**Digital Innovation Centre**

**Abstract**

Release Notes for the Software Product called Cloud Scheduler in the   
Cloud Environment

Release 1.0

**CLOUD SCHEDULER**

**CRITICALRIVER TECHNOLOGIES PVT LTD**

**Contents**

[**Release Overview:** 2](#_Toc111793874)

[**Audience:** 2](#_Toc111793875)

[**Document Overview:** 2](#_Toc111793876)

[**Prerequisites:** 2](#_Toc111793877)

[**Enhancement Details:** 3](#_Toc111793878)

[**Artifacts:** 3](#_Toc111793879)

[**Known Issues:** 3](#_Toc111793880)

[**Related Documents:** 4](#_Toc111793881)

# **Release Overview:**

This document provides a summary of information about the enhancements provided in the Cloud Scheduler 1.0 release. Cloud Scheduler provides a simple, straightforward approach to reduce cloud resources costs. With it, we can schedule non-production resources when idle to avoid incurring additional costs. We can also request on-demand resources.

# **Audience:**

This guide is intended for end-users, administrators, and technical personnel who uses Cloud Scheduler.

# **Document Overview:**

|  |  |
| --- | --- |
| TOPIC | DESCRIPTION |
| Prerequisites | Software requirements for the Cloud Scheduler to work |
| Enhancement Details | Provides a list of enhancements presented in this release, as well as impacted lines of business. |
| Artifacts | Describe the architecture, design, and function of the entire software development. |
| Related Documents | Business Use case, High-Level Design Diagram, and Documentation. |

# **Prerequisites:**

Following are the requirements for the Cloud Scheduler to work:

* Cloud Providers (AWS, GCP)
* Google Cloud CLI, AWS CLI
* Terraform
* Velero
* Version Control System (GitHub, Azure Repos)
* Jenkins

# **Enhancement Details:**

* Cloud Scheduler is designed to help save money when cloud resources are sitting idle and not being used.
* It helps to reduce monthly costs of cloud resources by 15% to 50%.
* It also has backup and restore mechanism for container-based resources.
* Through cron jobs it can automate the process of starting and stopping the resources on a given time.
* It can also start and stop the resources on demand as well.

# **Artifacts:**

Following are the artifacts as part of this release 1.0

* Jenkins Pipeline code for Cron Job

In the jenkinsfile of the cron job (autostsh.jenkins) we need to provide the following details.

Time Zone, Time of start-up and shutdown and variables parameter for Environments, mode, and Provider.

Below is an example:

Text

Description automatically generated

In this example timezone is set as TZ=Asia/Calcutta

For startup it will build the pipeline at 09:15 AM every day from Monday to Friday with parameter Environs=SBX;Mode=STARTUP;Provier=GCP.

For shutdown it will build the pipeline as 06:45 PM everyday from Monday to Friday with parameter Environs=SBX;Mode=STARTUP;Provier=GCP.

For cron job we can refer this website to convert time to cron [Crontab.guru - The cron schedule expression editor](https://crontab.guru/)

* Jenkins Pipeline code for VM Scheduling

In the jenkinsfile of the VM scheduling (vmstsh.jenkins) we need to provide the VM start and stop command of respective provider.

For Example:

Graphical user interface, text, application, email

Description automatically generated

In the example we are providing the gcloud command to start and stop the worker01 instances in zone=us-central1-a of project cr-test-356813.

* Jenkins Pipeline code for Terraform restore and destroy (tfcrdt.jenkins)
* Jenkins Pipeline code for Velero cluster backup and restore

In the Velero cluster backup and restore Jenkins file (vlrcbc.jenkins) we need to pass on the cluster name and bucket name

For Example:



In this command we are passing the cluster name as sbx-microservices-cluster and bucket name as wordpress.

* Terraform templates for creating K8s Cluster

In GCP for terraform we need to create a service account with the following permissions:

* 1. compute.admin
  2. iam.serviceAccountUser
  3. resourcemanager.projectIamAdmin
  4. container.clusterAdmin
  5. compute.viewer
  6. compute.securityAdmin
  7. container.developer
  8. iam.serviceAccountAdmin
  9. resourcemanager.projectIamAdmin
* Terraform templates for creating VMs
* K8s cluster backup scripts (Velero)

In GCP for velero we need to create a service account with the following permissions:

1. compute.disks.get
2. compute.disks.create
3. compute.disks.createSnapshot
4. compute.snapshots.get
5. compute.snapshots.create
6. compute.snapshots.useReadOnly
7. compute.snapshots.delete
8. compute.zones.get
9. storage.objects.create
10. storage.objects.delete
11. storage.objects.get
12. storage.objects.list

* K8s cluster restore scripts (Velero)

# **Related Documents:**

* High-Level Design Diagram
* Business Use Case
* Design Documentation
* Installation Guide