



lab



lab title

**AWS Command Line Interface (CLI)
V1.04**



Course title

**BackSpace Academy
AWS Certified Associate**



Table of Contents

Contents

Table of Contents	1
About the Lab.....	2
Using CloudShell for Command Line Access to AWS Resources.....	3
Clean Up.....	4
Creating a Cloud9 Development Environment on EC2	5
Sending CLI Commands from a Cloud9 Environment	10
Sending Commands from the Cloud9 Environment.....	11
Clean Up.....	14

About the Lab

Please note that not all AWS services are supported in all regions. Please use the US-East-1 (North Virginia) region for this lab.

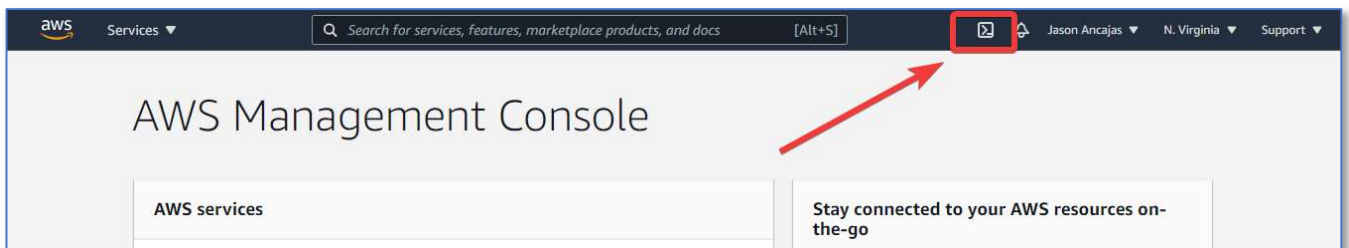
These lab notes are to support the AWS CLI lab of the AWS Certified Cloud Practitioner Course.

Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the latest version with any updates or corrections.

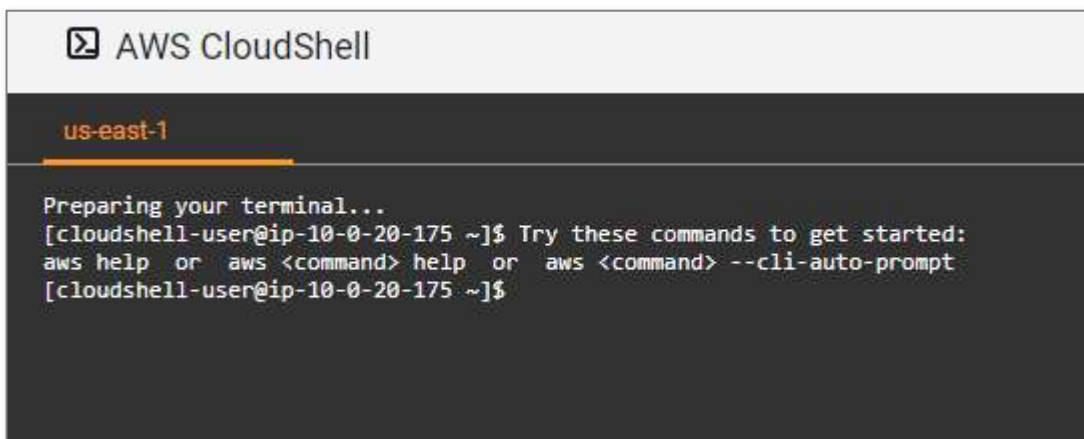
▶ Using CloudShell for Command Line Access to AWS Resources

In this section, we will use the CloudShell service to access AWS resources.

From the AWS console click the *CloudShell* icon



Your *CloudShell* environment will open.



Check that *AWS CLI* is already installed.

```
aws --version
```

```
[cloudshell-user@ip-10-0-20-175 ~]$ aws --version
aws-cli/2.1.28 Python/3.8.8 Linux/4.14.225-168.357.amzn2.x86_64 exec-env/CloudShell exe/x86_64.amzn.2 prompt/off
[cloudshell-user@ip-10-0-20-175 ~]$
```

Create a bucket using the `S3 mb` command

```
aws s3 mb s3://yourbucketname
```

```
[cloudshell-user@ip-10-0-20-175 ~]$ aws s3 mb s3://jancajas-bucket-cli
make_bucket: jancajas-bucket-cli
[cloudshell-user@ip-10-0-20-175 ~]$
```

Go to the *S3 management console* to see the newly created bucket

Amazon S3 console screenshot showing the 'Buckets (2)' list. The bucket 'jancajas-bucket-cli' is highlighted with a red box. A red arrow points from the 'Storage Lens' section in the left sidebar to the bucket list.

Name	AWS Region	Access	Creation date
elasticbeanstalk-us-east-1-991610390270	US East (N. Virginia) us-east-1	Objects can be public	July 19, 2021, 17:10:24 (UTC+08:00)
jancajas-bucket-cli	US East (N. Virginia) us-east-1	Objects can be public	July 28, 2021, 07:01:24 (UTC+08:00)

Clean Up

Now *Delete* the bucket

Amazon S3 console screenshot showing the 'Buckets (2)' list. The bucket 'jancajas-bucket-cli' is highlighted with a red box and labeled with a red circle '1'. The 'Delete' button is highlighted with a red box and labeled with a red circle '2'.

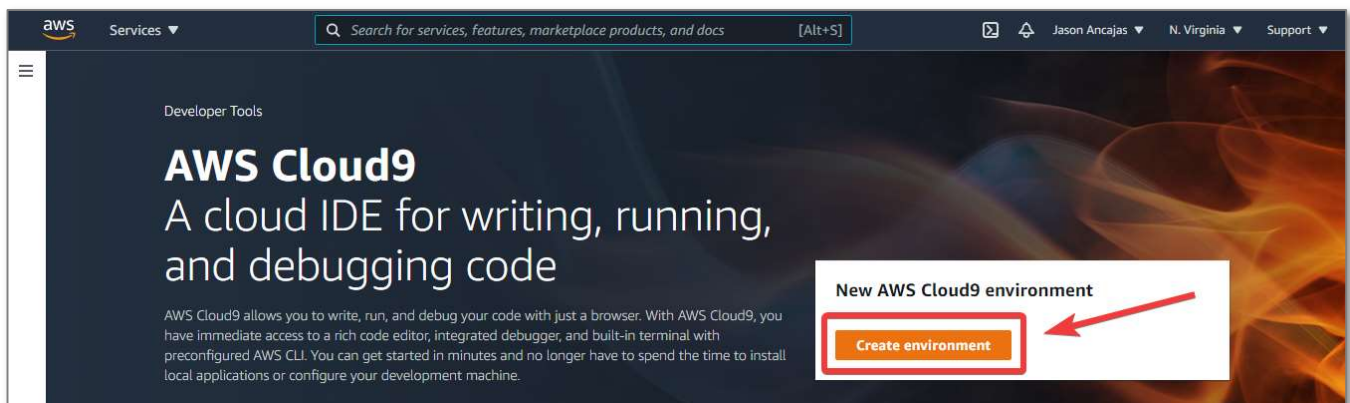
Name	AWS Region	Access	Creation date
elasticbeanstalk-us-east-1-991610390270	US East (N. Virginia) us-east-1	Objects can be public	July 19, 2021, 17:10:24 (UTC+08:00)
jancajas-bucket-cli	US East (N. Virginia) us-east-1	Objects can be public	July 28, 2021, 07:01:24 (UTC+08:00)

▶ Creating a Cloud9 Development Environment on EC2

In this section, we will use the Cloud9 service to create a development environment on an EC2 instance.

From the AWS console search *Cloud9*

Click *Create environment*



Give your environment a unique name.

Click *Next step*

AWS Cloud9 ×

AWS Cloud9 > Environments > Create environment

Step 1
Name environment

Step 2
Configure settings

Step 3
Review

Name environment

Environment name and description

Name
The name needs to be unique per user. You can update it at any time in your environment settings.
Backspace Labs - J Ancajas
Limit: 60 characters

Description - Optional
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.
Write a short description for your environment
Limit: 200 characters

Cancel **Next step**

Select *EC2 environment*.

Step 1
Name environment

Step 2
Configure settings

Step 3
Review

Configure settings

Environment settings

Environment type [Info](#)
Run your environment in a new EC2 instance or an existing server. With EC2 instances, you can connect directly through Secure Shell (SSH) or connect via AWS Systems Manager (without opening inbound ports).

- ☒ **Create a new EC2 instance for environment (direct access)**
Launch a new instance in this region that your environment can access directly via SSH.
- ☐ **Create a new no-ingress EC2 instance for environment (access via Systems Manager)**
Launch a new instance in this region that your environment can access through Systems Manager.
- ☐ **Create and run in remote server (SSH connection)**
Configure the secure connection to the remote server for your environment.

Select *t2 micro* to stay in the free tier

Instance type

☒ **t2.micro (1 GiB RAM + 1 vCPU)**
Free-tier eligible. Ideal for educational users and exploration.

☐ **t3.small (2 GiB RAM + 2 vCPU)**
Recommended for small-sized web projects.

☐ **m5.large (8 GiB RAM + 2 vCPU)**
Recommended for production and general-purpose development.

☐ **Other instance type**
Select an instance type.

t3.nano

Leave hibernation setting at 30 mins

Cost-saving setting
Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.

After 30 minutes (default)

Leave Network settings as *default*

Click *Next step*

Network settings (advanced)

Network (VPC)
Launch your EC2 instance into an existing Amazon Virtual Private Cloud (VPC) or create a new one. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an internet gateway (IGW) to your new VPC.

vpc-800a67fd (default) [Create new VPC](#)

Subnet
Select a public subnet in which the EC2 instance is created. (For a private subnet, you must create an environment that connects to its instance via Systems Manager.)

No preference (default subnet in any Availability Zone) [Create new subnet](#)

No tags associated with the resource.

[Add new tag](#)
You can add 50 more tags.

Cancel Previous step **Next step**

Click *Create environment*

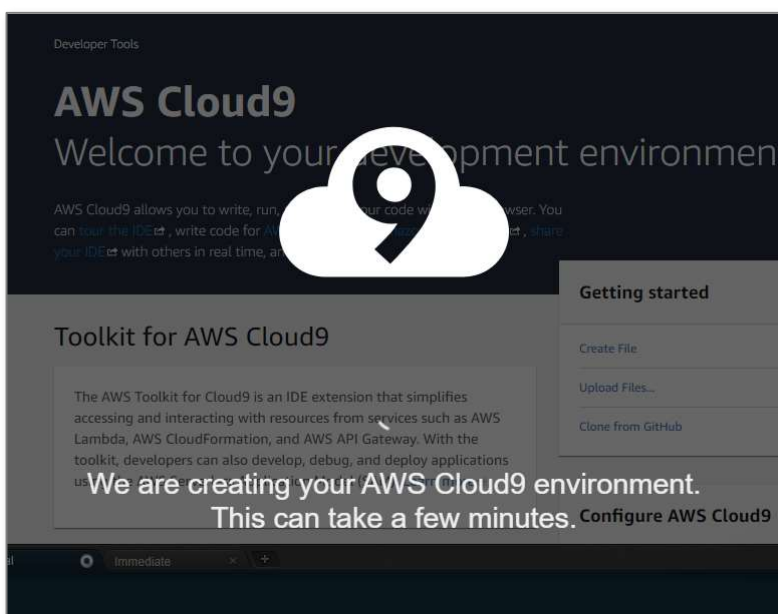
IAM role
AWSServiceRoleForAWSCloud9 (generated)

We recommend the following best practices for using your AWS Cloud9 environment

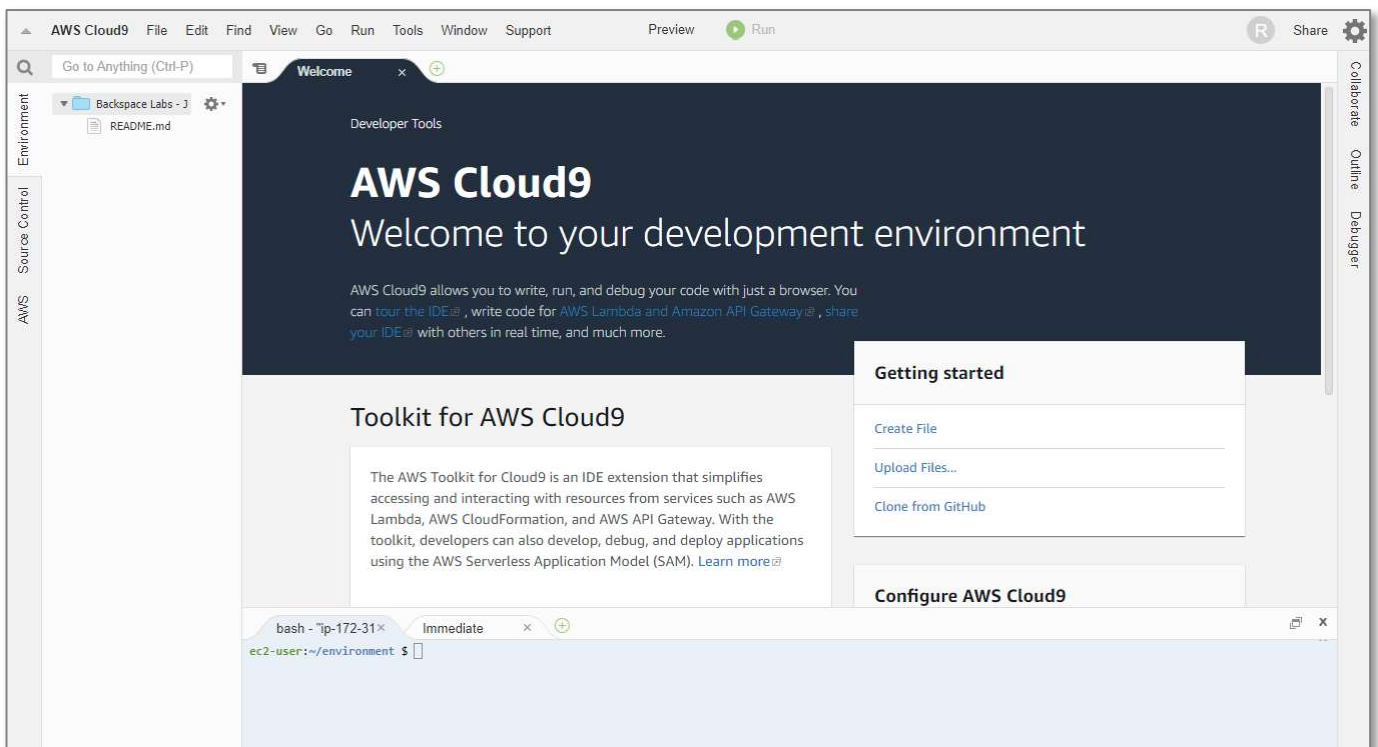
- Use **source control** and **backup** your environment frequently. AWS Cloud9 does not perform automatic backups.
- Perform regular **updates of software** on your environment. AWS Cloud9 does not perform automatic updates on your behalf.
- Turn on AWS CloudTrail** in your AWS account to track activity in your environment. [Learn more](#)
- Only share your environment with **trusted users**. Sharing your environment may put your AWS access credentials at risk. [Learn more](#)

Cancel Previous step **Create environment**

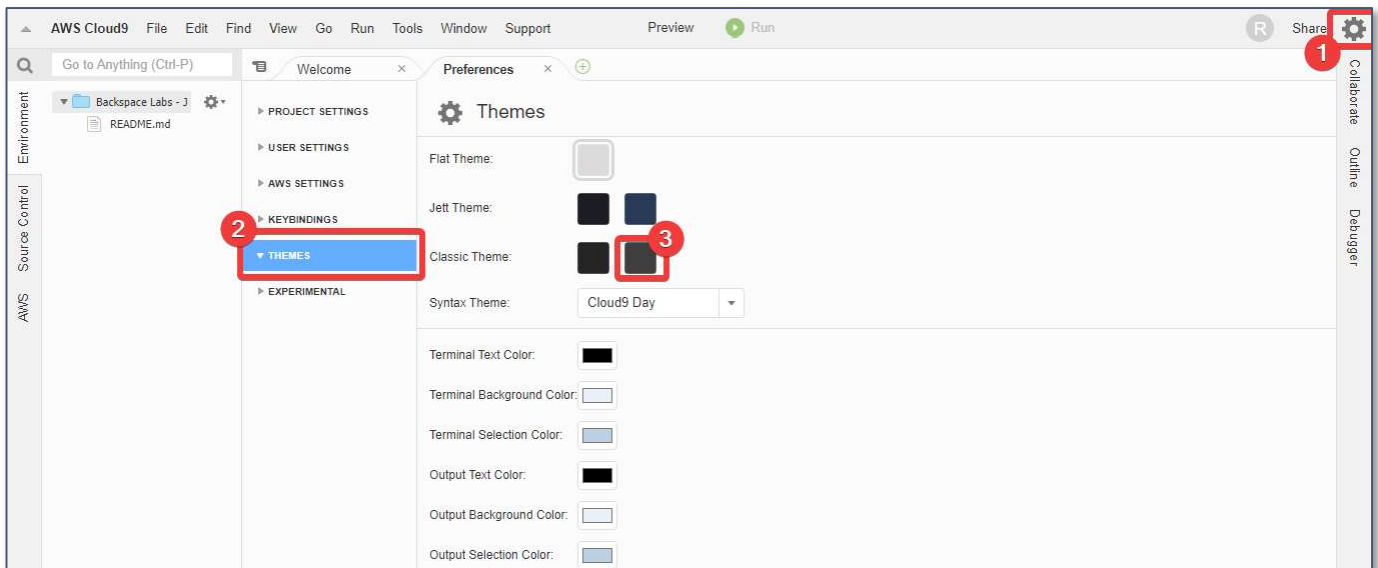
The environment creation process will begin



After some time, your environment will be created.



You can customize the look and feel of the IDE by selecting a theme from the *preferences*.



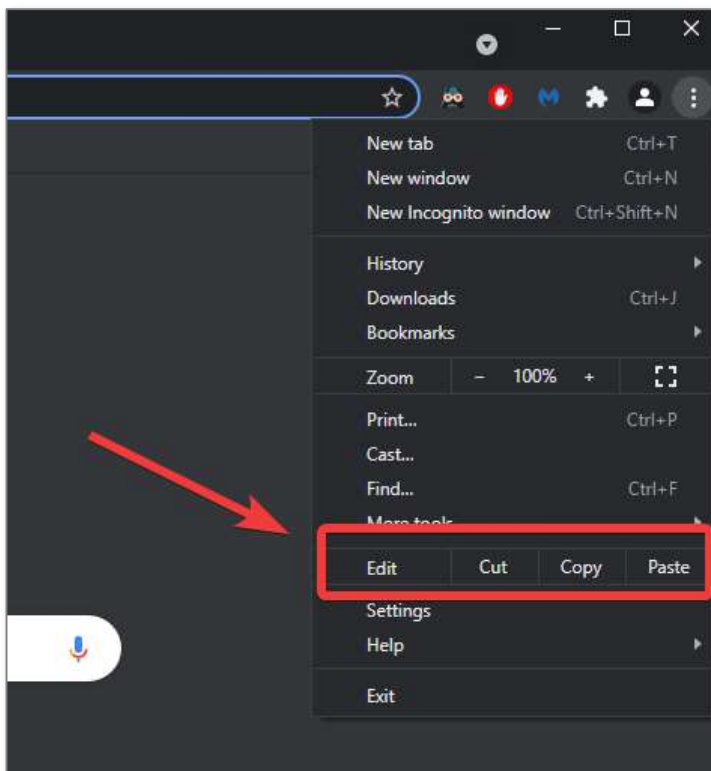
▶ Sending CLI Commands from a Cloud9 Environment

In this section, we will use the Cloud9 service to send CLI commands from the Cloud9 Environment.

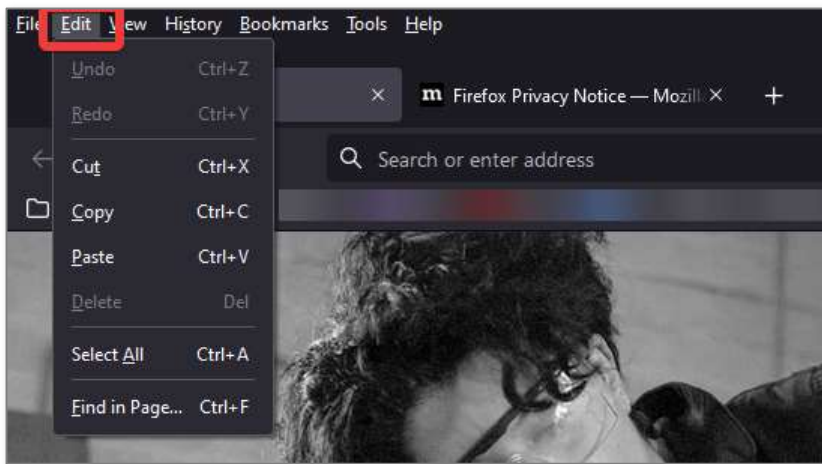
Please note:

Cut and Paste (right click or Cloud9 menu) may not work directly in Cloud9. If you cannot paste into Cloud9 then use the browser paste menu item or use `ctrl-v` (Windows) / `cmd-v` (MAC).

e.g., for Chrome:



e.g., for Firefox (Press Alt)



Sending Commands from the Cloud9 Environment

At the bottom of the screen will be the *Linux terminal console* panel.

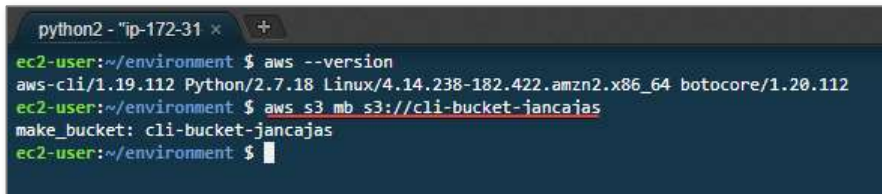
Check that *AWS CLI* is already installed.

```
aws --version
```

```
bash - "ip-172-31-25" x +
ec2-user:~/environment $ aws --version
aws-cli/1.19.112 Python/2.7.18 Linux/4.14.238-182.422.amzn2.x86_64 botocore/1.20.112
ec2-user:~/environment $
```

Create a bucket using the *S3 mb* command

```
aws s3 mb s3://yourbucketname
```

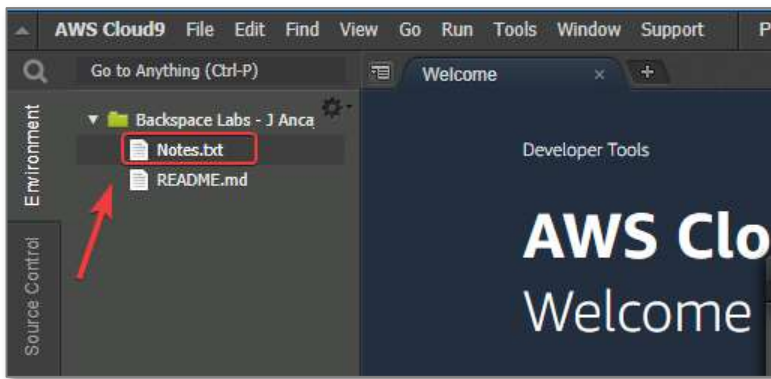


```
python2 - "ip-172-31" x +
ec2-user:~/environment $ aws --version
aws-cli/1.19.112 Python/2.7.18 Linux/4.14.238-182.422.amzn2.x86_64 botocore/1.20.112
ec2-user:~/environment $ aws s3 mb s3://cli-bucket-jancajas
make_bucket: cli-bucket-jancajas
ec2-user:~/environment $
```

Upload a local file to the *Cloud9 Environment*



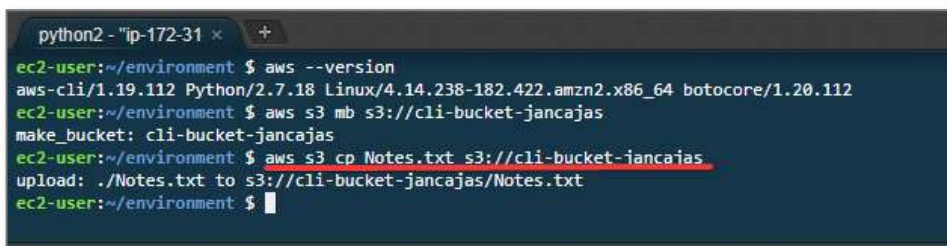
The file will appear in the file *treeview* after uploading



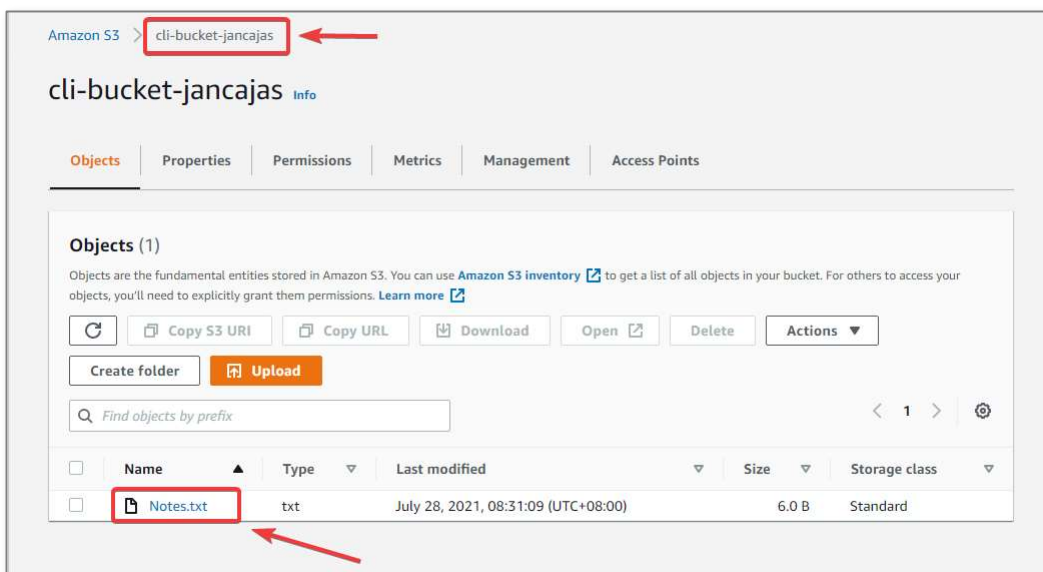
Now use the `S3 cp` command to copy the file to the newly created bucket

*Note filename is case sensitive

```
aws s3 cp yourfilename s3://yourbucketname
```

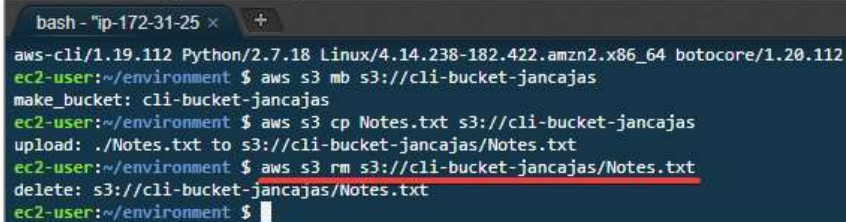


Go to the *S3 management console* to see the newly created bucket with its contents



Now we will delete the object using the *S3 rm* command

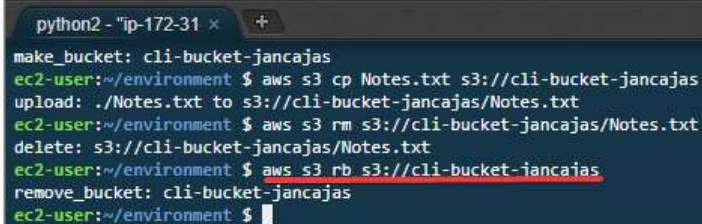
```
aws s3 rm s3://yourbucketname/yourfilename
```



```
bash - "ip-172-31-25" x +
aws-cli/1.19.112 Python/2.7.18 Linux/4.14.238-182.422.amzn2.x86_64 botocore/1.20.112
ec2-user:~/environment $ aws s3 mb s3://cli-bucket-jancajas
make_bucket: cli-bucket-jancajas
ec2-user:~/environment $ aws s3 cp Notes.txt s3://cli-bucket-jancajas
upload: ./Notes.txt to s3://cli-bucket-jancajas/Notes.txt
ec2-user:~/environment $ aws s3 rm s3://cli-bucket-jancajas/Notes.txt
delete: s3://cli-bucket-jancajas/Notes.txt
ec2-user:~/environment $
```

Now we will delete the empty bucket using the *S3 rb* command

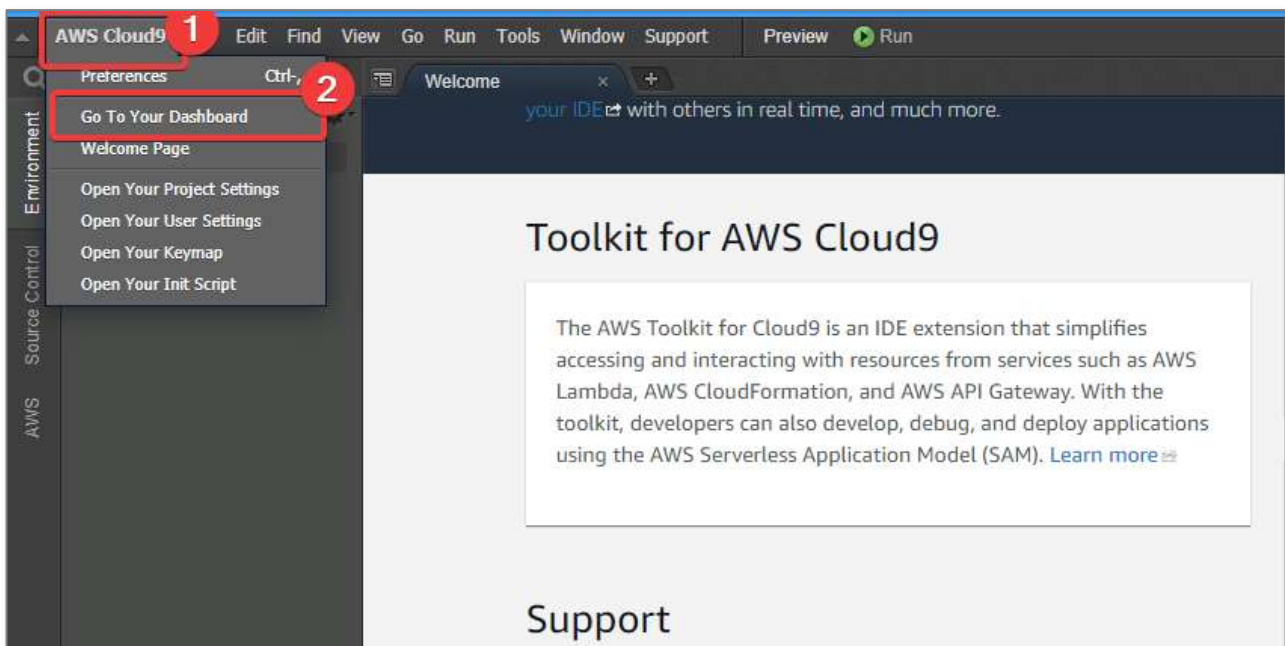
```
aws s3 rb s3://yourbucketname
```



```
python2 - "ip-172-31" x +
make_bucket: cli-bucket-jancajas
ec2-user:~/environment $ aws s3 cp Notes.txt s3://cli-bucket-jancajas
upload: ./Notes.txt to s3://cli-bucket-jancajas/Notes.txt
ec2-user:~/environment $ aws s3 rm s3://cli-bucket-jancajas/Notes.txt
delete: s3://cli-bucket-jancajas/Notes.txt
ec2-user:~/environment $ aws s3 rb s3://cli-bucket-jancajas
remove_bucket: cli-bucket-jancajas
ec2-user:~/environment $
```

Clean Up

Go back to your *Cloud9 Dashboard*



Select your environment and click *Delete*

