Linux Academy

Hands-On Training

Configuring an Auto Scaling Application

Contents

Introduction 1

Scenario 1

Creating a Launch Configuration 1

Create an Auto Scaling Group 2

Load Balancers 5

Self-Healing Applications 5

Introduction

In this lab, we will be using concepts learned in the **AWS Certified Solutions Architect** course, in particular, the videos *Concept to Production: Building a Highly Available WordPress Application* and *Auto Scaling and Bootstrapping*.

To begin, log into the AWS Console with the credentials provided on the lab page.

Scenario

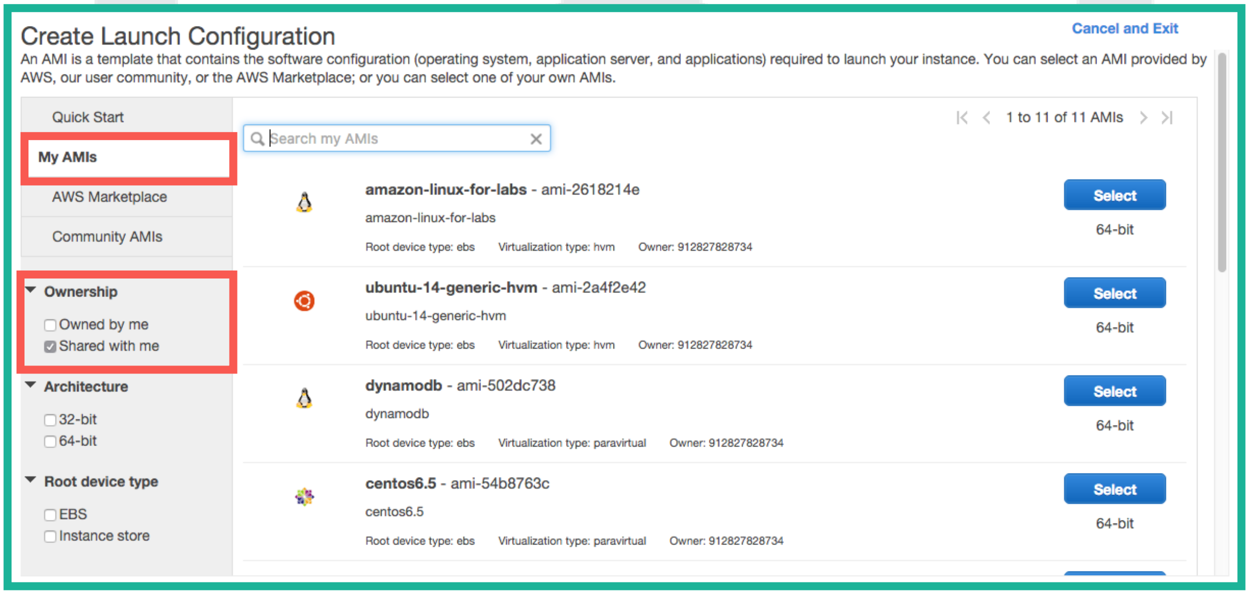
We are creating a highly-available web application called Grumpy Cat that will use AWS Auto Scaling to scale both up and down as needed, based on peak CPU loads.

Creating a Launch Configuration

Navigate to the **EC2 dashboard** from the AWS Console and select **Launch Configurations**, located in the left bar under **Auto Scaling**.

Press **Create Auto Scaling group**. AWS will provide you with a page giving you an overview of Auto Scaling group creation. Click **Create launch configuration**.

From here, you will need to select an Amazon Machine Image (AMI). On the left menu, select **My AMIs**, and check the **Shared with me** box under **Ownership**. An AMI titled *autoscaling-grumpycat* will be available from the list. Select this AMI.



Create Launch Configuration

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Ownership
□ Owned by me
☑ Shared with me

Architecture
□ 32-bit
□ 64-bit

Root device type
□ EBS
□ Instance store

amazon-linux-for-labs - ami-2618214e
amazon-linux-for-labs
Root device type: ebs Virtualization type: hvm Owner: 912827828734
64-bit

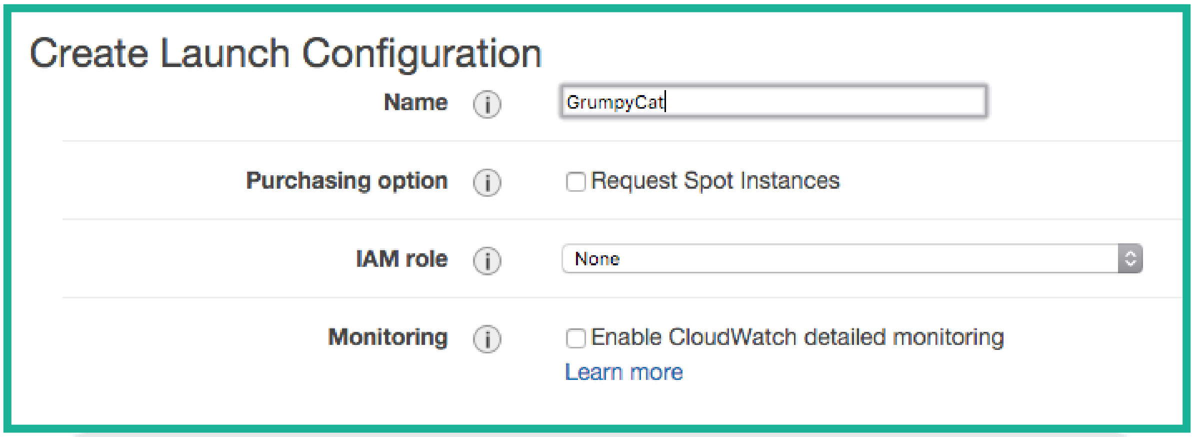
ubuntu-14-generic-hvm - ami-2a4f2e42
ubuntu-14-generic-hvm
Root device type: ebs Virtualization type: hvm Owner: 912827828734
64-bit

dynamodb - ami-502dc738
dynamodb
Root device type: ebs Virtualization type: paravirtual Owner: 912827828734
64-bit

centos6.5 - ami-54b8763c
centos6.5
Root device type: ebs Virtualization type: paravirtual Owner: 912827828734
64-bit

Use the default *t2.micro* plan, and press **Next: Configure details**.

We will be calling our launch configuration *GrumpyCat* (**Name**) since this is the application it will be used for. Do **not** check **Request Spot Instances**, and leave the **IAM role** set to *none*. Also, leave **Monitoring** unchecked.



Create Launch Configuration

Name: GrumpyCat

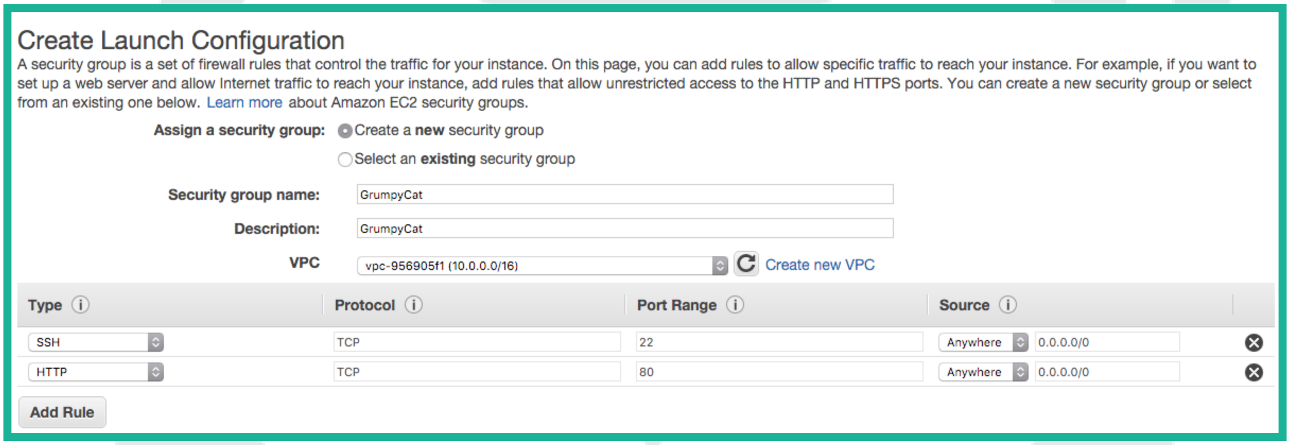
Purchasing option: □ Request Spot Instances

IAM role: None

Monitoring: □ Enable CloudWatch detailed monitoring

Press **Next: Add Storage**, and leave everything with the default settings. Go to **Next: Configure Security Group**.

We want to **Create a new security group**. We used *GrumpyCat* as both the security group name, and the description. If there is an option for a VPC with LinuxAcademy in the title, use this; otherwise, use the default option.



Create Launch Configuration

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: GrumpyCat

Description: GrumpyCat

VPC: vpc-956905f1 (10.0.0.0/16)

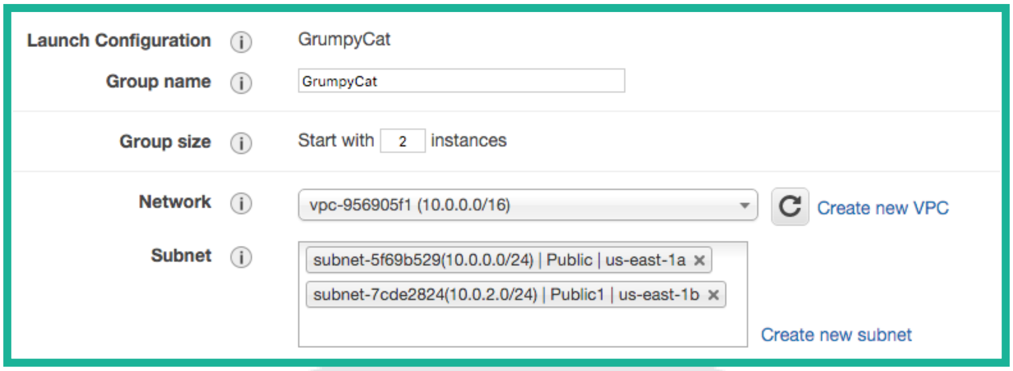
Type	Protocol	Port Range	Source	
SSH	TCP	22	Anywhere 0.0.0.0/0	⊗
HTTP	TCP	80	Anywhere 0.0.0.0/0	⊗

Press **Add Rule**, and add a rule for *HTTP (port 80)*. Select **Review**, and look through your settings to ensure everything is configured correctly before pressing **Create launch configuration**.

You will be prompted to select an existing or create a new key pair. Since we do not need a key pair, choose *Proceed without a key pair* from the dropdown menu, and read and check off the acknowledgment below. Press **Create launch configuration**.

Create an Auto Scaling Group

As before, we will be calling our group *GrumpyCat*. We want to start with a **Group size** of *2* instances, for high-availability purposes. Select the available VPC under **Network**, and add both subnets to the subnet area.



Launch Configuration ⓘ GrumpyCat

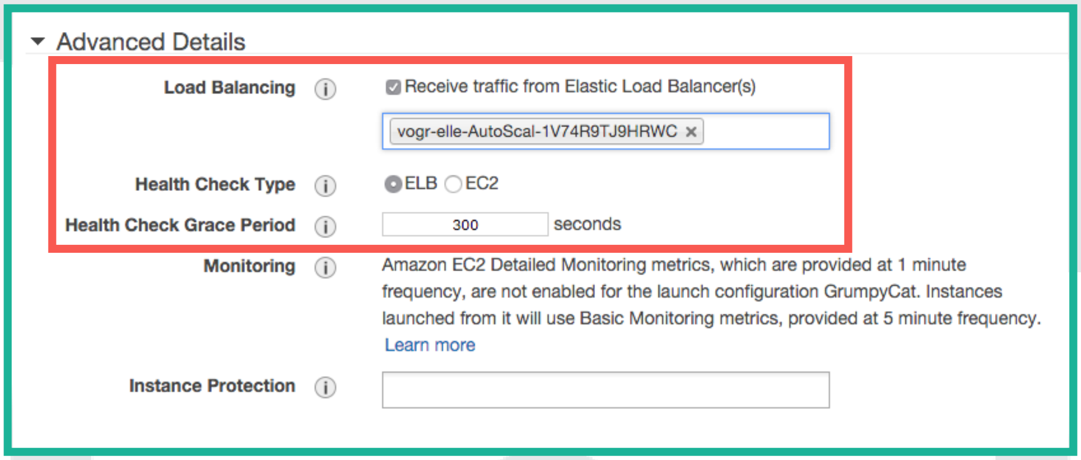
Group name ⓘ GrumpyCat

Group size ⓘ Start with 2 instances

Network ⓘ vpc-956905f1 (10.0.0.0/16)

Subnet ⓘ subnet-5f69b529(10.0.0.0/24) | Public | us-east-1a ×
subnet-7cde2824(10.0.2.0/24) | Public1 | us-east-1b ×

Below the subnet area is an **Advanced Details** section. Expand this so we can configure the load balancing portion of the application. Check **Receive traffic from Elastic Load Balancer(s)** and in the box below, select the single ELB available. Set the **Health Check Type** to *ELB*. You can leave the **Health Check Grace Period** at the default *300* seconds.



▾ Advanced Details

Load Balancing ⓘ ☑ Receive traffic from Elastic Load Balancer(s)

vogr-elle-AutoScal-1V74R9TJ9HRWC ×

Health Check Type ⓘ ● ELB ○ EC2

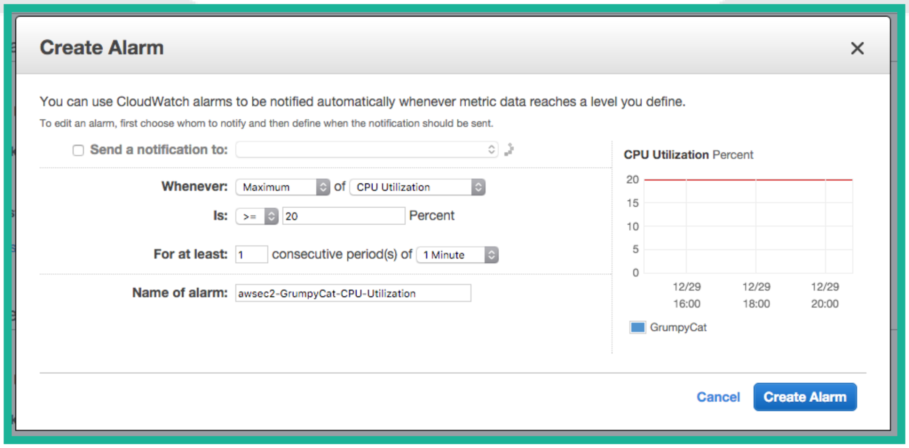
Health Check Grace Period ⓘ 300 seconds

Monitoring ⓘ Amazon EC2 Detailed Monitoring metrics, which are provided at 1 minute frequency, are not enabled for the launch configuration GrumpyCat. Instances launched from it will use Basic Monitoring metrics, provided at 5 minute frequency.

Instance Protection ⓘ

Press **Next: Configure scaling properties**. We want to **Use scaling policies to adjust the capacity of this group**. You will be presented with two options for actions and alerts: Increasing and Decreasing the group size. First, we must define the minimum and maximum amount of instances, however. Set it to **Scale between** *2* **and** *4* **instances**.

Within the **Increase Group Size** area, press **Add new alarm**. Uncheck the option to send out a notification, and change **Whenever** to be a *Maximum of CPU Utilization [that] is >= 20 Percent*. Set **For at least** to be *1* **consecutive period(s) of** *1 Minute*. Press **Create Alarm**.



Create Alarm ×

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

□ Send a notification to:

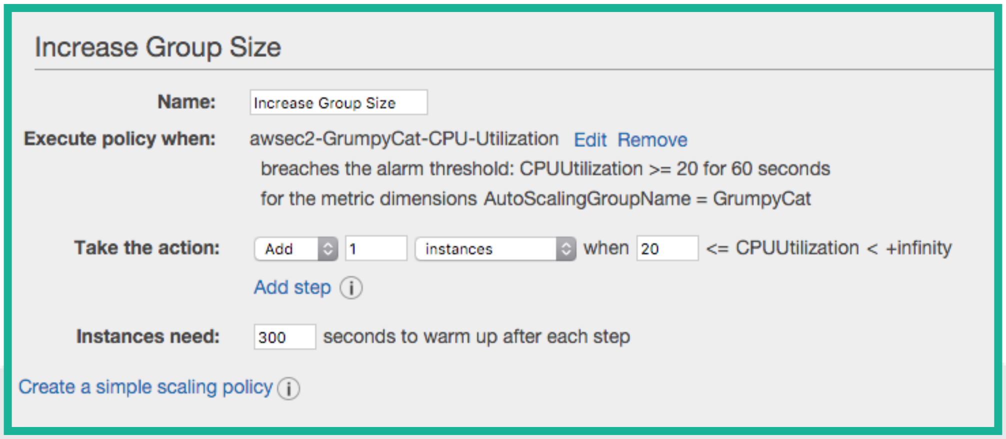
Whenever: Maximum of CPU Utilization

Is: >= 20 Percent

For at least: 1 consecutive period(s) of 1 Minute

Name of alarm: awsec2-GrumpyCat-CPU-Utilization

From here, we now need to define the action we want AWS to take when the alarm threshold is hit. In the **Take the action** area, we want to *Add 1* **instance**. Set the **Instances needed** to *300* **seconds to warm up after each step**.



Increase Group Size

Name: Increase Group Size

Execute policy when: awsec2-GrumpyCat-CPU-Utilization Edit Remove
breaches the alarm threshold: CPUUtilization >= 20 for 60 seconds
for the metric dimensions AutoScalingGroupName = GrumpyCat

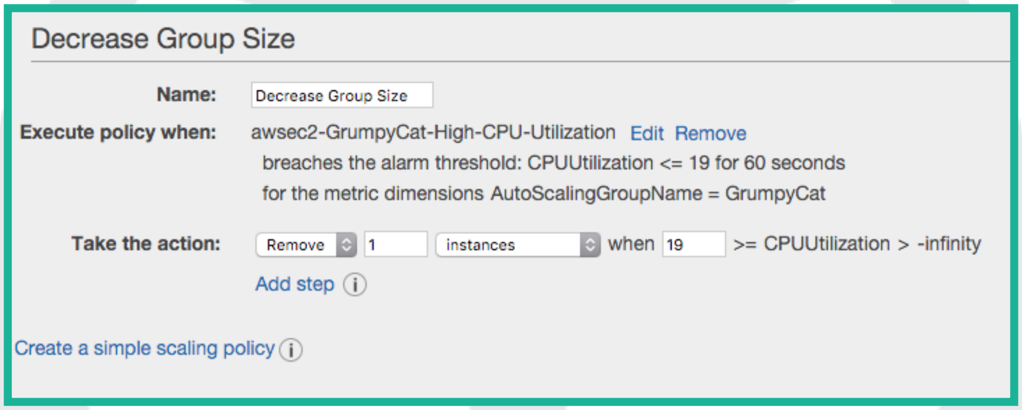
Take the action: Add 1 instances when 20 <= CPUUtilization < +infinity

Add step ⓘ

Instances need: 300 seconds to warm up after each step

Create a simple scaling policy ⓘ

Under **Decrease Group Size**, also **Add new alarm**. Again, deselect the send notification option. Set **Whenever** to a *Minimum of CPU Utilization [that] is <= 19 Percent* **for at least** *1 consecutive period of 1 Minute*. **Create Alarm**, then set **Take the action** to *Remove 1 instances*.



Decrease Group Size

Name: Decrease Group Size

Execute policy when: awsec2-GrumpyCat-High-CPU-Utilization Edit Remove
breaches the alarm threshold: CPUUtilization <= 19 for 60 seconds
for the metric dimensions AutoScalingGroupName = GrumpyCat

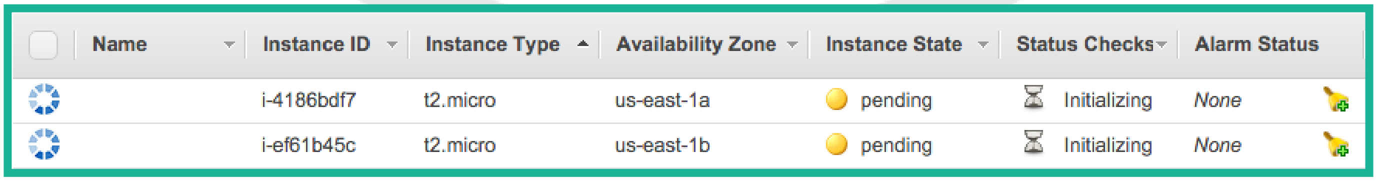
Take the action: Remove 1 instances when 19 >= CPUUtilization > -infinity

Add step ⓘ

Create a simple scaling policy ⓘ

Press **Next: Configure Notifications**, ignore the warning, then press **Next: Configure Tags**. We do not need to configure tags for this instance, but to foster good habits, you may wish to give the instance a name value. Press **Review**, and review your details. Click **Create Auto Scaling group**.

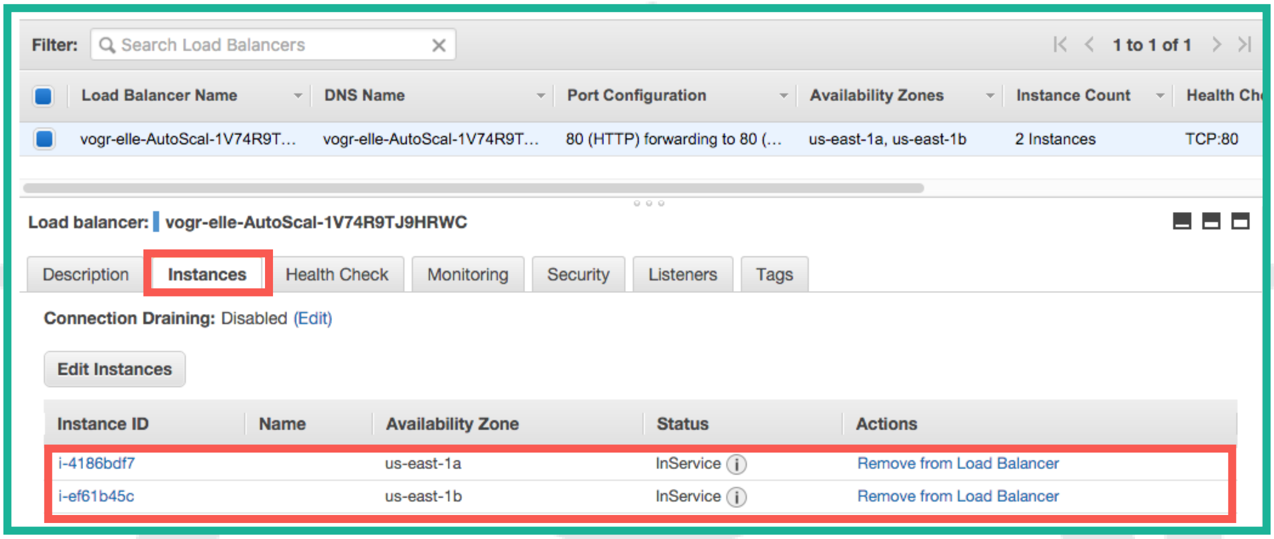
If you go to your **Instances**, you will see two EC2 instances starting up.



□	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
		i-4186bdf7	t2.micro	us-east-1a	● pending	⌛ Initializing	None
		i-ef61b45c	t2.micro	us-east-1b	● pending	⌛ Initializing	None

Load Balancers

If you navigate to your **Load Balancer** dashboard, located within the EC2 dashboard, under Load Balancing on the left menu, you can see the created load balancer. Select it, and then press the **Instances** tab below. Here you will see two instances, one for each zone. Wait for the **Status** of both load balancers to read *InService* before continuing. This can take up to ten minutes.



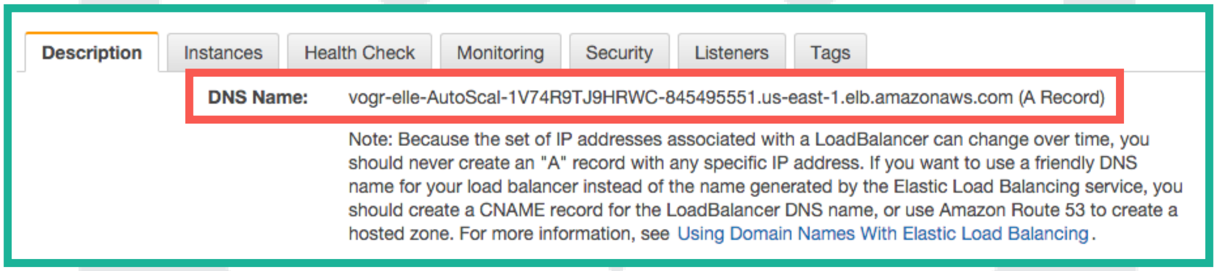
Load Balancer Name	DNS Name	Port Configuration	Availability Zones	Instance Count	Health Ch
vogr-elle-AutoScal-1V74R9T…	vogr-elle-AutoScal-1V74R9T…	80 (HTTP) forwarding to 80 (…	us-east-1a, us-east-1b	2 Instances	TCP:80

Load balancer: vogr-elle-AutoScal-1V74R9TJ9HRWC

Connection Draining: Disabled (Edit)

Instance ID	Name	Availability Zone	Status	Actions
i-4186bdf7		us-east-1a	InService	Remove from Load Balancer
i-ef61b45c		us-east-1b	InService	Remove from Load Balancer

Return to the **Description** tab, and copy and **DNS Name**, then paste it into your web browser. You should see your Grumpy Cat.



DNS Name: vogr-elle-AutoScal-1V74R9TJ9HRWC-845495551.us-east-1.elb.amazonaws.com (A Record)

Note: Because the set of IP addresses associated with a LoadBalancer can change over time, you should never create an "A" record with any specific IP address. If you want to use a friendly DNS name for your load balancer instead of the name generated by the Elastic Load Balancing service, you should create a CNAME record for the LoadBalancer DNS name, or use Amazon Route 53 to create a hosted zone. For more information, see Using Domain Names With Elastic Load Balancing.

Self-Healing Applications

To test if our instance is truly self-healing, we will now need to return to **Instances**, under the left menu. Here, we need to terminate one of the running instances: If our application is self-healing, another instance should be brought up, once the initial instance is terminated. You may also terminate the other running instance if so desired. Because we created rules that call for a minimum of two instances, that, too, will bring up a replacement instance. You may need to refresh the page to witness this.

Return to the link in your web browser and refresh to ensure Grumpy Cat is still available, even after your instances have been deleted and recreated.