


Services ▾

Resource Groups ▾



AppDevRole ▾

N. Virginia ▾

Support ▾

AWS Cloud9

×

Your environments

Shared with you

Account environments

How-to guide

AWS Cloud9 > Your environments

Your environments (1)

Open IDE

View details

Edit

Delete

Create environment

< 1 > ⚙

workshop-environment

Type

EC2

Permissions

Owner

Description

Cloud9 environment for the crypto builders python modules

Open IDE

Step 1:

- **Navigate to Cloud9 service**
- **Open the environment workshop-environment**
- **If you don't find the environment, you might not be in the right region for your workshop – this slide shows N.Virginia. Depending on where this workshop is being conducted the region might be different**

Feedback

English (US)

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

 README.md

```
bash - "ip-10-0-0 ×
```

```
admin:~/environment $
```

- **Notice that the Common Name is set in the CSR**

Go to Anything (% P)

Environment

Source Control

AWS

workshop-environment - /home/

data-protection

cf-templates

docs

images

site

usecase-1

usecase-4

usecase-5

usecase-9

cf-templates

code

templates

email_signing_cert.pem

smime-encrypt-cert.py

smimecerts-sign-cert.py

templates.py

environment-setup.sh

LICENSE

ok

README.md

setup.log

usecase-8

cf-templates

files

environment-setup.sh

LICENSE

README.md

CODE_OF_CONDUCT.md

CONTRIBUTING.md

environment-setup.sh

LICENSE

mkdocs.yml

smimecerts-sign ×

```
80 .subject_name(subject_name_csr)
81 .add_extension(basic_constraints, True)
82 .sign(csr_webserver_privkey, hashes.SHA256(), default_backend())
83 )
84
85 csr_pem = csr.public_bytes(encoding=serialization.Encoding.PEM)
86
87 response = acm_pca_client.issue_certificate(
88     ApiPassthrough={
89         'Extensions': {
90             'ExtendedKeyUsage': [
91                 {
92                     'ExtendedKeyUsageType': 'EMAIL_PROTECTION'
93                 },
94             ],
95             'KeyUsage': {
96                 'DigitalSignature': True,
97                 'NonRepudiation': True,
98             },
99             'SubjectAlternativeNames': [
100                 {
101                     'Rfc822Name': 'myemail@domain.com',
102                 },
103             ],
104         },
105         'Subject': {
106             'Country': 'US',
107             'Organization': 'customer',
108             'OrganizationalUnit': 'customerdept',
109             'State': 'Nevada',
110             'CommonName': 'S/MIME',
111         },
112     },
113 )
```

104:23 Python Spaces: 4

data-protection/u ×

Run

Command: data-protection/usecase-7/templates/smimecerts-sign-cert.py

Runner: Python 3 CWD ENV

This step will take about 2 minutes to complete

Successfully created S/MIME signing certificate called email_sign_cert.pem

Process exited with code: 0

Step 3:

- Scroll down to line 90
- Look through
 - ExtendedKeyUsage
 - KeyUsage
 - SubjectAlternativeNames
- Notice KeyUsage is using: *DigitalSignature* and *NonRepudiation*
- This happens via [API Passthrough](#)

AWS Cloud9 File Edit Find View Go Run Tools Window Support Preview **Run** A Share

Go to Anything (% P)

Environment

- workshop-environment - /home, ⚙
- data-protection
 - cf-templates
 - docs
 - images
 - site
 - usecase-1
 - usecase-4
 - usecase-5
 - usecase-9
 - cf-templates
 - code
 - templates
 - email_signing_cert.pem
 - smime-encrypt-cert.py
 - smimecerts-sign-cert.py
 - templates.py
 - environment-setup.sh
 - LICENSE
 - ok
 - README.md
 - setup.log
- usecase-8
 - cf-templates
 - files
 - environment-setup.sh
 - LICENSE
 - README.md
 - CODE_OF_CONDUCT.md
 - CONTRIBUTING.md
 - environment-setup.sh
 - LICENSE
 - mkdocs.yml

Source Control

AWS

```
1 #####
2 # Learning about using templates #
3 #####
4
5 from datetime import datetime
6 from datetime import timedelta
7 import os
8 import subprocess
9 import sys
10 import random
11 import time
12 import boto3
13
14 from cryptography import x509
15 from cryptography.hazmat.primitives import hashes
16 from cryptography.hazmat.backends import default_backend
17 from cryptography.hazmat.primitives import serialization
18 from cryptography.hazmat.primitives.asymmetric import rsa
19 from cryptography.x509.oid import NameOID
20 from cryptography.hazmat.primitives.serialization import NoEncryption
21
22 def main():
23     """
24     #####
25     # Creating an S/MIME signing certificate #
26     #####
27     """
28     try:
29         acm_pca_client = boto3.client('acm-pca')
30
31         current_directory_path = os.path.dirname(os.path.realpath(__file__)) + '/'
```

97:52 Python Spaces: 4 ⚙

data-protection/u × +

Run Command: data-protection/usecase-7/templates/smimecerts-sign-cert.py Runner: Python 3 CWD ENV

This step will take about 2 minutes to complete

Successfully created S/MIME signing certificate called email_sign_cert.pem

Process exited with code: 0

Step 4:

- Run the python script smime-sign-cert.py by clicking on the Run button
- After about 2 mins you should see the print

“Successfully created ...”

Go to Anything (% P)

EnvironmentSource ControlAWS

workshop-environment - /home

data-protection

cf-templates

docs

images

site

usecase-1

usecase-4

usecase-5

usecase-9

cf-templates

code

templates

codesigning_cert.pem

email_cert.pem

smimecerts.py

templates.py

environment-setup.sh

LICENSE

ok

README.md

setup.log

usecase-8

cf-templates

files

environment-setup.sh

LICENSE

README.md

CODE_OF_CONDUCT.md

CONTRIBUTING.md

smimecerts.py

```
2 # Learning about
3 #####
4
5 from datetime import
6 from datetime import
7 import os
8 import subprocess
9 import sys
10 import random
11 import time
12 import boto3
13
14 from cryptography
15 from cryptography
16 from cryptography
17 from cryptography.hazmat.primitives import serialization
18 from cryptography.hazmat.primitives.asymmetric import rsa
19 from cryptography.x509.oid import NameOID
20 from cryptography.hazmat.primitives.serialization import NoEncryption
21
22 def main():
23     """
24     #####
25     # Creating an S/MIME certificate
26     #####
27     """
28     try:
```

New File ^ N

New Terminal **⌘ T**

New Run Configuration

New Immediate Window

Output

Open Preferences ⌘ ,

Recently Closed Tabs

Preferences

Welcome

environment-setup.sh

templates.py

ashes

fault_backend

serialization

import rsa

import NameOID

import NoEncryption

21:1 Python Spaces: 4

Run data-protection/u

Command: data-protection/usecase-7/templates/smimecerts.py

Runner: Python 3 CWD ENV

This step will take about 2 minutes to complete

Successfully created code signing cert email_cert.pem

Process exited with code: 0

Step 5:

- Open a new bash terminal

Step 6:

- Change directory to templates

cd data-protection/usecase-9/templates/

- Run the following command

openssl x509 -in email_signing_cert.pem -text -noout

The screenshot displays a code editor interface with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure with a 'data-protection' directory containing 'usecase-9' and a 'templates' subdirectory. The 'smimecerts.py' file is selected in the 'templates' directory. The code editor shows the beginning of the 'smimecerts.py' file, which includes imports for datetime, os, subprocess, sys, random, time, boto3, cryptography, and NameOID. A context menu is open over the code, showing options like 'New File', 'New Terminal', and 'New Run Configuration'. The terminal window at the bottom shows the command 'data-protection/usecase-7/templates/smimecerts.py' being executed, with the output 'This step will take about 2 minutes to complete' and 'Successfully created code signing cert email_cert.pem'. The terminal also shows 'Process exited with code: 0'.

```
2 # Learning about
3 #####
4
5 from datetime import
6 from datetime import
7 import os
8 import subprocess
9 import sys
10 import random
11 import time
12 import boto3
13
14 from cryptography
15 from cryptography
16 from cryptography
17 from cryptography.hazmat.primitives import serialization
18 from cryptography.hazmat.primitives.asymmetric import rsa
19 from cryptography.x509.oid import NameOID
20 from cryptography.hazmat.primitives.serialization import NoEncryption
21
22 def main():
23     """
24     #####
25     # Creating an S/MIME certificate #
26     #####
27     """
28     try:
```

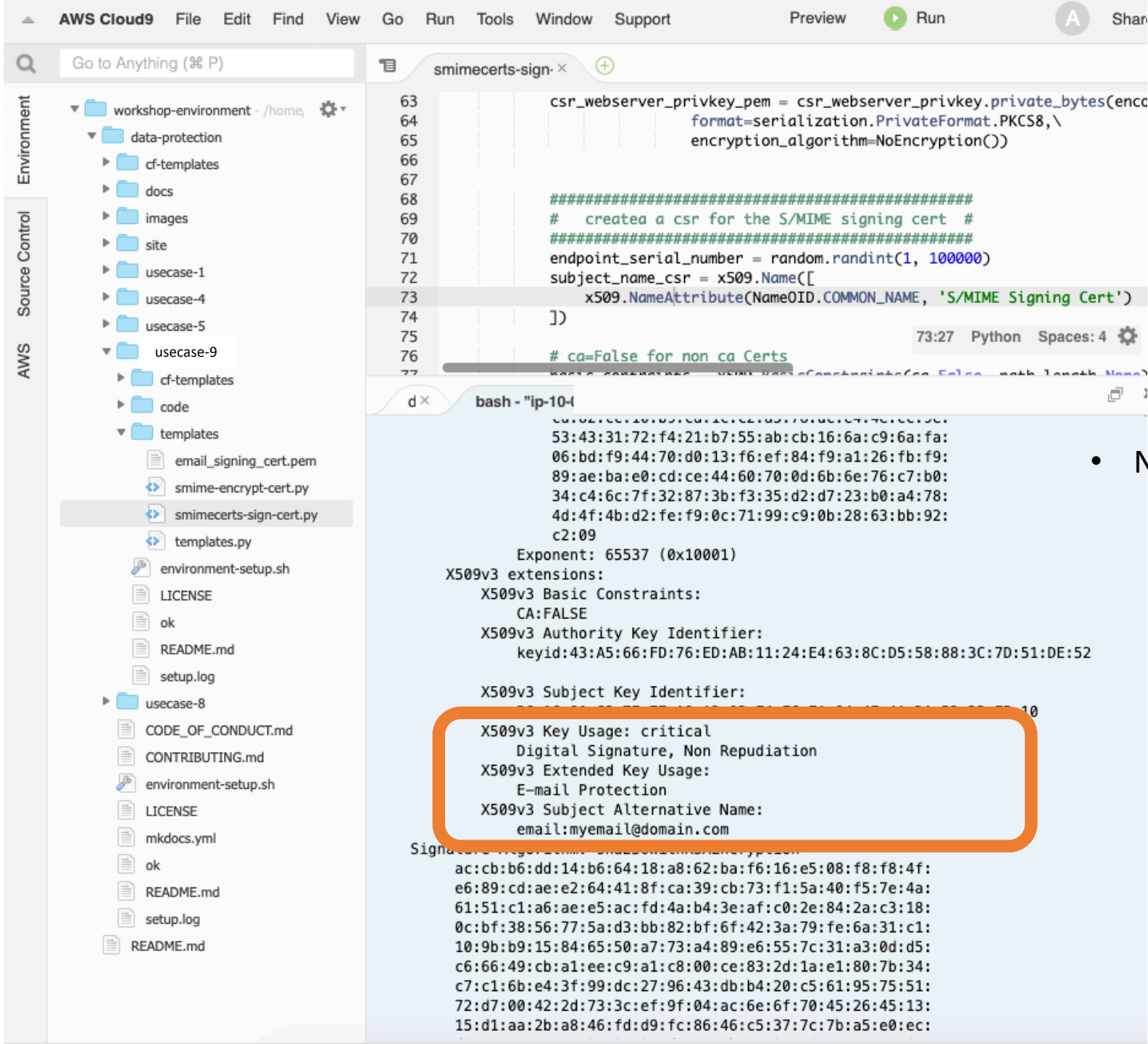
bash - "ip-10-0-0" data-protection/u

Run Command: data-protection/usecase-7/templates/smimecerts.py Runner: Python 3 CWD ENV

This step will take about 2 minutes to complete

Successfully created code signing cert email_cert.pem

Process exited with code: 0



- Notice that:
 - Key Usage and Extended Key Usage have been set by API Passthrough method