Class 15:

EC2 Dashboard

**1, Launch Templates**

Create a template to create an instance fast with fixed details

Launch template name*– required*:

Template version description:

AMI:

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type

ami-032930428bf1abbff

Catalog: Quick Startvirtualization: hvmarchitecture: 64-bit (x86)

Instance type: t2.microFree tier eligible

Key pair name: AWSB26-VPC-key pair

Networking platform: VPC

Subnet:

Security groups: Allow all

Resource tags: Key Value Resource types (instances)

Network interfaces: we don’t add network interfaces because we can’t change subnets if we want

Advanced details: user : used for coding like installation of nginx

|  |
| --- |
| #! /bin/bash  Yum update -y  Yum install -y nginx  Service nginx start |

* This template is created on single subnet, here if I want to create 10 instance all goes to same subnet, if I want to deploy in other subnets, what we can do.

Here we can manipulate an include in script for that we need **Access key & secrete key**

go to

IAM -> Users -> ADD USER

**User name:**

Access type: **Programmatic access**

**Next**

**|**

Attach existing policies directly

 [AdministratorAccess](https://console.aws.amazon.com/iam/home?region=us-east-1" \l "/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAdministratorAccess" \t "'_blank') 

|

Create user

|

Now we get **Access key & secrete key** Copy and save it

Access key ID: AKIAZ6OFFUXWRGOU32ZT

Secret access key: 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

Now go to instances

|

Select any instance and copy public DNS

|

Putty

|

**aws configure**

AWS Access Key ID [None]: AKIAZ6OFFUXWRGOU32ZT

AWS Secret Access Key [None]: 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

Default region name [None]: us-east-1

Default output format [None]: json

Aws ec2 describe-vpcs (this command is used to get all vpc’s)

Yum install -y jq (installing json parser to get cmd line colorful)

Aws ec2 describe-vpcs | jq (this command is used to get all vpc’s with pipe (shift+above enter butten) symbol and execute jq and output is colorful)

* So now we write script to place our instance template in 3 subnets

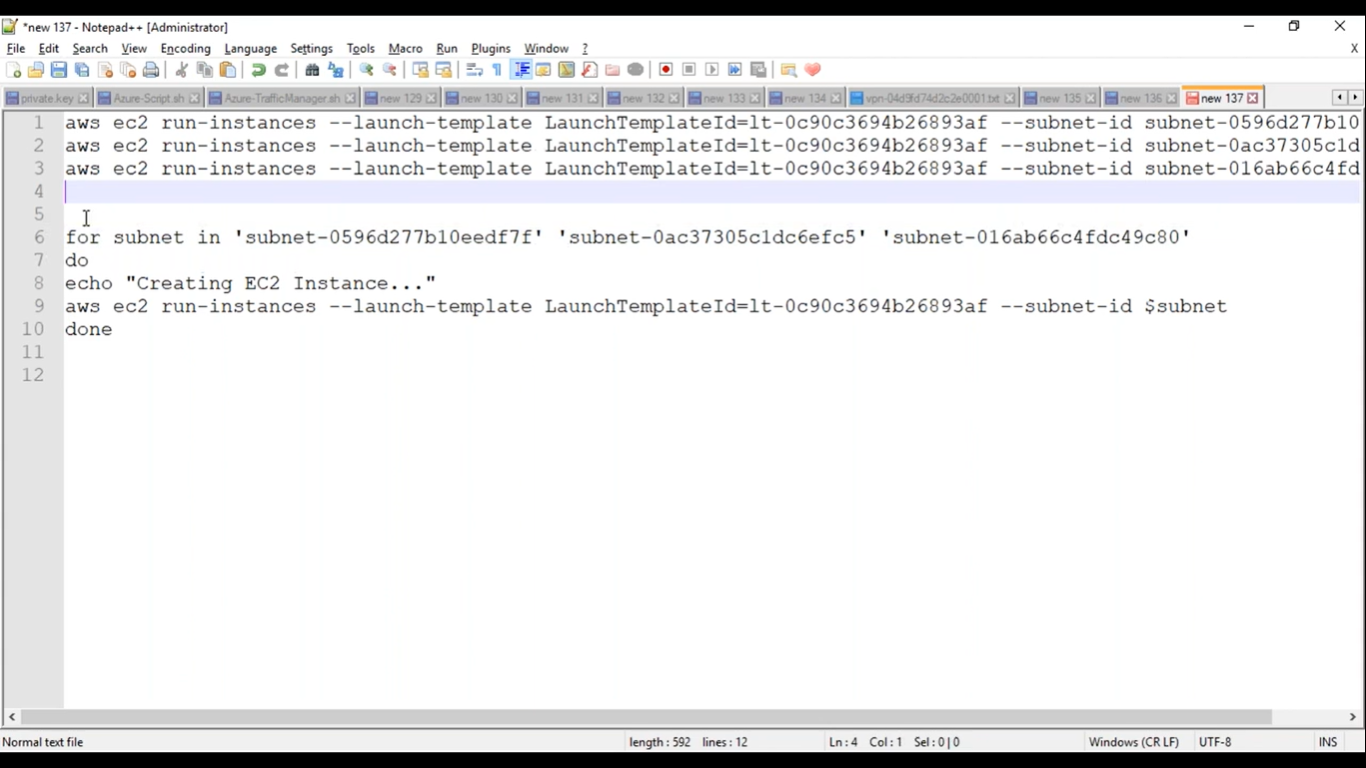
for subnet in 'subnet-057b872e980ff2a3e' 'subnet-0da79e8831a01c067' 'subnet-0560e3b1ee449d50e'

do

echo "Creating EC2 Instance..."

aws ec2 run-instances --launch-template LaunchTemplateId=lt-0ece7a959206facf7 --subnet-id $subnet

done



**2, Spot Requests**

Regular instances that we create are called “on-demand instances”

The free area that is empty is “spot request instance”

In server we create instance on spot request because it is cheaper, they provide us it for cheaper because it is empty and not using by anyone, if aws got demand for their server they will take it back from us.

When to use spot request

1, one-time request (if they get on demand they will remove. we use it for temporary testing)

2, maintain the capacity -> if I want “20 CPU and 40 GB” aws will give from different instance types like

T2.micro \* 20

T3.medium

M4.large

If one family got demand, aws will provide another type of instance type

3, Specific duration without downtime max of 6hrs

With in 6 hrs. the server wont shutdown

->Go to

Ec2 Navigation

|

Spot requests

**3, Reserved Instances**

Reserved Instances is one of the cost saving techniques (i.e., options to save money to company)

If we buy instance of 1-3 years duration, we can save 40% of on-demand price

**4, Scheduled Instances**

Scheduled Instances is same as Reserved Instances, but it is served for short time 1 week to 1 year

**5, Capacity Reservations**

Capacity Reservations is used to reserve “instance types” for particular time

Like t2.nano,t2.micro………………..

**6, saving plans**

saving plans is for reserving of 1/hr. for that we get discount compare to on-demand price