

|  |  |
| --- | --- |
| **Public DNS** | ec2-54-153-104-29.us-west-1.compute.amazonaws.com |
| **User name** | Administrator |
| **Password** | https://d2zusxcxuz7yd0.cloudfront.net/db9dc991063d2274fd962246154d884b8d9cabf7/ec2/clear.cache.gifGet Password  cST)XAd3X3U |

Instructions

**Lab Documents:**

* https://github.com/paulnguyen/cmpe281/blob/master/aws/1-aws-vpc-ec2-ami.md
* [http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html (Links to an external site.)Links to an external site.](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html)

In this Lab, you will be setting up your AWS account with SSH keys to access EC2 and then will launch a Amazon Linux "Free-Tier" AMI to install a LAMP Stack.  You will then deploy some PHP code to remotely create "Load" on the Linux VM and observe the CPU utilization.

**Key Steps Are:**

1. Setup Key Pair for EC2 and Download PEM file.
2. Create VPC:  cmpe281 (Using Wizard).
3. Launch EC2 Instance.
4. Connect to EC2 Instance.
5. PHP Setup on EC2 Linux AMI.
6. PHP Test.
7. Create PHP AMI Image.

**Using VisualOps with AWS:**

See Doc Here:  [http://docs.visualops.io/getting\_started/amazon\_web\_services.html (Links to an external site.)Links to an external site.](http://docs.visualops.io/getting_started/amazon_web_services.html)

**Key Steps:**

1. Create AWS User for VisualOps to Use
2. Grant User Access to Manage EC2 Instances  
   (*Note:  for the class only Read Access is needed.*   
   *We will not be deploying from VisualOps*)
3. Enter AWS User Credentials in VisualOps

**2. Lab 2:**

**Lab Documents:**

* [https://github.com/paulnguyen/cmpe281/blob/master/aws/2-aws-as-elb-classic.md (Links to an external site.)Links to an external site.](https://github.com/paulnguyen/cmpe281/blob/master/aws/2-aws-as-elb-classic.md)
* [http://docs.aws.amazon.com/autoscaling/latest/userguide/as-register-lbs-with-asg.html (Links to an external site.)Links to an external site.](http://docs.aws.amazon.com/autoscaling/latest/userguide/as-register-lbs-with-asg.html)
* [http://docs.aws.amazon.com/autoscaling/latest/userguide/autoscaling-load-balancer.html (Links to an external site.)Links to an external site.](http://docs.aws.amazon.com/autoscaling/latest/userguide/autoscaling-load-balancer.html)
* [http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-create-https-ssl-load-balancer.html (Links to an external site.)Links to an external site.](http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-create-https-ssl-load-balancer.html)
* [http://docs.aws.amazon.com/autoscaling/latest/userguide/as-add-availability-zone.html#as-add-az-console (Links to an external site.)Links to an external site.](http://docs.aws.amazon.com/autoscaling/latest/userguide/as-add-availability-zone.html#as-add-az-console)

In this Lab, you will be creating a small three instance auto-scaled cluster using the Linux AMI from your previous Lab.  You will then configure an Elastic Load Balancer and create load on instances to observe Cloud Elasticity at work.

**Key Steps Are:**

1. Create or Select a Launch Configuration
2. Create an Auto Scaling Group
3. Using a Load Balancer With an Auto Scaling Group