

## Study Guide: App Deployment (13%)

### Overview

The App Deployment section of the Salesforce Platform App Builder Certification Exam, weighted at 13%, tests your ability to manage the lifecycle of Salesforce customizations, from development to production. This topic is critical for ensuring that apps built with declarative tools (e.g., custom objects, fields, workflows) are safely and efficiently deployed across environments. It's all about understanding Salesforce's deployment tools and processes—no coding required—though knowing when to escalate to developers for programmatic solutions is also key.

You'll need to master sandboxes, change sets, packages (managed and unmanaged), and deployment best practices. This guide will dive deep into each area, offering practical insights, real-world scenarios, and study aids to help you succeed on the exam and in actual Salesforce deployments.

---

### Key Concepts

Here's what you'll need to master:

1. **Sandboxes:** Isolated environments for development and testing.
2. **Change Sets:** Declarative tool for deploying metadata between orgs.
3. **Packages:** Bundles of customizations for distribution (Managed vs. Unmanaged).
4. **Deployment Process:** Steps to move changes from sandbox to production.
5. **Metadata:** Components of a Salesforce app (e.g., objects, fields, layouts).
6. **Dependencies:** Ensuring all related components are included in a deployment.
7. **Testing and Validation:** Verifying deployments work as expected.
8. **Best Practices:** Strategies for safe, scalable deployments.

---

### Detailed Explanation

#### 1. Sandboxes

Sandboxes are isolated Salesforce environments used for development, testing, and training, cloned from a production org.

- **Types:**
  - **Developer Sandbox:** Small (200 MB data, 200 MB file storage), for individual development.
  - **Developer Pro Sandbox:** Larger (1 GB data, 1 GB file storage), for team development.
  - **Partial Copy Sandbox:** Includes sample data (5 GB data, 5 GB file storage), for testing.
  - **Full Sandbox:** Exact replica of production (same data/file storage), for final testing.
- **Creation:** Setup → Sandboxes → New Sandbox → Select type.
- **Example:** Build a new Flow:
  - Use a Developer Sandbox to create and test.
- **Real-World Scenario:** Test a major app update:
  - Partial Copy Sandbox → Load sample Accounts and Opportunities → Test.
- **Best Practices:**
  - Refresh sandboxes regularly (e.g., Developer: every 1 day, Full: every 29 days).
  - Use Full Sandboxes for user acceptance testing (UAT).
  - Limit data in Developer Sandboxes to avoid clutter.

## 2. Change Sets

Change Sets are the primary declarative tool for deploying metadata between connected Salesforce orgs (e.g., sandbox to production).

- **Components:**
  - **Outbound Change Set:** Created in source org, listing metadata to deploy.
  - **Inbound Change Set:** Received in target org for validation and deployment.
- **Process:**
  1. Setup → Outbound Change Sets → New.
  2. Add components (e.g., Custom Object, Flow).

3. Upload to target org.
4. Validate and deploy in target org.
  - **Example:** Deploy a custom object:
    - Change Set: “New Project Object” → Add Project\_\_c, fields, layout → Upload.
  - **Real-World Scenario:** Update a sales process:
    - Change Set: Add Opportunity\_Stage\_\_c picklist values, new Flow.
  - **Limitations:**
    - Only works between orgs with a deployment connection (e.g., sandbox to production).
    - No data migration—metadata only (e.g., no records).
    - Manual dependency tracking.
  - **Tips:**
    - Validate before deploying to catch errors.
    - Include all dependencies (e.g., fields referenced in a Flow).
    - Name Change Sets descriptively (e.g., “Release\_v1.2”).

### 3. Packages

Packages bundle Salesforce customizations for distribution, either within an org or externally (e.g., via AppExchange).

- **Types:**
  - **Unmanaged Package:**
    - Open, editable customizations.
    - Use: Sharing internally or with clients.
    - Example: Share a custom Survey\_\_c object with a partner.
  - **Managed Package:**
    - Locked, IP-protected customizations.
    - Use: Selling on AppExchange or enterprise distribution.
    - Example: A billing app with hidden logic.

- **Creation:** Setup → Package Manager → New Package → Add components.
- **Example:** Package a CRM extension:
  - Unmanaged: Lead\_Scoring\_\_c object, Flow, layout.
- **Real-World Scenario:** Distribute a reporting tool:
  - Managed Package: Custom dashboards, reports → Sell on AppExchange.
- **Best Practices:**
  - Use namespaces in Managed Packages (e.g., “myapp\_\_”) for uniqueness.
  - Test Packages in a clean org before release.
  - Document components for recipients.

#### 4. Deployment Process

The deployment process moves customizations from development to production in a structured way.

- **Steps:**
  1. **Develop:** Build in a Developer Sandbox.
  2. **Test:** Move to Partial Copy or Full Sandbox for QA/UAT.
  3. **Deploy:** Use Change Sets or Packages to production.
  4. **Validate:** Check functionality post-deployment.
- **Example:** Deploy a new app:
  - Developer Sandbox → Build Order\_\_c object.
  - Partial Copy → Test with sample data.
  - Change Set → Deploy to production.
- **Real-World Scenario:** Roll out a service update:
  - Full Sandbox → Test Case workflows → Change Set to production.
- **Tips:**
  - Plan deployments during off-hours to minimize disruption.
  - Backup production data before major deployments.

- Document each step for audit trails.

## 5. Metadata

Metadata is the configuration data that defines a Salesforce app, deployable via Change Sets or Packages.

- **Examples:**

- Objects (Account, Project\_\_c).
- Fields (Name, Status\_\_c).
- Page Layouts, Flows, Validation Rules.

- **Tools:**

- Change Sets: Move metadata between connected orgs.
- Packages: Distribute metadata broadly.

- **Example:** Metadata in a Change Set:

- Invoice\_\_c, Amount\_\_c field, Invoice\_Layout.

- **Real-World Scenario:** Update an existing app:

- Metadata: New Priority\_\_c field on Case, updated layout.

- **Best Practices:**

- Understand metadata types (see Salesforce Metadata API docs).
- Avoid deploying partial components (e.g., a Flow without its fields).
- Use sandboxes to test metadata integrity.

## 6. Dependencies

Dependencies are relationships between metadata components that must be included for a deployment to work.

- **Examples:**

- A Flow depends on fields it updates.
- A Layout depends on fields it displays.
- A Validation Rule depends on its formula fields.

- **Checking:** Change Sets auto-prompt for some dependencies (e.g., fields in a layout).
- **Example:** Deploy a Flow:
  - Include: Flow, Status\_\_c field, Case object.
- **Real-World Scenario:** Move a dashboard:
  - Dependencies: Dashboard, underlying reports, report fields.
- **Tips:**
  - Use “View/Add Dependencies” in Change Sets.
  - Test in a sandbox to catch missing pieces.
  - Document dependencies for complex apps.

## 7. Testing and Validation

Testing ensures deployments function as expected; validation checks for errors before deployment.

- **Validation:**
  - Run in target org via Change Set → “Validate.”
  - Checks permissions, dependencies, and conflicts.
- **Testing:**
  - Unit Tests: Not required for declarative (unlike Apex).
  - Functional Tests: Verify in sandbox (e.g., create a record).
- **Example:** Validate a Change Set:
  - Components: Task\_\_c object, Flow → No errors → Deploy.
- **Real-World Scenario:** Test a sales app:
  - Partial Copy Sandbox → Create Opportunities → Confirm Flow triggers.
- **Best Practices:**
  - Validate every Change Set—catches 90% of issues.
  - Test with real user scenarios (e.g., sales rep Profile).
  - Involve end-users in UAT.

## 8. Best Practices

- **Version Control:** Track changes manually (e.g., “v1.1: Added Field X”).
  - **Rollback Plan:** Backup data, note pre-deployment state.
  - **Communication:** Notify users of deployment schedules.
  - **Example:** Deploy safely:
    - Backup: Export Account via Data Loader.
    - Schedule: Weekend deployment → Notify via Chatter.
  - **Real-World Scenario:** Enterprise rollout:
    - Sandbox → Test → Change Set → Production → Post-deployment email.
- 

### Study Guide Tables

**Table 1: Sandbox Types**

Type	Data/File Storage	Refresh Interval	Use Case
Developer	200 MB / 200 MB	1 day	Initial dev
Developer Pro	1 GB / 1 GB	1 day	Team dev
Partial Copy	5 GB / 5 GB	5 days	QA with data
Full	Same as prod	29 days	Final testing

**Table 2: Change Sets vs. Packages**

Feature	Change Set	Package
Scope	Connected orgs	Any org/AppExchange
Type	Metadata only	Managed/Unmanaged
Example	Deploy Flow	Share app
Automation	Manual	Installable

**Table 3: Deployment Checklist**

Step	Task	Tool
Develop	Build components	Sandbox
Test	Verify functionality	Sandbox
Deploy	Move to production	Change Set
Validate	Confirm post-deploy	Production org

---

## Practical Examples

1. **Sandbox:** Test a new object:
    - Developer Sandbox → Create Event\_\_c → Test layout.
  2. **Change Set:** Deploy a workflow:
    - Components: Case Flow, Priority\_\_c field → Upload to prod.
  3. **Package:** Share a tool:
    - Unmanaged Package: Survey\_\_c, layout → Distribute to team.
  4. **Validation:** Check a deployment:
    - Change Set: Order\_\_c → Validate → Deploy.
- 

## Tips for Success

- **Hands-On Practice:** In a Developer org:
  - Create a sandbox, build a custom object.
  - Deploy via Change Set to another sandbox.
  - Package an app and install it.
- **Dependencies:** Always double-check related components.
- **Trailhead Modules:**
  - “App Deployment”
  - “Change Management”

- **Scenarios:** Practice questions like:
    - “Move X from sandbox to production.”
    - “Share Y with a client.”
  - **Test Thoroughly:** Deploy to a sandbox first, test all features.
  - **Timing:** Know refresh intervals—trick exam questions.
  - **Backup:** Practice exporting data as a rollback plan.
- 

### Bullet Point Summary

- **Sandboxes:**
  - Isolate dev/testing.
  - Choose type by need.
  - Refresh strategically.
- **Change Sets:**
  - Deploy metadata.
  - Validate first.
  - Track dependencies.
- **Packages:**
  - Managed for IP protection.
  - Unmanaged for sharing.
  - Test before release.
- **Deployment Process:**
  - Develop → Test → Deploy.
  - Plan downtime.
  - Document steps.
- **Metadata:**
  - Core of deployments.

- Include all pieces.
  - Understand types.
  - **Dependencies:**
    - Auto-prompted in Change Sets.
    - Test to confirm.
    - List manually if needed.
- 

### Study Plan

1. **Day 1:** Study Sandboxes (create one, explore limits).
  2. **Day 2:** Master Change Sets (deploy an object).
  3. **Day 3:** Practice Packages (build Managed/Unmanaged).
  4. **Day 4:** Explore Process and Testing (validate a deployment).
  5. **Day 5:** Review all concepts, test a full scenario.
  6. **Day 6:** Take a practice quiz on App Deployment.
- 

### Exam Scenarios to Practice

1. **Scenario:** “Deploy a new Task\_\_c object to production.”
  - Solution: Developer Sandbox → Change Set → Validate → Deploy.
2. **Scenario:** “Share a reporting app with a partner.”
  - Solution: Unmanaged Package → Add reports, dashboards → Distribute.
3. **Scenario:** “Test a major update before production.”
  - Solution: Full Sandbox → Load data → Test → Change Set.