








# 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
}

# 生成随机密码并设置环境变量
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地址来设置密码
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 インスタンス数
read -p "インスタンス数を入力してください: " 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 個の初期パスワードを生成します ..."
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
```

```
# 初期パスワードを生成
echo "5 個の初期パスワードを生成します ..."
aws ec2 wait instance-running --instance-ids "${instance_ids[@]}"
echo "5 個の初期パスワードを生成しました。 ..."
sleep 300
```

```
# 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 ID: $instance_id"
```

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

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

```
# 新しいパスワードを設定
echo "インスタンス $instance_id の ID: $instance_id"
change_password "$instance_id" "$new_password"
```

```
# CSV
```

```
echo "$instance_id,$public_ip,Administrator,$new_password" >> rdp_info.csv
```

```
echo "      $instance_id  " "
```

```
done
```

```
echo "  . RDP  rdp_info.csv  ."
```

```
echo "  :  " "
```

:

- EC2 .
- (AZ) .
- --image-id , --instance-type , --key-name , --security-group-ids , --subnet-id .

## 5. ?? CSV ?? ??

rdp\_info.csv .

: 30 74 .

```
COMMAND 1b3da9d1-b43a-45ea-9218-c9a84de97775 0 0 AWS-RunPowerShellSc
Pending Pending 1 3600
ALARMSCONFIGURATION False
CLOUDWATCHOUTPUTCONFIG False
INSTANCEIDS i-07645e5b837e01de1
NOTIFICATIONCONFIG
COMMANDS net user Administrator '2cgrAlq1aB4NOY4s'
Instance i-07645e5b837e01de1 processed
All instances launched and processed. RDP information saved in rdp_info.csv
Note: Passwords have been changed to randomly generated passwords.
bash windows-ec2-launcher-test.sh 74.10s user 19.63s system 17% cpu 8:57.42 total
Downloads cat rdp_info.csv
Instance ID,Public IP,Username,Password
i-05beb7f065f362ff0,34.228.82.48,Administrator,ABzamyLIHpVKTbNE
i-003609439dfccbfde,34.228.12.178,Administrator,BCQF06LZRay839Uv
i-03476b7a63d960265,3.94.111.223,Administrator,+27A//cBQZMJc0a9
i-0fd242034d9642b72,184.72.123.251,Administrator,XKWeoVukubQAUFz5
i-074e311ac5600fc83,34.229.13.242,Administrator,3HhMh750rjk/s+X5
i-0593965a77c0cbb7,54.91.100.59,Administrator,Wob7Q1/X7Rvkh282
i-025c2cc88ed07a15a,54.208.216.63,Administrator,VEV5c5L5GhmCzz
i-04aa56101b27769c5,52.204.169.43,Administrator,zeaX0FQWJ3iukTnQ
i-01f2197f0358ed1d5,18.212.253.208,Administrator,XZcq+Sex6dKyW64v
i-090fb5caa1b5a216,54.226.42.49,Administrator,LIzm//wNj4CyG4b8
i-0c41f291fe45d608,54.196.158.143,Administrator,50lUblVhndW3fdg+
i-0e7e48d3a99c443e0,3.80.32.199,Administrator,TMrDt18YtjBvyf9+
i-00378f991f56fb440,54.160.159.151,Administrator,8XqB0AYK6zu06V0+
i-030f540064f148f6e,54.226.244.201,Administrator,WDBXSxF+9a7U1451
i-0efc06006927d3e29,3.84.241.49,Administrator,9BxppHu8DGu0Q6vg
i-09ef1aa8de6580a92,50.19.8.0,Administrator,cZ0n5jyfA2+75Qre
i-08d9047258ffdc82,54.91.160.147,Administrator,hug8IBnKvrlEmdw3
i-0d35a9d989e6f452b,54.227.113.152,Administrator,XeeY1n79svVmwPjd
i-0313fa99bf49b851e,3.85.57.73,Administrator,hZkpARS6FSyKzNST
i-01b550d62590956e9,34.203.214.42,Administrator,nEjGF46CdKF2aRiz
i-0a8e878707f64b7f1,18.214.99.84,Administrator,86CovK/CfMoX0m2
i-0b3544c16c3ef44b2,54.165.96.206,Administrator,gA3GwvyKS060vaCP
i-06ff191f36f4848bf,3.87.83.39,Administrator,BVf1gh5q8rXhhjGC
i-002609e7e4bb87b2b,54.83.181.113,Administrator,sie6xab+YE7HzIXR
i-0416039a981a71378,100.26.35.17,Administrator,ZcIHQnNEXInGx1
i-02ed60054d20f14c,100.27.203.58,Administrator,QaQad/Y2Hi69K2VZ
i-041aedff50d147144,54.144.169.215,Administrator,QAIskQf17zJLGHrX
i-05e885e24e1b6817f,3.82.19.248,Administrator,KkbN94R2FbMESPgp
i-06a790a0c93b9a9e5,54.145.48.223,Administrator,tM5H4E8X3goBRQM
i-07645e5b837e01de1,34.228.8.154,Administrator,2cgrAlq1aB4NOY4s
```

??





# 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](https://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html)

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