CS 7001-03: Report for AWS Lab 3 - Platform/Application Provisioning and Auto Scaling Adaptation

Chanmann Lim

cl9p8@mail.mail.missouri.edu

March 31, 2015

1. Use `as-create-launch-config` command to create an AWS launch-configuration with options `--instance-type` set to 't1.micro', `--aws-credential-file` set to 'new-cf', `--image-id` set to 'ami-aa0237c2', `--key` set to 'new-key', `--group` set to 'new-group' and `--launch-config` set to 'new-lc':

```
# as-create-launch-config        \
  --instance-type t1.micro       \
  --aws-credential-file new-cf \
  --image-id ami-aa0237c2        \
  --key new-key                  \
  --group new-group              \
  --launch-config new-lc
```

Where 'ami-aa0237c2' is the custom AMI created from an instance of type 't1.micro'.

2. Use `as-create-auto-scaling-group` command to create an auto scaling group and pass the name 'new-asg' as its argument along with options `--aws-credential-file` set to 'new-cf', `--availability-zones` set to 'us-east-1b', `--launch-configuration` set to 'new-lc', `--load-balancers` set to 'ELB', `--max-size` set to '5' and `--min-size` set to '1':

```
# as-create-auto-scaling-group new-asg \
  --aws-credential-file new-cf          \
  --availability-zones us-east-1b       \
  --launch-configuration new-lc         \
  --load-balancers ELB                  \
  --max-size 5                          \
  --min-size 1
```

Where 'ELB' is an already created load balancer.

3. Use `as-put-scaling-policy` command to create or update auto scaling policy and pass the name 'scale-up-policy' as its argument along with options `--auto-scaling-group` set to 'new-asg', `--adjustment=2` to scale up two instances, `--type` set to 'ChangeInCapacity', and `--cooldown` set to '240' for re-evaluating interval in seconds:

```
# as-put-scaling-policy scale-up-policy \
  --auto-scaling-group new-asg \
  --adjustment=2 \
  --type ChangeInCapacity \
  --cooldown 240
```

4. We should use Amazon CloudWatch to create an Alarm with 'new-asg' CPUUtilization metric to measure the computing resources then define threshold condition(e.g. CPUUtilization $>= 50$) and to trigger the policy we need to add AutoScaling Action then set **Whenever this alarm**: 'State is ALARM', **From the group**: 'new-asg', and **Take this action**: 'scale-up-policy'.

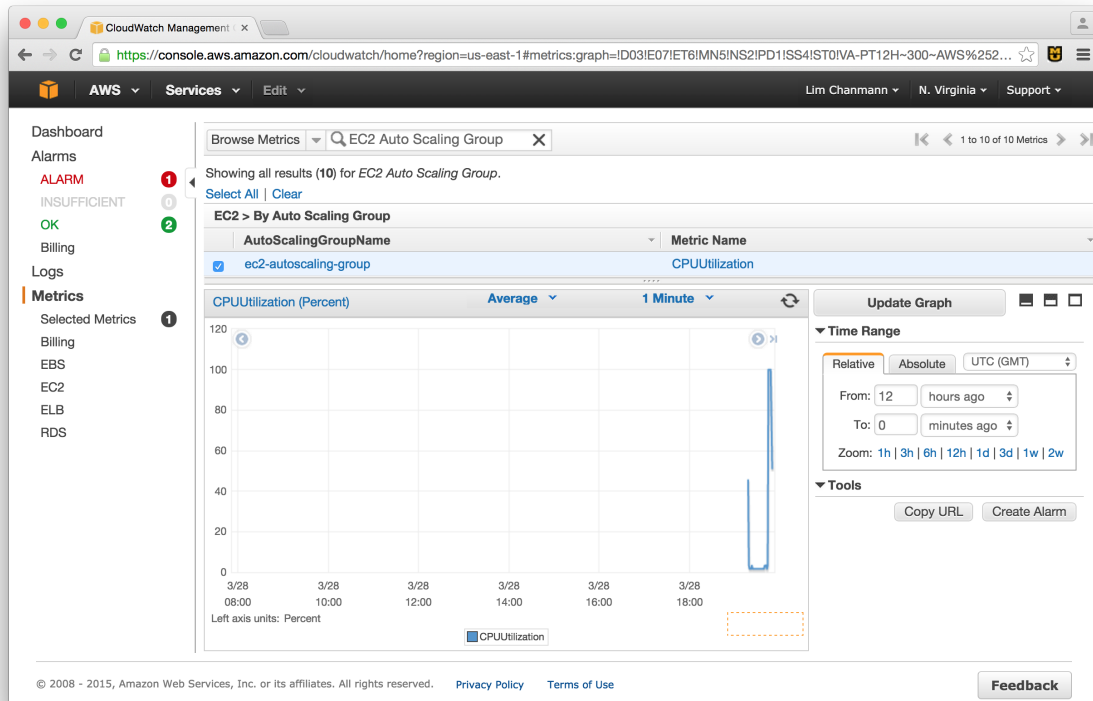**5.** The screenshots taken in Steps 3.5.3 and 3.5.4:
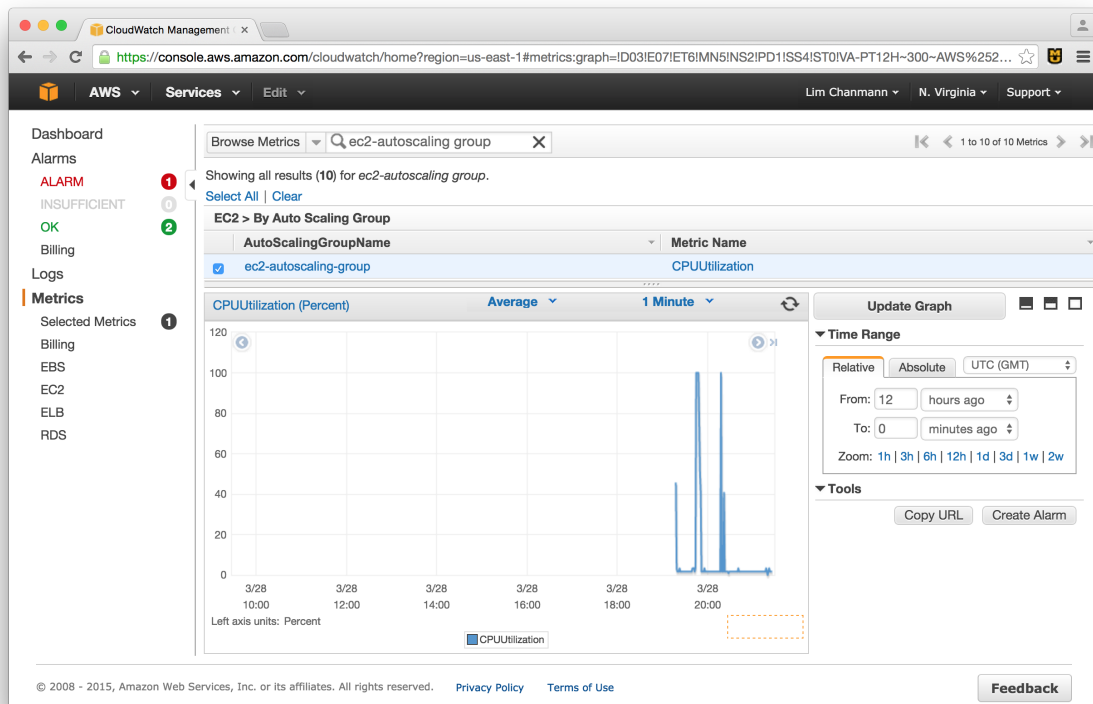

Figure 1: CPUUtilization metric during Apache Server benchmarking

Figure 2: CPUUtilization metric after stopping benchmarking