

Practical: AWS Parameter Store (Python)

Table of Contents

- [Prerequisites and references](#)
- [Creating parameters](#)
- [Accessing parameters programmatically](#)
- [Delete parameters when done](#)

AWS Parameter Store is part of the Systems Manager service. It's functionality is similar to Secrets Manager in that it essentially a key-value store that you can manage from within AWS. However, it doesn't have some of the functionality you would typically want for secret values, such as rotation, encryption at rest, etc.. The Parameter Store is intended to store configuration information that other resources can query. A good example might be storing the DNS name for your app, so that it can use that name when dynamically creating responses to HTTP queries. Storing it outside your application VM or container images means that you can configure your app without rebuilding those images.

Prerequisites and references

- [AWS CAB432 login](#) ➦ (<https://d-97671c4bd0.awsapps.com/start/#/?tab=accounts>)
- [AWS CAB432 Parameter Store](#) ➦ (<https://ap-southeast-2.console.aws.amazon.com/systems-manager/parameters/?region=ap-southeast-2&tab=Table>)
- [AWS Parameter Store docs](#) ➦ (<https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-parameter-store.html>)
- [parameterstore_python.zip](#) (<https://canvas.qut.edu.au/courses/20367/files/6722624/download>)

Creating parameters

Parameters are usually named with a structure similar to file paths. This makes it easier to organise them. There is additional functionality in Parameter Store for querying based on paths, such as retrieving all parameters within a certain prefix. We won't look at those in this practical, but you can explore them on your own.

We'll sketch out the process and leave it to you to work out the details.

- Go to the Parameter Store on the AWS console. Create a new parameter
- For the *Name* create one starting with your username, such as `/n1234567/demo_parameter`
- Fill in the *Value* with something of your choosing.
- Tag with `qut-username` set to your username like `n1234567@qut.edu.au`
- Create the parameter.

In the CAB432 environment you will only be able to access, delete or change parameters that are tagged with your QUT username.

It is possible to edit the parameter after creating it, which can allow you to adjust the configuration of your application.

Accessing parameters programmatically

- Create a new python project:

```
mkdir parameterstoredemo_python
cd parameterstoredemo_python
touch parameterstore_python.py
```

- Create `parameterstore_python.py` and fill with the following contents (replacing the user name with your own).

```
import boto3
from botocore.exceptions import ClientError
```

```
parameter_name = "/n1234567/demo_parameter" # Change to your own parameter name
region = "ap-southeast-2"

ssm = boto3.client('ssm', region_name=region)

def main():
    try:
        response = ssm.get_parameter(Name=parameter_name)
        print(response['Parameter']['Value'])
    except ClientError as error:
        print(error)

if __name__ == "__main__":
    main()
```

- Run with `python parameterstore_python.py`. You should see your value printed out.

Delete parameters when done

As usual, when you are done with your parameters please delete them.

Complete code: [parameterstore_python.zip \(https://canvas.qut.edu.au/courses/20367/files/6722624/download\)](https://canvas.qut.edu.au/courses/20367/files/6722624/download)

TEQSA PRV12079 | CRICOS 00213J | ABN 83 791 724 622