

BitPath-DevKit

AWS IoT Core FleetHub

RADIOSTUDIO



1 Table of Contents

1	Table of Contents	2
2	Revision History	3
3	Requirement	4
4	Solution	5
5	Architecture	6
5.1	Workflow	6
5.2	Epics	7
5.3	To view the execution output	13
5.4	Subscribe to the below topics	14
6	Running IoT Jobs on multiple devices	15
6.1	Select the IoT devices	16
6.2	Publishing message to topics	17
6.3	Post-execution of the jobs	17
6.4	Job Status	18
7	Executing the subscriber program as a daemon	19
8	Sample deployment steps	20
9	Rollback of the deployment	22
10	Fleet Hub	24
11	References	26

2 Revision History

Version No	Date	Author	Change Log
1.0	29/04/2024	RadioStudio	

3 Requirement

Exploring the remote execution of commands on Raspberry Pi Devices via AWS IoT Core / Fleethub. Also, studying the remote device management

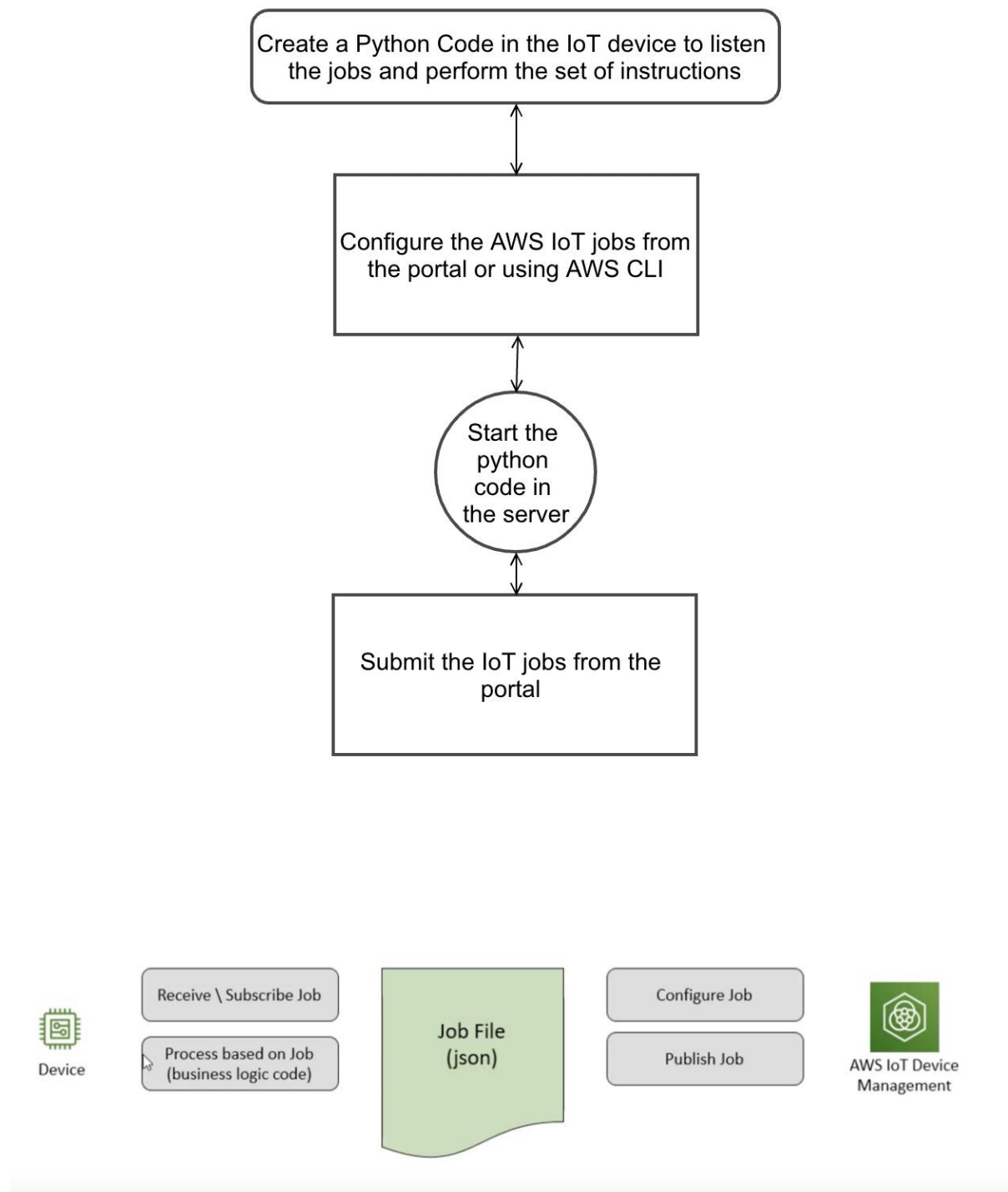
4 Solution

- **AWS IoT Jobs** - AWS IoT jobs can be used to define a set of remote operations sent to and executed on one or more devices connected to AWS IoT.
- **AWS Fleet Hub** - Fleet Hub is a web application that enables customers to manage their device fleets connected to AWS IoT.

Operation logic needs to be created in an IoT device. The job needs to be configured with instructions from the AWS IoT console, which the device can listen to and execute according to the instructions in the JSON file.

5 Architecture

5.1 Workflow



*Taken from a different website

5.2 Epics

1. Write a Python script - **joblistener-1.py** to connect with *AWS IoT jobs*

Code Snippet: The complete Python code is on the server under `/opt/iotcore-jobs/`. We are already running this code as a daemon.

Part I: First half of the code will read the device ID and the certificates installed on the device

```
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient
import time
import json
import requests
import os
import subprocess

#read the device id from the registration script output
with open('/opt/aws-iot-fleet-provisioning-v1/thingName.txt', 'r') as file:
    value = file.read().replace('\n', '')
deviceid = str(value)
print(deviceid)

topic = "bitpath/testing" #change the subscription topic as needed in the code and the AWS IoT subscription
dashboard.

myMQTTClient = AWSIoTMQTTClient(deviceid)
myMQTTClient.configureEndpoint("", 8883)
myMQTTClient.configureCredentials("/opt/certificate/amazon-root-ca.pem","/opt/certificate/3a1e966d47-
private.pem.key", "/opt/certificate/3a1e966d47-certificate.pem.crt")
myMQTTClient.connect()
print("Client Connected")
```

Part - II: This will read the IoT Job Operation Type and execute the jobs accordingly

```
if operationType == "install":
    print ("Job Execution Started")
    url = payload['execution']['jobDocument']['files']['url']
    r = requests.get(url)
    with open ('/opt/iotcore-jobs/install.py', 'wb') as f:

        f.write(r.content)

    output = subprocess.check_output('python /opt/iotcore-jobs/install.py',shell=True)
    message = str(output)
    myMQTTClient.publish(topic, message, 0)
    updateJobStatus(jobId,"SUCCEEDED")
```

```

if operationType == "uninstallation":
    print ("Job Execution Started")
    url = payload['execution']['jobDocument']['files']['url']
    r = requests.get(url)
    with open ('/opt/iotcore-jobs/rollback.py', 'wb') as f:

        f.write(r.content)

    output = subprocess.check_output('python /opt/iotcore-jobs/rollback.py',shell=True)
    message = str(output)
    myMQTTClient.publish(topic, message, 0)
    updateJobStatus(jobId,"SUCCEEDED")

```

2. Create an IoT job template in JSON format. You can name it according to the operation. I have tagged install-01.json and kept it in the AWS S3 bucket[us-east-1] named S3: *bitpath-jobs-demo*.

```

{
  "operation": "install",
  "version": "1.0",
  "packageName": "install.py",
  "autoStart": "true",
  "workingDirectory": "/opt/iotcore-jobs",
  "files": {
    "fileName": "install.py",
    "url": "${aws:iot:s3-presigned-url:https://s3.us-east-1.amazonaws.com/bitpath-jobs-demo/install.py}"
  }
}

```

3. Now, as described in the JSON, package name install.py [In this file, we will write the commands that need to be executed on the device to install/download software, etc.]

Create a python file *install.py* with the below sample code and upload it to the same S3 bucket folder; here, this code will print the system time; we can change it as per our requirement, like Github Repo clone and deploy - hardware update, etc.[Discussed in sample deployment steps]


```
import os
cmd = 'uptime'
os.system(cmd)

print("A Sample file. Downloaded to Device and Executed !")
```

4. Create IoT job from AWS Console: Now, we will go back to the AWS IoT core portal and configure an IoT Job: Name of “*Job and description*” Click on Next

The screenshot shows the AWS IoT console interface for creating a custom job. The breadcrumb navigation at the top reads: AWS IoT > Manage > Jobs > Create job > Create custom job. On the left, a sidebar lists three steps: Step 1: Custom job properties (active), Step 2: File configuration, and Step 3: Job configuration. The main content area is titled 'Custom job properties' with an 'Info' link. It contains a 'Job properties' section with a 'Name' field (placeholder: Job_name) and a 'Description - optional' field (placeholder: Description). A note states: 'Enter a unique name that contains only alphanumeric characters, hyphens, or underscores. Job names can't contain any spaces.' Another note states: 'The description can have up to 2,028 characters.' Below these fields is a 'Tags - optional' section with a right-pointing arrow. At the bottom right, there are 'Cancel' and 'Next' buttons.

5. Deploy the jobs: Select the device where you want to execute the command; right now, I have only limited devices, so I have selected “*things*” to run this job. In case we have multiple things/device, we can use “*Thing Groups*” :

AWS IoT > Manage > Jobs > Create job > Create custom job

Step 1
Custom job properties

Step 2
File configuration

Step 3
Job configuration

File configuration

Job targets Info
A custom job is a remote operation that is sent to and runs on one or more devices connected to AWS IoT. Job targets are the things and thing groups that represent the devices that should run this job.

Things to run this job
Choose existing things
bithpath-dev1634203379290 X

Thing groups to run this job
Choose existing thing groups

File Info
The job file defines the operation to perform on the job targets. For additional security, files can contain a presigned URL, or code signing placeholders.

Job file
A JSON file to upload to S3.

S3 URL
s3://bithpath-jobs-demo/install-01.jsor X View Browse S3

Pre-sign resource URLs
For an extra layer of security, you can pre-sign URLs that refer to resources in your job file, like a binary for a firmware update.

Pre-signing role
Choose an IAM role Clear Create role

Timeout
The duration of a pre-signing URL is available from 1 minute to 1 hour.

Minutes	Seconds
1	0
1-60	0-59

At the bottom, we must specify the JSON job file we created in step 2. We don't need to manually pre-sign the URL; it's automated as part of the AWS process. Ensure the IAM role is permitted to "GetObject from the S3 bucket you have created." Here, we are using RoleName - bitpath-dev-fleet-role. Enter the signed URL expiration time info.

File Info
The job file defines the operation to perform on the job targets. For additional security, files can contain a presigned URL, or code signing placeholders.

Job file
A JSON file to upload to S3.

S3 URL
s3://bithpath-jobs-demo/install-01.jsor X View Browse S3

Pre-sign resource URLs
For an extra layer of security, you can pre-sign URLs that refer to resources in your job file, like a binary for a firmware update.

Pre-signing role
bitpath-dev-fleet-role Clear Create role

Timeout
The duration of a pre-signing URL is available from 1 minute to 1 hour.

Minutes	Seconds
5	0
1-60	0-59

Cancel Previous Next

- Click on next, and we will get options to execute these jobs; it could be one time, or it can be configured to run continuously after adding the new devices. I have selected "snapshot" and kept the default for only the rest. [AWS Documentation reference for [Jobs](#)]

Job configuration [Info](#)

Job configuration

Job run type

Configure how your job will deploy to the job targets.

☒ **Snapshot**
Your job completes after it deploys to all of the job targets.

☐ **Continuous**
Your job continues to deploy to devices that are added to the thing groups in the job target list.

Additional configurations - optional

- ▶ Rollout configuration
- ▶ Job executions timeout configuration
- ▶ Abort configuration

[Cancel](#) [Previous](#) [Submit](#)

7. Before clicking Submit, ensure the process runs on the device created in step 1.[Ignore this step, as it's automatically running as a daemon].

8. Continue step 6 and submit the Job from the AWS portal - The Status will turn "in progress".

The screenshot shows the AWS IoT FleetHub console for a job named 'job-demo-1'. The job status is 'In progress'. The console displays job details, rollout configuration, and a table of job executions.

Thing name	Last updated date	Queued at	Status
bithpath-dev1634203379290	October 16, 2021, 14:20:45 (UTC+0530)	October 16, 2021, 14:20:44 (UTC+0530)	In progress

```
root@bitpath-dev-xx:/opt/jobs-testing# python3 joblistener2.py
Client Connected
Device Listening Started
Job Completed !
```

Verify your device terminal that the command has executed successfully, and you will be able to view the system time and the message as per the testconfig.py

```
root@bitpath-dev-xx:/opt/jobs-testing# python3 joblistener2.py
Client Connected
Device Listening Started
Job Completed !
Job Execution Started
09:50:46 up 1 day, 15:18, 1 user, load average: 0.02, 0.08, 0.08
A Sample file. Downloaded to Device and Executed !
```

Job-status also has switched from in progress to complete once completed.

job-demo-1 [Info](#)

[Refresh](#) [Edit](#) [Save as a job template](#) [Cancel](#) [Delete](#)

Job details

Job name job-demo-1 ARN arn:aws:iotus-east-1:685943795297:job/job-demo-1 Description -	Last updated October 16, 2021, 14:20:48 (UTC+0530) Created October 16, 2021, 14:20:43 (UTC+0530) Status Completed	Devices to update 1 thing Job run type SNAPSHOT Timeout configuration -
--	---	---

Rollout configuration

The rate at which the job is deployed to the target devices.

Abort configuration

If the number of job failures reaches the threshold percentage, and the job has deployed to the minimum number of devices, the job will cancel.

Job executions | [Job document](#) | [Job targets](#) | [Tags](#)

Job executions (1) [Info](#)

Devices currently processing this job.

[All job executions \(1\)](#) [Refresh](#) [Cancel](#) [Delete](#)

<input type="checkbox"/>	Thing name	Last updated date	Queued at	Status
<input type="checkbox"/>	bitpath-dev1634203379290	October 16, 2021, 14:20:46 (UTC+0530)	October 16, 2021, 14:20:44 (UTC+0530)	Succeeded

5.3 To view the execution output

To check the Inbound/Outbound request, please subscribe below topics from the Device

AWS IoT

- Monitor
- Activity
- Connect
- Manage
- Fleet Hub
- Greengrass
- Wireless connectivity
- Secure
- Defend
- Act
- Test
 - Device Advisor
 - MQTT test client

communicate their state to AWS IoT. AWS IoT Core provides a set of topics and publish MQTT messages to topics.

Subscribe to a topic

Topic filter [Info](#)
 The topic filter describes the topic(s) to which you want to subscribe.

Additional configuration

Subscriptions
Favorites

- \$aws/events/job/bit-test/canceled
- \$aws/events/#
- \$aws/things/#

5.4 Subscribe to the below topics

1. Subscribe to a topic
2. Subscription topic: \$aws/events/#
3. Subscribe
4. Subscribe to a topic
5. Subscription topic: \$aws/things/#
6. Subscribe
7. Subscribe to a topic
8. Subscription topic: bitpath/testing
9. Subscribe to topic: #: In case no topic is selected to view all the events.

Subscribe to a topic

Publish to a topic

Topic filter [Info](#)

The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.

► Additional configuration

Subscribe

Subscriptions

Favorites

\$aws/events/job/bit-test/canceled

\$aws/events/#

\$aws/things/#

bitpath/testing

general/outbound

All subscriptions

bitpath/testing

Pause

Clear

Export

Edit

▼ bitpath/testing

October 20, 2021, 14:17:58 (UTC+0530)

⊗ Message cannot be displayed in specified format.

b'

m:

99

total

7875

0

used

115

99\nA

free

7446

Sample file. Downloaded to Device and Executed !\n'

shared

8

buff/cache

313

available\nMe

7521\nSwap:

6 Running IoT Jobs on multiple devices

Starting the listener scripts in both devices [In our golden image, it's automated. We can ignore this step]:

```
root@bitpath-dev-xx:/opt/jobs-testing# python3 joblistener4.py
bithpath-dev1634203379290
Client Connected
Device Listening Started
```

```
root@ip-172-31-71-169:/opt/jobs-testing# python3 joblistener1.py
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.26.7) or chardet (3.0.4) doesn't match a supported version!
RequestsDependencyWarning)
Client Connected
Device Listening Started
Job Completed !
```

From the AWS IoT core Manage Jobs, we will trigger jobs to both devices:

The screenshot shows the 'Things (2)' page in the AWS IoT Core console. It includes a search bar and a table of devices. The table has columns for 'Name' and 'Thing type'. Two devices are listed: 'cloud-9-test' and 'bithpath-dev1634203379290', both with a 'Thing type' of '-'.

Name	Thing type
cloud-9-test	-
bithpath-dev1634203379290	-

For the same steps, click on Create Job and enter the required details.

The screenshot shows the 'Custom job properties' form. It has a 'Name' field with the value 'multi-device-testing' and a 'Description - optional' field with the value 'testing on two device checking free memory'. There are 'Cancel' and 'Next' buttons at the bottom right.

Job properties

Name
multi-device-testing
Enter a unique name that contains only alphanumeric characters, hyphens, or underscores. Job names can't contain any spaces.

Description - optional
testing on two device checking free memory
The description can have up to 2,028 characters.

► **Tags - optional**

Cancel Next

6.1 Select the IoT devices

Job targets [Info](#)

A custom job is a remote operation that is sent to and runs on one or more devices connected to AWS IoT. Job targets are the things and thing groups that represent the devices that should run this job.

Things to run this job

Choose existing things ▼

cloud-9-test ✕ bithpath-dev1634203379290 ✕

Thing groups to run this job

Choose existing thing groups ▼

File [Info](#)

The job file defines the operation to perform on the job targets. For additional security, files can contain a presigned URL, or code signing placeholders.

Job file

A JSON file to upload to S3.

S3 URL

✕

View [↗](#)

Browse S3

Pre-sign resource URLs

For an extra layer of security, you can pre-sign URLs that refer to resources in your job file, like a binary for a firmware update.

Pre-signing role

bitpath-dev-fleet-role ▼

Clear

Create role

Timeout

The duration of a pre-signing URL is available from 1 minute to 1 hour.

Minutes

Seconds

1-60

0-59

Cancel

Previous

Next

6.2 Publishing message to topics

As discussed above, we must subscribe to topics to view the logs. On job completion, we can view the output to the respective subscribed topics:

Here, we have subscribed to the following topics:

- 1. bitpath/testing
- 2. general/inbound

Subscriptions

Favorites

\$aws/events/job/bit-test/canceled

#

bitpath/testing

\$aws/events/#

\$aws/things/#

general/inbound

general/inbound

Pause

Clear

Export

Edit

6.3 Post-execution of the jobs

Cloud-9-test: Output from device:

```
RequestsDependencyWarning)
Client Connected
Device Listening Started
Job Completed !

Job Execution Started
b' total used free shared buff/cache available\nMem: 1949 562 258 2 1128 1203\nSwap: 488 0 488\nA Sample
file. Downloaded to Device and Executed !\n'
```

Bithpath-dev1634203379290 output from the device:

```
bithpath-dev1634203379290
Client Connected
Device Listening Started
Job Completed !

bithpath-dev1634203379290
bithpath-dev1634203379290
Job Execution Started
b' total used free shared buff/cache available\nMem: 7875 115 7412 8 347 7517\nSwap: 99 0
99\nA Sample file. Downloaded to Device and Executed !\n'
```

Same output from the IoT subscription:
bitpath/testing

Subscriptions

Favorites

\$aws/events/job/bit-test/canceled

#

bitpath/testing

\$aws/events/#

bitpath/testing

October 21, 2021, 11:03:27 (UTC+0530)

Message cannot be displayed in specified format.

b'	total	used	free	shared	buff/cache	available	nMem:	7875	115	7412
8	347	7517	99	0	99	nA	Sample file. Downloaded to Device and Executed !\n'			

general/inbound

Subscriptions

Favorites

\$aws/events/job/bit-test/canceled

#

bitpath/testing

\$aws/events/#

\$aws/things/#

general/inbound

general/inbound

October 21, 2021, 11:03:25 (UTC+0530)

Message cannot be displayed in specified format.

b'	total	used	free	shared	buff/cache	available	nMem:	1949	562	258
2	1128	1203	488	0	488	nA	Sample file. Downloaded to Device and Executed !\n'			

6.4 Job Status

Job details

Job name

multi-device-testing

ARN

arn:aws:iotus-east-1:685943795297:job/multi-device-testing

Description

testing on two device checking free memory

Last updated

October 21, 2021, 11:03:30 (UTC+0530)

Created

October 21, 2021, 11:03:22 (UTC+0530)

Status

Completed

Devices to update

2 things

Job run type

SNAPSHOT

Timeout configuration

-

Rollout configuration

The rate at which the job is deployed to the target devices.

Abort configuration

If the number of job failures reaches the threshold percentage, and the job has deployed to the minimum number of devices, the job will cancel.

Job executions

Job document

Job targets

Tags

Job executions (2)

Find job executions

All job executions (2)

< 1 >

	Thing name	Last updated date	Queued at	Status
<input type="checkbox"/>	bithpath-dev1634203379290	October 21, 2021, 11:03:28 (UTC+0530)	October 21, 2021, 11:03:25 (UTC+0530)	Succeeded
<input type="checkbox"/>	cloud-9-test	October 21, 2021, 11:03:25 (UTC+0530)	October 21, 2021, 11:03:25 (UTC+0530)	Succeeded

7 Executing the subscriber program as a daemon

Multiple ways to execute the jobs in the backend, like crontab and .bashrc. In this case, we have used the default OS service, **systemd**.

Systemd should be installed on all the machines, in case if it's missing, install it using the below command :

```
sudo apt-get install -y systemd
```

Once, installed create a file under : /etc/systemd/system/<NameOftheService>

```
-rw-r--r-- 1 root root 128 Oct 23 05:19 iot-services.service
root@bitpath-dev-01:/etc/systemd/system#
```

We have kept it as iot-services.service.

With very minimal configuration, to start the service

```
root@bitpath-dev-01:/etc/systemd/system# cat iot-services.service
[Unit]
Description=My IoT job
[Service]
Type=simple
Restart=always
ExecStart=/usr/bin/python3 /opt/jobs-testing/joblistener4.py
```

Once configured, reload the daemon with `sudo systemctl daemon-reload` and start the service with `sudo systemctl start iot-services.service`, which will start the listener service in the background.

8 Sample deployment steps

Deploying a sample Python code and installing dependencies on devices:

1. Cloning of a Sample Python code:

First, change the install.py as per the requirement[refer to step 3 code]; here, we are cloning a public repo on the device and installing the packages.

```
import os

clone1 = 'git clone https://github.com/stevemar/sample-python-app.git'
executer= 'cd sample-python-app && pip install -r requirements.txt'

os.system(clone1)
os.system(executer)

print("A Sample file. Downloaded to Device and Executed !")
```

Once the modification is completed, save the file and upload the file in the AWS S3 bucket.

2. IoT job configuration file: The install-01.json script does not change, as the filename is the same.

```
{
  "operation" : "install",
  "version" : "1.0",
  "packageName" : "install.py",
  "autoStart" : "true",
  "workingDirectory" : "/opt/jobs-testing",
  "files" : {
    "fileName" : "install.py",
    "url" : "${aws:iot:s3-presigned-url:https://s3.us-east-1.amazonaws.com/bitpath-jobs-demo/install..py}"
  }
}
```

3. Deploying to a group :

- a. [Bithpath-dev-2nd](#)
- b. [Cloud-9-test](#)

As a prerequisite, we already run Python listener code as a background process on all devices. Now, we will create an IoT job from the AWS Console or CLI and execute it—refer to Epics: 4th point.

bitpath-dev-location

EditDelete

Thing group details

Name

bitpath-dev-location

ARN

arn:aws:iot:us-east-1:685943795297:thinggroup/bitpath-dev-location

Parent group

-

Description

a group of devices with US as location

Date created

September 29, 2021, 18:15:40 (UTC+0530)

Type

Static

AttributesChild groupsThingsPoliciesJobsSecurity profilesTags

Things (3)

info

Remove thingsAdd things

Filter things

☐

Name

▲

☐

bitpath-dev-2nd

☐

bitpath-dev1634203379290

☐

cloud-9-test

Output

To view the Output, go to the MQTT test client:

Subscriptions

bitpath/testing

PauseClearExportEdit

Favorites

\$aws/events/job/bit-test/canceled

×

\$aws/events/#

×

\$aws/things/#

×

bitpath/testing

×

#

×

general/inbound

×

All subscriptions

▼ bitpath/testing

October 25, 2021, 23:12:53 (UTC+0530)

Message cannot be displayed in specified format.

b'Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple\nRequirement already satisfied: flask==1.1.1 in /usr/local/lib/python3.7/dist-packages (from -r requirements.txt (line 6)) (1.1.1)\nRequirement already satisfied: click>=5.1 in /usr/lib/python3/dist-packages (from flask==1.1.1->-r requirements.txt (line 6)) (7.0)\nRequirement already satisfied: Werkzeug>=0.15 in /usr/local/lib/python3.7/dist-packages (from flask==1.1.1->-r requirements.txt (line 6)) (2.0.2)\nRequirement already satisfied: itsdangerous>=0.24 in /usr/lib/python3/dist-packages (from flask==1.1.1->-r requirements.txt (line 6)) (0.24)\nRequirement already satisfied: Jinja2>=2.10.1 in /usr/local/lib/python3.7/dist-packages (from flask==1.1.1->-r requirements.txt (line 6)) (3.0.2)\nRequirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.7/dist-packages (from Jinja2>=2.10.1->flask==1.1.1->-r requirements.txt (line 6)) (2.0.1)\n\nSample file. Downloaded to Device and Executed !\n'

▼ bitpath/testing

October 25, 2021, 23:12:52 (UTC+0530)

Message cannot be displayed in specified format.

b'Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple\nRequirement already satisfied: flask==1.1.1 in /usr/local/lib/python2.7/dist-packages (from -r requirements.txt (line 6)) (1.1.1)\nRequirement already satisfied: Jinja2>=2.10.1 in /usr/local/lib/python2.7/dist-packages (from flask==1.1.1->-r requirements.txt (line 6)) (2.11.3)\nRequirement already satisfied: itsdangerous>=0.24 in /usr/local/lib/python2.7/dist-packages (from flask==1.1.1->-r requirements.txt (line 6)) (1.1.0)\nRequirement already satisfied: Werkzeug>=0.15 in /usr/local/lib/python2.7/dist-packages (from flask==1.1.1->-r requirements.txt (line 6)) (1.0.1)\nRequirement already satisfied: click>=5.1 in /usr/local/lib/python2.7/dist-packages (from flask==1.1.1->-r requirements.txt (line 6)) (7.1.2)\nRequirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python2.7/dist-packages (from Jinja2>=2.10.1->flask==1.1.1->-r requirements.txt (line 6)) (2.0.1)\n\nSample file. Downloaded to Device and Executed !\n'

RadioStudio

1.0 and 29/04/2024

Page 21 of 26

9 Rollback of the deployment

In case of deployment issues, we can roll back similarly.

1. Create a rollback Python script; here, we have created a sample rollback Python script—rollback.py [For this demo, no modification required] script in S3 and mentioned the commands that must be executed.
2. For uninstallation, we will use a different JSON file name as uninstall-01.json

Custom job properties Info

Job properties

Name

Enter a unique name that contains only alphanumeric characters, hyphens, or underscores. Job names can't contain any spaces.

Description - optional

Description

The description can have up to 2,028 characters.

► **Tags - optional**

Cancel Next

3. Select the device:

Job targets Info

A custom job is a remote operation that is sent to and runs on one or more devices connected to AWS IoT. Job targets are the things and thing groups that represent the devices that should run this job.

Things to run this job

Thing groups to run this job

File Info

The job file defines the operation to perform on the job targets. For additional security, files can contain a presigned URL, or code signing placeholders.

Job file

A JSON file to upload to S3.

S3 URL

View Browse S3

Pre-sign resource URLs

For an extra layer of security, you can pre-sign URLs that refer to resources in your job file, like a binary for a firmware update.

Pre-signing role

Clear Create role

Timeout

The duration of a pre-signing URL is available from 1 minute to 1 hour.

Minutes **Seconds**

1-60 0-59

Cancel Previous Next

Rollback of the deployment successful :

Outputs

Device1:

Subscriptions	general/inbound	Pause	Clear	Export	Edit
Favorites					
\$aws/events/job/bit-test/canceled					
\$aws/events/#					
\$aws/things/#					
bitpath/testing					
#					
general/inbound					

▼ general/inbound

October 26, 2021, 11:55:07 (UTC+0530)

Message cannot be displayed in specified format.

b'Uninstalling Flask-1.1.1:\n Successfully uninstalled Flask-1.1.1\nA Sample file. Downloaded to Device and Rollbacked the inst
allation !\n'

▼ general/inbound

October 26, 2021, 11:55:07 (UTC+0530)

Message cannot be displayed in specified format.

b'Uninstalling Flask-1.1.1:\n Successfully uninstalled Flask-1.1.1\nA Sample file. Downloaded to Device and Rollbacked the inst
allation !\n'

Device2:

Subscriptions	bitpath/testing	Pause	Clear	Export	Edit
Favorites					
\$aws/events/job/bit-test/canceled					
\$aws/events/#					
\$aws/things/#					
bitpath/testing					
#					
general/inbound					
All subscriptions					

▼ bitpath/testing

October 26, 2021, 11:55:10 (UTC+0530)

Message cannot be displayed in specified format.

b'Uninstalling Flask-1.1.1:\n Successfully uninstalled Flask-1.1.1\nA Sample file. Downloaded to Device and Rollbacked the inst
allation !\n'

▼ bitpath/testing

October 26, 2021, 11:55:10 (UTC+0530)

Message cannot be displayed in specified format.

b'Uninstalling Flask-1.1.1:\n Successfully uninstalled Flask-1.1.1\nA Sample file. Downloaded to Device and Rollbacked the inst
allation !\n'

10 Fleet Hub

The diagram below shows the process of using the fleet hub.

Applications
Fleet Hub web applications allow users to access your IoT data outside the AWS management console.

▼ How it works

Step 1. Create Fleet Hub application.
Create a Fleet Hub application to monitor the status of your device fleet. Choose [create application](#) to start.

Step 2. Add users to Fleet Hub application.
Add users to your Fleet Hub application at any time to give your users access. Click on each application detail page to add SSO users.

Step 3. Manage dependencies
Sync your thing groups, fleet indexing searchable fields, and job templates to expand Fleet Hub features. Head to the [related services](#) page to learn more.

Single Sign-On MFA devices | Sign out

Search

AWS IoT Fleet Hub
demo fleet

Terms of Use Powered by aws

Once we have set up the application and added the SSO user, the user will get a signup request in his email box and a URL to manage all the IoT devices. This portal will have all the information, including a device list, job status, etc., and also give permissions to execute the above jobs that we have just executed.

demo fleet ×

Dashboards

Device groups

Jobs

Fleet Hub alarms

English (US) EN

Logout

demo fleet > Dashboard US East (N. Virginia) ▼

Dashboard

All fields Search by values Search Filter

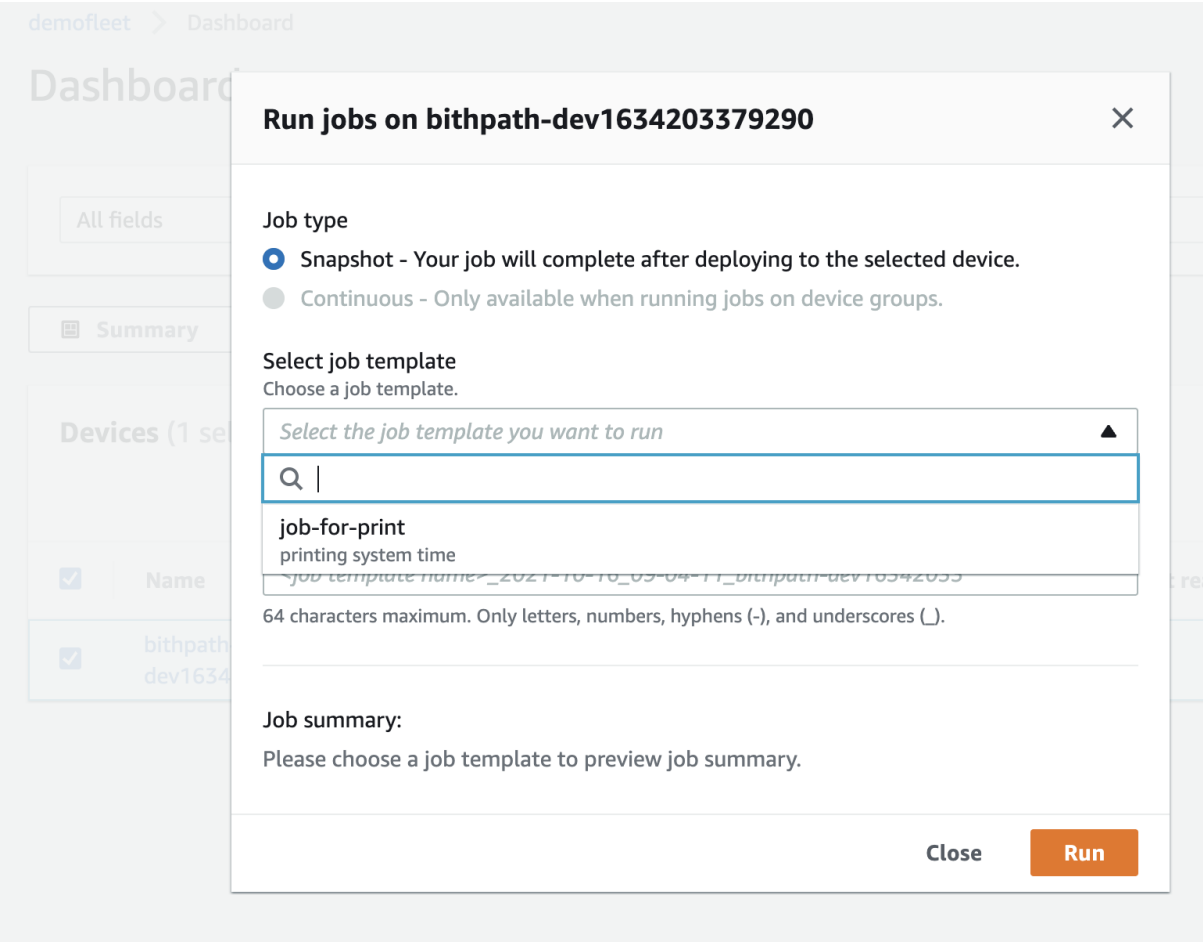
Summary Device list Create alarm

Devices (1) Export current page Run jobs

	Name	Connected	Last connection timestamp	Disconnect reason	Thing type	Thing groups
<input type="checkbox"/>	bithpath-dev1634203379290	✓ true	October 16, 2021 06:55 (UTC)	-	-	bithpath-dev-location

Privacy | Site terms | Cookie preferences | © 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Users can view the device list in the same portal and run jobs by selecting the device. [Note: to use jobs, the admin must create the job template first. Only then can the user choose and perform the operations.



Jobs	Job name	Job status	Target	Job type	Last update	Created at
Fleet Hub alarms	job-demo-1	Completed	1 device	Snapshot	October 16, 2021 08:50 (UTC)	October 16, 2021 08:50 (UTC)
	job-for-print_2021-10-16-06-56-23_bithpath-dev16342033	Completed	1 device	Snapshot	October 16, 2021 06:57 (UTC)	October 16, 2021 06:56 (UTC)
	job-for-print_2021-10-16-06-53-44_bithpath-dev16342033	Canceled	1 device	Snapshot	October 16, 2021 06:57 (UTC)	October 16, 2021 06:53 (UTC)

11 References