DevOps Release Notes

/ ForgeRock Identity Platform 6.5

Latest update: 6.5.2

David Goldsmith
Shankar Raman

ForgeRock AS.
201 Mission St., Suite 2900
San Francisco, CA 94105, USA
+1 415-599-1100 (US)
www.forgerock.com

Abstract

Information for deploying ForgeRock Identity Platform™ version 6.5 on Kubernetes. Includes late-breaking news about features, known issues, and using the `forgeops` repository.

Table of Contents

Preface

Read these release notes before you deploy ForgeRock software on Kubernetes or update your existing deployment.

These release notes cover the prerequisites for deployment, known issues and improvements, changes and deprecated functionality, and other important information.

Before You Begin

Before deploying the ForgeRock Identity Platform on Kubernetes, read the important information in Start Here.

About ForgeRock Identity Platform Software

ForgeRock Identity Platform™ serves as the basis for our simple and comprehensive Identity and Access Management solution. We help our customers deepen their relationships with their customers, and improve the productivity and connectivity of their employees and partners. For more information about ForgeRock and about the platform, see https://www.forgerock.com.

The platform includes the following components:

- ForgeRock® Access Management (AM)
- ForgeRock® Identity Management (IDM)
- ForgeRock® Directory Services (DS)
- ForgeRock® Identity Gateway (IG)

Chapter 1

What's New

The February, 2020 revision of the `forgeops` repository is a major revision that introduces new features, functional enhancements, and fixes.

A Completely New Approach

With this February, 2020 revision, the `forgeops` repository:

- Includes Kustomize and Skaffold example artifacts for building Docker images and deploying them in a Kubernetes cluster. Helm charts are no longer used in deployment examples.
- Includes Pulumi scripts as examples of Kubernetes cluster creation. The example bash scripts for cluster creation have been removed from the repository.
- Changes the way that AM, IDM, and IG configuration is stored and managed. Instead of relying on the external `forgeops-init` repository, configuration is now stored as part of the the Docker images for AM, IDM, and IG.

These Release Notes contain detailed information about these major changes, and about other important changes.

The `forgeops` repository no longer designates release branches. You now deploy using the latest development versions of artifacts, available in the `master` branch.

The `forgeops` repository's `release/6.5.2` branch, which contains artifacts for deploying ForgeRock's DevOps Examples and CDM using Helm charts, is deprecated. This branch has not yet been removed from the `forgeops` repository; however, using Helm charts to deploy the ForgeRock Identity Platform is now deprecated, and Helm charts will eventually be removed from the repository. ForgeRock strongly recommends that you start deploying the platform with Kustomize and Skaffold as soon as possible.

The previous version of the DevOps documentation, which described deployment using Helm charts, has been removed from ForgeRock BackStage. Should you need to refer to this documentation for existing deployments before you migrate to Kustomize and Skaffold, you can still find it in the `forgeops` repository's `legacy-docs` directory.

New Features

This section covers new features in, and major improvements to the open source `forgeops` repository, in the February, 2020 revision:

Docker images include the AM, IDM, and IG configuration

In this revision, the AM, IDM, and IG configurations are incorporated into the `am`, `idm`, and `ig` Docker images.

This change improves ForgeRock Identity Platform startup times. It also eliminates the startup dependency on the availability of an external Git repository.

Configurations for AM, IDM, and IG now reside in the `forgeops` repository's `config` directory. Before building a customized Docker image for the ForgeRock Identity Platform, you run the new **config.sh** script. This script copies a configuration to a staging area in the `forgeops` repository's `docker` directory.

For information about customizing Docker images for the ForgeRock Identity Platform, see:

- "*Developing Custom Docker Images for the Platform*" in the *DevOps Developer's Guide: Using Minikube*
- "*Developing Custom Docker Images for the Platform*" in the *DevOps Developer's Guide: Using a Shared Cluster*

Skaffold framework support

The `forgeops` repository contains new artifacts that let you deploy the ForgeRock Identity Platform using the Skaffold framework. Deploying with Skaffold lets you:

- Quickly and easily start the ForgeRock Identity Platform.
- Modify the AM, IDM, and IG configurations.
- Build updated Docker images that include your configuration changes.
- Restart the ForgeRock Identity Platform with the updated Docker images.

Before you can use Skaffold with ForgeRock Identity Platform, you'll need to install Skaffold software on your local computer. See the *DevOps Developer's Guides* for more information.

Kustomize framework support

This revision uses the Kustomize framework to orchestrate AM, DS, IDM, and IG on Kubernetes. You no longer use Helm charts to orchestrate the ForgeRock Identity Platform.

Before you can use the Kustomize framework with ForgeRock Identity Platform, you'll need to install Kustomize software on your local computer. See the *DevOps Developer's Guides* for more information.

The ForgeRock Cloud Developer's Kit

The ForgeRock Identity Platform documentation introduces the term *Cloud Developer's Kit* to describe what was previously referred to as the DevOps Examples.

For more information about the Cloud Developer's Kit, see:

- "Configure the Platform" in the *Start Here* Guide
- "About the Cloud Developer's Kit" in the *DevOps Developer's Guide: Using Minikube*

Identical configurations for the CDK and the CDM

The CDK and the CDM now use uniformly comprehensive AM, IDM, and IG configurations. Examples in the documentation now illustrate full-featured configurations, and are no longer based on minimally viable configurations. See Configuration in the `forgeops` repository's top-level README file for more information about the configurations.

Prior to this revision, different configurations were used for CDK and CDM deployments. The DevOps Examples used minimal configurations for AM, IDM, and IG, while the CDM used a more full-featured configuration.

Pulumi scripts for CDM cluster creation

This revision uses Pulumi scripts to create clusters for CDM deployments.

For information about how to create Kubernetes clusters for the CDM using Pulumi, see the *Creating and Setting up a Kubernetes Cluster* sections in the CDM Cookbooks.

The previous version used a set of bash scripts for cluster creation. These scripts have been removed from the `forgeops` repository.

Secrets generator

The ForgeRock secrets generator randomly generates all secrets for AM, IDM, and DS services running in the CDK and the CDM. Random secrets generation greatly improves security for CDK and CDM deployments from previous versions.

The secrets generator runs as a Kubernetes job before AM, IDM, and DS are deployed.

See the `forgeops-secrets` README file for more information about the secrets generator, including a list of which secrets it generates.

Completely revised AKS Cookbook

Because of the previous lack of support in AKS for multiple availability zones for AKS clusters, ForgeRock formerly recommended against deploying the platform on Azure in production. With support for zones available in AKS, Azure is now a supported platform for production deployments, and the Cloud Deployment Model Cookbook for AKS is no longer designated "evaluation-only."

Changes from the evaluation-only version of the Cookbook include:

- The CDM deployment topology on Azure now matches the CDM deployment topology on GCP and AWS.

- Pulumi scripts demonstrate AKS cluster creation.
- Benchmark results are available for a sample deployment with 10,000,000 users.

Chapter 2
Upgrading

Before moving from Helm-based deployments to deploying based on Skaffold and Kustomize, read "*What's New*" and "*Changes and Deprecated Functionality*" for information about new, changed, and removed features.

Then perform the following general steps:

1. Update to newer versions of third-party software as follows:
   - If you're deploying the CDK in Minikube, go here.
   - If you're deploying the CDK in a namespace in a shared GKE cluster, go here.
   - If you're deploying the CDK in a namespace in a shared EKS cluster, go here.
   - If you're deploying the CDK in a namespace in a shared AKS cluster, go here.
2. Delete existing clones of the `forgeops` repository, and then reclone the `forgeops` repository.
3. Create a new Kubernetes namespace.
4. Copy your AM, IDM, and IG configurations into your `forgeops` repository clone:
   a. Create a new directory; for example `my-config`, under `/path/to/forgeops/config/6.5`.
   b. Copy your current AM configuration from your `forgeops-init` repository to `/path/to/forgeops/config/6.5/my-config/amster`.
   c. Copy your current IDM configuration from your `forgeops-init` repository to `/path/to/forgeops/config/6.5/my-config/idm`.
   d. Copy your current IG configuration from your `forgeops-init` repository to `/path/to/forgeops/config/6.5/my-config/ig`.
5. Modify ForgeRock's Kustomize files as needed.

   If you've used ForgeRock's default Helm charts without modfying them, you should be able to use the default Kustomize bases and overlays in the `forgeops` repository.
6. Revise any code to customize the AM web container.

   Version 6.5 and previous versions came with the **customize-am.sh** script. You could use this script to customize the AM web container.

The **customize-am.sh** script is no longer available in this revision of the `forgeops` repository.

To customize the AM web container in this revision, add instructions to the `am` Dockerfile to copy your customizations into the `/usr/local/tomcat/webapps/am` directory.

Chapter 3

Changes and Deprecated Functionality

This chapter covers changed, deprecated, and removed features in the February, 2020 revision of the `forgeops` repository.

Changes to Existing Functionality

The following features have been changed in the February, 2020 revision of the `forgeops` repository:

Deployment with Skaffold and Kustomize instead of Helm

This revision uses Skaffold and Kustomize, instead of using Helm charts, to deploy the platform.

Skaffold can detect changes to the file system that holds the AM, IDM, and IG configurations. When it detects a change to one of those configurations, it rebuilds the `am`, `idm`, or `ig` Docker image. Then, it reorchestrates the ForgeRock Identity Platform deployment.

Note that changes to dynamic AM configuration data—policies and application data—are *not* automatically detected by Skaffold. Changes to dynamic AM configuration data still need to be exported using Amster.

For more information about customizing the ForgeRock Identity Platform configuration when working with Skaffold, see *Developing Custom Docker Images for the Platform* in the DevOps Developer Guides.

Changes to CDM zones and node pools

In the new revision, CDM deployment have three availability zones and two node pools.

In the previous version, CDM used two zones and a single node pool.

See the CDM architecture diagrams in the *CDM Overview* sections in the CDM Cookbooks for more information.

New scripts for installing third-party components

This revision includes improved bash scripts for installing the NGINX ingress controller, Certificate Manager, Prometheus, Grafana, and Alertmanager in a CDM cluster.

The new scripts are ingress-controller-deploy.sh, certmanager-deploy.sh, and prometheus-deploy.sh,

Helm tiller pod no longer required

Although the CDM still uses Helm charts to install the NGINX ingress controller and Prometheus, a Helm tiller pod is no longer needed in the CDM cluster.

In the previous version, CDM deployment required a running tiller pod to support Helm chart deployment.

Revised benchmarking technique

This revision uses Gradle to trigger AM and IDM simulations for benchmarking performance.

For more information, see *Benchmarking CDM Performance* in the CDM Cookbooks.

Revised backup technique

In this revision, backup is greatly simplified. Backups are made to local disks running in the same pods in which DS runs.

The previous version required an NFS-mounted external storage device (Google Filestore or EFS) to be available for backup. The external storage device is no longer needed.

For more information, see *Backing up and Restoring Directory Data* in the Cloud Deployment Guide.

Modified DS topology in the CDM

This revision's DS topology:

- Two DS services are used: the CTS and ID Repo services. CTS directories hold CTS tokens. ID Repo directories hold identities, configuration data, policies, application data, and IDM run-time data.
- Three replicas of each service are deployed.

The previous version's DS topology:

- Three DS services were used: CTS, AM userstore, and AM configuration store.
- Two replicas of each service were deployed.

For more information, see the CDM architecture diagrams in the *CDM Overview* sections in the CDM Cookbooks for more information.

IG not deployed by default

In this revision, IG is not deployed as part of the CDK or CDM. Benchmarks for IG are no longer available in the *CDM Cookbooks*.

You can still deploy IG with the CDK and the CDM by using the Kustomize base and overlays in the `/path/to/forgeops/kustomize/ig` directory.

CDM sizing and benchmarks

The CDM Cookbooks provide the steps for creating medium-sized (10,000,000 users) clusters.

You can still create small-sized (1,000,000 users) and large-sized (100,000,000) clusters using the artifacts in the `forgeops` repository.

Benchmarks for small, medium, and large clusters are available for Google GKE. Benchmarks for medium clusters only are available for Amazon EKS and Microsoft Azure AKS.

Randomly generated administrator passwords

The CDM and CDK use administrator passwords that are randomly generated by the secrets generator.

See "*Using the Platform*" in the *DevOps Developer's Guide: Using Minikube* for information about how to obtain the administrator passwords. Note that the technique to obtain the passwords is the same for the CDK and the CDM.

New Docker image and pod names

ForgeRock's Docker image repository names are now `am`, `idm`, and `ig`. In previous versions, the Docker image repository names were `openam`, `openidm`, and `openig`.

Kubernetes pod names now include the strings `am`, `idm`, and `ig`. In previous versions, the pod names included the strings `openam`, `openidm`, and `openig`.

New method for building base Docker images

As with previous versions, you must still build your own base Docker images for the ForgeRock Identity Platform in production on Kubernetes.

In this version, you must download the ForgeRock binaries manually before building the Docker images.

In the previous version, a script automatically downloaded the binaries from ForgeRock's Artifactory repository. This script has been removed from the `forgeops` repository.

For more information, see *Building Custom Docker Images* in the Cloud Deployment Guide.

customize-am.sh script removed

The **customize-am.sh** script is no longer available in this revision of the `forgeops` repository.

To customize the AM web container in this revision, add instructions to the `am` Dockerfile to copy your customizations into the `/usr/local/tomcat/webapps/am` directory.

New backup-loader.sh script

The new **backup-loader.sh** script lets you create PVCs from DS binary backups before you start the platform, so that DS instances in the platform use the data from the PVCs.

Different default URLs

Use the following default URLs to access ForgeRock Identity Platform services in this revision:

- AM: https://*namespace*.iam.*domain*/am
- IDM: https://*namespace*.iam.*domain*/admin
- IG: https://*namespace*.iam.*domain*/ig

Support for newer versions of CDM third-party software

The CDM includes more recent versions of these third-party components.

See these scripts for details about versions of third-party software currently used with the CDM: ingress-controller-deploy.sh, certmanager-deploy.sh, and prometheus-deploy.sh,

Certificate Manager no longer required for the CDK on Minikube

Support for self-signed certificates and signing certificates is built into the CDK when it runs on Minikube. Because of this, you no longer need to deploy Certificate Manager when deploying the CDK on Minikube.

Self-signed certificates for GKE CDM deployments

CDM deployments use Certificate Manager for SSL support. In previous versions, Certificate Manager was configured to call Let's Encrypt to provide certificates for CDM deployments on GKE.

In this revision, Certificate Manager is configured to provide a self-signed certificate for CDM deployments on GKE.

***DevOps Developer's Guide* replaced**

The *DevOps Developer's Guide* has been replaced with two new guides:

- DevOps Developer's Guide: Using Minikube
- DevOps Developer's Guide: Using a Shared Cluster

The content in the new guides is similar to the *DevOps Developer's Guide*. Each of the new guides limits its descriptions to a single type of cluster, thus simplifying procedures.

***Before You Deploy* section moved**

The information formerly in the *Before You Deploy* chapter of the *Release Notes* has been moved. This information is now available where it's needed instead of on linked pages.

Site Reliability Guide renamed to Cloud Deployment Guide

The *Site Reliability Guide* has been renamed to the *Cloud Deployment Guide*. The new title more accurately reflects the guide's content.

DevOps QuickStart Guide removed

The *DevOps QuickStart Guide* tutorial has been removed from the documentation.

To get the ForgeRock Identity Platform up and running quickly on Kubernetes, see DevOps Developer's Guide: Using Minikube.

CDM and CDK installation requires Linux or macOS

ForgeRock supports CDM and CDK installation on Linux and macOS only. If you use a Microsoft Windows computer, you'll need to create a Linux virtual machine for installing the CDM and the CDK.

Deprecated Features

The following feature is deprecated in the February, 2020 revision of the `forgeops` repository:

Helm charts

Using Helm charts to orchestrate the ForgeRock Identity Platform in a Kubernetes cluster is deprecated. You should migrate to the Kustomize framework as soon as possible.

The deprecated Helm charts currently remain in the `forgeops` repository, but will be removed in the near future.

Removed Features

The following features have been removed from the February, 2020 revision of the `forgeops` repository:

`forgeops-init` repository

The `forgeops-init` repository is no longer available on GitHub. Previously, you used this repository as a starting point for creating a custom configuration repository.

bash scripts for CDM cluster creation

The bash scripts used for CDM cluster creation have been removed from the `forgeops` repository.

The current revision uses Pulumi software to create clusters for CDM deployments instead of using bash scripts.

Downloader utility

In the previous version of the `forgeops` repository, you obtained binary images of ForgeRock Identity Platform software from Artifactory using the Downloader utility. The binary images were required for building Docker images for the ForgeRock Identity Platform.

The Downlader utility is no longer available in the `forgeops` repository. Instead, download ForgeRock Identity Platform software from backstage.forgerock.com before building base images.

For more information, see *Building Custom Docker Images* in the Cloud Deployment Guide.

Dockerfile source code moved

Dockerfiles for ForgeRock Identity Platform components have been moved to the `/path/to/forgeops/docker/6.5` directory. Subdirectories with the suffix `-base` now contain the Dockerfile source code.

customize-am.sh script

In previous versions of the `forgeops` repository, the **customize-am.sh** script provided a means of customizing the AM web container before AM started.

This script has been removed from the `forgeops` repository. To customize the AM web container in this revision, add instructions to the `am` Dockerfile to copy your customizations into the `/usr/local/tomcat/webapps/am` directory.

Chapter 4
Limitations

DS Limitations

DS live data and logs should reside on fast disks.

DS data requires high performance, low latency disk. Use external volumes on solid-state drives (SSDs) for directory data when running in production. Do not use network file systems such as NFS.

DS does not scale elastically.

DS does not support elastic scaling. Be sure to design your DS deployment architecture carefully, with this limitation in mind.

The dsreplication command cannot run in a container.

The **dsreplication** command does not support configuration expressions, which are used by artifacts in the `forgeops` repository. Therefore, do not execute the **dsreplication** command in a Kubernetes pod. For more information, see Setting Up Replication in the *ForgeRock Directory Services Deployment Guide*.

As a result, you cannot use the **dsreplication status** command to obtain diagnostic information about replication when running DS in a Kubernetes pod. Instead, use one of the following techniques:

- **Monitor replication using Prometheus**. Grafana charts display the number of replicated and unreplicated updates, and the replication delay for each replica.
- **Query the `cn=monitor` entry**. For more information, see LDAP-Based Monitoring in the *ForgeRock Directory Services Administration Guide*.

The DS restore command must be used if you need to recover DS directory data.

You cannot use the **dsreplication** command with a DS instance in a replication topology, because the command cannot run using configuration expressions for property value substitution. To restore directory data, use the DS **restore** command. For information about restoring directory data from a backup in a Kubernetes deployment, see "Initializing Directory Data From Binary Backup" in the *Cloud Deployment Guide*.

Fully test your backup and recovery procedures before deploying DS in Kubernetes in production. Failure to do so might result in data loss.

Caution

Deploy DS as Kubernetes stateful sets only if:

- You are highly experienced and extremely skilled in Kubernetes deployment. DS is a specialized distributed database. Deploying DS on Kubernetes requires a deep knowledge of both DS and Kubernetes.
- You plan to deploy DS following the ForgeRock documentation. Any deviation from our prescriptive documentation on deploying DS on Kubernetes could cause instability in the deployment, and might impair ForgeRock's ability to support you.
- You understand that the current pattern for deploying DS, unlike other ForgeRock components, is not elastic.
- You have read the limitations in this section and understand you might need to work around them.

Unless you have experience deploying both DS and Kubernetes stateful sets in production, ForgeRock recommends that you *not* deploy DS using Kubernetes stateful sets in production deployments. Instead, deploy DS in virtual machines on cloud platforms to support ForgeRock Identity Platform deployments with AM and IDM running in Kubernetes.

We strongly recommend that you review the following with a ForgeRock technical consultant or a ForgeRock certified partner before deploying DS containers in production:

- Your overall requirements.
- Your DS services design.
- Your strategy for testing functionality and performance.

AM Limitations

Several AM operations are stateful and require session stickiness.

Several operations in AM are stateful, requiring flows to return to the same server instance several times. For example, browser-based authentication that uses authentication chains, and some SAML flows, are stateful operations. If your deployment uses any stateful AM operations, you *must* configure your load balancer to use sticky sessions.

Even if your deployment does not use any stateful AM operations, ForgeRock recommends that you configure your load balancer to use sticky sessions to achieve better performance.

A subset of AM's full SAML v2.0 functionality does not work correctly in containers.

When implementing AM SAML v2.0 in containers:

- Enable session stickiness on the ingress controller.
- For SAML v2.0 single sign-on with the HTTP-Artifact binding, use SAML v2.0 failover. For more information, see Configuring Providers for Failover in the *Access Management SAML v2.0 Guide*.

- For SAML v2.0 single logout, use the HTTP-POST or HTTP-Redirect bindings. The SOAP binding is *not* supported when AM runs in a container.

IDM Limitations

There are no limitations for this release.

IG Limitations

There are no limitations for this release.

DevOps Limitations

Docker images are not available for use in production deployments.

Docker images for use in production deployments of the ForgeRock Identity Platform are not available. Unsupported, evaluation-only images are available in ForgeRock's public Docker registry. These images can be used *for evaluation purposes only*.

When deploying ForgeRock Identity Platform in production, you must build Docker images. For more information about building images for the ForgeRock Identity Platform, see *Building Custom Docker Images* in the Cloud Deployment Guides.

Docker images with the ssoadm command are not available.

The CDK and the CDM do not include example deployments of the AM **ssoadm** command. However, you can use the AM REST API and the **amster** command with the AM and DS deployment example.

The IDM repository is not configured for high availability.

The IDM repository configuration used with the CDK and the CDM is not suitable for production deployments. When running IDM in production, configure your repository for high availability. For more information about ensuring high availability of the identity management service, see Clustering, Failover, and Availability in the *ForgeRock Identity Management Integrator's Guide*.

Chapter 5
Documentation Updates

The following changes have been made to the documentation since the February, 2020 revision of the CDK and the CDM:

Date	Description
2020-08-26	Added step 4.d in "To Create a Kubernetes Cluster for CDM" in the *Cloud Deployment Model Cookbook for Amazon EKS* to avoid incorrect bucket creation while creating networking infrastructure components. Added two required environment variables to the settings required when running the OAuth 2.0 Authorization Code Grant Flow benchmark. The same change has been made to the GKE, EKS, and AKS Cookbooks. You can find an example of the two new environment variables in "Before You Begin" in the *Cloud Deployment Model Cookbook for Amazon EKS*.
2020-08-06	Fixed step 7.d.4 in "To Create a Kubernetes Cluster for CDM" in the *Cloud Deployment Model Cookbook for Amazon EKS* to set AMI IDs for the correct worker node names. Corrected the default repository name in step 5 of "To Set up Your Local Computer to Push Docker Images" in the *Cloud Deployment Model Cookbook for Amazon EKS*.
2020-06-25	The `forgeops` release tag is revised to `2020.06.24-laPaniscia`.
2020-06-19	The **print-secrets.sh** command has been simplified: • A version number is no longer required as a command parameter. • Options have been added to get passwords for a single identity. For example, to get the password for the `amadmin` user, you can run the **print-secrets.sh amadmin** command. See "*Using the Platform*" in the *DevOps Developer's Guide: Using Minikube* and "*Using the CDM*" in the *Cloud Deployment Model Cookbook for GKE* for examples. Some versions of third-party software have been updated. See the *Third-Party Software* sections in the *DevOps Guides* for more information.
2020-05-28	The `forgeops` release tag is revised to `2020.05.13-AlPomodoro.1`.
2020-05-07	The method for obtaining the `forgeops` repository has changed. We now recommend that users clone the repository, and the create a branch based on the current release tag. See "Obtaining the forgeops Repository" in the *DevOps Developer's Guide: Using Minikube* and similar sections in the DevOps Developer's Guide: Using a Shared Cluster and in all of the *CDM Cookbooks*. The *Site Reliability Guide* has been renamed to the *Cloud Deployment Guide*. The new title more accurately reflects the guide's content.

Date	Description
	A new chapter, "*About the forgeops Repository*" in the *Cloud Deployment Guide*, describes how to work with the `forgeops` repository. The chapter covers release tags and repository forks. It also describes the repository's contents, and strategies for updating `forgeops` repository clones. "Initializing Directory Data From Binary Backup" in the *Cloud Deployment Guide*, replaces the section, *Using CDM Restore*. The section provides instructions for using the new **backup-loader.sh** script to create PVCs from DS binary backups before you start the platform, so that DS instances in the platform use the data from the PVCs. Microsoft Windows-specific instructions and PowerShell example commands have been removed from the *DevOps Guides* and the *CDM Cookbooks*. Windows users are now required to install the CDM and the CDK in a Linux virtual machine. For more information, click the plus signs next to the text labeled *Windows users* in the *DevOps Guides* and the *CDM Cookbooks*.
2020-04-10	Corrected a bug in the example command in step 5e of "To Create a Kubernetes Cluster for CDM" in the *Cloud Deployment Model Cookbook for AKS* . Removed language in the "CDM Overview" in the *Cloud Deployment Model Cookbook for GKE* (and in the analogous sections of the EKS and AKS Cookbooks) that stated that IDM is configured to use AM for authentication. This is not the case in the current version of the CDM.
2020-04-03	A major revision of the Start Here Guide clarifies the relationship of the CDM to production deployments, better describes the set of activities performed when building a ForgeRock Identity Platform service in the cloud, and lists skills and expertise required for performing these activities.
2020-03-18	The CDM and CDK now use administrator passwords that are randomly generated by the secrets generator. See "*Using the Platform*" in the *DevOps Developer's Guide: Using Minikube* for information about how to obtain the administrator passwords. Note that the technique to obtain the passwords is the same for the CDK and the CDM.
2020-03-13	Fixed the command to forward port 3000 on your local computer to port 3000 on the Grafana web server. See "Monitoring the CDM" in the *Cloud Deployment Model Cookbook for GKE* and the analogous sections in the EKS and AKS Cookbooks.
2020-03-09	Added an example command to create repositories for the ForgeRock Identity Platform in the ECR registry. See "To Set Up Amazon EKS Cluster Dependencies" in the *Cloud Deployment Model Cookbook for Amazon EKS*. A single, consolidated Cloud Deployment Guide replaces the *Site Reliability Guide for GKE* and the *Site Reliability Guide for Amazon EKS*.
2020-03-03	Added a completely revised Cloud Deployment Model Cookbook for AKS to the DevOps documentation. The new Cookbook is no longer designated as evaluation-only. Because of the previous lack of support in AKS for multiple availability zones for AKS clusters, ForgeRock formerly recommended against deploying the platform on Azure in

Date	Description
	production. With support for zones available in AKS, Azure is now a supported platform for production deployments. Changes from the evaluation-only version of the Cookbook include: • The CDM deployment topology on Azure now matches the CDM deployment topology on GCP and AWS. • Pulumi scripts demonstrate AKS cluster creation. • Benchmark results are available for a sample deployment with 10,000,000 users.

Appendix A. Getting Support

This appendix contains information about support options for the ForgeRock Cloud Developer's Kit, the ForgeRock Cloud Deployment Model, and the ForgeRock Identity Platform.

ForgeRock DevOps Support

ForgeRock has developed artifacts in the `forgeops` Git repository for the purpose of deploying the ForgeRock Identity Platform in the cloud. The companion ForgeRock DevOps documentation provides examples, including the ForgeRock Cloud Developer's Kit (CDK) and the ForgeRock Cloud Deployment Model (CDM), to help you get started.

These artifacts and documentation are provided on an "as is" basis. ForgeRock does not guarantee the individual success developers may have in implementing the code on their development platforms or in production configurations.

Commercial Support

ForgeRock provides commercial support for the following DevOps resources:

- Artifacts in the `forgeops` Git repository:
  - Files used to build Docker images for the ForgeRock Identity Platform:
    - Dockerfiles
    - Scripts and configuration files incorporated into ForgeRock's Docker images
    - Canonical configuration profiles for the platform

  - Kustomize bases and overlays
  - Skaffold configuration files
- ForgeRock DevOps guides.

ForgeRock provides commercial support for the ForgeRock Identity Platform. For supported components, containers, and Java versions, see the following:

- *ForgeRock Access Management Release Notes*
- *ForgeRock Identity Management Release Notes*
- *ForgeRock Directory Services Release Notes*
- *ForgeRock Identity Gateway Release Notes*

Support Limitations

ForgeRock provides no commercial support for the following:

- Artifacts other than Dockerfiles, Kustomize bases, Kustomize overlays, and Skaffold YAML configuration files in the `forgeops` Git repository. Examples include scripts, example configurations, and so forth.
- Non-ForgeRock infrastructure. Examples include Docker, Kubernetes, Google Cloud Platform, Amazon Web Services, and so forth.
- Non-ForgeRock software. Examples include Java, Apache Tomcat, NGINX, Apache HTTP Server, Certificate Manager, Prometheus, and so forth.
- Production deployments that use ForgeRock's evaluation-only Docker images. When deploying the ForgeRock Identity Platform using Docker images, you must build and use your own images for production deployments. For information about how to build your own Docker images for the ForgeRock Identity Platform, see "*Building Base Docker Images*" in the *Cloud Deployment Guide*.

Third-Party Kubernetes Services

ForgeRock supports deployments on Google Kubernetes Engine (GKE), Amazon Elastic Kubernetes Service (Amazon EKS), Microsoft Azure Kubernetes Service (AKS), and Red Hat OpenShift.

Red Hat OpenShift is a tested and supported platform using Kubernetes for deployment. ForgeRock uses OpenShift tools such as the OpenShift installer, as well as other representative environments such as Amazon AWS for the testing. We do not test using bare metal due to the many customer permutations of deployment and configuration that may exist, and therefore cannot guarantee that we have tested in the same way a customer chooses to deploy. We will make commercially reasonable efforts to provide first-line support for any reported issue. In the case we are unable to reproduce a

reported issue internally, we will request the customer engage OpenShift support to collaborate on problem identification and remediation. Customers deploying on OpenShift are expected to have a support contract in place with IBM/Red Hat that ensures support resources can be engaged if this situation may occur.

Accessing Documentation Online

ForgeRock publishes comprehensive documentation online:

- The ForgeRock Knowledge Base offers a large and increasing number of up-to-date, practical articles that help you deploy and manage ForgeRock software.

  While many articles are visible to community members, ForgeRock customers have access to much more, including advanced information for customers using ForgeRock software in a mission-critical capacity.

- ForgeRock developer documentation, such as this document, aims to be technically accurate with respect to the sample that is documented. It is visible to everyone.

How to Report Problems or Provide Feedback

If you are a named customer Support Contact, contact ForgeRock using the Customer Support Portal to request information or report a problem with Dockerfiles, Kustomize bases, Kustomize overlays, or Skaffold YAML configuration files in the CDK or the CDM.

If you have questions regarding the CDK or the CDM that are not answered in the documentation, file an issue at https://github.com/ForgeRock/forgeops/issues.

When requesting help with a problem, include the following information:

- Description of the problem, including when the problem occurs and its impact on your operation.
- Steps to reproduce the problem.

  If the problem occurs on a Kubernetes system other than Minikube, GKE, EKS, OpenShift, or AKS, we might ask you to reproduce the problem on one of those.

- HTML output from the **debug-logs.sh** script. For more information, see "Reviewing Pod Descriptions and Container Logs" in the *DevOps Developer's Guide: Using Minikube*.
- Description of the environment, including the following information:
  - Environment type: Minikube, GKE, EKS, AKS, or OpenShift.
  - Software versions of supporting components:
    - Oracle VirtualBox (Minikube environments only).

  - Docker client (all environments).
  - Minikube (all environments).
  - **kubectl** command (all environments).
  - Kustomize (all environments).
  - Skaffold (all environments).
  - Google Cloud SDK (GKE environments only).
  - Amazon AWS Command Line Interface (EKS environments only).
  - Azure Command Line Interface (AKS environments only).
- `forgeops` repository branch.
- Any patches or other software that might be affecting the problem.

Getting Support and Contacting ForgeRock

ForgeRock provides support services, professional services, training through ForgeRock University, and partner services to assist you in setting up and maintaining your deployments. For a general overview of these services, see https://www.forgerock.com.

ForgeRock has staff members around the globe who support our international customers and partners. For details on ForgeRock's support offering, including support plans and service-level agreements (SLAs), visit https://www.forgerock.com/support.