High-level design document

Predictive Cloud Capacity and Cost Management

August 10, 2022

Criticalriver technologies PVT ltd

Bangalore

Contents

1. [Introduction: 2](#_Toc111131871)

[1.1 Purpose: 2](#_Toc111131872)

[1.2 Scope: 2](#_Toc111131873)

[1.3 Document Overview: 2](#_Toc111131874)

1. [System High-Level Design Overview: 3](#_Toc111131875)

[2.1 System high-Level Design Goals, Scope, and Objectives: 3](#_Toc111131876)

[**2.1.1 Code check-out From Version Control Systems:** 3](#_Toc111131877)

[**2.1.2 Cloud Instance Startup Job:** 4](#_Toc111131878)

[**2.1.3 Creating K8s Cluster using Terraform:** 4](#_Toc111131879)

[**2.1.4 Restoring the Backup:** 4](#_Toc111131880)

[**2.1.5 Code Check-out to From Version Control Systems:** 5](#_Toc111131881)

[**2.1.6 Backup of the cluster:** 5](#_Toc111131882)

[**2.1.7 Destroying Cluster:** 5](#_Toc111131883)

[**2.1.8 Destroying VMs:** 5](#_Toc111131884)

1. [Detailed System Functionality Explanations: 5](#_Toc111131885)
2. [Deployment: 6](#_Toc111131886)

[4.1 Overview: 6](#_Toc111131887)

[4.2 Tools required: 6](#_Toc111131888)

# **Introduction:**

## 1.1 Purpose:

This document describes the architecture and implementation of Cloud Instances scheduler. It discusses the considerations that underpinned its design and how it is implemented**.**

## 1.2 Scope:

This document defines the high-level functionality of the scheduler i.e., scheduling of VMs and Kubernetes Clusters on various cloud providers. This document provides a simple, straightforward approach to reducing costs and improving business efficiency. This scheduler intends to do the following:

* + **Cloud Instances Scheduling:** Creating VMs using the terraform templates and scheduling the VMs with Cron Job triggered by Jenkins
  + **Kubernetes Cluster Scheduling:** Creating Cluster using terraform templates and taking backup and restoring of cluster using Velero which is a third-party open-source tool. And scheduling cluster with Cron Job Triggered by Jenkins

## 1.3 Document Overview:

The High-Level Design Document is arranged in the following format:

**Section 1: Introduction:**

A brief explanation of the purpose, goals, and format of this system Design Document.

**Section 2: System High-Level Design Overview:**

An Overview of the goals and objectives of the scheduler project. This section also provides a short explanation of each component and process to be implemented.

**Section 3: Detailed System Functionality Explanations:**

This section documents the detailed design of all Modules within the System.

**Section 4: Deployment:**

This section lists the options for deploying the system and the technical infrastructure where applications or processes should be set up to support the system.

# **System High-Level Design Overview:**

## 2.1 System high-Level Design Goals, Scope, and Objectives:

Critical Rivers as a part of the Digital Innovation center in DevSecOps is building a scheduler to help save cost when cloud instances are idle and not being used. Initially, the scheduler will schedule instances and Kubernetes cluster using the Cron job.

**The following diagram outlines the high-level design for a start-up job.**

Graphical user interface

Description automatically generated

Fig1: High-Level Design for startup

### **2.1.1 Code check-out From Version Control Systems:**

Jenkins will be triggered at the scheduled time as mentioned in the Cron job and will check out the code from the version control system like GitHub, Azure Repos, Bitbucket, etc.

### **2.1.2 Cloud Instance Startup Job:**

Once the code is checked out Jenkins will trigger the VM startup and shutdown pipeline job. With the help of a startup script, VMs and other resources will be up and running.

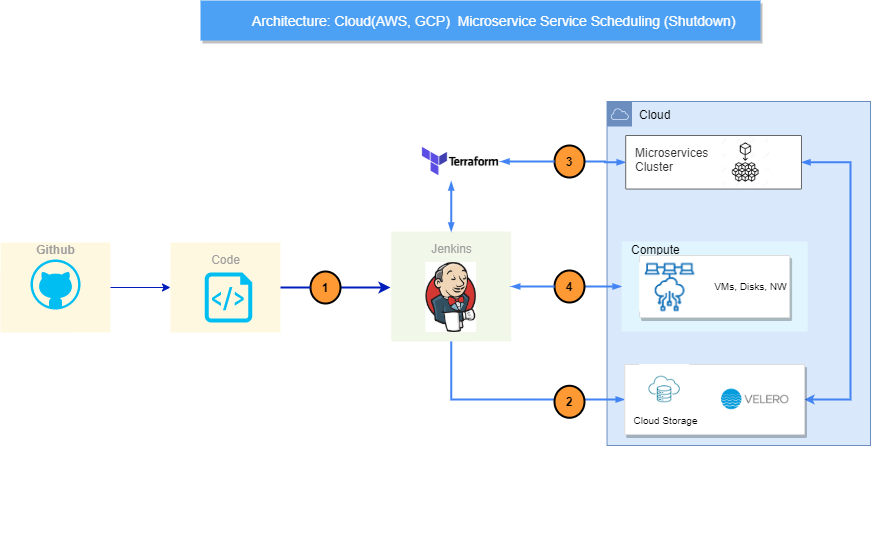
### **2.1.3 Creating K8s Cluster using Terraform:**

Once the VMs are up and running Jenkins will trigger the terraform pipeline job to create the cluster. With the help of Terraform templets, the Kubernetes cluster will be created.

### **2.1.4 Restoring the Backup:**

Once the cluster is created with Terraform templates it will start the Velero restore pipeline job. Velero will check for the latest backup in the cloud bucket and restore it.

**The following diagram outlines the high-level design for the Shut-down job.**

Fig2: High-Level Design for Shutdown

### **2.1.5 Code Check-out to From Version Control Systems:**

Jenkins will be triggered at the scheduled time as mentioned in the Cron Job and will pull the code from the version control system like GitHub, Azure Repos, Bitbucket, etc.

### **2.1.6 Backup of the cluster:**

Once the code is check-out in Jenkins it will start the Velero backup pipeline job. Velero will take the backup of the full cluster and store it in the cloud bucket.

### **2.1.7 Destroying Cluster:**

Once the backup is done Jenkins will trigger the terraform pipeline job to create the cluster. With the help of Terraform templates, the Kubernetes cluster will be destroyed.

### **2.1.8 Destroying VMs:**

Once the cluster is destroyed Jenkins will trigger the VM shutdown pipeline job to shut down the VM. With the help of the shutdown script, VMs and other resources will be shut down.

# **Detailed System Functionality Explanations:**

Cloud instances are always running unless customers specifically turn them off, which means customers are paying for computing time they don’t use. For this reason, the Digital Innovation Centre Team is designing and developing the scheduler to schedule the VMs and Kubernetes cluster that can be implemented on any cloud provider.

1. **Cloud Instances scheduling:**

This component is to be developed for scheduling the Cloud Instances so that Cloud Instances that are idle and not being used can be safely shut down. Also, it can be up during the productive hours of the day.

1. **Backup and Restore:**

This component is to be developed for taking backup and restoring for the Kubernetes cluster. This component will be responsible for the cluster backup and stored in the cloud bucket. And when the cluster is created back on working hours it can restore it from the cloud bucket.

1. **Kubernetes Cluster scheduling:**

This component Is to be developed for scheduling the Kubernetes cluster through terraform templets. It will be responsible for creating/destroying the Kubernetes cluster with the help of terraform script**.**

# **Deployment:**

This section identifies the different options for areas requiring configuration and implementation of the scheduling project.

## 4.1 Overview:

In this section, the deployment refers to the end-to-end flow of our project. Initially, when the Pipeline is triggered the required cloud resources (VMs & K8s Clusters) are created using Terraform templates. Later, the VMs which are scheduled will startup/shutdown using the respective scripts and Cron Job. Then, the Velero an open-source tool takes a backup of the K8s clusters and stores it in a cloud bucket. After taking backup the K8s Cluster will be destroyed again using Terraform templates.

**Timeline

Description automatically generated**

## 4.2 Tools required:

**Version control system:** GitHub **CICD Tool:** Jenkins **IaC Tool:**  Terraform **Disaster management tool:** Velero