User guide

Robert Bosch GmbH - BD/PLS3

Table of Contents

***The Bosch Development Cloud (BDC) is a framework to build custom** development toolchains using standard solution building blocks like Github, Artifactory and Mattermost. It provides self service capabilities as well as AIM compliant user permission management.*

Created at: 2023-12-19 16:22:26 UTC
Source branch: main

User Guide

This document provides an overview for the Bosch Development Cloud, or short BDC. We start with an introduction what the BDC is and how you can get started with the services offered by the BDC. This is followed by an overview on general topics like access and permission management, the BDC Portal and how we can provide support to you. We then go through the growing catalog of services, starting with an introduction for the service followed by useful information on how to use it.

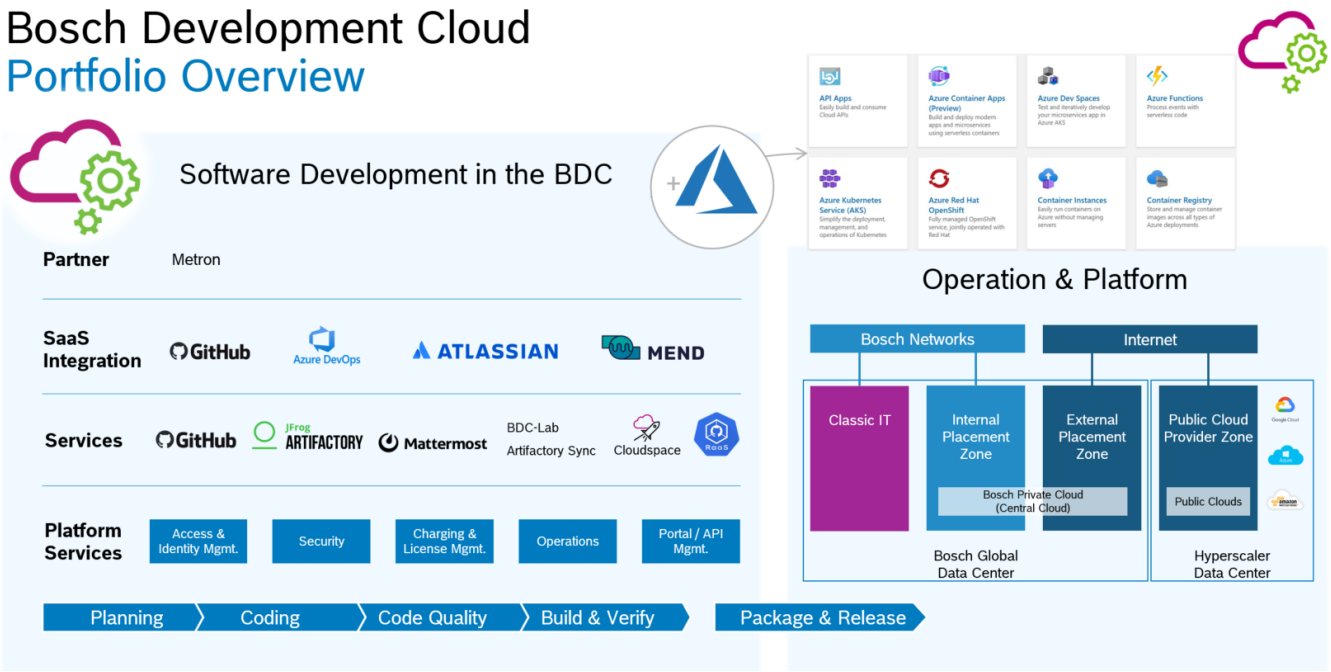
Chapter 1. Welcome to the Bosch Development Cloud

Purpose of the Solution

The BDC addresses the need of more and more software projects to be able to compose their toolchains using standard building block based on modern tools, tools being reachable from everywhere in the internet and an easy integration of external partners.

The BDC is not a monolithic solution but a collection of individual services which can be combined easily. Therefore, there is not "one" Cloud managed by the BDC, but different services available in the public internet which can be booked based on the individual requirements of the business use case.

These are our current BDC-services:



1.1. Service Catalog

In the following you can find a short overview of the different BDC Services that can be booked and managed in the BDC Portal. More details about each Service can be found in the service section itself.

Artifactory is a universal artefact repository manager to store binaries and to empower CI/CD workflows and is operated by the BDC in Microsoft Azure.

Atlassian Cloud products supported by the BDC are Jira Software (plan, track and release), Confluence (document collaboration) and Jira Service Management (ITSM), which are operated by the vendor Atlassian (SaaS).

Azure DevOps Services provide an integrated, collaborative environment that supports Git, continuous integration, and agile tools for planning and tracking work and are operated by

Microsoft (SaaS).

Cloudspace Service provides a secure and compliant space in the Microsoft Azure Cloud, where customers can explore and book Azure services and host development systems. Azure Cloud is operated by Microsoft.

GitHub Enterprise Server (github.boschdevcloud.com) is a git repository hosting service operated by the BDC in Microsoft Azure providing a collaboration platform for development in the internet and includes features like pages, wikis and issues.

GitHub Enterprise Cloud (github.com) is a git repository hosting service operated by the vendor GitHub itself (SaaS) providing a collaboration platform for development in the internet and includes features like pages, wikis and issues.

Runner as a Service (RaaS) include BDC managed runners on OpenShift and AKS for GitHub actions available for GHES.

Mattermost is a chat-ops service with persistent chat, search and integrations into other services (GitHub, ADO, etc.) operated by the BDC in Microsoft Azure.

Mend is an open-source management solution which helps SWD to identify used open-source licenses in their code base and find known security issues in used open-source components operated by the vendor Mend.io itself.

1.2. How to get started

What is a BDC business account?

Our services are normally not used by an individual person, instead they are used by teams for projects, product development, departments, collaboration, … You name it. We therefore decided to introduce the concept of a BDC business account. Whenever services of the BDC are ordered, they are ordered for a BDC Project and are not owned by the person who ordered it.

Technically, a BDC business account is a customer account in our portal which gives permission to order and manage the BDC services. Each BDC business account is identified by a unique ID and the short title.
The prerequisite to book and manage the different BDC services (like Artifactory, GitHub, Atlassian, etc.) is a BDC business account. Access to a BDC business account is managed via an IDM Access Right (BDC_cloud<bdc-id>_admin).

The BDC business account is for free. More details about the pricing of each individual service can be found in the section of each service. To get access to the service itself as a user, the individual IdM Access Right of the service is required.

Users and customers

You come here because you either want to use services of the BDC as a **user**, or because you want to order and manage services for your team as our **customer**. We make this differentiation because it will influence how you use the BDC. You might have both roles as well.

In the following, we will outline the steps you will have to take depending on your role.

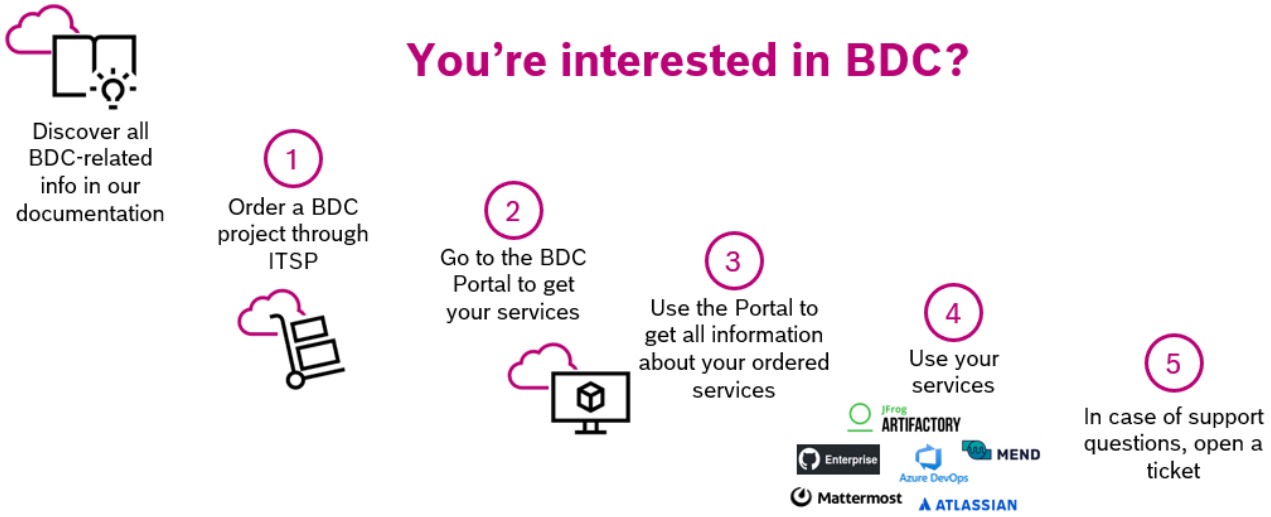
1.2.1. Get started as a user of the BDC

Someone has ordered BDC services (like Artifactory, GitHub, Atlassian, etc.) and you want to use them. Here are some basic steps to take.

- Access to all services in the BDC is managed via the IdM self service tool where you can apply for the service specific IdM Access Rights in the Bosch Development Cloud IT-Application by searching for it in the "find access rights" tab.
- We tried to add useful information to the IdM Access Right descriptions so try to search for your project name or other useful key words. Or the person who ordered the service simply provides you with the list of roles you want to request. You can also search for the access rights assigned to other persons in the Bosch Development Cloud IT-Application in the second tap "Access rights of others".

If the Bosch Development Cloud is not yet listed on the left side in IdM, search/select "Bosch Development Cloud" in the field IT Applications with Autom. Account Creation. Now that you have the target system assigned, click on it on the left side on the "Bosch Development Cloud" and go to the tab "Find Access Rights" to request the needed IdM Access Right(s)- There is also a training video available provided by IdM how to request access rights.

1.2.2. Get started as a customer of the BDC



You want to order and mange services of the BDC to help your team move forward. Let's get started!

Every order needs to be linked to a BDC business account. This ensures we know exactly which ordered services belong to which project/team/department, etc. To get access to this BDC business account as individual users, the IdM Access Right for that BDC business account is required (BDC_cloud<bdc-id>_admin).

The BDC business Account IdM Access Right (*BDC_cloud<bdc-id>_admin*) does not give access to the individual services (like GitHub, Atlassian, Artifactory, etc.). The access management for the services themselves are managed via different IdM

> Access Rights which are service specific and are documented in the section of each service

Order new BDC

- In case you or your team/project/department/... do *not* have a BDC business account already,
  - please order a BDC project via ITSP
  - apply for the IdM Access Right in the Bosch Development Cloud IT-Application in IdM self service tool. The name of the IdM Access Right is provided in the confirmation e-mail you receive after the order via ITSP

Tip: If the Bosch Development Cloud is not yet listed on the left side in IdM, search/select "Bosch Development Cloud" in the field IT Applications with Autom. Account Creation. Now that you have the target system assigned, click on it on the left side on the "Bosch Development Cloud" and go to the tab "Find Access Rights" to request the needed IdM Access Right(s)- There is also a training video available provided by IdM how to request access rights.

- After the Access Right Owner approved the Access Right Request it can take up to 24 hours until users can access the BDC Account in the BDC Portal.

> Some times you also might receive mails from our side like cost center is invalid. To verify the cost center is valid or not, please use the link **"http://rb-cae.de.bosch.com/SRSapigateway/api/CostCenter/<your_cost_center>"** and add your cost center at the end. In the result page, make sure the status of the **CostCenterEnabled** as **True** in the response.

Get Access to existing BDC

In case you or your team/project/department/ already have a business account, but you do not have the permission to access it, ask you colleagues for the IdM Access Right (name is BDC_cloud<bdc-id>_admin)and apply for it in the Bosch Development Cloud IT-Application in IdM self service tool .

Tip: If the Bosch Development Cloud is not yet listed on the left side in IdM, search/select "Bosch Development Cloud" in the field IT Applications with Autom. Account Creation. Now that you have the target system assigned, click on it on the left side on the "Bosch Development Cloud" and go to the tab "Find Access Rights" to request the needed IdM Access Right(s)- There is also a training video available provided by IdM how to request access rights.

- After the Access Right Owner approved the Access Right Request it can take up to 24 hours until users can access the BDC Account in the BDC Portal.

1.2.3. Terms and conditions & BDC Spec Sheet

Please find our current terms and conditions and BDC Spec Sheet here, including information about the valid security-classes.

1.3. Access to the Bosch Development Cloud

The following requirements are needed to access the BDC:

- Internet access
- A valid Bosch Account (internal or external IT user account)
- For Bosch internal employees an IT end device provided and managed by BD as required by EISA

1.3.1. Access Management

All permissions for the services in the Bosch Development Cloud are managed using role assignments in the IDM Tool. This ensures all compliance requirements are met. For each BDC Service we provide standard IdM Access Rights to get access to the services itself which is explained individually in the service specific chapters of this user guide in the permission management section.
Furthermore, the BDC offers the possibility to enhance this standard permission management by providing the bring your own group (BYOG) feature via the BDC Portal. This means that you can bring your own Azure AD group and assign it to one or multiple BDC Service(s) with a specific permission. To support this feature, the BDC offers APIs to manage and create your own **custom Role** in the BDC Target System in IdM and corresponding Azure AD group. The use of the IdM Access Right is only allowed for BDC Services and has by default no permissions. To assign permissions to the IdM Access Right for one or multiple BDC Services, please use the Bring Your Own Group API in the Service(s) you want to use the role.

Access requirements for Externals

Based on the central directive CD-07900, the following implementations apply for the authentication of the Bosch Developer cloud with regards to external partner and collaboration:

- CD-07900 applies for all IT systems and applications with Bosch data (independent of on premises and cloud hosting)
- The authentication of Bosch associates and external partners must be based on accounts from the Bosch Corporate Directory (BCD) provided by CI
- Strong authentication (Multifactor authentication) is required if the access origins from public Internet
- External partners can use their own corporate devices to access Bosch data with a limited and restricted access based on the authorization concept of the application

For further references, please refer to the CD-07900 regulation as follows: CD-07900

In addition, you can find more information about the user taxonomy on the following Docupedia page

Here is - from BDC perspective - the important statement on that page related to access for external partners to the BDC.

To sum this up, access for external partners is available by using external IT user accounts and assigning the required IdM Access Rights of the BDC Services.

More information about the Taxonomy of User Identities from an IT Security Perspective (B2E, B2C, B2B, B2B2x) including the different use cases can be found here.

External IT User Accounts

There are two types of external IT User accounts: the external standard user account and the external collaboration user (ECU). A comparison of both users can be found in the Docupedia page of the digital workplace. Both user accounts need the required IdM Access Right to get access to the BDC Service(s).

Please note that ECUs do not have access to IdM. Therefore, the IdM Access Rights requests for ECUs have to be done on their behalf by a user account with access to IdM.

For the BDC Services, the **ECU cannot be used for all services** due to missing Microsoft Licenses behind the ECU.

ECUs are blocked for accessing Azure Portal and Azure DevOps Services even with IdM Access Rights.

Therefore, please find below an overview to help you choose the right external IT User depending on the BDC Service you want to use:

BDC Service	External Collaboration User (ECU)	External Standard User
Atlassian Cloud	Yes	Yes
Artifactory	Yes	Yes
Azure DevOps Services	No	Yes
Cloudspace/Lab	No	Yes
GitHub Enterprise Server & Cloud	Yes	Yes
Mattermost	Yes	Yes
Mend	Yes	Yes

Password Change for Externals

External accounts also have a password expiration policy. Every 180 days the password expires and

needs to be changed, a self-set reminder for this is recommended. The password change can be done here.

Technical Users

There is often the need to use a technical user e.g. for CI/CD automation - so here are the steps to make a technical user ready to work in a BDC-environment.

- Request a technical user (use the ITSP form or ask your IT Partner or the CI-Hotline)
  - needs to have internet-access
  - needs an email-address
- Request permissions for the new account

Remark: Users without email-address are *not* synced to Azure by default! Please open a ticket at BD-Hotline.

Requirement

Please enter a valid information, who's responsible for this account in the public profile of your service user.
If our monitoring finds some unwanted behavior of such users, we could contact you first, before we shutdown this user's activities.

Login Method for Technical account Users:

To login as a technical user account in any application on Bosch development cloud (BDC portal, Artifactory, GitHub, etc..), just use a private window or incognito window in your browser. Then you can enter the credentials of this technical user account during the login.

1.3.2. Strong authentication via SAML

SAML (Security Assertion Mark-up Language) is a standard protocol, which provides i.a. single sign on (SSO) authentication.

In this authentication process, there is always a **service provider** (the application which the user wants to access, e.g. **GitHub**) and an **identity provider**, which holds the user information (a central directory like **AD**).

At **Bosch**, SAML is used as a way to connect web based systems with our **Active Directory** which holds information about Users like Username (commonly known as NT-ID), Email-Address or if a user account is active or retired.

To authenticate against a system like BDC-GitHub which uses SAML authentication, the users are redirected to the identity provider (at Bosch, Active Directory Federation Services), where they need to proof their identity.
On a Bosch managed device, this is mostly done fully automated.

After the login to the ADFS was successful, the user is redirected back to the Service (e.g. GitHub), where he is authenticated then.

1.3.3. Access via 2FA/MFA

The Bosch ADFS configuration requires two authentication factors for a successful login, which is known as Two/Multi-factor-authentication (2FA/MFA).

The first factor is your Bosch AD Account/Password ("Windows-Password").

For the second factor, there are multiple options. On a Bosch managed device, like a Notebook, the second factor is a certificate installed on the machine.

If the user doesn't use a Bosch managed device, the second factor might be a registered **cell phone** e.g. at the time of login, the user gets a call or has to open the **Microsoft Authenticator app** to provide the second proof of his identity against the ADFS.

In the following we describe how to get the 2FA/MFA for Bosch-internal employees as well as Bosch-external employees.

Prerequisites

- Existing Bosch account
- Existing 2FA/MA set
  - A certificate from Bosch-CA1-DE and E-mail encryption certificate
  - 2FA/MA via call to a mobile / cell phone
- Access to a mobile device with a number (to save time: DMTF-Support on the mobile)

Set-Up for Bosch-internal Employees

There is an exact documentation in the docupedia

Set-Up for Bosch-external Employees

There is now a way to get MFA for external employees via ITSP. It does not require email encryption certificate and PIN letter. Please follow the instructions here

Access to the non-Bosch-Device

You open your application and to log in and now you'll get a call/sms/push-notification to authenticate yourself using your mobile/cell phone. Afterwards you can work just like on a regular Bosch-Device.

Bosch-internal Employees have to use an IT end device provided and managed by BD to access Bosch data and therewith BDC Service from the Internet. More information can be found in EISA

1.4. Feedback

We look forward to receiving your feedback, which will help us to improve in the future!

Chapter 2. Help & Support

2.1. Support Channels, Incident Reporting & BDC Operations

This section gives an overview of the different support channels with the BDC and other users as well as how to report incidents and how the BDC operates its Services.

2.1.1. Support Channels

Support via Jira Service Desk

To enable customers and users to interact with us, in case of support requests, questions or problems that are not covered by our user guide in the section of each service, feel free to reach out to our BDC Service Desk.

If you do not have access please access the Site, enter your nt-user@bosch.com e-mail address and then you will be redirected to the Bosch Single Sign On. With this, an Atlassian account will be created. Afterwards, you can login to the BDC Service Desk and create a ticket.

BDC Bosch Connect Page

The BDC Bosch Connect Page is used for the communication to customers and users where the BDC regularly posts news & updates or announces service related events.

Bosch Tube

The BDC also has its own BDC Bosch Tube Channel which contains helpful support videos or the recordings of events.

User Group Meeting

The User Group Meeting is held monthly with the intend to give customers and users the opportunity to connect and interact with other users, share experiences and questions and provide some updates from BDC side.

Agenda for the User Group Meetings:

- BDC Updates
- Customer Updates (e.g. best practices or lessons learned from migrations)
- Q&A and Discussion of general topics

For individual questions, problems or support please use our BDC Service Desk.

On the following days the User Group Meeting will take place in 2024:

- In the odd-numbered months (January, March, May, July, September, November) every second Tuesday of the month at 10 CE(S)T.

- In the even-numbered months (February, April, June, August, October, December) one slot every second Wednesday of the month at 16 CE(S)T.

Download the **ics files** to import the series to your calendar:

- BDCUserGroupMeeting_TuesdaySlot
- BDCUserGroupMeeting_WednesdaySlot

MS Teams Channel

With the intend to provide a possibility for customers and users to interact and exchange with others, the BDC created a BDC MS Teams Channel in the DevCorner. If you have any individual questions or problems for which you would like to get a feedback from the Bosch Development Cloud Team, please open a ticket at the BDC Service Desk and do not use the MS Teams Channel.

Bosch Overflow

The BDC Bosch Overflow category can be used to ask BDC related questions to a community of developers. It's a modern platform, which offers you features like tagging items or rating answers. As this is a developer-forum everyone is encouraged to answer when (s)he knows a solution. The BDC team is scanning the questions regularly and takes care that answers are given.

Developer Idea Log (Devil)

The Developer Idea Log - short Devil - former VoteBox - is a pilot project by BD/PJ-NLI in which the Bosch Development Clouds participates to get input from users for improvements or new product ideas / features / services to extend the BDC Product Portfolio.
More information about the Developer Idea Log, its intentions & scope can be found here.

Code sharing

We use different options to share code with the community

Code snippets and scripts

We use a repository in the Bosch Org in Github to share some smaller code snippets with the community. You can access it here

Collaboration and joint development

We setup a Github org to allow us in specific cases and for projects to share and jointly develop code. The Org can be found here. Please get in touch with us in case you have a specific need or project.

2.1.2. Incident Reporting

For any incidents and problems related to the BDC services the following channels can be used:

- IT Service Desk / CI Hotline
  - ITServiceDesk@bosch.com
  - Call to 3311

This channel is intended only for the following incidents:

- Disruption of any BDC Service
- Any failure in the operation of a BDC service

A SMT ticket will be created from a call or from eMails to the ITServiceDesk.

- ICheck
  - ICheck can be used as an easy to use interface to the SMT system to create tickets from a call or from eMails to the ITServiceDesk.

2.1.3. Operations

Service Operation

The objective of Service Operation is to make sure that IT services are delivered effectively and efficiently. The Service Operation lifecycle stage includes the fulfilling of user requests, resolving service failures, fixing problems as described in the previous chapter as well as carrying out routine operational tasks.

The entire organizational environment for the supply and operation of services, as well as the planning and implementation of measures in exceptional circumstances (e.g. interruptions, emergencies), is described with the Operation to Satisfaction Processes at CI and documented in the CI process landscape, according to the ITIL process method.

More information is documented in the operational manual that describes the operational concept in the scope of CI.

All roles and responsibilities of the processes are described here: Preamble (BDC Operational Manual).

Release and Deployment Management

The BDC development team tests a new BDC release for one week, before it is deployed. All changes are automatically rolled out to the productive environment using a DevOps pipeline and the developers do not have permissions to make any manual changes in the productive subscription.

Development is done in four week Sprints and a new BDC release is being deployed at the end of a Sprint, with minimized outage times of the BDC services.

The planned downtime window is Thursday at 5:00 am UTC every 4 weeks and is communicated via Bosch Connect and SPOI. The windows can be moved in case of public holidays.

Maintenance & SPOI

Maintenance of BDC-Service(s) will be announced:

- "Events" in our BoschConnect Community (please become a member and you will be informed automatically)
- in the SPoI-System. See the following guide how to subscribe:

1. Navigate to SPOI page
2. choose the tab "Self-Service"
3. choose "Search IT Services"
4. fill in "BoschDevCloud" an run the search
5. open the explorer-like tree by clicking on "+"
6. click icon to subscribe
7. save

To unsubscribe from the SPoI:

1. Please navigate to SPOI page and follow the steps shown in the chapter above
2. click icon "unsubscribe"
3. save

Microsoft Azure Public Cloud general issues

You can see the status of the Microsoft Azure cloud on this website.

Incident Management

BDC follows the official CI Incident Management process O1 .

An incident is for example a disruption of one of the BDC services or a failure in the operation of a service. The incident management process is applicable only for these kinds of issues.

For an incident, an incident ticket has to be created in SMT. These tickets will be routed to the BDC team. The BDC team will add incidents that are relevant for the community to Bosch Connect.

Based on the severity of the incident , the "BDC Status Page" in the Bosch Connect overview will be updated.

BDC follows the official CI Problem Management Process.

2.1.4. Change Management

For any changes of our BDC services, we follow the official CI-Change-Management process. That means a SMT-Change Request (CR) is provided. See also here for the change management process in BD/PLS.

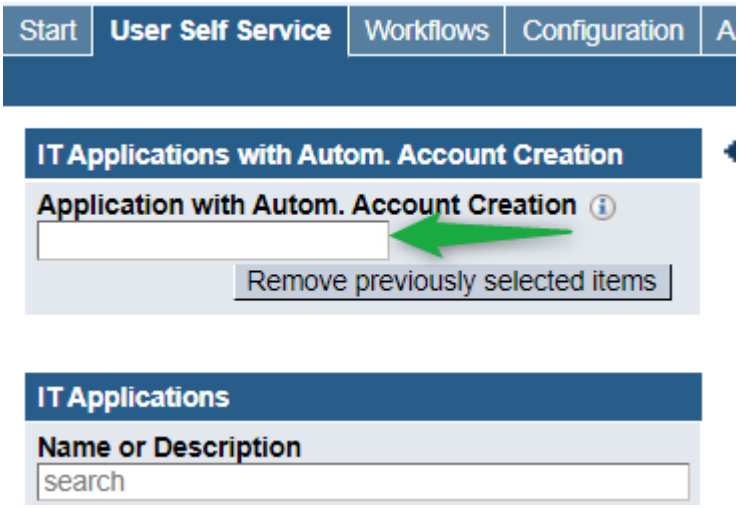
To get updated about our changes, you might use SPoI and subscribe to **SDE_BOSCHDEVCLOUD** (under IT-Service - Application - Engineering - SDE).

2.2. IdM Tool

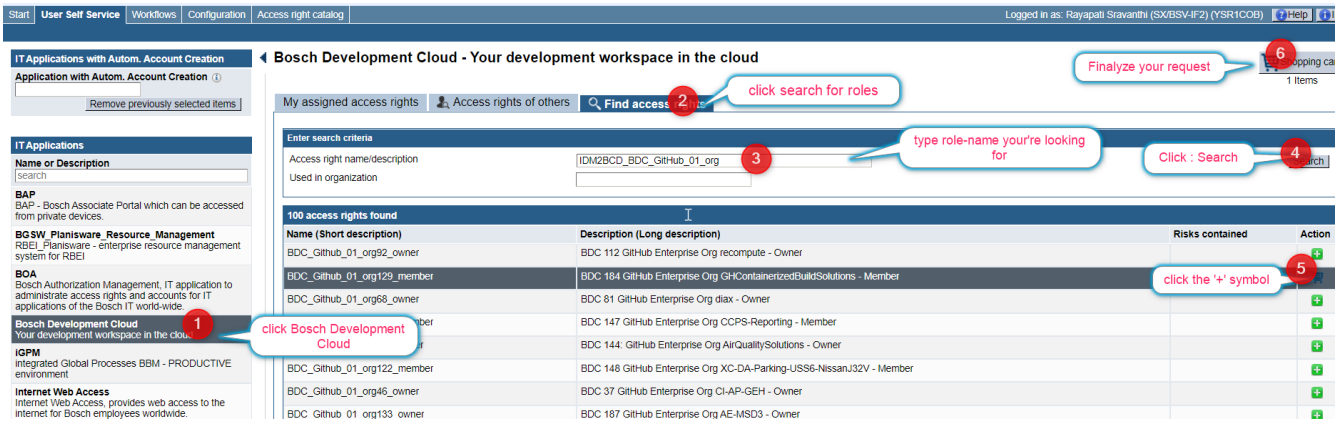
2.2.1. IdM Self-Service

Access to the BDC Portal and all BDC Services can be requested via the IDM Tool.

Here is a short guide how to navigate in IdM: If the Bosch Development Cloud is not yet listed on the left side in IdM, search/select "Bosch Development Cloud" in the field IT Applications with Autom. Account Creation.



After searching the target system, you will see it under IT Applications, and you are be able to assign access rights.



1. click on "Bosch Development Cloud"
2. click on tab "Find access rights"
3. search/ filter for IdM Access Right: We tried to add as much useful information as possible to the IdM Access Right descriptions so try to search for your project name or other useful key words. Or the person who ordered the service simply provides you with the list of roles you want to request. You can also search for the access rights assigned to other persons in the Bosch Development Cloud IT-Application in the second tap "Access rights of others".
4. click "Search"
5. click the "+"-symbol at the end of the line to add the IdM-role to your shopping cart
6. click on the shopping cart and add a "Reasoning" which is visible for the role approver and submit your "shopping-cart"

External colleagues cannot use the IdM Self-Service.

There is also a training video available provided by IdM how to request access rights.

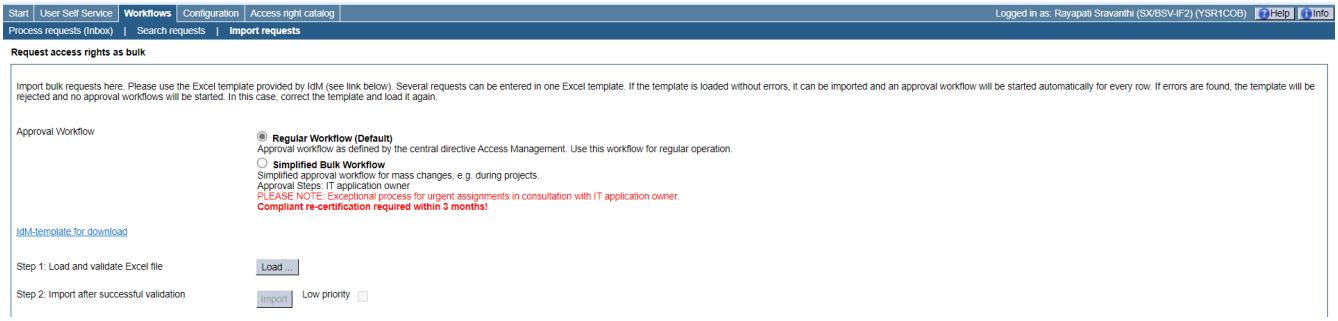
2.2.2. Non-Self-Service

If you want to do this for several accounts ("bulk operation") or an account for externals, you have the following options:

- User ITSP-Request "Manage IT roles/authorizations via IdM"
- Mail to BD-Hotline

For all ways you may prepare a template and attach it to the mail, the ITSP-Request or the incident ticket. You can find it here:

- IDM Tool
- Click the Workflows tab. (left upper corner in tab "Start")
- Display the Import requests view in the foreground.



- Click the IdM-template for download link and save the Excel file to your computer.

Fill in:

- Open the Excel file and, in the Action column, select the Assign access right option.
- Line 5: "Reason" to explain the approver the reason of the request
- Column A: Select the mode via drop-down
- Column B: IdM target system - for BDC: "Bosch Development Cloud"
- Column C: your account
- Column D: one of the roles you received when the service requested was delivered.

Mail to CI-Hotline

Use the filled-out template above and send off to CIHotline@bosch.com , preferred subject is :Service: IDM - USERADMINISTRATION PORTAL_EMEA - Please assign new IdM-Role to User

2.2.3. IdM for master of roles and approver

The master of role (MoR) has to approve every IdM Role assignment. There are multiple ways how a MoR can delegate approvals, either for all requests via a delegate or only for specific roles. When you click on the help button in IdM, the online help opens up. In the section "Configure organizations view", you will find instructions on all those options.

2.2.4. IdM Glossary

The IdM Glossary explains the key terms of the IdM (Identity Management) IT solution.

2.2.5. IdM Training

The IdM Training Area provides different trainings depending on your organizational role (user, user admin, manager, IT-Application Owner) in IdM.

2.3. BDC Portal and API Management

The central management of the BDC Services can be done by customers via the BDC Portal.

2.3.1. General information

The Portal is implemented via the Azure API management service from Microsoft Azure. It brings a lot of preconfigured features and functionality with it. Login to the Portal is possible for everyone who has a valid Bosch Account. Simply sign in with your Bosch account, it sometimes requires a second login (known bug already reported to Microsoft). We recommend using the Chrome browser.

As described in the about section, a **BDC business account** and corresponding IdM Role (*BDC_cloud<bdc-id>_admin*) is required to book and manage the BDC Services.

The BDC business account IdM Access Right (*BDC_cloud<bdc-id>_admin*) does not

> give access to the individual services (like GitHub, Atlassian, Artifactory, etc.). The access management for the services themselves are managed via different IdM Access Rights which are service specific and are documented in the section of each service.

2.3.2. APIs

The APIs are our way to allow you to automate the management of the BDC services you have ordered. We developed a standard set of APIs for each BDC service which we offer. They can be used to order and configure the services available for your BDC business account. We will add more and more APIs over time so check regularly for new functionality.

This comes in addition to the APIs of the individual services. Those are not limited or customized and can be used as designed by the vendor. For Cloudspace, you can use all Azure APIs, for GitHub Enterprise Server, all APIs are available and so on.

2.3.3. Manage my BDCs

Visual Service Explorer

The visual service explorer will guide you through the different activities. It allows you to drill down into your services and will invoke the respective APIs in the background. This functionality is not available for all services at the moment and we will add them step by step.

Currently available is:

- Cloudspace

> You need to be owner of a BDC via the respective IdM Access Right "IDM2BCD_BDC_cloud<xxx>_admin".

User Guide

This link will open the relevant part of our general user guide to bring you directly to the interesting content. The full user guide is always available via the top navigation.

Open API

This view replaces the native API reference page provided by the API management in a more common way by using the OpenAPI standard format. You see a list of different API end points with their allowed methods. Click on "Try it out", select parameter where applicable and then hit the execute button.

2.3.4. My BDCs (legacy)

This user interface is the out of the box UI from the Azure API Management Service. We will replace it step by step with the new "manage my BDCs" interface.

This page shows you all business accounts for which you have the corresponding IdM Access Right.

You can select any of them to request new services or to check what is already setup for you. Documentation of these products will be available in the API reference page.

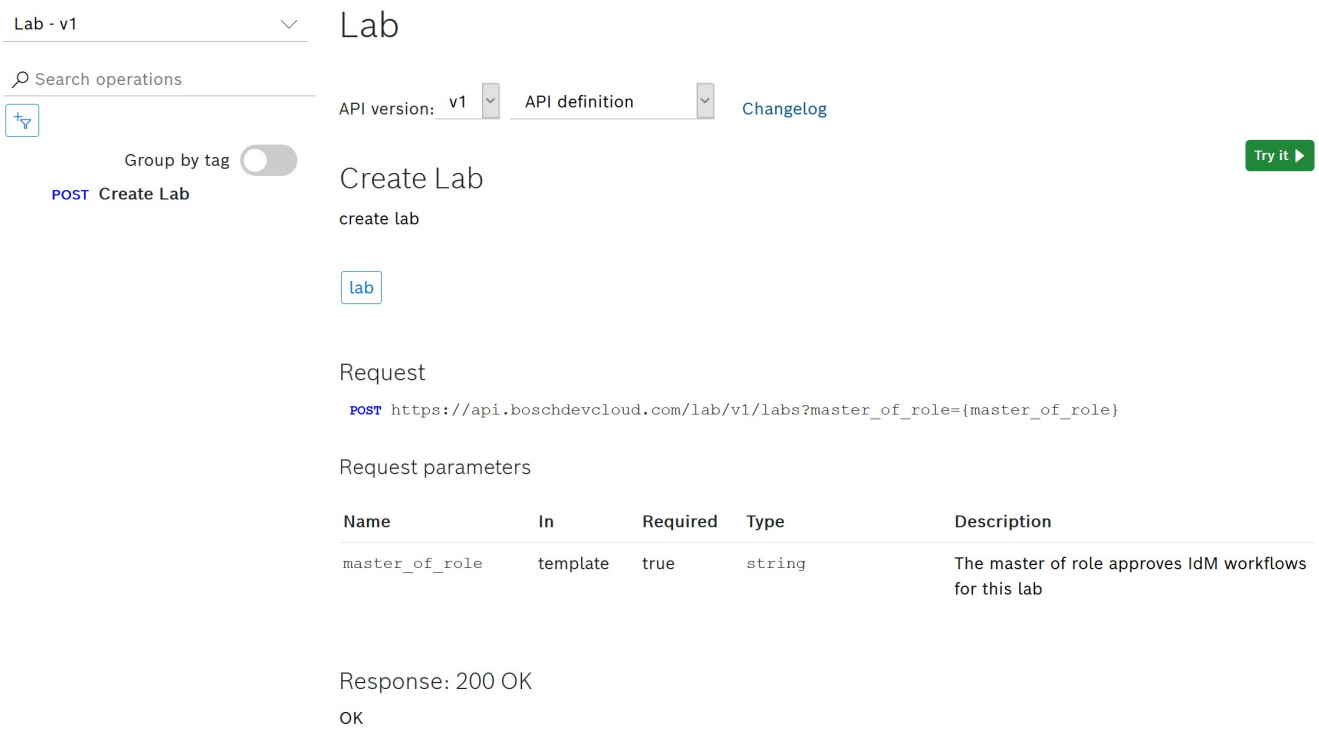
You need to be owner of a BDC via the respective IdM Access Right "IDM2BCD_BDC_cloud<xxx>_admin".

API reference page

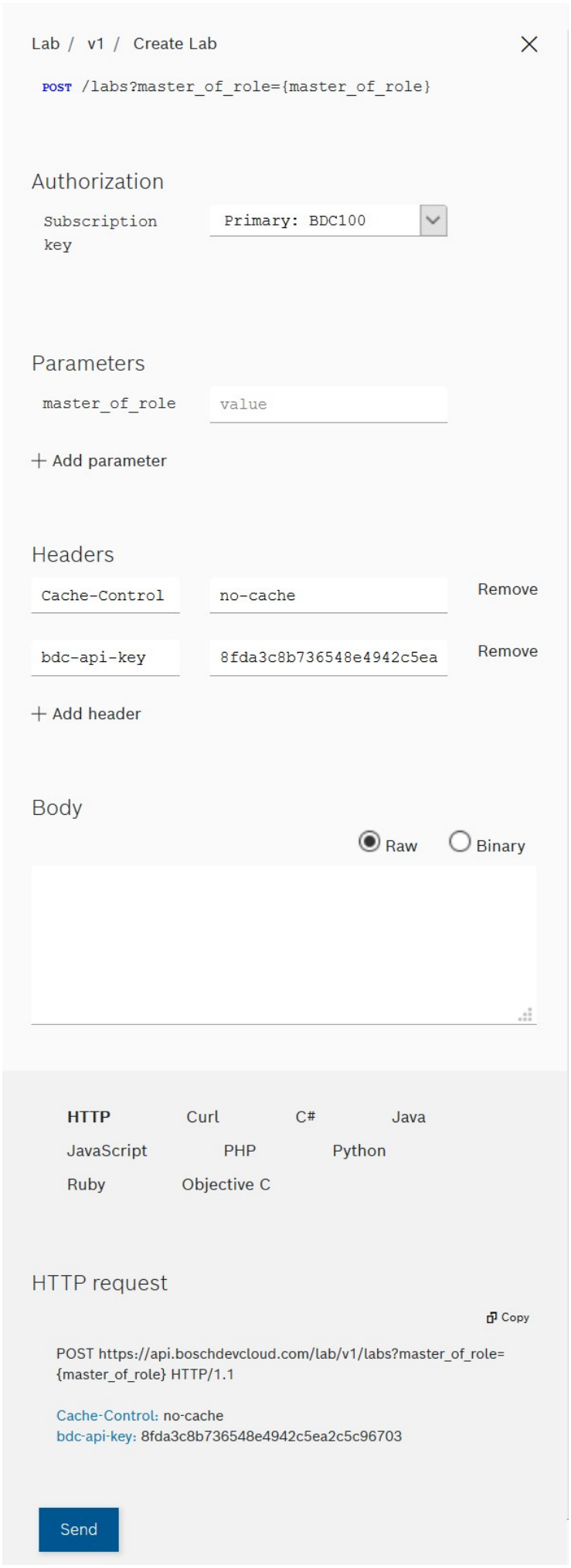
The left area of the screen allows you to search for API operations to check the range of functionalities offered by the API.

The middle section presents the documentation of the selected API operation, with a description containing the request type, endpoint, and all parameters.

You have to click on the "Try it" button to access the API execution area.



API execution area:


Lab / v1 / Create Lab
POST /labs?master_of_role={master_of_role}
Authorization
Subscription key
Primary: BDC100
Parameters
master_of_role
value
+ Add parameter
Headers
Cache-Control
no-cache
Remove
bdc-api-key
8fda3c8b736548e4942c5ea
Remove
+ Add header
Body
Raw
Binary
HTTP
Curl
C#
Java
JavaScript
PHP
Python
Ruby
Objective C
HTTP request
Copy
POST https://api.boschdevcloud.com/lab/v1/labs?master_of_role={master_of_role} HTTP/1.1
Cache-Control: no-cache
bdc-api-key: 8fda3c8b736548e4942c5ea2c5c96703
Send

This area allows you to "try" or better, execute API calls. Please understand that this is not a test area but the real execution of requests to the production system. When you try out how to order a service, you actually do order the service.

- Authorization requires you to select the subscription you have for this product. In case you have none, please go back to the product page and generate one there. Since a subscription key for a BDC is always bound to a user, we recommend using a technical user (with Bosch account), if you plan to integrate automatic requests to the BoschDevCloud API in your CI/CD environments. Just assign the IdM role BDC_cloud<id>_admin to the technical user account and log in to the BDC Portal with it to create a subscription key for your BDC<id>.
- Parameters allow you to specify required parameter for the API call execution. In the above example, I would need to add the approver for requests for access to the lab.
- A list of headers are send with the API call. You will find here the secret key from your subscription. The key in the screen shot is already replaced by a new secret ;-) The Cache-Control header is set by default but is not required.
- Body shows the payload send with the call. This example has no payload.
- Example calls allow you to see how your API call would look like using different languages. The screen shot shows the HTTP example but you can click on the other languages, e.g. Curl to see how you can use the API.
- Make sure you select the correct BDC business account (if you have more than one) when you use any of the APIs. The portal will always pick the one for the BDC with the lowest number so ensure you choose the correct BDC.

Remember: The API Key is your personal identifier, do not share it with other people. Handle it like you handle your passwords.

2.3.5. Profile

Account Details

The account details for your user cannot be changed. The Registration date helps you to ensure you celebrate every anniversary of Portal usage!

BDC Subscriptions

To be able to use the APIs, you need a so called subscription (i.e. personal token) for each BDC business account. By default, the BDC will create one for you, you can furthermore create additional ones. You can find a full list of all your subscription keys configured in the Portal. You need those keys for every API call or request via the Portal. At any point in time you can go to your profile and delete, rename, or recreate your subscriptions.

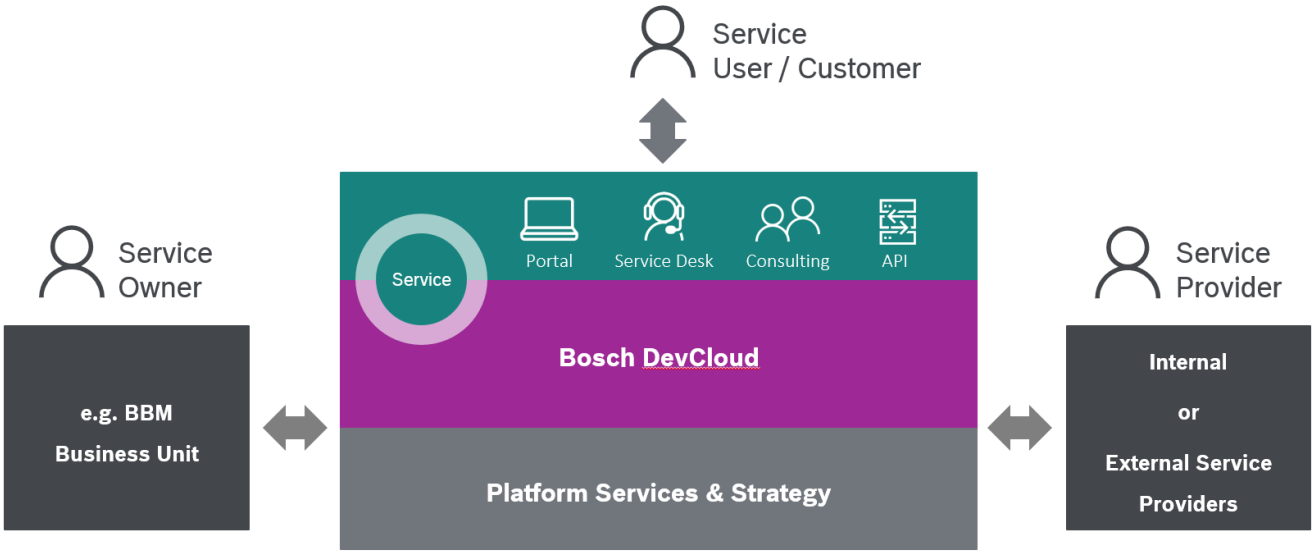
The API Key is your personal identifier, do not share it with other people. Handle it like you handle your passwords.

Chapter 3. General Services

3.1. BDC as Service distribution platform

The objective of the BDC is to offer an easy access to services required by software developers. Therefore, we are building a system in which we not only offer services developed by the BDC team, SaaS services operated by external partners but also services provided by partners inside Bosch (e.g. CloudIA by XC-ECO).

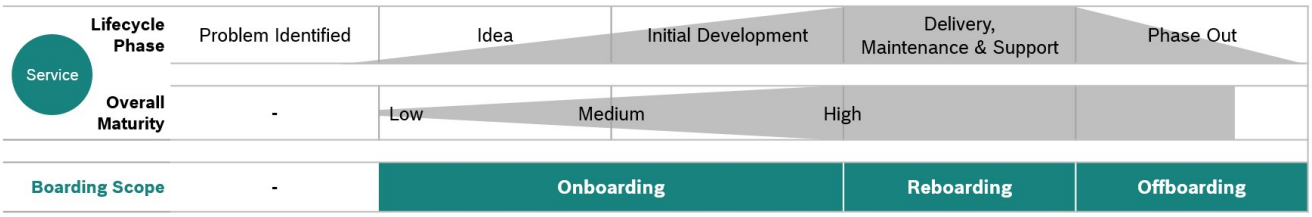
We will enhance this chapter while we learn how to implement such a platform. You will find those services integrated in our BDC Portal.



3.1.1. Service onboarding overview

There is a process of evaluation which manages how services are made available in the BoschDevCloud environment, e.g.:

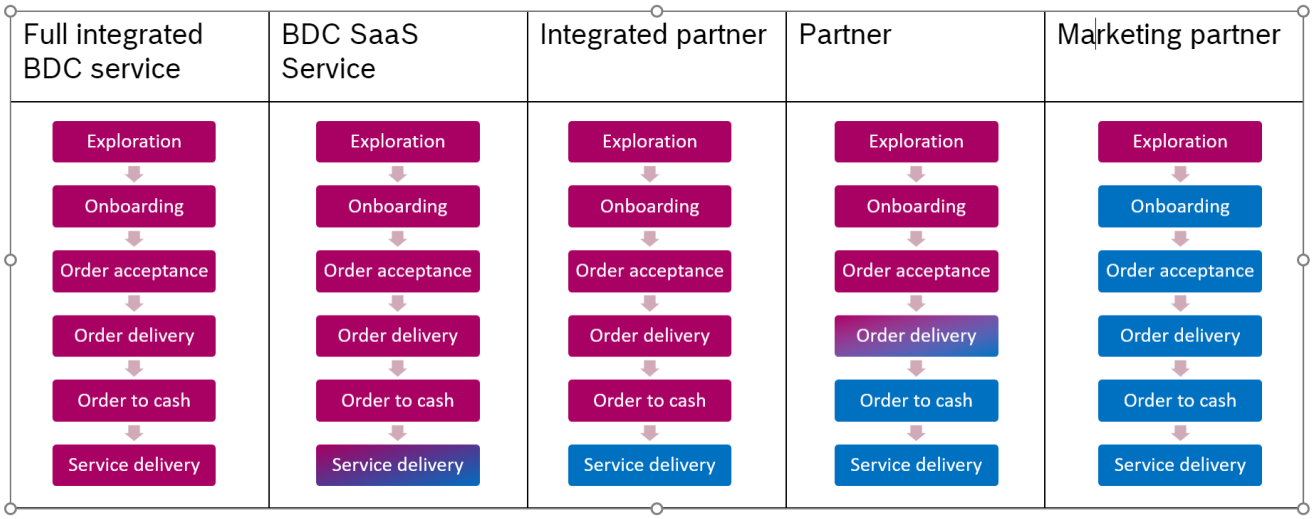
- Considering service lifecycle phase and overall maturity
- Addressing contributions, responsibilities & potential risks



3.1.2. Service integration model

Services will be integrated into the BDC following one of the below models. This integration model allows different level of accountability for the Service Provider and the BDC. The integration is shown from a customer point of view.

- **Exploration** : When looking through the available services, the partner services is listed along the BDC services
- **Onboarding** : The business account (BDC) and the Portal is available for the partner service
- **Order acceptance** : The partner service can be ordered via the BDC portal
- **Order delivery** : The partner service is deployed via the API management of the BDC
- **Order to cash** : Charging for the partner service is done via the BDC
- **Service delivery** : The responsibility for the delivery of the service



3.1.3. Terminology

Term	Definition
Partner	High-level description of an organization or person that cooperates for business reasons with BDC, e.g. for providing services.
Service Owner	Organization or person responsible for the service, e.g. for definition or creation of the service.
Service Provider	Organization or person that is either responsible or directly provides the operational and value creating part of the service, e.g. an organization providing expert consulting services.

3.2. Bosch Private Cloud (BPC) as hosting platform

3.2.1. What is the BPC?

The Bosch Private Cloud is an on-premise cloud installation. It allows you to host solutions, web-services, container workload, VM workload and more in our Bosch data centers. As a customer you can choose to host your application in one of the following runtimes: OpenShift, Cloud Foundry or OpenStack. Moreover, the Bosch Private Cloud offers Data Services, Messaging Services, and more in a marketplace which allows you to fully automate the services' lifecycles. A modern cloud

architecture allows you to easily onboard and deploy applications which would have been deployed historically in SL2 or SL4. A tight integration with the Bosch Development Cloud allows you to easily build Source Code and deploy Artifacts that you manage with the Bosch Development Cloud.

3.2.2. Marketplace for Runtimes and Services

Inside the Bosch Private Cloud Marketplace we offer an API and web portal which allows the ordering of services offered on the Bosch Private Cloud. Here, the Bosch Private Cloud offers services like container runtimes, databases, messaging and caching services, which users can book to develop and operate their services and solutions.

See the Bosch Private Cloud Documentation or simply login on to one of our marketplaces (e.g., the IPZ marketplace) for an overview about the services offered in the BPC marketplaces.

3.2.3. Architecture

The Bosch Private Cloud is separated into multiple segments which we call Cloud Instances. In every location where the Bosch Private Cloud is available there is at least one Cloud Instance. A data center is considered as an availability site of the cloud. A data center usually contains more than one Cloud Instance. For the purpose of scalability, security and operational reasons we segmented the Bosch Private Cloud into two types of Cloud Instances: Internal and External Placement Zones. Both types are technologically the same but there is a small difference between the services which can be used inside these Cloud Instances.

The Internal Placement Zone is intended for internal business and Bosch internally used applications. Services and applications running on Internal Placement Zones are not directly reachable from internet. Services and applications placed in Internal Placement Zone are not intended to be exposed to Internet though it is possible to publish parts or specific APIs of them by using e.g. API Gateways, API Management, Web Access Management, or other services offered in Enterprise IT or External Placement Zones.

The External Placement Zone is intended to host external business for Bosch customer or Bosch partners. It is intended for customer-facing solutions like, web pages, web shops and IT landscapes for Bosch products.

3.2.4. BDC Integration

Currently we offer documentation how to run your own GitHub Runner on the BPC, see here.

Stay tuned for a fully automated integration in the future!

3.2.5. Links

- Bosch Private Cloud Landing Page
- Onboard within Minutes to BPC
- IPZ marketplace
- EPZ marketplace

- Docs

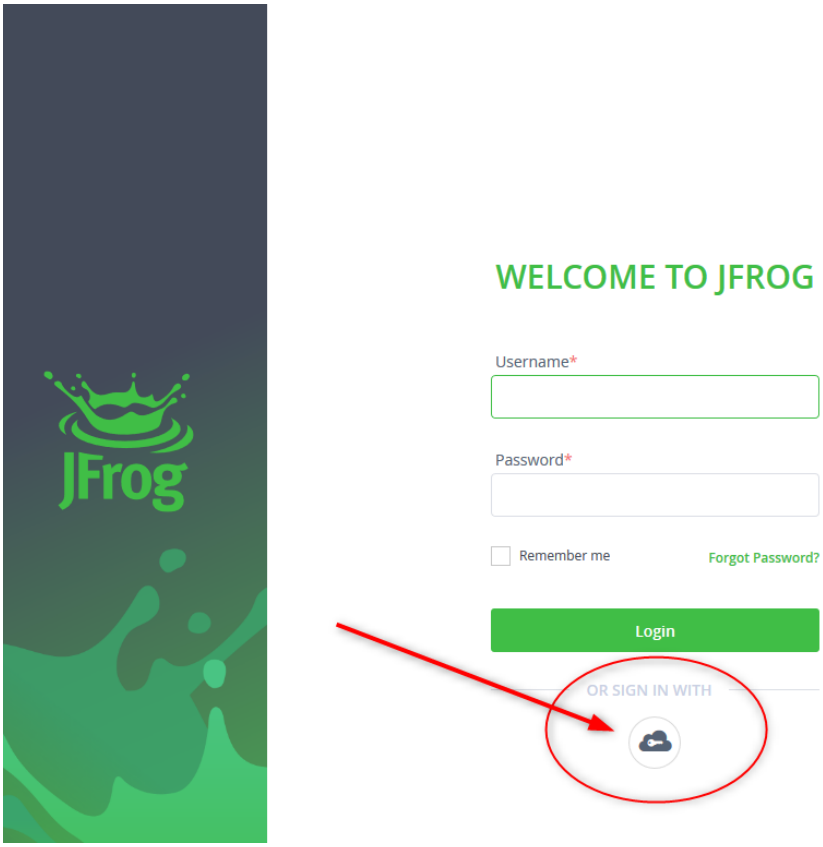
Chapter 4. Artifactory

BoschDevCloud Artifactory is an universal artefact repository manager to store binaries and to empower CI/CD workflows. Many repository types are available e.g. npm, conan, docker, ...

You can request all repository types supported by the product, including docker registry.

Access to the service is available at artifactory.boschdevcloud.com

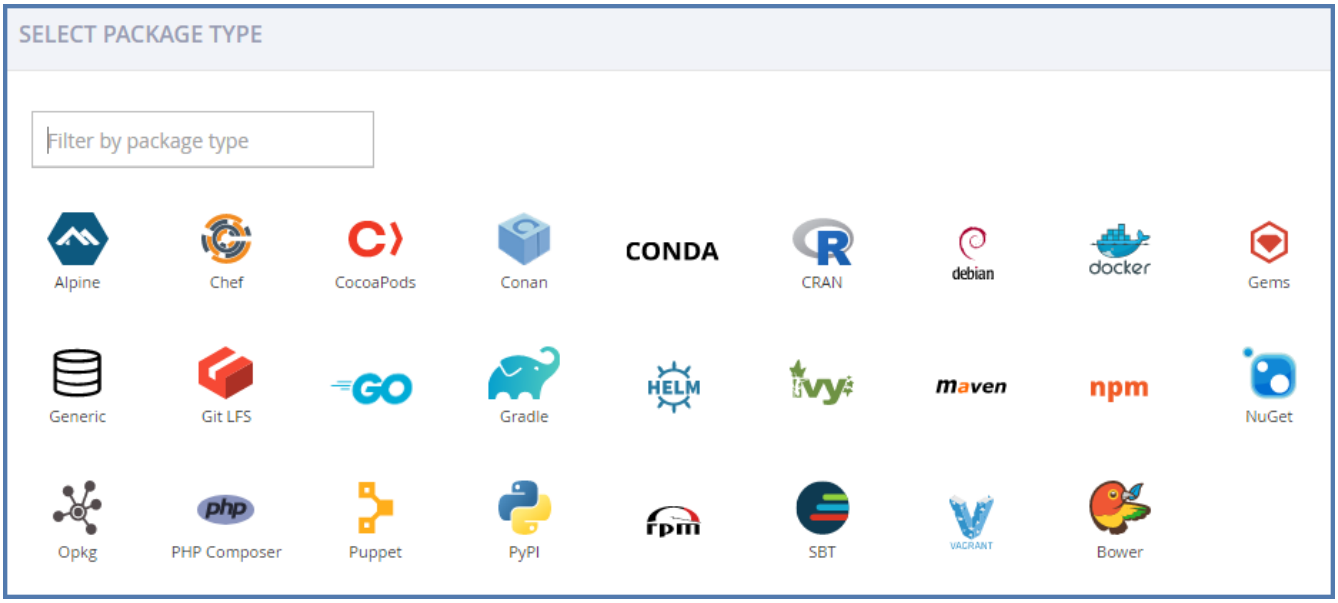
Artifactory uses SAML-authentication:



4.1. What do I get for Artifactory and how to order

You can request all repository types supported by the product, including docker registry.

You as BDC-owner/admin use our BDC-Portal to request both repositories and permission targets. For each permission target three IdM-roles will be created.



4.1.1. Types of Repositories

There are three repository types:

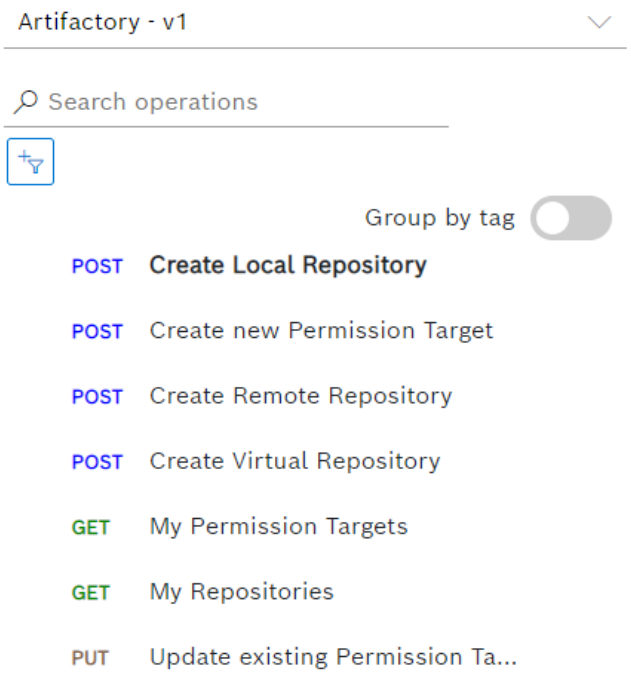
- Local – a physical, locally-managed repository into which you can deploy artifacts.
- Remote – a caching proxy for a repository managed at a remote URL. Artifacts are stored and updated in remote repositories according to various configuration parameters that control the caching and proxying behaviour. You can remove artifacts from a remote repository cache but you cannot actually deploy a new artifact into a remote repository.
- Virtual – an aggregated repository (that combines the local and remote repositories) under a common URL, used to create controlled domains for search and resolution of artifacts.

4.1.2. Request Repositories and Permission Targets

Only a BDC-owner can use our BDC-Portal to request repositories and permission targets.

Navigate in the portal to "My BDCs". When you do not see "BDC <xxx>", you have not been onboarded to the respective BDC. You can do so yourself via self-service. Check for role "IDM2BCD_BDC_cloud<xxx>_admin". We are currently onboarding our existing users to the new BDC-management, so you might not yet be onboarded to your respective role. Please contact us if this is the case! When the entry is visible select it.

Before you start the request check if you have already created a "subscription" (key) for this API-product. To do so you just enter a name and press "subscribe". Unfortunately the tool now jumps to your "Profile", so you have to navigate back to "My BDCs" and enter your BDC. Then please choose "Artifactory" where you have the following options:



We assume that you are already familiar with the concept of jfrog-artifactory. See "Repository Management" and "Identity and Access" in the Administration Guide of jfrog.

The main point is that Artifactory uses "permission-targets" which consists on the one hand of the repositories which are combined and on the other hand of user-groups which have access. Example: One repository belongs to at least one permission-target and one group of users. The user-groups are realized via IdM-roles. Two or more repositories can be combined together in one permission target. Also one repository can be part of more than one permission target.

When you start from scratch you create your first repository by e.g. "Create local repository". Please read carefully the description which we provide for each item in the portal. Unfortunately when clicking the "Try it"-button (actually the "Do-it"!) some parts of the description are hidden by the new window on the right border. We can't change this behavior. When you filled out the data requested you send off the request either by pressing "send" or by saving the command in the language or your choice and send it off later.

Via the item "My Repositories" you can check which repositories exist for your BDC.

To make the repos accessible you now create a permission target. When doing so three IdM-roles will be created which can be assigned via IdM-self-service:

IDM2BCD_BDC_Artifactory_<yy>_perm<xx>_nodelete	*permissions for repositories*: Read, Annotate, Deploy/Cache *permissions for builds*: Read, Annotate, Deploy
IDM2BCD_BDC_Artifactory_<yy>_perm<xx>_reader	*permissions for repositories*: Read *permissions for builds*: Read
IDM2BCD_BDC_Artifactory_<yy>_perm<xx>_user	*permission for repositories*: Read, Annotate, Deploy/Cache, Delete/Overwrite *permissions for builds*: Read, Annotate, Deploy

To find the IdM-roles in the IdM-Self-service you can use the name of the permission target. Usually

it takes a while until the roles are visible in the IdM-Self-service and even a bit longer to see them in the Artifactory@BDC.

Repositories can be linked to more than one permission target!

Updating an existing Permission Target

When you have created additional repositories you may update an existing permission target by choosing the respective entry in our portal. Please remember this needs to enter all repositories existing repos and the new one(s), as this is not an incremental process! The main difference is that during the update no new IdM-roles are generated.

4.2. Cost for Artifactory

The licensing model from JFrog for Artifactory is server based. The charging model takes the infrastructure and license cost and distributes it according to the repository size.

The current cost for Artifactory@BDC is based on the monthly average size of each repository.

	Costs per month	**Remark**	**Cost charged against:**
Artifactory@BDC Operations + Infrastructure	0,90€/1 GB/month (incl. License)	unlimited amount of users	cost center of the repo
Artifactory dedicated instance (on request)	1500€ plus Operations + Infrastructure	unlimited amount of users	cost center of the instance
Artifactory user	no additional costs	unlimited amount of users	

4.3. Permission management in Artifactory

Each Artifactory Repository has the following IdM roles available

- User: Read/write/delete access to the repository
- User Nodelete: Read/write access to the repository
- Readonly: Read access to the repository

Artifactory uses so called permission targets to consolidate user management across multiple repositories.

It takes a short time after logging in the first time to see the expected repos. There is a job running in the background to synchronize new users to the respective permission-groups inside Artifactory.

4.3.1. Technical User for Artifactory.

- If you would like to have technical accounts or BOT accounts for CI/CD automation, kindly refer to Technical User in Azure section.

4.4. Repository-names for Artifactory@BDC

For all repository types (including docker repo), we only offer the repository path method.

A repository-name should include hyphens only, no underscores.

Use of special characters

Artifactory is not able to deal with special characters like # in object names. As a workaround, you can put the object in a folder named with no special characters and download the whole folder. Files with special characters in the name cannot be restored in case they get deleted. This is a known issue with Artifactory.

4.5. Artifactory and BIOS

Artifactory can be used to share content related to a BIOS project. There are several requirements to consider.

- The repo owner will be charged for the cost of the stored data in the same way like we charge for private repos
- We have no way to highlight a repo is shared under the BIOS license.
- Ensure the reference to the source code is available in the Artifactory repo
- The compiled resources fall under the same rules as the source code for BIOS projects (e.g. sharing only within the BIOS user group)
- To enable access for all BIOS members, use the Bring-Your-Own-Group feature in the BDC portal and add the group "RB_SDE_SOCO_bios_user_UF"

4.6. Documentation

Most of the general documentation is valid for us as we use an out-of-the-box installation
Using Artifactory

JFrog also offers several training options in their JFrog Academy.

4.7. Known limitations of Artifactory

The JFrog product Artifactory has unfortunately some limitations and is not very good in dealing with them. This type of misuse is not prevented by the product and cause a lot of stability and availability issues.

4.7.1. Use of properties

You can define properties for artifacts and JFrog recommends to not use too many per artifact (<100). What happens in the background is that each property is stored as a separate entries in the properties database table (node_props). This will cause this table to grow a lot which makes is slower to query. This table is queried a lot as the properties feature is also used for general attributes which are required for displaying artifacts in the GUI or during AQL queries. Our largest

ever downtime of Artifactory was caused by the fact that this properties table grew so fast that the database optimization processes were not able to cope with it. As a result the table became unusable which caused the whole system to be unusable.

Artifact Property length should not exceed ~200 (or more) characters.

4.7.2. Deletion of Artifacts

While this is not a limitation, it is important to understand that when you delete an artifact, it is moved to the Trash Can. This changes the database entries, as the Trash Can is handled like a repository and the links to the deleted artifacts need to be overwritten. When we see a lot of artifact deletions (100 million) in short time intervals, this can also cause the database tables to become unusable.

4.8. Upload Limit due to Application Firewall

Due to security reasons, we operate artifactory and all other services behind an application firewall. The best currently available services in Azure has a maximum upload size of about 4GB per data chunk.

Artifacts type generic

The total file size limit is currently 4GB per file. It is normally uploaded in multiple chunks.

Artifacts type docker

The upload limit is not for the total size of the docker image but per docker level. Each docker level needs to be smaller than 4GB.

4.9. Docker Repos in Artifactory

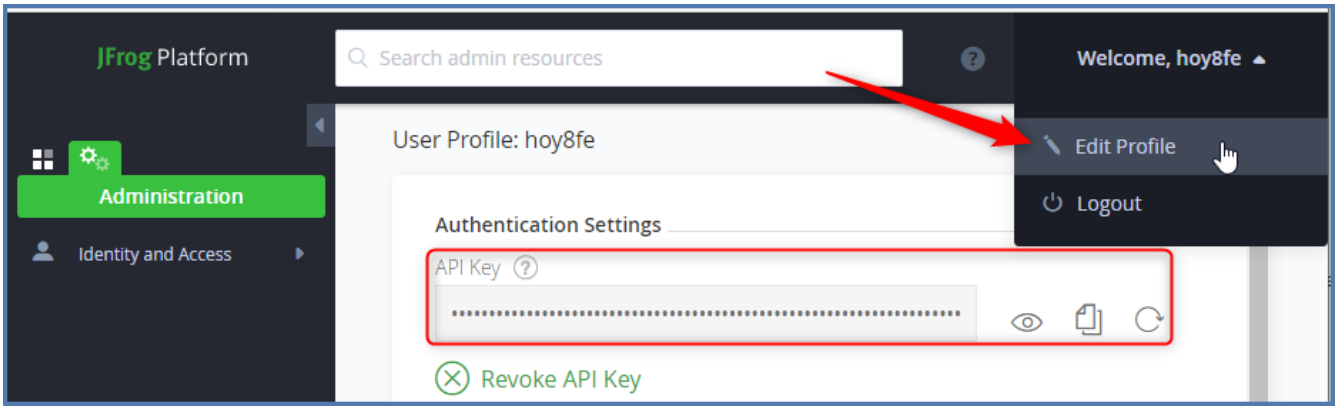
Since Artifactory 7.7 it is possible to host docker repositories in Artifactory@BDC!

There are three types of repositories supported:

- **Local repositories** are a place for your internal Docker images. Through Artifactory's security capabilities, these are secure private Docker registries.
- **Remote repositories** are used to proxy remote Docker resources such as Docker Hub.
- **Virtual repositories** can aggregate multiple Docker registries thus enabling a single endpoint you can use for both pushing and pulling Docker images.

Prerequisites

In order to use docker images, you have to create your own **API Key**. You can't use your regular password, since we're using a "single click" SAML authentication.
To create this API key, just log into Artifactory and click on your username in the top right corner. Now select "Edit profile". In this window you're able to create, regenerate, or view your API key. Please keep your API key as secure as your password, i.e. don't share or publish it anywhere and don't write it on a post-it sticked to your screen, not even in your home office.



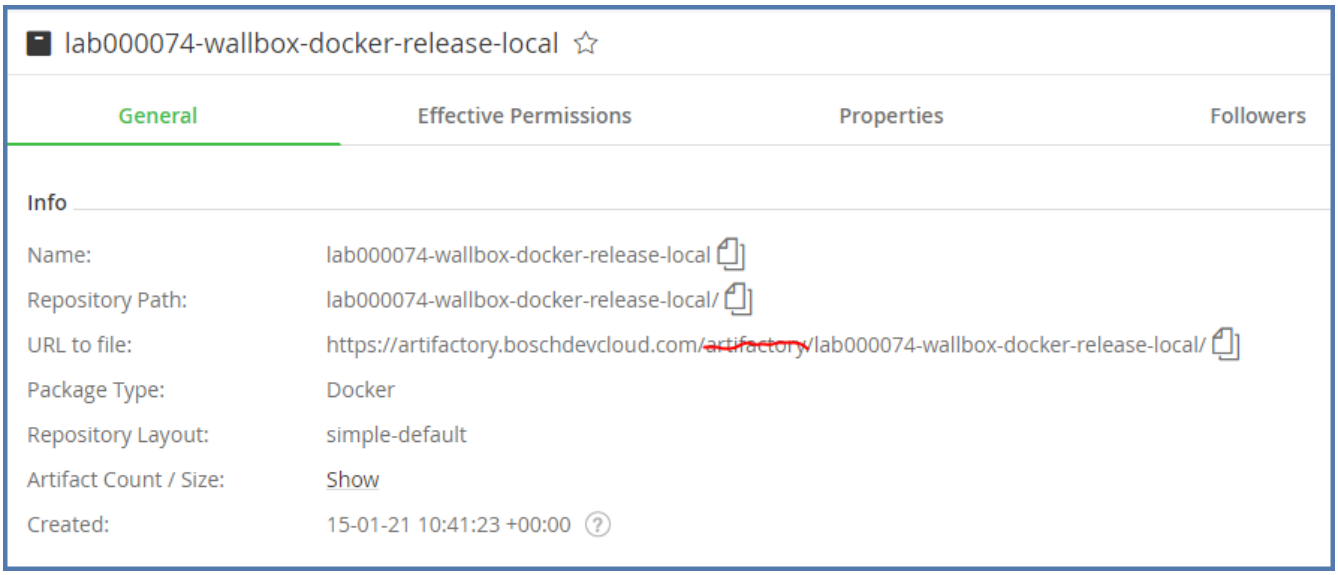
How To

Artifactory supports all relevant calls of the Docker Registry API so you can use Docker client to transparently access images through Artifactory:

01-Instance

$ docker login -u <userPrincipal> artifactory.boschdevcloud.com
e.g. docker login -u abc2fe artifactory.boschdevcloud.com
Remember to use your personal API Key as password, see "Prerequisites".

$ docker pull artifactory.boschdevcloud.com/<DOCKER_REPOSITORY>:<DOCKER_TAG>
e.g. docker pull artifactory.boschdevcloud.com/lab123456-mydocker-remote/hello-world:latest
Please do not use the shown URL in the artifactory repo window. Remove the "artifactory" part, else you will not be able to pull the image.



02 & 03 Instances

$ docker login -u <userPrincipal@bosch.com> artifactory-<02|03>.boschdevcloud.com
e.g. docker login -u abc2fe@bosch.com artifactory-02.boschdevcloud.com
Remember to use your personal API Key as password, see "Prerequisites".

For instances 02 & 03 users need to use "NTID@bosch.com" for docker login. As instances 02 & 03 uses OpenID.

4.10. Mirroring / Synchronization

In some cases you might think about a mirror repository or even a repository sync.

Mirror from On-Prem to BDC

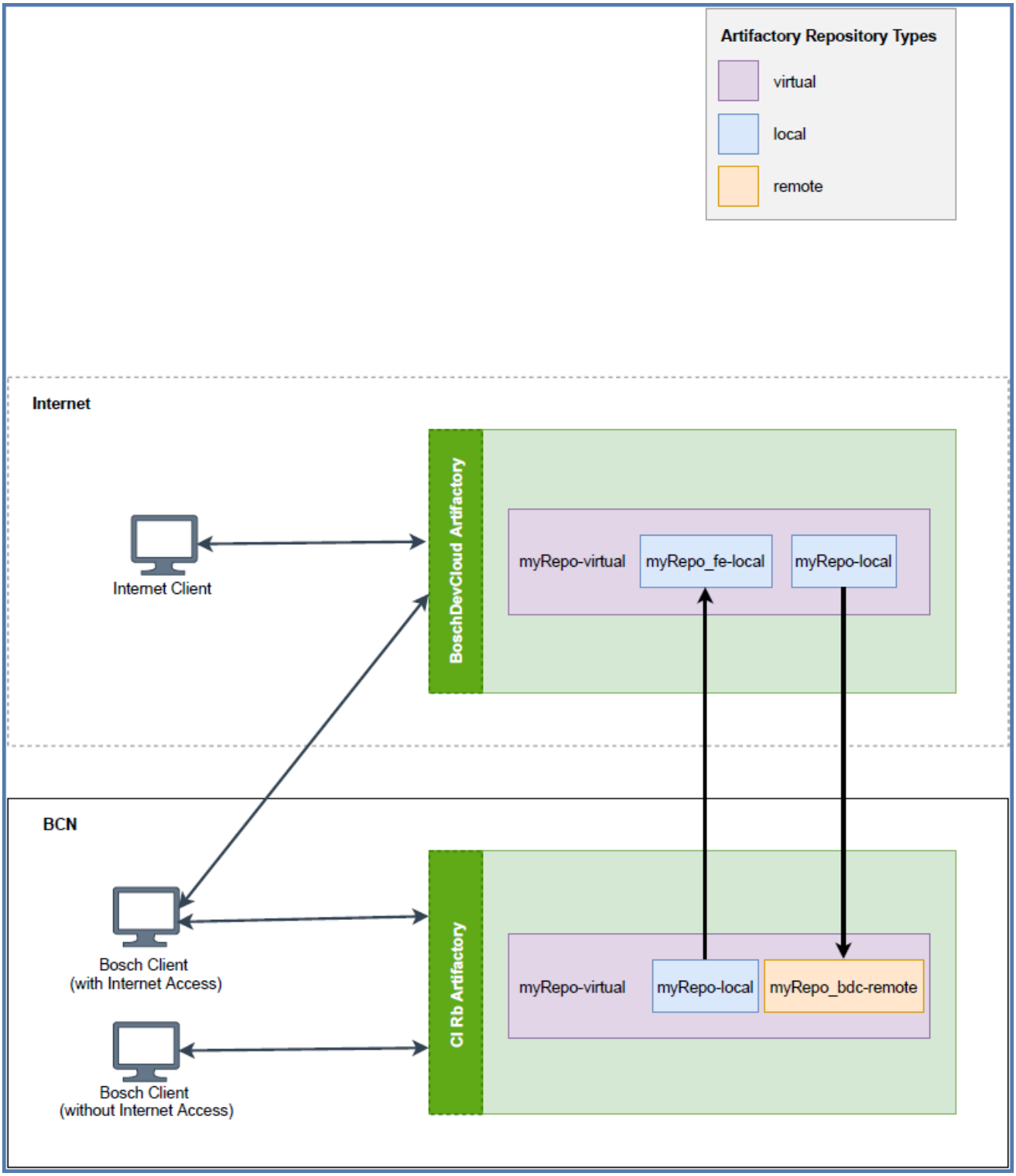
It’s possible to copy (i.e. "mirror") your on-prem repository into the BDC repository. We just need to change a few settings to realize it. Due to technical restrictions, we can’t push large files to our BDC-Artifactory. All data traffic has to pass our "Application Gateway" and our Firewall, thus the data-size is restricted to currently about 4GB (details see here).
Keep in mind: It’s only a one-way synchronization, therefore the target repository in our BDC will be read-only.

Mirror from BDC to On-Prem

It is possible to create a remote repo in the on-prem artifactory that refers to a repo of the BDC Artifactory.

Synchronization

You might have a specific use-case, which requires a bi-directional copy/sync process from the on-prem to/from the BDC repository. We could realize such a request, but it needs some effort. We’d have to set up a mix of local, remote and virtual repositories and an additional sync-user. Please get in contact with us, so we could define the details.



4.11. Retention Policy

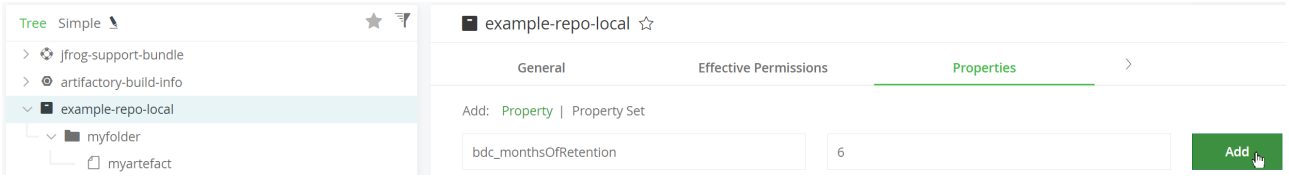
To enable retention of artifacts in a LOCAL repository you have to set the following properties

At the moment we do not support automatic cleanup of Docker and Conan repositories as there are many dependencies between folders and artifacts and deletion could result in failures during package creation.

Propertyname	Purpose	value	Applies to
bdc_monthsOfRetention	Defines after how many month unused data is deleted	positiv number > 0	repository
bdc_cleanupSkip	Skip cleanup on specific folders or artifacts	true	folder, artefact

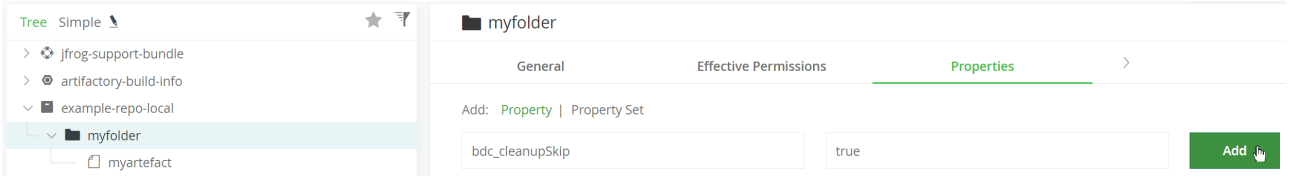
Please be sure to spell the properties and values as described - otherwise they won't work.

- **bdc_monthsOfRetention** (valid for Repositories)
  The value is the number of month an artefact is kept in the repository.
  It should be larger then 0, as set to 0, all artifacts would be deleted during cleanup.
  For example the property bdc_monthsOfRetention = 6 results in a deletion of all artifacts which haven't been downloaded within the last 6 months or have been created 6 months ago but never been downloaded.

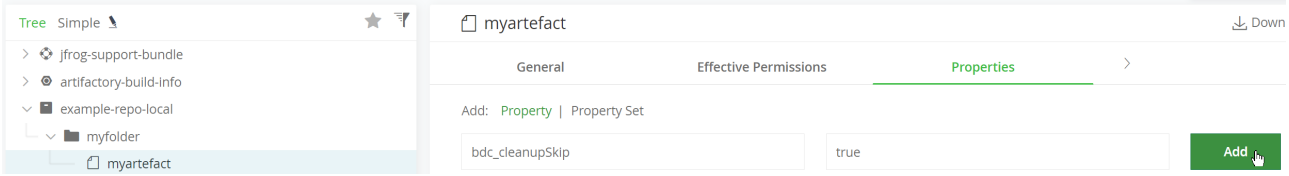


- **bdc_cleanupSkip** (valid for Directories(Folders) and Artifacts)
  Can be set to "true" on folders and individual artifacts

  Folders:
  This skips the deletion of artifacts inside the folder and all sub-folders, in a repository where the property bdc_monthsOfRetention is set.



  Artifacts:
  This skips the deletion of the artefact, in a repository where the property bdc_monthsOfRetention is set.



- **bdc_deleteEmptyDirs**
  This property has become obsolete as Artifactory will automatically delete empty folders after the last artifact has been removed.

4.12. Cleanup of retired artifacts

There are 2 scheduled jobs that are used to cleanup repositories, where the "bdc_monthsOfRetention" property has been set

- **Check for cleanup candidates**
  This jobs runs from Mo-Fr at 20:00 CET
  It only scans all the repositories, where the property "bdc_monthsOfRetention" has been set, to find retired files.
  The result of this check will be uploaded to the folder **_bdc_operations_logs/cleanup_candidates** within the related repository
  The retention of those results are set to 30 days
- **Cleanup retired artifacts**
  This jobs runs once a week on Sa at 20:00 CET
  It scans all the repositories, where the property "bdc_monthsOfRetention" has been set, to find retired files
  and moves them to the Trash Can, where they are kept for 10 more days.
  The result of this cleanup will be uploaded to the folder **_bdc_operations_logs/deleted_artifacts**
  The retention of those results are set to 365 days
- **Cleanup of empty folders**
  Artifactory will automatically delete empty folders after the last artifact has been removed.

4.13. Artifactory usage examples

4.13.1. Using AQL search

You can use AQL (Artifactory Query Language) to find items based on your search criteria.

Here a example how to find elements which have been modified in the last 3 days in repository "example-repo-local".

Note: You need to create an Artifactory Identity Token in your user profile first.

```
curl --location 'https://artifactory.boschdevcloud.com/artifactory/api/search/aql'
--header 'Content-Type: text/plain' --header 'Authorization: Bearer
cmV......addYourOwnIdentityToken...' --data 'items.find({"repo" : {"$match": "example-
repo-local"}, "modified" : {"$last" :
"3d"}}).include("repo","path","name","created_by","created")'

{
"results" : [ {
  "repo" : "example-repo-local",
  "path" : ".",
  "name" : "all-conan.txt",
  "created" : "2023-08-15T08:28:24.529Z",
  "created_by" : "eir2si"
},{
  "repo" : "example-repo-local",
  "path" : "_bdc_operations_logs/cleanup_candidates",
  "name" : "2023-08-14_cleanup_candidates.log",
  "created" : "2023-08-14T18:29:37.280Z",
  "created_by" : "bdcadmin"
```

```
},{
  "repo" : "example-repo-local",
  "path" : "_bdc_operations_logs/deleted_artifacts",
  "name" : "2023-08-12_cleanup.log",
  "created" : "2023-08-12T18:26:35.621Z",
  "created_by" : "bdcadmin"
} ],
"range" : {
  "start_pos" : 0,
  "end_pos" : 3,
  "total" : 3
}
}
```

4.14. Artifactory FAQ

4.14.1. Technical Questions

Is it possible to delete the Artifactory through BDC API portal?

No, You need to request the BDC team to delete the repos. You can create a ticket in Jira Service Desk. However, we are working on offering you an API in the future.

Is there any other option for permission management without using provided IdM roles?

Create a permission target and make use of the Bring Your Own Group option. With this you could add your Azure AD object (AAD group) to the permission target instead of using BDC provided IdM role.

How can I request access for a virtual repositories?

Permissions are configured for local and remote repositories only. So when accessing content that are aggregated in virtual repository, permissions for that resource will be granted based on the permission on local / remote repo permissions. Feel free to refer to official Artifactory docs.

I have created a Virtual Repository from BDC API portal, now I want update virtual repository with some more repos, how to do it as there is no option available in API portal?

Currently updating virtual repository is not possible via API portal, Kindly create a ticket in Jira Service Desk with BDC ID, Permission Target Name, repository list to add in virtual repositories.

I have ordered a new repo and not sure how to authenticate it from CLI?

Create an identity token from Artifactory and make use of set me up option in artifactory. In case of docker repos, refer to the respective Docker section.

Due to a compliance, Can I mirror Docker Hub or Cloud Artifactory to Artifactory?

Yes, you could make use of Artifactory remote repo feature with upstream repo as docker hub or

Cloud Artifactory repos.

4.15. Feedback

We look forward to receiving your feedback, which will help us to improve in the future!

Chapter 5. Atlassian Cloud

The Atlassian Cloud Service supported by the BDC currently includes the following products: Jira, Confluence and Jira Service Management.

To find out more about the features of each product please check the official Atlassian Website:

- Jira Software
- Confluence
- Jira Service Management(former Service Desk)

In addition to having a site, the site has to be managed via Atlassian Access to ensure authentication with the Bosch Account and proper licensing. This is part of the BDC Service as well as the SSO configuration and Idm Integration of the Atlassian Cloud.

5.1. What do I get for Atlassian Cloud and how to order

You can order a site and then chose which of those products you want to use in your site and how many user licenses you purchase per service. A site is similar to having your own virtual server instance of those products.

The order process is a bit more complex as usual with the BDC due to the business model of Atlassian to include partners in their sales process. Feel free to set up a meeting with Christine Welschof when you require more information about the different products and licenses or need support with the site or the project/space registration process.

The order process of your own Atlassian Cloud Site is the following.

1. Decide which products (Jira, Confluence or Jira Service Management) you would like to use and how many licenses per product are required. Please note that the BDC and the Atlassian Partner need two extra licenses per used product for the IdM Integration and set-up of the site.
2. Please reach out to the BDC Service Desk and provide the following information:
    - BDC BDC Business Account ID:
    - Site name you want to have that will be visible in the URL (recommendation: bosch-generic name, not department/group name in case of re-organizations):
    - Jira Software:
        - Number of Licenses:
        - Which License plan: Standard or Premium, see details here?
    - Confluence:
        - Number of Licenses:
        - Which License plan: Standard or Premium, see details here?
    - Jira Service Desk:
        - Number of Agent Licenses (only people working on tickets need a license, not customers

who create tickets):

        - Which License plan: Standard or Premium, here?
    - Plugins, if you wish to use please sent link of the Plugin within the Atlassian Marketplace.
        - Please note that Plugins have to be officially onboarded by the customer itself in some use cases. Therefore, please take a look at the guideline in our user guide.
    - Site Admins of the Atlassian Site (site admins can administer the whole site, should be only a few users):
    - Cost Center to which the license costs can be charged to (standard contract duration with Atlassian is 1-year, billing will be done on a monthly basis):

We are working with Atlassian to simplify this process. They are working on a better service for Enterprise customers and we hope this will simplify this process a lot going forward.

Start to set-up your Atlassian Cloud Site:

- After the order process is completed, your site on atlassian.net will be setup and handed over to you.
- Site Admins can then start in Atlassian and create Confluence spaces and Jira projects in which they want to collaborate with others.
- To set up the IdM Integration, every project and space created in the Atlassian Site has to be registered in the BDC Portal by the customer itself. This will create the required IdM access rights and implement the permission and user sync between IdM and Atlassian.net. Your colleagues can then use the IdM Access rights to work together with you when you share the link to your site and they login with their ntid@bosch.com mail address! More information about the registration of the projects/spaces and standard IdM access rights can be found in the BDC Portal under the corresponding API Call.

5.2. Cost for Atlassian Cloud

There are two cost points for the Atlassian Cloud: the Atlassian site & product licenses itself and the costs for the BDC Service (configuration of site in Bosch Organization in Atlassian, SSO, Idm Integration).
Below the overview:

	Costs per month	**Remark**	**Costs charged against**
Atlassian Cloud Operations	4,57€ per user or agent (not customers of Service Desk)	BDC Atlassian Cloud: SE-0004331	Cost center of the **User**

Costs for the Atlassian site	Depending on number of users and Atlassian Products required.	For a first estimation of the License cost check out the Atlassian Website (Jira Software, Confluence, Jira Service Management). Based on your request an individual offer will be requested which then includes the Bosch discount.	Customer cost center provided during ordering process

5.3. Permission management in Atlassian Cloud

A summary of the Standard BDC Idm Access Rights can be found under the POST API Calls in the BDC Portal.
The IdM Access rights are project/space specific. Therefore, each project/space has to be registered in the BDC Portal by the customer. Users can only see and work on projects/spaces for which they have the corresponding IdM access right. If you use the BYOG feature, the Azure AD Group needs to be registered to one or multiple projects in the BDC Portal.
Users can login to your site using their ntid@bosch.com mail address for which SSO is configured.

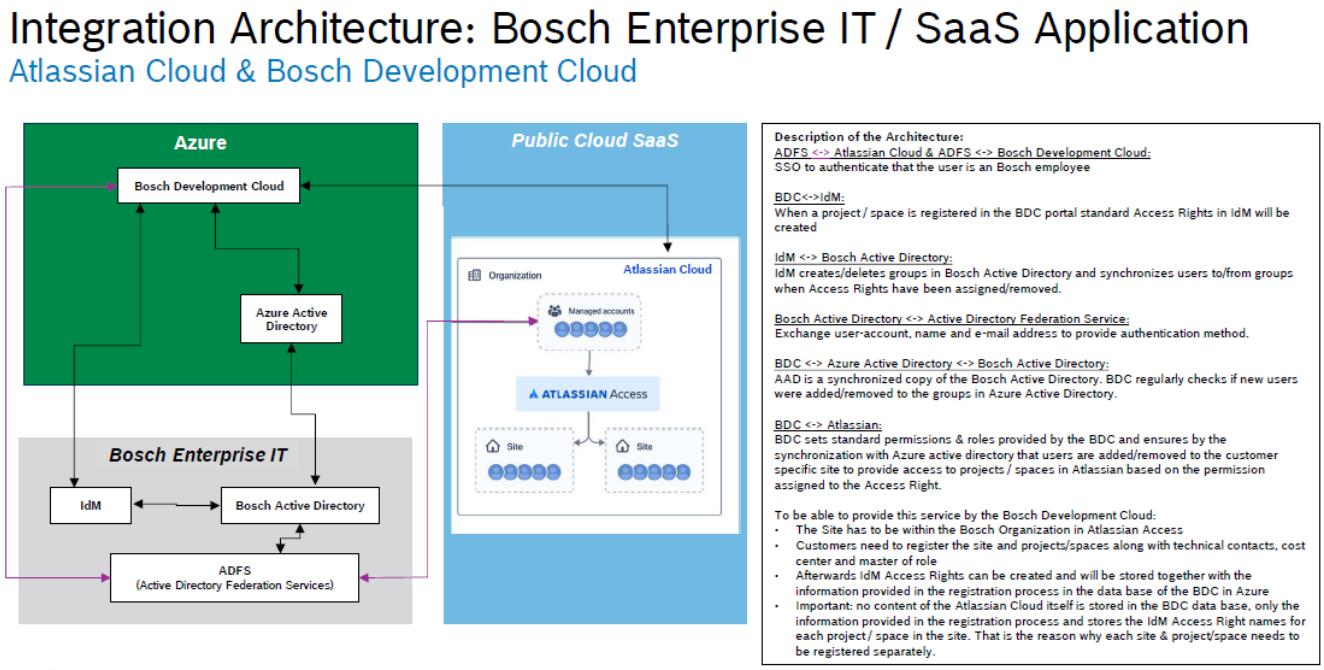
As soon as users do not have any IdM Access rights of your site assigned to them anymore or they are not part of any of your own Azure AD Groups, the licenses will be available again and can be re-used by others. However, the user will still remain in the Users List in Atlassian, but without site access. With that, the contributions of the user and its name will remain in your site.
Please make sure to not add any default access groups to the product access per product, otherwise the BDC user sync and IdM Integration to make licenses available again if IdM Access Rights are revoked will not work.

User synchronization from IdM to Atlassian:
The User Synchronization currently starts at 3am, 6am, 9am ,12am, 3pm, 6pm (UTC) and takes around 2 hours.

5.4. Shared responsibility for Atlassian Cloud

The user management is in the responsibility of the customer. Each project/space has to be registered in the BDC Portal to create the corresponding IdM access rights.
If customers want to use the BYOG feature, they are responsible that the group is AIM compliant and has successfully passed all Bosch security and approval processes. This includes for example that permissions and changes to this group are documented and reviewed periodically according to the corporate rules.
Furthermore, customers have to make sure that Plugins they use are conform with current Central Directives. Please take a look at the Guideline that is aligned with CI/PIP to find out more about this topic.

5.5. Architecture Overview



5.6. FAQ

How do I find out which projects are already registered and also the corresponding IdM Access rights?
Use the GET API Call in the BDC Portal

How can I use customized Atlassian Roles and the BYOG feature?
In Atlassian you can create roles and assign roles certain permission. To use this role, you can then use our Portal and assign your own group to that customized Atlassian Role. Important: only use roles, not groups. Groups are automatically created when you use the BDC Portal.

Can I change the permissions of the Standard BDC roles in Atlassian if I want to remove or add certain permissions to that?
No, if you want customized permissions, you have to create your own role in Atlassian and use the BYOG Feature in the Portal. Changes will be overwritten by the user-sync.

How can I see how many licenses per product we have or how many are still available?
Site Admins can check this in Atlassian directly in the Product Access Overview within the Admin View.

Can I get a licenses for e.g. 4 months only?
No, standard contract with Atlassian is one year, therefore this is the minimum. However, the license costs are then charged on a monthly basis towards the cost center provided during the ordering process.

Do customers for Jira Service Desk need a license?
No, only agents, the users who are working on the ticket, need a license.

5.7. Plugins

SaaS Providers like Atlassian, AzureDevOps Services or GitHub offer marketplaces that include first- or third-party plugins, apps, extensions or add-ons. In many cases, the plugins, apps, extensions or ad-ons can be easily installed. Some of them can be used for free, for others licenses have to be bought. However, before using them, it is important to check, if and under which conditions it is allowed from Bosch side to use these. Therefore, please check the FAQ section of C/IDO who is responsible for the SaaS - Onboarding Process.

To find out which plugins, apps, extensions or add-ons is already onboarded, please search for it in LeanIX.

5.7.1. Cost for Plugins

The costs for Plugins/Apps/Extensions can be found in the marketplace of the SaaS Provider.

5.7.2. Shared responsibility for Plugins

Customers have the possibility to choose if they want to use plugins, apps, extensions or add-ons. If decided to use plugins, apps, extensions or add-ons it is in the responsibility of the customer to make sure that they are compliant with all Bosch regulations (Central Directives, EISA, works council agreements).
The Bosch Development Cloud does not offer product support for plugins, apps, extensions or add-ons or the onboarding process in case it is needed.

5.8. Feedback

We look forward to receiving your feedback, which will help us to improve in the future!

Chapter 6. Azure Devops Services

BoschDevCloud Azure DevOps Services provide an integrated, collaborative environment that supports Git, continuous integration, and agile tools for planning and tracking work.

Choose Azure DevOps Services when you want:

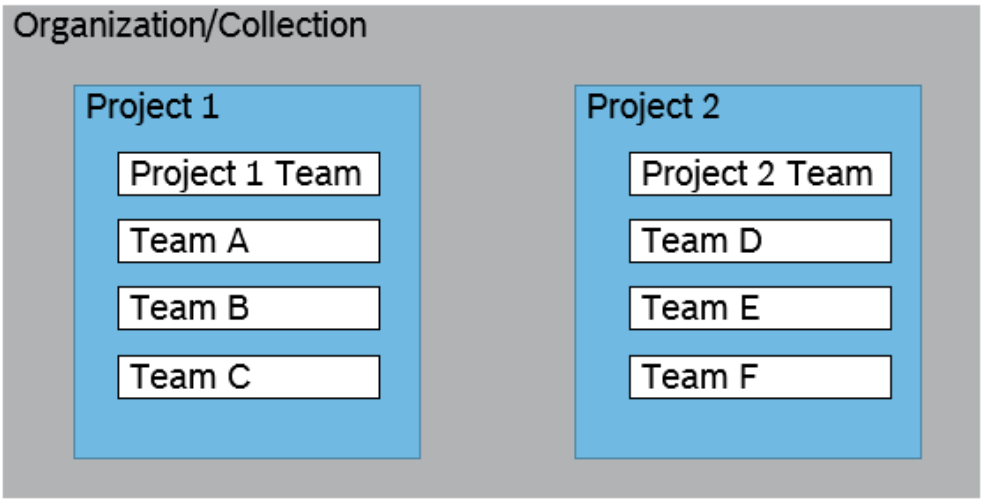
- Quick set-up
- Maintenance-free operations
- Easy collaboration across domains
- Elastic scale

You can access the service via https://dev.azure.com/<your organization name>.

6.1. What do I get for Azure Devops Services and how to order

What is an organization, a project, a team?

An organization is a collection which can hold several projects. Each project can hold several teams. Azure DevOps Services uses organizations as managing entity. You will get your own organization with the capabilities to full configure and administer it.



As soon as you have a BDC assigned you may now:

- order a new organisation or project
- register an existing ADO organization or project

Access to ADO managed by BDC is Bosch-compliant granting access by IdM-roles.

Ordering a new organisation or project

Ordering an Azure DevOps organization or a project can be done by API using the BDC-Portal.

Under "my BDCS" you will find two APIs, one to create a new organization and one to create a new project. So you need to first create your organisation and the project inside this organisation.

Please remember you have to have a BDC - if you don't have one, just provide us the information requested here.

Register an existing ADO organization or project

Existing ADO organization or projects from a Service-instance can be moved inside the BDC. First ensure you make the account "BSORFE" the owner of your organisation. You can then use the Portal to register first your organisation and then all your projects.

Alternatively you can send an email to BD/PLS BoschDevCloud Support (BD/PLS3) and provide the following information:

Organization	
Do you have already an AzureDevOps Organization in Azure DevOps Services?	( ) Yes, within the BDC ( ) Yes, but not within the BDC ( ) No, please create an organization for me in the BDC
Name of the organization in AzureDevOps *(mind no special characters like: @, /, _ or "blank" possible)*	
Who will be owner of the new organization? *When you already have an organization please check and make "BSORFE (BD/PLS)+ " an owner, otherwise we cannot proceed.*	
Who is technically responsible for the orga? *(nt-account)*	
Project	
Do you already have an AzureDevOps project?	( ) Yes ( ) No, please create a project for us
Name of the project in AzureDevOps *(mind no special characters like: @, /, _ or "blank" possible)*	
Who is technically responsible for the project? (nt-account)	
General	
Costcenter	
Master of Data *(a role, not a person)*	
Master of Role *(one(!) role, not a person)*	

6.2. Cost for Azure Devops Services

Currently the costs for Azure DevOps are:

	Costs per month	Costs charged against
Azure DevOps organization & project	depending on the resources booked on organization level - see directly at Azure	cost center of the DevOps-Org
MS Visual Studio License	see here	cost center of the user
SaaS operations	2,93€/user/month	cost center of the user

Users can use this link to subscribe for a license or for information:

- Azure DevOps CAL Service can be canceled immediately
- VS Subscriptions after the minimum service usage time of 12 month

To check which users use service open Reporting Services Web Portal (bosch.com) and filter accordingly (authorization required).

For more information on the MS Visual Studio License, please use the following link also mentioned in the table above: Visual Studio Subscriptions and Visual Studio Support

6.3. Permission management in Azure Devops Services

Which IDM-roles are created during the creation process?

These IDM-roles will be created:

BDC_AzureDevOps_org<org_idm_id>_proj<proj_idm_id>_admin
This role will be added to the Project Administrator-group on project-level in Azure Devops.

BDC_AzureDevOps_org<org_idm_id>_proj<proj_idm_id>_contributor
This role will be added to the Contributor-group on project-level in Azure DevOps.

BDC_AzureDevOps_org<org_idm_id>_admin
This role will be added to the Project Collection-group on organization-level in Azure DevOps. This is an optional role and will only be created on demand.

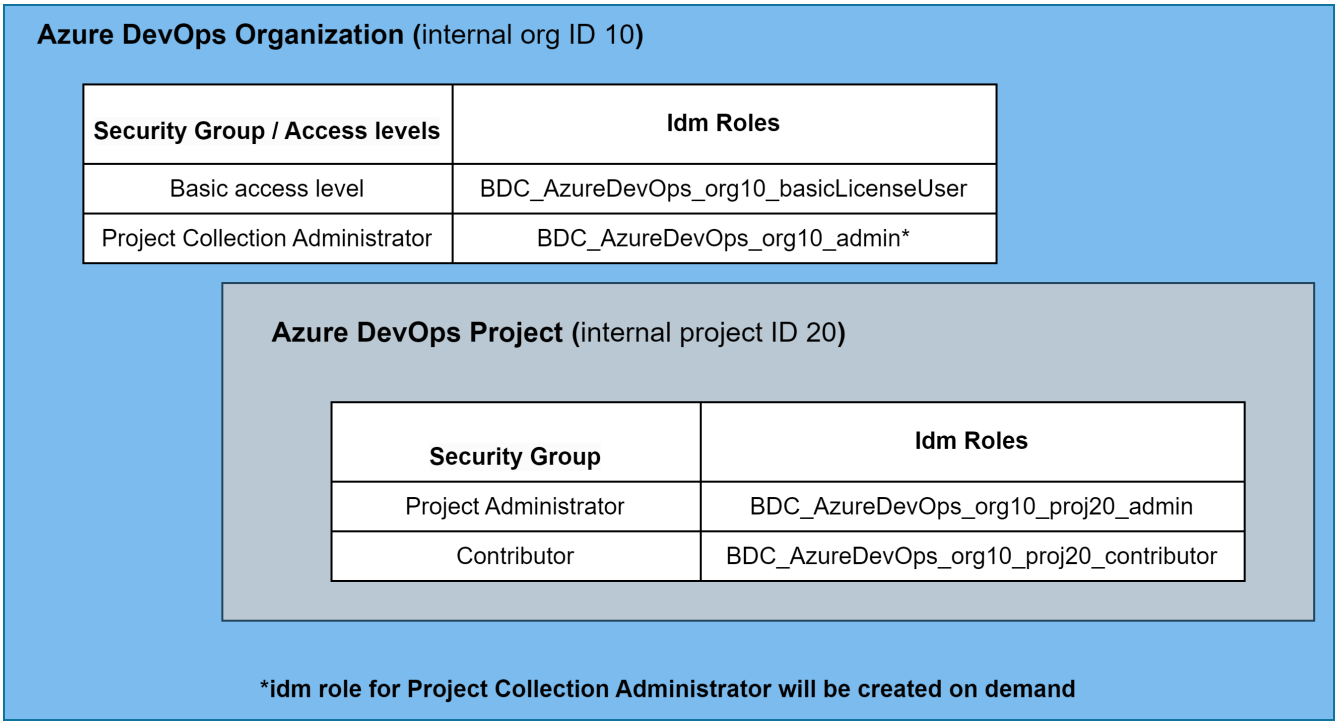
BDC_AzureDevOps_org<org_idm_id>_basicLicenseUser
This role will be added on organization-level in Azure DevOps and this role will give Basic license to the users.

1. If user have to be part of other groups(like Build Administrators, Release Administrators), then project admin has to assign them directly into these groups and also the respective user has to be either in idmroles of Contributor/Admin.
2. The project admin has to verify the users once a year whom are assigned other than contributor/Administrator

For more information about security groups and access levels, please refer this related Microsoft document.

How are the IDM-roles assigned to the groups in Azure DevOps?

The IDM-roles are assigned as illustrated below and here orgID 10 & projectID 20 are used as an example:



6.4. Shared responsibility for Azure Devops Services

Who is responsible for the used extensions?

Please be aware that the project admin is completely responsible for all the extensions which are used in the project. The list of possible extensions can be found in the Marketplace for extensions.

6.5. Documentation

The complete documentation from Microsoft is valid for us because we use Azure DevOps Services as a SaaS integration:

Please be aware of differences between Azure DevOps Server and Azure DevOps Services.

6.6. Support

Due to the fact that Azure Devops Services is integrated as a SaaS-Integration, the team of the Bosch Development Cloud does not provide application support for this service.

6.7. FAQs

6.7.1. What is the difference between Azure DevOps Server and Azure DevOps Services?

As both services are provided by Microsoft, we would like to forward you to the Microsoft

documentation.

Some important advantages of the cloud-solution are:

- Upgraded every three weeks
- Accessible from anywhere
- Cloud-first innovation
- Leverage developer services in the Microsoft cloud

6.7.2. How to migrate a project from one organization to another organization

Microsoft doesn't provide a solution for that. However, they provide a documentation about the possible ways to achieve this.

There are also some 3rd party tools which can be used, like Opshub Visual Studio Migration Utility.

6.7.3. How to move an organization (outside of BDC) into the BDC?

If you have an organization and you want to move into the Bosch Development Cloud, please use also the same ways as described under the section *Ordering* for registering the organization.

6.7.4. What to do if I want to adapt the standard process?

In some special cases it is required to have administrative access at organization level, e.g. to adapt the standard process to your needs. This is possible with an optional role. With this role it is possible to grant you the administration access at organization level. To order this role, please send an email to BD/PLS BoschDevCloud Support (BD/PLS3) including the MoR for this new role.
You can also submit a ticket in our Service Desk.
Please be aware that you are responsible for the onboarding of the used extensions if you add extensions to your organization.
In this case you will get this additional role: **BDC_AzureDevOps_org<org_idm_id>_admin**

6.7.5. How to add ARM service connections using a service principal

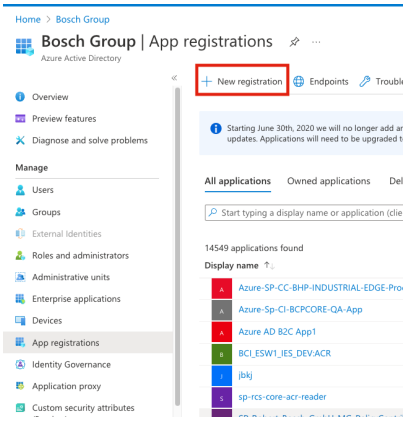
As a 'project administrator' you are able to create new service connections. Due to backend permissions you will not be able to 'automatically' validate a service principal, please select 'manual' and enter the service principal id and password. NOTE: In case you own a Lab within the BDC we already created a service principal for this lab. You can use it to get access to your Lab resource group. You can find the credential within the key vault for your Lab.

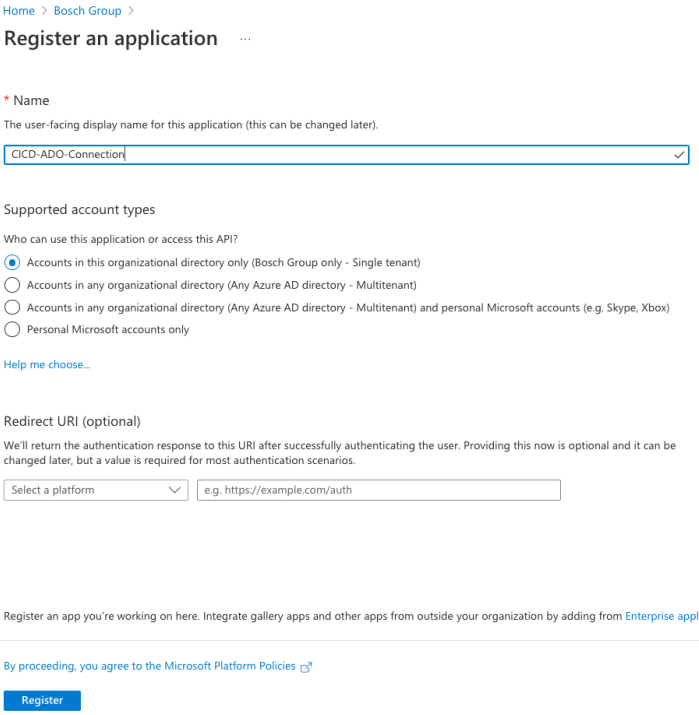
6.8. Azure DevOps Service connections

6.8.1. Service connection to a BDC Lab resource group

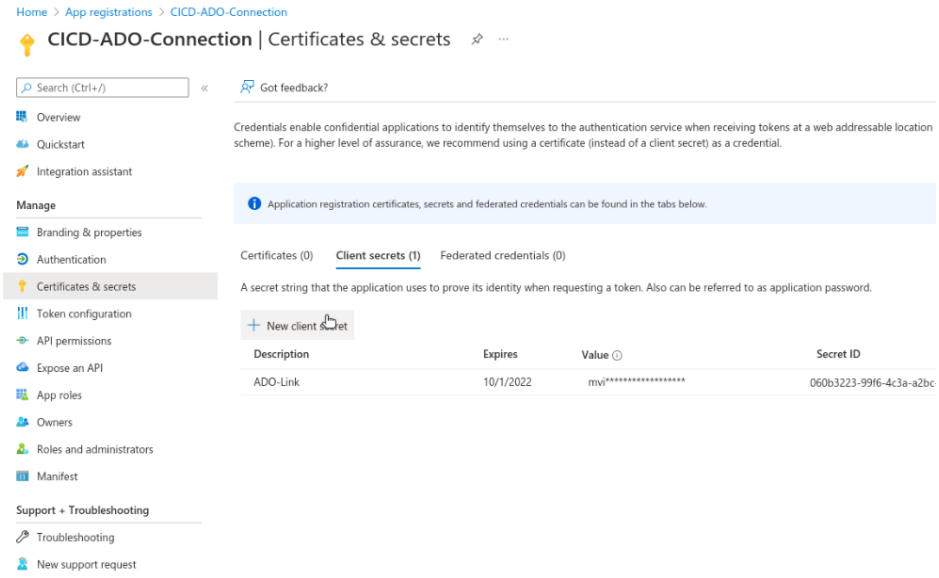
To create a service connection in Azure DevOps pointing to your Azure lab resource group, an AAD app registration is needed.

This app registration can be created in the Azure Active Directory.





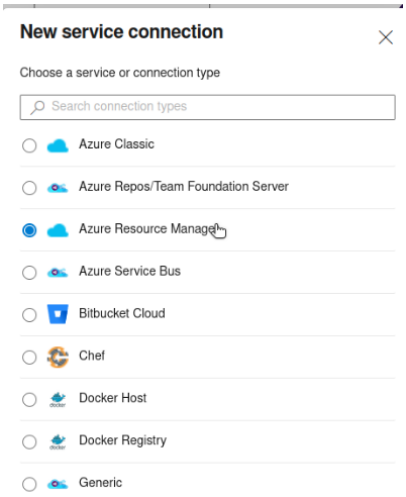
This app registration has to have a client secret which will be used in the service connection, so a new one might be created for this.

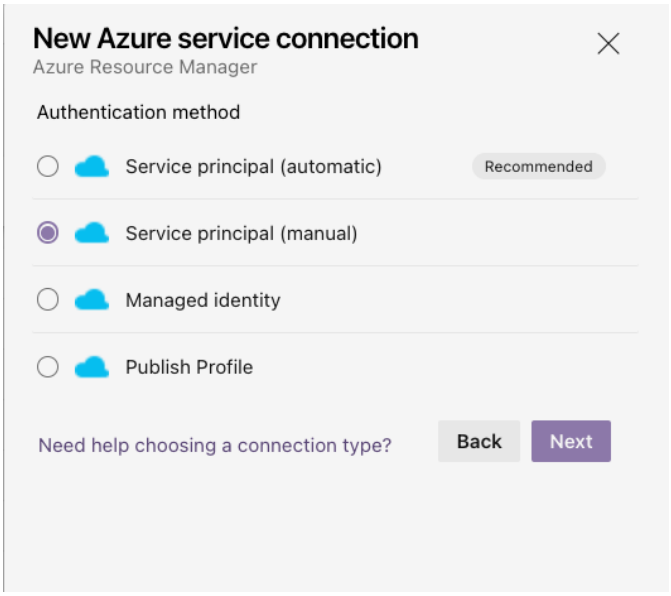


When the app registration was created, a BDC member has to assign it contributor permissions to

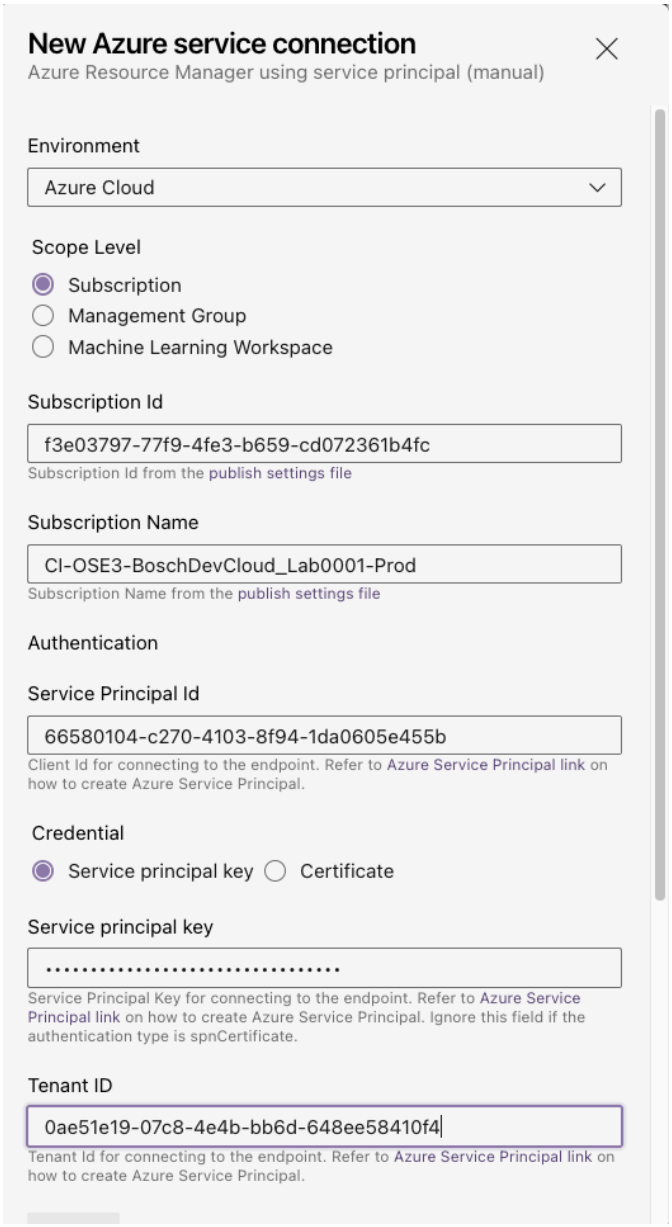
your resource group.
Feel free to send us a short request via our ServiceDesk or by mail with the name and application id of your app registration.

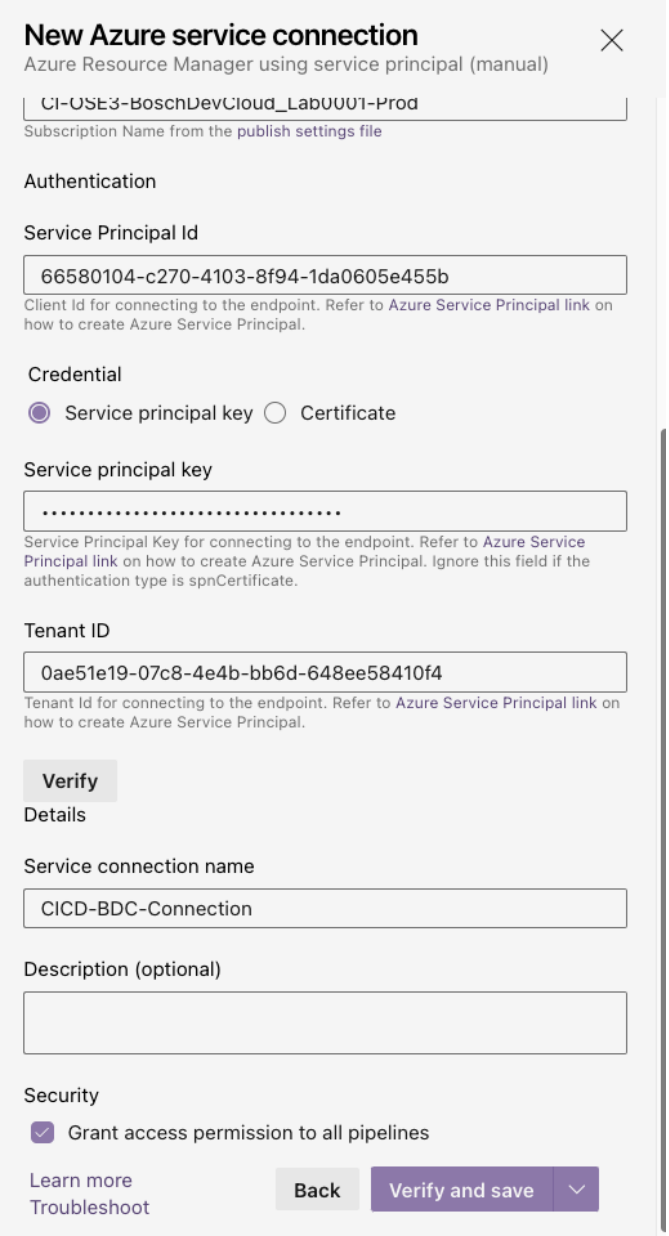
Afterwards, a new "Azure Resource Manager" service connection can be set up via "Service principal (Manual)".





Here the app id and the value of a created secret can be specified.


New Azure service connection
Azure Resource Manager using service principal (manual)
Environment
Azure Cloud
Scope Level
Subscription
Management Group
Machine Learning Workspace
Subscription Id
f3e03797-77f9-4fe3-b659-cd072361b4fc
Subscription Id from the publish settings file
Subscription Name
CI-OSE3-BoschDevCloud_Lab0001-Prod
Subscription Name from the publish settings file
Authentication
Service Principal Id
66580104-c270-4103-8f94-1da0605e455b
Client Id for connecting to the endpoint. Refer to Azure Service Principal link on how to create Azure Service Principal.
Credential
Service principal key
Certificate
Service principal key
Service Principal Key for connecting to the endpoint. Refer to Azure Service Principal link on how to create Azure Service Principal. Ignore this field if the authentication type is spnCertificate.
Tenant ID
0ae51e19-07c8-4e4b-bb6d-648ee58410f4
Tenant Id for connecting to the endpoint. Refer to Azure Service Principal link on how to create Azure Service Principal.



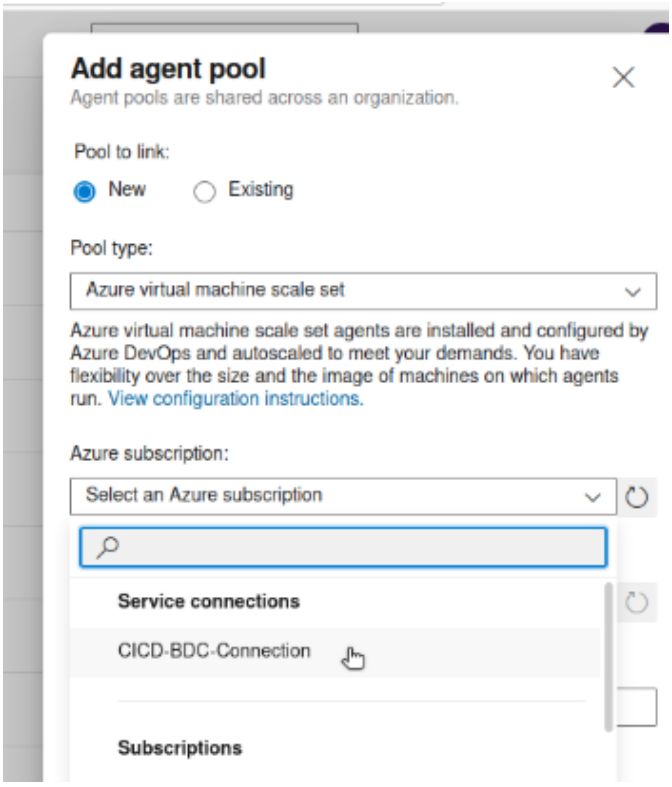
The other parameters are as follows:

Subscription Id
f3e03797-77f9-4fe3-b659-cd072361b4fc

Subscription Name
CI-OSE3-BoschDevCloud_Lab0001-Prod

Tenant ID
0ae51e19-07c8-4e4b-bb6d-648ee58410f4

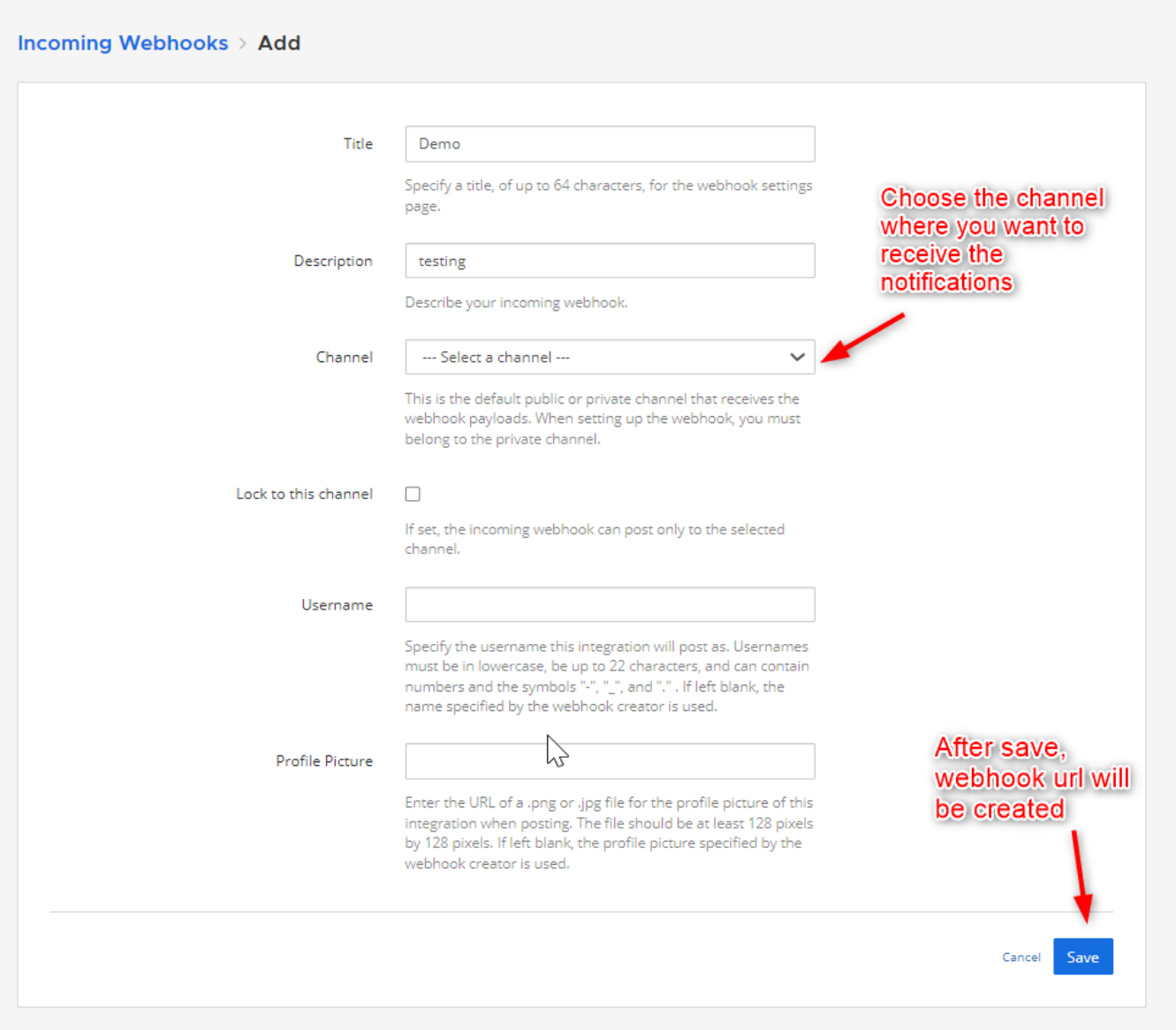
When this service connection is setup successfully, it can be used to create VMSS agent pools e.g



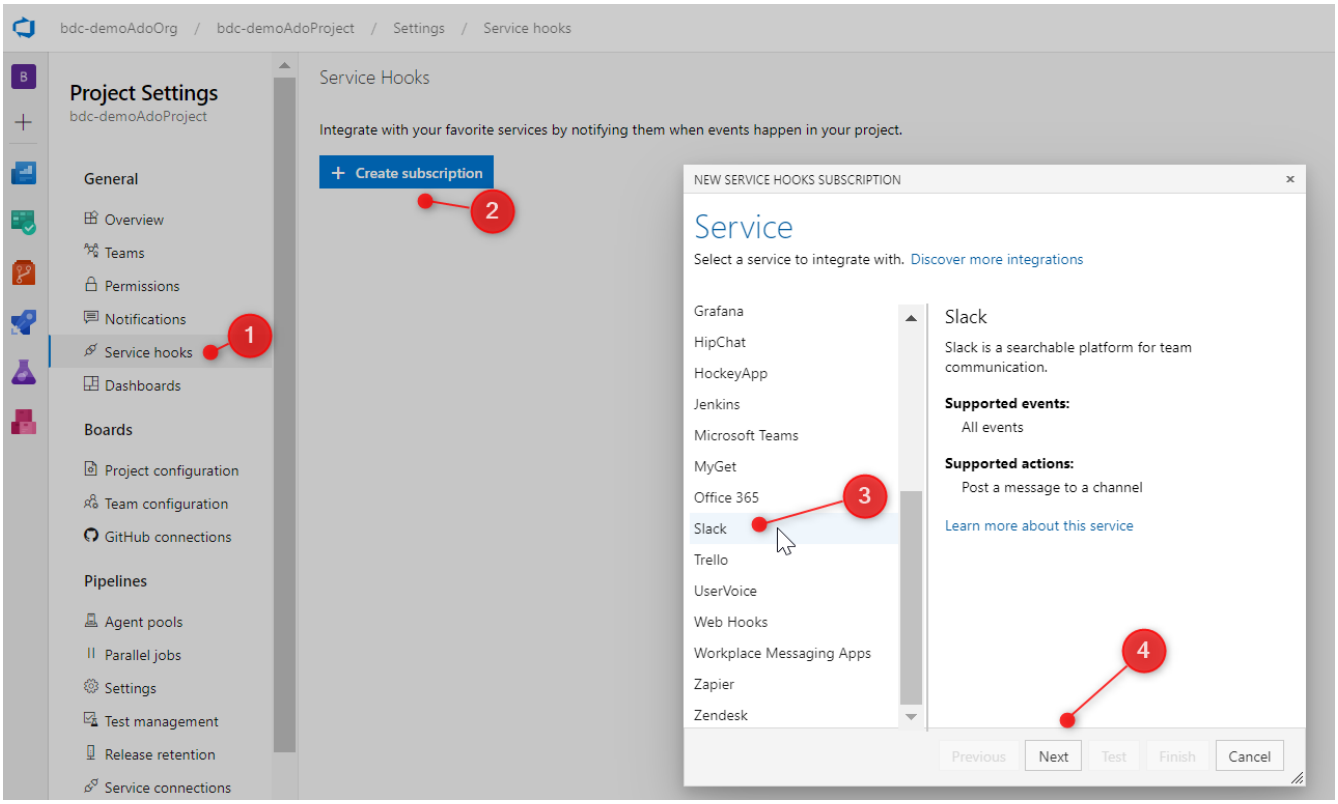
6.9. Incoming webhook configuration in Mattermost with Azure DevOps

- This chapter is written with context of posting notification message from Azure DevOps to Mattermost on an event based.
- In Mattermost, go to **Product menu** > **Integrations** > **Incoming Webhook** . Only Mattermost team admins can able to **add incoming webhook**
- Add a name and description for the webhook and select the channel to receive webhook payloads, then select **Save** to create the webhook.
- You will end up with a webhook endpoint that looks like below :

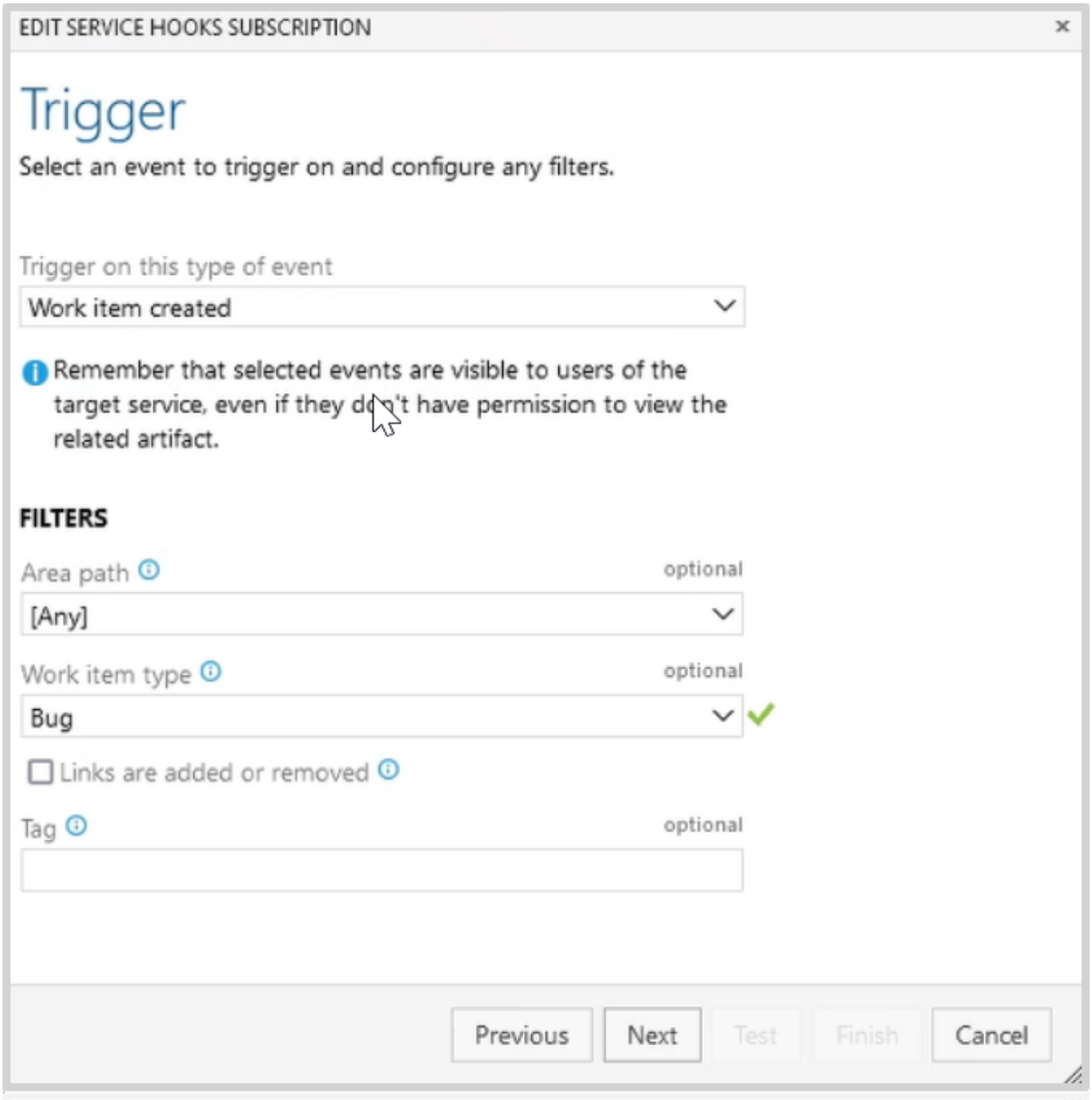
https://mattermost.boschdevcloud.com/hooks/xxx-generatedkey-xxx



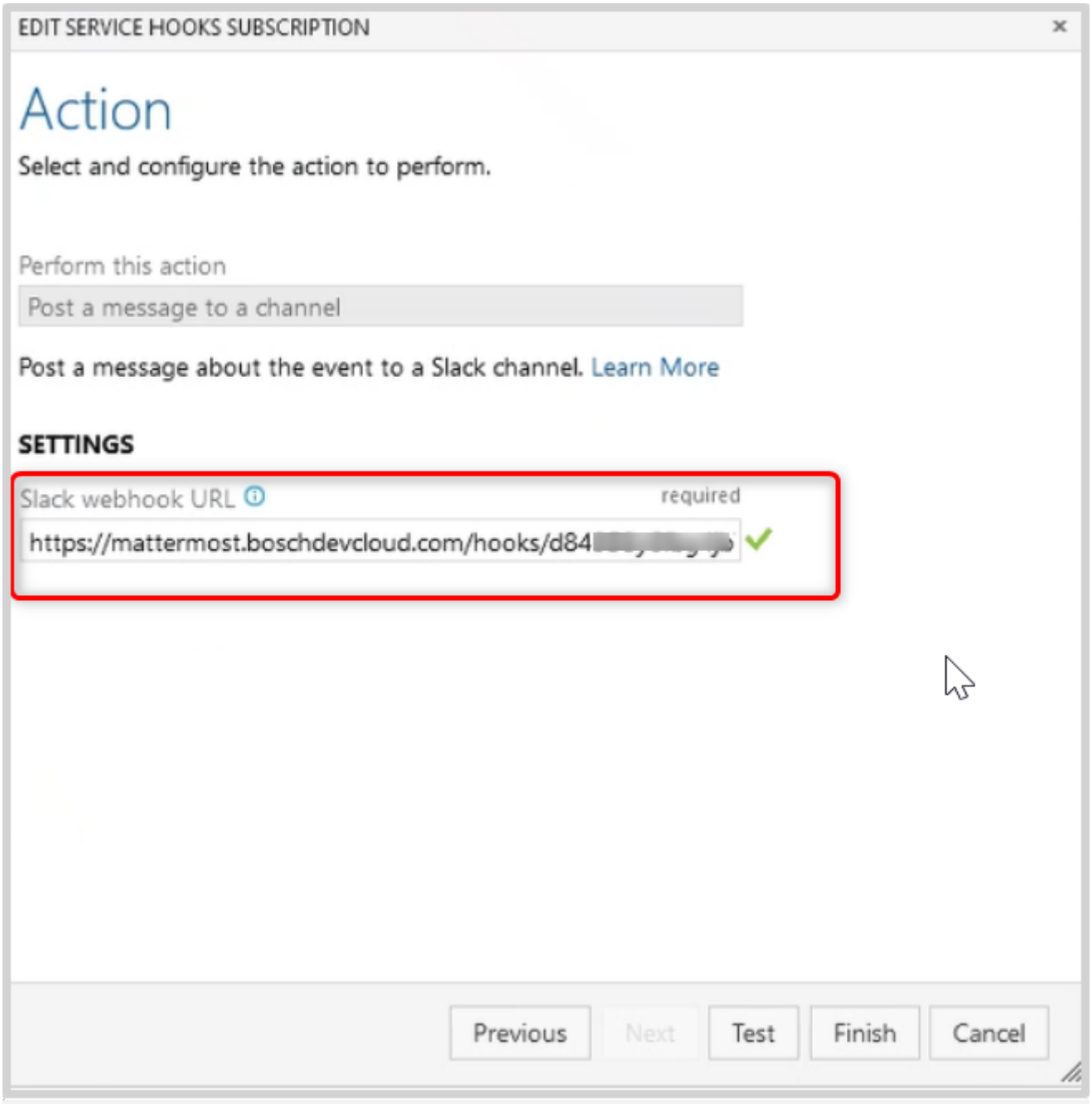
- User needs **Project Administrator** role in Azure DevOps project to add the above generated webhook url. In **Azure DevOps → Project settings → service hook → create subscription**. Please refer the below screenshot and here slack is chosen because their api is compatible with the mattermost,



- Then in the next page, choose the event on which notification has to be triggered. ex: events like Work item or Pull request created in devops and also choose the filters



- Then give the generated mattermost webhook url and click test. Once the testing succeeded, click finish to complete the service hook configuration in Azure DevOps.



6.10. Plugins

SaaS Providers like Atlassian, AzureDevOps Services or GitHub offer marketplaces that include first- or third-party plugins, apps, extensions or add-ons. In many cases, the plugins, apps, extensions or ad-ons can be easily installed. Some of them can be used for free, for others licenses have to be bought. However, before using them, it is important to check, if and under which conditions it is allowed from Bosch side to use these. Therefore, please check the FAQ section of C/IDO who is responsible for the SaaS - Onboarding Process.

To find out which plugins, apps, extensions or add-ons is already onboarded, please search for it in LeanIX.

6.10.1. Cost for Plugins

The costs for Plugins/Apps/Extensions can be found in the marketplace of the SaaS Provider.

6.10.2. Shared responsibility for Plugins

Customers have the possibility to choose if they want to use plugins, apps, extensions or add-ons. If decided to use plugins, apps, extensions or add-ons it is in the responsibility of the customer to make sure that they are compliant with all Bosch regulations (Central Directives, EISA, works council agreements).
The Bosch Development Cloud does not offer product support for plugins, apps, extensions or add-ons or the onboarding process in case it is needed.

6.11. Feedback

We look forward to receiving your feedback, which will help us to improve in the future!

Chapter 7. Cloudspace

7.1. What is Cloudspace

The BDC Cloudspace Service provides you with an own, secure and compliant space in the Microsoft Azure Cloud, where you can explore services and host your development systems.

It offers automated provisioning via API management, Kubernetes clusters, a virtual network, public/internet access to your web services and much more.

Cloudspaces are offered in different tiers:

- In a **shared tier**, where multiple Cloudspaces are located within the same Azure subscription and therefore share the same subscription limits like the maximum amount of storage accounts.
- In a **dedicated tier**, for bigger projects, where a Cloudspace is using a subscription on its own.

Find a direct comparison of the Cloudspace tiers as well as a comparison to the predecessor service BDC-Lab here: Cloudspace Comparison

With Cloudspace we support multiple Azure locations and one Cloudspace can also span multiple locations.
Currently supported locations are:

- West Europe
- West US 3

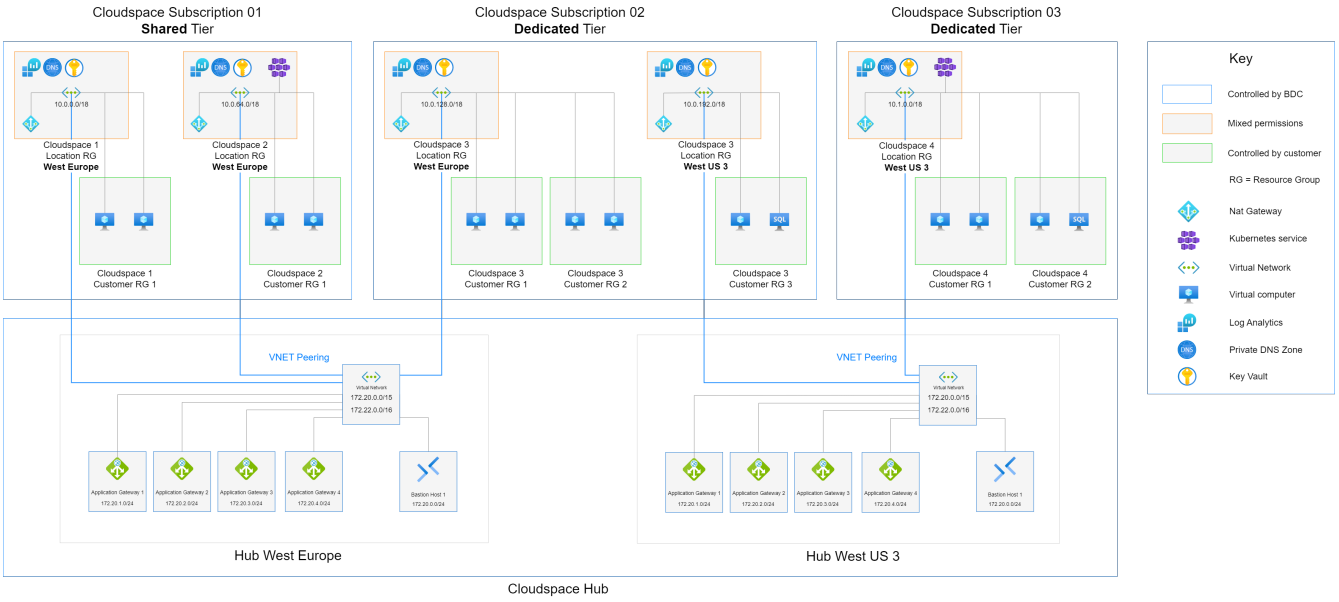
7.1.1. Architecture

The Cloudspace architecture consists of the shared and dedicated subscriptions and a central Cloudspace Hub subscription.

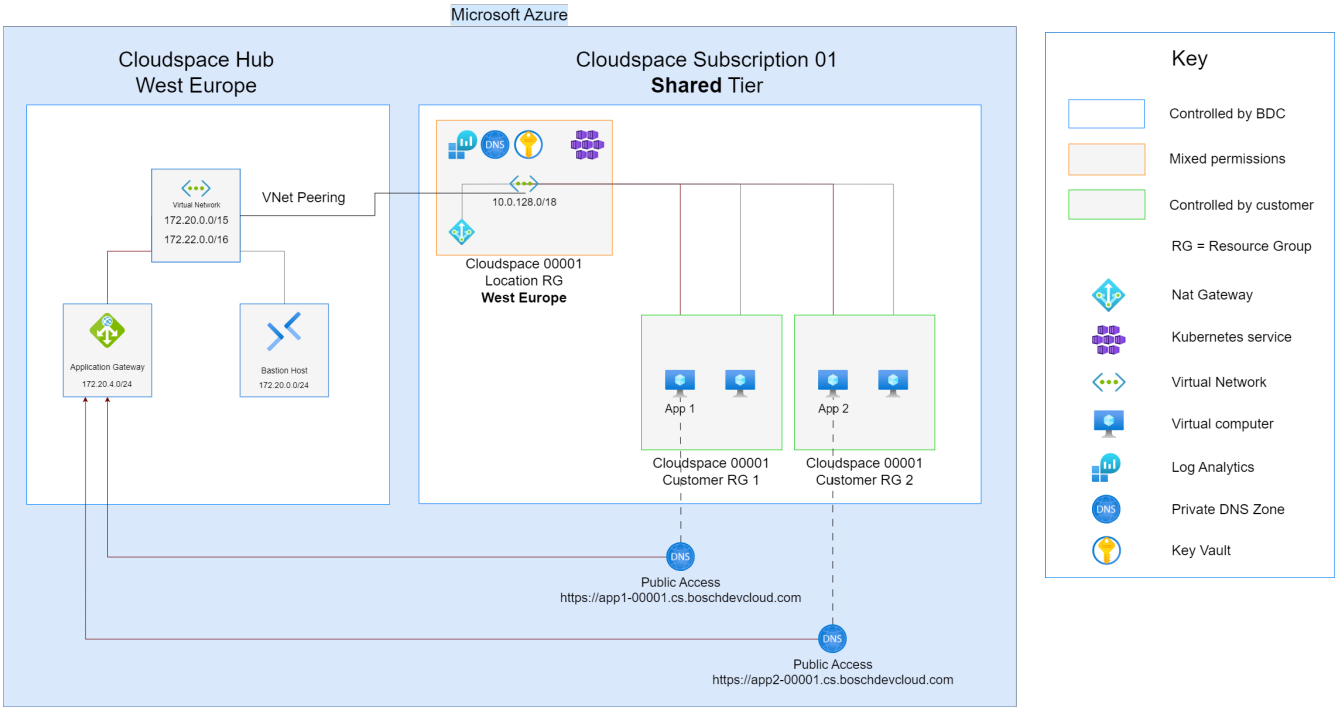
All Cloudspaces are connected with the Cloudspace Hub.

In this Cloudspace Hub, different shared and obligatory services for Cloudspace customers are located. These services are for example the Azure Application Gateways to make web services available through the internet or the Azure Bastion service, to connect to virtual machines.

In the following picture you can see the overall characteristics of the Cloudspace architecture.



Here you can find a detailed example of a single Cloudspace instance and its components within the shared tier:



7.1.2. Terminology

Cloudspace

Cloudspace is commonly abbreviated to "CS"

Cloudspace ID

The identifier of your Cloudspace, for example "00001"

Cloudspace Alias

The alias/name of a Cloudspace that is unique across all Cloudspaces.
If your Cloudspace has an alias, some resources will use the alias instead of the Cloudspace ID (e.g.: Public Access Domains).

The alias needs to follow this Regular Expression `^[a-z0-9]{1,20}$` (only alphanumeric characters and a length of 1-20 characters)
The alias can't be removed or modified after the initial order as of now!

Cloudspace Tier

A Cloudspace can be ordered in different tiers (shared or dedicated)

Cloudspace Hub

A central resource group which contains central services that are required and connected to each Cloudspace. This resource group is managed by the BDC-Team

Cloudspace Location

An Azure location represented by a specific resource group within your Cloudspace

Location Resource Group

A resource group within your Cloudspace that is managed by the BDC-Team, providing central and obligatory services for the specific Cloudspace Location

Customer Resource Group

Resource groups that you can order and that are managed by you

Naming Conventions

IDM Access Right

`IDM2BCD_BDC_CloudSpace<Cloudspace ID>_contributor`
Examples:
`IDM2BCD_BDC_CloudSpace00001_contributor`

Location Resource Group

`cs<Cloudspace ID>-<location geocode>-rg`
Examples:
`cs00001-we-rg`

Location Default VNet

`cs<Cloudspace ID>-<location geocode>-vnet`
Examples:
`cs00001-we-vnet`

Location Log Analytics Workspace

`cs<Cloudspace ID>-<location geocode>-loganalytics`
Examples:
`cs00001-we-loganalytics`

Location NAT Gateway

`cs<Cloudspace ID>-<location geocode>-nat`
Examples:
`cs00001-we-nat`

Location Key Vault

`cs<Cloudspace ID>-<location geocode>-<8 random chars>-kv`
Examples:
`cs00001-we-d0v6cqkf-kv`

AKS Cluster

`cs<Cloudspace ID>-<location geocode>-<AKS Instance ID>-aks`
Examples:
`cs00001-we-01-aks`
`cs00001-we-02-aks`

Public Access Domain

`<hostnamePrefix>-<[Cloudspace ID] XOR [Cloudspace Alias]>.cs.boschdevcloud.com`
Examples:
`myapp-00001.cs.boschdevcloud.com`,
`myapp-mycloudspace.cs.boschdevcloud.com`

Public Access Backend Certificate

`<hostnamePrefix>-<[Cloudspace ID] XOR [Cloudspace Alias]>-sslPem`
Examples:
`myapp-00001-sslPem`,
`myapp-mycloudspace-sslPem`

Customer Resource Group

`cs<Cloudspace ID>-<location geocode>-<custom name>-rg`
Examples:
`cs00001-we-mygroup-rg`

7.2. Getting started

7.2.1. How To Order

Prerequisites

To be able to order a Cloudspace

- you need to have a BDC-ID
- if you would like to order a Cloudspace in the dedicated tier, contact us prior

Order

A Cloudspace can be ordered through the BDC Portal with the Cloudspace API.

- Open the BDC-Portal
- Create a subscription/token for the BDC you want the Cloudspace be associated to
- Navigate to the Cloudspace API
  My BDCs - <Your desired BDC> - Cloudspace

- Open the Create Cloudspace API Call

For the creation of a Cloudspace you need to specify

- an Access Right Owner (Master of Role)
- Costcenter
- Technical Contacts (who we should contact in case of questions/issues with your Cloudspace)
- Alias (optional - if provided, it will be used instead of the Cloudspace ID for some resources)
    - The alias can't be removed or modified after the initial order as of now!
- Location

If the provided data was correct, your request for a Cloudspace will be accepted (Response Code 202) and the deployment will start in the background.

The response will contain the following **important data**:

- The new **Cloudspace ID** needed for further component orders
- **IDM Access Rights (Roles)** you need to apply for, to be able to access the Cloudspace resources in Azure

Check Cloudspace Status

Before you can order further components for your new Cloudspace, you need to wait for the deployment to finish.
To check the deployment status of your Cloudspace, you can use the "Get Cloudspace" API call.
In the result of the API call you will see some "processing" attributes and their values will reflect the current deployment status.
If all "processing" values are `null` and the "status" is `active`, your Cloudspace is ready to use.

7.2.2. Access and Permissions to use CS

Each Cloudspace will get an according IDM Access Right (Role) - `IDM2BCD_BDC_CloudSpace<Cloudspace ID>_contributor`.
As a member of this Access Right, you will get access to the Azure resources of this Cloudspace.
The Cloudspace API can only be used by people having the according BDC Admin Right.

The access to the Azure resources can be enhanced with your own AAD Objects by registering those via the Cloudspace API (Bring Your Own Object - BYOO).
Currently the Cloudspace Resource Group and Cloudspace AKS service support the BYOO feature.

7.2.3. Access to Azure Portal

To access and use your BoschDevCloud lab, please go to Azure Portal and login with your user account and password. Keep in mind to use the username as following:
<username>@bosch.com
You will be redirected to the main page of the Azure portal site.

Azure Cloud Shell (quick start)

If you want to start quickly interacting with Azure resources using the az cli or managing your Kubernetes Cluster with kubectl, but do not want to install any kind of software or struggling with proxy settings, feel free to checkout Azure Cloud Shell, which you will find in the right upper corner of the Azure Portal.

The Cloud Shell provides two different Shell environments, one using Bash and one using Powershell.
Both of them come completely preconfigured and have most of the tools installed you need to interact with Azure resources.
E.g.

- az cli
- kubectl
- helm
- terraform

It even comes with an file editor which you can easily start by typing:

```
1 code .
```

Find the Cloud Shell documentation here

Setup Azure Cloud Shell

When Cloud Shell is opened the first time, it asks for an Azure storage account where files and preferences can be stored.
Please be aware, that you must not have activated firewall settings on your storage account, like allowing access only from specific ip addresses.

> Azure storage firewall is not supported for cloud shell storage accounts.

Also only specific Azure datacenters are supported for Cloud Shell usage, which may be found here.

To get all storage accounts which were created or changed by Azure Cloud shell, filter for tag "ms-resource-usage" and value "azure-cloud-shell".

Command line Interface

Install AZ cli
AZ cli Documentation

7.3. Components and Operations

7.3.1. Cloudspace Location

With Cloudspace we support multiple Azure locations and a Cloudspace can also span multiple locations at once.
Currently supported locations are:

- West Europe
- West US 3

For each Cloudspace Location there will be a central **Location Resource Group** deployed with the naming convention `cs<Cloudspace ID>-<location geocode>-rg` (e.g. `cs00001-we-rg`).
This resource group includes central and obligatory services for a cloudspace location and is managed by the BDC-Team.

The core services deployed with this resource group are:

- Virtual Network
- NAT Gateway
- Log Analytics Workspace
- Key Vault
- Private DNS zone

Since the BDC team is accountable for this resource group, you only have read-only permissions on it as well as some higher privileges on specific resources (like the VNet) to be able to use them.

7.3.2. Network

Every Cloudspace location comes with a virtual network with **16.382** IP addresses (/18) which is part of the address space 10.0.0.0/8.

The default VNet of a Cloudspace Location can be found with the naming convention `cs<Cloudspace ID>-<location geocode>-vnet`.

Every Cloudspace network is connected to the Cloudspace Hub network in the same Azure location via VNet peering.
This peering must not be changed or deleted.

Feel free to create a peering to self created networks in your Customer Resource Group if needed.

Public Access endpoints and Bastion host can only be configured for resources in the default VNet

We have observed issues on the Azure Portal, where the available VNets were not listed/found correctly (especially during VM creation).
Using the CLI and referencing according names works fine though at the same time.

Subnets

A Cloudspace VNet is deployed with following subnets:

Table 1. Subnets example

Subnet Name	Address space	Last IP
zBdcAksSubnet01	10.1.192.0/24	10.1.192.255
zBdcAksSubnet02	10.1.193.0/24	10.1.193.255
zBdcReserved01	10.1.194.0/23	10.1.195.255
zBdcReserved02	10.1.196.0/22	10.1.199.255
DefaultSubnet	10.1.200.0/24	10.1.200.255

All subnets with the suffix "zBDC" are used by the BDC Team and its services and must not be changed.

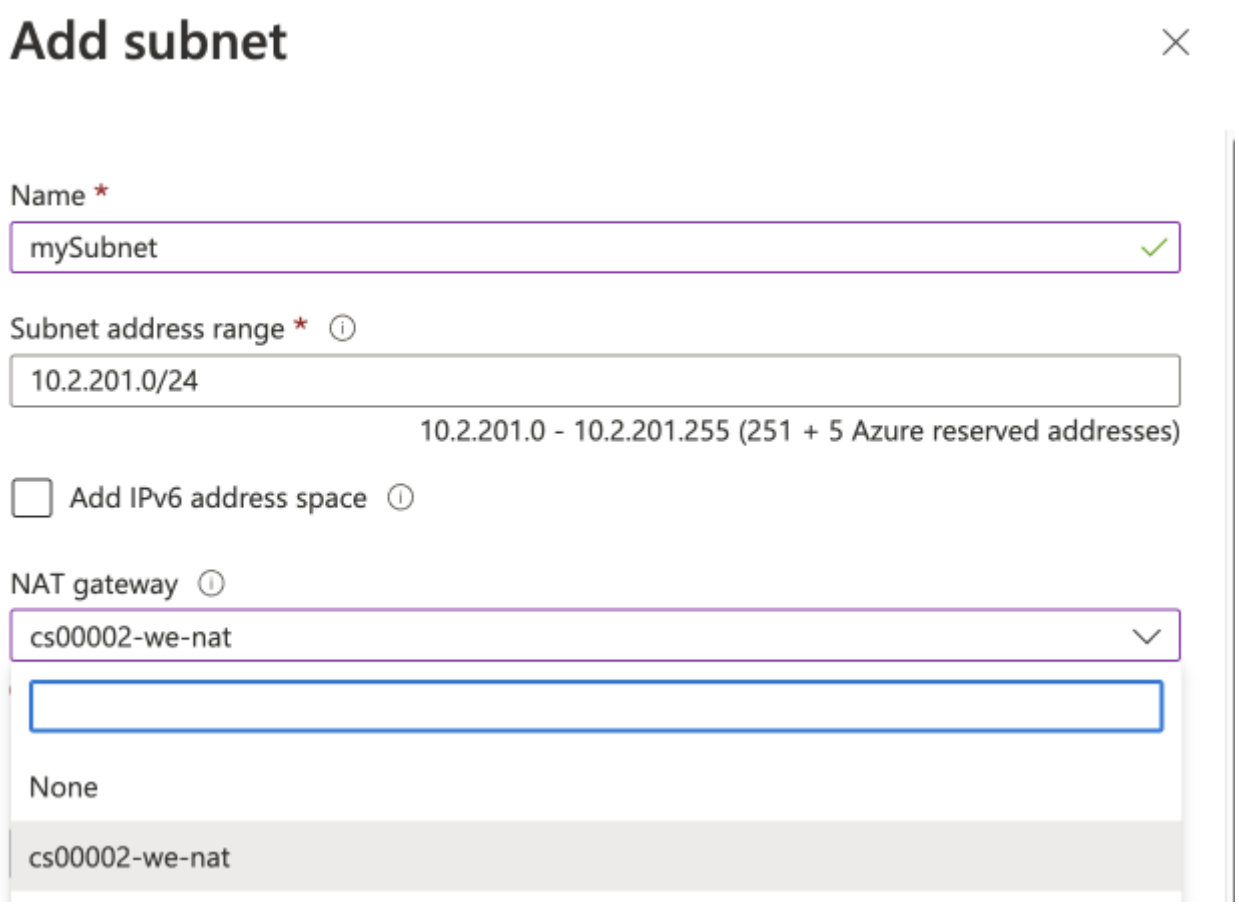
The **DefaultSubnet** is the entry point for all users, who don't want to deal with network related topics and enables them to directly deploy resources like virtual machines in it.
Settings like "Service Endpoints" of this subnet may be adapted to personal needs, but a deletion of this subnet is not possible.

Apart from these pre-configured subnets, the creation of any number of subnets, as well as changing or deleting them is possible.

NAT Gateway

In each Cloudspace, a NAT Gateway is used to make it possible to whitelist all systems coming from a Cloudspace e.g. in a foreign SaaS offering.
All pre-deployed subnets are configured to use this Gateway, in order to send all outgoing traffic through one static public IP.

Self-created subnets must be configured accordingly to use this gateway.



Private DNS

A private DNS zone resolves domain names in a VNet without having to add a customized DNS solution.

Each Cloudspace Location comes with a predefined private DNS zone "cs.boschdevcloud.com". There you can define and manage your own DNS records. The records contained in a private DNS zone can't be resolved from the internet.
The private DNS zone is linked to the default VNet of the location.

To make resources resolvable within a VNet, add the hostname to a private DNS zone.
For example, if you have public access and want to make it internally reachable, add the hostname prefix of your public access to the private DNS zone of the location resource group.

If you want to create your own private DNS zone with your custom name, you can do this in your own resource group.

Public IPs

The usage of public IPs is prohibited and the creation is denied by Azure Policies.

FAQs

How can I connect to BCN?

Please use Bastion or public access service to transfer data, as we can't provide a default tunnel.

▼ *Click to reveal the full answer...*

> Since we are unable to provide you a default tunnel for connections to BCN, you must use the Bastion or Public Access service to access the cloud data you wish to transfer to/from BCN.

How can I connect to client license servers?

BD/PLS4 offers license hosting services in Bosch. Please get in touch with the license team to discuss your request.

▼ *Click to reveal the full answer...*

> The BDC doesn't host any license servers in our network. There are internet exposed license servers which can be used from Cloudspace but there are limitations (license conditions, ...). Get in touch with the license team via: http://rb-cae.de.bosch.com/ServiceRegistration/?LicenseHosting to evaluate your request.

Can I use VNet peering to connect to other VNets?

VNet peering is allowed within your Cloudspace. For migration purposes it is also temporarily allowed to peer your Cloudspace and Lab(s).

▼ *Click to reveal the full answer...*

> In general, VNet peering is allowed within your Cloudspace, for example, to peer Location and Customer networks. It is also temporarily allowed to peer your Lab and Cloudspace networks.
> You are NOT allowed to peer networks of any other Azure subscription outside of the BDC. We monitor these violations and reserve the right to delete them.

7.3.3. Key Vault

The Key Vault within the Location Resource Group is used to provide managed secrets that are meant for your use - e.g.: Internal certificates for your Public Access.

You only have read permission on this Key Vault.

If you want to use a Key Vault for your own secrets, you can do so by creating a Customer Resource Group and deploying your own Key Vault to it.

7.3.4. Log Analytics

Every Cloudspace will be deployed with a Log Analytics Workspace (LAW) that is used to collect monitoring data of all resources deployed within the whole Cloudspace.

As this LAW is also used in case of using a managed AKS, the rights on this LAW are restricted to avoid deletion of deployed components.
There is a custom role (bdc-<env>-loganalytics-user-role) assigned to the user group that allows the creation and modification of alerts, scheduled searches and views.

7.3.5. Customer Resource Groups

Every Cloudspace comes with a Location Resource Group, where BDC-managed resources like the VNet are in (see also chapter Cloudspace Location).
To create own resources like virtual machines or others, you need to order a Customer Resource Group. You can order Customer Resource Groups via the Cloudspace API.
To delete Customer Resource Groups you need to use the "Delete a resource group" API. Manually deleted Customer Resource Groups are not permanently deleted and will be restored after an update.

Customer Resource Groups in Cloudspace are created in following convention:
`cs<Cloudspace ID>-<location geocode>-<custom name>-rg` (eg. `cs00001-we-mygroup-rg`)

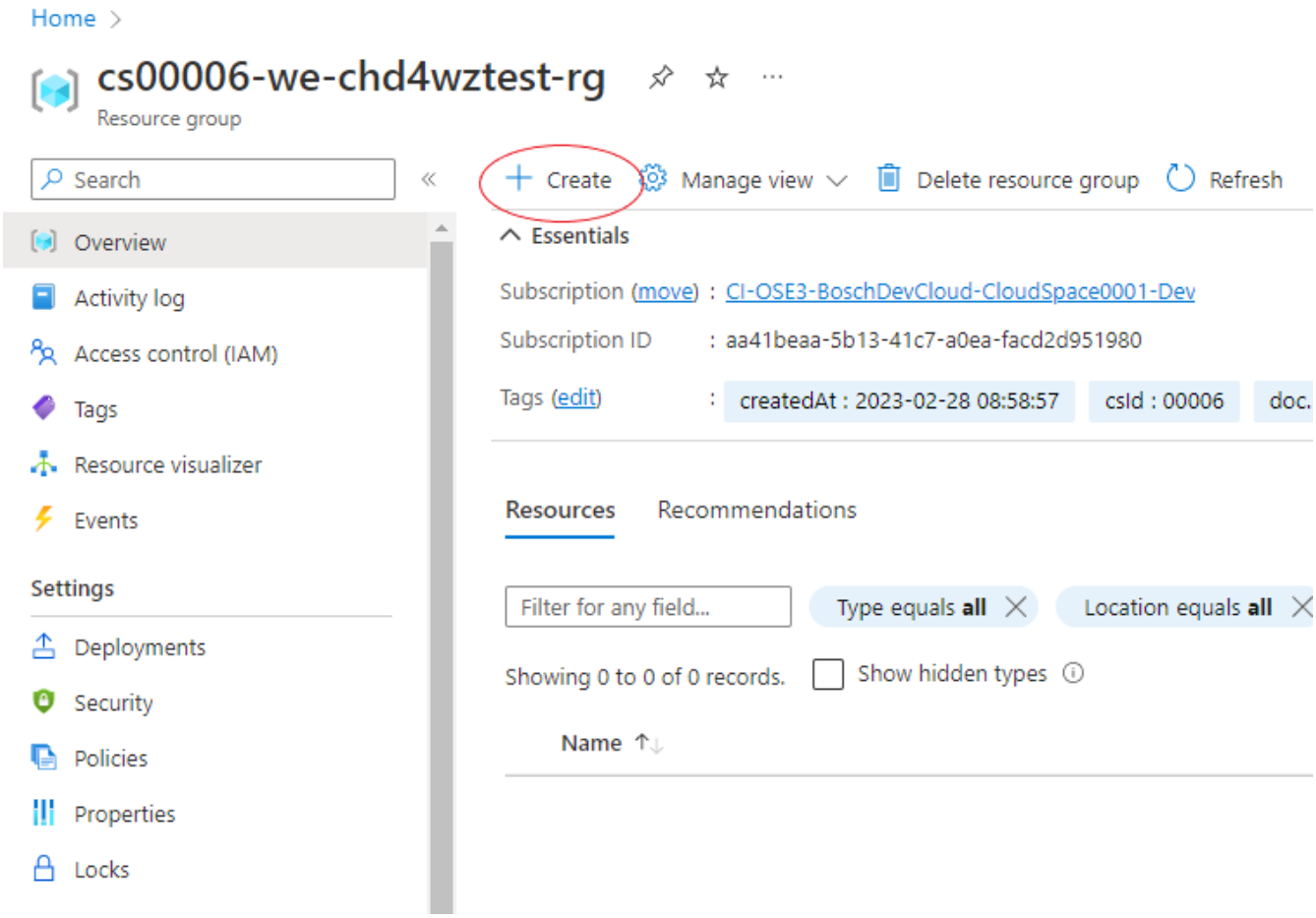
Each user with the Cloudspace IDM Access Right (`IDM2BDC_BDC_Cloudspace<cloudspaceId>_contributor`) has contributor rights on these created resource groups.

Resources like virtual machines in these resource groups can be connected with the Cloudspace Location VNet (needed if access via Bastion is necessary).

How to create a virtual machine

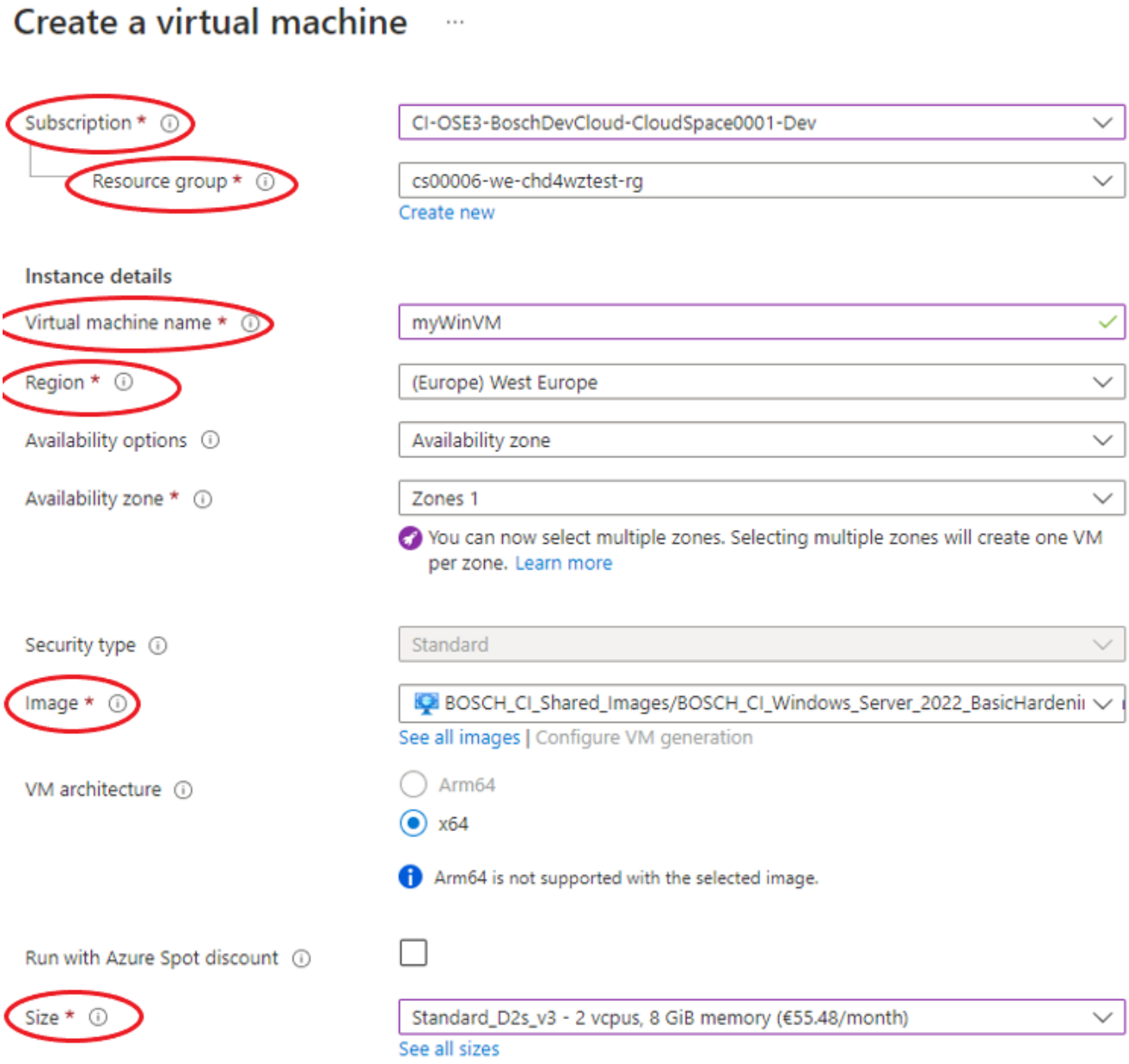
Virtual machines can be created in your **Customer Resource Group** (eg `cs00001-we-mygroup-rg`).

From the Azure portal, navigate to your Customer Resource Group and click on the "+ Create" button on top of the page.



A window pops up, which shows you resources that can be created from the Azure Market Place. In the search box type "Virtual Machine" and after selecting it "Create".

The "Create Virtual Machine" page opens as shown in the following picture:

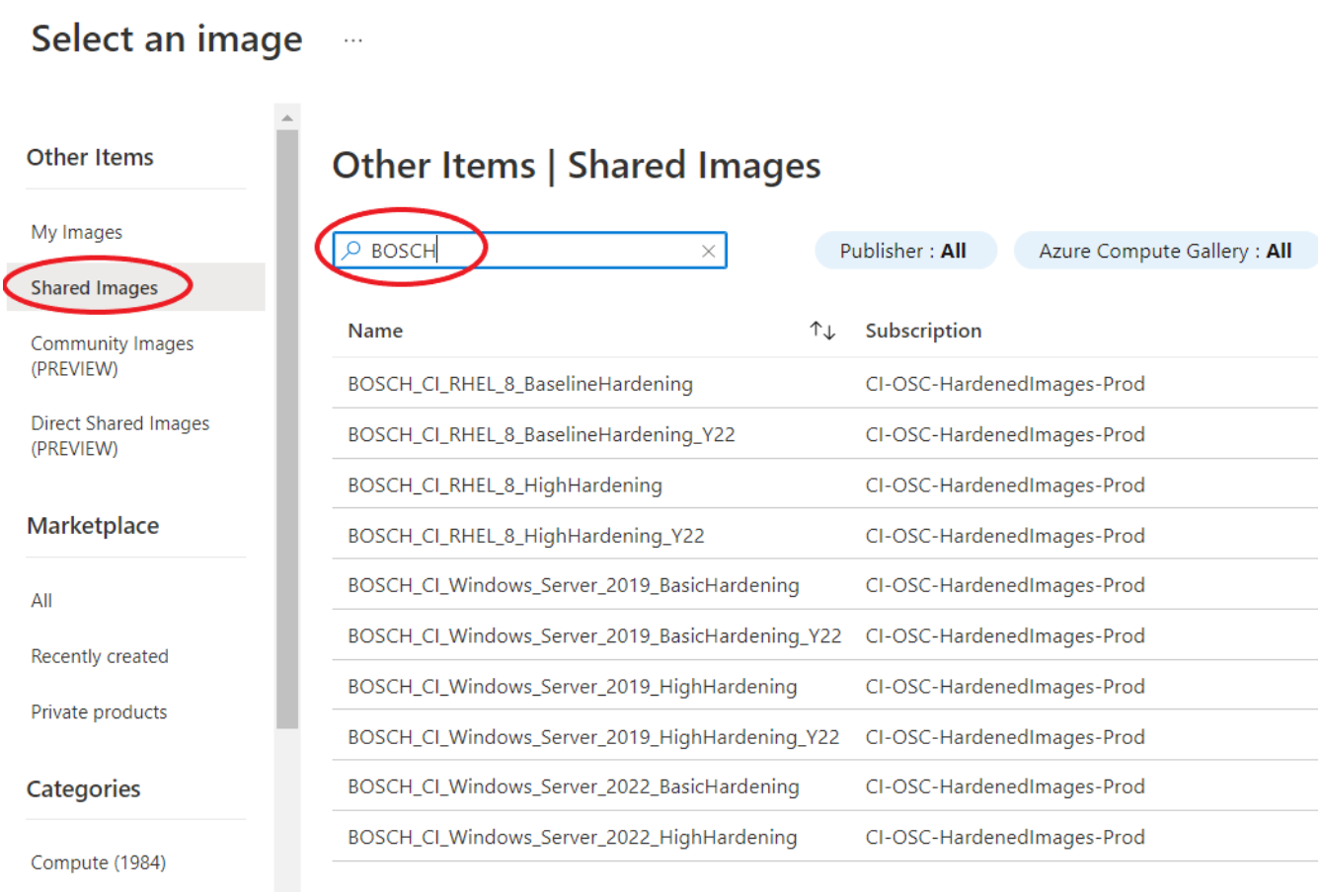


Values that should already be there:

- Subscription
  Subscription under which your cloudspace and customer resource group were created
- Resource group
  Correct Customer Resource Group (eg `cs00001-we-mygroup-rg`)

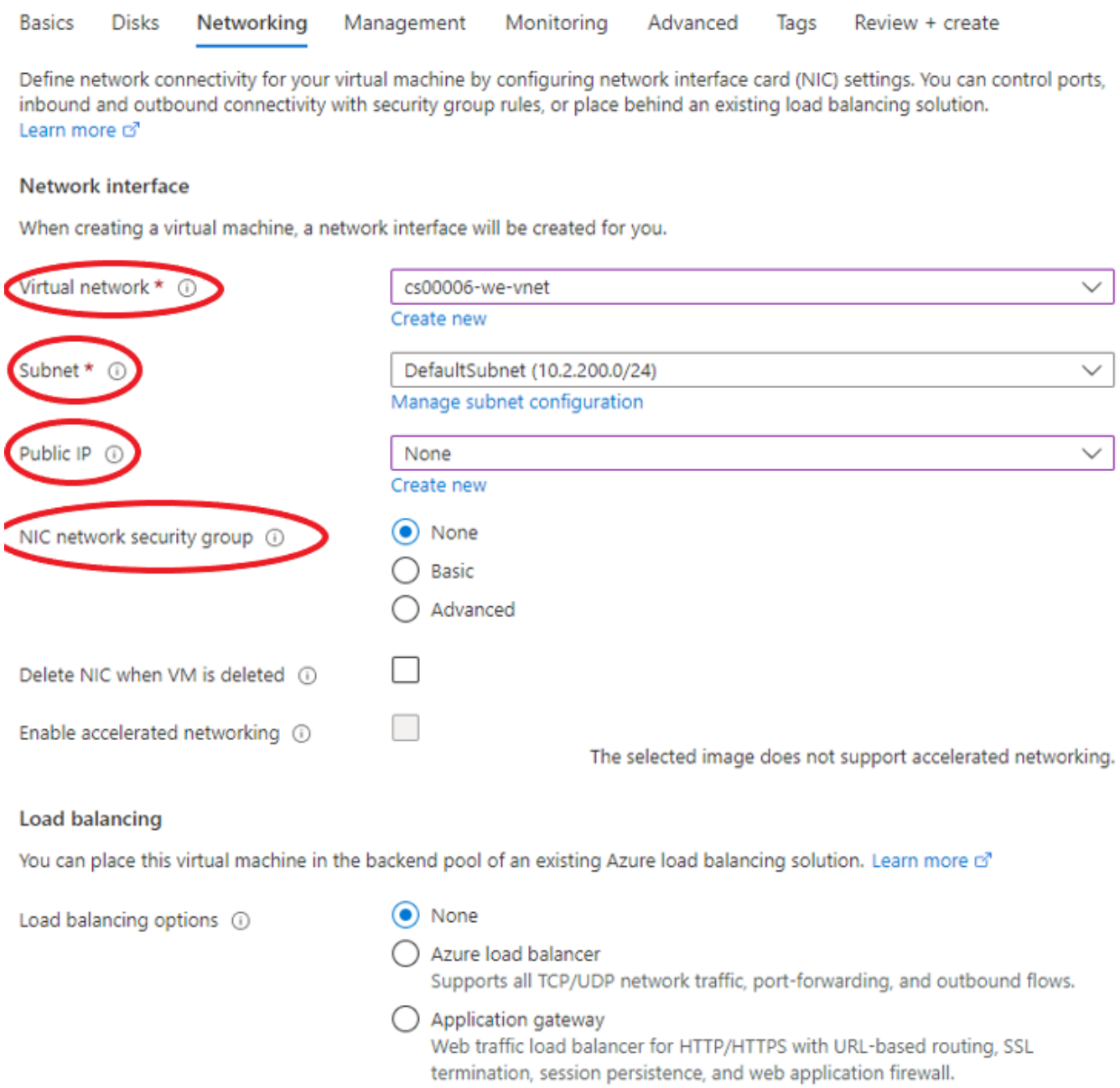
Enter values for the following parameters:

- Virtual machine name
- Image
  Users are free to choose fron any available images provided by Azure. Keep in mind that Bosch already provides hardened images for both Red Hat and Windows Server and they are recommended to use.



For additional information please check:
Red Hat Linux Server hardened images in AZURE
Windows Server hardened images in AZURE

- Size
  Users are free to choose from any available VM size that Azure provides. There is a huge variety of different VM types and sizes. The size of a VM can be changed at any time (just requires a reboot). Select the VM and re-size it.
- For Windows set "Username" and "Password" and for Linux choose "SSH public key"
- Select "None" for Public Inbound ports
- Chose License Type
- navigate to next page "Disk"
- navigate to next page "Networking"



- Choose Virtual network from your **Resource Group**
- for the subnet, choose **DefaulfSubnet** or create your own subnet
- select "**None**" for Public IP and the NIC network security group
- click on **"Review and Create"**
- review the values and finally click on **"Create"**

To connect to your new VM, navigate to this VM from the portal and connect using Bastion host with the credentials specified during VM creation.

You may also want to install some useful SW packages on your VM. Here is a small HowTo install some applications on a Windows Server VM. To install an application open Powershell, execute the commands below and then restart the Powershell.

- Chocolatey

```
Set-ExecutionPolicy Bypass -Scope Process -Force;
```

```
[System.Net.ServicePointManager]::SecurityProtocol                                    =
[System.Net.ServicePointManager]::SecurityProtocol    -bor    3072;    iex    ((New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```

- Enable the choco feature allowGlobalConfirmation
  `choco feature enable -n allowGlobalConfirmation`
- Azure CLI
  `choco install azure-cli`
- VSCode
  `choco install visualstudiocode`
- Firefox
  `choco install firefox`
- kubectl
  `az aks install-cli`
  In case of "CERTIFICATE_VERIFY_FAILED" please check FAQ
- notepad
  `choco install notepadplusplus`
- Git
  `choco install git`

A helpful price calculation tool for your VM might be found here: https://azureprice.net/?currency=EUR&region=westeurope&timeoption=month

Filetransfer into VMs

If you would like to copy data from your computer into your Azure VM, you can use Bastion for it.

There are some things to keep in mind:

- VNet: the default Location VNet should be selected (`cs<Cloudspace ID>-<location geocode>-vnet`, e.g. for access via Bastion)

> We have observed issues on the Azure Portal, where the available vnets were not listed/found correctly (especially during VM creation).
> Using the CLI and referencing according names works fine though at the same time.

- Subnet: can be created by yourself or the DefaultSubnet can be used
- Public IP: since the usage of public IPs is prohibited through Azure policies, "None" must be selected
- All other settings can be adjusted as needed

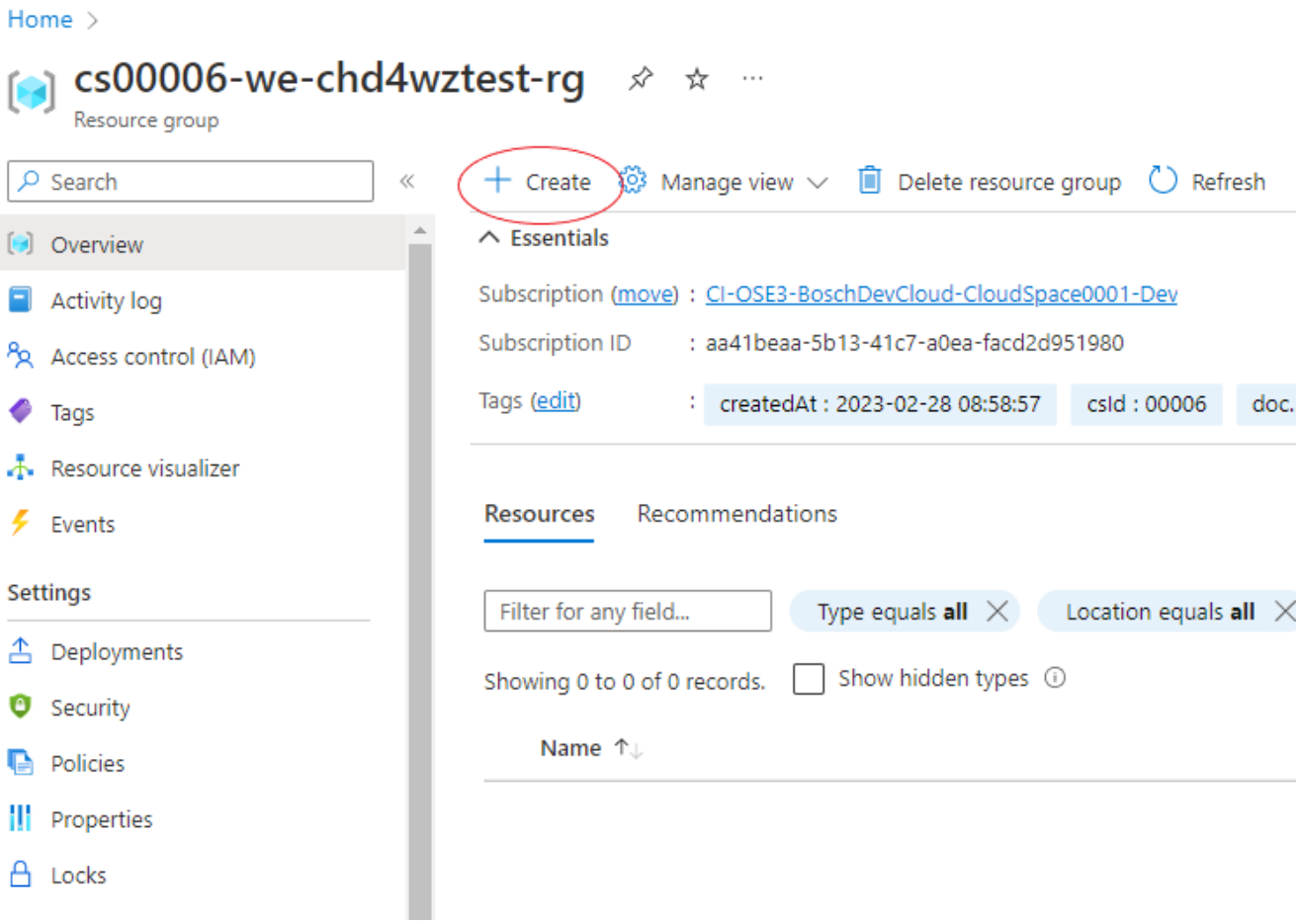
Storage Account

Azure storage account - is a resource to store different objects including data objects, file shares, queues, tables, disks etc.

If you want to use a Storage Account for your own objects, you can do so by creating a Customer Resource Group and deploying your own Storage Account to it.

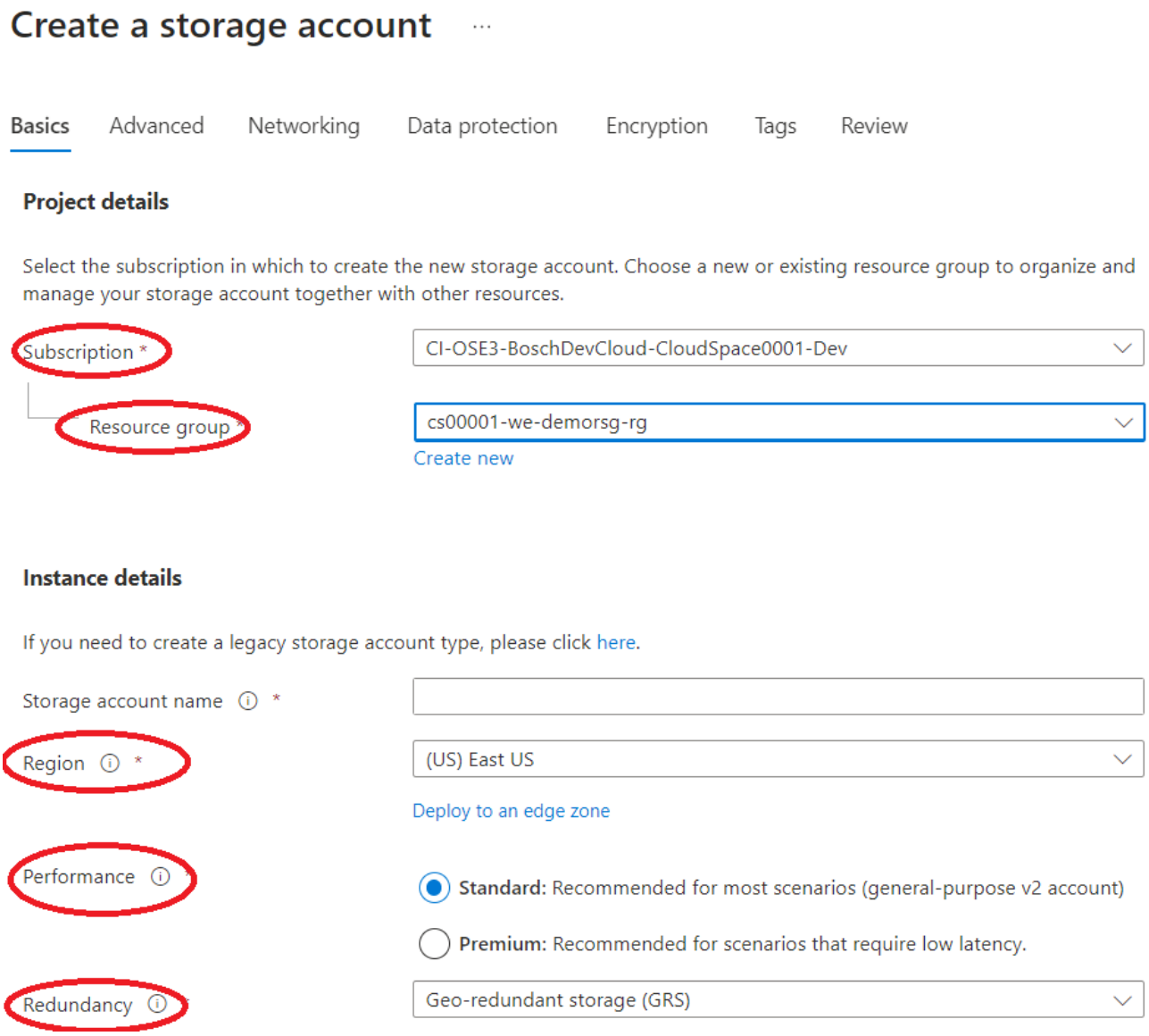
How to create storage account

From the Azure portal, navigate to your Customer Resource Group and click on the "+ Create" button on top of the page.



A window pops up, which shows you resources that can be created from the Azure Market Place. In the search box type "Storage account" and after selecting it "Create".

The "Create Storage account" page opens as shown in the following picture:



Values that should already be there:

- Subscription
  Subscription under which your cloudspace and customer resource group were created
- Resource group
  Correct Customer Resource Group (eg cs00001-we-demorsg-rg)

Enter values for the following parameters:

- Storage Accounts name
- Region
  Keep in mind that East US and West Europe region are recommended to use.
- Performance
  Standard version of storage is recommended to use.
- Redundancy
  If your SA exists for testing purpose only - LRS will be great to use, If you need to keep your data save use GRS.

How to keep SA compliant

- Secure transfer to storage accounts should be enabled
- Storage account public access should be disallowed
- Storage accounts should be migrated to new Azure Resource Manager resources
- Storage accounts should use customer-managed key for encryption
- Storage accounts should restrict network access using virtual network rules
- Storage accounts should use private link
- Storage accounts should restrict network access

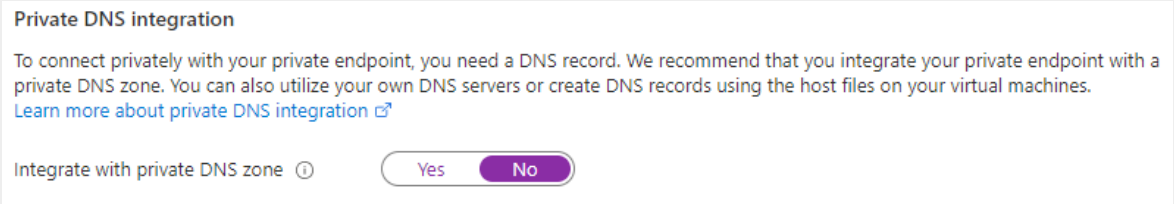
FAQs

How to connect privately with your Private Endpoint?

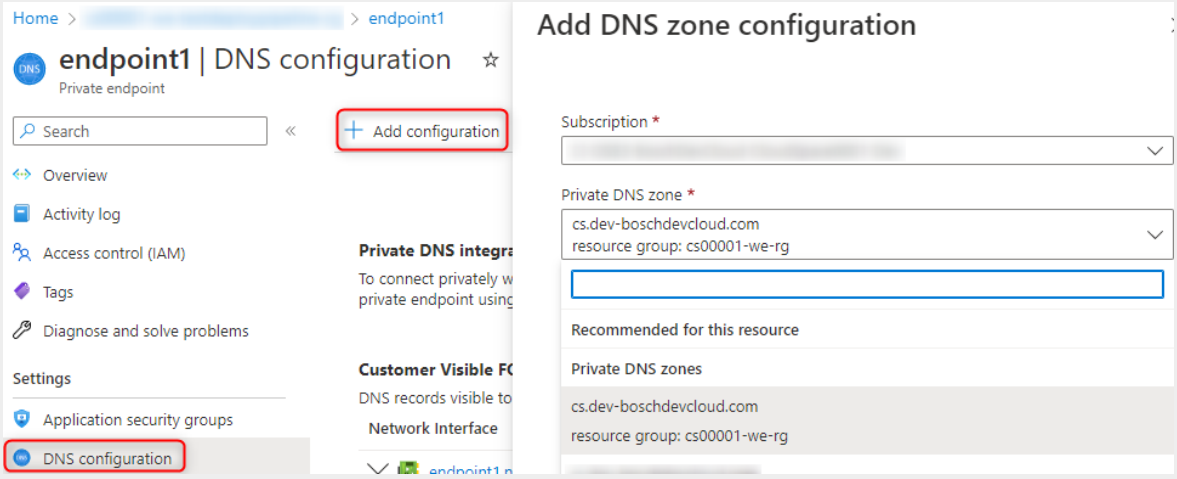
In order to establish a private connection with your private endpoint, you need to manually integrate your private endpoint into a private DNS zone.

▼ *Click to reveal the full answer...*

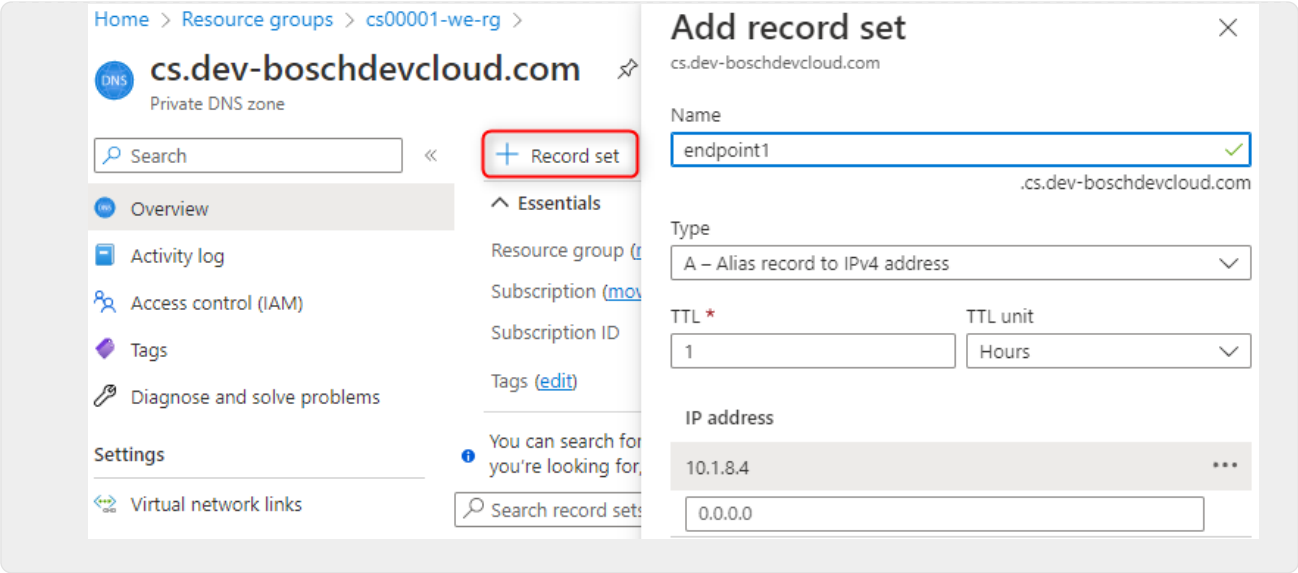
1. Deploy your resource without public network access and without private DNS integration. A private endpoint will be deployed. With an existing resource change the public network access to "Disabled" and create a private endpoint on your own.



2. Go to the private endpoint and select DNS configuration. Add a configuration and select the private DNS zone "cs.boschdevcloud.com".



3. To complete the DNS configuration you must manually add a DNS record for your private endpoint in the private DNS zone. Go to the location resource group, select "cs.boschdevcloud.com" and add a record set with the name of the private endpoint and its IP address.

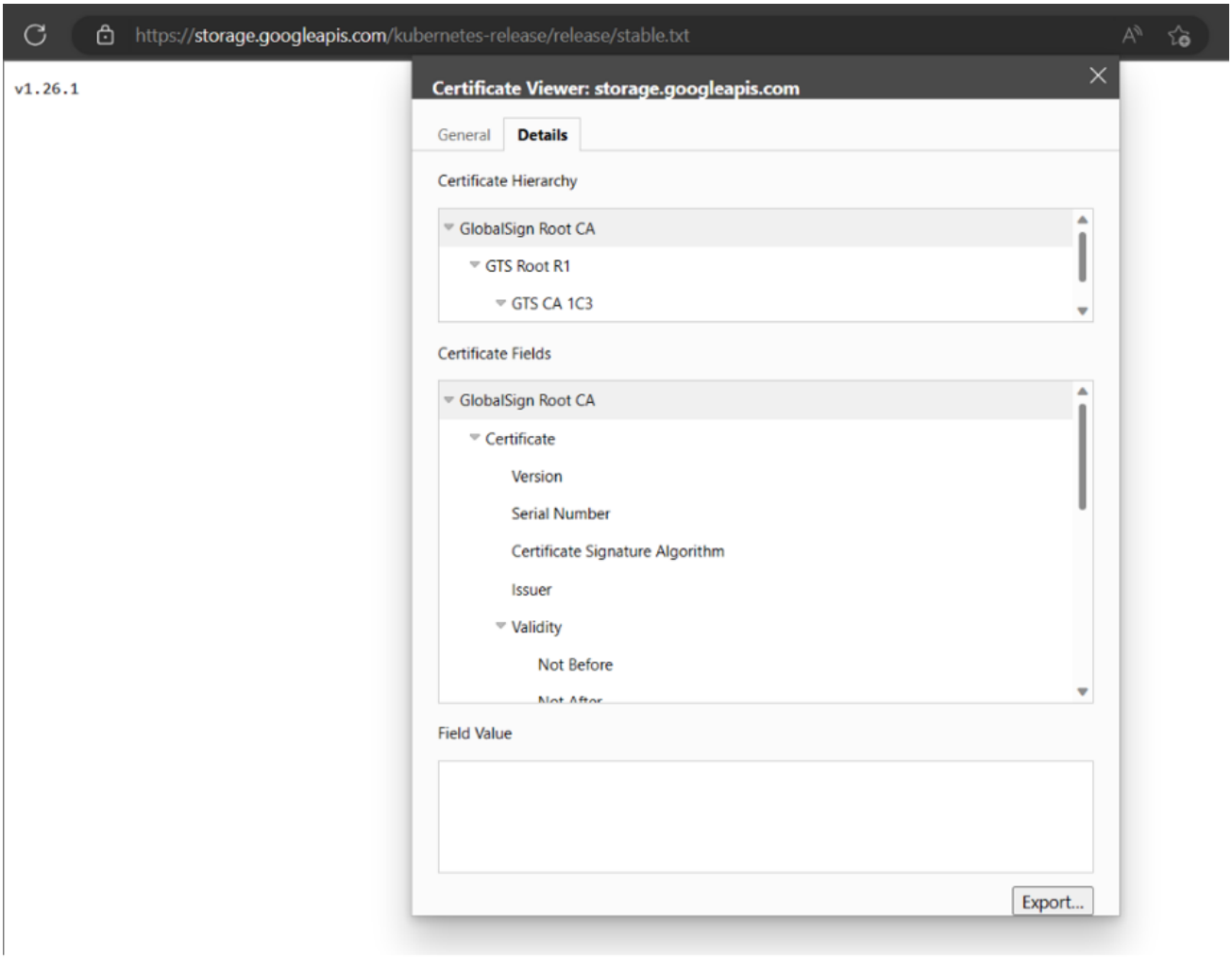


Since Microsoft allows adding a private endpoint to a private DNS zone only with a special naming scheme, it is mandatory to configure the private DNS zone manually and not during the creation of the private endpoint.

CERTIFICATE_VERIFY_FAILED error while installing kubectl

▼ *Click to reveal the full answer...*

While running from Windows powershell az aks install-cli command **CERTIFICATE_VERIFY_FAILED** error might occur for some Azure images. The reason for the exception to appear seems to be the lack of GlobalSign Root CA. Solution is browsing following url in Microsoft Edge and checking certificate.



After browsing the above url with Edge browser, no error occurs.

7.3.6. AKS

It is possible to deploy two independent Azure Kubernetes Services (AKS) per Cloudspace Location. These clusters are identified by the AKS instance ID (01 or 02). The current setup uses kubenet networking and has a preconfigured VM size (Standard_D2ds_v5) with Ephemeral OS disks for the default node pool.
It's planned to provide some more configuration options in the future (e.g. different network setup to support Windows nodes)
We no longer provide an ingress load balancer. You are now free to deploy a load balancer of your choice.
Also Role-based Access Control (RBAC) is enabled by default.

AKS Permissions

The following configuration rights are granted to users in the Azure portal for the AKS

- New node pools can be created with no restriction on VM size
- Node pools can be created with different sized VMs
- Autoscaling option can be used for all node pools
- Start and stop the cluster

- Update AKS cluster by yourself via API
- Read permissions are granted to the Infrastructure Resource Group in which the Scaleset is located.

Upgrade your AKS

You are responsible for updating your AKS cluster yourself. If you want to avoid this, you can use the AKS auto-upgrade feature.
We reserve the right to discontinue support if you are using an older version that is no longer supported by Azure. Here you can find the supported AKS versions.
Information on how to perform an AKS upgrade can be found here.

AKS auto-upgrade

AKS auto-upgrade provides automation to keep your clusters up to date and relieves you of the responsibility to regularly update your clusters.
You can choose between auto-upgrade options while creating your AKS cluster. However, the default value is "stable", but you can change it to one of the AKS auto-upgrade channels listed below.

You can choose between the following upgrade channels:

- none:
  Automatic upgrades are not applied. Upgrades must be initiated manually.
- stable:
  Upgrades the cluster to the latest patch version of the N-1 generally available minor version.
  For example, if your cluster runs on version 1.25.6 and versions 1.26.0, 1.26.3 are available, the cluster upgrades to 1.26.0.
- rapid:
  Updates the cluster to the latest patch version of the latest generally available minor version.
  For example, if your cluster runs on 1.25.6 and versions 1.26.0, 1.26.3 are available, the cluster upgrades to 1.26.3
- patch:
  Updates the cluster to the latest patch version within the specified minor version.
  For example, if you use 1.26.0 it will updates to the latest patch 1.26.3, but won't upgrade to a newer minor version.
- node-image:
  Updates the node operating systems to the latest available image but does not automatically updates the Kubernetes version. Microsoft provides patches and new images for image nodes weekly. Automatically updates your node images whenever a new version is available.

If you want to use AKS auto-upgrade these limitations apply:

- Be aware that there is currently no function to plan AKS upgrades! Upgrades will be applied when new updates are available. This can cause downtime at inconvenient times.
- If you are using node-image, your AKS cluster version (Kubernetes) won't get upgraded automatically. You need to take care about that. Information on how to upgrade your AKS

cluster can be found here.

- There is no option to upgrade the control plane first. Auto-upgrade always upgrades the control plane and the node pools together.
- If you are using node-image, Linux unattended upgrades is disabled by default.

To change AKS auto-upgrade on existing clusters use the Cloudspace API - AKS - Update customer AKS in the BDC portal.

Node OS image updates

To upgrade your node OS image, you can either set your AKS auto-upgrade channel to "node-image" or you can do it manually via the Azure portal. The node OS image is also automatically updated when an AKS version upgrade is performed.
Node OS auto-upgrade functionality only applies if you set your AKS auto-upgrade channel to "node-image". The node OS auto-upgrade will then automatically be set to "NodeImage" and will periodically update your node OS image. We don't currently provide a function to select a different channel for the node OS auto-upgrade feature.

Windows Nodes

To use Windows Nodes, you need to deploy a cluster with Azure CNI Overlay network plugin configured.
Use Cloudspace API - AKS - Create customer AKS to deploy a new cluster with "azure" as network plugin parameter. After deployment, you can create Windows Nodes via CLI or Azure Portal.

This is for new clusters only. There is currently no option to upgrade existing clusters with network type "kubenet".

To check what network type your cluster is using, there are several ways:

- Use Cloudspace API - AKS - Get customer AKS: in the response you can see your network profile.
- In the Azure Portal, on the overview of your cluster or on the Networking tab.
- Via CLI use "az aks list": in the network profile section you can find the networkPlugin and networkPluginMode parameters.

Additional information about Azure CNI Overlay can be found here.

Login to the cluster

As prerequisite to connect to the cluster and deploy workloads you need to install kubelogin and kubeclt. This can be done using Azure CLI:

Listing 1. install kubelogin and kubeclt

```
1 az aks install-cli
```

After this, it should be possible to connect to the cluster as follows:

Listing 2. connect to aks

```
az aks get-credentials --resource-group cs<cloudspaceId>-<location>-rg --name
cs<cloudspaceId>-<location>-<AKS Instance ID>-aks --subscription CI-OSE3-
BoschDevCloud-Cloudspace<number>-Prod

# example for cloudspace 00001, location westeurope and instance ID 01:
az aks get-credentials --resource-group cs00001-we-rg --name cs00001-we-01-aks
--subscription CI-OSE3-BoschDevCloud-Cloudspace0001-Prod
```

Logs

The cluster is configured to use a Log Analytics workspace to collect all logs (see Log Analytics). You can find the workspace in the same Location Resource Group as the AKS.

Listing 3. Kusto query to get container logs

```
ContainerLog
| where TimeGenerated < now()
| where TimeGenerated >= startofday(ago(1d))
| extend ClusterName = tostring(split(_ResourceId, "/")[-1])
| join kind = inner (
    KubePodInventory
    | project ContainerID, PodName=Name, ControllerKind, ControllerName, Namespace
    | distinct *
    )
    on ContainerID
| project TimeGenerated, ClusterName, PodName, LogEntry
```

Traefik Deployment with Certificate

This section serves as an example and is not provided by default!

Prerequisites:

- You have **NOT** ordered Public Access
- You are connected to your AKS cluster via CLI (you can also use the portal)
- You have a service/application/workload you want to publish
- You have openSSL installed on your client

Follow the steps below to create a Traefik load balancer, order Public Access and access your service via https.

1. Deploy Traefik load balancer via Helm with a dynamic configuration
   The following attributes must be part of the dynamic configuration:

   ```
   service:
     annotations:
   ```

```
3     service.beta.kubernetes.io/azure-load-balancer-internal: "true"
```

▼ *Example dynamic configuration file*

```
globalArguments: ①
  - --global.checknewversion=false
  - --global.sendanonymoususage=false

resources: ②
  limits:
    cpu: '2'
    memory: 400Mi
  requests:
    cpu: 200m
    memory: 40Mi

service:
  annotations:
    service.beta.kubernetes.io/azure-load-balancer-internal: "true"

rbac:
  enabled: true

ports:
  websecure:
    tls:
      enabled: true
```

① Some global arguments that are helpful. Can be adapted or deleted if not needed.

② Resources can be adjusted as needed.

2. Install Traefik with Helm

```
helm repo add traefik https://traefik.github.io/charts
helm repo update
helm install traefik traefik/traefik -f traefik.yaml
```

3. Order Public Access via BDC portal
   Now you can order Public Access via the BDC portal with the "Register Cloudspace public access in location" API. You need to choose a hostname prefix and check the external IP address of your Traefik to define the destination IP for your backend service. Also define a health probe path where your application is accessible and responds with an http status code. You can also specify which http status code the health probe should consider healthy.
   If you have already ordered Public Access, you can change the attributes with the "Change Cloudspace public access settings in location" API.
4. Get the certificate from the Location KeyVault

```
az keyvault certificate list --vault-name <KeyVault Name>
az keyvault secret download --vault-name <KeyVault Name> --name <Cert Name>-sslPem --file cert_from_kv.pem

#Example
az keyvault secret download --vault-name cs00001-we-abcd1234-kv --name test-00001-sslPem --file cert_from_kv.pem
```

You can also navigate to the Location KeyVault in the portal and download the certificate manually. Please note different names and file formats.

5. Export the private key from the certificate with openSSL

```
openssl rsa -in cert_from_kv.pem -out private.key
```

6. Create a secret on the AKS cluster
   Upload the certificate and private key from your client to the AKS cluster. If you downloaded the certificate manually, you may need to adjust the file paths. If you have created a custom namespace for your application, you must also create the secret in that namespace.

```
kubectl create secret generic traefik-cert --from-file=tls.crt=cert_from_kv.pem --from-file=tls.key=private.key --namespace <example>
```

7. Deploy an Ingress
   Deploy an Ingress with the following additional attributes. Also consider your custom namespace when deploying the ingress.

```
tls:
- hosts:
    - <hostname>.cs.boschdevcloud.com
  secretName: traefik-cert

#Example
tls:
- hosts:
    - test-00001.cs.dev-boschdevcloud.com
  secretName: traefik-cert
```

▼ *Full Ingress YAML*

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: tls-example-ingress
spec:
  tls:
```

```
  - hosts:
      - test-00001.cs.dev-boschdevcloud.com # Adjust this attribute
    secretName: traefik-cert
  rules:
  - host: test-00001.cs.dev-boschdevcloud.com # Adjust this attribute
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: <your-service> # Adjust this attribute
            port:
              number: 80
```

```
kubectl apply -f ingress.yaml --namespace <example>
```

8. Enter the URL into your browser and check if it is accessible.

FAQs

How to mount a `Azure Disk` which is located in my customer resource group?

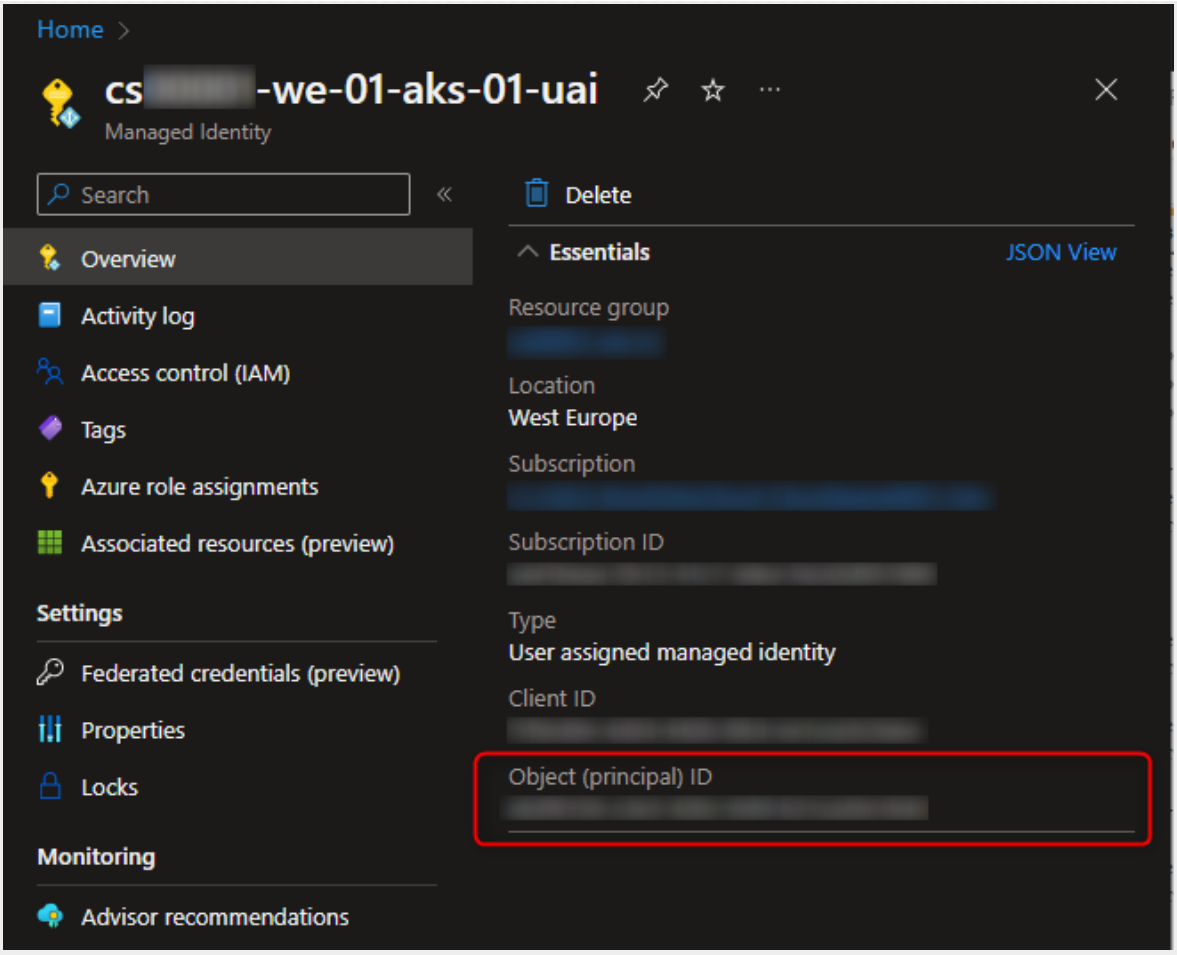
To mount such an Azure Disk you need to grant the AKS User assigned identity permission to access the Azure Disk.

▼ *Click to reveal the full answer...*

1. Get the `Object (principal) ID` of your Cloudspace AKS
   Via Azure CLI - az identity show or Azure Portal.

   Listing 4. example - Azure CLI - az identity show

   ```
   az identity show `
       --name "cs00001-we-01-aks-01-uai" `
       --resource-group "cs00001-we-rg" `
       --query [].principalId `
       --output tsv
   ```



2. Add the Object ID to your Resource Group Use the Cloudspace - Resource group - Bring your own AAD Object API to add the Object ID to your Resource Group. This process will take a few minutes. You can check the status of the process using the proper GET API call.
3. Configure a Pod to use a persistent volume for storage Use kubectl to configure a Pod to use a PersistentVolume for storage. You will need the resource ID of the disk which you can get via the Azure Portal or Azure CLI - az disk show

   Listing 5. example - az disk show

```
az disk show `
    --name $diskName `
    --resource-group $diskSourceRgName `
    --query [].id `
    --output tsv
```

Are Windows Nodes supported?

Yes, Windows Nodes are now supported in Cloudspace AKS.

▼ *Click to reveal the full answer...*

Corresponding information can be found in the AKS Windows Nodes section.
Be aware that Windows Nodes are only possible on newly created cluster. There is currently no option to upgrade existing cluster using Kubenet Networking.

Can I use a `service principle` to access the cluster?

Yes, you can enhance the access with your own AAD Objects by registering those via the Cloudspace API - AKS - Bring your own AAD Object (BYOO).

Why do I have to update my AKS cluster - can't the BDC-Team do this?

You know best which AKS version your applications currently supports. You can also decide when is the best time to update.

▼ *Click to reveal the full answer...*

As it is very individual which AKS version suits your applications best, we do not offer to update your AKS cluster. Moreover, you can decide for yourself when is a suitable time window for the update.
Please make sure you stay within the Azure supported AKS versions.

```
1 az aks get-versions --location <location geocode>
2
3 #Example
4 az aks get-versions --location westeurope
```

How can I upgrade my AKS cluster?

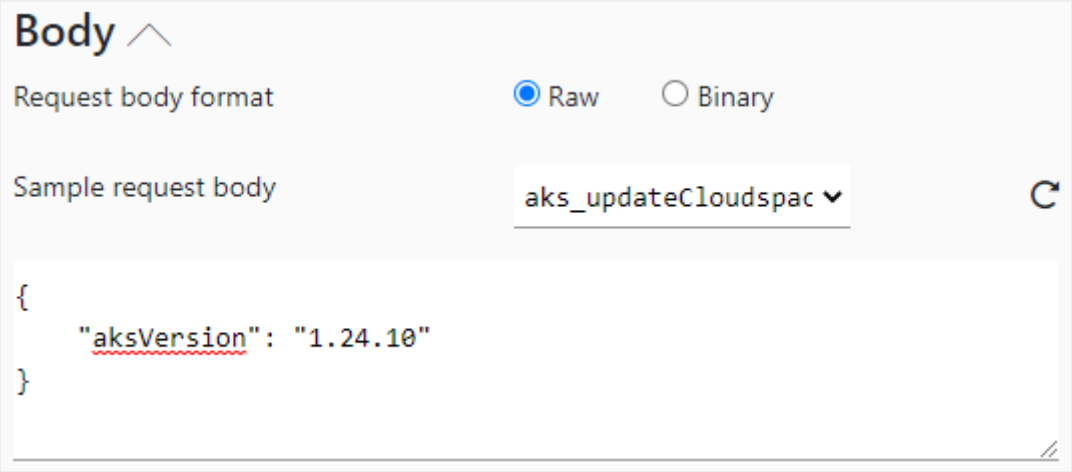
There is an API in the BDC portal to upgrade your AKS cluster.

▼ *Click to reveal the full answer...*

First check the available versions for your AKS cluster:

```
1 az aks get-upgrades --resource-group cs<Cloudspace ID>-<location geocode>-rg
  --name cs<Cloudspace ID>-<location geocode>-<AKS Instance ID>-aks --output
  table
2
3 #Example
4 az aks get-upgrades --resource-group cs00001-we-rg --name cs00001-we-01-aks
  --output table
```

In the BDC portal, you can upgrade via the "Update customer AKS" API in the Cloudspace - AKS section.
In the body, specify the AKS version you want to upgrade to.
Example:



The upgrade will take a few minutes, depending on the size of your cluster. Your nodes will be updated one after another.

7.3.7. Bastion

To be able to securely connect via Internet to virtual machines in Cloudspace, we provide the Azure Bastion service.

With it you can easily connect to a VM e.g. via RDP or SSH.

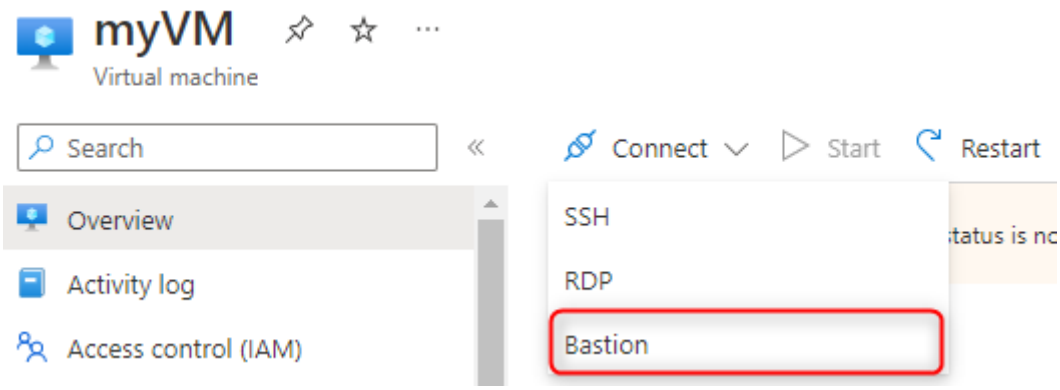
The service can either be used via web browser through Azure Portal or via Native client using Azure CLI.

In order for this to work with your manually created VMs, please ensure that the VM is located within the Cloudspace Location VNET.

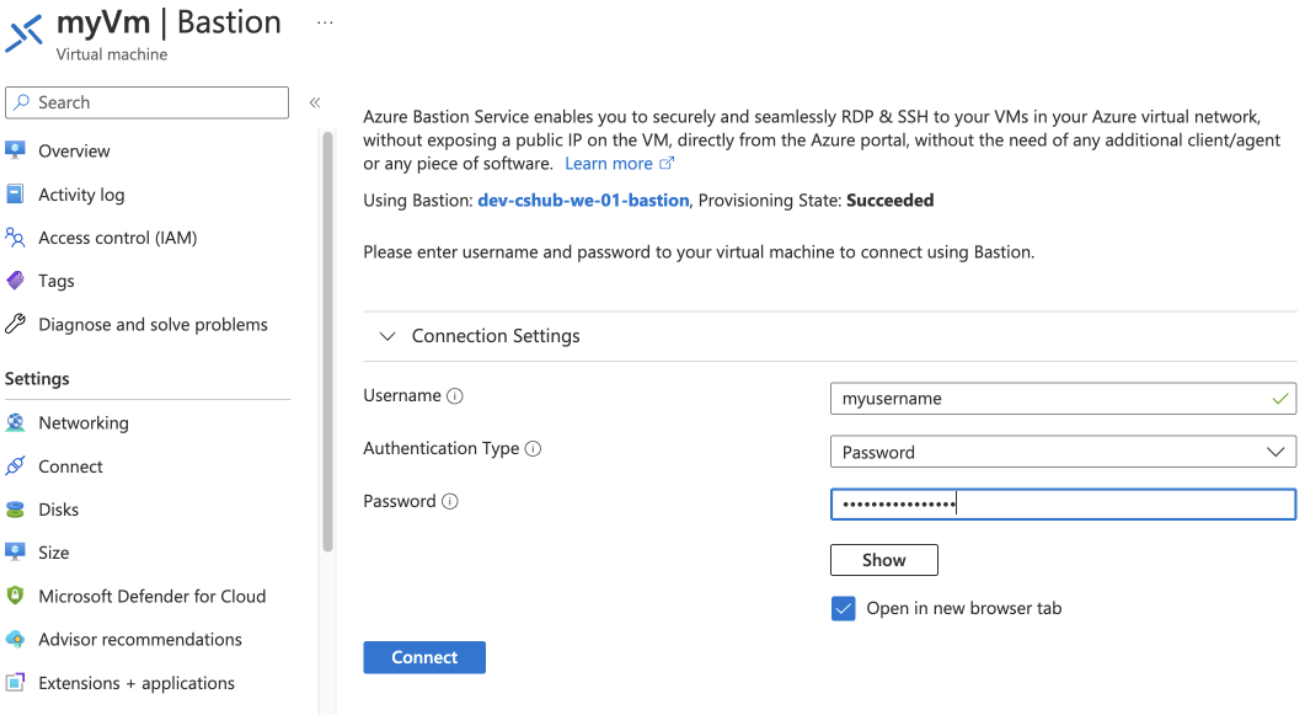
If you want to use your own VNet, you have to deploy a VM that is connected to both the Location VNet as well as your own VNet. This VM then acts as a jumphost to which you can connect through Bastion.
This workaround is caused by an Azure limitation that does not allow routing across three different VNets.

Connect via Azure Portal

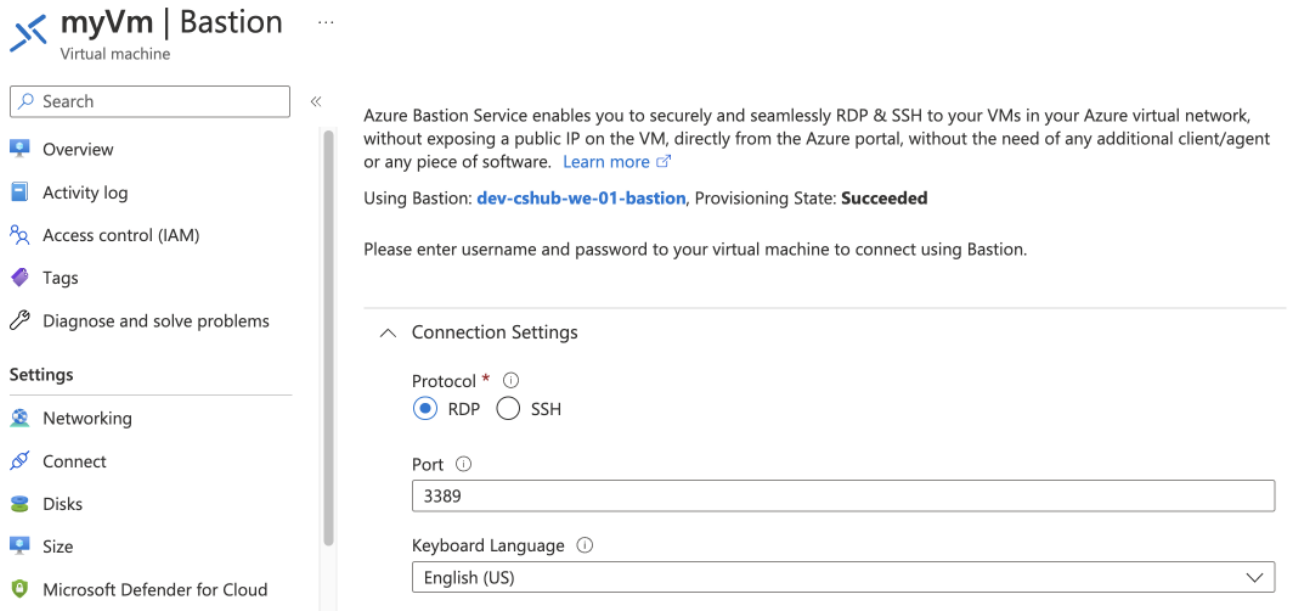
To open a connection to your VM using Bastion, open your desired virtual machine in the Azure Portal and click on **Connect** and **Bastion**.



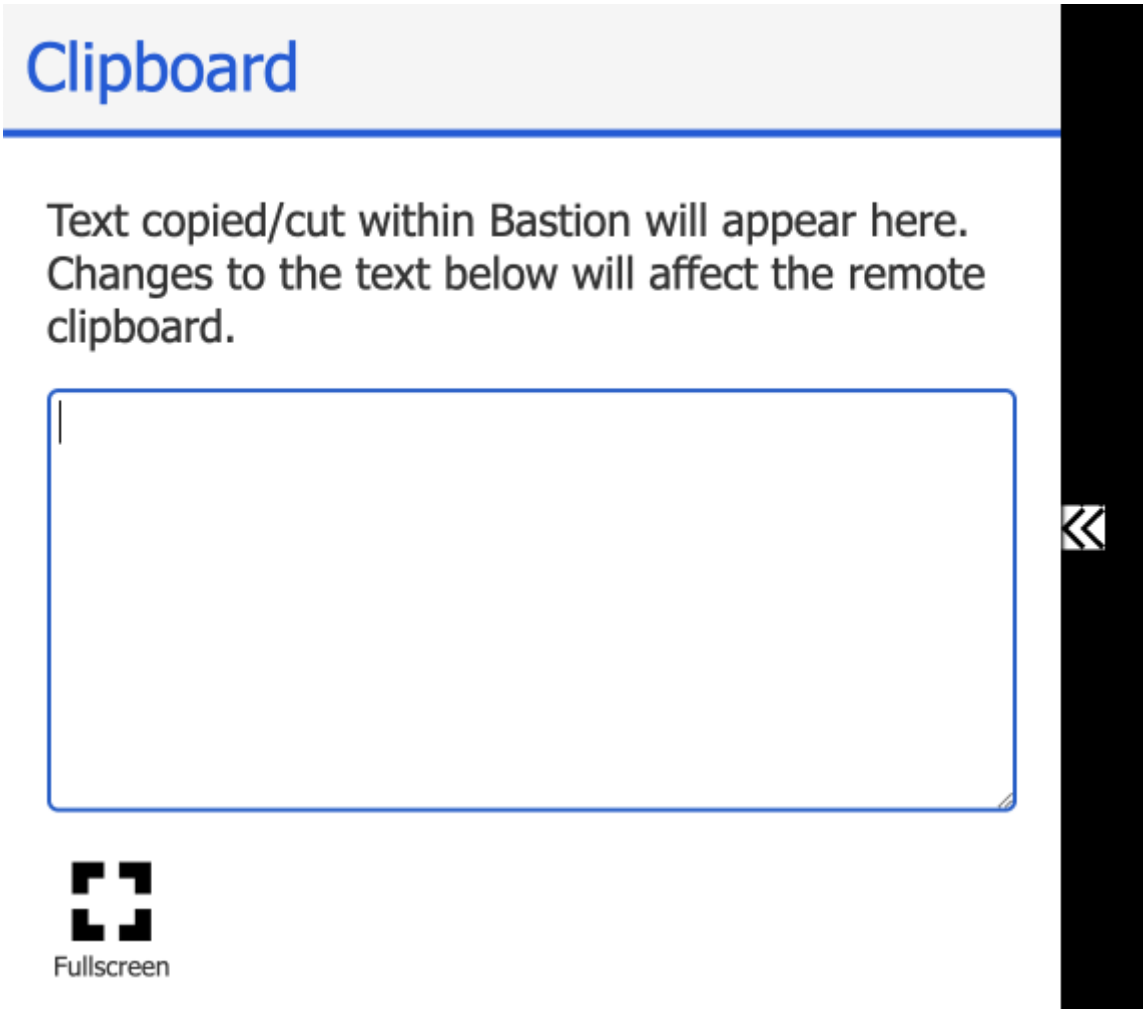
On the next page, enter your username and password which you configured while creating the virtual machine and click **Connect**.



If you want to change the protocol from RDP to SSH (or vice versa) or change the connection port, click on **Connection settings**.



Copy&Paste information from or to your virtual machine through Bastion works via Ctrl+C/Ctrl+V or via the Copy&Paste menu which can be opened by clicking the two arrows on the left site.

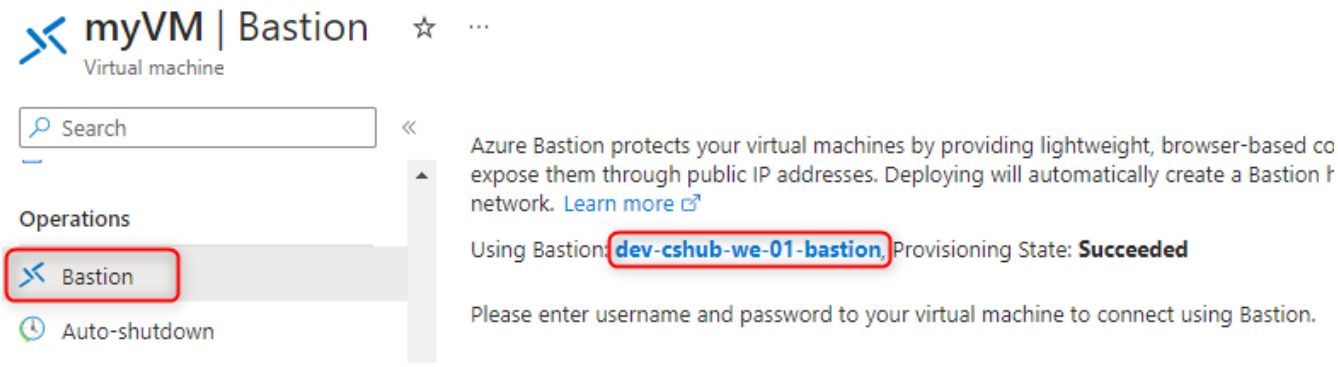


Connect via Native client

If you like to use a native SSH or RDP client on your local machine to connect to a Cloudspace VM, you can do so by using **Azure CLI** bastion commands.
The Cloudspace Bastion is already prepared to support native clients and you can use it directly.

You only need to install the Azure CLI on your computer which you can find here.

Then prepare the Azure CLI:
The subscription key depends on which Cloudspace Hub you are connected to. To get the right subscription ID, go to Operations → Bastion. The associated Bastion service is displayed. Click on it and there you will see the subscription ID.




```
az login
az account list
az account set --subscription "<Bastion subscription ID>"
# The subscription ID of the Bastion service that your VM is connected with.
```

Windows RDP

```
# For Cloudspaces in westeurope
az network bastion rdp --name "prod-cshub-we-01-bastion" --resource-group "prod-cshub-we-01-bastion-rg" --target-resource-id "<VMResourceId>"

# For Cloudspaces in westus3
az network bastion rdp --name "prod-cshub-wus3-01-bastion" --resource-group "prod-cshub-wus3-01-bastion-rg" --target-resource-id "<VMResourceId>"
```

Feel free to access the corresponding Azure documentation Connect to a Windows VM

Linux SSH

Connect via Username/Password

```
# For Cloudspaces in westeurope
az network bastion ssh --name "prod-cshub-we-01-bastion" --resource-group "prod-cshub-we-01-bastion-rg" --target-resource-id "<VMResourceId or VMSSInstanceResourceId>" --auth-type "password" --username "<Username>"

# For Cloudspaces in westus3
az network bastion ssh --name "prod-cshub-wus3-01-bastion" --resource-group "prod-cshub-wus3-01-bastion-rg" --target-resource-id "<VMResourceId or VMSSInstanceResourceId>" --auth-type "password" --username "<Username>"
```

Feel free to access the corresponding Azure documentation Connect to a Linux VM

Upload files

To exchange files between your local machine and your CS VM you can use Azure Bastion tunnel.

```
az network bastion tunnel --name "<BastionName>" --resource-group
"<ResourceGroupName>" --target-resource-id "<VMResourceId>" --resource-port
"<TargetVMPort>" --port "<LocalMachinePort>" --subscription
"<SubscriptionNameOfBastion>"

scp -P <LocalMachinePort>  <local machine file path>  <username>@127.0.0.1:<target
VM file path>
```

See also Azures documentation File upload and download to a VM using a native client

7.3.8. Public Access

To access webservices hosted in Cloudspace from the public internet, we offer a SSL and WAF secured connection to an endpoint within your Cloudspace network, called Public Access.
A central Application Gateway within the Cloudspace Hub will proxy the requests to your endpoint.

You can request a Public Access via our Cloudspace API in the BDC Portal. To create a Public Access you need to enter several data described within the Cloudspace API. The domain which will be generated is using the following pattern (if your Cloudspace has an alias defined, it will be used instead of the ID):

- `https://<hostnamePrefix>-<[Cloudspace ID] XOR [Cloudspace Alias]>.cs.boschdevcloud.com`

The Azure Application Gateway frontend only allows HTTPS traffic, HTTP requests will be redirected to HTTPS.
Traffic will be filtered by a Web Application Firewall with latest OWASP and Microsoft Bot Protection Rules.
For each Public Access you can specify the mode the WAF should operate in:

Prevention

Potentially malicious requests will be blocked with a 403-Forbidden response

Detection

Potentially malicious requests will be logged, but still forwarded to your service

Backend Traffic

The communication between the AGW and your endpoint must be encrypted as per EISA.
Therefore we currently expect your endpoint to provide an HTTPS listener on port 443.

The certificate for traffic encryption will be generated with your Public Access request and placed in the according Location Key Vault.
You can find the certificate with the following name-pattern: `<hostnamePrefix>-<[Cloudspace ID] XOR [Cloudspace Alias]>-sslPem`.
The certificate is signed by our internal Cloudspace Certificate Authority and the AGW will only trust certificates signed by this CA.
The certificate is valid for one year and can be renewed via the Cloudspace API.

The AGW will proxy incoming requests to your endpoint with the same hostname as the public

access.
Therefore you need to configure your webservice to present the generated certificate and recognize the hostname of your public access.

Sometimes it is necessary to provide a certificate prior to setting up the endpoint (or eases the process) - therefore the Public Access Creation API will also work for non-existent endpoints.

FAQs

How to get the certificate file and private key for my Public Access?

You can download the certificate file in different formats from the Azure Portal or use the Azure CLI.

▼ *Click to reveal the full answer...*

1. Get the certificate from the Location KeyVault

```
az keyvault certificate list --vault-name <KeyVault Name>
az keyvault secret download --vault-name <KeyVault Name> --name <Cert Name>-sslPem --file cert_from_kv.pem

#Example
az keyvault secret download --vault-name cs00001-we-abcd1234-kv --name test-00001-sslPem --file cert_from_kv.pem
```

   You can also navigate to the Location KeyVault in the portal and download the certificate manually. Please note different names and file formats.

2. Export the private key from the certificate with openSSL

```
openssl rsa -in cert_from_kv.pem -out private.key
```

3. Create a password protected PFX file

```
openssl pkcs12 -export -in cert_from_kv.pem -inkey private.key -out pass-protected-cert.pfx -passout pass:'secret-password'
```

Can I assign the same hostname to multiple endpoints/IPs?

No, a hostname can only be assigned to a single endpoint/IP Address.

▼ *Click to reveal the full answer...*

No, a hostname can only be assigned to a single endpoint/IP Address.
If you want to balance traffic for a hostname on multiple endpoints, you would have to

provide an according loadbalancer service as an endpoint.

Do you provide wildcard/multi-level subdomains for public access?

No, we don’t provide wildcard or multi-level subdomains.

▼ *Click to reveal the full answer…*

For wildcard and multi-level subdomains, we would need to provide wildcard and sub-domain certificates. These certificates can’t be managed automatically.

How can I reach/resolve my Public Access URL within my Cloudspace location?

In the Location Resource Group, you must add your Public Access domain prefix to the private DNS zone.

▼ *Click to reveal the full answer…*

Adding your Public Access domain prefix as a A record to the cs.boschdevcloud.com private DNS zone in the Location Resource Group, makes the URL resolvable within your VNet.
For example, if your Public Access URL is "test-00001.cs.boschdevcloud.com" you need to add "test-00001" with the corresponding IP as an A record in the private DNS zone.
If you provide your Public Access via AKS, you must use the external IP of the Traefik.

How can I reach/resolve my Public Access URL from other Cloudspaces or other Cloudspace Locations?

If you want to access other Cloudspace Public Access URLs, you must add the Public Access domain prefix with the Public IP of the Application Gateway to the private DNS zone in the Location Resource Group.

▼ *Click to reveal the full answer…*

To make Public Access URLs from different Cloudspaces or Cloudspace Locations resolvable, you need to add their domain prefix to the "cs.boschdevcloud.com" private DNS zone in the Location Resource Group.
In this case, you will need to know the Public IP of the Application Gateway that is assigned to the Cloudspace Location. To get the Public IP you can run nslookup/dig from your local client (or any other client outside your Azure VNet) with the Public Access URL you want to reach.
For example, your Public Access URL is "test-00001.cs.boschdevcloud.com" you need to add "test-00001" with the corresponding Public IP of the AGW as an A record to the private DNS

zone.

7.3.9. Cloudspace APIs

The Cloudspace setup is mainly managed through our Cloudspace APIs.

Those APIs are provided in the BDC-Portal under
My BDCs - <Your desired BDC> - Cloudspace

There you can find all the necessary descriptions of the single endpoints and operations.

7.4. Cloudspace Comparison

Please find a comparison of features for the different Cloudspace tiers in the table below.

Feature	Dedicated	Shared	Lab (old Service)
Dedicated Subscription for your Cloudspace	✔	✘	✘
Create MS Support Tickets on your own	✔	✘	✘
Use entire subscription limits	✔	✘	✘
Create multiple resource groups	✔	✔	✘
Use two pre-defined AKS instances	✔	✔	✘
Create Public Access to Web Resources	✔	✔	✘
Use BDC-Portal APIs	✔	✔	✘
Create custom VNets	✔	✔	✘
Create custom subnets	✔	✔	✘
Use 16.382 IP addresses	✔	✔	✘
Infrastructure as Code Deployment	✔	✔	✘
Use multiple Regions	✔	✔	✘
Grant permissions to own AAD Objects	✔	✔	✘
Shared Bastion Host	✔	✔	✔
Shared Application Gateway	✔	✔	✔

7.5. Costs

The following components will be charged for the BDC Cloudspace Service:

	Costs per month	**Remark**	**Costs charged against**
Shared Cloudspace	352,85€	Service-ID (LASL): SE-1007552	Cost center of the **Cloudspace**
Dedicated Cloudspace	2822,80€	Service-ID (LASL): SE-1007552	Cost center of the **Cloudspace**
Web access configuration via Application gateway	51€	Service-ID (LASL): SE-1007541	Cost center of the **Cloudspace**
Azure Infrastructure consumption	Depending on your resources used at Microsoft Azure	Costs are coming via C/IDA22 from Azure for e.g. VMs, storage etc. This costs are slightly higher than the one you find in the Azure Portal. Service-ID (LASL): SE-1007541	Cost center of the **Cloudspace**

7.6. Azure Hybrid Benefit Plan for Windows Servers

The Azure Platform management team enables the use of Microsoft's Azure Hybrid Benefit for Windows Servers program. Due to this program, Bosch is allowed to re-use existing on-premise server licenses for servers hosted on the Azure Cloud and thereby realizing remarkable cost savings.

By using the BD/PIP1- provided Azure Reporting for Hybrid Benefits it's possible to check how many licenses are still available at Bosch and see a list of all Virtual Machines per Subscription which could make use of them.

Here is a simple how to guide to activate AHB for your VM.

FAQ from Azure Platform management team:
1. Q: How do I get access to the Power BI Report "Azure Reporting - Hybrid Benefit"
A: Please check the documentation here.
2. Q: When activating AHB on my VM, will my running workload be affected?
A: No, activation of AHB on running VM`s will not cause the VM to stop or reboot.
3. Q: How much costs I can save on each VM?
A: Please use the above mentioned calculator to analyze the exact savings for your VM types.
4. Q: Can I configure this property via ARM/Terraform (IaC)?
A: Yes, this is a property of the VM.
5. Q: Do I have to pay something for activating this feature?
A: No, Bosch has already on-premises licenses (see PowerBI report) which can be used right away.

7.7. Azure Reservations

Azure reservations allow you to move away from the pay per use model and purchase a fixed contingent of selected Azure resources (e.g. a VM family). This allows you to get a better price for the resources but only makes sense when you use a consistent amount of the specific resource type. The scope of a reservation can be set to a resource group and all resources in that resource group can benefit from it.

You can use the BDC APIs to get the list for reservations used by your BDC account or to request a new one. Technically, the BDC team has to create the reservation and we decided that the best way to do this is to have a meeting together with you and take there the decision what reservations are useful for your use case.

7.8. Shared Responsibility

As you have full access to the azure service catalog, we cannot take over full responsibility for the resources you deploy and configure. We take responsibility for the resources we deploy to enable Cloudspace. More details will be provided when it becomes general available.

7.9. Compliance

We will provide you with a link to our compliance documentation for Cloudspace when it is ready.

7.9.1. Security Alerts and Recommendations

You can always check for compliance and security of your Cloudspace resources by visiting the Microsoft Defender for Cloud resource in the Azure Portal.

NOTE: Please check that the filter matches the Cloudspace Subscription where your Cloudspace is located for easier readability (e.g. CI-OSE3-BoschDevCloud-CloudSpace0001-Prod).

- Security Recommendations
- Security Alerts

You should always try to ensure maximum compliance by remediating listed security alerts or following security recommendations.

In case of major security alerts the BDC Team will reach out to you to clarify and remediate the issue in a timely manner.

7.9.2. Policies

You can check for compliance and security of your Cloudspace resources by visiting the Policy Compliance section in the Azure Portal.

IMPORTANT: You need to make sure that the scope is set on at least a resource group level for your relevant cloudspace resource groups in order for the correct compliance analysis to appear!
(e.g. Subscription: CI-OSE3-BoschDevCloud-CloudSpace0001-Prod - Resource Group: cs00001-we-rg,

cs00001-we-myrgname-rg, ...)

Beside specifying the scope, you can also go to the specific resource group itself in Azure Portal and check the policies section of it.

7.9.3. Policy exemptions

A lot of the requirements of our security governance is automatically monitored in Azure using Azure Policies. Bosch decided to use the Microsoft managed default policy set, Azure Defender Initiatives. They are automatically applied on the subscription level and are regularly enhanced by Microsoft.

This initiative consists of a defined set of policies which are in different states like preview and production. A security score is calculated out of the production level policies.

We plan to offer a new API function for you to submit exemptions for policy violations in your Cloudspace to document why you cannot comply to a policy.

7.10. FAQs

How can I increase subscription limits/quotas?

Open a BDC Service Desk ticket.

▼ *Click to reveal the full answer...*

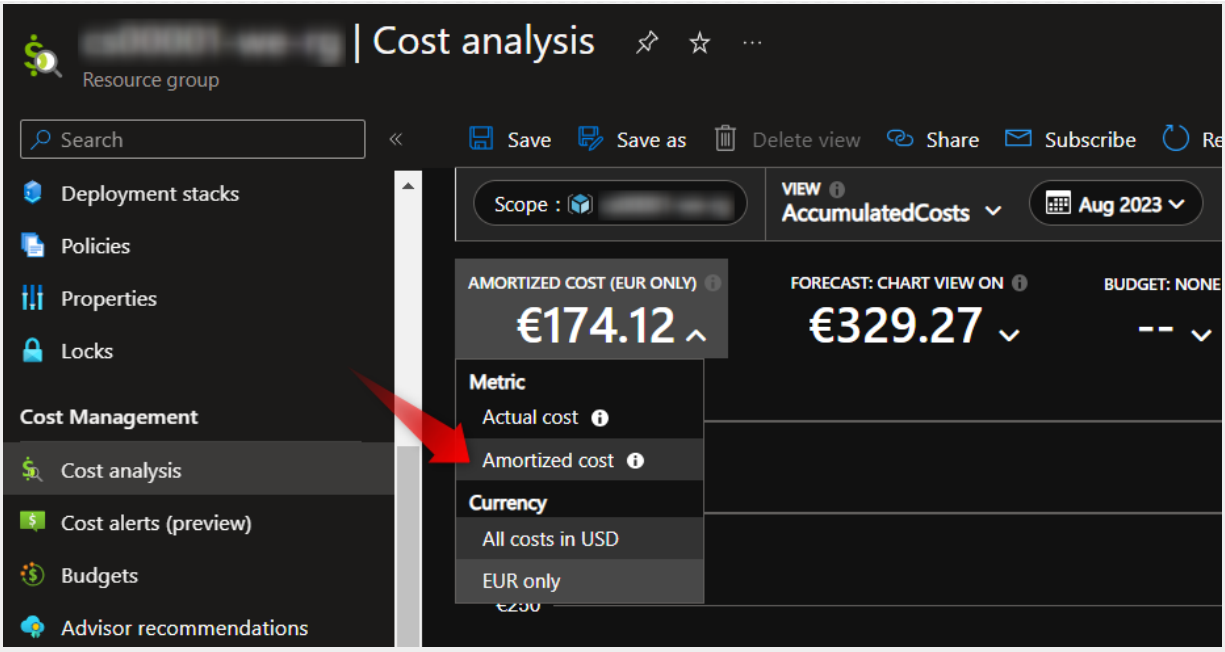
> An increase of Subscription limits/quotas must be ordered by the subscription owner via request to Microsoft, this is a manual process and cannot be completely automated. If it's possible to increase the limit/quotas this can be done via Service desk ticket, our support team will take over the communication with Microsoft and inform you about the status of the request.

Where can I see the cost of my azure resources?

For every resource group the costs can be found under `Cost analysis` in the Azure portal.

▼ *Click to reveal the full answer...*

> To see the complete costs including the resources which are billed using a central billing plan please select **Amortized cost**.



To get the costs of the AKS cluster you must also check the costs of the `MC···` resource group created for each cluster in the background.

The displayed costs gives you a rough overview what the resources will cost. The final costs can be slightly higher.

Can I change my tier from shared to dedicated or vice-versa?

There is no automation to change to a different tier. This will be a case by case assessment.

▼ *Click to reveal the full answer...*

If you wish to change your tier from shared to dedicated or vice-versa, we will need to take a look at your environment. Changing tiers will result in migrating to a different Azure subscription.

How can I migrate from BDC Lab service to Cloudspace?

There is no automation available to migrate from BDC Lab service to Cloudspace.

▼ *Click to reveal the full answer...*

To migrate from BDC Lab service to Cloudspace we recommend you to start from scratch in Cloudspace. Since there are different underlying architectures and permissions, we recommend to recreate your environment.
Azure tools are available to move resources to Cloudspace. Check if your resources are supported.
We can't offer anything similar to a migration guide, as the migration steps are very

individual.

Can I add Azure budgets for my Cloudspace?

Currently there are no options to add budgets.

▼ *Click to reveal the full answer...*

Budgets can be used to manage and monitor costs. This requires permissions on subscription level that we can't grant you.
Therefore it is currently not possible to use budgets in Cloudspace.

Can I change/remove the alias of my Cloudspace

No, as of now there is no possibility to change or remove the alias of a Cloudspace.

▼ *Click to reveal the full answer...*

The alias is tied to a Cloudspace instance upon creation and can't be modified afterwards. These operations might be available in future.

How can I delete my Cloudspace?

To delete your Cloudspace entirely, you must first delete all your services (Public Access, AKS, Customer Resource Group) and then delete the Location Resource Group.

▼ *Click to reveal the full answer...*

To delete your Public Access, use the Cloudspace API - Public access - Remove Cloudspace public access from location
To delete your AKS, use the Cloudspace API - AKS - Delete customer AKS
To delete your Customer Resource Groups, use the Cloudspace API - Resource Group - Delete a resource group. Manually deleted Customer Resource Groups are not permanently deleted and will be restored after an update.

After you have deleted all of your services, you can proceed to delete the Location Resource Group with the Cloudspace API - Location - Remove location from Cloudspace. Since you can have more than one location, your Cloudspace will not be completely deleted until you delete the last location.

If you try to delete a location but there are still services deployed, you will receive an error listing the remaining services.

How can I configure SMTP? Does BDC support SMTP mail configuration?

▼ *Click to reveal the full answer…*

> No, BDC does not offer services related to SMTP configuration. But there is an existing service offered from Bosch called Transactional Messaging System (TMS) that can be used. For more details about Bosch TMS, refer to this documentation.

7.10.1. Network

How can I connect to BCN?

Please use Bastion or public access service to transfer data, as we can't provide a default tunnel.

▼ *Click to reveal the full answer…*

> Since we are unable to provide you a default tunnel for connections to BCN, you must use the Bastion or Public Access service to access the cloud data you wish to transfer to/from BCN.

How can I connect to client license servers?

BD/PLS4 offers license hosting services in Bosch. Please get in touch with the license team to discuss your request.

▼ *Click to reveal the full answer…*

> The BDC doesn't host any license servers in our network. There are internet exposed license servers which can be used from Cloudspace but there are limitations (license conditions, …). Get in touch with the license team via: http://rb-cae.de.bosch.com/ServiceRegistration/?LicenseHosting to evaluate your request.

Can I use VNet peering to connect to other VNets?

VNet peering is allowed within your Cloudspace. For migration purposes it is also temporarily allowed to peer your Cloudspace and Lab(s).

▼ *Click to reveal the full answer…*

> In general, VNet peering is allowed within your Cloudspace, for example, to peer Location and Customer networks. It is also temporarily allowed to peer your Lab and Cloudspace networks.
> You are NOT allowed to peer networks of any other Azure subscription outside of the BDC.

We monitor these violations and reserve the right to delete them.

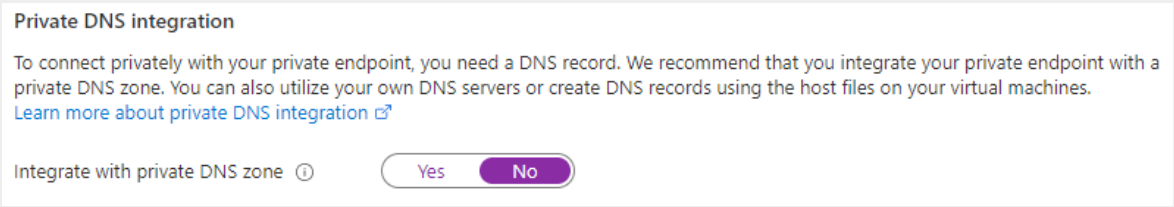
7.10.2. Customer Resource Group

How to connect privately with your Private Endpoint?

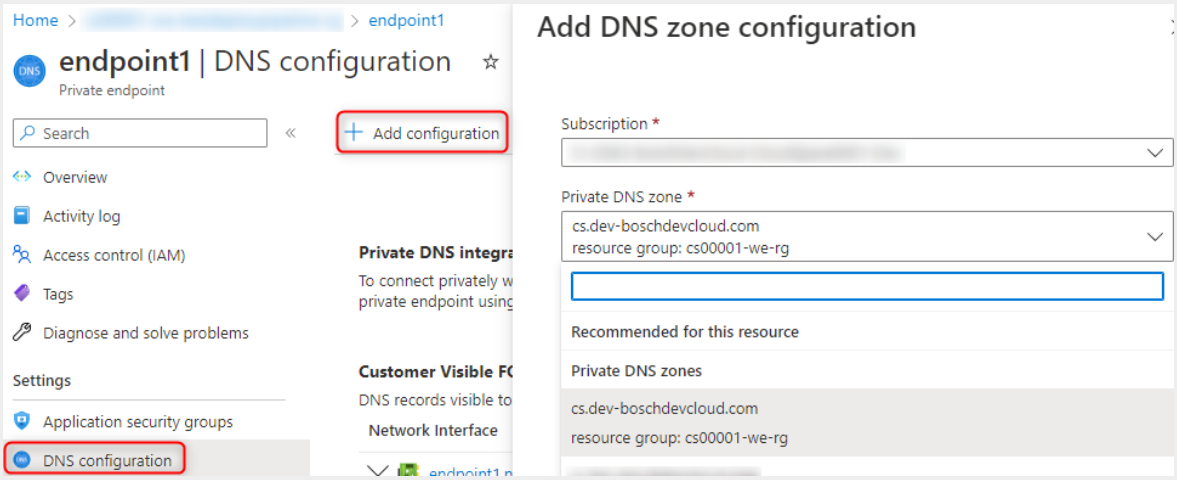
In order to establish a private connection with your private endpoint, you need to manually integrate your private endpoint into a private DNS zone.

▼ *Click to reveal the full answer...*

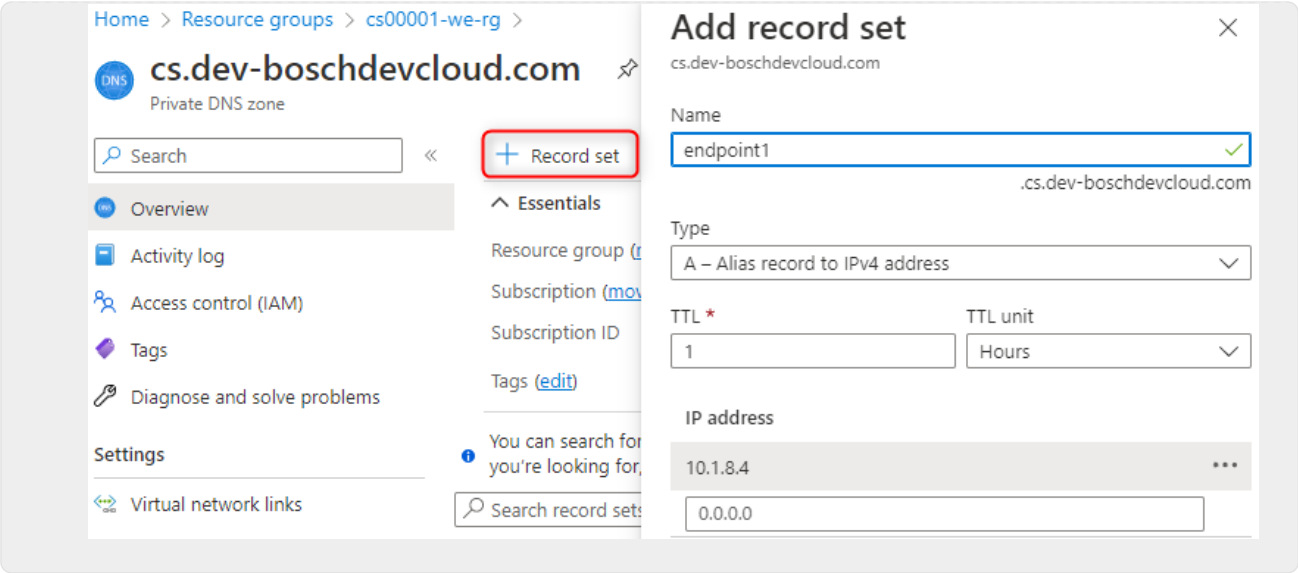
1. Deploy your resource without public network access and without private DNS integration. A private endpoint will be deployed. With an existing resource change the public network access to "Disabled" and create a private endpoint on your own.



2. Go to the private endpoint and select DNS configuration. Add a configuration and select the private DNS zone "cs.boschdevcloud.com".



3. To complete the DNS configuration you must manually add a DNS record for your private endpoint in the private DNS zone. Go to the location resource group, select "cs.boschdevcloud.com" and add a record set with the name of the private endpoint and its IP address.

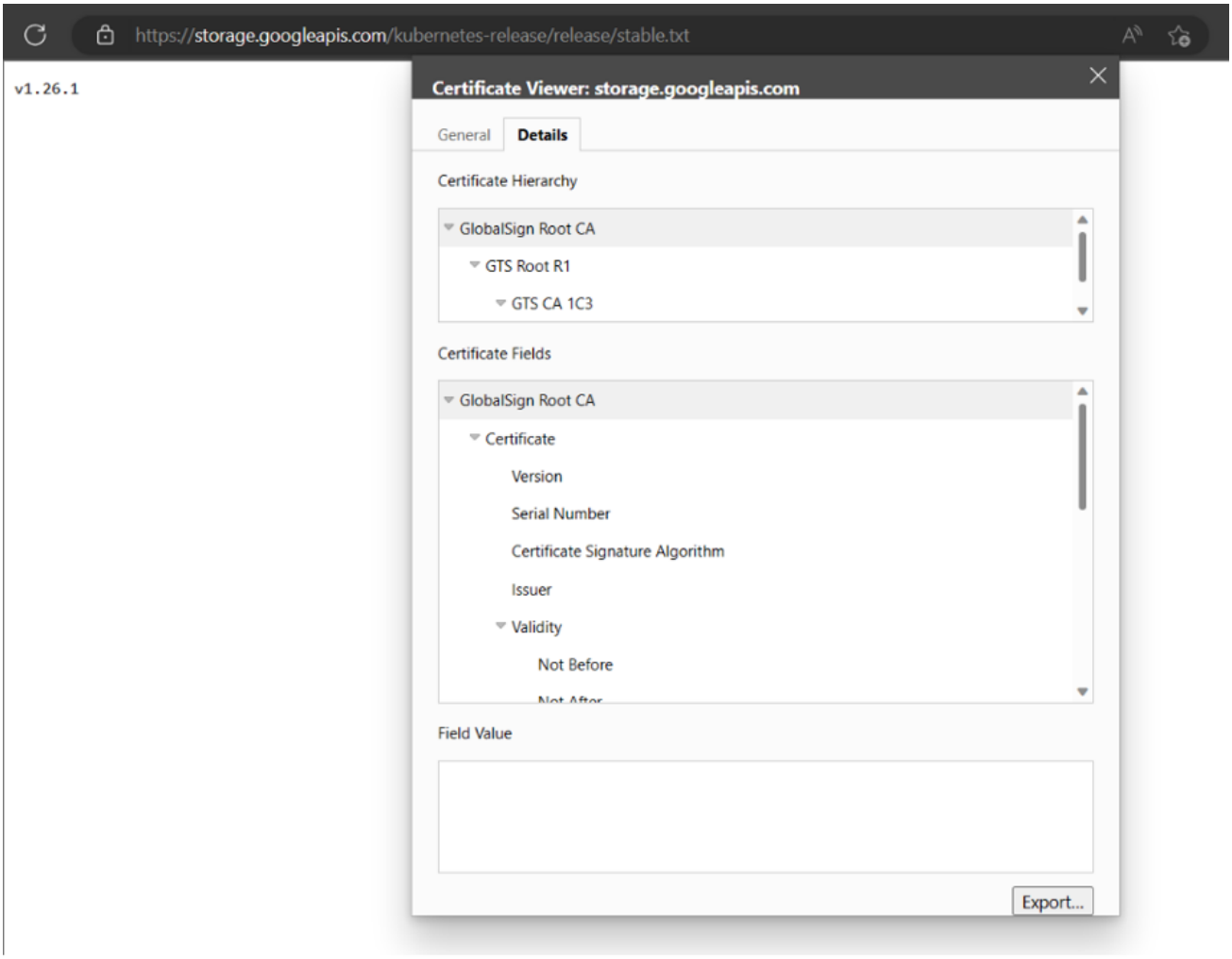


Since Microsoft allows adding a private endpoint to a private DNS zone only with a special naming scheme, it is mandatory to configure the private DNS zone manually and not during the creation of the private endpoint.

CERTIFICATE_VERIFY_FAILED error while installing kubectl

▼ *Click to reveal the full answer...*

While running from Windows powershell az aks install-cli command **CERTIFICATE_VERIFY_FAILED** error might occur for some Azure images. The reason for the exception to appear seems to be the lack of GlobalSign Root CA. Solution is browsing following url in Microsoft Edge and checking certificate.



After browsing the above url with Edge browser, no error occurs.

7.10.3. AKS

How to mount a `Azure Disk` which is located in my customer resource group?

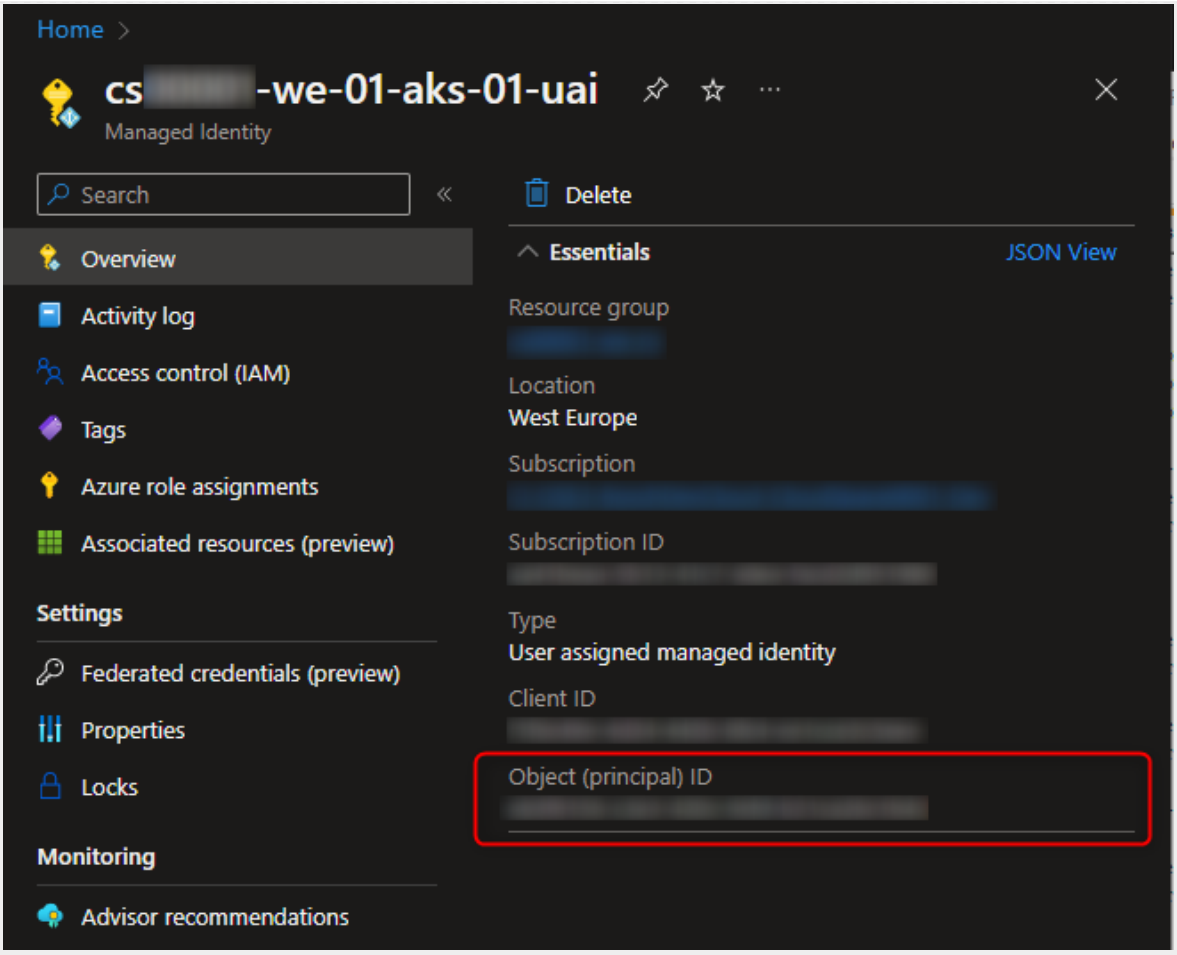
To mount such an Azure Disk you need to grant the AKS User assigned identity permission to access the Azure Disk.

▼ *Click to reveal the full answer...*

1. Get the `Object (principal) ID` of your Cloudspace AKS
   Via Azure CLI - az identity show or Azure Portal.

   Listing 6. example - Azure CLI - az identity show

```
az identity show `
    --name "cs00001-we-01-aks-01-uai" `
    --resource-group "cs00001-we-rg" `
    --query [].principalId `
    --output tsv
```



2. Add the Object ID to your Resource Group Use the Cloudspace - Resource group - Bring your own AAD Object API to add the Object ID to your Resource Group. This process will take a few minutes. You can check the status of the process using the proper GET API call.
3. Configure a Pod to use a persistent volume for storage Use kubectl to configure a Pod to use a PersistentVolume for storage. You will need the resource ID of the disk which you can get via the Azure Portal or Azure CLI - az disk show

Listing 7. example - az disk show

```
az disk show `
    --name $diskName `
    --resource-group $diskSourceRgName `
    --query [].id `
    --output tsv
```

Are Windows Nodes supported?

Yes, Windows Nodes are now supported in Cloudspace AKS.

▼ *Click to reveal the full answer...*

Corresponding information can be found in the AKS Windows Nodes section.
Be aware that Windows Nodes are only possible on newly created cluster. There is currently no option to upgrade existing cluster using Kubenet Networking.

Can I use a `service principle` to access the cluster?

Yes, you can enhance the access with your own AAD Objects by registering those via the Cloudspace API - AKS - Bring your own AAD Object (BYOO).

Why do I have to update my AKS cluster - can't the BDC-Team do this?

You know best which AKS version your applications currently supports. You can also decide when is the best time to update.

▼ *Click to reveal the full answer...*

As it is very individual which AKS version suits your applications best, we do not offer to update your AKS cluster. Moreover, you can decide for yourself when is a suitable time window for the update.
Please make sure you stay within the Azure supported AKS versions.

```
1 az aks get-versions --location <location geocode>
2
3 #Example
4 az aks get-versions --location westeurope
```

How can I upgrade my AKS cluster?

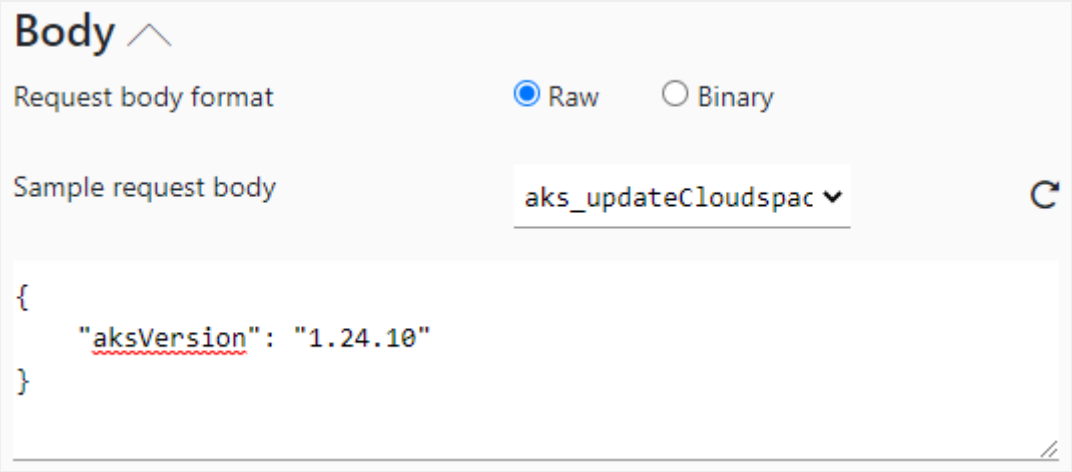
There is an API in the BDC portal to upgrade your AKS cluster.

▼ *Click to reveal the full answer...*

First check the available versions for your AKS cluster:

```
1 az aks get-upgrades --resource-group cs<Cloudspace ID>-<location geocode>-rg
  --name cs<Cloudspace ID>-<location geocode>-<AKS Instance ID>-aks --output
  table
2
3 #Example
4 az aks get-upgrades --resource-group cs00001-we-rg --name cs00001-we-01-aks
  --output table
```

In the BDC portal, you can upgrade via the "Update customer AKS" API in the Cloudspace - AKS section.
In the body, specify the AKS version you want to upgrade to.
Example:



The upgrade will take a few minutes, depending on the size of your cluster. Your nodes will be updated one after another.

7.10.4. Public Access

How to get the certificate file and private key for my Public Access?

You can download the certificate file in different formats from the Azure Portal or use the Azure CLI.

▼ *Click to reveal the full answer...*

1. Get the certificate from the Location KeyVault

```
az keyvault certificate list --vault-name <KeyVault Name>
az keyvault secret download --vault-name <KeyVault Name> --name <Cert Name>-sslPem --file cert_from_kv.pem

#Example
az keyvault secret download --vault-name cs00001-we-abcd1234-kv --name test-00001-sslPem --file cert_from_kv.pem
```

   You can also navigate to the Location KeyVault in the portal and download the certificate manually. Please note different names and file formats.

2. Export the private key from the certificate with openSSL

```
openssl rsa -in cert_from_kv.pem -out private.key
```

3. Create a password protected PFX file

```
openssl pkcs12 -export -in cert_from_kv.pem -inkey private.key -out pass-protected-cert.pfx -passout pass:'secret-password'
```

Can I assign the same hostname to multiple endpoints/IPs?

No, a hostname can only be assigned to a single endpoint/IP Address.

▼ *Click to reveal the full answer...*

No, a hostname can only be assigned to a single endpoint/IP Address.
If you want to balance traffic for a hostname on multiple endpoints, you would have to provide an according loadbalancer service as an endpoint.

Do you provide wildcard/multi-level subdomains for public access?

No, we don't provide wildcard or multi-level subdomains.

▼ *Click to reveal the full answer...*

For wildcard and multi-level subdomains, we would need to provide wildcard and sub-domain certificates. These certificates can't be managed automatically.

How can I reach/resolve my Public Access URL within my Cloudspace location?

In the Location Resource Group, you must add your Public Access domain prefix to the private DNS zone.

▼ *Click to reveal the full answer...*

Adding your Public Access domain prefix as a A record to the cs.boschdevcloud.com private DNS zone in the Location Resource Group, makes the URL resolvable within your VNet.
For example, if your Public Access URL is "test-00001.cs.boschdevcloud.com" you need to add "test-00001" with the corresponding IP as an A record in the private DNS zone.
If you provide your Public Access via AKS, you must use the external IP of the Traefik.

How can I reach/resolve my Public Access URL from other Cloudspaces or other Cloudspace Locations?

If you want to access other Cloudspace Public Access URLs, you must add the Public Access domain prefix with the Public IP of the Application Gateway to the private DNS zone in the Location

Resource Group.

▼ *Click to reveal the full answer...*

> To make Public Access URLs from different Cloudspaces or Cloudspace Locations resolvable, you need to add their domain prefix to the "cs.boschdevcloud.com" private DNS zone in the Location Resource Group.
> In this case, you will need to know the Public IP of the Application Gateway that is assigned to the Cloudspace Location. To get the Public IP you can run nslookup/dig from your local client (or any other client outside your Azure VNet) with the Public Access URL you want to reach.
> For example, your Public Access URL is "test-00001.cs.boschdevcloud.com" you need to add "test-00001" with the corresponding Public IP of the AGW as an A record to the private DNS zone.

7.11. Feedback

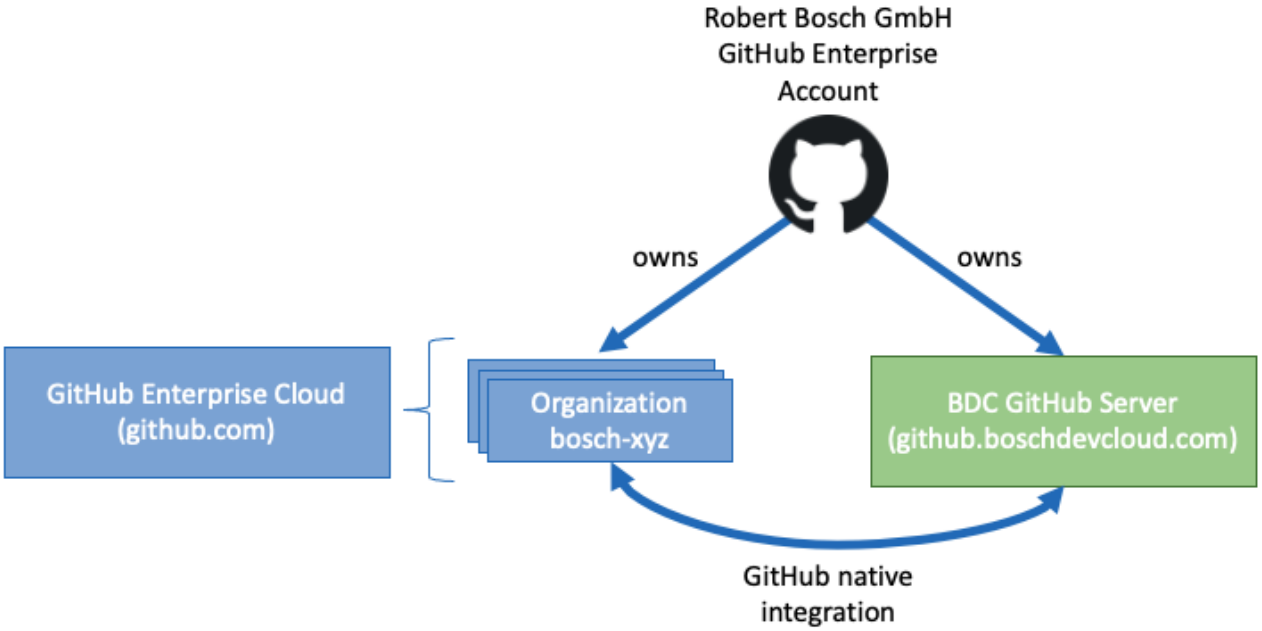
We look forward to receiving your feedback, which will help us to improve in the future!

Chapter 8. GitHub Enterprise

8.1. Introduction

Github is an eco-system for developers to manage their code, plan their work and execute CICD workloads.

GitHub Enterprise Server and GitHub Enterprise Cloud are sharing the same GitHub Enterprise Account. This means that with a single user license you can use both, the server and cloud. Both services are deeply integrated and are integrated into the BDC so that you can benefit from features like the unified search between the server and cloud.



Projects that want to host their code on GitHub should try to collaborate with other projects in the same organization to foster synergies among the projects and avoid silos. When building up your organization keep in mind that there are technical limitations on GitHub that may make a dedicated organization necessary. Repositories on github.com can be easily moved between organizations so even if you start with a single organization, you can still move all of our repositories into a dedicated organization if needed. Only org-level GitHub Project Boards and GitHub Teams can't be moved.

Unlike GitHub organizations on the Enterprise Server, organizations on github.com are public and all public hosted content there can be accessed by anybody from the internet (e.g. https://github.com/bosch). There is no way to hide GitHub organizations on github.com and hence we have to carefully name our organizations to avoid leaking sensitive information about our projects or customers.

General recommendation from GitHub is to reduce the number of organizations to a minimum and avoid that users are part of many organizations. Further information can be found in this document. As there are many up- and downsides of a single shared organization there is no simple

answer to the question how many organization you should have.

8.1.1. Pricing

	Costs per month	Remark	Cost charged against
GitHub Organization & repositories	no additional costs for GHES, additional costs possible for GHEC (see Billing on Organization level on GHEC)		
GitHub@BDC Operations and License	15,23 EURO/user		cost center of the user
GitHub.Com@BDC Operations and License	10,26 EURO/user		cost center of the user
GitHub@BDC Advanced security License	45,38 EURO/user	This is a required license for *every contributor* in a repo within the time period of 90 days which is configured for advanced security. This is in addition to the user license.	cost center of the user
GitHub@BDC Copilot	20,50 EURO/user	This cost includes the required license. Be aware that in order to use Copilot, you also need to have a regular access to github.com, see cost for GitHub.Com@BDC Operations and License	cost center of the user
GitHub dedicated instance (on request)	variable	This is in addition to the user license.	cost center of the instance

Billing on Organization level on GHEC

Billing happens on organization level and the assigned cost center for the organization will be charged on a monthly base. Monthly costs depends on your usage of the GitHub metered products within your organization, like GitHub managed Action Runners, Package Storage, Codespaces and so on. The pricing of the GitHub metered products can be found in the GitHub billing documentation.

There's also a **Pricing Calculator** from GitHub, where you could estimate the costs for your usage of github.com: https://github.com/pricing/calculator
Just enter your approximate usage of the different items, and it will give you the sum. These usage

costs of github.com will simply be passed-through by our billing mechanism to your cost-center. Attention: do not enter number of users as we have a different cost per user license than what is shown in the calculator, use it only for the metered services.

At this point it's not possible to set a cost limit on organization level but GitHub is actively working on this feature.

Each organization owner can download a full usage report directly from GitHub in form of a CSV file that looks like the following:

```
1 Date,Product,Repository Slug,Quantity,Unit Type,Price Per Unit,Actions Workflow
2 2021-02-12,actions,bosch/repositoryname,3,UBUNTU,$0.008,.github/workflows/ci.yml
3 2021-02-15,actions,bosch/repositoryname,1,UBUNTU,$0.008,.github/workflows/ci.yml
4 2021-02-
  15,actions,bosch/repositoryname,4,UBUNTU,$0.008,.github/workflows/release.yml
```

FAQ

I use GHEC and GHES, which license do I need and what is charged?

A license for GHES and GHEC is needed, and license and operation costs for GHES and GHEC will be charged.

I use GHES without any orgs, only private repos, what are the costs?

Normal GHES costs will be charged.

I use Github hosted runners in GHEC, will it be charged?

The charges are applicable and booked under the GHEC-Org cost center.

How to opt out from Github Service?

If you only have a private repository and do not login for more than 90 days, your user will be marked as dormant and will not be charged (it can be reactivated just by logging in). If you are part of any Github organisation, make sure to remove yourself from the respective IdM roles.

I would like to use Github pages, do colleagues who view them need a licence?

Yes, to view Github pages, they will have to login, and that requires a licence.

If I need to use Github, do I need to purchase a licence?

A license will be assigned to you automatically once you login to the Github instance. In the case of an Advanced Security License, seats are filled on a first-come, first-served basis, but we will purchase new seats based on demand, which may take some time.

Will I be charged twice if I am a member of two GHES organisations (one internal and one BIOS)?

Charges are based on user rather than organisation or repos, so you will not be charged twice.

8.1.2. What can I order?

You can order your own organization hosted on GHES, our BDC-GitHub on

github.boschdevcloud.com, or on GHEC on github.com.

Order an Organization

Before creating your organization, please make sure that you understand your options for code hosting on BDC. Have a look also at the Pricing. Following options for code hosting and software collaboration are offered by BDC:

- **GitHub Enterprise Server (GHES)**
  - A Github Organization
  - A BIOS GitHub Organization
- **GitHub Enterprise Cloud (GHEC)**

GitHub Enterprise Server (GHES)

GitHub Enterprise Server is a self-hosted version of GitHub's collaboration and version control platform. GitHub Enterprise Server is hosted by BDC and accessible via URL - `github.boschdevcloud.com`. It allows us, as Bosch, to host our own server instance and have complete control over the infrastructure and data.

If you're unsure which service to use or if you're not planning to use any of the GHEC advanced features like GitHub managed Action Runners and Codespaces, please refrain from creating a GHEC organization and create an organization on BDC-GitHub instead, our own GHES instance.

If you're just looking for a simple git-based code hosting solution or you're not yet familiar with GitHub, then the GHES is your safe harbor.

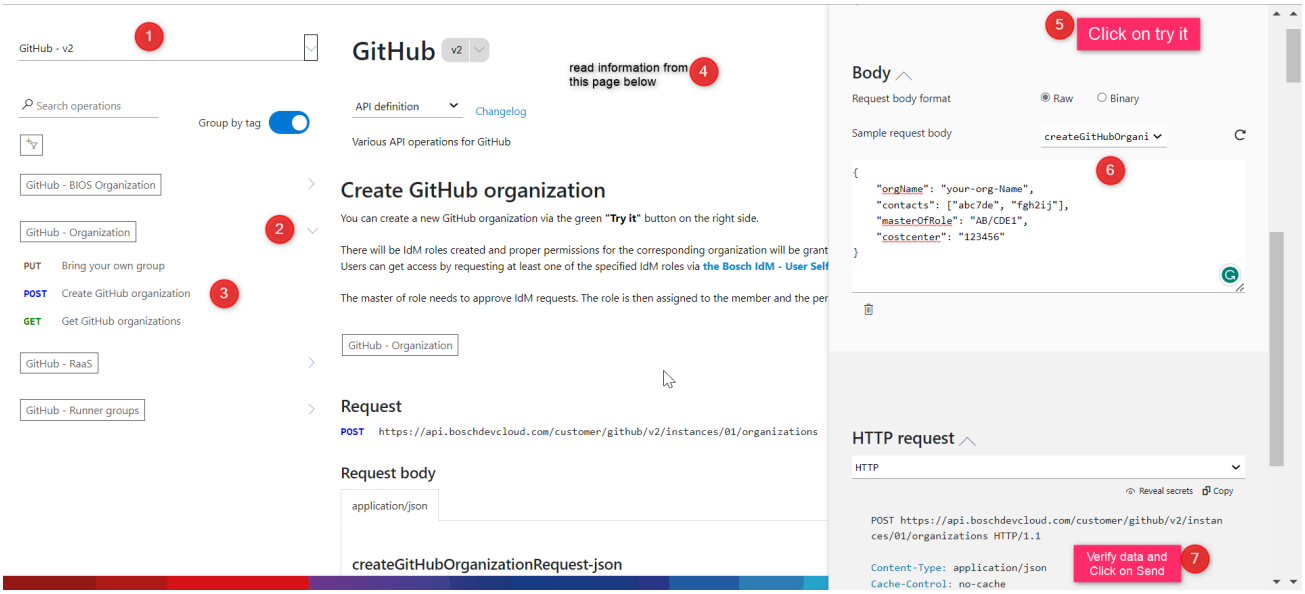
Please refer to the detailed guide below in GitHub Enterprise Server section.

Optionally, you could order a BIOS Organization, which is the Bosch implementation of Innersource. More information can be found here.

How to Order a GHES org?

- Go to the BDC Portal and login.
- You must be a BDC account admin to do this. If you are one, you can see your BDC ID (BDC-XXX).
- Click on it and go to the GitHub-v2 API.
- Use `POST Create GitHub organization` to create an organization.
- It will return 2 IdM roles for Organization Owner and Member access, Give system some time to create IdM roles, and use IdM self-service portal to request roles.

▼ *Click to reveal reference screenshot...*



In case you would like to order a BIOS GitHub Organization, kindly use API - `POST Create GitHub Bios organization` and not the above mentioned one.

GitHub Enterprise Cloud (GHEC)

GitHub Enterprise Cloud is a SaaS offering, is hosted by GitHub and accessible in URL - github.com.

Please make sure that you understand the differences between GHES and GHEC and the Organization Limits and Constraints of GHEC before proceeding with creating an own GHEC organization. If possible, consider joining an existing GHEC organization before creating your own one because being a member in many different organizations has a bad impact on the developer UX and is not recommended by GitHub.

Each organization must have a valid Bosch cost center number which will be charged for any use of the GitHub metered products, such as using the GitHub managed Action Runners, Package Storage and Codespace usage. The pricing of the metered products can be found in the GitHub billing documentation. If you want to avoid any additional costs in your organization, please create an organization on GHES instead. For more details have a look on the Pricing documentation.

Please refer to the detailed guide below in GitHub Enterprise Cloud section.

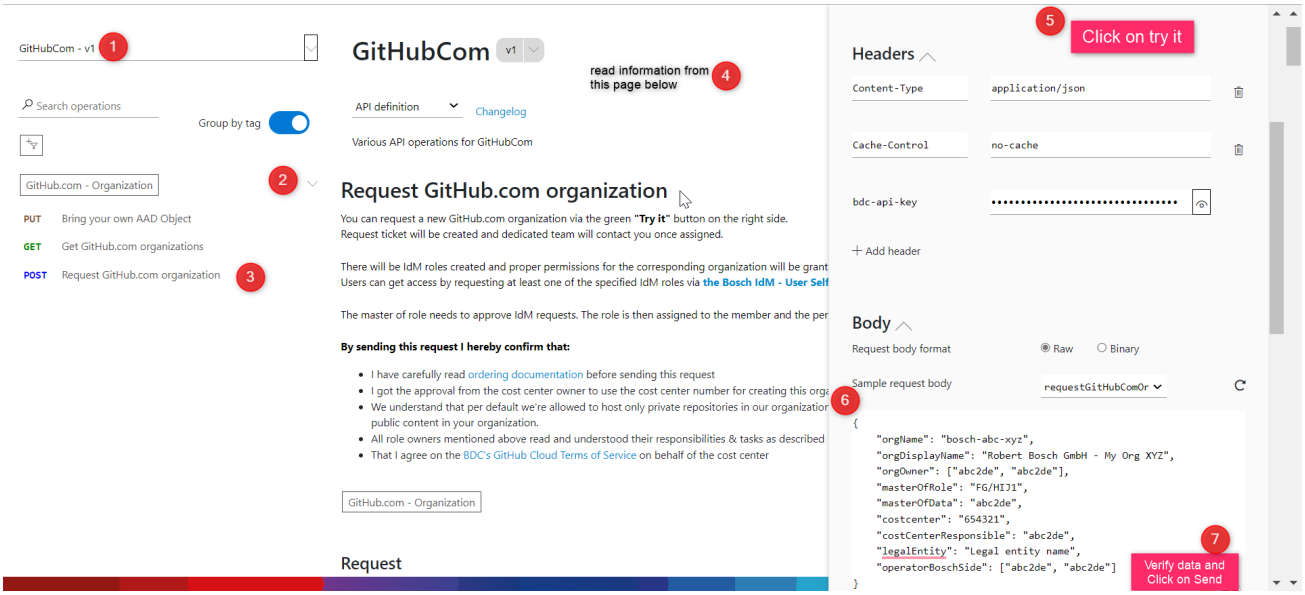
Our Bosch owned GHEC organizations are configured to host private repositories only. This means that you're not allowed to host public repositories or forks in your organization. If you need to host public content or forks in your organization, please get in touch with the BDC team via Service Desk so that we explain your options for hosting public repositories. We've implemented an automation that will turn public repositories into private ones and deletes forks as soon as they've been detected (as forks can't be made private) without further notice.
Discussions about hosting public repositories and forks in our Bosch enterprise account are currently ongoing.

How to Order a GHEC org?

- Go to the BDC Portal and login.
- You must be a BDC account admin to do this. If you are one, you can see your BDC ID (BDC-XXX).

- Click on it and go to the GitHubCom-v1 API.
- Use `POST Request GitHub.com organization` to request an organization.

▼ *Click to reveal reference screenshot...*



The ordering process can't be fully automated at this point, hence the creation of a new organization takes several days to complete.

> Some of the organization metadata on github.com like the organization name, description and so on are public and can be accessed by anybody on the internet (e.g. github.com/bosch). There is no way to hide this information like on GHES where all organizations are completely private. Therefore the organization name needs to be chosen carefully to avoid leaking sensitive information (> SC0) like customer information or other information that is not allowed to be shared in the public.

- The **organization name** must:
  - follow the naming scheme `bosch-abc-xyz-···` (for GHEC only)
  - needs to be conform to our code of conduct (nonviolent language, ...)
  - contains only lowercase alphanumeric characters + "-" (lowercase hyphenated)

It is a good idea to use a stable name and keep in mind that the name of your organizational unit might not be the best choice as it changes over the time.

- The **organization display name** can be any string and will be displayed as `"your legal entity" - "your display name"` inside of your GitHub organization. E.g. `Robert Bosch GmbH - My Org XYZ`
- Adding new members to an organization always requires the approval of the IdM Access right owner, who has been assigned to the IdM roles of your GitHub organization. The default Access right owner, who is responsible for the IdM WOM element is not always the best choice. Creating a new IdM WOM element is also very difficult hence consider adding an Access right owner delegate to your organization's IdM role. Have a look on the IdM help for setting up a MoR delegate.

By ordering an organization you're agreeing on the BDC's GHEC Terms of Service on behalf of the cost center that is assigned to the organization.

8.1.3. Access and permission management

Access to your service is available at github.boschdevcloud.com or github.com

GitHub uses SAML-authentication to achieve Single-Sign-On. By starting the web-application you will automatically get logged in with your regular Bosch account. If you want or need a different user, just log out and you will get a sign-in window.
Your user account gets created the moment you log into the system for the very first time.
If your login usually works, but one day the SAML login doesn't let you in anymore (maybe due to outdated cache data), you can manually logoff from the SAML provider with this link: https://stfs.bosch.com/adfs/ls/Idpinitiatedsignon.aspx

There's an automatic synchronization in place which will provide your account with the correct permissions. This can only be started after account creation so you might not see all resources when you login the first time.

Each Github Organization has the following IdM roles available:

- **Owner**: Organization owners are like the "admin" of an Organization, they have all full access to all settings.
- **Member**: The role for every regular user in this Organization.

For further details of these permissions, please visit GitHub's documentation.

Take a look at the FAQ for How to request access to an Github organization?

If you are onboarding new users, ask them to login to GitHub@BDC at least once. This will create an account for them and then our user-sync will add all users from the IdM group to the respective GitHub Organizations.

Teams for permission management

To comply with the Bosch central directive on identity and access management, the projects are strongly encouraged to use GitHub Teams. Direct access on repositories shall be avoided if possible.

GitHub Enterprise Cloud customers can make use of a feature called "Team synchronization", With this you can synchronize a GitHub team with an IdP group. You can assign an IdP group to multiple GitHub Enterprise Cloud teams and can connect up to five IdP groups to a GitHub Enterprise Cloud team. More on this topic can be found in GitHub Team IdP Sync docs. Team synchronization is not a user provisioning service and does not invite non-members to join organizations. This means a user will only be successfully added to a team if he is already an organization member. This feature is in private beta for GitHub Enterprise Server and not enabled.

As every project will have an unique team setup (some may have only one team, multiple teams, or a nested team structure) it's almost impossible to find a common approach on how the team setup should be reflected inside of the organization. Projects can also use CODEOWNERS files to further

control the permissions on a repository in combination with GitHub Teams. Ultimately, it's the project's responsibility to decide how they want to manage access to their GitHub Resources and it's also their responsibility to keep their solution CD conform.

Outside Collaborators

GitHub also has a feature called outside collaborators but the usage of this feature is not permitted due to Central Directives CD 07900 & CD 05106 at Bosch. Access to Bosch Data needs to be documented at Bosch side via an Identity and Access Management Tool so that access could also be centrally revoked, if needed. Therefore, if externals or colleagues of other departments need access to a specific repository within your Org to collaborate with you, they need to apply for the IdM Member Role. Manually adding outside collaborators in GitHub is NOT Bosch compliant.

8.1.4. Organization Role Definitions

Cost Center Responsible

Description: The Cost Center Responsible is the initial role, which has to be defined, before creating an organization or other GitHub Resources. He/She nominates the other roles (see below). He/She is responsible for the costs and has the overall responsibility for the organizations and GitHub Resources that are assigned to his/her cost center.

Responsibilities & Tasks:

- Can delegate responsibilities & tasks
- Nominates other roles (see below)
- Approves by default as "Master of Role"
- Is responsible for the organization costs
- Ensures regular cost monitoring

Master of Data

Source: RB/GF 105, CD 02900-001, CD 02981.

Description: The Master of Data is responsible for the Bosch data stored in the GitHub solution. He/She must classify the data based on the criteria Confidentiality "C", Integrity "I", Availability "A".

Responsibilities & Tasks:

- Identifies and classifies the information to be processed (stored, transferred etc.) in the GitHub solution
- Overall responsibility, that the necessary measures from CD 02900 are identified and implemented to protect the information and to be compliant with legal data protection regulations

Organization Owner

Source: CD 05106.

Description: Manages the GitHub organization.

Responsibilities & Tasks:

- Responsible for management of the GitHub organization
- Ensures CD compliant user administration and role assignment in the GitHub organization
- Maintain and implement the authorization concept for the GitHub organization incl. annual check
- Disables or removes not required GitHub Resources

Operator (Bosch-side)

Source: RB/GF 105, CD 02900, CD 07900.

Description: The Operator (Bosch-side) ensures ongoing operation, maintenance, and troubleshooting of the organization.

Responsibilities & Tasks:

- Responsible for maintenance and operation of the GitHub organization (e.g. GitHub Action workflows)
- Responsible for the secure operation of the GitHub organization during its whole lifecycle according to our CD
- Continuously monitors the organization for vulnerabilities or delegates this task to members or teams of each repository
- Continuously monitors the Bosch CERT Security Advisories
- Triggers in case of affected/vulnerable components the implementation of patches
- Responsible for reporting security incidents according to Bosch's Information Security Incident Response Communication Plan (IRCP)
- Single point of contact for incident response assistance

Competencies:

- Familiarized with Bosch's Information Security Incident Response Communication Plan (IRCP)

Member

Every user, who should have access to one of the repositories of an organization needs to be a member of the organization.

8.1.5. GitHub products and terminology

Glossary

GitHub Resources: Repositories (includes packages, releases, project boards, issues, actions, ...), teams, org-level project boards and GitHub/OAuth Apps.

See also the official GitHub glossary.

GitHub products

GitHub provides a good overview regarding its products and the differences between them. It helps to understand the difference between what you use in private when you do work on github.com to the work in the Bosch context with Github Enterprise. We offer both GitHub Enterprise Server (GHES) and GitHub Enterprise Cloud (GHEC) in the BDC. All other products are not available as they are not designed for enterprise customers.

GitHub's product overview

GitHub Connect

With GitHub Connect, we can share certain features and data between our GitHub Enterprise Server instance and our GitHub Enterprise Cloud Organization or enterprise account on github.com. We are constantly evaluating which features we can use and how (and if GitHub would like to get money for them).

Right now, we have all features activated:
- Server statistics
- License sync
- Unified search of github.com (including private repos)
- Dependabot (without notifications)
- Actions usage from github.com
- Contribution sharing with github.com (if you have an account there and connected the GHES and GHEC accounts via GitHub Connect in the settings)

We don't share any code with github.com, no line of code leaves our BDC-GitHub Server instance!

8.1.6. Organizations

Github uses Organizations as managing entity. You will get your own Organization with the capabilities to fully configure and manage it.

Teams in your own Organization

As Organization Owner, you are free to create (and delete) teams. You may only add people to a team when they have created an account in GitHub during their first login.
You're able to create new teams within GitHub. But there's no automation behind it, you need to manage all team members on your own. You need to be repository admin to create a new team. Teams can only be added on the Organization-level, you can't add teams in your own repositories.

Repositories

In case you're an owner, you're able to add existing GitHub users to the list of collaborators. You have to decide, which role the new user will have in this repository. By default there are these roles: Admin, Maintain, Write, Triage, or Read. But you can define your own Custom roles for each team. It's also possible to add a team to the list of collaborators. Here you can also decide between the available roles for that team regarding that repository.

The default visibility of a repository is *private*.
Repository invitation is de-activated.

Repository deletion is de-activated, but can be executed by the organization's admins

There are different types of repositories:

- **personal**: You can create personal repositories which are only accessible by you. Keep in mind that data/code should be stored in the business context - i.e. an organization and *not* in a personal repository. Use of personal repositories is not recommended and their content will be deleted automatically when your account gets disabled.
  Personal repositories could be private or public, hence we regularly check for public repos and change their visibility back to private.
- **internal**: Internal repositories are used for BIOS only. All repositories set to internal can be read by every member of the BIOS community by default. They should reside in BIOS organizations and not in general organizations. See the chapter BIOS on GitHub Enterprise Server for more information.
- **private**: Private repositories are only visible to people who have been granted access to that repo. This is the type you should choose for repositories other than BIOS repositories.
- **public**: Such repos are visible to everyone who can login to the server. So not public in the sense of "the world" but for everyone who logs onto the server with a Bosch Account. You will share everything in a public repo with all Bosch GitHub users.

As long as you can't ensure, that it's 100% legal (e.g. in terms of Bosch CDs, international laws, and/or taxes) to share your code within all the distributed Bosch companies, please don't use public repositories!

Here is some more information from GitHub about repository visibility.

All repositories are *unprotected* by default. If you need some kind of branch protection (e.g. no direct commits, but only with a pull request) you have to set all these restrictions on your own. Restrictions can be defined as a member of the according repository's admin group.
Further information about protected branches and how to define them can be found at GitHub Documentation.

The upload limit for each file in a GitHub repository is 50MB.

It's possible to **transfer repositories** from one organization to another one. But keep in mind, that the access rights for the transferred repository will get messed up. You will probably lose all admin access to your repository and need to ask the target org owner to grant your elevated access rights again. This is unfortunately the intended behavior from GitHub, as we're told in a support ticket regarding this topic.

Projects

In GitHub you are able to create your own Project boards. Projects can be added to different levels:
- Organization
- Repository
- Team

The difference between these three project boards are:

- Project boards on the **Organization level** can be added by every GitHub user and linked to several repositories. Of course, you need to be a member of all the repositories, you'd like to connect. As a creator of this project board, you can decide, if the visibility is public or private (not during creation, though, but in its settings). You can define the general access to your project board for all Organization members: Admin, Write, Read and None.
- Project boards on the **Repository level** are allowed by default and are pretty simple. Every member of the repository can add one, but you can't set any access rights nor could you link other repositories to it. The only thing, you can do is setting a name and a description; and - of course - you could delete it again. If you're sure, you'll never need project boards, you can deactivate them completely for your repository in its settings (you need to be member of the repository's admin team).
- Project boards on the **Team level** are not possible for regular GitHub users (members), you need to be an Organization admin to define them. You can't create a new project board on this level anyways, it's only possible to link existing boards to existing teams. If you need such a link, just get in contact with the according organization owner and ask them to do it for you.

8.1.7. GitHub Collaboration

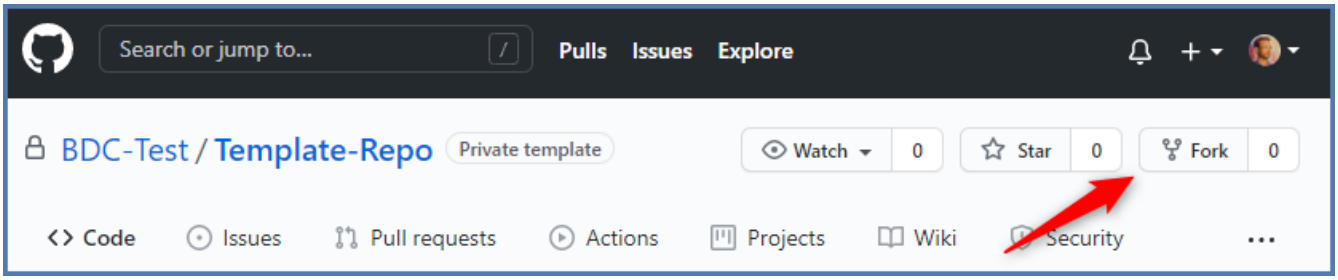
A huge advantage of GitHub is its support of collaborative work.
In the first step, you have to assign each person the according IdM role, so they get access to your organization. There are *owners* and *members* in your organizations, see here for details.

Access Rights

Inside your organization you're free to define all needed access rights. You can - and have to - define the access rights for all repositories in your organization. A good way might be the creation of **teams**, but you could also assign all these access rights manually for every user.
To learn everything about the different repository roles in GitHub (by default: Read, Triage, Write, Maintain, Admin), please take a look at the GitHub documentation.

Forking

Another way of collaboration is **forking**.
To fork a repository basically means, to create a copy of it. This copied repository is linked to the repository, which it was forked from, the so called root repository.
You have to decide on your own, if forking is the best way for you. Maybe this post "Talk, don't Fork!" might help with your decision.
To fork a repository, you simply have to click on the "Fork" button in the top right corner of the repository, which you'd like to fork.

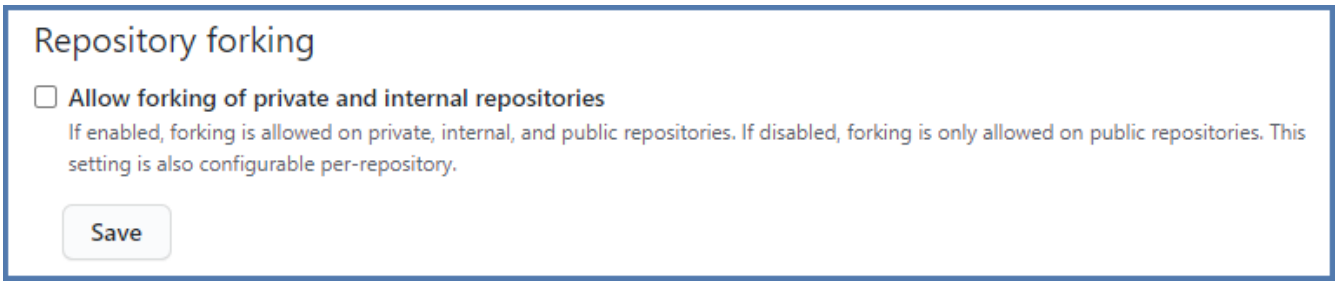


In case, you can't fork a private repository, you might get in contact with the organization owner

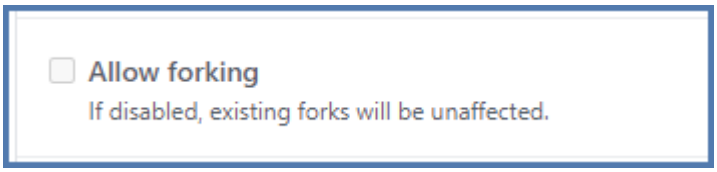
and explain your request. You can see all organization owners with this link: https://github.boschdevcloud.com/orgs/<org-name>/people?query=role%3Aowner

- Settings

Within your organization, you can decide, if your users should be able to create forks from private (and internal) repositories. If you don't do this, only forks of public repositories are allowed.

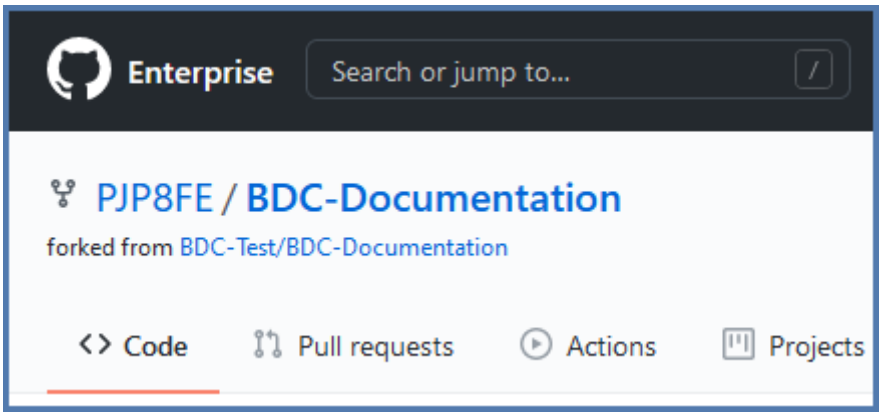


Additionally you could define in the settings of each repository, if you want to allow the forking of this repository. If you don't set this option, nobody can fork this repository. You can see it easily by the greyed-out "Fork" button on the top right.

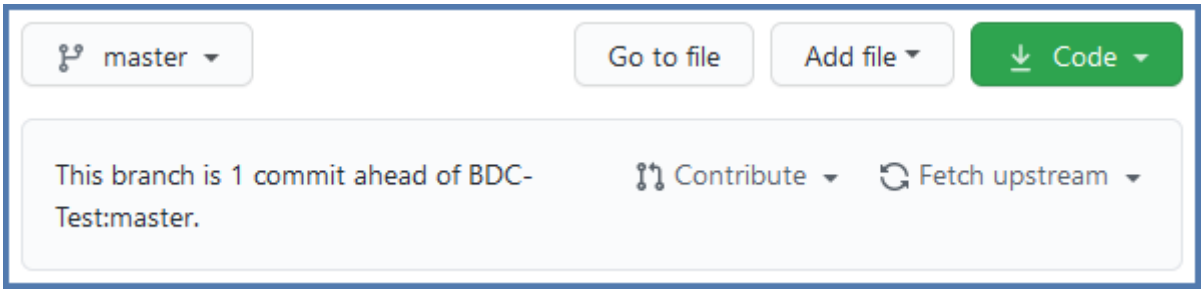


- How-to fork

To see if your repository is forked, just look at its name in the top left. If it's forked, you can see the fork icon in front of the name. And you can also see, where it's forked from, i.e. its root repository.

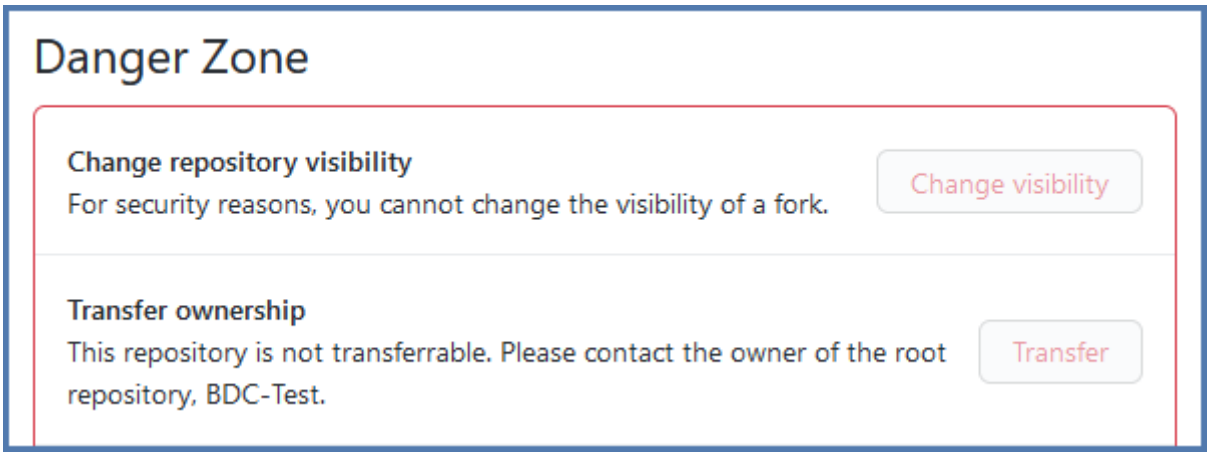


Additionally you can also see the differences between your forked and the original repository.
You can always push your changes to the original repository by clicking on the "Contribute" button and create a Pull Request to the source repository. And you can get all changes from the original repository by clicking on the "Fetch upstream" button.



- Fork's Visibility

A fork will keep the visibility of its root repository. Forks from private repositories will stay private (internal repositories are private, if you fork them into your personal space), public repositories will stay public.
Keep in mind, that you can't change the visibility of a forked repository, anymore. Neither can you transfer such a fork somewhere else.



- Changes in the Root Repository

Also there are consequences, when you change the visibility of the source/root repository of your fork or even delete it: it will break the link between the fork and its root.
It's described here in detail, what will happen in these cases.

8.1.8. Organization Limits and Constraints

GitHub Apps, OAuth Apps, GitHub Actions or any other integrations into GitHub that depends on 3rd party SaaS offerings needs to be approved by Bosch. For more information have a look on the Bosch Cloud Onboarding documentation.

Marketplace is accessible but only free apps can be installed. If you need to install an app that requires a paid subscription, please reach out to us via the BDC Service Desk.

By default, your organization will be classified as "private" organization meaning that you're not allowed to host public repositories, gists or pages. If you plan to host public content on github.com in your organization, please raise a support request. Let's try to strengthen Bosch's open source brand by moving our OSS projects into the central github.com/bosch OSS organization instead of having OSS projects spread over many different Bosch related organizations.

Table 2. Enforced GitHub organization settings

Platform	Setting	Reason
GHEC	Profile	All our private organizations need to have a common public layout to avoid leaking sensitive information as we can't hide those organizations
GHEC	Member privileges > Repository creation is set to private only	Our organizations are private by default and we need to disable the internal visibility level too as we're collaborating with external partners in our organizations and have as of now no way to exclude them from accessing our internal repositories.
GHEC	Member privileges > Allow forking of private and internal repositories is disabled	Forking private or internal repositories from our organizations into your private GitHub account has the effect that those forked repositories are no longer protected by our Bosch SSO. This means that even if a users looses access to the organization or has no active Bosch SSO session the user has still access to the forked repository. Bosch requires strong authentication via SAML for all of our Bosch data. Hence we needed to disable this feature. Forks are meant for working in the OSS environment but is not really for private development. Consider using branches on the repositories instead.
GHEC	Member privileges > Allow members to invite outside collaborators to repositories for this organization is disabled	Just like forking repositories, this feature will effectively bypass our Bosch SSO and needs to be disabled. Bosch requires strong authentication via SAML for all of our Bosch data.
GHEC	Member privileges > Members will be able to publish sites with only the selected access controls is disabled	Same reasoning as for disabling public and internal repositories.
GHEC	Organization Security > Require two-factor authentication for everyone	Strongly recommended by GitHub to protect your private account and to reduce the risk that others might hijack your account and act as Bosch employee in the public.
GHEC	Organization Security > Enable & require SAML authentication	All Bosch data must be protected by strong authentication via SAML.
GHEC	Organization Security > Team synchronization enabled	Make it easier for organizations to be compliant to our CD's regarding access management.

Platform	Setting	Reason
GHEC	Third-party access > Third-party application access policy - Policy: Access restricted	Applications must be granted explicitly the right to access our Bosch organizations.

A list of limitations that might force a project into its own organization on GHEC:

- Heavy-duty workflows that require a lot of quota from the GitHub metered products (i.e. causing a lot of possibly unpredictable costs). In that case it makes sense to set a hard cost limit and this can only happen on an organization level. Heavy-duty means something like multiple Action workflows that are running non-stop 24x7 or frequent download of large GitHub Packages (all data transferred out, when triggered by GitHub Actions, is for free). See pricing for GitHub Actions and GitHub Packages.
- Need to install a GitHub or OAuth app on organization level that can't be added to your current GitHub organization (e.g. due to security or data privacy issues). There is no general rule of thumb for that so this decision must happen on a per app base. Be aware that any GitHub app that interacts with a 3rd party provider (e.g. Travis, ZenHub, ...) requires a Bosch Cloud Onboarding before you can use them.
- Specific security needs that requires a dedicated/isolated GitHub organization.

8.1.9. Differences between BDC's GitHub Enterprise Cloud Offering and BDC's GitHub Enterprise Server Offering

With a single GitHub license you can use both solutions as GitHub Cloud and our BDC GitHub Server is linked to the same Bosch GitHub Enterprise Account. The following table shall help you to better understand the difference and find the best solution for your project.

Table 3. GitHub Enterprise Cloud and GitHub Enterprise Server comparison

Feature	GitHub Enterprise Cloud (GHEC)	GitHub Enterprise Server (GHES)
Hosting	https://github.com	https://github.boschdevcloud.com
Operator (Bosch-side)	BD/PLS3 (management of the Bosch Enterprise Account, ...)	BD/PLS3 (everything including IT operations)
Bosch Data Security Classifications	Up to C-SC2, I-SC2, A-SC2	Up to C-SC2, I-SC2, A-SC2
Data Hosting Location	USA	Europe
SLAs and Performance	99.9% uptime SLA, global availability and a strong infrastructure with a very good network performance (VPOPs and CDNs)	99.5%, for details see spec sheet

Feature	GitHub Enterprise Cloud (GHEC)	GitHub Enterprise Server (GHES)
Terms of Service (ToS)	GitHub's Terms of Service and BDC's GHE Terms of Service are applying for every user	BDC's GHE Terms of Service are applying for every user
API	API limits are defined and enforced by GitHub. Currently 15,000 API requests per hour per user and 15,000 requests per hour per repository	The **Abuse Rate Limits** are set, they limit the usage depending on the load on the server.
Large repositories and large Git-LFS files (>> 5GB)	Repository limits and Git-LFS limits are defined and enforced by GitHub	Repository size limit: 100GB, warnings at 75%. Size limitation for maximum git-object size is 50MB for the repositories (not LFS). You can't upload files with >500MB via the firewall.
Feature Rollout	Bosch has no control over when a certain feature gets rolled out by GitHub. Beta features are typically rolled out first on GitHub Cloud (github.com) and it takes around 2-18 month (avg. 4 month) until a beta features goes GA. Have a look on the public roadmap for GitHub Cloud for upcoming features	Bosch operators have full control over when a certain feature gets rolled out (by up- or downgrading the server version). Typically new features will be released in beta on github.com before they turn into GA and will be available on GHES, as well. Usually we're also in most "limited betas" for new features. Some features are exclusive to GHES. Have a look on the public roadmap for GitHub Cloud for upcoming features
Exclusive Features	Ready to use GitHub Action runners (pay-as-you-go including Linux, Windows and macOS runners), Codespaces (unclear yet if and when this feature will be also available on server)	Full control over the IT and network infrastructure by the Bosch operators. Some advanced enterprise server configurations
Marketplace	Access to the GitHub marketplace. Only free apps and actions can be installed and the Organization Owner is responsible for installing only trusted and well maintained apps and actions. Apps and actions that depend on 3rd party SaaS offerings (e.g. Travis, ZenHub, ...) need to be approved by Bosch before you can use them. See Organization Limits and Constraints for more details.	Access to marketplace is disabled.

Feature	GitHub Enterprise Cloud (GHEC)	GitHub Enterprise Server (GHES)
BIOS-approved platform	No, you can't host at the moment your BIOS projects on GHEC	Yes
Organizations	Profile of the organization is always public (name, avatar and description). All other information like repositories, members, activity are private	Completely private organizations
User Profiles	No direct control over the public user profiles and all profiles are public but the user can decide if he wants to reveal his real identity or stay completely anonymous in the public.	Bosch operators have full control over the user profiles and all profiles are private and are provisioned by Bosch. Names, usernames and email addresses are automatically set.
Authentication & Authorization	Access to Bosch owned organizations (i.e. access to issues, repositories, ...) is protected by Bosch SSO and login requires an active Bosch NT user account. Users will require to have both, a Bosch NT user account and a GitHub account. Access and permission management on GitHub Team level can be seamless linked to Bosch IdM groups	Access to the server is protected by Bosch SSO and login requires an active Bosch NT user account. No additonal GitHub account is required and users can simply login with their Bosch NT user account.
Backups & Disaster Recovery	GitHub's Terms of Service apply and deleted repositories can be only restored within 90 days. No additional backups are getting created from our side.	We run a nightly, incremental backup of the complete BDC-GitHub instance, which could be used for a disaster recovery.

8.2. GitHub Enterprise Server (GHES)

BoschDevCloud GitHub Enterprise is a git repository hosting service. While git is a command line tool, GitHub provides a web-based graphical user interface. GitHub is a collaboration platform for development in the internet and includes features like pull requests, pages, wikis, and project boards with issues. **G**it**H**ub **E**nterprise **S**erver follows very closely the feature set of the public SaaS service on github.com (also called GitHub Enterprise Cloud).

We're currently running **version 3.9**, usually with the latest patch or close to it
Additionally there might be a preview instance, running **3.10**

8.2.1. Backup

There's a nightly backup job, which creates an incremental backup (compared to the latest run) of the complete GitHub data. In case something crashes, there's the possibility to run a disaster recovery with this backup.

The restore of single elements from this backup after accidentally deletion is not part of our backup strategy. In some emergencies we might be able to run a restore on a different environment to restore the lost elements. Please get in touch with BDC-team as soon as possible.

The option *Legal Hold* is not activated, i.e. data can get deleted and will automatically be purged after a defined timespan.

Deleted repositories are available for 90 days in a trash can. If you delete it by accident, we might be able to restore it again from this trash before it gets purged, but you still loose all forks, attachments, and team configurations (see here). If it really happened, please get in contact with the BDC team, so we could support you.

8.2.2. SSH Access

SSH Access is not possible. Please use HTTPS and Personal Access Tokens.

8.2.3. API Calls

GitHub Enterprise offers the same great GitHub API you maybe know from github.com.
Please consider to chose from **REST API** calls or using the **GraphQL API**, depending on your needs. You can specify more detailed requests in the GraphQL API while you get more or less unfiltered results via the REST API.
Check the documentation to get all the details about these two possibilities: here

You could also use the gh tool to execute API requests, see here; or just use e.g. cURL.

Rest API

Please be aware that URI is different on the server compared to the cloud:
https://github.boschdevcloud.com/api/v3/ (instead of: https://api.github.com)

To access it, it makes sense to authenticate with your personal access token in an Authorization header. Please consult the documentation about the whole topic of Authentication.
A simple request for all your user's repositories could look like this in the REST API:

```
curl -H 'Authorization: token ghp_xxxxxxxxxxxxxxxxxxxxxx'
https://github.boschdevcloud.com/api/v3/user/repos
```

GraphQL

GraphQL requests are more complicated but also more powerful.
For example, a similar request your first three repos looks like this in the GraphQL:

```
curl -s --request POST \
  --url https://github.boschdevcloud.com/api/graphql \
  --header 'Authorization: Bearer ghp_xxxxxxxxxxxxxxxxxxxxxx' \
  --header 'Content-Type: application/json' \
  --data '{"query":"query {\n\tviewer {\n\t\trepositories(first: 3) {\n\t\t\tnodes {
```

```
  nameWithOwner visibility}\n\t\t}\n\t}\n}"}'
```

If you run the same request and pipe it to jquery, the output is better to read.

```
curl -s --request POST \
  --url https://github.boschdevcloud.com/api/graphql \
  --header 'Authorization: Bearer ghp_xxxxxxxxxxxxxxxxxxxxxx' \
  --header 'Content-Type: application/json' \
  --data '{"query":"query {\n\tviewer {\n\t\trepositories(first: 3) {\n\t\t\tnodes {
nameWithOwner visibility}\n\t\t}\n\t}\n}"}' \
  | jq
```

A good tool to help with GraphQL queries is graphiql

Rate Limits

Please be aware, that there are Rate Limits in place. A rate limit is the limitation of the API requests. They're on github.com as well as on the BDC-GitHub server. The limits on our BDC-GitHub are defined like this:

If you hit a rate limit, you will get an error message telling you about this issue. Please have a look at the notification. It tells you, if your personal rate limit is reached or if the rate limit for unauthenticated calls is reached. In this case it will tell you, that the requests for your IP are reached.

Additionally there are further rate limits, which might block your requests, in case the server load is very high.

8.2.4. Proxy configuration

From all devices outside the Bosch network, you probably don't need to bother with a proxy.
Inside the Bosch network, all outgoing traffic is routed via one of the Bosch proxy servers. By default, these settings are already configured, in all Bosch software (e.g. the browsers).

Using your own software (like GitHub Desktop or Visual Studio Code) needs some adjustments about the proxy settings. One way would be to install your own local proxy server, which takes care about all outgoing network traffic. There's a description about this topic in Bosch Connect.

Now you just have to check the git-settings (.gitconfig), if they're correct:

```
git config -l --[global|local|system]
```

And you could change e.g. the proxy settings like that:

```
git config --add http.proxy http://127.0.0.1:3128
```

Maybe you first need to remove existing settings, as well:

```
git config --unset-all http.proxy
```

8.2.5. Repository Upload Limit

There's a size limit for uploaded files of 50MB. Larger files might be uploaded via Git-LFS.
We're monitoring the repository sizes.

8.2.6. github.com Beta features

From time to time GitHub develops some cool new features which are available in a Beta status on GitHub.com. So the question arises, whether these features will be available in the BDC GitHub service, too.

Sadly that's not the case, as the BDC Service uses a GitHub version called "GitHub Enterprise Server" (GHES) which is a VM appliance created by GitHub and hosted by the BDC team in Azure.
At the time of writing GitHub doesn't release these Beta features in the GitHub Enterprise version.
Normally they get available in GHES soon after they reached the status "General Availability" on GitHub.com, as GitHub wants to ensure absolute stability and maturity of their GHE product.

8.2.7. BDC-GitHub Beta features

Sometimes GitHub announces new features for the GitHub Enterprise Server. Unfortunately there's no communication channel, where we can see all these announced Betas.
If we get the information about such a Beta-feature, we usually ask GitHub to activate it for our instances. If you hear something about any feature in Beta for the GHES and want (or even need) to test it, please get in contact with us and we will approach GitHub regarding that potential new feature.

8.2.8. GitHub Preview System

To realize our approach, always releasing the newest versions to our customers as soon as possible, we offer a special GitHub Preview Instance.
https://github-preview.boschdevcloud.com

On this instance, we deploy new major versions of GitHub Enterprise, before they get ready for production and we are able to deploy it on our main instance:
https://github.boschdevcloud.com.

Normally it takes three to five minor versions (patch versions), until we get the go by GitHub, that a new major version is production ready.
Until then, every BDC customer is able to try out new features on this instance and check if there are breaking changes in their workloads e.g. in GitHub Actions pipelines, before this version is applied on the main instance.
After the new version was applied on the main instance, we will take down the preview instance until the next major version. There's no backup or extended monitoring on this preview instance. If it breaks, it will be rebuild from scratch without any data.
This preview instance follows the same lifecycle steps as the main instance. Whenever we're

running a hotpatch on the main instance, we also install the latest hotpatch to the preview instance.

8.2.9. Service acceptlisting

Outgoing connections (egress) from Github Enterprise Server come from following IP address 20.103.214.16. Feel free to use this IP if you need to acceptlisting connections from it.

Incoming connections (Ingress) to our Github Enterprise Server don't have a static IP as of now due to a technical limitation, refer to this post

8.2.10. I-SC3 with GHES

GHES is designed for I-SC2 data. There are ways to configure GHES to allow the use for I-SC3 data. Here is a documentation from one of our customers how they can use I-SC3 on GHES.

8.2.11. GitHub Known Issues / Limitations

There are a few limitations for GitHub@BDC (GHES)

Check-in of large Files

According to the recommendation of GitHub, we set the maximum file upload size to 50MB. The reason is, that all bigger files should be transferred via LFS. Git is a protocol for readable code files and there are hardly any bigger text files.

Checking in a file above the limit brings an error message:

> GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com

```
:/mnt/c/nmm/GIT/github.boschdevcloud.com/bdc-test$ git push
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 60.02 KiB | 147.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote: error: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.
remote: error: File output.file is 60.00 MB; this exceeds GitHub Enterprise's file size limit of 50.00 MB
To https://github.boschdevcloud.com/bosch/bdc-test.git
 ! [remote rejected] master -> master (pre-receive hook declined)
error: failed to push some refs to 'https://09c2f2a63c9e356a5268f3ecb30d16b73d6b3539@github.boschdevcloud.com/bosch/bdc-test.git'
```

In case you'd like to push several files with a size of >50MB, it's not a problem.
But you might get the same message, in case your file sizes are above 25MB, now as a warning and not as an error message.

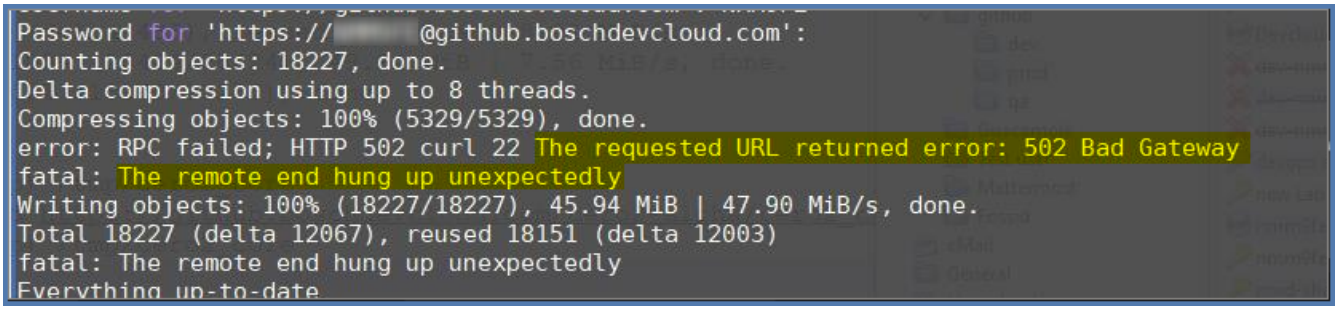
```
/mnt/c/nmm/GIT/github.boschdevcloud.com/bdc-test$ git push
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 30.18 KiB | 147.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote: warning: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.
remote: warning: File output.file1 is 30.00 MB; this is larger than GitHub Enterprise's recommended maximum file size of 25.00 MB
To https://github.boschdevcloud.com/bosch/bdc-test.git
   b057751..8bd3f80  master -> master
```

Maybe this tool (which is also mentioned in the official GitHub documentation for the cli commands) could help you to get rid of these huge files: https://rtyley.github.io/bfg-repo-cleaner/

If you want to know more on Git LFS, please have a look into our chapter on GitHub LFS.

Unable to commit existing Repositories

In case you'd like to commit a repository from an existing git-based version control systems, you might run into problems. Usually it's not possible to commit your code, because it will exit with an error message. It might look like this "502 Bad Gateway" and an unexpected hang up of the remote end:



To further analyze issues like this, it's recommended to run a check on this repository with this command:

```
1 git fsck
```

As a result, you could get outputs like this. It mentions an object-ID and the according issue:

```
1 error in tree fa762a959f60690a67eb37964bce108238afe40a: duplicateEntries: contains
  duplicate file entries
```

Up to now, we heard about these kinds of problematic repositories:

- **hasDotdot: contains '..'**
  In pretty old git versions, it was allowed to include '..' into commits. Due to security reasons it wasn't allowed anymore. But it's of course still in the repository history available, see below.
- **duplicateEntries: contains duplicate file entries**
  no further analysis done yet. But a history rewrite should help here.
- **missingSpaceBeforeDate: invalid author/committer line - missing space before date**
  no further analysis done yet. But a history rewrite should help here.
- **missingNameBeforeEmail: invalid author/committer line missing space before email**
  no further analysis done yet. But a history rewrite should help here.

Potential Solutions

Rewrite Repository History

To rewrite the repository history, it's recommended to use the git-filter-repo commands - or, if you know, what you're doing, git-filter-branch.

Change Settings on GitHub Server

If necessary, we might change the settings on our GitHub server, so that these checks won't prevent a commit. But it's not the recommended way, because the misbehavior in the repository is still available and if it will be transferred to a different GitHub server (e.g. github.com) it will appear again (and, of course, we can't change anything on github.com). Please always try to rewrite the repository history. The BDC-team needs to execute these commands on the GitHub server, depending on the error message:

```
admin@github-dev-boschdevcloud-com:~$ ghe-repo <Org-Name>/<Repo-Name>
git@github-dev-boschdevcloud-com:/data/repositories/<repo-id>.git$ git config
receive.fsck.duplicateEntries warn
git@github-dev-boschdevcloud-com:/data/repositories/<repo-id>.git$ git config
receive.fsck.missingNameBeforeEmail warn
git@github-dev-boschdevcloud-com:/data/repositories/<repo-id>.git$ git config
receive.fsck.missingSpaceBeforeDate warn
```

And after the successful import, we should unset these settings again:

```
git config --unset receive.fsck.duplicateEntries
```

All further information about that configuration can be found here: https://git-scm.com/docs/git-config#Documentation/git-config.txt-fsckltmsg-idgt

And here's the documentation about all GitHub cli commands: https://docs.github.com/en/enterprise-server@3.9/admin/configuration/configuring-your-enterprise/command-line-utilities#ghe-repo

8.2.12. Integrations

Jira Github Integration

If you like to connect your Jira project with your Github organization, feel free to request an integration from your T&R support team.

On BDC side, the only thing you have to do is, to add the service user **Atlassian BDC-GitHub - TBD4FE@bosch.com** to your designated GitHub Organization using IDM.

See also the corresponding T&R Jira Docupedia pages.

8.2.13. Roadmap for the Github Enterprise Server

We don't immediately upgrade to the next major version that becomes available because of the GHES application's reliability, performance, and availability.

Instead, we wait until releases `3.x.3` or higher in each major version (e.g., `GHES 3.7`). At this stage, the application should be fairly reliable. After testing this major version in our development and quality assurance environments, we also ask Github for advice on whether we should proceed with the upgrade or if they know any objections.

TIPP: Please be aware that there can be bugs that delay the release process, and this has happened in the past already.

We will upgrade the Github server based on the findings and the resolutions to the issues we run across during testing. Unfortunately we can't really create a solid release schedule, since GitHub doesn't publish any release dates of their software, let alone we know about all the misbehaviors they put into the new versions.

Next major version roadmap: `GHES 3.9.0` was released on June 08, 2023 and it may take some more time for the stable versions as mentioned above.

You may find more information on what to anticipate with each release in the GitHub release notes if you're interested.

8.3. GitHub Enterprise Cloud (GHEC)

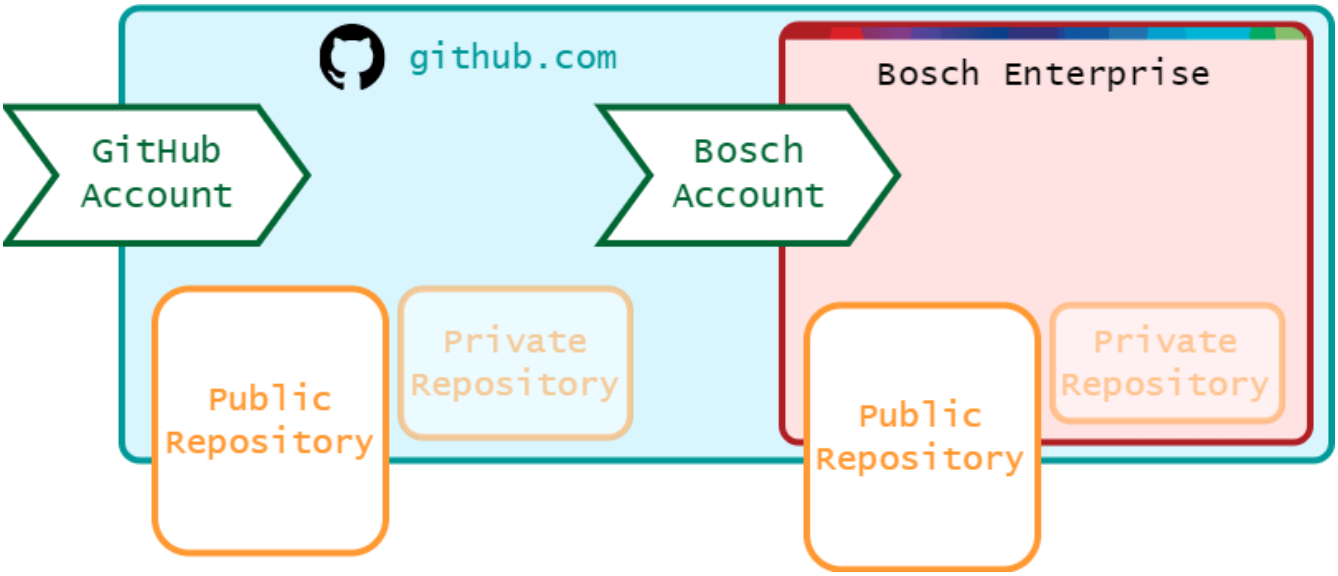
GitHub Enterprise Cloud (GHEC) is a git repository hosting service. While Git is a command line tool, GitHub provides a Web-based graphical interface. Github is a collaboration platform for development in the internet. GitHub Enterprise Cloud is a SaaS offering from GitHub. It's hosted on github.com and has been integrated into the Bosch Development Cloud (BDC). Overall, BDC provides a BDC managed GitHub Enterprise Server (GHES) service and the GitHub Enterprise Cloud (GHEC) service.

GHEC @ BDC is provided to you with □□ by the DAN SC GitHub Cloud team.

8.3.1. TL;DR

- Host your Bosch internal projects on github.com up to security class 2 (SC2)
- github.com is an open platform in the internet and you represent Bosch out there in the world. Keep in mind common sense and remember that social media can be an amazing amplifier
- Github.com Q&A
- Pricing
- How to order an own organization
- How to get access to an existing organization
- General questions regarding GitHub @ Bosch should be asked on Bosch Overflow in the BoschDevCloud category
- For any other questions or issues please create a support ticket via the BDC Service Desk, see Support and Operations > Service Desk

8.3.2. Login Process



To enter all Bosch related content on github.com you need two logins.

1. A (personal) GitHub Account
   Login to github.com with your GitHub account. Hints for this account can be found in the next chapter and also in the Q&A.
2. A Bosch Account
   Whenever you want to access an organization or repository in the Bosch enterprise, you need to validate yourself against the Bosch Active Directory. This is done with a SAML login. On a Bosch managed device this should only be one click. On any other, personal device, you need to login and provide a second factor (answer a phone call or use the Microsoft Authenticator).

By enabling the Bosch SAML (required for SSO) in our github.com organizations, GitHub will associate your `ntuser@bosch.com` ID with our GitHub user account. You can revoke anytime your active SAML session from GitHub or unlink your GitHub account from the Bosch SAML. The linked identity and your Bosch email address is the minimal information GitHub needs to store in order to provide the SAML based Bosch login functionality. Besides that you have full control over which further personal information you want to share with GitHub.

If the SAML-login is not working anymore, here's a link, how to manually logoff from the SAML provider.

8.3.3. GitHub Account

We know that many of you care a lot about the contributions shown in your GitHub □□□□. Please decide early if you want to have only a single account or a separated work account to avoid loosing your contributions that you've collect over the time within Bosch. For further information have a look on our Q&A.

Using your private-professional GitHub account is preferred but you can, of course, create a dedicated GitHub account for interacting with the Bosch organizations. It is also up to you if you want to reveal your real identity in your GitHub profile or keep your profile completely anonymous. Consider choosing an easy memorable account name as otherwise your team members will have a difficult time collaborating with you. Make sure that your account name is conform to our code of conduct (nonviolent language, ...) or otherwise you will be removed from all of our

Bosch organizations on GitHub. Using your NT user ID or year of birth in your GitHub account name is a bad choice.

Do not use your Bosch NT password for your GitHub account!

You represent Bosch out there in the world. Keep in mind common sense and remember that social media can be an amazing amplifier. Let's amplify great messages, projects and open contributions. Review the Bosch social media guidelines and policies in case you've got any questions.

Before you can request access to an existing Bosch organization on github.com the following requirements have to be met:

When activating 2FA on your GitHub account please make sure to **update your GitHub account recovery options and store your recovery codes in a safe place!** Otherwise you risk loosing access to your account, **permanently**.

- You have an active Bosch user account
- You have a GitHub.com account (see account name recommendations above)
- You have added your Bosch `ntid@bosch.com` (`tja6fe@bosch.com`) email address to your GitHub account via your GitHub account settings. It's not required that the Bosch email is your primary email address. Without setting this email address, you won't be able to join any organization. Make sure that you're using this email address for your Git commits as otherwise your contributions won't be included in your profile
- You have 2FA enabled on your Github.com account
- For BCN device users: To upload files to github.com (e.g. screenshots), you will need an additional IdM role; it's **RB_ExtendedInternet_FileSharing**, since GitHub hosts all these files on an AWS cloud. Users on Internet Clients are not affected by this limitation.

Here are some additional recommendations from our side:

- We encourage you to treat your GitHub account like your LinkedIn account: use a single account if you want to promote your community open source participation, use common sense, and help build Bosch's strong open source brand
- Make sure that you read our Q&A about leaving an organization and possible consequences to your contributions in your profile
- Please maintain a professional profile on GitHub, especially for those who plan to publicize their membership in the Open Source Software (OSS) related Bosch organizations
- Consider uploading a great profile photo to add personality to your account and community presence
- Indicate that you work for Bosch by including e.g. "@bosch" or "Robert Bosch GmbH" or "Bosch Corporation" within the company field of your GitHub profile.
- A well maintained GitHub profile can be a powerful component of an engineer's resume

8.3.4. Renaming Github.com Organization

Renaming the github.com organization will create many issues like the failure of SAML authentication and for users the access gets denied for the particular organization.

As we are dependent on other teams, there will be delays in processing such requests hence contact the support team if you want to change the name of your organization. Please get in contact with the BDC team, before you rename your organization.

8.3.5. Get access to an existing Organization

Requirements for joining an organization:

- Valid Bosch Account
- A GitHub account as described here

If you put a valid Bosch eMail-address in your GitHub account, your user will be automatically assigned a BDC-GitHub user license after joining a BDC-GitHub organization. The costs center of the user will be charged with the license and operation costs, see Pricing. With a single license you can join as many organizations as you wish, no further costs are generated.
When a user leaves all BDC-GitHub related organizations then this user won't be charged any further.

Access to an existing organization on github.com can be requested via IdM.

You can request access to an organization by assigning yourself one of the two IdM roles listed below.

Once the Access Right Owner for that organization has approved your request, you'll receive an automated invitation email from GitHub (`noreply@github.com`) that allows you to join the organization. The synchronization process might take up to three hours once your role was approved.

If you don't receive any invitation mail from GitHub, please check your junk mail or use the following link for joining your organization https://github.com/orgs/your-org-name/sso. This link will only work once your IdM role was approved and has been synced.

Available IdM roles for each organization:

- `BDC_Githubcom_orgXXX_member` - All members of this role will be the added as GitHub organization members
- `BDC_Githubcom_orgXXX_owner` - All members of this role will be the added as GitHub organization owners

GitHub has also a feature called outside collaborators but the usage of this feature is not permitted (and is disabled on organization level) as it bypasses the mandatory Bosch SSO login procedure.

8.3.6. Deleting Github.com Organization

We strictly recommend **NOT** to delete the Github.com organization even though the owners of the organization has rights to do it.

If the users want to delete their organization please send a request to BDC Team via Jira Service Desk.

Additionally, if you manually delete the Github.com organization the billing will still continue.

8.3.7. Terms of Service

GitHub's Terms of Service and BDC Terms and Conditions always apply and have priority when using GitHub Enterprise Cloud.

Bosch Data

It is only allowed to host Bosch data inside of the Bosch organizations up to a security class categorization of SC2. Security class categorizations are defined in CD-02900-001 (1.3.1).

It is explicitly not allowed to host any Bosch data that is categorized as SC1 or above inside of your private GitHub profile.

Backups

We do not provide backups and GitHub won't keep historic backups of your resources on GitHub too. A deleted repository must be restored within 90 days or otherwise it's gone for good. It is in your own responsibility to create regular backups of your important data on GitHub. See documentation from GitHub on backing up repositories.

Termination

Organization owners are encouraged to delete their organizations in case they're not needed anymore. The deletion of an organization can be triggered via a support ticket.

We preserve the right to delete organizations or particular resources of it at any time, with or without notice. Among others, the following could lead to a deletion or deactivation of your organizations or particular resources inside of it:

- Reported critical security or compliance findings are not fixed (e.g. mining crypto, not fixing critical vulnerabilities in your GitHub Actions, or using insecure GitHub apps that might put the Bosch GitHub organizations at risk)
- Invalid cost center
- Can't get in contact with the organization owners over a longer period

No backup will be created by us when deleting an organization or particular resources of it. Of course, we'll try to get in contact first with the organization owners before acting but depending on the critically we might be forced to act also without the consent of the organization owners.

8.3.8. Github.com Q&A

GitHub Accounts

Q: What do I need to consider when leaving an organization?

A: Make sure to give other organization members proper access to repository that you're owning. Maybe some spring cleaning and remove old repositories of yours? Also, make sure that to read the other Q&A's below regarding your contributions in your profile.

Q: What do I need to consider when leaving Bosch?

A: Everything from above. You may also keep your verified work email address in your profile if you wish to keep your public code contributions in your profile.

Q: Can I keep my contributions on public repositories in my profile when leaving the organization that owns the repository (changing projects, leaving Bosch, ...)?

A: Yes, see official GitHub documentation

Q: Can I keep my contributions on private repositories in my profile when leaving the organization that owns the repository (changing projects, leaving Bosch, ...)?

A: No, all your contributions will be removed from your profile. However, they'll be added again to your profile once you've regained access to the repository.

Q: Can I keep my contributions on private repositories in my profile when I'm still in the organization that owns the repository but lost access to the repository?

A: No, all your contributions will be removed from your profile. However, they'll be added again to your profile once you've regained access to the repository.

Q: I initially started working in a Bosch managed organization with my private account but decided now to create a dedicated work account. Can I keep all related historic contributions in my private profile when creating a dedicated work account?

A: No, your code contributions are bound to the email address that you've used for your commits. If you move this email address to your new work account then all related code contributions will be removed from your private profile. Contributions in form of issues on public repositories will remain in your private profile. Contributions on private repositories will be removed. Please be aware that when moving your email address to another account your historic code contributions won't be restored.

If you're really in this situation here is a workaround: Instead of moving your current work email address just use an alternative email address for your work account. E.g. keep `your.name@de.bosch.com` in your current account and add `your.name@bosch.com` to your new account. Both email addresses are valid and can be used for GitHub. Other workaround could be to ask CI to create a new email address for you.

Organizations

Q: I didn't receive the invitation email for joining my organization. What shall I do?

A: If you didn't receive any invitation mail from GitHub (`noreply@github.com`), please check your junk mail or use the following link for joining your organization https://github.com/orgs/<your-org-name>/sso. This link will only work once your IdM role was approved and has been synced.

Q: Why can't I browse public content in my organization when I'm logged in with my GitHub user but have no valid SAML session?

A: This is a known issue to GitHub but there is no fix yet planned for that. As a workaround start an incognito session for browsing the organization in case you can't login via Bosch SSO.

Q: Can I access my repositories from IoT devices during development?

A: Yes, consider using deploy keys directly on your repository. If that doesn't fit your needs you may also create a personal access token (PAT) for this purpose but make sure that you give it only minimum permissions required. Personal access tokens needs to be manually authorized to your Bosch organizations.

Q: Can I login into the Bosch organizations only from within the BCN?

A: Internet and BCN devices are allowed for login into our Bosch organizations.

Q: Which devices can be used for accessing our Bosch organizations on github.com?

A: Access to our BDC managed GitHub organizations is possible for all Bosch managed devices regardless of if they're located in the BCN (Bosch intranet) or are internet clients. If your device is located within the BCN (or SAZ), then the Bosch managed device check will be omitted and just 2FA will be sufficient for getting access to our organizations.
Managed device check gets only enforced on devices that are accessing Github from the public internet. For more details, please have a look on the EISA-ACC-103.2 and EISA-ACC-103.4. Under certain circumstances it's possible to disable the managed device check in your organization. For further information please raise an issue in the BDC service desk and explain your situation.

Additional Features

Q: What happens with additional costs related to the usage of github.com?

A: Some features of github.com include an additional usage fee; e.g. running Actions, using Codespaces, storing and transferring data from Actions and Packages, etc.
The BDC team bills the complete usage of your organization to its cost center. All organization owners can see the detailed usage of these features in the organization's settings, by clicking on "Billing & plans" or using this link link: https://github.com/organizations/<organization-name>/settings/billing

To get a rough estimate, how much cost your organization could create, you might use the GitHub Pricing Calculator.

8.4. GitHub Innersource, Opensource and BIOS support

8.4.1. Opensource on GitHub Enterprise Cloud

GitHub is a great service to host open source projects, this is fundamentally what this service was originally designed for.

How to get started

You want to host an open source project and don't know how to get started? First of all, contact the Open Source Officer (OSO) for your organizational unit. These people can guide you through the process and make sure that all checks are done.

How do I get my project on github.com?

The BDC team and the Open Source Expert Team is currently in discussion how we best govern open source on GitHub. We do have an organization named "bosch" reserved for open source. We will find the best place together with your OSO.

If you want to know more about Open Source at Bosch, you can find a lot of information at the Open Source Management docupedia page.

8.4.2. Innersource and BIOS on GitHub Enterprise Server

BIOS is the Bosch implementation of Innersource. BIOS projects are only supported on GHE Server, not on GHE Cloud.

Request access to the BIOS

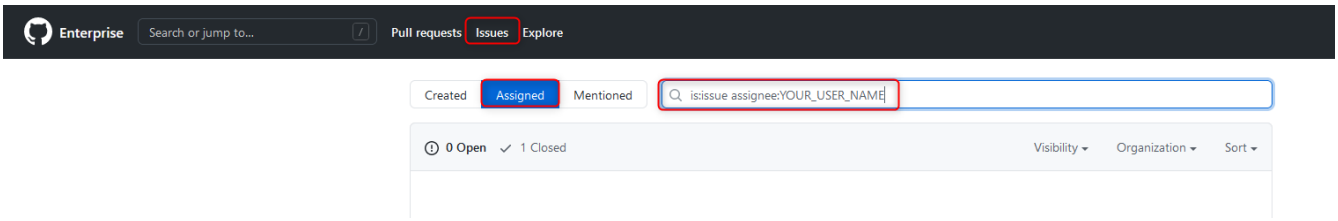
Every member of the BIOS community has access to all BIOS GitHub Organizations. This also means that if you do not have access, you are not a member of the BIOS community.

To onboard you as a member of the Github BIOS community, we're relying on the onboarding process of the Social Coding platform; i.e. you should get access to this platform first.

Here's a document, that describes all the steps needed to become a member of the Social Coding community.

Accept BIOS Policy in GitHub Enterprise Server

After you log in to GitHub for the first time, you will get a GitHub issue assigned. This issue contains terms and conditions regarding the BIOS policies. You may find the issue quickly, as shown here:



The query string is: `is:issue assignee:<YOUR-USER-NAME>`
You must also read and close this issue and act accordingly. Only after this issue has been closed,

you will have access to BIOS organizations and repositories.

Here is the direct link: https://github.boschdevcloud.com/issues?q=%22ACTION+REQUIRED%3A+BIOS+Policy%22+assignee%3A%40me

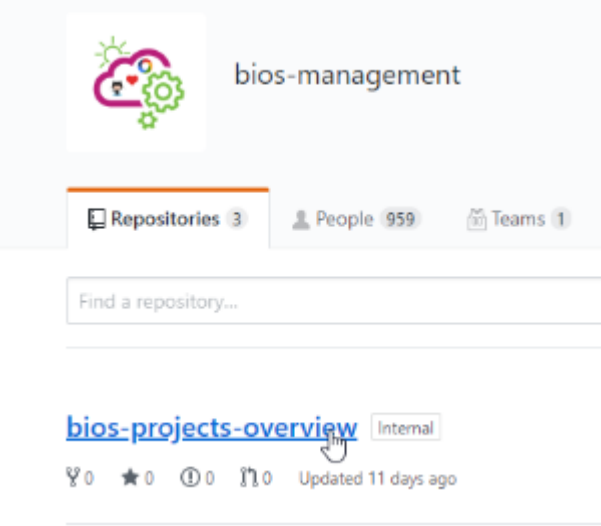
How to create and work with BIOS projects in GHES

For each BIOS project, we're offering a dedicated GitHub Organization. You need to name at least one responsible person for this Organization, who will get all administrative rights for it, and you're completely responsible for this organization on your own.

This means you can - and you have to - take care of all configuration settings.

Please do not remove administrative accounts from us like `bdcadmin`, `bdcautomation`, `bdcautomationbios` and `bdcmonitoring`.

If you just wanted to use other BIOS Projects that are available and collaborate, visit this bios-projects-overview repository for a list of already-available BIOS Organizations. The `bios-management` Organization is the central place for BIOS related information.



Permission Management for BIOS Projects

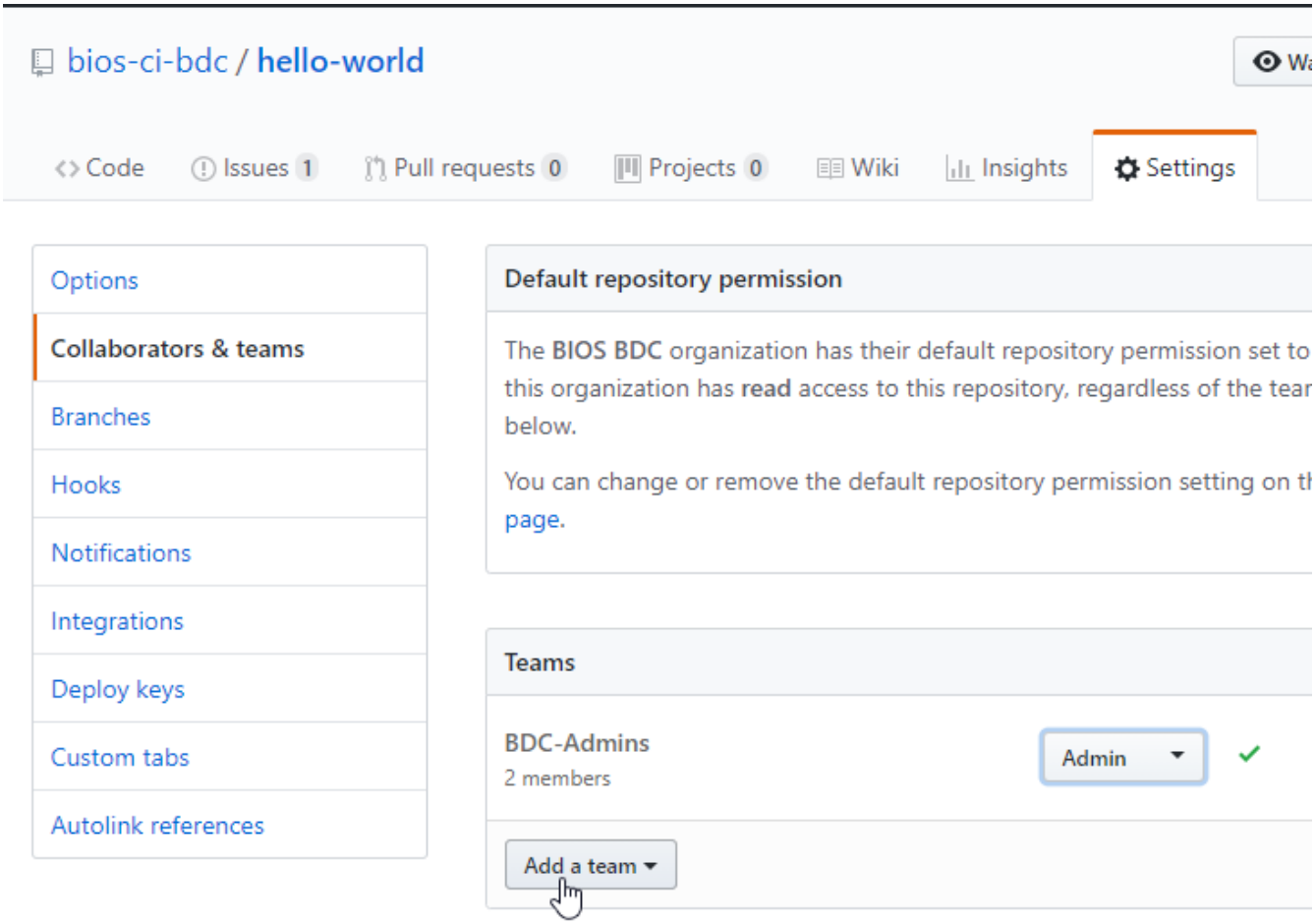
Relevant for Users:

- A BIOS user can access all BIOS projects, and if they wish to collaborate, they can ask the owner of the organization to add them as members.

Relevant for organization owner:

- Organization owners are free to create as many repositories as needed.
- To ensure that all BIOS members get access to all BIOS repositories as defined in the BIOS rules, the repository visibility must be set to "internal". The only exception of this rule is the special repo .github, which might be set to public.
  Please be aware of this setting, when creating new repositories.
- To grant repository permissions to your colleagues, you can create teams in your Organization. These teams can be assigned to repositories with different permission like write, admin or

maintain. Just go to your repositories settings to set team permissions accordingly. Refer to the public github documentation about the permissions for more information.

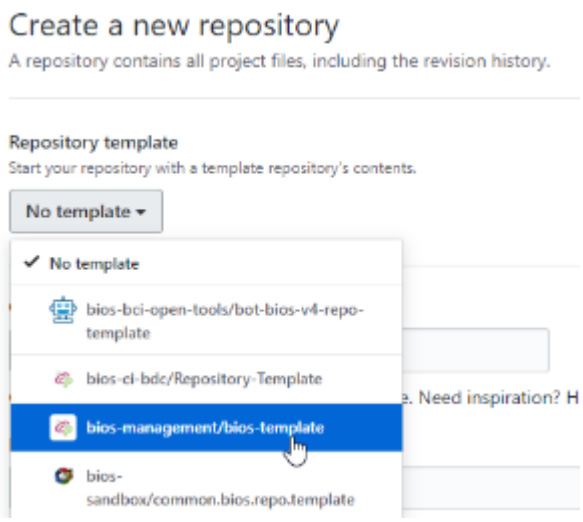


BIOS License

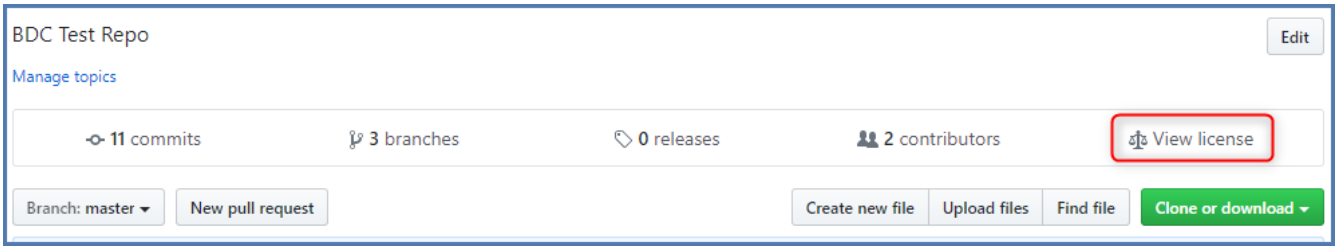
All Bosch Internal Open Source repositories have to include the current BIOS license information. You can find the current BIOSL v4 - Bosch Internal Open Source License v4 on this docupedia page: BIOS License V4

By creating a new repository, you've the option to create it from an existing template repository.

We have a template you can use which includes the license file: https://github.boschdevcloud.com/bios-management/bios-template.



Or you can easily add a file called LICENSE.md to your repositories and it will be recognized as its license. You can see a "View license" button on the repository's main page, as soon as you added a license file.



Ordering a BIOS-orga

Please use our BDC-Portal to order a bios-organisation. You need to supply the following information:

- name of the orga: It has to start with "bios-"
- account(s) of repsonsible person(s)
- account(s) of org owner(s)

8.5. GitHub Actions

8.5.1. Actions

> GitHub Actions makes it easy to automate all your software workflows. Build, test, and deploy your code right from GitHub. Make code reviews, branch management, and issue triaging work the way you want.

Besides this marketing slogan you can find the complete and official documentation at https://docs.github.com/en/enterprise-server@3.9/actions

Actions are workflows that execute a list of task defined in a YAML (.yml) file in your repository. They're stored in its .github/workflows/ subfolder.
To define your own workflow.yml, just create the directory `.github/workflows` and write a .yml file

in there. Or you could simply use the "New Workflow" button in your repository. Then the new workflow file will open automatically and in the correct directory.

Organisation and Repository Settings

All existing GitHub users can ask their Organization-Admins to activate this feature. An owner just needs to go to the organization's "Settings" tab and check for the "Actions" option on the left. Here the owner can configure on which repos actions may be used.

As soon as this Actions feature is activated on your repository via the parent organization you're almost ready to go.
In the Settings of your repository, you can see the Actions option in the left menu and you now you can decide, if you'd like to use Actions in your repository.

Action Runner

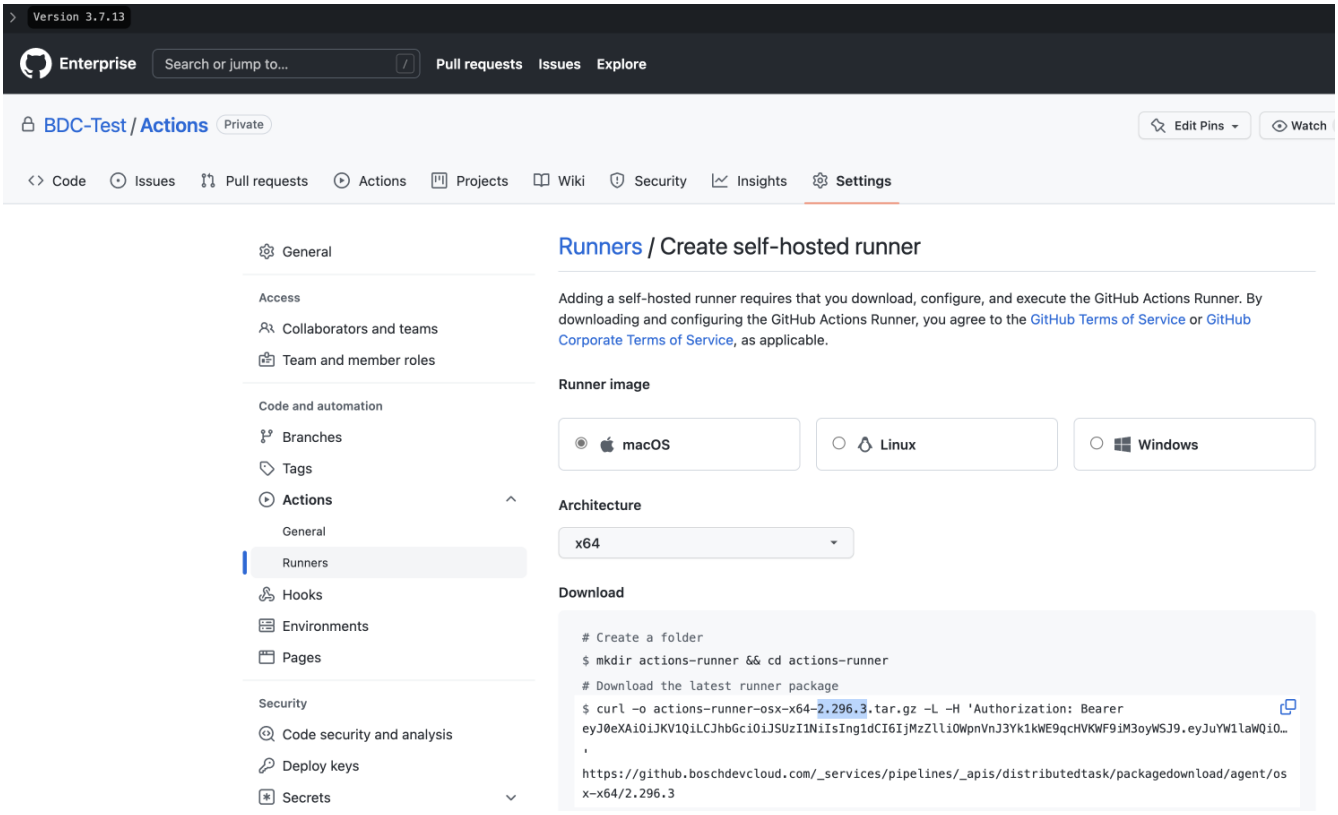
Remember: you need a "Runner" (i.e. an agent), which executes your action. In your repo or organization you can add a new self hosted runner via the action settings. All code to install and configure your runner agent will be shown to you. For details, please see the next chapter.
As an alternative, you might use our "Runner as a Service" offering.

See adding-self-hosted-runner and using-self-hosted-runner for more details on how to setup your runner.
After this, be sure to add one or more labels your runners as you would expect them to be.

github/actions is a collection of templates maintained by open source community, which you can use in your workflows.

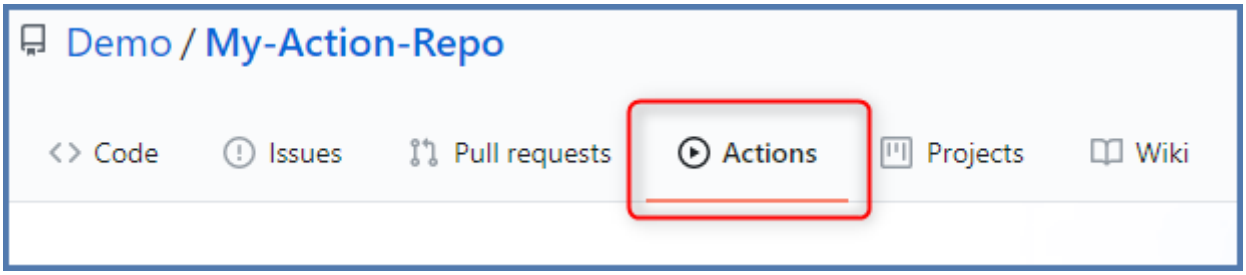
Runner version

Each GHES version ships a default runner version, which can be found, in the runner creation page of the GitHub GUI.
To get there, you have to open the settings page of any repo and click on "Actions" → "Runners" → "New self-hosted runner".



Workflows

Maybe you know the concept of workflows already from other tools, where they might have different names, e.g. "pipelines". GitHub calls these files, which define their automatisms **workflows**.
There are no local actions/workflows provided by the BDC-team on the BDC-GitHub server. But we activated the GitHub Connect feature - therefore you can (technically) use all available actions from the GitHub Marketplace - and there are literally thousands! Just look up, how to use the action in its documentation, which you'd like to use in your workflow.
And remember to check the legal aspects, as well (e.g. their licenses).

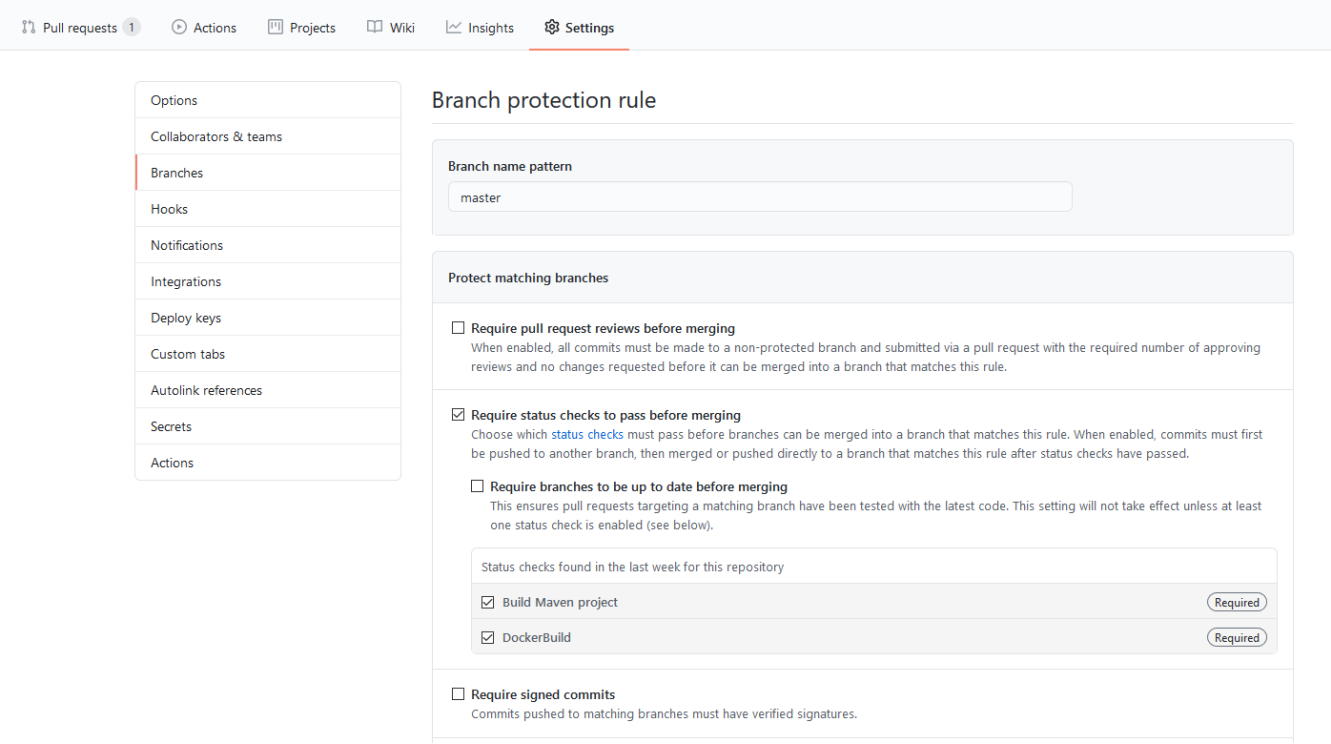


Unfortunately there's a small difference between the github.com workflows and the workflows in the BDC-GitHub: On github.com you're automatically connected to the GitHub marketplace. In the BDC-GitHub you can only see some workflows from the "Starter Workflows" package. But the advantage is, that these workflows are automatically adapted to your repository and your runners. The branch name is changed to the "main/master" branch and the self-hosted label is written into the "runs-on" option:

```yaml
# Controls when the workflow will run
on:
  # Triggers the workflow on push or pull request events but only for the main branch
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]
```

On the right side of your screen, you will see more information about the possibilities in these workflow yaml files, including links to the according documentation pages.

By using branch rules you can even define that all tests have to exit successfully for a pull request to be mergable.



Versions

In your workflow files, you define, which version of the action you want to use. The recommendation is, to define it as specific as possible. The best option is the usage of a SHA, see Security guide. In this way you can be sure, which code your workflow will execute.
As an example you could use this line to use the SHA, which introduced version 2.6.0 of the checkout-action:

```yaml
    # Steps represent a sequence of tasks that will be executed as part of the job
    steps:
      # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
      - uses: actions/checkout@dc323e67f16fb5f7663d20ff7941f27f5809e9b6
```

If you specify a very general information, like `@v2` it will take just the latest available version of this action. In this case it wouldn't be 2.6.0, but 2.7.0. This could lead to unwanted behavior, in case

something changed what you haven’t tested; or in the worst case this latest version could even have a bug and can’t execute at all.
You can find all available versions of each action in the "Releases" section of that repository, maybe you have to switch to the Tags; it depends how that action is handling releases.

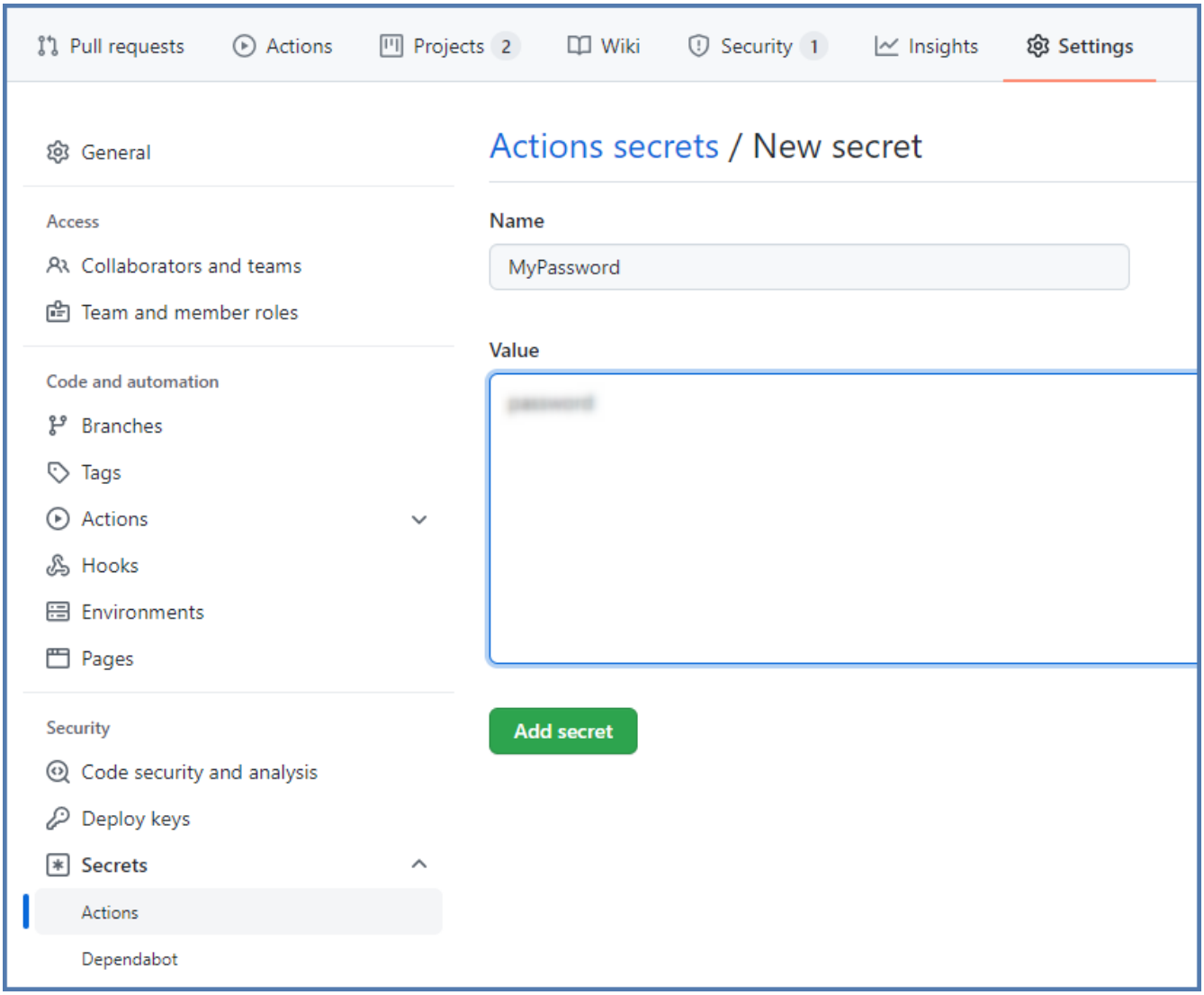
Github Enterprise Server ships a default set of actions which are available in the /actions as well /github organization. These actions come with a specific version, depending on the GHES version.
As these versions are behind their original ones on Github.com, we implemented a synchronization job, which updates all these actions to the most current one.

This synchronization happens every Monday at 5 PM CET.

Secrets

Of course, you should never write passwords hardcoded into these workflow files (or any other file). Therefore you could use the **Secrets** functionality of GitHub. Similar to other tools, you can store e.g. usernames and passwords in a secure way into a GitHub vault. This Secrets vault is in the repository or organization settings just above the Actions. You could define secrets for a single repository or put them in the vault of an organization. This makes them available in all of their repositories, but for each of these secrets you could restrict its usage and define its scope.

N.B. you could use the Dependabot to check for tokens/secrets in all files of your repositories, just go to the repo’s security tab.



8.5.2. GitHub Runner (self-hosted)

To use Actions, you need so called "Runners", or better "Self-hosted Runners". Depending on your needs, you might provide the runners on the organization level for all your repositories - or provide dedicated runners for each repository.

We guess, it would make sense in many cases, if you provide the runners on the organization level.

Containers might have some restrictions (i.e. you can't run docker inside a docker container), therefore GitHub doesn't offer them, yet. You have to use a virtual machine as a runner, and probably all common operating systems are supported. As long as you don't need to access any data or services inside the Bosch network, you could get just any Azure VM, and install & configure the runner code on it (see next chapter).

If you need access to data and/or services **inside** the Bosch network, you should take a look at the SL4 zone. You could refer to the documentation from BCAI on how to set it up.

BDC offers a managed, self-hosted runner service called Runner as a Service. You could take a look.

There's a good and comprehensive documentation from GitHub about the runners, here's just a quick summary.

Security

Please take a look at the Security Hardening chapters of the runners. And please take it seriously!
If your runners are publicly available on the internet, they're at a much higher risk than all machines inside the Bosch network.

Installation

To install a runner and connect it to your organization (or repository), you just go to the org's or repo's settings and scroll down in the Actions chapter, in there is a sub-chapter for the runners. Going there, you see a green "Add Runner" button. In case you have already an existing runner, you might click on the green "Add new" button.
Clicking this buttons shows you all the needed steps for your desired operating system, which you have to execute on your virtual runner machine:

- download the runner's code
- extract the downloaded code
- configure the runner, which includes:
  - installing the code
  - putting labels (i.e. tags) to the runner
  - set the runner application as a service

On Windows runners you need to install the agent software in a useful user account, i.e. don't use the proposed system accounts like NETWORK-SERVICE or any other internal system account. All tasks on the runner are executed by this user account. It's also recommended to use a folder in the file system's root directory to avoid potential "file name too long"-errors.

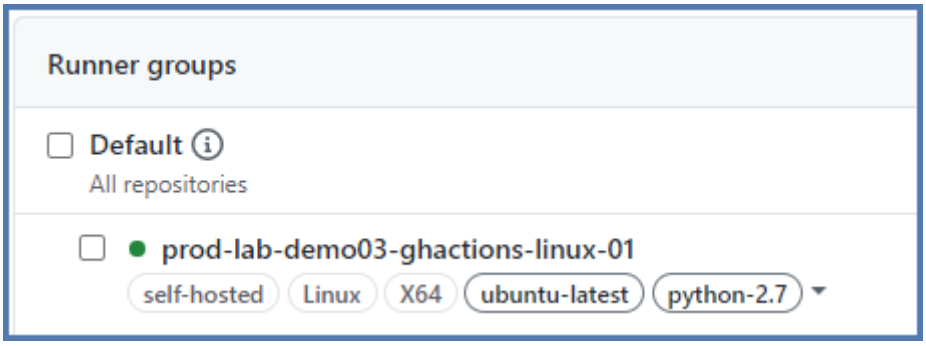
Update

The runners should auto-update, in case a new GitHub software version is used on the BDC-GitHub and as long as they have a connection to the BDC-GitHub server.

If you want to disable the auto-update, you have to specify the `--disableupdate` parameter, while executing the runner's `./config.sh` script.
See also https://github.blog/changelog/2022-02-01-github-actions-self-hosted-runners-can-now-disable-automatic-updates/

Labels

Your runners will automatically get some default labels (or tags). It might be useful for your workflow's yaml code to add some good additional labels. Just click on the little triangle next to the last label to see the "Add" option. During the runner's provisioning it already gets the three default labels "self-hosted", "Linux"/"Windows"/"macOS" and "X64"/"ARM".



Groups

You might have the need to restrict some of your organization's runners to specific repositories. Groups offer the option to restrict access to their assigned runners for defined repositories.

Runners for multiple Organizations

You can define and set up your own runner(s) for your organization and define its scope for all the containing repositories. You can't setup a runner for several organizations. In case you need one runner for more than one organization, you currently have two possibilities:

- install several runner agents, one for each organization. It's possible to have multiple runner agents running on your self hosted runner. Just don't use the same directory for the runner agent.
- if it's not too urgent, you could wait some time. There's a project going on to provide an API for your self-hosted runners for multiple organizations.

Support

Unfortunately we can't help you with the installation of your self-hosted runners. If something's not working as expected, maybe you can find a solution in Bosch Connect, or in Bosch Overflow. Other people at Bosch set up their runners successfully, maybe they documented it somehow for all others or can help you with your questions.

8.5.3. Managed Github Actions runner for BIOS community

We provide a Github Actions Runner group for our BIOS community.
The Action Runner Admins from the BIOS community voluntarily manage those runners.

To use the BDC runner, configure the actions runner group on all repositories in your organization

Org Settings → Actions → BIOS-Runner → "Edit repository access" → All repositories

The available Github runners are displayed in the repo settings.

Details about the Runners and their labels may be found here.+

Please execute all your workflow steps inside a docker container to not influence the pipelines of other BIOS colleagues

Examples

Check our workflow examples here

8.6. GitHub Advanced Security

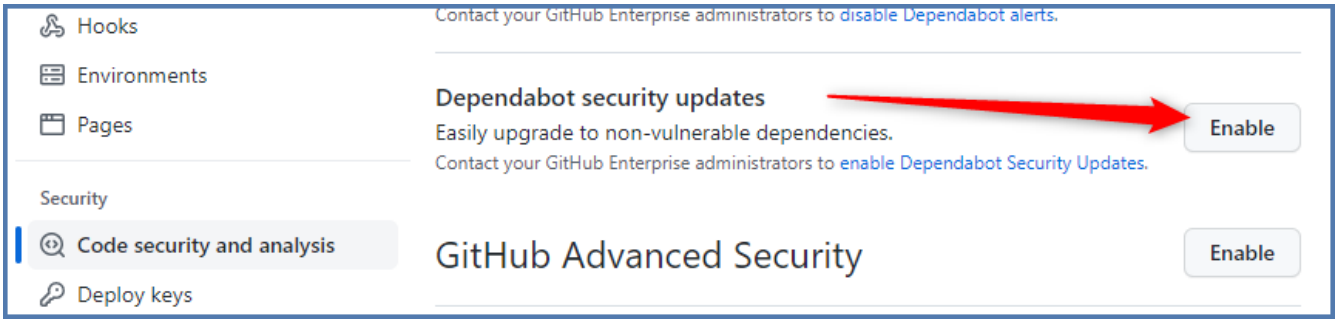
GitHub Advanced Security (GHAS) is the service portfolio around all topics of improving security and to enable developers to improve security while they do their work

8.6.1. Dependabot

With the GitHub Dependabot feature you can automatically check for insecure and/or outdated dependencies in your code.
All information about it can be found in the official documentation from GitHub or browse to their blog post about this feature.

We set up everything on the admin side for you to use it.
And we put the `dependabot` label on the BIOS runners.

You only need to activate this feature in the settings of your repository. Go to the chapter "Code security and analysis" of the settings (if you're repo admin). Now you can enable the "Dependabot security updates". It's a good idea to check the "Access to alerts" further down on this page and decide, which additional people should have access to your security alerts.



In the next step you need a runner with the `dependabot` label (and `linux` and `self-hosted`, they're not case-sensitive) available for your repository. All dependabot jobs will be executed on this runner. You probably need some additional packages on this linux runner. It's hard to describe, since it changes from time to time and maybe for different languages. To update a python dependency we needed to provide a working docker installation on the runner. If your dependabot action is not running successfully, you might get helpful error messages.

Whenever you make changes to your dependency list, this action will run and check them for security incidents.
And every insecure dependency will end up in a automatically generated pull request in this repository with a change in the version of the dependency.

If you want to run frequent checks on your code, you need to add a dependabot.yml file in your code. Take a look here at the GitHub documentation about it.

8.6.2. GitHub Dependency Vulnerabilities Check - Dependabot

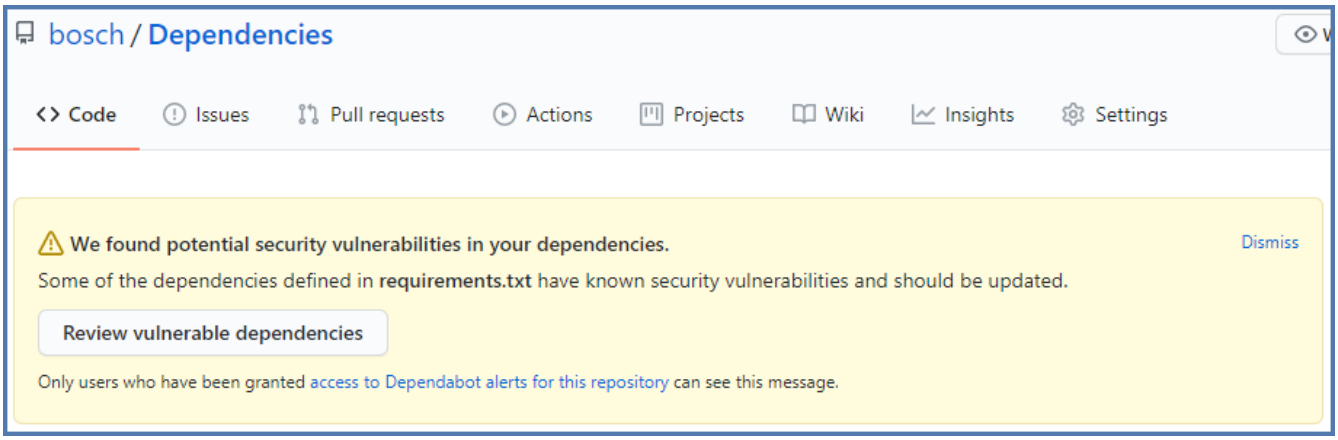
BDC-GitHub is able to automatically check your code repositories for vulnerable dependencies!

To achieve this, GitHub retrieves all CVE information regarding insecure and outdated dependencies every hour. All checks are executed locally on our GitHub Enterprise Server, so no line of code leaves our instance.

The best thing: it works completely in the background and you don't have to do anything to get your code analysed.

Here's the documentation, where they describe, at which dependencies they look: https://docs.github.com/en/enterprise-server@3.9/github/visualizing-repository-data-with-graphs/about-the-dependency-graph Especially the chapter "Supported package ecosystems".

As soon as you have a dependency with a known security vulnerability in your code, you will get such a message on the repo's main page:

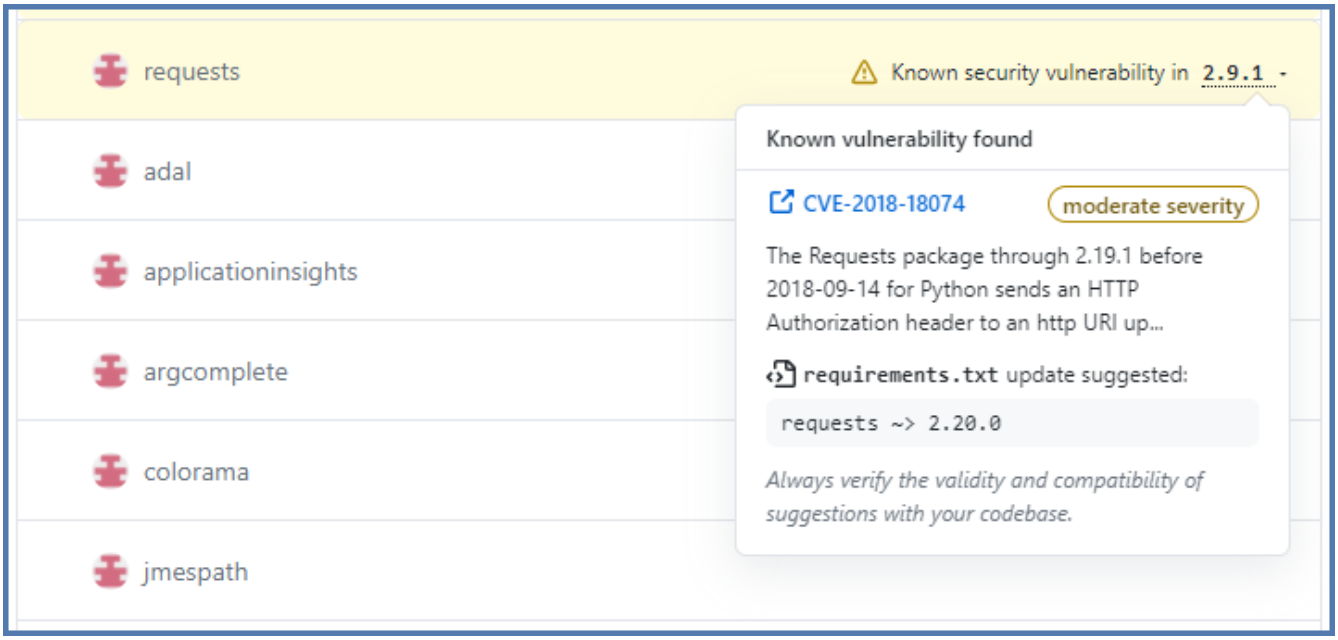


The same message also appears during each commit, which introduces such issue(s). If you're using the git commands in a terminal, you will also get a warning in your push operation:

```
C:\GitHub\Dependencies>git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 894 bytes | 298.00 KiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 2 local objects.
remote:
remote: GitHub found 11 vulnerabilities on bosch/Dependencies's default branch (7 high, 4 moderate). To find out more, visit:
remote:      https://github.dev-boschdevcloud.com/bosch/Dependencies/network/dependencies
remote:
To https://github.dev-boschdevcloud.com/bosch/Dependencies.git
   51c28f8..ac79d35  master -> master
```

By clicking on the "Review vulnerable dependencies"-Button it will show you all files with security risks; you might also use the "Insights" of the repository's menu bar.

It also gives you a popup window with some basic information about the issue, a link to the according CVE, and a suggestion, how to fix it:



By default repository and organization owners only can see this yellow alert message. But you could give regular repository members or teams access to this message as well. Just click on the link of the message or use this one:

https://github.boschdevcloud.com/<org>/<repo>/settings/security_analysis

Every user can define in their notification settings, in which way they will get this information. It's always a good thing to check these settings from time to time and maybe remove some unneeded or unwanted notifications.

8.7. Github Copilot

GitHub Copilot is the AI solution from GitHub. You can find all information about it here.

Do not to use the private Copilot option at work!

8.7.1. Copilot at Bosch

GitHub Copilot is a powerful tool which has to be used in the right way. Please evaluate your business needs carefully and use GitHub Copilot in a way that it is beneficial for your project and not put you in legal risk.

There is also a generic approach to evaluate such AI based solutions, check out the Maturity Level Model for AI based programming companions. This will provide some guidance on how to use Copilot in your project. In addition, there are further details directly related to GitHub Copilot available.

Copilot requires you to use a github.com account to verify if you are allowed to use Copilot or not. The enablement of Copilot requires your account to be a member of a GitHub Enterprise Cloud Organization.
Those are the key requirements from a product perspective.

We created a GHEC organization which we use solely for the purpose of granting the permission to

use Copilot.
The membership for this org is managed via an IDM role and your manager has to approve the role assignment. The associated costs are to be covered by your cost center. Please check the GitHub pricing chapter for detailed cost information.

8.7.2. Risks and mitigation strategy

The following table provides some insights into identified risks. Please perform a risk analysis for your specific business case before you use GitHub Copilot.

Risks	**Mitigation options**
GitHub Copilot learned from publicly available code and accidently reproduces code snippets from this code. If the used code is not hosted Github, the GitHub snippet filter will not warn you. This may lead to infringement of copyright of third parties	GitHub Snippet Filter is activated centrally. Use Snippet Scanner (FossID). Snippets that are in the knowledge base will be detected.
Risk of dissemination of confidential information	The contracts with Microsoft and GitHub contain adequate confidentiality clauses. Data up to SC2 can be uploaded. Use cases involving the processing of personal data should be aligned with responsible DSOs and C/ISP.
Code generated by AI is not protectable by copyright	Protect created code by adequate measures (to be treated as trade secret). If AI generated code is used in customer projects: make sure, that is allowed under the contracts with the customer.
US Class Actions against providers of Foundation models may lead to a ban of certain LLMs.	Don’t rely to much on the availbality of GitHub CoPilot. Mark AI code for the unlikely case that litigation will de directed towards users of LLMs.
Determine the Risk of using Copilot for OE Product Portfolio	Decide if and for which use cases Github Copilot is allowed.
Bad code quality is injected in the projects by GitHub Copilot	Ensure that the software development processes and guidelines are also valid for AI generated Code. AI tools can produce wrong or misleading information. Do not solely relay on AI tools for making critical decisions. Ton esure the quality and reliability of the output produced by Copilot, we recommend conducting careful and independent quality checks.
Security vulnerabilities are injected by AI generated code.	Security processes and assessments are also valid for AI generated code

Risks	Mitigation options
Copyright protected Code Snippet is smaller than 150 characters and will not be detected by GitHub filter. (e.g implementation of inverse square root algorithm)	Mandatory to use Snippetscanner (FossID). Snippets that are publicly available in the internet AND in the Scanners Database will be found.
In case of legal disputes it is not transparent which code was created by a developer and which code has been generated by Copilot.	Decide for each project with C/LS if documentation and which degree of documentation is necessary. A tool that automizes this documentation is currently under evaluation.
Customer contract includes obligation for Bosch to grant copyright protected code to customer.	Use of GitHub Copilot has to be approved by customer and AI generated code has to be marked. Supplier have to be obliged to provide this information.

8.7.3. Copilot for Business

There are two versions of Copilot: Private and Business. Keep in mind, that there are differences between the "Business Solution" and the private usage of Copilot:

- There are different Terms of Service including such things as defense of third party claims: According to the specific terms for Copilot for Business, GitHub takes over indemnification coverage for Copilot for Business customers against potential third party claims.
- Persistence of code snippet data: In GitHub Copilot for Business, GitHub does never save or use any of the prompts for product improvements or similar. All code-related prompt and response data is garbage collected as soon as we have given the suggestion back to the developer, and as such business customers can be sure that their code is not used for any training purpose etc.

As you can see, some additional "enterprise-focused" thoughts went into the Copilot for Business variant, that may not show in the individual developer experience but makes a big difference in its usage.

To implement these terms technically, we had to **disable** following two options in the Robert Bosch GmbH Enterprise, which effects all GHEC Copilot users:

- **Suggestions matching public code**
  GitHub Copilot can allow or block suggestions matching public code. See the GitHub Copilot documentation to learn more.
- **GitHub Copilot Chat**
  Grant beta access for all organizations with access to Copilot within this enterprise. If allowed, organizations can access GitHub Copilot Chat feature and you agree to pre-release terms.

The first option controls, if Copilot can suggest code, which is publicly available.
If this option is disabled, Copilot uses filter, which prevent, that already existing code snippets are suggested that may be under OSS or other licenses and bring risks to your development.

The second option controls, if the GitHub Copilot Chat feature (another plugin for your IDE), might

be used.
We disabled this option as this feature is in a Beta phase currently, where **pre-release** terms apply which differ from the normal Terms of Service and allow Github to collect usage data.

Whenever Github publishes the Copilot Chat feature as GA in conjunction with the business terms of service, we will activate it.

Do not to use the private Copilot option at your workplace!

8.7.4. How to get started

Short summary GitHub Copilot is an extension to your local IDE, e.g. visual studio code. You need to have a valid license in order for it to work. The following instructions will guide you through the required steps to get your license and to enable it in your IDE. The license is managed via the GitHub Enterprise Cloud, this is the reason while some of the required steps have to be performed on github.com.

▼ *Prerequisite: setup a GitHub account (for new users only)*

Setup a GitHub Account

- Username - Please pick a publicly & humanly readable username! Not your Bosch ntuser!
- Create the account with your private email address.
- Log in.
- Go to https://github.com/settings/emails
- Add ntuser@bosch.com
- Verify the email address
- Activate two-factor authentication - https://github.com/settings/security

---

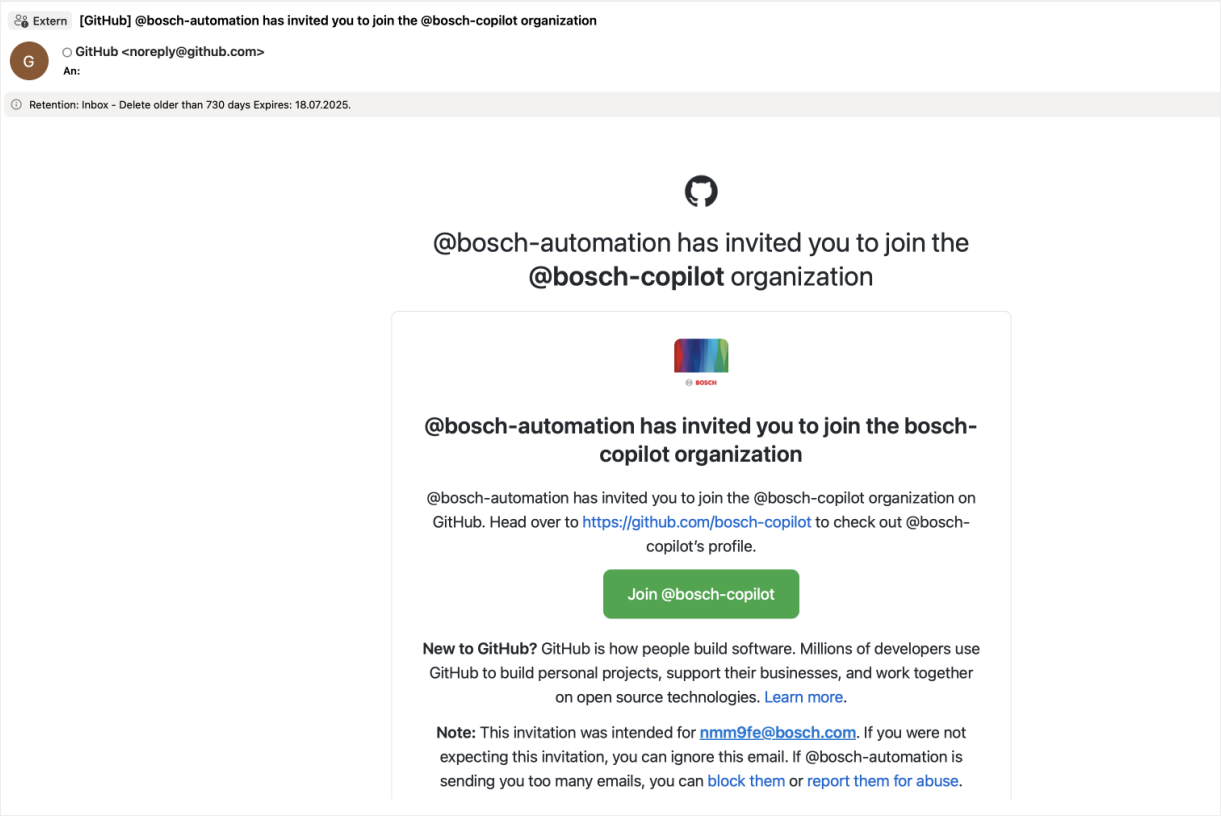
▼ *Step 1: Request the required IdM access right*

To get a Copilot Business license, your user needs to get part of the Bosch Copilot organization. All you have to do is to request the following IDM role: **IDM2BCD_BDC_Githubcom_CoPilot** in the Target System "Bosch Development Cloud". Your manager has to approve the role assignment.

You can do this via User Self Service. Select Bosch Development Cloud on the left side (or enter it in the top left field if you have no BDC roles assigned so far). Then search for the role and assign it to you. Don't forget to send the request at the end (with a good justification for the approver to quickly approve it).

▼ *Step 2: Invitation email*

As soon as your IDM workflow was completed and your user got synced, you should receive an email about the invitation to the organization:

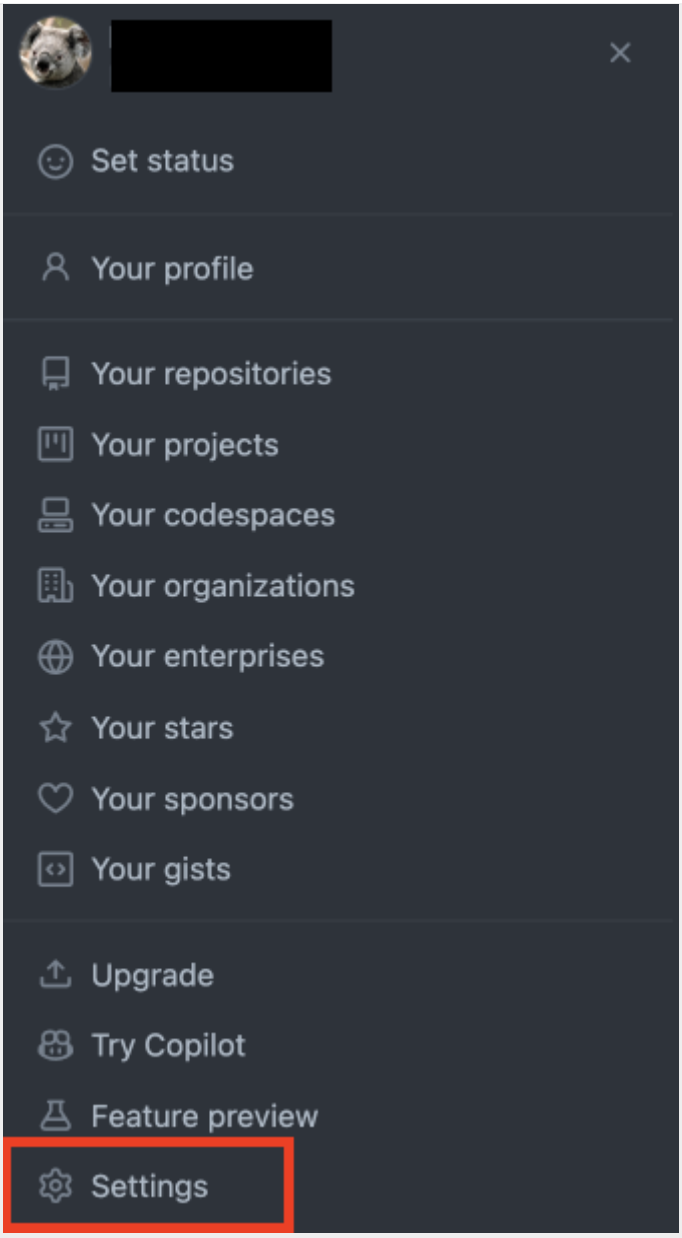


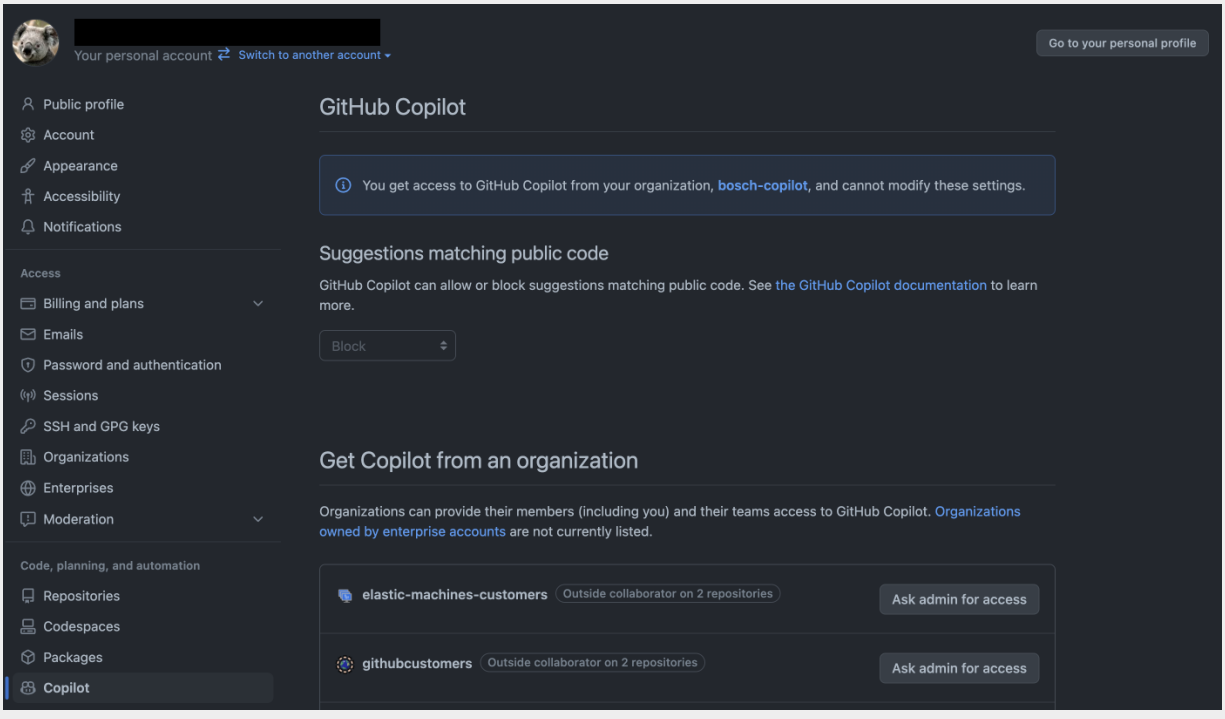
You have to accept this invitation, to get added to the Copilot organization and to receive the license.

In case you didn't receive the email or you deleted it as spam, you can also join directly, but only after your IdM assignment is completed.

▼ *Step 3: Verify the license*

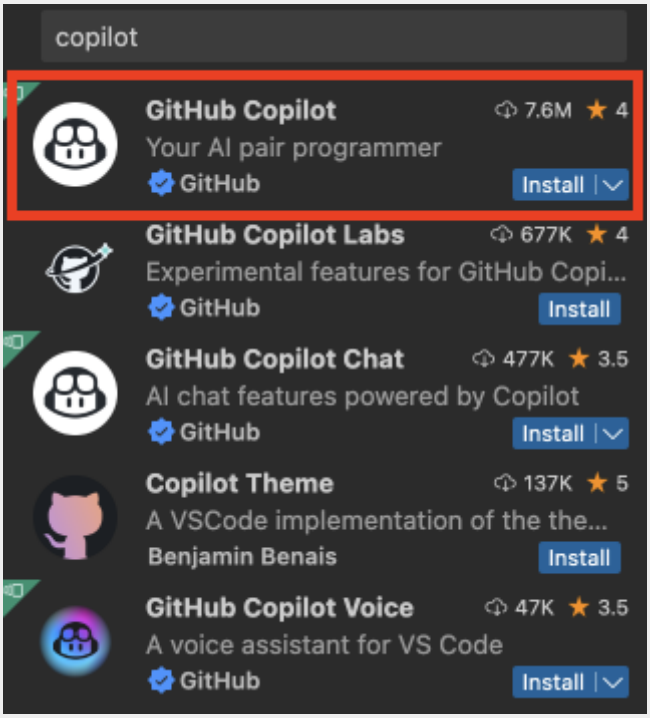
After you got access to the Copilot org, you should get a Copilot license automatically.
You can check this in your Profile settings as follows:


Set status
Your profile
Your repositories
Your projects
Your codespaces
Your organizations
Your enterprises
Your stars
Your sponsors
Your gists
Upgrade
Try Copilot
Feature preview
Settings

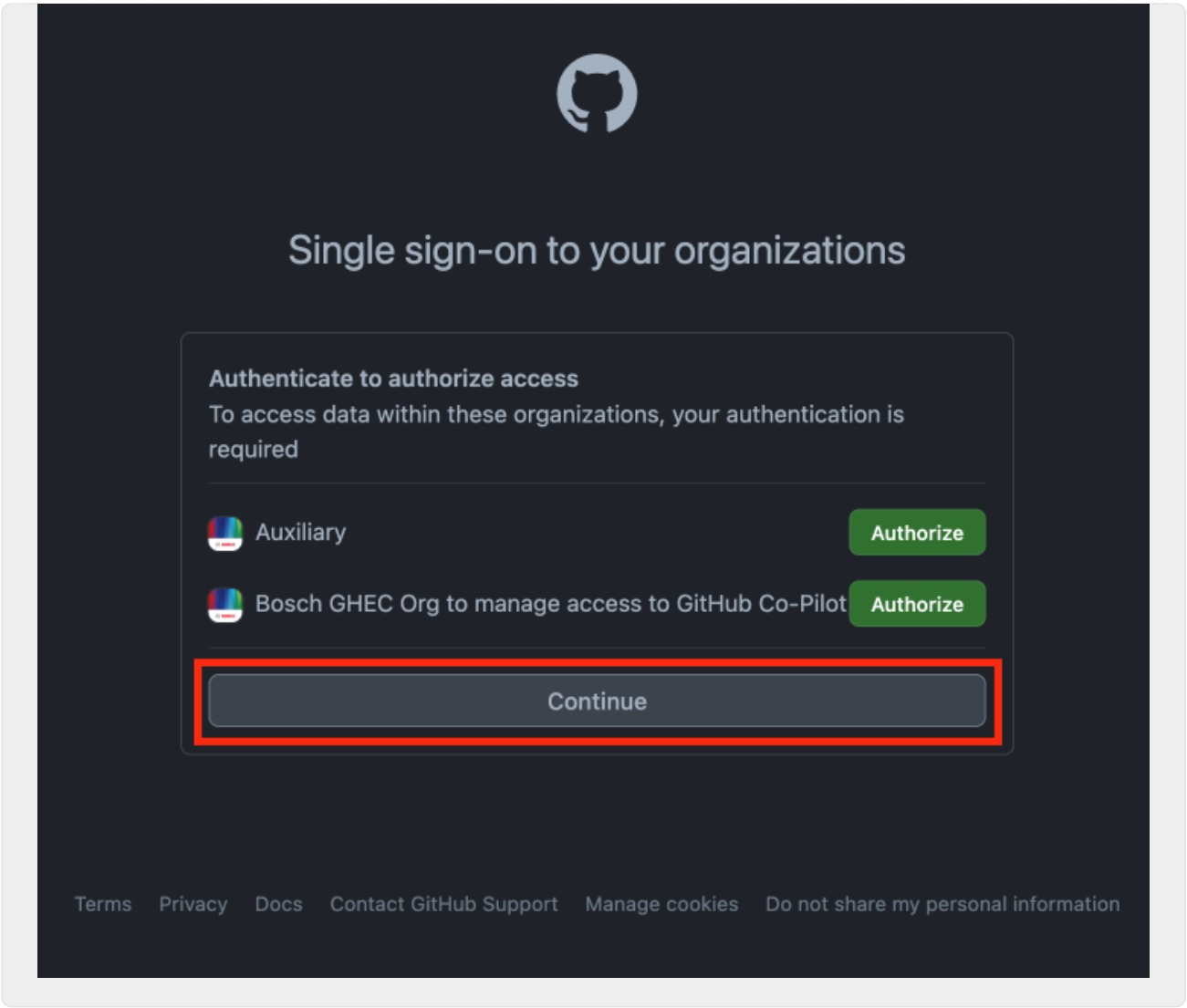

Your personal account
Switch to another account
Go to your personal profile
Public profile
Account
Appearance
Accessibility
Notifications
Access
Billing and plans
Emails
Password and authentication
Sessions
SSH and GPG keys
Organizations
Enterprises
Moderation
Code, planning, and automation
Repositories
Codespaces
Packages
Copilot
GitHub Copilot
You get access to GitHub Copilot from your organization, bosch-copilot, and cannot modify these settings.
Suggestions matching public code
GitHub Copilot can allow or block suggestions matching public code. See the GitHub Copilot documentation to learn more.
Block
Get Copilot from an organization
Organizations can provide their members (including you) and their teams access to GitHub Copilot. Organizations owned by enterprise accounts are not currently listed.
elastic-machines-customers
Outside collaborator on 2 repositories
Ask admin for access
githubcustomers
Outside collaborator on 2 repositories
Ask admin for access

▼ *Step 4: Enable the extension in VSC*

To use Copilot you just need to install the corresponding extension in your IDE (e.g. Visual Studio Code), after you acquired a license as described in the prerequisites above.



If you have not authorized Visual Studio Code in your GitHub account, you will be prompted to sign in to GitHub.
In the dialog that appears, just click "Continue", an authorization of specific organizations is not necessary.

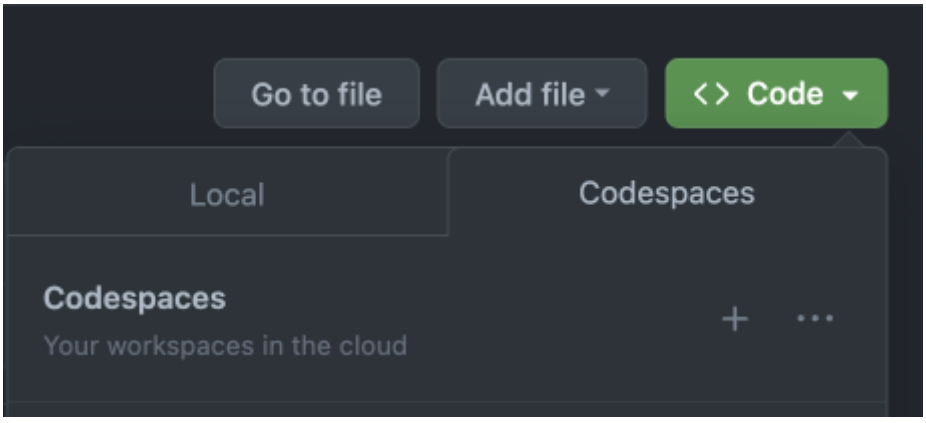


Getting Started

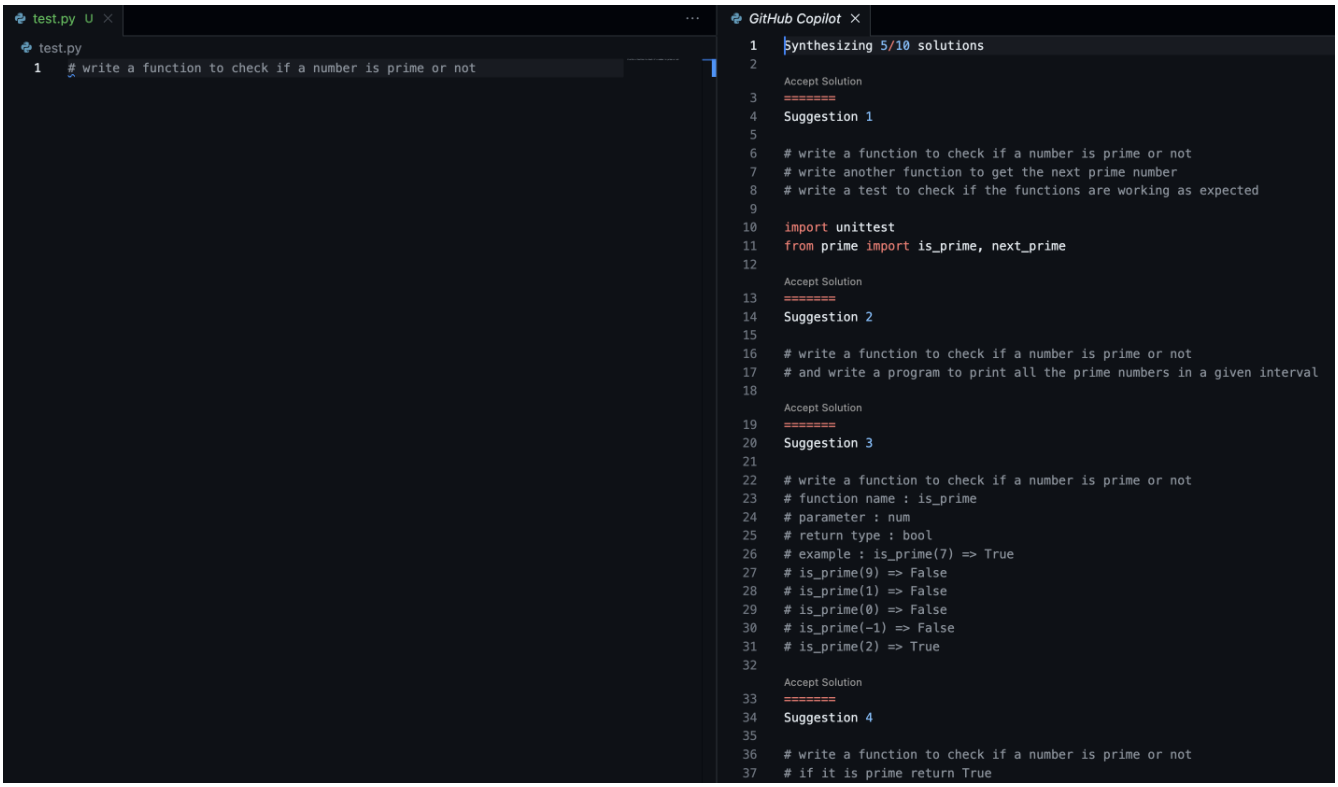
There's a "Getting Started" documentation from GitHub with all information, how to get your first AI suggestion.

One option to start with Copilot is to use a Github Codespaces environment, as it provides an encapsulated container where you can try out things, without changing your local IDE.

To do so, just create a new repository on Github.com and click on '< > Code' → 'Codespaces'



And that's already it, you can open a source code file and start typing while Copilot will try to suggest you helpful snippets.
Or you write a comment and hit "CTRL + ENTER" to get different solutions where you can choose from.



8.8. GitHub Features

8.8.1. Github Codespaces

Codespaces is a feature of GitHub Enterprise Cloud, i.e github.com. It's not available on our BDC-GitHub and there's no statement from GitHub, if and when it could be made available for our GitHub server.
A Codespace is your virtual dev environment in the cloud. There's a marketing page from GitHub, which tells you all the great options of it. And there's a hands-on documentation, which guides you through all the necessary steps, how to setup your codespace(s), how to use them and how to organize and manage them.

Codespaces will create additional costs to your GitHub organization. To get a rough idea, how much that might be, you can use the GitHub Pricing Calculator.

Retention period

Per default, your codespaces will be deleted after 30 days of inactivity but you can also shorten this period to reduce cost. Here is a link to a how-to.

Updates and patches

GitHub will take care of patching the host VM for your codespace while you are responsible to keep the specified container up to date. If you don't specify an image, GitHub provides an universal

image which they will keep up to date.

8.8.2. GitHub Packages

Packages are currently not supported on GHES.

GitHub offers the possibility to create and use so called *packages*. All details about this feature can be found at the official GitHub documentation.
To publish a package, you need to use this link in your workflow:

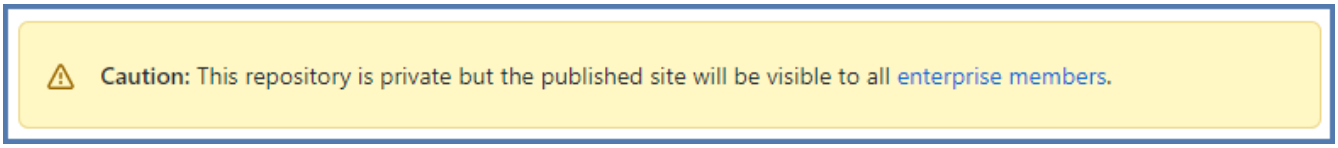
```
1 registry-url: https://github.boschdevcloud.com/_registry/npm
```

Besides JavaScript/npm you could also use Ruby/gem, Java/mvn, Java/gradle, docker and .NET.

8.8.3. GitHub Pages

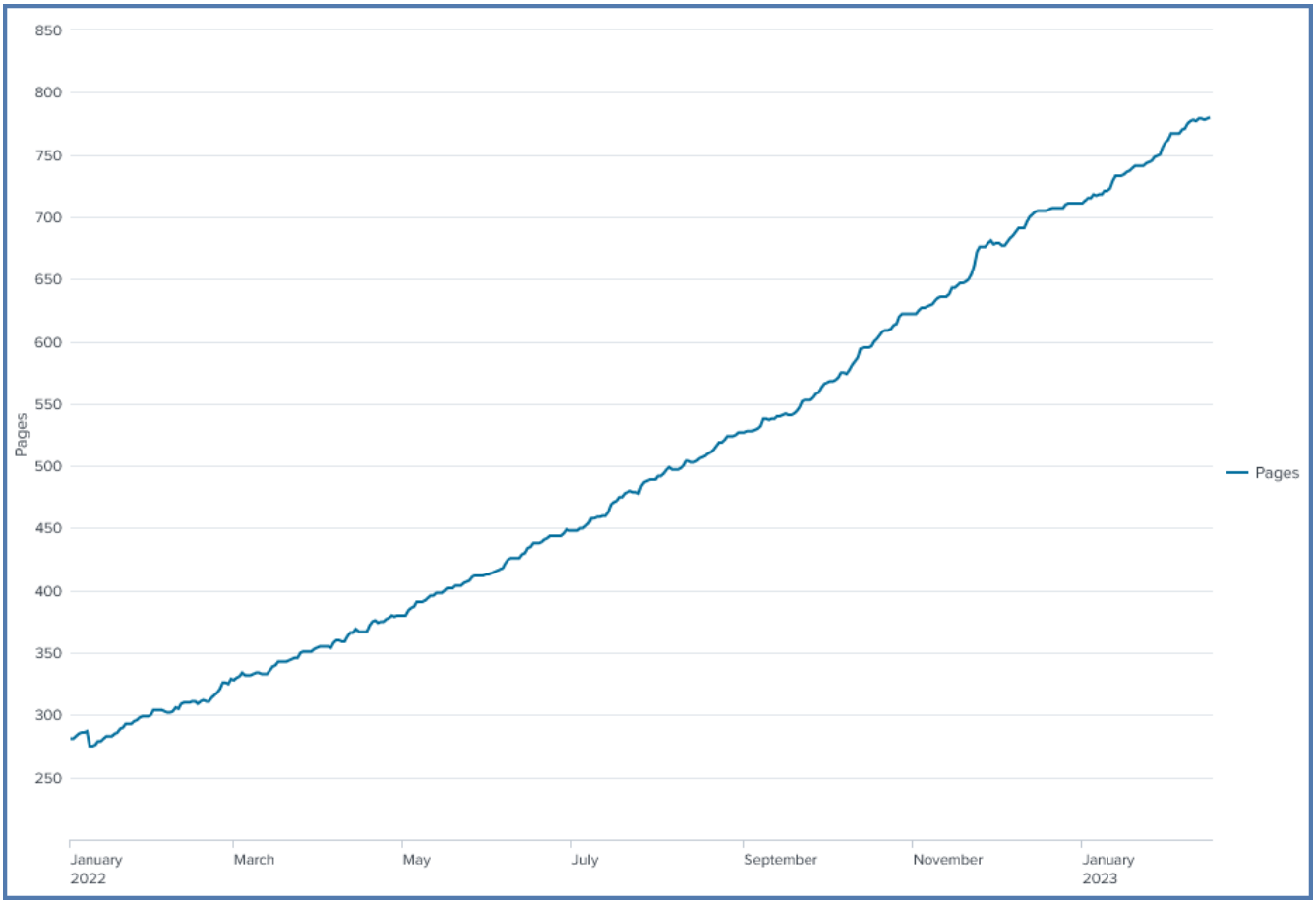
There's a feature in GitHub called **"Pages"**. It's activated in our BDC-GitHub.
Every page you publish is visible to all GitHub Enterprise members. That means, every GitHub user is able to see these published pages. If you want to access this page, your current GitHub session will be used or you will get asked to login.
If you don't have a GitHub account (where you need to pay for the license), it will get created automatically when you click on the green "Login with Bosch account" button.

We don't have the "public pages" option activated; since it's not possible to do this due to our Bosch central directives. This means, you need a GitHub account for every access to a GitHub page. But after your login, you will be able to see **all** GitHub pages. There's no option in GitHub to restrict the visibility of your repository's pages (e.g. to GitHub teams).
You will see this information also in the pages chapter of your repo's settings:



How to use the feature is pretty good documented in the official GitHub documentation: https://docs.github.com/en/enterprise-server@3.9/pages/getting-started-with-github-pages/creating-a-github-pages-site

And it really looks like you make use of that GitHub Pages feature.



8.8.4. Git Large File Storage (LFS)

Git Large File Storage (LFS) is a git extension, which replaces large and binary files such as audio samples, videos, datasets, and graphics with text pointers inside git. And it stores the file content on a remote server, outside the git protocol (which is not designed for such files).

> **Default settings**
> Per default GitHub Enterprise's file size limit is 50MB as recommended by GitHub. The maximum file size in Git LFS is 5GB.
> If you regularly push large or binary files, consider introducing Git LFS as part of your workflow.
> In best case, you enable LFS **before** adding any files.

Some Examples for Usage

Hint: The settings will be saved in the .gitattributes-file.

Configure Git LFS to track a specific file type:

```
git lfs track *.psd
```

Configure Git LFS to track only files in a specific directory:

```
git lfs track 'images/'
```

Check which file types are currently under Git LFS:

`git lfs track`

Remove files from tracking:

```
git lfs untrack '<file-type>'
git rm --cached '<file-type>'
git add '<file-type>'
git commit -m "restore '<file-type>' to git from lfs"
```

Enable Git LFS in your repository

Hint: If you add additional file-types to be tracked through LFS in an existing project, it won't change that project's history. Only from that point in time when you added it, the files will be stored to the external storage.

1. If you received Git-for-Windows over SCCM, then Git LFS is already included. If not, download and install the Git command line extension.
2. Once downloaded and installed, set up Git LFS for your account by running the command below. Nevertheless you must run the command once for each user account, in case you're using more than one.
   `git lfs install`
3. In each Git repository where you want to use Git LFS you have to select the file types you'd like Git LFS to manage, i.e.:
   `git lfs track "*.psd"`
   `git lfs track "*.mov"`
   `git lfs track "*.bin"`

Make sure that the file ".gitattributes" is tracked:
`git add .gitattributes`

Migrate an existing non-LFS repository to a LFS repository

As the way to migrate depends on the version of your git lfs, it's described very detailed here.

Migrate from Bitbucket to GitHub when using LFS

General hint
If you don't use LFS in your repository but want to migrate from BitBucket to GitHub, these steps are also valid. In this case you can skip the steps (4. and 6.).

1. Open Git bash
2. Create a bare clone of the repository

   ```
   git clone --bare https://<old-repository>.git
   ```

3. Navigate to the repository you just cloned

```
cd old-repository.git
```

4. Pull in the repository's Git Large File Storage objects

```
git lfs fetch --all
```

5. Mirror-push to the new repository

```
git push --mirror https://<new-repository>.git +
(If you have files which are larger than 50 MB, you will get an error because it
exceeds GitHub Enterprise file size limit.)
```

6. Push the repository's Git Large File Storage objects to your mirror

```
git lfs push --all https://<new-repository>.git
```

7. Remove the temporary local repository you created earlier

```
cd ..
rm -rf <old-repository>.git
```

8.8.5. Github Data migration

There are some ways to migrate existing git repositories to our BDC GitHub.

Importing a Git repository using the command line

https://docs.github.com/en/enterprise-server@3.9/get-started/importing-your-projects-to-github/importing-source-code-to-github/importing-a-git-repository-using-the-command-line

Source code migration tools

https://docs.github.com/en/enterprise-server@3.9/migrations/importing-source-code/using-the-command-line-to-import-source-code/source-code-migration-tools

Migrate your repositories using ghe-migrator

https://github.blog/2016-05-16-migrate-your-repositories-using-ghe-migrator/

8.8.6. GitHub Command Line Interface

GitHub added its own Command Line Interface (CLI). Using it might make your work much easier. With this dedicated command line interface you're able to work with all GitHub elements, like pull requests or issues, via the cli and don't need to go to the web interface.

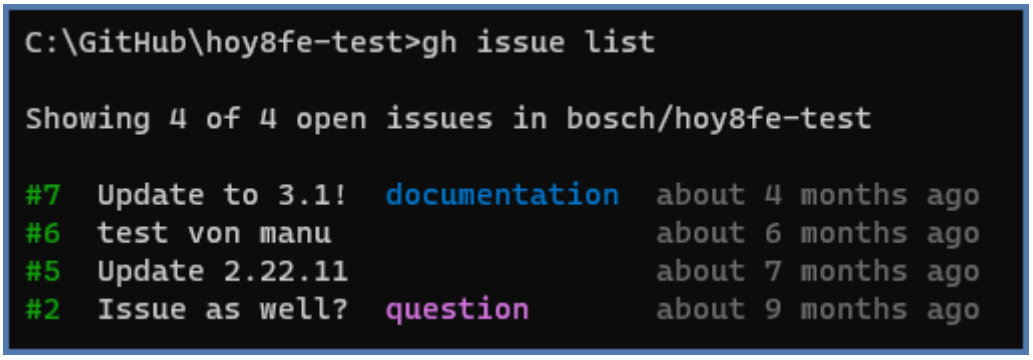
Take care, when using the `api` option! It defaults to github.com and will probably

> lead directly into a rate limit exceeded status (see below).

Some examples

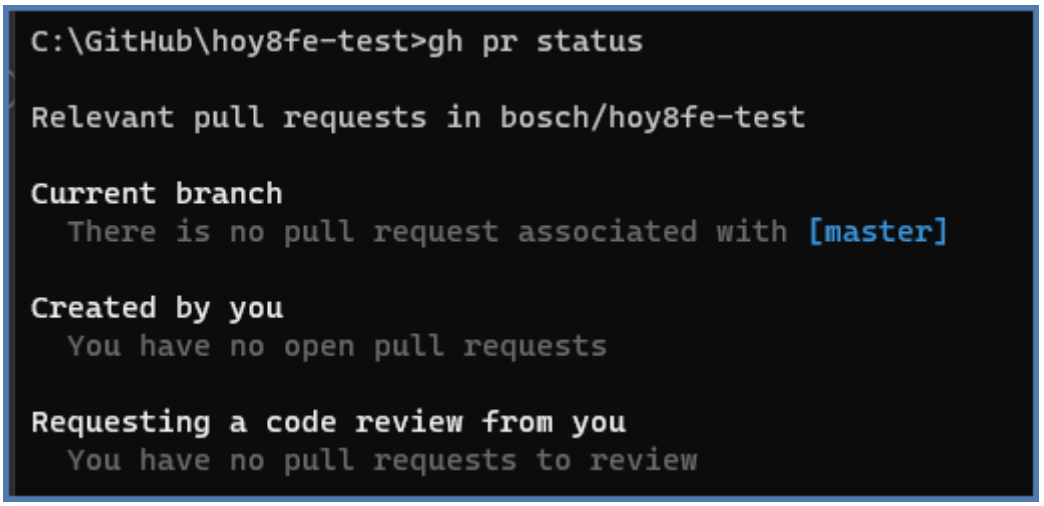
Get all issues of your repository

```
gh issue list
```



Get status of pull requests

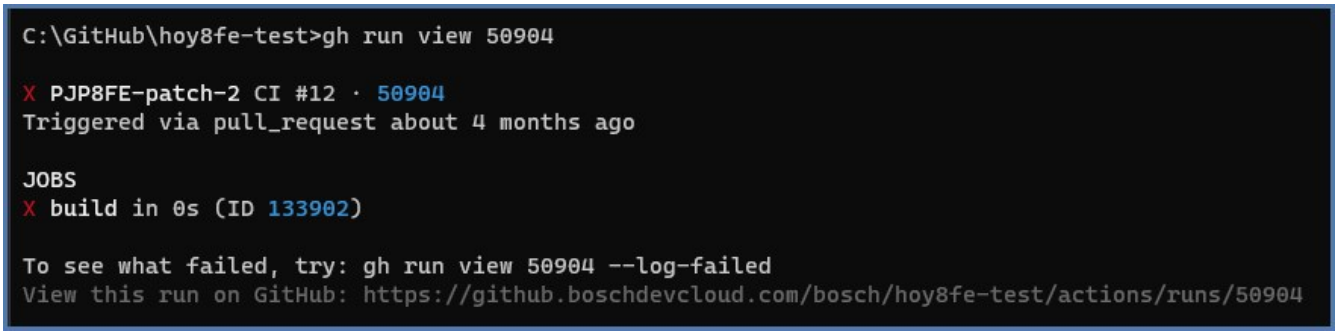
```
gh pr status
```



Get details of a workflow run

```
gh run view 50904
```

(you could get this ID via the list subcommand first)



Execute an API request

This request brings by default the first 100 repositories, which are accessible to you. They're probably all public repos. It might be useful, to redirect the output into a textfile (by adding `> output.txt` to the end).

```
gh api repositories --hostname github.boschdevcloud.com
```

Take a look at the documentation of this subcommand to get additional, useful options, like:
- pagination: --paginate
- simple filter: --jq '.[].full_name'
- more advanced filter: --jq '.[] | .full_name, .fork'
There's a pretty good documentation about the jq usage.

You can get all necessary information about it in the official documentation:
https://cli.github.com
Or just install the GitHub CLI on your machine. There are packages and instructions for Win, Mac and Linux available on the public github repository.
Now you only have to type gh to get all its possible commands, subcommands, and options.

```
gh
```

For a quick start, here's a working example for a graphql request with this gh tool to get all repos and their visibility for the logged in user:

```
gh api graphql --paginate -f query='
    query($endCursor: String) {
      viewer {
        repositories(first: 100, after: $endCursor) {
          nodes { nameWithOwner visibility}
          pageInfo {
            hasNextPage
            endCursor
          }
        }
      }
    }
  '
```

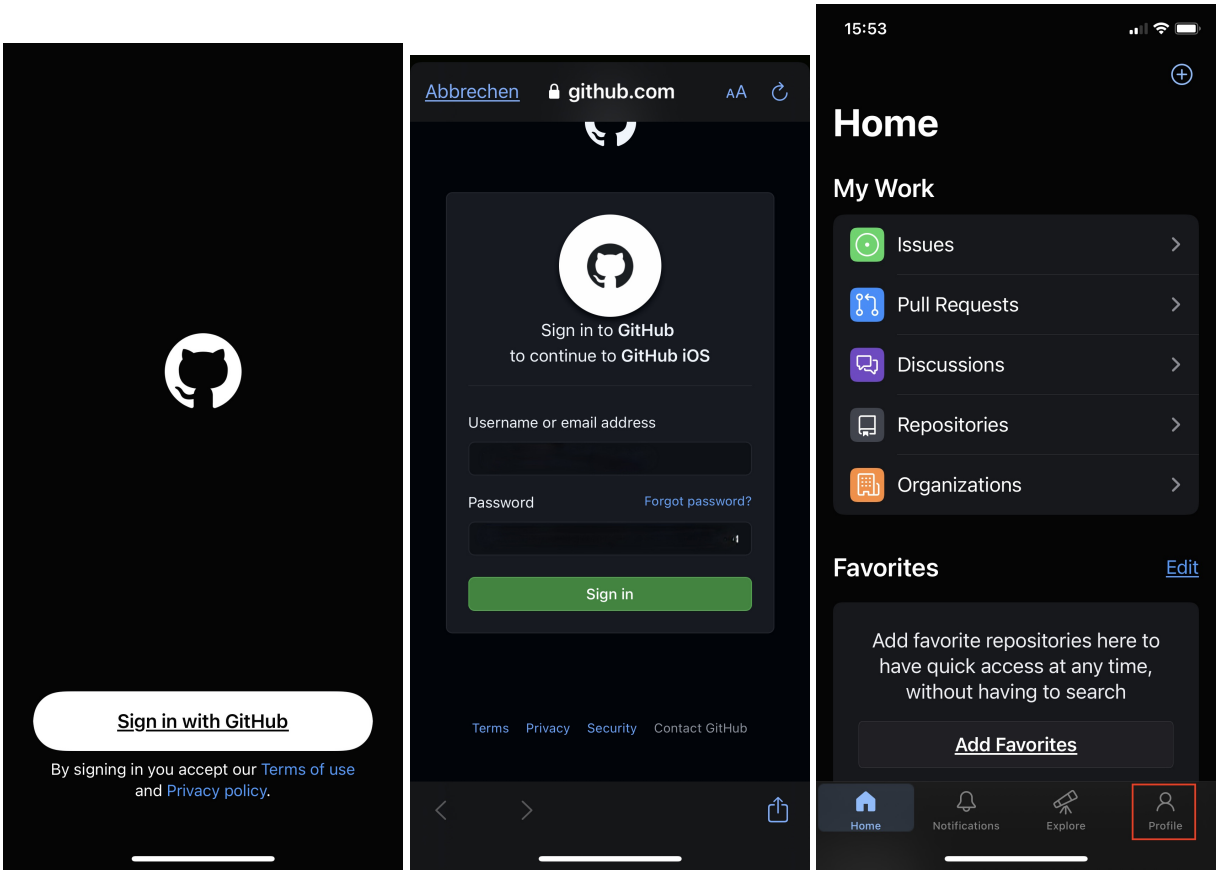
8.8.7. GitHub Desktop

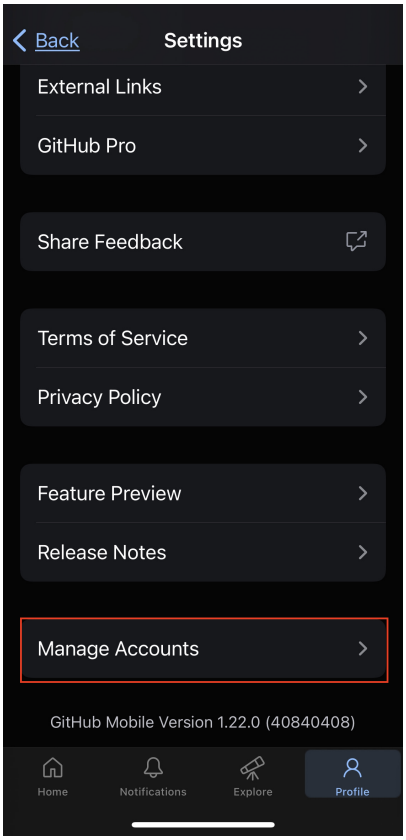
If you're uncomfortable with git commands on the command line, you might take a look at GitHub Desktop. It's a graphical tool to do most of the usual git operations on your repositories.

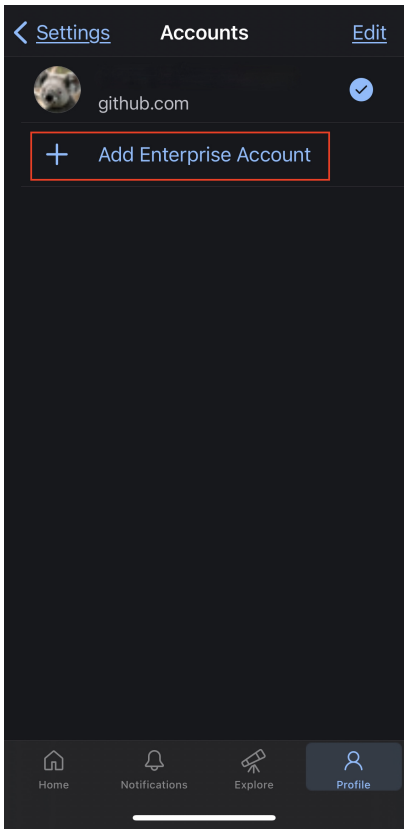
8.8.8. GitHub for mobile

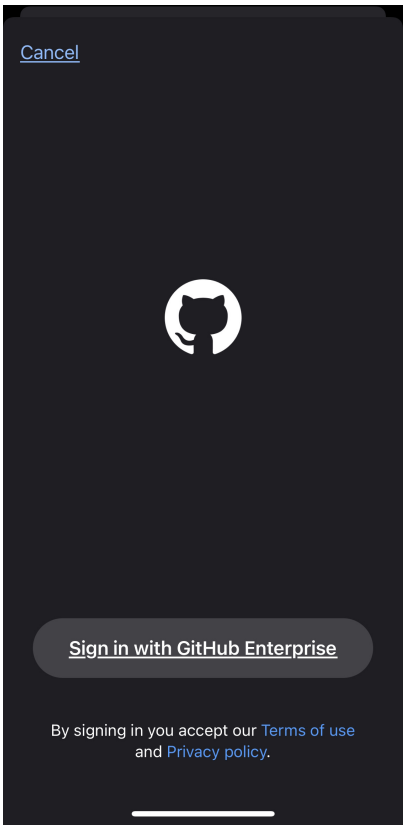
GitHub offers a dedicated mobile app for Android and iOS. You can install it on your phone and connect to github.com and to our BDC-GitHub. You probably can't develop code with this app, but it's designed to watch repositories and organizations, work on pull requests, and take part in discussions.
You can download it from your app store and find all further information here. It only contains one tracker in the android app.+ Since it's a bit tricky to connect to this app, we have a short screenshot documentation.

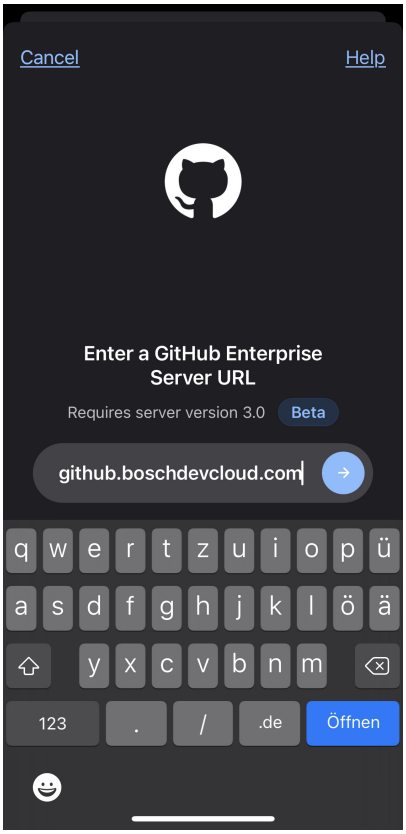
Prerequisite:
Your account needs 2FA to use it - as described here.

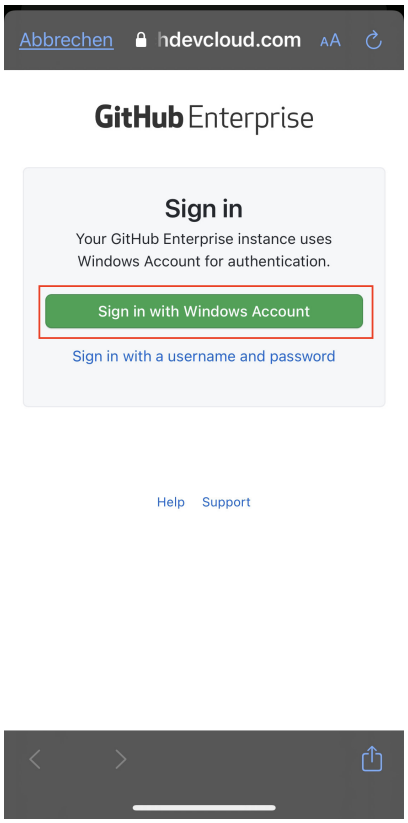


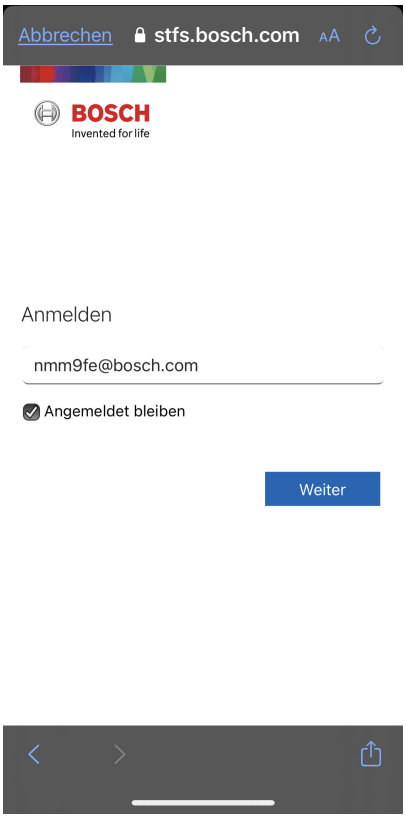

Back
Settings
External Links
GitHub Pro
Share Feedback
Terms of Service
Privacy Policy
Feature Preview
Release Notes
Manage Accounts
GitHub Mobile Version 1.22.0 (40840408)
Home
Notifications
Explore
Profile

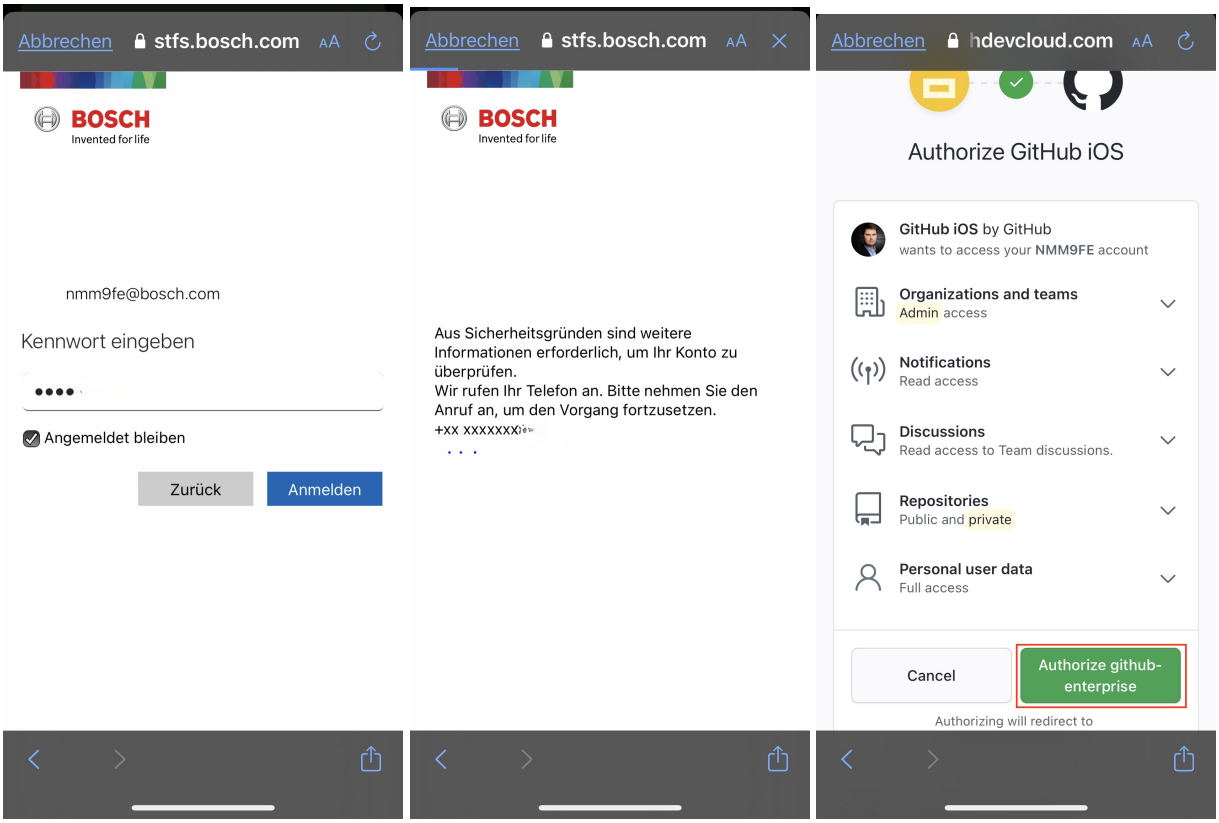

Settings
Accounts
Edit
github.com
Add Enterprise Account
Home
Notifications
Explore
Profile

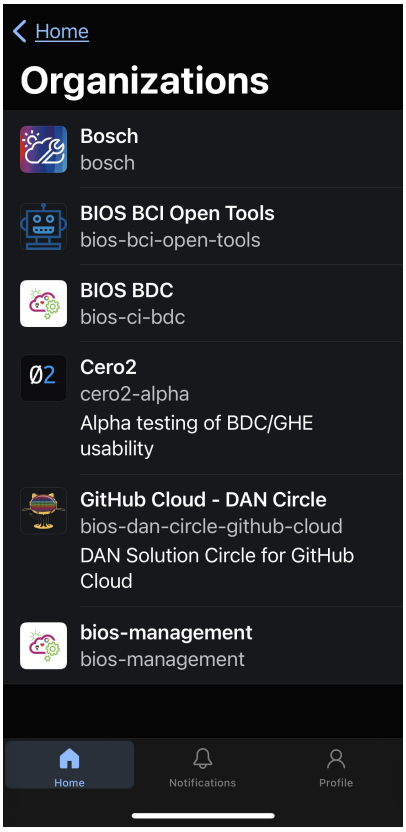

Cancel
Sign in with GitHub Enterprise
By signing in you accept our Terms of use and Privacy policy.


Cancel
Help
Enter a GitHub Enterprise Server URL
Requires server version 3.0
Beta
github.boschdevcloud.com
Öffnen


Abbrechen
hdevcloud.com
AA
GitHub Enterprise
Sign in
Your GitHub Enterprise instance uses Windows Account for authentication.
Sign in with Windows Account
Sign in with a username and password
Help
Support


Abbrechen
stfs.bosch.com
AA
BOSCH
Invented for life
Anmelden
nmm9fe@bosch.com
Angemeldet bleiben
Weiter





8.9. Plugins

SaaS Providers like Atlassian, AzureDevOps Services or GitHub offer marketplaces that include first- or third-party plugins, apps, extensions or add-ons. In many cases, the plugins, apps, extensions or ad-ons can be easily installed. Some of them can be used for free, for others licenses have to be bought. However, before using them, it is important to check, if and under which conditions it is allowed from Bosch side to use these. Therefore, please check the FAQ section of

C/IDO who is responsible for the SaaS - Onboarding Process.

To find out which plugins, apps, extensions or add-ons is already onboarded, please search for it in LeanIX.

8.9.1. Cost for Plugins

The costs for Plugins/Apps/Extensions can be found in the marketplace of the SaaS Provider.

8.9.2. Shared responsibility for Plugins

Customers have the possibility to choose if they want to use plugins, apps, extensions or add-ons. If decided to use plugins, apps, extensions or add-ons it is in the responsibility of the customer to make sure that they are compliant with all Bosch regulations (Central Directives, EISA, works council agreements).
The Bosch Development Cloud does not offer product support for plugins, apps, extensions or add-ons or the onboarding process in case it is needed.

8.10. GitHub Support

Sometimes the software behaves different, than you expect it. This could have a few different reasons.

1. It simply behaves in a different way, as you would think or as you might know it from different systems. Please check the GitHub documentation, if you can find something about your topic: https://docs.github.com/en/enterprise-server@3.9
2. It behaves in a different way as it's documented. This would be considered as a bug (or at least a wrong documentation). In this case, you might send us the information about the misbehavior and we could create a support ticket at GitHub.
    a. There are different ways for a solution. Depending on the priority it could be fixed immediately either via a hotpatch or some other "fix-magic". Or it might be added to the GitHub-internal feature request list. Or it is already on this list and we will be added to the "requesters".
    b. If it's not a high-priority for GitHub, there's the chance to buy so called "Professional Services" from GitHub. Within these "Professional Services" we could request a solution from GitHub for our issues. If this should be needed one day, please get in contact with us from the BDC-team.

If you're a key user of GitHub and want to skip the CI-middle-men for your support process, please get in contact with the BDC-team. We have a few free seats for the GitHub Support and it might be helpful, if you get in direct contact with GitHub.

8.11. GitHub FAQ

8.11.1. Technical FAQ

I am getting an error during git clone which says "Could not resolve host: github.boschdevcloud.com"?

You should set up git proxy by referring to this Proxy configuration. If you are using internet client or non-Bosch client this is not required.

How should I authenticate in `git` cli tool?

You can use Personal Access Token (PAT). Feel free to check detailed answer below.

▼ *Click to reveal the full answer...*

Since SAML is used as authentication method, you must not use your Bosch user password to e.g. clone a repository. In this case, you need to create a **Personal Access Token** (or short PAT). There's an article on the GitHub website, which describes, how to create such a PAT.

This token has to be handled similar to your password. Do not share it and keep it in a secure storage (for example in Keepass).
It's a good practice to set an expiration date for each PAT.

In case you get an error while cloning a repository on the command line, it might be the case, that an old/outdated/invalid password is stored in the credential manager. In this case, just open the "Credential Manager" via the windows start menu / control panel and remove all entries, which might store a wrong password.
Afterwards you could try a git clone again and will probably get asked two passwords, one for the proxy (use your windows account here) and the second one for the GitHub access (use your GitHub username and the PAT here).

A short troubleshooting page about connectivity issues can be found here in BCI's Docupedia

How to request access to an Github organization?

Please get in touch with your Github organization owner, but here are some hints.

Github Enterprise Server:

- **Normal Github Organization**: Kindly co-ordinate with your Organization owner and add yourself to the respective Idm roles as below,
  - Owner - IDM2BCD_BDC_Github_01_org<ID>_owner: Organization owners are like the "admin" of an GitHub org, they have all full access to all settings.
  - Member - IDM2BCD_BDC_Github_01_org<ID>_member: this is the role for every regular user in this GitHub org.
- **BIOS Github organization**: Refer to the Permission Management for BIOS Projects section.

Login to Github once and then add yourself to respective Idm roles.

Github Enterprise Cloud:

- Refer to the section Get access to an existing Organization.

How to login to GitHub Enterprise cloud - GHEC?

To login github.com organization, refer to Login Process document.

I have access to the Github Organization, but unable to login!

Must be due to temporary cache issue, try clearing cache and check again.

Login to GHES from non Bosch device does not work?

- If you are a Bosch employee, you must use a Bosch device to login to Github.
- If you are an external employee, you will be able to login with company provided laptop, but you should have MFA configured as described in external access section.

Are webhooks in ghes possible?

Yes, webhooks can be created in GHES, but network limitations must be considered.

- Webhook to server in Public cloud - This is possible, if you have firewall enabled, don't forget to whitelist Github outgoing IP which is mentioned in Service acceptlisting.
- Webhook to Bosch Network - If target server is located in SL3/SL4 network zones, communication from internet (Github) is completely blocked whereas if your server is located in SL2 zone, communication might be possible with firewall whitelisting.

Is it possible to point an internal DNS to Github pages?

No, this cannot be done, as even if we use CName to do it, the Github pages primary URL will not know how to handle this request.

Why Contribution chart does not show information on private repositories?

GitHub creates a *contribution chart* on every user's profile page. Contributions are e.g. creating issues or committing to a repository, which happened during the last year. By default only public contributions are visible in your chart, but you can change it, to show all private contributions, as well.
When you connect your github.com user account with your BDC-GitHub account (Settings - GitHub Connect), you can send your contributions to your github.com account. Only plain numbers are sent, nothing else, i.e no content.

Can I have Anonymous Read Access in Github?

Due to the regulations of Bosch given in the Central Directives it is not allowed to login in to GitHub@BDC without a valid Bosch account. Services provided by Bosch Development Cloud are running in the cloud/public internet so an anonymous login allows access to every person in the world. This is not acceptable for a commercial company. Find more information here (CD) or here (IdM)

Is there any recommendation for force pushes?

In case you're using the option Force Push in one of your repositories, you should provide a protected branch, as well.
You can learn everything about protected branches on the GitHub documentation.

can I change repo visibility for repo in my Organization?

In context of normal Github Organization, we advise you to use repos with `private` type. Also refer to our Repositories document.

In context of BIOS Github Organization, you must only use repos with `Internal` type to adhere BIOS policy.

8.11.2. General FAQ

Which one should I choose, GHES or GHEC?

Each offering will have their own advantages and disadvantages. We have documented the differences for both offering, kindly refer to the following section Differences between BDC's GitHub Enterprise Cloud Offering and BDC's GitHub Enterprise Server Offering for the same.

We are a big team, how many organization should we create?

In general, GitHub recommends minimizing the number of organizations you create. Take a look into Github best practices documentation.

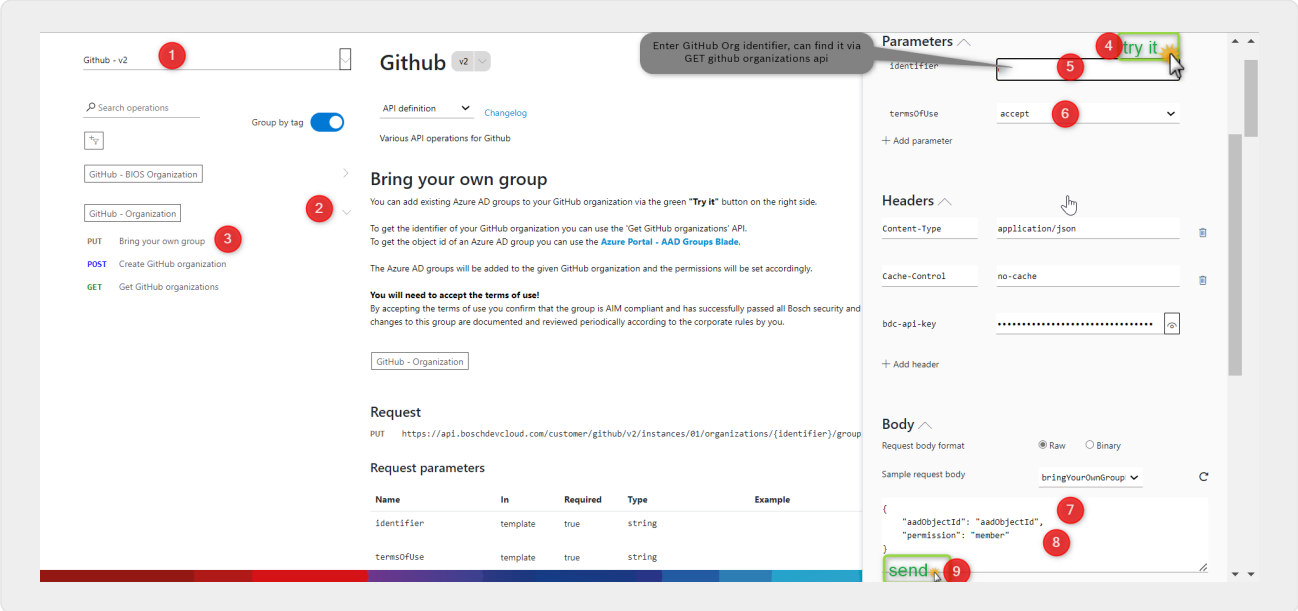
As an organisation owner, how to allow access for `user-a` to `repo-x` but not `repo-y`?

You can use Github Teams feature to achieve this, refer to the section Teams for permission management and Github teams documentation.

Is there any other option for permission management without using provided IdM roles?

You can use `Bring your own group (BYOG)` feature. Go to BDC portal and use Github API to add existing Azure AD groups to your GitHub organization.

▼ *Click to reveal more information...*



We are new to Github, could you advise on some best practices that we can follow?

- Provide access to other members via IdM roles and not by directly adding them to Organization.
- Refer to Github documentation on Best practices for organizations
- Refer to Github documentation on Best practices for repositories

I am not receiving any notification mails from GitHub for activities like pull requests..

Check your junk folder in outlook and if mails are found there, kindly adjust your outlook settings. Also make sure respective settings are enabled in GitHub

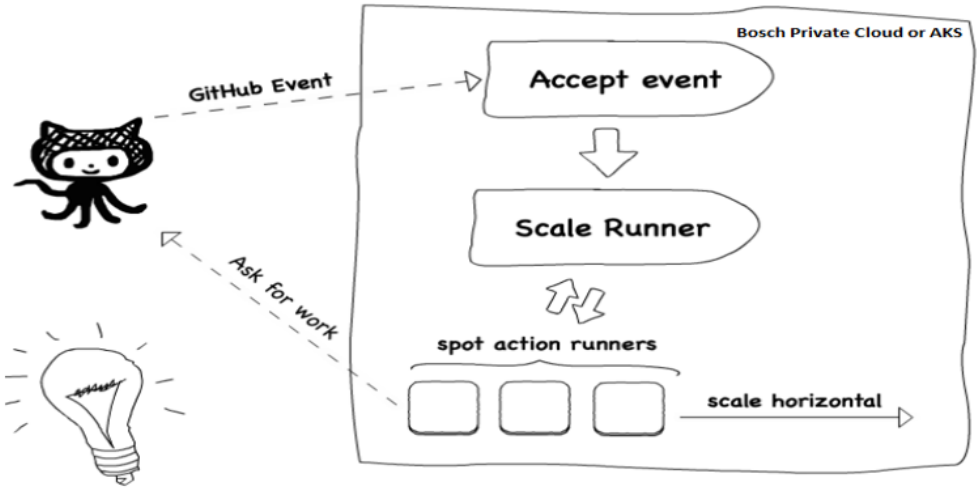
8.12. Feedback

We look forward to receiving your feedback for GitHub and for GitHub.com, which will help us to improve in the future!

Chapter 9. Runner as a Service

BoschDevCloud Runner as a Service, or simply RaaS, provides BDC-managed runners based on the GitHub custom workflow model, GitHub Actions. Most often, developers shape these action workflows to implement build and deployment workflows.

As a picture can be worth a thousand words, below is a simple carton to illustrate how the event handling of GitHub Enterprise and RaaS work together to support developer build scenarios.



9.1. What do I get for Runner as a Service and how to order

9.1.1. Runners

Runner as a Service offers container-based runners for GitHub Enterprise Server as a BDC standard service.

Runner providers are available in dedicated and shared instance types and linked to Azure (Public Cloud) and Bosch Private Cloud (designation and zone BPC-IPZ) host environments. When choosing a runner provider, please choose carefully. Each provider has its advantages. Find additional information in the respective chapters below.

Table 4. Availability zones for Dedicated and Shared Instance

Instance Type/Availability Zones	**Azure (Public Cloud)**	**BPC-IPZ (Private Cloud)**
Shared	raas-azure-shared (**generally available**)	raas-bpc-ipz (**generally available**)
Dedicated	raas-azure (**generally available**)	feature under consideration

There are several advantages and disadvantages of having runners in Azure or BPC-IPZ. Some of them are listed below,

Table 5. comparison of runners in Azure Dedicated and BPC-IPZ Shared Instances

	Azure (Public Cloud) - Dedicated	**Azure (Public Cloud) - Shared**	**BPC-IPZ (Private Cloud) - Shared**
Cost	Higher costs due to dedicated Azure cloud services.	Lower cost savings for shared K8s instance.	Lower cost savings for shared K8s instance. Costs are internal.
Access to runner	yes	not possible	not possible
Access to Bosch resources	Only resources exposed to internet can be accessed	Only resources exposed to internet can be accessed	yes, refer to the below chapter to see whitelisted resources.
Maximum number of runners in runner deployment	unlimited	10	10
Resource limitation of Runner	No resource limitation set	8 GiB Memory + 1 vCPU	8 GiB Memory + 1 vCPU
Node size	• **Standard** type with machine size Standard_D2ads_v5 (Default) • **Performance** type with machine size Standard_D8ads_v5	not applicable	not applicable

RaaS offers scalable integration mechanisms to host runners and execute GitHub build workflows in a scalable container context.

To set up RaaS, use self-service APIs to create a runner group, a RaaS project, and a runner itself. The deployment of these components generally follows the following API flow:

1. Create a runner group - The purpose of a runner group is to control access to runners at the organization and/or enterprise levels. Please note that Runner Groups are created at the enterprise level so that it can be accessed across multiple GitHub Organizations.
2. RaaS project creation - is a logical representation of your RaaS application in shared or dedicated instances.
3. Runner deployment - the runner instance itself.

9.1.2. How to order

The BDC API portal offers a self-service model to order RaaS services. Self-service APIs support the RaaS project setup in an AKS or a BPC-OpenShift K8s host environment. You can instantiate runner groups and link them to one or more of your GitHub Organizations using the APIs. This flexibility allows you to share runners between GitHub Organizations associated with a runner group. You can also add auto-scaling runners to your runner groups.

It’s important to note that the GitHub workflow specifier 'runs-on:' specification determines which

runners are used to execute the workflow by enumerating runner labels.

Listing 8. Sample workflow to choose runner based on label

```
name: Workflow name

on:
  workflow_dispatch:

jobs:
  your-job-name:
    runs-on: [ raas-hosted ]
...
```

Prerequisite:

- A GitHub organization
- A dedicated RaaS configuration (Azure-based) requires BDC Cloud Space as a prerequisite. The BDC Cloud Space instance includes a K8s host environment that RaaS utilizes.
- All resources (GitHub organization, Cloud Space) should be under same BDC. Make sure to deploy also RaaS under same BDC.

Ordering steps

Order RaaS via GUI in BDC Portal or CLI based on your preference.

Order via BDC Portal (GUI)

- Go to the BDC Portal and login.
- You must be a BDC account admin to do this. If you are one, you can see your BDC ID (BDC-XXX).
- Click on it and go to the GitHub API.
- Make sure you create your RaaS Project in the same BDC where GitHub Organization is deployed.
- Order runner group, RaaS project, and runner.
- Detailed steps are given below.

▼ *Click to reveal full information...*

Get GitHub Organization details

To create a Runner Group, you will need the GitHub identifier of the particular Org, so kindly execute this API to get details and save results to use it later.

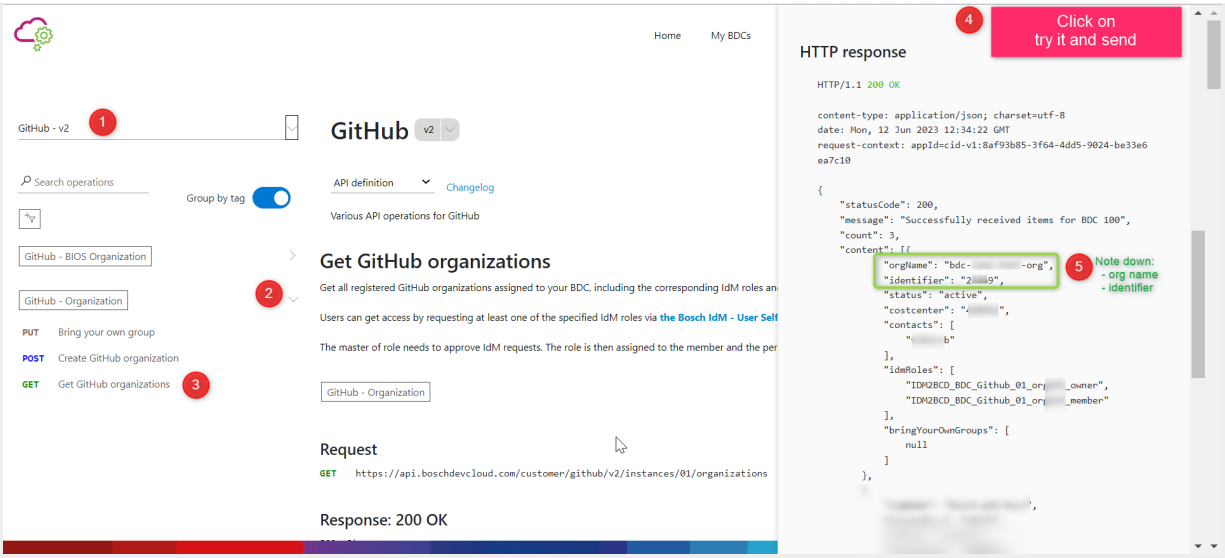


Figure 1. get github org details

To view images clearly, right click and open image in new tab.

Create runner group

runner group is a pre-requisite for RaaS. It facilitates managing access for runners inside and between multiple organizations.

more information about runner groups can be found here

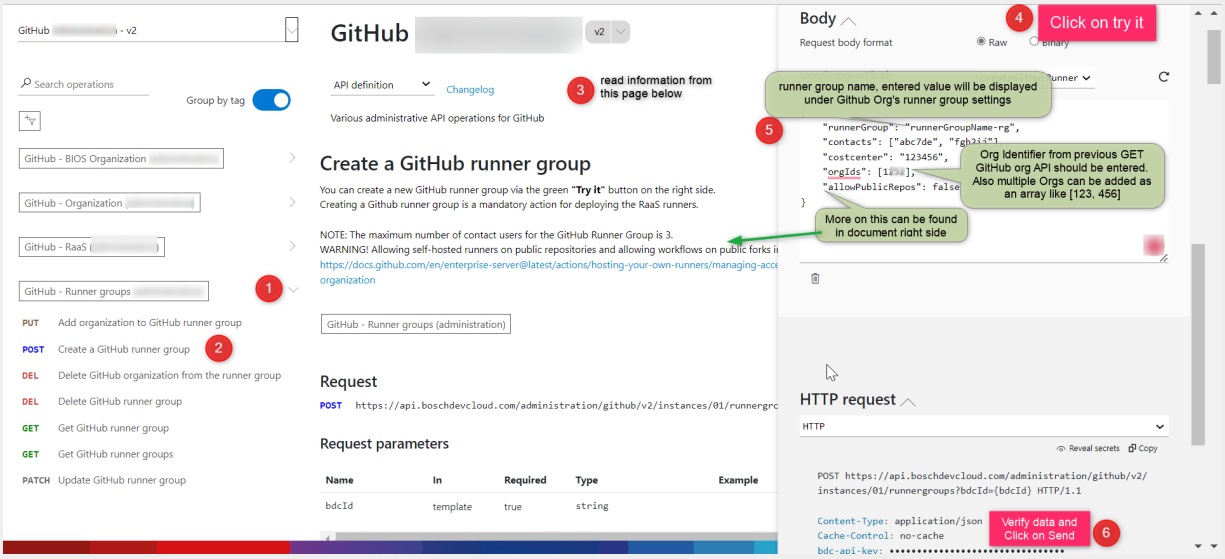


Figure 2. runner group creation

Create raas-project

Create raas-project for **dedicated/shared** based on your requirement. Available raas service instances for the shared environment are: raas-bpc-ipz and raas-azure-shared. For dedicated environment it is: raas-azure. Refer to full documentation before creating.

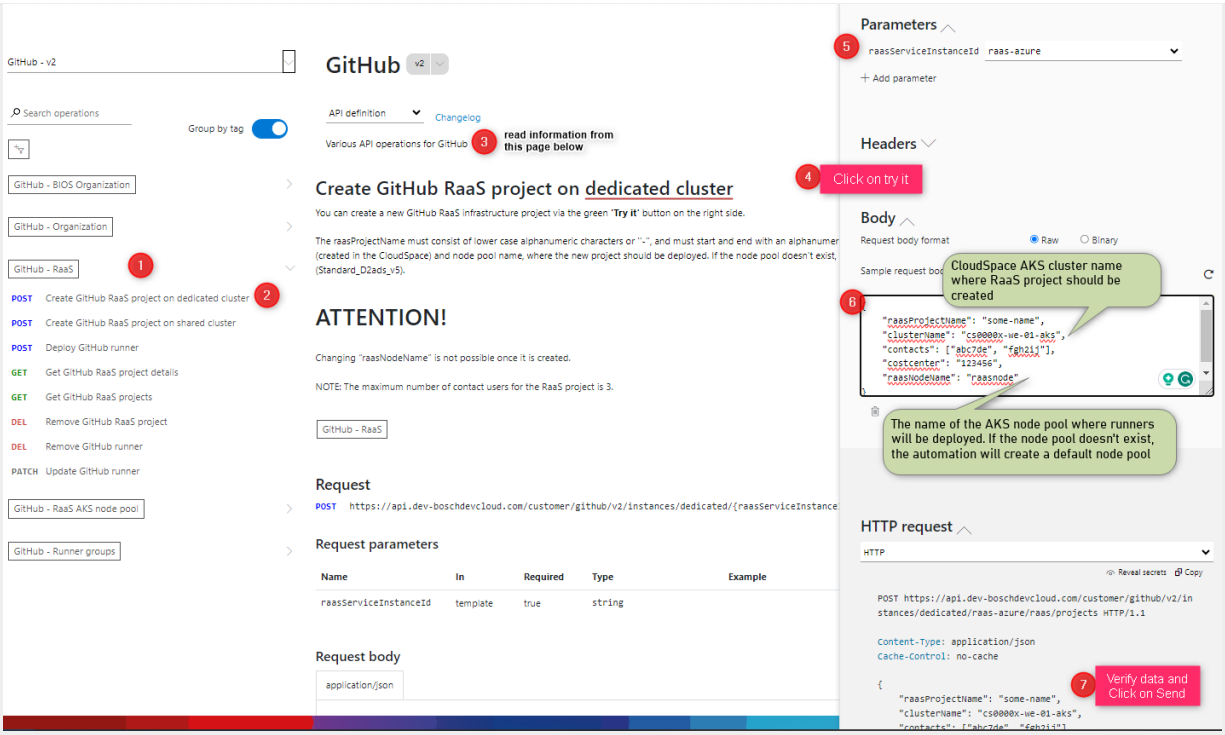


Figure 3. raas-project for dedicated

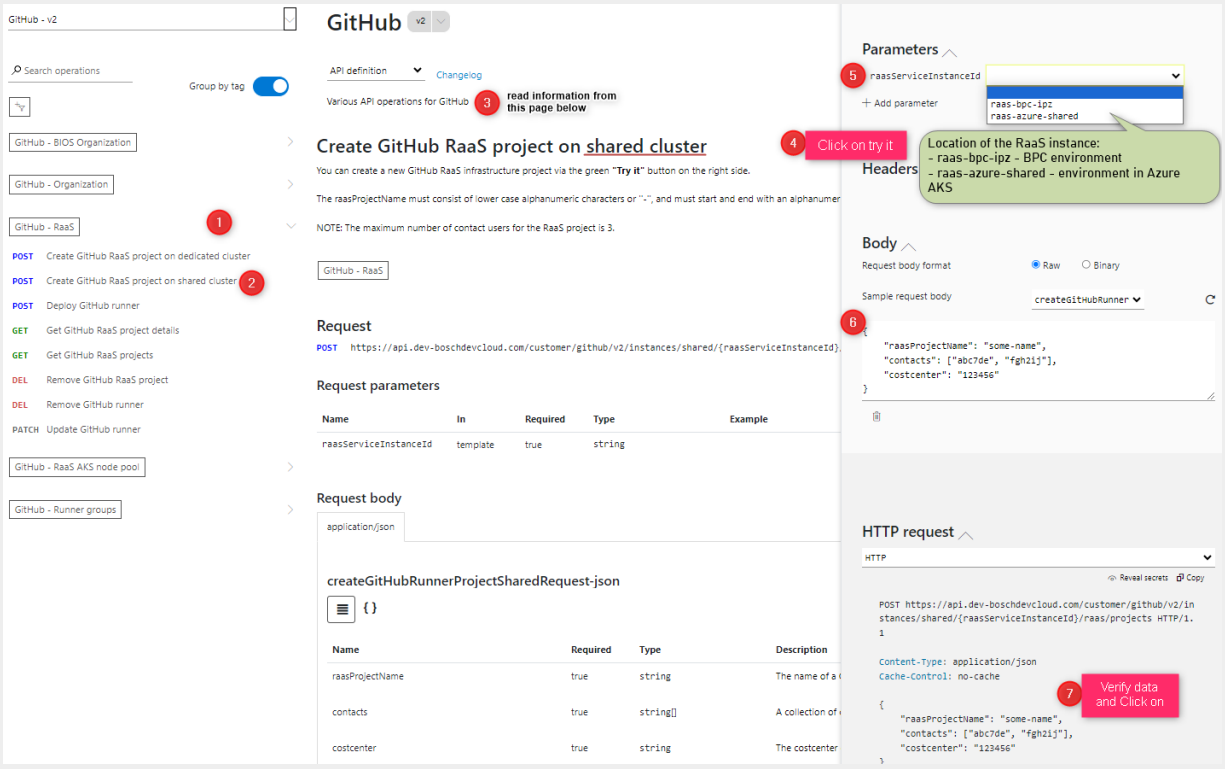


Figure 4. raas-project for shared

Deploy Runner

Deploy runner in one of the available raas service instances for the shared environment are: raas-bpc-ipz and raas-azure-shared. For dedicated environment it is: raas-azure. Refer to full documentation before creating.

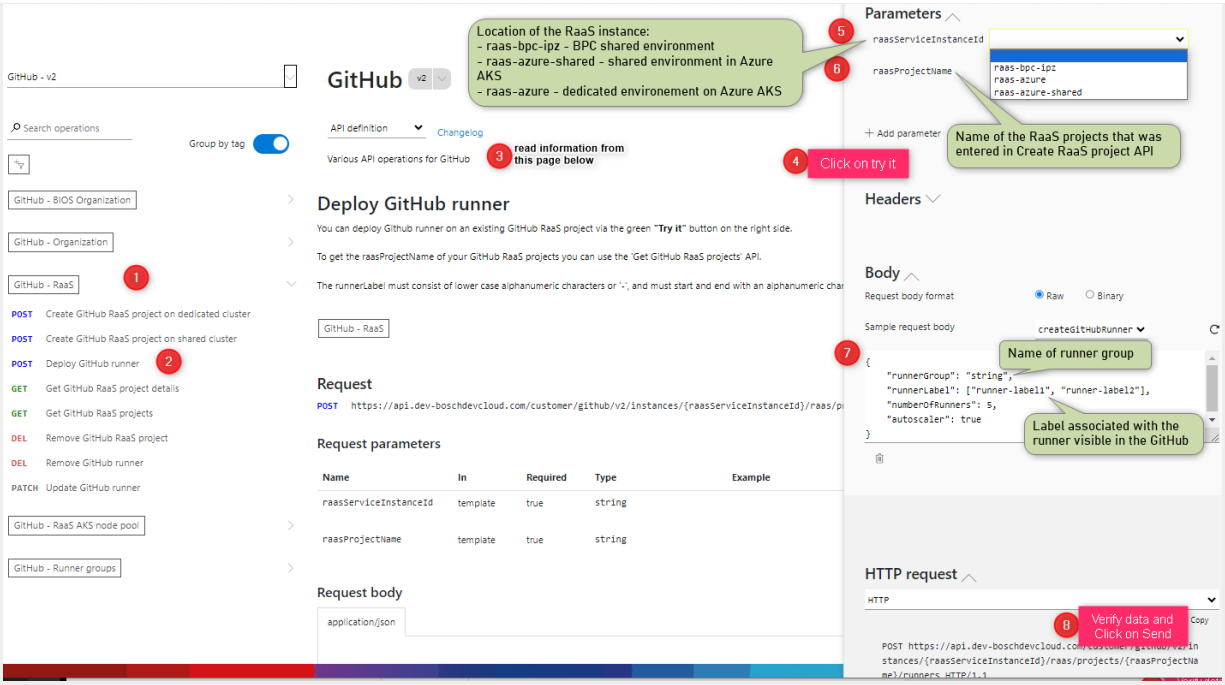


Figure 5. runner deployment

Order via CLI

The command line is also an option for setting up RaaS. Follow the client-side API invocation steps documented below to realize a new RaaS instance. The expression of individual API calls follows the Curl syntax. However, developers can trigger the APIs from any appropriate API client or context—for example, a Postman collection or a Python script.

▼ *Click to reveal full information...*

get github org details

Identify an existing GitHub organization to associate with a new RaaS instance. If org does not exist, please review the user guide contents on how you can order an organization (either regular or BIOS) for GitHub Enterprise. Once an org is identified or created, you can confirm the specific details of the target GitHub organization by using the following curl command:

```
curl -L -X GET
"https://api.boschdevcloud.com/customer/github/v2/instances/01/organizations
?bdcId={bdcId}" -H "Accept: application/vnd.github+json" -H "bdc-api-key:
my_key"
```

This API invocation requires the following two key arguments:

- bdcId - the id associated to your BDC space
- my_key - the primary or secondary key associated to the (product) subscription you created earlier

In case you have to create a new GitHub org, you will also need to request the appropriate IdM role as described here in order to have access to the newly created GitHub

organization.

Prerequisite - BDC API Key

An API key is required to consume the various APIs. The BDC API key can be obtained from the Bosch Dev Cloud portal. Follow these steps to obtain your API key value:

1. Browse to BDC
2. Click on the API category GitHub Administration
3. Select a specific API (e.g., the GET form of Get GitHub Organizations)
4. Use the eye button to unmask the API key bdc-api-key
5. Copy the hexadecimal value of the API key and save it (as will be required in curl or similar API invocation steps below)

TODO: note API tool selection (BDC portal, curl, Postman, ..)

Create Runner Group

Enterprise runner groups is one of the two foundational objects that must be instantiated before you can create/bind a runner. This object creates an association with your GitHub organization that is also utilized by the BoschDevCloud team to realize behind-the-scenes runtime integration and management. The structure of this API call follows the Curl syntax outlined below:

```
curl -L -X POST
"https://api.boschdevcloud.com/customer/github/v2/instances/01/runnergroups?
bdcId=1" -H "Accept: application/vnd.github.v3+json" -H "bdc-api-key:my_key"
-H "Content-Type: application/json" --data-raw "{\"runnerGroup\":
\"runnerGroup24\", \"contacts\": \"abc1de\", \"costcenter\": \"456123\",
\"orgIds\": \"1\"}"
```

This POST-oriented API invocation requires the following five key arguments:

- my_key - the primary or secondary key associated to the (product) subscription you created earlier
- runner group - a string uniquely identifying your runner group
- contacts - a contact (e.g., Bosch AD user id)
- cost center - the cost center where consumption of this service is charged to
- orgIds - the parent GitHub organization (can be more than one [])

Note, following invocation it will take several minutes for background activities to complete. This behavior is reflected in status code 402, where further processing is noted.


```
{
    "statusCode": 202,
    "message": "Accepted request for further processing. To check if the request is completed or still in the processing status
    "count": 1,
    "content": [
        {
```

To check on the creation status, first inspect the result of the POST API invocation. Later, you can use the GET form based on the Uri administration/github/v2/instances/01/runnergroups to confirm creation completion status.

RaaS Project Creation - shared

The second foundational object that must be instantiated for RaaS setup is the project object. In this step, the K8s provider objects including namespace are created in the background as a result of project API invocation.

```
curl -L -X POST
"https://api.boschdevcloud.com/customer/github/v2/instances/shared/{raasServ
iceInstanceId}/raas/projects?bdcId={bdcId}" -H "Accept:
application/vnd.github.v3+json" -H "bdc-api-key: my_key" -H "Content-Type:
application/json" --data-raw "{\"raasProjectName\": \"some-name\",
\"contacts\": [\"abc7de\", \"fgh2ij\"], \"costcenter\": \"123456\"}"
```

This POST-oriented API invocation requires the following four key arguments:

- bdcId - the id associated to your BDC space
- raasServiceInstanceId - an id of the instance where you want to deploy your runners, for RaaS shared it can be raas-bpc-ipz if you want to use BPC infrastructure or raas-azure-shared if you want to use an Azure infrastructure with Internet access
- my_key - the primary or secondary key associated to the (product) subscription you created earlier
- raasProjectName - a name of the RaaS project which you want to created and deploy runners later
- contacts - a contact (e.g., Bosch AD user id)
- cost center - the cost center where consumption of this service is charged to

You can use the GET form of the RaaS project-oriented API to check on the creation status.

RaaS Project Creation - dedicated

The second foundational object that must be instantiated for RaaS setup is the project object. In this step, the K8s provider objects including namespace are created in the background as a result of project API invocation.

```
curl -L -X POST
"https://api.boschdevcloud.com/customer/github/v2/instances/dedicated/{raasS
erviceInstanceId}/raas/projects?bdcId={bdcId}" -H "Accept:
application/vnd.github.v3+json" -H "bdc-api-key: my_key" -H "Content-Type:
application/json" --data-raw "{\"raasProjectName\": \"some-name\",
\"clusterName\": \"some-cluster-name\", \"contacts\": [\"abc7de\",
\"fgh2ij\"], \"costcenter\": \"123456\", \"raasNodeName\": \"raasnode\"}"
```

This POST-oriented API invocation requires the following four key arguments:

- bdcId - the id associated to your BDC space
- raasServiceInstanceId - an id of the instance where you want to deploy your runners, for RaaS dedicated it can be only raas-azure
- my_key - the primary or secondary key associated to the (product) subscription you created earlier
- raasProjectName - a name of the RaaS project which you want to created and deploy runners later
- clusterName - the name of a existing CloudSpace AKS cluster
- contacts - a contact (e.g., Bosch AD user id)
- cost center - the cost center where consumption of this service is charged to
- raasNodeName - the name of a node pool name, where the new project should be deployed. If the node pool doesn't exist, the automation will create a new one with 'standard' machine size (Standard_D2ads_v5).

You can use the GET form of the RaaS project-oriented API to check on the creation status.

Runner Deployment

The final provisioning API deploys the Action Runner Controller (ARC) and setups the runner instance(s).

```
curl -L -X POST
"https://api.boschdevcloud.com/customer/github/v2/instances/{raasServiceInst
anceId}/raas/projects/{raasProjectName}/runners?bdcId={bdcId}" -H "Accept:
application/vnd.github.v3+json" -H "bdc-api-key::my_key" -H "Content-Type:
application/json" --data-raw "{\"runnerGroup\": \"runnerGroup24\",
\"runnerLabel\": [\"my-runner-label\"], \"autoscaler\": \"false\",
\"numberOfRunners\": 5,}"
```

This POST-oriented API invocation accepts the following key arguments:

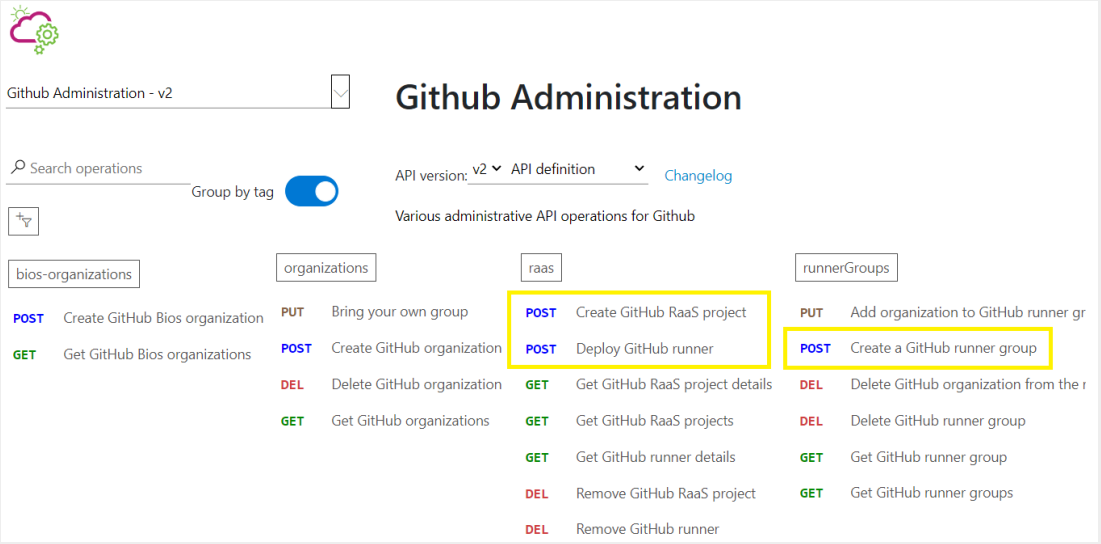
- bdcId - the id associated to your BDC space
- raasServiceInstanceId - an id of the instance where you want to deploy your runners, for RaaS dedicated it can be only raas-azure. For RaaS shared it can be raas-bpc-ipz if you have a raas Project in BPC infrastructure or raas-azure-shared if you have it in Azure
- raasProjectName - the name of the RaaS project you want to deploy the runners to
- my_key - the primary or secondary key associated to the (product) subscription you created earlier
- runner group - a string uniquely identifying your runner group
- runner label - a unique label that will identify the runner instance (use the workflow run-ons to bind to a particular runner)
- autoscaler - boolean flag to enable K8s auto-scaling

- numberOfRunners - if the autoscaler is set to true, this will define the maximum number of the runners deployed. If it is set to false, then it shows the total number of runners

You can browse over the full set of APIs available in RaaS by importing the API definition from this Url:

api-details#api=github-customer-v2https://portal.boschdevcloud.com/api-details#api=github-customer-v2

Here is a tabular visualization of the APIs available via the portal.



As noted earlier, RaaS APIs can be invoked from a number of tools and contexts, including an external API tool such as Postman.

api-details#api=github-customer-v2

9.1.3. Advanced options

Update existing Runner Deployment

You can update your existing Runner Deployment by executing PATCH Update GitHub runner API call. In the Parameters section, choose the environment were your deployment was deployed, provide the name of the RaaS project and the Id of the runner deployment you want to update. Next in the Body, fulfil the details you want to change, the one which you want to keep as they are can be removed from the Body call. Click on Send button and wait few minutes, till the Runner will be redeployed. To check the status and the details of the RaaS project, please execute the GET RaaS Project API call.

▼ *Click to reveal full information...*

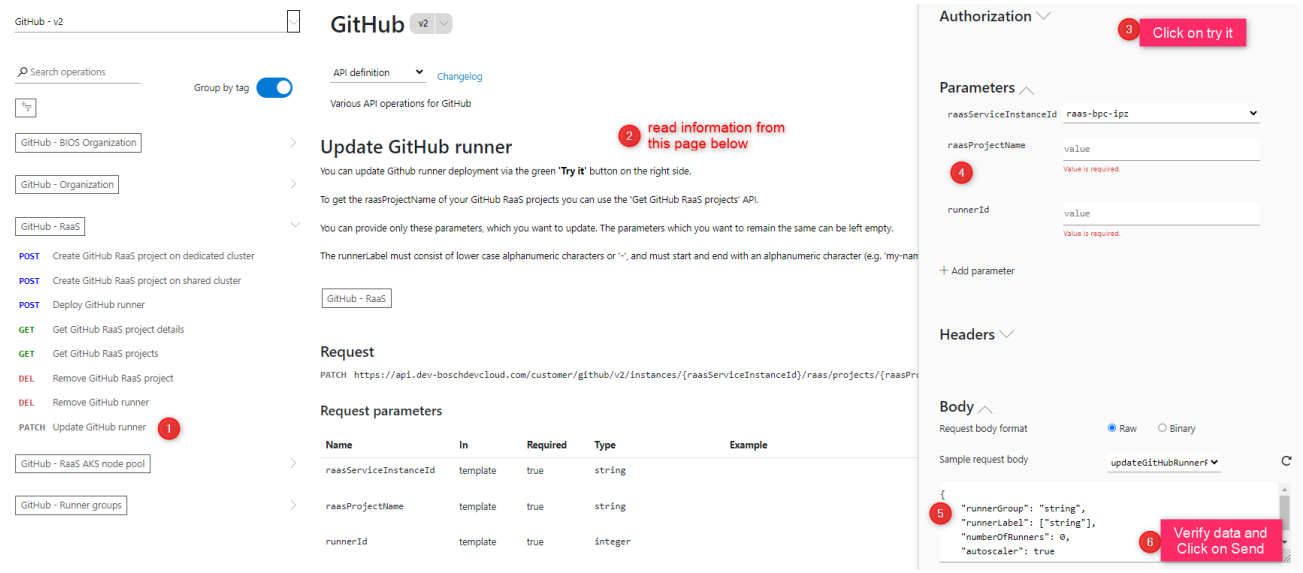


Figure 6. raas-runner update api call

▼ *Click to reveal full information...*

```
curl -v -X PATCH
"https://api.boschdevcloud.com/customer/github/v2/instances/{raasServiceInstanceId
}/raas/projects/{raasProjectName}/runners/{runnerId}" -H "Content-Type:
application/json" -H "Cache-Control: no-cache" -H "bdc-api-key: my_key" --data-raw
"{ \"runnerGroup\": \"string\", \"runnerLabel\": [\"string\"],
\"numberOfRunners\": 0, \"autoscaler\": true}"
```

This PATCH-oriented API invocation accepts the following key arguments:

- bdcId - the id associated to your BDC space
- my_key - the primary or secondary key associated to the (product) subscription you created earlier
- raasServiceInstanceId - the id of the instance where your runners are deployed
- raasProjectName - the name of the RaaS project you want to deploy the runners to
- runnerId - the Id of the runner you want to update
- runner group - a string uniquely identifying your runner group
- runner label - a unique label that will identify the runner instance (use the workflow run-ons to bind to a particular runner)
- autoscaler - boolean flag to enable K8s auto-scaling
- numberOfRunners - if the autoscaler is set to true, this will define the maximum number of the runners deployed. If it is set to false, then it shows the total number of runners

9.2. Cost for Runner as a Service

	Costs per month	Remark	Costs charged against

Shared Instance (BPC / Azure)	35€ per runner without autoscaler enabled or 35€ per runner deployment with autoscaler enabled	-	Customer cost center provided during ordering process
Dedicated Instance (Azure)	35€ per runner without autoscaler enabled or 35€ per runner deployment with autoscaler enabled	Azure Infrastructure consumption costs depending on resources used at Microsoft Azure are charged against cost center of the Cloudspace	Customer cost center provided during ordering process

We are working with GitHub to check options to change this to a more dynamic charging (pay per minute).

9.3. RaaS - Dedicated Instance

RaaS dedicated instance is currently only available in Cloudspace environment. Therefore, the precondition is to have a shared or dedicated Cloudspace tier of which one AKS is needed to deploy the runner by creating a separate node pool.

General information about Cloudspace Service and differences between shared and dedicated Cloudspace tiers can be found in Cloudspace section.

As Cloudspace tiers are owned by the customer, one of the main advantages is that you have access to the runner infrastructure. This results in the responsibility of the customer to take care of maintenance such as AKS upgrade, monitoring etc.

9.3.1. AKS Node Pool Management

By default the BDC creates one node pool to your AKS automatically. The scaling option can be enhanced by changing the set-up of the node pool by additional nodes to the node pool. By default BDC node pool includes a maximum of 3 standard nodes.

Via the BDC-Portal you can

- deploy additional AKS node pools, the following two machine types can be used:
  - standard - which represents the machine type Standard_D2ads_v5
  - performance - which represents the machine type Standard_D8ads_v5
- get list of AKS node pools
- remove AKS node pool(s)
- update AKS node pool (incl. default node pool created by BDC) to increase/reduce number of nodes and/or change machine type .

9.3.2. Runner configuration

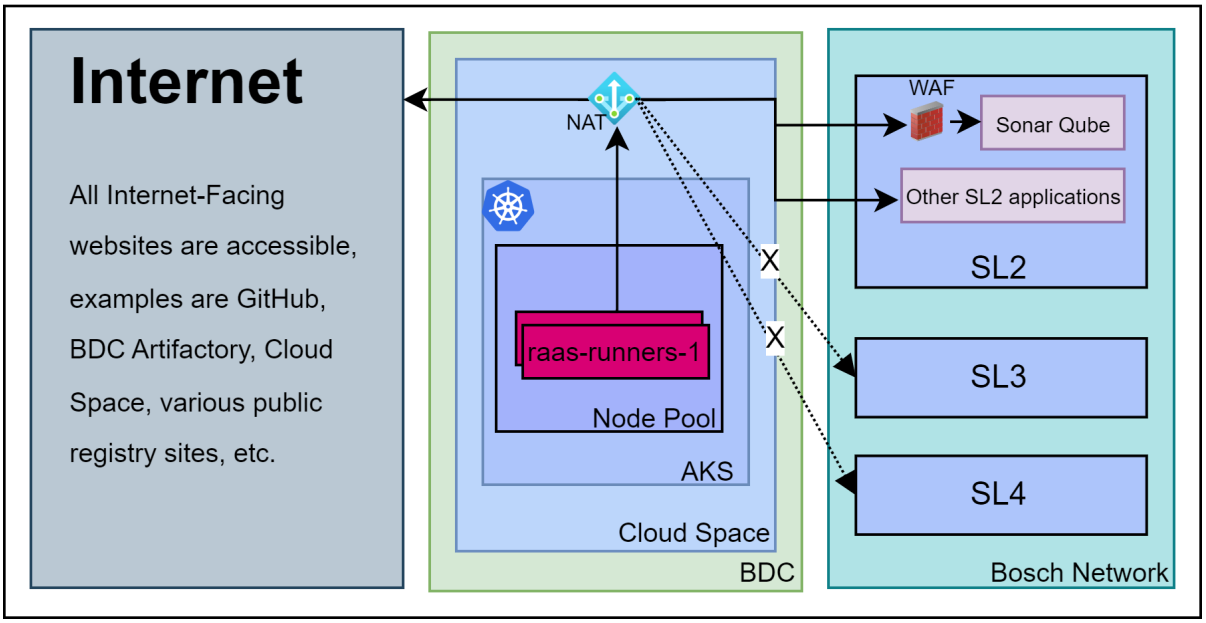
Runners are configured without any resource limitation. But please be aware of the nodepool instance size you choose during the runner deployment.

Runner Image: summerwind/actions-runner (based on Ubuntu 22.04)

A container job with custom docker image can be configured. A detailed documentation is at runner configuration section.

9.3.3. Architecture

Runners are located in Cloud Space - AKS in a nodepool deployed by us. Refer to below image to understand the Network connectivity. As these runners are on Internet (Cloud Space), You will not have restrictions to reach internet based resources.



9.3.4. Network

Egress (Outgoing) Traffic to Internet

Traffic from runner to internet is routed via a Azure NAT Gateway(Network Address Translation). This provides you a static IP for egress traffic which you could whitelist in your services if required. E.g: SonarQube

9.4. RaaS - Shared Instance (BPC)

As runners can be deployed to BPC-IPZ (Private Cloud), the RaaS team has configured proxy and no_proxy based on guidance shared in the Bosch Private Cloud documentation.This configuration is required to reach resources on the Internet and on-prem, respectively.

Runners are deployed in so called projects in OpenShift terms which is simply a namespace in k8s terms. Once you deploy runner, a single runner will be always listening for jobs and it could scale up based on the initial configuration to maximum of 10.

A big advantage of runners in BPC-IPZ is that you could access Bosch internal resources like rb-artifactory and many BCN resources. New access to on-prem resources can be requested, for which you will need to contact BDC team. A full list is provided below.

9.4.1. Runner configuration

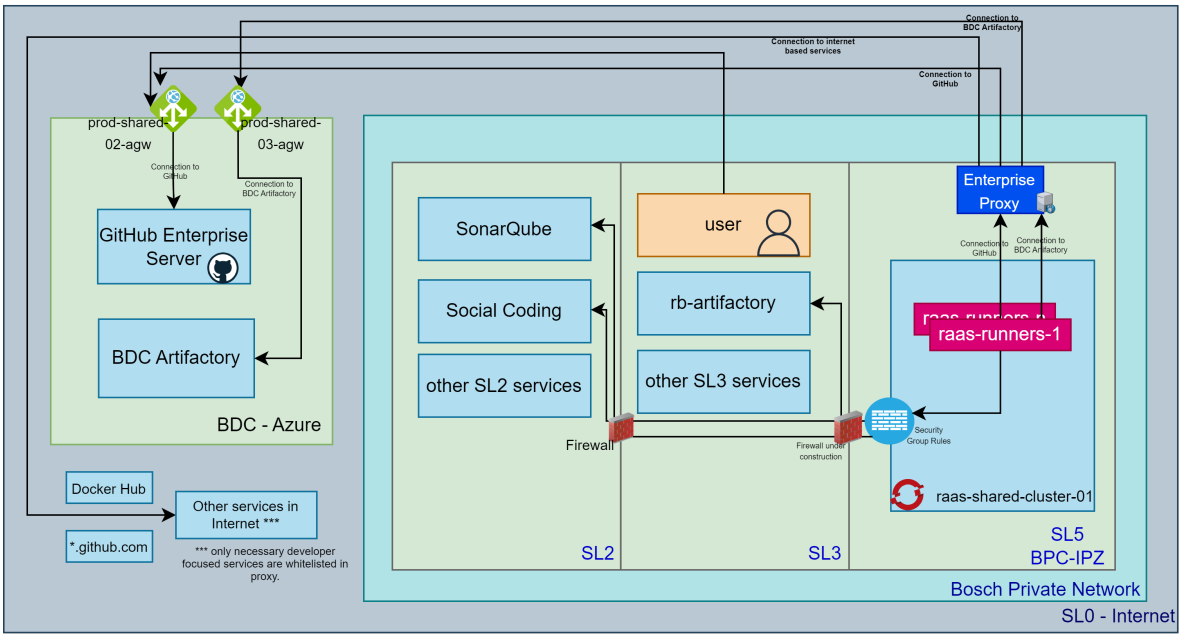
Runners are configured with 1 vCPU and 8 Gi of Memory.

Runner Image: summerwind/actions-runner (based on Ubuntu 22.04)

A container job with custom docker image can be configured. A detailed documentation is at runner configuration section.

9.4.2. Architecture

As runners are located in BPC-IPZ (Private Cloud), we have configured proxy and no_proxy as suggested by Bosch Private Cloud team in their documentation. This is required to reach resources in the internet and on-prem respectively.



9.4.3. Network

Egress (Outgoing) Traffic to on-prem services

Access to various on-prem in different SL zones is managed through network-oriented security groups and EIT Firewall. The existing whitelisted domains are documented below and for additional whitelisting, kindly reach BDC team. To know more refer to this BPC document.

The list of IP addresses for the RaaS Shared OpenShift worker nodes: "10.140.170.195","10.140.171.244","10.140.171.47","10.140.171.147","10.140.170.142","10.140.170.173", "10.140.170.235"

The egress traffic from RaaS Openshift worker nodes to the resources located in the EPZ is opened by default. If you want to connect from the runners to any of your EPZ resource, you would need to

open the ingress traffic for the listed IPs above only on your site.

Egress (Outgoing) Traffic to Internet

In the case of Egress connections to Internet-based resources, they are routed through the Proxy Server. You could also refer to **IPZ specific** Network diagram in BPC document - Egress (Outgoing) Traffic to Internet. Please be advised that the proxy server allows only https traffic.

Proxy-Service configuration

Below you can find the list of whitelisted domains for the Proxy-Service for RaaS:

*.dev-boschdevcloud.com
*.qa-boschdevcloud.com
api.loganalytics.io
ml.azure.com
adb-5407587042408609.9.azuredatabricks.net
data.pyg.org
install.determinate.systems
cache.nixos.org
zero-to-nix.com
codeload.github.com
objects.githubusercontent.com
api.github.com
ghcr.io
github.com
pkg-containers.githubusercontent.com
www.mathworks.com
dl.fbaipublicfiles.com
edconnect.edcastapi.com
*.edcastpreview.com
edconnect.edcastapi.eu
*.edcast.com
*.edcast.io

- Only development-focused URLs are whitelisted in the Proxy server. If the connection does not work to a resource on the internet via proxy, get in touch with us at Jira Service Desk. We will investigate if it is really required and proceed with your request.
- In case connection to the internet times out, you might have to adjust or pass proxy details in your tools (e.g.: Docker).
- By default Enterprise IT domain names (DNS) are not resolvable in the Bosch Private Cloud environment. A whitelisting process is required to make them resolvable in the BPC. Incase DNS resolution is failing, get in touch with us.
- Just a heads up, BPC runners and SL3 network connectivity are being controlled by Openshift feature for now. BPC team is planning to add a firewall soon.

Table 6. Allowed standard connections to services in other SL zones

Service Name / Description	Service URL	IP Address	port
RB Artifactory	rb-artifactory.de.bosch.com	10.35.31.174	443
RB Artifactory (DAD)	fe-artifactoryha.de.bosch.com	10.35.29.147	443
Docupedia	inside-docupedia.bosch.com	10.3.34.113	443
Track+Release	rb-tracker.bosch.com	10.3.34.136	443
Sonar Qube - SL2	sonarqube.dev.bosch.com	10.73.18.8	443
ALM	rb-alm-05-<p\q\d>.de.bosch.com	10.139.231.79, 10.139.231.12, 10.58.192.195	443
ALM	rb-ubk-clm-02.de.bosch.com	10.139.230.227	9443
Social Coding platform	sourcecode.socialcoding.bosch.com	10.139.209.84	443
Bosch Container Registry	bcr-de01.inside.bosch.cloud	10.140.180.156	443
Jmaas/Fossid		10.139.48.0/22	443,50014
OSD Mirror	http://mirror-osd.de.bosch.com	10.139.92.186	80
EMEA APIM	rb-jmaas.de.bosch.com	10.58.194.16	443,8075
RB-secrets	rb-secrets.bosch.com	10.73.86.60	8200
IDM API		10.35.29.22	443
IDM API	ews-bcn.api.bosch.com	10.35.31.9	443, 8075
DTR	rb-dtr.de.bosch.com	10.139.12.77	443
BPC-IPZ			Any
BPC-EPZ		10.143.0.0/17	Any

9.5. RaaS - Shared Instance (Azure)

RaaS shared Azure instance is currently deployed in the same way as a part of the Cloudspace environment. The infrastructure deployed as a part of the RaaS Azure shared instance is fully managed by the BoschDevelopmentCould team.

Runners are deployed in a seperate namespaces which are called from the API perspective RaaS project. Once you deploy runner, a single runner will be always listening for jobs and it could scale up based on the initial configuration to maximum of 10.

The runners deployed in the Azure shared instance has opened access to the Internet.

9.5.1. Runner configuration

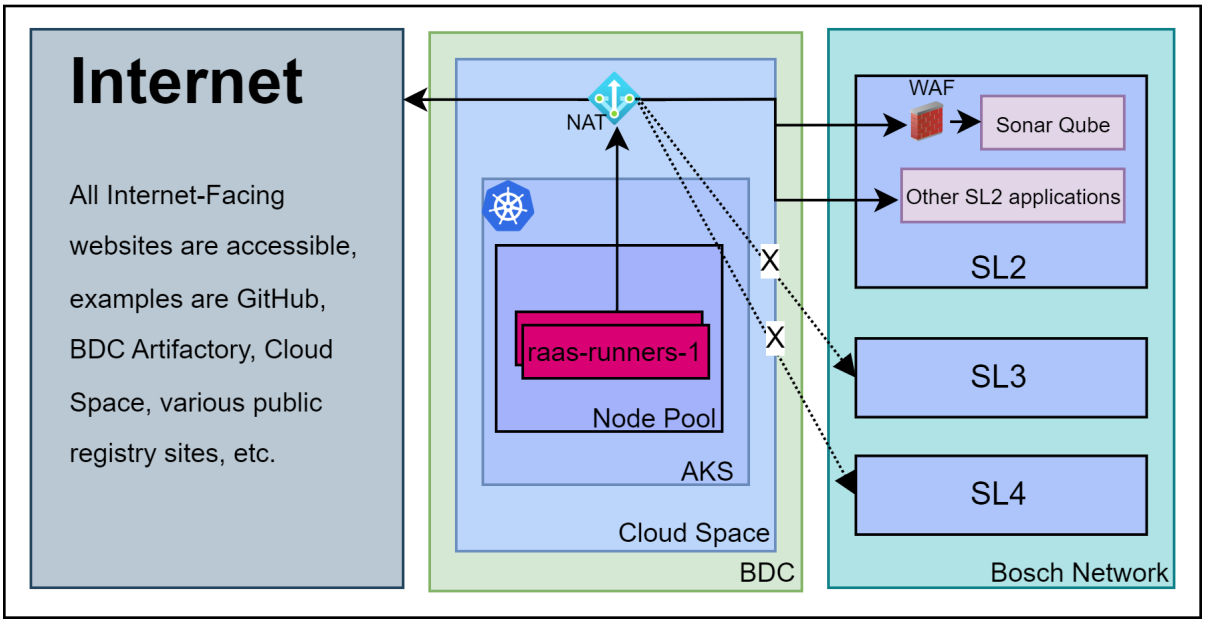
Runners are configured with 1 vCPU and 8 Gi of Memory.

Runner Image: summerwind/actions-runner (based on Ubuntu 22.04)

A container job with custom docker image can be configured. A detailed documentation is at runner configuration section.

9.5.2. Architecture

Runners are located in shared Cloud Space - AKS in a nodepool deployed by us. Refer to below image to understand the Network connectivity. As these runners are on Internet (Cloud Space), You will not have restrictions to reach internet based resources.



9.5.3. Network

Egress (Outgoing) Traffic to Internet

Traffic from runner to internet is routed via a Azure NAT Gateway(Network Address Translation). The public IP address of the Azure NAT Gateway is 104.45.45.134. This provides you a static IP for egress traffic which you could whitelist in your services if required. E.g: SonarQube

9.6. Runner Configuration

Depending on the project, you may need a unique set of tools for tasks ranging from building to deploying. Primarily you have two options for configuring runners,

- Configure runner with setup-actions.
- Container jobs with custom docker images.

9.6.1. Configure runner with setup-actions

Setup action such as `actions/setup-python`, `actions/setup-node` could be used to setup the environment. More information can be found on respective repo of each package type in public GitHub Actions organization.

Avoiding rate limit issues

We recommend using actions with PAT tokens whenever possible to avoid rate limit issues.

Most actions, come pre-installed on the appliance with GHES. When dynamically downloading package distributions, action `setup-<packageName>` downloads distributions from `actions/<packageName>-versions` on github.com (outside of the appliance). These calls to `actions/<packageName>-versions` are by default made via unauthenticated requests, which are limited to 60 requests per hour per IP, BPC runners are highly affected here as the outgoing traffic uses limited Bosch IPs. If more requests are made within the time frame, then you will start to see rate-limit errors during downloading that look like this:

Listing 9. Snippet of expected error

```
1 ##[error]API rate limit exceeded for YOUR_IP. (But here's the good news:
  Authenticated requests get a higher rate limit. Check out the documentation for more
  details.)
```

How to configure action with PAT is documented in official GitHub docs inside each actions, some are listed here - setup-python, setup-node, setup-go.

root access

In case you wish to install packages directly without actions such as `actions/setup-python` or perform tasks that require root permissions in the runner, you might run into issues with using sudo. It's because root access is disabled on the runner.

Listing 10. Error when switching to root

```
1 sudo: The "no new privileges" flag is set, which prevents sudo from running as root.
```

However, you can still gain root access on the runner by using a container job as described in below section. This will allow you to install packages or perform tasks that require root permissions.

9.6.2. Container jobs with custom docker images

You could also create a docker images for your usecase with all the required packages pre-installed and use it as base runner image. As a pre requisite, you need to build an image and store it in a

docker registry, alternatively you could also use `Bosch - Open Source Desktop (OSD)` image and configure it during each runs based on requirement.

Below is a snippet on how you could pass docker images in the workflow,

Listing 11. Container job sample code

```
jobs:
  job-name-goes-here:
    runs-on: [self-hosted, Linux]
    container:
      image: artifactory.boschdevcloud.com/repo-name-docker-local/python/python-image:latest
      credentials:
        username: ${{ secrets.ARTIFACTORY_USER }}
        password: ${{ secrets.ARTIFACTORY_TOKEN }}
```

Additional information of a container jobs can be found in GitHub documentation.

9.7. Compliance for Runner as a Service

We will provide you with a link to our compliance documentation for Runner as a Service when it is ready.

9.8. Shared responsibility for Runner as a Service

As the service is still in development, the content for this chapter are still being designed. Below is an early preview of key facts regarding Runner-as-a-Service and the self-service APIs that are now available to select customers (for early preview and evaluation).

9.9. Scope of Support

As the application provider, BoschDevCloud could support you on,

- Connectivity issues from runner to other services inside Bosch Network or in Internet.
- Infrastructure issues incase of shared cluster.

Please be advised that we are unable to assist you in creating workflows or building custom images for RaaS.

9.10. Best practices

- Stick to `https` procotol with secure ports when connecting to Bosch Internal resources/Internet (Eg.: Servers in SL3)
- If possible, try avoiding connecting to SL3 based services, sooner or later SL3 network zone will be depreciated.

9.11. RaaS FAQs

9.11.1. Technical FAQ

Can I access to Sonar Qube from Dedicated runners in AKS?

The SonarQube is now located in SL2 network zone, which is accessible from the Internet. You need to request the SonarQube team to whitelist your static egress IP in cloud space, which is a public IP associated with a NAT gateway in the location resource group.

How can I use my own custom image?

You could pass the image name in the workflow, Refer to the bios workflow here.

9.11.2. General FAQ

Runner does not pickup the job in my public repository

As documented here, GitHub recommends to only use self-hosted runners with private repositories. This is because forks of your public repository can potentially run dangerous code on your self-hosted runner machine by creating a pull request that executes the code in a workflow.
To mitigate the risk of code execution in public repositories, customers have the ability to configure Runner group deployment settings to either allow or disallow public repositories. This can be a helpful measure to prevent any unwanted actions from being executed in public repositories. This setting might prevent runner from picking up jobs in public repository.

Is it possible to get windows based runners?

Unfortunately, Windows nodes are not currently supported by Cloud Space. However, the development team is actively working on implementing this feature. Once it becomes available, we can explore the possibility of enabling it for RaaS.

I cannot find my BIOS GitHub org under my BDC with Get API in BDC Portal, what am I doing wrong?

If a BIOS organization is created without a corresponding BDC ID, then the BIOS organization is mapped to default BDC ID 1, due to this it may not appear under your BDC. To resolve this issue, kindly create a JSD ticket for the BDC team and share the necessary information such as the BIOS organization name and BDC ID details. This will expedite the process of mapping your BIOS repository to your BDC.

Can the runners from RaaS be used with GitHub Cloud?

Currently, the RaaS is offered only on Bosch Development Cloud GitHub Enterprise Server, however we are planning to integrated them with GitHub Cloud in the future.

9.12. Feedback

We look forward to receiving your feedback, which will help us to improve in the future!

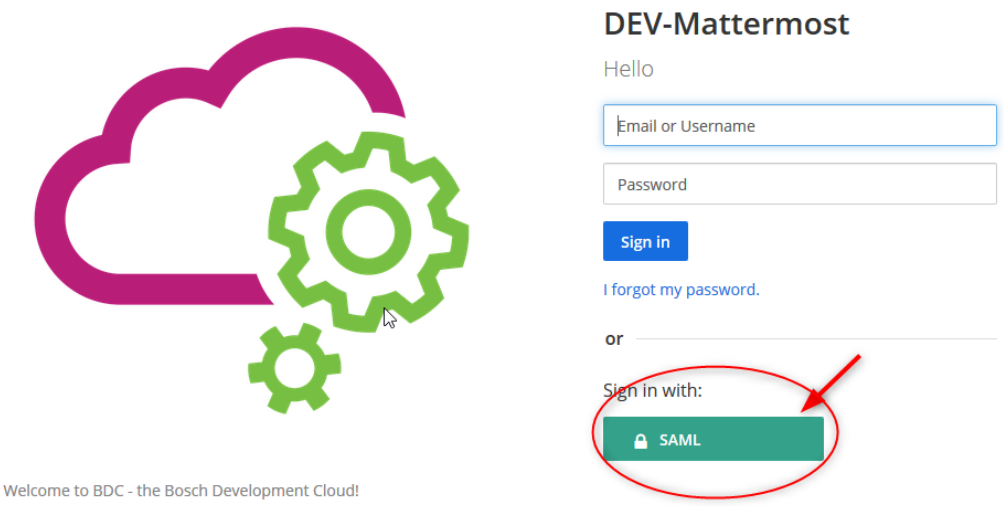
Chapter 10. Mattermost

BoschDevCloud Mattermost is a **chat-ops service** with persistent chat, search and integrations into other development services such as GitHub, Jira and many more. It is compatible with Slack, can be used via CLI and webhooks can be added on team-level.

Mattermost cannot be used as an audio/video-tool as MS Teams (and Skype) are the company wide solution for these communication channels.

Data up to SC2 are allowed.

- Go to https://mattermost.boschdevcloud.com or use your Mattermost-desktop-client
- Click on the button "SAML"



10.1. What do I get for Mattermost and how to order

You can order a team in Mattermost, you will have the possibility to create multiple channels, integrations, ...

An owner of a BDC can request a mattermost team via our BDC-Portal. Choose "My BDCs", and select your BDC, make sure you have created a "subscription" (key) and choose entry "Mattermost".

In the description you will see the IdM-roles which will be created and where all members of the team need to onboard to.

Please note, currently Mattermost-Teams names are restricted to 15 characters.

10.2. Cost for mattermost

The current costs for Mattermost@BDC:

	Costs per month	Costs charged against:
Mattermost team	no additional cost	
Mattermost@BDC License and Operations	17,46€/user/month	cost center of the user

The user can be a member of multiple teams without being charged with additional costs.

10.3. Permission management in Mattermost

The service can be access via https://mattermost.boschdevcloud.com via the Browser or via the desktop Mattermost Client.

Our Mattermost instance offers Single-Sign-On authentication based on SAML.

Each Mattermost Team has the following IdM roles available

- Admin: Can manage the team and integrations
- User: Can post messages

10.3.1. Preconditions to use Mattermost @ BDC

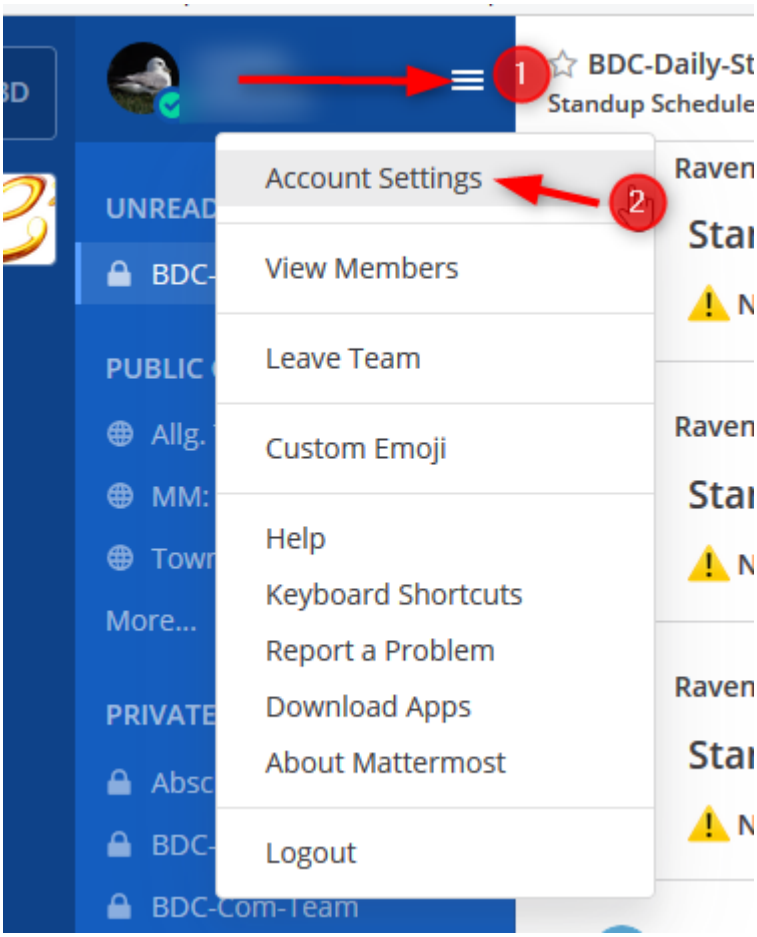
- The respective IdM-Role for the desired team is required for your Bosch-account: To request an IdM-role please follow the instructions here. Best search for "mattermost" and your team's name. This page is working for internal Bosch-users only, externals please ask your contact-person.
  - Admins need both the IdM-roles ending "BDC_Mattermost_01_teamXXXX_ admin" and "BDC_Mattermost_01_teamXXXX_user"
  - Users only need the IdM-role ending "BDC_Mattermost_01_teamXXXX_user"
- If you use a non-Bosch devices as an external employee, you also need to have MFA running

Your username in Mattermost is your **Bosch**-account!

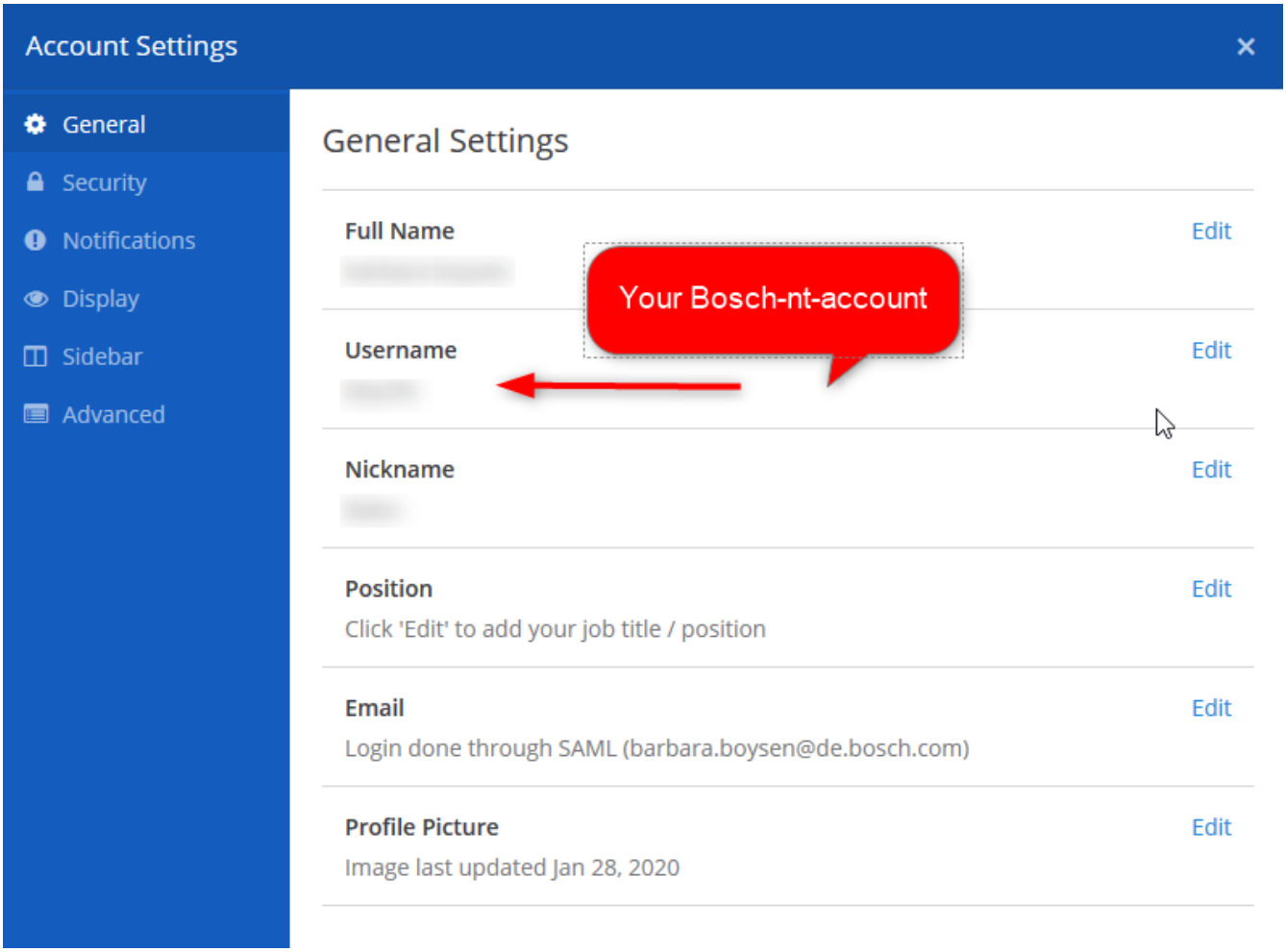
10.3.2. Check your Username, Add a Nickname

You may give yourself a nickname!

- Open the "burger-menu" of your account an fill in your nickname



- Your Bosch-account is the one you use to log onto your personal machine inside the BCN.



- Similarly you can enter a "Nickname" which your team-colleagues will see.

10.4. GitHub-Plugin

Mattermost offers an interface to the BDC-GitHub. It's a plugin which establishes the connection between Mattermost and GitHub. We have already enabled this plugin. Here are all further details about the plugin: → mattermost/mattermost-plugin-github: GitHub plugin for Mattermost

Precondition: You can watch one or many GitHub repository(s) or organization(s) hosted on BDC-GitHub.

Using the plugin means you will get a private message from the @github user, for every change in your watched GitHub org(s) or repo(s) hosted on BoschDevCloud GitHub Enterprise server (GHE).

General help about this plugin is available on here

10.4.1. Connection

Just connect your Mattermost account with your GitHub account by typing:

```
1 /github connect
```

Whenever you need help, this command will show you all available "slash commands":

```
/github help
```

10.4.2. Watching Repositories

You need to tell the plugin, which organization and which repositories you'd like to monitor. So you type this command:

```
/github subscribe <Organization>[/<Repository>]
```

You may limit the elements you want to watch by adding them into the optional parameter [features]. Just look into the /github help to see all available features. Without limits, you get a notification for each change in every feature.

10.4.3. Get current ToDos in GitHub

You can always type this slash command to get all your current todos in GitHub. But you will get also a daily reminder about them. At least, as long as you don't change the settings with /github settings.

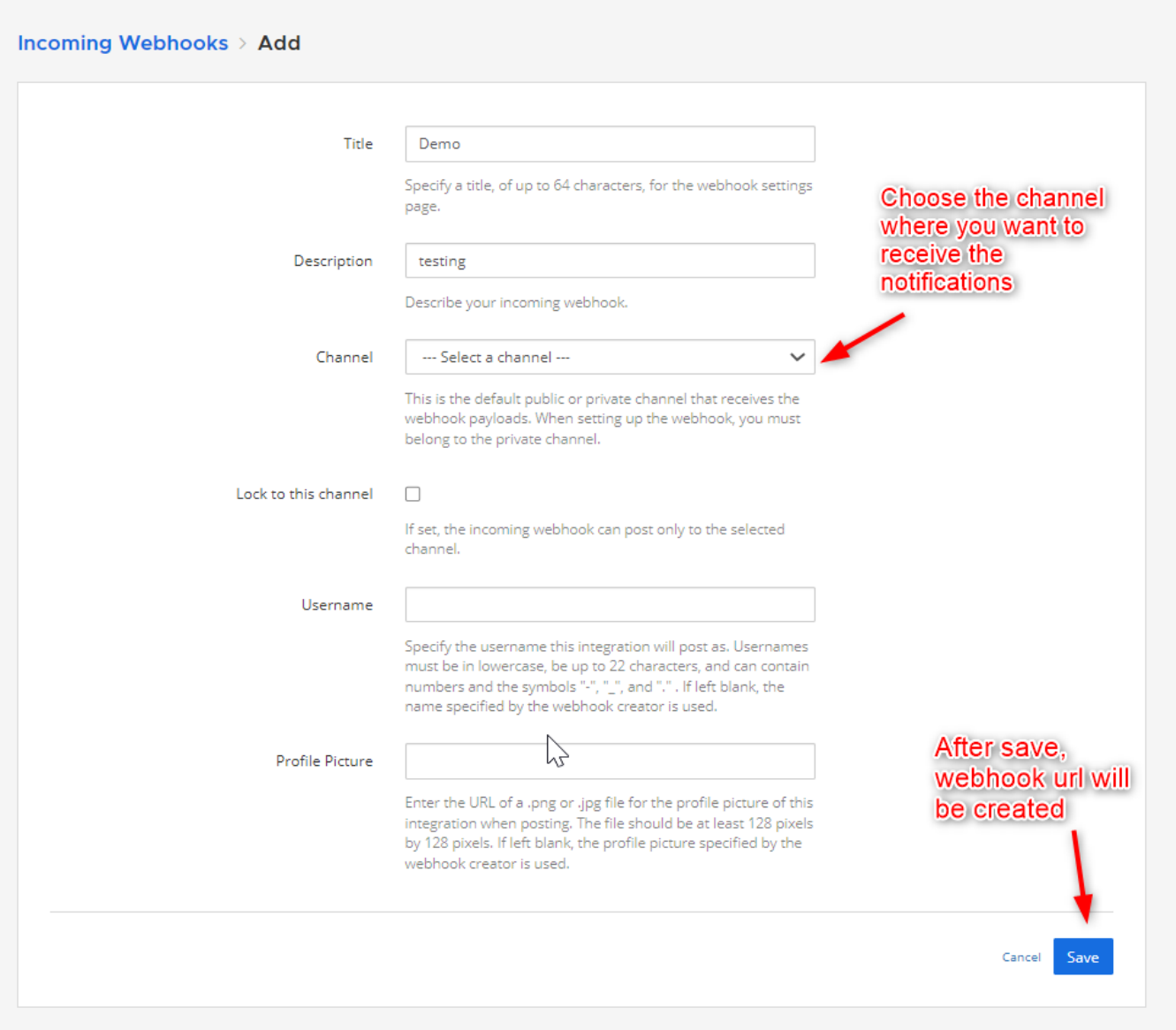
```
/github todo
```

In the result shown you may click on the links to go directly to the GitHub-items.

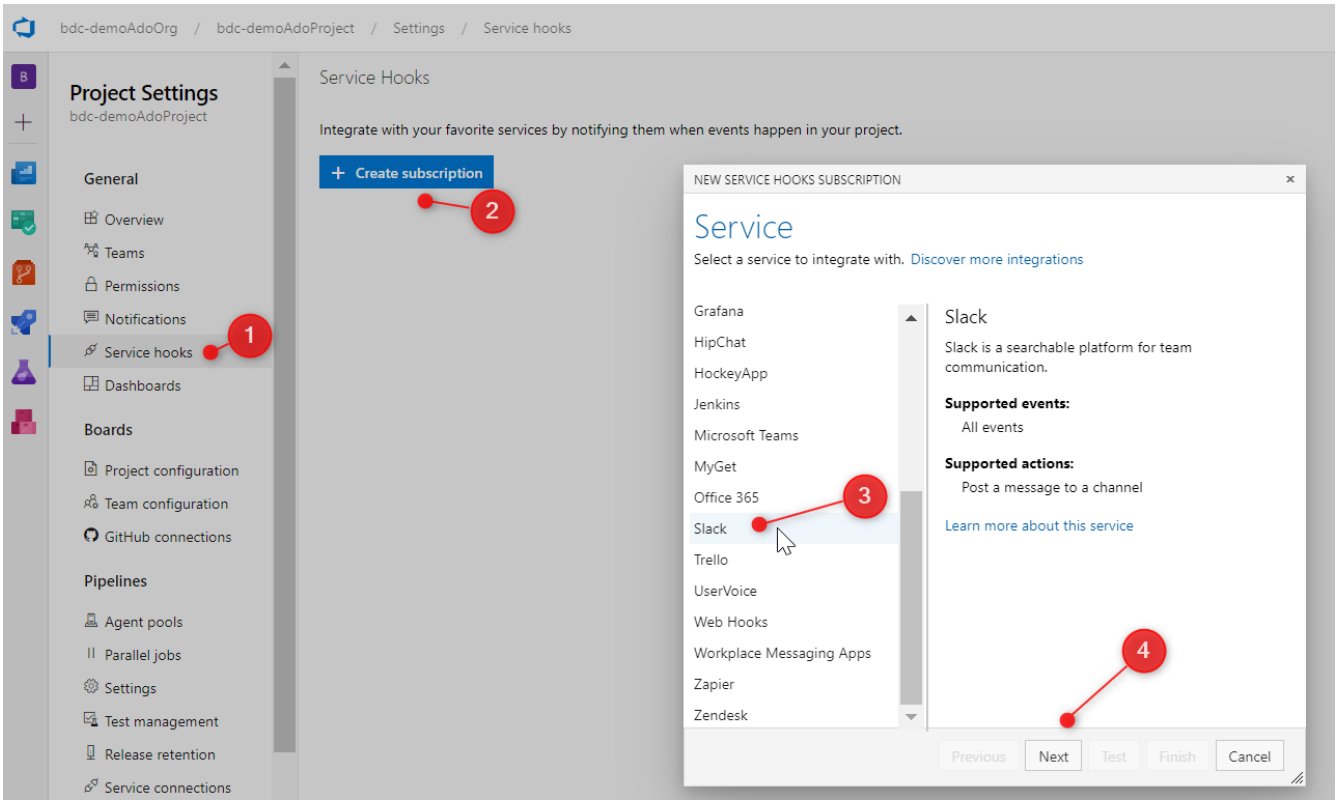
10.5. Incoming webhook configuration in Mattermost with Azure DevOps

- This chapter is written with context of posting notification message from Azure DevOps to Mattermost on an event based.
- In Mattermost, go to **Product menu** > **Integrations** > **Incoming Webhook** . Only Mattermost team admins can able to **add incoming webhook**
- Add a name and description for the webhook and select the channel to receive webhook payloads, then select **Save** to create the webhook.
- You will end up with a webhook endpoint that looks like below :

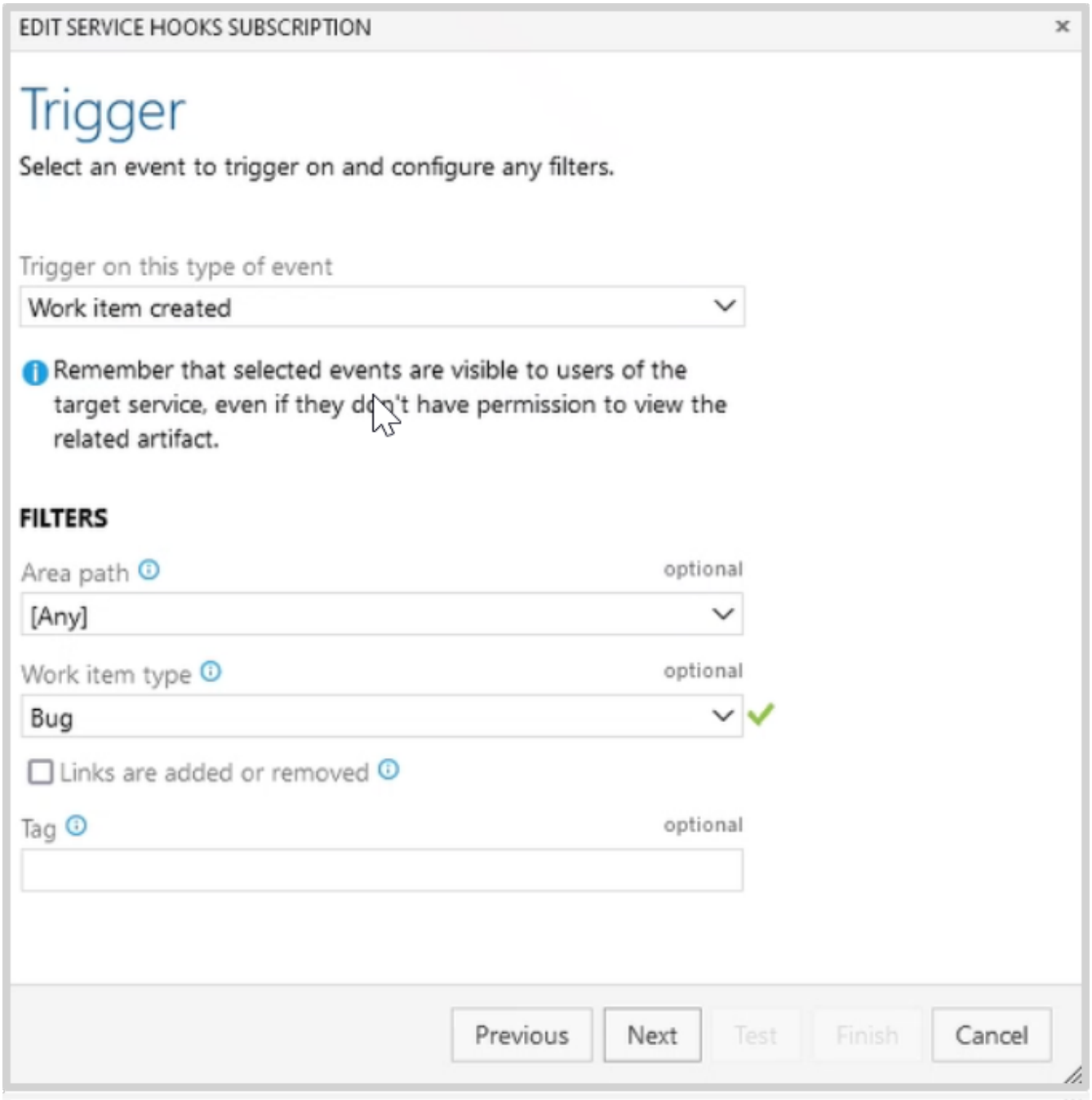
https://mattermost.boschdevcloud.com/hooks/xxx-generatedkey-xxx



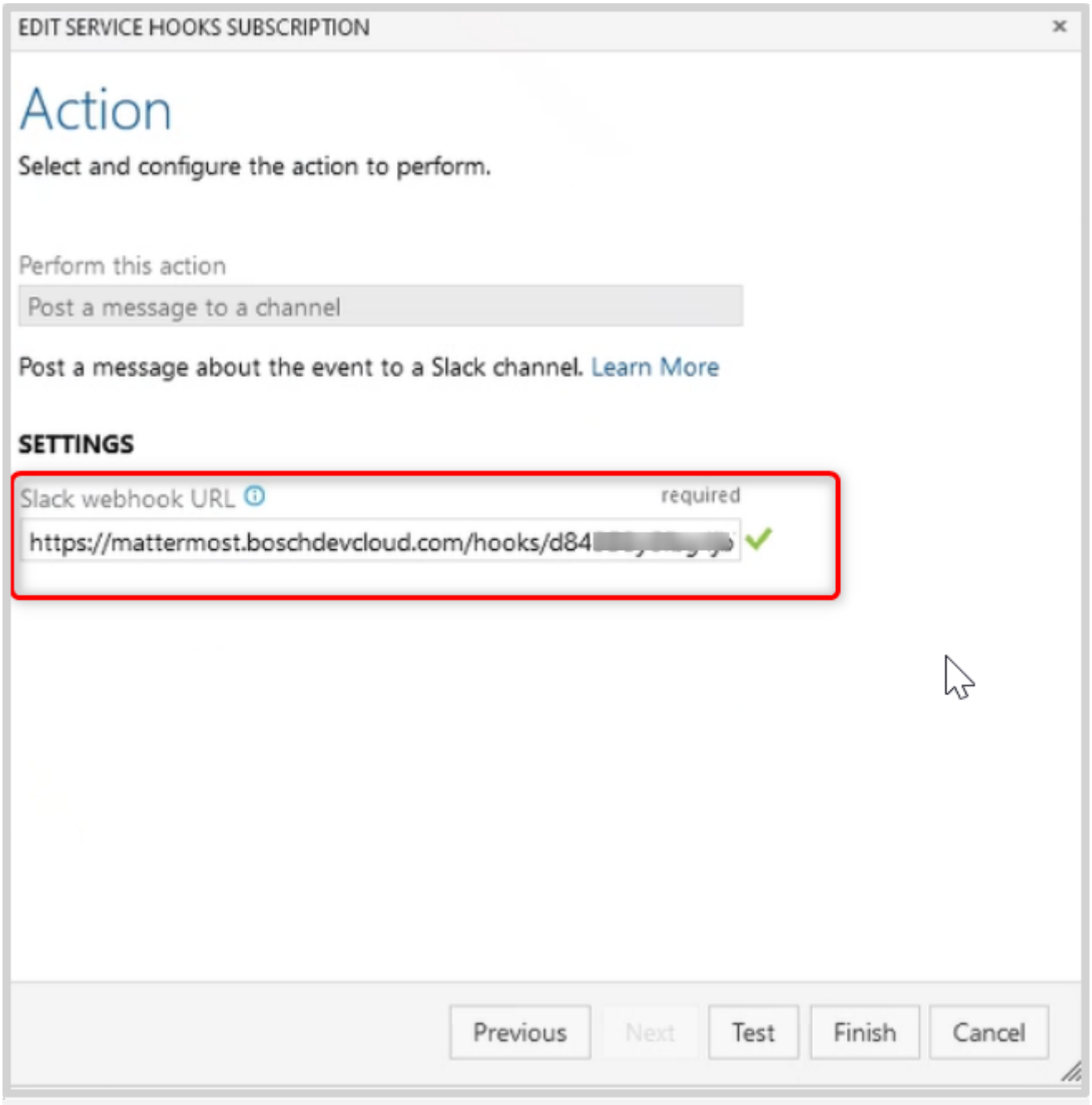
- User needs **Project Administrator** role in Azure DevOps project to add the above generated webhook url. In **Azure DevOps → Project settings → service hook → create subscription**. Please refer the below screenshot and here slack is chosen because their api is compatible with the mattermost,



- Then in the next page, choose the event on which notification has to be triggered. ex: events like Work item or Pull request created in devops and also choose the filters



- Then give the generated mattermost webhook url and click test. Once the testing succeeded, click finish to complete the service hook configuration in Azure DevOps.



10.6. Service acceptlisting

Outgoing connections from Mattermost come from following IP address 20.103.214.16. Feel free to use this IP if you need to acceptlisting connections from it.

10.7. FAQs

10.7.1. Do I need to install a client?

No - the web app has all the features from the installable clients. There are only two known minor features the desktop client has:

- the ability to connect to more than one Mattermost server (solution: use separate tabs in browser for each server)

- automatic start of the client (solution: user the "pin tab" [Right-click on the tab you want to pin and select Pin Tab from the menu] feature of your browser and place your browser into autostart of you operating system)

Thus there is no "SCCM"-package available

10.7.2. Can I use mattermost on a mobile phone?

Yes - you need to have MFA enabled. You will receive a phone call you have to verify. You can download Mattermost app from the android or Apple app store.

10.7.3. Can External Colleagues use Mattermost?

Yes, they can when they have a valid Bosch-account and the MFA enabled.

10.7.4. Export a Channel

To export/archive a channel, please refer to the Mattermost-documentation.

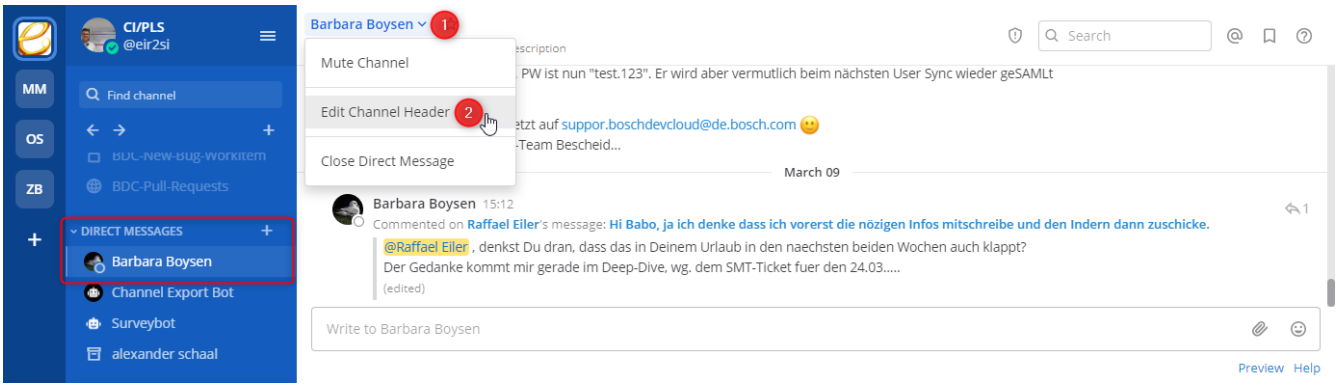
10.8. Tips and Tricks

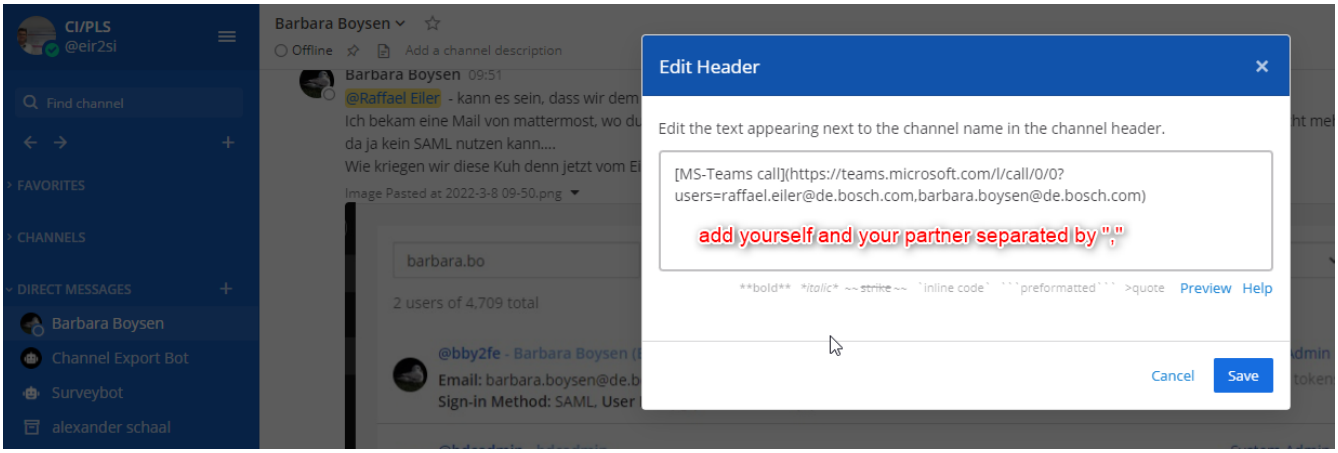
10.8.1. Microsoft Teams calls from Mattermost

To initiate a call from Mattermost to a person using Microsoft Teams you can use deep links into MS-Teams.

A typical use-case is to add such kind of deep link into header of Mattermost direct messages for quick access.

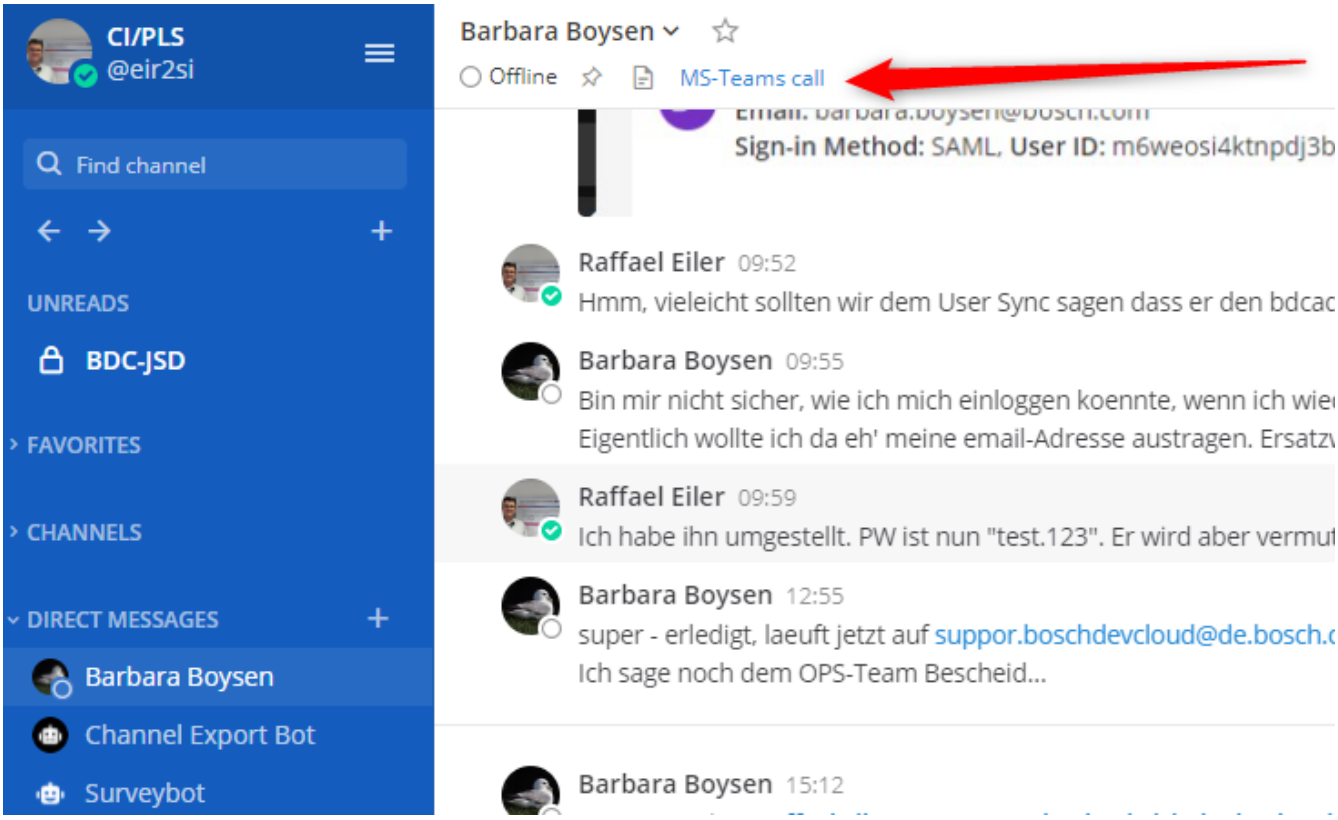
Example to add this link into direct conversation between me (Raffael) and Barbara:



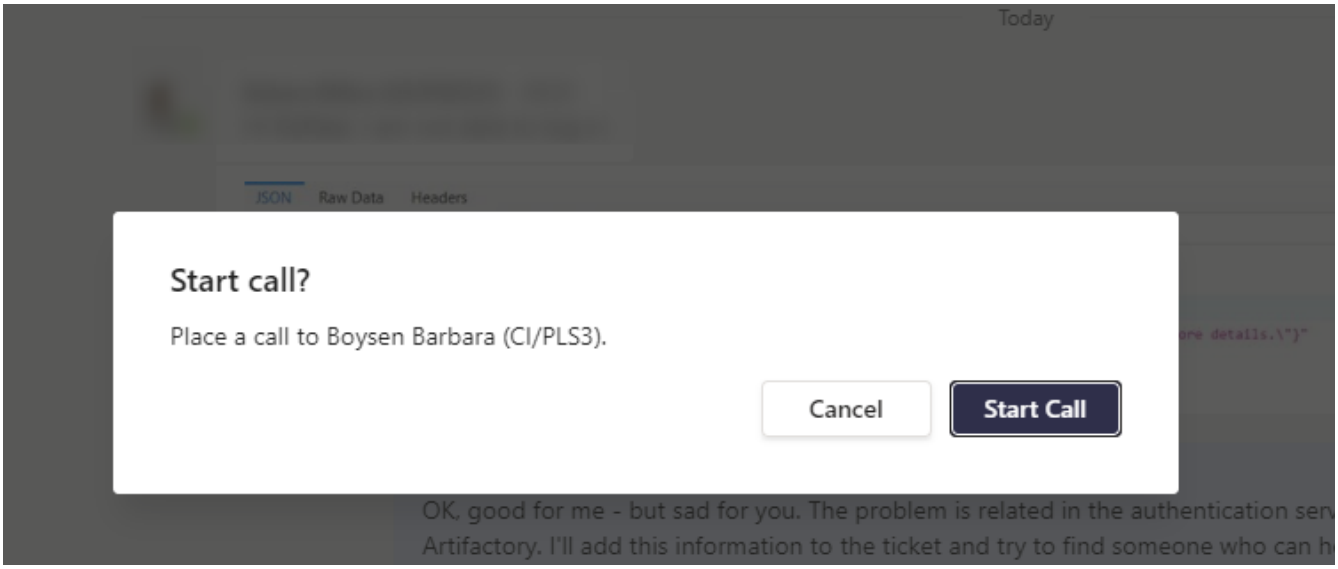


Text to insert (must be changed to your values):

```
[MS-Teams call](https://teams.microsoft.com/l/call/0/0?users=firstnameA.lastnameA@de.bosch.com,firstnameB.lastnameB@de.bosch.com)
```



Maybe you need to accept in this step that MS-Teams application shall be used in the future to make these calls.



10.9. Feedback

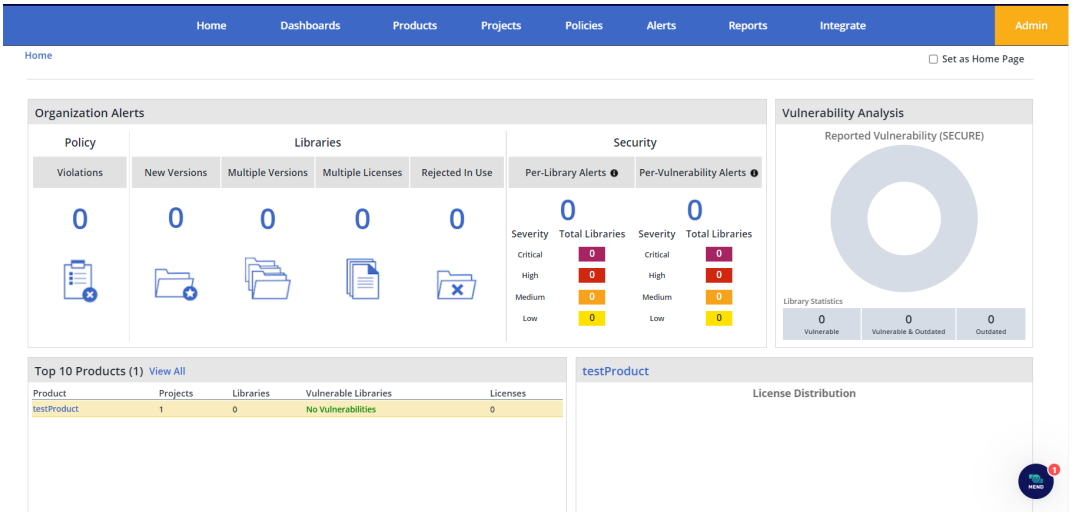
We look forward to receiving your feedback, which will help us to improve in the future!

Chapter 11. Mend

The Mend Service is supported by BDC. Mend is an open source management solution which helps development teams to identify used open source licenses in their code base and to find known security issues in used open source components.

In comparison to Fossid, the code is not scanned for open source snippets but only on component level (linux kernel, libraries etc.).
This makes scanning much faster and is used where components are either not changed by the developer or when the own code is separated from the original open source components.
In this case, Mend identifies the used components and only the own written code must be scanned by Fossid to apply to the Bosch compliance.

Mend is offered as a SaaS-Service only via:
https://app-eu.whitesourcesoftware.com/

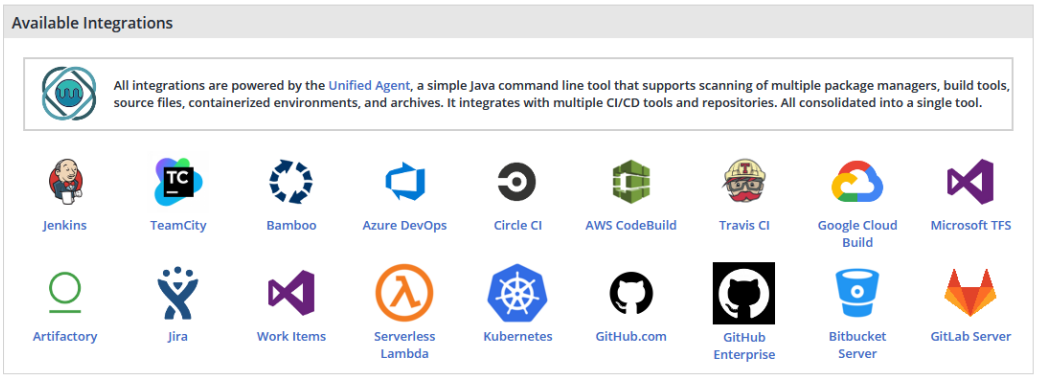


You can find the full documentation of the service on the Mend documentation web site.

11.1. What do I get for Mend and how to order

You can order a Mend Organization via the BDC portal which requires that you have permissions to order for an existing BDC project. If you are not a BDC customer, you can order a new project via ITSP. You will have the possibility to create Products, Projects and policies. You can find needed information concerning policies in the following links:
https://docs.mend.io/bundle/sca_user_guide/page/managing_automated_policies.html
https://docs.mend.io/bundle/api_sca/page/policies_api.html

You can also create multiple integrations like Github etc. in your Organization.



The order process includes a procurement activity to get the required license. Such processing can take up to some weeks. We will setup the Organization for you already but you will have to wait for the license before you start using it. You will be responsible to ensure you stay in compliance with the license (number of used seats).

11.2. Cost for Mend Service

There are two cost points for Mend:

- costs for the BDC Operations (configuration of SSO, Idm Integration, adding the Org to Central Bosch Org).
- costs for the Mend licenses itself.

Below the overview:

	Costs per month	**Remark**	**Costs charged against**
BDC Operations for Mend	4,14€/user	-	Customer cost center provided during ordering process
Mend License	Depending on number of contributors who contribute to the orgs/repos defined in Mend	For a first estimation of the License cost check out the Mend Website. Based on your request an individual offer will be requested which then includes the Bosch discount. Please contact the License Team (in person currently Bernhard Baumheuer) for the procurement of the licenses.	Customer cost center provided during ordering process

11.3. Permission management in Mend

A summary of the Standard BDC Idm Access Rights can be found under the POST API Calls in the

BDC Portal.
The IdM Access rights are organization specific. Therefore, each organization has to be registered in the BDC Portal. Users can only see and access organizations for which they have the corresponding IdM access right. If you use the BYOG feature, the Azure AD Group needs to be registered to one or multiple organizations in the BDC Portal.

As soon as users do not have any IdM Access rights of your Organization assigned to them anymore or they are not part of any of your own Azure AD Groups, the users will be deleted through our users synchronization process.

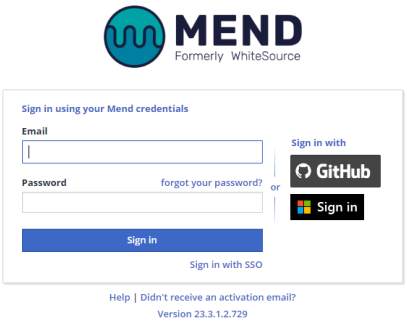
User synchronization from IdM to Mend:
The User Synchronization currently starts at 6am,14am,10pm (EST) and takes around 5 mins.

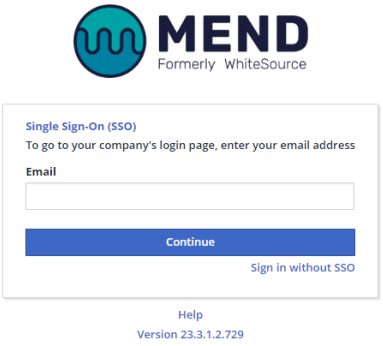
Use of Service Users:
Service users are users created directly in Mend and should NOT be used as local accounts are not allowed. For automation use cases, use Bosch technical accounts instead.

11.4. Authentication

We activated SAML-Authentication for all Bosch Mend Organization:

To login via SAML for the first time, just click on "Sign in with SSO" and provide your email address to get forwarded to the Bosch AAD Tenant.





11.5. Feedback

We look forward to receiving your feedback, which will help us to improve in the future!

Chapter 12. Project Specific Environment - aka "lab"

Please be aware that we will discontinue the creation of new labs. Labs will still be supported for the next months but we will work with the owners of existing labs to find better service offerings we can provide. Everyone who needs a new lab should consider either Cloudspace or Runner as a Service.

The Lab is based on Azure Service DevTest Labs. It provides features to manage VM infrastructure for team.

We have added an optional Azure Kubernetes Service to the lab. If required, you can also request the configuration of an HTTPS access to resources in the lab.

All services in the Lab are connected to a virtual network. This network is deployed in the region **west europe** and all resources connected to this network need to be deployed in the same region.

12.1. What can I order?

Different services are provided,

- Add virtual machines, both linux and windows
- Add other Azure services into your virtual network which do not require a public IP
- (optional) Use Bastion Host to access those virtual machines inside your BDC-environment.
- (optional) Use an Azure Kubernetes Service (AKS)
- (optional) Use the Azure Container Registry (ACR)
- (optional) Use Storage Accounts (blobs, files, queues, tables, disks ...)
- (optional) Internet access to web based resources inside the lab
- (optional) Authentication proxy to protect web based resources inside the lab

12.2. Pricing

	Costs per month	**Remark**	**Costs charged against**
Lab@BDC	30,86€/user	Each user is charged only once with the basic fee - regardless the number of labs he is admin or user	cost center of the **user**

Lab@BDC Azure Infrastructure consumption	depending on your resources used at Microsoft Azure	Costs are coming via C/IDA22 from Azure for e.g. VMs, storage etc. This costs are sligthly higher than the one you find in the Azure Portal	cost center of the **lab**
(opt.) Lab@BDC Internet Access	75€	required for access to web-based resources inside the lab from the internet	cost center of the **lab**
(opt.) Lab@BDC Authentication Proxy	500€	Enforces user authentication for access to web-based resources inside the lab from the internet	cost center of the **lab**

12.3. How to order?

1. Order Lab via BDC Portal → My BDC → BDC <ID> → Lab-v1
2. Lab can be created with an API call - POST *Request new Lab*, Click on "*Try it*"
3. Fill in required details,
    - **master_of_role** - The master of role (MoR) approves IdM workflows for this service. MoR must be organizational offices! A single user ID is not possible. Only a single organizational office will be accepted, something like "XY/ABC1"
    - **technical_contact** - The technical contact of the service. This contact defines the operator of the services from an EISA perspective and use NT ID, not user name.
    - **costcenter** - The cost center which is charged with the service.
    - **description** - A short naming description for your service e.g. 'BD/PLS3 - BDC'.
    - **include_bastion** - Deploys a bastion which is required to connect to your vm ('true'). Default is 'false'. Refer Azure Bastion section for more details.
4. Click "Send" → use GET *My Labs* API call on the same page to get IdM roles.
5. You will be provided with 2 IdM roles for admin and user groups, to know more about these 2 groups, IdM roles
   . You can start assigning users in IdM roles after 1 hour. If you have any trouble in finding IdM roles after an hour, you can reach BDC support team as this may happen due to incorrect details during lab creation API call.
6. Your lab will be created in background within 24 hrs and you can access lab as mentioned in below section.

12.4. Access and permission management

12.4.1. IdM roles

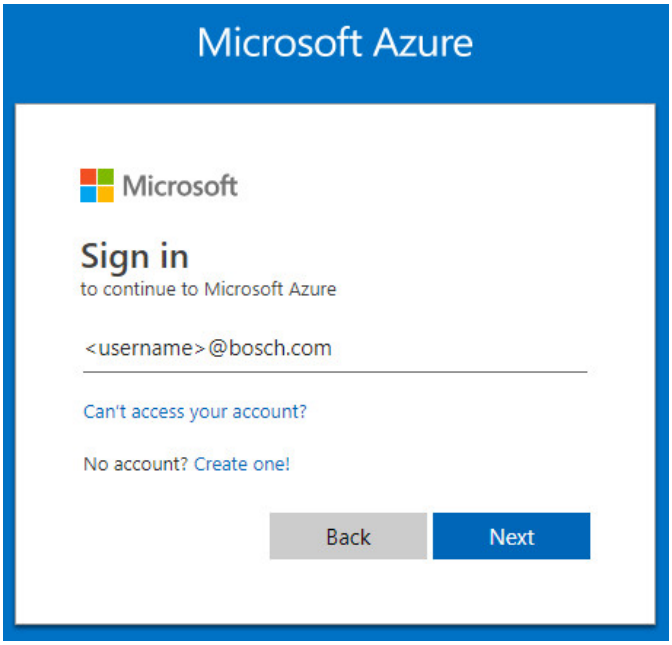
- Access to BDC Lab is provided via 2 IdM roles with RBAC roles - Contributor and Reader.
- These roles can be used to grant access to the lab-specific resource group.
  - IDM2BCD_RB_BoschDevCloud_labxxxxxx_admin
  - IDM2BCD_RB_BoschDevCloud_labxxxxxx_user

The `user` IdM role maps to the reader role, the `admin` IdM role maps to the contributor role of the corresponding resources group.

If you already are a member of the IdM BoschDevCloud Target system you can assign your user to the respective IdM groups. If not, more information about the IdM system can be found in the chapter User-Management of the user-guide documentation.

12.4.2. Azure Portal access

1. To access and use your BoschDevCloud lab, please go to Azure Portal and login with your user account and password. Keep in mind to use the username as following:
   <username>@bosch.com
   You will be redirected to the main page of the Azure portal site.



1. To access your resources, go to Resource groups in Azure portal and then click on resource group with your lab number (Naming convention: labxxxxx-rg). You can create resources and play around.

Access will be provided as Contributor or Reader with Resource Group scope.

12.5. Automated access via az cli or Terraform

To access Azure resources in your lab via automation tools you need to use an "App registration".
App registrations are "Azure AD" objects which replace service accounts from Windows Active Directory.

Every Lab comes with a preconfigured app registration which has "contributor" permissions on your lab resource group (e.g lab000001-rg).
With it, you can not only access your resources but also create new ones.

You can find the necessary id and password of the app registration in your default Lab Key Vault under the category **Secrets**, named:
- appRegistration-01-id-utf8
- appRegistration-01-password-utf8

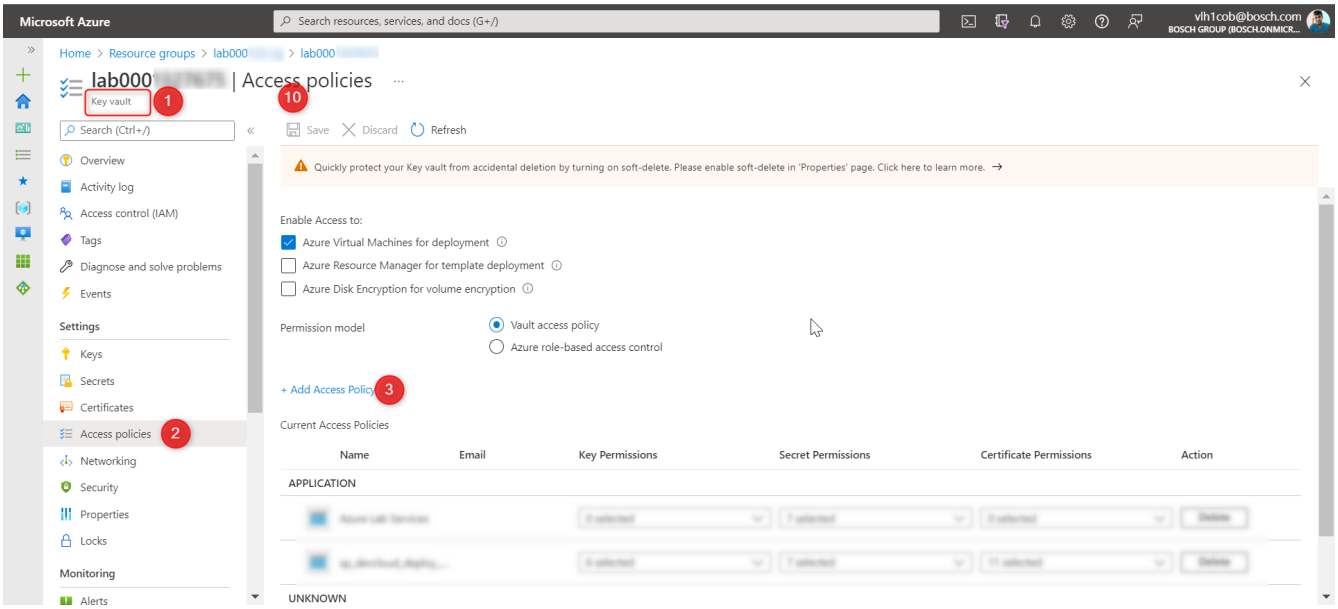
For compliance and security reasons, these secrets are rotated once a year. We usually append new ones about 20 days before they expire and delete old ones after they expire. During this period, both old and new secrets work. You must check your key vault for new secrets and update your systems.

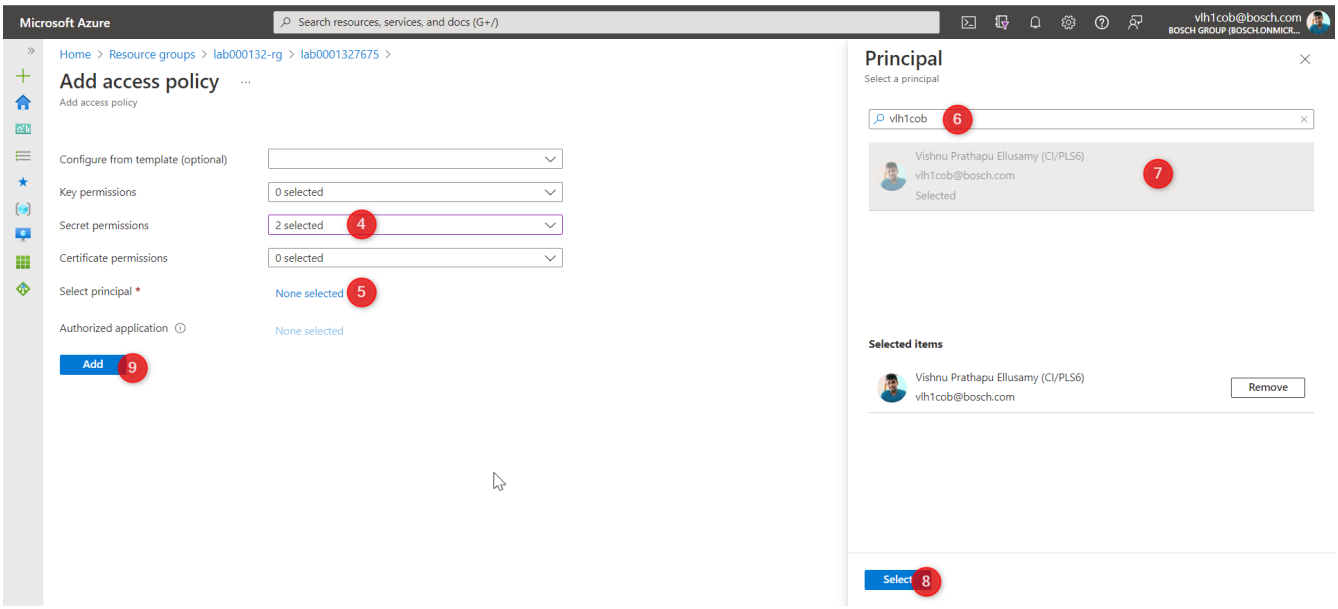
12.6. Lab Keyvault access and Guidelines

- We create a default Key Vault (Naming Convention: lab000xxxxxxx) to store the credentials that are managed by BDC.
- BDC manged service principal (appRegistration-01-id-utf8 and appRegistration-01-password-utf8) credentials and ACR Credentials are stored here. You can access it without modifying or adding other secrets to this Key Vault.

Users are requested to create their own keyvault for storing the user specific/application specific secrets.

- Lab admins have to avail themself access to default Lab Keyvault by changing key vault settings as below,
- key vault (lab000xxxxxxx) → access policy → + Add Access Policy → Set secret permission to **Get, List** → set "Select principal" as users who need access → add → Save.





12.7. Azure Active Directory Identities and admin consent

For running services and queries of the AAD, you need Identities in AAD. Those could be service principals, Enterprise Apps, or other types of identities. Whenever those identities require permissions in AAD, e.g. to query membership of groups, an admin consent is required. There are two ways to get this consent, chose the better option for your use case:

- https://inside-docupedia.bosch.com/confluence/display/dirservices/%232+App+Consent+Approval+Workflow
- https://inside-docupedia.bosch.com/confluence/display/dirservices/%233+Pre-Approval+Bosch+Development

When you develop Azure Functions you also need to get an "Admin consent" which you can request in advance via the following steps:

- Add an Identity provider to the App (Microsoft)

- Add GroupMember.ReadAll delegated API permission
- Write the email based on the template from Docupedia.

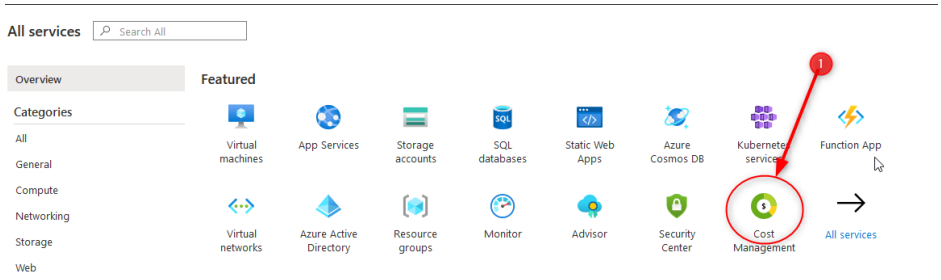
Usually the request requests one working day, sometimes even the same day

12.8. Check the costs in your environment

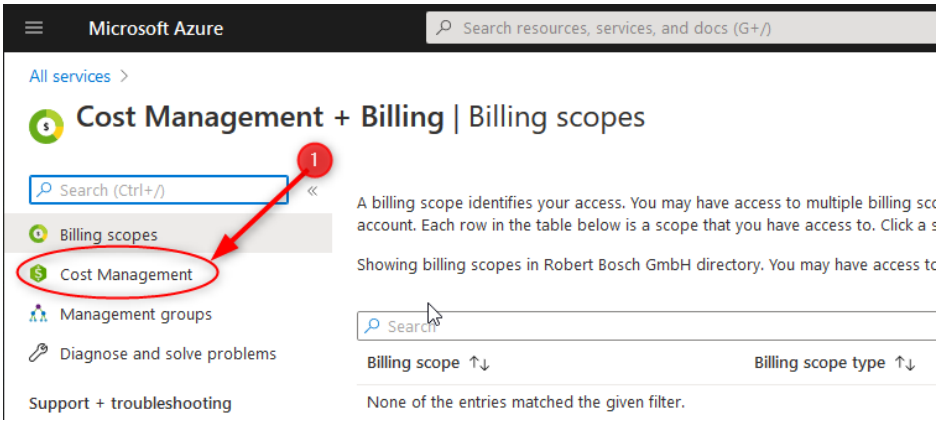
For privileged users Azure provides the possibility to check the costs for the various resources. Privileged means a user need to have the "Contributor"-role to their "full" lab. That means: Users only using services of BDC like GitHub, Artifactory, Mattermost or Jira cannot check costs. Please contact us if you're an admin of an existing environment and you want to check the costs.

Here is a short how-to:

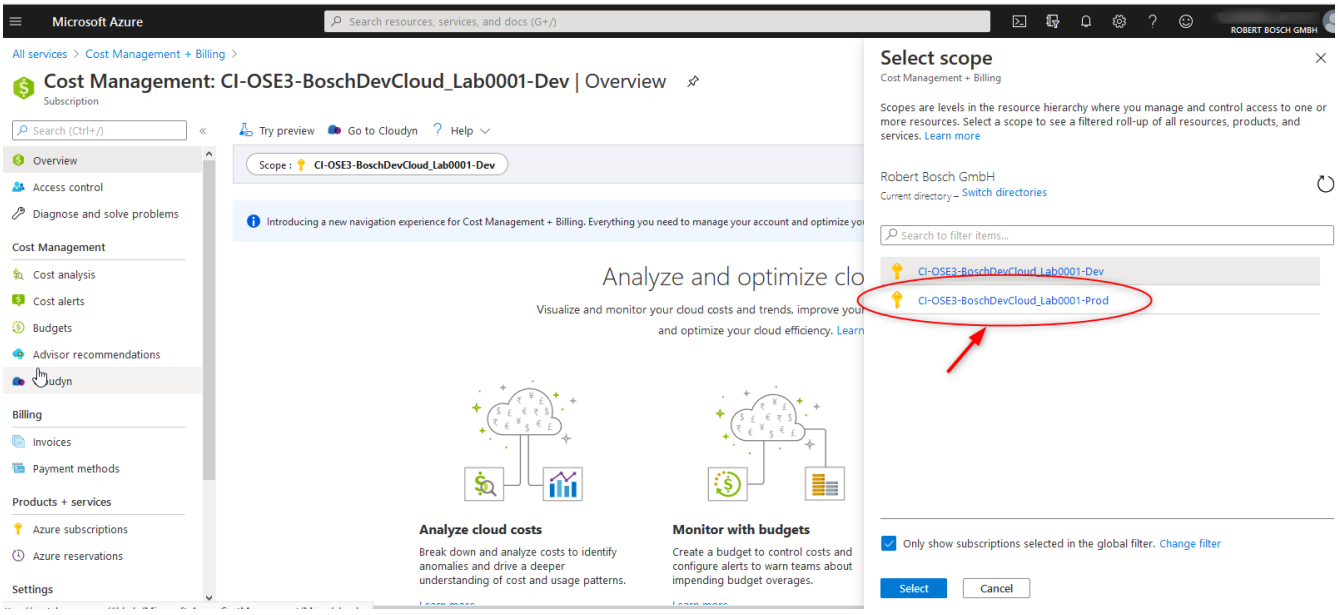
First you open the cost-management item from "All Services" in the azure-portal:



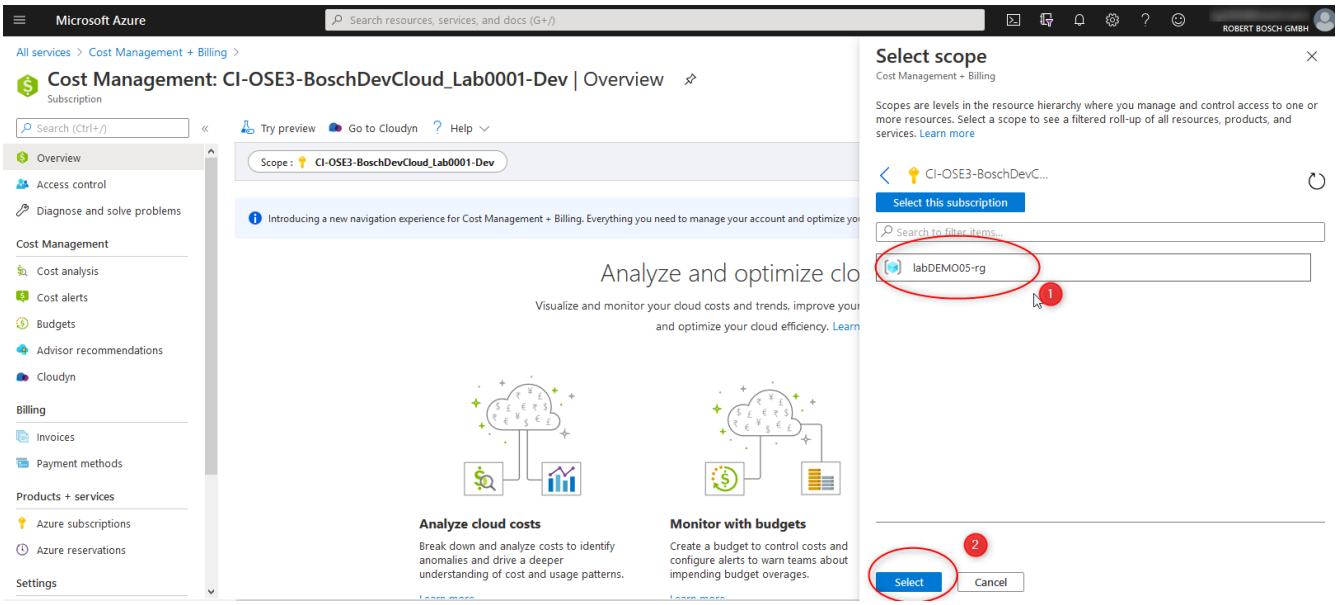
Now you select the "Cost Management" on the left-hand bar



Important is now to select the correct subscription: "CI-OSE3-BoschDevCloud_Lab0001-Prod"



and the respective resource-group

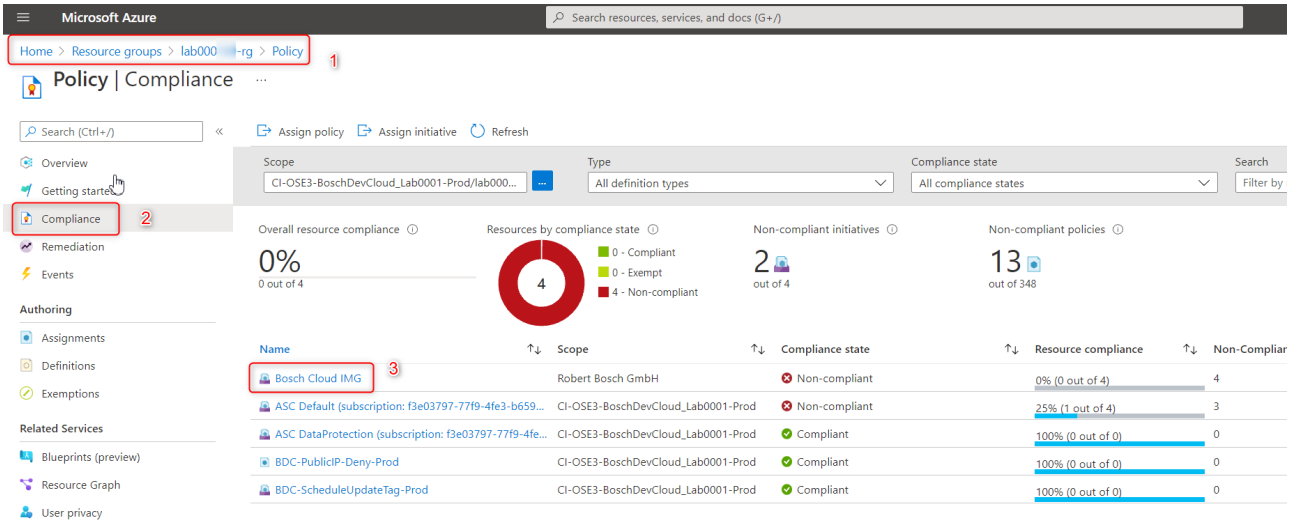


When you have selected all these item you should see now the costs and it is up to you to select which view you like.

12.9. Check your EISA compliance

As an operator of a lab, it is your responsibility to ensure what you deploy and configures meets the compliance requirements of Bosch. Please note, as operator of a lab you will have limited access to shared resources and no access to other labs.

- You can find them when you search for your Resource Group(s) on the Azure portal and select the "Policies" tab. You will need to select "Bosch Cloud IMG" to look for the Bosch owned policies.

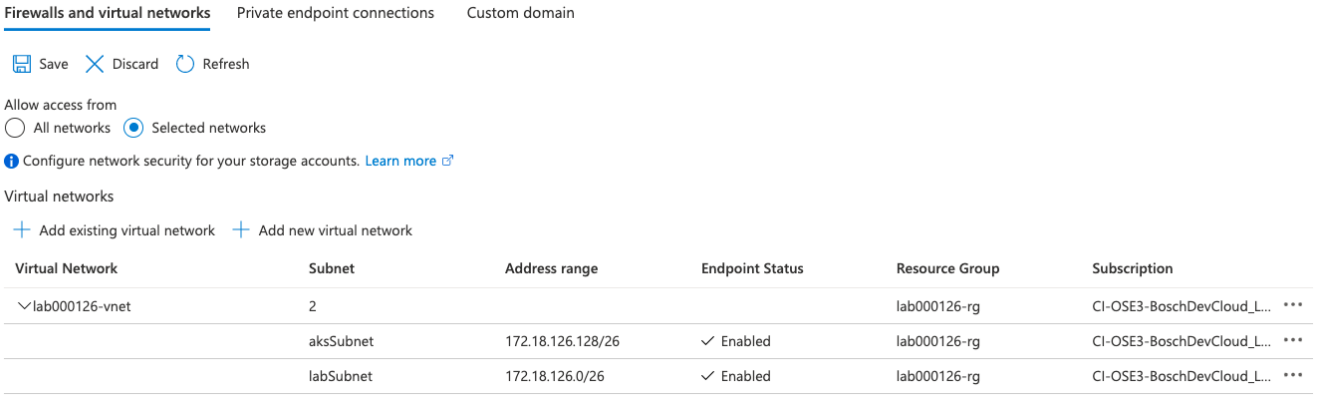


- you can take a look at Azure Implementation Guide → Security Configuration. This will give more details about each policy and its security configuration to be compliant.

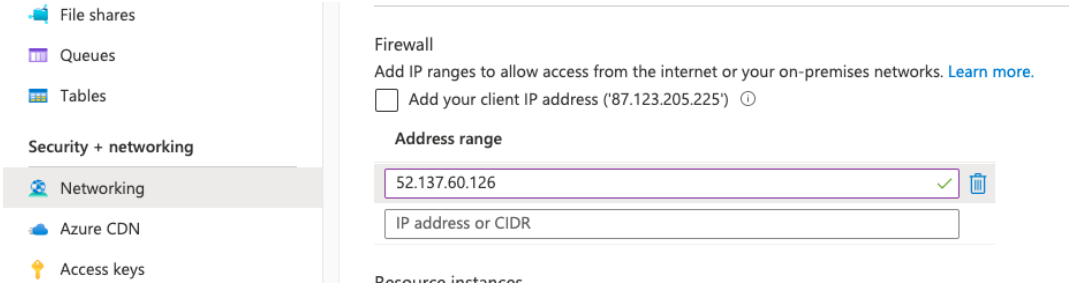
12.9.1. Protecting Azure SaaS-Services

If you create Azure SaaS services like Storace Accounts, Databases etc. you have to ensure their EISA compliance.
This includes that you configure firewall rules to protect the service from authorized access.

The best way to achieve this, is to only allow access from your lab vnet:



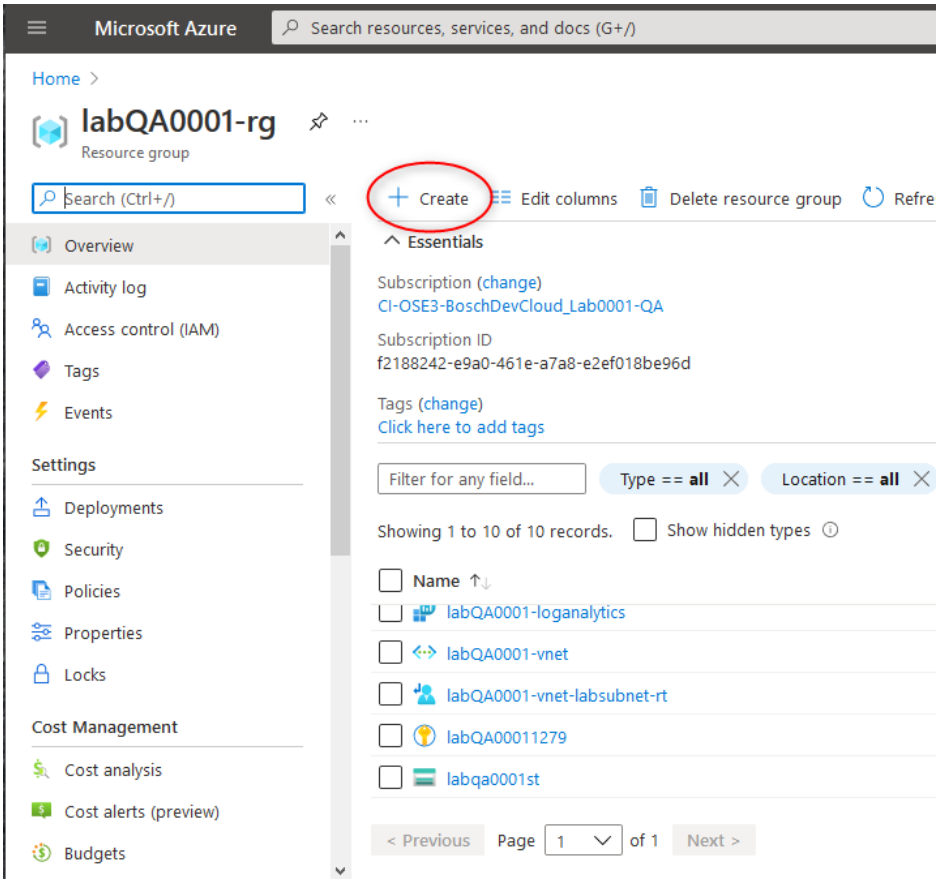
Or if not possible, only allow specific public ip addresses.
To allow access of your lab resources, include the BDC Lab Firewall ip address **52.137.60.126/32**



12.10. Virtual Machines

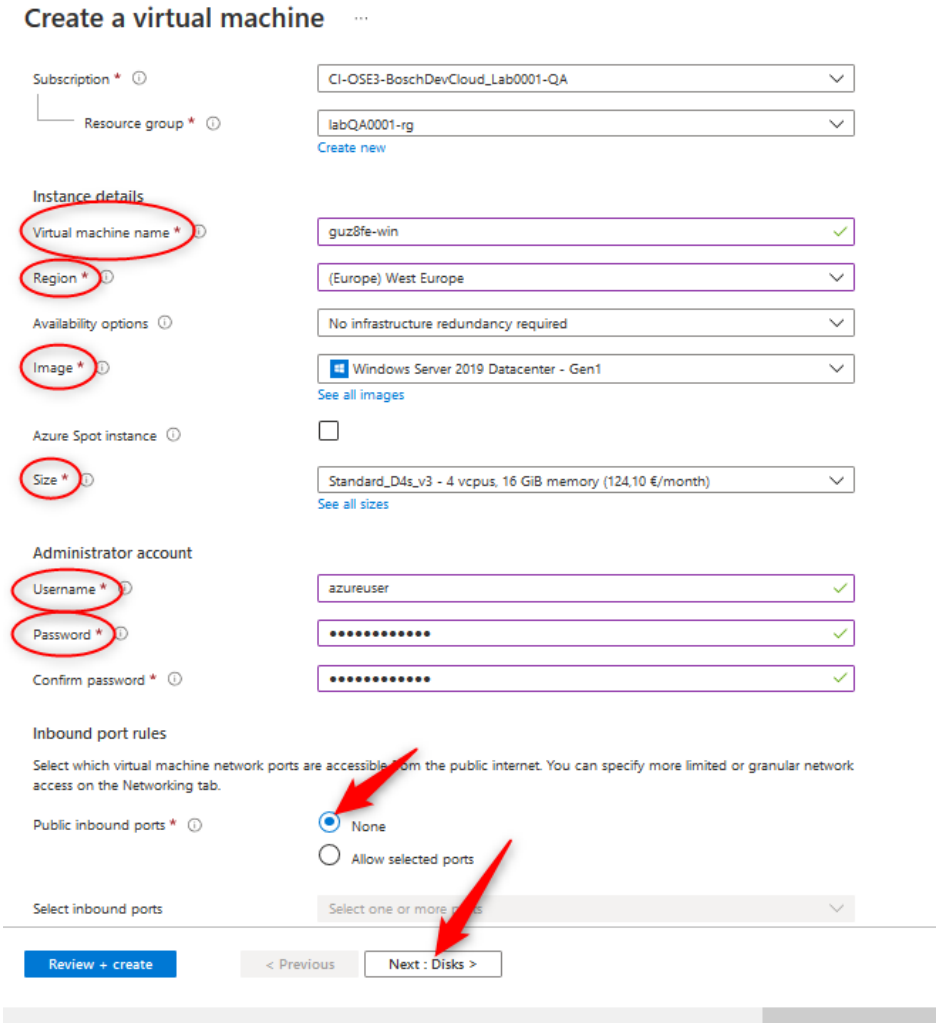
Virtual machines can be created in your lab resource group (eg lab000001-rg).

From the Azure portal, navigate to your lab RG and click on the "+ Create" button on top of the page.



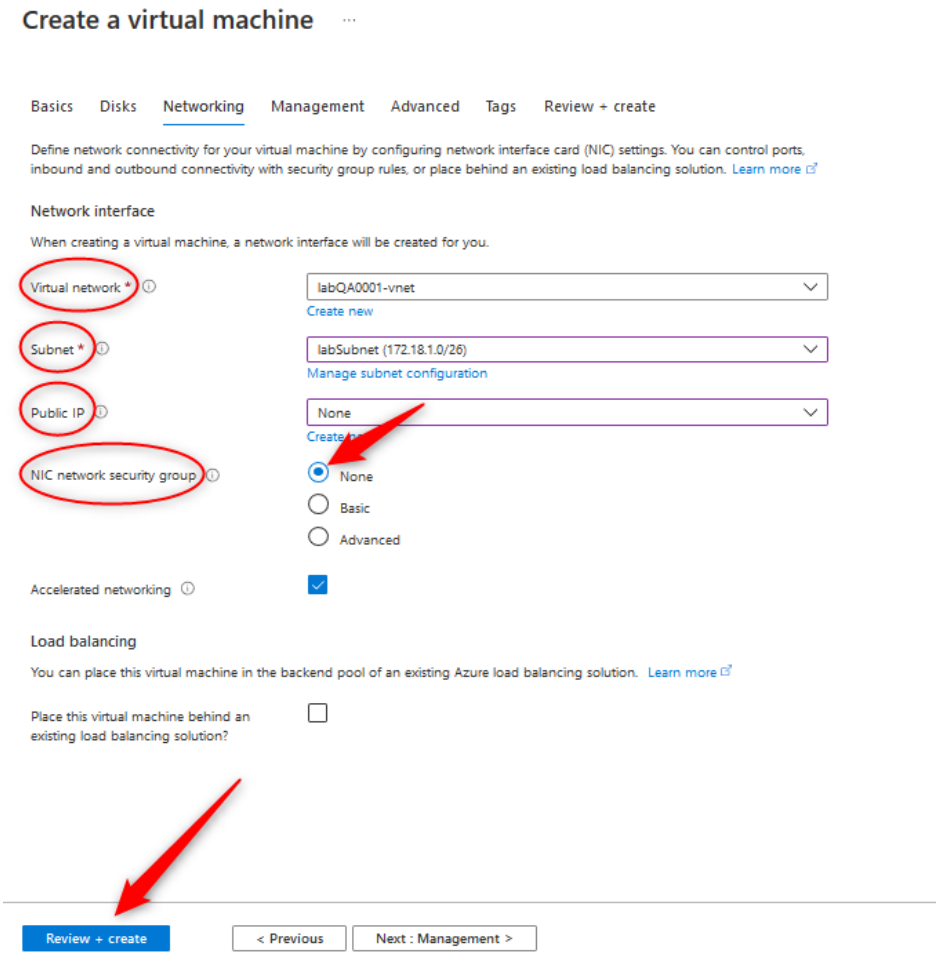
A window pops up, which shows you resources that can be created from the Azure Market Place. In the "Categories" row at the left side of the page select the "Compute" category and then "Virtual Machine".

The "Create Virtual Machine" page opens as shown in the following picture:



Enter values for the following parameters:

- Virtual machine name
- Region (**must** be (Europe) West Europe)
- Image
  Users are free to choose from any available VM size that Azure provides. There is a huge variety of different VM types and sizes. VMs can only be deployed in Microsoft datacenter "West Europe" due to the current network configuration.
  As we cannot keep track on all the changes Microsoft executes, please checkout yourself: https://azure.microsoft.com/en-us/pricing/details/virtual-machines/series/
- Size
  The size of a VM can be changed at any time (just requires a reboot). You just have to go to the Lab in case it was created via the old DevtestLab GUI or otherwise, go to the Azure Portal. Select the VM and re-size it.
- Choose "Password" as authentication type
- Select "None" for Public Inbound ports
- navigate to next page "Disk"
- navigate to next page "Networking"



- for the subnet, choose **labSubnet**
- select "None" for Public IP and the NIC network security group
- click on "Review and Create"
- review the values and finally click on "Create"

To connect to your new VM, navigate to this VM from the portal and connect using Bastion host with the credentials specified during VM creation.

You may also want to install some useful SW packages on your VM. Here is a small HowTo install some applications on a Windows Server VM. To install an application open Powershell, execute the commands below and then restart the Powershell.

- Chocolatey
  ```
  Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
  ```
- Enable the choco feature allowGlobalConfirmation
  ```
  choco feature enable -n allowGlobalConfirmation
  ```
- Azure CLI
  ```
  Invoke-WebRequest -Uri https://aka.ms/installazurecliwindows -OutFile .\AzureCLI.msi;
  ```

```
Start-Process msiexec.exe -Wait -ArgumentList '/I AzureCLI.msi /quiet'; rm .\AzureCLI.msi
```

- VSCode
  ```
  choco install visualstudiocode
  ```
- Firefox
  ```
  choco install firefox
  ```
- kubectl
  ```
  choco install kubernetes-cli
  ```
- notepad
  ```
  choco install notepadplusplus
  ```
- Git
  ```
  choco install git
  ```

A helpful price calculation tool for your VM might be found here: https://azureprice.net/?currency=EUR&region=westeurope&timeoption=month

12.11. VM Update Service

Every virtual machine deployed under the Azure Update Management solution is deployed with a default tag which ensures engagement. The default tag is persistent and it can only be cycled to an alternative option which will either set the frequency of the updates or join the machine into a group where it will exclude all incoming updates.

- The value cycle consists of a predefined values in form of Tags e.g. (scheduledUpdates.Value). These tags are non-changeable to keep the consistency of the solution. Below is the presented table of the available options.
- The update management solution is present in both Linux & Windows environments.

Tag Name: (scheduled.Updates) / Tag Value: **Monthly**

This option presents the default update pattern that is occurring once a month (every second Wednesday).

The option has the following update criteria; (Critical, Security, UpdateRollup, FeaturePack, ServicePack, Definition, Tools, Updates)

Reboot: If required, a reboot of the VM is done automatically.

Tag Name: (scheduled.Updates) / Tag value: **Weekly**

This option presents the most frequent pattern for enrolling updates on a weekly occurrence (every Wednesday).

The option has the following update criteria; (Critical, Security, UpdateRollup, FeaturePack, ServicePack, Definition, Tools, Updates)

Reboot: If required, a reboot of the VM is done automatically.

Tag Name (scheduled.Updates) / Tag Value: **Exclude**

This option presents the **exclusion of regular updates**, where the virtual machine is receiving only the necessary updates like Critical and Security ones.
The running occurrence is once a month (every second Wednesday).

The option has the following update criteria; (Critical, Security)

Reboot: No automatic reboot will be triggered - this has to be done manually by customers and is important for all security updates to take effect.

This tag is not visible in the Azure DevTestLab view of your virtual machine, but only through opening your virtual machine settings by entering the Azure Service "Virtual machines" - This is a known limitation of the Azure DevTestLab service.

Filetransfer into VMs

If you would like to copy data from your computer into your Azure VM, you can achieve this by using a Storage Account.

To do this quickly, we offer you one storage account in your Lab resource group by default.

12.12. Jenkins Helm template

For a detailed usage documentation please refer to our bdc_share

12.13. Azure Bastion

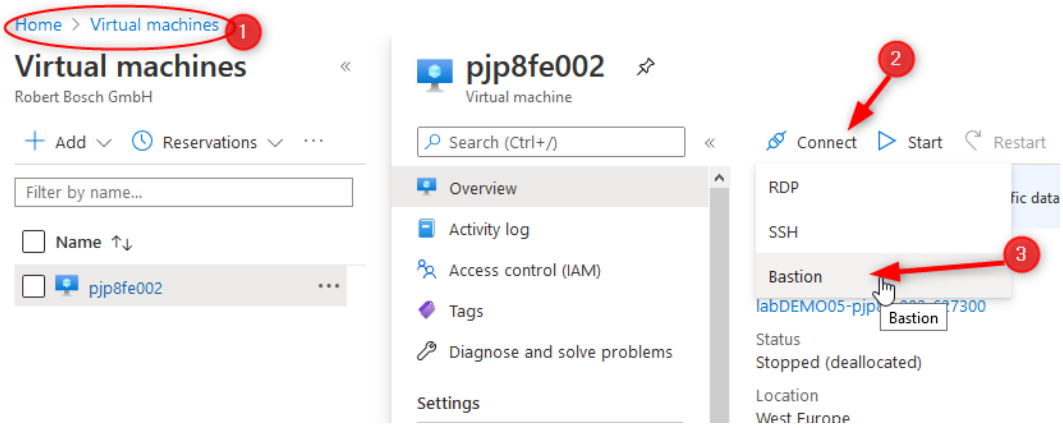
To connect to Virtual Machines created in your lab resource group we offer the service "Azure Bastion". It is **not** possible to connect from your computer to a VM in Azure via *ssh* or *rdp* directly.

The Bastion Service can be ordered at lab creation. While ordering a lab, set include_bastion flag to true (recommended).
In case the lab already exists, you have to explicitly request bastion via jira service desk (BDC Portal → Support).

Follow these steps to connect to your VM:

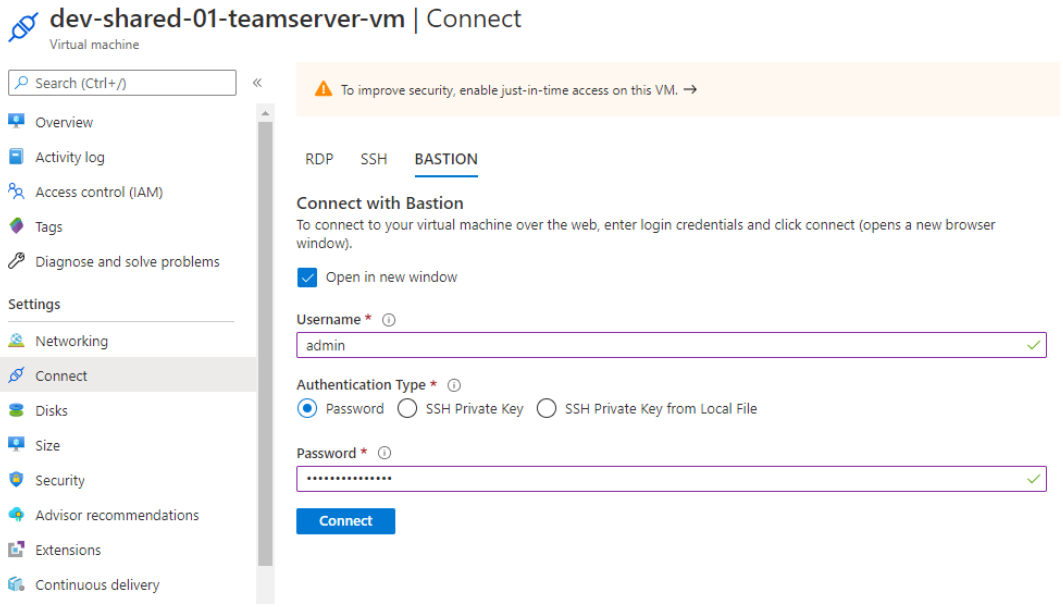
- Open Azure Portal
- Find your specific VM (the one with the dark blue computer icon - when you select the bright blue cube icon you won't be able to click on select)
- **Notice: It is important, that you find your VM via the Azure menu "Virtual machines", not via "Resource groups" or via your DevTestLab context. In this case, the Connect button is not usable and shown in gray**

- open the VM overview and click **Connect**.
- In the menu which is displayed now, click on **Bastion**.



The menu entries "ssh" and "rdp" do not work as the Lab VMs does not have public IP-Adresses for security reasons.

- a new page is shown, again click **Bastion**.
- Fill in your credentials and click **Connect**.



After the login was successfull, you should now see the command prompt of your Linux VM or the Desktop environment of your Windows VM.

Connection to Linux VMs For connection to Linux VMs, Azure Bastion supports SSH via password and ssh_keys.

- Changing the ssh port number is not possible.
- RDP for Linux machines is not available.

Connection to Windows VMs For connections to Windows VMs, Azure Bastion supports RDP.

- SSH or WinRM is not available.

Transfer Data to your VM

Datatransfer works via a "storage account" - see generic documentation on Microsoft website or chapter "Azure Storage Account" here in this document.

There is already a storage account deployed to your lab called "labxxxxxxst"

12.14. Azure Container Registry - ACR (optional)

The Azure Container Registry is a SaaS Azure resource, which offers an internet facing Docker registry with RBAC control.

When we create an ACR, we also create three different SPNs which have the following permissions on the ACR:

- sp_devcloud_lab_labDEV001acrowner (Owner Permission)
- sp_devcloud_lab_labDEV001acracrpull (ACRPull Permission)
- sp_devcloud_lab_labDEV001acracrpush (ACRPush Permission)

See https://docs.microsoft.com/en-us/azure/container-registry/container-registry-roles

The ID and Password of each SPN is stored in the default lab Key vault of the Lab Resource group. Refer this document on how to avail access.

For compliance and security reasons, these secrets are rotated once a year. We usually append new ones about 20 days before they expire and delete old ones after they expire. During this period, both old and new secrets work. You must check your key vault for new secrets and update your systems.

12.15. Lab Internet Access (optional)

We offer a reverse-proxy solution using Azure Application Gateways in combiation with AKS Traefik Reverse Proxy to allow HTTPS access from the internet to your web-based resources in your lab. Have a look at the AKS reverse proxy documentation on how to configure your endpoints.

Be aware that this opens your web-based services to the public internet. You need to ensure they are secured and enforce the use of Bosch accounts for authentication.

- The default URL that we offer is `aks-<lab_name>.boschdevcloud.com`. We offer customization of URL's, format: `<subdomainPrefix>-<alias>.boschdevcloud.com`

- `subdomainPrefix`: app1, app2, app3, as you require.
- `alias`: Team Name (by default lab name is used)

- Customized URL looks something like, `app1-teamname.boschdevcloud.com`
- As this service costs money, you need to be careful in reducing the number of URLs by

configuring traefik to work with `/path` like `demo-teamname.boschdevcloud.com/app1`

- To enable this service, Kindly raise a support ticket in Jira Service Desk.

As a prerequisite, you must have AKS deployed in your lab to configure traefik. Refer to next section for more details.

12.16. Azure Kubernetes Service - AKS (optional)

Azure Kubernetes Service (AKS) is a fully managed Kubernetes Cluster by Azure which can be used to deploy and manage containerized applications and services quite easily.

- AKS will only be deployed by the BDC team, feel free to create a support ticket in Jira Service Desk.

Please do not try to create an AKS cluster by yourself.

- Only one AKS per lab is allowed due to technical limitation, if you need second AKS, another lab is required.
- This AKS is also used for enabling internet-access for your services in lab.

As an admin of a lab which hosts an AKS you're responsible **to check regularly** for the latest version of the AKS software. Please see following chapter.

12.16.1. Update your AKS

As already described the admin of a "lab" has the task to keep the AKS up-to-date regularly. If you do not do so after a while you won't be able to update anymore. The only solution is to remove the AKS and redeploy it via pipeline. This can only be done by the BDC-team.

Current Recommended AKS version by BDC: 1.26.10

Important hints before you start the upgrade:

- First save all your existing crucial codes (we assume you did "everything as code")
- It will take some minutes for the upgrade, dependent on the cluster size. Your nodes will be updated one after the other and your applications will be moved to new nodes during the upgrade.
- Downgrades are not possible

You may also check:

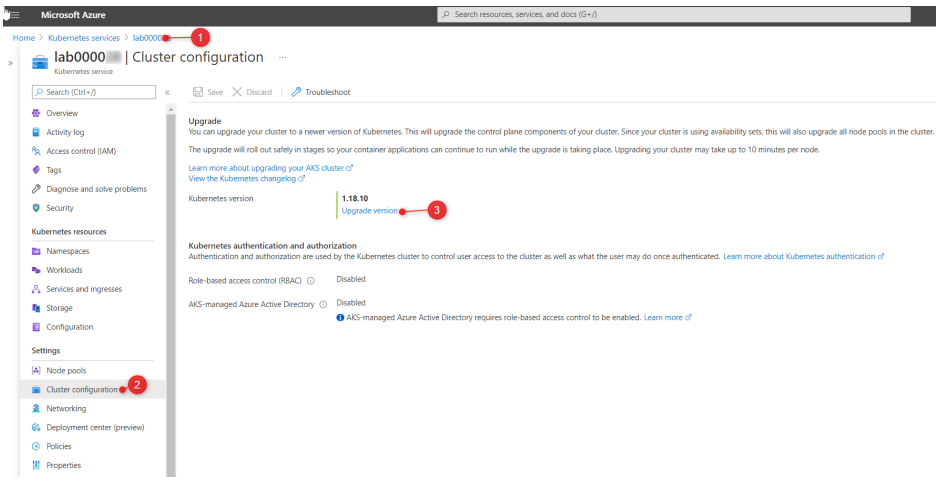
https://docs.microsoft.com/en-us/azure/aks/supported-kubernetes-versions

In a shell you can check for the versions available (BDC is hosted in Azure-region "Europe" only)

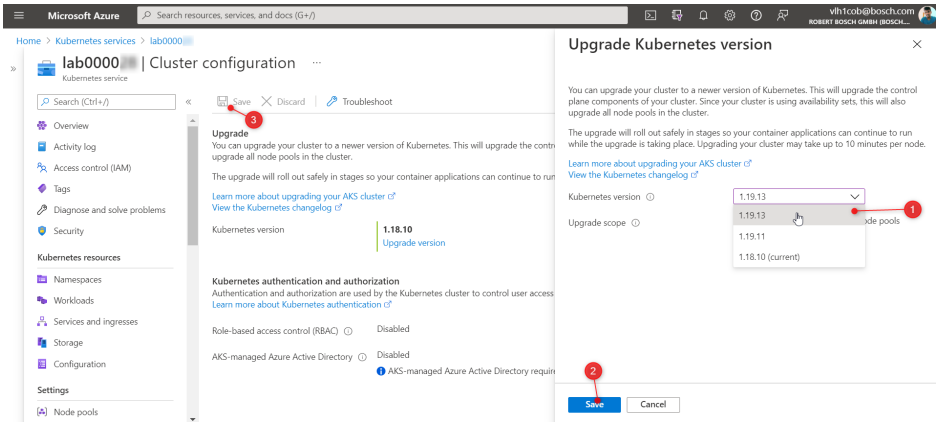
```
1 az aks get-versions -l westeurope -o table
```

Steps to upgrade

Check for "all services" in the azure portal and select "Kubernetes" (1), then click "Cluster configuration" (2) and then click on Upgrade version (3) as shown in the example below:



You cannot skip major/minor version (e.g. upgrade from 1.14 to 1.18 directly) and in such cases you have to start the upgrade twice. So first use the drop-down menu and chose the highest version possible (1) and then press "Save" (2)



Some more information from Microsoft directly: https://docs.microsoft.com/de-de/azure/aks/tutorial-kubernetes-upgrade-cluster

12.16.2. AKS reverse proxy - Traefik

Every Lab AKS is deployed with a preconfigured reverse proxy named Traefik.
This reverse proxy makes Kubernetes services/pods available from outside an AKS and is doing TLS termination.

A lab admin can adjust the AKS Traefik/ingress configuration and define additional internal lab services to make them public available.

For these services usually other containers are used within the AKS, but also services hosted on VMs located in the lab might be used.
The templates below can be adapted and used for this purpose.
Check also the corresponding kubernetes ingress documentation

Listing 12. add a ingress configuration to your aks [powershell]

```
az aks get-credentials  --resource-group <RESOURCEGROUP_NAME> `
                        --name <YOUR_LABNAME> `
                        --subscription <SUBSCRIPTION_ID>
kubectl apply -f <PATH_TO_YAML_FILE>
```

Listing 13. example Treafik Kubernetes Ingress configuration **for a Kubernetes pod as endpoint** *[yaml]*

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  labels:
    app: test-app
    env: prod
  name: ingressname1             # ingress name and should be unique in namespace
  namespace: default             # preferabally app namespace
  annotations:
    kubernetes.io/ingress.class: traefik
spec:
  rules:
  - host: <HOSTNAME>.boschdevcloud.com    #URL offered by BDC team, looks simillar to aks-<lab_name>.boschdevcloud.com
    http:
      paths:
      - backend:
          service:
            name: service1       # Name of your kubernetes service mentioned below
            port:
              number: 80         # Port where your below service is listening on
        path: /apple             # Path where ingress listens e.g hostname.boschdevcloud/jenkins1/, if you want users to access with just hostname without path - just use `/`
        pathType: Prefix
---
apiVersion: v1
kind: Service
metadata:
  labels:
    app: test-app
    env: prod
    role: frontend
  name: service1
  namespace: default             #preferabally app namespace
spec:
  ports:
  - port: 80                     # Port where this service is listening on
    protocol: TCP
    targetPort: 5678             # Port where your kubernetes backend service is listening on
```

```
  selector:
    app: test-app               #lables used in the pod
    env: prod
```

Listing 14. If you like to test this config, below is an example to create a pod.

```
kind: Pod
apiVersion: v1
metadata:
  name: test-app
  labels:
    app: test-app
    env: prod
spec:
  containers:
    - name: test-app
      image: hashicorp/http-echo # this image runs on default port 5678.
      args:
        - -text=apple
```

Once pod is up and running, you can visit your URL to check if it works. Output should display `apple` in URL.

Listing 15. example Traefik Kubernetes Ingress configuration **for a virtual machine as endpoint** *[yaml]*

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  labels:
    app: myservice
  name: myservice
  namespace: default
  annotations:
    kubernetes.io/ingress.class: traefik
spec:
  rules:
  - host: <HOSTNAME>.boschdevcloud.com #URL offered by BDC team, looks simillar to aks-<lab_name>.boschdevcloud.com
    http:
      paths:
      - backend:
          service:
            name: myservice      # Name of your below service
            port:
              number: 80         # Port where your below service is listening on
        path: /                  # Path where ingress listens e.g hostname.boschdevcloud/path/
        pathType: Prefix
---
apiVersion: v1
```

```
kind: Service
metadata:
  labels:
    name: myservice
    role: frontend
  name: myservice
  namespace: default
spec:
  ports:
  - port: 80                    # Port where this service is listening on
    protocol: TCP
    targetPort: 80              # Port where your service on the VM is listening
on
  selector:
    name: myservice
  type: ExternalName
  externalName: <IP_ADDRESS>    # Internal IP address of your virtual machine
```

Restrict Internet Access to specific ip ranges

If you want to restrict the access of your public available service, you can configure Traefik to only allow specific source IPs.

This is done via Traefik middlewares.

Whitelisting Bosch ip addresses for example, might be done via following example:

Listing 16. traefik whitelist middleware - For the current list of Bosch used IPs please visit: Docupedia

```
apiVersion: traefik.containo.us/v1alpha1
kind: Middleware
metadata:
  annotations:
  generation: 1
  name: ip-whitelist
  namespace: default
spec:
  ipWhiteList:
    ipStrategy:
      depth: 1
    sourceRange:
    - 194.39.218.10
    - 194.39.218.16
    - 194.39.218.17
    - 194.39.218.18
    - 194.39.218.19
    - 194.39.218.20
    - 194.39.218.21
    - 194.39.218.22
    - 194.39.218.23
    - 103.4.125.25
```

```
    - 119.40.64.9
    - 119.40.64.26
    - 209.221.240.193
    - 177.11.252.15
    - 103.205.152.154
    - 194.39.218.14
    - 194.39.218.13
    - 139.15.3.133
    - 139.15.143.3
    - 209.221.240.152
    - 209.221.240.153
    - 195.11.167.73
    - 103.4.127.176
    - 119.40.72.61
    - 194.39.218.11
    - 194.39.218.12
    - 194.39.218.15
    - 103.4.125.23
    - 119.40.64.12
    - 209.221.240.196
    - 177.11.252.18
```

To use the middleware, it has to be referenced in an IngressRoute rule:

Listing 17. traefik ingressroute whitelist

```
apiVersion: traefik.containo.us/v1alpha1
kind: IngressRoute
metadata:
  annotations:
  generation: 5
  name: ip-whitelists-ingressroute
  namespace: default
spec:
  entryPoints:
  - websecure
  routes:
  - kind: Rule
    match: Host(`myservice1.boschdevcloud.com`) && PathPrefix(`/mypath`)    # Only match at a specific path on a specific hostname
    middlewares:
    - name: ip-whitelist
    services:
    - name: myservice1
      port: 80
  - kind: Rule
    match: Host(`myservice2.boschdevcloud.com`)                             # Only match for any path on a specific hostname
    middlewares:
    - name: ip-whitelist
    services:
```

```
24     - name: myservice2
25       port: 80
```

Please notice:
Middleware can only be used in conjunction with a Traefik ingressRoute, which is a Kubernetes Custom Resource Definition(CRD) object and differs to standard Kubernetes Ingress rules.

If you have an ingress route in place and want to use whitelisting, please delete the ingress rule and create a Traefik IngressRoute object instead.

Update traefik version

The Traefik service is deployed and upgraded by the BDC team. There is no need to upgrade the service yourself.
Typically an update to the newest available version (including new self signed certificates) is done every 4 weeks during our maintenance.

If you have made changes on the Traefik deployment (e.g. changed the log level) this changes will be reverted by our upgrade!

Do not create any resources (example ingress) in namespace bdc-traefik (managed by BDC team) , the resources under this namespace might under go changes during our regular release. Please create your own namespace for easy management of your resources.

12.16.3. Troubleshoot AKS traefik/ingress configuration

If your configured service is not properly available through the internet, you may consider following troubleshooting steps:

First of all we should check, if we can reach the hostname of the configured lab access e.g. aks-lab000XXX at all.
This might be done using the path "ping", to see if we get an response from the traefik reverse proxy on the lab AKS.
https://aks-lab000XXX.boschdevcloud.com/ping
Which should result in a simple "ok"-message.

Then we can test the traefik ingress configuration directly from within the AKS cluster, without accessing the service via BDC Application Gateway:

Prerequisite
Please ensure you have following tools installed:
- Azure cli
- kubectl

```
Replace all "XXX" placeholders with your lab number as well every url or used service
name with your information.
```

```
# login with azure cli and get aks credentials of your cluster for kubectl
az login
az aks get-credentials -g lab000XXX-rg -n lab000XXX --overwrite-existing

# create a new pod from ubuntu image and execute bash in interactive mode
kubectl run mytestpod --rm -it --image ubuntu -- bash

# install vi and curl
apt update; apt install -y vim curl

# open the containers hosts file
vi /etc/hosts

# add your lab access hostname and point it to the traefik ip.
# The third octet in the ip address is the lab number.
# The fourth octet is the ip of the azure "internal-loadbalancer" resource,
# which might be found in the AKS resource group "MC_lab000XXX-rg_lab000XXX_westeurope"
172.18.XXX.133  aks-lab000XXX.boschdevcloud.com

# test the connection
curl --insecure https://aks-lab000XXX.boschdevcloud.com/myservice
```

If this results in the same error as via web browser using the internet, a problem with the application gateway can be excluded. If not, please open a ticket and the BDC Team will have a look at it.

Next, check the ingress rule and the corresponding backend service:

```
# get the ingress rule in detail and double check the configured hostname
# and the service name of the backend.
kubectl describe ingress myservice

Rules:
  Host                          Path  Backends
  ----                          ----  --------
  aks-lab000XXX.boschdevcloud.com
                                /   myservice:80 (<none>)

# get the details of the configured service
# the name must be the same as the backend in the above ingress listing
kubectl describe svc myservice

Name:              myservice
Namespace:         default
Labels:            name=myservice
                   role=frontend
Annotations:       <none>
Selector:          name=myservice
```

```
Type:              ExternalName
IP Families:       <none>
IP:
IPs:               <none>
External Name:     172.18.XXX.36
Port:              http  80/TCP
TargetPort:        80/TCP
Endpoints:         <none>
Session Affinity:  None
Events:            <none>
```

In the above service output the value of "External Name" is pointing to an ip address of a virtual machine.
To test if this vm is serving the application properly, we jump back to the self-created test pod and change the hosts file to this ip address.

```
kubectl exec -it mytestpod -- /bin/bash

vi /etc/hosts
# this time we take the ip address of the vm instead of the traefik reverse proxy
172.18.XXX.36  aks-lab000XXX.boschdevcloud.com

# test it again - this time without https, but only http as the application is
  running on port 80
curl http://aks-lab000XXX.boschdevcloud.com/myservice
```

When this also results in an error, there has to be a problem with the backend webserver.

But if the test above was successful we can pin the problem to the ingress/traefik configuration.
If so, we can check the logs of the traefik reverse proxy.

Listing 18. Get logs for traefik

```
kubectl logs --selector "app.kubernetes.io/name=traefik" -n bdc-traefik

{"level":"error","msg":"Near line 944 (last key parsed 'frontends.aks-
  lab000XXX.boschdevcloud.com/myservice.whiteList.sourceRange'): strings cannot
  contain newlines","time":"2021-11-12T15:13:54Z"}
{"level":"error","msg":"Near line 948 (last key parsed 'frontends.aks-
  lab000XXX.boschdevcloud.com/myservice.whiteList.sourceRange'): strings cannot
  contain newlines","time":"2021-11-12T15:13:54Z"}
```

In the above listing, we can see that we sadly put newlines in the ingress configuration, which prevented traefik from serving properly.
To fix this, we edit the corresponding ingress rule and restart the traefik service by deleting the current pod (which will be created again automatically)

Listing 19. Correct ingress definition

```
# edit the misconfigured ingress rule
kubectl edit ingress myservice

# delete the traefik pod
kubectl -n kube-system delete pod traefik-7bbf85b6c8-77976
```

Now we can check the logs of the restarting traefik again.

Listing 20. Get logs for traefik

```
kubectl logs --selector "app.kubernetes.io/name=traefik" -n bdc-traefik # use '-f' to follow the log file

{"level":"info","msg":"Server configuration reloaded on :80","time":"2021-11-08T15:14:44Z"}
{"level":"info","msg":"Server configuration reloaded on :443","time":"2021-11-08T15:14:44Z"}
{"level":"info","msg":"Server configuration reloaded on :8080","time":"2021-11-08T15:14:44Z"}
```

If you don't get the necessary information in the traefik log to find the cause of an issue, you can adjust the log level of traefik via editing the traefik deployment:

Listing 21. Increase traefik log level

```
kubectl -n bdc-traefik edit deployment traefik
      containers:
      - args:
        - --log.level=WARNING   # possible values are: DEBUG, PANIC, FATAL, ERROR, WARN, and INFO
```

You can also get the logs using log analytics workspace in the azure portal. You need to open `lab000XXX-loganalytics` resource (from your resource group) and navigate to `Logs`. Here you can query the logs using the following kusto query:

Listing 22. Kusto query to get traefik logs

```
ContainerLog
| where TimeGenerated < now()
| where TimeGenerated >= startofday(ago(1d))
| extend ClusterName = tostring(split(_ResourceId, "/")[-1])
| join kind = inner (
    KubePodInventory
    | project ContainerID, PodName=Name, ControllerKind, ControllerName, Namespace
    | distinct *
    )
    on ContainerID
```

```
| where PodName startswith "traefik-"
| extend Log=parse_json(LogEntry)
| project TimeGenerated, Log.level, Log.msg, Log.entryPointName, Log["time"],
  ClusterName, PodName, LogEntry
| sort by TimeGenerated desc
```

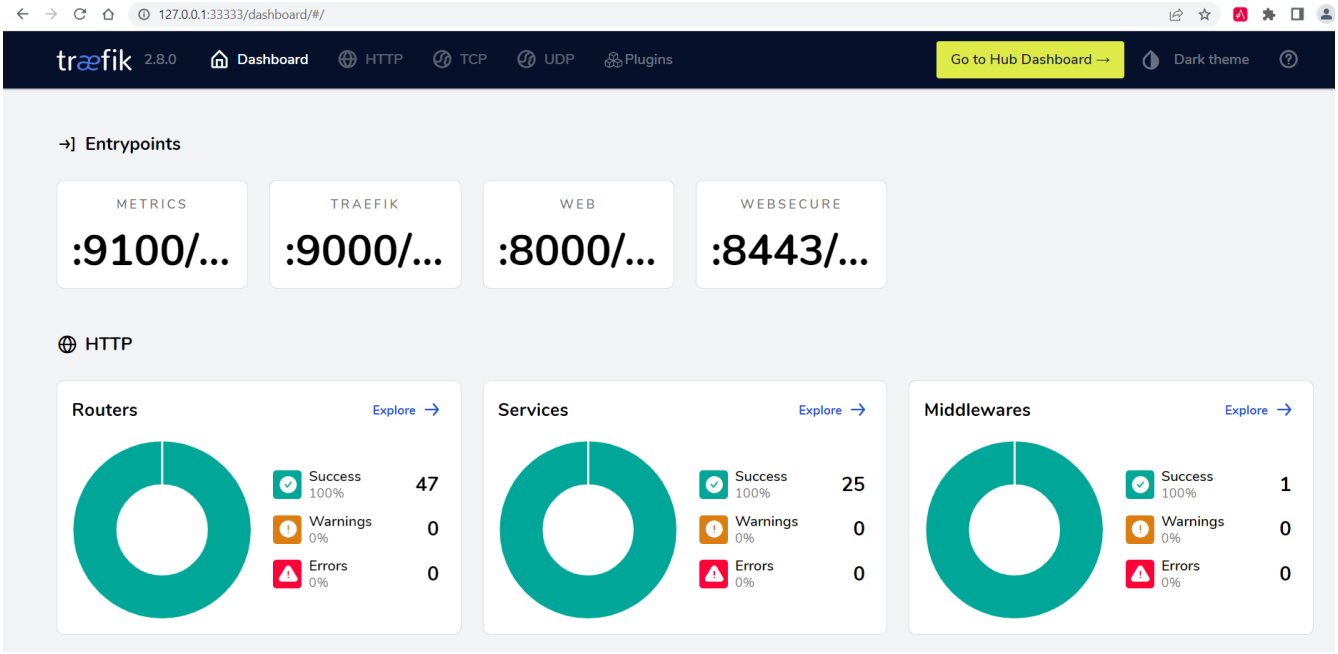
After checking the logs, we can see that we fixed the issue and traefik is working properly.
This can be also seen via the traefik dashboard which we can forwarded to our local computer using (not working in Azure Cloud Shell):

Listing 23. traefik dashboard

```
kubectl port-forward $(kubectl get pods --selector "app.kubernetes.io/name=traefik"
  --output=name -n bdc-traefik) -n bdc-traefik 33333:9000
```

Now you can open following url in the browser:
http://127.0.0.1:33333/dashboard/

Here we should see all configured reverse proxy urls and the corresponding backends.



12.17. Authentication Proxy (optional)

In addition to the lab internet access, the authentication proxy solution enforces a user login with the Bosch account.

12.18. Azure Storage Account

For easy data handling we offer you a fully prepared storage account.

You may find it in your Lab resource group with the name

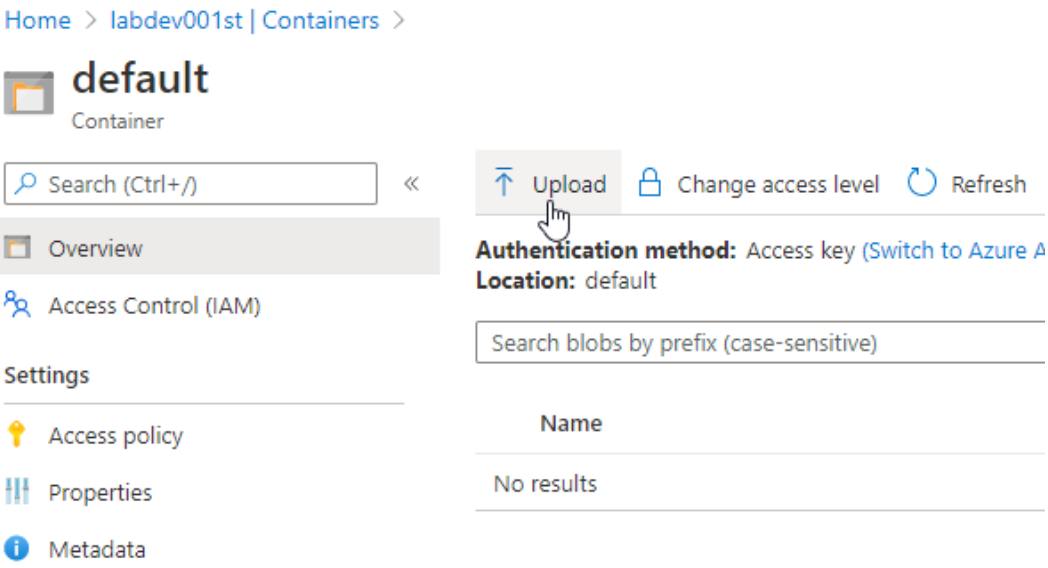
```
lab0000xx-rg\lab0000xx**st**
```

In this storage account we created a "default" container(kind of folder) where you can upload your files into. Feel free add more container if you need.

There are different ways to handle files within this Storage Account.

File Upload

The easiest way is to open the Storage account resource in the Azure Portal, open the container and click on "Upload"



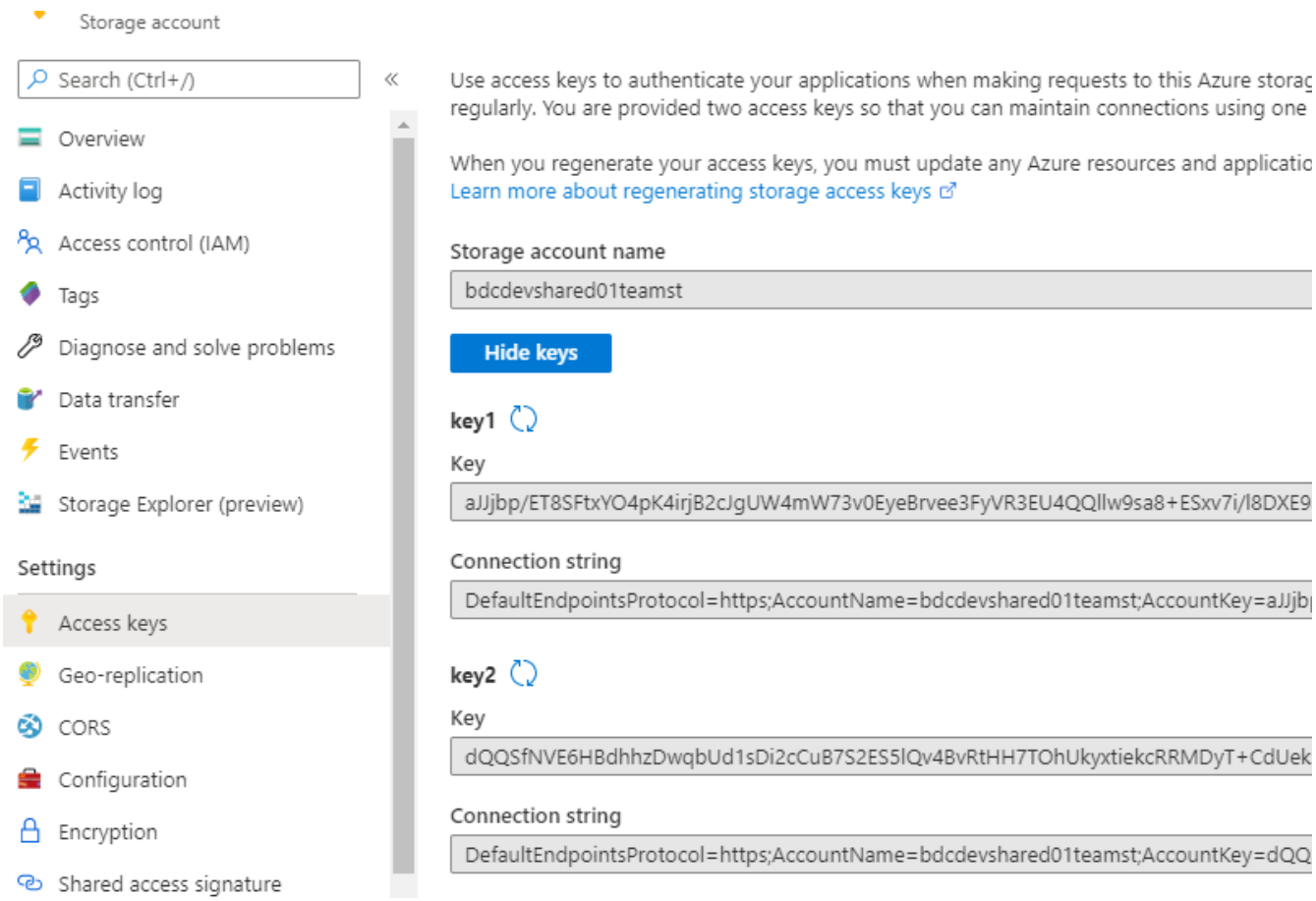
Azure Storage Explorer

To handle files in a more advanced way, there is the specific application "Azure Storage Explorer". More information you may find here:
https://azure.microsoft.com/en-us/features/storage-explorer/

If you are working inside the BCN, you need to have the Bosch proxy and its self-signed certificate configured in the application settings.

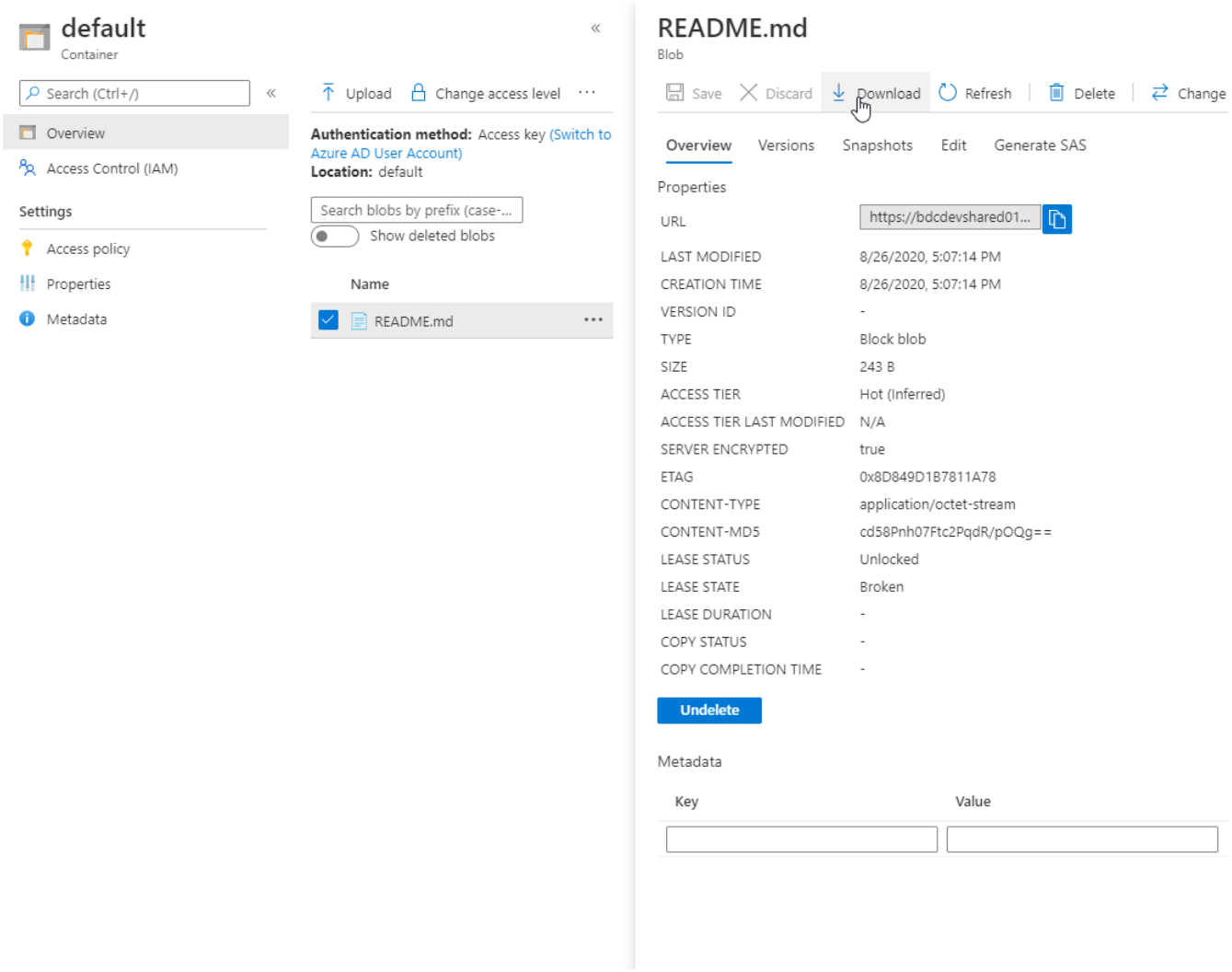
File Download

To work with a storage account in general, you have to authenticate against it using the automatically generated access keys.

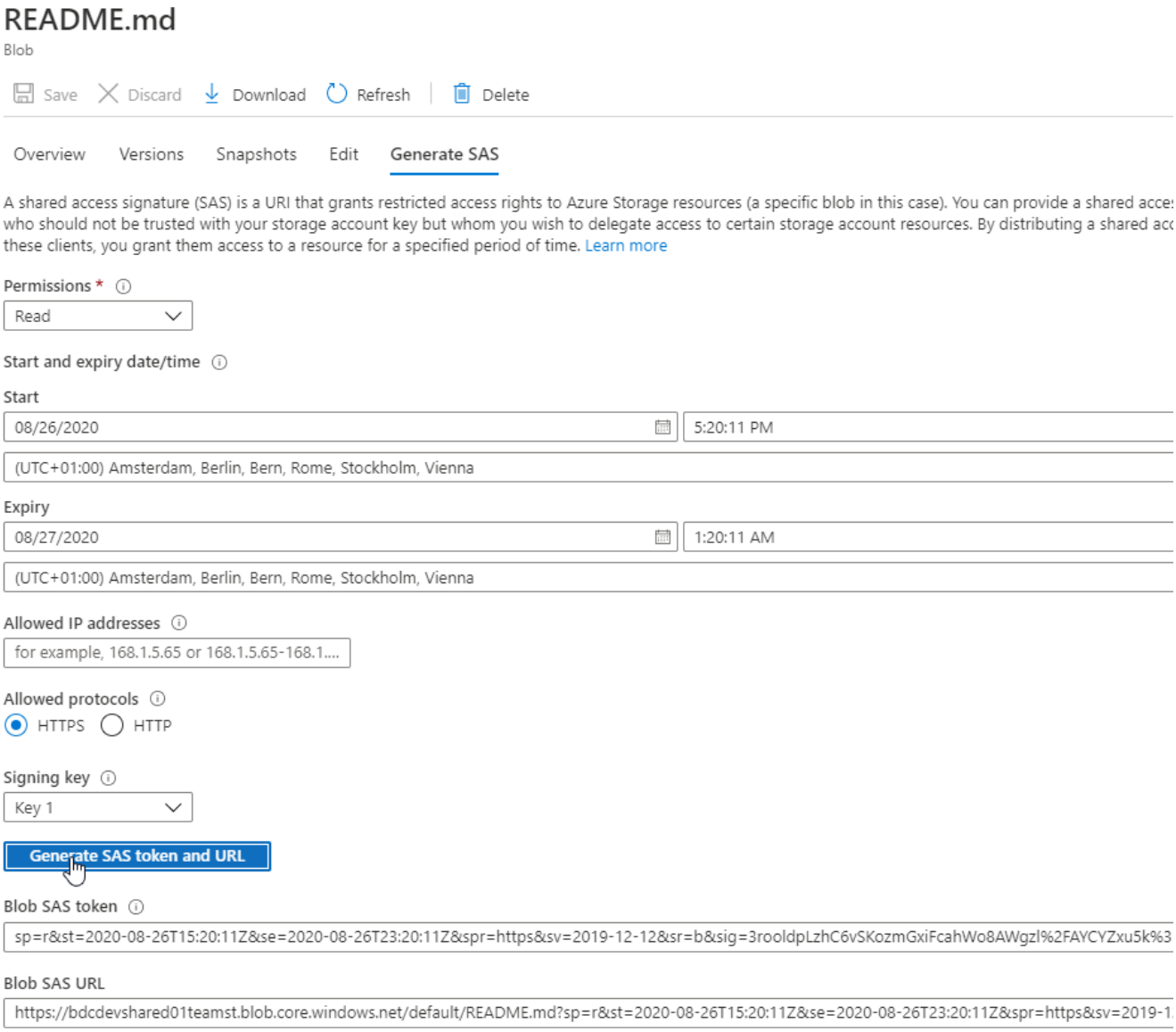


Quick download of files

If you just need to download specific files from your storage account, you can just do so by using the azure portal and click "Download in the property section of a file".



Download link If your system does not provide a graphical user interface, you can also download files via curl or wget.
To get a direct copy&paste link of a file, you can generate a shared access signature (SAS) URL via the Azure Portal.
This creates a link which is only valid for a certain time.



Please be are that you quote (') the "Blob SAS URL" when using it via curl/wget.

```
curl -sO 'https://bdcdevshared01teamst.blob.core.windows.net/default/README.md?sp=r&st=2020-08-26T15:20:11Z&se=2020-08-26T23:20:11Z&spr=https&sv=2019-12-12&sr=b&sig=3rooldpLzhC6vSKozmGxiFcahWo8AWgzl%2FAYCYZxu5k%3D'
```

12.19. Certificates

When an AKS is available within the lab, a container based loadbalancer (traefik) with a custom certificate is deployed too. This LB is used as kubernetes ingress resource, to make container services available via http and https from outside the cluster.

As the used certificate is a self signed one, you need to install the corresponding root and intermediate certificates on your VM, to properly access the exposed AKS service via https.
This avoids error messages like:

```
curl: (60) SSL certificate problem: self signed certificate in certificate chain
```

1. The chained certificate is stored in the lab keyvault as a base64 encoded PFX file.

12.19.1. Get root / intermediate certificates from KeyVault

The following example shows how to download the certificate using azure cli commands in bash or powershell

Listing 24. Download certificate [bash]

```
# define environment
labName="lab000132"
tempDir="./cert"
mkdir $tempDir

az login
az account set --subscription CI-OSE3-BoschDevCloud_Lab0001-Prod`
kvName=`az lab get --name $labName --resource-group "$labName-rg" --query vaultName
-o tsv | cut -d '/' -f9`

# Prerequsist - you need get permission on the lab keyvault
# userId=az ad signed-in-user show --query id -o tsv
# az keyvault set-policy -n $kvName --secret-permissions get list --object-id
$userId

# Download certificate
az keyvault secret download --vault-name $kvName --name aksTraefik-01-
sslPfxChained-base64 --encoding base64 --file "$tempDir/cert.pfx"

# Extract ca certs
openssl pkcs12 -in "$tempDir/cert.pfx" -cacerts -nokeys -chain -out "
$tempDir/certs.pem" -passin pass:
```

Listing 25. Download certificate [powershell]

```
# define environment
$labName="lab000132"
$tempDir="c:/temp/cert"
mkdir $tempDir

az login
az account set --subscription CI-OSE3-BoschDevCloud_Lab0001-Prod
$kvId=az lab get --name $labName --resource-group "$labName-rg" --query vaultName
-o tsv
$kvName=($kvId -split "/")[-1]

# Download certificate
az keyvault secret download --vault-name $kvName --name aksTraefik-01-
sslPfxChained-base64 --encoding base64 --file "$tempDir/cert.pfx"

# Extract ca certs
openssl pkcs12 -in "$tempDir/cert.pfx" -cacerts -nokeys -chain -out "
$tempDir/certs.pem" -passin pass:
```

Now you just need do split the root and intermediate certificate inside of certs.pem. The certificate after "subject ... CN = boschdevcloud-CA-P" is the intermediate and after "subject=C... CN = boschdevcloud-CA" is the root certificate.
Save the two certificates into separate files (root.crt and intermediate.crt)

12.19.2. Install root / intermediate certificates on VM

Now install the certificates on your VM.

Listing 26. Windows

```
# Note:  certificate files must be without  (-----BEGIN CERTIFICATE----- ...
-----END CERTIFICATE-----)

$root=Get-Content "root.crt" -raw
[System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String($root)) |
Out-File -Encoding "ASCII" "r.pem"
Import-Certificate -FilePath "r.pem" -CertStoreLocation Cert:\\LocalMachine\\Root

$intermediate=Get-Content "intermediate.crt" -raw
[System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String($intermed
iate)) | Out-File -Encoding "ASCII" "i.pem"
Import-Certificate -FilePath "i.pem" -CertStoreLocation Cert:\\LocalMachine\\CA
```

Listing 27. Linux (Red Hat)

```
sudo cp root.crt /etc/pki/ca-trust/source/anchors/
sudo cp intermediate.crt /etc/pki/ca-trust/source/anchors/
sudo update-ca-trust extract
```

Listing 28. Linux (Ubuntu)

```
sudo cp root.crt /usr/local/share/ca-certificates/
sudo cp intermediate.crt /usr/local/share/ca-certificates/
sudo update-ca-certificates
```

12.20. Network

12.20.1. Lab vnet

Every Lab has its own virtual network, where Lab resources are connected to.

The vnet is structured in four subnets:

Subnet Name	Subnet address	Last ip	Available addresses	Usage
labSubnet	172.18.1.0/26	172.18.1.63	57	Customer VMs

Subnet Name	Subnet address	Last ip	Available addresses	Usage
authProxySubnet	172.18.1.64/27	172.18.1.95	27	AuthProxy
aksSubnet	172.18.1.128/26	172.18.1.191	57	Kubernetes Cluster
AzureBastionSubnet	172.18.1.224/26	172.18.1.255	57	Bastion Host

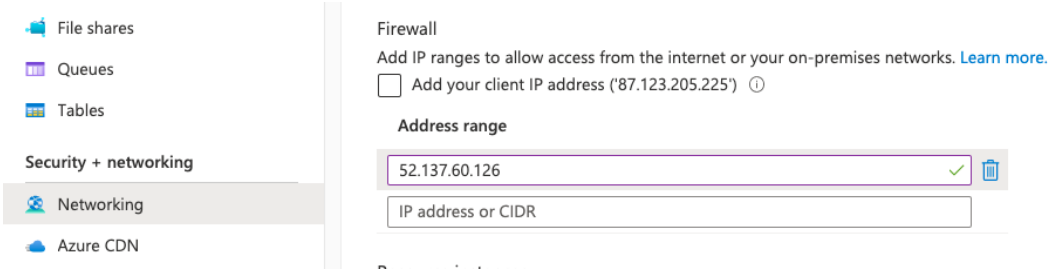
Customer created resources like virtual machines have to be connected to the subnet **labSubnet**. All other subnets must not be used in any kind.

It is not allowed to make changes on this virtual network. If changes are made, they will be rolled back without any notice.

12.20.2. Lab Firewall

All internet traffic which is requested by a lab resource is routed through our common lab firewall which has following public ip:
52.137.60.126/32
That means, that if you do a `curl ifconfig.me` via your lab virtual machine or AKS pod for example, this request is sent from the above ip.

To protect Azure services use the Networking→Vnet-Integration to only allow access from your lab vnet or if not available, activate the firewall feature on this service and allow above ip address to only allow traffic from BDC Labs to your resource.



12.21. BDC Lab Support

We offer support on Azure resources that are deployed and managed by BDC. As all projects are different and have different architecture, customers should be taking care of managing their resources which includes but is not limited to patching OS and Software vulnerabilities, upgrading AKS, checking EISA compliance, and configuring infrastructure in compliance with Bosch advised Security Configurations.

Please refer below examples of what resources managed by who,

1. Resources Managed by BDC: Application Gateway, Vnet
2. Resources managed by Customers: VM's, storage account that are deployed by customers etc..
3. Resources that has responsibilities at both sides:

a. AKS is deployed by BDC team and regular AKS upgrades and patching should done form customers end.

b. Traefik will be upgraded to latest version by BDC team during planned release and its configuration should be managed by customer as per the reference configuration provided in user guide.

12.22. Feedback

We look forward to receiving your feedback, which will help us to improve in the future!

Chapter 13. Network zone in the Bosch network for build agents

SL4 network zone is a bridge between Bosch Network and Bosch Development Cloud (internet). In case this setup doesn't match your needs, you can order your own network zone and reference our setup (EPITOP reference number: CICLCRM-3279).

As the communication from Internet to Bosch Internal Services is restricted with firewall. This service provides a network zone which can be used to place VM based build agents (Github Actions, Azure DevOps Agents or Jenkins agents) with access to several Bosch **internal** services.

If you need a managed GitHub runner access to the Bosch network, take a look at link: GitHub Runner as a Service.

If you only need a GitHub self-hosted runner without access to the Bosch network, take a look at link: GitHub self hosted Runners.

13.1. What can I order?

As customer, you can order VMs to be placed into this network zone and you might further need to setup below networking configurations.

13.1.1. How can I order a VM?

A VM can be ordered via ITSP, follow below steps.

Go to IT Service Portal and look for **virtual server** and select the one as per below screenshot,

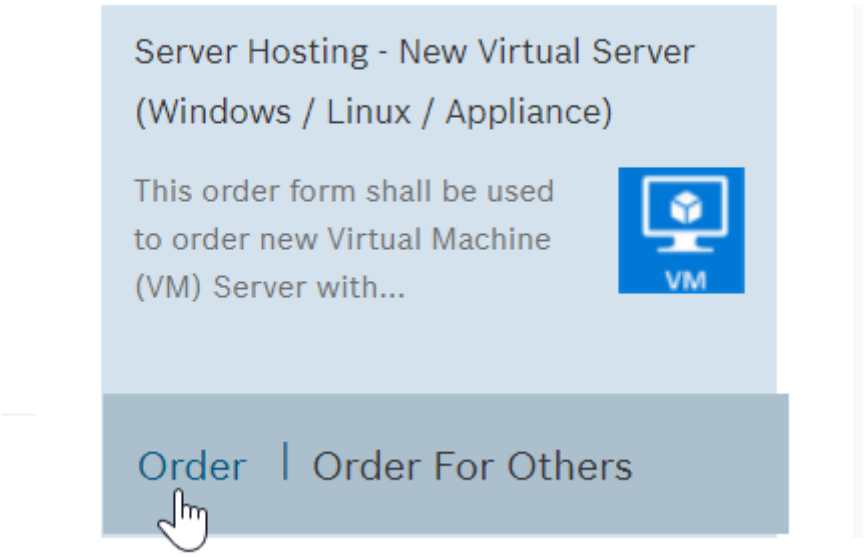


Table 7. Some useful information

Question	Answer
Security Zone Server	Yes
Region	EMEA
Country	Germany
Server Location	Feuerbach
Security Zone Layer	RSZ-SL4
Security Zone	RSZ-SL4-0260-BDCCICD_ST

Everything else is up to you to define.

The following instructions are an example from the BIOS-community: https://github.boschdevcloud.com/bios-emthacks/fossid-action/wiki/VM-in-SL4

13.1.2. Technical details for the network zone

- Network Name: RSZ-SL4-0260_BDCCICD_ST
- Location: Stuttgart
- IPv4 Network: 10.82.214.128/25
- IPv4 Gateway: 10.82.214.129
- VLAN: 260
- Type+Security Layer: RSZ-SL4
- Active Directory: BCD
- Distribution Area: CCSt_DA-CCS-1

13.2. Network connectivity

You might also require some additional configurations, while we have some zone-wide connections already enabled.

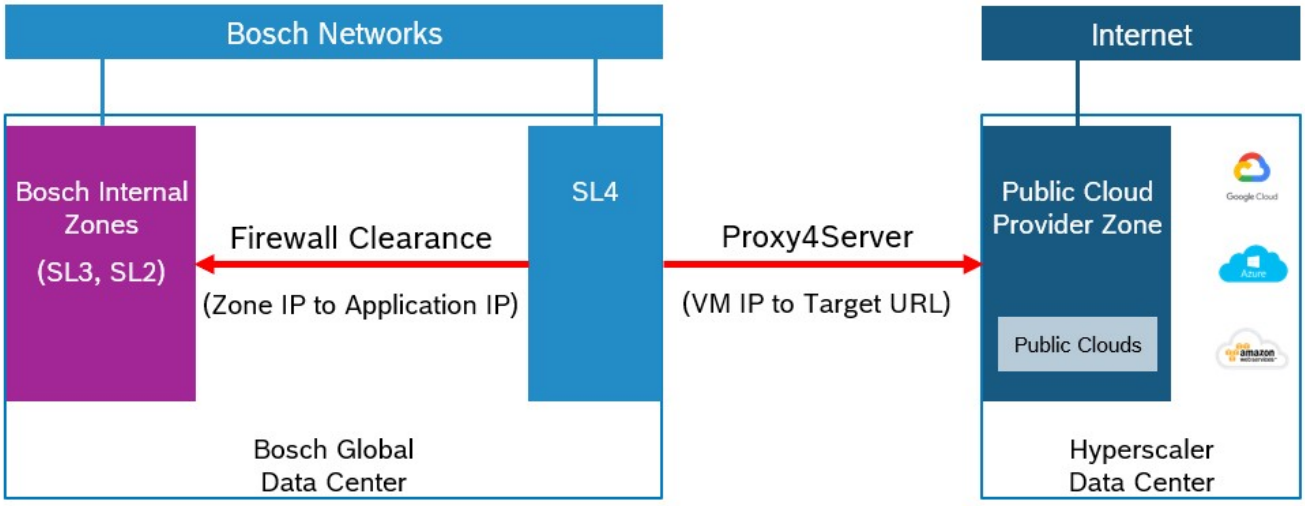
1. Proxy4Server - This is to connect to resources in internet. Connection from your VM to GitHub, Azure Devops, or other internet based resources.
2. Firewall clearance - If you would like to access some servers in Bosch Network that are not listed below. You need to request firewall clearance to whitelist your server IP.

13.2.1. Bosch network zone concept

The Bosch network has several security layers, SL0 to SL5. The higher the number, the more secure and trusted is the network. In general, more trusted networks can open connections to less trusted networks, e.g. from SL4 to SL3. Connections from less trusted networks to more trusted networks require additional security services like API gateways of Web Application Firewalls.

We decided to place our VMs in a SL4 zone as we can connect from here to the standard BCN network/SL3 (via firewall clearance) as well as to the internet (via Proxy4Server). As the internet is

untrusted, we cannot allow resources in the internet to open a connection to a VM in the SL4 zone. Instead, the VM has to open the connection to the internet service, e.g. GitHub as this ensures the more trusted resource controls the connection. Data can flow in both directions after the connection is established. (Same applies to all less trusted networks, e.g. BCN).



13.2.2. Which Bosch internal services can be accessed from this network zone?

Here is a list of the currently enabled connections for the whole zone. In case you need additional services for the zone enabled, please get in touch with us.

The following services are enabled (all only HTTPS on Port 443)

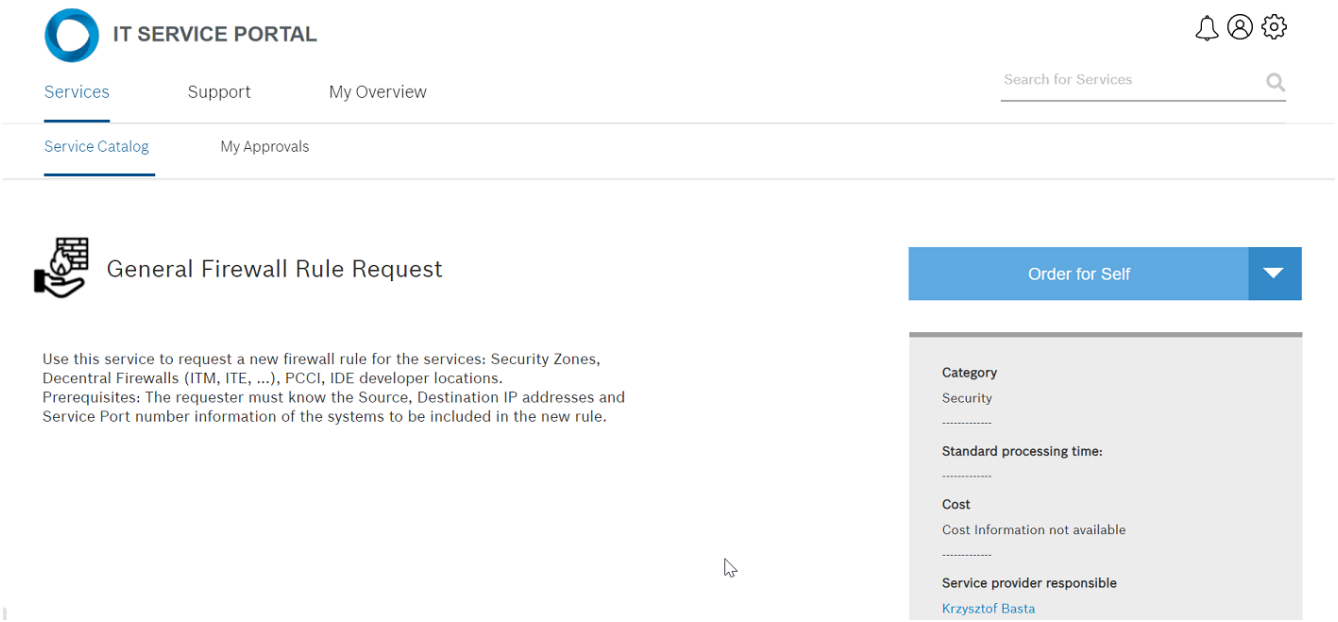
- Social Coding
- SonarQube
- RB Artifactory
- Docupedia
- Track+Release
- ALM-5
- ALM-11 (AE)
- JMaaS
- DTR
- mirror - OSD
- FossID
- AI Platform K8s

13.2.3. Firewall clearance

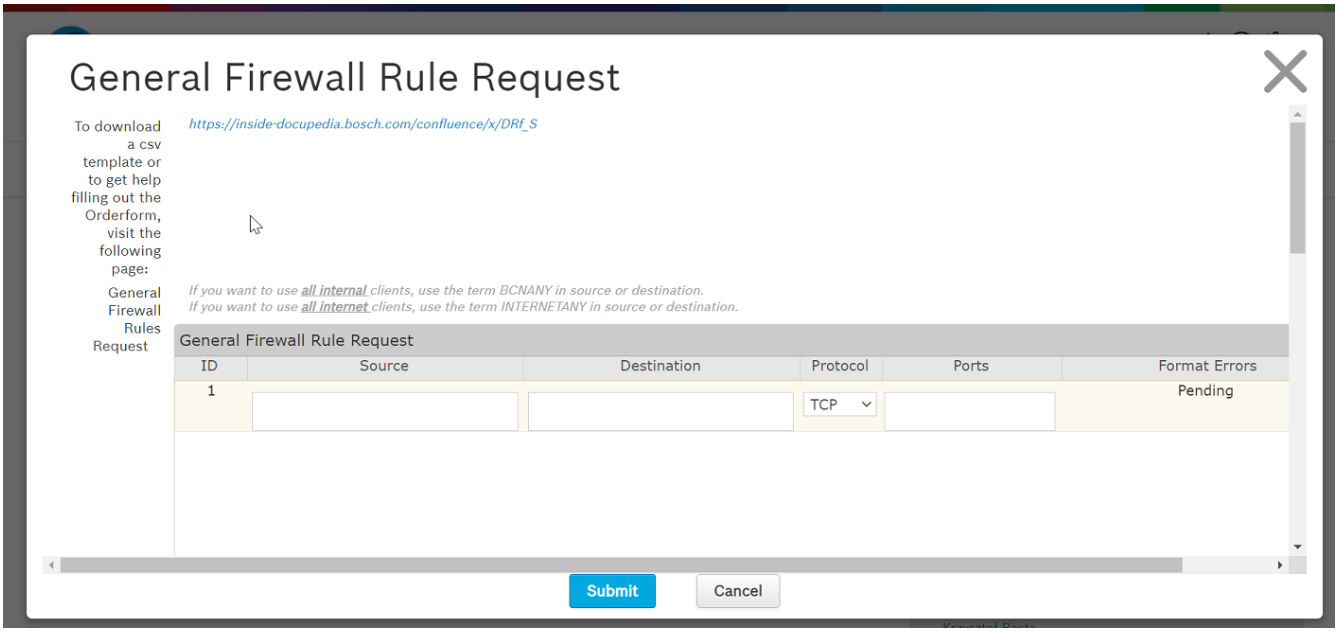
In order to access the Bosch Internal services, you have to request Firewall clearance via the following form: ITSP portal

For more details refer to this documentation.

Refer the below screenshot to order Firewall clearance request -



Refer the below screenshot to raise a Firewall clearance request -

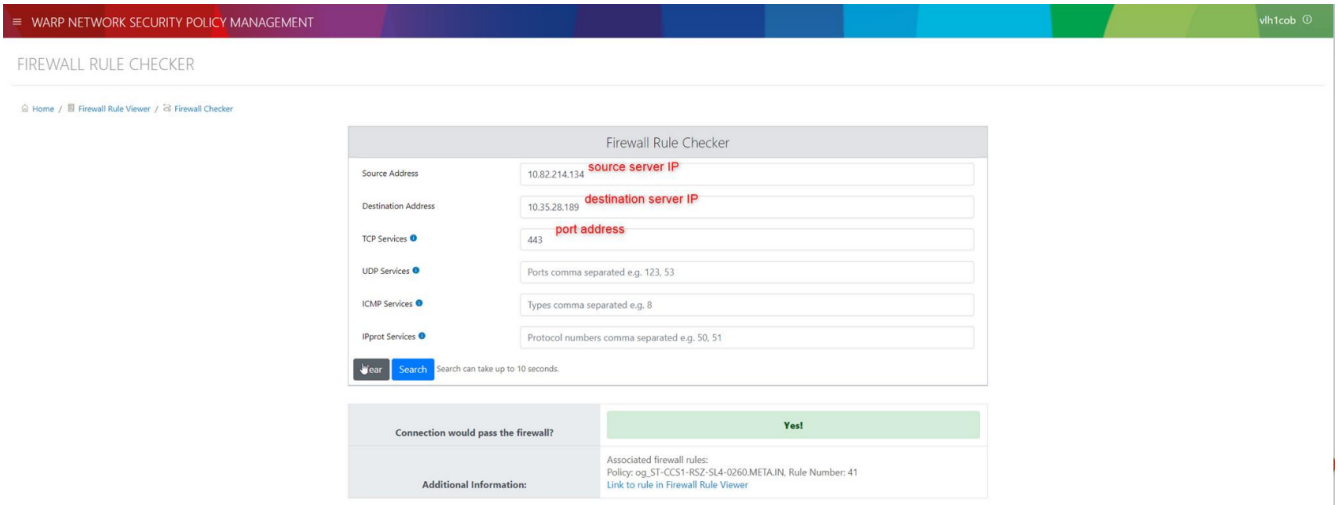


Request you to raise firewall clearance request for single IP and port and do not use ranges.

Hint: How to check whether firewalls allows your traffic?

Use Firewall Rule Checker tool to determine if the traffic is allowed. Access to this tool and usage of the tool is documented here.

Refer the below screenshot in order to check the firewall rule -

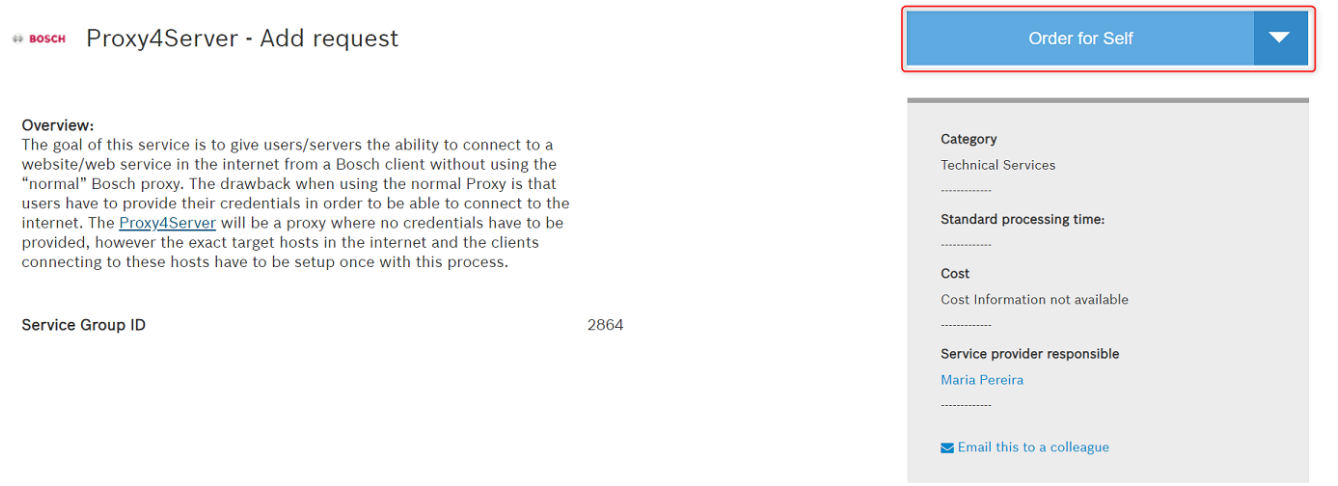


13.2.4. Proxy4Server

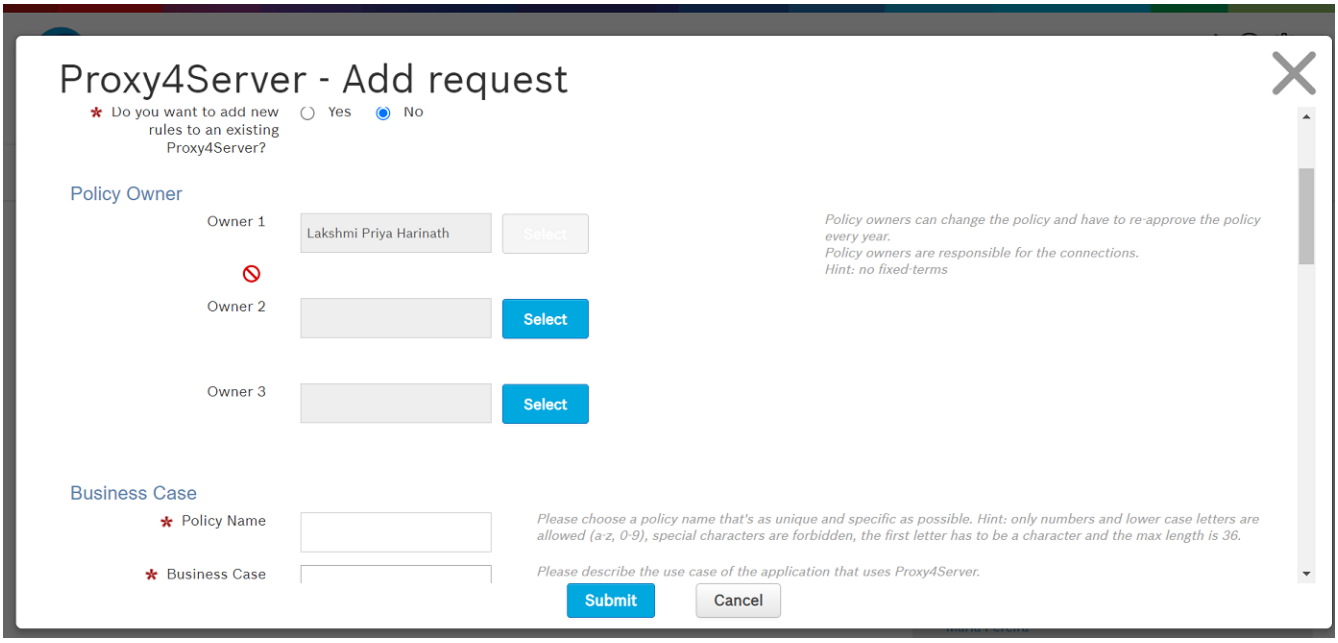
In order to access the Internet, request a Proxy4server configuration via ITSP portal

For more details refer this documentation.

Refer the below screenshot to order Proxy4server configuration -



Refer the below screenshot to place a new request for Proxy4server configuration -



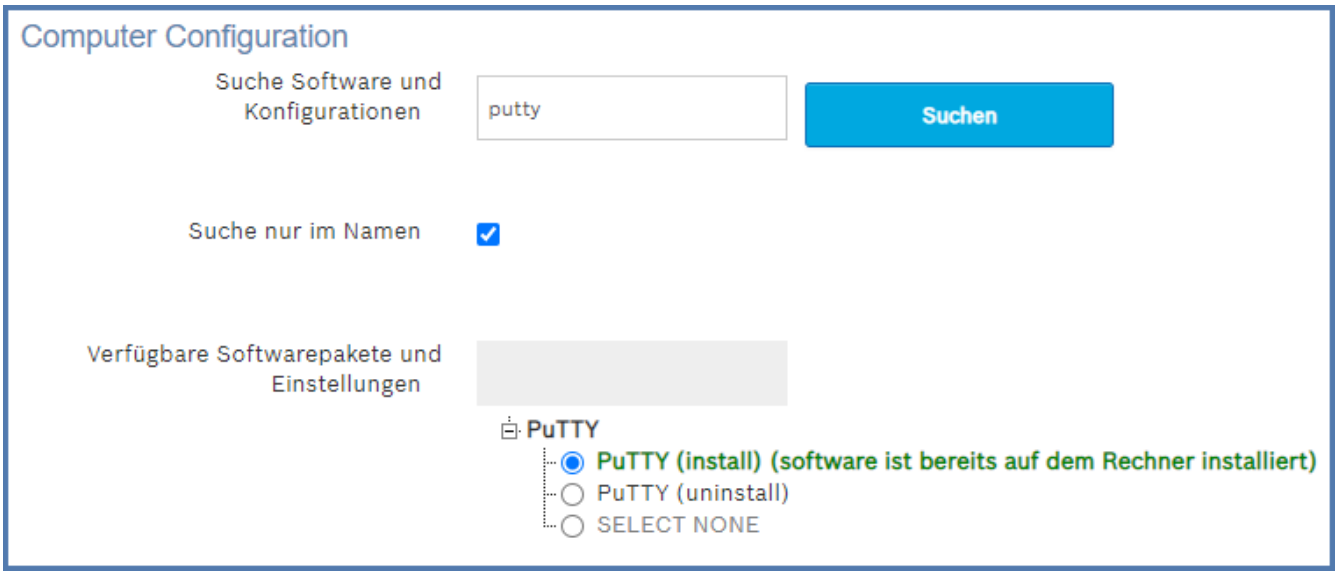
Question `Did you take care of the Cloud Onboarding?` can be marked as yes, if you wanted to connect to BDC resources - for example Github.

Make sure you request all required resources, incl. SAML login pages and sub-domains.

13.3. Preparation for PAM Access

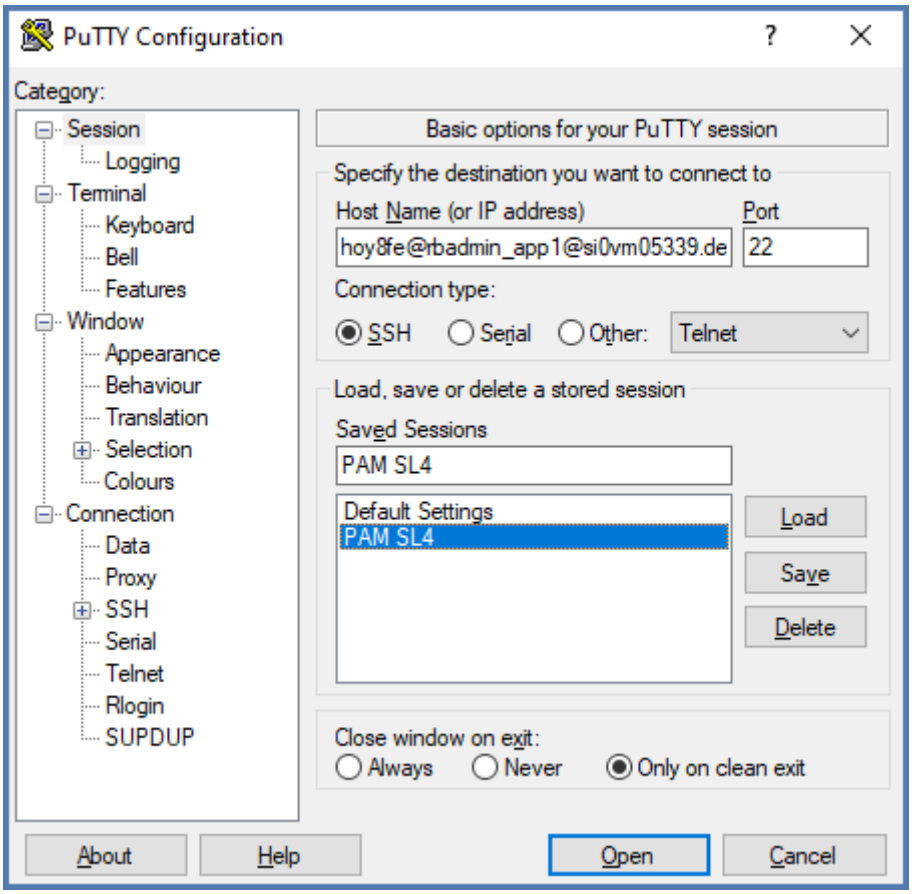
In case you need access to the PAM (Personal Access Management) mechanism for your Server / VM in SL4 Zone , please refer the RB-PAM User's Guide

To access a PAM server, you need PuTTY. You can get it via our MyIT Service Portal.
Here are some more details in the Client Software.

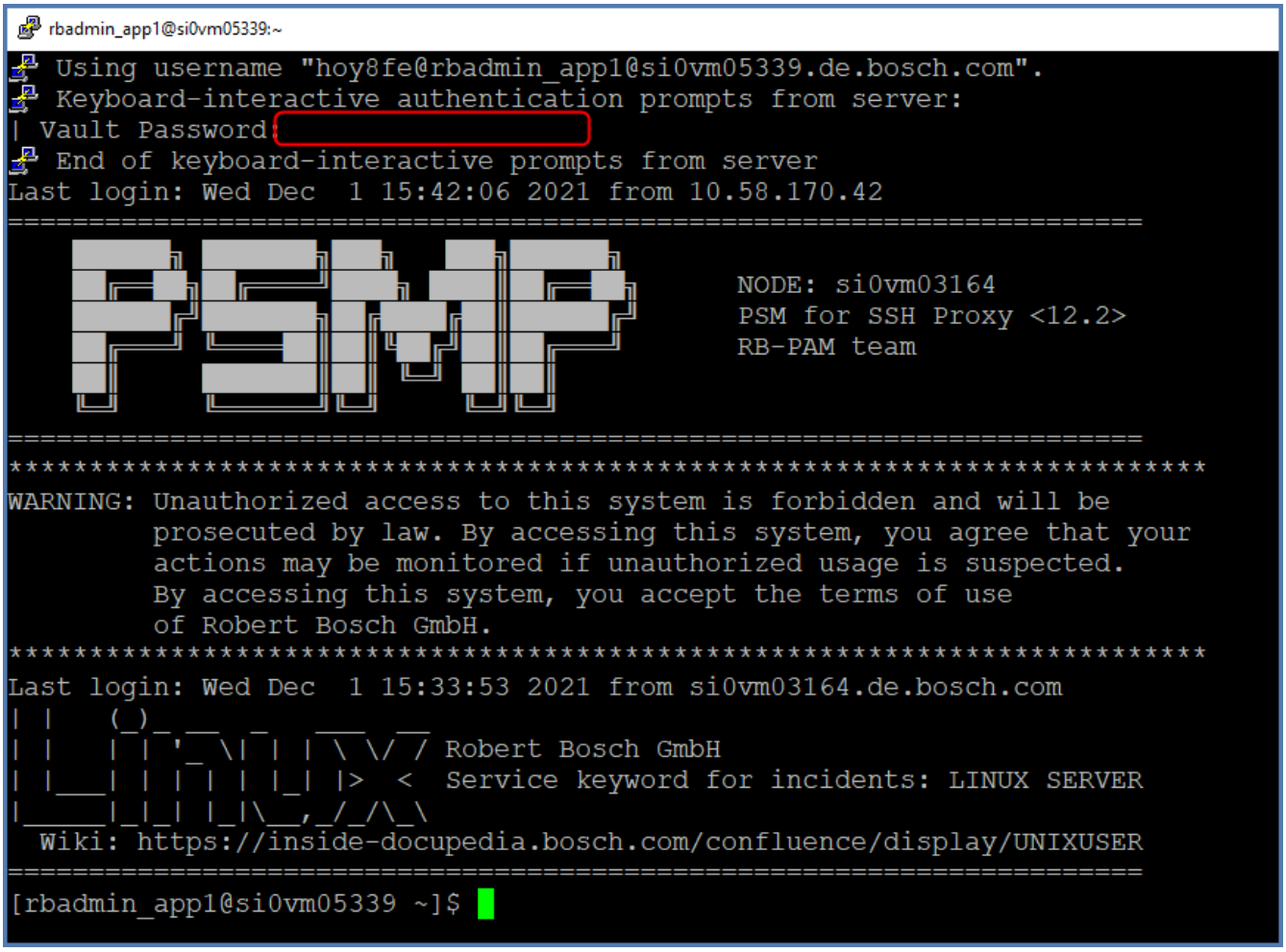


13.4. Using PAM via putty

- Start PuTTY and create a new session.
- Enter `<user_name>@<target_user>@<fq-servername>@rb-psmp.bosch.com` in the Host Name and port `22`, then save it.
- Now you can use these settings via one click each time. If you want, you can set some more default settings (like a bigger font size or different colors).
- You can find the possible target-users on this website, you can search for servername and it will list possible usernames/target_users.
- In my case, it would be `hoy8fe@rbadmin_app1@si0vm05339.de.bosch.com@rb-psmp.bosch.com`
- At the next start, you only need to "Load" these settings or double click on this setting to start.



Now you only have to enter your password, when you're asked for the Vault Password.



That's it - you're connected to your virtual machine in the SL4 zone!

13.5. How to install a self-hosted GitHub Runner

You really need to understand to 100%, what your GitHub Actions / Workflows are doing! By default your SL4 server does not have access to the internet, at all. *Every access* to the internet needs to be configured with your Proxy4Server configuration. The action/workflow itself comes from the github.com server. Therefore you at least need these three URLs in the Proxy4Server config:

- github.com
- api.github.com
- codeload.github.com

And everything else is up to you.
If you don't fully understand your Action and which internet access it needs, you probably will be pretty busy with troubleshooting for some time.

About the basic GitHub Runner installation: It's documented in the official GitHub documentation.
And also here on our BDC-GitHub in an internal BIOS repository.

Now you just need to follow the documentation from GitHub.

13.6. How to connect a Jenkins agent in SL4 with a master in Azure

There is a great documentation on how to connect your agent on Docupedia.

13.7. FAQs

Connection to license server for the SL4 servers is possible?

Yes, license servers are located in SL3 zone and hence you will require to raise a Firewall request as mentioned in one of the above chapters.

How to identify their server is located in which security zone?

Using server IP address, we can identify the security zone. Refer the WARP tool for the same.

Where can I find detailed information about Network Segmentation in the Bosch?

Refer to the document from network team.

Chapter 14. CloudIA

14.1. What is CloudIA?

CloudIA is a cloud service offering developed and operated by XC-ECO, with a strong focus on the needs of automotive SW projects.

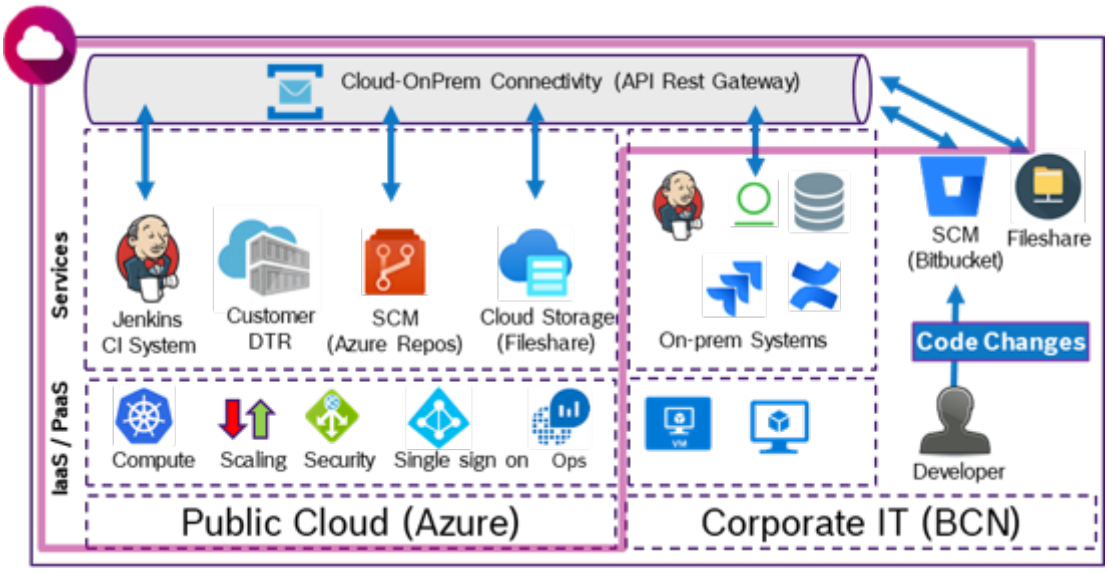
With CloudIA, we aim at enabling Software-centric projects in the following areas, by offering modular cloud services & solutions.

- DevOps: Automate CI/CD workflows
- Scalability of PMT infrastructure
- Co development with external partners & 3rd party suppliers
- Continuous SW Updates over lifetime

More information can be found here

14.2. SCALABLE CI:

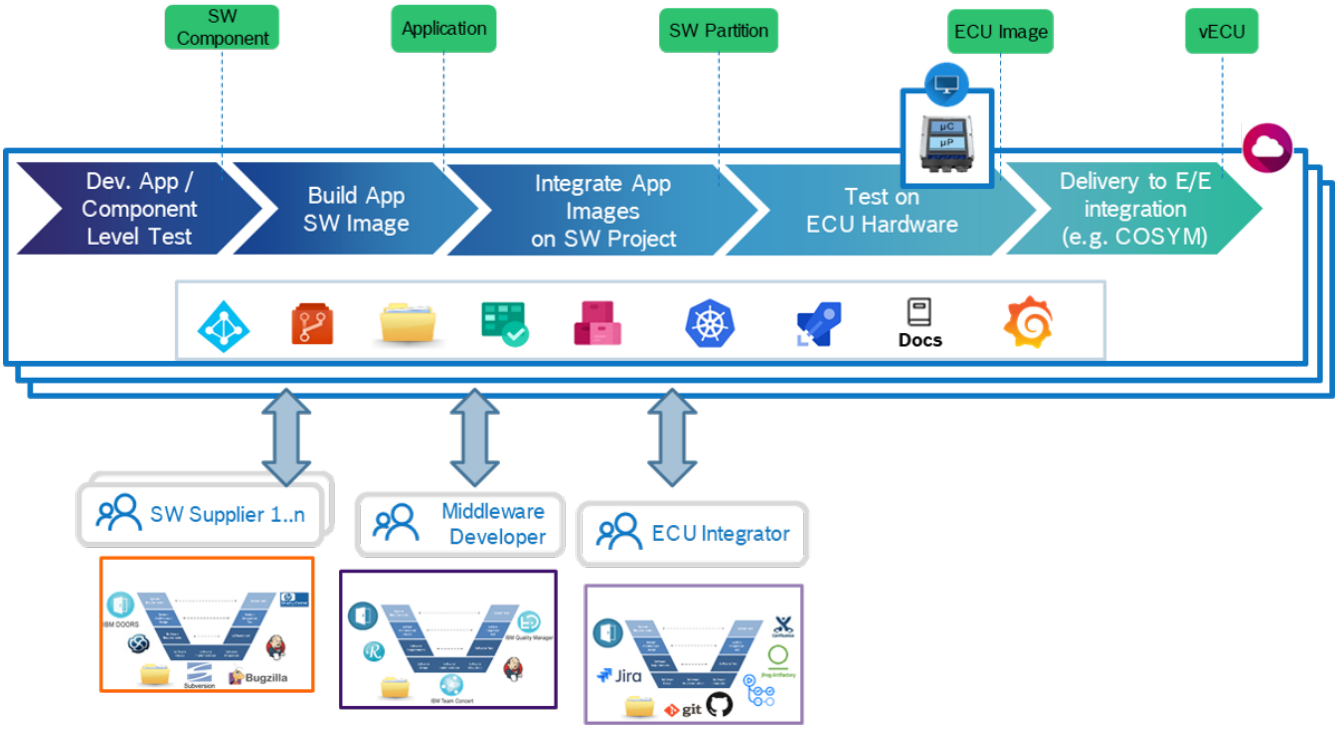
Projects striving for a Continuous-X (CX) integration workmodel aim to speed up the SW development & delivery iterations, by automating their complete SW integration workflows. This workmodel increases the demand for CI/CD pipeline executions, which often results in a bottleneck of CI Infrastructure. With CloudIA, we enable SW Projects to scale their CX pipelines using Cloud CI Infrastructure, by connecting on-premise and cloud systems.



14.3. CODE@CLOUD:

In cloud centric projects, developers work directly with cloud repositories for their SW artefacts, and the projects use SaaS and self-hosted cloud services to make best use of open market solutions. The resulting challenge for these projects are increased efforts for compliance & security of cloud services, and the migration of automotive specific tools and workflows to cloud environments. With

CloudIA, we provide managed tools and workflows for automotive development in the cloud, and offer a production-ready environment for external SW collaborations with partners.



Chapter 15. Metron

15.1. What is Metron?

Metron is a framework which allows you to seamlessly monitor your Software engineering efficiency by displaying various KPIs of your working process. Supported KPI's include the built duration, the success ratio, the branch lifetime, the avg. approval time and many more.

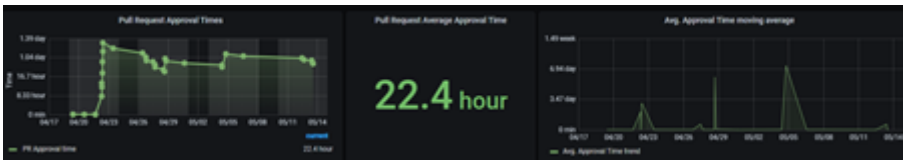
By providing an overview of the data surrounding you Software Engineering Process you can uncover bottlenecks in your workflow and speed up your development as well as the delivery.

You can find more information about Metron on the corresponding Docupedia page.

15.2. What KPIs are currently supported and what are the corresponding data sources?

Displayed KPI	Supported tool
Success Ratio	Jenkins
Build Duration	Jenkins
Builds per Day	Jenkins
Mean time between failures/ Mean Time to Restore	Jenkins
Avg. Branch Lead Time	
Avg. Branch Lifetime	Github, Bitbucket
Avg. Approval Time	Github, Bitbucket
Avg. Review Time	Github, Bitbucket

This is an example of how the KPI Avg. Approval Time is displayed by the Metron framework:



15.3. How can I order and set up a Metron subscription?

A Metron subscription can be ordered via the BDC Portal.

The initial deployment should take less than 30 minutes. Once completed, an e-mail will be send containing all the information needed to set up your tools to report to Metron. The status of your Metron subscriptions can be tracked via the BDC Portal. The technical contact specified for each Metron subscription will also receive e-mail notifications when any kind of change is made to their

subscription.

Steps of the ordering process:

- Order via the BDC Portal
- Await confirmation e-mail
- Set up your tools (eg. Github, Jenkins)
- View dashboards
- Manage access with IDM roles

Use this link to find more detailed instructions for each step of the ordering process of Metron: link

Chapter 16. Release Notes

The release notes for our regular updates can be found here