**Create Amazon EC2 Instance using AWS CLI**

If you want to create a new EC2 instance from AWS CLI, you need to login to aws running instance and connect with a user with **AmazonEC2FullAccess.**

First you need to create a new user via IAM service.

Services 
Find all services 
Home 
Resource Groups 
IA M 
search for IAM 
Compu 
EC2 
Ligntsail 
ECR 
ECS 
EKS 
Lambda 
aatcn 
Elastic Beanstalk 
Serverless Application Repository 
Storage 
FSX 
java2786 
Gol 
Robotics 
AWS RoboMaker 
Blockchain 
Amazon Managed alockcnain 
Satellite 
Ground Station 
Management & Govemance 
CloudWatcn 
AWS Auto Scaling 
CloudFormation 
Analytics 
Athena 
EMR 
CloudSearcn 
Elasticsearcn Service 
o 
Kinesis 
QuickSight 
Data Pipeline 
AWS Glue 
You can find here as 
Secur , Identity, 
Co iance 
IAM 
well 
Resource Access Manager 
Business App 
Alexa tor Busine 
Amazon Chime 
WorkMaiI 
End User Com 
Workspaces 
AppStream 2.0 
WorkDocs 
WorkLink 
Intemet Of Thi 
IOT core 
Amazon FreeRT 

while creating user select Access type as Programmatic access

Services v 
Resource Groups 
2 
3 
java2786 
4 
Global 
Add user 
Set user details 
You can add multiple users at once witn the same access type and permissions. Learn more 
5 
User name* 
Select AWS access type 
demo-userl 
O Add another user 
Select now these users will access AWS_ Access keys and autogenerated passwords are provided in the last step. Learn more 
Access type* 
* Required 
Programmatic access 
Enables an access key ID and secret access key tor the AWS API, CLI, SDK, and 
other development tools. 
AWS Management Console access 
Enables a password that allows users to sign-in to the AWS Management Console. 
Cancel 
Next: Permissions 

policy to assigned here is AmazonEC2FullAccess so that this user can EC2 instances

Services v 
C) abc 
Resource Groups 
3 
java2786 
4 
Global 
Add user 
Set permissions 
m Add user to group 
5 
Copy permissions trom 
existing user 
Attach existing policies 
directly 
e way to manage user's permissions by job functions. Learn more 
Add user to an existing group or create a new one. using groups is a best-pra 
Add user to group 
Create group 
Q Search 
Group 
Refresh 
Attach AmazonEC2FullAccess Policy 
Attached policies 
AmazonEC2FullAccess and 1 more 
Cancel 
Previous 
Showing 1 result 
Next: Tags 

Create user and note down Access Key Id and Secret Access Key (may be you want to download csv file)

Download .csv 
User 
demo-user 
Access key ID 
AKIAlBKUDGR6KOQYSZ4Q 
Secret access key 
snow 
Close 

Now connect to running instance using

Use putty

or

In linux terminal

$ chmod 400 downloaded-key-pair.pem

$ ssh -i downloaded-key-pair.pem ec2-user@<EC2-INSTANCE-PUBLIC-IP-ADDRESS>

After successful connection, In Amazon Instance terminal, change to root user

$ sudo su

Find your current region(3 different ways to find it)

1. aws 
   https://console.aws.amazon.com/console/home?region=us-east-1 
   Services v 
   Resource Groups v 
   AWS Management Console 
   AWS services 
   Find Services 
   You can enter names, keywords or acronyms. 
   Q Example: Relational Database Service, database, RDS 
   Recently visited services 
   java2786 
   N. Virginia A 
   US East (N. Virginia) 
   US East (Ohio) 
   US West (N. California) 
   US West (Oregon) 
   Support 
   current region value 
   Access resources on 
   Access the Manag 
   Console Mobile 
   Explore AWS 
   Current Region Name 
   Asia Pacific (Mumbai) 
   Asia Pacific (Seoul) 
   Asia Pacific (Singapore) 
   Asia Pacific (Sydney) 
   Asia Pacific (Tokyo) 
   Canada (Central) 
   ELI (Frankfurt) 
   ELI (Ireland) 

1. <https://docs.aws.amazon.com/general/latest/gr/rande.html>
2. $ aws ec2 describe-region

Now configure our create user

Cec2-user@ip-172-31-41-194 -IS sudo su 
Croot@ip-172-31-41-194 aws configure 
AWS Access Key ID [None]: AKIAIBKUDGR6KOQYSZ4Q 
AWS secret Access Key [None]: KIY4SK+PTR/GnvJ6CZOQ) 
Default region name [None]: us-east-I 
Default output format [None] : 

Now we are ready to create new EC2 Instance

Create new instance with this user using following command:

$ aws ec2 run-instances --image-id <ami-id> --count <number\_of\_instances> --instance-type <instance\_type> --key-name <key\_pair\_name> --region <region> --security-group-ids <security\_group\_id>

Or

$ aws ec2 run-instances \

--image-id <ami-id> \

--count <number\_of\_instances> \

--instance-type <instance\_type> \

--key-name <key\_pair\_name> \

--region <region> \

--security-group-ids <security\_group\_id>

Find ami\_Id

aws 
1. Choose AMI 
Services v 
Resource Groups v 
java2786 
N. Virginia 
Support 
2. Choose Instance Type 
3 Configure Instance 
4 _ Add Storage 
5. Add Tags 
6. Configure Security Group 
Review 
Step 1: Choose an Amazon Machine Image (AMI) 
Free tier only (j) 
Amazon Linux 
Free tier eligible 
Red Hat 
Free tier eligible 
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type ami-0080e4c5bc078760e 
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS 
and Java The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages. 
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes 
mand line tools, Python, Ruby, Perl, 
Red Hat Enterprise Linux 7.6 (HVM), SSD Volume Type - ami-011b3ccf1bd6db744 (64-bit x86) / ami- 
Oe3688b4a755ad736 (64-bit Arm) 
Red Hat Enterprise Linux version 7.6 (HVM), EBS General Purpose (SSD) Volume Type 
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes 
SUSE Linux Enterprise Server 15 (HVM), SSD Volume Type - ami-06ea7729e394412c8 
Cancel and Exit 
Select 
64-öit (x86) 
Select 
@ 64-bit (x86) 
O 
64-bit (Arm) 

**Find instance-type**

Choose AMI 
2. Choose Instance Type 
3. Configure Instance 
4. Add Storage 
5. Add Tags 
6. Configure Security Group 
T Review 
Step 2: Choose an Instance Type 
Amazon EC2 provides a wide selection of instance types optimized to Vit different use cases. Instances are virtual servers that can run applications. They have varying combinations ot CPU, memory, st' 
networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs. 
Filter by: 
All instance types v 
Current generation v Show/Hide Column 
Instance type 
Currently selected: t2micro (Variable ECL's, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB 
Family 
Genera] puppse 
Type 
t2.nano 
t2micro 
Free tier eligible 
vCPUs 
Memory (GiB) 
0.5 
ory, EBS onlY) 
Instance Storage (GB) 
(j) 
ESS only 
ESS only 
EBS-Optimized Available 
Network Performance 
(j) 
Low to Moderate 
Low to Moderate 
s, 

**Find security-groups-id**

aws 
https://cansale.aws.amazon.com/ec2/v2/home?region—us-east-1*Hame: 
Services v 
Resource Groups v 
EC2 Dashboard 
Events 
Tags 
click here 
Reports 
Limits 
INSTANCES 
Instances 
Launch Templates 
Spot Requests 
Reserved Instances 
Dedicated Hosts 
Scheduled Instances 
Resources 
You are using the following Amazon EC2 resources in the US East (N. Virginia) region: 
0 
2 
0 
Running Instances 
Dedicated Hosts 
Volumes 
Key Pairs 
Placement Groups 
0 
0 
0 
3 
find security id here 
Elastic IPS 
Snapshots 
Load Balancers 
Security Groups 
Learn more about the latest in AWS Compute trom AWS relnvent by viewing the EC2 Videos 
Create Instance 

Create Security Group 
Actions v 
Q Filter by tags and attributes or search by keyword 
Name 
Group ID 
sg-0939ba7db6b60Sb34 
sg-09bO d8fS3e98326 
sg-e701c 5 
Group Name 
Ec2LinuxSecurity 
WindowsSecurity 
default 

$ aws ec2 run-instances --image-id ami-0080e4c5bc078760e --count 1 --instance-type t2.micro --key-name KeyLinux --region us-east-1 --security-group-ids sg-0939ba7db6b605b34

Croot@ip-172-31-41 
-194 clear 
Croot@ip-172-31-41 
-194 aws ec2 run-instances 
--7 mage 
0080e4c5bc078760e - 
--instance-type t2.micro 
--key-name 
-count 1 
--region us-east-I 
-security-group-ids sg-0939ba7db6b605b34 
x 
-id ami- 
KeyLinu 

copy InstanceId from output and

find StateReason in output: it is pending, wait for few seconds

{   
 "Instances": [   
 {   
 "Monitoring": {   
 "State": "disabled"   
 },   
 "PublicDnsName": "",   
 "StateReason": {   
 "Message": "pending",   
 "Code": "pending"   
 },   
 "State": {   
 "Code": 0,   
 "Name": "pending"   
 },   
 "EbsOptimized": false,   
 "LaunchTime": "2019-03-07T12:59:18.000Z",   
 "PrivateIpAddress": "172.31.81.37",   
 "ProductCodes": [],   
 "VpcId": "vpc-d203b4a8",   
 "CpuOptions": {   
 "CoreCount": 1,   
 "ThreadsPerCore": 1   
 },   
 "StateTransitionReason": "",   
 "InstanceId": "i-000dfbfd58c3edce9",   
 "ImageId": "ami-0080e4c5bc078760e",   
 "PrivateDnsName": "ip-172-31-81-37.ec2.internal",   
 "KeyName": "KeyLinux",   
 "SecurityGroups": [   
 {   
 "GroupName": "Ec2LinuxSecurity",   
 "GroupId": "sg-0939ba7db6b605b34"   
 }   
 ],   
 "ClientToken": "",   
 "SubnetId": "subnet-f72445d9",   
 "InstanceType": "t2.micro",   
 "NetworkInterfaces": [   
 {   
 "Status": "in-use",   
 "MacAddress": "12:de:69:c0:b0:3c",   
 "SourceDestCheck": true,   
 "VpcId": "vpc-d203b4a8",   
 "Description": "",   
 "NetworkInterfaceId": "eni-0486ae20bfbea7503",   
 "PrivateIpAddresses": [   
 {   
 "PrivateDnsName": "ip-172-31-81-37.ec2.inte  
rnal",   
 "Primary": true,   
 "PrivateIpAddress": "172.31.81.37"   
 }   
 ],   
 "PrivateDnsName": "ip-172-31-81-37.ec2.internal",   
 "Attachment": {   
 "Status": "attaching",   
 "DeviceIndex": 0,   
 "DeleteOnTermination": true,   
 "AttachmentId": "eni-attach-096c6f4c09d94be49",  
   
 "AttachTime": "2019-03-07T12:59:18.000Z"   
 },   
 "Groups": [   
 {   
 "GroupName": "Ec2LinuxSecurity",   
 "GroupId": "sg-0939ba7db6b605b34"   
 }   
 ],   
 "Ipv6Addresses": [],   
 "OwnerId": "571705239115",   
 "SubnetId": "subnet-f72445d9",   
 "PrivateIpAddress": "172.31.81.37"   
 }   
 ],   
 "SourceDestCheck": true,   
 "Placement": {   
 "Tenancy": "default",   
 "GroupName": "",   
 "AvailabilityZone": "us-east-1d"   
 },   
 "Hypervisor": "xen",   
 "BlockDeviceMappings": [],   
 "Architecture": "x86\_64",   
 "RootDeviceType": "ebs",   
 "RootDeviceName": "/dev/xvda",   
 "VirtualizationType": "hvm",   
 "AmiLaunchIndex": 0   
 }   
 ],   
 "ReservationId": "r-06af027fa0bc2aec7",   
 "Groups": [],   
 "OwnerId": "571705239115"   
}

once instance status is running, you can verify using following command

aws ec2 describe-instances --instance-id copied\_id

note down PublicDnsName or PublicIp

And you are done...