Canary Deployment Strategy: Short-Term and Long-Term Solutions

Prepared by: Vamshi Krishna Veggalam

# 1. Problem Statement

Our production deployment process is tightly integrated with Jira, ServiceNow, CloudBees, and UDeploy. Even when manifest files are updated to reflect new app names (e.g., for canary releases), the process continues to deploy the original app name due to fixed parameters used in UDeploy. This restricts our ability to test alternate versions safely in production without impacting existing traffic.

# 2. Short-Term Workaround (Within One Sprint)

✅ Clone the Bitbucket repository or branch to isolate the canary version.  
✅ Create a new CloudBees pipeline pointing to the new repo or manifest (e.g., `manifest-canary.yml`).  
✅ Duplicate or slightly modify the existing UDeploy component to deploy the canary app using the new app name.  
✅ Trigger deployment manually or via temporary Jira/ServiceNow record if needed.

# 3. Long-Term Enterprise-Ready Solution

Introduce a parameterized, scalable approach aligned with existing release workflow.

## Steps:

1. Add an 'App Name Override' field to the Jira/ServiceNow form.  
2. Modify CloudBees pipeline to accept `APP\_NAME` and `MANIFEST\_FILE` as input parameters.  
3. Update UDeploy component to consume these parameters for flexible cf push.  
4. Promote this model as a reusable, governed rollout pattern for all progressive delivery scenarios.

# 4. Benefits of the Recommended Solution

- ✅ Safe deployment of canary and main versions side-by-side  
- ✅ No manual changes needed post-UAT  
- ✅ Fully auditable and aligned with existing governance flow  
- ✅ Minimal change to existing process, just parameter injection  
- ✅ Enables feature rollout flexibility using Apigee-based header and developer app flag routing

# 5. Next Steps

• Discuss with Platform/DevOps teams to enable app name/manifest override in CloudBees and UDeploy  
• Request ServiceNow/Jira integration team to expose override fields in the change form  
• Prepare routing rules in Apigee for `X-Feature-Variant` + `canary\_access\_enabled`  
• Begin pilot rollout using the canary model