

How to encrypt scenario's ec2 cloud config files using API calls

There are now options in AIQ UX to encrypt different files with sensitive data needed for CICD pipeline jobs as seen in the screenshot below for ec2 cloud config files `/AppvanceServer/rest/file/encryptFileContent` call. Below you will find examples on how to encrypt ec2 cloud files used in scenario to start and stop ec2 instances using the en. If you need to encrypt the CICD credential file used to upload dashboard to Amazon S3 service can use the same `encryptFileContent` HTTP call but in this case the payload is not a JSON but the simple text of that file.

The screenshot displays the AIQ UX interface with a sidebar on the left containing navigation links: Home, Web Testing, Mobile Testing, Autonomous Testing, API Test Designer, Services Workbench, Scenario Editor, Run Scenario, Reports, Test Data, Traffic, Global Options, Output Logs, AIQ Status, Preferences, and Cloud Config. The main panel is titled 'EC2 Configuration' and includes tabs for 'Test Configuration', 'All Started Machine Details', and 'Functional machines'. Action buttons 'Load Configuration', 'Save', and 'Save As' are located in the top right. The form contains the following fields:

Account Name	AWS Access Key	AWS Secret Key	Region
			US East (Ohio) v
AMI	Instance Type	Key Pair	Security Group ID
	t2.micro v		
VPC ID	Subnet ID	AIQ Protocol	AIQ Port
optional	optional	HTTP v	8080
Ec2 starting time out (mins)			
25			

Sometimes you need to be able to encrypt files in an automated way using CURL or service suite files to rotate the AWS credentials. Here we list two ways of achieving the same using AIQ Rest API calls.

EC2 Cloud config sample unencrypted structure

```
<?xml version="1.0" encoding="UTF-8"?> <cloud type="EC2" version="1"> <config-param
key="AIQProtocol" value="http"/>      <config-param key="AIQPort" value="8080"/>      <config-
param key="Ec2Timeout" value="25"/>      <config-param key="AccountName"
value="appvance"/>      <config-param key="AWSKey" value="DUMMYDUMMYDUMMYDUMMY"/>      <config-
param key="AWSSecretAccessKey" value="dummDUMMYDUMMYDUMMYDUMMYDUMMYDUMMYDUMMYDUMMY"/>      <config-param
key="AMI" value="ami-0dfc3da39bb6f571c"/>      <config-param key="InstanceType"
value="t3a.xlarge"/>      <config-param key="KeyPair" value="AppvanceOps"/>      <config-param
key="Security" value="sg-00d4ef05804c7a117"/>      <config-param key="AWSRegion" value="us-
west-1.ec2.amazonaws.com"/>      <config-param key="vpcId" value=""/>      <config-param
key="subnetId" value=""/> </cloud>
```

Using CURL

By issuing the following sequence of CURL commands in a Mac/ubuntu terminal you can grab the resulting encrypted from the resulting JSON response.

Please replace the AIQ controller DNS and the AIQ user credentials, as well as the actual XML text that you want to encrypt.

```
curl -c cookies.txt -X POST -d "username=appvance&password=appvance"
http://localhost:8080/AppvanceServer/rest/admin/loginForServices -H "ContentType:
application/x-www-form-urlencoded"curl -b cookies.txt -X POST -d "content=Replace here the xml
of the ec2 cloud file" http://localhost:8080/AppvanceServer/rest/file/encryptFileContent - H
"Content-Type: application/x-www-form-urlencoded"
```

For Windows OS, you must to modify `curl` to `curl.exe` in the above code.

The response for the second curl (`encryptFileContent`) will be similar to

```
{
  "data":
    "pma9QDLKQn10Mlkce6L4AHww8d/xS3yZuvdeUTPrnCv4s3kNSGzTGicu4tuCTvusxiUfNSVL7A6Z5Br3/lW248GS3j8WA05EF7w9sWogHyC5L2JLo/rHAJqb6qI/NGq4ZE9hy3sobAKkVq7sadFUH3Fg/Pz+0CD94J/PR5X9CbZ9mJbcctEh73+kqZLf8v2lrgCYibSs957WJUub0sPC3E9XDdgjT53y0GafA8xiIbHbDxaEL4WikoFqglzjLAvkY3h4u/Pcb412XjVNw0UCkx51ML294jJRJs7ZQgFYbunL0x0PhJ2R9xzbALjqIBXaiic9VvNmbgJX5RBdcWnw4js6/+wDFkTSHgHrht6KxmWUG7ASghNz8pujjf4IOZX2JGPerxqAfD++M5iLKf5f2buwYZiBQBFyNCW9F7H9Rwu+y2q2M73NZ6WIXzCmQfMeL+ZHHWSFTxV10eSPdeaXCtSDwWrY7eKWS0EvQYtPmKRDwiAnkInTDZSBDHn12WNnZTAY1x/izUdS1L7BldjV0m9X3jP41zIEiUuo6wXUqx5BhLT7a22lVQiSfIK4DZqENvjMDrnY0dHBLi2mX40kbALfsxrYjj3cm1PI3cVkdUs1skM0q0L8Zs0ZnmqkUTshoPoxjKvmN32KaN9Py6UdaoE0HBIekmR7K2cottKvbyBuPQlWSbU8/W2/bMWx03UDPtHDvYiVHLQax+XhuNe00p5Lfa19jWED3WAdFXxy7KpXJXh6dyx0dKtF1SI+F8IR4w40PJcG2kSSuE0yzbH0bjxXDKiFiozwMYSJrWQobqBHkdX/4V1nw2vUHjI6FASt3c3aWId417LLkZzEFQxiv3bfEAnSwkCTp0raw2jW/04aHpgpDwv0YqCtnHPZHN0tvk0I8Jd0Prf2fMnRIdzjj+8lWG0d1UtXPgIZPRdxXhippZA8Nejom0cj0sQt+NrW8YE4c6o/39JRW56e9M3Un6Kf/MWq1k4a62HQBChGHAF81+60Wc7HGzN+HSVPWPoRQv8Rgk+ULAcX+2U2m04w==",
  "success": true
}
```

Just grab the data contents in the above JSON and save it as a .cloud file name.

Using ServiceSuite

Here is a service suite script that will make the same calls seen above in the CURL example:

```
// MODIFY THIS TO POINT TO YOUR CONTROLLER
var $controller = "http://localhost:8080"
var $username = "appvance";
var $password = "appvance";
// THIS VARIABLE CAN BE USED TO CONSTRUCT THE EC2 CLOUD XML AS A STRING. PROVIDED BELOW SAMPLE DATA:
var ec2CloudText = '<?xml version="1.0" encoding="UTF-8"?> <cloud type="EC2" version="1"> <config-param key="AIQProtocol" value="http"/> <config-param key="AIQPort" value="8080"/> <config-param key="Ec2Timeout" value="25"/> <config-param key="AccountName" value="appvance"/> <config-param key="AWSKey" value="DUMMYOWGVPHYCUXYASDF"/> <config-param key="AWSSecretAccessKey" value="dummyxqmHB6IFeJ3KAar7zh3l7Id4AeQR8Uasdf"/> <config-param key="AMI" value="ami-0dfc3da39bb6f571c"/> <config-param key="InstanceType" value="t3a.xlarge"/> <config-param key="KeyPair" value="AppvanceOps"/> <config-param key="Security" value="sg-00d4ef05804c7a117"/> <config-param key="AWSRegion" value="us-west-1.ec2.amazonaws.com"/> <config-param key="vpcId" value=""/> <config-param key="subnetId" value=""/> </cloud>';

// LOGIN
var data = new FormData();
data.put("username", $username);
data.put("password", $password);
var resp = post($controller +
```

```
"/AppvanceServer/rest/admin/loginForServices",data);log(resp.getDataString());// ENCRYPT
DATAvar url = $controller + '/AppvanceServer/rest/file/encryptFileContent';var data = new
formData();data.put("content", ec2CloudText);var json = JSON.parse(post(url,
data).getDataString());log("Encryption success: "+json.success);var encryptedData =
json.data;log("Encrypted text:\n"+encryptedData);
```

After executing this script the encrypted contents are in the encryptedData variable. Then you can use writeText service suite command to write it to a file in your repository.

File

Options

Play Script

ServiceSuite_EncryptCloudFile.js

```

key="AMI" value="ami-0dfc3da39bb6f571c"/> <config-param key="InstanceType" value="t3a.xlarge"/> <config-param key="KeyPair" value="AppvanceOps"/>
<config-param key="Security" value="sg-00d4ef05804c7a117"/> <config-param key="AWSRegion" value="us-west-1.ec2.amazonaws.com"/> <config-param
key="vpcId" value=""/> <config-param key="subnetId" value=""/> </cloud> ;
8
9 // LOGIN
10 var data = new formData();
11 data.put("username",$username);
12 data.put("password",$password);
13 var resp = post($controller + "/AppvanceServer/rest/admin/loginForServices",data);
14 log(resp.getDataString());
15
16
17 // ENCRYPT DATA
18 var url = $controller + '/AppvanceServer/rest/file/encryptFileContent';
19 var data = new formData();
20 data.put("content", ec2CloudText);
21 var json = JSON.parse(post(url, data).getDataString());
22 log("Encryption success: "+json.success);
23 var encryptedData = json.data;
24 log("Encrypted text:\n"+encryptedData);

```

ServiceSuite_EncryptCloudFile.js (Done)

POST: http://localhost:8080/AppvanceServer/rest/admin/loginForServices. Request Header & Body. Response Header

{"logged": true}

POST: http://localhost:8080/AppvanceServer/rest/file/encryptFileContent. Request Header & Body. Response Header

Encryption success: true

Encrypted cloud:

pma9QDLKQn1OMlkce6L4AHwW8d/xS3yZuvdeUTPrnCv4s3kNSGzTGicu4tuCTvusxiUfNSVL7A6Z5Br3/1W248GS3j8WA05EF7w9sWogHyC5L2JLo/rHAJqb6qI/NGq4ZE9hy3sobAK
kVq7sadFUH3Fg/Pz+OCD94J/PR5X9CbZ9mJbcctEh73+kqZLf8v2lrgCYibSs957WJUub0sPC3E9XDdgjT53y0GafA8xiIbHbDxaEL4WikoFqglzjLAvkY3h4u/Pcb412XjVNwOUckx5
1ML294jJRJs7ZQgFYbunLOxOPhJ2R9xzbaLjqiBXaiaic9VvNmbgJX5RBdcWnw4js6/+wDFkTSHGhrht6KxmwUG7ASghNz8pujjf4IOZX2JGPerxqAFD++M5i1Kf5f2buwYz1BQBFyNCW
9F7H9Rwu+y2q2M73NZ6WixzCmQfMeL+ZHHWSFTxV1OeSPdeaXCtSDwWrY7eKWS0EvQYtPmKRDwiAnkInTDZSBDHn12WNnZTAY1x/izUdSL7BldjV0m9X3jP41zIEiUuo6wXUqx5BhLT
7a221VQiSfIK4DZqgENvJmDrnY0dHBLi2mX40kbALfsxrYjj3cm1PI3cVkdUs1skM0qOL8zsoZnmqkUTshoPoxjKvmN32KaN9Py6UdaoEOHBIEkmR7K2cottKvbyBuPQ1WSbU8/W2/bM
Wx03UDPtHDvYiVHLQax+XhuNeOOp5Lfai9jWED3WadFXxy7KpXJXh6dyx0dKtF1SI+F8IR4w4OPJcG2kSSuE0yZbH0bjxXDKiFiozwMYSJrWQobqBHkdX/4Vlnw2vUHji6FAST3c3aWI
d417LLkZzEFQxiv3bfEAnSwkCTp0raw2jW/O4aHpgpDwv0YgCtnHPZHN0tvkOI8Jd0Prf2fMnRIdzjj+81WG0d1UtXPgIZPRdxXhippZa8Nejom0cjosQt+Nrw8YE4c6o/39JRW56e9M
3Un6Kf/MWqlk4a62HQBChGAF81+60Wc7HGzN+HSVPWPoRQv8Rgk+ULAcX+2U2m04w==

Playback completed successfully.

Last updated: April 28, 2025

If you have any feedback on this topic, please contact us at documentation@appvance.ai

Revision #6

Created 2026-07-31 15:39:10 UTC by Admin

Updated 2026-10-14 17:59:08 UTC by mmora