

# MiCollab Advanced Messaging 9.3 Containerized Deployment Guide

For version 9.3 and above

## Notice

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Networks™ Corporation (MITEL®). Mitel makes no warranty of any kind with regards to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes.

No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

## Trademarks

The trademarks, service marks, logos and graphics (collectively "Trademarks") appearing on Mitel's Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively "Mitel") or others. Use of the Trademarks is prohibited without the express consent from Mitel. Please contact our legal department at [legal@mitel.com](mailto:legal@mitel.com) for additional information. For a list of the worldwide Mitel Networks Corporation registered trademarks, please refer to the website: <http://www.mitel.com/trademarks>.

© Copyright 2022, Mitel Networks Corporation

All rights reserved

# Contents

<b>Preface</b>	<b>5</b>
References	5
Documentation	5
Documentation Updates	6
Help	6
Document Conventions	6
Frequently Used Terms	7
Acronyms and Abbreviations	8
<b>Overview</b>	<b>9</b>
Prerequisites	9
Architectural Diagram	10
<b>Usage of gcloud for kubectl access</b>	<b>11</b>
Download gcloud	11
Connect to the cluster	11
<b>Create a StorageClass for NTFS volume disk</b>	<b>12</b>
<b>Pushing Images to GCP Container Registry</b>	<b>13</b>
<b>Deploying Containers</b>	<b>14</b>
MiCollab AM System Server	14
MiCollab AM Call Server	14
Web Phone Manager	15
MiCollab AM Mobile Service	15
MiCollab AM Web Client	16
<b>Upgrading Containers</b>	<b>17</b>
MiCollab AM	17
Edit and apply manifest	17
Web Applications	17
Edit and apply manifest	17
<b>Tips and Concerns</b>	<b>19</b>
Useful Kubernetes commands	19

Remote access to the Windows node (VM) of the cluster	20
Deleting deployments	20

# Preface

## References

A catalog of technical documentation is included on the MiCollab AM Installation Media. If you are installing any advanced applications, such as Networking and Fax Server applications, you should refer to the appropriate technical documentation for application and installation information.

## Documentation

The technical documentation is produced in the PDF format and requires the PDF reader to view it. The MiCollab AM Documentation Library includes the following documents and resources:

- **Administration Documentation.** Available as a PDF only. Contains the following:
  - **Administration Guides.** Available as a PDF only. Contains administrative guides for administrators about how to manage and configure the messaging system.
  - **Quick Reference Cards (QRC).** Contains shortcuts and quick instructions telling subscribers how to access and use the messaging system.
  - **User Guides.** Available as a PDF only. Contains user guides for subscribers about accessing the messaging system and checking and sending messages.
- **Server Documentation.** Available as a PDF only. Contains the following:
  - **Developer Resources.** Contains programming guides and API references for developers for integrating the server clients and web applications with MiCollab AM.
  - **Installation and Configuration.** Available as a PDF only. Contains installation and configuration guides for server administrators about how to install and configure the messaging system.
  - **Integration Technical Notes (ITN).** Contains a set of guides that describe the integration methods and instructions for a variety of phone systems to work with MiCollab AM. The ITNs are generally used by resellers or administrators who are experienced with MiCollab AM and familiar with the integration procedures and terminology.
  - **Spare Parts Documentation.** Contains a set of guides that describe the instructions for installing and configuring hardware parts to work with MiCollab AM. These documents are written for Mitel-certified MiCollab AM technicians who are experienced with MiCollab AM and familiar with the procedures and terminology.
- **Software Release Notice (SRN).** This notice introduces the new features, capabilities, and hardware/software requirements for the corresponding MiCollab AM version.

## Documentation Updates

Documentation updates may be available from the following sources:

- Mitel-certified technicians can view or download documents and program files from our partner web site: [www.mitel.com](http://www.mitel.com)

## Help

The primary source of information about MiCollab AM is the online help available within any of its administrative utilities. You can access **Help** by clicking the **Help** button in the dialog box or window in which you are working.

## Document Conventions

The following conventions are used in this document:

- **Key Names.** Names of keys on the keyboard are shown in a box.

Example: **Enter**

When two keys must be pressed simultaneously, they are joined by a + sign.

Example: **Alt** + **Tab**

- **Reference to Document** Titles of other documents are shown in italics.

Example: See the *System Installation and Configuration Guide*.

- **User Interface (UI) Element Names.** Names of UI elements such as dialog boxes, windows, screens, menu items, tabs, buttons, and icons are shown in bold.

Example: On the **Startup** screen, click the **Start** icon.

- **User Input.** Information required to be typed is shown in italics.

Example: Type the password *voicemail*.

- **Warning, Caution, Important, and Notes.** Text for the contents that require attention are shown as follows:

**WARNING** A warning paragraph advises you of circumstances that can result in the loss of data, harm to the MiCollab AM System Server platform, or personal harm.

**CAUTION** Failure to follow these recommendations can result in unauthorized access to the system and consequent loss of data.

**IMPORTANT** An important paragraph gives decision-making information or informs you of the order in which tasks need to be completed.

**NOTE** A note gives additional information, provides an explanation, or indicates an exception to the information in the preceding text.

For more related documents, refer to the following list of references:

Table 1. References

Document Number	Document Title
	<i>Containerized MiCollab AM System Installation and Configuration Guide</i>

## Frequently Used Terms

Table 2. Frequently Used Terms

Term	Description
Container	Similar to a Virtual Machine, an isolated set of processes with a specific role.
Image	The template for a container.
System Server	Term refers to an organization's computer platform(s) that have MiCollab AM software installed and handles the core system functions such as storing messages, database. It can also refer generically to the System Server platform, the Call Server platform, or both. The term is most often used to describe a software or hardware installation or configuration practice where the role of the server platform is not specifically expressed.
Call Server	Term refers to an organization's computer platforms that have MiCollab AM software installed and serve as the interface to the system (PBX). The Call Server(s) interface with the System Server for the purpose of accessing messages, and database.

# Acronyms and Abbreviations

Table 3. References

Term	Description
GCP	Google Cloud Platform
GKE	Google Kubernetes Engine



# Overview

This guide describes how to deploy the containerized MiCollab AM in Google Cloud.

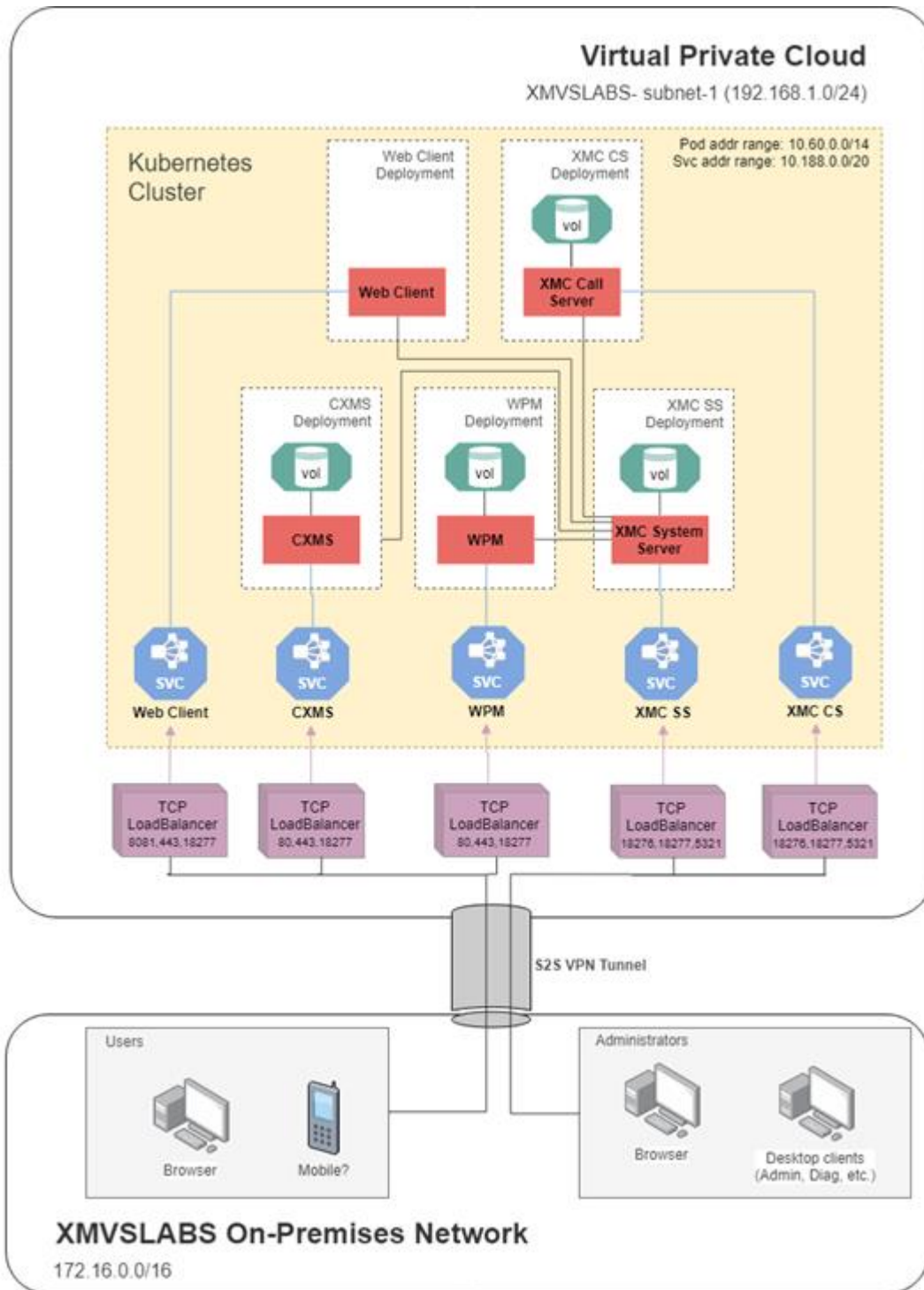
**Note** Upon completion of the steps in the document, the containers will be deployed but not configured. To configure the various containers, refer to *Containerized MiCollab AM System Installation and Configuration Guide*.

The following prerequisites are expected to be met.

## Prerequisites

- The administrator needs to have access to the yaml files this document is referencing.
- The administrator needs to have access to the registry images deployments need to use.
- The administrator needs to have a working knowledge of Kubernetes.
- The administrator needs to have a working knowledge of GCP and GKE.
- The administrator needs to know the name and have account access to the project on GCP where MiCollab AM is being deployed.
- A site-to-site (S2S) VPN needs to have been configured to connect GCP to the local network.

## Architectural Diagram



# Usage of gcloud for kubectl access

In order to run Kubernetes commands in Google's cloud use the following steps to download and connect using gcloud.

## Download gcloud

Follow this quickstart guide to install gcloud on your computer:

<https://cloud.google.com/sdk/docs/quickstart-windows>

## Connect to the cluster

After connecting to the project in your gcloud shell, to connect to the cluster:

- Go to the *Kubernetes clusters* page in GCP web console.
- Click the *Connect* button next to the next cluster.
- Use the code provided in the *Command-line access* box in your shell.
- Use a `kubectl` command to verify the cluster is connected.

# Create a StorageClass for NTFS volume disk

When a container is deployed, Kubernetes pod will request a PersistentVolume, which might trigger a dynamic Persistent Disk provisioning. The disks provisioned for MiCollab AM containers follow a custom StorageClass. If this StorageClass does not exist before pod deployment, it will not succeed. Therefore, this StorageClass must first be created by running the following command in the gcloud shell:

```
kubect1 apply -f ntfs-sc-gcp.yaml
```

# Pushing Images to GCP Container Registry

Images need to exist in the GCP Container Registry before being deployed. Push images to the GCP Container registry using the following steps.

Initialize gcloud to be authorized for Google-supported Docker registries

```
gcloud auth configure-docker
```

Tag the image with the GCP Container Registry location

```
docker tag [LOCAL-IMAGE-NAME]:[TAG] gcr.io/[PROJECT_ID]/[IMAGE-NAME]:[TAG]
```

Push the image to the GCP Container Registry

```
docker push gcr.io/[PROJECT_ID]/[IMAGE-NAME]:[TAG]
```

**Note** Go to the *Container Registry* page in GCP UI console to see your repository.

# Deploying Containers

Containerized MiCollab AM deployments include the following containers:

- System Server
- Call Server
- MiCollab AM Web Client
- Web Phone Manager
- MiCollab AM Mobile Service

The typical process for deployment is:

1. Create the service.
2. Apply the deployment, which will carry out volume operations, then deploy pod(s) and mount specified volumes.

**Note** Be sure to deploy the MiCollab AM System Server before any other containers.

## MiCollab AM System Server

Run the following commands:

Create a LoadBalancer service

```
kubectl apply -f xmc-ss-service-gcp.yaml
```

Apply the deployment

```
kubectl apply -f xmc-ss-deployment-gcp.yaml
```

Verify the deployed pod is "Running"

```
kubectl get pods
```

## MiCollab AM Call Server

**Note** Each Call Server has a unique yaml file. The following refers to the first Call Server.

Run the following commands:

Create a LoadBalancer service

```
kubectl apply -f xmc-cs1-service-gcp.yaml
```

Apply the deployment

```
kubectl apply -f xmc-cs1-deployment-gcp.yaml
```

Verify the deployed pod is "Running"

```
kubectl get pods
```

## Web Phone Manager

Run the following commands:

Create a LoadBalancer service

```
kubectl apply -f wpm-service-gcp.yaml
```

Apply the deployment

```
kubectl apply -f wpm-deployment-gcp.yaml
```

Get endpoint IP of the application for subsequent configuration

```
kubectl get svc wpm-service
```

**Note** You can also get pod and service information in the *Workloads* and *Services & Ingress* pages under the *Kubernetes Engine* section in GCP UI console.

## MiCollab AM Mobile Service

Run the following commands:

Create a LoadBalancer service

```
kubectl apply -f cxms-service-gcp.yaml
```

Apply the deployment

```
kubectl apply -f cxms-deployment-gcp.yaml
```

Get endpoint IP of the application for subsequent configuration

```
kubectl get svc cxms-service
```

**Note** You can also get pod and service information in the *Workloads* and *Services & Ingress* pages under the *Kubernetes Engine* section in GCP UI console.

## MiCollab AM Web Client

Run the following commands:

Create a LoadBalancer service

```
kubectl apply -f web-client-service-gcp.yaml
```

Apply the deployment

```
kubectl apply -f web-client-deployment-gcp.yaml
```

Get endpoint IP of the application for subsequent configuration

```
kubectl get svc web-client-service
```

**Note** You can also get pod and service information in the *Workloads* and *Services & Ingress* pages under the *Kubernetes Engine* section in GCP UI console.



# Upgrading Containers

Before upgrading the containers, the newer images first need to be pushed to the GCP Container Registry. See [Pushing Images to GCP Container Registry](#).

## MiCollab AM

The upgrade steps for System Server and Call Server are identical. Be sure to upgrade System Server **before** upgrading any Call Server.

**Note** Before upgrade, it is always recommended to run daily maintenance on the System Server and each of the Call Servers to create a recent backup of the data.

## Edit and apply manifest

**WARNING** Before upgrading MiCollab AM, stop MiCollab AM using either System Status or Line Status, to ensure there are no active calls that would be dropped.

Open `xmc-ss-deployment-gcp.yaml` and `xmc-cs1-deployment-gcp.yaml`, find the image property under Deployment. Modify the value with the new image name to upgrade to.

Apply the manifest as usual:

```
kubectl apply -f xmc-ss-deployment-gcp.yaml
OR
kubectl apply -f xmc-cs1-deployment-gcp.yaml
```

**Note** Anytime a new image is being deployed, it must first be pulled from gcr.io to the host node, which can be very time consuming. While this is happening, the original container will remain in a Running state, while the new container will have a ContainerCreating state. Using `kubectl describe pod [POD-NAME]` will indicate that the image is being pulled during this period.

## Web Applications

### Edit and apply manifest

Open the appropriate file:

- `wpm-deployment-gcp.yaml`

- cxms-deployment-gcp.yaml
- web-client-deployment-gcp.yaml

Find the image property under Deployment. Modify the value with the new image name to upgrade to.

Apply the above manifest file as usual:

```
kubectl apply -f [FILE-NAME].yaml
```

## Tips and Concerns

### Useful Kubernetes commands

**Note** Common resource types are - pod, service/svc, deployment/deploy, storageclass/sc, persistentvolume/pv, persistentvolumeclaim/pvc

Get help on a specific command

```
kubectl [COMMAND] --help
```

Get all objects of a resource type in the default namespace

Include `-o wide` flag if you want to get pods' IPs

```
kubectl get [RESOURCE-TYPE]
```

Get all objects of a resource type in a specific namespace

```
kubectl get [RESOURCE-TYPE] -n [NAMESPACE]
```

Two ways to get details of a specific object in the default namespace

The *describe* command includes events list, which is useful for debugging

```
kubectl get [RESOURCE-TYPE] [OBJECT-NAME] -o yaml  
kubectl describe [RESOURCE-TYPE] [OBJECT-NAME]
```

To connect to a PowerShell session within a container

```
kubectl get pods -o wide  
kubectl exec -it [POD-NAME] -- powershell
```

Delete an object of a resource type in the default namespace

```
kubectl delete [RESOURCE-TYPE] [OBJECT-NAME]
```

Force delete an object

```
kubectl delete [RESOURCE-TYPE] [OBJECT-NAME] --grace-period=0 --force=true
```

## Remote access to the Windows node (VM) of the cluster

In some cases, we would want to remotely access the Windows node of the cluster. For example:

- Test readinessProbe/livenessProbe manually by using curl to our pods from their host.
- Test network connection to and from this cluster node.

In that case:

- Go to the *Compute Engine/VM instances* page in GCP web console.
- Find the VM instance with the name *gke-[cluster-name]-[node-pool-name]-....*
- Click on the down arrow next to the *RDP* button, select *Set Windows password*. Use any username. Copy the password.
- Remote into the cluster node VM using its **Internal IP** shown in the GCP web console. Use a machine in the local network which connects to GCP via the VPN tunnel.
- Use the username and password given earlier.

## Deleting deployments

Deleting a deployment will also delete the volume claims, so this is highly discouraged unless you know how to re-connect a new claim to the original volume.