

Lab 1: Reference Architecture 1

Custom EC2 Instance with S3 Bucket and IAM Policy
DS 5220 - Advanced Cloud Computing

Student: xdy6sg | Date: January 23, 2026

Part 1: CloudFormation Stack Deployment

The CloudFormation stack was successfully deployed with status CREATE_COMPLETE.

Stack Parameters

Parameter	Value
Stack Name	refarch1
Instance Type	t3.micro
Key Name	ds5220
SSH Location	0.0.0.0/0
Subnet ID	subnet-09d9aeac372c739e9
UVA ID	xdy6sg
VPC	vpc-022aefb2869b1d62f

Stack Creation Process :

The screenshot shows the AWS CloudFormation console with the 'refarch1' stack selected. The 'Events' tab is active, displaying a list of 23 events. The events are listed in chronological order, starting from 'CREATE_COMPLETE' on 2026-01-22 23:42:00 UTC-0500 and ending with 'User Initiated' on 2026-01-22 23:39:25 UTC-0500. The events include the creation of EC2 instances, IAM roles, and S3 buckets, along with various consistency checks and resource creation steps.

Event ID	Timestamp	Logical ID	Status	Detailed Status	Status Reason	Hook Invocations
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:42:00 UTC-0500	refarch1	CREATE_COMPLETE	-	-	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:43:58 UTC-0500	EC2InstanceProfile	CREATE_COMPLETE	-	-	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:41:03 UTC-0500	EC2Instance	CREATE_COMPLETE	-	-	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:40:52 UTC-0500	refarch1	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:40:52 UTC-0500	EC2Instance	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:40:50 UTC-0500	EC2Instance	CREATE_IN_PROGRESS	-	Resource creation initiated	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:40:50 UTC-0500	S3BucketAccessPolicy	CREATE_COMPLETE	-	-	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:39:46 UTC-0500	EC2Instance	CREATE_IN_PROGRESS	-	-	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:39:47 UTC-0500	S3BucketAccessPolicy	CREATE_IN_PROGRESS	-	Resource creation initiated	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:39:47 UTC-0500	EC2InstanceProfile	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:39:47 UTC-0500	EC2Instance	CREATE_IN_PROGRESS	-	Resource creation initiated	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:39:47 UTC-0500	S3Bucket	CREATE_IN_PROGRESS	-	-	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:39:47 UTC-0500	EC2SecurityGroup	CREATE_COMPLETE	-	-	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:39:48 UTC-0500	EC2SecurityGroup	CREATE_IN_PROGRESS	-	Resource creation initiated	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:39:23 UTC-0500	S3Bucket	CREATE_IN_PROGRESS	-	Resource creation initiated	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:39:23 UTC-0500	EC2InstanceRole	CREATE_IN_PROGRESS	-	Resource creation initiated	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:39:28 UTC-0500	EC2SecurityGroup	CREATE_IN_PROGRESS	-	-	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:39:28 UTC-0500	EC2InstanceRole	CREATE_IN_PROGRESS	-	-	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:39:28 UTC-0500	S3Bucket	CREATE_IN_PROGRESS	-	-	-
cft7593bb-beef-4dca-9344-799608144606	2026-01-22 23:39:25 UTC-0500	refarch1	CREATE_IN_PROGRESS	-	User Initiated	-

Part 2: Resource Identification and Exploration

Resource 1: EC2 Instance

Field	Value
Instance ID	i-02ef18f0d56f0be53
Public IP Address	52.207.88.151
Private IP Address	172.31.39.21
Instance Type	t3.micro
AMI ID	ami-07ff62358b87c7116
AMI Name	al2023-ami-2023.10.20260105.0-kernel-6.1-x86_64
Availability Zone	us-east-1d
Key Pair	ds5220
Launch Time	January 22, 2026, 23:40 (UTC-05:00)

Web Server Test: Successfully accessed <http://52.207.88.151> and received a welcome page displaying: "Hello xdy6sg from EC2 with S3 Access! This EC2 instance has access to S3 bucket: xdy6sg-refarch1"

IP Address : <http://52.207.88.151/>



Resource 2: Security Group

Field	Value
Security Group ID	sg-080071e5e14e79d04
Security Group Name	refarch1-EC2SecurityGroup-ncLg0jaZN2Hk
VPC ID	vpc-022aefb2869b1d62f

Inbound Rules

Port	Protocol	Source	Description
22	TCP	0.0.0.0/0	Allow SSH access
80	TCP	0.0.0.0/0	Allow HTTP traffic from anywhere

- What traffic is allowed into the EC2 instance?
 - SSH traffic on port 22 (for remote terminal access)
 - HTTP traffic on port 80 (for web server access)
- Why are these specific ports open?
 - **Port 22 (SSH):** Allows you to remotely connect to the instance's command line to manage it, install software, and run commands — like when you used EC2 Instance Connect or tried SSH from your Mac
 - **Port 80 (HTTP):** Allows web browsers to access the Apache web server running on the instance — this is why you could see the "Hello xdy6sg" welcome page in your browser

Resource 3: S3 Bucket

Field	Value
Bucket Name	xdy6sg-refarch1
Bucket ARN	arn:aws:s3:::xdy6sg-refarch1
Bucket Versioning	Enabled
MFA Delete	Disabled
Default Encryption	SSE-S3 (Amazon S3 managed keys)
Block All Public Access	On (all 4 settings enabled)
Bucket Policy	None
ACL	Bucket owner only

S3 Access Test from EC2

All S3 operations were successfully executed from the EC2 instance:

- aws s3 ls s3://xdy6sg-refarch1/ - Listed bucket contents
- echo "Hello from EC2" > test.txt - Created test file
- aws s3 cp test.txt s3://xdy6sg-refarch1/ - Uploaded successfully
- aws s3 cp s3://xdy6sg-refarch1/test.txt downloaded.txt - Downloaded successfully
- cat downloaded.txt - Verified contents: "Hello from EC2"
- aws s3 rm s3://xdy6sg-refarch1/test.txt - Deleted successfully

Resource 4: IAM Role

Field	Value
Role Name	refarch1-EC2InstanceRole-05gDpGgRaJ2I
Role ARN	arn:aws:iam::441700731647:role/refarch1-EC2InstanceRole-05gDpGgRaJ2I
Number of Policies Attached	2
Creation Date	January 22, 2026, 23:39 (UTC-05:00)
Maximum Session Duration	1 hour

Attached Policies

Policy Name	Type
AmazonSSMManagedInstanceCore	AWS Managed
S3BucketAccessPolicy	Customer Inline

S3BucketAccessPolicy - Allowed Actions

Action	Description
s3:GetObject	Download/read objects from the bucket
s3:PutObject	Upload/write objects to the bucket
s3:DeleteObject	Delete objects from the bucket
s3>ListBucket	List the contents of the bucket

Resources the policy applies to:

- arn:aws:s3:::xdy6sg-refarch1/* (all objects inside the bucket)
- arn:aws:s3:::xdy6sg-refarch1 (the bucket itself)

Resource 5: IAM Instance Profile

Field	Value
Instance Profile Name	refarch1-EC2InstanceProfile-RThoQEuAKkFI
Instance Profile ARN	arn:aws:iam::441700731647:instance-profile/refarch1-EC2InstanceProfile-RThoQEuAKkFI
Attached IAM Role	refarch1-EC2InstanceRole-05gDpGgRaJ2I
Verified on EC2 Security Tab	Yes - IAM Role matches

Part 4: CloudFormation Outputs

Key	Value	Export Name
IAMRoleArn	arn:aws:iam::441700731647:role/refarch1-EC2InstanceRole-05gDpGgRaJ2I	refarch1-IAMRoleArn
InstanceId	i-02ef18f0d56f0be53	refarch1-InstanceId
InstancePublicIP	52.207.88.151	refarch1-InstancePublicIP
S3BucketArn	arn:aws:s3:::xdy6sg-refarch1	refarch1-S3BucketArn
S3BucketName	xdy6sg-refarch1	xdy6sg-refarch1
SecurityGroupId	sg-080071e5e14e79d04	refarch1-SecurityGroupId

Why might these outputs be useful if you were building a larger infrastructure?

Outputs are useful because they:

1. Enable cross-stack references — Other CloudFormation stacks can import these values. For example, a second stack deploying a Lambda function could reference the S3BucketArn to grant it access to the same bucket.
2. Simplify automation — Scripts and CI/CD pipelines can retrieve outputs programmatically (using aws cloudformation describe-stacks) to configure other services without hardcoding values.
3. Provide documentation — Outputs serve as a quick reference for important resource identifiers. When troubleshooting or connecting services, you don't need to hunt through the console.
4. Support modular architecture — You can break infrastructure into smaller, reusable stacks that communicate through outputs and imports, making templates easier to maintain and reuse across projects.

Part 5: Stack Resources

Total resources created: 6

Logical ID	Type	Status
EC2Instance	AWS::EC2::Instance	CREATE_COMPLETE
EC2InstanceProfile	AWS::IAM::InstanceProfile	CREATE_COMPLETE
EC2InstanceRole	AWS::IAM::Role	CREATE_COMPLETE
EC2SecurityGroup	AWS::EC2::SecurityGroup	CREATE_COMPLETE
S3Bucket	AWS::S3::Bucket	CREATE_COMPLETE
S3BucketAccessPolicy	AWS::IAM::Policy	CREATE_COMPLETE

[Insert screenshot of Resources tab here]

Logical ID	Physical ID	Type	Status	Module
EC2Instance	I-02x1f180e056f0be51	AWS::EC2::Instance	CREATE_COMPLETE	-
EC2InstanceProfile	refarch1-EC2InstanceProfile-RThoQEuAxFI	AWS::IAM::InstanceProfile	CREATE_COMPLETE	-
EC2InstanceRole	refarch1-EC2InstanceRole-05gDvGpRaJZl	AWS::IAM::Role	CREATE_COMPLETE	-
EC2SecurityGroup	sg-080071e1e14e79d04	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
S3Bucket	xdy0sg-refarch1	AWS::S3::Bucket	CREATE_COMPLETE	-
S3BucketAccessPolicy	refar-53Buc-80HpZuzMxDzA	AWS::IAM::Policy	CREATE_COMPLETE	-

Operation ID	Timestamp	Logical ID	Status	Detailed status	Status reason	Hook invocations
2c3c70d5-317a-4b66-9907-48ec3be068be	-	-	-	-	-	-
2c3c70d5-317a-4b66-9907-48ec3be068be	2026-01-23 11:54:24 UTC-0500	S3BucketAccessPolicy	DELETE_COMPLETE	-	-	-
2c3c70d5-317a-4b66-9907-48ec3be068be	2026-01-23 11:34:23 UTC-0500	EC2Instance	DELETE_IN_PROGRESS	-	-	-
2c3c70d5-317a-4b66-9907-48ec3be068be	2026-01-23 11:34:23 UTC-0500	S3BucketAccessPolicy	DELETE_IN_PROGRESS	-	-	-
2c3c70d5-317a-4b66-9907-48ec3be068be	2026-01-23 11:34:20 UTC-0500	refarch1	DELETE_IN_PROGRESS	-	User Initiated	-

Discussion Questions

1. What advantages does CloudFormation provide compared to manually creating these resources through the console?

- Repeatability: Deploy the exact same infrastructure multiple times using the same template, ensuring consistency across environments (dev, staging, production).
- Version Control: Templates are code (YAML/JSON), so you can track changes in Git, review modifications, and roll back if needed.
- Automation: A single click deploys all 6 resources in the correct order with proper dependencies - no manual clicking through multiple console pages.
- Documentation as Code: The template itself documents your infrastructure architecture.
- Easy Cleanup: Deleting the stack removes all resources automatically, preventing orphaned resources and unexpected charges.
- Error Reduction: Eliminates human error from manual configuration (typos, forgotten settings, etc.).

2. Can you identify the created resources that have to do with computational resources? network security? various types of storage?

Category	Resources
Compute	EC2Instance, EC2InstanceProfile, EC2InstanceRole, S3BucketAccessPolicy
Network Security	EC2SecurityGroup (controls inbound/outbound traffic on ports 22 and 80)
Storage	S3Bucket (object storage for files and data)

3. How does the IAM role enable the EC2 instance to access S3 without storing credentials?

The IAM role uses temporary security credentials through the AWS Security Token Service (STS). When the EC2 instance needs to access S3, it automatically retrieves short-lived credentials from the instance metadata service (available at <http://169.254.169.254>). These credentials are rotated automatically by AWS, so you never need to store long-term access keys on the instance. The Instance Profile acts as a container that attaches the IAM role to the EC2 instance, enabling this seamless, secure authentication.

4. What would happen if you deleted the S3 bucket manually before deleting the CloudFormation stack?

The CloudFormation stack deletion would fail. CloudFormation expects to manage all resources it created. When it tries to delete the S3 bucket and finds it already gone, it will report a "resource not found" error and the stack will enter a DELETE_FAILED state. To resolve this, you would need to either skip that resource during deletion or manually remove the bucket reference from the stack before retrying the delete operation.

5. Why is the security group configured to allow only ports 22 and 80? What are those ports used for?

Port	Protocol	Purpose
22	SSH	Secure Shell - allows administrators to remotely connect to the instance's command line for management and configuration
80	HTTP	Web traffic - allows users to access the web server running on the instance via their browser

This follows the principle of least privilege: only the minimum required ports are open. Opening unnecessary ports increases the attack surface and security risk. For a basic web server that needs remote administration, ports 22 and 80 are the essential minimum.