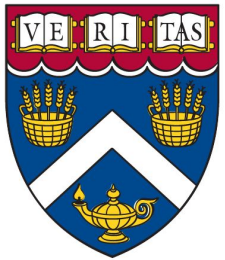


HARVARD
Extension School

Week 5

Amazon AWS
High Availability website
Command Line Interface (CLI)

This week..

- **Week 05:** Amazon AWS Command Line Interface (CLI)

  This lecture will cover the AWS command line interface and how to use AWS CLI in a bash script to create the same environment created by the AWS console in week 04. If time allow we will introduce the High availability websites.
  - AWS Identity Access Management (IAM) - Users, Groups, Roles and Policies
  - S3 and Static website
  - AWS Command Line Interface(CLI)
  - AWS Credential management best practices
  - Building AWS resource using AWS CLI
  - Building high available services.

  Assignment 2

Identity Access Management - IAM

- Authentication, Authorization, and Accounting
  - Authentication: Allows you to get in.
  - Authorization: Defines what you're allowed to do.
  - Accounting: Determines what you did.
- Used to manage users and their access level to all AWS services on the same account.
- Features:
  - MFA (Multi-Factor Authentication)
  - Identity Federation (SSO like AD , FB , et)
  - Shared access between AWS accounts

https://console.aws.amazon.com/iam/

IAM Objects

- ★ IAM **Users** and **groups** : people
  - **NOTE: Root access is an account but not an IAM user. It is the default user who has full access**
- ★ **Policies** : access level , permissions
  - Identity-based policies, Related to the IAM users and groups,
    - AWS-Managed policies
    - Customer-Managed policies (inline policies)
  - Resource-based policies, Permissions directly assigned to different resources, such as S3 buckets or EC2 instances.
- ★ **Roles** : services (EC2 , lambda , etc)

Simple Storage Services (S3)

- Object storage component (data, metadata, and the unique identifier).
- Consists of Buckets and objects contained within buckets.
- Features:
  - Secure , reliable and durable Storage
  - Provides 99.999999999% durability
  - Unlimited but the max single file size is 5TB
  - Versioning, encryption, static site, access policy
  - Region-based service: The user interface shows all your buckets, in all regions. But buckets exist in a specific region and you need to specify that region when you create a bucket. Therefore You can access any bucket from any region.
  - Universal Namespace - bucket name should be unique.
- Standard, One Zone-Infrequent Access (S3 Z-IA), Glacier.

https://aws.amazon.com/s3/

Create Static Website using S3

- Static Sites : No dynamic contents, no php ,etc
- Serverless: Build and run applications and services with no servers.
- Advantages:
  - Less cost
  - No server management and administration
  - High Availability

Static website in S3 -

- Create a bucket - remember unique name
- Upload an object, a text file with content, or html script. Give it the name index.html
- From the S3 Console click on properties tab
- Click on static website hosting
- Select the first option “Use this bucket to host a website”
- Enter “index.html” in the index document field.
- Take a note of the Endpoint as this will be your website address
- Save
- Go to permissions tab and click on the “bucket policy” button.
- Add the policy to allow All to perform GetObject policy.. On the next slide

Static website in S3 (cont.)

```
{
  "Version":"2012-10-17",
  "Statement":[{
      "Sid":"PublicReadGetObject",
        "Effect":"Allow",
        "Principal": "*",
      "Action":["s3:GetObject"],
      "Resource":["arn:aws:s3:::example-bucket/*"
      ]
    }
  ]
}
```

- To create a DNS for the website and customize the website address, Route 53 is required.

Building high available service

- Bootstrap
  - User data in EC2
- Load balancer
  - Application LB, Network LB and Classic LB
- Autoscaling
  - Launch Configuration

Bootstrap Script

- User data in EC2 can be used to configure and setup the initial configuration needed first time the EC2 machine launches.
- Shebang symbol + /bin/bash, will be executed as a Bash script ⇒ #!/bin/bash
- Example 1: Create a web server

```
#! /bin/bash
# yum update -y <- if it takes too long (depend on the last time the AMI update) will cause delay, so what?
# Install httpd
yum install httpd -y
#Create a file with the content CSCI E-91
echo "CSCI E-91 " > /var/www/html/index.html
# Start and enable the service
systemctl start httpd
systemctl enable httpd
```

Bootstrap Script (cont.)

- Example 2: Add user and ssh keys to the instance

```
# Delete the user if exist
userdel -r fadel 2> /dev/null
# Create user fadel with home directory and bash for shell
useradd fadel --create-home --shell /bin/bash
# As a user root, create .ssh folder for fadel
mkdir /home/fadel/.ssh
# As a user root, create authorized_keys file inside .ssh folder
cat > /home/fadel/.ssh/authorized_keys << EOF
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC7a98xqol/emQKxyK4d317fpZQLM5aiRKXAqZVZ7W9F3yHqVeJ+w8tRW4h0vxxvqMAzD62Pef7aDWww
pmcJaxpvudL2zwuopDEdQZ/OLqxZpQQtnLVcOQmyj1rdRhg6Q37P5y68TNj0rtM4XtblnmUiMVa2ANFPXCmBUIXji1/SQZDCfz1VBHF1btpTc
HQiQyWwRiIZH1nuEayI+FnK1bvvgNAMZFoS/b1Z1LXvtPvfGFKwZe7aLIKuc2qzkZ5W+yVU8SW2hrbRCHuCMwN+N8ajx+keBAZlMOY9ZQRLyc
JVjqvSC4kfmvGiYM0YECX2M1SrbOC2/7vpHGBzUbBFkLF
EOF
```

Bootstrap Script (cont.)

- Example 3: Permission and ownership

```
# Change the ownership of /home/fadel/.ssh and its files from root(the user
# who created the folder) to the user fadel
chown -R fadel.fadel /home/fadel/.ssh
# Change the permission of the folder and the file
chmod 700 /home/fadel/.ssh
chmod 600 /home/fadel/.ssh/authorized_keys
# Add fadel to the sudoers user to have permission to be root
echo "fadel  ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/fadel
# Add to the content of the webpage the ips
public_ipv4=$(curl -s http://169.254.169.254/latest/meta-data/public-ipv4)
local_ipv4=$(curl -s
http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:b9:85:fe:6c:ae/local-ipv4s/)
echo My Public IP is : $public_ipv4  >> /var/www/html/index.html
echo My Local IP is : $local_ipv4  >> /var/www/html/index.html
```

404 - Not Found

Note: You will receive an error. The curl was wrong : The MAC is for different machine.

The right command is :
 http://169.254.169.254/latest/meta-data/local-ipv4/

local_ipv4=$(curl -s http://169.254.169.254/latest/meta-data/local-ipv4/)

- ➔ How to update the script?
  - ◆ From the EC2 console > Actions > Instance Settings
- ➔ Can we use Git repository for the script?
  - ◆ Use Jenkins

AWS CLI

- Control AWS services and resources through script
- Automation is easier with the CLI

AWS CLI - installation and configuration

1. Installation:

```
$> pip install awscli
```

https://docs.aws.amazon.com/cli/latest/userguide/installing.html

2. Configuration:

```
$> aws configure
```

https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html

AWS CLI - Authorization

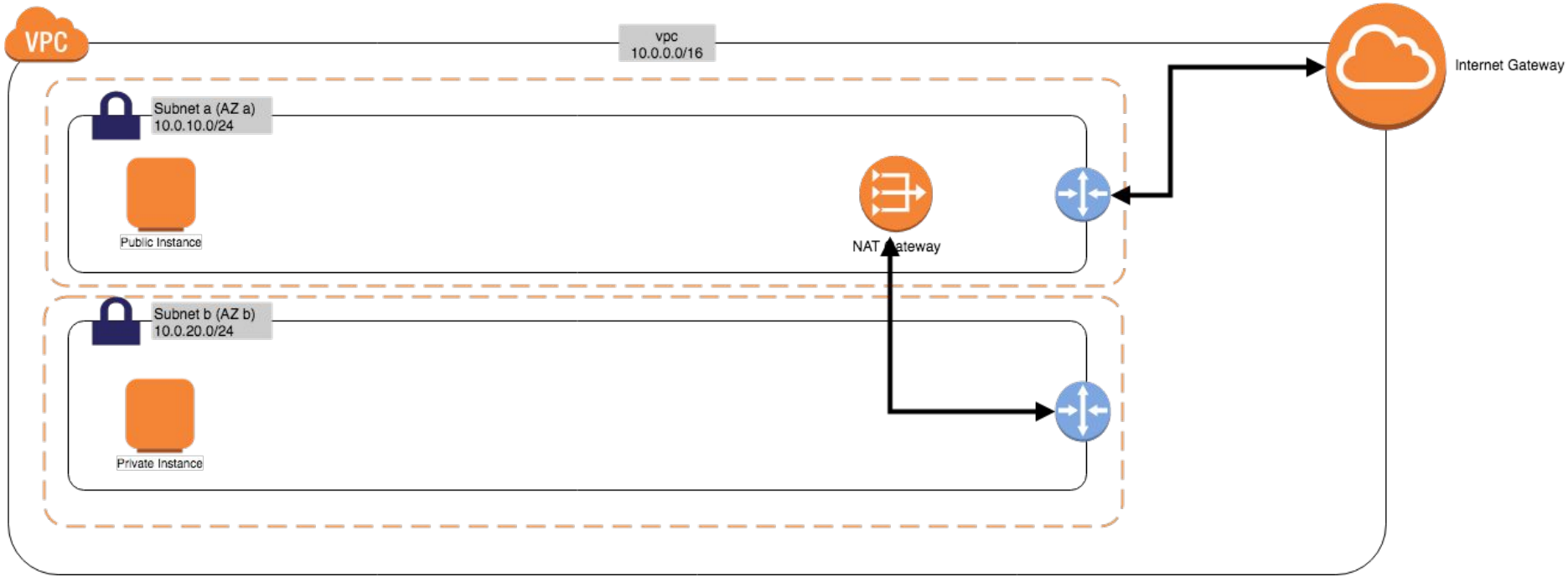
- IAM User Credentials:
  - Access Key
  - Secret Key
  - To create Admin user : AWS Console -> AMI -> Users -> Add user -> User name “Administrators” -> check both Access Type boxes. -> Next -> Create User

- Roles:
  - EC2 role
  - Lambda role
  - To create Admin Role : AWS Console -> AMI -> Roles -> Create Role -> EC2 -> AdministratorAccess -> Role Name “AdminFullAccess”

Review of mostly used AWS CLI commands

```
apt -get update -y
apt-get install python3-pip
pip3 install awscli
aws configure
aws help
aws ec2 help
aws ec2 describe-instances
```

Environment we have so far..



```
apt-get update -y

apt-get install python3-pip

pip3 install awscli

aws configure

aws ec2 describe-instances

aws ec2 describe-volumes

aws ec2 describe-vpcs  --query 'Vpcs[*].{ID:VpcId}' --output text
```

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16

aws ec2 create-subnet --cidr-block 10.0.0.0/17 --vpc-id vpc-03938376907823078 --availability-zone us-east-1a

aws ec2 create-subnet --cidr-block 10.0.128.0/17 --vpc-id vpc-03938376907823078 --availability-zone
us-east-1c

aws ec2 create-internet-gateway

aws ec2 attach-internet-gateway --vpc-id vpc-03938376907823078 --internet-gateway-id  igw-047ae6de9733b792d

aws ec2  create-route-table --vpc-id vpc-03938376907823078

aws ec2 create-route --route-table-id rtb-0747899edc9f679c9 --destination-cidr-block 0.0.0.0/0 --gateway-id
igw-047ae6de9733b792d

aws ec2 describe-subnets | grep us-east-1a

aws ec2 describe-route-tables | grep 10.0.0

aws ec2 associate-route-table --route-table-id rtb-0747899edc9f679c9 --subnet-id subnet-0d5ff97b62f73ae31

aws ec2 create-security-group --description 'open ports 22 and 80 to the world' --group-name open-ssh-and-web
--vpc-id vpc-03938376907823078
```

```
# Group name does not work with the non default VPC
#aws ec2 authorize-security-group-ingress --group-name open-ssh-and-web --protocol tcp --port 22 --cidr 0.0.0.0/0

aws ec2 authorize-security-group-ingress --group-id sg-0d2b86703c21f7cf6 --protocol tcp --port 80 --cidr 0.0.0.0/0

aws ec2 run-instances --image-id ami-04681a1dbd79675a5 --count 1 --instance-type t2.micro --key-name e91_key
--security-group-ids sg-0d2b86703c21f7cf6 --subnet-id subnet-0d5ff97b62f73ae31

aws ec2 run-instances --image-id ami-04681a1dbd79675a5 --count 1 --instance-type t2.micro --key-name e91_key
--security-group-ids sg-0d2b86703c21f7cf6 --subnet-id subnet-0d5ff97b62f73ae31 --associate-public-ip-address

aws ec2 describe-instances --filter "Name=instance-state-name,Values=running" | grep INSTANCES

aws ec2 terminate-instances --instance-ids i-0d7aa731bbd63eb5e
```

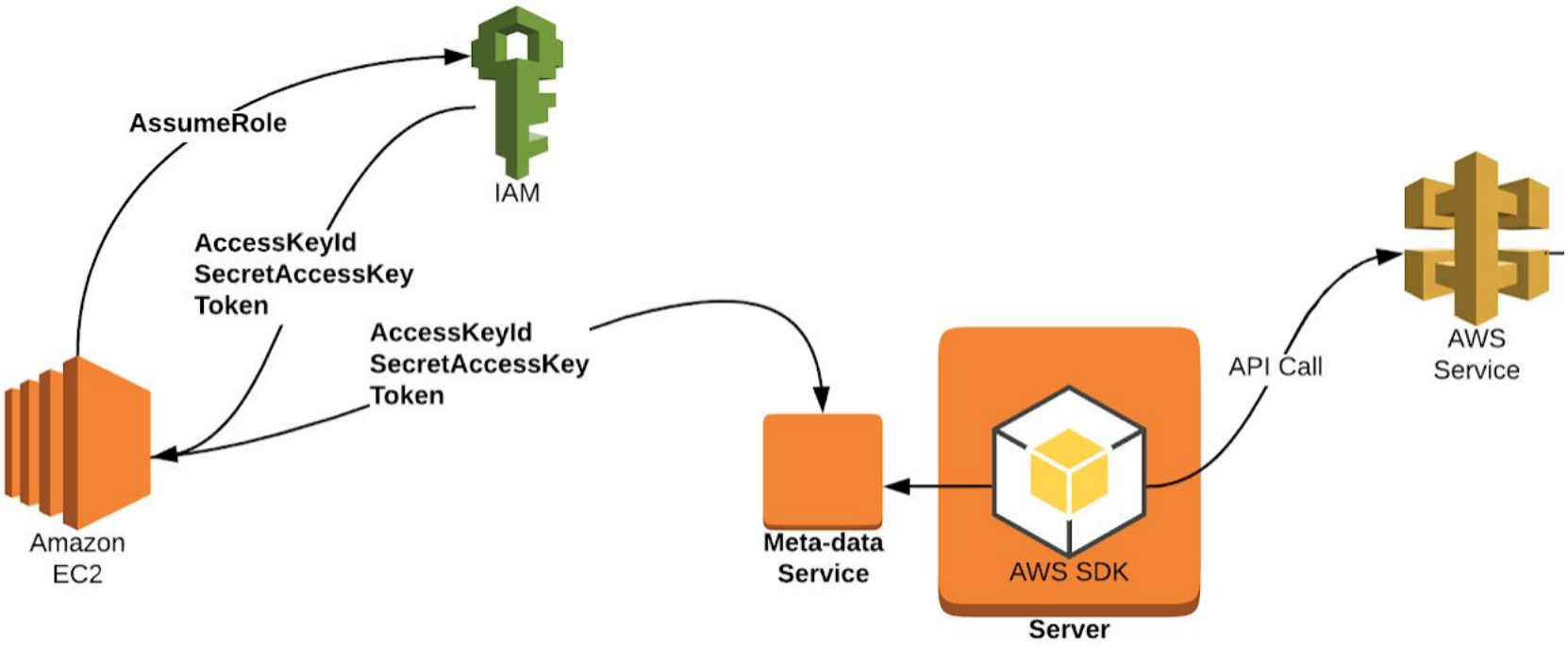
```
#!/bin/bash

VPCID=$(aws ec2 create-vpc --cidr-block 10.0.0.0/16 --output text | grep VPC | awk '{print $7}')

PUBLICSUB=$(aws ec2 create-subnet --cidr-block 10.0.0.0/17 --vpc-id $VPCID --availability-zone us-east-1a --output
text | awk '{print $9}')
PRIVATESUB=$(aws ec2 create-subnet --cidr-block 10.0.128.0/17 --vpc-id $VPCID --availability-zone us-east-1a
--output text | awk '{print $9}')

IGW=$(aws ec2 create-internet-gateway | awk '{print $2}')
aws ec2 attach-internet-gateway --vpc-id $VPCID  --internet-gateway-id  $IGW
```

Role and Security Token Service



```
ssh EC2WithRoleIP 'curl http://169.254.169.254/latest/meta-data/iam/security-credentials/EC2FullAccess'
{
  "Code" : "Success",
  "LastUpdated" : "2018-10-02T19:08:01Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIARGRHK43OKHVSUYP2",
  "SecretAccessKey" : "j5vJ1dcXMNNtWuKv/RSkmpKDdOH9b9ufOixhz4Mq",
  "Token" : "FQoGZXIvYXdzEPX ..=”,
  "Expiration" : "2018-10-03T01:28:14Z"
}

$ export AWS_ACCESS_KEY_ID=ASIARGRHK43OKHVSUYP2
$ export AWS_SECRET_ACCESS_KEY=j5vJ1dcXMNNtWuKv/RSkmpKDdOH9b9ufOixhz4Mq
$ export AWS_SESSION_TOKEN=FQoGZXIvYXdzEPX...<remainder of security token>
$ aws ec2 describe-instances
```

Load Balancer

- Layer 7 HTTP/HTTPS Application Layer Load Balancer
  - Target groups and Rules to route paths
- Layer 4 Network Load Balancer
  - Extreme performance is required.
  - Low latency while handling millions of requests per second
- Classic Load Balancer: Layer 4 and can use layer 7
  - X-Forwarding
  - sticky session

Live Demo Load Balancer

Auto scaling

- ➔ Launch Configuration :
  - ◆ EC2 launch configuration with userdata
- ➔ Auto scaling :
  - ◆ Auto scaling configuration using the Launch configuration and Auto Scaling Groups

Live Demo Auto Scaling

404 - Not Found

How to update the Script in launch configuration ?

1) Copy the launch configuration and choose new name
    a) Update the user data
    b) Make sure all config is ok
2) Replace the Autoscaling with the new Launch Configuration
3) Remove instances: for each instance in the LB
    a) terminate the instance
    b) wait for the new instance to come up.
    c) Test the new instance functionality
4) Remove the old launch configuration

Resources

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html

https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html

https://docs.aws.amazon.com/cli/latest/userguide/cli-roles.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html