

Introduction - Why are we even doing it like this

Salesforce deployments are essential for managing and evolving Salesforce environments, especially in a consulting company setting. There are several methods for deploying metadata between organizations, including Change Sets, the Metadata API, and the Salesforce Command Line Interface (CLI). Each method has its unique advantages, but the introduction of Salesforce DX (SFDX) has revolutionized the process, making SFDX-based deployments the standard for the future.

The main reasons are because it is easy to deploy, and easy to revert to a prior version of anything you deploy as well - proper CI/CD depends on GIT being used, which ensures that everything you do can be rolled back in case of bugs.

A table of deployment methods with advantages and disadvantages

| Deployment Method | Advantages | Disadvantages |
|----------------------------|--|---|
| Change Sets | <ul style="list-style-type: none">- Easy to use with a graphical interface- No additional setup required | <ul style="list-style-type: none">- Limited to connected orgs- Manual and time-consuming- No version control- Can be done ad-hoc |
| Metadata API | <ul style="list-style-type: none">- Supports complex deployments- Can be automated- Broad coverage | <ul style="list-style-type: none">- Requires programming knowledge- Steeper learning curve |
| Salesforce CLI (sf) | <ul style="list-style-type: none">- Advanced automation- Supports modern DevOps practices- Version control | <ul style="list-style-type: none">- Steeper learning curve- Initial setup and configuration required- Requires trained staff to maintain |
| Third-Party Tools | <ul style="list-style-type: none">- User-friendly interfaces- Advanced features and integrations | <ul style="list-style-type: none">- Additional costs- May have proprietary limitations |

Despite the complexity inherent in SFcli-based deployments, the benefits are substantial. They enable easy and frequent deployments, better testing by customers, smoother go-lives, and a general reduction in stress around project development and deployment cycles. The structured approach of SF ensures that deployments are reliable, repeatable, and less prone to errors.

To stay fact-based: SF deployments allow deploying multiple times a *week* in a few minutes per deployment. This allows **very easy user testing**, and also allows finding *why* a specific issue cropped up. You can check the Examples section to see how and why this is useful.

It is perfectly true that these deployments require more technical knowledge than third-party tools like Gearset or Changesets. It is our opinion that the tradeoff in productivity is worth the extra training and learning curve.

One thing that is often overlooked - you can NOT do proper CI/CD without plugging the deployment to your project management. This means the entire project management **MUST** be thought around the deployment logic.

This training is split into the following chapters:

- **Chapter 1: The Why, When and By Whom** This chapter explores the fundamental considerations of Salesforce deployments within the context of consulting projects. It addresses:
 - **Why Deploy:** The importance and benefits of deploying Salesforce metadata throughout the project lifecycle, from the build phase to UAT to GoLive.
 - **When:** When in the project timeline should deployments be planned and executed to ensure smooth progress and mitigate risks. (Hint - it's often, but not in every org)
 - **By Whom:** Roles and responsibilities involved in the deployment process, such as consultants committing changes, architects reviewing commits and system elements, and release managers overseeing and executing deployments.

Chapter 2: The What, How and How Frequently This chapter delves into the practical aspects of Salesforce deployments:

- **What:** Overview of the deployment tools used, including GIT, SFDX, SGD, Bitbucket, and your trusty Command Line.
- **How:** Detailed workflows and methodologies for using these tools effectively, tailored to specific roles within the project team (consultants, architects, release managers).
- **How Frequently:** Recommendations on the frequency of deployments throughout the project timeline to maintain agility, minimize conflicts, and ensure continuous integration and delivery.

Chapter 3: An Example Project and Deployment Flow This chapter provides a hands-on example to illustrate a typical project scenario and the corresponding deployment processes:

- **Example Project:** Overview of a hypothetical Salesforce consulting project, including its scope and objectives.
- **Deployment Flow:** Step-by-step walkthrough of the deployment lifecycle, from initial planning and setup through to execution and validation.
- **Best Practices:** Highlighting best practices and potential challenges encountered during the deployment process.

Chapter 4: Configurations, Templates and Setup This chapter focuses on the essential configurations and setup required to streamline the deployment process:

- **Configurations:** Detailed guidance on configuring Salesforce environments for efficient deployment management.
- **Templates:** Templates and reusable patterns for standardizing deployments and ensuring consistency across projects.
- **Setup:** Practical tips and strategies for setting up deployment pipelines, integrating with version control systems, and automating deployment tasks.

These chapters collectively provide a comprehensive guide to mastering Salesforce deployments within a consulting company, covering both strategic considerations and practical implementation details.

Chapters 5 and further contain information for daily work and should be viewed as cheat sheets.

Revision #5

Created 2024-06-18 14:04:49 UTC by Windyo

Updated 2026-03-04 16:32:18 UTC by Windyo