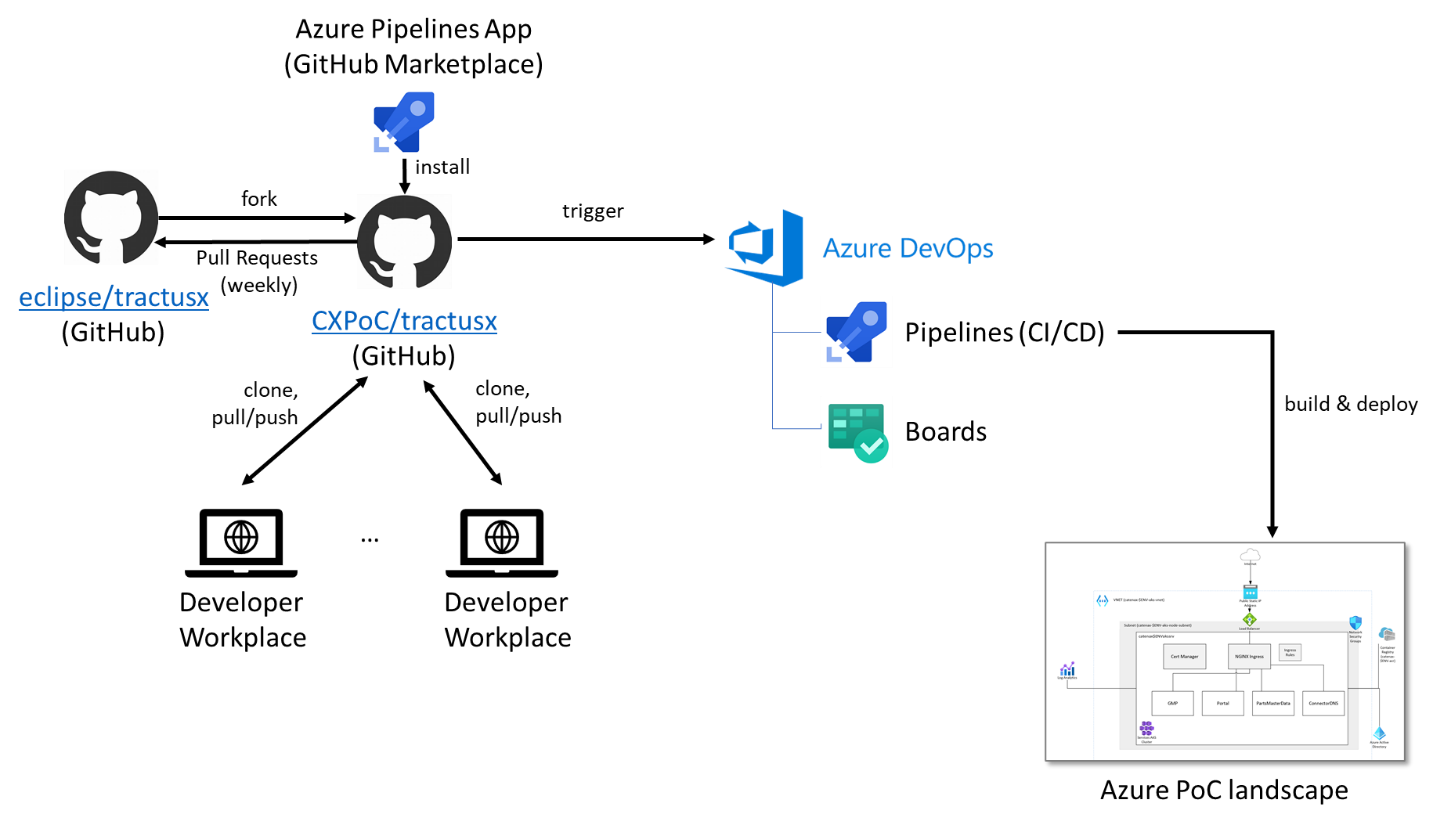
[Access management in the source code repository 6](#_Toc73655750)

[Access to the Azure subscription 6](#_Toc73655751)

[Managing PoC resources in the Azure subscription 6](#_Toc73655752)

# Overview

The following diagram shows the main components of the setup for development, continuous integration and deployment in the context of the Catena-X Proof-of-Concept (PoC)



* The PoC code is hosted on the Eclipse foundation’s GitHub organization’s Tractus-X repository at <https://github.com/eclipse/tractusx>. An overview of the Tractus-X Eclipse project can be found at [Eclipse Tractus-X | projects.eclipse.org](https://projects.eclipse.org/projects/automotive.tractusx). Only a few [PoC project team](https://github.com/eclipse/tractusx/graphs/contributors) members have committer access to this repository. Therefore we’ve setup a fork of this repository in a separate GitHub organization account [CXPoC (github.com)](https://github.com/CXPoC).
* [CXPoC (github.com)](https://github.com/CXPoC) forks the Tractus-X repository from Eclipse which can be accessed at <https://github.com/CXPoC/tractusx>. Here, all developers in the PoC team are members with administrative permissions to all repositories. This means they can clone, pull, push and add new collaborators to all repositories. As agreed in the Catena-X PoC board, we will create pull requests (PRs) every end of the week towards the Eclipse GitHub repository.
* The forked [repository](https://github.com/CXPoC/tractusx) is connected to the Azure DevOps environment for the PoC (<https://dev.azure.com/tractusx/Tractus-X>) via the Azure Pipelines Apps from the GitHub marketplace.
* Continuous integration and continuous deployment (CI/CD) is configured via [Pipelines](https://docs.microsoft.com/en-us/azure/devops/pipelines/?view=azure-devops) in the Azure DevOps project. Each service in the PoC (e.g. Connector DNS, Parts Master Data, …) defines its pipeline as YAML file in its corresponding folder in the repository (see next paragraph for more details). As an example, the Connector DNS service defines its CI/CD pipeline in the file [connectordns/azure-pipelines-connectordns.yml.](https://github.com/CXPoC/tractusx/blob/main/connectordns/azure-pipelines-connectordns.yml)

# PoC Repository Structure

The repository is using the following folder structure to separate the different services in the PoC:

* /connectordns: Code and CI/CD automation for the connector “Domain Name Service” (DNS) which is responsible for resolving an organization’s OneID to a connector instance ID.
* /gpm: Code and CI/CD automation for the Business Partner Management service (aka “Geschäftspartermanagement”, or GPM) to retrieve the “golden record” for a company’s business master data
* /kmuuploadapp: Backend logic to manage and store CSV uploads of parts master data from SMEs (aka “kleine- und mittelständische Unternehmen (KMUs)).
* /partsmasterdata: Code and CI/CD automation for the central parts master data service
* /portal: UI frontend of the Catena-X portal
* /terraform: Terraform scripts for the PoC landscape on Azure to automate the infrastructure deployment
* /manifests: Kubernetes manifests to deploy the services to the central Azure Kubernetes Service (AKS) cluster and other services (such as IDS Connectors)

# Workplace Setup

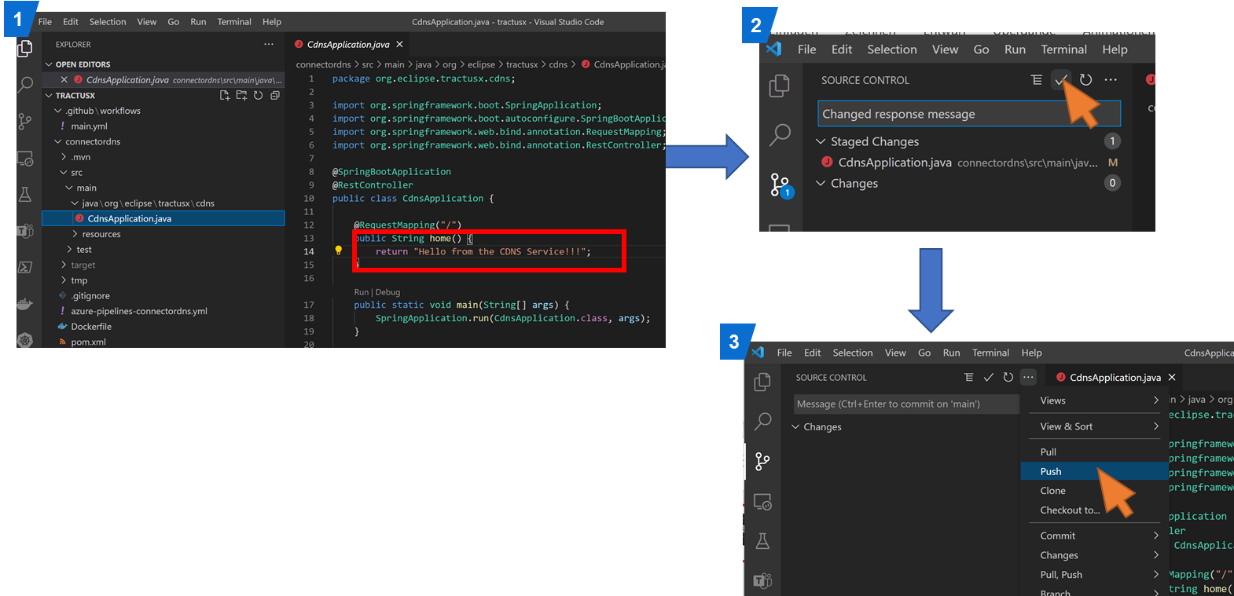
This guide makes no assumptions on the tools to use for development. As a developer, you can choose a simple text editor or integrated development environment (IDE). To create a copy of the Tractus-X code repository, install a Git client of your choice from [Git - Downloads (git-scm.com)](https://git-scm.com/downloads). Then clone the forked Tractus-X repository from the [CXPoC (github.com)](https://github.com/CXPoC) organization with the command

git clone https://github.com/CXPoC/tractusx.git

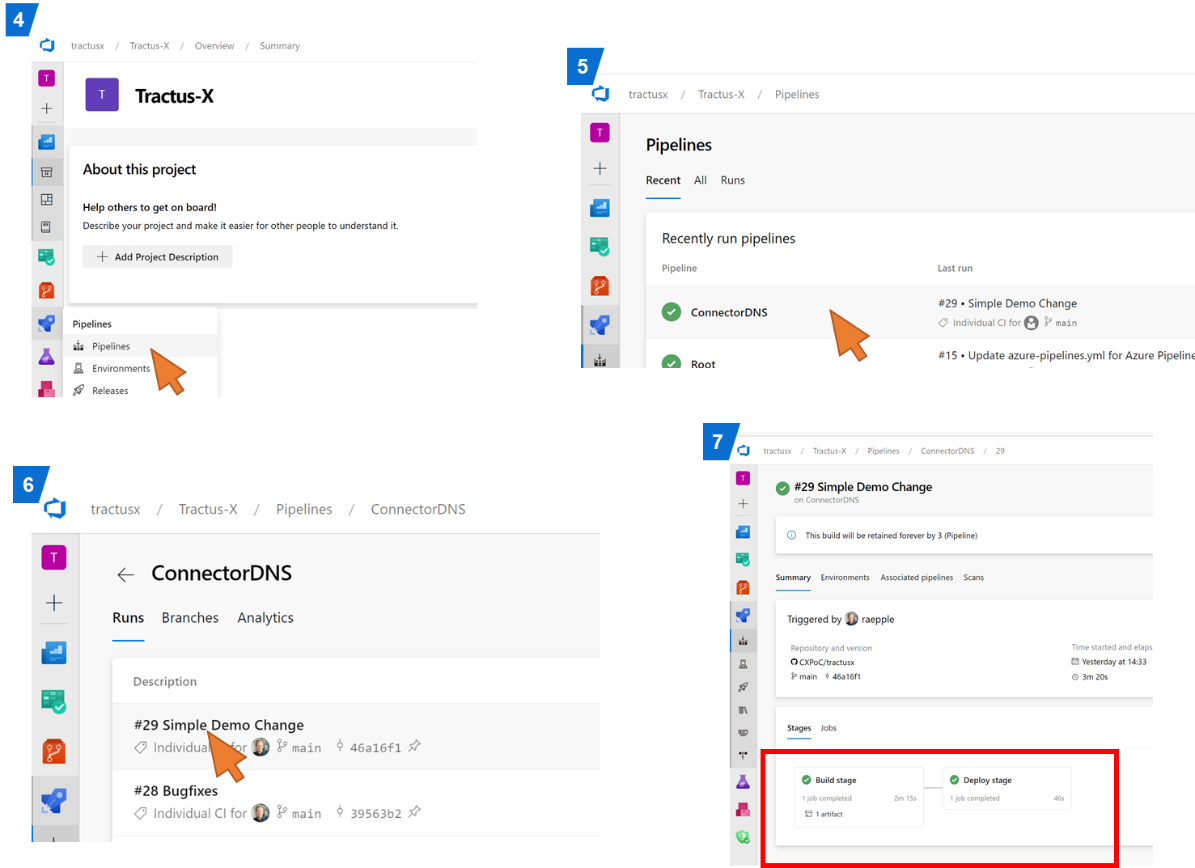
The following section on a typical developer workflow uses Visual Studio Code (<https://code.visualstudio.com/Download>) to edit the code of a service.

# Development Workflow

Let’s go though a simple development workflow step-by-step. We’ll take a look at source code management and CI/CD integration. This example is based on the Connector DNS service (folder [/connectordns](https://github.com/CXPoC/tractusx/tree/main/connectordns) in the Git repository). This service has a pipeline setup in Azure DevOps.



* Step 0: Open the root directory of the local clone of the Git repository with an IDE of your choice (here Visual Studio Code)
* Step 1: Implement the changes in the source code and save them (here we changed the message text returned by the home() method of class [tractusx/CdnsApplication.java at main · CXPoC/tractusx (github.com)](https://github.com/CXPoC/tractusx/blob/main/connectordns/src/main/java/org/eclipse/tractusx/cdns/CdnsApplication.java). Save the changes.
* Step 2: Switch to the Source Control view and stage your changes. Enter a commit message and click on Commit.
* Step 3: Push the changes to the remote repository on GitHub.



* Step 4: Go to the [Tractus-X project](https://dev.azure.com/tractusx/Tractus-X) on Azure DevOps and select *Pipelines* from the left-side navigation
* Step 5: Select the pipeline for the service you’ve just modified
* Step 6: You should see a new pipeline run appear in the list. Depending on the size of the pipeline, it may take a few minutes to complete the run. Click on the most recent item in the list to see the details.
* Step 7: Verify that all stages of the pipeline were executed successfully. If one of them failed, you can check for the reason by clicking on the stage to see the execution log.
* Step 8: If the pipeline finished successfully, verify the change(s) you made. For the Connector DNS service in this example, we changed the message text returned by the root path. Open a browser with the service URL as specified by the ingress rule

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

# Access Management

Access for developers is managed on two levels:

1. Access to the source code in the [CXPoC tractus-x repository](https://github.com/CXPoC/tractusx)
2. Access to the Azure subscription for the PoC

While all developers in the PoC team need committer access to the source code repository, not every developer may require access to the Azure subscription to create or modify the PoC resources deployed to Azure, such as the AKS cluster or container registry. The following sections describe the steps to request or change access to these resources.

## Access management in the source code repository

For the [CXPoC tractus-x repository](https://github.com/CXPoC/tractusx), the following people can add you as a member: [People · CXPoC (github.com)](https://github.com/orgs/CXPoC/people?query=role%3Aowner).

Please send your **GitHub account name** to one of these contacts to request access as a member. Members are able to clone, pull and push all repositories. If you need Owner privileges, please also contact one of the people above.

## Access to the Azure subscription

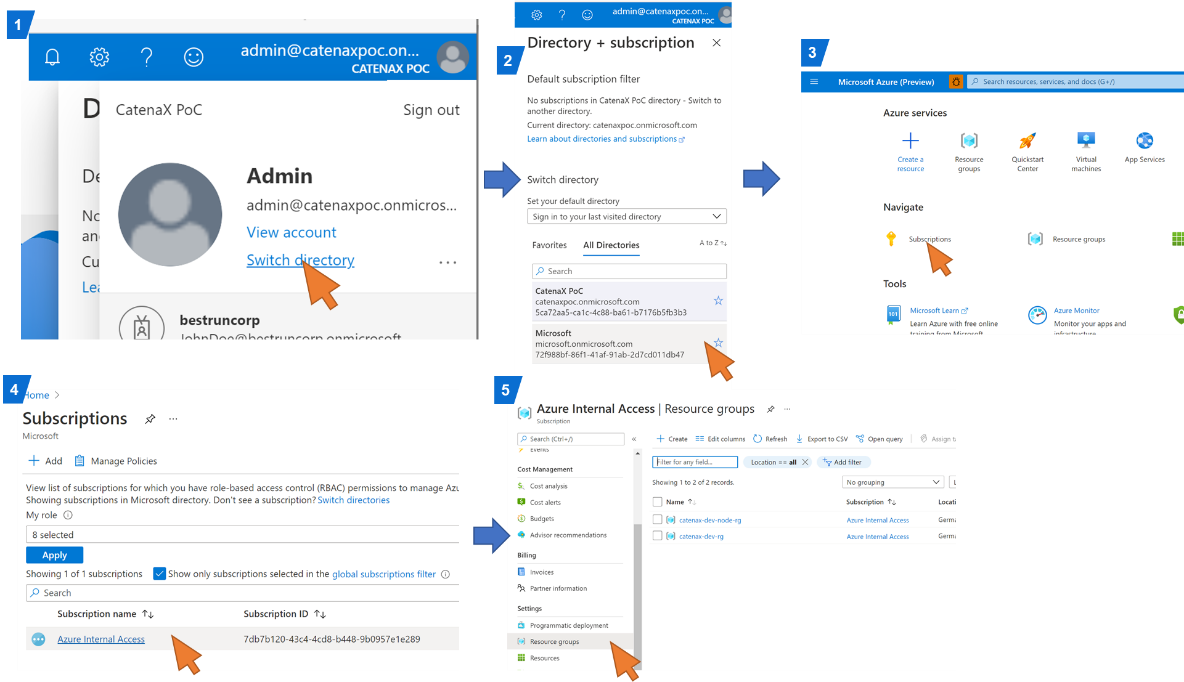
Please contact Martin Raepple or Lei Liu to request access to the Azure subscription. They will add you to the CatenaPoCTeam group which has Contributor rights in the Azure subscription resource groups.

Ein Bild, das Text, Screenshot, schwarz enthält.

Automatisch generierte Beschreibung

# Managing PoC resources in the Azure subscription

If you have been assigned access to the Azure subscription, you can use the Azure portal to access the resources deployed in the PoC landscape following this process:



1. Login with your Azure user account to the Azure Portal (<https://portal.azure.com>). If you are logged in to your company’s Azure AD tenant (e.g. <yourcompany>.onmicrosoft.com), click on the user icon in the upper right corner and select **Switch directory**
2. Select the Microsoft Azure AD tenant (**microsoft.onmicrosoft.com**) from the list
3. Select **Subscriptions** from the portal landing page
4. Select the subscription **Azure Internal Access** from the list
5. Select **Resource Groups** from the left side navigation menu. You will see the resource groups containing all PoC resources. You have the Contributor role assigned in these resource groups, which allows you to create, modify and delete Azure resources in these groups.

**Note:** When managing any resources in these resource groups, please **don’t use the Azure Portal**. We agreed for the PoC to follow the “Infrastructure-as-Code” (IaC) principle by writing scripts to automate the deployment and configurations management of the PoC infrastructure. We use the Open Source project [Terraform](https://www.terraform.io/) for writing these scripts. The current set of Terraform scripts can be found in the [/terraform](https://github.com/CXPoC/tractusx/tree/main/terraform) subdirectory of the PoC GitHub repository.

Any changes to the PoC infrastructure (for example adding a new Virtual Machine or database instance, renaming an existing namespace in the central Kubernetes cluster, deleting a subnet) should be implemented in the Terraform scripts first. This allows us to review the changes first before they are applied to the landscape. In addition, naming conventions can be consistently applied. Please follow the pattern applied to the existing resources in the script.