

Oracle® Cloud

Administering Visual Builder Studio



Release 24.10.1

F96930-02

August 2024



Copyright © 2020, 2024, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	viii
Documentation Accessibility	viii
Diversity and Inclusion	viii
Related Resources	viii
Conventions	ix

1 The Basics

What Is Oracle Visual Builder Studio?	1-1
What's an Extension?	1-1
Build Enterprise Apps in Minutes	1-2
Use VB Studio to Manage Your Development Cycle	1-4
How Do I Set Up VB Studio?	1-4
Key Concepts, Components, and Terms	1-5
OCI Resources in VB Studio	1-7

2 Create an Oracle Visual Builder Studio Generation 2 Instance

Before You Begin	2-1
Get the Right Roles	2-2
Create an OCI Compartment	2-2
Create the VB Studio Instance	2-3
Access VB Studio from the Oracle Cloud Home Page	2-6

3 Set Up VB Studio for CI/CD

Step 1: Decide whether to use the free VM build executor	3-1
VB Studio's Free VM Build Executor	3-2
Configure VB Studio to Use the Free VM Executor for Build Jobs and Pipelines	3-3
Add the Free VM Build Executor Manually	3-4
Step 2: Decide whether to run VM build executors or Docker executors	3-5
What are Build Executors and Build Templates?	3-5
VM Build Executors	3-5

Docker Build Executors and Docker Images	3-10
Step 3: Set up VB Studio to run VM build executors	3-13
Create and Manage Build Executor Templates	3-14
Add and Manage VM Build Executors	3-16
What Determines How Many VMs I Can Create?	3-22
Connect to Your OCI Account and Add VM Build Executors	3-23
Run VM Build Executors in Virtual Cloud Network	3-24
Use VB Studio's Default VCN	3-24
Run VM Build Executors in Another Compartment's VCN and Subnets	3-26
Software for Build Executor Templates	3-35
Step 4: Set up VB Studio to run builds on Docker executors	3-37
Add Your First Docker Deployment VM	3-38
Migrate to Docker	3-41
Create and Manage Docker Images	3-42
Add and Manage Docker Deployment VMs	3-44
Add Docker Deployment VMs	3-44
Manage Docker VMs	3-45
Manage Docker VM Resources	3-46
Reset Docker and Move to VM Build Executors	3-48
Step 5: Set Up IDCS Users and Groups	3-48
Step 6: Configure your OCI resources	3-49
Set Up the OCI Account	3-50
Get the Required OCI Input Values	3-52
Set Up the OCI Connection	3-53

4 Set Up VB Studio to Extend Oracle Cloud Applications

Before You Begin	4-1
Essential Set-up Tasks	4-4
Create a Simple Extension	4-5
What's Next?	4-10
Set Up VB Studio Users	4-11
How Do I Assign the Roles?	4-12
What Project Artifacts Are Created?	4-21
Advanced Administrator Tasks	4-23
Set Merge Restrictions on the main Branch	4-23
Set Up the Project to Deploy to Production	4-24
Add the Oracle Cloud Application's Production Instance to an Environment	4-25
Create a Production Branch	4-26
Create and Configure Production Build Jobs	4-26
Create and Configure the Production Pipeline	4-31
Run Production Pipelines	4-32

Set Up the Project to Deploy to Other DEV and TEST Instances	4-32
Add an Oracle Cloud Application Instance to an Environment	4-32
Create and Configure Deployment Build Jobs and Pipelines	4-34
Run the Pipeline	4-38
Delete an Extension	4-38
Delete an Extension Manually	4-38
Delete an Extension Version Manually	4-40
Configure a Job to Delete an Extension	4-40
Manual Administrator Tasks	4-42
Manually Create a Project for Extensions	4-42
Access VB Studio from Oracle Cloud Applications	4-44

5 Set Up VB Studio for Developing Visual Applications

Set Up Visual Builder Studio for Visual Applications	5-1
Get Access to Visual Builder Instances	5-2
Get the Required IDCS Roles	5-3
Set Up the VB Studio Project	5-3
Create and Set up a Project	5-3
Create and Set Up a Project for Development (Different Identity Domain)	5-4
Create and Set Up the Project for Development (Same Identity Domain)	5-6
Create a Project	5-7
Configure the Deployment Job	5-10
Run the Pipeline Manually	5-11
View the Deployed Visual Application	5-11
Add Users to the Project	5-12
Optional Configuration	5-13
Add More VM Build Executors or Build Executor Templates	5-14
Configure the Packaging Job	5-15
Set Merge Restrictions on the main Branch	5-16
Develop Your Application	5-16
Set Up Additional Deployment Instances	5-17
Set Up the Project to Deploy for Production	5-18
Create a Production Branch	5-18
Create and Configure Production Build Jobs	5-18
Create a Production Packaging Build Job	5-19
Create a Production Deployment Build Job	5-20
Configure a Production Job's Privacy Setting	5-21
Create and Configure a Pipeline	5-21
Undeploy a Visual Application	5-22
Undeploy a Visual Application Manually	5-22

6 Manage the VB Studio Instance

Change the CI/CD Compartment	6-1
Switch to the Built-In Free Account	6-1
Change Your OCI Account	6-1
Validate the OCI Connection	6-2
Manage VB Studio Instance Tags	6-2
Write IAM Policies that Control Access to VB Studio	6-3
Create Rules that Use Service Event Notifications	6-4
Move the VB Studio Instance	6-5
Terminate the VB Studio Instance	6-5

7 Manage the Organization

Update the Organization's Display Name and Description	7-1
Set the Organization's Default Wiki Markup Language	7-1
Restrict Project Creation to Organization Administrators	7-1
Restrict Sharing a Project to Organization Administrators Only	7-2
Limit the Number of Personal Access Tokens a User Can Create	7-2
Assign a Project's Ownership to Another User	7-4
Manage Your Organization's Projects	7-4
Manage Your Organization's Groups	7-7

8 Manage Your Component Exchange

What is a Component Exchange?	8-1
Use a Project as the VB Studio Component Exchange	8-1
Use an External Server as the VB Studio Component Exchange	8-3
Share the Component Exchange's URL	8-3
Workflow for Web Components	8-4

9 Troubleshooting

Unable to Access Instance After Switching Users to Federated IDCS	9-1
Unable to See the "Edit Page in Visual Builder Studio" Option in an Oracle Cloud Application	9-1
How Do I Resolve a "You Are Not a Member of This Organization" Error?	9-2
How Do I Access the IDCS Console From the Oracle Cloud Console?	9-2
Resolve Access Issues After Allow-Listing Your VB Studio IP Address	9-2

Part I Appendices

A Set Up Oracle Visual Builder Studio (Non-Native)

Get the Right IDCS Roles	A-2
Create the VB Studio Instance	A-3
Access VB Studio from the Oracle Cloud Home Page	A-5
Configure Your OCI Resources for CI/CD	A-6
Oracle Cloud Infrastructure: Terms of Use	A-7
Set Up the OCI Account	A-7
Get the Required OCI Input Values	A-9
Set Up the OCI Connection in VB Studio	A-10
Manage Your VB Studio Instance	A-12
Verify Your Compute and Storage Connections	A-12
Change Your OCI Account	A-13
Switch to Another OCI Account	A-13
Switch to an OCI Account From the Built-In Free Account	A-13
Switch to the Built-In Free Account From an OCI Account	A-14

B Migrate Oracle Cloud Applications Configurations

Before You Configure CSM Build Jobs and Pipelines	B-1
Create and Configure Build Jobs to Migrate Your Oracle Cloud Application's Configurations	B-3
Configure a Job to Export Configuration	B-4
Configure a Job to Import Configuration	B-5
Configure a Job to Cancel the Configuration Migration	B-7
Configure a Job to Apply and Save the Configuration Migration	B-8
Configure a Job to Restore Configuration Migration	B-9
Create and Configure the CSM Pipeline	B-10
Create and Configure Build Jobs to Deploy the App Extension to the Sandbox	B-14
Create the Production Sandbox Packaging Job	B-14
Create the Production Sandbox Deployment Build Job	B-16
Create and Configure the Production Sandbox Pipeline	B-18
Run the Pipelines	B-21

C Public IP Addresses for VB Studio

Preface

Topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Resources](#)
- [Conventions](#)

Administering Visual Builder Studio describes how to create the Visual Builder Studio instance and set it up for developing extensions, visual applications, and popular build frameworks such as Maven, Gradle, Ant, and Node.js.

Audience

This document is intended for Oracle Visual Builder Studio administrators who administer and set up the service.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

For more information, see these Oracle resources:

- [Oracle Public Cloud](#)

<http://cloud.oracle.com>

- Manage Your Development Process in *Using Visual Builder Studio*
- Create Visual Applications in VB Studio in *Building Responsive Applications with Visual Builder Studio*
- What Can You Do with Oracle Visual Builder Studio? in *Extending Oracle Cloud Applications with Visual Builder Studio*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Videos and Images

Your company can use skins and styles to customize the look of the application, dashboards, reports, and other objects. It is possible that the videos and images included in the product documentation look different than the skins and styles your company uses.

Even if your skins and styles are different than those shown in the videos and images, the product behavior and techniques shown and demonstrated are the same.

1

The Basics

Learn about Oracle Visual Builder Studio, its projects, components, roles, and how to access the service.

See Oracle Cloud Terminology in *Getting Started with Oracle Cloud* for definitions of terms found in this and other documents in the Oracle Cloud library.

What Is Oracle Visual Builder Studio?

Oracle Visual Builder Studio (VB Studio) is a robust application development platform that helps your team effectively plan and manage your work throughout all stages of the app dev lifecycle: design, build, test, and deploy. It makes it easy for your entire team to develop the artifacts they need, including:

- Oracle Cloud Applications developers, who need to extend their applications with business-specific customizations;
- Low-code developers, who want to create applications using a visual designer;
- Experienced programmers, who want to modify the source code for applications created by others, or to develop bespoke apps using the web programming language of their choice.

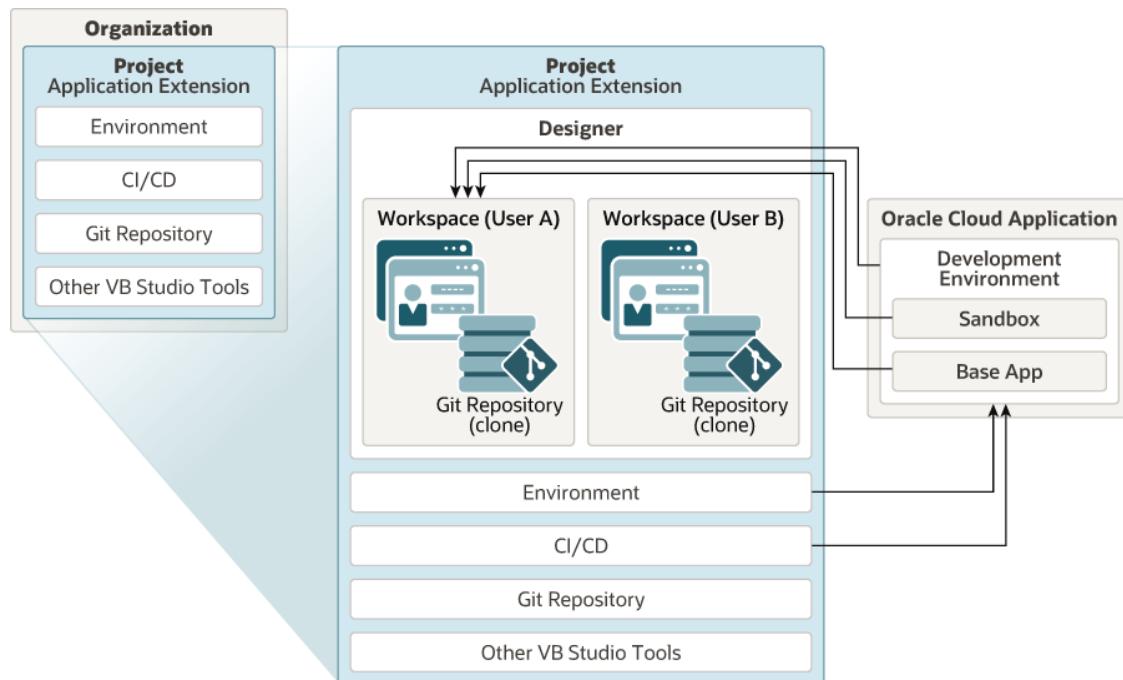
With VB Studio you get:

- A rich visual designer integrated with source control (Git) so that developers can manage changes, apply version control best practices, and collaborate with their teammates to develop applications
- The ability to build and display different flavors of the UI to meet the needs of discrete users of certain Oracle Cloud Applications (those built with VB Studio and Oracle JavaScript Extension Toolkit (Oracle JET)), also within a Git framework
- Built-in repositories for hosting code in Git and for hosting binaries, such as Maven dependencies
- A continuous integration service so you can automate your build and test systems
- A continuous delivery service that tightly integrates with Oracle Cloud Applications
- Agile boards and an issue tracking system for tracking sprints, tasks, defects, and features

VB Studio enables developers to easily deploy their applications to their preferred target, whether it's a staging or production instance of Oracle Cloud Applications or an Oracle Cloud Infrastructure (OCI) service instance.

What's an Extension?

The changes you make to your Oracle Cloud Application in VB Studio are stored in an artifact called an *extension*. Physically, the source files associated with the extension are stored in a Git repository.



Here are the key components of the VB Studio ecosystem as it pertains to extensions:

- Within a single Oracle Cloud Application environment family—which comprises all your development, test, and production instances—you have multiple instances of VB Studio, all of which are tied together into a single *organization*. Within your organization, you will likely belong to one or more *projects*, each of which is devoted to a discrete software effort. For example, you might have a project for building extensions for Oracle Cloud Applications, and another project for building a bespoke application your own department will use. A project brings together all the tools you need to create those artifacts, such as a Git repository for storing your source code, a pipeline to provide continuous integration and delivery, an issue tracking system, team wikis, and more.
- When working on an extension, Oracle best practice dictates that you use **one** project for all the extension work you do within an Oracle Cloud Application environment family. If you're using the same environment family to extend multiple *pillars*—HCM, CRM, and so on—you can create one Git repository for each, but they should all be hosted within the same project. That is, multiple developers can work together to develop the extension, but they should all be working from the same repository.
- When you work on an extension, you do so within the editors provided in the VB Studio Designer. Depending on which aspect of the page you're customizing, VB Studio invokes the proper editor to provide the experience you need.
- All of your work in VB Studio is done in the context of a *workspace*, a completely private area where you can work on your extension.

Build Enterprise Apps in Minutes

VB Studio makes it easy for you to build custom interfaces that meet your unique needs. No coding, no setup, and no IT resources required. It provides cloud-based visual tools to help you rapidly create and host engaging applications.

With VB Studio, you can:

- Use the what-you-see-is-what-you-get (WYSIWYG) designer to visually create your app: simply drag and drop UI components onto your app's pages, customize their behavior, and

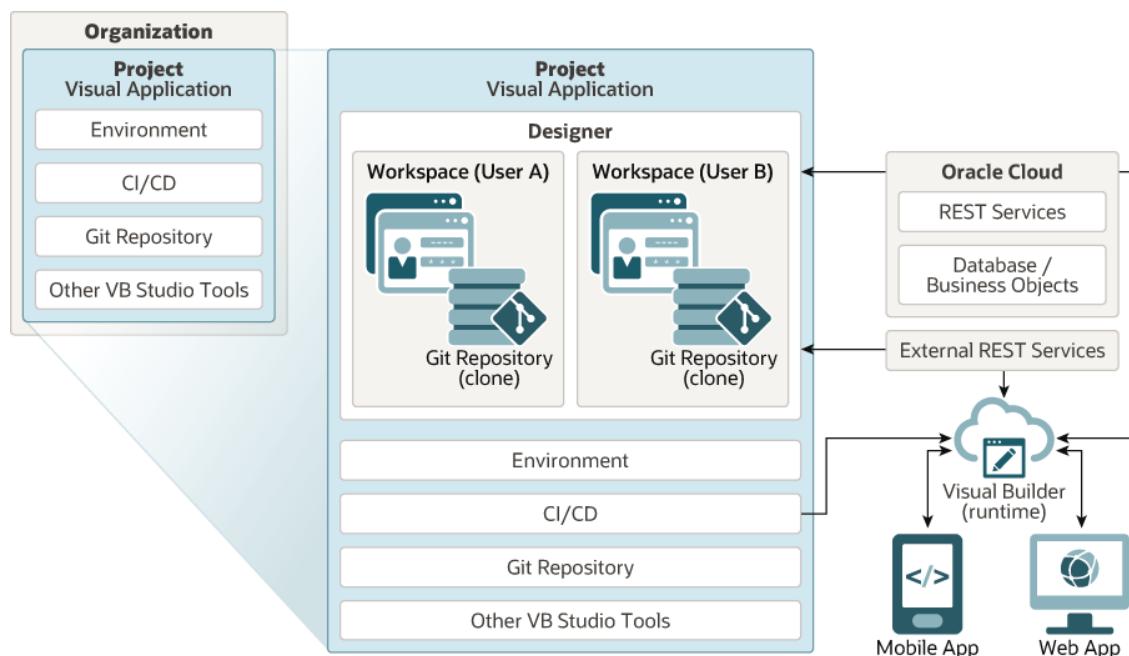
arrange them however you want. For more complex needs, you can extend functionality using standard JavaScript, HTML5, and CSS.

- Connect to data through REST-based services: you can create your own business objects that implement your app's business logic and store its data, pick data objects exposed by an integrated catalog of Oracle REST services (such as Oracle Cloud Applications and Oracle Integration), or pull data from any external REST service with just a few clicks.
- Deploy your application at the click of a button and make it available to users. This means you can publish your app instantly and access it on any device, anytime. You can even deploy it as a Progressive Web App (PWA)—combining the on-device mobile experience with a web app's ease of distribution—so it can be installed on user devices to leverage capabilities such as camera access and location information.

VB Studio also provides:

- Integration with Git repositories so you can share application resources with team members
- Infrastructure for securing access to your application, its data, and the Oracle Cloud services it consumes.

All of this becomes available to you within a *project*, which is the starting point for building a visual application in VB Studio. Here are the key components of the VB Studio ecosystem as it pertains to visual applications:



- Within a single VB Studio instance, you and your team members who use that instance are considered an *organization*. Within your organization, you will likely belong to one or more *projects*, each of which is devoted to a discrete software effort. For example, you might have a project for building a new Financial app, and a different project for creating an HR application. A project brings together all the tools you need to create those artifacts, such as a Git repository for storing your source code, a pipeline to provide continuous integration and delivery, an issue tracking system, team wikis, and more.
- All of your work in VB Studio is done in the context of a *workspace*, a completely private area where you can work on your visual application. Your work is stored in your own clone of the project's Git repository, and is not visible to others until you push it to the project's

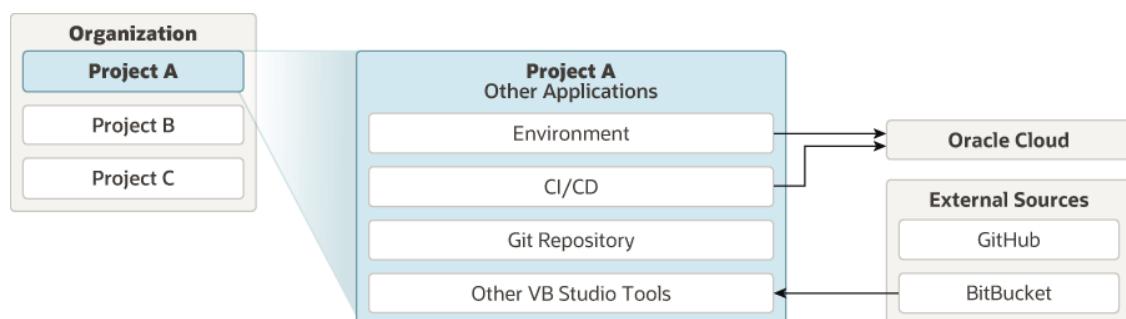
Git repo, choose to share it with others for testing, or publish it. A workspace also includes a pointer to the development or test environment where you plan to deploy your app, which must be a separate Oracle Visual Builder instance (also known as an environment). You can deploy your app manually, or wire it up to a pipeline to do it automatically, such as when a developer on your project publishes their changes.

- Your workspace is your ticket into the *Designer*, a declarative environment that helps you build visual applications. Specifically, you get a page editor, a Components palette, and a Structure view, for designing the layout of a page. Within this environment, you have access to a rich set of UI components, including those based on Redwood, the Oracle standard for user experience. You can use these components—all based on Oracle JET—to develop apps that provide the same look and feel as apps delivered from Oracle.
- The data for your visual applications can come from a database (in the form of business objects), Oracle REST services, or external REST services.

If you previously developed visual applications in Visual Builder and now work with VB Studio, you'll see that some things have changed. For details about tasks that you now do differently in VB Studio, see [For Visual Builder Users](#).

Use VB Studio to Manage Your Development Cycle

The VB Studio components shown in the diagram are used to manage the development process for applications other than visual applications or application extensions:



Though many users will rely on Visual Builder Studio to help them create visual applications or create application extensions for Oracle Cloud Applications, you can use VB Studio purely as a tool to help you manage your development processes:

- Within a single Visual Builder Studio instance, you and your team members who use that instance are considered an organization. Within your organization, you will likely belong to one or more projects, each of which is devoted to a discrete software effort. A project brings together all the tools you need to create those artifacts, such as a Git repository for storing your source code, a pipeline to provide continuous integration and delivery to the Oracle Cloud, an issue tracking system, team wikis, and more.
- Although VB Studio provides a Git repository for you, you can choose to use external repos, such as GitHub and BitBucket.

How Do I Set Up VB Studio?

Using VB Studio's hosted development and delivery platform, you and your team can develop software applications using any language you choose. You can also develop new visual applications or customize certain Oracle Cloud Applications by extending them. These extensions are then managed just like any other VB Studio artifact.

Before your team can use VB Studio you, as the administrator, need to set it up, configure it, add your team members to Oracle Identity Cloud Service (IDCS), and assign them proper roles.

 **Note:**

Only one IDCS is supported per service instance.

Most administrators will follow this path to set up VB Studio:

1. Everyone needs to read and follow the instance setup procedures in [Create an Oracle Visual Builder Studio Generation 2 Instance](#) and the ones for the build system in [Set Up VB Studio for CI/CD](#).
Users who deploy their extensions manually or deploy their visual applications using Grunt, don't use the CI/CD system. For these users, setting up the CI/CD system is not required. See [Deploy an Extension From Your Local System](#) for extensions or [Build Your Application With Grunt](#) for visual applications.
2. Then, depending on your needs, you can move on to either the [Set Up VB Studio to Extend Oracle Cloud Applications](#) or the [Set Up VB Studio for Developing Visual Applications](#) chapter.
3. If you just want to manage the development process, you can stop after you complete the CI/CD setup.

Key Concepts, Components, and Terms

Before you use VB Studio, it helps to become familiar with these key concepts, components, and terms. (If you have general questions about Oracle Cloud features and services, see [Oracle Cloud Terminology](#) in [Getting Started with Oracle Cloud](#).)

Term	Description
Oracle Cloud Infrastructure (OCI)	Oracle Cloud Infrastructure is a set of cloud services that enable you to build and run a wide range of applications and services in a highly available hosted environment. Oracle Cloud Infrastructure offers high-performance compute capabilities (as physical hardware instances) and storage capacity in a flexible overlay virtual network that is securely accessible from your on-premise network.
Oracle Cloud Infrastructure Compute (OCI Compute)	Service that hosts virtual machines (VMs) on Oracle Cloud with all the necessary storage and networking resources. VB Studio uses the VMs to run project builds.
OCI Compute VM instance	A virtual machine that runs on top of physical bare metal hardware. To learn more about a Compute instance, see Overview of the Compute Service .
OCI Compute VM shape	A shape is a template that determines the number of CPUs, amount of memory, and other resources allocated to a newly created VM Compute instance. To find more about shapes, see VM Shapes .
OCI region	OCI is hosted in regions and availability domains. A region is a localized geographic area, and an availability domain is one or more data centers located within a region. To learn more about regions and availability domains, see Regions and Availability Domains .
OCI Virtual Cloud Network (VCN) and subnets	A VCN is a software-defined network that you set up in Oracle Cloud Infrastructure data centers in a particular region. To find out more about VCNs and subnets, see VCNs and Subnets and Overview of Networking .

Term	Description
Oracle Cloud Infrastructure Object Storage (OCI Object Storage)	Oracle Cloud service that hosts containers on Oracle Cloud to store project data. VB Studio uses the buckets to archive build artifacts and Maven artifacts, and export project data.
Oracle Cloud Applications	Oracle Cloud Applications are a set of modular Cloud-ready enterprise applications. To learn more, see https://www.oracle.com/applications/ .
Application extension (App UI)	An App UI is an application that includes a user interface component in the form of Visual Builder pages and flows. Some App UIs are created by Oracle—like some Oracle Cloud Applications—but you can build your own App UIs and deploy them as peers alongside Oracle's App UIs in your Oracle Cloud Applications ecosystem.
Visual Builder	A Visual Builder instance that provides the server for delivering pages in web applications, and services your apps might use to access data, including the database used to store data and the proxy server for managing connections to REST services.
Visual application	A responsive application developed using VB Studio's browser-based development environment. You deploy a visual application to a Visual Builder instance.
Project	A project is a collection of VB Studio features. You can use a project to host source code files, track issues, collaborate on code, build, and deploy your applications. A project can host multiple Git repositories. Each Git repository can have multiple branches and hundreds of code files. You can create a merge request for each branch of the Git repository and ask reviewers to review the code. You can create and configure multiple build jobs to generate different project artifacts that you can deploy to Oracle Cloud or your on-premise web server.
Organization	The top-most entity in the project structure of VB Studio. Think of an organization as the umbrella for all the projects in a given identity domain.
VB Studio Designer	VB Studio's browser-based development environment.
Git repository	A Source Code Management (SCM) and distributed version control tool to host source code files.
Maven repository	A hosted binary repository to store build artifacts, library files, and dependencies for Maven applications.
Issue tracker	A built-in issue management system to create and track tasks, defects, and features.
Environment	Defines the target Oracle Cloud Applications, Visual Builder, Oracle Cloud SaaS, or Oracle Cloud Infrastructure service instance as a single entity. You'll define an environment to deploy an application to a service instance or to get information from a service instance.
Merge request and code review	A method to merge a Git repository branch with another branch. Before merging the branches, team members can review differences between files of both branches and provide their feedback.
Wiki	Built-in wiki system to help your team author and manage wiki pages.
Build system	A built-in system to define and automate builds of your applications.
Continuous Integration (CI) and Continuous Delivery (CD)	Continuous integration is a set of practices that allow development teams to implement small code changes and push the code to version control repositories, such as Git, frequently. Continuous delivery is a practice that enable developers to produce software in short cycles.

Term	Description
VM build executor	An OCI Compute VM instance dedicated to run VB Studio builds. Only one build can run on a VM build executor at a time. To learn more, see VM Build Executors .
Build executor template	A template that defines the operating system and the software installed on a VM build executor. To learn more, see Build Executor Templates .
Docker image	Defines the operating system and software packages your organization needs to run builds on a Docker executor. A Docker image can be either imported from an external Docker registry or created from a build executor template.
Docker deployment VM	An OCI VM Compute instance dedicated to run builds of jobs defined in VB Studio projects using Docker executors.
Docker executor	When a job is created, a Docker image is associated as a build template with the job. Then, when the job's build is triggered, a Docker executor is created using the Docker image and deployed on any Deployment VM to run the build. Unlike a VM executor, a Docker executor is not directly associated with a specific VM.
Job (or build job)	A configuration that defines your application's builds. You can create a job to perform various actions, such as package artifacts, run shell commands, run unit test scripts, and deploy application artifacts.
Build	The result from a job's run.
Pipeline	A path or a chain of builds. A pipeline helps you run continuous integration jobs and reduce network traffic.
Oracle Java Cloud Service	Oracle Cloud service to deploy web applications to a public Oracle WebLogic Server domain on Oracle Cloud.

OCI Resources in VB Studio

In VB Studio, you use *continuous integration and continuous delivery* (CI/CD) build jobs to package and deploy your application. To run builds, VB Studio requires access to your OCI resources, such as your OCI Compute *virtual machines* (VMs).

You can use OCI resources either from your VB Studio instance's built-in free account, Oracle Free Tier account, or your own OCI account.

- Built-in free account: Available in all regions in the OC1 realm and will be made available in select other realms, such as OC2, OC3, and OC4, in the future. The VB Studio instance comes pre-configured with a built-in free account that offers one OCI Compute VM to run builds.
- Free Tier account: All OCI accounts have a set of resources that are free of charge for the life of the account. The free tier account offers [free micro Compute VMs](#) that your organization's members can use to run builds. To use the OCI Compute VMs, configure VB Studio to connect to the Free Tier account.
- OCI account: If you're subscribed to OCI, you can use your own account's OCI Compute VMs to run builds. To use OCI Compute VM, configure VB Studio to connect to your OCI account.

This table summarizes the differences between the built-in free account, free tier account, and your OCI account:

Features	Built-In Free Account	Free Tier Account	Your OCI Account	What it means?
Number of build executors	1	2-4 Depending on the size of the boot volume and the number of OCPUs that you allocate to each Ampere A1 Compute instance, you can create up to four compute instances.	Unlimited	A VM executor can run only one build at a time. The more VM executors you have available, the more builds your organization's members can run at the same time. If members trigger multiple builds while VM executors are already running, they must wait until a build running on the VM executor is complete. If VB Studio is connected to your OCI account, you can add multiple VM executors to reduce the build wait time for your organization's members.

Features	Built-In Free Account	Free Tier Account	Your OCI Account	What it means?
Storage for build artifacts	50 GB	200 GB	Unlimited	<p>When your organization's members run builds, VB Studio saves the generated build artifacts in its configured storage space. If you're using the built-in free account or the free tier account, the build artifacts are stored in a limited storage space. If you run out of storage space, builds that generate artifacts will fail. If this happens, you must either remove some artifacts from the storage, or switch to an OCI account. You can't increase the storage space allocated to your built-in free account or free tier account.</p> <p>If VB Studio is connected to your OCI account, the artifacts are stored in OCI Object Storage buckets. You can always increase the storage space to store more artifacts.</p>

Features	Built-In Free Account	Free Tier Account	Your OCI Account	What it means?
VM shape	Fixed shape	Flex shape	Custom shapes, including shapes offered by OCI	The built-in free account offers a fixed shape OCI Compute VM. The free tier account offers the fixed VM.Standard.E2.1. Micro micro shape as well as the flexible ARM-based VM.Standard.A1.Flex shape. Docker cannot be used with the VM.Standard.A1.Flex shape. See Flexible Shapes .
VCN	VB Studio's default VCN	2 VCNs	VB Studio's default VCN or your custom VCN	If VB Studio is connected to your OCI account, you can choose any VM shape offered by OCI. To learn more about shapes, see VM Shapes . If VB Studio is connected to your OCI account, you can add VM executors in a custom VCN to access Oracle Cloud services that are running in the same VCN.

Features	Built-In Free Account	Free Tier Account	Your OCI Account	What it means?
Wait time	5 minutes (fixed)	Custom wait time	Custom wait time	After running a build, a VM executor waits for some time to run any queued builds. If no builds run in the wait time period, the VM executor stops. If a build triggers after a VM executor has stopped, the VM executor takes some time to start before it could run the build. If VB Studio is connected to your OCI account, you can customize the wait time for your VM executors.
Maven and NPM repositories	Not available	Available	Available	If VB Studio is connected to your OCI account, your organization's members can use their projects' hosted Maven and NPM repositories to upload custom dependencies and binary files.

Create an Oracle Visual Builder Studio Generation 2 Instance

Everyone needs to follow the instructions in this chapter, regardless of how your organization intends to use VB Studio. At a minimum, you should read:

- [Before You Begin](#)
- [Create an OCI Compartment](#)
- [Create the VB Studio Instance](#)

Like other Oracle Cloud services, you need to create an instance of Oracle Visual Builder Studio (VB Studio) before you can start using it. When you sign up for Oracle Cloud Infrastructure, a set of *service limits* is configured for your tenancy (account), which is the quota set for a given resource. (To learn more about service limits, see [Service Limits](#).) The service limit for VB Studio is one instance per region.

VB Studio Generation 2 (Gen 2) refers to VB Studio running natively on Oracle Cloud Generation 2 Infrastructure. [Public Cloud Regions](#) lists the commercial, EU Sovereign, and US/UK Government regions in which Visual Builder Studio Generation 2 is currently available.

Note:

It's important that you determine if you're running on Gen 2 so you know which chapter to read for set-up instructions. Let's do a quick test: Open the OCI console, then open Developer Services. Scan the list for "Visual Builder Studio". If you don't see it, that means you're not running on Gen 2 and should refer to [Set Up Oracle Visual Builder Studio \(Non-Native\)](#) instead of this chapter. If you do see it, then this chapter is the one you need.

Important:

After you finish creating an instance and setting it up, you'll need to set up the build system for CI/CD, by referring to [Set Up VB Studio for CI/CD](#). Then, if you want to set up VB Studio to customize certain Oracle Cloud Applications through application extensions, see [Set Up VB Studio to Extend Oracle Cloud Applications](#) or, if you want to set up VB Studio to develop responsive applications and deploy them to Visual Builder instances, see [Set Up VB Studio for Developing Visual Applications](#).

Before You Begin

Before you do anything, you must make sure that you have the required roles to create a VB Studio instance. Once you have these necessary roles and permissions, you need to open the OCI console and create a compartment for hosting VB Studio's resources: Compute instances

and Storage objects. Your organization's members are going to need these resources to run CI/CD builds.

 **Note:**

You can also run VB Studio on the free build executor (see [VB Studio's Free VM Build Executor](#)), but if you want to add more executor templates to reduce the wait time for your organization's members, create custom executor templates, or use advanced features (such as use your own VCN or use a different VM shape), you should configure VB Studio to connect to your own OCI account.

See [Get the Right Roles](#) and make sure that you have the right roles for creating a compartment and the VB Studio instance that uses that compartment's resources.

Get the Right Roles

If you're the OCI account subscriber, you already have the required roles to set up the VB Studio instance. If you're not the subscriber, make sure that you've been assigned the correct roles in Oracle Identity Cloud Service (IDCS).

You must be assigned this role:	To ...
Cloud Account Administrator or Tenancy Administrator	<p>Set up Oracle Cloud Infrastructure (OCI) compartments, policies, dynamic groups, and permission to create and manage the VB Studio instance.</p> <p>If you're not an account or tenancy administrator, contact the administrator to add you to the OCI_Administrators IDCS group.</p>
Identity Domain Administrator or User Administrator	<p>Add users and assign IDCS roles.</p> <p>If you're not an IDCS domain or user administrator, make sure that your OCI group is assigned a policy like this:</p> <pre>Allow group MyGroup to inspect identity-providers in tenancy</pre>

Create an OCI Compartment

An OCI compartment is where you host your VB Studio CI/CD infrastructure; that is, the compute and storage instances needed for your build executors. You'll actually set this up when you get to [Set Up VB Studio for CI/CD](#); for now, you're just creating the compartment itself.

1. On the OCI console, in the upper-left corner, click **Navigation Menu** .
2. Select **Identity & Security** and then under **Identity**, select **Compartments**.
3. On the Compartments page, create a compartment to host VB Studio resources.
 - a. To create the compartment in the tenancy (root compartment), click **Create Compartment**.
 - b. In the Create Compartment dialog box, fill in the fields and click **Create Compartment**.

To learn more about compartments, see [Managing Compartments](#).

Create the VB Studio Instance

You can create only one VB Studio instance in an Oracle Cloud account. You must create it in your tenancy's root compartment.

 **Note:**

The VB Studio instance is created in the identity domain you're logged into; there is no mechanism to request a different domain. As long as you have the proper user credentials, you should be able to create the instance in any domain you wish. All users who wish to access the new VB Studio instance must be in its same domain.

1. In a web browser, go to <https://cloud.oracle.com>.
To view the list of supported browsers for the OCI Console, see [Supported Browsers](#).
2. In **Cloud Account Name**, enter your Oracle Cloud account or tenancy name and click **Next**.
3. On the Single Sign-On (SSO) panel, if required, select your identity provider and click **Continue**.
4. Enter your Oracle Cloud account credentials and click **Sign In**.
The Oracle Cloud Console, also called the OCI console, opens. If you've recently created your Oracle Cloud account, wait for some time to see your services in the Oracle Cloud Console.
5. In the upper-left corner, click **Navigation Menu** .
6. Select **Developer Services** and then select **Visual Builder Studio**.
7. On the Visual Builder Studio Instances page, from the **Compartment** drop-down list, select the root compartment.
8. Click **Create VB Studio Instance**.
The Instance Name page is displayed.
9. In **Instance Name**, enter the VB Studio's instance name.
10. The **Compartment** field always displays the root compartment you selected in step 7.
This field is read-only. The the root compartment (displayed) is the only compartment allowed.
11. (Optional) Add one or more tags to this resource.
Default tags, which help you identify the instance, always get created.
If you aren't sure whether to add tags, skip this option now. You can always add tags later. For more information about tagging resources, see [Resource Tags](#).
 - a. Click **Show Advanced Options**  to display the Tag section.
 - b. Select the **Tag namespace**, or select **None** to apply a free-form tag.
 - c. Select or enter the **Tag key**.
 - d. In **Tag value**, either enter a value or select one from the list.
 - e. To apply another tag, click **+ Additional Tag**.

- f. Click **Add tag** (or **Add tags**, if you are adding multiple tags).
12. Click **Next**.
13. On the CI/CD Setup page, click the **Yes, I authorize this** check box. This authorizes VB Studio to create and use Compute and Storage instances.

In addition, this option automatically creates an additional policy that authorizes VB Studio to perform the necessary operations in the selected compartment for the CI/CD resources. For this automatic policy creation to succeed, the user who creates the instance must also have permissions to manage policies in the root compartment. See [Get the Right Roles](#).

Typically, tenancy administrators have these permissions, so the easiest way to create a VB Studio instance is to have a tenancy administrator do it. However, if that isn't feasible, it's also possible to create the VB Studio instance without the CI/CD authorization and then set it up later by performing an edit operation.

 **Note:**

If you want to use the built-in free account's free VM build executor , don't select the check box. See [VB Studio's Free VM Build Executor](#) for information about the resources provided with the free VM build executor.

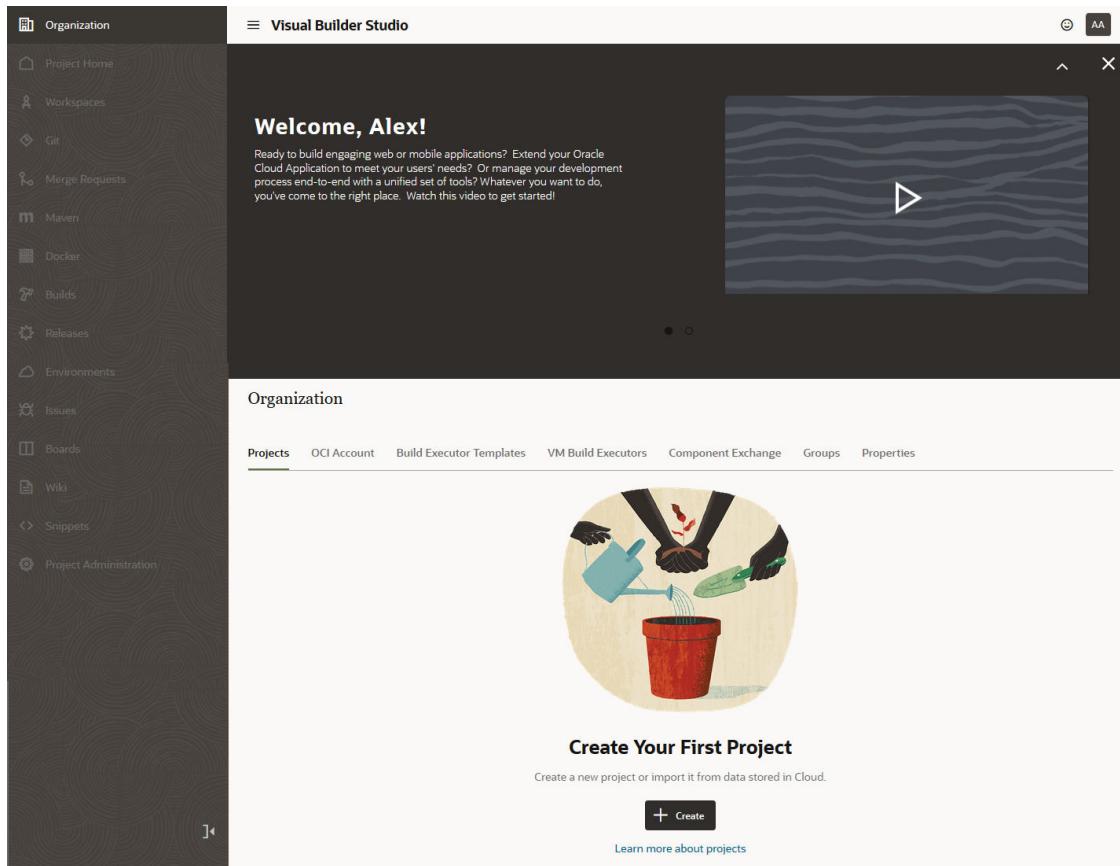
14. If you've selected the **Yes, I authorize this** check box, in **Compartment**, select the compartment you created to host VB Studio's Compute and Storage instances.

When you select the check box, the required policies are applied to the selected compartment.

15. Click the **Create Visual Builder Studio** button.

Once the service instance is created, click **Service Console** to open VB Studio's Organization page.

Here's an example:



What Are the Next Steps?

After you create the instance:

1. You're assigned the DEVELOPER_ADMINISTRATOR IDCS role, which enables you to set up VB Studio, manage the VM build executors, and update the organization details. After you're assigned the role, you become a VB Studio *Organization Administrator*.
2. You'll receive an email with the subject **Verify your Oracle Visual Builder Studio email**. Open the email and click the URL link in the email body to verify your email. If you don't do this, you won't receive important email notifications.
3. After verifying the email address, you'll receive another email with the subject **Welcome to Oracle Visual Builder Studio**. This email contains the VB Studio URL that you can bookmark.
4. You now need to provide access for additional VB Studio users. See [Step 5: Set Up IDCS Users and Groups](#) to add individual users or [Manage Your Organization's Groups](#) to add groups of users.
5. Set up VB Studio's build system for CI/CD, which enables your organization's developers to create jobs to build, package, test, and deploy their apps. See [Set Up VB Studio for CI/CD](#).
6. If your users want to customize Oracle Cloud Applications by creating extensions, see [Set Up VB Studio to Extend Oracle Cloud Applications](#). If they want to develop visual applications and deploy them to Visual Builder runtime instances, see [Set Up VB Studio for Developing Visual Applications](#).

Access VB Studio from the Oracle Cloud Home Page

If you've signed out of the Oracle Cloud or VB Studio, follow these steps to sign back in.

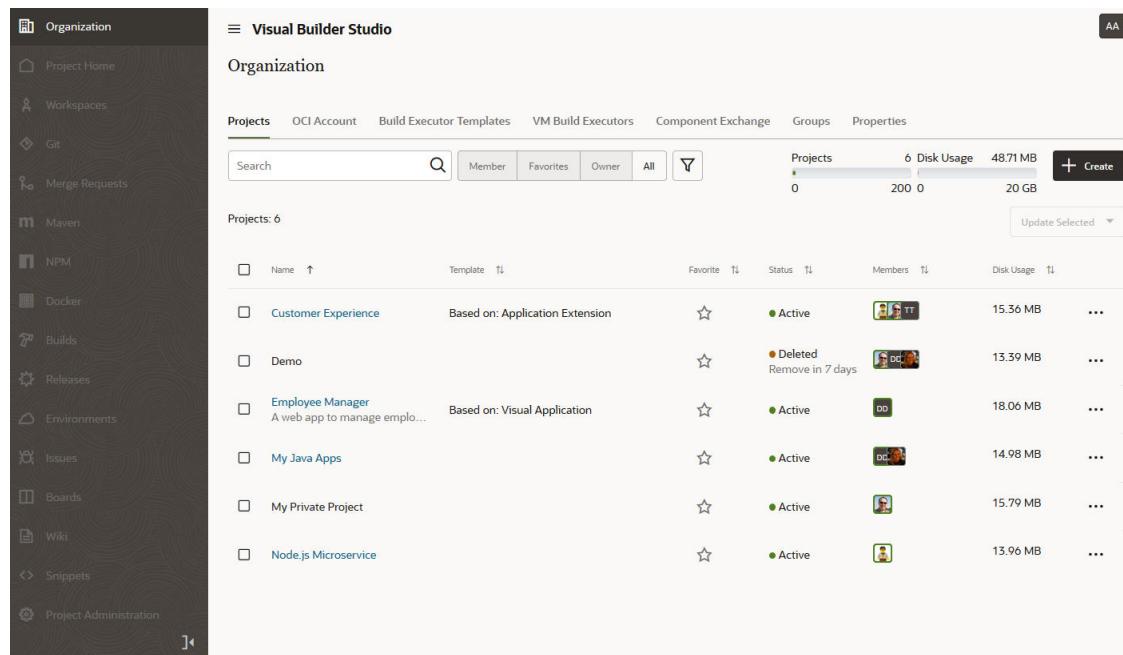
1. In a web browser, go to <https://cloud.oracle.com>.
2. On the Sign-In page, in **Cloud Account Name**, enter your Oracle Cloud account or tenancy name and click **Next**.
3. On the Single Sign-On (SSO) panel, if required, select your identity provider and click **Continue**.
4. Enter your Oracle Cloud account credentials and click **Sign In**.

The Oracle Cloud Console, also called the OCI console, opens. If you've recently created your Oracle Cloud account, wait for some time to see your services in the Oracle Cloud Console.

5. In the upper-left corner, click **Navigation Menu** .
6. Select **Developer Services** and then select **Visual Builder Studio**.
7. On the Visual Builder Studio Instances page, click the VB Studio instance's name.
8. On the Instance Details page, click **Service Console**.

The VB Studio **Organization** page opens, which displays all the projects you're a member of, as well as your favorite projects, the projects you own, and all the shared projects in your organization.

Here's an example of the Organization page with some projects:



The screenshot shows the Oracle Cloud Visual Builder Studio Organization page. The left sidebar contains navigation links for Project Home, Workspaces, Git, Merge Requests, Maven, NPM, Docker, Builds, Releases, Environments, Issues, Boards, Wiki, Snippets, and Project Administration. The main content area is titled "Visual Builder Studio" and "Organization". It features a search bar and filters for Member, Favorites, Owner, All, and a "Create" button. A summary table shows 6 projects with disk usage of 48.71 MB. Below this is a detailed table of projects:

Name	Template	Status	Members	Disk Usage	Actions
Customer Experience	Based on: Application Extension	Active	TT	15.56 MB	...
Demo		Deleted Remove in 7 days	pd	13.39 MB	...
Employee Manager	A web app to manage emplo...	Active	dd	18.06 MB	...
My Java Apps		Active	dd	14.98 MB	...
My Private Project		Active	S	15.79 MB	...
Node.js Microservice		Active	Q	13.96 MB	...

3

Set Up VB Studio for CI/CD

Everyone should read this chapter, regardless of what type of application your organization's members are building. The build system is one of VB Studio's most integral parts.

The build system in VB Studio enables your organization's developers to create jobs that build, package, test, and deploy applications and application extensions. In VB Studio, builds run either in VM build executors (VM executors, for short) or in Docker executors. Build executor templates define the operating system and software packages for VM executors, and Docker images define the operating system and software packages for Docker executors.

To build their applications, your organization's developers define *build jobs* in their VB Studio projects. A *job* is a configuration that defines the location where the application's source files are found, the commands needed to run the job and test scripts, the instructions for generating and packaging artifacts, and the location where those artifacts will be deployed.

To run builds, your organization's developers need some Compute instances with operating systems and software packages installed. For example, to build Node.js applications, they'll need Node.js installed on an OCI VM Compute instance, which they'll use to run to run their CI/CD builds.

Before you start configuring CI/CD for VB Studio, you need to make several decisions that will guide you through the configuration process:

- [Step 1: Decide whether to use the free VM build executor](#)
This step is required. See [VB Studio's Free VM Build Executor](#) to understand what this build executor provides and whether it meets the needs of your organization's members.
- [Step 2: Decide whether to run VM build executors or Docker executors](#)
This step is required. In VB Studio, builds run either in VM build executors or in Docker executors. If you're not sure which type you need, see [What are Build Executors and Build Templates?](#).
- [Step 3: Set up VB Studio to run VM build executors](#)
This step is optional. If you decided to go with VM Build executors in Step 2, use this step. If you decided to use Docker executors, go to Step 4. You must use one or the other.
- [Step 4: Set up VB Studio to run builds on Docker executors](#)
This step is optional. If you decided to go with Docker executors in Step 2, use this step. If you decided to use VM build executors, go to Step 3. You must use one or the other.
- [Step 5: Set Up IDCS Users and Groups](#)
This step is required. If you want to add groups instead of adding users individually, see [Manage Your Organization's Groups](#).
- [Step 6: Configure your OCI resources](#)
This step is optional. You only need to perform this task when you switch OCI compartments or accounts.

Step 1: Decide whether to use the free VM build executor

This step is required.

Decide whether to use the free build executor or an OCI account before you create the VB Studio instance. If you change your mind later, you can switch between the two, but you may have to manually migrate some of the data.

See [VB Studio's Free VM Build Executor](#) to learn more about the capabilities of the free VM build executor.

VB Studio's Free VM Build Executor

The free VM build executor is available in the VB Studio's built-in free account and runs on VB Studio's default VCN.

 **Note:**

The free VM build executor feature is available only for Commercial OC1 realm regions.

The more VM executors you have available, the more builds your organization's members can run at the same time. As the built-in free account offers only one VM executor, your VB Studio organization members can run only one build at a time. If members trigger multiple builds while a build is already running, they must wait until the build running on the VM executor is complete. The VM executor then runs the queued builds on a first-come-first served basis.

The free VM build executor template, called **System Default OL7 for Visual Builder**, includes the necessary software to run extension and visual application builds. You can't change the default executor template's configuration. This table lists the software defined in **System Default OL7 for Visual Builder**:

Software	Version
Oracle Linux operating system	8.x
Node.js	18.x
Oracle Java SE	17.x
Ant	1.9.6
C++ Compiler (cpp/gcc)	4.8.x
Firefox	78.7.x (or later)
Git	2.2.x (or later)
Jq	1.5.x (or later)
JUnit 4	4.11 (or later)
Maven	3.6.x
Python2 (including Virtualenv)	2.7.x
Ruby	2.0.0p648
Xvfb	1.20.x

If you want to add more executor templates to reduce the wait time for your organization's members, to create custom executor templates with different software configurations, or to use advanced features (such as using your own VCN or using a different VM shape), you should configure VB Studio to connect to your own OCI account.

Configure VB Studio to Use the Free VM Executor for Build Jobs and Pipelines

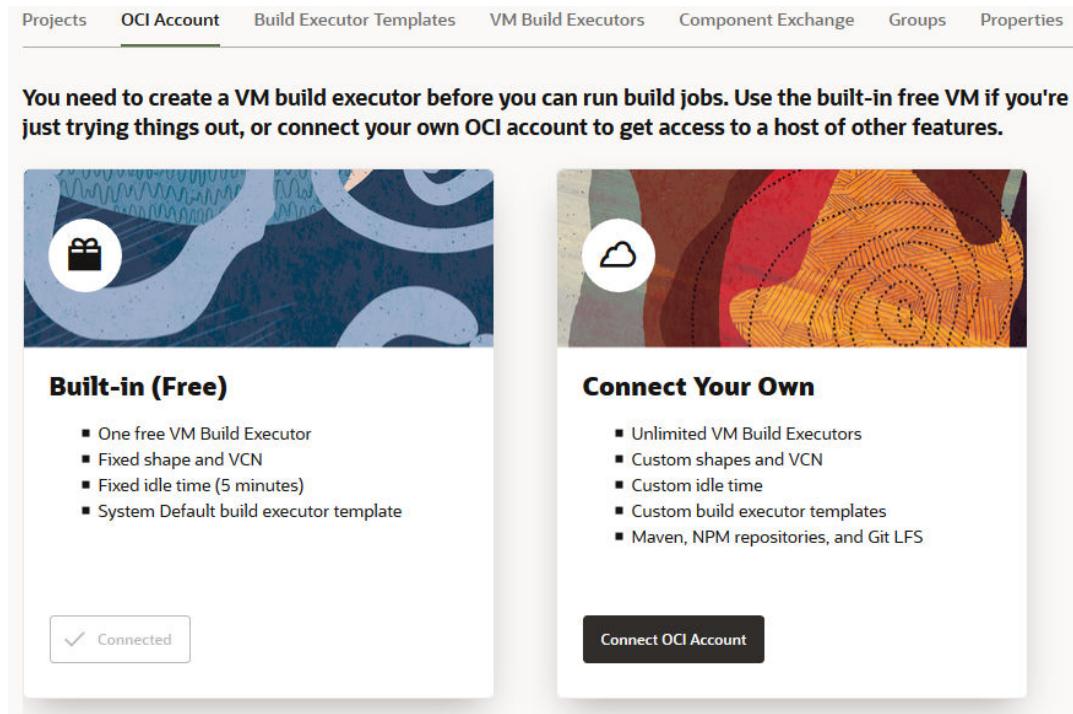
In a VB Studio visual application project, you use build jobs to compile the source code, package the visual application, and deploy it to a Visual Builder instance. The builds and pipelines run on build executors, also called VM build executors. These VM executors are Oracle Cloud Infrastructure (OCI) VM Compute instances dedicated to run VB Studio builds.

In some Oracle Cloud regions and data centers, VB Studio is available pre-configured with a built-in free account, which provides one free VM executor that you can use to run build jobs that package and deploy your extensions. However, there are some limitations (see [VB Studio's Free VM Build Executor](#)) associated with the free VM build executor, so you may want to connect to your own OCI account instead, if you have one. See [OCI Resources in VB Studio](#) for more a more comprehensive comparison between the built-in free account, free tier account, and your OCI account.

To find out whether your VB Studio instance is connected to the built-in free account, follow these steps:

1. In the left navigator, click **Organization** .
2. Click the **OCI Account** tab.

You should see a similar page:



Projects OCI Account Build Executor Templates VM Build Executors Component Exchange Groups Properties

You need to create a VM build executor before you can run build jobs. Use the built-in free VM if you're just trying things out, or connect your own OCI account to get access to a host of other features.



Built-in (Free)

- One free VM Build Executor
- Fixed shape and VCN
- Fixed idle time (5 minutes)
- System Default build executor template

 Connected



Connect Your Own

- Unlimited VM Build Executors
- Custom shapes and VCN
- Custom idle time
- Custom build executor templates
- Maven, NPM repositories, and Git LFS

Connect OCI Account

Depending on your VB Studio's data center, you may or may not see the **Built-in (Free)** option.

What do you see?	What you need to do:
I see the Built-in (Free) option	If you're trying out extensions, no additional configuration is required. Go ahead and create your extension project . VB Studio creates the free VM build executor when you create your first project.

What do you see?	What you need to do:
I see the Built-in (Free) option, but want to run builds without any limitations	Configure VB Studio to connect to your OCI account and add VM executors. If you're new to OCI, see Welcome to Oracle Cloud Infrastructure .
I don't see the Built-in (Free) option, but have access to an OCI account	The built-in free account isn't available in your data center. You should configure VB Studio to connect to your OCI account and add VM executors .
I don't see the Built-in (Free) option and don't have access to an OCI account either	You can still use VB Studio to create the extension project. To run builds, create an OCI account or the Oracle Cloud Free Tier account. The free tier account offers free micro Compute VMs that your organization's members can use to run builds. After creating the free tier account, create OCI resources as described in Set Up the OCI Account and get their details as described in Get the Required OCI Input Values . Then, Set Up the OCI Connection in VB Studio. If you can't create an OCI account or the free tier account, wait for the built-in free account to be available in your data center.

Before you create a project, note that in a VB Studio instance that has no projects and no VM executors, the first VM executor is created for you when you create the first project. If the project isn't the initial one, the VM executor must be created manually.

Add the Free VM Build Executor Manually

If you're using VB Studio's built-in free account, the free VM build executor is added when you create your first project. Here's how to add the VM executor manually if you've deleted it or it wasn't added when you created the project:

1. In the left navigator, click **Organization** .
2. Click the **VM Build Executors** tab.

If you see a VM executor of the **System Default OL7 for Visual Builder** template, the free VM build executor is available in your VB Studio instance. Ignore the remaining steps.

Overview		1 VM	None of your 1 VM build executors are currently running.				None of your VMs are in an error state.			
<input type="text" value="Search"/> 			<input type="button" value="Sleep Timeout"/> 	<input type="button" value="Update Selected"/> 	<input type="button" value="Create Free VM"/> 					
			Template	Region	Shape	VCN OCID	Details	Status	Actions	
<input type="checkbox"/>	System Default OL7 for Visual Builder			us-ashburn-1	VM.Standard.E2.2			Pending		

If you don't see the VM executor, jump to the next step.

3. Click **Create Free VM Build Executor**.

The screenshot shows the 'VM Build Executors' tab selected in the navigation bar. Below it, two main options are presented:

- Free VM Build Executor:** This section features a windmill icon and a list of attributes:
 - Fixed shape and VCN
 - Fixed idle time (5 minutes)
 - System Default Build Executor Template
 A red-bordered button labeled 'Create Free VM Build Executor' is at the bottom.
- Custom VM Build Executor:** This section features a cloud icon and a list of benefits:
 - Unlimited VM Build Executors
 - Custom shapes and VCN
 - Custom idle time
 - Custom Build Executor Templates
 - Maven, NPM repositories, and Git LFS
 A black button labeled 'Connect OCI Account' is at the bottom.

Step 2: Decide whether to run VM build executors or Docker executors

This step is required.

In VB Studio, builds run either in *VM build executors* (VM executors, for short) or in Docker executors. Before your developers create jobs and run builds, you must decide whether to run your organization's builds on VM executors or on Docker executors, and then create the required resources.

If you already know what you want to do, skip ahead to either [Step 3: Set up VB Studio to run VM build executors](#) or [Step 4: Set up VB Studio to run builds on Docker executors](#).

If you're not sure which one to use, see [VM Build Executors](#) or [Docker Build Executors and Docker Images](#).

What are Build Executors and Build Templates?

Builds in a build system run in *build executors*. The operating system and software packages required to run builds on build executors are defined in *build templates*. In VB Studio, builds run either in VM build executors (VM executors, for short) or in Docker executors.

VM Build Executors

VM build executors are OCI VM Compute instances dedicated to run builds of *jobs*, which your organization's members define in VB Studio projects.

A VM executor is always associated with a build executor template. When your organization's members create jobs, they simply associate the appropriate executor template with the job. When the job's build triggers, the VM executor that's associated with the executor template

starts automatically. Oracle charges you only when a VM executor is active, runs a build, or is preparing itself to run a build.

This table describes the different states of a VM executor:

State	What does it mean?	Does it cost?
Pending	After you add a VM executor, it is in this state until it runs a build. When a VM executor starts from this state, it takes some time to install the operating system and software packages.	No
Starting	The VM executor is starting. If the VM executor starts from the Pending state, VB Studio installs the operating system and software packages on the VM executor's assigned boot volume. This takes time. If the VM executor starts from the Stopped state, VB Studio uses software packages and the operating system from the previous run's saved boot volume. VB Studio periodically checks all executor templates for updates. If an executor template is found with new updates, VB Studio deletes the preserved boot volume of all stopped VM executors that reference the executor template and changes their status from Stopped to Pending .	Yes
Available	Operating system and software packages are installed, and the VM executor is ready to run a build.	Yes
In Use	The VM executor is running a build. After the running build is complete, the VM executor returns to the Available state.	Yes
Stopping	The VM executor is shutting down. Before shutting the VM executor down, VB Studio saves the operating system and software packages to the VM executor's assigned boot volume.	Yes
Stopped	The VM executor has shut down.	No
Error	There's a hardware or a software issue on the VM executor. Check the VM executor's log to find more about the cause.	No

State	What does it mean?	Does it cost?
Destroying	The VM executor is being deleted.	No
Error Unrecoverable	<p>This state is most likely caused when a customer changes the Compute account's OCI access so that control of the VM executor OCI resources are blocked.</p> <p>This state can also occur if the build executor is in an Error status and VB Studio is not able to remove all used OCI resources or if there is a temporary network glitch during the removal process.</p> <p>VB Studio tries once a day to clear OCI resources used by the VM executor in the Error Unrecoverable state. To manually clear these resources, you can also use the Try to Reset action.</p>	No

Some key points to remember about VM executors:

- After creating a VB Studio instance, VB Studio creates a VM executor when you create your first project (assuming you don't already have one). The VM executor is associated with the **System Default OL7 for Visual Builder** executor template in the connected OCI account.
- When you add a VM executor manually, you must specify the executor template, choose an OCI region from the connected OCI account's subscribed regions, specify the OCI Compute VM's shape, and select the VCN (optional).
- Your OCI account may have some Compute instance limits set. When you add a VM executor, VB Studio looks into the specified OCI region's Availability Domains, finds available OCPUs with the specified shape, calculates the number of Compute instances, and displays the number of Compute VM instances you can add from your OCI account's set limit.

Here's an example of the VB Studio's Add VM Build Executors dialog box that displays the number of VM executors you can add with the VM.Standard.E2.1 shape:

Add VM Build Executor

Quantity ▼ ▲

Build Executor Template

Region

Shape ▼

VCN Selection VMs allowed by this shape: 103

Virtual Cloud Network (VCN) is a software-defined network set up in Oracle Cloud Infrastructure data centers. Choose **Default** to use the Visual Builder Studio compartment's VCN, or **Custom** to use another compartment's VCN.

Default

Custom

Cancel Add

- When you add a VM executor, it is added in the **Pending** state and doesn't cost you anything. You can add more VM executors than the number of available Compute VM instances.
Remember, VB Studio creates a Compute VM instance when a VM executor starts, not when you add it.
- You can add standard and legacy VM shapes with these series:
 - VM.Standard1
 - VM.Standard2
 - VM.Standard.E2
 - VM.Standard3.Flex
 - VM.Standard.E3.Flex
 - VM.Standard.E4.Flex
 - VM.Standard.E5.Flex
 - VM.Standard.B1
 - VM.Standard.Intel.Generic
 - VM.Standard.x86.Generic

- VM.Standard.AMD.Generic

For more details about the above shapes, see [Standard shapes](#) and [Legacy shapes](#).

- A VM executor can run one build at a time.
- When a job's build runs, if VB Studio finds multiple VM executors allocated for the job's executor template, it runs the build on any one of them. You can't choose or specify a particular VM executor to use for the build.
- If you expect your organization's members to run parallel builds of jobs that refer to a common executor template, add multiple VM executors for that executor template. If you're not sure, you can start with one VM executor and add more VM executors later.
- While adding multiple VM executors that reference a common executor template:
 - Add all VM executors in the same VCN. If you add VM executors with a common executor template in different VCNs (such as some VM executors in the default VCN and other VM executors in a custom VCN), your builds might behave unpredictably.
 - Add all VM executors with the same shape. If you add VM executors with different shapes (such as some VM executors of the VM.Standard1.1 shape and some of the VM.Standard2.8 shape), your builds may run slow or fast depending on the VM executor it runs on.
- After a build is complete, a VM executor continues to be in the **Available** state and waits for some time for any queued builds. This wait time is called sleep timeout. If no builds run on the VM executors in this duration, VB Studio automatically stops the VM executors.
- The more VM executors you have running at a specific time, the higher the cost. To minimize the higher cost, configure the sleep timeout to stop inactive VM executors after some time. The sleep timeout setting applies to all your organization's VM executors.

Build Executor Templates

A build executor template defines the operating system and the software packages your organization's members need to run builds on VM executors.

VB Studio offers some executor templates out-of-the-box, but you can create more templates if the default templates don't meet your requirements. See [Create and Manage Build Executor Templates](#).

Some key points to remember about executor templates:

- Each executor template contains an Oracle Linux operating system, Java, and some required software packages. If required, you can install more software packages available in the [VB Studio Software Catalog](#).
- You can't add a software package or a version that's not available in the VB Studio Software Catalog.
- In the Software Catalog, some software packages have multiple versions. You can add only one version to the executor template. However, you can add multiple versions of Java to an executor template.
- Executor templates don't consume storage space or increase your cost. You can create as many executor templates as you require.
- Some software packages, such as Fn and Oracle Jet, require other software packages in the same executor template. When you add these software packages, VB Studio prompts you to add the dependent software packages.

- Don't add unnecessary software packages to an executor template. The more software packages you add to an executor template, the more storage space its VM executors consume when they run builds.

What Happens When a Build Runs in a Build Executor?

When VB Studio runs a build, it follows a set order to select a VM executor. You can't choose or specify a particular VM executor to use for the build. If no VM build executors are found, VB Studio fails the build.

When multiple VM executors are found, VB Studio selects a VM executor to run the build in this order:

- If a VM executor is in the **Available** state, VB Studio runs a build on it.
- If no VM executors are in the **Available** state, VB Studio starts a **Stopped** VM executor, installs the operating system and the executor template's software packages from the saved boot volume, and then runs the build on it.
- If no VM executors are in the **Available** or the **Stopped** state, VB Studio starts a **Pending** VM executor, installs the operating system and the executor template's software packages, and then runs the build on it.
- If all VM executors are running builds, VB Studio waits for a VM executor to complete its build, and then runs a build on it.

Note:

A VM executor in the **Stopped** or **Pending** state will take several minutes to start because VB Studio installs the operating system and the executor template's software packages before running the build.

The VM executor should start subsequent builds more quickly. You can adjust the sleep timeout to avoid start-up delays.

The steps for a build running in a build executor include:

1. VB Studio checks the job's build executor template and then finds a VM executor allocated to it.
2. VB Studio checks the job's configuration and runs the commands in the specified order.
3. After the build is complete, VB Studio copies any generated artifacts to the configured OCI Object Storage bucket.
4. The VM executor waits for some time for any queued builds. If no builds run during the wait time period, the VM executor stops.

Before stopping the VM executor, VB Studio saves the operating system and software packages to the VM executor's assigned boot volume.

Docker Build Executors and Docker Images

A *Docker image* defines the operating system and software packages your organization's members need to run builds on a *Docker executor*. You can either import a Docker image from an external Docker registry, such as DockerHub, or create it from a build executor template.

Unlike a VM executor, a Docker executor is not directly associated with any specific VM. When your organization's members create jobs, they simply associate a Docker image as a build

template with the job. When the job's build triggers, VB Studio runs the build on any *Docker deployment VM*.

Some key points to remember about Docker images:

- When you create an image from a build executor template, VB Studio creates a new Docker image by installing the software configured in that template. When you create an image from a registry, VB Studio pulls the image from the registry, adds a build agent, and creates a new image.
- You can specify the maximum number of executors that can be created from the image.
- You can create as many Docker images as you require, as long as the Management VM has enough space.

Docker Deployment VMs and Docker Executors

Docker executors run in a Docker deployment VM. Like a VM build executor, a Docker deployment VM is an OCI VM Compute instance dedicated to run builds.

To run builds in Docker executors, you add Docker deployment VMs. A Docker executor can run only one build at a time, but a Docker deployment VM can run multiple builds simultaneously, depending on how many Docker executors are configured for the deployment VM. The maximum number of Docker executors that can be run by a given Docker deployment VM is configurable between 1 and 10.

When a job's build is triggered, VB Studio starts a Docker deployment VM if it is in a stopped state, creates a Docker executor in it, and builds the job using the executor.

Remember, a Docker deployment VM costs you only when it's in a running state and actively building jobs. When the Docker deployment VM is idle and not building jobs, it'll be put to sleep. You can configure the number of minutes the deployment VM remains idle before it is put to sleep.

This table describes the Docker deployment VM states:

State	What does it mean?	Does it cost?
Pending	After you add a Docker deployment VM, it is in this state until the management VM finishes building, and then it will start automatically.	No
Starting	The Docker deployment VM is starting. If the Docker deployment VM starts from the Pending state, VB Studio installs the operating system and software packages on the Docker deployment VM's assigned boot volume. This takes time. If the Docker deployment VM starts from the Stopped state, VB Studio uses software packages and the operating system from the previous run's saved boot volume.	Yes

State	What does it mean?	Does it cost?
Ready	The operating system and Docker agent are installed and started. The VM is ready to deploy Docker executors.	Yes
Stopping	The Docker deployment VM is shutting down. Before shutting the Docker deployment VM down, VB Studio saves the operating system and software packages to the Docker deployment VM's assigned boot volume.	Yes
Stopped	The Docker deployment VM has shut down.	No
Error	There's a hardware or a software issue on the Docker deployment VM. Check the Docker deployment VM's log to find more about the cause.	No

Some key points to remember about Docker deployment VMs and Docker executors:

- Once created, the Docker Management VM is always running and costs you 24-7. To keep your costs lower, if you are planning to run fewer builds with fewer images, select a smaller shape with fewer OCPUs.
- To run builds in Docker executors, you add Docker deployment VMs and specify the number of Docker executors it can run, select the OCI region from the connected OCI account's subscribed regions, the OCI Compute VM's shape, and optionally the VCN.
- Your OCI account may have some Compute instance limits set. When you add a Docker deployment VM, VB Studio looks into the specified OCI region's Availability Domains, finds available OCPUs with the specified shape, calculates the number of Compute instances, and displays the number of Compute VM instances you can add from your OCI account's set limit.
- When you add your first Docker deployment VM, VB Studio adds one more VM, called the Docker Management VM, to manage Docker images.
- The Docker Management VM is responsible for creating Docker images and Docker executors and for deploying images to Docker executors.
- When you create your first Docker deployment VM, the Docker Management VM is also created with the same shape.
- VB Studio provides a default shape selection for the management VM. This default is only a recommendation. Some things to consider:
 - You could still choose a lower-end configuration for your specific purposes, but the default is what VB Studio determines to be the optimal powered VM for the management VM. Although a low-end VM (such as Standard 1.1 with 1 CPU and 8 GB of memory) could be used for Docker Image management in scenarios that have few Executor templates and create just a few Docker images, it is better to use the default recommendation. If you try to select a lower-powered VM than what is recommended, you'll see a warning.
 - For scenarios that have several Executor templates and will create several Docker images, using a high-end VM (such as VM.Standard.E3.Flex with 4 OCPU and 32 GB

RAM) instead is recommended. Using a low-end VM in demanding scenarios will severely hamper overall performance.

- If a Docker deployment VM is not in use, it will eventually time out and enter the Stopped state, which has no associated cost to you.
- The higher the number of Docker deployment VMs running Docker executors at any specific time, the higher the associated cost. To minimize the higher cost, you should configure the sleep timeout to stop inactive Docker deployment VMs after some period of time. The default timeout is 300 minutes. The sleep timeout setting will be applied to all your organization's Docker deployment VMs.

What Happens When a Build Runs on a Docker Executor?

1. VB Studio checks the Docker image that corresponds with the template configured in the job and finds whether a build of the image has run before.
If this is the Docker image's first build, it finds a Docker deployment VM to run the image in an executor, and then it downloads the image from the Docker Management VM if the image doesn't already exist in the VM.

If one or more Docker deployment VMs are found, VB Studio selects a Docker deployment VM to run the build in this order:

- a. If a Docker deployment VM is in the **Ready** state and has at least one Docker executor that's not running, VB Studio runs the build on it. A Docker deployment VM is full if VB Studio has already deployed the maximum number of executors on that VM.
- b. If no Docker deployment VMs are in the **Ready** state, VB Studio starts a **Stopped** Docker deployment VM and runs the build on one of its Docker executors.
- c. If all Docker executors of all Docker deployment VMs are running builds, VB Studio waits for a Docker executor to complete its build, and then runs a build on it.

If this is not the Docker image's first build, VB Studio finds the Docker deployment VM that ran the image's Docker executor earlier and uses it to run the build again, unless the Docker deployment VM is full. If all Docker executors of the Docker deployment VM are busy running builds, VB Studio selects another Docker deployment VM to run the build in the above described order.

2. VB Studio checks the job's configuration and runs the commands in the specified order.
3. After the build is complete, VB Studio copies any generated artifacts to the configured OCI Object Storage bucket.
4. The Docker deployment VM waits for some time for any queued builds. If no builds run during the wait time period, the Docker deployment VM stops.
Before stopping the Docker deployment VM, VB Studio saves the Docker image to the Docker deployment VM's assigned boot volume.

Step 3: Set up VB Studio to run VM build executors

This step is optional.

If you decided to go with VM build executors in Step 2, use this step. If you decided to use Docker executors, go to [Step 4: Set up VB Studio to run builds on Docker executors](#). You must use one or the other.

To run VM build executors, you'll need to create build executor templates and then add VM executors.

Create and Manage Build Executor Templates

You can create and manage build executor templates from the Organization Administration page's **Build Executor Templates** tab.

 **Note:**

You can't create or manage executor templates if VB Studio is connected to the built-in free account. If you want to use software that isn't available in the default executor templates, configure VB Studio to connect to your OCI account. See [Change Your OCI Account](#).

VB Studio offers these executor templates out-of-the-box for your organization to use. You can't modify or delete these executor templates.

Build executor template	Description
System Default OL7 for Visual Builder	Use this template to package and deploy extensions and visual applications. The template defines these software packages on Oracle Linux 8: <ul style="list-style-type: none">• Node.js 18• Default software packages
System Default OL8	Use this template for build jobs that use the default software packages on Oracle Linux 8.
System Default OL7	Use this template for build jobs that use the default software packages on Oracle Linux 7.

If none of these meet your requirements, you can create as many executor templates as you need. Executor templates don't consume storage space or increase your cost.

You can create and manage executor templates from the **Build Executor Templates** tab on the Organization page:

1. In the left navigator, click **Organization** .
2. Click the **Build Executor Templates** tab.

This table describes the actions you can perform to create and manage executor templates.

Action	How To
Create a custom executor template	<p>1. Click + Create Template.</p> <p>2. In the New Build Executor Template dialog box, enter a name and description of the executor template. In Platform, select the operating system to run on the VM executor.</p> <p>3. Click Create.</p> <p>4. Click Configure Software.</p> <p>5. If necessary, in Filter, enter the search term and click Search . To see the latest version of the software package, select the Show latest versions only check box. This is helpful if multiple versions of the same software are available in the catalog.</p> <p>6. In the software catalog, select the software package tile's check box to add it. The tile is highlighted in green and the software package adds to the Selected Software list on the right. To remove a selected tile, deselect the check box or click Remove this software  under Selected Software. You can't remove the required software packages that are highlighted in grey. If a software package depends on another software package or if there's a conflict between software packages, a message under Selected Software informs you about it.</p> <p>7. Click Done.</p>
Configure an executor template's software	<p>1. From the executor template list on the left, select the executor template and click Configure Software.</p> <p>2. In the software catalog, to search a software package, enter the search term in Filter and click Search . To see the latest version of the software package, select the Show latest versions only check box. This is helpful if multiple versions of the same software are available in the catalog. To add a software, select the software package tile's check box. The tile is highlighted in green and the software package adds to the Selected Software list on the right. To remove a software package, deselect its check box or click Remove this software  under Selected Software. You can't remove the required software packages that are highlighted in grey. If a software package depends on another software package or if there's a conflict between software packages, a message under Selected Software informs you about it.</p> <p>3. Click Done.</p>
Edit an executor template's name or description	<p>1. From the executor template list on the left, select the template you want to modify and click the Edit button on the far right side of the panel.</p> <p>2. In the Edit Build Executor Template dialog box, update the name or description.</p> <p>3. Click Save.</p>

Action	How To
Delete an executor template	<p>When you delete an executor template, its VM executors are deleted too.</p> <ol style="list-style-type: none">From the executor template list on the left, select the template you want to delete.Either click the Delete  icon next to it in the list or click the Delete button to the right in the template's panel.In the Delete Build Executor Template dialog box, click Yes to confirm.

Add and Manage VM Build Executors

A VM executor can run one build at a time. When you add a VM build executor, you allocate an OCI VM Compute instance to run VB Studio builds. If you expect your organization's members to run builds in parallel jobs that refer to a common executor template, you should add multiple VM executors for that executor template. Note that the more VM executors you have running at a specific time, the higher the cost. To minimize the higher cost, use the Sleep Timeout setting to automatically shut down inactive VM executors.

 **Note:**

You can't add or manage VM executors if VB Studio is connected to the built-in free account. If you want to add more than one VM executor, configure VB Studio to connect to your OCI account.

Here's an example of multiple VM executors that use a common executor template.

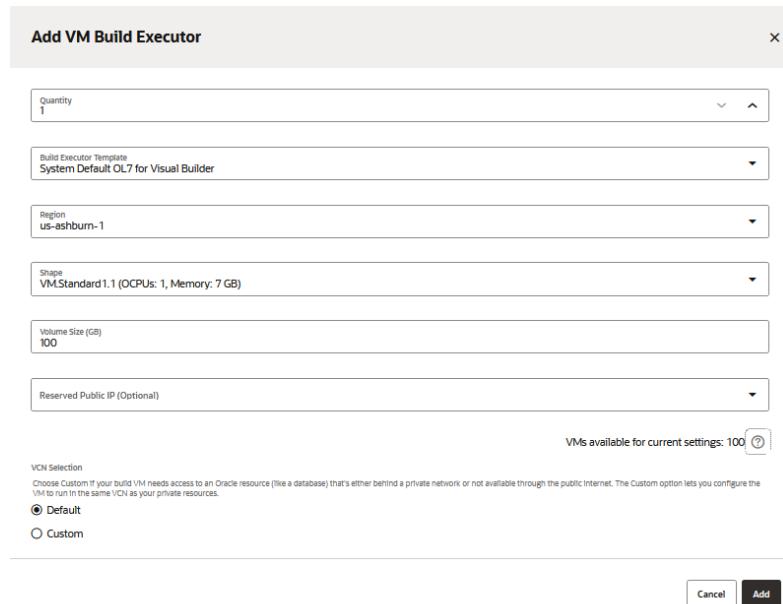
You add and manage VM executors from the **Build Executors** tab on the **Organization** page:

1. In the left navigator, click **Organization** 
2. Click the **Build Executors** tab.

This table describes the actions you can perform to manage VM executors.

Action	How To
Add the free VM build executor in the built-in free account	<p>You can add and use the free VM build executor only if your VB Studio instance is connected to the built-in free account.</p> <p>The free VM build executor will be automatically created upon creation of the first project in the org. You can also add it by clicking Create Free VM Build Executor on the Build Executors tab.</p> <p>VB Studio creates the free VM build executor that uses the System Default OL7 for Visual Builder template. You can't change the VM executor's template.</p> <p>The new VM executor is in the Pending state until you manually start it or trigger a job's build that references the associated executor template.</p>
Find VM executors of an executor template	In the search box, enter the executor template's name.
Sort VM executors	Click the arrow icon in the column's header to sort VM executors. For example, to sort VM executors by their state, click the arrow icon in the Status column's header.

Action	How To
Add a VM executor in the VB Studio's default VCN	<p>1. Find VM executors of the executor template you want to use. If there are no VM executors of the executor template, jump to the next step.</p> <p>If you find one or more VM executors that use the executor template, get each VM executor's VCN name. If the VCN's name isn't vbs-executor-vcn, click *** and select Delete to delete the VM executor. You should not add VM executors to different VCNs that use the same executor template.</p> <p>2. Click + Create VM.</p> <p>3. In the Add VM Build Executor dialog box, in Quantity, specify the number of VM executors you want to create. To minimize build execution delays, set the number to the number of jobs that you expect to run in parallel using the template you'll specify in Build Executor Template. If you're not sure, start with one VM executor. You can add more VM executors later, based on your actual usage.</p> <p>4. In Build Executor Template, select the executor template.</p> <p>5. In Region, select the region that's closer to you geographically. The drop-down list displays regions your OCI account is subscribed to.</p> <p>6. In Shape, select the VM executor's shape. For example:</p>



Notice the message below the **Reserved Public IP (Optional)** field. Click the question mark icon [image] to see how the number of available VMs was calculated for the current settings. Here's an example of what you might see.

Action	How To				
VMs available for current settings: 100 					
Availability Domain	Max by OCPUs	Max by Volume	Max by Memory	Max VMs this AD	
ZQib:US-ASHBURN-AD-1	25	31	N/A	25	
ZQib:US-ASHBURN-AD-2	41	159	N/A	41	
ZQib:US-ASHBURN-AD-3	34	284	N/A	34	
Total					100

The display shows the realtime statistics that were used to make this calculation, using your settings. As you change your settings, new statistics are computed and displayed. See [What Determines How Many VMs I Can Create?](#) for more information about how the calculations are made.

If the required number of Compute VM instances aren't available, choose another shape.

7. Select the **Reserved Public IP** that you want to add to the VM executor. You can add one Reserved Public IP to each VM executor. An allow listing for your VM executor is possible only if you specify an OCI account by assigning a Reserved Public IP to a build executor. The Reserved Public IP must be defined in the compartment of the specified OCI account.
8. In **VCN Selection**, select **Default**.
9. Click **Add**.

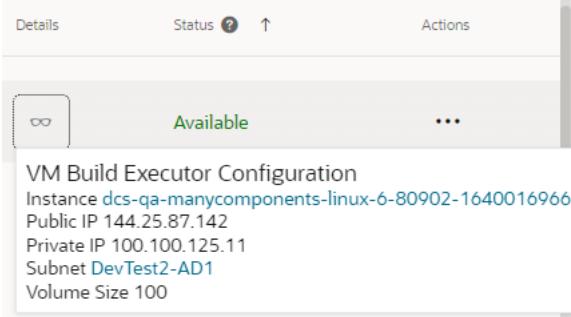
The new VM executor is in the Pending state until you manually start it or trigger a job's build that references the associated executor template.

Action	How To
Add a VM executor in another compartment's VCN	<p>Before you add a VM executor, make sure you've added a public subnet in the VCN. See Create and Configure a Public Subnet in a VCN.</p> <ol style="list-style-type: none"> Find VM executors of the executor template you want to use. If there are no VM executors of the executor template, jump to the next step. If you find one or more VM executors that use the executor template, get each VM executor's VCN name. If the VCN's name is different from your VCN, click *** and select Delete to delete the VM executor. You should not add VM executors to different VCNs that use the same executor template. Click + Create VM. In the Add VM Build Executor dialog box, in Quantity, specify the number of VM executors you want to allocate. In Build Executor Template, select the executor template. In Region, specify the VM executor's region. In Shape, select the VM executor's shape. For example:

The screenshot shows the 'Add VM Build Executor' dialog box. It has fields for Quantity (set to 1), Build Executor Template (System Default OL7 for Visual Builder), Region (us-ashburn-1), Shape (VM.Standard1.1 (OCpus: 1, Memory: 7 GB)), Volume Size (GB) (set to 100), and Reserved Public IP (Optional). A note at the bottom right says 'VMs available for current settings: 100' with a question mark icon.

Notice the message below the **Reserved Public IP (Optional)** field. Click the question mark icon [image] to see how the number of available VMs was calculated for the current settings. Here's an example of what you might see.

The screenshot shows a table with columns: Availability Domain, Max by OCpus, Max by Volume, Max by Memory, and Max VMs this AD. Data rows include ZQib:US-ASHBURN-AD-1 (25, 31, N/A, 25), ZQib:US-ASHBURN-AD-2 (41, 159, N/A, 41), ZQib:US-ASHBURN-AD-3 (34, 284, N/A, 34), and a Total row (0, 0, 0, 100). A note at the top right says 'VMs available for current settings: 100' with a question mark icon.

Action	How To
	<p>The display shows the realtime statistics that were used to make this calculation, using your settings. As you change your settings, new statistics are computed and displayed. See What Determines How Many VMs I Can Create? for more information about how the calculations are made.</p> <p>If the required number of Compute VM instances aren't available, choose another shape.</p>
	<p>7. Select the Reserved Public IP that you want to add to the VM executor. You can add one Reserved Public IP to each VM executor. An allow listing for your VM executor is possible only if you specify an OCI account by assigning a Reserved Public IP to a build executor. The Reserved Public IP must be defined in the compartment of the specified OCI account.</p> <p>8. In VCN Selection, select Custom.</p> <p>9. In VCN Compartment, select the compartment. If you're an Oracle Cloud OS Management Service (OSMS) user, don't select the OSMS compartment or a compartment with an OSMS policy.</p> <p>10. In VCN, select the VCN.</p> <p>11. In Subnets Compartments, select the compartments where your public subnets are. By default, it adds your VCN's compartment. If required, you can add more compartments.</p> <p>12. In Subnets, select a subnet. The list shows public subnets only. You can add multiple public subnets. If VB Studio can't create a VM executor on the first subnet you've added, it tries to create it on the second subnet, and so on.</p> <p>13. Click Validate Network Setup.</p> <p>14. After successful validation, click Add.</p>
	<p>The new VM executor is in the Pending state until you manually start it or trigger a job's build that references the associated executor template.</p>
Get the name and IP addresses of a VM executor's VCN subnet.	<p>1. In the VM executor's Details column, click Show machine details .</p> <p>2. Note the VCN subnet's name.</p>  <p>The VM executor's subnet name is displayed only if it's in the Available state. If the VM executor is in the Stopped or the Pending state, click ... and select Start to start the VM executor.</p>

Action	How To
View a VM executor's log	The VM executor's log has entries for all events along with information about when the events occurred, the type of event, and event details. 1. Select the VM executor, click Actions *** and select Display Log . 2. In the Provisioning Log window, review the log. To download the log file to your computer, click Download Log .
Start or stop a VM executor manually	When a build of a job triggers, its VM executor starts automatically if it was in the stopped state. It takes some time to start a VM executor, and the user must wait for a VM executor to start before the job's build runs on it. Similarly, a VM executor stops automatically if no builds run on it during the sleep timeout period. At times, you may want to manually start a VM executor before triggering a job's build or stop it to free resources immediately. To start or stop a VM executor, click Actions *** and select Start or Stop . To start or stop multiple VM executors, select their check boxes, click Update Selected and select Start selected VMs or Stop selected VMs .
Delete a VM executor	To delete a VM executor, click Actions*** and select Delete . If the Delete action doesn't delete the VM executor, you can force a delete using the Force Delete action. To delete multiple VM executors, select their check boxes, click Update Selected and select Delete selected VMs .
Change the sleep timeout of all VM executors	Set the sleepout time to stop inactive VM executors automatically. By default, it is 30 minutes. The higher the sleep timeout value, higher will be your cost as inactive VM executors continue to be in the Available state for a longer time. If you don't expect your organization's users to run builds frequently, specify a lower sleep timeout. When a VM executor starts from the Stopped state, it installs the operating system and all software packages of the executor template. This takes time and your organization's users must wait for the build to start until the VM executor is in the Available state. If you expect your organization's users to run builds frequently, specify a higher sleep timeout value. 1. Click Sleep Timeout . 2. In the Sleep Timeout dialog box, change the timeout duration to define how long should a VM executor be in the Available state if no builds run on it. 3. Click Save .
Reset the VM executor	Use the Try to Reset action if the build executor is stuck in the Error Recoverable status. The Try to Reset action will try to remove all used OCI resources of the VM executor. To reset a VM executor, click Actions *** and select Try to reset .

After adding VM executors, you and your organization's members can configure jobs to use executor templates.

What Determines How Many VMs I Can Create?

There are three factors that determine how many VMs you can create:

- **OCPUs** - The maximum number of VMs varies according to the shape selected

- **Memory** - The maximum number of VMs is based on the Flex shape selected and the Allocated Memory setting. (This restriction is used for Flex shapes only.)
- **Volume** - The maximum number of VMs is based on the Volume Size setting.

After you specify a volume size (in GB), the **Add VM Build Executor** dialog displays real time numbers for OCPU and Memory, just below the line that tells you how many VMs are available for you to use. These numbers come directly from the OCI Console.

You might notice that **Quantity** isn't on the list. The quantity of VMs that you actually create has no impact on the limits. The limits represent the maximum numbers of what you could *use* but VB Studio doesn't restrict what you can *create*. This is actually beneficial, since availability constantly changes. Instead, the runtime simply limits what you can use in real time.

Connect to Your OCI Account and Add VM Build Executors

You may want to configure VB Studio to connect to your own OCI account if you need more VM build executors to reduce the wait time for your organization's members, you want to create custom build executor templates, or you want to use advanced features for VM executors (such as use your own VCN or use a different VM shape).

1. Set up your OCI account and get the required input values. If you don't have authorization to create and manage OCI resources, ask some one who can create the resources and share their details.
See [Set Up the OCI Account](#) and [Get the Required OCI Input Values](#).
2. In the left navigator, click **Organization** .
3. Click **Connect OCI Account**.
4. Enter the required details and click **Validate**.
5. After successful validation, click **Save**.

To create custom executor templates, see [Create and Manage Build Executor Templates](#). Remember to add Node.js 18 (or a higher version) to the executor template. Node.js 18 is the minimum version required for packaging extensions.

To add more VM executors:

1. Click the **VM Build Executors** tab.
2. Click **+ Create VM**.
3. In the Add VM Build Executor dialog box, in **Quantity**, specify the number of VM executors you want to allocate.

To minimize build execution delays, set the number of VM executors to the number of jobs that you expect to run in parallel using that template. If the OCI VM quota is available, that number of VM executors will be added to the **Virtual Machines** tab.

If you're not sure about the number of VM executors you'll need, start with one VM executor and then add more as required. Note that the more VM executors you have running at a specific time, the higher the cost of OCPUs. To minimize the higher cost, use the Sleep Timeout setting on the **VM Build Executors** tab to automatically shut down inactive VM executors. You can always return to the **VM Build Executors** tab to remove or add VM executors, based on your actual usage.

4. In **Build Executor Template**, select **System Default OL7 for Visual Builder** or a executor template with Node.js 18 (or higher) software.
5. In **Region, Shape**, and **VCN Selection** specify the VM's region, shape, and Virtual Cloud Network (VCN).

- A region is a localized geographic area where the data centers are located. Remember, a VM executor is a VM on OCI Compute. Choose the region where your VB Studio account is or the one that's closest to you geographically.
- A shape is a template that determines the number of CPUs, amount of memory, and other resources allocated to the created instance. Choose a shape of your preference.
- A VCN is a software-defined network that you set up in the Oracle Cloud Infrastructure data centers in a particular region. By default, builds run in your VB Studio's compartment. To run builds in your own VCN, select **Custom** and specify its details.

To learn more about regions and shapes, see [Regions and Availability Domains](#) and [Compute Shapes](#). To find more about VCNs, see [VCNs and Subnets](#).

6. Click **Add**.

Run VM Build Executors in Virtual Cloud Network

A Virtual Cloud Network (VCN) is a software-defined network that you set up in the OCI data centers of a region. A subnet is a subdivision of a VCN. A VM build executor always runs in a VCN's public subnet.

You can choose to run a VM executor in the VB Studio compartment's VCN or in another compartment's VCN. Run the VM executor in the VB Studio compartment's VCN if you don't have a VCN or want to use the default option without any additional configuration. Run the VM executor in another compartment's VCN if you want the VM executor to access Oracle Cloud services running in that compartment's VCN.

 **Note:**

You can't run VM executors in a VCN if VB Studio is connected to the built-in free account. To use another compartment's VCN, configure VB Studio to connect to an OCI account.

Use VB Studio's Default VCN

VB Studio compartment's VCN, also called the default VCN, is automatically created for you when the first VM build executor that uses it starts.

VB Studio's default VCN is called `vbs-executor-vcn` and resides in the VB Studio's compartment. When a VM executor that uses the default VCN starts, VB Studio checks the OCI compartment for the default VCN. If it doesn't exist, VB Studio creates a VCN called `vbs-executor-vcn` with CIDR block `10.0.0.0/16` and public subnets in all availability domains. If the VCN exists, VB Studio uses it to run your VM executors.

When VB Studio creates the default VCN, it also creates these components and adds them to the VCN:

- An Internet Gateway
- A Route Table that uses the Internet Gateway as the routing rule
- Security Rule Ingress rules that allow TCP traffic on:
 - Destination port 22 (SSH), 9003 (Executor agent debug), 9005 (VM agent debug), 9082 (Executor agent), and 9085 (VM agent), and 8095 (Docker Agent), and 9001-9010 from source 0.0.0.0/0 and any source port.

- Destination port 443 from sources in the 10.0.0.0/16 IP range.
- A Security Rule that allows Egress to any destination from any protocol.
- Three public subnets, one for each availability domain. Their CIDR is set to 10.0.0.0/24, 10.0.1.0/24, and 10.0.2.0/24.

Here's an example of the default VCN:

vbs-executor-vcn

Move Resource Add Tags Terminate

VCN Information Tags

Compartment: VBStudioCompartment **OCID:** ...l7jzfq [Show](#) [Copy](#)

Created: Tue, Sep 15, 2020, 18:23:34 UTC **DNS Resolver:** [vbs-executor-vcn](#)

CIDR Block: 10.0.0.0/16 **Default Route Table:** [Default Route Table for vbs-executor-vcn](#)

DNS Domain Name: DNS isn't enabled for this VCN

Subnets *in* VBStudioCompartment Compartment

Create Subnet				
Name	State	CIDR Block	Subnet Access	Created
vbs-executor-subnet-icfc-US-ASHBURN-AD-3	Available	10.0.2.0/24	Public (icfc:US-ASHBURN-AD-3)	Tue, Sep 15, 2020, 18:23:37 UTC
vbs-executor-subnet-icfc-US-ASHBURN-AD-2	Available	10.0.1.0/24	Public (icfc:US-ASHBURN-AD-2)	Tue, Sep 15, 2020, 18:23:36 UTC
vbs-executor-subnet-icfc-US-ASHBURN-AD-1	Available	10.0.0.0/24	Public (icfc:US-ASHBURN-AD-1)	Tue, Sep 15, 2020, 18:23:36 UTC

Showing 3 Items < 1 of 1 >

As soon as the default VCN is available, you have full control over it and can modify it. You can add private subnets for your private services, add more public subnets or delete the existing subnets, modify security lists, and add or remove other components.

 Note:

- When a VM executor runs on the default VCN, it runs on any of its available public subnets. You can't specify which subnet it should run on.
- If you plan to remove some public subnets of the default VCN, make sure that at least one public subnet is available in the VCN. If there are no public subnets, VM executors in the default VCN won't run and your builds will fail.
- The default VCN is created once and continues to stay until it is deleted manually.
- If your organization's members configure jobs that access Oracle Cloud services in the private or public subnets of the VCN, ask them to configure their jobs to access the services using private IPs or Fully Qualified Domain Name (FQDN).

Run VM Build Executors in Another Compartment's VCN and Subnets

To allow a VM build executor access your Oracle Cloud services in a compartment's VCN, you should configure the VM executors to run in the same VCN. This allows the VM executor to access Oracle Cloud services easily without any complex networking configuration.

Before you configure the VCN, make a note of these:

- A VM executor always runs in a public subnet.
- In the VCN, you must create a public subnet or configure an existing public subnet to allow inbound access from and outbound access to VB Studio. See [Create and Configure a Public Subnet in a VCN](#).
- Make sure that the public subnet is regional.
- Instead of modifying an existing security list's security rules, create a new security list for the public subnet.
For the public subnet, create a security list and add ingress rules from source CIDR 0.0.0.0/0 for VB Studio ports 22 (SSH), 9082 (Executor Agent), and 9085 (VM Agent). This is required to allow VB Studio access the VM executors in the VCN.
- For the subnet's compartment, assign the `use virtual-network-family` OCI policy to the user whose OCID you specified when you set up the OCI connection in VB Studio. This is required for networking permissions and builds to run in the VCN's subnet. This statement assigns the policy to the user's group:
`allow group <group-name> to use virtual-network-family in compartment <subnet-compartment-name>`

Here's an example of the `use virtual-network-family` policy added to the policies you created in [Set Up the OCI Account](#).

VBStudioPolicies

[Edit Policy](#) [Add Tags](#) [Delete](#)

Policy Information

Tags

OCID: ...ji6zjy2a [Show](#) [Copy](#)

Version Date: Keep version current

Compartment: myaccount (root)

Description: Policies for the VB Studio Group

Created: Tue, Jun 16, 2020, 08:32:08 UTC

Statements

[Edit Policy Statements](#)

allow group VBStudioGroup to manage all-resources in compartment MyCompartm

allow group VBStudioGroup to read all-resources in tenancy

allow group VBStudioGroup to use virtual-network-family in compartment MyServicesCompartm

- Make sure that the VCN has a route table with a rule that allows Internet access.
- To allow the VM executor to access the VCN's private subnet's services and resources, configure the private subnet's security rules to allow incoming traffic from the public subnet used by the VM executor.
- While adding a VM executor, you can specify multiple public subnets. If VB Studio can't create the VM executor on the first specified public subnet, it tries to create it in the second subnet, and so on.
- After configuring a VM executor to run in another compartment's VCN, ask your organization's members to configure their build jobs to use the private IP addresses or the Fully Qualified Domain Name (FQDN) of services that are running in the VCN. Tell them not to use public IP addresses, because when VM executors are in the same VCN as the service, public IP addresses will route the traffic outside the VCN, causing builds to fail.

This table describes what you need to do if you have a VCN.

If ...	Then :
You have a VCN without a public subnet	<ol style="list-style-type: none"> 1. Create and Configure a Public Subnet in a VCN. You'll also configure the subnet's security list to allow inbound access from and outbound access to VB Studio. 2. To allow a VM executor access a service running in the private subnet, configure the private subnet's security list. See Allow VM Build Executors to Access a Private Subnet's Resources. 3. Add and Manage VM Build Executors in the VCN.
You have a VCN with a public subnet	<ol style="list-style-type: none"> 1. In your VCN, create a security list with ingress rules and egress rules as described in steps 6-11 in Create and Configure a Public Subnet in a VCN. 2. Open your public subnet's details page and add the security list. See step 14 in Create and Configure a Public Subnet in a VCN. 3. To allow a VM executor access to services running in the private subnet, configure the private subnet's security list. See Allow VM Build Executors to Access a Private Subnet's Resources. 4. Add and Manage VM Build Executors in the VCN.
You don't have a VCN and want to create one	<ol style="list-style-type: none"> 1. Use the VCN wizard to create a VCN with a public subnet and an internet gateway. See Virtual Networking Quickstart. 2. Create and Configure a Public Subnet in a VCN. You'll also configure the subnet's security list to allow inbound access from and outbound access to VB Studio. 3. To allow a VM executor access to services running in the private subnet, configure the private subnet's security list. See Allow VM Build Executors to Access a Private Subnet's Resources. 4. Add and Manage VM Build Executors in the VCN.

Create and Configure a Public Subnet in a VCN

Before you can run VMs in another compartment's VCN, you must first create a public subnet in your VCN with security rules that allow inbound access from and outbound access to VB Studio.

1. Sign in to Oracle Cloud Console.
2. In the upper-left corner, click **Navigation Menu** .

3. Select **Networking** and select **Virtual Cloud Networks**.
4. Under **List Scope**, select the compartment.
5. From the VCNs list, click the VCN's name.
6. Under **Resources**, click **Security Lists**, and then click **Create Security List**.
7. In **Name**, enter a name for the security list.
8. In **Create in Compartment**, ensure that the correct compartment is selected.
9. In **Allow Rules for Ingress**, click **+ Another Ingress Rule** and follow these steps:
 - a. In **Source Type**, select **CIDR**.
 - b. In **Source CIDR**, enter **0.0.0.0/0**.
 - c. In **Destination Port Range**, enter **9082**.
 - d. (Optional) In **Description**, add a description.

Here's an example:

Create Security List Help Cancel

A security list contains ingress and egress rules that specify the types of traffic allowed in and out of instances. [Learn more about Security Lists](#)

NAME

Public Subnet Security List for VB Studio Build VMs

CREATE IN COMPARTMENT

MyCompartment ▼
myaccount (root)/MyCompartment

Allow Rules for Ingress

Ingress Rule 1 X

Allows TCP traffic 9082

STATELESS (i)

SOURCE TYPE	SOURCE CIDR	IP PROTOCOL (i)
CIDR ▼	0.0.0.0/0 Specified IP addresses: 0.0.0.0-255.255.255.255 (4,294,967,296 IP addresses)	TCP ▼

SOURCE PORT RANGE <small>OPTIONAL</small> (i)	DESTINATION PORT RANGE <small>OPTIONAL</small> (i)
All Examples: 80, 20-22	9082 Examples: 80, 20-22

DESCRIPTION <small>OPTIONAL</small>
TCP Traffic for ports: 9082 Maximum 255 characters

+ Another Ingress Rule

- e. Click **+ Another Ingress Rule** and repeat steps 9a through 9d to add ports **9085** and **22**.
- f. (Only if you are using Docker executors) Click **+ Another Ingress Rule** and repeat steps 9a through 9d to add ports **8095** and **9001-9010** to **0.0.0.0/0** and add port **443** to your VCN CIDR (for example **10.0.0.0/16**).
10. In **Allow Rules for Egress**, click **+ Another Egress Rule** and follow these steps:

- a. In **Source Type**, select **CIDR**.
- b. In **Source CIDR**, enter **0.0.0.0/0**.
- c. In **IP Protocol**, select **All Protocols**.
- d. (Optional) In **Description**, add a description.

11. Click **Create Security List**.

After creating the security list, click its name to verify the ingress and egress rules you added.

Here's an example of ingress rules:

Ingress Rules

	<input type="checkbox"/> Stateless	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows
	<input type="checkbox"/>	No 0.0.0.0/0	TCP	All	22		TCP traffic for ports: 22
	<input type="checkbox"/>	No 0.0.0.0/0	TCP	All	9082		TCP traffic for ports: 9082
	<input type="checkbox"/>	No 0.0.0.0/0	TCP	All	9085		TCP traffic for ports: 9085

Here's an example of the egress rule:

Egress Rules

	<input type="checkbox"/> Stateless	Destination	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows	Description
	<input type="checkbox"/>	No 0.0.0.0/0	All Protocols				All traffic for all ports	:>
0 Selected								
Showing 1 item < Page 1 >								

12. Return to the VCN's details page.

13. Under **Resources**, select **Subnets** and follow these steps to create a public subnet:

If you want to edit an existing public subnet, jump to the next step.

- a. Click **Create Subnet**.
- b. In **Name**, enter the subnet's name.
- c. In **Create in Compartment**, select the correct compartment.
- d. In **Subnet Type**, make sure that **Regional** is selected.
- e. In **CIDR Block**, enter the subnet's CIDR block.
Don't set it to **172.17.0.0/16** as it's the default subnet allocated to Docker.
- f. In **Route Table**, select the VCN's route table.
- g. In **Subnet Access**, make sure that **Public Subnet** is selected.
- h. In **DHCP Options**, select the VCN's DHCP options.
- i. In **Security List**, select the security list you created in step 6.

- j. Fill in the other fields as required.

Here's an example:

Create Subnet

NAME
Public Subnet for VB Studio Build VMs

CREATE IN COMPARTMENT
VBStudioCompartment
devcstest (root)/VBStudioCompartment

SUBNET TYPE

Regional (Recommended)
Instances in the subnet can be created in any availability domain in the region. Useful for high availability.

Availability Domain-specific
Instances in the subnet can only be created in one availability domain in the region.

CIDR Block

CIDR BLOCK
10.0.4.0/24
Specified IP addresses: 10.0.6.0-10.0.6.255 (256 IP addresses)

ROUTE TABLE COMPARTMENT IN DEMOCOMPARTMENT ([CHANGE COMPARTMENT](#))
Route Table for Private Subnet-MyVBStudioVCN

SUBNET ACCESS

Private Subnet
Prohibit public IP addresses for Instances in this Subnet

Public Subnet
Allow public IP addresses for Instances in this Subnet

DNS RESOLUTION
 USE DNS HOSTNAMES IN THIS SUBNET (i)
Allows assignment of DNS hostname when launching an instance

DNS LABEL
PublicSubnetfor
Only letters and numbers, starting with a letter. 15 characters max.

DNS DOMAIN NAME READ-ONLY
<dns-label>.myvbstudiovcn.oraclevcn.com

DHCP OPTIONS COMPARTMENT IN DEMOCOMPARTMENT ([CHANGE COMPARTMENT](#))
Default DHCP Options for MyVBStudioVCN

Security Lists

You can associate up to 5 network security lists with the subnet.

SECURITY LIST COMPARTMENT IN DEMOCOMPARTMENT ([CHANGE COMPARTMENT](#))
Public Subnet Security List for VB Studio Build VMs

+ Another Security List

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values that can be attached to resources.
[Learn more about tagging](#)

TAG NAMESPACE	TAG KEY	VALUE
None (add a free-form tag)		

+ Additional Tag

Create Subnet [Cancel](#)

- k. Click **Create Subnet**.

14. If you want to edit an existing subnet, follow these steps:

- a. Under **Resources**, select **Subnets** and click the public subnet's name.
- b. Click **Add Security List**.
- c. In the Add Security List dialog box, in **Security List**, select the security list you created in step 6.
- d. Click **Add Security List**.

That's it. After creating or editing the public subnet, your VM executors can now run in the VCN.

Allow VM Build Executors to Access a Private Subnet's Resources

After adding a public subnet in a VCN, to allow VM build executors access the resources and services (such as Java Cloud Service or a VM-based Database) running in the VCN's private subnet, configure the private subnet's security rules to allow incoming traffic from the public subnet used by VM executors.

For example, to allow VM executors access Java Cloud Service running in a private subnet, configure the subnet's security list to add the VM executors CIDR ranges to the Ingress rule associated with the JCS Admin port.

1. Sign in to Oracle Cloud Console.
2. In the upper-left corner, click **Navigation Menu** .
3. Select **Networking** and select **Virtual Cloud Networks**.
4. On the Virtual Cloud Networks page, click the VCN.
5. Under **Resources**, click **Security Lists**, and then click the private subnet's security list.
6. Click **Add Ingress Rules**.

If you want to modify an existing rule, click the Actions icon (three dots), and then select **Edit**.

7. In **Source Type**, select **CIDR**.
8. In **Source CIDR**, enter the VM executor's public subnet's CIDR range.
9. In **Destination Port Range**, enter the service's port number.
10. (Optional) In **Description**, add a description.

Here's an example of Java Cloud Service port 7002 with a source CIDR of 10.0.4.0/24:

The screenshot shows the 'Add Ingress Rules' dialog box. At the top, it says 'Ingress Rule 1' and 'Allows TCP traffic 7002'. Below this, there are fields for 'SOURCE TYPE' (CIDR), 'SOURCE CIDR' (10.0.4.0/24), 'IP PROTOCOL' (TCP), 'SOURCE PORT RANGE' (All), 'DESTINATION PORT RANGE' (7002), and a 'DESCRIPTION' field ('TCP Port for Java Cloud Service'). At the bottom, there are 'Add Ingress Rules' and 'Cancel' buttons, and a '+ Another Ingress Rule' link.

- 11.** Click **Add Ingress Rules**.
- 12.** If required, repeat steps 6-11 for each service's port.

VB Studio Public IP Addresses in OCI Data Centers

If you're running VM executors in another compartment's subnet and don't want to use the CIDR 0.0.0.0/0 range in your ingress and egress rules, specify VB Studio's public IP address in the subnet's ingress and egress rules.

For a list of public IP addresses, see [Public IP Addresses for VB Studio](#).

You can also use the following list of IP addresses to restrict outbound access from your build VM executors:

Server type	IP address
Storage	129.150.1.1/32
	129.150.1.2/32
	134.70.80.3/32
	134.70.84.3/32
	130.162.1.2/32
	130.162.1.1/32
Public yum	23.212.73.81/32
Shared services yum	130.35.147.212/32

Server type	IP address
Yum regional mirrors	192.29.194.161/32 129.149.81.204/32 129.149.51.216/32 192.29.166.112/32 155.248.131.128/32 129.149.16.241/32 129.148.209.47/32 138.1.66.178/32 192.29.138.86/32 129.148.132.80/32 147.154.1.235/32 192.29.25.15/32 192.29.113.113/32 129.149.65.253/32 192.29.242.73/32 138.1.16.168/32 129.149.114.199/32 192.29.222.78/32 129.149.98.107/32 129.149.122.166/32 192.29.38.186/32 147.154.123.93/32 129.149.39.7/32 129.148.179.162/32 129.148.160.57/32 192.29.153.28/32 129.149.3.154/32 129.148.144.226/32 192.29.82.26/32 192.29.69.154/32 192.29.181.241/32
NPM (registry.npmjs.org)	104.16.16.35/32 104.16.17.35/32 104.16.18.35/32 104.16.19.35/32 104.16.20.35/32 104.16.21.35/32 104.16.22.35/32 104.16.23.35/32 104.16.24.35/32 104.16.25.35/32 104.16.26.35/32 104.16.27.35/32
Node.js (nodejs.org)	104.20.22.46/32 104.20.23.46/32

Server type	IP address
Docker - OL8 only (download.docker.com)	108.156.91.110/32
	108.156.91.29/32
	108.156.91.41/32
	108.156.91.7/32

Software for Build Executor Templates

VB Studio provides various software packages in the build executors' software catalog. Some software packages are available by default in each executor template.

Platforms

These platforms are available:

- Oracle Linux 8
- Oracle Linux 7 (default)

Default Software Packages

These software packages are available by default in each executor template. You can't edit or remove these software packages from an executor template.

Software	Version in Oracle Linux 8	Version in Oracle Linux 7
Oracle Java SE	17.x	17.x
Ant	1.10.5 (or later)	1.9.15
C++ Compiler (cpp/gcc)	8.5.0 (or later)	4.8.5 (or later)
Git	2.39.3 (or later)	2.31.1
Git LFS	3.2.0	2.10.0
Jq	1.6 (or later)	1.5 (or later)
Maven	3.6.3	3.6.3
Python2 (including Virtualenv)	2.7.18 (or later)	2.7.5 (or later)

Software Packages in the Software Catalog

Here's a list of the software available in the VB Studio's software catalog. You can add only one version to an executor template, even if multiple versions of the software are available.

Software	Version in Oracle Linux 8	Version in Oracle Linux 7	Notes
Docker	26.1.1 (or later)	19.03.11 (or later)	
Firefox	ESR 115.5.0 (or later)	ESR 115.5.0 (or later)	
Fn	NA	0.6.33	Requires Docker, OCICli, and Python3 on 3.7 or later.
GraalVM EE for Java 8	21.2.0	21.2.0	For Java 1.8_301 (or later), version 21.2.0 is available.
Gradle	7.5.1	7.5.1	
Groovy	4.0.4	4.0.4	
Helm	3.9.3	3.9.3	

Software	Version in Oracle Linux 8	Version in Oracle Linux 7	Notes
Java SE	22.0.1 21.0.3 17.0.11 11.0.23 1.8.0_411	22.0.1 21.0.3 17.0.11 11.0.23 1.8.0_411	<p> Note:</p> <p>Java 17.x is automatically included in all new Build Executor templates (instead of Java 1.8.x) and is automatically added to any existing templates.</p> <p>Templates created before VB Studio 22.10.0 include Java 1.8.x, but they've been updated to also include Java 17.x.</p>
JUnit 4	4.12 (or later)	4.11 (or later)	Users can add multiple versions of Java to a build executor template by selecting the Java version to use in a job from the job's configuration page.
Kubectl	1.24.3	1.24.3	
Node.js grunt <i>pre-install</i>	grunt v1.6.1 gruntcli v1.4.3	grunt v1.6.1 gruntcli v1.4.3	
Node.js	20.8.0 18.18.1 17.9.1 16.17.0 14.20.0	17.9.1 16.17.0 14.20.0	
Node.js Driver for Oracle Database	6.2.0 (or later)	6.2.0 (or later)	
OCICli	3.37.1 (or later)	3.37.1 (or later)	Requires Python3.

Software	Version in Oracle Linux 8	Version in Oracle Linux 7	Notes
Oracle Developer Studio 12c 12.5	12.5	12.5	
Oracle Forms Developer	12.2.1.4.0	12.2.1.4.0	
Oracle Instant Client 12c	21.12.0.0.0 (or later) 19.15.0	21.12.0.0.0 (or later) 19.15.0	
Oracle JDeveloper Studio	12.2.1.4.0	12.2.1.4.0	
Oracle JET Command-line Interface	16.0.0 (or later)	16.0.0 (or later)	Requires Node.js
Oracle SOA Suite 12	12.2.1.4.0	12.2.1.4.0	
Podman	4.6.1 (or later)	N/A	

Note:

Podman will not work if Docker is being used for build executors in your organization.

PSMcli	1.1.28	1.1.28	Requires Python3.
Python3	3.10.6 3.9.13 3.8.13 3.7.13	3.10.6 3.9.13 3.8.13 3.7.13	To invoke virtualization for your environment, run this command: <code>python3 -m venv <env_name></code>
Ruby	2.5.9p229 (or later)	2.0.0p648 (or later)	
SQLcl	24.2.0.0 (or later)	24.2.0.0 (or later)	Requires Java 11 (or later)
Xvfb	1.20.11 (or later)	1.20.4 (or later)	

Step 4: Set up VB Studio to run builds on Docker executors

This step is optional.

If you decided to go with Docker executors in Step 2, use this step. If you decided to use VM build executors, go to [Step 3: Set up VB Studio to run VM build executors](#). You must use one or the other.

In VB Studio, you can run builds on Docker executors. You may want to use Docker executors for your execution platform if you have several templates and frequent jobs. Docker requires one management VM to run full-time, as well as one or more deployment VMs that can switch between templates as needed by different builds. Another reason you may want to use Docker

executors instead of VM executors is if you want to include custom Docker images as the base for build executor templates.

You can import Docker images from your private or public Docker registry or create them from build executor templates, then deploy these images on VB Studio's Docker deployment VMs.

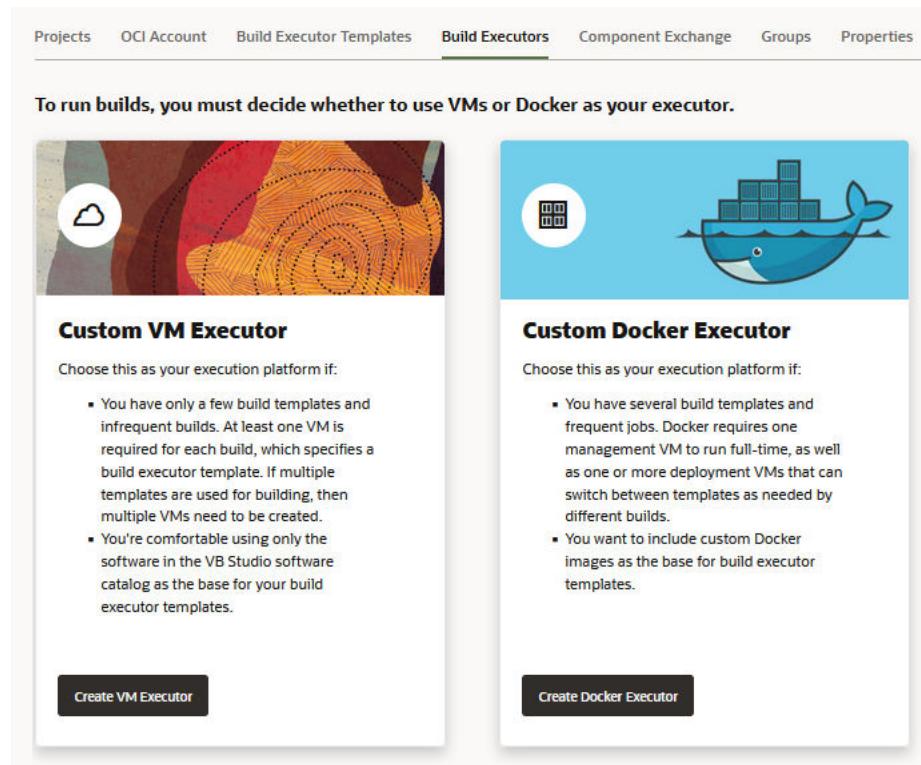
Add Your First Docker Deployment VM

If you haven't configured VB Studio to add VM executors, you can add your first Docker deployment VM from the Organization page's Build Executors tab.

Note:

If you have created a VCN for your VM executors, make sure that you have the VCN configured to support Docker executors. For instructions, see [Use VB Studio's Default VCN](#) or [Create and Configure a Public Subnet in a VCN](#) for a custom VCN.

1. Configure VB Studio to connect to your OCI account.
2. In the left navigator, click **Organization** .
3. Click the **Build Executors** tab.
4. In the **Custom Docker Executor** tile, click **Create Docker Executor**.



5. Add details to the Add VMs for Creating Docker Executors dialog:
 - a. **Number of VMs**: Specify the number of Docker deployment VMs you want to create.

The number you specify doesn't include the Docker Management VM. For example, if you specify 4 in **Number of VMs**, VB Studio will create five VMs: four Docker deployment VMs and one Docker Management VM.

- b. **Executors created per VM:** Enter the maximum number of Docker executors you want to deploy in each Docker deployment VM.

- c. **Region:** Select the region that's closer to you geographically.

The drop-down list displays regions your OCI account is subscribed to.

- d. **Shape:** Select the Docker executor's shape.

As this is your first Docker deployment VM, the Docker Management VM is also created with the same shape. Choose a shape with at least two OCPUs and 16 GB of RAM.

After selecting the shape, wait for a few seconds. VB Studio calculates the number of Compute VM instances that can be created with the selected shape and displays it in the dialog box. If the required number of Compute VM instances aren't available, choose another shape.

- e. **Volume:** Specify the storage capacity for each Docker deployment VM.

- f. **Number of OCPUs:** If you've selected a Flex shape, specify the number of OCPUs to add in each Docker deployment VM.

- g. **Amount of Memory:** If you've selected a Flex shape, specify each Docker deployment VM's memory.

- h. **Use only for project(s):** (Optional) Select one or more projects to assign to this Docker deployment VM. After the Docker deployment VM is created, it will be restricted to running builds for the assigned projects. Builds for these projects will only run on this VM, and builds from other projects won't run on this VM.

6. For **VCN Select**, choose **Default** or **Custom**.

- If you chose **Default**, click **Create**. You are finished with this task.
- If you chose **Custom**, continue to the next step.

7. Fill out the additional details for the custom VCN:

- a. **VCN Compartment:** Select the compartment.

If you're an Oracle Cloud OS Management Service (OSMS) user, don't select the OSMS compartment or a compartment with an OSMS policy.

- b. **VCN:** Select the VCN.

- c. **Subnets Compartments:** Select the compartments where your public subnets are. By default, it adds your VCN's compartment. If required, you can add more compartments.

- d. **Subnets:** Select a subnet. The list shows public subnets only.

You can add multiple public subnets. If VB Studio can't create a Docker executor on the first subnet you've added, it tries to create it on the second subnet, and so on.

8. Click **Validate Network Setup**.

9. Click **Create**.

Before you try to use your Docker executors and images, you should check their states by visiting the tabs on the Organization page's **Build Executors** tab:

- The **Docker Images** tab shows the number of Docker images that are in use or are in an error state.

Docker Images VM Pool Active Executors (0) Auto Refresh

4 Images

None of your images are in use.
There are 2 images in an error state.

Name Type Size Active Images Status Actions

<input type="checkbox"/> DCS QA Req Comp Linux 7	Template	3.49 GB	0 of 4 allowed	Pending	...
<input type="checkbox"/> DCS QA Req Comp Linux 8	Template	2.08 GB	0 of 4 allowed	Pending	...
<input type="checkbox"/> System Base OL7	System	1.88 GB		Error	...
<input type="checkbox"/> System Base OL8	System	2.08 GB		Error	...

- The **VM Pool** tab shows how many of your VMs are running or are in an error state.

Docker Images **VM Pool** Active Executors (0) Auto Refresh

3 VMs

One of your VMs is running.
None of your VMs are in an error state.

VM Id Type Restricted Details Active Executors Disk Usage Status Actions

<input type="checkbox"/> 961	Management		10%	Ready	...
<input type="checkbox"/> 962	Deployment		0%	Stopped	...
<input type="checkbox"/> 963	Deployment		0%	Stopped	...

- The **Active Executors** tab shows how many Docker build executors are currently active.

Docker Images VM Pool **Active Executors (0)** Auto Refresh

Job Name Project Name Docker Image VM Id Started Status Actions

No Docker build executors are currently active

Migrate to Docker

If you've created one or more VM executors and want to use Docker executors to run builds, you can migrate those VM executors to Docker.

You can choose to run builds for your organization on either VM executors or on Docker executors; you can't choose both. If the builds in your organization are already running on VM executors, you can migrate to Docker, and all builds in your organization going forward will use Docker executors.

When you perform a Docker migration, all of your VM executors are destroyed and recreated in Docker as images, separate from deployment VMs. If you want to revert back to VM executors, you will perform a reset that destroys all the Docker images and deployment VMs. You will then have to recreate your VM executors manually, so be sure to capture the details of your current VM executors before performing a migration in case you need to reset and go back.

Note:

If you have VM executors that specify different custom VCNs, you *must* capture the details of all of the custom VCNs *before* migrating to Docker. The migration dialog will only let you choose one VCN, and you will need to recreate the deployment VMs with the details of your other custom VCNs.

1. If you're using the free VM build executor or are connected to the OCI free tier account, configure VB Studio to connect to your OCI account.
2. In the left navigator, click **Organization** .
3. Click the **Build Executors** tab.
4. Capture details for your current VM executors in case you want to revert back from Docker executors to VM executors.
 - a. To capture ID, template, and shape information, copy the contents of the **Build Executors** table and paste to a text file.
 - b. To capture the contents of the **Details** column, click **Show machine details**  for each VM executor, copy the **VM Build Executor Configuration** details, and paste to a text file.
5. Click **Migrate to Docker**.
6. In the Do you want us to re-create your VM executors as Docker images dialog, enter the required details.
 - a. In **Number of VMs**, specify the number of Docker deployment VMs you want to create. The number you specify doesn't include the Docker Management VM. For example, if you specify 4 in **Number of VMs**, VB Studio will create five VMs: four Docker deployment VMs and one Docker Management VM.
 - b. In **Executors created per VM**, enter the number of Docker executors you want in each Docker deployment VM.
 - c. In **Region**, select the region that's closer to you geographically. The drop-down list displays regions your OCI account is subscribed to.

- d. In **Shape**, select the Docker deployment VM's shape.

As this is your first Docker deployment VM, the Docker Management VM is also created with the same shape. For the image management VM's optimum performance, select a shape with at least 2 OCPUs and 16 GB memory.

Wait for a few seconds. VB Studio calculates the number of Compute VM instances that can be created with the selected shape and displays it in the dialog box. If the required number of Compute VM instances aren't available, choose another shape.

- e. In **Volume**, specify the storage capacity for each Docker deployment VM.
- f. If you've selected a Flex shape, then in **Number of OCPUs**, specify the number of OCPUs to add in each Docker deployment VM.
- g. If you've selected a Flex shape, then in **Amount of Memory**, specify each Docker deployment VM's memory.

7. For **VCN Select**, choose **Default** or **Custom**.

- If you chose **Default**, click **Migrate**. You are finished with this task.
- If you chose **Custom**, continue to the next step.

8. Fill out the additional details for the custom VCN:

- a. **VCN Compartment**: Select the compartment.

If you're an Oracle Cloud OS Management Service (OSMS) user, don't select the OSMS compartment or a compartment with an OSMS policy.

- b. **VCN**: Select the VCN.

- c. **Subnets Compartments**: Select the compartments where your public subnets are. By default, it adds your VCN's compartment. If required, you can add more compartments.

- d. **Subnets**: Select a subnet. The list shows public subnets only.

You can add multiple public subnets. If VB Studio can't create a Docker executor on the first subnet you've added, it tries to create it on the second subnet, and so on.

9. Click **Validate Network Setup**.

10. Click **Migrate**.

Create and Manage Docker Images

In VB Studio, you can import a Docker image from an external Docker registry or create a Docker image from a build executor template.

You'll need to configure VB Studio to connect to your OCI account before you can proceed. See [Set Up the OCI Connection](#) in VB Studio.

 **Note:**

You can't create or manage Docker images if VB Studio is connected to the built-in free account.

VB Studio provides out-of-the-box base Docker images for your organization's users. You can't modify or delete these Docker images, and they can't be used for building your job. These base images are used to create Docker images from Docker build executor templates.

Docker image	Description
System Base OL7	Oracle Linux 7 base image with required software packages
System Base OL8	Oracle Linux 8 base image with required software packages

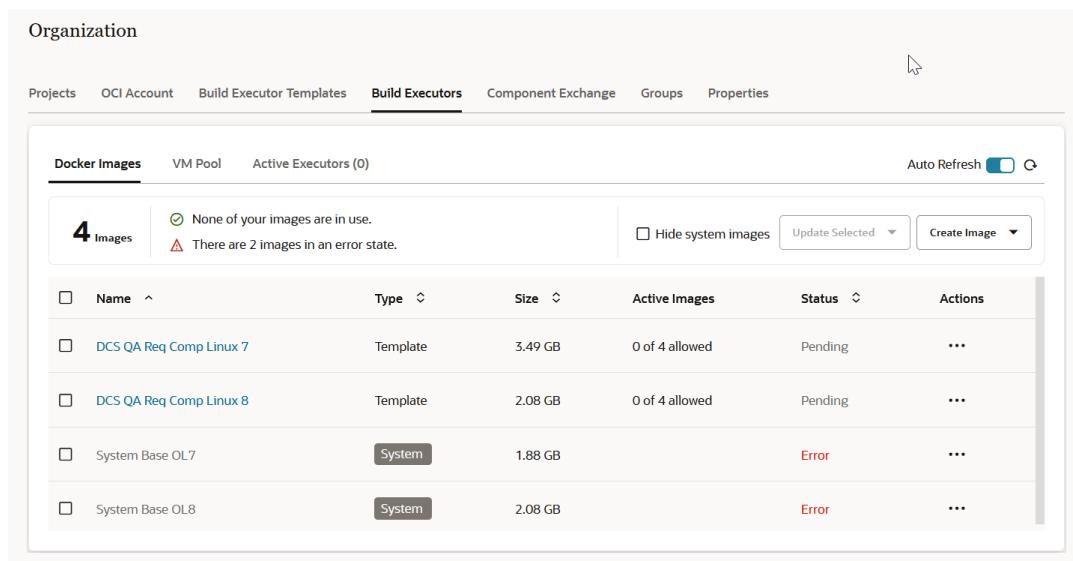
You can create and manage Docker images from the **Build Executors** tab on the Organization page.

1. In the left navigator, click **Organization** .

2. Click the **Build Executors** tab.

3. Click the **Docker Images** tab.

The tab shows the number of Docker images that are in use or are in an error state.



Notice that the system base images, indicated with the **System** badge, are displayed by default. To hide them, click the **Hide system images** checkbox to the right.

This table describes the actions you can perform to create and manage Docker images.

Action	How To
Import a Docker image from an external Docker registry	<ol style="list-style-type: none"> 1. Click Create Image and select Create Image from Registry. 2. In the Add Custom Docker Image dialog box, in Name, enter the image's name. 3. In Registry Host, enter the Docker registry's host name. 4. In Username and Password, enter the credentials of the users who can access the registry. If it's a public registry, leave the fields empty. 5. In Image Name, enter the Docker image's name to be imported. 6. In Version Tag, specify the Docker image's tag 7. In Max Executors, specify the maximum number of executors to be created from the imported image. 8. Click Add.

Action	How To
Create a Docker image from an executor template	You can create only one Docker image from an executor template. 1. Click Create Image and select Create Image from Build Executor Template . 2. From the Build Executor Template drop-down list, select the executor template. 3. In Max Executors , specify the maximum number of executors to be created from the image. 4. Click Save .
Change a Docker image's executor template	Click Action *** and select Edit . In Build Executor Template , change the template and click Save .
Change a Docker image's maximum executors	Click Action *** and select Edit . In Max Executors , change the number of executors and click Save .
Re-create a Docker image	Click Action *** and Recreate Image . In the confirmation dialog box, click OK .
View a Docker image's log	Click Action *** and select Show Log .
Delete a Docker image	Click Action *** and select Delete . In the Delete Docker Image dialog box, click Delete .

Add and Manage Docker Deployment VMs

You can add and manage Docker deployment VMs from the Organization Administration page's Build Executors tab.

Note:

You can't add or manage Docker deployment VMs if VB Studio is connected to the built-in free account. If you want to add Docker deployment VMs, you'll need to configure VB Studio to connect to your OCI account.

1. In the left navigator, click **Organization** .
2. Click the **Build Executors** tab.
3. Click the **VM Pool** tab and follow one of these procedures:
 - To add Docker deployment VMs after you have added the initial VM, see [Add Docker Deployment VMs](#).
 - To view, start, stop, delete, or view the Docker VM logs, see [Manage Docker VMs](#).
 - To manage disk usage for a Docker VM, see [Manage Docker VM Resources](#).

Add Docker Deployment VMs

When you add a Docker deployment VM, you allocate an OCI VM Compute instance to run your builds in Docker executors.

1. From the **VM Pool** tab, click **+ Add VMs**.
2. Add details to the Creating Docker Executors dialog:
 - a. **Number of VMs**: Specify the number of Docker deployment VMs you want to create.
 - b. **Executors created per VM**: Enter the maximum number of Docker executors you want to deploy in each Docker deployment VM.
 - c. **Region**: Select the same region that you chose for the Management VM in Step 5c of [Add Your First Docker Deployment VM](#).
The drop-down list displays regions your OCI account is subscribed to.
 - d. **Shape**: Select the Docker executor's shape.
Wait for a few seconds. VB Studio calculates the number of Compute VM instances that can be created with the selected shape and displays it in the dialog box. If the required number of Compute VM instances aren't available, choose another shape.
 - e. **Volume**: Specify the storage capacity for each Docker deployment VM.
 - f. **Number of OCPUs**: If you've selected a Flex shape, specify the number of OCPUs to add in each Docker deployment VM.
 - g. **Amount of Memory**: If you've selected a Flex shape, specify each Docker deployment VM's memory.
 - h. **Use only for project(s)**: (Optional) Select one or more projects to assign to this Docker deployment VM. After the Docker deployment VM is created, it will be restricted to running builds for the assigned projects. Builds for these projects will only run on this VM, and builds from other projects won't run on this VM.
3. For **VCN Select**, choose **Default** or **Custom**.
 - If you chose **Default**, click **Create**. You are finished with this task.
 - If you chose **Custom**, continue to the next step.
4. Fill out the additional details for the custom VCN:
 - a. **VCN Compartment**: Select the compartment.
If you're an Oracle Cloud OS Management Service (OSMS) user, don't select the OSMS compartment or a compartment with an OSMS policy.
 - b. **VCN**: Select the VCN.
 - c. **Subnets Compartments**: Select the compartments where your public subnets are. By default, it adds your VCN's compartment. If required, you can add more compartments.
 - d. **Subnets**: Select a subnet. The list shows public subnets only.
You can add multiple public subnets. If VB Studio can't create a Docker executor on the first subnet you've added, it tries to create it on the second subnet, and so on.
5. Click **Validate Network Setup**.
6. Click **Create**.

Manage Docker VMs

You can view, start, stop, or delete a Docker deployment VM, view the VM log from the Organization Administration page's Build Executors tab, or show available resources for the VM. You can also reset, view the log, or show available resources for the Docker management VM.

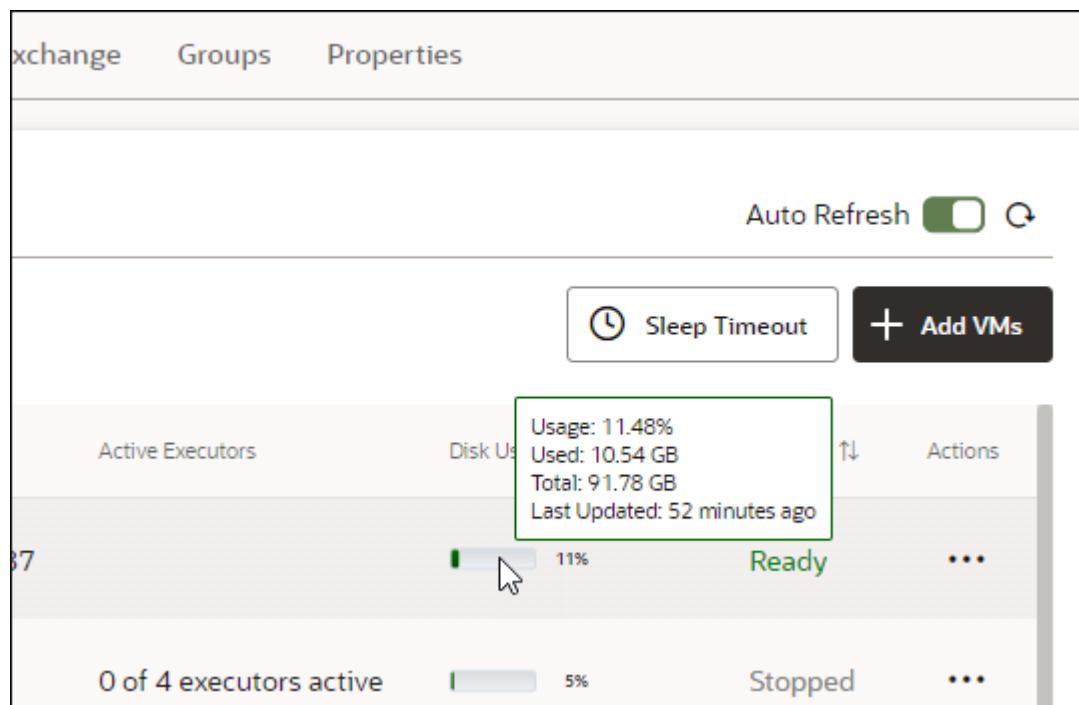
From the **VM Pool** tab, choose one of the actions shown in the following table:

Action	How To
View a Docker deployment VM's or Docker management VM's log	Click Action *** and select Show Log . In the Docker Executor VM Log window, review the log.
Start or stop a Docker deployment VM	Click Action *** and select Start or Stop . To start or stop multiple Docker deployment VMs, select their check boxes, click Update Selected and select Start selected VMs or Stop selected VMs .
Delete a Docker deployment VM	Click Action *** and select Delete . In the confirmation dialog box, click OK . To delete multiple Docker deployment VMs, select their check boxes, click Update Selected and select Delete selected VMs .
Reset a Docker management VM	Click Action *** and select Reset . In the confirmation dialog box, click OK .
View and clean up VM resources	Click Action *** and select Show Resources.. For more information, see Manage Docker VM Resources .
View the projects associated with a restricted Docker deployment VM.	In the Restricted column, hover over to view the projects associated with the VM.

Manage Docker VM Resources

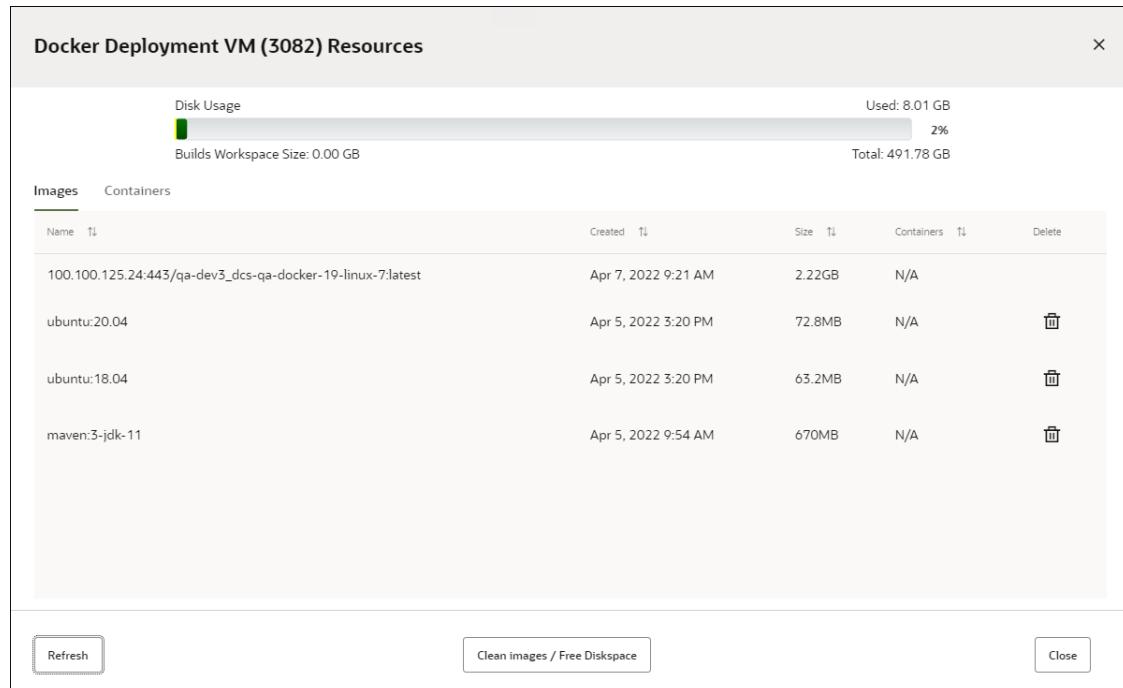
You can view the total disk usage available for a Docker Management VM or Docker Deployment VM and delete images or containers to free up disk space.

For a quick look at the disk usage for a Docker VM, hover over the Disk Usage bar to view usage percentage, GB used, total GB available, and the date that the data was last updated.



The data on this page is only updated every hour or so. For real time usage and additional detail, view the Resources page for the VM.

To view the Resources page for a Docker VM, click *** and select **Show Resources**.



Note:

If you are viewing Docker Management VM resources, you will see the Internal Registry Size instead of the Builds Workspace Size.

From here you can view total disk usage for the VM, as well as usage for individual images and containers.

The following table describes actions you can take to free disk space.

Action	How To
Delete an individual image (for Docker deployment VMs).	In the Images tab, click the trash can next to the image.
Free disk space by removing images that are no longer in use.	In the Images tab, click the Clean Images/Free Diskspace button at the bottom.

Reset Docker and Move to VM Build Executors

If you want to run your organization's builds on VM build executor, you can reset Docker on the Management VM from the Organization page's Build Executors tab.

 **Note:**

If you reset Docker, *all of your VMs and images will be destroyed* and your VM executors will need to be created from scratch. Capture the details of your images and deployment VMs before you begin so they can be recreated manually when you switch to VM executors.

1. In the left navigator, click **Organization** .
2. Click the **Build Executors** tab.
3. Click the **VM Pool** tab.
4. For the **Management VM**, click *** and select **Reset**.
5. In the Delete Docker Deployment VM dialog box, click **OK**.

When you've finished with the migration, you will need to create the VM build executors. See [Add and Manage VM Build Executors](#).

Step 5: Set Up IDCS Users and Groups

This step is required.

To add IDCS users to VB Studio and its projects, make sure they are added to the identity system and assigned the appropriate VB Studio roles. If you want to add groups instead of adding IDCS users individually, see [Manage Your Organization's Groups](#).

To federate with your existing identity provider, see [Federating with Identity Providers](#).

To add IDCS users manually to the identity system, follow these steps:

1. Open the Oracle Cloud Console page.
2. In the upper-left corner, click **Navigation Menu** .
3. Under **Governance and Administration**, select **Identity**, and then select **Federation**.
4. On the Federation page, click the identity service provider's link.
5. On the Identity Provider Details page, click **Create IDCS User**.
6. In the Create IDCS User dialog box, enter the new user's details and click **Create**.
7. To send the password reset instructions and URL to the new user, click **Email Password Instructions**.
8. Click **close**.
9. On the Identity Provider Details page, click the user's IDCS Username link.
10. On the User Details page, click **Manage Service Roles**.
11. On the Manage Service Roles page, search for the service with **Developer Cloud Service** description, click the Actions icon (three dots) and select **Manage Instance Access**.

12. On the Manage Access page, in the **Instance Role** column, select the role you want to grant to the user. A user must be assigned one of these two roles to access VB Studio.

This VB Studio role...	Enables a user to:
DEVELOPER_ADMINISTRATOR	Set up VB Studio, manage all projects, manage VM executors and executor templates, and update the organization details. The user with this role is also called the <i>Organization Administrator</i> . Assign this role role to users who can administer VB Studio.
DEVELOPER_USER	Create and access VB Studio projects. All non-admin users of VB Studio must be assigned this role. Note that this role doesn't allow the user to update the organization details.

13. Click **Save Instance Settings**.

14. On the Manage Service Roles page, click **Apply Role Settings**.

For more details about adding users to IDCS and assigning them roles, see [Managing Oracle Identity Cloud Service Users in the Console](#) and [Managing Instance Roles in the Console](#).

Step 6: Configure your OCI resources

This step is optional. You need to perform task only if you switch OCI compartments or accounts.

To allow your organization's members to run CI/CD builds, you need to configure VB Studio to connect to an OCI account. Before doing that, you need to open the OCI console and set up your OCI account:

1. Create a compartment.
2. Create a group and a user to access the compartment.
3. Create a policy that defines access to the compartment resources.

See [Set Up the OCI Account](#). After creating the resources, get their details. This is described in [Get the Required OCI Input Values](#). You'll need those details to [Set Up the OCI Connection](#) in VB Studio.

You can use the root compartment and the tenancy user that was created when the OCI account was created, but it's recommended to create a dedicated compartment to host VB Studio resources. This allows you to organize VB Studio resources better because they aren't mixed with other tenancy resources. You can also restrict users and control read-write access to the compartment without affecting other resources. To learn more about compartments, see [Understanding Compartments](#).

 **Note:**

If you use Oracle Cloud OS Management Service (OSMS), don't configure VB Studio to use the OSMS compartment, or a compartment with an OSMS policy. OSMS compartments aren't compatible with VB Studio VM build executors.

If you aren't authorized to create and manage OCI resources, ask someone who can and have them share the details of the resources they created.

Set Up the OCI Account

When you switch OCI accounts or change from using the Free VM account to using an OCI account, you may need to manually set up the OCI account and then connect it to VB Studio.

1. Open the OCI console and, in the upper-left corner, click **Navigation Menu** .
2. Select **Identity & Security** and then, under **Identity**, select **Compartments**.
3. On the Compartments page, create a compartment to host VB Studio resources.
 - a. To create the compartment in the tenancy (root compartment), click **Create Compartment**.
 - b. In the Create Compartment dialog box, fill in the fields and click **Create Compartment**.

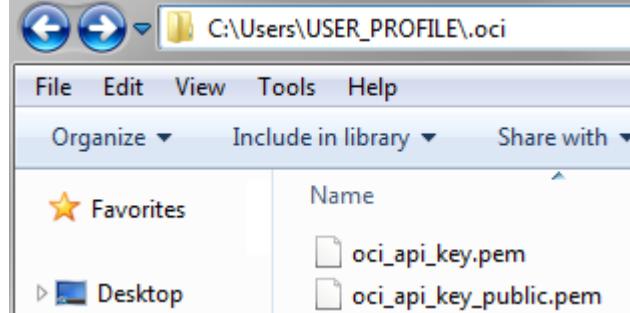
To learn more about compartments, see [Managing Compartments](#).

4. Create a user that can access the VB Studio compartment:
 - a. In the navigation menu, select **Identity & Security**.
 - b. Under **Identity**, select **Users**.
 - c. Click **Create User**.
 - d. Select the **IAM** user type.
 - e. In the Create User dialog box, fill in the fields, then click **Create**.

To learn more about OCI users, see [Managing Users](#).

5. On your computer, generate a private-public key pair in the PEM format.
To find out how to generate a private-public key pair in the PEM format, see [How to Generate an API Signing Key](#).

Here's an example of private-public key files on a Windows computer:



6. Upload the public key to the user's details page:
 - a. Open the public key file in a text editor and copy its contents.
 - b. In the navigation menu, select **Identity and Security**.
 - c. Under **Identity**, select **Users**.
 - d. Click the user's name created in Step 4.
 - e. Under **Resources**, click **API Keys**.
 - f. Click **Add API Key**.

- g. In the Add API Key dialog box, select the **Paste Public Key** option, paste the contents of the public key file, then click **Add**.

To learn more about uploading keys, see [How to Upload the Public Key](#).

7. On the Groups page, create a group for the user who can access the VB Studio compartment and add the user to that group:

- a. In the navigation menu, select **Identity and Security**.
- b. Under **Identity**, select **Groups**.
- c. Click **Create Group**.
- d. In the Create Group dialog box, fill in the fields and click **Create**.
- e. On the Groups page, click the group's name.
- f. On the Group Details page, click **Add User to Group**.
- g. In the Add User to Group dialog box, select the user created in Step 4 and click **Add**.

To learn more about groups, see [Managing Groups](#).

8. In the root compartment, not the VB Studio compartment, create a policy to allow the group created in step 6 to access the VB Studio compartment.

- a. In the navigation menu, select **Identity and Security**.
- b. Under **Identity**, select **Policies**.
- c. On the left side of the Policies page, from the **Compartment** list, select the root compartment.
- d. Click **Create Policy**.
- e. In **Name** and **Description**, enter a unique name and a description.
- f. In **Compartment**, select the root compartment.
- g. In **Policy Builder**, click **Show manual editor** and add these statements:
 - allow group <group-name> to manage all-resources in compartment <compartment-name>
This grants all permissions to the VB Studio group users to manage all resources within the VB Studio compartment.
 - allow group <group-name> to read all-resources in tenancy
This grants read permissions to the VB Studio group so that its users can read—but not use, create or modify—all resources inside and outside the VB Studio compartment. The group users can't use, create, or modify the resources.

Here's an example:

Create Policy

[Help](#)

Name
No spaces. Only letters, numerals, hyphens, periods, or underscores.

Description

Compartment

Policy Builder Show manual editor

allow group VBStudioGroup to manage all-resources in compartment VBStudioCompartment
allow group VBStudioGroup to read all-resources in tenancy

Example: Allow group [\[group_name\]](#) to [\[verb\]](#) [\[resource-type\]](#) in compartment [\[compartment_name\]](#) where [\[condition\]](#)

h. Click **Create.**

To learn more about policies, see [Managing Policies](#).

Get the Required OCI Input Values

Every OCI resource has a unique Oracle-assigned ID called an Oracle Cloud Identifier (OCID). You'll need and use this information (and more) to set up the OCI connection in VB Studio.

To connect to OCI, you need the account's tenancy OCID, home region, the OCID of the compartment that hosts VB Studio resources, and the OCID and the fingerprint of the user who can access the VB Studio compartment. To connect to OCI Object Storage, you need the Storage namespace. You can get these values from the OCI Console pages.

This table describes how to get the OCI input values required for the connection.

To get these values ... Do this:

Tenancy OCID, Home Region, and Storage Namespace

1. Open the OCI console and in the navigation menu, select **Governance & Administration**.
2. Under **Account Management**, select **Tenancy Details**.
The **Tenancy Information** tab displays the tenancy details in these fields:
 - **OCID**: tenancy OCID
 - **Home Region**: home region
 - **Object Storage Namespace**: storage namespace

To get these values ... Do this:

User OCID and fingerprint	<ol style="list-style-type: none"> 1. Open the OCI console and, in the navigation menu, select Identity & Security. 2. Under Identity, select Users. 3. On the Users page, click the user's name. <ul style="list-style-type: none"> • The User Information tab displays the user OCID in the OCID field. Click the Copy link to copy the OCID to the clipboard. • To get the fingerprint of the public key that's associated with your OCI account, scroll down to Resources, select API Keys, and copy the fingerprint value.
Compartment OCID	<ol style="list-style-type: none"> 1. Open the OCI console and, in the navigation menu, select Identity & Security. 2. Under Identity, select Compartments. 3. On the Compartments page, click the compartment's name. The Compartment Information tab displays the compartment's OCID in the OCID field. 4. Click the Copy link to copy the compartment's OCID to the clipboard.

Set Up the OCI Connection

Before you can connect to OCI, you must first get the compartment's details, user details, and the required OCID values. You need to exit VB Studio and open the OCI console to retrieve the details you need. If you're not the OCI administrator, you need to get the details from the OCI administrator.

After you have the OCI input values, you can create an OCI connection from VB Studio:

1. In the left navigator, click **Organization** .
2. Click the **OCI Account** tab.
3. Click **Connect**.
4. In **Tenancy OCID**, enter the tenancy's OCID copied from the Tenancy Details page.
5. In **User OCID**, enter the OCID of the user who can access the VB Studio compartment.
6. In **Home Region**, select the home region of the OCI account.
7. In **Private Key**, enter the user's private key who can access the VB Studio compartment.

The private key file was generated and saved on your computer when you created the private-public key pair in the PEM format. See step 5 in [Set Up the OCI Account](#).

Make sure that the private key you enter contains the -----BEGIN RSA PRIVATE KEY----- and -----END RSA PRIVATE KEY----- markers.

8. In **Passphrase**, enter the passphrase used to encrypt the private key. If no passphrase was used, leave the field empty.
9. When you enter a private key and a passphrase, the **Fingerprint** field is automatically populated. Ensure that the automatically populated fingerprint value matches the fingerprint value of your private-public key pair. If it doesn't, update it to enter the correct value.
10. In **Compartment OCID**, enter the compartment's OCID copied from the Compartments page.

11. In **Storage Namespace**, enter the **Object Storage Namespace** value copied from the Tenancy Details page.
12. Select the terms and conditions check box.
13. Click **Validate**.
14. After validating the connection details, click **Save**.

Here's an example of an **OCI Account** tab filled with required OCI details.

Configure OCI Account

OCI Credentials

Enter your OCI credentials below. We will use this account for storing artifacts and running builds.

Tenancy OCID
ocid1.tenancy.oc1..abcdefgij12345k0l7m8n9o0abcd1234efgh5678xyz321zyx123abcdefg

User OCID
ocid1.user.oc1..abcdefghi09876jklm5n6o7pq8rst901ghi0n6o7pabcfklmzyxuvwtrsp

Home Region
us-ashburn-1

Private Key
-----BEGIN RSA PRIVATE KEY-----
MIIBCD123efghijklm5n6o7pq8rst901ghi0n6o7pabcfklmzyxuvwtrsp
...
cbNF375h59hjgfdDGN4n5ji9J85VH75544FGHHJk9HJK98seFe45Da==
-----END RSA PRIVATE KEY-----

Passphrase

Fingerprint
a1:b2:c3:d4:e5:f6:1e:2d:3c:ab:5a:6b:7c:8d:9e:0a

Compartment OCID
ocid1.compartment.oc1.iad.abcdefghaaaaaaaaaaaaaaaaaaaaaaa09876m8n9o0abcd1234901ghi0n6o7p7gg64q

Storage Namespace
myaccount

* Visual Builder Studio requires these credentials to use the Compute, Network and Object Storage resources of your OCI account. This usage may be extensive. By checking the box, you acknowledge that you have read and accepted the terms specified [here](#).

Validate
● Compute Connection Successful ● Storage Connection Successful
Cancel
Save

ORACLE®

3-54

Set Up VB Studio to Extend Oracle Cloud Applications

This chapter tells you how to set up Oracle Visual Builder Studio (VB Studio) so your users can work with Oracle Cloud Applications in an extension.

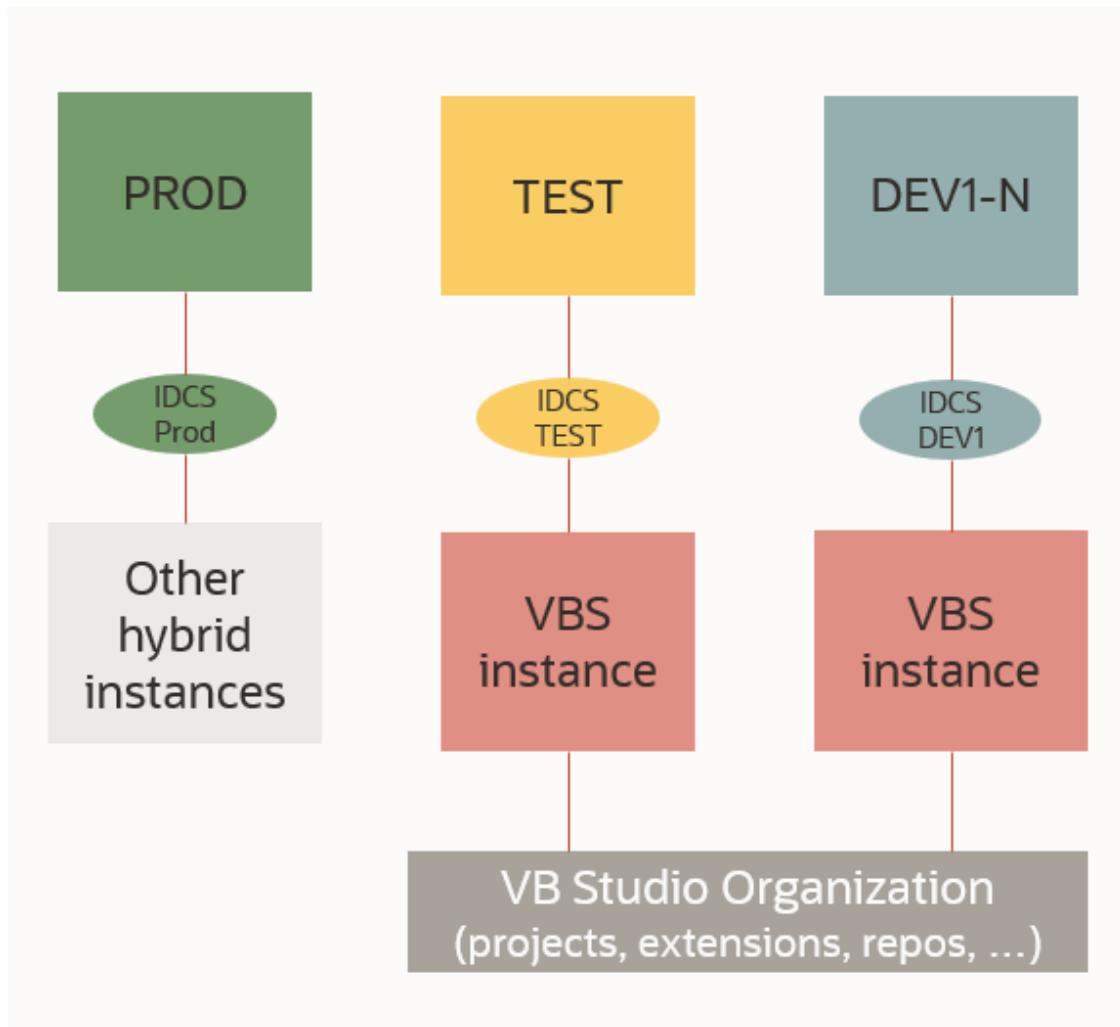
An *extension* is what your team members use to deliver new capabilities into Oracle Cloud Applications. In most cases, an extension contains the work a user has done to customize an app to meet specific business needs. See [What Is an Extension?](#) to learn more.

While this chapter explains everything you need to set up VB Studio for use by others, there are only a handful of activities that MUST be done before a user tries to use VB Studio. Those activities are described in [Essential Set-up Tasks](#). After you have completed these steps, users need only click the **Edit Page in Visual Builder Studio** button in their Oracle Cloud Application, and VB Studio will guide them through the necessary steps so they can begin their work. Of course, there are a few other activities that you may want to perform after completing the essential tasks. Those activities are described in [Advanced Administrator Tasks](#), but they are purely optional.

Before You Begin

Before you begin working through the essential set-up tasks, it's important to understand how your Oracle Cloud Application and VB Studio instances are organized.

Oracle is in the process of provisioning multiple instances of Visual Builder Studio for each Oracle Cloud Applications customer. New customers will see this configuration as soon as they are provisioned with Oracle Cloud Applications; existing customers are being migrated to the new landscape over the next several months. In this new configuration, every TEST and DEV instance in your Oracle Cloud Application environment family receives its own instance of VB Studio, which in turn are tied together by a single VB Studio organization:



By sharing a common organization, users working in VB Studio associated with a TEST instance, for example, are able to access the same projects, repositories, and extensions as users working in a completely different instance—say a DEV instance tied to a different VB Studio instance, or another TEST instance. They can see each other's changes, review and approve each other's merge requests, and collaborate on the same wikis, all as though they were in the same VB Studio instance. This arrangement also underpins and facilitates the use of a **single** project and Git repository for **all** extensions to your Oracle Cloud Applications—no matter if they're from HCM, CRM or another App—which just so happens to be Oracle's recommended best practice. (See Extension Best Practices for more extension-related best practices.)

Here are some important considerations to keep in mind when using multiple VB Studio instances:

- You can use different VB Studio instances for different purposes. For example, you might maintain one or more extensions using VB Studio in your PROD instance, while simultaneously evaluating new functionality in an upcoming Oracle Cloud Applications version using a different VB Studio (likely one associated with a DEV instance running the newer version).
- Having multiple VB Studio instances gives you flexibility in case the instance where you normally work is unavailable for some reason. For example, suppose you usually work in Project A on the DEV1 instance, but DEV1 is down for some reason. As long as your

identity on DEV2 is also a member of Project A—and assuming that you've pushed your latest changes to Git—you can access the project through the VB Studio associated with DEV2 and pick up your work from there. (See [Set Up VB Studio Users](#) for more about user identities.) Note that both DEV1 and DEV2 must be running the same release of Oracle Cloud Applications in order for this scenario to work properly.

- You don't have to have a VB Studio instance associated with an Oracle Cloud Application instance in order to deploy to it. That is, you can deploy an extension to a PROD instance from any DEV or TEST instance in your environment family, despite the fact that PROD is not associated with a VB Studio instance. See [Set Up the Project to Deploy to Production](#) for more.
- If you are developing an extension in one instance and want to test it in another, follow the instructions in [Set Up the Project to Deploy to Other DEV and TEST Instances](#). After you press the **Publish** button in the Designer to commit the changes you made in the extension to the Git repository in VB Studio, you'll need to do the following:
 - Define a new environment and map it to an Oracle Cloud Applications instance, providing the URL to the instance and the credentials for a user that can deploy extensions there. These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.
 - Create a new deployment job that is based on the built-in deployment job that was created for your extension then, in the configuration for the new job, switch the environment used for the deploy step and update the user credentials, as needed.
 - Optionally, include the new build job in your pipelines.

After the job runs successfully you can see your extension deployed on the new instance. You can track the existing deployments through the **Deployment** tab for your environment in VB Studio.

- Although you can use the VB Studio instance that was provisioned alongside your Oracle Cloud Applications account to create visual applications, you'll need an instance of Visual Builder to deploy them to. By default this instance is in a different identity domain than VB Studio, so you'll need to follow the procedures described in [Create and Set Up a Project for Development \(Different Identity Domain\)](#). If you decide to create this instance in the same domain as your VB Studio and Oracle Cloud Applications instance, see [Create and Set Up the Project for Development \(Same Identity Domain\)](#).
- Your organization may have multiple environment families, each with its own TEST instance. If you need to determine which TEST instance VB Studio is associated with, contact your Oracle Cloud account administrator, who has access to your organization's cloud account and can retrieve this information.

 **Note:**

If you don't yet have multiple instances of VB Studio available, you can still move your instance of VB Studio to a different Oracle Cloud Applications instance. By default your VB Studio was provisioned with your first Oracle Cloud Applications TEST instance. To use VB Studio with a different Oracle Cloud Applications TEST instance—or a DEV instance—file a service request so the VB Studio instance associated with your current TEST instance can be terminated, then a new one can be created and associated to your preferred instance. Provisioning VB Studio for PROD instances isn't supported. When you file the service request, please ask for instructions to backup your VB Studio projects so you don't lose any data. For details about what else to include in the service request, see [Move the VB Studio Instance](#). Only one move between pods is allowed, unless a pod must be terminated for some reason and therefore requires another association.

Essential Set-up Tasks

Before you can start setting up or using VB Studio, there are a few tasks that must be performed first:

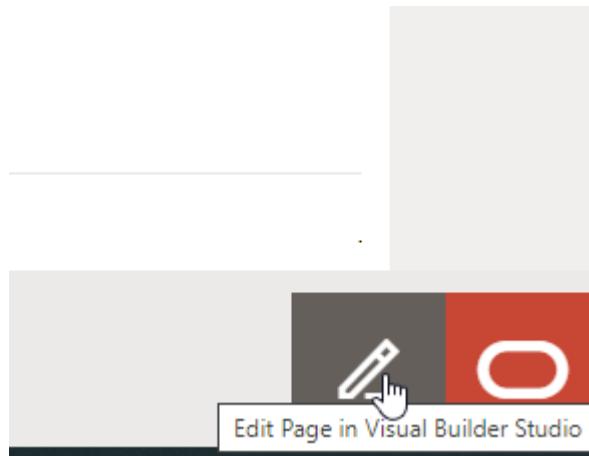
1. Ensure that IDCS user sync is enabled on the IDCS domain associated with **any** instance where you plan to use VB Studio for developing extensions. Developers who plan to use VB Studio must be created and managed in your Oracle Cloud Application instance.
2. Make sure you have the FND_ADMINISTER_SANDBOX_PRIV privilege or any of these roles in your development and production instances:
 - APPLICATION ADMINISTRATOR
 - APPLICATION DEVELOPER
 - SALES_ADMINISTRATOR
 - CUSTOMER RELATIONSHIP MANAGEMENT APPLICATION ADMINISTRATORYou'll need these permissions to set up VB Studio and your Oracle Cloud Applications instances, and to deploy an extension to an Oracle Cloud Application environment.
3. If IDCS role synchronization is enabled (recommended), the Oracle Cloud Applications roles listed in step 2 are sufficient and you should have access to VB Studio after 30 minutes or so. If role synchronization is disabled, you'll need to use the IDCS console and explicitly grant VB Studio access by assigning yourself the DEVELOPER_USER or DEVELOPER_ADMIN VB Studio role in your instance.
See [Assign VB Studio Roles in Oracle Identity Cloud Service \(IDCS\)](#). Without these role assignments, the first user (you or one of your users) who begins to create a simple extension will see the error, "You are not a member of this Organization."
If you can't do this yourself, please consult with the infrastructure administrator at your site who has the necessary IDCS identity administrator privileges.
4. If your Oracle Cloud Applications instances restrict ingress IP access, you must add your VB Studio infrastructure IP address to the list of allowed IP addresses in your Web Application Firewall (WAF) and/or Location Based Access Control (LBAC) configuration. If you have done this, but are experiencing access problems, see [Resolve Access Issues After Allow-Listing Your VB Studio IP Address](#).

Create a Simple Extension

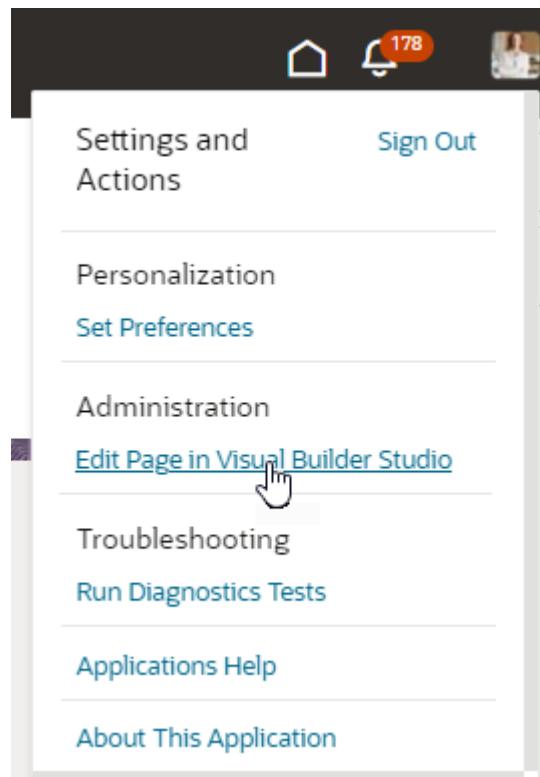
The easiest way to set up VB Studio for others and ensure everything is working properly is to create a simple extension yourself, which walks you through the required steps. There are some other steps you may want to complete later, but those are optional.

The best practice is to create one *project* for each Oracle Cloud Application your users will want to configure, so that all the code for that Application is stored in the same Git repository. (A VB Studio project gathers all the resources you need for your software development effort, depending on what that is.) For that reason, you'll need to repeat the following steps for each such Oracle Cloud Application:

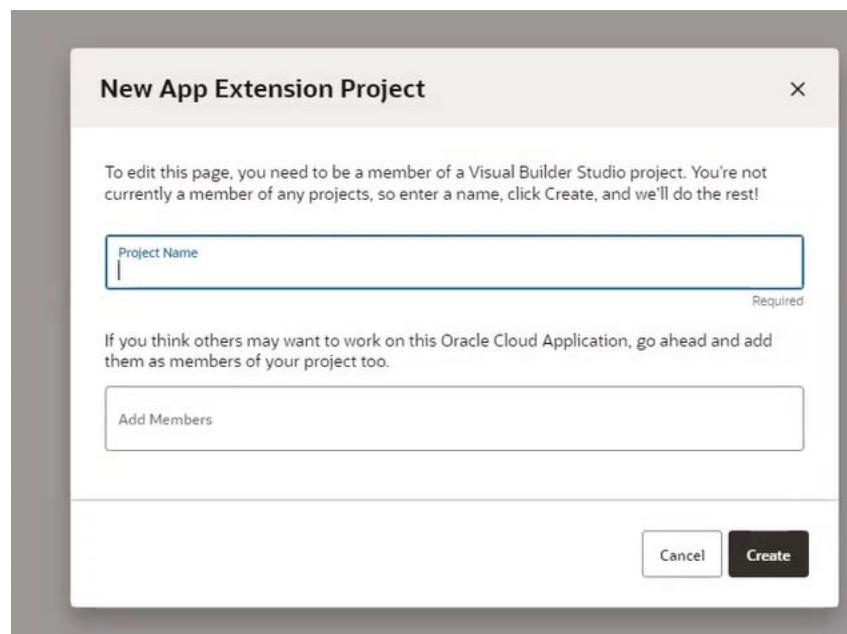
1. Navigate to a page in the Oracle Cloud Application, then click **Edit Page in Visual Builder Studio**:



If you don't see the pencil in the lower right corner, open the Settings and Actions menu and click **Edit Page in Visual Builder Studio**:



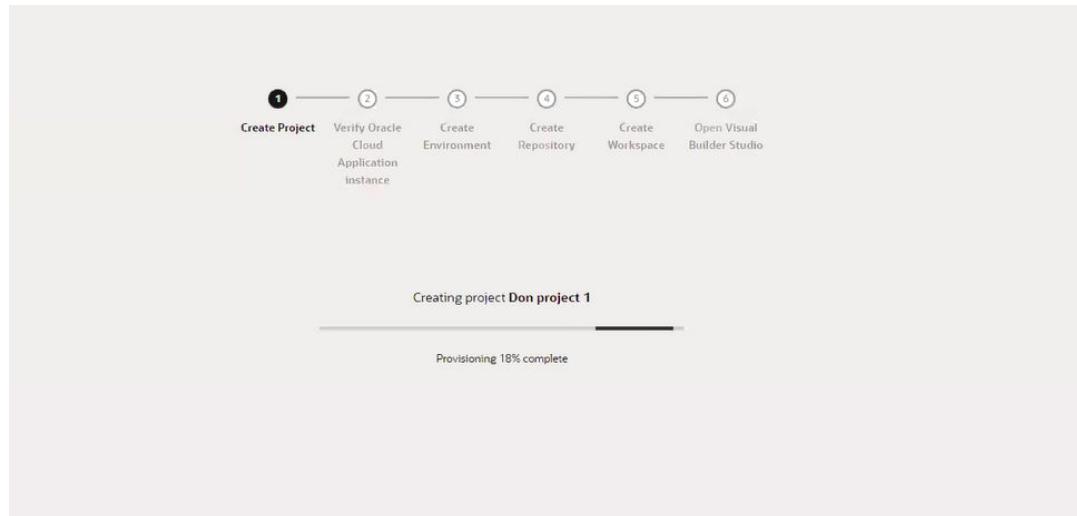
2. In the New App Extension Project dialog, enter a unique name for the project, as well as some team members or groups you think might use this project to extend this Oracle Cloud Application:



 **Note:**

After you've completed this procedure, make sure that the team members you've added have the proper roles, by following the steps in [Set Up VB Studio Users](#).

VB Studio shows you a status dialog so you can follow along as each asset is created:



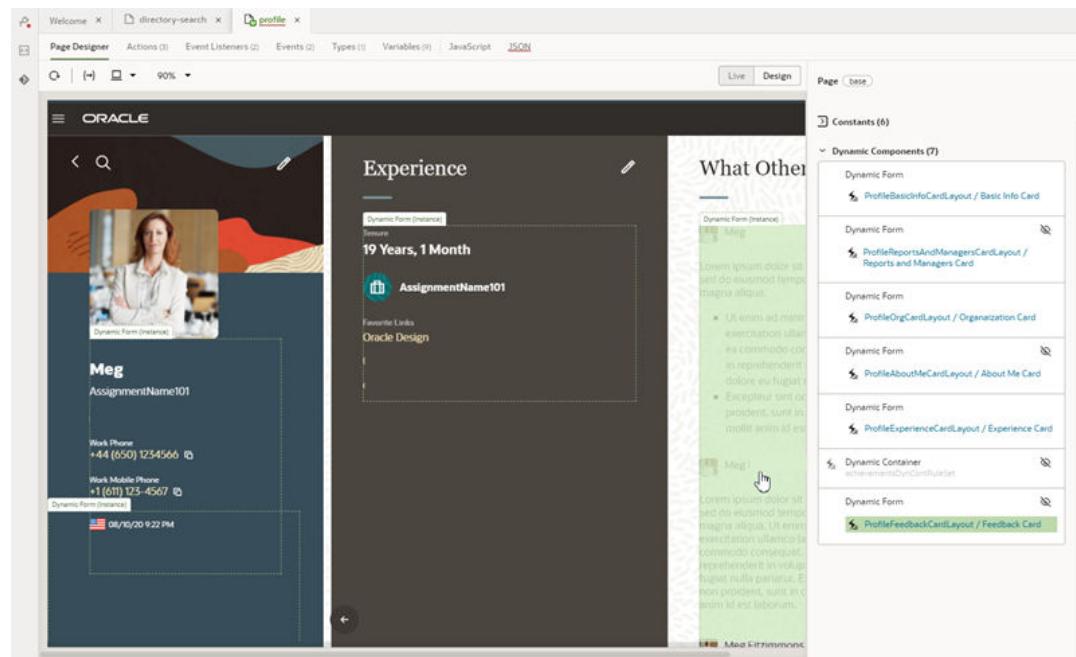
If you're curious about these assets, you can find out more about them in [What Project Artifacts Are Created?](#).

By default, VB Studio creates a private project that's *discoverable*, which means that users you haven't added explicitly to the project will be allowed to choose the project when they click **Edit Page in Visual Builder Studio** from the associated Oracle Cloud Application. Remember, it's best to keep all work for a given Oracle Cloud Application within the same project, so VB Studio will place a "Recommended" badge next to the project's name to encourage users to select it.

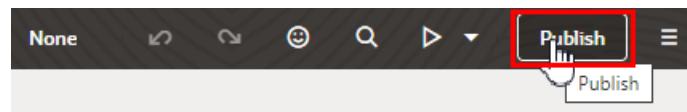
 **Note:**

The default workspace name generated when you jump over to VB Studio typically uses the pillar of the page you're trying to extend, in the format `Workspace PILAR`, for example, `Workspace HCM`. If that name already exists, a number is added to the name and incremented as needed, for example, `Workspace HCM 1`, `Workspace HCM 2`, and so on.

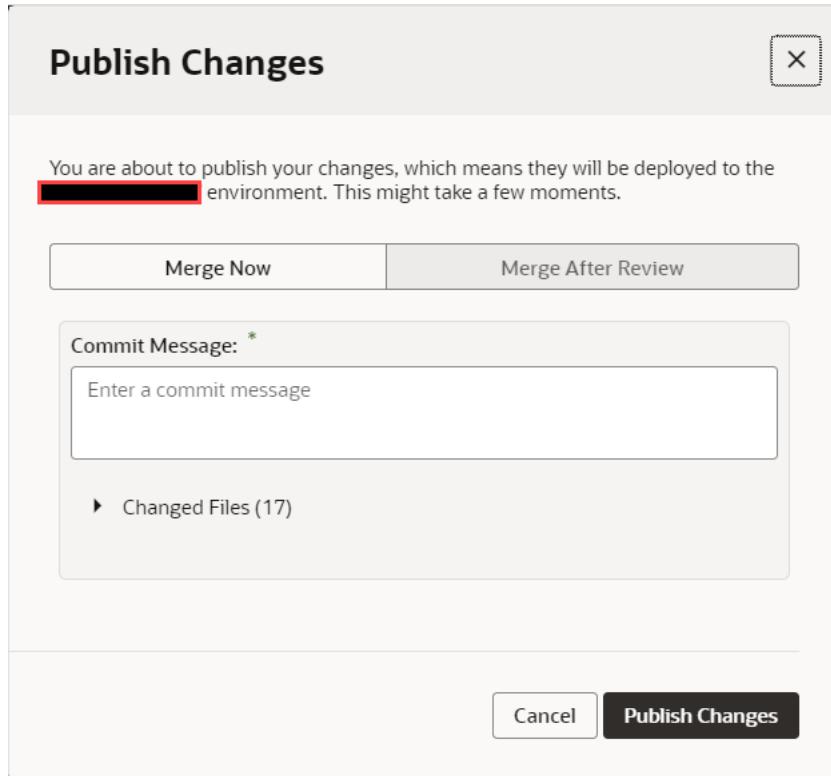
3. VB Studio opens the page you were just viewing in the Designer. When you hover over an extendable component on the canvas, it is outlined in blue, like this:



4. Make a small change to the page, just for testing purposes. You can drag a Heading component onto the page, for example, or add or remove a field on a dynamic form. VB Studio implicitly saves your changes as you work.
5. In the Designer's header, click **Publish**:



The Publish action automatically commits your changes and pushes them to the Git repo, then opens this dialog:

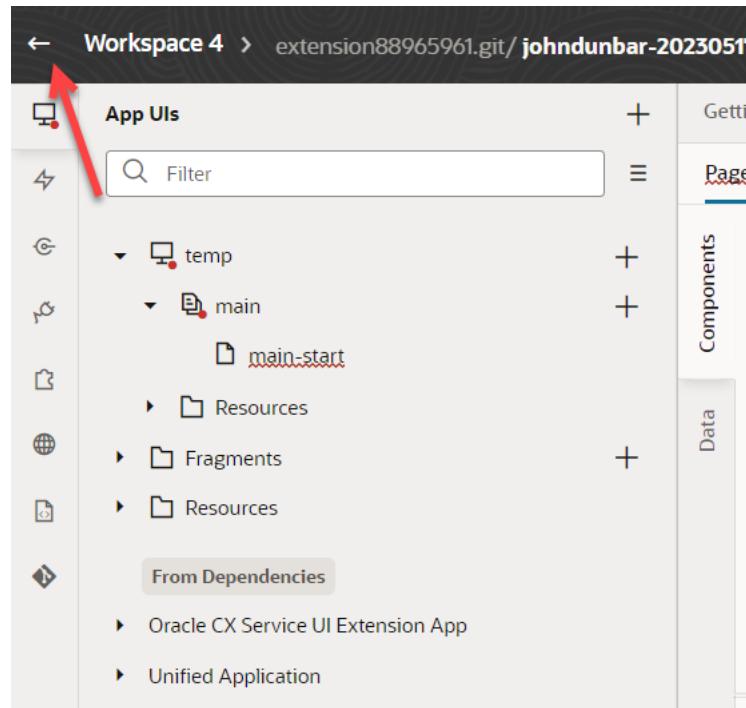


6. Add a brief comment to describe your changes, then click **Publish Changes**.
7. Add valid credentials for this Oracle Cloud Applications instance. Be sure to supply Oracle Cloud Application credentials, as opposed to those for Visual Builder Studio, SSO, or any other system. Once you do this, other users deploying to this instance will not be prompted for credentials.

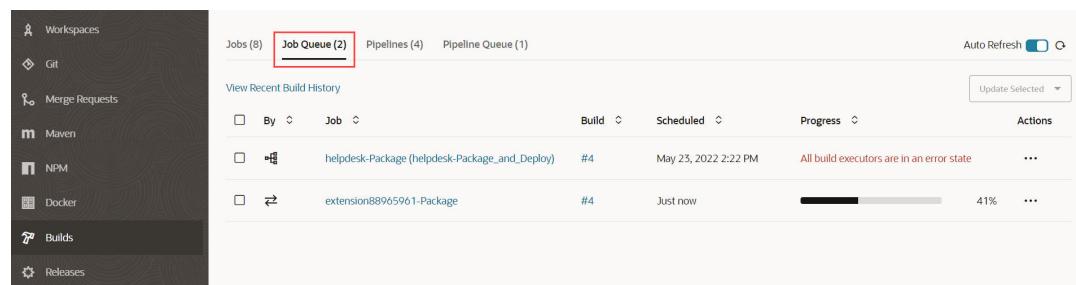
The pipeline that was created as part of the project contains two jobs: a package job (called `extensionName-Package`), which generates the extension's artifact file, and a deploy job (`extensionName-Deploy`, which deploys the extension's artifact file to the Oracle Cloud Application's Development environment. After testing, you can set up a separate pipeline to publish the build artifact to the live (PROD) instance; see [Set Up the Project to Deploy to Production](#).

Let's take a look at the pipeline that's kicked off as part of the Publish process.

8. Click the upper left hand arrow to exit the Designer:



9. In the VB Studio left navigator, click **Builds**, then click **Job Queue**:



In this example, you can see that the package job has already begun running, which will be followed by the deploy job. Depending on how long it takes for you to arrive on the Job Queue tab, you may see a message in the Progress field that says the job is waiting for a VM executor to become available. This should resolve itself shortly.

Take a moment now to investigate the Jobs, Pipelines, and Pipeline Queue tabs.

10. View your published changes.

After the deploy job completes, you'll be able see your work by going to your Oracle Cloud Applications instance and pointing your browser to the App UI you just configured. You may need to re-authenticate to see your latest changes.

11. As this was just a simple test and not a real extension, you may want to remove it. To do so, see [Delete an Extension Version Manually](#).

What's Next?

At this point, you have several options:

- Set up the VB Studio users you added in step 1, if you haven't already. (This must be done before your users can access VB Studio.) See [Set Up VB Studio Users](#).

- Take advantage of some the advanced settings you can apply. See [Advanced Administrator Tasks](#).
- Set up a pipeline to deploy to your PROD instance. See [Set Up the Project to Deploy to Production](#).
- Change some of the project's attributes, like the markup language used to format wiki pages and comments. See [What Else Can You Do with Your Project?](#).

Set Up VB Studio Users

To extend Oracle Cloud Applications applications with VB Studio, you need to assign roles to your users in both Oracle Cloud Applications and IDCS.

Note that in VB Studio, a single user is assigned a unique name to reflect each Oracle Cloud Applications instance where the user is active. For example, UserA in the development instance DEV1 might be known as `UserA@<Oracle Cloud Applications-instance-name>-DEV1` to VB Studio, UserB might be `UserB@<Oracle Cloud Applications-instance-name>-DEV2`, and so on. This means that you may need to assign roles for the same user multiple times, one for each user name/identity.

There are two set-up procedures that you can use to assign your Oracle Cloud Applications users the suggested roles, depending on whether IDCS role sync has been turned off or not.

- If you're a customer that has *not* disabled role sync, you can follow the instructions in [Assign Oracle Cloud Application Roles](#). Oracle Cloud Applications will synchronize those users with IDCS. There is no need for you to manually assign the VB Studio IDCS roles to your users' Oracle Cloud Application roles.
- If you are a customer that *has* disabled role sync between your Oracle Cloud Applications development instance that's associated with VB Studio and its corresponding IDCS stripe, after you follow the instructions in [Assign Oracle Cloud Application Roles](#), you'll need to log into the IDCS console and follow the instructions in [Assign VB Studio Roles in Oracle Identity Cloud Service \(IDCS\)](#) for assigning VB Studio access.

How do you know if role sync has been disabled? There are two ways to find out:

- You can reach out to your infrastructure and identity administrator, who has identity administrator privileges in the IDCS stripe, and have them review this configuration.
- You can wait 30 minutes after you assign the Oracle Cloud Applications roles. If you continue to receive an error from VB Studio indicating that you are not part of the Organization, then role sync is not enabled (or not working), and you'll either need to re-enable it or assign VB Studio roles directly by using the IDCS console and the instructions in [Assign VB Studio Roles in Oracle Identity Cloud Service \(IDCS\)](#).

As an administrator, you must have certain privileges to assign the necessary roles:

- In Oracle Cloud Applications, you must have one of these privileges: cloud administrator, service administrator, or application administrator. Everyone who assigns Oracle Cloud Applications roles to their users needs one of these privileges.
- In IDCS, you must have either an Identity Domain Administrator or User Administrator role. You only need these roles if IDCS role sync has been turned off and you need to assign the VB Studio IDCS roles manually.

This table shows the VB Studio IDCS roles that are mapped to users with Oracle Cloud Application roles:

Oracle Cloud Applications users with these roles:	Need to be mapped to this VB Studio IDCS role:
Application Administrator (ORA_FND_APPLICATION_ADMINISTRATOR_JOB) (OB)	VB Studio administrator (DEVELOPER_ADMINISTRATOR)
Sales Administrator (ORA_ZBS_SALES_ADMINISTRATOR_JOB)	
Customer Relationship Management Application Administrator (ORA_ZCA_CUSTOMER_RELATIONSHIP_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB) (OB)	
Synchronization Enabled Administrator Identity (ORA_SYNC_ENABLED_ADMINISTRATOR_ABSTRACT)	
Human Capital Management Application Administrator (ORA_HRC_HUMAN_CAPITAL_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB)	
Application Developer (ORA_FND_APPLICATION_DEVELOPER_JOB)	VB Studio user (DEVELOPER_USER)
Synchronization Enabled Developer Identity (ORA_SYNC_ENABLED_DEVELOPER_ABSTRACT)	

How Do I Assign the Roles?

If IDCS role synchronization hasn't been disabled for your Oracle Cloud Applications environment, you can follow the instructions in [Assign Oracle Cloud Application Roles](#), wait a half hour while the roles are synchronized with the IDCS system, and then your users should be able to use VB Studio without any further involvement on your part.

If role synchronization has been disabled, role assignment is a two-step process:

1. Create an Oracle Cloud Applications user, then assign a role to that user using the Oracle Cloud Applications Identity manager. See [Assign Oracle Cloud Application Roles](#).
2. In IDCS, manually assign a VB Studio role (DEVELOPER_ADMIN or DEVELOPER_USER) to the Oracle Cloud Applications user. See [Assign VB Studio Roles in Oracle Identity Cloud Service \(IDCS\)](#).

Assign Oracle Cloud Application Roles

Before beginning the process, review some background information:

- Standard roles (listed in step 7 below) are predefined. Their permissions are automatically updated as necessary, such as when new features or services are added.
- Custom roles are created as substitutes for standard roles, allowing only specific privileges. These privileges are assigned in the Oracle Cloud Applications security console.

A custom role can be one of the following:

- If role synchronization is enabled, one of the Oracle Cloud Applications abstract roles (ORA_SYNC_ENABLED_ADMINISTRATOR_ABSTRACT or ORA_SYNC_ENABLED_DEVELOPER_ABSTRACT) will need to be assigned in the Oracle Cloud Applications security console along with the

FND_ADMINISTER_SANDBOX_PRIV and FND_MANAGE_SANDBOX_PRIV privileges.

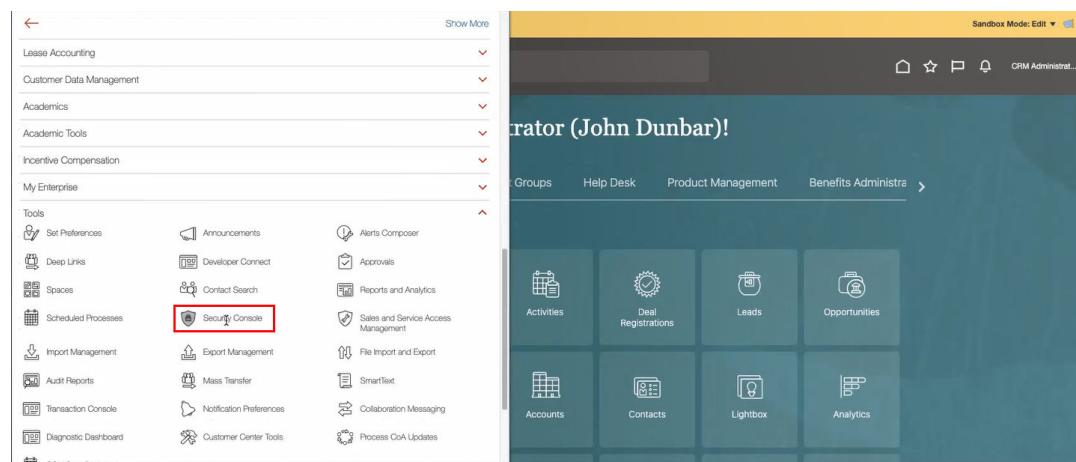
- If role sync is disabled, one of the VB Studio IDCS roles (DEVELOPER_ADMINISTRATOR or DEVELOPER_USER) will need to be assigned instead of the Oracle Cloud Applications abstract roles. The VB Studio IDCS roles are assigned in the IDCS console. The custom roles are assigned to VB Studio IDCS roles in the same manner as standard roles, with administrative roles mapping to the VB Studio administrator role in IDCS and non-administrative roles being assigned the VB Studio developer role in IDCS.

To assign an Oracle Cloud Applications role to a new or existing Oracle Cloud Applications user:

1. Sign in to the Applications Console.

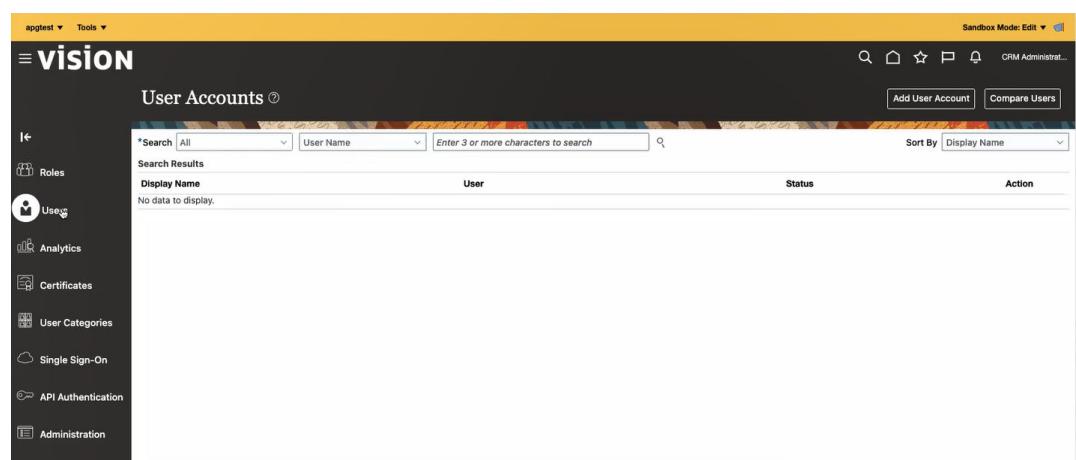
Make sure that you specify the identity domain where you want to create the user.

2. Click **Navigation**  menu to open the navigation menu and, under **Tools**, click **Security Console**.



3. Click **Users** in the navigation pane.

The User Accounts page is displayed.



4. To create a new user, click **Add User Account** and follow the prompts. To assign a role to an existing user, use the Search field to find the user you want, then skip to step 6.
The Add User Account page is displayed.
5. Fill out the User Information fields (**First Name**, **Last Name**, **Email**, **Password**, **Confirm Password**).

The screenshot shows the 'Add User Account' page in the Oracle Cloud Applications interface. The left sidebar includes links for Roles, Users, Analytics, Certificates, User Categories, Single Sign-On, API Authentication, and Administration. The main form has sections for 'Associated Person Information' (Associated Person Type dropdown set to 'None'), 'User Information' (User Category dropdown set to 'DEFAULT', with fields for First Name ('Alex'), Last Name ('Cloud'), Email ('alex.cloud@example.com'), User Name ('Alex.Cloud'), Password ('.....'), and Confirm Password ('.....')), and 'Roles' (Role dropdown set to 'Application Administrator'). There are also buttons for 'Save and Close' and 'Cancel' at the top right.

The **User Name** field has been filled in for you, using the first and last names you entered separated by a period.

6. Click **Add Role**.

The Add Role Membership from Role page is displayed.

7. Select one of the standard Oracle Cloud Applications roles:

- Application Administrator (ORA_FND_APPLICATION_ADMINISTRATOR_JOB)
- Sales Administrator (ORA_ZBS_SALES_ADMINISTRATOR_JOB)
- Customer Relationship Management Application Administrator (ORA_ZCA_CUSTOMER_RELATIONSHIP_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB)
- Synchronization Enabled Administrator Identity (ORA_SYNC_ENABLED_ADMINISTRATOR_ABSTRACT)
- Application Developer (ORA_FND_APPLICATION_DEVELOPER_JOB)
- Synchronization Enabled Developer Identity (ORA_SYNC_ENABLED_DEVELOPER_ABSTRACT)
- Human Capital Management Application Administrator (ORA_HRC_HUMAN_CAPITAL_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB)

 **Note:**

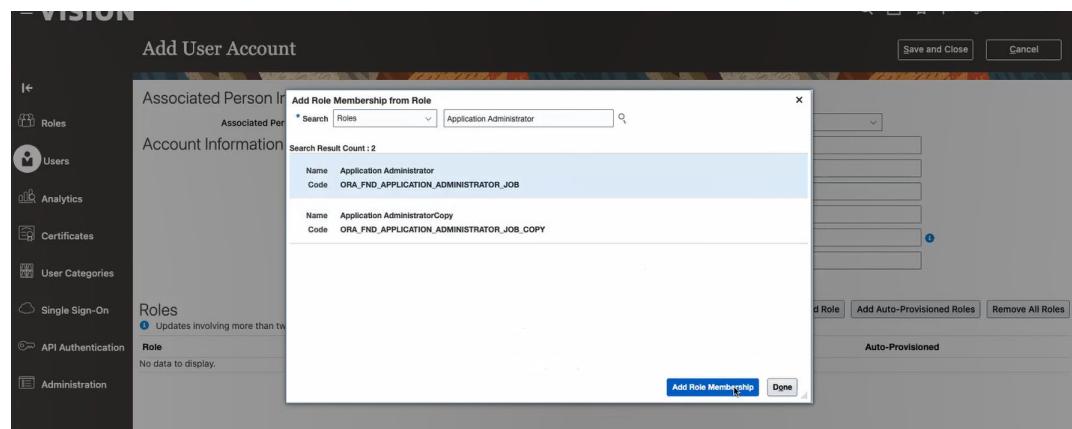
To see the list of assignable roles when extending Oracle Cloud Applications, users need to be assigned a role with the PER_REST_SERVICE_ACCESS_USERS_AND_ROLES_LOVS_PRIV privilege, such as ORA_PER_EMPLOYEE_ABSTRACT or ORA_HRC_HUMAN_CAPITAL_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB.

 **Note:**

The ORA_SYNC_ENABLED_ADMINISTRATOR_ABSTRACT and ORA_SYNC_ENABLED_DEVELOPER_ABSTRACT roles grant access to VB Studio, but do not offer any additional privileges for Oracle Cloud Applications.

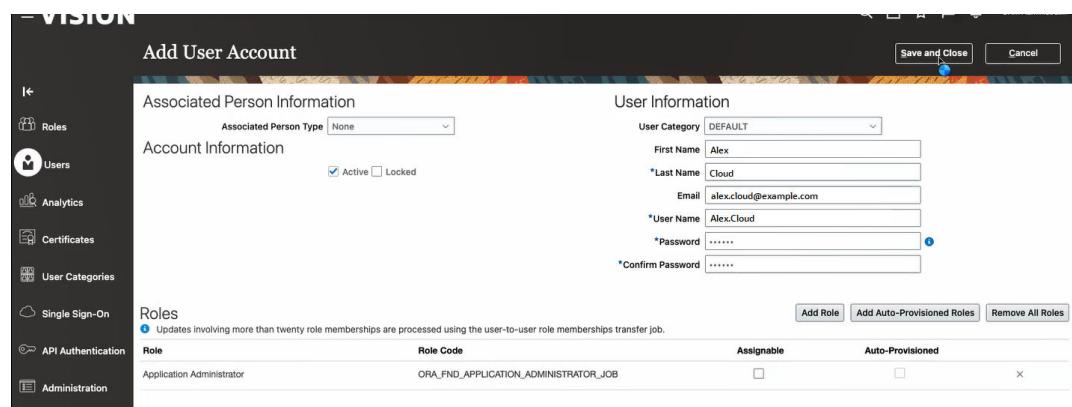
If you created a custom role, you can select it instead of a standard role.

- After you select the role to assign to the user, click **Add Role Membership**.



- Click **Done**.

The Add User Account page is displayed, showing the new user with an assigned role.



10. Click Save and Close.

From the time role membership was granted via the security console, it will take approximately 30 minutes for a user to gain access. If, after 30 minutes, a user is still seeing a warning message indicating that they are not a member of the organization, it is very likely that IDCS role sync has been disabled on your environment, and you need to follow the instructions in [Assign VB Studio Roles in Oracle Identity Cloud Service \(IDCS\)](#) to assign one of the VB Studio roles in IDCS to the Oracle Cloud Application user.

Assign VB Studio Roles in Oracle Identity Cloud Service (IDCS)

If you just created a new user in Oracle Cloud Applications, it will take at least 30 minutes for the user profile to show up in IDCS so that you can assign a VB Studio user role to it.

IDCS has two different interfaces (upgraded and Classic), so choose the process that's applicable to you.

Assign Roles with the Upgraded IDCS Interface

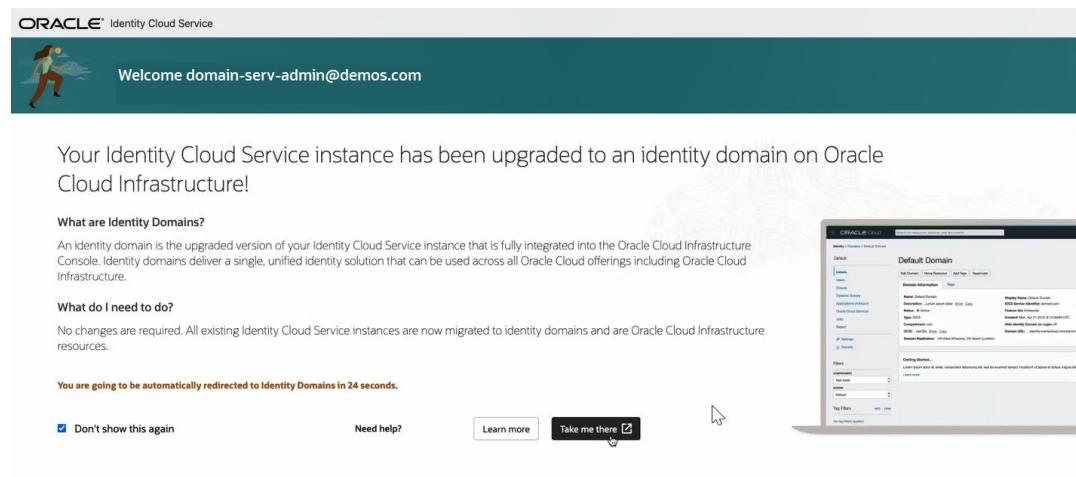
Assign VB Studio access to users in the IDCS stripe associated to the Oracle Cloud Application instance:

1. Sign in to your Identity domain using your credentials and the URL you received.

Note:

If you don't have access to the URL, see [How Do I Access the IDCS Console From the Oracle Cloud Console?](#)

If your Identity Cloud Service instance has been upgraded to an identity domain on Oracle Cloud Infrastructure (OCI), you'll see this page:



2. Click the **Take me there** button.

The domain's Overview page is displayed.

 **Tip:**

If you click the **Don't show me this again** check box, the page with the upgrade notice will be bypassed and you'll see the domain's Overview page after you log in.

If you see a page that looks quite different, your IDCS hasn't been upgraded and you're using the Classic interface. See [Assign Roles with the Classic IDCS Interface](#).

3. Select **Visual Builder Studio** from the **Oracle Cloud Services** list under the Identity domain.
4. In the **Resources** list, click **Application roles**. All member roles, in this case **DEVELOPER_ADMINISTRATOR** and **DEVELOPER_USER**, that can be assigned for the VB Studio service are displayed.

Name	Description
DEVELOPER_ADMINISTRATOR	Visual Builder Studio Administrator
DEVELOPER_USER	Visual Builder Studio User

5. Click the down arrow on the right side of the role's row to display the list of resources you can manage:

Name	Description
DEVELOPER_ADMINISTRATOR	Visual Builder Studio Administrator

Assigned users: 2 Manage
Assigned groups: - Manage
Assigned applications: - Manage

6. Click **Manage** next to **Assign Users**. The **Manage user assignments** dialog is displayed.

The screenshot shows a table titled "Assigned users (2)". It includes columns for First name, Last name, Email, Mobile phone number, and Member type. Two users are listed: John Dunbar (john.dunbar_etest@demos.com, Direct) and Alex Admin (alex.admin_etest@demos.com, Direct). A "Revoke" button is at the top left, and a search bar at the top right. A "Show available users" link is at the bottom left.

<input type="checkbox"/>	First name	Last name	Email	Mobile phone number	Member type
<input type="checkbox"/>	John	Dunbar	john.dunbar_etest@demos.com	-	Direct
<input type="checkbox"/>	Alex	Admin	alex.admin_etest@demos.com	-	Direct

0 selected Showing 2 users < Page 1 >

+ Show available users

Close

- Click **+ Show available users**, then use Search to locate the user name you are searching for.

The screenshot shows a table titled "Available users (2)". It includes columns for First name, Last name, Email, Mobile phone number, and Member type. One user is listed: Alex Cloud (alex.admin_etest@demos.com, Unassigned). A "Hide available users" link is at the top left, and a search bar at the top right. A "Assign" button is at the top left of the table area.

<input type="checkbox"/>	First name	Last name	Email	Mobile phone number	Member type
<input type="checkbox"/>	Alex	Cloud	alex.admin_etest@demos.com	-	Unassigned

0 selected Showing 1 user < Page 1 >

Hide available users

Assign

Cloud

Close

- After you locate the user, click the **Assign** checkbox on the left side of the user's row, then click the **Assign** button.
- The **Assigned users** section of the panel shows the new user you assigned.

9. Click the **Close** button.
10. Repeat steps 7-10 to assign additional users.

Assign Roles with the Classic IDCS Interface

To assign VB Studio access to Oracle Cloud Applications users in the Classic stripe:

1. Sign in to your Identity domain using your credentials and the URL you received in the Welcome email.
The Identity Cloud Service Welcome page is displayed.

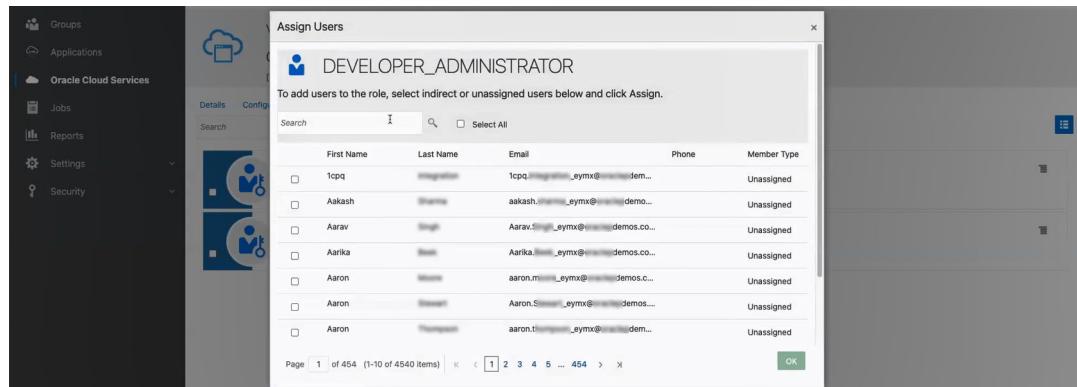
2. Click the **Navigation** menu.
The Navigation drawer is displayed.
3. Click **Oracle Cloud Services** to display the list of service instances that are available in your identity domain.
4. Scroll to find the service (or use the Search bar to find it), then click the box on the left side of the row to select it:



The App Details page is displayed.

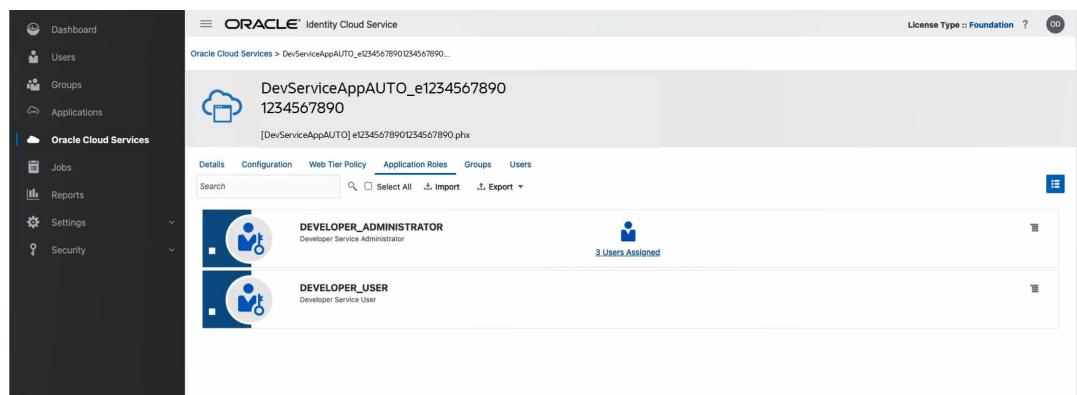
5. Select the **Application roles** tab, click the **Menu** icon on the right and select **Assign Users**.

6. In the Assign Users page, use **Search** to locate the user that you want to assign the VB Studio IDCS role to, then click the check box on the left side of the row to select the user.



7. Click OK.

The **Application Roles** tab should now show an additional user (the one you just assigned the role to).

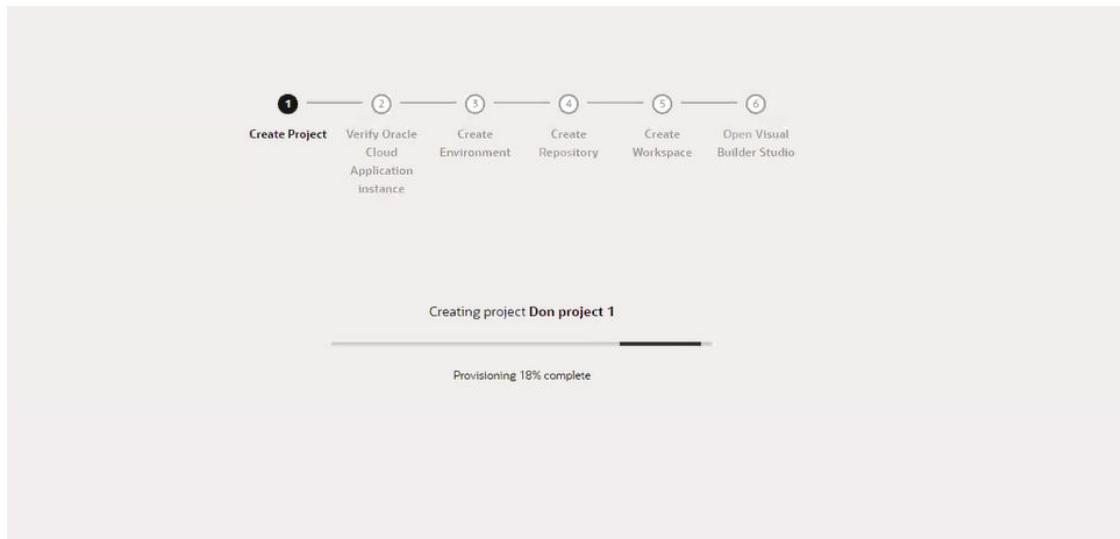


8. To verify that the user was assigned correctly, click the # Users Assigned link.

The **Users Assignments** page is displayed, listing all the users who've been assigned the VB Studio IDCS role.

What Project Artifacts Are Created?

When you create a new project, several other VB Studio artifacts are also created for you. VB Studio shows you a status diagram so you can follow along as each artifact is generated:

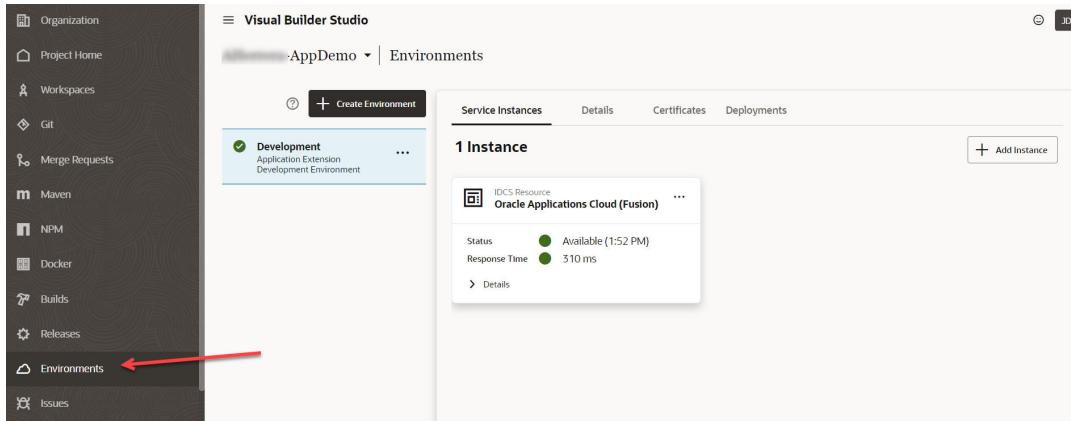


These artifacts include:

- A Git repository, which will contain the source code for all the extensions based on this project.
To see the Git repository's files, go to the **Project Home** page, click the **Repositories** tab, then click the Git repository name:

The screenshot shows the 'Repositories' tab selected in the navigation bar. Below it, there is a search bar labeled 'Filter Git repositories' with a magnifying glass icon. There are two buttons: 'All' and 'Favorites'. To the right of the search bar is a dropdown menu labeled 'Favorites First'. At the bottom left is a star icon next to the text 'digital-sales-app-extension.git'. At the bottom right is a 'Clone' button with a dropdown arrow.

- A **Development** environment, pointing to the development instance where the Oracle Cloud Application you plan to customize is running. (A VB Studio *environment* can include one or more instances of Oracle Cloud Applications, Visual Builder, Oracle Cloud SaaS, or Oracle Cloud Infrastructure service instances, so you can treat them as a single entity.) To see the definition for the Development environment, click **Environments** in the VB Studio navigator:



- A workspace, which is a private area where you work on your extensions. Each user has a unique workspace, although users can collaborate on the same extension by cloning each other's workspace (which also clones the project's Git repository).
- Build jobs that package and deploy the extension's artifact to Oracle Cloud Application's development instance.
The package job, named **extensionName-Package**, generates the extension's artifact file. The deploy job, **extensionName-Deploy**, deploys the extension's artifact file to the Oracle Cloud Application instance defined in the Development environment. (If you [create a project manually](#), these jobs are named **Application-Extension-Package** and **Application-Extension-Deploy**.)
- The **extensionName-Package and Deploy** pipeline, which runs the package and deploy build jobs in a sequence. (If you create a project manually, this pipeline is named **Application Extension - Package and Deploy**.)
- A VM executor, but only if this is the first project in this VB Studio's instance and this instance did not already have any VM executors. (Builds in a build system run in *build executors*. The operating system and software packages required to run builds on build executors are defined in *build templates*.) The VM executor uses the **System Default OL7 for Visual Builder** executor template. You can use this VM executor to run build jobs that reference the **System Default OL7 for Visual Builder** template in the current project and other projects as well.

Advanced Administrator Tasks

Once you've completed the essential tasks and set up one or more projects, there are a few other things you may want to do to tailor operations so they better suit your organization's needs:

- [Set merge restrictions](#) on the `main` branch for the project's Git repository;
- Set up the project to [deploy to your PROD instance](#);
- Set up the project to [deploy to a different DEV or TEST instance](#).

Set Merge Restrictions on the `main` Branch

By default, the `main` branch is accessible to all project users and anyone can make changes to its files. To restrict changes and who can push commits to it, you may want to set restrictions on it and allow branch merges only after they are approved.

Be sure to repeat this process for each project you created in [Create a Simple Extension](#).

1. In the left navigator, click **Project Administration**.
2. Click **Branches**.
3. In **Repository and Branches**, select the Git repository and the `main` branch.
4. Select the **Requires Review** option.
5. In **Default Reviewers**, enter and select the users.

A default reviewer is a project member who is automatically added as a reviewer when a merge request is created on the branch.

6. From the **Approvals** drop-down list, select the minimum number of reviewers who must approve the review branch of a merge request, where the selected branch is the target branch
7. (Optional) To allow a review branch to be merged to the selected branch only if the last build of the linked job in Merge Request is successful, select the **Require successful build** check box.
To use this option, link a build job to a merge request.
8. (Optional) To reset the approval status of reviewers if change is pushed to a branch after they have approved the merge request, select the **Reapproval needed when branch is updated** check box.
9. (Optional) To ensure changes pushed to the target branch match the contents of the review branch, select the **Changes pushed to target branch must match review content** check box.
10. (Optional) In **Merge Request Exempt Users**, specify users who can bypass the branch restrictions and merge the review branch of a merge request outside VB Studio or without required approvals.
This is useful if you want to allow some users to merge the review branch irrespective of review conditions being met.

11. Click **Save**.

Set Up the Project to Deploy to Production

By default, a project deploys extensions to the Development environment, which is the Oracle Cloud Application instance you were using when you clicked **Edit Page in Visual Builder Studio**. Follow these steps to deploy to the PROD instance, so users can deploy to it after their development and test cycles are complete.

The Oracle Cloud Applications PROD instance resides in a different identity domain from the DEV and TEST instances, which means you'll need the URL for the PROD instance, as well as user credentials that VB Studio can use to deploy to this instance. These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.

To set up a project to deploy to PROD:

1. [Add the PROD instance to an environment](#)
2. [Create a production branch](#)
3. [Create and configure production build jobs](#)
4. [Configure the pipeline so that the packaging and deployment jobs run in succession](#)
5. [Run the production pipeline.](#)

 **Tip:**

For a simpler interface where you can manage the lifecycle of all your extensions, use the Manage Extension Lifecycle page. From this single page, you can deploy extensions to your PROD instance without the need to create build jobs and pipelines, as long as that instance has an authentication method of OAuth.

Add the Oracle Cloud Application's Production Instance to an Environment

Only one Oracle Cloud Applications instance can be added to an environment. As the Development environment already points to the TEST instance (or DEV, if you asked Oracle to re-associate it), you must create a new environment for the PROD instance.

To create a new environment for the PROD instance:

1. In the VB Studio navigator, click **Environments**.
2. Click **+ Create Environment**.
3. In the Service Instances tab, click **+ Add Instance**.
4. In the Add Service Instances dialog box, select the **Oracle Cloud Applications** option button.
5. Under Authentication Method, select the **Application Credentials** option.
6. In **Base URL**, enter the Oracle Cloud Application's base URL.
7. In **Instance Name**, if required, update the instance's display name. The name will be displayed in the **Service Instances** tab.
8. In **Username** and **Password**, enter the credentials of a user who can access this Oracle Cloud Applications instance.

These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.

9. Under Authorization Type, select **OAuth 2.0 or Basic**.
10. Click **Add**.
11. If you chose OAuth 2.0 as the Authorization Type, you may be prompted to authorize IDCS to acquire OAuth tokens for accessing the target instance. Click **Authorize** when prompted, then sign in to IDCS with credentials to access the Oracle Cloud Applications instance.

 **Note:**

OAuth tokens (access and refresh) are typically cycled during regular use. If a token expires during an extended period of inactivity, like when you're away on vacation, click **Actions ...** and select **Renew OAuth Access** to restart the OAuth workflow.

Note that the username that shows in the environment Details section isn't the VB Studio user, but the user for the connected instance whose credentials were used to create the connection's refresh token. The details don't show the VB Studio username, which isn't relevant for a 3-legged OAuth Environment connection.

If the newly added instance stays in the **Unknown** status for some time, it usually indicates that the IDCS Application provisioning may have failed. VB Studio added the Oracle Cloud Applications instance, but can't access it. In this case, click **Actions** *** and select **Remove** to remove the Oracle Cloud Applications instance from the environment, and then click **Add** to add it again.

Create a Production Branch

Follow your organization's guidelines to create a branch and protect it from unverified changes. To protect the branch, you can set merge restrictions, make the branch private, and restrict who can push commits to it, or freeze it.

1. In the left navigator, click **Git** .
2. Click the **Refs** view and then click **Branches** .
3. From the **Repositories** drop-down list, select the repository.
4. Click **+ Create Branch**.
5. In the New Branch dialog box, in **Name**, enter the branch name. From the **Base** drop-down list, select the `main` branch as the base branch.
6. Click **Create**.

After creating the production branch, any changes pushed to the `main` branch aren't automatically available in the production branch. You must create a merge request or manually push the changes to the production branch.

If you want to set merge restrictions on the production branch, see Set Review and Merge Restrictions on a Repository Branch. To freeze the branch or make it private, or set other restrictions, see Protect a Branch.

Create and Configure Production Build Jobs

You need to set up some packaging and deployment jobs before you can deploy extensions to your Oracle Cloud Application's PROD instance. Follow this process:

1. Migrate the configurations to the production Oracle Cloud Application instance. See [Overview of Configuration Life Cycle](#) and [Overview of Migration](#) in *Configuring and Extending Applications* for instructions.
2. Create a build job that packages the extension. See [Create the Production Packaging Build Job](#) for instructions.
3. Create a build job that deploys the extension to the production instance. See [Create the Production Deployment Build Job](#) for instructions.
4. (Optional) Restrict who can see or edit the production build jobs or run their builds. See [Configure a Job's Privacy Setting](#) for instructions.
5. Configure the pipelines to run the packaging and deployment jobs in succession. See [Create and Configure the Production Pipeline](#) for instructions.
6. Run the production pipeline to package the extension and deploy it to the production instance. See [Run Production Pipelines](#) for instructions.

Before You Configure Build Jobs and Pipelines

Here are some things you need to know before you configure and run build jobs and pipelines:

- Make sure that the source and target instances are of the same release, with the same standard and one-off patches applied to both environments.
- In the development packaging job, if you changed the default artifact's file name, get the new name and its path.
- If you configured the development packaging job to overwrite the application's version defined in `visual-application.json`, get the new version. You'll configure the production's packaging job to use the same version.

Create the Production Packaging Build Job

The packaging job generates an extension artifact that's ready to deploy to the mainline.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. In **Template**, select the **System Default OL7 for Visual Builder** template.
6. Click **Create**.
7. Click **Configure** .
8. Click the **Git** tab.
9. From the **Add Git** list, select **Git**.
10. In **Repository**, select the Git repository. In **Branch or Tag**, select the production branch.
11. Click the **Steps** tab.
12. From **Add Step**, select **Application Extension**, and then select **Package**.
13. By default, the packaging step minifies the application's source code before running the build. If you don't want to minify the source files, deselect the **Optimize extension** check box.

Minification is a process to remove the unnecessary characters (such as blank spaces, new lines, and comments) from the source code and reduce the size of the files, making the transfer of files consume less bandwidth and storage.

Note:

When you deselect the **Optimize extension** check box, this warning is displayed: **Optimization not selected. Packaging without optimization can cause performance problems, so avoid doing so unless you're debugging or troubleshooting.**

14. (Optional) If you want to change the artifact file's name, in **Artifact**, enter the new name. By default, it is `extension.vx`.
15. (Optional) If you configured the development packaging job to overwrite the extension's default version defined in the `visual-application.json` file, specify the same version in **Extension Version**.
16. Click the **After Build** tab in the Job Configuration window.
17. From **Add After Build Action**, select **Artifact Archiver**.

18. In **Files to archive**, enter the build artifact name.

You can use wild characters. For example, * .vx. Make sure that you include the path for the Application Extension build artifact.

If you entered a value in the **Target Directory** field in **Copy Artifacts** for the deployment job, it is considered to be a subdirectory of the workspace and should be part of the artifact's path in the **Build Artifact** field.

19. If you want to discard the build's old artifacts, click **Settings** . In the **General** tab, select the **Discard Old Builds** check box and specify the discard options.
20. Click **Save**.

Create the Production Deployment Build Job

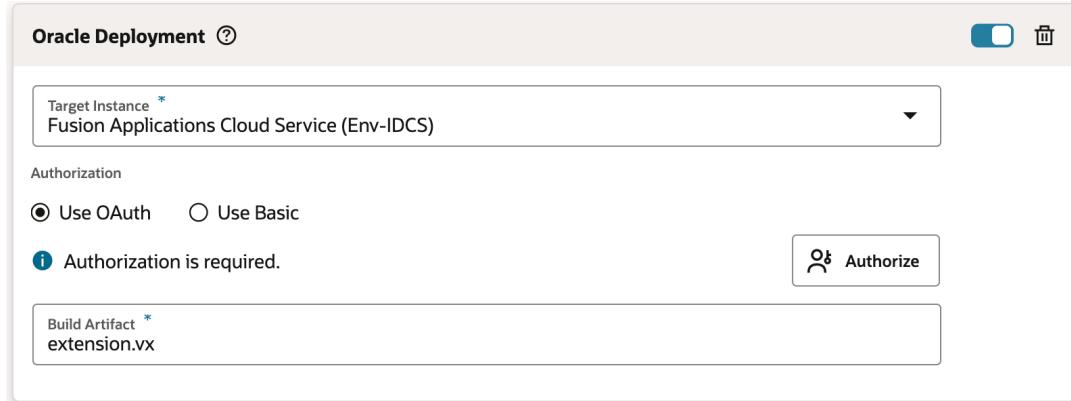
The deployment job deploys the extension's artifact that was generated in the packaging job to the Oracle Cloud Application's production instance. Before you create the job, make sure you have credentials that VB Studio can use to access the Oracle Cloud Application's PROD instance.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. In **Template**, select the **System Default OL7 for Visual Builder** template.
6. Click **Create**.
7. Click **Configure** .
8. Click the **Before Build** tab.
9. From **Add Before Build Action**, select **Copy Artifacts**.
10. In **From Job**, select the packaging job that generated the extension's artifact.
11. In **Which Build**, select one of the following:
 - Last successful build (default)
 - Last keep forever build
 - Upstream build in this pipeline instance
 - Specified by permalink
 - Specific build
 - Specified by build parameter

Depending on what you select, you may be prompted to select which permalink, build, or build parameter you want to be used.

12. Leave other fields with their default or empty values.
13. Click the **Steps** tab.
14. From **Add Step**, select **Oracle Deployment**.

This image shows the Oracle Deployment build job page that's been partially filled in.



15. In **Target Instance**, select the environment with the Oracle Cloud Application's production instance.
16. In the Authorization section, select **Use OAuth** or **Use Basic** to specify the type of authorization used to run this build step.

If you selected **Use OAuth**, before authorization, the **Authorization is required** message indicates that this build step needs to be authorized, either by clicking the **Authorize** button or by running the build manually and entering the credentials when you are prompted for them at that point. After authorization, the **Authorization has been provided** message and the **Renew Authorization** button are displayed.

If you selected **Use Basic**, in **Username** and **Password**, enter the credentials of an IDCS user who is not only an Oracle Cloud Applications user, but one who can connect and undeploy from the Oracle Cloud Application's production instance. These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.

17. If you changed the archive artifact files default file names in the production packaging job, in **File names**, select **Use custom file names**.

In **Sources and Build Artifact**, enter the same file names (with the path) that you specified in the packaging job.

18. Click **Save**.

Note:

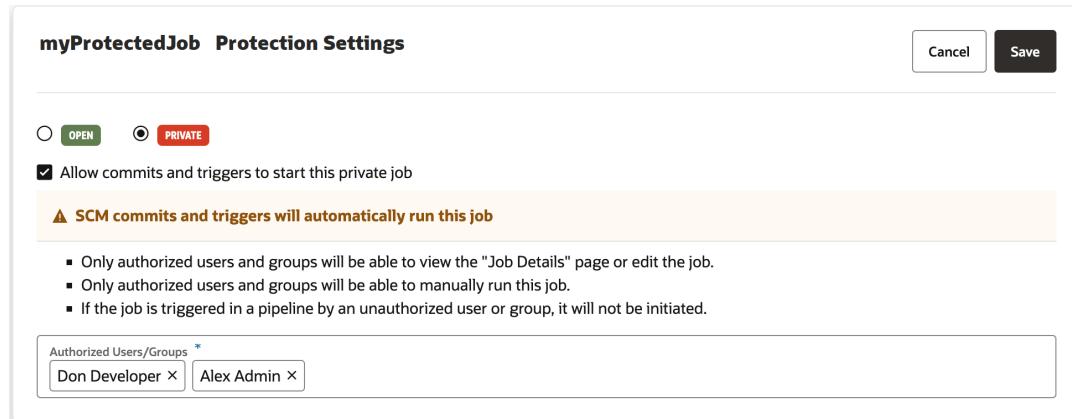
If you develop an extension on, say, 23D in your Test environment, then want to deploy the extension to your 23C Prod environment, you'll have to wait until your Prod instance has been upgraded to 23D before you can deploy successfully. In most cases, there shouldn't be more than a two week gap between prod upgrades.

Configure a Job's Privacy Setting

The project owner can mark a job as private to restrict who can see or edit a job's configuration, or run its build:

1. In the left navigator, click **Project Administration** .
2. Click **Builds**.
3. Click the **Job Protection** tab.

4. From the jobs list, select the job.
5. Select the **Private** option.
6. Select the **Allow commits and triggers to start this private job** checkbox if you want SCM commits and triggers to be able to automatically run this job.



With the checkbox selected, periodic triggers will run any job or pipeline, including private jobs set to allow commits and triggers to start the private job. If a private job with this option selected is included in a pipeline and a non-authorized user attempts to run the pipeline manually, the private job won't run but periodic triggers and SCM commits will.

Leave the checkbox unselected if you don't want the job to be started when it is triggered by an SCM commit or timer.

 **Note:**

Best Practice:

If you use the checkbox to enable the protected build to be triggered by an SCM commit, you need to protect the branch that the build job is tied to. If you don't do this, anyone can trigger the protected build by making a commit to trigger the build.

7. Click in the **Authorized Users/Groups** field to view a list of **Users** and **Groups** that have permission to access this project.

From the list, select one or multiple users and groups. Don't forget to add yourself.

The screenshot shows a list of authorized users and groups. At the top, there are two buttons: 'Alex Admin' and 'Don Developer'. Below them, under 'Groups', is 'QA-Group (2)'. Under 'Users', there are three entries: 'Mary Jane', 'Tanya Testsuite', and 'Tina Testsuite', each with a small user icon next to their name.

8. Click **Save.**

You can see if a job is private from several places in the VB Studio user interface. A private job is indicated by a **Lock** icon:

- In the jobs list found on the **Project Administration** tile's **Builds** page's **Job Protection** tab, to the right of each protected job's name.
- In the **Private** column on the **Builds** page's **Jobs** tab.
- In the jobs shown in the the **Builds** page's **Pipelines** tab.

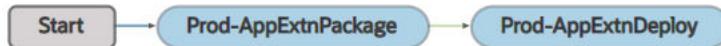
An unauthorized user can't run a private build job manually, or through a pipeline, or via an SCM/periodic trigger.

Create and Configure the Production Pipeline

To ensure the deployment job runs automatically after the packaging job, create a pipeline and set the dependency.

1. In the left navigator, click **Builds**.
2. Click the **Pipelines** tab.
3. Click **+ Create Pipeline**.
4. In the Create Pipeline dialog box, in **Name** and **Description**, enter a unique name and description.
5. Click **Create**.
6. On the Pipeline Configuration page, right-click the **Start** node and select **Add New Start jobs**.
7. Click in the **Select new on success job(s)** field, select the packaging job and click **Save**.
8. Right-click the packaging job and select **Add**, then **Add New On Success Jobs**.
9. Click in the **Select new on success job(s)** field, select the deployment job, and click **Save**.

Here's an example of the final pipeline:



10. Click **Save**.

Run Production Pipelines

Before you run production build jobs or a production pipeline, make sure that all code changes have been pushed to the production branch and there are no open merge requests.

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. In the production pipeline's row, click the **Actions** *** menu and select **Run Pipeline**.

After a successful build, you can see the deployed application's link in the **Environments** page's **Deployments** tab. You should also notify the Oracle Cloud Applications administrator that the extension has been pushed to the Oracle Cloud Application's production instance.

Set Up the Project to Deploy to Other DEV and TEST Instances

If you've followed this chapter to this point, you've configured VB Studio to deploy extensions to your primary Oracle Cloud Applications Development environment. This section explains how to set up other Oracle Cloud Application instances, such as DEV and TEST, where your team members may want to test their extensions before deploying to PROD.

This involves creating an environment for each instance you want to deploy to, as well as configuring jobs and pipelines to package up and deploy extensions to those target instances.

Before you proceed, make sure that the Oracle Cloud Applications instances are up and running.

Tip:

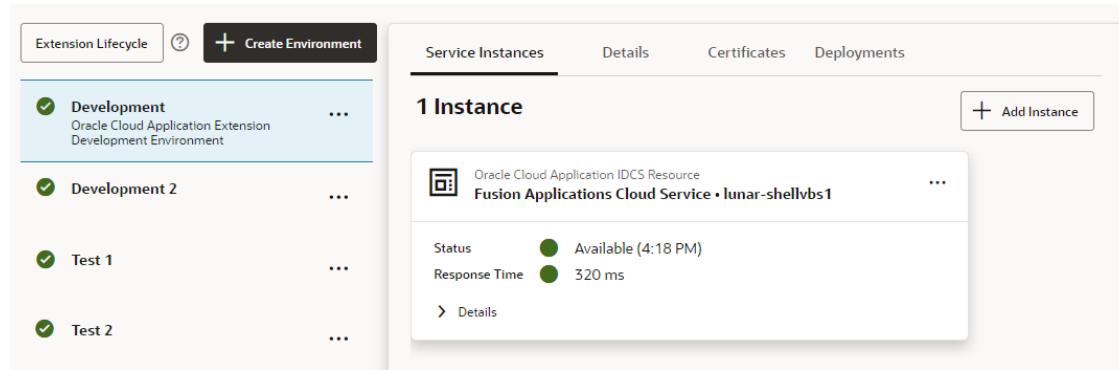
For a simpler interface where you can manage the lifecycle of all your extensions, use the Manage Extension Lifecycle page. From this single page, you can deploy extensions to DEV and TEST instances without the need to create build jobs and pipelines, as long as those instances have an authentication method of OAuth.

Add an Oracle Cloud Application Instance to an Environment

To deploy an extension to an Oracle Cloud Applications instance, you must create a VB Studio environment and add the instance to it. You can add only one Oracle Cloud Applications instance to each environment.

Each Oracle Cloud Applications DEV and TEST instance has its own identity domain. This means you'll need the target instance's URL—as well as a user's credentials that can access the instance—to set up the environment. These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.

Here's what your Environments page might look like after you've added three new environments, one for another DEV instance and two for some TEST instances:



To create a new VB Studio environment:

1. In the VB Studio navigation menu, click **Environments**.
2. Click **+ Create Environment**. In **Environment Name** and **Description**, enter a unique name and description, then click **Create**.
3. In the Service Instances tab, click **+ Add Instance**.
4. In the Add Service Instances dialog box, select the **Oracle Cloud Applications** option button.
5. Under Authentication Method, select the **Application Credentials** option.
6. In **Base URL**, enter the Oracle Cloud Application's base URL.
7. In **Instance Name**, if required, update the instance's display name. The name will be displayed in the Service Instances tab.
8. In **Username** and **Password**, enter the credentials of a user who can access the Oracle Cloud Applications instance.
These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.
9. Under Authorization Type, select **OAuth 2.0 or Basic**.
10. Click **Add**.
11. If you chose OAuth 2.0 as the Authorization Type, you may be prompted to authorize IDCS to acquire OAuth tokens for accessing the target instance. Click **Authorize** when prompted, then sign in to IDCS with credentials to access the Oracle Cloud Applications instance.

 **Note:**

OAuth tokens (access and refresh) are typically cycled during regular use. If a token expires during an extended period of inactivity, like when you're away on vacation, click **Actions ...** and select **Renew OAuth Access** to restart the OAuth workflow.

Note that the username under the Details section of the connected Oracle Cloud Applications endpoint shown in the **Environments** page's **Instances** tab displays the user that is logged in to the Oracle Cloud Applications instance and who creates the Refresh Token that the connection is using is shown. The details don't show the VB Studio username, which isn't relevant for a 3-legged OAuth Environment connection.

If the newly added instance stays in the **Unknown** status for some time, it usually indicates that the IDCS Application provisioning may have failed. VB Studio added the Oracle Cloud Applications instance, but can't access it. In this case, click **Actions** *** and select **Remove** to remove the Oracle Cloud Applications instance from the environment, and then click **Add** to add it again.

Create and Configure Deployment Build Jobs and Pipelines

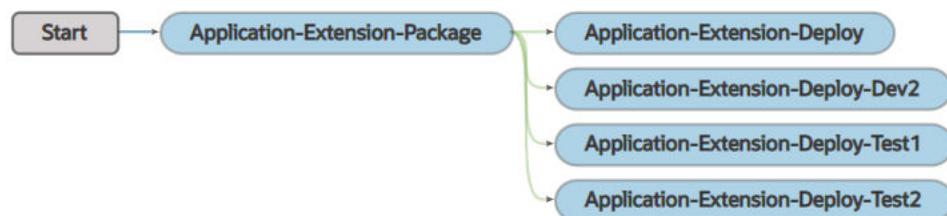
To deploy extensions to other Oracle Cloud Applications instances, you need to set up or create a packaging job, one or more deploy jobs, and a pipeline. Specifically:

- A packaging job packages the extension and creates a build artifact. Instead of creating new packaging jobs, you can use the `extensionName-Package` job that was created with the project. (If you created this project manually, the package job is called **Application-Extension-Package**.) This job packages the extension and creates a build artifact from the same branch your team members used to deploy to the primary Oracle Cloud Applications Development instance. Remember, this packaging job is configured to trigger a build on every SCM commit.
- Create one deploy job for each Oracle Cloud Applications instance you plan to deploy to. You'll also need a user's credentials who can deploy to each Oracle Cloud Applications instance. To create the job, see [Create a Deployment Build Job](#).
- After you've created deployment jobs, create pipelines to run the package job and deployment jobs in sequence. You can create either of these pipelines:
 - A pipeline to deploy the extension to the primary and other Oracle Cloud Applications instances.

Instead of creating a new pipeline, configure the existing `extensionName-Package and Deploy` pipeline to deploy to other instances as well. (If you created this project manually, this pipeline's name is **Application-Extension Package and Deploy**.) See [Configure the Default Pipeline to Deploy to Other Oracle Cloud Applications Instances](#). Remember, the default pipeline runs automatically when a code change is pushed to the `main` branch. After you've configured the pipeline, it deploys the extension to all development and test instances on every SCM commit to the `main` branch.

After creating this pipeline, you'll have one pipeline in your project that deploys to all DEV and TEST instances.

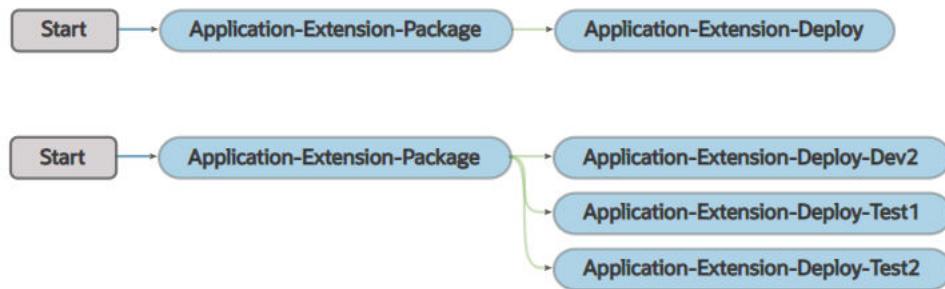
Here's an example of what this pipeline could look like:



- A pipeline to deploy the extension to other DEV and TEST Oracle Cloud Applications instances manually. See [Create and Configure a Pipeline to Deploy to Other Oracle Cloud Applications Instances](#). Create this pipeline if you don't want to deploy the extension to other instances on every SCM commit, but only after your team members have validated the extension on the primary development instance.

After creating this pipeline, you'll have two pipelines in the project: 1) Your primary pipeline, which deploys to the primary development instance, and 2) this pipeline, which deploys to other DEV and TEST instances.

Here's an example of what these pipelines could look like:



Create a Deployment Build Job

The deployment job deploys the extension's artifact that was generated in the default packaging job to the Oracle Cloud Application's instance. Before you create the job, make sure you have credentials that VB Studio can use to access the instance.

1. In the left navigator, click **Builds**.
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. Select the **Copy From Existing** check box.
6. From the **Copy From** drop-down list, select the **Application-Extension-Deploy** job.
7. In **Template**, make sure that the **System Default OL7 for Visual Builder** template is selected.
8. Click **Create**.
9. From the Job Configuration screen, click the **Steps** tab.
The Oracle Deployment screen is displayed.
10. In **Target Instance**, select the environment with the target Oracle Cloud Applications instance.
11. In the Authorization section, select **Use OAuth** or **Use Basic** to specify the type of authorization used to run this build step.
 - If you selected **Use OAuth**, the **Authorization is required** message indicates that this build step needs to be authorized, either by clicking the **Authorize** button or by running the build manually and entering the credentials when you are prompted for them at that point. After authorization, the **Authorization has been provided** message and the **Renew Authorization** button are displayed.
 - If you selected **Use Basic**, in **Username** and **Password**, enter the credentials of a user who can deploy to the Oracle Cloud Applications instance. These credentials must belong to a local IDCS user, not a federated identity, and must not require multi-factor authentication.

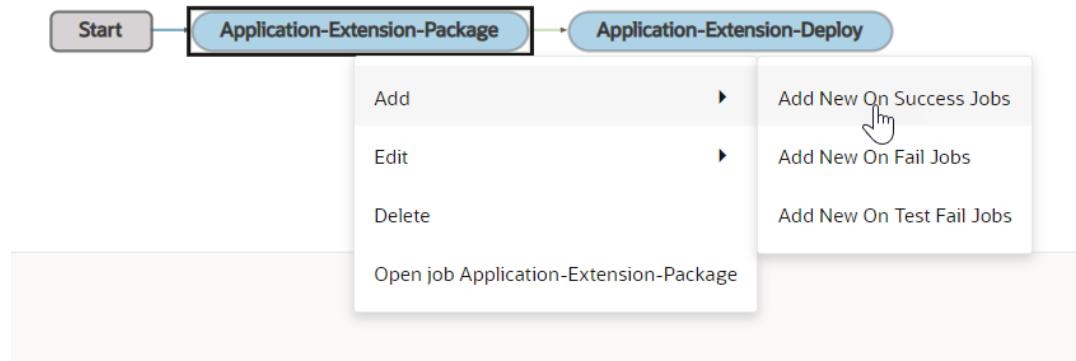
12. The **Build Artifact** field shows the same artifact name that was used in the packaging build step.
13. Click **Save**.

Configure the Default Pipeline to Deploy to Other Oracle Cloud Applications Instances

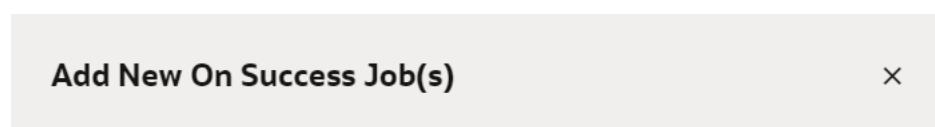
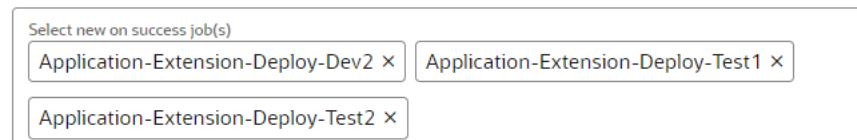
If you want to deploy to other DEV and TEST instances along with your primary Oracle Cloud Applications instance, you can configure the default pipeline to deploy to other instances as well.

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. In the `extensionName-- Package and Deploy` pipeline row, click the **Actions** menu and select **Configure Pipeline**. (If you created the project manually, the pipeline name is **Application Extension - Package and Deploy**.)
4. Right-click the `extensionName-Package` job and select **Add**, then **Add New On Success Jobs**.

Here's an example:



5. Click in the **Select new on success job(s)** field, select the jobs that you want to add from the list, and click **Save**.

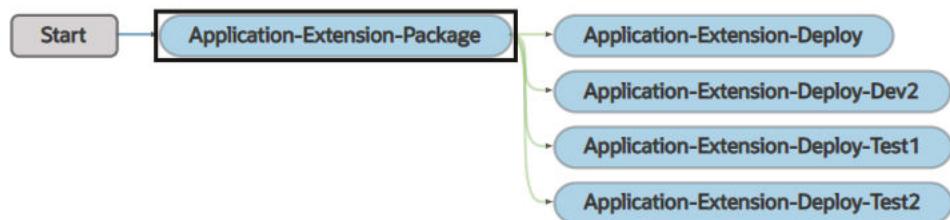



The screenshot shows the 'Add New On Success Job(s)' dialog with the following content:

- Text input field: 'Select new on success job(s)'
- List of selected jobs:
 - Application-Extension-Deploy-Dev2 ×
 - Application-Extension-Deploy-Test1 ×
 - Application-Extension-Deploy-Test2 ×

Cancel **Save**

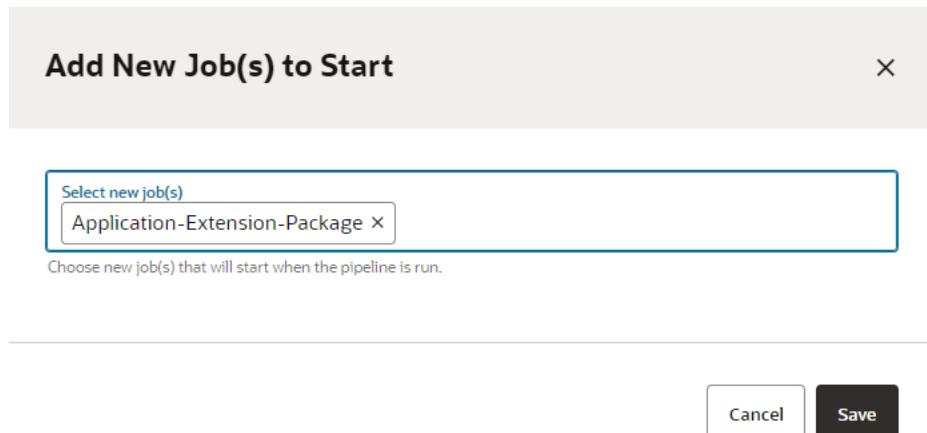
Here's an example of the final diagram:



6. Click **Save**.

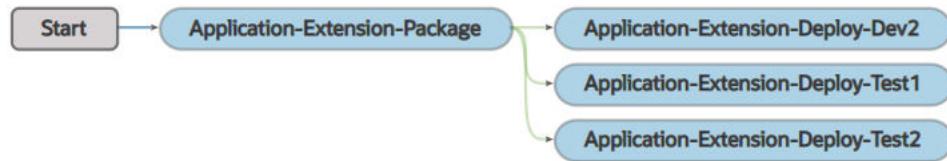
Create and Configure a Pipeline to Deploy to Other Oracle Cloud Applications Instances

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. Click **+ Create Pipeline**.
4. In the Create Pipeline dialog box, in **Name** and **Description**, enter a unique name and description.
5. Deselect the **Auto start when pipeline jobs are built externally** check box.
6. Click **Create**.
7. On the Pipeline Configuration page, right-click the **Start** node and select **Add New Start jobs**.
8. Click in the **Select new on success job(s)** field, select **extensionName-Package**, and click **Save**.



9. Right-click the **extensionName-Package** job and select **Add**, then **Add New On Success Jobs**.
10. Click in the **Select new on success job(s)** field, select the jobs that you want to add from the list, and click **Save**.

Here's an example of a finished pipeline:



- 11.** Click **Save**.

Run the Pipeline

The test build pipeline runs automatically when a commit is pushed to the Git repository's test branch specified in the packaging job.

If you want to run the pipeline manually:

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. For the pipeline you want to run, click **Build** .

To monitor the pipeline, click the pipeline's name on the **Builds** page. After a successful build, you'll find the deployed application's link in the **Deployments** tab of the **Environments** page.

If you want to run a job's build manually, open the job's details page and click **Build Now**. You can monitor its build on the job's details page. To view the latest build log of a job, open the **Builds** page, click the job's name, and then click **Build Log** .

Delete an Extension

The method you use to delete an extension depends on if you want to delete the *entire* extension or only an extension *version*. Delete an entire extension using the Manage Lifecycle Extension page. To delete an extension version, the method is different depending on if the extension is deployed to an Oracle Cloud Applications instance that's in the same identity domain as VB Studio, or a different one.

- You delete an extension version *manually* from its environment's **Deployments** tab, if the extension is deployed to an Oracle Cloud Applications instance that's in the same identity domain as VB Studio (by default a TEST pod, although it could be a DEV pod or a different TEST pod if you requested that Oracle re-associate VB Studio to a different pod). You could also configure a build job to delete it.
- If the extension is deployed to a different identity domain (typically your PROD instance) and you want to delete an extension version, you must configure a build job to delete it.

Delete an Extension Manually

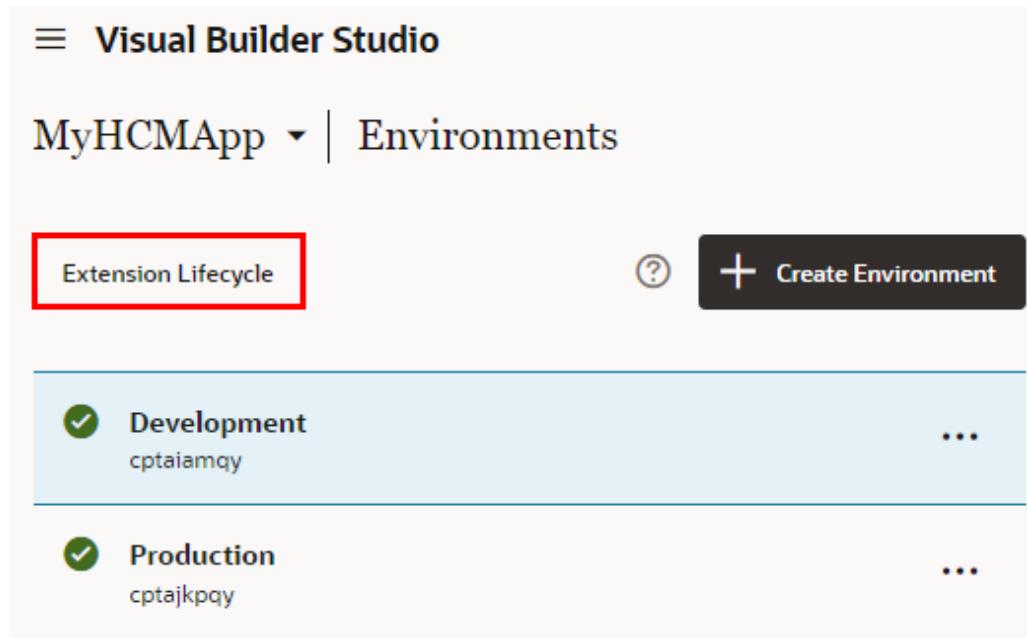
Use the Manage Extension Lifecycle page to delete extensions deployed from any project and to any Oracle Cloud Applications instance, provided the instance it's deployed to has an authentication method of OAuth.

If the extension was deployed to an Oracle Cloud Applications instance whose authentication method is something other than OAuth, you must either [use its environment's Deployments tab](#) to delete the extension or [configure a build job](#) to delete it.

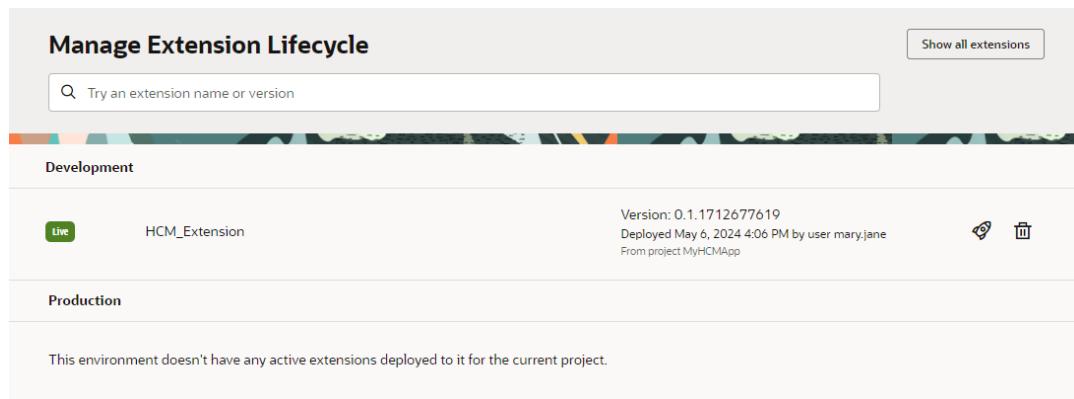
To delete an extension using the Manage Extension Lifecycle page:

1. Access the Manage Extension Lifecycle page:

- From the menu in the Designer's header, click **Extension Lifecycle**.
- Or, from the VB Studio left navigator, click **Environments**, then click **Extension Lifecycle**. You won't see this option if you haven't yet added an Oracle Cloud Applications instance to your environment.



You should now see the Manage Extension Lifecycle page, which lists all the environments that are available. You can work with an environment on this page as long as the authentication method for its Oracle Cloud Applications instance is OAuth.



Under each environment, you'll see a list of the extensions already deployed. Each entry includes the extension name plus additional details, such as extension version, deployment date, and project name. The Manage Extension Lifecycle page can show all extensions deployed from the current project or across projects. Use **Show all extensions** and **Only show extensions for my project** to toggle between both views.

2. To delete an extension from an Oracle Cloud Applications instance, find the extension that



you want to delete and click .

Tip:

Before deleting an extension from your PROD instance, delete the extension from the DEV instance (or TEST instance, if applicable) and make sure there aren't any adverse effects.

Delete an Extension Version Manually

You can delete a *version* of an extension from its environment's **Deployments** tab, if an extension is deployed to the Oracle Cloud Applications instance in the same identity domain as your VB Studio instance.

1. In the left navigator, click **Environments** .
2. Select the **Development** environment where the extension is deployed.
3. Click the **Deployments** tab.
4. Expand the base application's name.
5. Click **Actions** *** and select **Delete** next to the extension you want to delete.

Note that there can be only one *active* extension version at a time. The recommended best practice is to avoid deleting the active extension version because, if you delete it, the extension will become inactive as if it wasn't installed at all. You can, however, delete inactive versions without impacting the currently active deployed extension.

6. In the confirmation dialog box, click **Delete**.

For an extension deployed to an Oracle Cloud Applications instance in the same identity domain as VB Studio, you can also use a build job to delete it, if preferred.

Configure a Job to Delete an Extension

To delete an extension *version* that's deployed to your PROD Oracle Cloud Applications instance (or to any Oracle Cloud Applications instance in a different identity domain from VB Studio), configure a build job and run it.

Note that there can be only one *active* extension version at a time. The recommended best practice is to avoid deleting the active extension version because, if you delete it, the extension will become inactive as if it wasn't installed at all. You can, however, delete inactive versions without impacting the currently active deployed extension.

For an extension deployed to an instance in a different identity domain, you can also use the Manage Extension Lifecycle page to delete it, but keep in mind that this deletes the entire extension. You must use a build job, if:

- You want to delete an extension *version*, not an entire extension
- The extension is deployed to an Oracle Cloud Applications instance that has an authentication method of something other than OAuth

Before you configure and run the job, delete the extension from the DEV instance (or TEST instance, if applicable) and make sure there aren't any adverse effects. For example, let's

assume you have an attribute that's hidden in both the extension's business object and the user interface. After you delete the extension, the user interface shows the attribute that is still hidden in the business object. This may cause an error.

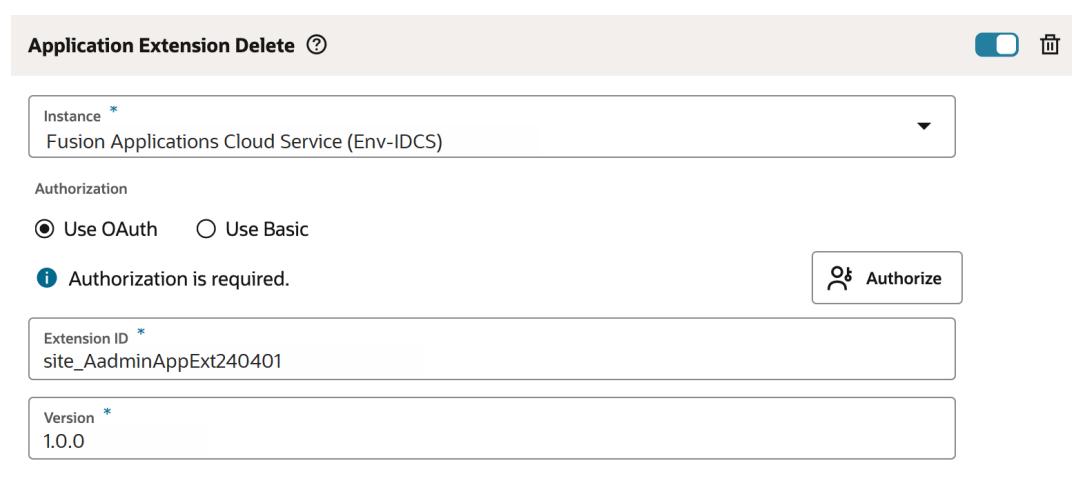
To configure the job, make sure you have valid credentials for the Oracle Cloud Application's instance where the extension is deployed.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. In **Template**, select **System Default OL7 for Visual Builder**.
6. Click **Create**.

The Job Configuration page opens.

7. Click the **Steps** tab.
8. From **Add Step**, select **Application Extension**, and then select **Delete**.

This image shows the Application Extension Delete build job page that's partially filled in.



Application Extension Delete 

Instance *
Fusion Applications Cloud Service (Env-IDCS)

Authorization

Use OAuth Use Basic

 Authorization is required. 

Extension ID *
site_AdminAppExt240401

Version *
1.0.0

9. In **Instance**, select the Oracle Cloud Applications instance where the application is deployed.
10. In the Authorization section, select **Use OAuth** or **Use Basic** to specify the type of authorization used to run this build step.
 - If you selected **Use OAuth**, the **Authorization is required** message displayed before authorization indicates that this build step needs to be authorized, either by clicking the **Authorize** button or by running the build manually and entering the credentials when you are prompted for them at that point. After authorization, the **Authorization has been provided** message and the **Renew Authorization** button are displayed.
 - If you selected **Use Basic**, in **Username** and **Password**, enter the credentials of an IDCS user who is not only an Oracle Cloud Applications user, but one who can access the Oracle Cloud Application's production instance and undeploy from it. These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.

11. Enter the extension's identifier in **Extension ID**.

 **Tip:**

The identifier is displayed in a column of the same name on the **Deployments** tab of the environment where the extension is deployed. The extension name can be viewed in a tooltip if you hover over the identifier. You can also see the extension name from Settings in the top right hamburger menu. This option displays the visual view of the file.

12. Enter the extension's version in **Version**.

You can find the Extension Manager version on the **Deployments** tab of the environment where the extension is deployed. The Application Extensions section lists deployments for the current project only. Optionally, you can use the **Show Previewed/Shared versions** checkbox to display extensions that were deployed through the Designer, not through a build step.

13. Click **Save**.

14. To run a build, click **Build Now**.

Manual Administrator Tasks

VB Studio is designed to make accessing the product and creating a project quick and easy, simply by clicking the **Edit Page in Visual Builder Studio** link in an Oracle Cloud Application. However, there are times when you may want to complete these tasks manually.

Manually Create a Project for Extensions

The easiest way to create a project for extensions is through the **Edit Page in Visual Builder Studio** link in an Oracle Cloud Application. However, you do still have the option to create a project manually. Ideally, you'll create one project for each Oracle Cloud Application you plan to extend, such as Sales, Journeys, and so on.

To create an extension project manually:

1. On the Organization page, click **+ Create**.
2. On the **Project Details** page of the New Project wizard, in **Name** and **Description**, enter a unique name and description for the project.
3. In **Security**, select the project's privacy setting:
 - a. Select **Private** to restrict access to project members only.

The **Discoverable** checkbox, selected by default for private projects, allows organization members that aren't org admins or project members to see basic information, such as name and owner contact information, about your private project. Private projects that aren't discoverable won't be exposed to non-members.

 **Note:**

If you want this project to be visible to users when they use the **Edit Page in Visual Builder Studio** option to make changes to their Oracle Cloud Applications page in VB Studio, mark it as Discoverable. This gives users an opportunity to select a project that's already associated with the page they want to modify (or they can create a new project, if they prefer).

- b. Select **Shared** to make the project code, wiki docs, tasks, and builds available to anyone inside your organization.
4. In **Preferred Language**, specify the language for the email notifications your project users will receive.

You can change the language in which the user interface appears in your user preferences.

5. Click **Next**.
6. On the **Template** page, select the **Application Extension** template, then click **Next**.

When you create a project, VB Studio also creates a workspace and an extension for you. Use these to create your own extension, if you like, or let one of your team members use it.

If you don't see the **Application Extension** template, you're in an OCI region that doesn't support it. Use these steps to create your project instead of the steps shown here:

- a. Create an empty project.
- b. Open the Environments page and create a new environment.
- c. Add your Oracle Cloud Applications instance to the environment.
- d. Open the Workspaces page, click **New**, then select **New Application Extension**, and fill in the fields on each page, clicking **Next** to advance to the next page or **Create** after completing the last page.
- e. Add packaging and deployment jobs, then add them to pipelines, as needed.
7. On the **Project Properties** page:
 - a. In **Extension Name**, if necessary, change the extension's name. Make sure that the extension name is unique across all the extensions created for this Oracle Cloud Application. By default, the name is `<project_name> Extension`.
 - b. In **Extension Id**, if necessary, change the extension's ID. Make sure that the ID is unique across all the extensions created for this Oracle Cloud Application. By default, the ID is `site_<extension_name>`.
 - c. In **Workspace Name**, if required, change your private workspace's name. By default, it is `Workspace1`.

A workspace contains all the artifacts that you need to develop extensions, including a clone of this project's Git repository—and the branch—containing the source files. To learn more about workspaces, see [What Is a Workspace?](#) Again, you don't have to use this workspace to develop an extension; it's created purely for your convenience. (When a team member accesses VB Studio from their Oracle Cloud Application, a workspace is automatically created for the user if one doesn't already exist.)

- d. In **Oracle Cloud Applications Development Instance**, VB Studio preselects your current identity domain's Oracle Cloud Applications instance and uses it as the development instance to develop extensions.

If there are multiple Oracle Cloud Applications instances in your identity domain, you'll be asked to choose. Make sure to select the Oracle Cloud Applications instance you want to use to develop extensions. Make sure that the selected instance has a VB Studio instance provisioned and also has the Oracle Cloud Application you want to extend.

- e. From **Base Application**, select the Oracle Cloud Application you want to extend in this project.
- f. In **Git Repository Name**, change the Git repository's default name, if required. When a team member creates a workspace, a copy of this repository is made for the user's local changes. When the user executes the Git Push command, the changes are copied to the project's repository (also known as the "remote repository", in VB Studio terms).
- g. In **Working Branch Name**, if required, change the workspace's working branch name. By default, it is `branch1`.

Follow your organization's guidelines for creating and managing Git repository's branches. When the project is provisioned, the Git repository's `main` branch is ready for you to start working on an extension, should you opt to do so. Whenever someone creates a workspace, VB Studio creates a copy of the main branch, renames it with the specified name, and uses it as the workspace's working branch. This documentation assumes that you'll continue to use `main` as the development branch and create a separate branch for production.

8. On the Team page:
 - a. Click **Add Members** and select Oracle Cloud Applications users or groups to add to the project, from the list displayed, if you know they may work in this project.
 - b. Select the membership (Project Owner, Developer Full Access, Developer Limited Access, or Contributor) that the members you're adding will have in the project: See [What Are Project Memberships?](#) for more information about each membership.
 - c. Click **Add**.
 - d. Repeat substeps a, b, and c for different users and groups with various membership types, if needed.
9. Click **Finish**.

After the project is provisioned, the **Project Home** page opens where you can see a summary of the project's provisioning activities; default environment; and Git, Maven, and NPM repositories. Review the activity feed and the **Environments** box for any errors.

Access VB Studio from Oracle Cloud Applications

While the **Edit Page in Visual Builder Studio** link in an Oracle Cloud Application is the easiest way to access VB Studio, there is an alternative approach:

1. In your Oracle Cloud Applications instance, click **Navigator** .
2. In the navigation menu, under **Configuration**, select **Visual Builder**.

← Show More

Home

Me

My Team

Tools

Configuration

Visual Builder

Sandboxes

Migration

Appearance

Structure

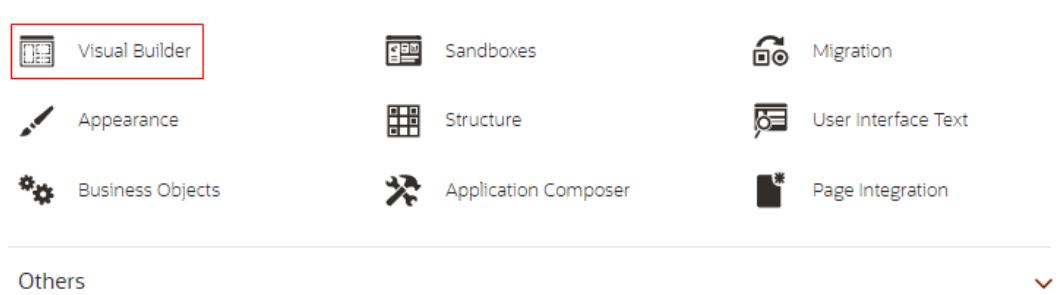
User Interface Text

Business Objects

Application Composer

Page Integration

Others



The VB Studio **Organization** page opens, which displays all the projects you're a member of, as well as your favorite projects, the projects you own, and all the shared projects in your organization.

Set Up VB Studio for Developing Visual Applications

This chapter tells you how to set up Oracle Visual Builder Studio (VB Studio) so that your users can create *visual applications*.

A visual application is a responsive application developed using VB Studio's browser-based development environment. You deploy a visual application to a Visual Builder instance or to a Visual Builder instance available in Oracle Integration. The Visual Builder instance must be version 19.4.3.1, or later.

Assuming you have already created a VB Studio instance, the setup process for visual applications involves three phases:

- [Set Up Visual Builder Studio for Visual Applications](#)
- [Set Up the VB Studio Project](#)
- [Set Up the Project to Deploy for Production](#)

Set Up Visual Builder Studio for Visual Applications

Here's a summary of how to set up VB Studio for developing visual applications:

To perform this action:	See this:	Why do I need to perform this action?
1. Get access to Visual Builder instances.	Get Access to Visual Builder Instances	To deploy an app, your users need credentials that allow them to access the Visual Builder instance (deployment target). If an instance isn't available for them, you must create it.
2. Get the required IDCS roles assigned to you.	Get the Required IDCS Roles	To create and set up the VB Studio instance, you must be assigned some specific IDCS roles.
3 (Optional) Configure VB Studio to run build jobs and pipelines.	Configure VB Studio to Use the Free VM Executor for Build Jobs and Pipelines	To run build jobs and pipelines, you must configure VB Studio to connect to an OCI account or the built-in free account.

Note:

If you're developing a visual application using an instance of VB Studio that was provisioned in your Oracle Cloud Applications account, follow the procedures in [Create and Set Up a Project for Development \(Different Identity Domain\)](#).

Get Access to Visual Builder Instances

You can deploy a visual application to a standalone Visual Builder instance or to a Visual Builder instance that's part of Oracle Integration. You cannot deploy to Visual Builder Studio itself.

If you don't already have an instance to deploy to:

1. [Create a Visual Builder Service Instance](#).
To create an instance, you'll need the IDCS_VISUALBUILDER_ENTITLEMENT_ADMINISTRATOR service role.
2. [Add an IDCS User to Your Cloud Account](#) and assign the user a Visual Builder role to connect and deploy applications.

You'll also need to configure the instance once it's created. If you're a Visual Builder runtime administrator, sign in to the Visual Builder runtime to complete these tasks, as described in *Administering Oracle Visual Builder, Gen 2*:

- [Access Instance Settings](#)
- [Configure Security Options for Applications](#)
- [Set Page Messages for Access Denied Errors](#)
- [Allow Other Domains Access to Services](#)
- [Switch to Your Own Oracle DB Instance](#)
- [Reset an Expired Password or ATP Wallet for Your Oracle DB Instance](#)
- [Manage Self-signed Certificates](#)
- [Configure Support for a Custom Domain](#)

It's important to keep these things in mind before you deploy a visual application to a Visual Builder instance:

- The Visual Builder instance must be version 19.4.3.1 or later.
- To ensure that business objects work properly, your Visual Builder administrator must manually add the VB Studio hostname to the allow list for each Visual Builder instance. For more information, see [Add an Allowed Origins Domain](#).
- If you need to create a VB Studio instance to deploy to, create it in the same Oracle Cloud account and identity domain as the Visual Builder Studio instance you are developing the visual application with. When you need to deploy your app for testing or to go live, you'll need to create instances for those environments using working credentials in different identity domains. See [Create the VB Studio Instance](#).
- Follow your organization's guidelines to create and set up Visual Builder instances for deployment targets. Your guidelines may suggest that you create instances for different deployment environments, such as development, integration, test, pre-stage, stage, pre-production, and production. You can create these instances in different identity domains or in a common identity domain.
This documentation assumes that you have created and set up Visual Builder instances you can deploy to in your development and production environments, where each instance is in a different identity domain.

To deploy your visual application to Visual Builder development and production instances, you'll need credentials to access them. You'll be prompted for these credentials when you set up the instances or when you perform a manual deployment to those instances.

Get the Required IDCS Roles

Make sure that you have the necessary identity domain roles you'll need to set up VB Studio on an OCI tenancy:

You must be assigned this role:	To ...
Identity Domain Administrator or User Administrator	Add users and assign IDCS roles.
DCS_INSTANCE_ENTITLEMENT_ADMINISTRATOR OR	Create the VB Studio instance.
DEVELOPER_ADMINISTRATOR	Set up VB Studio. After you're assigned the role, you're considered VB Studio's <i>Organization Administrator</i> .
OCI_Administrator (OCI Administrator)	Set up OCI compartments and buckets, which are required to set up the VB Studio build system.

Set Up the VB Studio Project

After setting up VB Studio, follow these steps to create and set up a project for developing your visual applications.

To perform this action:	See this:	Why do I need to perform this action?
1. Create a project for visual applications and set up the project for development.	Create and Set up a Project	To develop a visual application, you must create a VB Studio project and set up the project for your developers.
2. Add users to the project.	Add Users to the Project	To allow your team members to access the visual application project, invite them to join the project.
3. (Optional) Configure advanced settings.	Optional Configuration	These settings are optional, depending on your configuration.
4. Developers can now create visual applications.	Develop Your Application	This is the main development stage for your visual app.

Create and Set up a Project

A project gathers all the resources you need for developing software—in this case, visual applications. After you create the project, you must set it up.

The procedures that you use to create and set up the project depend on whether your Visual Builder instance is in the same identity domain as your VB Studio instance or in a different domain.

Select the appropriate procedures for your situation:

- If the Visual Builder instance is in a different identity domain than the VB Studio instance, see [Create and Set Up a Project for Development \(Different Identity Domain\)](#).
- If the Visual Builder instance is in the same identity domain as the VB Studio instance, see [Create and Set Up the Project for Development \(Same Identity Domain\)](#).

Create and Set Up a Project for Development (Different Identity Domain)

To develop visual applications using a VB Studio instance that's in a different identity domain than your Visual Builder instance (for example, a VB Studio instance provisioned in your Oracle Cloud Application account), you'll need to add a Visual Builder instance to your VB Studio project.

Follow these steps to create and set up the project, configure the deployment job, and add users to the project:

- After you've set up VB Studio to develop visual applications, follow the steps in the this table to set up the project, configure the deployment job, and add users to the project:

Action	How to
1. Create an empty VB Studio project.	<ol style="list-style-type: none">Sign in to VB Studio. See Access VB Studio from the Oracle Cloud Home Page.On the Organization page, click + Create.On the Project Details page, enter a unique name and description for the project.In Security, select the project's privacy setting. A private project is accessible to invited users only. Users who aren't invited can't access it or make changes to it. You can invite users after creating the project. A shared project is accessible to all users of the organization. Any user can view the source code, create or update issues, edit wiki pages, and interact with project builds. However, only invited users can make updates to the source code in Git repositories, create and run build jobs, and perform deployment operations.In Preferred Language, specify the language for the email notifications your project users will receive. You can change the language in which the user interface appears in your user preferences.Click Next.On the Template page, select the Empty Project template, then click Next.On the Properties page, select the wiki markup type, then click Finish.

After the project is provisioned, the Project Home page opens where you can see a summary of the project's provisioning activities; default environment; default workspace; and Git, Maven, and NPM repositories. Review the activities feed and the Environments box for any errors.

Action	How to
2. In the project, create an environment with your Visual Builder instance.	<ol style="list-style-type: none"> 1. In the navigation menu, click Environments , and then click + Create Environment. 2. In Environment Name and Description, enter a unique name and description, and click Create. 3. In the Service Instances tab, click Add Instance. 4. For Instance Type, select the Visual Builder, and for Authentication Method, select Visual Builder Credentials. 5. In Base URL, enter the Visual Builder instance's URL. The Instance Name is automatically filled, but you can update that if you wish. The Instance Name will be displayed in the Service Instances tab. 6. In Username and Password, enter the credentials of a user who can access the Visual Builder instance. 7. Click Add.
3. When users are ready, they can create this workspace in the project.	<ol style="list-style-type: none"> 1. In the navigation menu, click Workspaces . 2. Click New and select New Visual Application. 3. In Workspace Name, add a name for your workspace. 4. In Development Environment, select the environment you created in the previous step. 5. In Visual Application Template, don't change the default (Default VBCS Application). 6. In Git Repository, select Create new repository, and enter a repository name and working branch name.

Action	How to
4. The user that created the visual application will need to configure their deployment job.	<ol style="list-style-type: none"> 1. In your Visual Builder Studio navigation menu, click Builds . 2. Click Configure, then click Configure . 3. Click the Steps tab. 4. In Target Instance, make sure that the environment you created in Step 2 is selected. 5. In Username and Password, enter yours or a user's credentials who can connect and deploy to the Visual Builder development instance. The credentials will be used when the package and deploy build pipeline runs. 6. (Optional) Choose Use custom file names and add custom file name details. 7. (Optional) To overwrite the application's default version, specify the new version in Application Version. Leave it empty to use the version defined in the application's <code>visual-application.json</code> file. Don't deselect the Include the application version in the URL check box. 8. (Optional) In Application Profile, specify the development application profile. Leave it empty to use the application's default profile. Using application profiles, you can define different combinations of servers and security settings for each of your environments, and use them when deploying the application to an environment. To learn more, see What Are Application Profiles? 9. (Optional) To use the existing application's database, in Data Management, select Keep existing environment data. To use a clean database for the application, in Data Management, select Use clean database. 10. Click Save.
5. Add users to the project.	To allow your team members to access the visual application project, invite them to join the project. See Add Users to the Project .

Create and Set Up the Project for Development (Same Identity Domain)

To develop visual applications using a VB Studio instance that's in the same identity domain as your Visual Builder instance, you'll first need to first create a project based on the Visual Application template, then make a few configuration settings in the project so that it's ready for developing visual applications.

Here's a summary of how to create and set up a VB Studio project for development:

To perform this action:	See this:
1. Create a project.	Create a Project
2. Configure the deployment job. By default, the deployment job lacks the credentials for connecting to the target development instance, so you must specify them manually.	Configure the Deployment Job
3. Verify your credentials by running the development pipeline.	Run the Pipeline Manually
4. View the deployed visual applications.	View the Deployed Visual Application
5. Add team members to the project.	Add Users to the Project

Create a Project

If you're developing a visual application using a VB Studio instance that's in the same identity domain as your Visual Builder instance, you'll need to base your project on the Visual Application project template.

1. Sign in to VB Studio. See [Access VB Studio from the Oracle Cloud Home Page](#).
2. On the Organization page, click **+ Create**.
3. On the **Project Details** page of the New Project wizard, enter a unique name and description for the project.
4. In **Security**, select the project's privacy:
 - a. Select **Private** to restrict access to project members only.
Select the **Discoverable** checkbox to allow organization members that aren't org admins or project members to see basic information, such as name and owner contact information, about your private project. Private projects that aren't discoverable won't be exposed to non-members.
 - b. Select **Shared** to make the project code, wiki docs, tasks, and builds available to anyone inside your organization.
5. In **Preferred Language**, specify the language for the email notifications your project users will receive.
You can change the language in which the user interface appears in your user preferences.
6. Click **Next**.
7. On the **Template** page, select the **Visual Application** project template, and then click **Next**.

Tip:

If you don't see the **Visual Application** template, use these steps to create a new visual application when a Visual Application project template isn't available::

- a. Create an empty project.
- b. Open the Environments page and create an Environment that uses your VB runtime instance.
- c. Open the Workspaces page, click **New**, then select **New Visual Application**.

These steps produce the same result as using the Visual Application project template.

8. On the **Project Properties** page:
 - a. In **Git Repository Name**, change the Git repository's default name, if required.
 - b. In **Development VB Instance**, if not already selected, select the Visual Builder development instance.

If there is only one Visual Builder instance in your VB Studio's identity domain, VB Studio selects it as the development instance. If there are more than one Visual Builder instances in your identity domain, the drop-down lists all the Visual Builder instances.

If you're assigned the PaaSAdministrator IDCS role, you'll see both current identity domain's service instances and PSM Visual Builder instances in this list. If you see both, select a PSM Visual Builder instance as it provides options to control the instance.

- c. In **Visual Application Template**, select a visual application template available on the selected development instance.

By default, VB Studio uses the **Default VBCS Application** template. To select another template, click **Change Template**, select the template and click **Use Selected**.

- d. (Optional) In **Workspace Name**, if required, change your private workspace's name. By default, it is `Workspace1`.

A workspace contains all the artifacts that you need to develop visual applications, including a clone of this project's Git repository—and the branch—containing the source files. To learn more about workspaces, see [What Is a Workspace and Why Do I Need One?](#)

- e. (Optional) In **Working Branch Name**, if required, change the workspace's working branch name. By default, it is `branch1`.

When the project is provisioned, the Git repository's `main` branch contains your application's files. While creating the workspace, VB Studio creates a copy of the `main` branch, renames it with your specified name and uses it as the workspace's working branch.

- f. Click **Next**.

9. On the **Team** page:

- a. Click **Add Members** and select users or groups to add to the project, from the list displayed.

- b. Select the membership (Project Owner, Developer Full Access, Developer Limited Access, or Contributor) that the members you're adding will have in the project.

See [What Are Project Memberships?](#) for more information about each membership.

- c. Click **Add**.

- d. Repeat substeps a, b, and c for different users and groups with various membership types, if needed.

10. Click **Finish**.

After the project is provisioned, the **Project Home** page opens where you can see a summary of the project's provisioning activities; default environment; default workspace; and Git, Maven, and NPM repositories. Review the activities feed and the Environments box for any errors.

When you create a project using the Visual Application template, these artifacts are created for you:

- A Git repository, which contains the visual application's source code.
To see the Git repository's files, go to the **Project Home** page, click the **Repositories** tab, then click the Git repository name:

The screenshot shows the GitHub interface within the Oracle Cloud Infrastructure. At the top, there are tabs for 'Repositories', 'Statistics', and 'Team'. Below the tabs, the word 'Git' is displayed in large letters, followed by a 'Create Repository' button. A search bar labeled 'Filter Git repositories' with a magnifying glass icon is present. Below the search bar are two buttons: 'All' and 'Favorites', and a dropdown menu 'Favorites First'. In the main area, a repository named 'employee-management.git' is listed, indicated by a star icon. To the right of the repository name is a 'Clone' button with a dropdown arrow.

- A **Development** environment pointing to the Visual Builder development instance. In the left navigator, click **Environments** to see the **Development** environment:

The screenshot shows the 'Environments' page. On the left, a sidebar lists environments: 'Development' (selected, highlighted in green), 'Visual Application Development Environment', and an ellipsis. On the right, a table titled '1 Instance' displays one row: 'Name' (VisualBuilder-Dev), 'Type' (IDCS Resource), and 'Status' (Available (2:27 PM)). There is also a '+ Add Instance' button.

- Build jobs that package and deploy the visual application's artifact to the Visual Builder development instance. By default, **Visual-Application-Package** and **Visual-Application-Deploy** jobs are created for you. The **Visual-Application-Package** job generates the visual application's artifact file. The **Visual-Application-Deploy** job deploys the visual application's artifact file to the Visual Builder development instance.

In the left navigator, click **Builds** and then click the **Jobs** tab to see the build jobs:

The screenshot shows the 'Builds' page. At the top, it says 'Jobs (16)', 'Job Queue (0)', 'Pipelines (10)', 'Pipeline Queue (0)', and 'Auto Refresh'. Below this is a table with columns: Status (dropdown), Private, Weather, Job (dropdown), Last Successful, Last Unsuccessful, Duration, and Actions. Two rows are visible: 'Prod-Visual-Application-Package' (Last Successful: N/A, Last Unsuccessful: N/A, Duration: N/A) and 'Prod-Visual-Application-Deploy' (Last Successful: N/A, Last Unsuccessful: N/A, Duration: N/A).

To run builds of the package and deploy jobs, you must first allocate VM build executors and make the appropriate deployment configurations. Without the appropriate configuration or VM executors, the builds won't run.

- A pipeline to run the build jobs in a sequence.

In the left navigator, click **Builds** and then click the **Pipelines** tab. In the pipeline's row, click the **Actions** *** menu and select **View Layout**.



- A private workspace to edit the visual application in the VB Studio Designer.
In the left navigator, click **Workspaces** to see the workspace:

Workspace	Repository	Current Branch	Current Sandbox	Environment	Last Opened	Size
Workspace1	employee-management.git	branch1		Development	3 minutes ago	9.7 KB
Visual Application						

- By default, the project uses the organization's default markup language. Your project's users use the markup language to format wiki pages and comments. If required, you can change the project's markup language from the **Project Administration** page. See [Change a Project's Wiki Markup Language](#).
- A VM executor is created if this project is VB Studio's first project and no build VM executors had existed when you created the project. The VM executor uses the **System Default OL7 for Visual Builder** build executor template. You can use this VM executor to run build jobs that reference the **System Default OL7 for Visual Builder** template in the current project and other projects as well. See [VM Build Executors](#) for more information about VM executors and build executor templates.

In the left navigator, click **Organization** and then click the **VM Build Executors** tab to see the VM executor.

Overview		1 VM	None of your 1 VM build executors are currently running.		None of your VMs are in an error state.	
			Sleep Timeout	Update Selected	Create Free VM	
	Template	Region	Shape	VCN OCID	Status	Actions
	System Default OL7 for Visual Builder	us-ashburn-1	VM.Standard.E2.2		Pending	

Configure the Deployment Job

The deployment job deploys the visual application's build artifact to your Visual Builder development instance. In the job, specify the application's version and profile, and the credentials required to connect and deploy build artifact to your Visual Builder development instance.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click the deployment job.
3. Click **Configure**.
4. Click **Configure** .

5. Click the **Steps** tab.
6. In **Username** and **Password**, enter yours or a user's credentials who can connect and deploy to the Visual Builder development instance.

The credentials will be used when the package and deploy build pipeline runs.

7. (Optional) To overwrite the application's default version, specify the new version in **Application Version**. Leave it empty to use the version defined in the application's visual-application.json file.

Don't deselect the **Include the application version in the URL** check box.

8. (Optional) In **Application Profile**, specify the development application profile. Leave it empty to use the application's default profile.

Your visual application accesses data from different servers for REST services and may need different security settings for different environments, such as development and production. Using application profiles, you can define different combinations of servers and security settings for each of your environments, and use them when deploying the application to an environment. This simplifies management of the visual application as you move through development to production. To learn more, see [About Application Profiles](#).

9. (Optional) To use the existing application's database, in **Data Management**, select **Keep existing environment data**.

To use a clean database for the application, in **Data Management**, select **Use clean database**.

10. Click **Save**.

Run the Pipeline Manually

The development build pipeline runs automatically when a commit is pushed to the Git repository's branch specified in the packaging job.

If you want to run the pipeline manually:

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. In development pipeline's row, click the **Actions** *** menu and select **Run Pipeline**.

To monitor the pipeline and see each job's status, click the pipeline's name. To see a job's build log, click the job's name and click **View Log**.

If you want to run a job's build manually, open the job's details page and click **Build Now**. You can monitor its build on the job's details page.

View the Deployed Visual Application

After the deployment job has successfully run, you can view the deployed applications in the **Deployments** tab of the **Environments** page.

1. In the left navigator, click **Environments** .
2. Select the Visual Builder environment.
3. Click the **Deployments** tab.
4. If not enabled, click the **Visual Applications** toggle button.

5. If the Visual Builder instance is from a different identity domain, provide its access credentials.
6. Expand the app's name to see the deployed app's link.

The **Deployments** tab displays the applications you've deployed from the current project. It doesn't show applications deployed by other users of the project, or applications deployed from other projects.

Example:

The screenshot shows the 'Deployments' tab in the Oracle Visual Builder interface. The table lists the following information for each deployment:

Git Repository	Application URL Root	Service (Target)	Schema	Status	Version	Updated	Actions	
▼ myvisualapp	myvisualapp	VisualBuilder-Dev	SP15594091...		Deployed	0.1	Yesterday at 9:35 AM	...
└ webapp								
▶ myvisualapp	myvisualapp	VisualBuilder-Dev	SP15594091...		Deployed	1.1	Yesterday at 11:52 AM	...
▶ myvisualapp	myvisualapp	VisualBuilder-Dev	SP15594091...		Shared	vbshare_1	Mon at 3:22 PM	...

Add Users to the Project

You must explicitly add users before they can work within a project, as explained in this table:

If you want to:	Do this:
Add a user to the project	<p>1. In the left navigator, click Project Home <p>2. Click the Team tab.</p> <p>3. Click + Create Member.</p> <p>4. Click the Username drop-down list. If you can't find a particular user, enter the user's name or username in the search box. As you begin typing, users matching the search term are displayed.</p> <p>5. Under Users, select the user.</p> <p>6. From the membership option types, select the user's membership.</p> <p>7. Click Add.</p> </p>

If you want to:	Do this:
Add a group to a project	<ol style="list-style-type: none"> 1. In the left navigator, click Project Home . 2. Click the Team tab. 3. Click + Create Member. 4. Click the Username drop-down list. 5. Under Groups, select the group. 6. From the membership option types, select the membership you want to assign to the group's members. 7. Click Add.
Add multiple users or groups to the project	<ol style="list-style-type: none"> 1. In the left navigator, click Project Home . 2. Click the Team tab. 3. Click + Create Member. 4. Click the Username drop-down list. 5. From the drop-down list, select a user or a group. Click Username again to select another user or group. If you can't find a particular user, enter the user's name or username in the search box. As you begin typing, users matching the search term are displayed. 6. From the membership option types, select the user's membership. 7. Click Add.
Change a user's or a group's project membership	To change a user's or a group's project membership, click the Change Membership icon  . From the dropdown, select a new project membership (Contributor, Developer, Developer Limited, or Project Owner).
Remove a user or a group from the project	<p>Before removing a user, change the ownership of any assigned issues and merge requests to another user.</p> <p>For the user or the group you to remove, click Remove .</p>

Optional Configuration

After setting up the project, you can follow these optional steps to configure some advanced settings in your project. You can perform these steps at any time during your development cycle.

To perform this action:	See this:
Add more VM executors or executor templates. By default, VB Studio is connected to its built-in free account and uses one free VM build executor with fixed software packages in the executor template. If you want to add more VM executors or customize executor template's software packages, configure VB Studio to connect to your OCI account.	Add More VM Build Executors or Build Executor Templates

To perform this action:	See this:
Configure the packaging job to change the visual application's archive file names. By default, the packaging job generates two archive files: <code>sources.zip</code> and <code>built-assets.zip</code> . If you want to specify custom file names, configure the packaging job.	Step 7 in Configure the Packaging Job
Protect the Git repository's <code>main</code> branch for unapproved code updates. By default, a branch is accessible to all project users and anyone can make changes to its files. To restrict changes and push commits to the <code>main</code> branch, set restrictions on it to allow branch merges only after they are approved.	Set Merge Restrictions on the main Branch

Add More VM Build Executors or Build Executor Templates

If you want to add more VMs to reduce the wait time for your organization's members or create custom VM templates, or use advanced features for VMs (such as use your own VCN or use a different VM shape), you should configure VB Studio to connect to your own OCI account.

1. Set up your OCI account and get the required input values.
See [Set Up the OCI Account](#) and [Get the Required OCI Input Values](#).
2. In the left navigator, click **Organization** .
3. Click **Connect OCI Account**.
4. Enter the required details and click **Validate**.
5. After successful validation, click **Save**.

The executor template and the free VM build executor of your built-in free account migrate to the OCI account.

To create custom executor templates, see [Create and Manage Build Executor Templates](#). Remember to add Node.js 18 (or a higher version) to the VM template. Node.js 18 is the minimum version required for packaging visual applications.

To add more VM executors:

1. Click the **VM Build Executors** tab.
2. Click **+ Create VM**.
3. In the Add VM Build Executor dialog box, in **Quantity**, specify the number of VM executors you want to allocate.

To minimize build execution delays, set the number of VM executors to the number of jobs that you expect to run in parallel using that template. If the OCI Compute VM quota is available, that number of VM executors will be added to the **Virtual Machines** tab.

If you're not sure about the number of VM executors you'll need, start with one VM executor and then add more as required. Note that the more VM executors you have running at a specific time, the higher the cost of OCPUs. To minimize the higher cost, use the Sleep Timeout setting on the **VM Build Executors** page to automatically shut down inactive VM executors. You can always return to the **VM Build Executors** tab to remove or add VM executors, based on your actual usage.

4. In **Build Executor Template**, select **System Default OL7 for Visual Builder** or a executor template with Node.js 18 (or higher) software.
5. In **Region**, **Shape**, and **VCN Selection** specify the OCI VM's region, shape, and Virtual Cloud Network (VCN).
 - A region is a localized geographic area where the data centers are located. Remember, a VM executor is a VM on OCI Compute. Choose the region where your VB Studio account is or the one that's closest to you geographically.
 - A shape is a template that determines the number of CPUs, amount of memory, and other resources allocated to the created instance. Choose a shape of your preference.
 - A VCN is a software-defined network that you set up in the Oracle Cloud Infrastructure data centers in a particular region. By default, builds run in your VB Studio's compartment. To run builds in your own VCN, select **Custom** and specify its details.

To learn more about regions and shapes, see [Regions and Availability Domains](#) and [VM Shapes](#). To find more about VCNs, see [VCNs and Subnets](#).

6. Click **Add**.

Configure the Packaging Job

From the source files in the Git repository's `main` branch, the packaging job generates two archive files: a source archive file that contains the visual application's source files and a build artifact archive file.

If you don't want to change the visual application's default archive file names, there's no need to configure the packaging job. However, if you do want to specify custom names, follow these steps:

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click the packaging job.
3. Click **Configure**.
4. Click **Configure** .
5. Click the **Steps** tab and navigate to the **Visual Application Packaging** section.
6. By default, the build jobs minifies the application's source code before running the build. If you don't want to minify the source files, deselect the **Optimize application** check box. Minification is a process to remove the unnecessary characters (such as blank spaces, new lines, and comments) from the source code and reduce the size of the files, making the transfer of files consume less bandwidth and storage.
7. If you want to change the default names of the archive files, in **File names**, select **Use custom file names**.

In **Sources**, specify the visual application source archive file's name and path. In **Build Artifact**, specify the build artifact archive file's name and path.

You'll need both archive files to deploy the visual application.

8. Click **Save**.

If you change the default file names of archive files, then remember to configure the deployment job to use the same file names.

1. In the **Jobs** tab, click the deployment job.
2. Click **Configure**.

3. Click **Configure** .
4. Click the **Steps** tab and navigate to the **Oracle Deployment** section.
5. In **File names**, select **Use custom file names**.
6. In **Sources and Build Artifact**, enter the same file names (with path) you specified in the packaging job.
7. Click **Save**.

Set Merge Restrictions on the `main` Branch

By default, the `main` branch is accessible to all project users and anyone can make changes to its files. To restrict changes and who can push commits to it, you may want to set restrictions on it to allow branch merges only after they are approved.

1. In the left navigator, click **Project Administration** .
2. Click **Branches**.
3. In **Repository and Branches**, select the Git repository and the `main` branch.
4. Select the **Requires Review** option.
5. In **Default Reviewers**, enter and select the users.

A default reviewer is a project member who is automatically added as a reviewer when a merge request is created on the branch.
6. From the **Approvals** drop-down list, select the minimum number of reviewers who must approve the review branch of a merge request, where the selected branch is the target branch
7. (Optional) To allow a review branch to be merged to the selected branch only if the last build of the linked job in Merge Request is successful, select the **Require successful build** check box.

To use this option, link a build job to a merge request.
8. (Optional) If you want to reset the approval status of reviewers if change is pushed to a branch after they have approved the merge request, select the **Reapproval needed when branch is updated** check box.
9. (Optional) To ensure changes pushed to the target branch match the contents of the review branch, select the **Changes pushed to target branch must match review content** check box.
10. (Optional) In **Merge Request Exempt Users**, specify users who can bypass the branch restrictions and merge the review branch of a merge request outside VB Studio or without required approvals.

This is useful if you want to allow some users to merge the review branch irrespective of review conditions being met.
11. Click **Save**.

Develop Your Application

Now that you've set up VB Studio and created a visual application project and added users to it, guide your organization members who develop visual applications to these documents to learn more about VB Studio.

- Build Enterprise Apps in Minutes in *Building Responsive Applications with Visual Builder Studio* describes how to create, edit and publish visual applications.
- Get Started in *Using Visual Builder Studio* is a good resource, as it explains how to manage a project, create and manage issues and Agile boards, review source code with merge requests, and more.

Set Up Additional Deployment Instances

If you want to deploy a visual application to additional Visual Builder instances (for example, a production or QA instance), you must create a separate VB Studio environment and add the instance to it. You can add only one Visual Builder instance to an environment.

1. In the left navigator, click **Environments** .
 2. Click **+ Create Environment**.
- In **Environment Name** and **Description**, enter a unique name and description, and click **Create**.
3. In the Service Instances tab, click **Add Instance**.
 4. In the Add Service Instances dialog box, select the **Visual Builder and Oracle Integration** option button.
 5. Do one of the following:
 - If the Visual Builder instance is in the *same* identity domain, select the instance you want to deploy to from the list of available instances displayed on the right and click **Add**.
 - If the Visual Builder instance is in a *different* identity domain, add the service instance using the Visual Builder instance's base URL and credentials that can access the instance:
 - a. Select the **Visual Builder Credentials** option.
 - b. In **Base URL**, enter the base Visual Builder instance's URL. For example: `https://yourserver`.
 - c. In **Instance Name**, if required, update the instance's display name. The name will be displayed in the **Service Instances** tab.
 - d. In **Username** and **Password**, enter the credentials of a user who can access the Visual Builder instance.
 - e. Click **Add**.

After you add a Visual Builder instance to an environment, the **Service Instances** tab displays a tile that shows basic information, such as status and response time, about the instance. Click  **Details** to see additional information (Visual Builder release version, Service URI, etc.) or click the Home Page link to visit its home page URL.

If the newly added instance stays in the **Unknown** status for some time, it usually indicates that the IDCS Application provisioning may have failed. VB Studio added the Visual Builder instance but can't access it. In such a case, click **Actions** *** and select **Remove** to delete the Visual Builder instance from the environment, and then click **Add** to add it again.

Set Up the Project to Deploy for Production

After your development and test cycles are complete, you may want to configure the project to build and deploy visual applications to the Visual Builder production instance.

Before you proceed, contact the Visual Builder administrator and make sure that the Visual Builder production instance is properly configured and running. For example, make sure that the production instance's security options are configured and the instance points to the correct database.

Here's a summary of how to set up the VB Studio project for deployment:

To perform this action:	See this:
1. In the VB Studio project, create an environment for the Visual Builder production instance. The Visual Builder production instance usually resides in another identity domain.	Set Up Additional Deployment Instances
2. Create a production branch from the main branch. Use this branch to push your application to production.	Create a Production Branch
3. Create the production packaging and deployment build jobs, and set up the pipeline.	Create and Configure Production Build Jobs
4. (Optional) Restrict users who can edit the production jobs or run their builds.	Configure a Job's Privacy Setting

Before you run production jobs, make sure that all code changes have been pushed to the production branch and there are no open merge requests.

Create a Production Branch

Follow your organization's guidelines to create a branch and protect it from unverified changes. To protect the branch, you can set merge restrictions, make the branch private and restrict who can push commits to it, or freeze it.

1. In the left navigator, click **Git** .
2. Click the **Refs** view and then click **Branches** .
3. From the **Repositories** drop-down list, select the repository.
4. Click **+ Create Branch**.
5. In the New Branch dialog box, in **Name**, enter the branch name. From the **Base** drop-down list, select the `main` branch as the base branch.
6. Click **Create**.

After creating the production branch, if you want to set merge restrictions on it, see Set Review and Merge Restrictions on a Repository Branch. To freeze the branch or make it private, or set other restrictions, see Protect a Branch.

Create and Configure Production Build Jobs

You need to set up some packaging and deployment jobs before you can deploy visual applications to your Visual Builder production instance. This topic explains how to do that.

Before you configure production build jobs, make a note of these:

- In the production packaging job, use the Git repository's production branch to generate production artifacts.
- In the development packaging job, if you changed the default file names of archive artifact files, get the new names and their paths. You'll need them when you configure the production packaging job.
- If you configured the development packaging job to overwrite the application's version defined in `visual-application.json`, get the new version. You'll need it when you configure the production packaging job.
- While configuring deployment build jobs, you specify whether to include the application's version in its URL. A visual application without the version in its URL is called a **Live** application. Usually, you deploy a live application to a production instance.
- If you deploy a version of a visual application that's never been deployed, VB Studio overwrites the last deployed version with the new version.
VB Studio doesn't undeploy the previously deployed version from the production instance. It continues to remain on the target instance, but is inaccessible.
- If you want to redeploy a live application or a previously deployed version, undeploy it first, else the deploy build fails. To undeploy a previously deployed visual application version, [configure a undeploy build job and run it](#). You can't undeploy it manually from the **Environments** page.
- If you've created an application profile for production, get its name. You'll need it when you configure the production deployment job.

Create a Production Packaging Build Job

The production packaging job generates a visual application artifact that's ready to deploy.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. In **Template**, select the **System Default OL7 for Visual Builder** template.
6. Click **Create**.
7. Click **Configure** .
8. Click the **Git** tab.
9. From the **Add Git** list, select **Git**.
10. In **Repository**, select the Git repository. In **Branch or Tag**, select the production branch.
11. Click the **Steps** tab.
12. From **Add Step**, select **Visual Application**, and then select **Package**.
13. By default, the build jobs minifies the application's source code before running the build. If you don't want to minify the source files, deselect the **Optimize application** check box.
Minification is a process to remove the unnecessary characters (such as blank spaces, new lines, and comments) from the source code and reduce the size of the files, making the transfer of files consume less bandwidth and storage.

14. If you want to change the default names of the archive files, in **File names**, select **Use custom file names**.

In **Sources**, specify the visual application source archive file's name and path. In **Build Artifact**, specify the build artifact archive file's name and path.
You'll need both archive files to deploy the visual application.

15. Click the **After Build** tab.
16. From **Add After Build Action**, select **Artifact Archiver**.
17. In **Files to archive**, enter the build artifact name. You can also use wild characters. For example, *.zip.
18. If you want to discard old artifacts of the build, click **Settings** . In the **General** tab, select the **Discard Old Builds** check box and specify the discard options.
19. Click **Save**.

Create a Production Deployment Build Job

The production deployment job deploys the visual application's artifact that was generated in the production packaging job to the Visual Builder production instance. Before you create the job, get the access credentials of a user who can connect and deploy to the Visual Builder production instance.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. In **Template**, select the **System Default OL7 for Visual Builder** template.
6. Click **Create**.
7. Click **Configure** .
8. Click the **Before Build** tab.
9. From **Add Before Build Action**, select **Copy Artifacts**.
10. In **From Job**, select the production packaging job that generated the visual application's artifact.
11. In **Which Build**, select **Last Successful Build**.
12. Leave other fields with their default or empty values.
13. Click the **Steps** tab.
14. From **Add Step**, select **Oracle Deployment**.
15. In **Target Instance**, select the Visual Builder production instance.
16. In **Username** and **Password**, enter the user's credentials who can connect and deploy to the Visual Builder production instance.
17. In the production packaging job, if you changed the default file names of archive artifact files, then in **File names**, select **Use custom file names**.

In **Sources** and **Build Artifact**, enter the same file names (with path) you specified in the packaging job.
If you didn't change the file names, use the default option.

18. (Optional) If you configured the development deployment job to overwrite the application's default version, specify the same version in **Application Version**. Leave it empty to use the version defined in the application's `visual-application.json` file.

If the version is already deployed, undeploy it first.

19. Deselect the **Include the application version in the URL** check box.

When deploying to a production instance, don't include the application's version in the deployed application's URL.

20. (Optional) In **Application Profile**, specify the production application profile. Leave it empty to use the application's default profile.

Your visual application accesses data from different servers for REST services and may need different security settings for different environments, such as development and production. Using application profiles, you can define different combinations of servers and security settings for each of your environments, and use them when deploying the application to an environment. This simplifies management of the visual application as you move through development to production. To learn more, see [About Application Profiles](#).

21. To use a clean database for the application, in **Data Management**, select **Use clean database**.

22. Click **Save**.

Configure a Production Job's Privacy Setting

Mark a job as private to restrict who can see or edit a job's configuration, or run its build.

A private job must be run manually. It won't run if a non-authorized user tries to run the job directly, through an SCM/periodic trigger or a pipeline.

1. In the left navigator, click **Project Administration** .
2. Click **Builds**.
3. Click the **Job Protection** tab.
4. From the jobs list, select a production job.
5. Select the **Private** option.
6. In **Authorized Users**, add yourself.

Then, to add other users, select them individually or, under Groups, select a group (or groups) of users.

7. Repeat the steps 4-6 for the deployment production job.
8. Click **Save**.

A private job shows a  icon in the jobs list on the right side of the Job Protection page, in the **Jobs** tab of the **Builds** page, and in the pipelines.

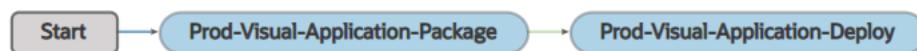
Create and Configure a Pipeline

To ensure the production deployment job runs automatically after the production packaging job, create a pipeline and set the dependency.

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. Click **+ Create Pipeline**.

4. In the Create Pipeline dialog box, in **Name** and **Description**, enter a unique name and description.
5. Click **Create**.
6. On the Pipeline Configuration page, right-click the **Start** node and select **Add New Start** jobs.
7. Click in the **Select new on success job(s)** field, select the packaging job and click **Save**.
8. Right-click the packaging job and select **Add**, then **Add New On Success Jobs**.
9. Click in the **Select new on success job(s)** field, select the deployment job, and click **Save**.

Here's an example of the finalized pipeline:



10. Click **Save**.

Undeploy a Visual Application

If you want to undeploy a visual application that's deployed to Visual Builder, you can do so manually or through a job configuration.

You can undeploy a visual application manually if it is deployed to a Visual Builder instance that's in the same identity domain as VB Studio. If the visual application is deployed to a Visual Builder instance in a different identity domain than VB Studio (such as your production instance) or the Visual Builder instance was added to an environment through credentials, you should configure a build job to undeploy it.

Undeploy a Visual Application Manually

You can undeploy a visual application that's deployed to your current identity domain's Visual Builder manually from the **Deployments** tab of its environment.

Note:

If your visual application is deployed to an identity domain that's different than your VB Studio instance, you'll need to follow the instructions in [Configure a Build Job to Undeploy a Visual Application](#) to undeploy the application.

1. In the left navigator, click **Environments** .
2. Select the environment where the visual application is deployed.
3. Click the **Deployments** tab.
4. Expand the application.
5. For the visual application to undeploy, click **Actions** *** and select **Undeploy**.
6. If the visual application is a live application (without the version number in the URL), select the **Yes, I'm sure** check box.

7. Click **Undeploy**.

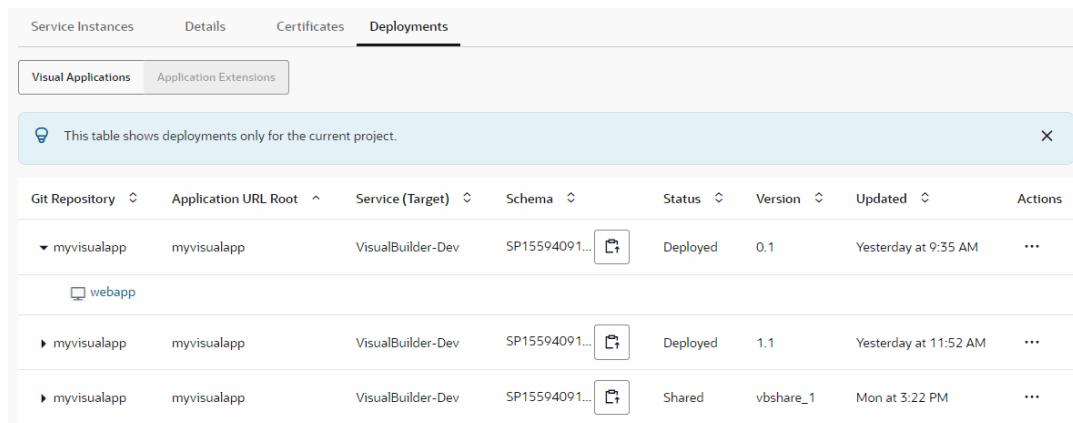
Configure a Build Job to Undeploy a Visual Application

To undeploy a visual application deployed to the production instance, or another identity domain's Visual Builder instance, configure a build job. Before you configure the job, get the access credentials of a user who can access the Visual Builder instance where the visual application is deployed.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. In **Template**, select the **System Default OL7 for Visual Builder** template.
6. Click **Create**.
7. Click **Configure** .
8. Click the **Steps** tab.
9. From **Add Step**, select **Visual Application**, and then select **Undeploy**.
10. In **Instance**, select the Visual Builder instance where the application is deployed.
11. In **Username** and **Password**, enter the user's credentials who can connect and undeploy to the Visual Builder instance.
12. In **Application URL Root** and **Application Version**, enter the visual application's root URL and its version.

You can find the application's root URL and its version from the **Deployments** tab of the environment where the visual application is deployed.

Example:



Git Repository	Application URL Root	Service (Target)	Schema	Status	Version	Updated	Actions
▼ myvisualapp	myvisualapp	VisualBuilder-Dev	SP15594091...		Deployed	0.1	Yesterday at 9:35 AM
webapp							
▶ myvisualapp	myvisualapp	VisualBuilder-Dev	SP15594091...		Deployed	1.1	Yesterday at 11:52 AM
▶ myvisualapp	myvisualapp	VisualBuilder-Dev	SP15594091...		Shared	vbsshare_1	Mon at 3:22 PM

13. Click **Save**.

14. To run a build, click **Build Now**.

After the build is successful, open the build's log. You'll see a similar message:

```
[2020-01-01 12:00:00] === Begin VB App Ops Undeploy Application===
[2020-01-01 12:00:00] Application employee-manager-visual-web-app/0.1 has
been undeployed
```

```
[2020-01-01 12:00:00] === End VB App Ops Undeploy Application ===  
Executor log size 364 B (364)  
[2020-01-01 12:00:02]  
[2020-01-01 12:00:02] Build completed.  
[2020-01-01 12:00:02] Status: DONE Result: SUCCESSFUL Duration: 7.0 sec
```

6

Manage the VB Studio Instance

After creating the VB Studio instance, you change the CI/CD compartment and the instance's display name, but can't change the name displayed in the instance's URL.

Change the CI/CD Compartment

You can switch to another OCI compartment any time. When you switch, your current compartment's executor templates and VM executors migrate to the new compartment, but Object Storage buckets do not migrate. The data in buckets may include build artifacts, Maven artifacts, NPM artifacts, and exported project data.

1. Open the Visual Builder Studio Instances page and click the VB Studio instance.
2. Click **Edit**.
3. In the Edit Instance box, under **CI/CD Setup**, select the compartment you want to use for VM executors and storage.
4. Click **Submit**.

The VB Studio instance is updated to use the specified compartment.

If you switch to the built-in free account, you'll lose access to data in Object Storage, and your current account's executor templates and VM executors. Instead, you'll get access to only one free VM build executor and the default executor template.

Switch to the Built-In Free Account

If you switch to the built-in free account, your current account's build executor templates and build executors will be deleted and you'll have access to only one free VM build executor with fixed software in the default build executor template. You can't add more build executors or edit the executor template's software.

1. Open the Visual Builder Studio Instances page and click the VB Studio instance.
2. Click **Edit**.
3. Deselect the **Yes, I authorize this** check box.
4. In the **Confirm Delete** dialog box, click **Delete Resources**.
5. In the Edit Instance dialog box, click **Submit**.

Change Your OCI Account

If you are using the built-in free account or your own OCI account, you can switch to another OCI account any time and use the new account's resources.

When you switch to another OCI account:

- Your current OCI account's build executor templates and their metadata migrate to the new OCI account. For example, If you have 10 executor templates in your current OCI account, VB Studio creates the same number of executor templates with the same software in the new OCI account.

- Your current OCI account's VM build executors and their metadata migrate to the new OCI account. For example, if you have 234 VM executors in your current OCI account, VB Studio creates the same number of VM executors of the same shape, VCN, and executor templates in the new OCI account.
- Your current OCI account's Object Storage buckets do not migrate to the new account. The data in buckets may include build artifacts, Maven artifacts, NPM artifacts, and exported project data.

To switch from the built-in free account to your OCI account:

1. Open the Visual Builder Studio Instances page in the OCI Console and select the VB Studio instance.
2. Click **Edit**.
3. Select the **Yes, I authorize this** check box.
4. In the **Confirm Delete** dialog box, click **Delete Resources**.
5. In the Edit Instance dialog box, click **Submit**.

If you switch to the built-in free account, you'll lose access to data in Object Storage, and your current account's executor templates and VM executors. Instead, you'll get access to only one free VM build executor and the default executor template.

Validate the OCI Connection

If you notice any Compute or Storage issues, validate your OCI connection:

1. In the left navigator, click **Organization** .
2. Click the **OCI Account** tab.
3. Click **Validate**.

If the validation fails, click **Edit** to edit the connection, or switch to another OCI account. If your OCI account has reached Compute and Storage service limits, contact the OCI administrator to [increase the service limit](#).

Manage VB Studio Instance Tags

OCI tags enable you to tag your OCI resources, such as your VB Studio instance, and help you organize resources based on your business needs. You can manage VB Studio tags from the VB Studio Instance Details page.

If you're new to OCI tags, see [Tagging Overview](#).

Action	How To
Add a tag	<ol style="list-style-type: none">1. On the VB Studio Instance Details page, click Add Tags.2. In the Add One Or More Tags To This Resource dialog box, enter the tag's namespace, key, and value.3. Click Add Tags.

Action	How To
Edit a tag	<ol style="list-style-type: none"> On the VB Studio Instance Details page, click the Tags tab. To edit a tag, click its Edit  icon. In the Edit Tag dialog box, edit the tag and click Save.
Remove a tag	<ol style="list-style-type: none"> On the VB Studio Instance Details page, click the Tags tab. To edit a tag, click its Edit  icon. In the Edit Tag dialog box, click Remove Tag.

Write IAM Policies that Control Access to VB Studio

Identity and Access Management (IAM) is an Oracle service that helps you manage the permissions that control which OCI resources your users can access. You use IAM to control who is authenticated (signed in) and who is authorized (has permissions) to use resources. In essence, IAM ensures that the right users have the appropriate access to resources. For VB Studio, that means the vbstudio-instance resource. This topic provides information you can use to write policies that control this access.

Supported Variables

You use variables when you add conditions to a policy. These conditions are evaluated by the OCI authorization service when API requests are authorized.

For the complete list of general variables that are applicable to all requests, including those that can be used in policy statements authorizing VB Studio API requests, see [General Variables for All Requests](#).

Details for Meta-Verb + Resource-Type Combinations

The following table shows the visualbuilder-instance resource type permissions and API operations that are covered by each meta-verb. The level of access is cumulative as you go from inspect to read to use to manage.

Verb	Permissions	APIs Fully Covered	APIs Partially Covered
INSPECT	<ul style="list-style-type: none"> VBS_INSTANCE_INSPECT 	<ul style="list-style-type: none"> ListVbsInstances ListWorkRequests 	None
READ	<ul style="list-style-type: none"> Inherits from INSPECT: <ul style="list-style-type: none"> VBS_INSTANCE_INSPECT VBS_INSTANCE_READ 	<ul style="list-style-type: none"> GetVbsInstance GetWorkRequest ListWorkRequestErrors ListWorkRequestLogs 	<ul style="list-style-type: none"> None
USE	<ul style="list-style-type: none"> Inherits from READ: <ul style="list-style-type: none"> VBS_INSTANCE_INSPECT VBS_INSTANCE_READ VBS_INSTANCE_UPDATE 	<ul style="list-style-type: none"> UpdateVbsInstance 	<ul style="list-style-type: none"> None

Verb	Permissions	APIs Fully Covered	APIs Partially Covered
MANAGE	<ul style="list-style-type: none"> • Inherits from USE: <ul style="list-style-type: none"> - VBS_INSTANCE_INSPECT - VBS_INSTANCE_READ - VBS_INSTANCE_UPDATE • VBS_INSTANCE_CREATE • VBS_INSTANCE_DELETE 	<ul style="list-style-type: none"> • CreateVbsInstance • DeleteVbsInstance 	None

Permissions Required for Each API Operation

These are the permissions that are required for each API operation for instances and work requests.

API Operation	Permissions Required to Use the Operation	Description
ListVbsInstances	VBS_INSTANCE_INSPECT	List the VB Studio instances
GetVbsInstance	VBS_INSTANCE_READ	Get the VB Studio instance
CreateVbsInstance	VBS_INSTANCE_CREATE	Create a VB Studio instance
DeleteVbsInstance	VBS_INSTANCE_DELETE	Delete a VB Studio instance
UpdateVbsInstance	VBS_INSTANCE_UPDATE	Update a VB Studio instance
ListWorkRequests	VBS_INSTANCE_INSPECT	List a Visual Builder instance's work requests
GetWorkRequest	VBS_INSTANCE_READ	Get a Visual Builder instance's work request
ListWorkRequestErrors	VBS_INSTANCE_READ	List a Visual Builder instance's work request errors
ListWorkRequestLogs	VBS_INSTANCE_READ	List a Visual Builder instance's work request logs

Sample OCI IAM Policy Statements that Grant These Permissions

This example grants fine-grained permissions to a group called devteam1:

```
Allow devteam1 to
{VBS_INSTANCE_INSPECT,VBS_INSTANCE_READ,VBS_INSTANCE_CREATE,VBS_INSTANCE_UPDATE,VBS_INSTANCE_DELETE,VBS_INSTANCE_MOVE} in tenancy
```

Here's another example that uses the friendlier meta-verbs to grant the same permissions to group devteam1 and a lesser set to another group, devteam2:

```
Allow devteam1 to manage vbstudio-instances in tenancy
Allow devteam2 to use vbstudio-instances in tenancy
```

Create Rules that Use Service Event Notifications

This step is optional.

Many actions that you perform, such as creating, modifying, or deleting a VB Studio instance, emit event notifications. You can use the OCI console to define *rules* that trigger a specific action when an event occurs and a notification about that event is sent. For example, you

might want to define a rule that sends a notification to an administrator (or group of administrators) when a VB Studio instance is created, updated, or deleted.

For more information about events and how to get started using them in rules, see [Overview of Events](#) and [Get Started with Events](#).

Here are the Visual Builder Studio events that you can use in a rule:

Event Name	Event Type
Visual Builder Studio - Create Instance - begin	com.oraclecloud.vbstudioinst.createvbsinstance.begin
Visual Builder Studio - Create Instance - end	com.oraclecloud.vbstudioinst.createvbsinstance.end
Visual Builder Studio - Delete Instance - begin	com.oraclecloud.vbstudioinst.deletevbsinstance.begin
Visual Builder Studio - Delete Instance - end	com.oraclecloud.vbstudioinst.deletevbsinstance.end
Visual Builder Studio - Update Instance - begin	com.oraclecloud.vbstudioinst.updatevbsinstance.begin
Visual Builder Studio - Update Instance - end	com.oraclecloud.vbstudioinst.updatevbsinstance.end

Move the VB Studio Instance

To use VB Studio with a different Oracle Cloud Applications instance, you'll need to file a service request so the VB Studio instance associated with your current TEST or DEV instance can be terminated, then a new one can be created and associated with your preferred instance.

Include these details in the service request:

- Current instance URL for the Oracle Cloud Applications pod that's associated with your VB Studio instance
- Your VB Studio service instance URL
See [How Do I Find the URL of My VB Studio Instance?](#)
- Instance URL for the destination Oracle Cloud Applications pod that you want to move VB Studio to
- Your request for assistance with backing up your existing VB Studio project, if you've used your current VB Studio instance and want to retain project data
- Your approval to terminate the current VB Studio instance

Terminate the VB Studio Instance

When you terminate your VB Studio instance, all projects and their data (such as Git repositories, workspaces, build jobs, and build artifacts) are deleted. OCI resources, such as VM executors, are destroyed and OCI Object Storage buckets are deleted too.

1. Open the Visual Builder Studio Instances page and click the VB Studio instance.
2. Click **Terminate**.
3. In the Confirm dialog box, enter the VB Studio instance's name and click **Terminate**.

Wait for some time. OCI takes a few minutes to terminate the VB Studio instance.

Manage the Organization

From the Organization page, you can manage projects and update the organization's properties.

Update the Organization's Display Name and Description

By default, the organization's display name is <your_vbstudio_instance_name>-<your_oracle_account_name>.

1. In the left navigator, click **Organization** .
2. Click the **Properties** tab.
3. In **Name**, edit the organization name.
4. In the Confirm Organization Name Change dialog box, click **Yes**.
5. In **Description**, if necessary, enter or edit the organization's description.

Set the Organization's Default Wiki Markup Language

When you create a project, the project creation wizard displays the organization's default wiki markup language. If you don't change the wiki markup language, the default markup language is used as the project's markup language. You can change the default markup language for new projects, but note that existing projects continue to use their original markup language.

1. In the left navigator, click **Organization** .
2. Click the **Properties** tab.
3. In **Markup Language**, select the default wiki markup language.

Restrict Project Creation to Organization Administrators

By default, all VB Studio users (users with the DEVELOPER_ADMINISTRATOR IDCS role or the DEVELOPER_USER IDCS role) can create a project. As the Organization Administrator, if required, you can restrict project creation to yourself and other organization administrators (users with the DEVELOPER_ADMINISTRATOR IDCS role).

1. In the left navigator, click **Organization** .
2. Click the **Properties** tab.
3. Select the **Limit project creation to organization administrators only** check box.

The **+ Create** button on the All Projects page and in the Switch Project menu is now hidden for users assigned the DEVELOPER_USER IDCS role. To allow them to create projects, deselect the **Limit project creation to organization administrators only** check box.

Restrict Sharing a Project to Organization Administrators Only

By default, all VB Studio users can create shared projects. Users who are project owners can share their private projects and make them accessible to all users of the organization.

As the Organization Administrator, if required, you can restrict the users' ability to create a shared project or share a private project. Once this restriction is set, only you and other organization administrators can create a shared project or share a private project.

1. In the left navigator, click **Organization** .
2. Click the **Properties** tab.
3. Select the **Limit shared project to organization administrators only** check box.

After the check box is selected, the **Shared** option under **Security** in the New Project wizard and the **Project Administration > Properties** page is disabled for users who are assigned the DEVELOPER_USER IDCS role. They can't change a private project's security to shared, but they can change a shared project's security to private.

Limit the Number of Personal Access Tokens a User Can Create

An organization administrator can limit the maximum number of tokens a user in the organization is allowed to create. This action can prevent users from creating excessive numbers of personal access tokens. It also makes it easier for an administrator to maintain existing tokens. The **Organization Properties** page provides an option for controlling the maximum number of tokens each user can create. The default limit is 10 personal access tokens per user, but this limit can be increased to a maximum of 50 tokens per user. The set value applies to all users, including the organization administrator, who sets the limit.

To set a new limit for the number of personal access tokens a user can create:

1. In the left navigator, click **Organization** .
2. Click the **Properties** tab.
3. In the Access Token Controls section, at the bottom of the page, select the **Enforce a per-user limit on the number of personal access tokens** check box.
4. In the **Number of tokens** field, use the selector to set the maximum number of personal access tokens each user in the organization can have.

The default value is 10. The maximum number is 50.

After the check box is selected and a limit set, the **User Preferences**, **Maven Browse**, and **NPM Browse** pages will use the set limit to deal with the situation that arises when a user's token limit has been reached. When the limit is decreased, the new limit applies to new token creation only. It doesn't force users to delete any of their existing tokens. In other words, it doesn't affect the number of already-created tokens but it can prevent them from creating any new tokens until they delete enough of the existing tokens to be under the limit.

To remove the limit, deselect the **Enforce a per-user limit on the number of personal access tokens** check box.

Note:

VB Studio automatically deletes tokens 30 days after they expire, but not without giving users plenty of advance notice first:

- An "Expires Soon" status label is displayed next to the token 5 days before the actual expiration date.

The screenshot shows the 'Personal Access Tokens' section of the Visual Builder Studio interface. A token named 'TestToken' is selected, and its details are shown in a modal window. The modal displays the token's basic details, including its label ('TestToken'), creation date ('Created October 3, 2023'), and expiration date ('Expires on October 28, 2023 4:06 PM +0530'). A red oval highlights the 'Expires Soon' status indicator next to the expiration date.

- An email notification is sent to the user 5 days before the user's personal token is about to expire.

The screenshot shows an email from 'Oracle Visual Builder Studio <do-not-reply@oracle.com>' to 'To: Alex Admin'. The subject of the email is 'Personal access tokens are about to expire'. The email body contains a message stating: 'This is a courtesy email, as access tokens cannot be refreshed prior to their expiration date, nor can they be restored after expiration. We suggest you create a new token now and replace the existing token to avoid disruption.' Below this message, there is a table titled 'Personal Access Token Details' showing three tokens: 'GitAccess' (expiring on WED, OCT 11, 2023 10:07 PM UTC), 'ContributorToken' (expiring on FRI, OCT 13, 2023 5:27 AM UTC), and 'ForDevLimitedAccess' (expiring on MON, OCT 16, 2023 9:15 AM UTC). At the bottom of the email, there is a link 'Go to Oracle Visual Builder Studio'.

Assign a Project's Ownership to Another User

At times, you may want to assign a project's ownership to another user. For example, when a project's owners are removed or have left the organization, you should assign the Project Owner role to another project user.

1. In the left navigator, click **Organization** .
2. Click the **Projects** tab.
3. Click the **Owner** filter button.

If you can't find the project in the filtered list, you aren't assigned the project's Project Owner membership. You should assign the project's ownership to yourself before you assign it to another user.

- a. Click the **All** filter button.
- b. Locate the project.
- c. Click **Update Project** ***.
- d. Select **Assign Me as Owner**.
4. Click the project's name to open it.
5. Click the **Team** tab.
6. If the user whom you want to assign the ownership is already a member, click the **Change Membership**  icon and select **Project Owner**.
If the user is not a member, add the user to the project.
 - a. Click **+ Add Member**.
 - b. Click the **Username** drop-down list and select the user.
 - c. Select the **Project Owner** membership.
 - d. Click **Add**.
7. After adding the user, you can remove yourself from the project. Click **Remove**  next to your name. In the dialog box that opens, click **OK**.

Manage Your Organization's Projects

As the Organization Administrator, you can access and manage all projects. You can assign a project's Project Owner role to yourself, delete a project, revoke the project's deletion, or postpone the project's deletion by a few days.

1. In the left navigator, click **Organization** .
2. Click the **Projects** tab.

Example:

The screenshot shows the Visual Builder Studio interface. On the left is a dark sidebar with various project management icons: Project Home, Workspaces, Git, Merge Requests, Maven, NPM, Docker, Builds, Releases, Environments, Issues, Boards, Wiki, Snippets, and Project Administration. The main area is titled "Visual Builder Studio" and "Organization". At the top, there are tabs for Projects, OCI Account, Build Executor Templates, VM Build Executors, Component Exchange, Groups, and Properties. Below the tabs is a search bar and filter buttons for Member, Favorites, Owner, All, and a detailed filter icon. A summary bar shows "Projects: 6", "Disk Usage: 48.71 MB / 20 GB", and a "+ Create" button. The main content area is titled "Projects: 6" and lists six projects with columns for Name, Template, Favorite, Status, Members, and Disk Usage. The projects listed are Customer Experience, Demo, Employee Manager, My Java Apps, My Private Project, and Node.js Microservice.

Name	Template	Favorite	Status	Members	Disk Usage
Customer Experience	Based on: Application Extension	☆	Active	TT	15.56 MB
Demo		☆	Deleted Remove in 7 days	DC	13.39 MB
Employee Manager	A web app to manage emplo...	☆	Active	DD	18.06 MB
My Java Apps		☆	Active	DC	14.98 MB
My Private Project		☆	Active	S	15.79 MB
Node.js Microservice		☆	Active	H	13.96 MB

Here are the project management actions you can perform:

Action	How To
Create a project	Click + Create . See Create a Project.
Open a project	In the Name column, click the project name. See Open a Project. You can open a project only if you're a member or an owner. Projects where you aren't a member or an owner don't appear as links. You can't open a project that's in the deleted state.
View a project's team members	The Members column shows the gravatars of the project's team members. A blue circle around a user's gravatar indicates the user is assigned the project's Project Owner role. Click the gravatars to display names and email addresses of all team members.
Search for projects	In the search box above the projects list, enter the search term. The term is matched against your organization's project names and their descriptions, as well as their members' names, usernames and email addresses.
Filter projects	In Member/Owner/Favorite/All , click a filter option to enable it and filter the projects list. Click Owner to view projects where you're a Project Owner. Click the Member button to view projects where you're a team member (any membership including Project Owner). Click Favorite to view your favorite projects. Click All to view organization's all projects. To see more filter options, click Detailed Filter .
Mark a project as a favorite	In the Favorite column, click Favorite to add it to your favorites list.

Action	How To
Assign a project's Project Owner role to yourself	<p>You can assign a project's Project Owner role to yourself. This is usually helpful in a case when all project's owners are removed or have left the organization. As the Organization Administrator, you can then take the ownership of such projects and then assign the Project Owner role to another project user.</p> <p>For the project whose ownership you want to assign to yourself, click Update Project *** and select Assign Me as Owner.</p>
Delete a project	<p>As the Organization Administrator, you can delete any project.</p> <ol style="list-style-type: none"> 1. For the project you want to delete, click Update Project *** and select Delete. 2. In the Delete Project dialog box, click Delete. <p>The project's State is marked as Deleted - Project will be removed automatically in <i>N</i> days., but the project isn't removed immediately. In the Delete Project dialog box, read the message before you click Delete. Also, check the project's delete message to know when the project will be permanently removed. To immediately remove the project permanently, click *** and select Remove Forever.</p>
Postpone a deleted project's removal	<p>At times, you may be unsure if you want to revoke a project's deletion that's close to its permanent removal date. For such projects, you can choose to postpone their removal by a few days.</p> <p>For the deleted project you want to postpone, click Update Project *** and select Postpone Removal.</p> <p>The project's permanent removal date is rescheduled and the message is displayed at the top. In Status, check the project's delete message to know when the project will be permanently removed.</p>
Undelete a project	<p>When you undelete a project, you get an option to update the name and description of the project. When the deletion is revoked, the project URLs will change if a different name of the deleted project is specified.</p> <ol style="list-style-type: none"> 1. For the deleted project you want to undelete, click Update Project *** and select Undelete. 2. In the Undelete Project dialog, if necessary, update the project name in Name, and update the description in Description. <p>You would have to change the project name if another project with the same name was created after this project was deleted.</p> <ol style="list-style-type: none"> 3. Click Undelete.
Select multiple projects	<p>You can select multiple projects and perform a common action on them.</p> <ol style="list-style-type: none"> 1. If necessary, filter and sort the projects as desired. 2. Press the Ctrl key or the Shift key and select project rows. <p>To select all projects, click the check box to the left of the Name column header.</p> <ol style="list-style-type: none"> 3. Click Update Selected and select the desired action from the menu. <p>Some options, such as Undelete, are not available in the Update Selected menu.</p>

Action	How To
View usage metrics of projects	<p>Next to the Detailed Filter  , review the projects and the disk usage metrics of the organization.</p> <p>For each project, the Disk Usage column shows the project's disk usage. Click the number to open a pop-up with details of usage by each service in the project.</p>

Manage Your Organization's Groups

In VB Studio, you can create groups of your VB Studio organization's users or import existing Oracle Identity Cloud Service (IDCS) groups, and then add these groups to VB Studio projects. When you add a group to a project, you assign a common membership type to all group's users.

After creating a group, you can add more members to it and add the group to projects. You can also import IDCS groups to VB Studio, but can't add or remove users from the imported IDCS group. To do that, sign in to the IDCS Console. Note that the imported IDCS group is synced with IDCS every five minutes.

This table describes the actions you can perform to create and manage groups.

Action	How To
Create a VB Studio group	<ol style="list-style-type: none">1. In the left navigator, click Organization .2. Click the Groups tab.3. Click + Create Group.4. In Type, if not already selected, select the VB Studio tile.5. In Name and Description (optional), enter the group's name and description.6. To see all members you can add, click the members list. Users who are assigned the DEVELOPER_ADMINISTRATOR or the DEVELOPER_USER IDCS role are displayed.7. From the users drop-down list, select users to add to the group. If you can't find a particular user, enter the user's name or username in the search box. As you type, the drop-down list displays users matching the search term.8. Click Create.

Action	How To
Import an IDCS group	<p>1. In the left navigator, click Organization .</p> <p>2. Click the Groups tab.</p> <p>3. Click + Create Group.</p> <p>4. In Type, select the IDCS tile.</p> <p>5. Under the filter box, select the group to import. The drop-down list displays IDCS groups that are assigned either the DEVELOPER_ADMINISTRATOR or the DEVELOPER_USER IDCS role. To select multiple groups, press the Ctrl key or the Shift key, and then select groups. If you can't find a particular group, enter the group's name in the search box. As you type, the drop-down list displays groups matching the search term.</p> <p>6. Click Create.</p> <p>From VB Studio, you can't add or remove users of an IDCS group. You can do that from the IDCS Console. See Manage Oracle Identity Cloud Service Groups.</p>
See a group's members	<p>1. In the left navigator, click Organization .</p> <p>2. Click the Groups tab.</p> <p>3. Locate the VB Studio group and click the user gravatars in the Members column.</p>
Add members to an existing VB Studio group	<p>1. In the left navigator, click Organization .</p> <p>2. Click the Groups tab.</p> <p>3. Locate the VB Studio group and click Add Member to the Group .</p> <p>4. Click the members drop-down list. All users who are assigned the DEVELOPER_ADMINISTRATOR or the DEVELOPER_USER IDCS role are displayed.</p> <p>5. Select the user from the drop-down list. If you can't find the user, enter the user's name or username in the search box. As you type, users matching the search term are displayed.</p> <p>6. Click Add.</p> <p>7. In the Members tab, verify the added members.</p> <p>8. Click Close.</p>
Remove members from a VB Studio group	<p>1. In the left navigator, click Organization .</p> <p>2. Click the Groups tab.</p> <p>3. Locate and double-click the VB Studio group.</p> <p>4. In the Members tab, select the members to remove.</p> <p>5. Click Remove Members.</p> <p>6. Click Remove Members to confirm.</p>

Action	How To
Add a group to a project	<p>To add a group to a project, you must be assigned the project's Project Owner membership.</p> <ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Locate the VB Studio group and click Add Group to a Project . 4. From the project drop-down list, select the project. 5. From the roles list, select the role you want to assign to the group's members. 6. Click Add. 7. Click Close.
Remove a group from a project	<p>To remove a group from a project, you must be assigned the project's Project Owner membership.</p> <ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Locate and double-click the VB Studio group. 4. In the Projects tab, select the projects to remove. 5. Click Remove Group from Projects. 6. Click Remove Group from Projects to confirm.
See the projects a group is added to	<ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Locate the VB Studio group and click the user gravatars in the Projects column. 4. Click Close.
Edit a group's name or description	<ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Locate and double-click the VB Studio group. 4. Click Actions *** and select Edit. 5. Update the group's name and description 6. Click Save. 7. Click Close.
Delete a group	<ol style="list-style-type: none"> 1. In the left navigator, click Organization . 2. Click the Groups tab. 3. Locate the VB Studio group. 4. Click Delete. 5. Click Delete to confirm.

Manage Your Component Exchange

If your team develops custom components for visual applications and want the components to be available to all users in the Visual Builder Components tab, you'll need to first set up a component exchange. This chapter tells you how to set up the Component Exchange in Visual Builder.

What is a Component Exchange?

A Component Exchange is a repository of custom components that you use in your visual applications. The custom components include web components, application templates, UI patterns, and actions and are available to all VB Studio users.

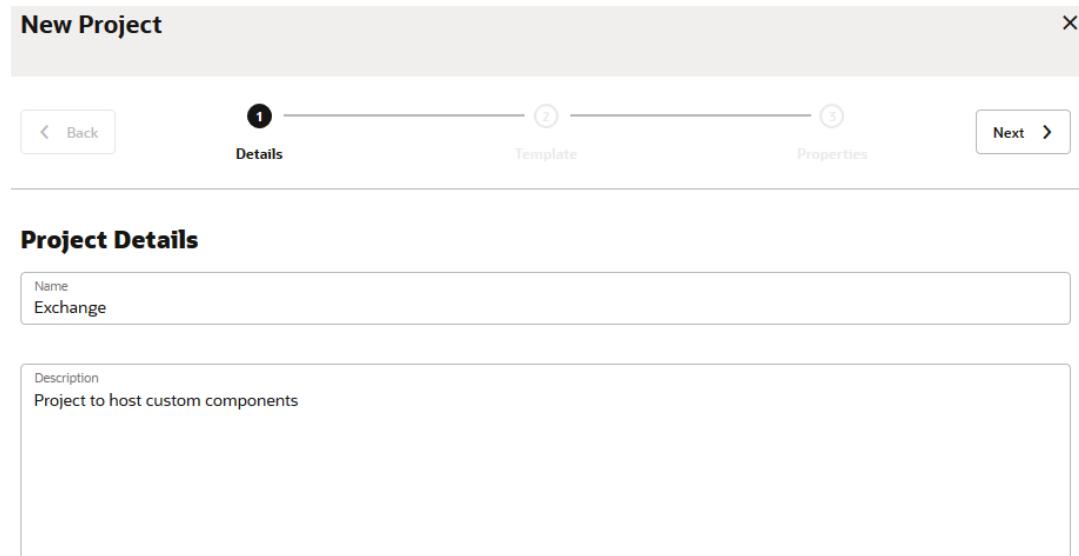
To host the custom components, you can define either a VB Studio project or an external server as the Component Exchange. Any VB Studio project can be used as the Component Exchange, but you should create and set up a separate project to host components. If you have multiple VB Studio instances across multiple tenancies, you can specify an external server as the common Component Exchange for all VB Studio instances.

To use a project or a server as the Component Exchange, specify its details on the **Organization** page. When set, the VB Studio Designer automatically shows the Component Exchange's hosted components in the Components tab.

Use a Project as the VB Studio Component Exchange

1. On the Organization page, click **+ Create**.
If you've already created a project to use as Component Exchange, jump to step 10.
2. On the **Details** page, enter a unique name and description for the project.

Example:



3. In **Security**, select **Shared**.
4. In **Preferred Language**, specify the language for the email notifications your project users will receive.

You can change the language in which the user interface appears in your user preferences.

5. Click **Next**.
6. On the Template page, select **Empty Project**, and click **Next**.
7. In **Wiki Markup**, select the project's wiki markup language.

Project's users use the markup language to format wiki pages and comments. If you're not sure which markup language to choose, use the default markup language. You can change the markup language later from the **Project Administration** page. See Change a Project's Wiki Markup Language.

8. Click **Finish**.
- After the project is provisioned, you're navigated to the **Project Home** page.
9. Add users to the project and assign them proper memberships. See Manage Project Users and Groups.

Add users who would develop the components and assign them either the Developer or Developer Limited membership. Assign the Project Owner membership to trusted users only. Remember that a user with the Project Owner membership can delete the project and Git repositories.

10. In the left navigator, click **Organization** .
11. Click the **Component Exchange** tab.
12. Click **Edit**.
13. Select the **Enable organization-wide component hosting** check box.
14. Select the **In a project** option.

15. In **Select Project**, select the project you created.

16. Click **Save**.

That's it. When your team members upload the components to the Component Exchange, they'll be automatically available in the Designer's Component tab.

Use an External Server as the VB Studio Component Exchange

1. In the left navigator, click **Organization** .
2. Click the **Component Exchange** tab.
3. Select the **Enable organization-wide component hosting** check box.
4. Select the **On a custom server** option.
5. In **Server URL**, enter the server's URL.
6. In **Username** and **Password**, enter the access credentials.
7. Click **Test Connection**.
8. When the connection is successful, click **Save**.

Share the Component Exchange's URL

After defining a project or an external server as the Component Exchange, share its URL with developers who'll upload components.

If you've defined an external server as the Component Exchange, share its URL as-is with developers.

If you've defined a project as the Component Exchange, share its URL in the `https://
{hostname}/{org_id}/s/{project_id}/compcatalog/0.2.0/` format. These steps tell you how to construct the URL in the required format.

1. Open the Component Exchange project's **Project Home** page.
2. In the **Maven** section of the **Repositories** tab, click the **Project Repository's URLs** dropdown list and then click the HTTPS URL's **Copy**  icon.

Example:

The screenshot shows the Oracle Cloud developer console interface. At the top, there are tabs for 'Repositories', 'Statistics', and 'Team'. Below that, the 'Git' section has a 'Create Repository' button and a search bar. The 'Maven' section is shown below, with a 'Project Repository' URL listed: <https://myorg-vbs.developer.ocp.oc.com/profile/myorg-vb...>. This URL is highlighted with a red box. To the right of the URL is a 'Clone' button with a dropdown arrow.

3. Paste the copied URL in a text editor.
4. In the copied URL, remove the `profile/` and `maven/` elements.
For example, if the copied URL is `https://vbs-vbstest.developer.ocp.oraclecloud.com/profile/vbs-vbstest/s/vbs-vbstest_my-project_902/maven/`, then remove the elements marked in bold.
After removing the elements, the URL should be similar to this:
`https://vbs-vbstest.developer.ocp.oraclecloud.com/vbs-vbstest/s/vbs-vbstest_my-project_902/`
5. Append `compcatalog/0.2.0` at the end of the URL.
For example: `https://vbs-vbstest.developer.ocp.oraclecloud.com/vbs-vbstest/s/vbs-vbstest_my-project_902/compcatalog/0.2.0`
6. Share the URL with developers who'll upload components to the Component Exchange.

Workflow for Web Components

To find out how to create a web component; test it; use it in your application; build, package, and deploy it to the component exchange, see Typical Workflow for Working with Oracle JET Web Components in *Developing Applications with Oracle JET*.

Troubleshooting

These topics cover some common issues and how to address them.

Unable to Access Instance After Switching Users to Federated IDCS

If your users are federated in IDCS and they are no longer able to access Visual Builder Studio, you need to set the primary email address on their user account.

To set the primary email address on the user account:

1. Sign in to Oracle Cloud.
2. From the Infrastructure Console, click the navigation menu in the top left corner, expand **Platform Services**, and then click **Visual Builder** to open the Instances page.
3. In the Oracle Visual Builder console, click **Users**.
4. On the User Management page in the Oracle Visual Builder console, click **Identity Console**.
5. On the User Management page, click **Action** for the user account you want to change and select **Edit**.
6. Enter the primary email address for the user in the **User Name/Email** field.
7. Click **Save**.

Unable to See the "Edit Page in Visual Builder Studio" Option in an Oracle Cloud Application

If your users don't see the **Edit Page in Visual Builder Studio** option in their Oracle Cloud Application, it could be because:

- They don't have the right privileges to access VB Studio. See [Set Up VB Studio Users](#) and [Add Users to the Project](#).
- They may not be working in an environment that has an instance of VB Studio associated with it, such as a TEST environment. See [Before You Begin](#).
- Their Oracle Cloud Application hasn't yet adopted Oracle's new Redwood design pattern, so this page isn't extensible using VB Studio. In that case, refer them to *Oracle Applications Cloud: Configuring and Extending Applications* for instructions on how to customize their Oracle Cloud Applications with App Composer to meet their business needs.

 **Tip:**

If their application page URL contains .../faces/..., then they can't edit this page using VB Studio.

How Do I Resolve a "You Are Not a Member of This Organization" Error?

If your users see a You are not a member of this organization error when trying to access VB Studio from their Oracle Cloud Application, they probably haven't been granted either the required Oracle Cloud Applications roles or VB Studio roles.

See [Set Up VB Studio Users](#).

How Do I Access the IDCS Console From the Oracle Cloud Console?

If you don't have a direct URL to the IDCS console, you can access it from the Oracle Cloud Console instead.

1. In a web browser, go to <https://cloud.oracle.com>.
2. On the Sign-In page, in **Cloud Account Name**, enter your Oracle Cloud account or tenancy name and click **Next**.
3. Sign in with the user credentials that you received when you set up your account.
The Oracle Cloud Console, also called the OCI console, opens.
4. In the upper-left corner, click **Navigation Menu** .
5. Select **Identity & Security** and then select **Domains**.
6. Select the appropriate domain where your VB Studio instance is located.
The domain's Overview page is displayed.
7. From here, you can now complete IDCS tasks.

For example, you can assign VB Studio IDCS roles manually, which is a required step if IDCS role sync has been turned off. To do this, start with step 3 in [Assign VB Studio Roles in Oracle Identity Cloud Service \(IDCS\)](#).

Resolve Access Issues After Allow-Listing Your VB Studio IP Address

If you've added your VB Studio infrastructure IP address to the list of allowed IP addresses in your Web Application Firewall (WAF) and/or Location Based Access Control (LBAC) configuration and are now experiencing access issues, try one of these possible solutions:

- If you have LBAC enabled, but you haven't added your VB Studio IP address to the LBAC allow list, your users may receive error messages that say: "We are not able to determine the status of environment <environment name>. We couldn't retrieve the list of sandboxes from <instance name> due to :Forbidden."

- If you see an "Unauthorized" status in the Service Instances tab on VB Studio's Environments page, the issue *could be* related to LBAC or WAF. However, it could also be an issue with your credentials not having access to the FA status endpoints. Only certain endpoints, not all of them, are protected by the LBAC configuration.
- Ask the Oracle Cloud Applications administrator to enable Cross-Origin Resource Sharing (CORS) by adding the VB Studio's root URL to the Oracle Cloud Application's ORA_CORS_ORIGINS profile option. This option is used to specify which domains can talk to each other.

 **Note:**

Production to Test (P2T) and other operations that replace the database may cause the profile option value to change. If you perform any operation that replaces the database, you must repeat these steps.

To enable CORS, the Oracle Cloud Applications admin must:

1. Open VB Studio from the Oracle Cloud Applications instance and copy the VB Studio URL from the browser's address bar.
2. In the Oracle Cloud Applications instance, click **Navigator** .
3. Under **Others**, click **Setup and Maintenance**.
4. On the right side of the page, click **Tasks**,  then click **Search**.
5. In the search box, enter **Manage Administrator Profile Values** and click the Search icon.
6. In the search results, click **Manage Administrator Profile Values**.
7. In Profile Options Code, enter **ORA_CORS_ORIGINS**, then click Search.
8. Under **ORA_CORS_ORIGINS: Profile Values**, click **New**.
9. In Profile Level, select **Site**.
10. In Profile Value, enter '**'self'**' (including the single quotes), followed by VB Studio's root URL. For example: '**'self'** <https://abcd-test-DEVCSSAPP-07012210-2070-abcd.developer.ocp.oraclecloud.com>

 **Note:**

Don't include port numbers, wild cards, or extraneous characters after the root. For example, if the VB Studio's full URL is <https://abcd-test-DEVCSSAPP-07012210-2070-abcd.developer.ocp.oraclecloud.com:443/abcd-test-devcsapp-07012210-2070-abcd>, the root is simply <https://abcd-test-DEVCSSAPP-07012210-2070-abcd.developer.ocp.oraclecloud.com>.

Your screen should look something like this:

The screenshot shows a web-based configuration interface for Oracle Database profile values. The title bar reads "ORA_CORS_ORIGINS: Profile Values". Below the title are standard navigation buttons: Actions (dropdown), View (dropdown), New (+), Delete (X), and Edit (pencil icon). The main area has a table header with columns: "Profile Level" (with a dropdown arrow), "Product Name", "User Name", and "Profile Value". A sub-table below is titled "Site" and contains a single row with the value "self https://abcd-test-DEVCSSAPP-07012210-2070-ab".

11. Before you finish the CORS setup, you should check the CORS headers to be added. Although all of the other CORS options should be configured by default, in every pod, some of these settings may be missing or may have been erased inadvertently. See [Configure Cross-Origin Resource Sharing](#) for additional information about all the settings that can be configured, including a list of acceptable values for each option.
 12. Click **Save and Close**.
- If you are still experiencing problems, contact your Oracle Support representative and file an SR so Oracle can assist you in setting up access to VB Studio.

Appendices

These appendices provide information about creating and setting up a non-native VB Studio instance, migrating Oracle Cloud Applications configurations, and a list of all IP addresses by region and data center:

Topics:

- [Set Up Oracle Visual Builder Studio \(Non-Native\)](#)
- [Migrate Oracle Cloud Applications Configurations](#)
- [Public IP Addresses for VB Studio](#)

A

Set Up Oracle Visual Builder Studio (Non-Native)

Note:

This appendix explains how to create and set up your VB Studio instance if you are running VB Studio non-natively on underpinnings that are not OCI Generation 2. All the information that applies to the non-native setup and configuration is self-contained in this appendix.

Like other Oracle Cloud services, you must create an instance of Oracle Visual Builder Studio (VB Studio) before you can start using it. You can create only one VB Studio instance in an Oracle Cloud account.

VB Studio Generation 2 refers to VB Studio running natively on Oracle Cloud Generation 2 Infrastructure. This appendix addresses setting up VB Studio that is **not running natively** on Oracle Cloud Infrastructure.

Note:

It's important that you determine if you're running on Gen 2 so you know which chapter to read for set-up instructions. Let's do a quick test: Open the OCI console, then open Developer Services. Scan the list for VB Studio. If you see it, that means you are running natively on OCI, and should refer to [Create an Oracle Visual Builder Studio Generation 2 Instance](#) instead of this appendix. If you don't see it, then this appendix is the one you need.

Here's a summary of the initial set up steps for VB Studio:

To perform this action:	See this:	Why do I need to perform this action?
1. Get the right IDCS roles.	Get the Right IDCS Roles	To create and set up the VB Studio instance, you must be assigned some specific IDCS roles.
2. Create the VB Studio instance.	Create the VB Studio Instance	To use any Oracle Cloud service, you must first create an instance of the service. If you're an Oracle Cloud Applications user, you don't need to create a VB Studio instance as an instance is provisioned with your first Oracle Cloud Applications instance.

To perform this action:	See this:	Why do I need to perform this action?
3. Set up OCI connections.	Configure Your OCI Resources for CI/CD	You need a connection to OCI Compute VMs because this is where your VB Studio builds will run.
4. Set the build system up for CI/CD.	Set Up VB Studio for CI/CD	Builds in VB Studio run on VM build executors. Only VB Studio administrators can create and manage the VM build executors.
5. (Optional) Set your user preferences.	Set Your User Preferences	To change your display name, set an avatar image, change the email address, or set email notification preferences.

After you finish creating an instance and setting it up, you will need to do some additional set up if you are developing visual applications or extending Oracle Cloud applications.

 **Note:**

If you are developing other types of applications using popular build frameworks such as Maven, Gradle, Ant, and Node.js; or standard infrastructure interfaces such as Docker, Kubernetes, and Terraform, you are ready to start building your application.

To perform this action:	See this:
Set up VB Studio to customize Oracle Cloud Applications through extensions	Set Up VB Studio to Extend Oracle Cloud Applications
Set up VB Studio to develop visual applications and deploy them to Visual Builder instances	Set Up VB Studio for Developing Visual Applications

Get the Right IDCS Roles

If you're the Oracle Cloud account subscriber, you're assigned the required roles to set up the VB Studio instance. If you're not the subscriber, ensure that you are assigned the correct roles in Oracle Identity Cloud Service (IDCS).

You must be assigned this role:	To ...
Cloud Account Administrator	Set up Oracle Cloud Infrastructure (OCI) compartments and buckets. OCI compartments and buckets are required to set up the VB Studio build system. If you're not an account administrator, contact the administrator to add you to the OCI_Administrators IDCS group.
Identity Domain Administrator or User Administrator	Add users and assign IDCS roles. If you're not an administrator, make sure that your OCI group is assigned a policy like this: Allow group MyGroup to inspect identity-providers in tenancy

You must be assigned this role:	To ...
DCS_INSTANCE_ENTITLEMENT_ADMINISTRATOR OR	Create the VB Studio instance. If you're not the account administrator or the IDCS administrator, contact the IDCS admin to assign you this instance role. Without this role you won't see the user interface to provision the VB Studio instance.

Create the VB Studio Instance

You can create only one VB Studio instance in an Oracle Cloud account, so make sure you don't already have one before you get started.

If you're an Oracle Cloud Applications user, you don't need to create a VB Studio instance as an instance is provisioned with your first Oracle Cloud Applications *-TEST instance. To access it, [get the required roles](#) and [open VB Studio from your Oracle Cloud Applications instance](#).

Note:

Only one VB Studio instance can be provisioned in an Oracle Cloud account. By default, a VB Studio instance is provisioned with your first Oracle Cloud Applications TEST instance that your organization's members can use to develop application extensions. If you wish to use VB Studio with a different Oracle Cloud Applications TEST or DEV instance, file a service request so that we can terminate the VB Studio instance associated with your current *-TEST or *-DEV instance, create a new one, and associate this new one to your preferred instance. (Provisioning VB Studio for PROD instances is not supported.) Only one move between pods is allowed, unless a pod must be terminated for some reason and therefore requires another association.

If you're not an Oracle Cloud Applications user, follow these steps to create the VB Studio instance:

1. In a web browser, go to <https://cloud.oracle.com>.

To view the list of supported browsers, see https://docs.oracle.com/en-us/iaas/Content/GSG/Tasks/signingin.htm#supported_browsers.

2. In **Cloud Account Name**, enter your Oracle Cloud account or tenancy name and click **Next**.
3. On the Single Sign-On (SSO) panel, if required, select your identity provider and click **Continue**.
4. Enter your Oracle Cloud account credentials and click **Sign In**.

The Oracle Cloud Console, also called the OCI console, opens. If you've recently created your Oracle Cloud account, wait for some time to see your services in the Oracle Cloud Console.

5. In the upper-left corner, click **Navigation Menu** .
6. Select **OCI Classic Services** and then select **Developer**.

Here's an example:

The screenshot shows the Oracle Cloud OCI Classic Services console. On the left, there's a sidebar with various service categories like Home, Compute, Storage, etc. Below that is a section for 'OCI Classic Services' which includes options like Platform Services, Analytics Classic, API Platform, etc. The 'Developer' option is highlighted with a red box. At the bottom of the sidebar, there's a button labeled 'OCI Classic Services'.

Platform Services	Classic Data Management...
Analytics Classic	Big Data - Compute Edition
API Platform	Database Classic
API Platform Classic	Data Hub
Apairy	Event Hub
Application Container	Event Hub - Dedicated
Blockchain Platform	Exadata Classic
Container	MySQL
Container Pipelines	Classic Infrastructure Services
Content and Experience	Compute Classic
Content and Experience Classic	Storage Classic
Data Integration Platform	Load Balancer Classic
Data Integration Platform Classic	
Developer	
Goldengate	
Integration Classic	
IoT Enterprise	
Java	
Management Cloud	
Mobile	

If you don't see the **Developer** option, make sure you're assigned the [required IDCS roles](#).

7. In the **Instances** tab, click **Create Instance**.
8. On the Create New Instance page, in **Instance**, enter a unique name. In **Description**, enter a description.

The name helps you to identify the service instance.

9. In **Notification Email**, enter your email address where you'd like to receive email notification when the instance is ready.

10. In **Region**, select your home region.

You'll find your region on the OCI console's header.

11. Click **Next**.

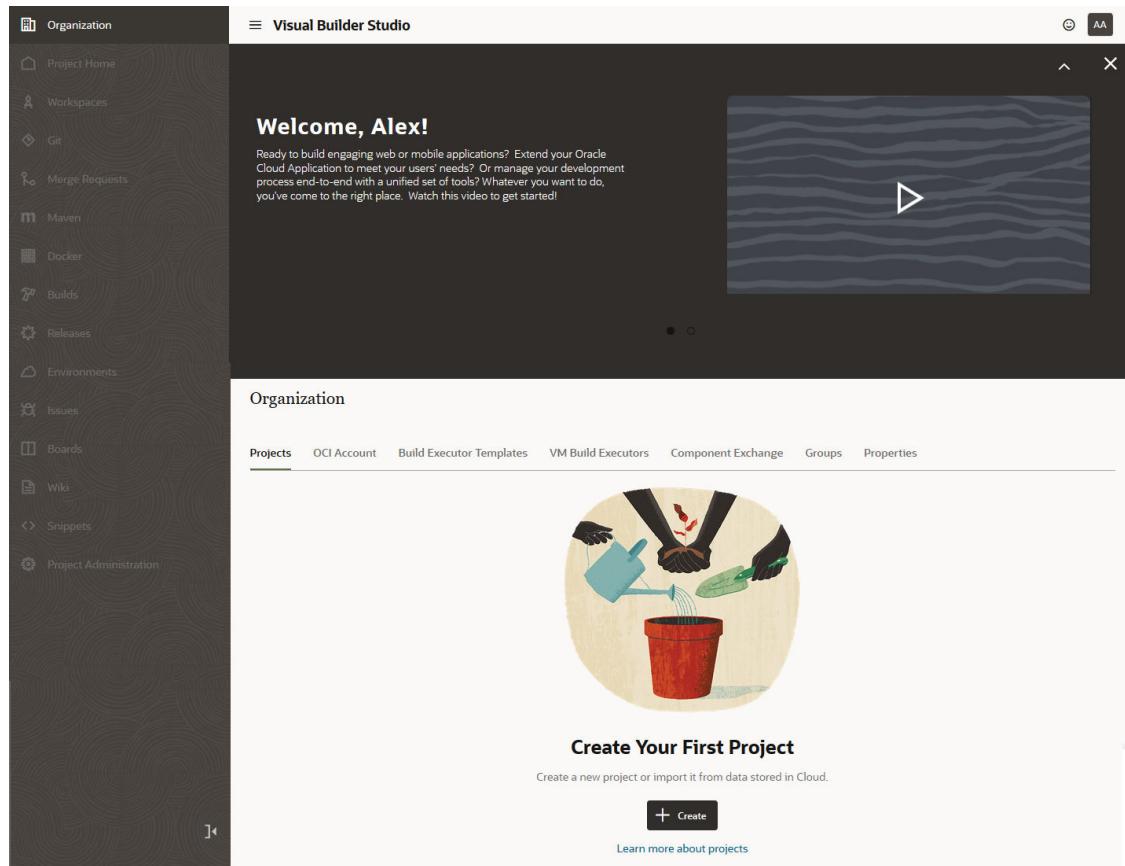
12. On the Service Details page, verify the entered details and click **Next**.

13. On the Confirmation page, click **Create**.

Expand the **Instance Create and Delete History** section to track the status.

Once the service instance is created, click **Action**  and then select **Access Service Instance** to open VB Studio's Organization page.

Here's an example:



After you create the instance:

- You're assigned the DEVELOPER_ADMINISTRATOR IDCS role, which enables you to set up VB Studio, manage the VM build executors, and update the organization details. After you're assigned the role, you're also called the VB Studio's *Organization Administrator*.
- You receive an email with the subject **Verify your Oracle Visual Builder Studio email**. Open the email and click the URL link in the email body to verify your email. This is required to receive email notifications from the service. After verifying the email address, you'll receive another email with the subject **Welcome to Oracle Visual Builder Studio**. This email contains the VB Studio URL that you can bookmark.
- You should configure VB Studio to connect to OCI and set up the build system.

Access VB Studio from the Oracle Cloud Home Page

If you've signed out of the Oracle Cloud or VB Studio, follow these steps to sign back in.

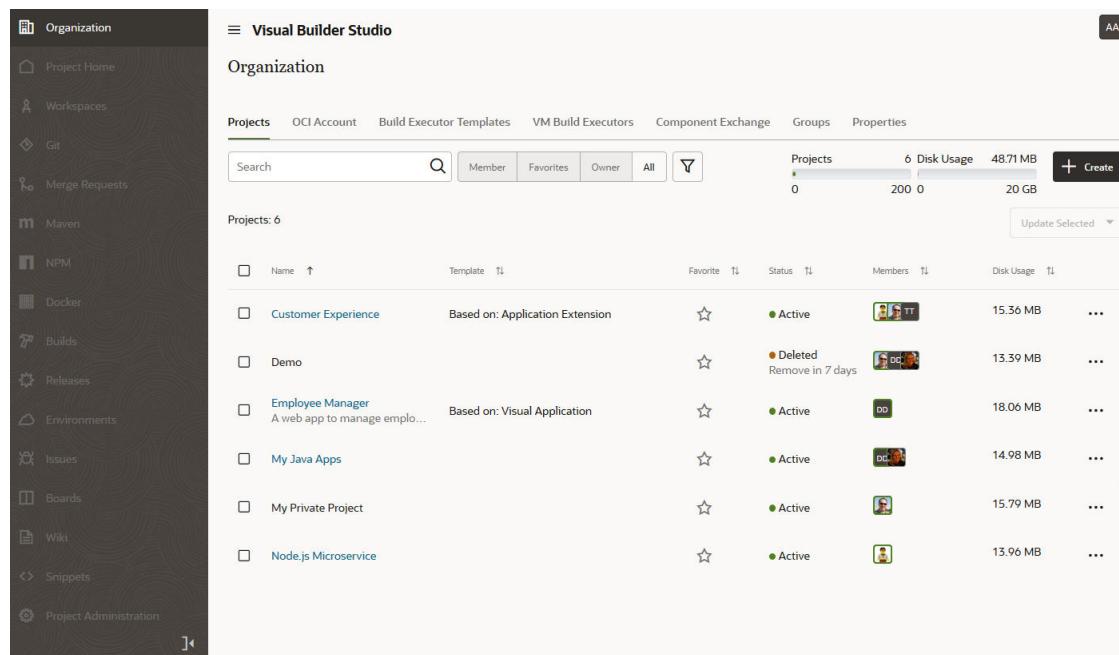
1. In a web browser, go to <https://cloud.oracle.com>.
2. On the Sign-In page, in **Cloud Account Name**, enter your Oracle Cloud account or tenancy name and click **Next**.
3. On the Single Sign-On (SSO) panel, click **Continue**.
4. Enter your Oracle Cloud account credentials and click **Sign In**.

The Oracle Cloud Console, also called the OCI console, opens.

5. In the upper-left corner, click **Navigation Menu** .
 6. Select **OCI Classic Services** and then select **Developer**.
- If you don't see the **Developer** option, make sure you're assigned the required IDCS roles.
7. In the **Instances** tab, click **Manage this instance**  and select **Access service instance**.

The VB Studio **Organization** page opens, which displays all the projects you're a member of, as well as your favorite projects, the projects you own, and all the shared projects in your organization. You can also set up OCI connections, virtual machines, and the properties of the organization.

Here's an example of the Organization page with some projects:



The screenshot shows the Visual Builder Studio Organization page. On the left is a dark sidebar with various project management options like Project Home, Workspaces, Git, Merge Requests, Maven, NPM, Docker, Builds, Releases, Environments, Issues, Boards, Wiki, Snippets, and Project Administration. The main area has a header "Visual Builder Studio" and "Organization". Below is a table titled "Projects" with the following data:

Name	Template	Favorite	Status	Members	Disk Usage	Actions
Customer Experience	Based on: Application Extension		Active		15.36 MB	...
Demo			Deleted Remove in 7 days		13.39 MB	...
Employee Manager	A web app to manage emplo...		Active		18.06 MB	...
My Java Apps			Active		14.98 MB	...
My Private Project			Active		15.79 MB	...
Node.js Microservice			Active		13.96 MB	...

Configure Your OCI Resources for CI/CD

To allow your organization's members to run CI/CD builds, you should configure VB Studio to connect to an OCI account.

Before you do that, set up your OCI account. Open the OCI console and create a compartment, a group and a user to access the compartment, and a policy that defines access to the compartment resources. See [Set Up the OCI Account](#). After creating the resources, get their details as described in [Get the Required OCI Input Values](#) because you'll need them to [Set Up the OCI Connection in VB Studio](#).

You can use the root compartment and the tenancy user that was created when the OCI account was created, but it's recommended to create a dedicated compartment to host VB Studio resources. This allows you to organize VB Studio resources better because they aren't mixed with the other resources of your tenancy. You can also restrict users and control read-write access to the compartment without affecting other resources. To learn more about compartments, see [Understanding Compartments](#).

 **Note:**

If you use Oracle Cloud OS Management Service (OSMS), don't configure VB Studio to use the OSMS compartment, or a compartment with an OSMS policy. OSMS compartments aren't compatible with VB Studio VM build executors.

If you don't have authorization to create and manage OCI resources, ask some one who can create the resources and share their details.

Oracle Cloud Infrastructure: Terms of Use

VB Studio requires access to your OCI Compute, Network, and Object Storage resources. For example, VB Studio build executors run on OCI Compute virtual machines (VMs) in your virtual cloud network (VCN), and artifacts for your VB Studio projects are stored in an OCI Object Storage bucket.

Here are some special legal terms and guidance that apply to the usage of OCI resources:

- You must create a dedicated OCI compartment for use by VB Studio.
- You must create a dedicated OCI Identity and Access Management (IAM) user, group, and policies for use by VB Studio. When creating a username, avoid including personal names or personal information (like birthdays).
- You should allow only the dedicated OCI IAM user to access the VB Studio compartment. Be aware that a user with the read-write access to the VB Studio compartment can view, create, update, and delete resources such as VM Compute instances, storage volumes, security policies, and other users. Failure to maintain security best practices may create a high risk for you and your organization.
- Notwithstanding VB Studio terms, you acknowledge that Oracle isn't responsible or liable for any action you take in accessing or creating access to VB Studio.

Set Up the OCI Account

1. On the OCI console, in the upper-left corner, click **Navigation Menu** .
2. Select **Identity & Security** and then under **Identity**, select **Compartments**.
3. On the Compartments page, create a compartment to host VB Studio resources.
 - a. To create the compartment in the tenancy (root compartment), click **Create Compartment**.
 - b. In the Create Compartment dialog box, fill in the fields and click **Create Compartment**.

To learn more about compartments, see [Managing Compartments](#).

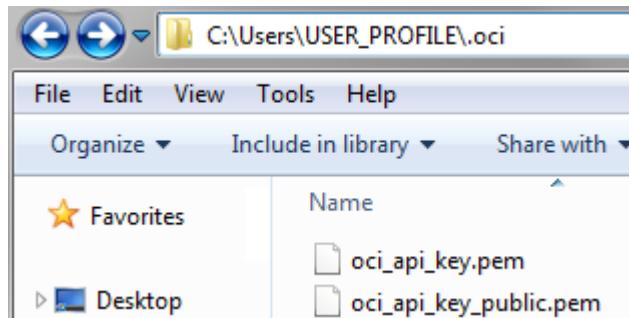
4. Create a user to access the VB Studio compartment.
 - a. In the navigation menu, select **Identity & Security**. Under **Identity**, select **Users**.
 - b. Click **Create User**.
 - c. Select the **IAM** user type.
 - d. In the Create User dialog box, fill in the fields and click **Create**.

To learn more about OCI users, see [Managing Users](#).

5. On your computer, generate a private-public key pair in the PEM format.

To find out how to generate a private-public key pair in the PEM format, see [How to Generate an API Signing Key](#).

Here's an example of private-public key files on a Windows computer:



6. Upload the public key to the user's details page.

- a. Open the public key file in a text editor and copy its contents.
- b. In the navigation menu, select **Identity & Security**. Under **Identity**, select **Users**.
- c. Click the user's name created in Step 4.
- d. Under **Resources**, click **API Keys**.
- e. Click **Add API Key**.
- f. In the Add API Key dialog box, select the **Paste Public Key** option, paste the contents of the public key file, then click **Add**.

To learn more about uploading keys, see [How to Upload the Public Key](#).

7. On the Groups page, create a group for the user who can access the VB Studio compartment and add the user to the group.
 - a. In the navigation menu, select **Identity & Security**. Under **Identity**, select **Groups**.
 - b. Click **Create Group**.
 - c. In the Create Group dialog box, fill in the fields and click **Create**.
 - d. On the Groups page, click the group's name.
 - e. On the Group Details page, click **Add User to Group**.
 - f. In the Add User to Group dialog box, select the user created in Step 4 and click **Add**.

To learn more about groups, see [Managing Groups](#).

8. In the root compartment, not the VB Studio compartment, create a policy to allow the group created in step 6 to access the VB Studio compartment.
 - a. In the navigation menu, select **Identity & Security**. Under **Identity**, select **Policies**.
 - b. On the left side of the Policies page, from the **Compartment** list, select the root compartment.
 - c. Click **Create Policy**.
 - d. In **Name** and **Description**, enter a unique name and a description.
 - e. In **Compartment**, select the root compartment.

f. In **Policy Builder**, click **Show manual editor** and add these statements:

- allow group <group-name> to manage all-resources in compartment <compartment-name>
This grants all permissions to the VB Studio group users to manage all resources within the VB Studio compartment.
- allow group <group-name> to read all-resources in tenancy
This grants read permissions to the VB Studio group so that its users can read—but not use, create or modify—all resources inside and outside the VB Studio compartment. The group users can't use, create, or modify the resources.

Here's an example:

The screenshot shows the 'Create Policy' page. The policy details are as follows:

- Name:** VBStudioPolicies
- Description:** VB Studio policies
- Compartment:** myaccount (root)
- Policy Builder:** Show manual editor (switch is on)
- Statements:**

```
allow group VBStudioGroup to manage all-resources in compartment VBStudioCompartm
allow group VBStudioGroup to read all-resources in tenancy
```

At the bottom, there is an example text: "Example: Allow group [group_name] to [verb] [resource-type] in compartment [compartment_name] where [condition]".

g. Click **Create**.

To learn more about policies, see [Managing Policies](#).

Get the Required OCI Input Values

Every Oracle Cloud Infrastructure resource has an Oracle-assigned unique ID called an Oracle Cloud Identifier (OCID).

To connect to OCI, you need the account's tenancy OCID, home region, the OCID of the compartment that hosts VB Studio resources, and the OCID and the fingerprint of the user who can access the VB Studio compartment. To connect to OCI Object Storage, you need the Storage namespace. You can get these values from the OCI Console pages.

This table describes how to get the OCI input values required for the connection.

To get these values ...	Do this:
Tenancy OCID, Home Region, and Storage Namespace	<ol style="list-style-type: none"> 1. Open the OCI console and in the navigation menu, select Governance & Administration. 2. Under Account Management, select Tenancy Details. The Tenancy Information tab displays the tenancy details in these fields: <ul style="list-style-type: none"> • OCID: tenancy OCID • Home Region: home region • Object Storage Namespace: storage namespace
User OCID and Fingerprint	<ol style="list-style-type: none"> 1. Open the OCI console, and in the navigation menu, select Identity & Security. 2. Under Identity, select Users. 3. On the Users page, click the user's name. <ul style="list-style-type: none"> • The User Information tab displays the user OCID in the OCID field. Click the Copy link to copy the OCID to the clipboard. • To get the fingerprint of the public key associated with your OCI account, scroll down to Resources, select API Keys, and copy the fingerprint value.
Compartment OCID	<ol style="list-style-type: none"> 1. Open the OCI console, and in the navigation menu, select Identity & Security. 2. Under Identity, select Compartments. 3. On the Compartments page, click the compartment's name. The Compartment Information tab displays the compartment's OCID in the OCID field. 4. Click the Copy link to copy the compartment's OCID to the clipboard.

Set Up the OCI Connection in VB Studio

To connect to OCI, you will need to first need to get the VB Studio compartment's details, user details, and the required OCID values. You will need to leave VB Studio and open the OCI console in order to get the details you need. If you're not the OCI administrator, get the details from the OCI administrator.

After you have the OCI input values, you can create an OCI connection from VB Studio.

1. In the left navigator, click **Organization** .
2. Click the **OCI Account** tab.
3. Click **Connect**.
4. In **Tenancy OCID**, enter the tenancy's OCID copied from the Tenancy Details page.
5. In **User OCID**, enter the OCID of the user who can access the VB Studio compartment.
6. In **Home Region**, select the home region of the OCI account.
7. In **Private Key**, enter the user's private key who can access the VB Studio compartment.

The private key file was generated and saved on your computer when you created the private-public key pair in the PEM format. See Step 5 in [Set Up the OCI Account](#).

Make sure that the private key you enter contains the -----BEGIN RSA PRIVATE KEY----- and -----END RSA PRIVATE KEY----- markers.

8. In **Passphrase**, enter the passphrase used to encrypt the private key. If no passphrase was used, leave the field empty.
9. When you enter a private key and a passphrase, the **Fingerprint** field is automatically populated. Ensure that the automatically populated fingerprint value matches the fingerprint value of your private-public key pair. If it doesn't, update it to enter the correct value.
10. In **Compartment OCID**, enter the compartment's OCID copied from the Compartments page.
11. In **Storage Namespace**, enter the **Object Storage Namespace** value copied from the Tenancy Details page.
12. Select the terms and conditions check box.
13. Click **Validate**.
14. After validating the connection details, click **Save**.

Here's an example of an **OCI Account** tab filled with required OCI details.

Configure OCI Account

X

⚠ Fill in these fields to set up your new OCI account. Existing build artifacts in Object Storage won't be migrated, but new artifacts will be created when your build jobs start running again.

OCI Credentials

Enter your OCI credentials below. We will use this account for storing artifacts and running builds.

Tenancy OCID
 ocid1.tenancy.oc1..abcdefgij12345k0l7m8n9o0abcd1234efgh5678xyz321zyx123abcdefg

User OCID
 ocid1.user.oc1..abcdefghi09876jklm5n6o7pq8rst901ghi0n6o7pabcfklmzyxuvwtrsp

Home Region
 us-ashburn-1

Private Key
 -----BEGIN RSA PRIVATE KEY-----
 MIIABCD123efgh456XYZ321UAT654ORA1290A2b3C4d5hym28a8b1c0AyugOFsdg
 ...
 cbNF375h59hjgfdDGN4n5ji9J85VH75544FGHHJk9HJK98seFe45Da==
 -----END RSA PRIVATE KEY-----

Passphrase

Fingerprint
 a1:b2:c3:d4:e5:f6:1e:2d:3c:ab:5a:6b:7c:8d:9e:0a

Compartment OCID
 ocid1.compartment.oc1.iad.abcdefghiaaaaaabbbbbbb09876m8n9o0abcd1234901ghi0n6o7p7gg64q

Storage Namespace
 myaccount

* Visual Builder Studio requires these credentials to use the Compute, Network and Object Storage resources of your OCI account. This usage may be extensive. By checking the box, you acknowledge that you have read and accepted the terms specified [here](#).

Validate
Compute Connection Successful
Storage Connection Successful
Cancel
Save

[Help me find this information](#)
[Go to my OCI Accounts](#)

Manage Your VB Studio Instance

After creating the VB Studio instance, you can't rename it but you can change your OCI resources or switch to another account.

Verify Your Compute and Storage Connections

If you notice any Compute or Storage issues, verify your OCI connection.

1. In the left navigator, click **Organization** .
2. Click the **OCI Account** tab.
3. Click **Validate**.

ORACLE®

A-12

If the validation fails, click **Edit** to edit the connection, or switch to another OCI account. If your OCI account has reached Compute and Storage service limits, contact the OCI administrator to [increase the service limit](#).

Change Your OCI Account

If you are using the built-in free account or your own OCI account, you can switch to another OCI account any time and use the new account's resources.

When you switch to another OCI account:

- Your current OCI account's build executor templates and their metadata migrate to the new OCI account. For example, if you have 10 executor templates in your current OCI account, VB Studio creates the same number of executor templates with the same software in the new OCI account.
- Your current OCI account's VM build executors and their metadata migrate to the new OCI account. For example, if you have 234 VM executors in your current OCI account, VB Studio creates the same number of VM executors of the same shape, VCN, and executor templates in the new OCI account.
- Your current OCI account's Object Storage buckets do not migrate to the new account. The data in buckets may include build artifacts, Maven artifacts, NPM artifacts, and exported project data.

If you switch to the built-in free account, you'll lose access to data in Object Storage, and your current account's executor templates and VM executors. Instead, you'll get access to only one free VM build executor and the default executor template.

Switch to Another OCI Account

To switch to another OCI account, [get the required OCI input values](#) and follow these steps:

1. In the left navigator, click **Organization** .
2. Click the **OCI Account** tab.
3. Click **Edit**.
4. Enter the OCID details of the new OCI account and click **Validate**.
5. After successful validation, click **Save**.

After you save, build executor templates and VM build executors of your old OCI account migrate to the new OCI account.

Switch to an OCI Account From the Built-In Free Account

To switch to an OCI account, [get the required OCI input values](#) and follow these steps:

1. In the left navigator, click **Organization** .
2. Click the **OCI Account** tab.
3. Click **Connect OCI Account**.
4. In **Account Type**, select **OCI**.
5. Enter the OCID details of the new OCI account as described in [Set Up the OCI Connection in VB Studio](#) and click **Validate**.
6. After successful validation, click **Save**.

After you save, the build executor template and the free VM build executor of your built-in free account migrate to the OCI account.

Switch to the Built-In Free Account From an OCI Account

If you switch to the built-in free account, build executor templates and VM build executors of your current OCI account are deleted and you get access to only one free VM build executor with fixed software in the default build executor template. You can't add more VM executors or edit the executor template's software.

1. In the left navigator, click **Organization** .
2. Click the **OCI Account** tab.
3. Click **Edit**.
4. Deselect the terms and conditions check box, and click **Remove Account**.
5. In the Confirm Removal of OCI Account dialog box, click **Remove Account**.

After you remove your OCI account, you're automatically connected to the built-in free account.

Migrate Oracle Cloud Applications Configurations

Configurations are changes saved in an Oracle Cloud Applications sandbox. A configuration set is an archive file that contains all your configurations across all product families. When you migrate configurations, you export a configuration set from the source Oracle Cloud Applications instance (such as your development or test instance) and import it to a target Oracle Cloud Applications instance (such as your production instance).

If you're new to migration, see [Configuration Life Cycle](#) and [Migration](#) in *Configuring and Extending Applications* to learn about configuration migration.

You can export and import configurations manually from the Oracle Cloud Applications instance's Migration page, or configure Configuration Set Migration (CSM) build jobs and pipelines in VB Studio to automate the export and import process. To learn more about VB Studio build jobs and pipelines, see [What Are Jobs and Builds?](#) and [What is a Pipeline?](#).

This document describes how to configure VB Studio build jobs and pipelines to migrate configurations from the source development instance to a sandbox on the target production instance. This document also describes how to deploy your extension to the production instance's sandbox so you can test it with the migrated configurations.

Before You Configure CSM Build Jobs and Pipelines

Here are some best practices to follow before you configure and run CSM build jobs and pipelines:

General Considerations

- Synchronize the source and target environments to ensure successful configuration migration:
 - Always ensure that the source and target instances are of the same release, with the same standard and one-off patches applied to both instances.
 - In the source instance, register the target instance. See [Migration](#) in *Configuring and Extending Applications*.
 - Make sure that both the source and target instances are synchronized. To sync them, make application changes to the source instance, test the changes, then migrate them to the target instance.
 - Create all configurations in the source instance, then migrate them to the target instance. Never create configurations directly on the target instance.
 - Test all application changes and configurations that were made in the source instance before starting with the migration process.
 - Ensure that the target instance doesn't have any re-keyed configurations or ones that come from another source. If the source and target instances differ, re-provision the target instance from same common ancestor that provisioned the source instance.

 **Note:**

If these guidelines aren't closely adhered to, CSM Migration-related issues and Import/Export-related errors will be encountered. Also, in some instances, your source and target environments will be out of sync and you have to perform a P2T (Production-to-Test) operation to restore order to your environments.

- Always create new and unverified configurations in a sandbox. The migration process will only migrate configurations that have been created in a published sandbox.
 - If you make configuration changes on a sandbox and don't publish to mainline, the configuration changes won't be included in the migration package.
- You should use CSM to migrate lookups and menu configurations, instead of existing Functional Setup Manager (FSM) tasks.
- Functional security artifacts for standard objects are not migrated.
 - If you migrate any functional security associated with roles in the source to a target instance, and the corresponding role doesn't exist in the target instance, an error will occur on import.
 - To avoid these errors, you can selectively create these roles in the target instance when you import your migration set. See <http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=OSCUS3429288> in Oracle CX: Securing CX Sales and B2B Service.
- After you apply configurations, Oracle Cloud Application end users won't see any changes to the user interface until they log out and log back in.

Delta Migration

- A normal migration flow migrates everything. Because this type of migration contains everything, it can be time-consuming. Think of it as an all-or-nothing proposition. A Delta Migration, on the other hand, only migrates what you've changed since the last time you migrated. By only migrating the latest set of changes, you can potentially save considerable execution time during the migration.
- Note that you can't selectively pick and choose which of your latest changes should get migrated. Delta migration takes all of the changes since last time. Yes, you can select which modules you want to migrate, but you have to take all changes within the module.
- To perform a delta migration, use the sandbox feature in both the source and target instances. See [Configuration Life Cycle](#) in *Configuring and Extending Applications*.

Roles Migration

- If you need to review and create roles on the target instance, don't run the migration pipeline. Run the export configurations job manually, create the required roles on the target instance, and then manually run the import configurations job. See [Migrate Your Configurations](#).
- Get the access credentials of an Oracle Cloud Applications user who can connect and deploy to the target instance.

Application Extensions Using Visual Builder Studio

- In VB Studio, [create a project for application extensions](#).
- In your VB Studio project, make sure you have environments that point to both source and target Oracle Cloud Applications instances. See [Add an Oracle Cloud Application's Instance to an Environment](#) in *Administering Visual Builder Studio*.

- If you want to import the configuration set into multiple target instances, create an import job for each target instance.
- When build jobs are running, don't export configurations manually from the source instance or make security changes.
- After you have triggered the migration pipeline, do not cancel it or retry it if it seems slow. Check the build job's log to monitor the migration status.

To deploy your extension to the Oracle Cloud Applications production instance's sandbox, you'll create these jobs and pipelines:

- A packaging job that packages the extension for the Oracle Cloud Applications production instance's sandbox.
- A deployment job that deploys the extension to the Oracle Cloud Applications production instance's sandbox.

Here are some tips that help you speed up your migration process:

- Perform a delta migration first, as it reduces the migration set size and reduces the time to migrate configurations.
To do that, configure your export configurations job to move only the latest and incremental changes.
- To reduce failures, include only the required optional modules in the configuration data set and exclude the modules that have no application changes.
The configuration set migration process involves migrating components, such as CRM, ADF, BI, SOA, as well as workspaces. If your base application has CRM and BI components, for example, but you don't have any BI or Analytics related changes, exclude the BI component from the configuration set when you configure your export configurations job.

Create and Configure Build Jobs to Migrate Your Oracle Cloud Application's Configurations

You need to set up build jobs to export and import configurations. This topic explains how to do that.

To do this:	See this:
Generate the configuration set from the source Oracle Cloud Applications instance and create the configuration set ID. The configuration set ID is required by other CSM build jobs.	Configure a Job to Export Configuration
Import the generated configuration set from source Oracle Cloud Applications instance to the target Oracle Cloud Applications instance	Configure a Job to Import Configuration
Delete the imported configuration set	Configure a Job to Cancel the Configuration Migration
Apply and save the imported configuration set on the target instance	Configure a Job to Apply and Save the Configuration Migration
Restore the target Oracle Cloud Applications instance to the state before you applied the configuration set	Configure a Job to Restore Configuration Migration

After creating jobs, [create and configure a CSM pipeline](#) to run the jobs in a sequence.

Configure a Job to Export Configuration

Configure a job to generate the outbound configuration set from the source Oracle Cloud Applications instance. This job will also generate a unique configuration set ID that is required by other CSM build jobs.

1. Open your VB Studio project.
2. In the left navigator, click **Builds** .
3. In the **Jobs** tab, click **+ Create Job**.
4. In the New Job dialog box, in **Name**, enter a unique name.
5. In **Description**, enter the job's description.
6. In **Template**, select the **System Default OL7 for Visual Builder** template.
7. Click **Create**.
8. (Optional) Click **Configure** .
9. Click the **Steps** tab.
10. From **Add Step**, select **Configuration Set Migration**, and then select **Export Data**.
11. In **Source Instance**, select the environment that points to the Oracle Cloud Application's source instance.
12. (Optional) In **ID Parameter Name**, if necessary, change the parameter name for the configuration set. By default, it's set to `CONFIGURATION_SET_ID`.

When a build runs, the job generates a unique configuration set ID, assigns it to the parameter name defined in **ID Parameter Name**, and then passes the parameter to the downstream jobs when the CSM jobs run in a pipeline. The downstream jobs can access the parameter's value by calling it in the `$<PARAMETER_NAME>` or `${<PARAMETER_NAME>}` format.

If you change the parameter name, note it down. You'll need it when you configure other CSM jobs.

13. In **CSM/Sandbox Name**, enter a name for the configuration set.

When you import configurations later, the target Oracle Cloud Applications instance will create an internal sandbox with this name.

14. (Optional) In **Description**, enter a description for the configuration set.
15. In **Username** and **Password**, enter the credentials of an IDCS user who is also an Oracle Cloud Applications user.
16. (Optional) Under **Optional Modules**, specify the optional modules you want to export. By default, all modules are included.

To specify a few modules only, deselect the **Include all modules** check box. In the Search Optional Modules dialog box that opens, enter the credentials of an IDCS user who is also an Oracle Cloud Applications user and click **Search Optional Modules** . Select the check boxes of optional modules you want to include and click **OK**.

17. (Optional) To export configurations if the target isn't registered on the source Oracle Cloud Applications instance, select the **Create migration set even if target isn't registered** check box.
18. (Optional) By default, the migration will move the latest application changes. To move all application changes, select the **Move all changes** check box.

19. Click **Save**.

When you run a build of this job, open the build's log to note the configuration set ID. Here's an example:

```
[2020-07-01 08:00:00] Build scheduled. Build started by pipeline CSM
[2020-07-01 08:00:00] Build task id: a1b2c123-abc1-12d3-6789-5bc4d76531f2
.
.
.
[2020-07-01 08:01:35] Status: Export Started
[2020-07-01 08:01:35] Configuration Migration Set ID is 1239873457659876
[2020-07-01 08:01:39] IN_PROGRESS...
[2020-07-01 08:30:02] SUCCESS...
[2020-07-01 08:35:10] CREATED...
.
.
.
[2020-07-01 09:30:10] 2020-07-01 08:36:02.990: The outbound migration of CRM common objects completed successfully.
[2020-07-01 09:35:10] === End CSM Export Execution ===
Executor log size 4.6 KB (4,558)
[2020-07-01 09:35:12]
[2020-07-01 09:35:12] Build completed.
[2020-07-01 09:35:12] Status: DONE Result: SUCCESSFUL Duration: 35 min
```

After a successful build, if you want to view details of the exported migration set or download it, open the source Oracle Cloud Applications instance's Migration page and click the **Outgoing** tab. You don't need to download it if you plan to use the import configuration job to import configurations.

Here's an example of the Migration page:

The screenshot shows the Oracle Cloud Applications Migration page. The left sidebar has tabs for 'Incoming' (selected), 'Outgoing' (highlighted in blue), and 'Environment Info'. The 'Outgoing' tab shows a message: 'Your migration set is ready for export. If you make changes to this environment after creating this set, create a new migration set to export your changes.' Below this are sections for 'Sandbox Changes' and 'Outgoing Migration Sets'. The 'Outgoing Migration Sets' section lists a single item: 'CSM-Release1.0.1'. It shows a message: 'Your migration set is ready, and available for import in the target environment.' There are buttons for 'Move New Changes', 'Move All Changes', 'Delete', and 'Content Read Me'. At the bottom, it shows 'Created By alex.cloud', 'Release Version 11.1.13.0.0', 'Created On 2020-07-07 09:11:53 UTC', and 'Modules'.

Configure a Job to Import Configuration

Configure a job to import the configuration set to the target Oracle Cloud Applications instance.

When a build of this job runs, it requires the unique configuration set ID generated by the export configuration job. If you're not planning to run this job in a pipeline with the export configuration job, you'll need to provide the configuration set ID manually.

- In the left navigator, click **Builds**

2. In the **Jobs** tab, click the export configuration job.
3. Click **Configure**.
4. Click the **Steps** tab and note the **Configuration Set ID** field's parameter name.
5. Click **Cancel**.
6. Return to the **Jobs** tab and click **+ Create Job**.
7. In the New Job dialog box, in **Name**, enter a unique name.
8. In **Description**, enter the job's description.
9. In **Template**, select the **System Default OL7 for Visual Builder** template.
10. Click **Create**.
11. (Optional) Click **Configure** .
12. If you plan to run this job manually and not in a pipeline with the export configuration job, add a build parameter to accept the configuration set ID at runtime. Ignore this step if you plan to run the job in a pipeline with the export configuration job.
 - a. Click the **Parameters** tab.
 - b. From **Add Parameter**, select **String Parameter**.
 - c. In **Name**, enter the Configuration Set ID parameter name you noted in Step 4. If you didn't change the default parameter name, enter `CONFIGURATION_SET_ID`.
 - d. (Optional) In **Description**, enter the parameter's description. For example, you could add this: Enter the configuration ID copied from the CSM Export job's build log.
13. Click the **Steps** tab.
14. From **Add Step**, select **Configuration Set Migration**, and then select **Import Data**.
15. In **Target Instance**, select the environment that points to the Oracle Cloud Application's target instance.
16. In **Configuration Set ID**, enter the Configuration Set ID parameter name you copied in Step 4 in the `$<PARAMETER_NAME>` or `${<PARAMETER_NAME>}` format. By default, it's set to `${CONFIGURATION_SET_ID}`.

When a build of this job runs, the job will get the unique configuration set ID generated by the export configuration job. If no ID is found, the build will fail.
17. In **Username** and **Password**, enter the credentials of an IDCS user who is also an Oracle Cloud Applications user.
18. (Optional) To skip active sandboxes check, select the **Ignore unpublished sandboxes with Application Composer** check box.
19. Click **Save**.

When you import configurations, the target Oracle Cloud Applications instance creates an internal sandbox to test the migrated configurations. The build job gets the sandbox ID from the target Oracle Cloud Applications instance and displays it in the build's log.

Here's an example:

```
[2020-07-01 09:36:13] Build scheduled. Build started by pipeline CSM
[Upstream Job: CSM-Release1.0.1 Build: #1]
[2020-07-01 09:36:13] Build task id: d2c1b321-cba8-45e5-2233-2dd3a25678a1
.
```

```
[2020-07-01 09:36:32] Configuration Set ID = 1239873457659876
[2020-07-01 09:36:49] Status: Import started
.

.

.

[2020-07-01 09:42:32] IMPORT_IN_PROGRESS...
[2020-07-01 09:42:39] CSM Sandbox Id: abc01def2fe345fedcba12345ab01234
[2020-07-01 10:01:46] IMPORT_SUCCESS...

.

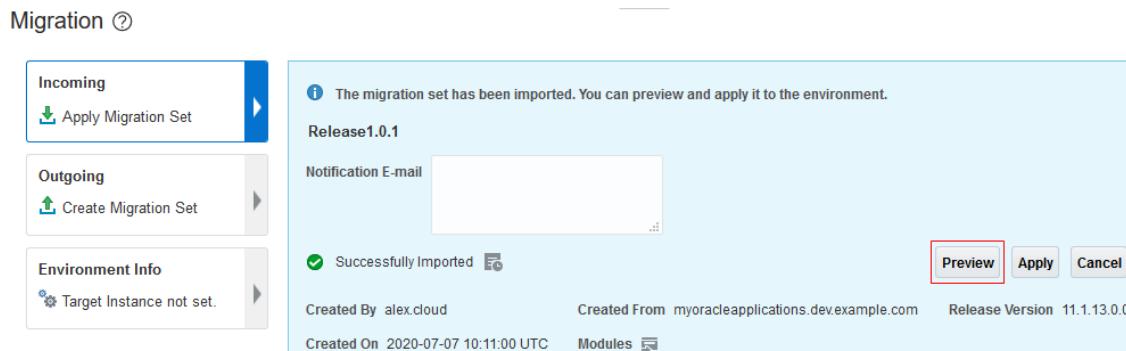
.

.

[2020-07-01 10:07:44] === End CSM Import Execution ===
Executor log size 7.6 KB (7,616)
[2020-07-01 10:07:46]
[2020-07-01 10:07:46] Build completed.
[2020-07-01 10:07:46] Status: DONE Result: SUCCESSFUL Duration: 31 min
```

After a successful build, to view details of the imported migration set, open the target Oracle Cloud Applications instance's Migration page and click the **Incoming** tab.

Here's an example:



To test the imported configurations on the newly created sandbox, click **Preview**.

Configure a Job to Cancel the Configuration Migration

If for some reason your import configuration job fails, configure a job to cancel the migration process.

This job requires the unique configuration set ID generated by the export configuration job. If you're not planning to run this job in a pipeline with the export configuration job, you'll need to provide the configuration set ID manually.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click the export configuration job.
3. Click **Configure**.
4. Click the **Steps** tab and note the **Configuration Set ID** field's parameter name.

5. Click **Cancel**.
6. Return to the **Jobs** tab and click **+ Create Job**.
7. In the New Job dialog box, in **Name**, enter a unique name.
8. In **Description**, enter the job's description.
9. In **Template**, select the **System Default OL7 for Visual Builder** template.
10. Click **Create**.
11. (Optional) Click **Configure** .
12. If you plan to run this job manually and not in a pipeline with the export configuration job, add a build parameter to accept the configuration set ID at runtime. Ignore this step if this job runs in a pipeline with the export configuration job.
 - a. Click the **Parameters** tab.
 - b. From **Add Parameter**, select **String Parameter**.
 - c. In **Name**, enter the Configuration Set ID parameter name you copied in Step 4. If you didn't change the default parameter name, enter `CONFIGURATION_SET_ID`.
 - d. (Optional) In **Description**, enter the parameter's description. For example: Enter the configuration ID copied from the CSM Export job's build log.
13. Click the **Steps** tab.
14. From **Add Step**, select **Configuration Set Migration**, and then select **Cancel Migration**.
15. In **Target Instance**, select the environment that points to the Oracle Cloud Application's target instance.
16. In **Configuration Set ID**, enter the Configuration Set ID parameter name you noted in Step 4 in the `$<PARAMETER_NAME>` or `${<PARAMETER_NAME>}` format. By default, it's set to `${CONFIGURATION_SET_ID}`.

When a build of this job runs, the job gets the unique configuration set ID generated by the export configuration job. If no ID is found, it fails.
17. In **Username** and **Password**, enter the credentials of an IDCS user's credentials who is also an Oracle Cloud Applications user.
18. Click **Save**.

Configure a Job to Apply and Save the Configuration Migration

After your import job is successful and you have manually tested the changes on the target Oracle Cloud Applications instance, configure a migration job to save the configuration set.

This job requires the unique configuration set ID generated by the export configuration job. If you don't know the ID while configuring this job, use a build parameter to accept the configuration set ID at runtime. Don't add this job in the pipeline with the import configuration job, but run this job manually after you've verified all imported configurations on the target Oracle Cloud Applications instance.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click the export configuration job.
3. Click **Configure**.
4. Click the **Steps** tab and note the **Configuration Set ID** field's parameter name.

5. Click **Cancel**.
6. Return to the **Jobs** tab and click **+ Create Job**.
7. In the New Job dialog box, in **Name**, enter a unique name.
8. In **Description**, enter the job's description.
9. In **Template**, select the **System Default OL7 for Visual Builder** template.
10. Click **Create**.
11. (Optional) Click **Configure** .
12. Click the **Parameters** tab.
13. From **Add Parameter**, select **String Parameter**.
14. In **Name**, enter the Configuration Set ID parameter name you copied in Step 4.
If you didn't change the default parameter name, enter `CONFIGURATION_SET_ID`.
15. (Optional) In **Description**, enter the parameter's description. For example: Enter the configuration ID copied from the CSM Export job's build log.
16. Click the **Steps** tab.
17. From **Add Step**, select **Configuration Set Migration**, and then select **Apply Migration**.
18. In **Target Instance**, select the environment that points to the Oracle Cloud Application's target instance.
19. In **Configuration Set ID**, enter the Configuration Set ID parameter name you copied in Step 4 in the `$<PARAMETER_NAME>` or `${<PARAMETER_NAME>}` format. By default, it's set to `$ {CONFIGURATION_SET_ID}`.
When a build of this job runs, the job gets the unique configuration set ID generated by the export configuration job. If no ID is found, it fails.
20. In **Username** and **Password**, enter the credentials of an IDCS user's credentials who is also an Oracle Cloud Applications user.
21. Click **Save**.

Configure a Job to Restore Configuration Migration

Configure a job to restore the target Oracle Cloud Applications instance to the state before the configuration set was applied.

This job requires the unique configuration set ID generated by the export configuration job. If you don't know the ID while configuring this job, use a build parameter to accept the configuration set ID at runtime. Don't add this job in the pipeline with the import configuration job, but run this job manually after you've verified all imported configurations on the target Oracle Cloud Applications instance.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click the export configuration job.
3. Click **Configure**.
4. Click the **Steps** tab and note the **Configuration Set ID** field's parameter name.
5. Click **Cancel**.
6. Return to the **Jobs** tab and click **+ Create Job**.

7. In the New Job dialog box, in **Name**, enter a unique name.
8. In **Description**, enter the job's description.
9. In **Template**, select the **System Default OL7 for Visual Builder** template.
10. Click **Create**.
11. (Optional) Click **Configure** .
12. Click the **Parameters** tab.
13. From **Add Parameter**, select **String Parameter**.
14. In **Name**, enter the Configuration Set ID parameter name you copied in Step 4. If you didn't change the default parameter name, enter `CONFIGURATION_SET_ID`.
15. (Optional) In **Description**, enter the parameter's description. For example: Enter the configuration ID copied from the CSM Export job's build log.
16. Click the **Steps** tab.
17. From **Add Step**, select **Configuration Set Migration**, and then select **Restore Migration**.
18. In **Target Instance**, select the environment that points to the Oracle Cloud Application's target instance.
19. In **Configuration Set ID**, enter the Configuration Set ID parameter name you copied in Step 4 in the `$<PARAMETER_NAME>` or `${<PARAMETER_NAME>}` format. By default, it's set to `${CONFIGURATION_SET_ID}`.
When a build of this job runs, the job gets the unique configuration set ID generated by the export configuration job. If no ID is found, it fails.
20. In **Username** and **Password**, enter the credentials of an IDCS user's credentials who is also an Oracle Cloud Applications user.
21. Click **Save**.

Create and Configure the CSM Pipeline

To ensure the configuration jobs runs automatically in the correct order, create a pipeline and set the dependency.

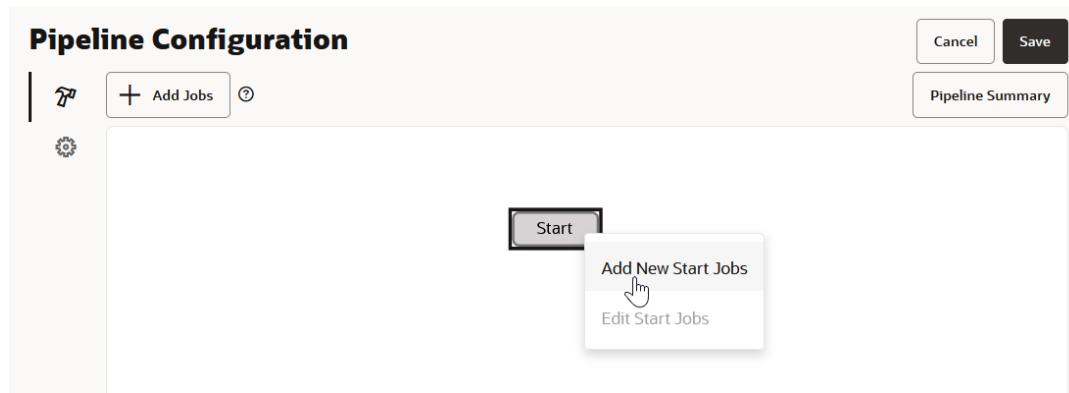
When you define the pipeline, make sure that the import configuration job triggers after the export configuration job is successful. If the import configuration job fails, run the cancel configuration migration job. Don't add the apply configuration job to the pipeline. You should run it manually after verifying your changes on the target Oracle Cloud Applications instance.

1. In the left navigator, click **Builds** .
2. Click the **Pipelines** tab.
3. Click **+ Create Pipeline**.
4. In the Create Pipeline dialog box, in **Name** and **Description**, enter a unique name and description.
5. Click **Create**.

The Pipeline Configuration window is displayed with the Start node in the center.



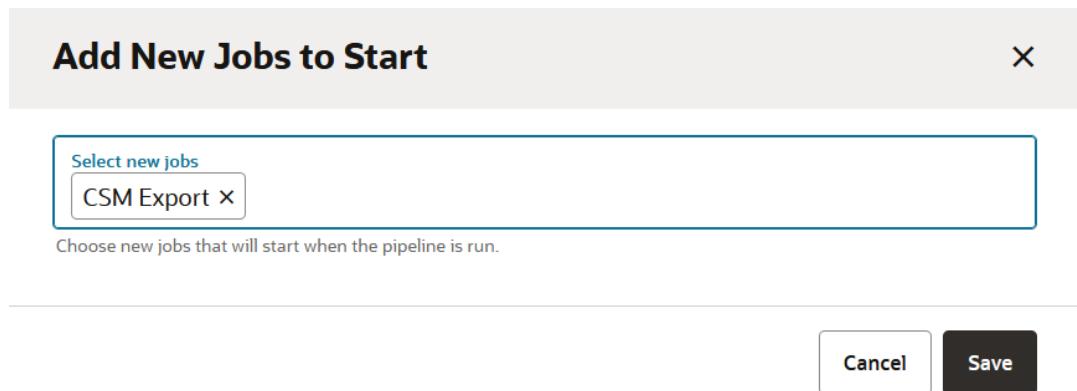
- Right-click the Start node and select **Add New Start Jobs**.



The **Add New Jobs to Start** dialog box is displayed.

- Click in the **Select new jobs** area to choose new jobs that will start when the pipeline is run.

Select one or more jobs from the list displayed.

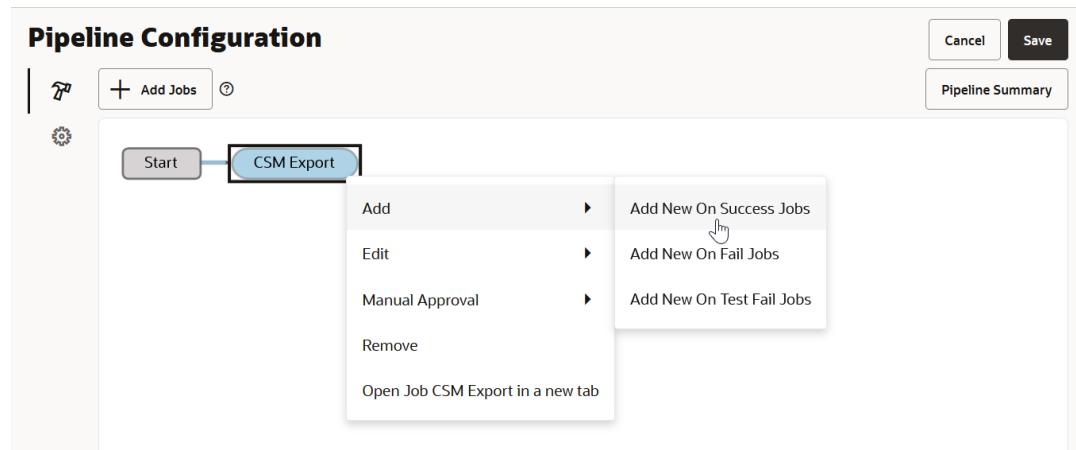


- Click **Save**.

The Pipeline Configuration window displays the Start node with a blue line that ends in an arrow connecting to the node that represents the job you just added.



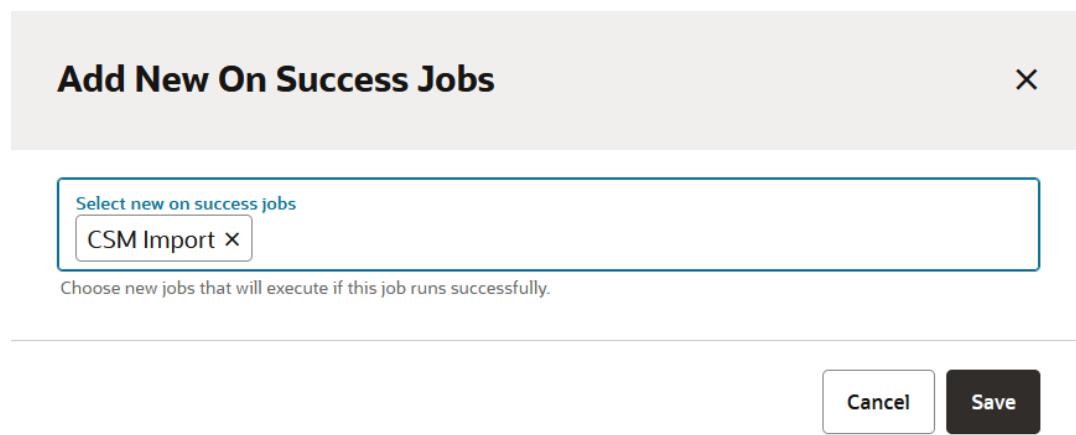
9. Right click the node you just added and select **Add** then select **Add New On Success Jobs**.



The **Add New On Success Jobs** dialog box is displayed.

10. Click in the **Select new on success jobs** area to choose new jobs that will execute if this job runs successfully.

Select one or more jobs from the list displayed.

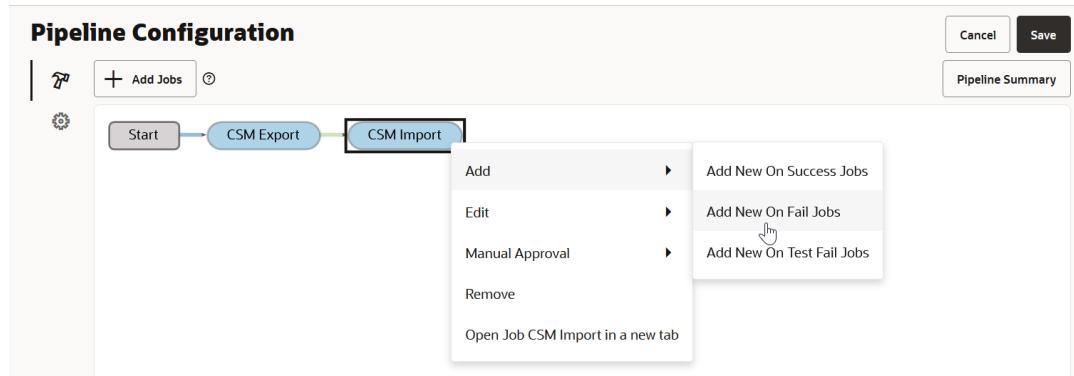


11. Click **Save**.

The Pipeline Configuration windows displays the Start node, the node with the build job you previously selected in step 7, and the node with the build job you just selected, all connected by lines that end in an arrow.



12. Right click the node you just added and select **Add** then select **Add New On Fail Jobs**.



The **Add New On Failure Jobs** dialog box is displayed.

13. Click in the **Select new on failure jobs** area to choose new jobs that will execute if this job's run fails.

Select one or more jobs from the list displayed.

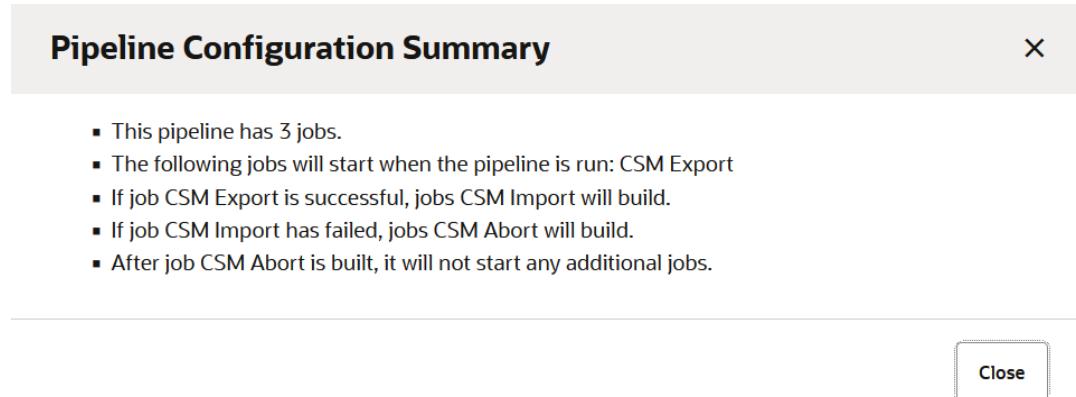


14. Click **Save**.

The Pipeline Configuration windows displays the Start node, the node with the build job you previously selected in step 7, the node with the build job you previously selected in step 10, and the node with the build job you just selected, all connected by lines that end in an arrow.



15. Click the Pipeline Summary button to display a summary screen that provides many details about the pipeline configuration, similar to the example shown here.



Create and Configure Build Jobs to Deploy the App Extension to the Sandbox

You need to set up packaging and deployment jobs to deploy app extensions to your Oracle Cloud Application's production instance's sandbox. This topic explains how to do that.

To do this:	See this:
Package extension for the Oracle Cloud Application's production sandbox	Create the Production Sandbox Packaging Job
Deploy the packaged extension artifact to the Oracle Cloud Application's production instance's sandbox.	Create the Production Deployment Build Job
Configure the production sandbox pipeline	Create and Configure the Production Sandbox Pipeline

Create the Production Sandbox Packaging Job

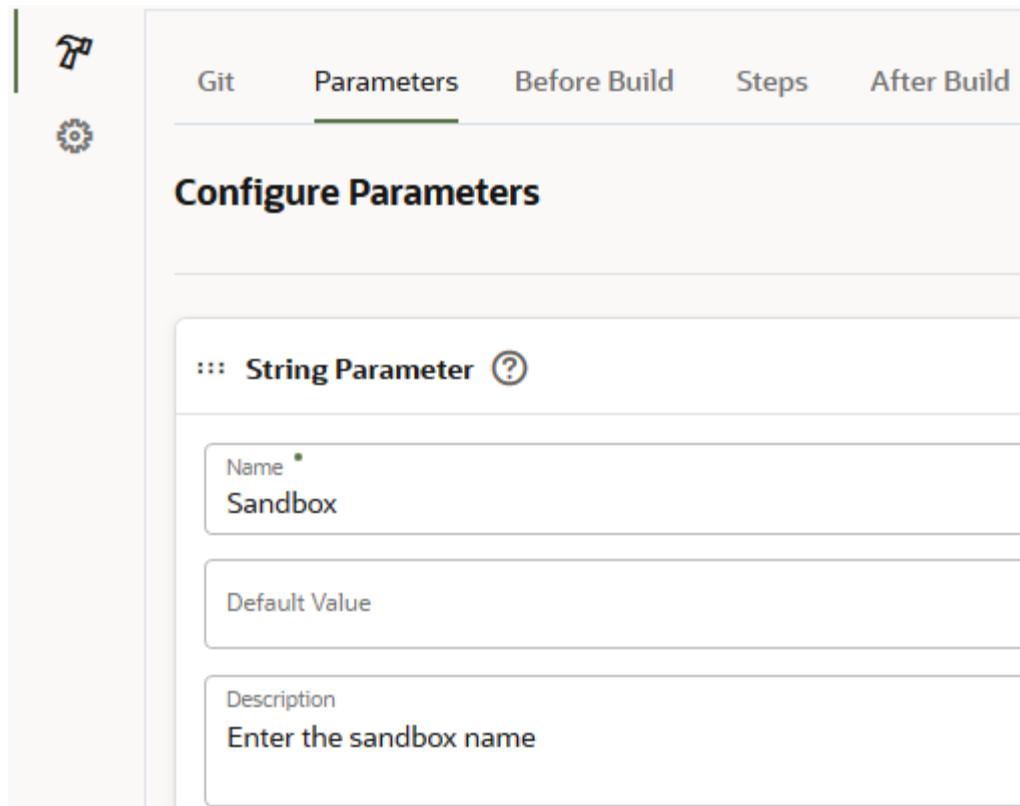
The sandbox packaging job generates an extension artifact that's ready to deploy to the Oracle Cloud Applications production instance's sandbox. Configure this job only if you want to deploy the extension to the production instance's sandbox.

To package extension for the sandbox, you'll create a build parameter that accepts the sandbox name when the build runs.

- In the left navigator, click **Builds** .
- Click **+ Create Job**.
- In the New Job dialog box, in **Name**, enter a unique name.
- In **Description**, enter the job's description.
- In **Template**, select the **System Default OL7 for Visual Builder** template.
- Click **Create**.
- Click **Configure** .

8. Click the **Git** tab.
9. From the **Add Git** list, select **Git**.
10. In **Repository**, select the Git repository. In **Branch or Tag**, select the production branch.
11. Click the **Parameters** tab.
12. From **Add Parameter**, select **String Parameter**.
13. In **Name**, enter a parameter name. You'll use this name to accept the sandbox's name when this job's build runs.

Here's an example:



14. (Optional) In **Description**, enter the parameter's description. For example: Enter the sandbox name.
15. Click the **Steps** tab.
16. From **Add Step**, select **Application Extension**, and then select **Package**.
17. By default, the packaging step minifies the application's source code before running the build. If you don't want to minify the source files, deselect the **Optimize extension** check box.

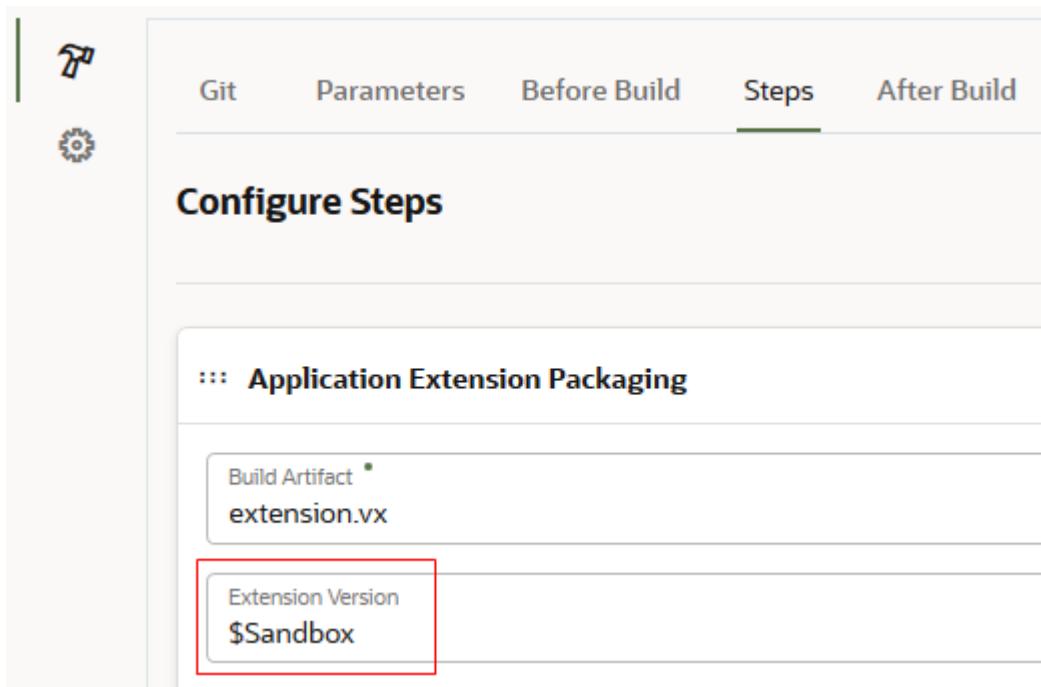
Minification is a process to remove the unnecessary characters (such as blank spaces, new lines, and comments) from the source code and reduce the size of the files, making the transfer of files consume less bandwidth and storage.

 **Note:**

When you deselect the **Optimize extension** check box, this warning is displayed: **Optimization not selected. Packaging without optimization can cause performance problems, so avoid doing so unless you're debugging or troubleshooting.**

18. (Optional) If you want to change the artifact file's name, in **Artifact**, enter the new name. By default, it is `extension.vx`.
19. In **Extension Version**, enter the sandbox parameter name you added in step 13. The name must be in the `$<PARAMETER_NAME>` or the `${<PARAMETER_NAME>}` format.

Example:



20. Click the **After Build** tab.
21. From **Add After Build Action**, select **Artifact Archiver**.
22. In **Files to archive**, enter `extension.vx`, the build artifact's name. You can also use wild characters. For example, `*.vx`.
23. If you want to discard the build's old artifacts, click **Settings** . In the **General** tab, select the **Discard Old Builds** check box and specify the discard options.
24. Click **Save**.

Create the Production Sandbox Deployment Build Job

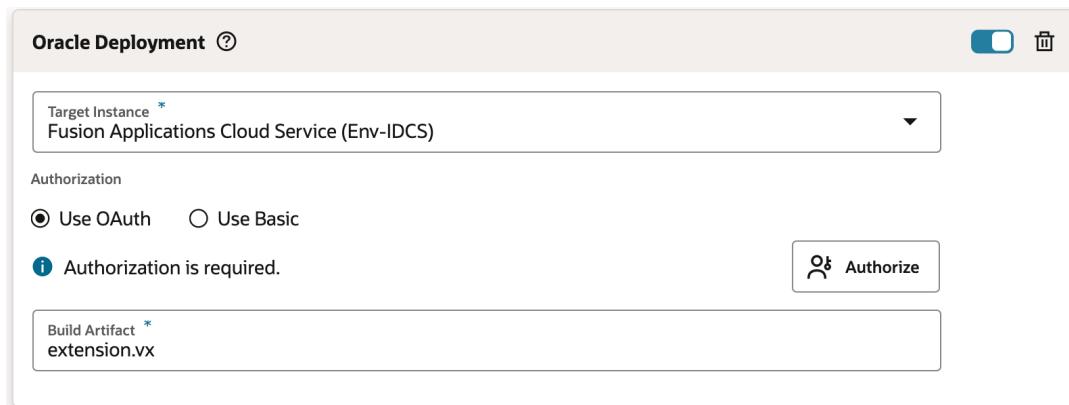
The sandbox deployment job deploys the extension's artifact that was generated in the sandbox packaging job to the Oracle Cloud Applications production instance's sandbox. Before you create the job, make sure you have credentials that VB Studio can use to access the Oracle Cloud Applications production instance's sandbox.

1. In the left navigator, click **Builds** .
2. In the **Jobs** tab, click **+ Create Job**.
3. In the New Job dialog box, in **Name**, enter a unique name.
4. In **Description**, enter the job's description.
5. In **Template**, select the **System Default OL7 for Visual Builder** template.
6. Click **Create**.
7. Click **Configure** .
8. Click the **Before Build** tab.
9. From **Add Before Build Action**, select **Copy Artifacts**.
10. In **From Job**, select the packaging job that generated the extension's artifact.
11. In **Which Build**, select one of the following:
 - Last successful build (default)
 - Last keep forever build
 - Upstream build in this pipeline instance
 - Specified by permalink
 - Specific build
 - Specified by build parameter

Depending on what you select, you may be prompted to select which permalink, build, or build parameter you want to be used.

12. Leave other fields with their default or empty values.
13. Click the **Steps** tab.
14. From **Add Step**, select **Oracle Deployment**.

This image shows the Oracle Deployment build job page that's been partially filled in.



The screenshot shows the Oracle Deployment configuration page. At the top, there is a header with the title "Oracle Deployment" and a help icon. To the right of the title are three icons: a switch, a trash can, and a refresh symbol. Below the header, there is a section for "Target Instance" with a dropdown menu containing "Fusion Applications Cloud Service (Env-IDCS)". Underneath this, there is an "Authorization" section with two radio buttons: "Use OAuth" (which is selected) and "Use Basic". A note below the radio buttons states "Authorization is required." To the right of this note is a button labeled "Authorize" with a person icon. At the bottom of the form, there is a field for "Build Artifact" with the value "extension.vx".

15. In **Target Instance**, select the environment with the Oracle Cloud Application's production instance.
16. In the Authorization section, select **Use OAuth** or **Use Basic** to specify the type of authorization used to run this build step.

If you selected **Use OAuth**, before authorization, the Authorization is required message indicates that this build step needs to be authorized, either by clicking the **Authorize** button or by running the build manually and entering the credentials when you are prompted for them at that point. After authorization, the Authorization has been provided message and the **Renew Authorization** button are displayed.

If you selected **Use Basic**, in **Username** and **Password**, enter the credentials of an IDCS user who is not only an Oracle Cloud Applications user, but one who can connect and undeploy from the Oracle Cloud Applications production instance's sandbox. These credentials must be those of a local user, not a federated identity, and must not require multi-factor authentication.

17. In **Build Artifact**, enter the same artifact name that you used in the sandbox packaging build step.
18. Click **Save**.

Create and Configure the Production Sandbox Pipeline

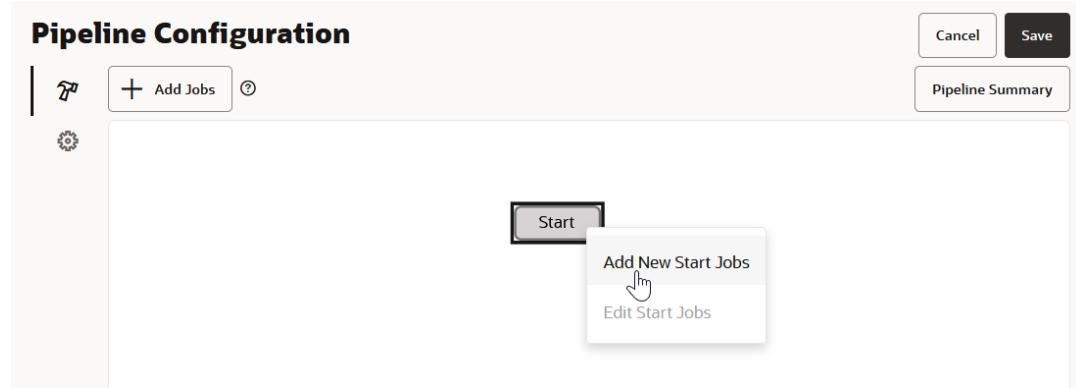
To ensure the package and deploy jobs runs in the correct order, create a pipeline and set the dependency.

1. In the left navigator, click **Builds**.
2. Click the **Pipelines** tab.
3. Click **+ Create Pipeline**.
4. In the **Create Pipeline** dialog box, in **Name** and **Description**, enter a unique name and description.
5. Click **Create**.

The Pipeline Configuration window is displayed with the Start node in the center.



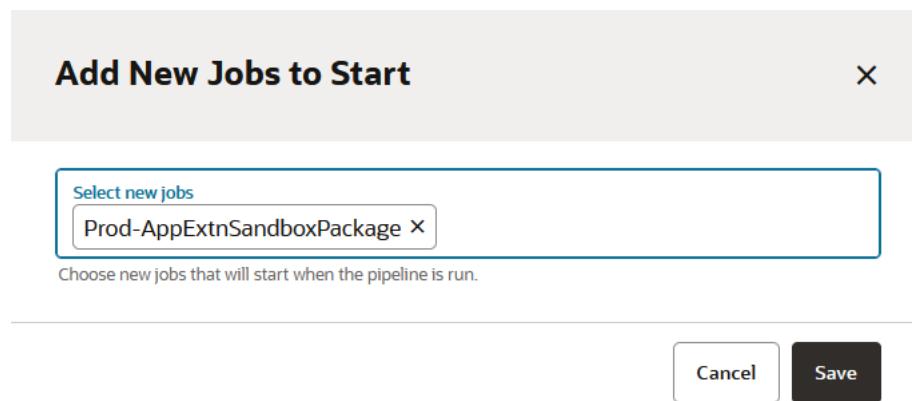
6. Right-click the Start node and select **Add New Start Jobs**.



The **Add New Jobs to Start** dialog box is displayed.

- Click in the **Select new jobs** area to choose new jobs that will start when the pipeline is run.

Select one or more jobs from the list displayed.

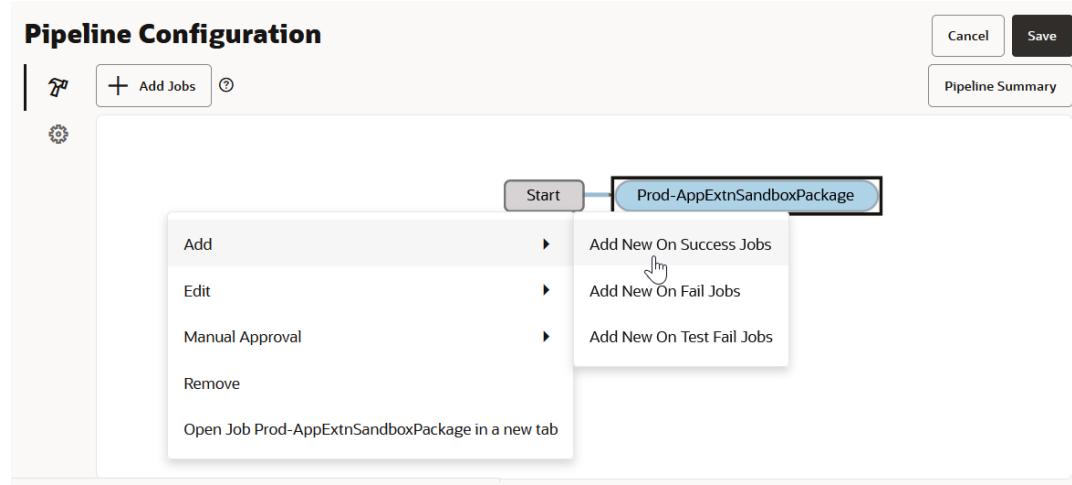


- Click **Save**.

The Pipeline Configuration window displays the Start node with a blue line that ends in an arrow connecting to the node that represents the job you just added.



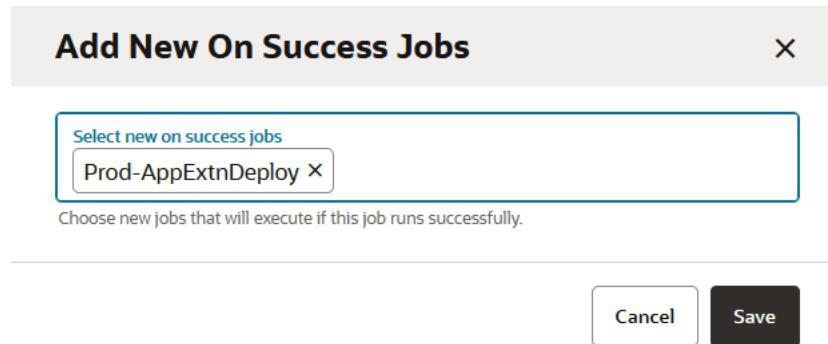
- Right click the node you just added and select **Add** then select **Add New On Success Jobs**.



The **Add New On Success Jobs** dialog box is displayed.

- Click in the **Select new on success jobs** area to choose new jobs that will execute if this job runs successfully.

Select one or more jobs from the list displayed.



- Click **Save**.

The Pipeline Configuration windows displays the Start node, the node with the build job you previously selected in step 7, and the node with the build job you just selected, all connected by lines that end in an arrow.



- Click the Pipeline Summary button to display a summary screen that provides many details about the pipeline configuration, similar to the example shown here.

Pipeline Configuration Summary

This pipeline has 2 jobs.
The following jobs will start when the pipeline is run: Prod-AppExtnSandboxPackage
If job Prod-AppExtnSandboxPackage is successful, jobs Prod-AppExtnDeploy will build.
After job Prod-AppExtnDeploy is built, it will not start any additional jobs.

Close

To see even more information about each item in the pipeline's configuration, move the mouse over the node or connection you want to learn more about. For example, in this pipeline configuration, if you hover over the the connection between the Prod-AppExtnSandboxPackage and Prod-AppExtnDeploy nodes, you'll see this information displayed.



Run the Pipelines

Before you deploy your extension to the production instance, run the CSM pipeline to migrate configurations and the production sandbox pipeline to deploy extension to the sandbox where you migrated configurations.

You should run pipelines and jobs in this order:

1. Run the CSM pipeline to migrate configurations.
Before you run the pipeline or the import configurations job, make sure that the source and the target instances match and the target instance does not have any configurations rekeyed or from another source.
2. Run the production sandbox pipeline to deploy the extension to the sandbox where you've migrated configurations.
3. Preview the sandbox and test the extension.
4. If the testing is successful, run the apply configuration job.

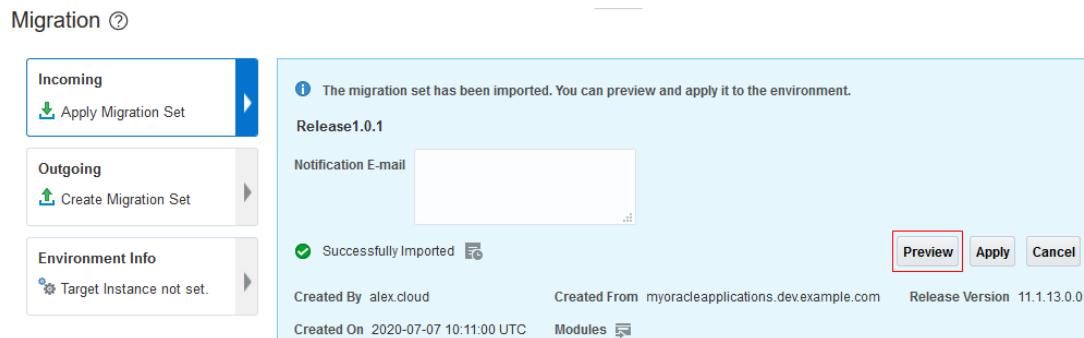
Follow these steps to run pipelines in the above order:

1. In the left navigator, click **Builds**
2. Click the **Pipelines** tab.
3. On the right side of the CSM pipeline, click **Build**

Wait for configurations to migrate to the production Oracle Cloud Applications instance's sandbox. Click the pipeline's name to monitor the progress of its build jobs.

4. In the pipeline, if the import configuration job is successful, verify the migrated configurations in the sandbox's preview mode. Open the Oracle Cloud Applications production instance's migration page, click **Incoming** and click **Preview**.

Here's an example:



To find out on what you can see in the sandbox preview mode, see [Preview Mode in Unified Migration](#) in *Configuring and Extending Applications*.

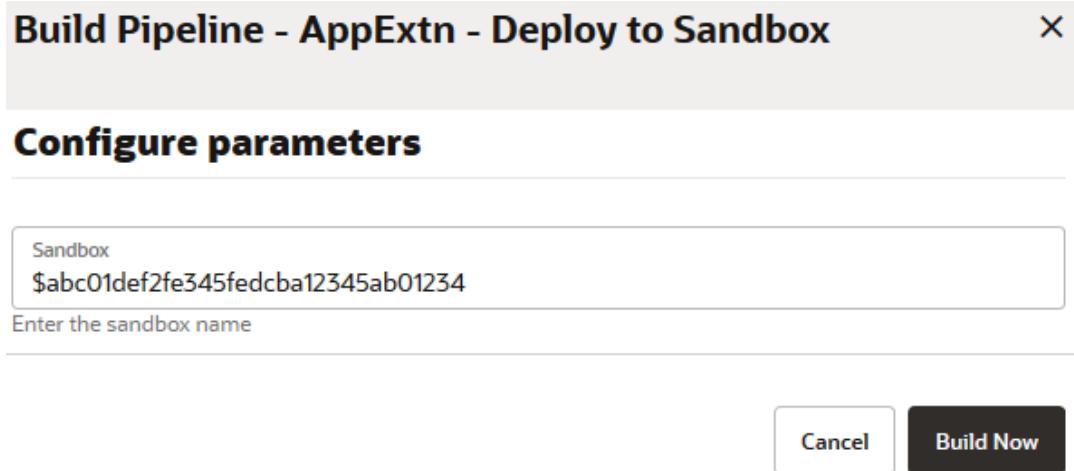
5. Return to VB Studio's **Build** page.
6. Open the import configuration job's build log and note the CSM Sandbox ID. You'll need the sandbox ID when you deploy the extension to the sandbox.

Here's an example:

```
[2020-07-09 09:31:20] BACKUP_IN_PROGRESS...
[2020-07-09 09:41:31] IMPORT_IN_PROGRESS...
[2020-07-09 09:42:01] BACKUP_DONE...
[2020-07-09 09:42:32] IMPORT_IN_PROGRESS...
[2020-07-09 09:43:02] CSM Sandbox Id: abc01def2fe345fedcba12345ab01234
[2020-07-09 10:26:46] IMPORT_SUCCESS...
[2020-07-09 10:26:46]
```

7. Click the **Pipelines** tab. On the right side of the production sandbox pipeline, click **Build** .
8. In the Configure Parameters dialog, enter the sandbox's name in the \$<Sandbox_Name> format and click **Build Now**.

Here's an example:



9. After the pipeline is successful, verify the extension along with the migrated configurations in the sandbox's preview mode. Open the Oracle Cloud Applications production instance's migration page, click **Incoming** and click **Preview**.
10. Return to VB Studio's **Build** page.
11. If the imported configurations meet your expectations, save the migration set.
 - a. In the **Jobs** tab, open the export configuration job's build log and note the Configuration Migration Set ID.

Here's an example:

```
[2021-01-14 06:18:16] Status: Export Started
[2021-01-14 06:18:16] Configuration Migration Set ID is 1234432100056789
[2021-01-14 06:19:47] IN_PROGRESS...
[2021-01-14 06:42:43] SUCCESS...
[2021-01-14 06:46:19] CREATED...
```
 - b. Open the apply configuration job and click **Build Now**.
 - c. In the Configure Parameters dialog box, enter the configuration set ID and click **Build Now**.
12. Deploy the extension to the production mainline. See [Set Up the Project to Deploy to Production](#) in *Administering Visual Builder Studio*.

After you apply the configurations, Oracle Cloud Application end users won't see any changes to the user interface until they log out and log back in to the App.

C

Public IP Addresses for VB Studio

VB Studio's list of public IP addresses has been removed from this guide and is now available as a separate document. Please update your bookmarks to point to the new URL: Public IP Addresses for Oracle Visual Builder Studio.