Customer Feature Parity Matrix

Domain	Capability	Current Gen CD	Next Gen	Migrating Over with tool	Migration Path
Account	Account Default	✅	✅	YES	• We use Account Variables - `<+variables.account.[var_id]>` • `${account.defaults.var_id} -> <+variables.account.[var_id]>`
	Resource Constraints	✅	✅	NO - User can create a Pipeline and add a queuing as a step	N/A
	Tag Management	✅	❌	NO - No Centralized Tag Management, however customer can have tags for each entity Test each entity can take that tag and can be referenced in the pipeline - <+tag>	• Application Tag `<+project.tag.{var_id}>` • Service Tags `<+service.tags.{var_id}>` • Environment Tags `<+env.tags.{var_id}>`
	Application Stacks	✅	❌	NO - We are not carrying over Application Stacks from Current Gen	N/A
Delegate	Kubernetes	✅	✅	NO - Delegates will not be migrated over to NG	N/A
	SSH	✅	✅	NO - Delegates will not be migrated over to NG	• User's can install Podman, then install the docker delegate
	ECS	✅	✅	NO - Delegates will not be migrated over to NG	• Harness has docs to create an ECS Delegate based of our Docker Delegte
	Helm	✅	✅	NO - Delegates will not be migrated over to NG	N/A
	Docker	✅	✅	NO - Delegates will not be migrated over to NG	N/A
Cloud Provider	Azure	✅	✅	YES	Azure Environment (CG) Azure Environment U.S. Government (CG) U.S. Government TenantID (CG) Tenant ID Client ID (ApplicationID) [CG] ApplicationID Usage Scoping **Will Not Migrate Over** Delegate Selectors **Will Not Migrate Over**
	GCP	✅	✅	YES	Inherit From Delegate (cg) Inherit Credtianl from Delegate Use Encrypted Key (cg) Specify Credential w/ a Service Account Secret Usage Scoping **Will Not Migrate Over** Delegate Selectors **Will Not Migrate Over**

	AWS	✅	✅	YES	Assume STS Role (CG) Assume STS Role Role ARN (CG) Cross account role ARN External Id (CG) External Id Override Default Region (CG) Test Region Assume IAM Role on Delegate (CG) Assume IAM Role on Delegate Use IRSA (CG) Use IRSA Enter AWS Access Keys Manually (CG) AWS Access Key Usage Scoping **Will Not Migrate Over** Delegate Selectors **Will Not Migrate Over**
	Kubernetes	✅	✅	YES	
	Cloud Foundry (Tanzu Application Services)	✅	✅		
	Physical Datacenter	✅	✅		
	SpotInst	✅	✅		
	Rancher	✅	🚧 Q1		
Template Library - Account	Shell Script	✅	✅	YES	
	HTTP Step	✅	✅	YES	
	CF Command	✅	✅		
	Service Command	✅	✅	YES	Service Commands in First Gen moved as first class Pipeline Steps in Next Gen
	Artifact Source	✅	✅	YES	
Service Deployment Type	Kubernetes	✅	✅	YES	The Service Deployment Type attribute will migrate with the Service from CG to NG (Kubernetes encompasses Kustomize, Openshift and Helm K8s)
	Kustomize	✅	✅	YES	The Service Deployment Type attribute will migrate with the Service from CG to NG (Kubernetes encompasses Kustomize, Openshift and Helm K8s)
	Openshift	✅	✅	YES	The Service Deployment Type attribute will migrate with the Service from CG to NG (Kubernetes encompasses Kustomize, Openshift and Helm K8s)
	Native Helm	✅	✅	YES	If a Native Helm Service is configured in CG the Upgraded service in NG will retain that property
	ECS	✅	✅	NO	
	Cloud Foundry	✅	✅	YES	
	AMI/ASG	✅	✅	YES	
	CodeDEploy	✅	✅	NO	• Will be solved via Deployment Template Offering
	SSH	✅	✅	YES	
	WinrRM	✅	✅	YES	
	Azure Web App	✅	✅	YES	
	Azure VMSS	✅	🚧	NO	
	SpotINst	✅	✅	YES	

Connectors	Git	✓	✓	YES	
	Github	✓	✓	YES	
	Gitlab	✓	✓	YES	
	Bitbucket	✓	✓	YES	
	JIRA	✓	✓	YES	
	ServiceNow	✓	✓	YES	
	Docker	✓	✓	YES	
	Artifactory - Docker, Non-Container	✓	✓	YES	
	Nexus - Docker, Non-Container	✓	✓	YES	
	Jenkins	✓	✓	YES	
	Bamboo	✓	✗		
	Helm HTTP	✓	✓	YES	
	Azure Artifacts	✓	✓		
	Custom Artifact Source	✓	✓	YES	
Verification Providers	Prometheus	✓	✓	YES	
	AppDynamics	✓	✓	YES	
	Datadog	✓	✓	YES	
	ELK	✓	✓	NO	
	Splunk	✓	✓	YES	
	Custom Metrics Provider	✓	✓	NO	
	NewRelic	✓	✓	YES	
	Cloudwatch	✓	✓	NO	
	Custom Log Provider	✓	✓	NO	
	Stackdriver	✓	✓	YES	
	Dynatrace	✓	✓	YES	
	SumoLogic	✓	✗	YES	
	Bugsnag	✓	✗	NO	
	Logz.io	✓	✗	NO	
	Instana	✓	✗	NO	
	Jira Integration	✓	✓	YES	
	Log Feedback	✓	✗	NO	
	Deeplinking to verification providers	✓	✗	NO	
Application	Application Defaults	✓	✓	YES	Application Defaults are going to become Project Variables

	Mandate Webhook Secrets for Github Triggers	✓	✓	NO	We will configure this in the Project Setting Section
	Template Library	✓	✓	YES	Template Library at the project scope
	Event Rules	✓	✗		
	Service	✓	✓	YES	
	Environment	✓	✓	YES	
	Workflow	✓	✗	YES	These will now become stages
	Pipeline	✓	✓	YES	
	Trigger	✓	✓	NO	We will not migrate the triggers users need to recreate these
	Infrastructure Provisioner	✓	✓	NO	Infrastructure Provisioners are now Steps of Infra provisioning
Template Library - Application	Shell Script	✓	✓	YES	
	HTTP Step	✓	✓	YES	
	CF Command	✓	✓		
	Service Command	✓	✓	YES	
	Artifact Source	✓	✓	YES	
Service	Artifact from Manifest	✓	✓	YES	
	Enable Kubernetes V2	✓	✓		K8s V2 is Default No More K8s V1
	Inline Manifests	✓	✓	YES	
	Remote Manifests	✓	✓	YES	
	Custom Remote Manifests	✓	✓	YES	
	Upload Inline Manifests	✓	✓	YES	
	Config Variables	✓	✓	YES	
	Config Files	✓	✓	YES	
	Values YAML Override	✓	✓		
	Artifact History	✓	✗		
Environment	Infra Definition	✓	✓	YES	
	Scope to Specific Service	✓	✗		
	Service Configuration Override - Variable	✓	✓		
	Service Configuration Override - Values. yaml	✓	✓		
	Service Configuration Override - Application Setting	✓	✓		

	Service Configuration Override - ConfigMap YAML	✅	❌		
	Service Configuration Override - Connection String (Inline)	✅	✅		
	Service Configuration Override - Connection String (Remote)	✅	✅		
	Service Configuration Override - ConfigMap YAML	✅	❌		
	Service Configuration Override - File Override	✅	✅		
	Service Configuration Override - Openshift Params	✅	✅		
	Service Configuration Override - TAS Manifest	✅	❌		
	Service Configuration Override - Helm Chart Repo	✅	❌		
Workflow	Basic	✅	✅	YES	1 Deployment Stage in a Pipeline that performs a basic deployment (SSH, WinRM, PCF, Azure VMSS, Azure Web APP, Helm) `${workflow.variables.[var]} <+pipeline.variables.[var]>`
	Canary	✅	✅	YES	1 Deployment Stage in a Pipeline that performs a Canary deployment `${workflow.variables.[var]} <+pipeline.variables.[var]>`
	Blue-Green	✅	✅	YES	1 Deployment Stage in a Pipeline that performs a Blue Green deployment `${workflow.variables.[var]} <+pipeline.variables.[var]>`
	Rolling	✅	✅	YES	1 Deployment Stage in a Pipeline that performs a Rolling deployment `${workflow.variables.[var]} <+pipeline.variables.[var]>`
	Multi Service	✅	✅	NO	We will have a first class Multi Service Deployment capability in NG that doesn't leverage the current gen's multi service behavior, Workflow type isn't widly used (low usage)
	Build	✅	✅	YES	We will create 1 Custom Stage in a Pipeline wtih the steps used from the Build Workflow Now Build Stages are replaced by A Custom Stage or a CIE Stage `${workflow.variables.[var]} <+pipeline.variables.[var]>`

Pipeline	Option to Skip Step - Do Not Skip	✅	✅	YES	We can configure these properties at the stage level
	Option to Skip Step - Always Skip	✅	✅	YES	• We can configure these properties at the stage level
	Option to Skip Step - Based on Assertion	✅	✅	YES	• We can configure these properties at the stage level
	Approval Step	✅	✅		
	Execution Step	✅	✅		
	Execute in Parallel with Previous Step	✅	✅	YES	
Trigger	On New Artifact	✅	✅	NO	Users will need to recreate their triggers in NG
	On Pipeline Completion	✅	✅	NO	We are replacing this with Pipeline Chaining
	On Time Schedule	✅	✅	NO	Users will need to recreate their triggers in NG
	On Webhook Event	✅	✅	NO	Users will need to recreate their triggers in NG
	On Manifest Changes	✅	✅	NO	Users will need to recreate their triggers in NG
	Last Collected	✅	✅	NO	Users will need to recreate their triggers in NG
	Last Successfully Deployed	✅	❌	NO	Users will need to recreate their triggers in NG
	From Triggering Source	✅	✅	NO	Users will need to recreate their triggers in NG
	Webhook Authorization	✅	✅	NO	
	Webhook Authentication	✅	🚧	NO	
Secrets Manager	AWS KMS	✅	✅	YES	
	AWS Secrets Manager	✅	✅	YES	
	Azure Key Vault	✅	✅	YES	
	Google KMS	✅	✅	YES	
	Google Secrets Manager	✅	🚧		
	HashiCorp Vault	✅	✅	YES	
	Custom Secrets Manager	✅	✅		
	Hashicorp Vault - SSH Secrets Engine	✅	❌		
Secrets	Encrypted Text	✅	✅		
	Encrypted Files	✅	✅		
	SSH	✅	✅		
	Winrm Credential	✅	✅		
	Direct access to secrets in YAML	✅	✅		
	Setup Usage	✅	✅		
	Runtime Usage	✅	✅		

User Management	Users - UI, API, SCIM and LDAP	✅	✅		
	User Groups - UI, API, SCIM and LDAP	✅	✅		
Access Control	RBAC on entities	✅	✅	NO	
	Scope to applications	✅	❌	NO	
	API Keys	✅	✅	NO	
	IP Whitelist Management	✅	✅	NO	
	Consolidated view of entities	✅	✅	NO	
Audit Trails	CRUD operations	✅	✅	NO	
	YAML Diff	✅	✅	NO	
Authentication Settings	OAuth Provider-Github	✅	✅	NO	
	OAuth Provider-Bitbucket	✅	✅	NO	
	OAuth Provider-Gitlab	✅	✅	NO	
	OAuth Provider-Linkedin	✅	✅	NO	
	OAuth Provider-Google	✅	✅	NO	
	OAuth Provider-Azure	✅	✅	NO	
	SAML based SSO	✅	✅	NO	
	LDAP based Auth	✅	✅	NO	
	LDAP - CRON expression for scheduling	✅	🚧	NO	
	Password Policy - Quality	✅	✅	NO	
	Password Policy - Expiration	✅	✅	NO	
	Password Policy - Lockout Policy	✅	✅	NO	
	Security Options - Enforce Two Factor Authentication	✅	✅	NO	
	Security Options - Restrict users to email domains	✅	✅	NO	
	Local Login	✅	✅	NO	
Kubernetes	Rolling Deployment Step	✅	✅		
	Blue Green Deployment	✅	✅		
	Canary Deployment	✅	✅		
	Traffic Shifting With Istio	✅	✅	NO	• We are Supporting with a Kubernetes Apply step with Service yaml providing traffic percentage no longer using istio integration. • It has 0 customer usage right now

	Traffic Shifting without Istio	✅	✅	YES	• We are Supporting with a Kubernetes Apply step with Service yaml providing traffic percentage
	Support Multiple Values.yaml for the Service	✅	✅	YES	• In the Service Configuration we can add multiple values.yaml files to the service
	Kubernetes Apply Step Apply From service configuration YAML	✅	✅	YES	• We will Migrate the step over as is
	Kubernetes Apply Step Apply Inline YAML	✅	✅	YES	
	Kubernetes Apply Step Apply Remote	✅	✅	YES	
	CRD Deployments	✅	✅	YES	• We can Deploy CRDs today with our regular K8s steps
	Rancher Deployment	✅	🚧 (Q1)		
	Kubernetes Dry-Run	✅	✅	NO	• We will configure this as a seperate step in NG and improve it so other users can benefit, righ now only 1 customer uses it
	Custom Remote Manifests	✅	✅	YES	
Native Helm	Version 2	✅	✅	YES	These configurations are defined in the Service
	Version 3	✅	✅	YES	These configurations are defined in the Service
	Helm Basic Deployment	✅	✅	YES	It will now become a Helm Rolling Deployment - We will migrate over the Helm Deploy Step
	Custom Remote Manifests	✅	✅	YES	
	Helm Command Flags	✅	✅	YES	These configurations are defined in the Service
Kustomize	Patches Support	✅	✅	YES	
	Rolling	✅	✅	YES	
	Canary	✅	✅	YES	
	Blue-Green	✅	✅	YES	
Openshift	Rolling	✅	✅	YES	
	Canary	✅	✅	YES	
	Blue-Green	✅	✅	YES	
	Custom Remote Manifests	✅	✅		
Amazon ECS	Rolling	✅	✅	NO	These capabilities are replaced by the new ECS Swimlane behavior
	Canary	✅	✅	NO	These capabilities are replaced by the new ECS Swimlane behavior
	Blue-Green	✅	✅	NO	These capabilities are replaced by the new ECS Swimlane behavior
	ECS Service Setup	✅	✅	NO	These capabilities are replaced by the new ECS Swimlane behavior
	ECS Steady State Check	✅	✅	NO	These capabilities are replaced by the new ECS Swimlane behavior

	Run Task	✅	✅	NO	These capabilities are replaced by the new ECS Swimlane behavior
	Upgrade Container Step	✅	✅	NO	These capabilities are replaced by the new ECS Swimlane behavior
	Visualization of Task Definition	✅	✅		This can be enhanced to view TaskDefinition with native experience from Harness File Store
Amazon AMI /ASG	Rolling	✅	✅	YES	
	Canary	✅	✅	YES	
	Blue-Green	✅	✅	YES	
Amazon Lambda	Basic	✅	🚧	NO	We will design a V2 of this swimlane
Amazon CodeDeploy	Basic	✅	❌	NO	AES CodeDeploy can be supported with Deployment Template
Cloud Foundry	Rolling	✅	✅	YES	
	Canary	✅	✅	YES	
	Blue-Green	✅	✅	YES	
	Custom Remote Manifest	✅	✅	YES	
	Charles Schwab - PCF Deployment Behaviors	✅	✅	YES	
Azure VMSS	Rolling	✅	🚧		
	Canary	✅	🚧		
	Blue-Green	✅	🚧		
Azure Web Apps	Rolling	✅	✅	YES	
	Canary	✅	✅	YES	
	Blue-Green	✅	✅	YES	
Deployment Templates	Deployment Template	✅	✅	NO	
Infrastructure Provisioners	Terraform	✅	✅	NO	We are moving away from the construct of infrastructure provisioners they are now just steps in a pipeline
	Terraform Apply	✅	✅	YES	The infra provisioner configuration will be mapped to the step, the terraform apply
	Terraform Provision	✅	❌	NO	Terraform Provision was very similar to Terraform Apply, we now removed the redundancy. Terraform Plan, Terraform Apply, Terraform Destroy. The infra provisioner configuration will be mapped to the steps
	CloudFormation	✅	✅	YES	We are moving away from the construct of infrastructure provisioners they are now just steps in a pipeline
	Cloudformation Create Stack	✅	✅	YES	
	CloudFormation Delete Stack	✅	✅	YES	
	Shell Script Provisioner	✅	✅	YES	We are moving away from the construct of infrastructure provisioners they are now just steps in a pipeline
	Terragrunt	✅	✅	YES	We are moving away from the construct of infrastructure provisioners they are now just steps in a pipeline

	Azure ARM/Blueprint	✅	✅	YES	We are moving away from the construct of infrastructure provisioners they are now just steps in a pipeline
Governance	Pipeline Governance v1	✅	✅	NO	We are not migrating the old PIpeline Governance experience over we have our new OPA based Pipeline Governance
	Deployment Freeze	✅	✅	NO	
Orchestration Behaviors	Multi Infrastructure Deployments	✅	✅	NO	Users will move to the new version of Multi Infrastructure deployments in NG, the configuration and desgin is differet
	Execution Inputs	✅	✅		
Dashboards	Custom Dashboards	✅	✅	YES	We will keep the existing CG data available as long as its within the retention period so the Custom Dashboards pointing to CG data will remain visible to the user
	Service Dashboard	✅	✅	NO	We are not migrating the Service Dashboard data from CG to NG, It will start fresh
	Environment Dashboard	✅	✅	NO	We are not migrating the Environment Dashboard data from CG to NG, It will start fresh
Steps	Barrier	✅	✅	YES	
	Email	✅	✅	YES	
	Shell Script	✅	✅	YES	Powershell support for script in Progress
	Approval - Harness	✅	✅	YES	
	Jira Create	✅	✅	YES	
	Jira UPDATE	✅	✅	YES	
	ServiceNow Create	✅	✅	YES	Support for Import sets in Progress
	ServiceNow Update	✅	✅	YES	
	Jenkins	✅	✅	YES	
	Bamboo	✅	❌		
	Google Cloud build	✅	❌		
	Resource Consraint	✅	✅	YES	
Stage: Approval	ServiceNow	✅	✅	YES	
	JIRA	✅	✅	YES	
	Harness	✅	✅	YES	
	Custom Approval	✅	✅	YES	
APIs	GraphQL API	✅	❌	REST API	We will not be migrating the customers API based tools, customers will need to rebuild around our REST API Harness is no longer exposing a GRaphQL API to our customers
	API Endpoints	✅	❌	REST API	Not all enpoints will be migrated over because the objects either don't exist in NG or the construct is different Users need to reqirte their automation to leverage the new endpoints. Users will need to communicate via REST
Git Experience	GitHub - Saas and Enterprise	✅	✅	NO	

	Bitbucket - SaaS and On-Prem	✅	✅	NO	
	Azure Repo	✅	✅	NO	
	Gitlab	✅	❌	NO	
	Generic Git Connector	✅	❌	NO	
	Git activity	✅	❌	NO	
Artifact Sources	Docker	✅	✅	YES	
	Jenkins	✅	✅	YES	
	Amazon S3	✅	✅	YES	
	Artifactory	✅	✅	YES	
	Nexus	✅	✅	YES	
	ECR	✅	✅	YES	
	Google Container Registry	✅	✅	YES	
	Custom Artifact Source	✅	✅	YES	
	Azure Artifacts	✅	✅	YES	
	Azure Container Registry	✅	✅	YES	
	Azure VMSS	✅	🚧		
	Github Package Registry	❌	✅		
	Google Artifact Registry	❌	✅	YES	
	GCS	✅	✅	YES	
	Amazon AMI	✅	✅	YES	
	Bamboo	✅	❌		
	SFTP	✅	❌		
	SMB	✅	❌		