Start Here

/ 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

Introduction to using ForgeRock's DevOps artifacts for cloud deployments.

Table of Contents

Take Stock of Your Readiness Level

ForgeRock provides several resources to help you get started in the cloud. These resources demonstrate how to deploy ForgeRock Identity Platform™ (the platform) on Kubernetes. Before you proceed, review the following precautions:

- Deploying ForgeRock software in a containerized environment requires advanced proficiency in many technologies. See "Assess Your Skill Level" for details.

- If you don't have experience with complex Kubernetes deployments, then either engage a certified ForgeRock consulting partner or deploy the platform on traditional architecture.

- Don't deploy ForgeRock software in Kubernetes in production until you have successfully deployed and tested the software in a non-production Kubernetes environment.

For information about obtaining support for ForgeRock Identity Platform software, see "*Getting Support*" in the *DevOps Release Notes*.

Introducing the CDK and CDM

The forgeops repository and DevOps documentation address a range of our customers' typical business needs. The repository contains artifacts for two primary resources to help you with cloud deployment:

- **Cloud Developer's Kit (CDK)**. The CDK is a minimal sample deployment for development purposes. Developers deploy the CDK, and then access AM's and IDM's GUI consoles and REST APIs to configure the platform and build customized Docker images for the platform.

- **Cloud Deployment Model (CDM)**. The CDM is a reference implementation for ForgeRock cloud deployments. You can get a sample ForgeRock Identity Platform deployment up and running in the cloud quickly using the CDM. After deploying the CDM, you can use it to explore how you might configure your Kubernetes cluster before you deploy the platform in production.

  The CDM is a robust sample deployment for demonstration and exploration purposes only. *It is not a production deployment*.

	CDK	CDM
Fully integrated AM, IDM, and DS installations	✔	✔
Randomly generated secrets	✔	✔

	CDK	CDM
Resource requirement	Namespace in a GKE, EKS, AKS, or Minikube cluster	Dedicated GKE, EKS, or AKS cluster
Can run on Minikube	✔	
Multi-zone high availability		✔
Replicated directory services		✔
Ingress configuration		✔
Certificate management		✔
Prometheus monitoring, Grafana reporting, and alert management		✔

ForgeRock's DevOps documentation helps you deploy the CDK and CDM:

- **DevOps Developers Guide**. (For Minikube | For Shared Clusters) Tells you how to install the CDK, modify the AM and IDM configurations, and create customized Docker images for the ForgeRock Identity Platform.

- **CDM Cookbook**. (For GKE | For EKS | For AKS) Tells you how to quickly create a Kubernetes cluster on Google Cloud Platform (GCP), Amazon Web Services (AWS), or Microsoft Azure, install the ForgeRock Identity Platform, access components in the deployment, and run lightweight benchmarks to test DS, AM, and IDM performance.

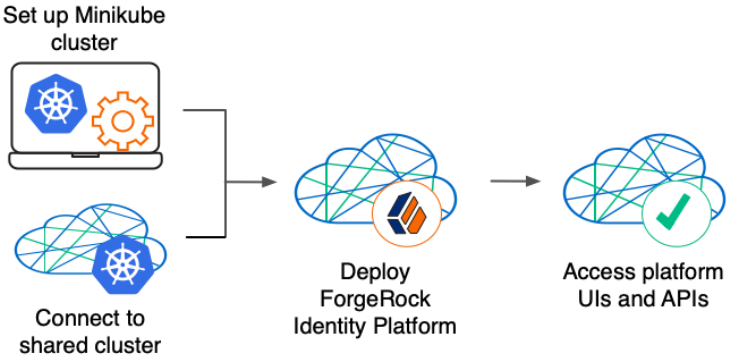
- **Cloud Deployment Guide**. Contains how-tos for customizing monitoring, setting alerts, backing up and restoring directory data, and modifying CDM's default security configuration.

- **DevOps Release Notes**. Keeps you up-to-date with the latest changes to the `forgeops` repository.

Try Them Out

Before you start planning a production deployment, deploy either the CDK or the CDM—or both. If you're new to Kubernetes, or new to the ForgeRock Identity Platform, deploying these resources is a great way to learn. And when you've finished deploying them, you'll have sandboxes suitable for exploring ForgeRock cloud deployment.

Deploy the CDK


Set up Minikube cluster
Connect to shared cluster
Deploy ForgeRock Identity Platform
Access platform UIs and APIs

The CDK is a minimal sample deployment of the ForgeRock Identity Platform. If you have access to a cluster on GCP, EKS, or AKS, you can install the CDK in a namespace on your cluster. But even if you don't have access to a cloud-based cluster, you can still deploy the CDK on a local computer running Minikube, and when you're done, you'll have a namespace on a local Kubernetes cluster with the ForgeRock Identity Platform.

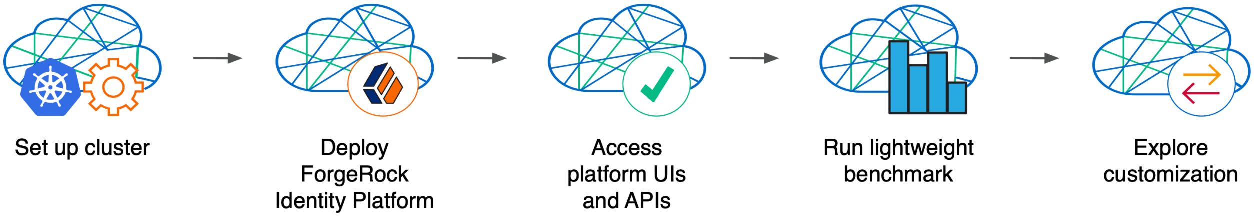
Prerequisite technologies and skills:

- Git
- Docker
- Kubernetes, running on a cloud platform or on Minikube

More information:

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

Deploy the CDM


Set up cluster
Deploy ForgeRock Identity Platform
Access platform UIs and APIs
Run lightweight benchmark
Explore customization

Deploy the CDM on GCP, EKS, or AKS to quickly spin up the platform for demonstration purposes. You'll get a feel for what it's like to deploy the platform on a Kubernetes cluster in the cloud. When you're done, you won't have a production-quality deployment. But you will have a robust, reference implementation of the platform that you can use to explore optional deployment customizations.[1]

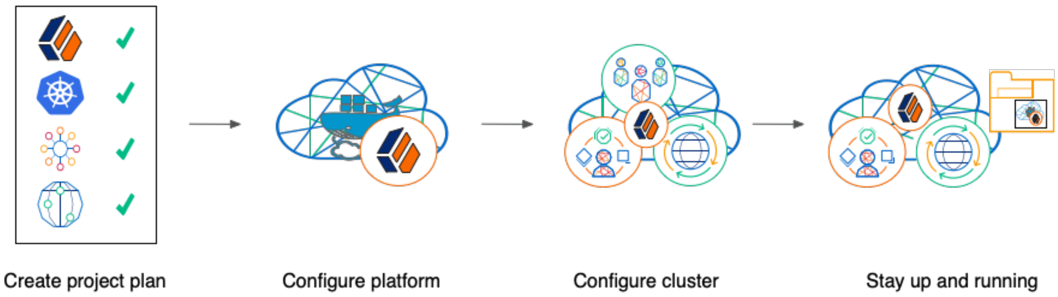
Prerequisite technologies and skills:

- Git
- GCP, AWS, or Azure
- Kubernetes, running on GCP, AWS, or Azure

More information:

- Cloud Deployment Model Cookbook for GKE
- Cloud Deployment Model Cookbook for Amazon EKS
- Cloud Deployment Model Cookbook for AKS

Build Your Own Service



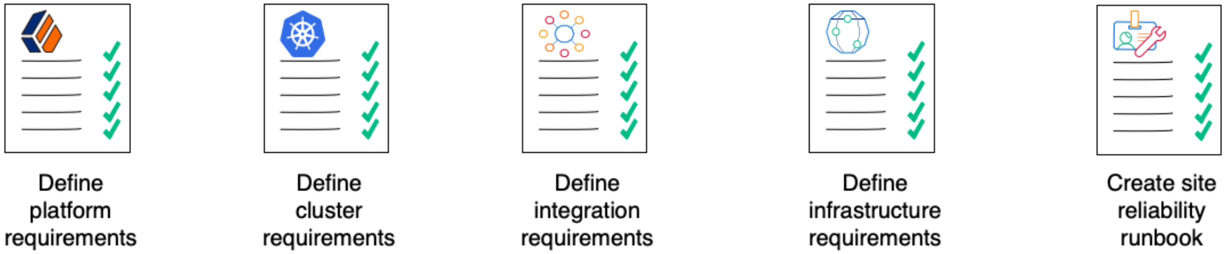
Perform the following activities to customize, deploy, and maintain a production ForgeRock Identity Platform implementation in the cloud:

- "Create a Project Plan"
- "Configure the Platform"
- "Configure Your Cluster"

[1] Optional customizations are deployment options that you might want to use in production that are not part of the CDM. Examples include, but are not limited to securing SSL with a certificate that's dynamically obtained from Let's Encrypt; using an ingress controller other than the NGINX ingress controller; resizing the cluster to meet your business requirements; configuring Alert Manager to issue alerts when usage thresholds have been reached.

- "Stay Up and Running"

Create a Project Plan


Define platform requirements
Define cluster requirements
Define integration requirements
Define infrastructure requirements
Create site reliability runbook

After you've spent some time exploring the CDK and CDM, you're ready to define requirements for your production deployment. *Remember, the CDM is not a production deployment*. Use the CDM to explore deployment customizations, and incorporate the lessons you've learned as you build your own production service.

Analyze your business requirements and define how the ForgeRock Identity Platform needs to be configured to meet your needs. Identify systems to be integrated with the platform, such as identity databases and applications, and plan to perform those integrations. Assess and specify your deployment infrastructure requirements, such as backup, system monitoring, Git repository management, CI/CD, quality assurance, security, and load testing.

Prerequisite technologies and skills:

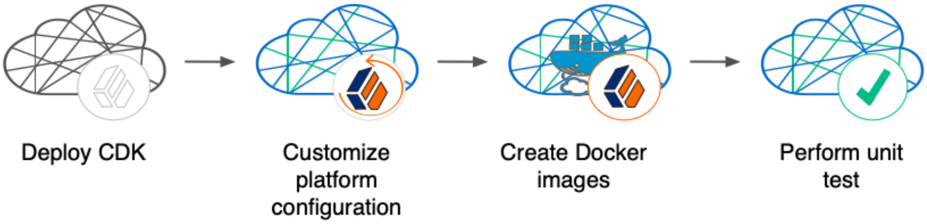
- Project planning and management
- Git
- Docker
- GCP, AWS, or Azure
- Kubernetes, running on GCP, AWS, or Azure
- ForgeRock Identity Platform
- Applications and databases that you plan to integrate with ForgeRock Identity Platform
- CI/CD for a production deployment in the cloud
- Integration testing
- Deployment hardening and security

- Benchmarking and load testing
- Site reliability

More information:

- All the DevOps documentation at https://backstage.forgerock.com/docs/forgeops/6.5

Configure the Platform



With your project plan defined, you're ready to configure the ForgeRock Identity Platform to meet the plan's requirements. Install the CDK on your developers' computers. Configure AM and IDM. If needed, include integrations with external applications in the configuration. Iteratively unit test your configuration as you modify it. Build customized Docker images that contain the configuration.

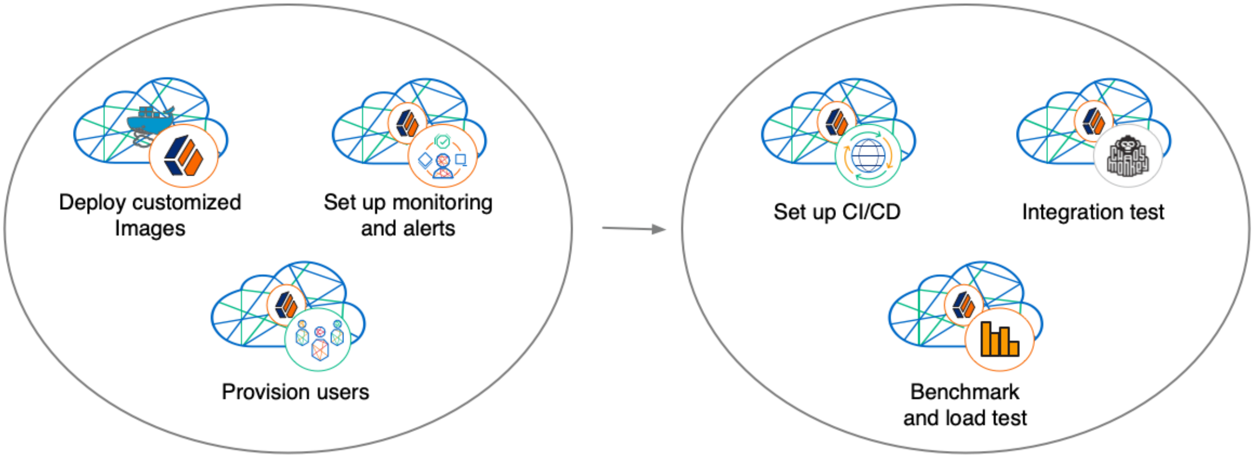
Prerequisite technologies and skills:

- ForgeRock Identity Platform
- Git
- Kubernetes, running on a cloud platform or on Minikube
- Docker

More information:

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

Configure Your Cluster



With your project plan defined, you're ready to configure a Kubernetes cluster that meets the requirements defined in the plan. Install the platform using the customized Docker images developed in "Configure the Platform". Provision the ForgeRock identity repository with users, groups, and other identity data. Load test your deployment, and then size your cluster to meet service level agreements. Perform integration tests. Harden your deployment. Set up CI/CD for your deployment. Create monitoring alerts so that your site reliability engineers are notified when the system reaches thresholds that affect your SLAs. Implement database backup and test database restore. Simulate failures while under load to make sure your deployment can handle them.

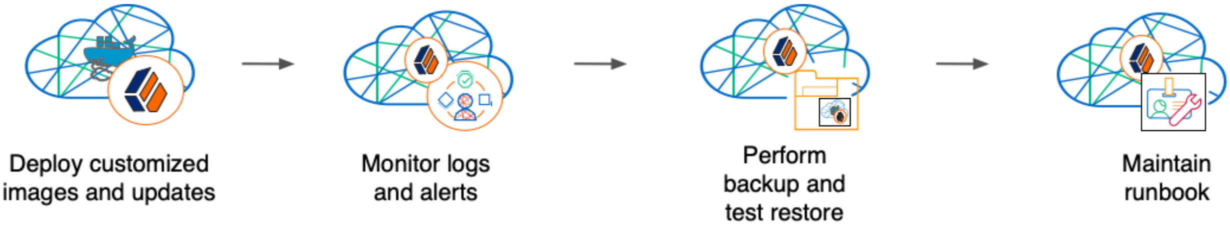
Prerequisite technologies and skills:

- GCP, AWS, or Azure
- Git
- Kubernetes, running on GCP, AWS, or Azure
- ForgeRock Identity Platform
- CI/CD for a production deployment in the cloud
- Integration testing
- Deployment hardening and security
- Benchmarking and load testing
- Site reliability

More information:

- Cloud Deployment Guide
- Cloud Deployment Model Cookbook for GKE
- Cloud Deployment Model Cookbook for Amazon EKS
- Cloud Deployment Model Cookbook for AKS

Stay Up and Running


Deploy customized images and updates
Monitor logs and alerts
Perform backup and test restore
Maintain runbook

By now, you've configured the platform, configured a Kubernetes cluster, and installed the platform with your customized configuration in the cluster. Run your ForgeRock Identity Platform deployment in your cluster, continually monitoring it for performance and reliability. Take backups as needed.

Prerequisite technologies and skills:

- Git
- GCP, AWS, or Azure
- Kubernetes, running on GCP, AWS, or Azure
- ForgeRock Identity Platform
- CI/CD for a production deployment in the cloud
- Site reliability

More information:

- Cloud Deployment Guide

Assess Your Skill Level

Benchmarking and Load Testing

I can:

- Write performance tests, using tools such as Gatling and Apache JMeter, to ensure that system meets required performance thresholds and service level agreements (SLAs).
- Resize a Kubernetes cluster, taking into account performance test results, thresholds, and SLAs.
- Run Linux performance monitoring utilities, such as **top**.

CI/CD for Cloud Deployments

I have experience:

- Designing and implementing a CI/CD process for a cloud-based deployment running in production.
- Using a cloud CI/CD tool, such as Tekton, Google Cloud Build, Codefresh, AWS CloudFormation, or Jenkins, to implement a CI/CD process for a cloud-based deployment running in production.
- Integrating GitOps into a CI/CD process.

Docker

I know how to:

- Write Dockerfiles.
- Create Docker images, and push them to a private Docker registry.
- Pull and run images from a private Docker registry.

I understand:

- The concepts of Docker layers, and building images based on other Docker images using the **FROM** instruction.
- The difference between the **COPY** and **ADD** instructions in a Dockerfile.

Git

I know how to:

- Use a Git repository collaboration framework, such as GitHub, GitLab, or Bitbucket Server.
- Perform common Git operations, such as cloning and forking repositories, branching, committing changes, submitting pull requests, merging, viewing logs, and so forth.

External Application and Database Integration

I have expertise in:

- AM policy agents.
- Configuring AM policies.
- Synchronizing and reconciling identity data using IDM.
- Managing cloud databases.
- Connecting ForgeRock Identity Platform components to cloud databases.

ForgeRock Identity Platform

I have:

- Attended ForgeRock University training courses.
- Deployed the ForgeRock Identity Platform in production, and kept the deployment highly available.
- Configured DS replication.
- Passed the ForgeRock Certified Access Management and ForgeRock Certified Identity Management exams (highly recommended).

GCP, AWS, or Azure (Basic)

I can:

- Use the graphical user interface for GCP, AWS, or Azure to navigate, browse, create, and remove Kubernetes clusters.
- Use the cloud provider's tools to monitor a Kubernetes cluster.
- Use the command user interface for GCP, AWS, or Azure.
- Administer cloud storage.

GCP, AWS, or Azure (Expert)

In addition to the skills and expertise listed in "GCP, AWS, or Azure (Basic)" I can:

- Read the Pulumi scripts in the `forgeops` repository to see how the CDM cluster is configured.
- Create and manage a Kubernetes cluster using an infrastructure-as-code tool such as Pulumi, Terraform, or AWS CloudFormation.
- Configure multi-zone and multi-region Kubernetes clusters.

- Configure cloud-provider identity and access management (IAM).
- Configure virtual private clouds (VPCs) and VPC networking.
- Manage keys in the cloud using a service such as Google Key Management Service (KMS), Amazon KMS, or Azure Key Vault.
- Configure and manage DNS domains on GCP, AWS, or Azure.
- Troubleshoot a deployment running in the cloud using the cloud provider's tools, such as Google Stackdriver, Amazon CloudWatch, or Azure Monitor.
- Integrate a deployment with certificate management tools, such as cert-manager and Let's Encrypt.
- Integrate a deployment with monitoring and alerting tools, such as Prometheus and Alertmanager.

I have obtained one of the following certifications (highly recommended):

- Google Certified Associate Cloud Engineer Certification.
- AWS professional-level or associate-level certifications (multiple).
- Azure Administrator.

Integration Testing

I can:

- Automate QA testing using a test automation framework.
- Design a chaos engineering test for a cloud-based deployment running in production.
- Use chaos engineering testing tools, such as Chaos Monkey.

Kubernetes (Basic)

I've gone through the tutorials at kubernetes.io, and am able to:

- Use the **kubectl** command to determine the status of all the pods in a namespace, and to determine whether pods are operational.
- Use the **kubectl describe pod** command to perform basic troubleshooting on pods that are not operational.
- Use the **kubectl** command to obtain information about namespaces, secrets, deployments, and stateful sets.
- Use the **kubectl** command to manage persistent volumes and persistent volume claims.

Kubernetes (Expert)

In addition to the skills and expertise listed in "Kubernetes (Basic)" I have:

- Configured role-based access to cloud resources.
- Configured Kubernetes objects, such as deployments and stateful sets.
- Configured Kubernetes ingresses.
- Passed the Cloud Native Certified Kubernetes Administrator exam (highly recommended).

Project Planning and Management for Cloud Deployments

I have planned and managed:

- A production deployment in the cloud.
- A production deployment of ForgeRock Identity Platform.

Security and Hardening for Cloud Deployments

I can:

- Harden a ForgeRock Identity Platform deployment.
- Configure TLS, including mutual TLS, for a multi-tiered cloud deployment.
- Configure cloud identity and access management and role-based access control for a production deployment.
- Configure encryption for a cloud deployment.
- Configure Kubernetes pod security and network security policies.
- Configure private Kubernetes networks, deploying bastion servers as needed.
- Undertake threat modeling exercises.
- Scan Docker images to ensure container security.
- Configure and use private Docker container registries.

Site Reliability Engineering for Cloud Deployments

I can:

- Manage multi-zone and multi-region deployments.

- Implement DS backup and restore in order to recover from a database failure.
- Manage cloud disk availability issues.
- Analyze monitoring output and alerts, and respond should a failure occur.
- Obtain logs from all the software components in my deployment.
- Follow the cloud provider's recommendations for patching and upgrading software in my deployment.
- Implement an upgrade scheme, such as blue/green or rolling upgrades. software in my deployment.
- Create a Site Reliability Runbook for the deployment, documenting all the the procedures to be followed and other relevant information.
- Follow all the procedures in the project's Site Reliability Runbook, and revise the runbook if it becomes out-of-date.