**Title: GoCD Installation on devops-gocd Cluster**

**Revision History:** V1, Date: 20th Sep 2023

**Creator:** Pravat Kumar Sahu (Email: [psahu@atd-us.com](mailto:psahu@atd-us.com))

**Contains:**

1. Basic dev Cluster reference Environment
2. Pre-requisite
3. Architecture
4. GoCD Server Installation
5. GoCD agent configuration
6. GoCD agent configuration

-------------------------------------------------------------------------------------------------------------------------

1. Basic dev Cluster reference Environment:

* ATD GCP Project: atd-datamgt-npe
* Cluster: devops-gocd
* Control plane zones: us-east1-b
* GKE Control plane and Client version: 1.27.3-gke

1. Pre-requisite:

* Kubernetes CLI kubectl
* Configured gcloud SDK
* GKE cluster with separately managed node pools (Because this allows to customize GoCD GKE cluster based on GoCD server and Elastic agent pod resource requirement)
* Helm Client v3
* GKE node pool server account has roles and permissions to access Google Container Registry (GCR).

1. Architecture (reference from GoCD official diagram):

A diagram of a cluster of information

Description automatically generated

1. GoCD Server Installation:

* Download Helm Chart from <http://stash.prod.icd/projects/GOC/repos/gocd-pipeline/browse?at=GoCDHelm>
* (Optically you can git clone Helm Chart from <http://stash.prod.icd/scm/goc/gocd-pipeline.git> )
* Run the Install command.

Command: helm install gocd ./gocd --values ./gocd/values.yaml --namespace gocd --create-namespace

GoCD Helm Chart will automatically install GoCD Server, Elastic Scheduler, Agent POD, Kubernetes Cluster profile with plugins kubernetes-elastic-agents, docker-registry-artifact and okta-oauth-authorization.

* After few minutes of installation GoCD Server is available.

commands: kubectl get deployments --namespace gocd

* Get the GoCD server URL by running these commands:

(It may take a few minutes before the IP is available to access the GoCD server.)

Command: echo "GoCD server public IP: <http://$(kubectl> get ingress gocd-server --namespace=gocd -o jsonpath='{.status.loadBalancer.ingress[0].ip}')"

* Now it’s time setup GOCD server accessible by DNS with SSL certificate.

Setup on <https://devops.gcp.atd-us.com/go>

Update the DNS records to point to GCP resource by pointing to the IP address of Ingress controller and then apply Ingress resource.

Ensure that the GOCD server accessible using the domain name <https://devops.gcp.atd-us.com/go>

* Access and Authorize GoCD using ATD Okta account.

Go to GoCD admin >> security >> auth\_configs and Click Add for adding new Authorization Configuration.

Select Plugin as Okta oauth authorization plugin and give a name as ID.

Enter ATD Okta Endpoint, OAuth Client ID and Client Secret and save the configuration.

* Start login GoCD with your Okta account.

1. GoCD agent configuration

Note: Pre-exist and ready to use following agent profile are present on GCR Container Registry.

Python Profile:

gcr.io/atd-datamgt-npe/gocdagent-python:03

Java profile:

gcr.io/atd-datamgt-npe/gocdagent-java:02

GO Profile:

gcr.io/atd-datamgt-npe/gocdagent-go:02

Node profile:

gcr.io/atd-datamgt-npe/gocdagent-node:02

Docker profile:

gcr.io/atd-datamgt-npe/gocd-agent-docker-dind:v23.3.0

* Go to GoCD >> admin >> elastic\_agent\_configurations
* Add Elastic Agent Profiles under k8-cluster-profile.

Give appropriate Elastic Profile Name and Profile image name as an example gcr.io/atd-datamgt-npe/gocdagent-node:02

Specify Elastic Agent pod.yaml as below:

apiVersion: v1

kind: Pod

metadata:

name: gocd-agent-{{ POD\_POSTFIX }}

labels:

app: web

spec:

serviceAccountName: default

containers:

- name: gocd-agent-container-{{ CONTAINER\_POSTFIX }}

image: gcr.io/atd-datamgt-npe/gocdagent-node:02

securityContext:

privileged: true

* Save the Elastic profile Agent configuration.

1. GoCD agent configuration (Option steps how to use Elastic agent in GoCD Pipeline)

* Go to Pipeline JOBS SETTINGS and add the Elastic Agent Profile Id.