The world of software development is undergoing rapid changes. Long drawn-out software development processes are giving way to a “build fast, fail fast” approach in which organizations deliver products and prototypes within weeks, if not days. With Agile becoming the go-to development methodology, new tools and techniques like wiki, Bamboo, Bitbucket, and GitHub are now industry standard in the product development lifecycle.

As a developer, your decisions must align with these expectations. Imagine you’re part of a startup that has built a working prototype and has secured funding. You plan to set up a full-scale development environment to take your business to the next growth phase. Your options: You could purchase the core stack components, including the operating systems, integrated development environments (IDEs), servers, and databases, to set up the development environment. Or you could purchase a pay-as-you-go subscription to a fully-integrated environment hosted on the cloud.

Setting up your own development environment from scratch can take weeks, if not months. A subscription, on the other hand, would let you provision the development environment within minutes, with minimal cost of ownership.

This scenario highlights why so many organizations are moving development activities to hosted environments. The benefits of quick provisioning combined with minimal up-front payment and maintenance costs are game changers for developers who want to get going, and fast.

Automate Your CI/CD Workflows with Oracle Developer Cloud Service
A Developer’s Guide

Contents

Why Oracle Developer Cloud Service? 2
Why Oracle Java Cloud Service? 2
What is in the Toolbox? 2
What Is the Cost Advantage? 3
Where Do You Want to Focus Your Resources? 3
How Do I Get Started? 4
Launching Your Oracle Developer Cloud Service 4
Setting Up Preferences 4
Developing Applications 4
Building and Deploying Applications 5
Using Agile and Sprint Tools for Project Management 6
Scenario: How One Customer Uses Oracle Developer Cloud Service 7
Why Oracle Cloud? 7
Getting Started 7
Running Multiple Environments on Oracle Java Cloud Service 7
Developing the Application Using Oracle DevCS 8
The Solution: Two Web Applications and Two Deployments 8
Creating the Application Structure 8
Creating the User Interface 9
Checking Performance 10
Summary 11
Additional Resources 11
Why Oracle Developer Cloud Service?

Oracle Developer Cloud Service (DevCS) provides developers with a fully provisioned environment, hosted on Oracle Cloud. It comes loaded with tools for building and deploying applications, source control management, code review, issue tracking, agile project management, continuous integration, and team collaboration.

Using Oracle Developer Cloud Service, you can develop new apps, deploy to other Oracle Cloud services such as Oracle Java Cloud Service, or deploy in your on-premises infrastructure. You can collaborate, track issues, manage your code, and deploy continuously in the cloud.

Why Oracle Java Cloud Service?

Oracle Java Cloud Service provides a runtime platform for Java and Java EE based applications. Oracle Java Cloud Service comes with a complimentary instance of Oracle Developer Cloud Service. You can use Oracle Developer Cloud Service to automate the deployment of applications to Oracle Java Cloud Service.

You can deploy your Java application to Oracle Java Cloud Service – SaaS Extension, Oracle Java Cloud Service, or Oracle Application Container Cloud Service.

Learn how to set up and create a deployment target in Oracle Developer Cloud Service, in order to deploy an application artifact to an Oracle Java Cloud Service instance.

What Is in the Toolbox?

Oracle Developer Cloud Service provides a rich set of tools that includes:

- Code Version Management – Git
- Build Automation
  - Ant, Maven, Gradle, npm, Grunt, Bower, Gulp, SQLcl, PSM, Shell
- Test Automation
  - Selenium, JUnit, Findbugs
- Continuous Integration Engine
- Deployment and Provisioning Automation
- Issues Tracking
- Agile Process Management
- Code Review
- Wiki
- Activity Stream
Where Do You Want to Focus Your Resources?

Oracle Developer Cloud Service (DevCS) provides a complete set of developer services that eliminates the overhead of deploying and managing applications. Focus instead on doing the things that differentiate your business, such as building innovative features into your applications and creating exceptional user experiences.

Let’s explore how Oracle Developer Cloud Service leads to a productivity boost that can translate to more features and a faster time to market.

All components are available in a project, which is a collection of features and services provided by Oracle Developer Cloud Service.

DevCS also provides seamless integration with Eclipse, JDeveloper and NetBeans. It also lets you rapidly build and deploy your applications to Oracle Java Cloud Service and your on-premises infrastructure.

Oracle Developer Cloud Service provides easy ways to set up projects, configure repositories and Webhooks, manage ATOM/RSS feeds, and import external CI jobs.

What Is the Cost Advantage?

Focusing your development resources on application development while minimizing headcount for nonessential expertise naturally translates into reduced costs. But the advantages extend beyond the obvious. Choosing Oracle Developer Cloud Service over a development environment of your own leads to reduced costs.

Oracle Developer Cloud Service provides easy ways to set up projects, configure repositories and Webhooks, manage ATOM/RSS feeds, and import external CI jobs.

With Oracle Developer Cloud Service, you are tapping into an economy of scale. Consider the number of hours it would take to set up the core stack—the platform-level components—for your applications. Add to that the hours expended on an ongoing basis to maintain the stack. Now consider those same costs multiplied across hundreds of applications, and the cost-saving proposition of a cloud-based development environment becomes clear.

Agility and Time to Market: Possibly the most obvious reason for using DevCS to deploy and run your applications is that it enhances your agility. DevCS eliminates the time and resources spent on setting up, maintaining, and adapting your infrastructure.

Operational Simplification: DevCS removes the need for development teams to perform mundane tasks, such as installation and upgrades. You can replace scripted operations with automated developer services. DevCS takes on the operational and security burden and makes sure the development software and processes that you rely on are up to date and optimized for best performance.

Expertise: Platform as a service (PaaS) is an operationally complex infrastructure with a myriad of processes and dependencies. Finding technically skilled experts to install, configure, and manage such an environment can be expensive and resource intensive. DevCS removes this layer from development teams and allows them to focus on application development—a clear win for your business.
How Do I Get Started?

Use your Oracle Cloud account to request a trial subscription for one of the following services. (If you don’t have an Oracle Cloud account, you can create one as part of your request.) The only way to receive Oracle Developer Cloud Service is to order a Cloud Service that includes it as part of a subscription.

- Oracle Java Cloud Service
- Oracle Java Cloud Service - SaaS Extension
- Oracle Messaging Cloud Service
- Oracle Mobile Cloud Service
- Oracle Application Container Cloud Service
- Oracle SOA Cloud Service
- Oracle Database Cloud Service
- Oracle Containers Cloud Service

When approved, you’ll get an email asking you to activate your trial. After you get your Oracle Cloud account, refer to the workflow for using Oracle Developer Cloud Service.

Launching Your Oracle Developer Cloud Service

You can access Oracle Developer Cloud Service from the Oracle Cloud web interface and Eclipse, NetBeans, and JDeveloper. In addition, you’ll be able to check code into the repository from any environment that supports the Git command line.

Learn how to start using an Oracle Developer Cloud Service instance.

Setting Up Preferences

In the DevCS environment you can set up your preference including your user profile, add the SSH key, and configure your email notifications and language preferences.

Developing Applications

To begin developing applications, first review the workflow for Oracle Developer Cloud Service. The descriptions in the workflow can help you start setting up projects, creating Git repositories, using snippets, managing and reviewing code, and building and deploying project artifacts.

Creating Projects: A project is a collection of features and services provided by DevCS. You can create a project to 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.

Using Git: A Git repository provides a control system for your source code that easily integrates with existing repositories. Oracle Developer Cloud Service uses Git source control management to save and manage your application source code files.

Learn about using Git in Oracle Developer Cloud Service.
Using Maven and Gradle: Manage your builds by using Maven and Gradle to achieve sustained and critical build-process conformity and stability, regardless of the size or complexity of your teams and projects.

Managing Releases: Create or edit a release in Oracle Developer Cloud Service. A release is a collection of specific tags or branches of Git repositories, specific binary artifacts of the project Maven repository, specific archived build artifacts, and pointers to your documentation wiki pages.

Using Snippets: Create and share a snippet with your project members. Snippet files contain small pieces of useful and reusable content, such as code and commands.

Reviewing Source Code: Review and merge your source code to avoid bugs, identify design issues, and catch design and implementation problems that could affect the performance of the application.

Learn more about code reviews.

Tracking Issues: Track issues to manage tasks, defects, and features. You can create an issue to track a new feature request or to file a bug against a product. You can create a task to be performed and then assign it to yourself or another project member.

Using Agile Methodology: Create a board to manage Agile sprints and issues in DevCS. The Agile page displays the boards you own, your favorite boards, and all boards associated with the project.

Learn more about implementing Agile methodology.

Building and Deploying Applications

Oracle Developer Cloud Service includes continuous integration services to build project source files.

Creating Jobs: Create and configure a job to run builds and generate artifacts that you want to deploy. You can view information about all build jobs. You can also provide links to configure and manage them on the Jobs Overview page.

Managing Builds: Customize your automatic builds by determining their frequency and their response to test and build failures.

Deploying Applications: Configure your project to deploy successful builds automatically to an Oracle Java Cloud Service or an on-premises environment for testing or production.
Configuring Webhooks: A Webhook is a user-defined HTTP callback that sends notifications when events occur. The information that a generic Webhook sends is delivered using an HTTP POST request with the application/json content-type, with the UTF-8 character set, in a Message object. Webhooks in Oracle Developer Cloud Service support only HTTP/HTTPS communication.

Oracle Developer Cloud Service supports the following types of Webhooks:

- Generic Webhook
- GitHub-compatible Webhook
- HipChat Webhook
- Hudson/Jenkins Webhook
- Hudson/Jenkins Build Trigger Webhook
- Jenkins Build Notification Webhook
- Oracle Social Network Webhook
- Slack Webhook

Learn how to configure Webhooks in Oracle Developer Cloud Service to integrate with Slack for messaging.

Using Wiki Pages: Oracle Developer Cloud Service provides a Wiki instance for document authoring and collaboration. Using the Wiki page, you can create wikis and share information with your team.

Using IDEs: You can access Oracle Developer Cloud Service projects from IDEs such as Oracle Enterprise Pack for Eclipse, Oracle JDeveloper, and Netbeans IDE.

Learn how to access Oracle Developer Cloud Service from IDEs.

Using Agile and Sprint Tools for Project Management

Oracle Developer Cloud Service provides a powerful way to foster communication between teams by supporting the Agile development methodology. The rich set of Agile and Sprint tools that come with DevCS simplify task management and help you coordinate project delivery from start to finish by closely monitoring progress in the sprints.

DevCS provides an issue tracking system that helps you document and track your to-do items, whether they are features, tasks or bugs. You can also use stories and epics if you follow agile development methodology. Issues can be assigned to specific team members, targeted for specific release, assigned a priority and a host of other attributes that you can customize. In addition, you can estimate the time or complexity of a task to help you plan your project activity and time line.

To track the progress of development activities in DevCS, you start by creating a board from an Agile page. This will let you view a list of backlog issues that are waiting your actions. Next, you create a sprint and add issues that you target for resolution in this sprint.
Getting Started

Red Samurai used the following Oracle Cloud and on-premises technologies for building the invoicing system:

→ Oracle Java Cloud Service to build the application
→ Oracle Developer Cloud Service to perform DevOps
→ Oracle JavaScript Extension Toolkit (Oracle JET) to create the user interface
→ Oracle ADF Business Components to create REST services

Running Multiple Environments on Oracle Java Cloud Service

Red Samurai used Oracle Java Cloud Service to deploy the invoicing application and decided to run Oracle ADF Business Components and Oracle JET on Oracle Java Cloud Service. ADF can’t be deployed to an Oracle Application Container Cloud Service, so Red Samurai hosted Oracle JET on Oracle Java Cloud Service. This decision made it possible to host the server-side code and the client-side code on the same Oracle Java Cloud Service instance. Developers could run the demonstration environment and the production environment on the same Oracle Java Cloud Service instance, targeted to different managed servers. This also made maintenance easier.

Learn how to build Oracle ADF Applications with Maven using Oracle Developer Cloud Service.
Creating the Application Structure

The invoicing application was divided into two parts—Oracle ADF Business Components REST and the Oracle JET user interface—and deployed separately. Oracle ADF Business Components REST implements back-end REST services and provides authentication and authorization controls. Red Samurai used Oracle ADF Business Components development together with the REST interface provided in Oracle ADF Business Components 12.2.1.

The following image shows the structure of the invoicing application in Oracle Java Cloud Service:

The Solution:
Two Web Applications and Two Deployments

Red Samurai adopted a two-way approach to build the invoicing solution.

Developing the Application Using Oracle DevCS

Red Samurai developed the invoicing application by using Oracle Developer Cloud Service (DevCS). The developers used the Agile and Sprint management dashboard extensively to monitor sprint progress. They used Eclipse for managing Oracle JET code, and they used Git to manage source code files. They used the continuous build automation feature to build and save changes to the Oracle ADF Business Components REST repository.

After each build process was completed, the developers used the DevCS deploy feature to deploy the latest Enterprise Application archive (EAR) package directly to Oracle Java Cloud Service. This feature helped save time by eliminating the need to repeat redeployment steps after each build.
Creating the User Interface

Red Samurai used Oracle JET to develop the user interface for the application. The menu structure was designed using the Oracle JET router. The application’s home page has a dashboard that displays financial data for orders, customers, and invoice items. In addition, the application includes a new-invoice page, a template-setup page, an invoice-search page, a customer-setup page, and a supplier-setup page.

The user interface for the invoicing application is based on the Oracle JET user interface, which uses a modular architecture approach, implementing each use case in a separate module. The common logic, such as the REST service definition, was moved out into controller modules. The client-side business logic was implemented in the Oracle JET module by using JavaScript functions and was rendered with the Oracle JET user interface components.

The following image shows the Model-View-Controller (MVC) structure of the invoicing application:

The application runtime data shows that the Oracle ADF Business Components REST application executes the Oracle ADF REST servlet to produce the REST request and response. The custom PDF servlet and ADF authentication servlet execute the logout session.

The Oracle JET wrapper application doesn’t run any server-side logic. It simply returns the Oracle JET application content to the client. Only one file servlet is invoked in the operation.

The following image shows that only one file servlet is invoked:
Checking Performance

Oracle JET, which runs on the client side, results in fewer server-side calls compared to applications built on Oracle ADF Faces. In this invoicing system scenario, when an invoice is submitted, several REST calls are executed, and they complete running within 100 microseconds. This speed indicates that Oracle JET REST calls don’t block client functionality. Calls execute asynchronously, allowing users to work in the application.

When users sign out of the invoicing application, a request is executed for the ADF authentication servlet with the logout parameter set to true, allowing the Oracle ADF Business Components REST session to close on the server side.

The following image shows the REST call execution time:
Summary

DevCS is a complete PaaS development environment for the enterprise that offers simplification via an automatically provisioned development platform supporting the full development life cycle. This turnkey development platform, which is securely hosted in the cloud, lets developers collaborate through integrated tools to manage and track tasks, builds, and documentation. With Oracle Developer Cloud Service, it’s not only easy to integrate with other services running in the cloud, but it also provides for automatic deployment to Oracle Java Cloud Service or your local on-premises infrastructure.

Additional Resources

→ Building Oracle ADF Applications with Ant Using Oracle Developer Cloud Service
→ Deploying an Application from Oracle Developer Cloud Service to Oracle Java Cloud Service
→ Developing Applications with Eclipse and Oracle Developer Cloud Service
→ Performing Code Reviews in Oracle Developer Cloud Service

Try It Yourself

Free Cloud Trial ➔

Scenario: How One Customer Uses Oracle Developer Cloud Service

Automate Your CI/CD Workflows with Oracle Developer Cloud Service