# [AWS Security](https://confluence.ellucian.com/display/PD/AWS+Security)

# AWS Security - Secrets Manager and Parameter Store

1.Create a secret in AWS Secrets Manager and use it in an application by referencing its ARN.

A screenshot of a computer

Description automatically generated with medium confidence

Using it in the application :

-----------------------------------

import boto3

secret\_name = "your-secret-name"

secret\_arn = "arn:aws:secretsmanager:us-west-2:123456789012:secret: arn:aws:secretsmanager:us-west-2:571456924570:secret:secret-test-FZywYQ "

secrets\_client = boto3.client("secretsmanager")

response = secrets\_client.get\_secret\_value(SecretId=secret\_arn)

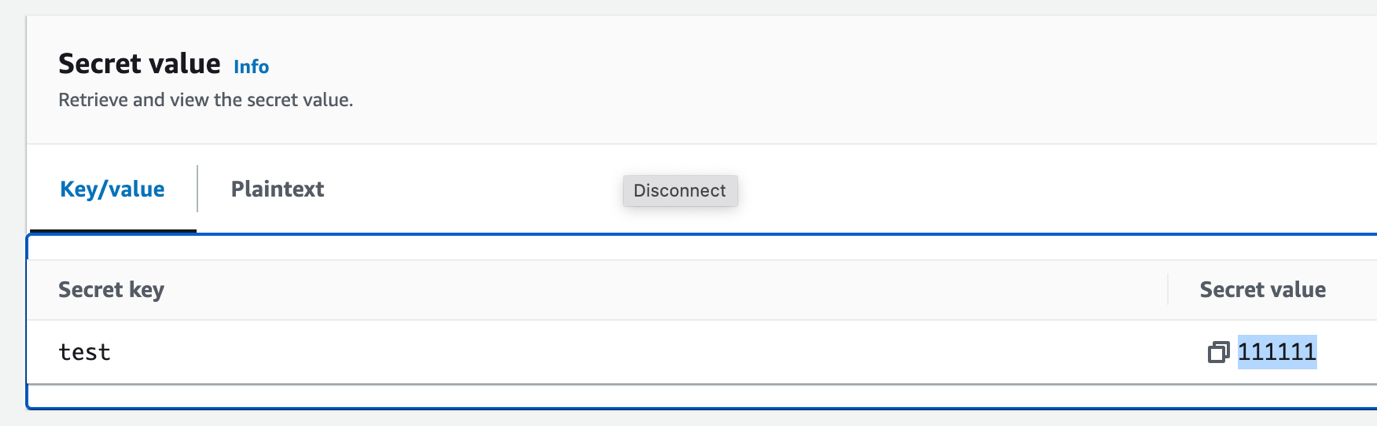
secret\_value = response["SecretString"]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2. Rotate a secret in AWS Secrets Manager and update the application to use the new version of the secret.

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Here's an example using the AWS SDK for Python (Boto3) to retrieve a secret in the application code:

import boto3

# Create a Secrets Manager client

secrets\_client = boto3.client('secretsmanager')

# Specify the secret's ARN or name

secret\_id = "secret-test"

# Retrieve the secret value

response = secrets\_client.get\_secret\_value(SecretId=test)

secret\_value = response['123456']

# Use the secret value in the application

# Update your application logic to use the new secret value

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3. Use AWS CLI to retrieve a parameter from the Parameter Store and display its value.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

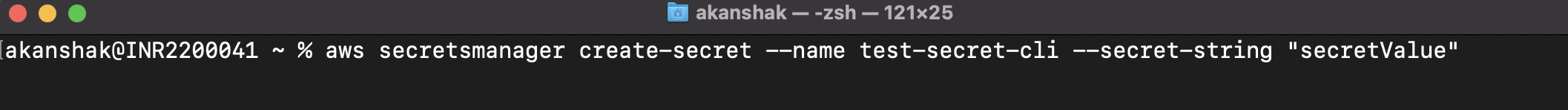
Description automatically generated with medium confidence



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

4. Use AWS CLI to create a secret in AWS Secrets Manager and update its value.

To create secret using CLI :



To update value :



5. Use AWS SDK to retrieve a parameter from the Parameter Store and display its value in a Javascript application.

1. npm install aws-sdk
2. // In JavaScript file

process.env.AWS\_ACCESS\_KEY\_ID = 'Test-Parameter’;

process.env.AWS\_SECRET\_ACCESS\_KEY = '1000’;

process.env.AWS\_REGION = 'us-east-1';

1. **Retrieve Parameter from Parameter Store**:

const AWS = require('aws-sdk');

// Create a new instance of the AWS SDK with the configured credentials

const ssm = new AWS.SSM();

// Define the name of the parameter you want to retrieve

const parameterName = ‘Test-Parameter’;

// Create a request to retrieve the parameter

const params = {

Name: parameterName,

WithDecryption: true // Set this to true if the parameter is encrypted

};

// Retrieve the parameter value

ssm.getParameter(params, (err, data) => {

if (err) {

console.error('Error retrieving parameter:', err);

} else {

const parameterValue = data.Parameter.Value;

// Display the parameter value in your application

console.log('Parameter value:', parameterValue);

}

});

**4.Run the JavaScript Application**:

node your-app.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*