Chillo Deployments Using Ansible

This document outlines the plan to modernize Chillo deployments by integrating Red Hat Ansible as the standard automation and orchestration platform.

Why Ansible

Currently, multiple teams maintain custom automation tools for artifact deployments, including scripts and Java applications. To standardize and streamline automation efforts, the organization is adopting Red Hat Ansible—a flexible and extensible framework capable of handling various automation tasks, including orchestrated artifact deployments across multiple environments.

Impact on Chillo

No changes will be made to existing Chillo repositories, application code, or logic. The transition focuses solely on replacing the deployment mechanism with Ansible while keeping all other aspects of the system intact.

Automation Overview

GitHub Actions will be used as the primary interface for initiating deployments, providing a user experience similar to the current Jenkins-based process. This approach simplifies adoption and leverages modern CI/CD capabilities.

Implementation Plan

1. **GitHub Workflow Inputs** – A workflow_dispatch event will trigger the process with the following parameters:
- TAG
- Branch
- Environment
- Actions
- Component Filter
- Server Tag Filter

Additional parameters may be added in future iterations.

2. **Execution Plan Generation** – A Docker container will run the Chillo CLI to produce:
- `execution-plan.json` (same as the Jenkins-generated file)
- `servers.yml` (server inventory for the selected environment)

3. **YAML Conversion for Ansible** – A lightweight executable will convert `execution-plan.json` and `servers.yml` into `execution-plan.yml`, a concise yet complete representation for Ansible orchestration.

Example snippet:

```yaml
- name: sdp-canola-config
type: SERVICE
dependsOn:
- componentName: iceamazoncorretto.17.0.3.6.1
currentAction: INSTALL
containers:
- type: RPM
groupId: com.theice.clearing.deliveries
artifactId: ICEdel-config-server-rpm
version: 1.10.3.2.0-SNAPSHOT
serviceCommands:
start: /usr/bin/sudo /var/opt/delcs/scripts/startStop.sh start icus
```

4. **Deployment via Ansible API** – The generated `execution-plan.yml` will be passed as an environment variable (`EXECPLAN`) to Red Hat Ansible, which will orchestrate the deployment process.

5. **Monitoring and Reporting** – Once the deployment completes, Ansible's job status will be checked. On success, the process ends cleanly. On failure, detailed error information and a URL to view the complete Ansible logs will be provided.

Key Benefits

- **Standardization:** Aligns with the company-wide Ansible automation framework.
- **Scalability:** Enables coordinated deployments across multiple servers and environments.
- **Visibility:** Provides clear logging, reporting, and troubleshooting mechanisms.
- **Reusability:** GitHub Actions and Docker-based execution ensure consistent, repeatable automation.
- **Future-Readiness:** Establishes a foundation for integrating broader automation use cases beyond deployments.