



# Salesforce

DevOps for Salesforce

# Table of Contents

## **Part 1** | An Introduction to Salesforce and DevOps

- 4** What is Salesforce?
- 4** What do organizations want?
- 5** Where does Salesforce fall short?
- 6** Salesforce before & after DevOps
- 7** How do I resolve these challenges?
- 9** What is FlexDeploy?
- 11** What does FlexDeploy have to do with Salesforce?
- 13** Exploring FlexDeploy's Support for Salesforce

## **Part 2** | A Step-by-Step Guide to Support for Salesforce

- 18** Source Control Your Salesforce Changes
- 22** Deploy Salesforce Apps and Metadata Objects
- 25** Build and Deploy Salesforce DX Apps
- 29** Unit Test and Test Automation for Salesforce

# Part 1

## An Introduction to Salesforce and DevOps

### What is Salesforce?

Salesforce is an extremely popular cloud computing software as a service (SaaS) based customer relationship management (CRM) solution used by companies around the world. The Customer 360 platform integrates an organization's departments (including marketing, sales, commerce, service, and IT) to power all aspects of the sales lifecycle and "bring companies and customers together".

Customer 360 hosts all the Salesforce apps in one platform, which allows everyone to access all apps from anywhere. With all information in a central location, teams can access the same information to streamline communication and nurture leads through their customers' journey.

**Salesforce is the world's #1 CRM  
with over 150,000 customers.**

The Customer 360 CRM is just one technology owned by Salesforce. The Salesforce ecosystem includes MuleSoft, Einstein AI, Tableau, Slack, and others.

Over 150,000 customers embrace and enjoy Salesforce because it merges critical data, streamlines internal communication, personalizes customer experiences, and enables smarter and faster decision-making for business success.

### What do organizations want?

Running a successful enterprise business that delivers products and/or services is impossible without technology. Most enterprises have a wide and varied technology landscape to deliver quality products/services and respond to customers' needs.

The pace of today's business demands IT departments increase innovation and deliver high-quality outputs while maintaining strict budgets. Every organization is striving to develop and deliver software in a quick, high-quality, and cost-effective manner.

The goal is to wow customers. This means teams must develop high-quality software faster, more repeatably, and with less cost and risk.

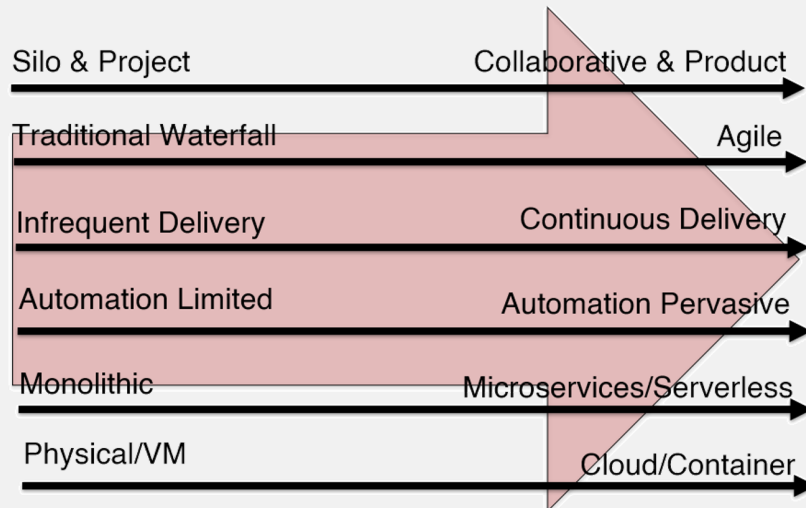
*Automation and agility allow organizations to achieve these goals.*

Each organization is somewhere along their agile and automation journey, which has traditional waterfall methodologies and limited automation on one end of the spectrum and agile practices and extensive automation on the other.

It is essential for the success of a business to continue evolving their processes and enabling technology. The next step may be an evolution toward agile and automated processes. Maybe the next step is to become more continuous in how frequently capabilities are delivered. Perhaps it is more general, like adopting new technologies, such as cloud, microservices, serverless, or containers.



With companies focused on delivering high quality software faster, many are looking to DevOps and automation solutions which help them achieve speed at scale, while improving quality and more effectively managing the cost and risk associated with software delivery. Generally, the shift is from left to right on the diagram below, with a desire to leverage existing investment as well.



## Where does Salesforce fall short?

Salesforce is a powerful tool that is used and loved by over 150,000 companies around the world. So why would you need to incorporate any other tools to make it better?

*Migrating Salesforce changes across environments is a manual, tedious, and error-prone process.*

It is difficult to tell what needs to be migrated across sandbox environments, and there isn't clear visibility into what has been migrated to each environment, when the changes were made, or by whom.

Companies have tried various solutions, but keeping metadata in sync between sandboxes and organizations is a time consuming and error-prone challenge to say the least.

**This is where DevOps comes in!**

# Salesforce before & after DevOps

## Before DevOps

- ❌ Slow and error-prone changes across all environments, leading to unhappy IT leaders, Business, and Customers.
- ❌ Manual activities and script development/maintenance is costly, which reduces the time spent on activities that directly contribute to business value.
- ❌ Deployments into test and production environments are error-prone, negatively impacting business results and increasing the time spent troubleshooting and rectifying deployment-related outages.
- ❌ Employee satisfaction is negatively impacted because developers and other highly skilled professionals spend time on lower-level activities which do not utilize their skills.
- ❌ Lack of automation results in poor visibility to changes and too much time spent figuring out what is deployed where, when changes were made, and who made the changes.

## After DevOps

- ✅ Increased speed and agility of software delivery results in the ability to deliver consistently, more often, and reduce the feedback cycle.
- ✅ Development, Operations, and supporting teams can focus their talents and energy on applications and services versus lower-level tooling and manual activities.
- ✅ Automated deployments reduce errors, time spent troubleshooting, and the negative impact of outages across test and production environments.
- ✅ Employee satisfaction and retention improves significantly. Development and IT professionals do what they do best, and avoid stressful nights and weekends spent dealing with production releases and issues.
- ✅ Visible comparison of changes from sandbox or scratch org environments makes identification of changes easy. Users can manage packages with ease and track where they got deployed.
- ✅ Automated testing enables consistent execution across environments.
- ✅ Sandbox Orgs make adopting Salesforce DX and DevHub based development easy.
- ✅ Control your Salesforce environments with notifications, and quality gates such as approvals, schedules, and automated test validation.

## How do I resolve these challenges?

### What solutions are out there?

As you can see, there are many potential challenges with Salesforce development and delivery. Fortunately, there are resources and solutions available, including:

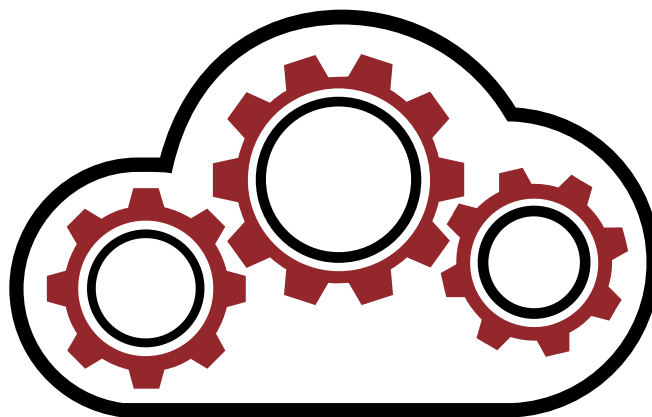
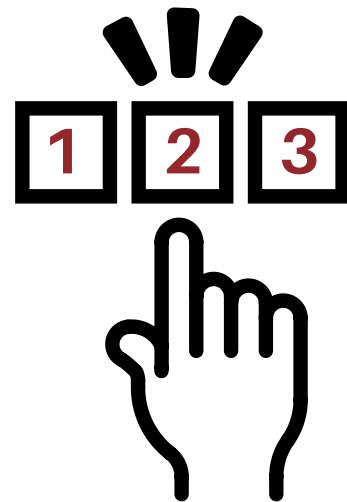
1. Salesforce's own DevOps Center
2. Niche solutions focused on Salesforce
3. Enterprise DevOps Platforms

### What is the right solution for me?

Each solution solves the problems and challenges inherent to Salesforce development and delivery. However, there are limitations and tradeoffs you need to consider when choosing an option.

Teams must determine which solution is best for them based on their larger technology landscape and their organization's goals. Evaluate both where you are now and where you want to go. With these two situations in mind, you can determine which solution will help you in this progression.

Let's explore each.



**“The most powerful tool we have as developers is automation.”**

**Scott Hanselman**

### 1) Salesforce's DevOps Center

The DevOps Center is a new addition for Salesforce, providing change and release management directly on their platform. This streamlines processes for developers and admins.

However, there are limitations. It isn't an enterprise play, meaning you need other DevOps tools for the remainder of your technology landscape. It doesn't work cross technology, cross vendor, and/or cross cloud like other solutions.



### 2) Salesforce-Specific Solutions

Another set of DevOps solutions are focused on Salesforce and Salesforce alone. With this niche focus, you encounter the benefits of a best-of-breed tool because it is built for Salesforce teams.

However, like Salesforce's DevOps Center, these solutions are limited to Salesforce. You do not receive the benefits of DevOps for your other technologies. If you want to extend DevOps across your technology landscape, you will need multiple solutions. This can quickly become a burden on your time and budget.



### 3) Enterprise DevOps Platforms

Finally, there are enterprise DevOps, or Continuous Delivery and Release Automation (CDRA), platforms.

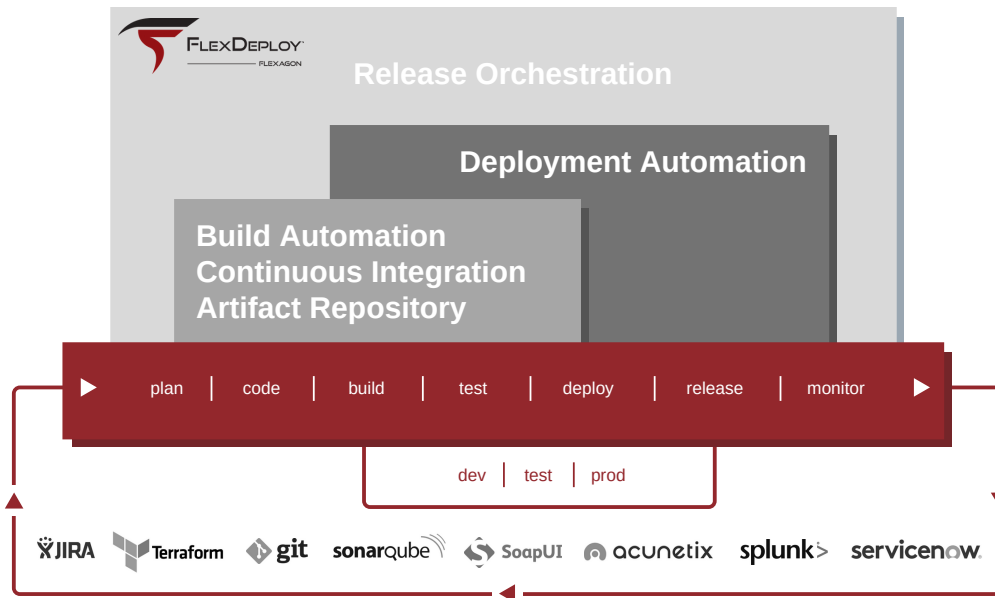
Unlike the previous two solutions that focus only on Salesforce, a good enterprise DevOps platform integrates with other tools in your toolchain, whether on-premises or in the Cloud. This simplifies the development process, allowing for an efficient, centralized, and consistent developer experience.

As Salesforce brings an organization's departments together (including marketing, sales, commerce, service, and IT) into one centralized platform, an enterprise DevOps platform integrates with an IT team's technologies. Through integration, you and your team can build, test, deploy, release, and monitor your software delivery life cycle (SDLC) in one platform.

## What is FlexDeploy?

FlexDeploy is an enterprise DevOps Platform for Continuous Delivery and Release Automation. FlexDeploy enables enterprises to achieve better software delivery by providing automation, controls, and visibility to implement higher quality software faster, with lower cost and risk.

Since its inception, the focus has been the developer experience. This has resulted in first-class support for Build, Deploy, Release, and Orchestration processes.



With native support for over 100 open-source and commercial technologies, FlexDeploy integrates with tools and technologies across your delivery toolchain.

FlexDeploy supports a variety of source control tools, unlike some other Salesforce DevOps tools that only integrate with Git. This allows your team to stay in their developer tools, further simplifying their experience.

Additionally, there is integration with other essential developer tools like agile project management and containers so you can streamline your Agile planning.

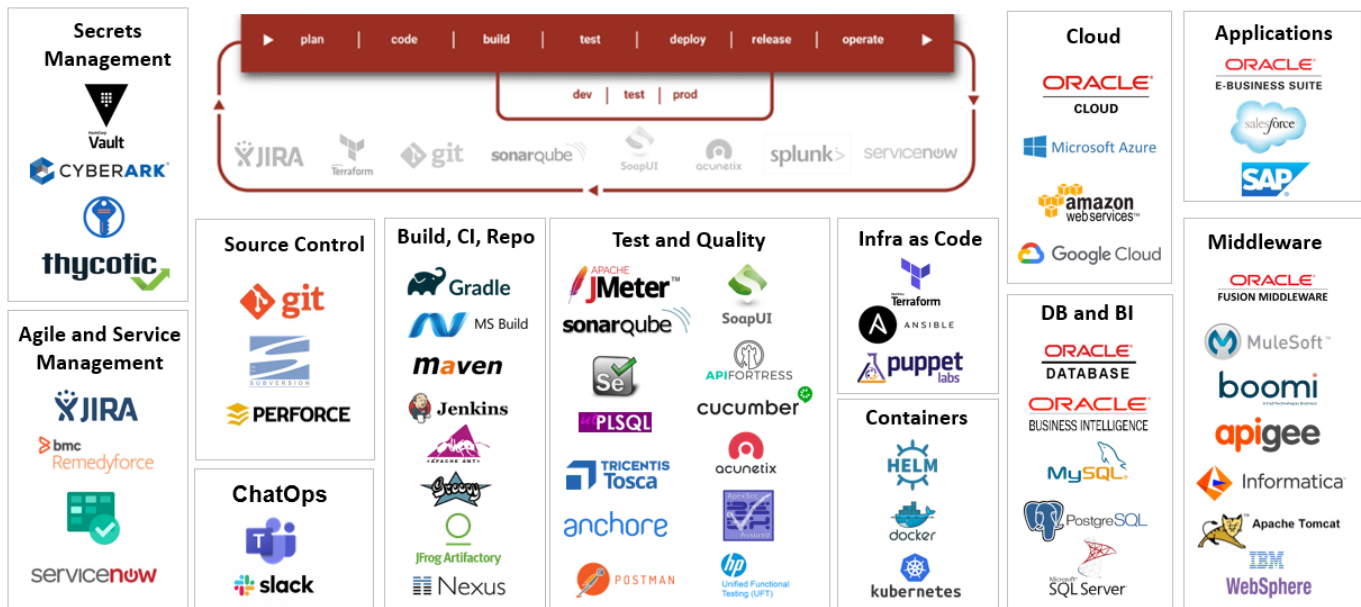
**“FlexDeploy has dramatically improved our ability to deliver software fast and with high quality, positioning us to meet our aggressive business growth strategy.”**

**CIO, Global Food Manufacturer**

This solution doesn't stop at Salesforce or the build environment. It extends to delivery and deployment for the entire Salesforce Ecosystem, including MuleSoft, and the rest of your technology landscape. FlexDeploy supports you wherever you are in your automation journey, from traditional waterfall to Agile software methodologies.

FlexDeploy streamlines deployment and release pipelines across on-premises and cloud environments and provides value to organizations of all sizes so they can scale with confidence.

### FlexDeploy's Pre-Built Integrations and Plugins



### With FlexDeploy, you can...

- **Automate and standardize** the provisioning, build, deploy, and release lifecycle across environments.
- **Speed up** the delivery of software, better meeting internal and external customer needs.
- **Decrease costs** associated with building and maintaining scripts, performing manual steps, and troubleshooting problems.
- **Reduce errors and lower the risk** of deployment failures and environment outages.
- **Gain visibility** into the who, what, when, and where across all environments and applications.
- **Improve team satisfaction** by reducing stress and allowing staff to perform more valuable and enjoyable work.

## What does FlexDeploy have to do with Salesforce?

FlexDeploy has out-of-the-box support for Salesforce proper, and the larger Salesforce Ecosystem (MuleSoft APIs, Database deployments, Java Applications, Node.js, Kubernetes, and many commercial and open source technologies running in the cloud and on premises).

The fully integrated solution for Salesforce deployment and release management has rich features for continuous integration and continuous delivery, enterprise release management, and handling of technologies.

FlexDeploy makes it easy to establish repeatable and efficient processes for Salesforce implementations, including extensive automation, improved controls and repeatability, and visibility like you've never had before. It provides a comprehensive and integrated solution across Build, Deploy, and Release Automation, helping across the software delivery lifecycle.

FlexDeploy's rich DevOps features, flexibility in controlling release automation, and integrated Salesforce support provides a highly efficient solution for your enterprise.

You can establish an automated and repeatable process for making changes across your sandbox, test, and production environments. FlexDeploy helps automate many Salesforce development methodologies – Org Development, Source-Driven Development, No-Code Development, or Low-Code Development, and supports Advanced Developers.

FlexDeploy supports Salesforce DX-Based Development and it helps when users migrate from Sandbox-Based Development to DevHub-Based Development.

FlexDeploy increases the speed of making changes by automating tasks which would have otherwise been performed manually or with customized scripts. Because changes are automated, quality increases significantly, driving down the time and cost of delivery. Visibility to what, when, where, and who improves, and users can use dashboards and reports to gain insight to real-time and historical data associated with a Salesforce implementation. No more wondering what caused the issues, or having to scramble to execute audits successfully. All the information you need, at your fingertips.



FlexDeploy offers robust Salesforce org management and integration with source control and automated release pipelines.

## Integration Features

**FlexDeploy integration capabilities with your Salesforce Org:**

- Identify and retrieve changes from Salesforce development orgs (sandboxes or scratch orgs). Compare the changes with the files in your source control tool (e.g. Git).
- Commit changes selectively from Salesforce into source control.
- Create branches for easy and controlled deployment.
- Manage Packages with Unit test execution settings.
- Continuous Integration of changes through a gated release pipeline.

## Plugin Features

**The FlexDeploy Salesforce plugin provides the following capabilities:**

- Deploy changes to Classic, Lightning, DevHub (Salesforce DX) environments.
- Deploy metadata, settings, Apex classes, and profiles to Salesforce Orgs and sandboxes.
- Continuous Integration from source control to a sandbox.
- Continuous Delivery of source control changes through a gated release pipeline.
- Rollback change when needed.
- Execute Apex test cases and view the results in FlexDeploy.
- Automate promotions based on test results.

## Who uses FlexDeploy?

Companies around the globe rely on FlexDeploy to help them deliver better quality software and maintain a competitive advantage, including:



Flexagon was named a Leader in the Forrester Wave: Continuous Delivery and Release Automation, Q2 2020





## Exploring FlexDeploy's Support for Salesforce

### Supported Metadata Types

You can make your Salesforce org even more useful and responsive to your users by developing custom applications and integrating your external applications. You can deploy all your code changes with FlexDeploy as it supports all commonly used metadata types.

### Identify and View Changes

FlexDeploy helps you to identify and view changes made in a sandbox or development org or scratch org. Once identified, FlexDeploy helps pick what can go into your Source Control Management (SCM) tool, such as Git, and can commit changes directly.

Select	State	Type	Full Name	Last Modified By
<input type="checkbox"/>	+	ApexPage	/pages/HeatMapMock.page	Karl Henselin
<input type="checkbox"/>	+	ApexPage	/pages/Opportunity_PDF.page	Karl Henselin
<input type="checkbox"/>	+	PermissionSet	/permissionsets/FlexDeploy_Developer.permissionset	Karl Henselin
<input type="checkbox"/>	+	PermissionSet	/permissionsets/Flex_Developer.permissionset	Karl Henselin
<input type="checkbox"/>	!	PermissionSet	/permissionsets/Trailblazer_Developer.permissionset	Karl Henselin
<input type="checkbox"/>	+	Profile	/profiles/Analytics_Cloud_Integration_User_profilePasswordPolicy15910708646...	Karl Henselin
<input type="checkbox"/>	+	Profile	/profiles/Custom_Sales_Profile_profilePasswordPolicy15910708646...	Karl Henselin
<input type="checkbox"/>	+	ProfileSessionSetting	/profileSessionSettings/Analytics_Cloud_Integration_User_profileSessionSetting15910708646...	Karl Henselin
<input type="checkbox"/>	+	ProfileSessionSetting	/profileSessionSettings/Custom_Sales_Profile_profileSessionSetting15910708646...	Karl Henselin
<input type="checkbox"/>	+	Profile	/profiles/Admin.profile	Karl Henselin

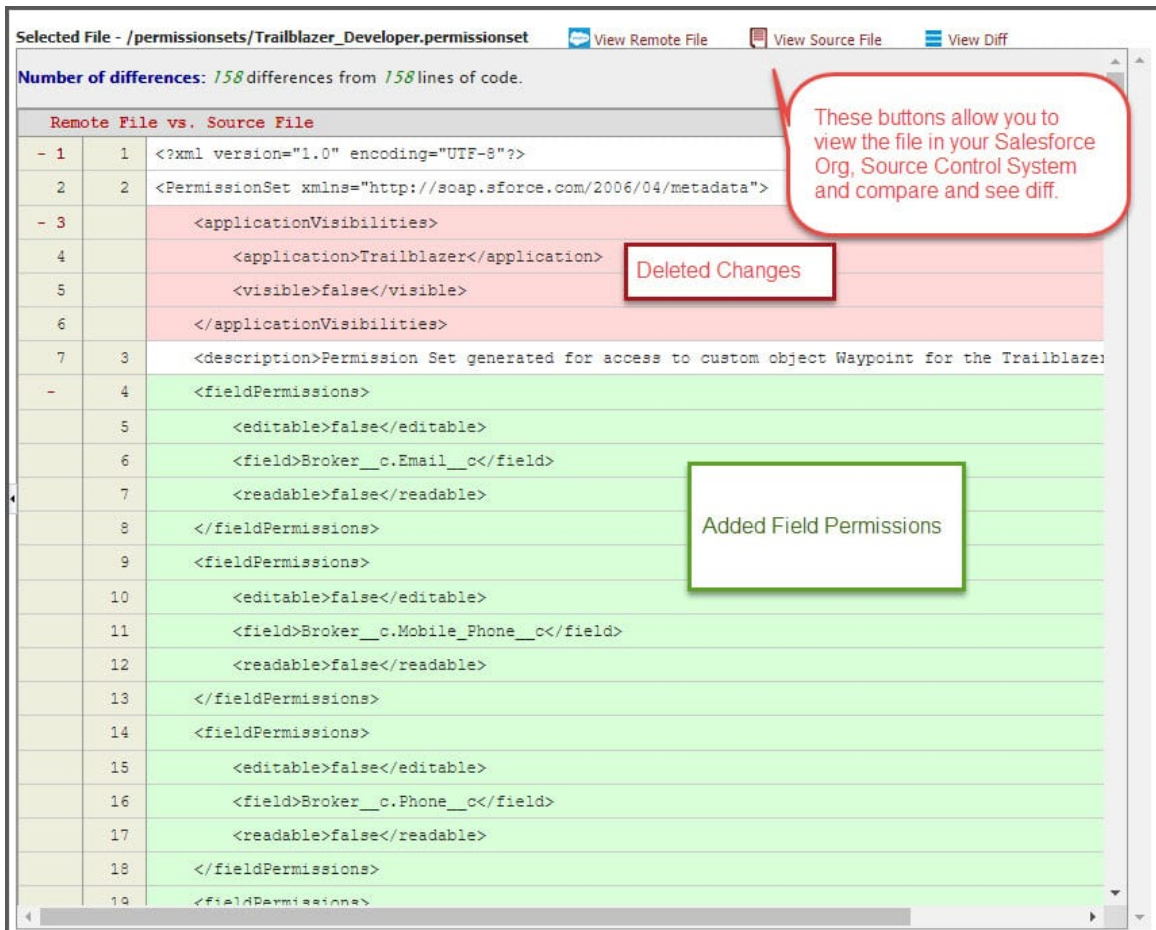
Indicates Page was added.

Indicates Profile Set was modified

User Name who worked on the file.

### Diff the Modified Files

You can view the file contents in your org or SCM and view the line-by-line difference to identify what sections were added or removed.



## Auto Generation of Package XML

With so many types of customizations, it is difficult to hand-code your package.xml to pull the changes from Salesforce and deploy to other environments. FlexDeploy automatically generates the package.xml and eliminates developers' manual work.

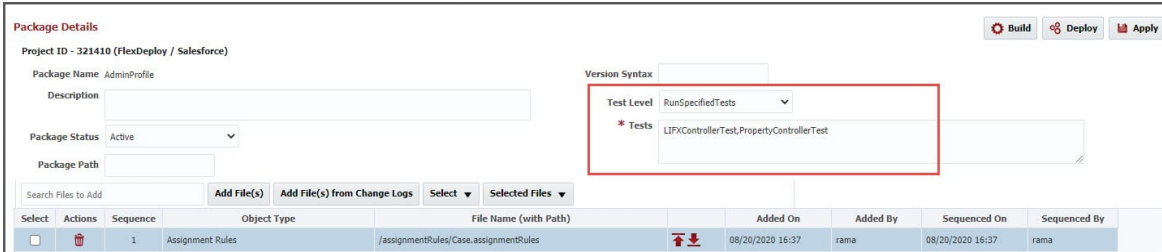
## Create Packages and Deploy

FlexDeploy allows you to select multiple files and create a package, which is especially handy when you have a lot of changes. You can track the packages with the latest version and the version that has been deployed to each environment org.

Packages										
Project ID - 321410 (FlexDeploy / Salesforce)										
Name or Description		Updated After		05/28/2020 18:18						
Package Status		Updated By		rana						
Actions	Package Name	Description	Package Status	Latest Version	Development	Test	QA	Production	Package Path	Test Level
	AdminProfile 		Active	1.0.295						RunSpecifiedTests
	Homepage 		Active	1.0.289	1.0.289					NoTestRun

## Control Unit Test Executions

You have full control to define the test level for each package. This ensures any needed tests execute after deploying and avoids missing test executions.



Package Details

Project ID - 321410 (FlexDeploy / Salesforce)

Package Name: AdminProfile

Description:

Package Status: Active

Package Path:

Version Syntax:

Test Level: RunSpecifiedTests

\* Tests: LJFXControllerTest,PropertyControllerTest

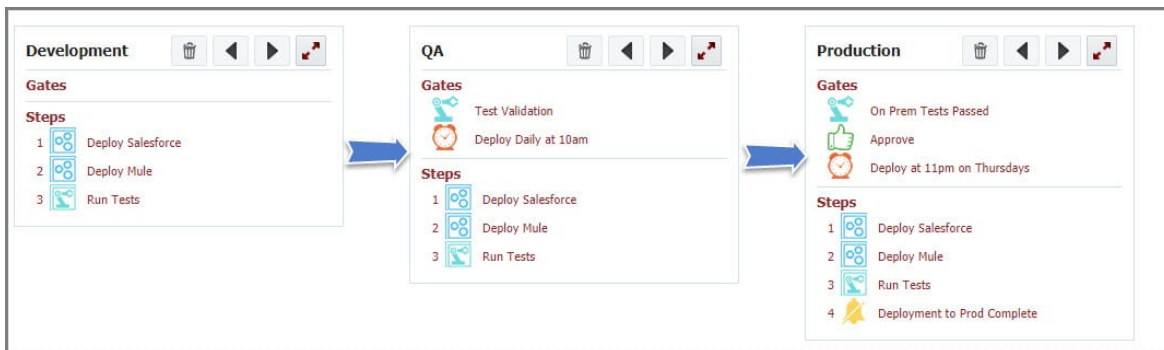
Select	Actions	Sequence	Object Type	File Name (with Path)	Added On	Added By	Sequenced On	Sequenced By
<input type="checkbox"/>		1	Assignment Rules	/assignmentRules/Case.assignmentRules	08/20/2020 16:37	rama	08/20/2020 16:37	rama

## Notifications and Approvals

With FlexDeploy you can enable notifications to keep respective teams informed of any continuous deployments. Notifications include the details of changes being promoted. You can also set up approvals, including external approvals that delegate the approval to tools like ServiceNow.

## Fully Automated Your Release with Dependencies

FlexDeploy's in-depth CI/CD and release features will help you automate the complete Salesforce release process. With a simple drag-and-drop editor for creating reusable release pipelines, users can automate and control release processes. In addition to Salesforce, you can deploy changes to any technology like Mule Services, Web Applications, etc. The release pipelines can include automated and manual steps.



## Test Automation

You can automate Salesforce Apex Tests or UI automation tests using test tools like Selenium, Tosca, HP UFT, Automation Anywhere, and more. Simply configure FlexDeploy to execute the tests and automate the next steps (e.g. code promotions, failures, notifications, etc.) based on the test results.



Strategy	Test Sets	Test Definitions	Statistics
* Test Type	Unit Tests	* Environment	Development
* Test Type	Functional Tests	* Environment	QA
* Test Type	Smoke Tests	* Environment	Production

## Easy Rollbacks

Most of the time we push a release forward and don't look back. But sometimes a release fails or causes issues, which are only compounded when that release is to a production environment. FlexDeploy has plugin operations to rollback any Salesforce packages, if needed. You can ensure the quality of an environment by automating the release to rollback the package based on test results.

## Salesforce DX Ready

All the FlexDeploy features work with Salesforce DX, per Salesforce's recommendations and best practices. If you already have DX in your organization or you are planning to move to DX, it is easy and seamless with FlexDeploy! FlexDeploy integrates with your DevHub orgs and scratch orgs.

## Integration with Other Tools

FlexDeploy's native integration with other tools like Git, Jira, and ServiceNow will help achieve process-oriented development within your organization.

## Keep Reading

In Part 2, we'll walk through how you can utilize FlexDeploy with Salesforce, and start getting the benefits of a DevOps platform for Continuous Delivery and Release Automation.

# Part 2

## A Step-by-Step Guide to FlexDeploy's Support for Salesforce

## Source Control Your Salesforce Changes

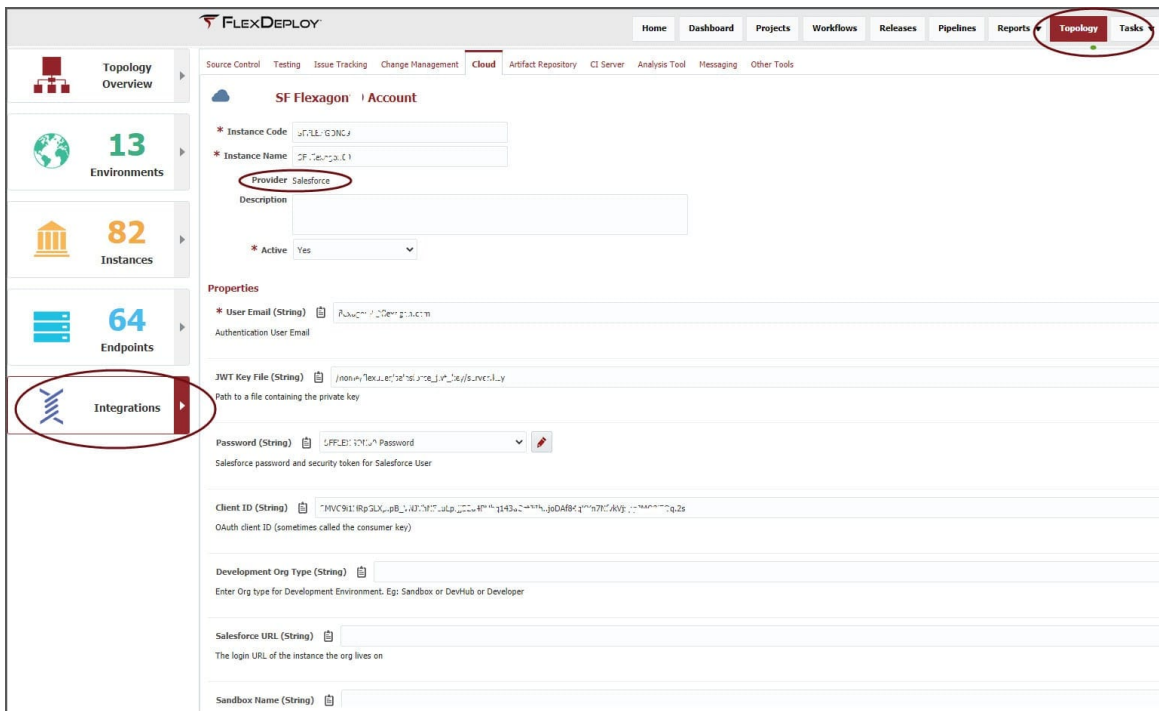
In this chapter, discover how easy it is to find the changes from your development Org(s) and commit to your source control repository. We recommend using a source control management (SCM) tool such as Git, SVN, or TFS for storing Salesforce changes.

### Configuring FlexDeploy to Use Salesforce

Follow these steps to complete a one-time configuration to use FlexDeploy with Salesforce in your enterprise. You can use Blueprints to complete this configuration using a quick and easy wizard.

#### Configuring Salesforce Org Accounts in FlexDeploy

Salesforce accounts need to be configured to connect to your Orgs. This can be done under the Cloud Integration in FlexDeploy's Topology screen. You can connect using User Email and Password or User Email, OAuth Key, and Client Id combination.

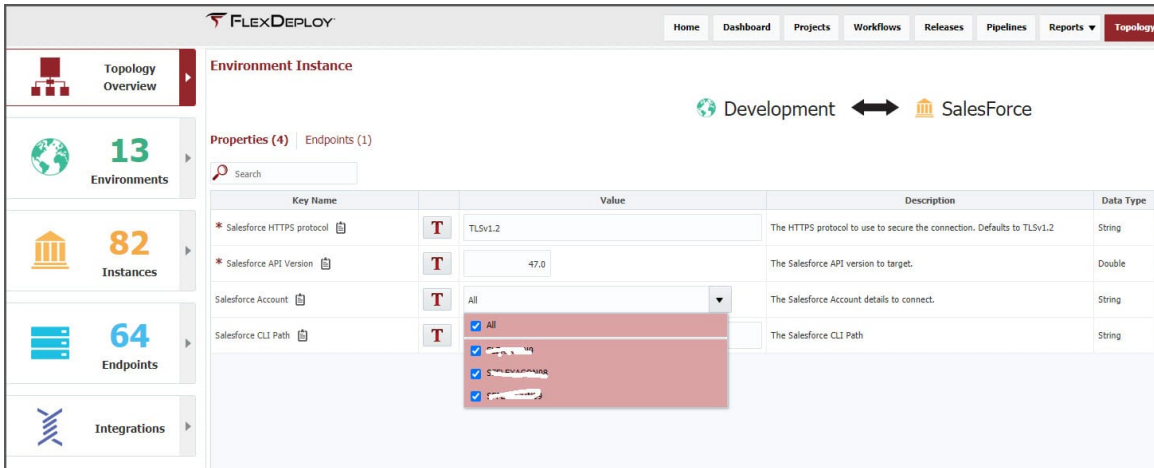


The screenshot shows the FlexDeploy interface with the 'Topology' tab selected. On the left sidebar, the 'Integrations' icon is circled. The main content area displays the configuration for a 'SF Flexagon Account'. The 'Provider' is set to 'Salesforce'. The 'Active' checkbox is checked. The 'Properties' section includes fields for 'User Email', 'Authentication User Email', 'JWT Key File', 'Password', 'Client ID', 'OAuth client ID', 'Development Org Type', 'Salesforce URL', and 'Sandbox Name'. The 'Password' field is highlighted with a red circle.

Repeat the connection creation for your other environment Orgs.

## Associate Salesforce Accounts to Environments

After configuring the Salesforce Org accounts, you need to associate them with the environment. To achieve this, go to Topology Overview and click the Environment Instance you would like to associate with the accounts.



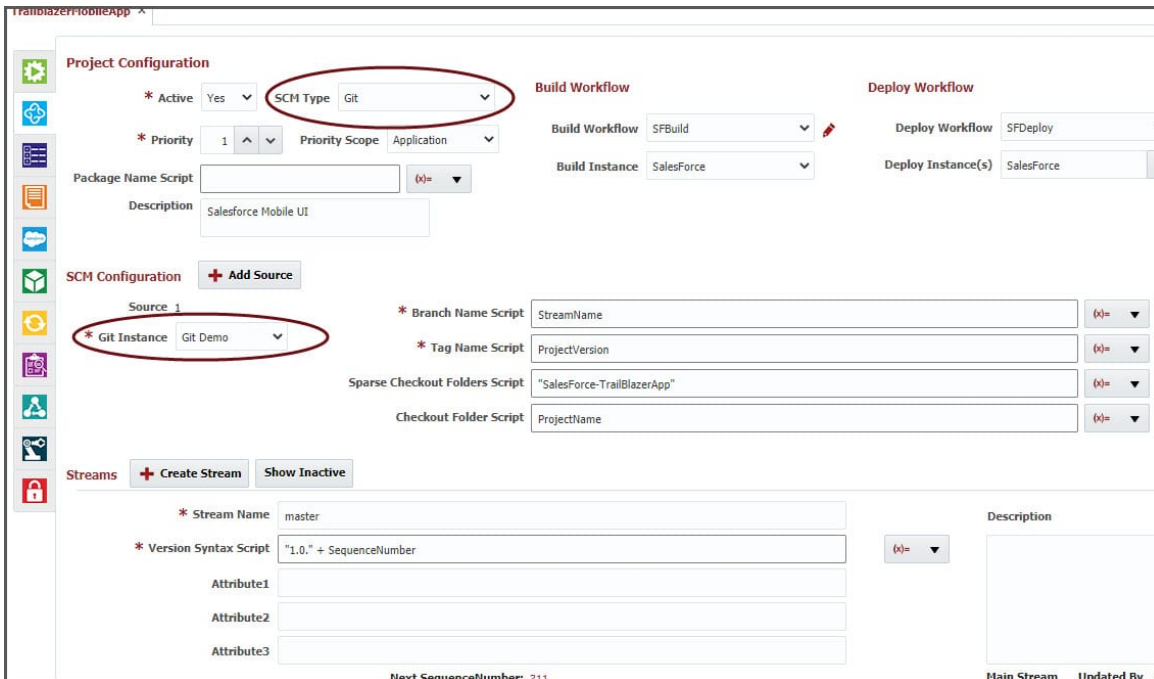
The screenshot shows the 'Environment Instance' configuration page in FlexDeploy. The page is titled 'Development' and 'SalesForce'. It displays a table of properties for the environment. The 'Salesforce Account' property is highlighted, showing a dropdown menu with 'All' and several other accounts selected.

Key Name	Value	Description	Data Type
* Salesforce HTTPS protocol	TLSv1.2	The HTTPS protocol to use to secure the connection. Defaults to TLSv1.2	String
* Salesforce API Version	47.0	The Salesforce API version to target.	Double
Salesforce Account	All	The Salesforce Account details to connect.	String
Salesforce CLI Path		The Salesforce CLI Path	String

You can select one or many accounts per environment.

## Project Configuration

Create a project in FlexDeploy as a Salesforce type and configure your SCM in the Project Configuration tab. Select SCM type and instance. In this example, we use Git. That is all you need to make your Salesforce development source driven. FlexDeploy's seamless integration with SCMs will take care of adding, retrieving, committing, pushing, etc.



The screenshot shows the 'Project Configuration' page in FlexDeploy. The page is titled 'Project Configuration' and shows various configuration options for a project. The 'SCM Type' is set to 'Git' and the 'Git Instance' is set to 'Git Demo'. The 'Build Workflow' is set to 'SFBUILD' and the 'Deploy Workflow' is set to 'SFDDeploy'.

**Project Configuration**

- \* Active: Yes
- \* Priority: 1
- Package Name Script: (x)=
- Description: Salesforce Mobile UI

**SCM Configuration**

- Source 1: \* Git Instance: Git Demo
- \* Branch Name Script: StreamName
- \* Tag Name Script: ProjectVersion
- Sparse Checkout Folders Script: "SalesForce-TrailBlazerApp"
- Checkout Folder Script: ProjectName

**Build Workflow**

- Build Workflow: SFBUILD
- Build Instance: Salesforce

**Deploy Workflow**

- Deploy Workflow: SFDDeploy
- Deploy Instance(s): Salesforce

**Streams**

- \* Stream Name: master
- \* Version Syntax Script: "1.0." + SequenceNumber
- Attribute1:
- Attribute2:
- Attribute3:

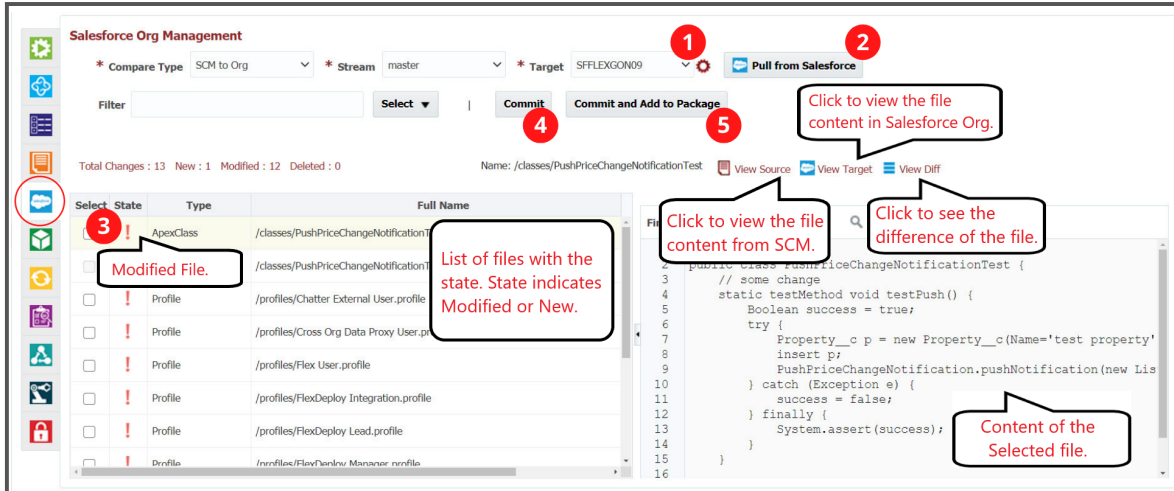
Next SequenceNumber: 311

Main Stream Updated By: dl



## Review Salesforce Changes in FlexDeploy and Commit to SCM

FlexDeploy's integration with Salesforce provides rich features to retrieve changes from a Salesforce Org. The Org can be your Sandbox, Scratch Org, Development or Production edition. This screen allows you to select the Salesforce Org and retrieves the changes. It also displays the status of the file compared with your SCM.



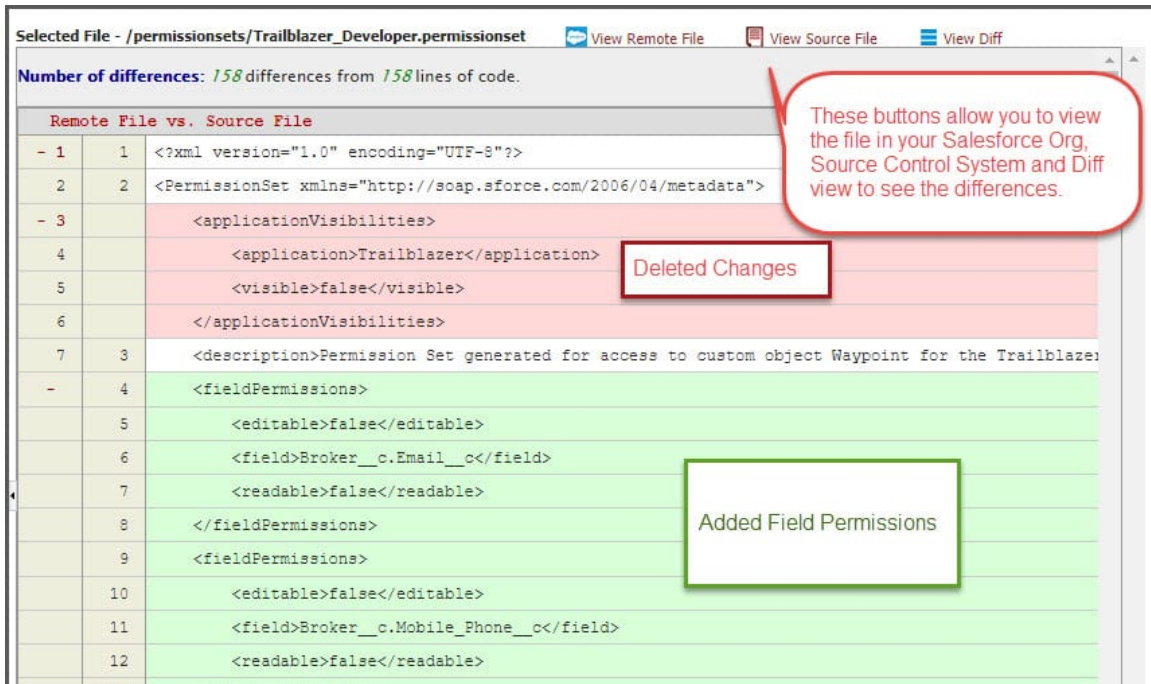
1. Select the Salesforce Org you would like to retrieve changes from and view.

2. Click on the Pull from Salesforce.

3. FlexDeploy will display the list of files with the State, Type, Full file name, and User who worked on it.

a. You can select the file and view the content on the right panel.

b. You can select to view the Remote file, i.e. what is in Salesforce, file in your SCM, and the difference.



Selected File - /permissionsets/Trailblazer\_Developer.permissionset

View Remote File View Source File View Diff

Number of differences: 158 differences from 158 lines of code.

Remote File vs. Source File

Deleted Changes

Added Field Permissions

These buttons allow you to view the file in your Salesforce Org, Source Control System and Diff view to see the differences.



4. Once the review is completed, you can select the files that you would like to commit to the SCM.
5. You can also choose to commit and create a package in FlexDeploy to build, and deploy that package to other Salesforce environments.

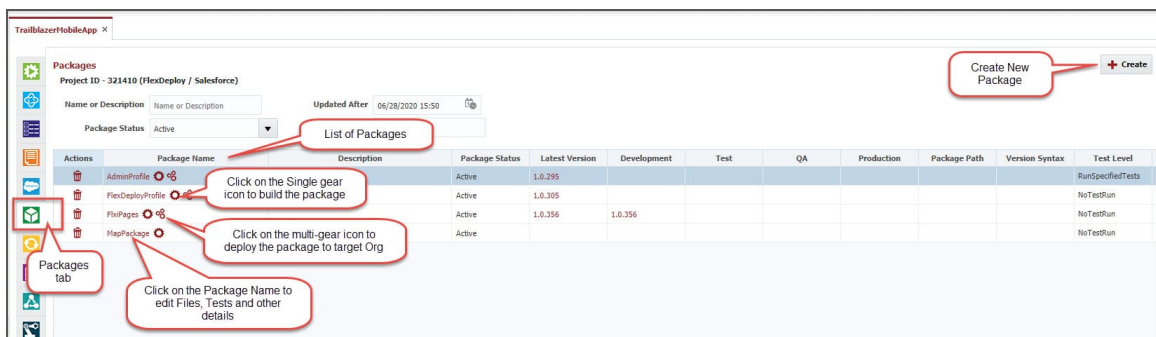
FlexDeploy's extensive integrations with Salesforce, SCM like GIT, Project Management tools like Jira, Service/Change Management tools like ServiceNow, and communication tools like Slack, MS Teams, and email will make retrieving changes from Salesforce, reviewing, and tracking in SCM easy for developers as well as administrators.

## Deploy Salesforce Apps and Metadata Objects

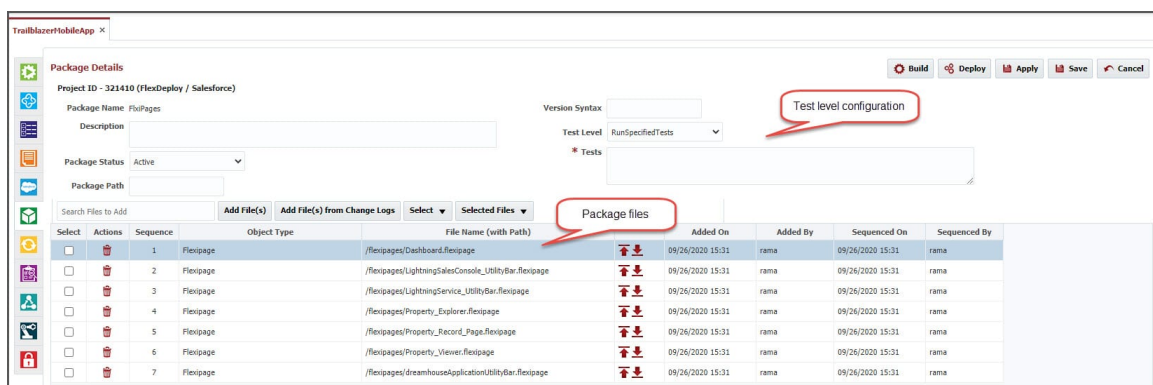
In this chapter, we will see how easy it is to deploy your changes to your Test and production Orgs with FlexDeploy.

### Create Salesforce Package

The step to create a Package is optional with FlexDeploy. But for better control and flexibility of deployments, we suggest you use FlexDeploy's Packages features. The Packages tab on your FlexDeploy project allows you to view existing packages and create or modify the files within the existing packages.



These packages can also be created directly when you commit changes on the Salesforce tab. Click on the Package name to view and edit the details. The Package Details screen displays the files in and other configurations about the package. You can add new files or remove files from the package. A package can consists of any number of different types of files. You can control what unit tests need to be executed by setting the Test Level and Tests.

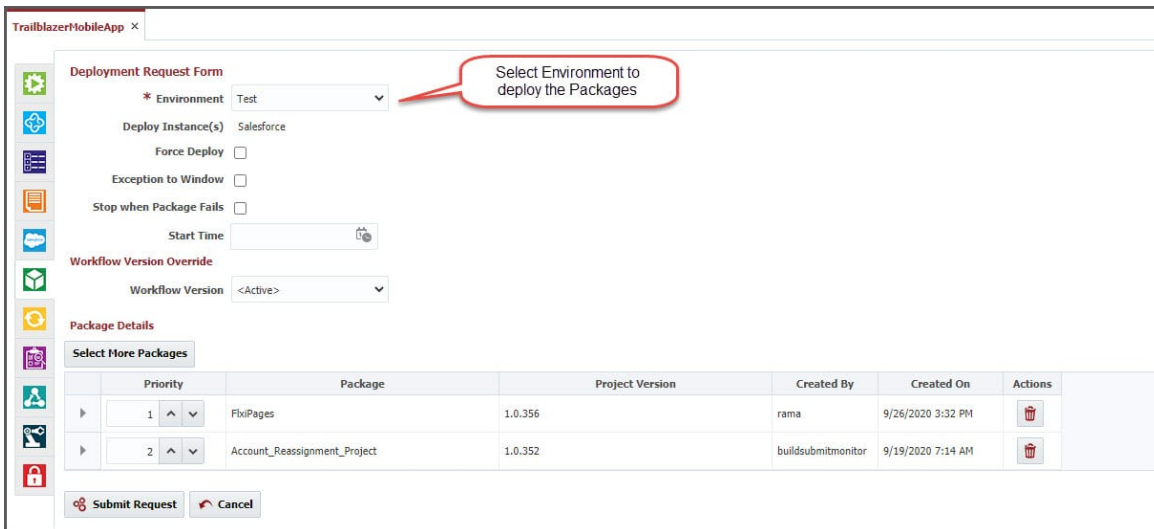


## Build Salesforce Package

Once you have created the Package, you can click on the Build button (or gear icon) to build the package. It will bring up the build request page where you can select the branch you would like to build from if different than what is configured. The Build operation will pull the changes from your SCM and create the package that can be deployed to any Salesforce environment. FlexDeploy will generate the package.xml based on the files selected and test configurations. The generated package artifact will be versioned and added to the Artifact Repository.

## Deploy Salesforce Package(s)

After completing the build, you can choose to deploy one or many packages to your target Salesforce environment. The Deploy request page will allow you to select the Salesforce environment where you would like to deploy. Simply select the environment, review the packages and click Submit Request.



**Deployment Request Form**

\* Environment: Test

Deploy Instance(s): Salesforce

Force Deploy: ☐

Exception to Window: ☐

Stop when Package Fails: ☐

Start Time:

Workflow Version Override: <Active>

**Package Details**

Select More Packages

Priority	Package	Project Version	Created By	Created On	Actions
1	FixiPages	1.0.356	rama	9/26/2020 3:32 PM	
2	Account_Reassignment_Project	1.0.352	buildsubmitmonitor	9/19/2020 7:14 AM	

Submit Request Cancel

All the selected packages will be deployed, and any unit tests will be executed as configured. You can see the status on the Execution screen and drill down for all the details and unit test execution details.

TrailblazerMobileApp

Project ID - 321410 (FlexDeploy / Salesforce)

Project Activity

View

Filter

Workflow Requests

Workflow Executions

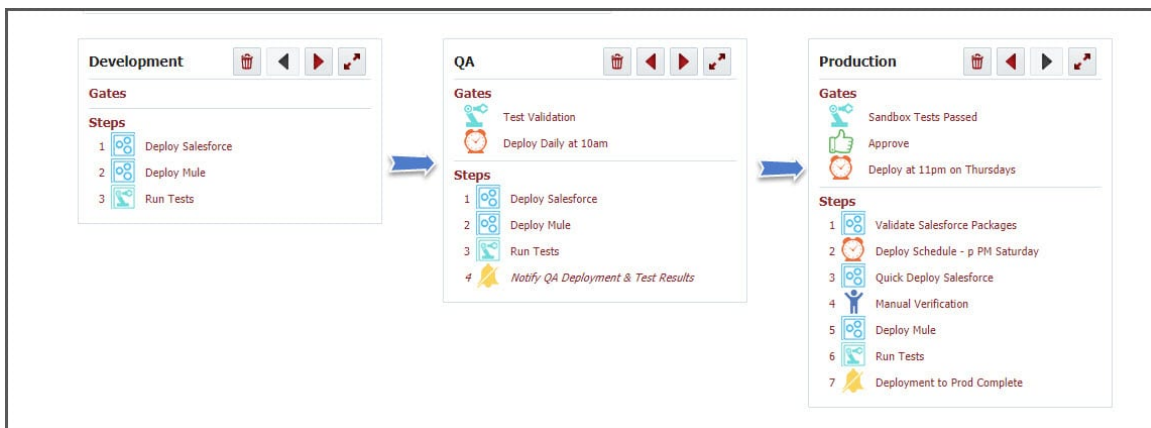
Request Id	Request Status	Requested By	Requested On	Environment	Instance	Workflow Type	Status	Package Name	Project Version	Stream	Execution Id	Start Time	End Time	Ref Name
1281511	Completed	rama	09/26/2020 15:37	Development	Salesforce	DEPLOY	Success	FixiPages	1.0.356	master	1834389	09/26/2020 15:37	09/26/2020 15:39	
1281507	Completed	rama	09/26/2020 15:32	Development	Salesforce	BUILD	Success	FixiPages	1.0.356	master	1834365	09/26/2020 15:32	09/26/2020 15:32	
1280462	Completed	buildsubmitmonitor	09/24/2020 07:56	Development	Salesforce	BUILD	Success	Account_Reassignment_Project	1.0.355	master	1831408	09/24/2020 07:56	09/24/2020 07:56	Weekly HuleSoft and S...
1280447	Completed	stepworker	09/23/2020 20:33	Development	Salesforce	TEST	Success	Hortage	1.0.347	master	1831282	09/23/2020 20:33	09/23/2020 20:34	Continuous Deployment...
1280444	Completed	buildsubmitmonitor	09/23/2020 20:33	Development	Salesforce	BUILD	Success	Account_Reassignment_Project	1.0.354	master	1831269	09/23/2020 20:33	09/23/2020 20:33	Continuous Deployment...
1279980	Completed	stepworker	09/21/2020 17:57	Development	Salesforce	TEST	Success	Hortage	1.0.347	master	1829803	09/21/2020 17:57	09/21/2020 17:58	Oracle EBS_OIC_Sales...
1279889	Completed	buildsubmitmonitor	09/21/2020 17:56	Development	Salesforce	BUILD	Success	Account_Reassignment_Project	1.0.353	master	1829552	09/21/2020 17:56	09/21/2020 17:56	Oracle EBS_OIC_Sales...
1279317	Completed	stepworker	09/19/2020 07:21	Development	Salesforce	TEST	Success	Hortage	1.0.347	master	1828016	09/19/2020 07:21	09/19/2020 07:21	Weekly HuleSoft and S...
1279261	Completed	buildsubmitmonitor	09/19/2020 07:14	Development	Salesforce	BUILD	Success	Account_Reassignment_Project	1.0.352	master	1827591	09/19/2020 07:14	09/19/2020 07:14	Weekly HuleSoft and S...
1279158	Completed	stepworker	09/17/2020 19:43	Development	Salesforce	TEST	Success	Hortage	1.0.347	master	1827263	09/17/2020 19:43	09/17/2020 19:43	Continuous Deployment...

It is that simple to start deploying your Salesforce changes with FlexDeploy.

## Release Automation

FlexDeploy's Pipelines and Release Automation features take Salesforce support to the next level with end-to-end automation and orchestration to deploy your changes to Production.

You can create a Pipeline with a simple drag and drop editor to automate your complete release process. FlexDeploy allows you to include any technology and govern the process to ensure high quality and repeatable releases with visibility to everyone.



With the extensive features provided in FlexDeploy's Pipeline and Release Management, you can automate your process however you would like.

- Set up your CI and CD to create and deploy packages on commits.
- Schedule your deployments so they can execute during non-peak hours and avoid conflicts.
- Automate your Apex Unit tests and UI Automation testing using FlexDeploy's Test automation framework.
- Validate Salesforce Packages first and schedule the quick deployment (commit the package) in Production.
- Fail the release based on test results or code coverages. As a result notify a Developer to fix.
- Notify team members of and assign tasks based on test results

FlexDeploy allows you to start a simple or more complex process, so that based on your current process and maturity level you can start and keep enhancing your release process to achieve higher quality and faster releases.

## Build and Deploy Salesforce DX Apps

Next, we will see how easy it is to deploy your changes from SFDX to your test and production orgs with FlexDeploy.

### Salesforce Developer Experience

Salesforce Developer Experience (DX) is a new way to develop and manage Salesforce apps using scratch orgs and version control. Scratch orgs are available for the Dev Hub enabled Salesforce platforms. These scratch orgs consist of Salesforce code or metadata that can be easily created or destroyed, helping to speed up the standard development workflow.

Salesforce DX is primarily used by advanced developers with version control and using Visual Studio. FlexDeploy provides all the features needed to embrace the new development process with best practices. FlexDeploy enables all DX features to administrators and no-code or low-code developers, regardless of their experience using tools like version control or Visual Studio. At the same time, FlexDeploy allows advanced Salesforce developers to take advantage of full CI/CD automation. If you are thinking about moving to SFDX, FlexDeploy helps you transform to new SFDX without needing to learn new tools.

Once you enable the Dev Hub feature in your Salesforce org, you can manage scratch orgs and packages from FlexDeploy and integrate with version control seamlessly. FlexDeploy allows you to deploy to test and prod orgs using release and pipelines automation, ensuring quality, approvals, and visibility.

### Enable Dev Hub Features in Your Org

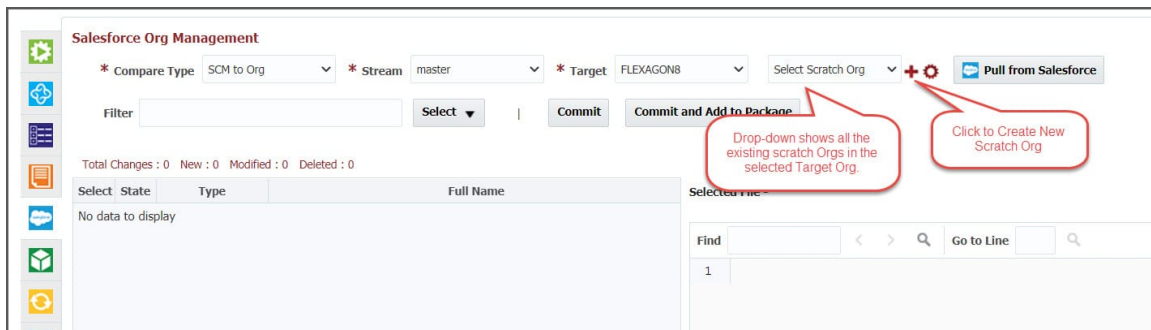
Enabling Dev Hub in a development, production, or business org is safe and doesn't cause any performance or customer issues. To enable Dev Hub in an org:

- Log in as System Administrator to your Developer Edition, trial, or production org (for customers), or your business org (for ISVs).
- From Setup, enter Dev Hub in the Quick Find box and select Dev Hub.
- To enable Dev Hub, click Enable.

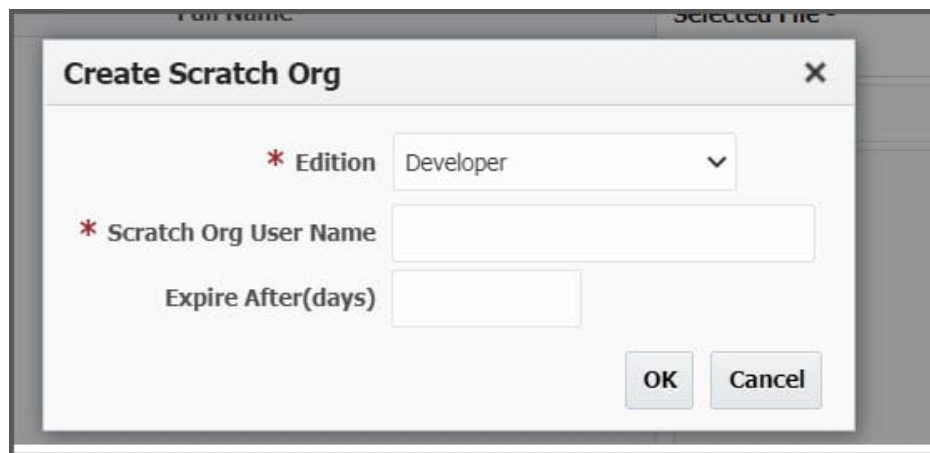
\*NOTE: After you enable Dev Hub, you can't disable it.

## Manage Scratch Orgs

FlexDeploy's Salesforce Org Management screen automatically brings up all the scratch orgs created if the Dev Hub org is selected. You can view and create your new scratch org by clicking on the + icon next to the "Select Scratch Org" drop-down.



After clicking the + to create new, you can enter the details for the new scratch org and click OK to create. Once the new scratch org is created, it will be available in the drop-down for comparing and pulling changes.



## Pull Changes from Your Scratch Org

From the Salesforce Org Management screen, you can select your desired scratch org from the drop down, and retrieve all your changes made. This will allow you to compare changes with version control or with any org, such as a test or prod, and view the differences. On this screen you can view the new, modified, or deleted changes, compare each file, and view the differences. Once you review the changes, you can simply click a button to commit to your version control if you have it enabled for your enterprise.

## Create and Manage Packages

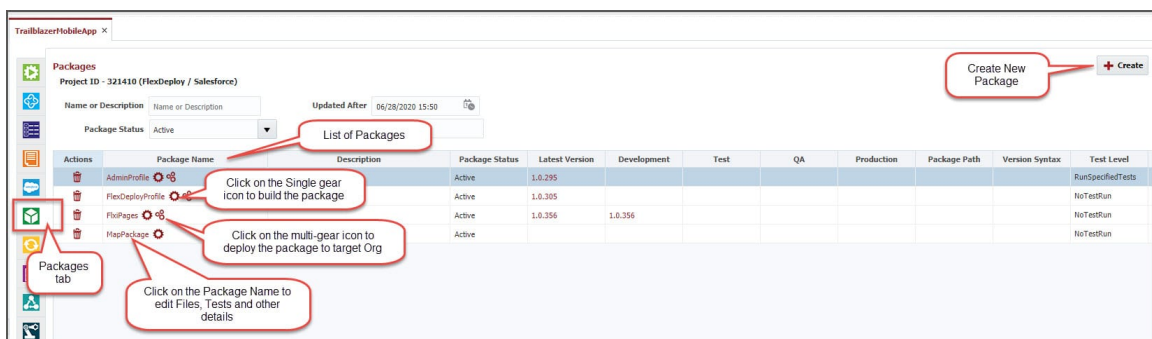
FlexDeploy's Packages screen enables you to view your packages and create or edit the existing packages.

If you are an advanced developer using Visual Studio or any other tool for your development, FlexDeploy allows you to view / edit your packages. As a result, all the packages are in sync with your source. If you edit existing or create new packages, FlexDeploy will update or create a new package in your code and you can view the changes in your tool.

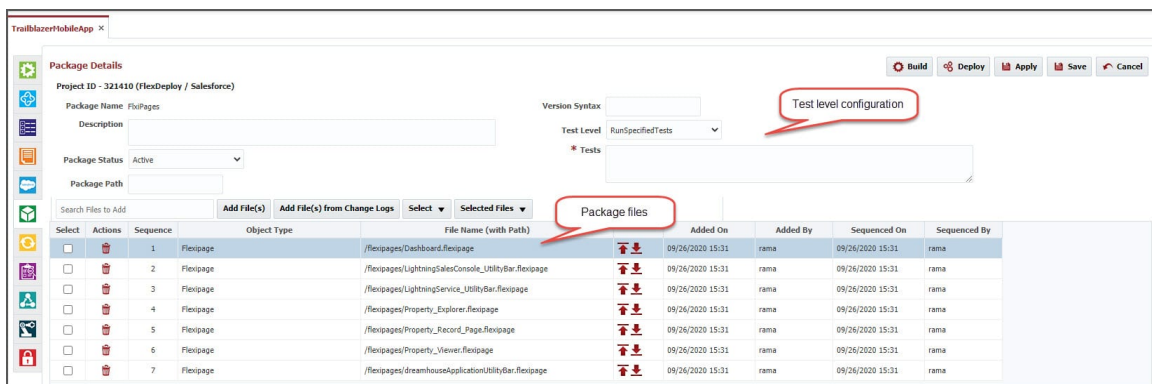
For Administrator or no-code / low-code Salesforce development, you can completely manage your packages on the FlexDeploy Packages screen so that you can create new packages and manage them with few clicks.

Regardless of the type of development, users can:

- Define unit test type for packages in FlexDeploy.
- Manage dependencies by selecting multiple packages with desired order to deploy to target orgs.
- Create package versions (Build) from FlexDeploy Package screen.
- View what versions of packages are installed in various test and prod orgs.



The Package details screen show the details of a selected package.



## Release Automation

Once you have changes pulled from scratch org and created packages in FlexDeploy, you can use CI/CD and release automation features. FlexDeploy helps automate the complete Salesforce release with a simple drag-and-drop for creating reusable pipelines so you can build and deploy Salesforce DX Apps with ease.

This process is similar to non-DX projects or any other technology in your enterprise. In addition to Salesforce, you can deploy changes to any technology like Mule Services, Web Applications, etc. The release pipelines can include automated and manual steps.



FlexDeploy allows you to transition from traditional Salesforce development to modern DX seamlessly. As a result, you can achieve high quality release of your changes with best practices and automated quality checks. If you are a DX developer, take advantage of the advanced features to achieve high quality releases with speed.



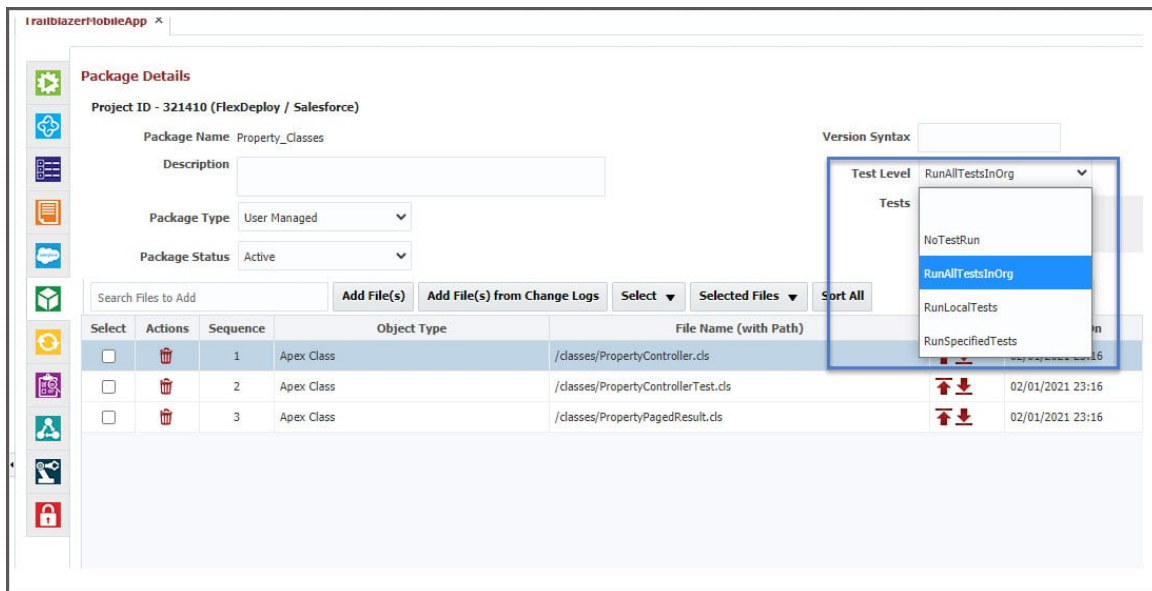
## Unit Test and Test Automation for Salesforce

FlexDeploy's extensive set of features for Salesforce and Test automation with test plugins:

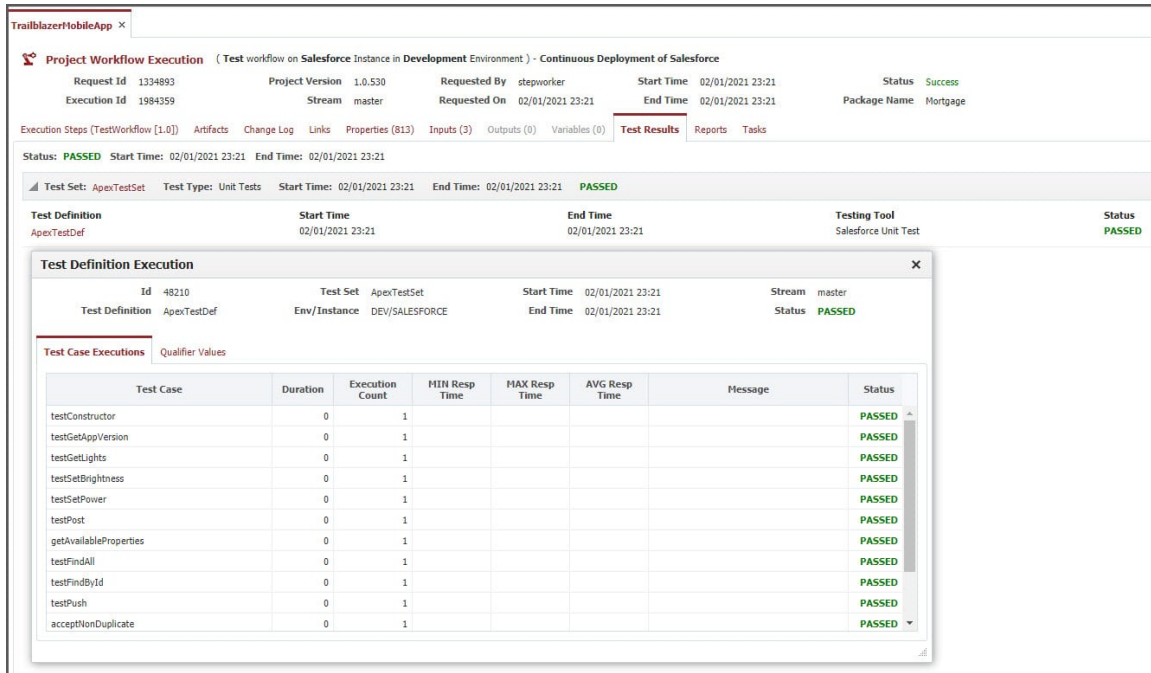
- Ensure that Salesforce changes work as expected.
- Provide a suite of regression tests that can be rerun every time so you don't break existing functionality.
- Meet code coverage requirements as per your preset thresholds.
- Ensure high-quality deployment to production and happy business users.

### Unit Tests

The Apex testing framework enables you to write and execute tests for your Apex classes and triggers. Apex unit tests ensure high quality for your Apex code and let you meet the requirements for deploying Apex. You can configure your test levels at the package level and ensure that tests are executed during deployment.



Apex tests will be executed as part of the deploy or validate package and results will be captured. You can view the results in FlexDeploy and also automate deployment promotions based on test results, ensuring your package will get promoted to the next environment only if you have successful results. Users can be confident the required unit tests are executed and have visibility on what got executed.



**Project Workflow Execution** (Test workflow on Salesforce Instance in Development Environment) - Continuous Deployment of Salesforce

Request Id: 1334893 | Project Version: 1.0.530 | Requested By: stepworker | Start Time: 02/01/2021 23:21 | Status: Success  
 Execution Id: 1984359 | Stream: master | Requested On: 02/01/2021 23:21 | End Time: 02/01/2021 23:21 | Package Name: Mortgage

Execution Steps (TestWorkflow [1.0]) | Artifacts | Change Log | Links | Properties (813) | Inputs (3) | Outputs (0) | Variables (0) | **Test Results** | Reports | Tasks

Status: **PASSED** | Start Time: 02/01/2021 23:21 | End Time: 02/01/2021 23:21

Test Set: ApexTestSet | Test Type: Unit Tests | Start Time: 02/01/2021 23:21 | End Time: 02/01/2021 23:21 | **PASSED**

**Test Definition**  
 ApexTestDef | Start Time: 02/01/2021 23:21 | End Time: 02/01/2021 23:21 | Testing Tool: Salesforce Unit Test | Status: **PASSED**

**Test Definition Execution**

Id	Test Set	Start Time	Stream	Status
48210	ApexTestSet	02/01/2021 23:21	master	<b>PASSED</b>
Test Definition	ApexTestDef	Env/Instance: DEV/SALESFORCE	End Time: 02/01/2021 23:21	<b>PASSED</b>

**Test Case Executions** | Qualifier Values

Test Case	Duration	Execution Count	MIN Resp Time	MAX Resp Time	AVG Resp Time	Message	Status
testConstructor	0	1					<b>PASSED</b>
testGetAppVersion	0	1					<b>PASSED</b>
testGetLights	0	1					<b>PASSED</b>
testSetBrightness	0	1					<b>PASSED</b>
testSetPower	0	1					<b>PASSED</b>
testPost	0	1					<b>PASSED</b>
getAvailableProperties	0	1					<b>PASSED</b>
testFindAll	0	1					<b>PASSED</b>
testFindById	0	1					<b>PASSED</b>
testPush	0	1					<b>PASSED</b>
acceptNonDuplicate	0	1					<b>PASSED</b>

## Code Coverage

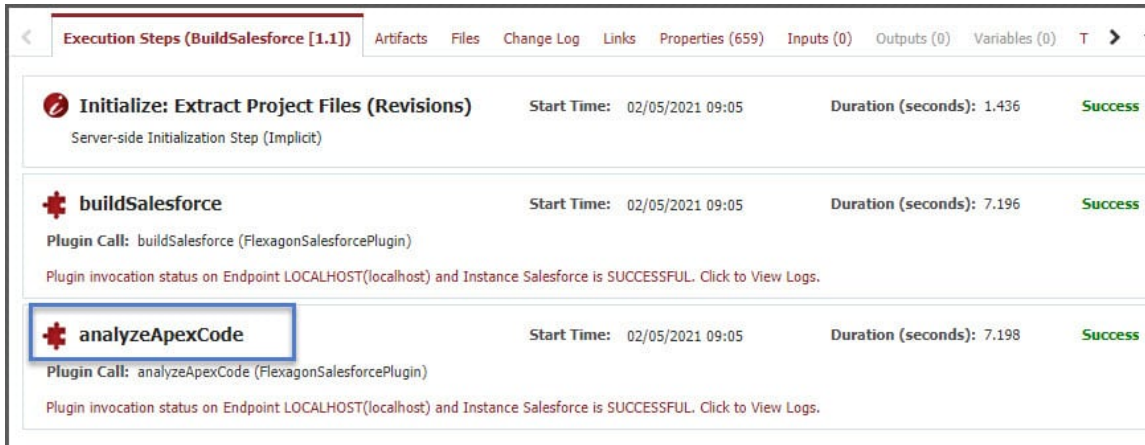
Define code coverage thresholds in FlexDeploy. For example, you can stipulate that at least 75% of Apex code must be covered, and all those tests must pass. From your Apex Test, the code coverage percentage will return as an output. You can look at those results for manual approval or define a gate in your pipeline to act upon.

For example, FlexDeploy can automatically notify the developer/team to work on the unit tests to increase the coverage.

## Static Code Analysis

Along with Unit testing, most of our customers use static code analysis tools like PMD and SonarQube to increase the quality of their code. These tools help find Bugs, Security Vulnerabilities, and Code Smells. FlexDeploy has support for both of these tools such that you can automate the execution after pulling the code from your Org or before deploying. You can view the results in FlexDeploy and also take automated actions based on the results.

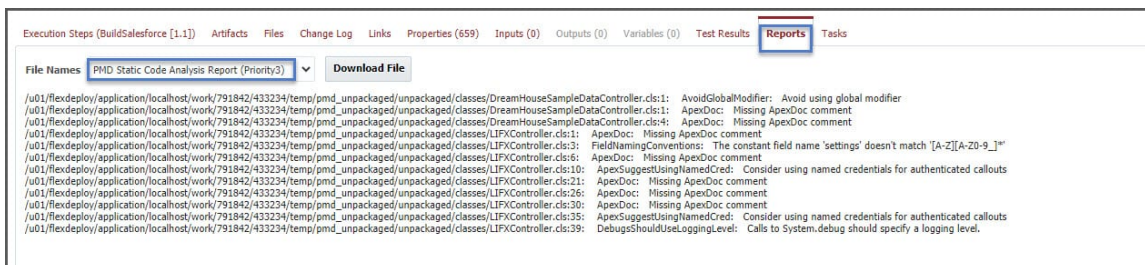
This analysis is normally done after pulling the Apex code from a Salesforce Org or Source Control (like Git). You could also configure your workflow to re-run the analysis before approving the deployment to each environment.



The screenshot shows the 'Execution Steps (BuildSalesforce [1.1])' interface. The 'analyzeApexCode' step is highlighted with a blue box. The step details include:

- Step Name:** analyzeApexCode
- Start Time:** 02/05/2021 09:05
- Duration (seconds):** 7.198
- Status:** Success
- Plugin Call:** analyzeApexCode (FlexagonSalesforcePlugin)
- Message:** Plugin invocation status on Endpoint LOCALHOST(localhost) and Instance Salesforce is SUCCESSFUL. Click to View Logs.

FlexDeploy will execute the analysis based on any custom rule files if you have them. Otherwise, it uses the default rules. You can view reports in FlexDeploy for manual approvals or you can configure to fail or auto promote the package based on the results.



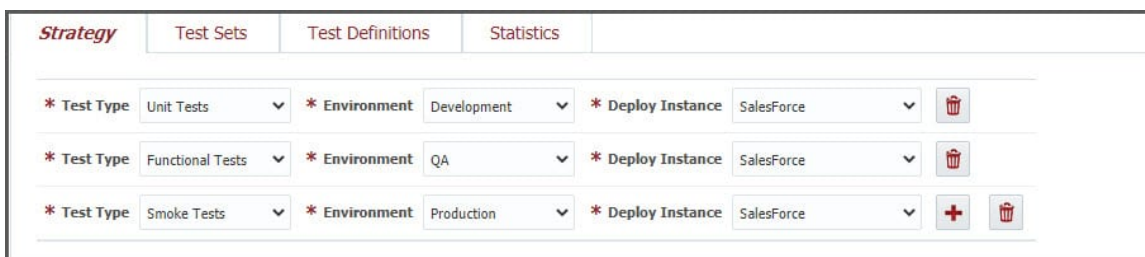
The screenshot shows the 'Reports' tab in FlexDeploy. A dropdown menu is open, showing a list of file names and their corresponding issues. The issues are categorized by priority and include details about the specific code problems.

File Names	Issues
/u01/flexdeploy/application/localhost/work/791842/433234/temp/pmd_unpackaged/unpackaged/classes/DreamHouseSampleDataController.cls:1	AvoidGlobalModifier: Avoid using global modifier
/u01/flexdeploy/application/localhost/work/791842/433234/temp/pmd_unpackaged/unpackaged/classes/DreamHouseSampleDataController.cls:1	ApexDoc: Missing ApexDoc comment
/u01/flexdeploy/application/localhost/work/791842/433234/temp/pmd_unpackaged/unpackaged/classes/DreamHouseSampleDataController.cls:4	ApexDoc: Missing ApexDoc comment
/u01/flexdeploy/application/localhost/work/791842/433234/temp/pmd_unpackaged/unpackaged/classes/LIFXController.cls:1	ApexDoc: Missing ApexDoc comment
/u01/flexdeploy/application/localhost/work/791842/433234/temp/pmd_unpackaged/unpackaged/classes/LIFXController.cls:3	FieldNamingConventions: The constant field name 'settings' doesn't match '[A-Z][A-Z0-9_]'
/u01/flexdeploy/application/localhost/work/791842/433234/temp/pmd_unpackaged/unpackaged/classes/LIFXController.cls:6	ApexDoc: Missing ApexDoc comment
/u01/flexdeploy/application/localhost/work/791842/433234/temp/pmd_unpackaged/unpackaged/classes/LIFXController.cls:10	ApexSuggestUsingNamedCred: Consider using named credentials for authenticated callouts
/u01/flexdeploy/application/localhost/work/791842/433234/temp/pmd_unpackaged/unpackaged/classes/LIFXController.cls:21	ApexDoc: Missing ApexDoc comment
/u01/flexdeploy/application/localhost/work/791842/433234/temp/pmd_unpackaged/unpackaged/classes/LIFXController.cls:26	ApexDoc: Missing ApexDoc comment
/u01/flexdeploy/application/localhost/work/791842/433234/temp/pmd_unpackaged/unpackaged/classes/LIFXController.cls:30	ApexDoc: Missing ApexDoc comment
/u01/flexdeploy/application/localhost/work/791842/433234/temp/pmd_unpackaged/unpackaged/classes/LIFXController.cls:35	ApexSuggestUsingNamedCred: Consider using named credentials for authenticated callouts
/u01/flexdeploy/application/localhost/work/791842/433234/temp/pmd_unpackaged/unpackaged/classes/LIFXController.cls:39	DebugsShouldUseLoggingLevel: Calls to System.debug should specify a logging level.

Based on the priority, any issues get a separate report file to view and/or download.

## Test Automation

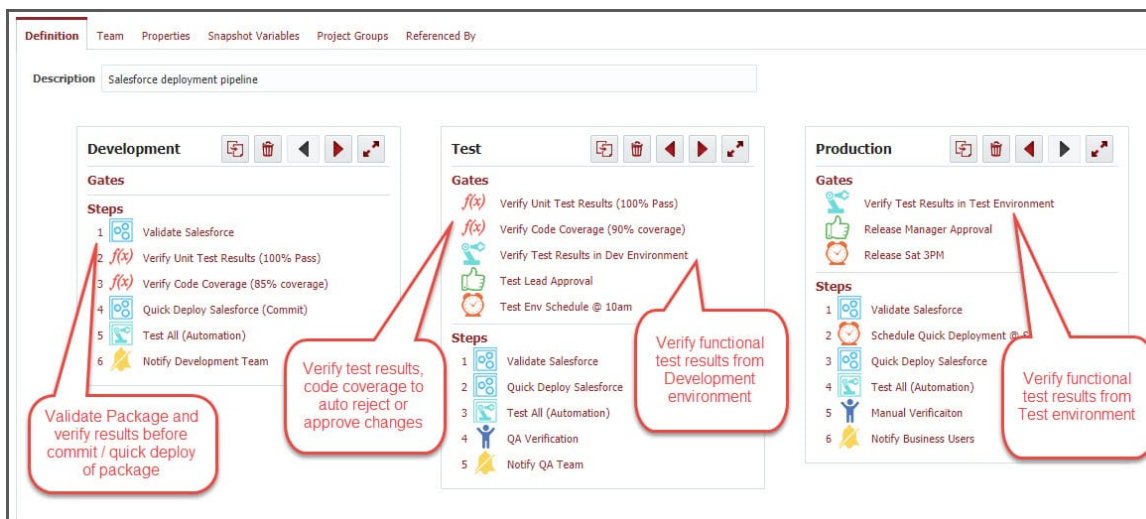
FlexDeploy's full-grown Test Automation framework can be utilized to configure what tests run in each environment. Test automation tools like Selenium, Tosca, HP UFT, and Automation Anywhere can be used to execute tests. Simply configure FlexDeploy to execute the tests and automate the code promotions based on the test results.



The screenshot shows the 'Strategy' configuration table in FlexDeploy. The table has columns for Test Sets, Test Definitions, and Statistics. The configuration is as follows:

Test Sets	Test Definitions	Statistics
* Test Type: Unit Tests	* Environment: Development	* Deploy Instance: Salesforce
* Test Type: Functional Tests	* Environment: QA	* Deploy Instance: Salesforce
* Test Type: Smoke Tests	* Environment: Production	* Deploy Instance: Salesforce

FlexDeploy has a robust pipeline and release automation system. With that, all the results from the same environment or any of the previous environments can be verified prior to promoting the changes



FlexDeploy provides many capabilities in Test and Code review to achieve high-quality Salesforce deployments. But you don't need to have all of it on day one. You can mature the process as you go with simple configurations.

## In Conclusion

While the Salesforce ecosystem is widely used and loved globally, migrating its changes across environments is a manual, tedious, and error-prone process that lacks visibility. This is challenging for IT teams, leads to unhappy leaders, and results in disappointed customers.

*The solution? DevOps.*

There are several DevOps solutions for Salesforce, including Salesforce's own DevOps Center, Salesforce-specific DevOps platforms, and enterprise DevOps platforms. While each solves the challenges previously outlined, each option has limitations. Thus, teams must evaluate their entire technology landscape and larger goals as an IT department and organization.

To support the Salesforce ecosystem and other technologies in your toolchain, an enterprise DevOps platform is the best option. It enables you and your team to build, test, deploy, release, and monitor your SDLC in one platform. A good platform focuses on the developer experience.

FlexDeploy is an enterprise DevOps platform for continuous delivery and release automation. With native support for over 100 open source and commercial technologies and a continuous focus on the developer experience, FlexDeploy supports you wherever you are in your automation journey.

FlexDeploy is used by customers around the globe and is a leader in the Forrester Wave for Continuous Delivery and Release Automation.

If you'd like to streamline your deployment and release pipelines and deliver higher quality software faster, and with lower cost and risk, please reach out to us at [info@flexagon.com](mailto:info@flexagon.com) or check out [flexagon.com](https://flexagon.com) to learn more.

# DevOps for Salesforce



Any questions or comments?  
[info@flexagon.com](mailto:info@flexagon.com)