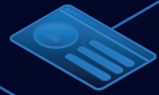
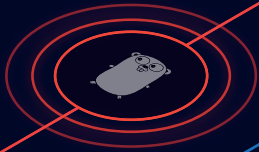




# Securing Privileged User Accounts



# Overview

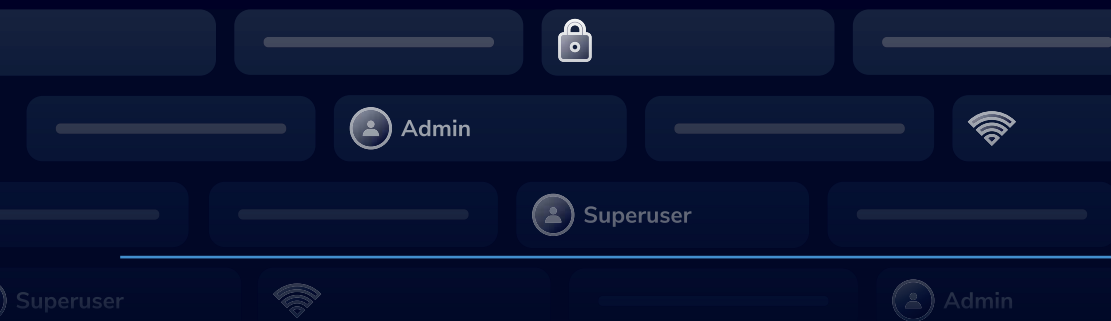
One of the most sensitive secrets in your organization are without a doubt the credentials for your superuser accounts. These accounts, such as the root account for a Linux server, the Administrator account for a Windows server, or the Admin accounts for a network device, have virtually unlimited privileges. Anyone with the credentials for these privileged user accounts effectively has free reign over your systems, from creating and deleting users to installing software and changing system settings.

Because of the sensitivity of privileged-user credentials, many security standards require that you periodically rotate your superuser account passwords to remain compliant. As these accounts are often shared by multiple applications and users, ensuring everyone has the updated password is an administrative nightmare, and often, just distributing a new password increases the risk of it being intercepted and abused. Once the password is out there, you have no control over how it is managed, stored, and used.

# Automating Password Rotation

The Akeyless Vault Platform introduces rotated secrets – a simple way to automatically rotate and protect privileged account credentials. With rotated secrets, your superuser passwords are:

- Rotated according to a predefined schedule
- Encrypted using Akeyless's patented Distributed Fragments Cryptography™ (DFC) technology
- Securely stored in the Akeyless vault, from where they can be retrieved by authorized clients. In case of emergency, for example, if you know a password has been compromised, you can also manually trigger a password reset. Let's take a closer look at how all this works.



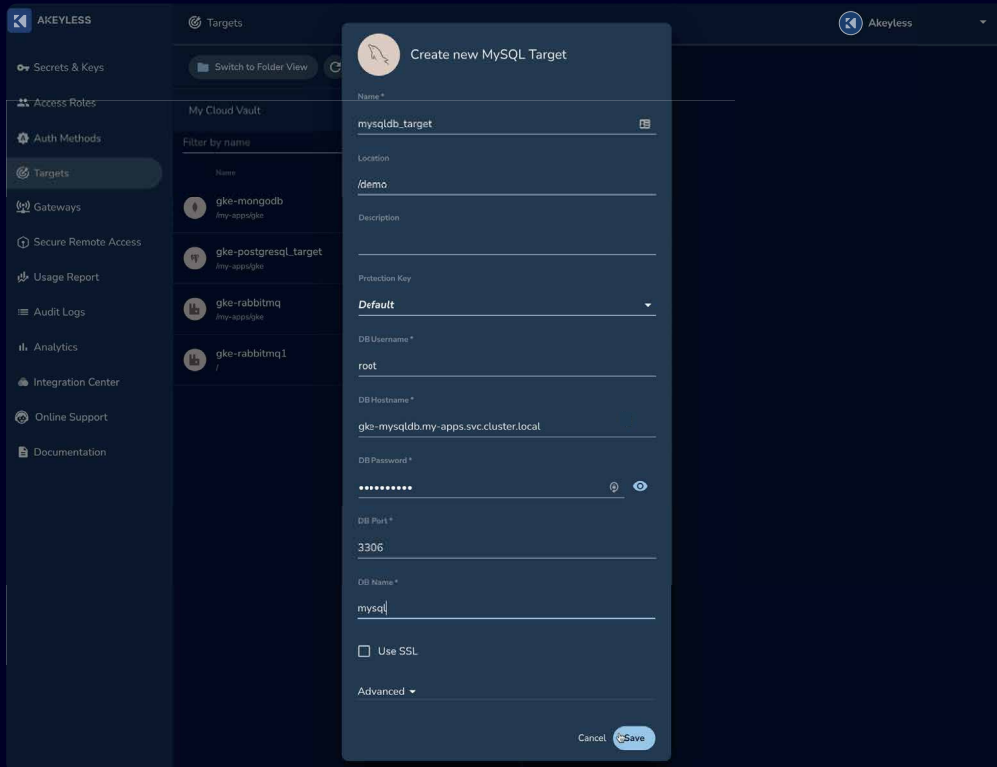
# Create a Target

Get started with a rotated secret in the Akeyless Vault Platform by setting up the target for the secret, which could be a database or other server for which the superuser credentials are defined. The rotated secret itself is a privileged user account on the target.

Akeyless supports a growing list of targets, including AWS, Kubernetes, various databases, and SSH. New targets are continuously being added;

check out the [online documentation](#) for the most up-to-date list. To set up a target, first define its location, such as a URL or hostname, and specify the credentials required to connect to the target and update the privileged user account password.

For example, for this SQL server target, specify the database name, hostname, and port for connecting to the database, as well as the required username and password.



# Create a Rotated Secret

Now it's time for the rotated secret itself. When you create a rotated secret, you need to name it and define the secret settings, such as how often the secret should be rotated, and the secret target. You can define a rotated secret to automatically update the password at defined intervals, or manually trigger a password update from the CLI or from the Akeyless Console. Similarly, you can set the rotated secret to reset the privileged user account password using the credentials in the target, or set it to use the privileged user account credentials to reset itself.

For example, for this SQL rotated secret, specify the rotated secret name, the target for which the secret is defined, the username and password to be updated, and the rotation schedule.

The screenshot shows the 'Create Rotated Secret' dialog in the Akeyless console, with the 'Basic Configuration' tab selected. The form fields are as follows:

- Name: `mysqldb_rotate`
- Location: `/demo`
- Description: (empty)
- Tags: (empty)
- Target: `demo/mysqldb_target`
- Authenticate with the following credentials: `User credentials`
- Rotator Type: (empty)

Buttons at the bottom: Finish, Next

The screenshot shows the 'Create Rotated Secret' dialog in the Akeyless console, with the 'Basic Configuration' tab selected. The form fields are as follows:

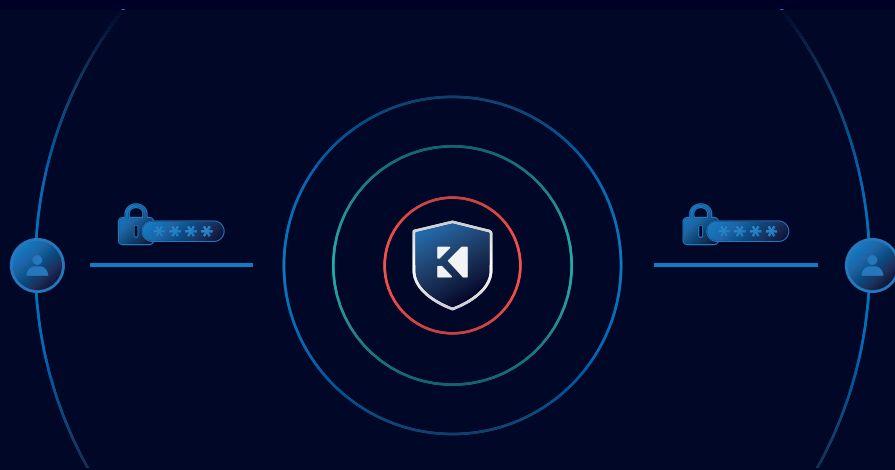
- Username: `root`
- Password: (masked with dots)
- Gateway: `GCP-GW (acc-qvuumbdx1gjp-u91zmregsrwb/defaultCluster)`
- Protection Key: `Default`
- Auto Rotate:  Recurrent  Manual Only
- Days to rotate: `7`
- Rotation Hour (local time zone): `00:00`

Buttons at the bottom: Finish, Next

# Get the Rotated Secret Value

According to the schedule defined in the rotated secret, or when triggered from the CLI or from the Akeyless Console, the Akeyless Vault Platform generates a new password for the privileged user account, resets it on the target machine, and stores the updated secret value so that it can be retrieved when required.

To keep the secret value safe, add your rotated secret to an access role, with the appropriate permissions. Akeyless will only provide the secret value to authenticated clients who are authorized to view it.



The screenshot displays the Akeyless console interface for a rotated secret named 'mysqladb\_rotate'. The 'Password' field is highlighted, showing the newly generated secret value: 'qaVQ3oLDsd'. Other fields include 'Username' (root), 'Gateway' (GCP-GW), and 'Protection Key' (Default). A 'Rotate Secret Now' button is visible, along with a 'Secure Remote Access' dropdown menu.

The screenshot displays the Akeyless console interface for a rotated secret named 'mysqladb\_rotate'. The 'Password' field is highlighted, showing a masked secret value: '\*\*\*\*\*'. A 'Next Rotation' notification is displayed at the bottom, indicating the secret will be automatically rotated in 7 days. Other fields include 'Username' (root), 'Gateway' (GCP-GW), and 'Protection Key' (Default). A 'Rotate Secret Now' button is visible, along with a 'Secure Remote Access' dropdown menu.

# Conclusion

Superuser credentials are often static account credentials that are used by multiple applications and users. The sensitive nature of these accounts necessitates that their passwords be rotated frequently.

By automating this process, Akeyless rotated secrets enable you to implement an aggressive rotation schedule and limit the risk if the credentials were exposed. In addition, by securely storing the password in a central vault, Akeyless eliminates the need to spread the password, reducing the risk of the password being compromised.



A Few Words About



The Akeyless Vault Platform is a unified, SaaS-based solution for Secrets Management, Secure Remote Access, and Data Protection that enables DevOps, Cloud Transformation, and Zero-Trust Access use cases. Using a patented, FIPS 140-2 certified, innovative KMS technology (Akeyless DFC™), organizations can leverage an all-in-one solution to manage credentials, certificates, and keys for humans and workloads. The Akeyless Vault Platform is SOC2 certified and operates across a highly distributed environment, delivering Just-in-Time secrets and access to humans and machines (i.e., applications, microservices) efficiently and at infinite scale. To learn more, or schedule a demo, visit [www.akeyless.io](http://www.akeyless.io).