








EC2 ?????

-   EC2  
- Rstudio   

??? ??? EC2 ????? ???

이 레슨에서는 AWS CLI를 사용하여 EC2 인스턴스를 실행하는 방법을 보여줍니다. 이 레슨에서는 AWS CLI를 사용하여 EC2 인스턴스를 실행하는 방법을 보여줍니다. 이 레슨에서는 AWS CLI를 사용하여 EC2 인스턴스를 실행하는 방법을 보여줍니다.

?? ?????

- AWS CLI를 설치하고 구성합니다.
- Bash 셸을 사용하여 AWS CLI를 실행합니다.
- AWS CLI를 사용하여 EC2 인스턴스를 실행합니다.

?? ???????

1. EC2 인스턴스를 실행하는 데 필요한 리소스를 확인합니다.
2. EC2 인스턴스를 실행하는 데 필요한 리소스를 확인합니다.
3. EC2 인스턴스를 실행하는 데 필요한 리소스를 확인합니다.
4. EC2 인스턴스를 실행하는 데 필요한 리소스를 확인합니다.
5. EC2 인스턴스를 실행하는 데 필요한 리소스를 확인합니다.

?? ??

1. ?? EC2 ????? ??

이 레슨에서는 AWS CLI를 사용하여 EC2 인스턴스를 실행하는 방법을 보여줍니다. 이 레슨에서는 AWS CLI를 사용하여 EC2 인스턴스를 실행하는 방법을 보여줍니다. 이 레슨에서는 AWS CLI를 사용하여 EC2 인스턴스를 실행하는 방법을 보여줍니다.

```
aws ec2 run-instances \
  --image-id ami-04df9ee4d3dfde202 \
  --instance-type m5.large \
  --key-name your-key-pair-name \
  --security-group-ids sg-your-security-group-id \
  --subnet-id subnet-your-subnet-id \
```

```
--count 1 \  
--associate-public-ip-address
```

❏ : --image-id , --instance-type , --key-name , --security-group-ids , --subnet-id ❏ ❏❏❏ ❏❏❏ ❏❏❏❏ .

2. ??? ?????? ?? ? ?????? ??

RDP❏ ❏❏❏ ❏❏❏ ❏❏❏❏ ❏❏❏ ❏ , ❏❏❏ ❏❏❏❏❏ ❏❏❏❏❏ .

3. ??? AMI ??

❏ Bash ❏❏❏❏❏ ❏❏❏❏ ❏❏❏ AMI❏ ❏❏❏ ❏❏❏❏❏ :

```
#!/bin/bash  
  
# AMI ❏❏❏❏❏  
create_ami() {  
    local instance_id=$1  
    local ami_name=$2  
    local ami_description=$3  
  
    aws ec2 create-image \  
        --instance-id "$instance_id" \  
        --name "$ami_name" \  
        --description "$ami_description" \  
        --no-reboot \  
        --query 'ImageId' \  
        --output text  
}  
  
# AMI ❏❏❏❏❏❏❏❏❏❏❏  
check_ami_status() {  
    local ami_id=$1  
    aws ec2 describe-images \  
        --image-ids "$ami_id" \  
        --query 'Images[0].State' \  
        --output text  
}
```

```

# 1. 创建AMI
echo "实例ID: $instance_id:"
read instance_id

echo "AMI名称: $ami_name:"
read ami_name

echo "AMI描述: $ami_description:"
read ami_description

echo "AMI名称: $ami_name"
ami_id=$(create_ami "$instance_id" "$ami_name" "$ami_description")

echo "AMI名称: $ami_name. AMI ID: $ami_id"
echo "AMI名称: $ami_name. AMI ID: $ami_id"

while true; do
    status=$(check_ami_status "$ami_id")
    if [ "$status" = "available" ]; then
        echo "AMI名称: $ami_name. AMI ID: $ami_id"
        break
    elif [ "$status" = "failed" ]; then
        echo "AMI名称: $ami_name. AMI ID: $ami_id"
        exit 1
    else
        echo "AMI名称: $ami_name. AMI ID: $ami_id. AMI ID: $ami_id"
        sleep 30
    fi
done

echo "AMI名称: $ami_name. AMI ID: $ami_id"

```

4. 创建AMI? 创建AMI? 创建AMI? 创建AMI? 创建AMI?

1. Bash 脚本 创建AMI 脚本 创建AMI 脚本 创建AMI 脚本 创建AMI 脚本 :

```

#!/bin/bash

# 1. 创建AMI
generate_password() {

```

```

    openssl rand -base64 12
}

# 生成随机字符串
launch_instance() {
    local instance_number=$1
    local initial_password=$2
    aws ec2 run-instances \
        --image-id ami-your-custom-ami-id \
        --count 1 \
        --instance-type m5.large \
        --key-name your-key-pair-name \
        --security-group-ids sg-your-security-group-id \
        --subnet-id subnet-your-subnet-id \
        --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=WindowsInstance-
'$instance_number'}]' \
        --user-data "net user Administrator '${initial_password}'" \
        --query 'Instances[0].InstanceId' \
        --output text
}

# 获取实例的公共IP地址
get_public_ip() {
    local instance_id=$1
    aws ec2 describe-instances \
        --instance-ids $instance_id \
        --query 'Reservations[0].Instances[0].PublicIpAddress' \
        --output text
}

# SSM 通过实例ID和公共IP地址来更改密码
change_password() {
    local instance_id=$1
    local new_password=$2
    aws ssm send-command \
        --instance-ids "$instance_id" \
        --document-name "AWS-RunPowerShellScript" \
        --parameters "commands=[\"net user Administrator '${new_password}'\"]" \
        --output text
}

```

```
# 設定 初期パスワード
read -s -p "初期パスワードを入力してください: " INITIAL_PASSWORD
echo
```

```
# 設定 インスタンスの数を取得
read -p "インスタンスの数を指定してください: " INSTANCE_COUNT
```

```
# CSV ファイルを作成
echo "Instance ID,Public IP,Username,Password" > rdp_info.csv
```

```
# 初期パスワードを生成
echo "$INSTANCE_COUNT 個のインスタンスを作成します ..."
instance_ids=()
for i in $(seq 1 $INSTANCE_COUNT); do
    instance_id=$(launch_instance $i "$INITIAL_PASSWORD")
    instance_ids+=($instance_id)
    echo "インスタンス $i の ID: $instance_id"
done
```

```
# インスタンスの起動を確認
echo "起動を確認中..."
aws ec2 wait instance-running --instance-ids "${instance_ids[@]}"
echo "起動完了。5分間待機します ..."
sleep 300
```

```
# 設定 初期パスワードを生成
for instance_id in "${instance_ids[@]}"; do
    echo "インスタンス $instance_id の初期パスワードを生成します"
```

```
# 設定 IP アドレスを取得
public_ip=$(get_public_ip $instance_id)
```

```
# 設定 新しいパスワードを生成
new_password=$(generate_password)
```

```
# 設定 パスワードを変更
echo "パスワードを変更します ..."
change_password "$instance_id" "$new_password"
```

- `aws ec2 run-instances` command is used to create instances.
- `aws ec2 run-instances` command is used to create instances.
- `--image-id`, `--instance-type`, `--key-name`, `--security-group-ids`, `--subnet-id` are the options used to create instances.

rdp_info.csv

74

??

Rstudio ????? ??? ??

Amazon EC2 Ubuntu 24 . . . !

Amazon EC2 User data User Data .

install_rstudio.txt .

```
#!/bin/bash

# update indices
apt update -qq

# install two helper packages we need
apt install -y --no-install-recommends software-properties-common dirmngr

# add the signing key (by Michael Rutter) for these repos
# To verify key, run gpg --show-keys /etc/apt/trusted.gpg.d/cran_ubuntu_key.asc
# Fingerprint: E298A3A825C0D65DFD57CBB651716619E084DAB9
wget -qO- https://cloud.r-project.org/bin/linux/ubuntu/marutter_pubkey.asc | tee -a
/etc/apt/trusted.gpg.d/cran_ubuntu_key.asc

# add the repo from CRAN -- lsb_release adjusts to 'noble' or 'jammy' or ... as needed
add-apt-repository -y "deb https://cloud.r-project.org/bin/linux/ubuntu $(lsb_release -cs)-cran40/"

# Install R
apt install -y --no-install-recommends r-base

# Install RStudio dependencies
apt install -y gdebi-core

# Install R package dependencies
apt install -y build-essential

wget https://download2.rstudio.org/server/jammy/amd64/rstudio-server-2024.12.1-563-amd64.deb
```

```
gdebi -n rstudio-server-2024.12.1-563-amd64.deb
```

```
# Add rstudio user and set password
```

```
useradd -m -s /bin/bash rstudio
```

```
echo "rstudio:rstudio" | chpasswd
```

```
R -e "options(HTTPUserAgent = sprintf('R/%s R (%s)', getRversion(), paste(getRversion(), R.version['platform'],  
R.version['arch'], R.version['os'])))"); install.packages('TwoSampleMR', repos = c('https://mrcieu.r-  
universe.dev/bin/linux/noble/4.4/', 'https://p3m.dev/cran/_linux_/noble/latest', 'https://cloud.r-project.org'))"
```

❏ ❏ ❏ ❏ **EC2** ❏ ❏ ❏ ❏ . ❏ ❏ ❏ **Amazon EC2** ❏ ❏
❏ ❏ ❏ **Command line** ❏ ❏ ❏ ❏ .

```
export AMI_ID={Ubuntu ❏ ❏ OS❏ ❏ Base AMI}
```

```
export INSTANCE_TYPE=t3.large
```

```
export KEY_NAME={❏❏❏ KeyName}
```

```
# ❏❏❏ security group (sg-❏❏❏ ❏❏ )❏ SnapshotId❏ ❏❏ ❏❏❏ .
```

```
aws ec2 run-instances --image-id "${AMI_ID}" --instance-type "${INSTANCE_TYPE}" --key-name "${KEY_NAME}" \
\
--user-data file://install_rstudio.txt \
--block-device-mappings
'{"DeviceName":"/dev/sda1","Ebs":{"Encrypted":false,"DeleteOnTermination":true,"Iops":3000,"SnapshotId":"sn-  
ap-0dbe62bb8f1f21357","VolumeSize":100,"VolumeType":"gp3","Throughput":125}}' \
--network-interfaces '{"AssociatePublicIpAddress":true,"DeviceIndex":0,"Groups":["sg-  
0d2f7724e68ddff15","sg-0e2c103f2a28a9be7"]}' \
--credit-specification '{"CpuCredits":"unlimited"}' --tag-specifications
'{"ResourceType":"instance","Tags":[{"Key":"Name","Value":"rstudio server"}]}' \
--metadata-options '{"HttpEndpoint":"enabled","HttpPutResponseHopLimit":2,"HttpTokens":"required"}' \
--private-dns-name-options '{"HostnameType":"ip-  
name","EnableResourceNameDnsARecord":true,"EnableResourceNameDnsAAAARecord":false}' \
--count "1" --region ap-northeast-2
```

EC2 ❏ ❏❏❏❏ . ❏❏❏ ❏❏ ❏❏❏ ❏❏❏ ❏❏ ❏ **AMI** ❏❏❏ ❏❏❏ .

```
aws ec2 create-image \
```

```
--instance-id {❏❏❏ ❏❏ ❏❏❏ ❏❏❏ } \
```

```
--name "My Rstudio server" \
```

```
--description "An AMI for my Rstudio server with TwoSampleMR R package" \
```

--region ap-northeast-2

awscli --region ap-northeast-2 **AMI** --help

awscli --region ap-northeast-2 **AMI** --help

AWS CLI --region ap-northeast-2 **Quota** --help

https://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html

awscli --region ap-northeast-2 **Quota** --help