Response from Harness Team

1. **Do we have something like a Deployment dashboard and Release dashboard? If yes, how to configure it?**  
   Yes, Harness provides dashboards for deployment-related insights. You can find them in the global view under the dashboard panel in the UI.

<https://developer.harness.io/docs/category/harness-dashboards>

1. **How good are the configuration management capabilities? Can it offer similar functionality to Ansible Tower?**  
   Could you please provide more details on your specific use case so that I can assist you further? This will help me better understand how Harness can meet your configuration management needs.
2. **How do deployments happen? Do we need any agents installed on the target infrastructure?**  
   In Harness, delegates are responsible for performing all tasks on the target infrastructure. You can learn more about delegates and their setup in our documentation: <https://developer.harness.io/docs/platform/delegates/delegate-concepts/delegate-overview/>
3. **What deployment artifacts are supported (e.g., jar, war, exe, nuget, etc.)?**  
   You can find the list of supported deployment artifacts in our documentation <https://developer.harness.io/docs/artifact-registry/whats-supported/>.

My Research

2) Config Management

Harness includes a built-in Secret Management feature that enables you to store encrypted secrets, such as access keys, and use them in your Harness account.

Harness Manager does not have access to your key management system, and only the Harness Delegate, which sits in your private network, has access to it.

<https://developer.harness.io/docs/platform/secrets/secrets-management/harness-secret-manager-overview/>

<https://developer.harness.io/docs/platform/secrets/secrets-management/store-authentication-credentials/>

3) Gating Option

During deployment, the user group members use the Harness Manager to approve or reject the deployment manually.

Approvals are usually added in between stage steps to prevent the stage execution from proceeding without an approval.

<https://developer.harness.io/docs/continuous-delivery/x-platform-cd-features/cd-steps/approvals/using-harness-approval-steps-in-cd-stages/>

<https://developer.harness.io/docs/continuous-delivery/x-platform-cd-features/cd-steps/approvals/using-jira-and-service-now-approval-steps-in-cd-stages/>

4) Deployments

Harness Delegate is a service you run in your local network or VPC to connect your artifacts, infrastructure, collaboration, verification, and other providers with Harness Manager.

The delegate performs all operations, including deployment and integration.

How Harness Manager picks delegates[​](https://developer.harness.io/docs/platform/delegates/delegate-concepts/delegate-overview/#how-harness-manager-picks-delegates)

Harness uses delegates for all operations. For example:

* **Connectors**: Connectors are used for all third-party connections.
* **Pipeline** **Services** and Infrastructure: Connectors are used in Pipeline Service connections to repos and Pipeline Infrastructure connections to target environments (deployment targets, build farms, etc).
* **Pipeline** **Steps**: You can select a delegate in each pipeline step to ensure that the step only uses that delegate to perform its operation.

<https://developer.harness.io/docs/platform/delegates/delegate-concepts/delegate-overview/#task-assignment>

<https://developer.harness.io/docs/platform/connectors/cloud-providers/pdc-connector/>

5) Artifacts

Harness Artifact Registry support a variety of registry types, and new ones are often added. We currently support the registry types listed below.

<https://developer.harness.io/docs/artifact-registry/whats-supported/>

<https://developer.harness.io/docs/platform/connectors/cloud-providers/ref-cloud-providers/artifactory-connector-settings-reference/>