Media Sharing Website

Part 1: Media uploads

Table of Contents

Overview

The objective of this lab is to create a simple media sharing website. In this first part, we will create the core architecture of the system, providing basic features such as browsing, uploading and deleting content. Media content will be limited to images, but the concepts covered here also apply to other types of media such as documents (PDF, RTF, presentations, etc.), music, videos, etc.

In this first part, the system will provide a web interface for users to browse and store images.

Step 1 – Media storage

Images could be stored on EBS volumes but we need to provision capacity in advance, and manage the operation of scaling up this storage layer by adding volumes. Besides, those volumes need to be attached to an EC2 instance to serve the content via HTTP. This creates a single point of failure in the system if the data is not replicated and served from another instance.

A better approach is to use Amazon S3 as storage repository for media files. S3 provides high durability of data and the ability to serve content via HTTP out of the box. It also saves on EC2 capacity.

There is no limit to the number of objects that can be stored in a bucket and no variation in performance whether you use many buckets or just a few. You can store all of your objects in a single bucket, or you can organize them across several buckets.

For more information about Amazon S3, visit:
http://docs.amazonwebservices.com/AmazonS3/latest/dev/Introduction.html

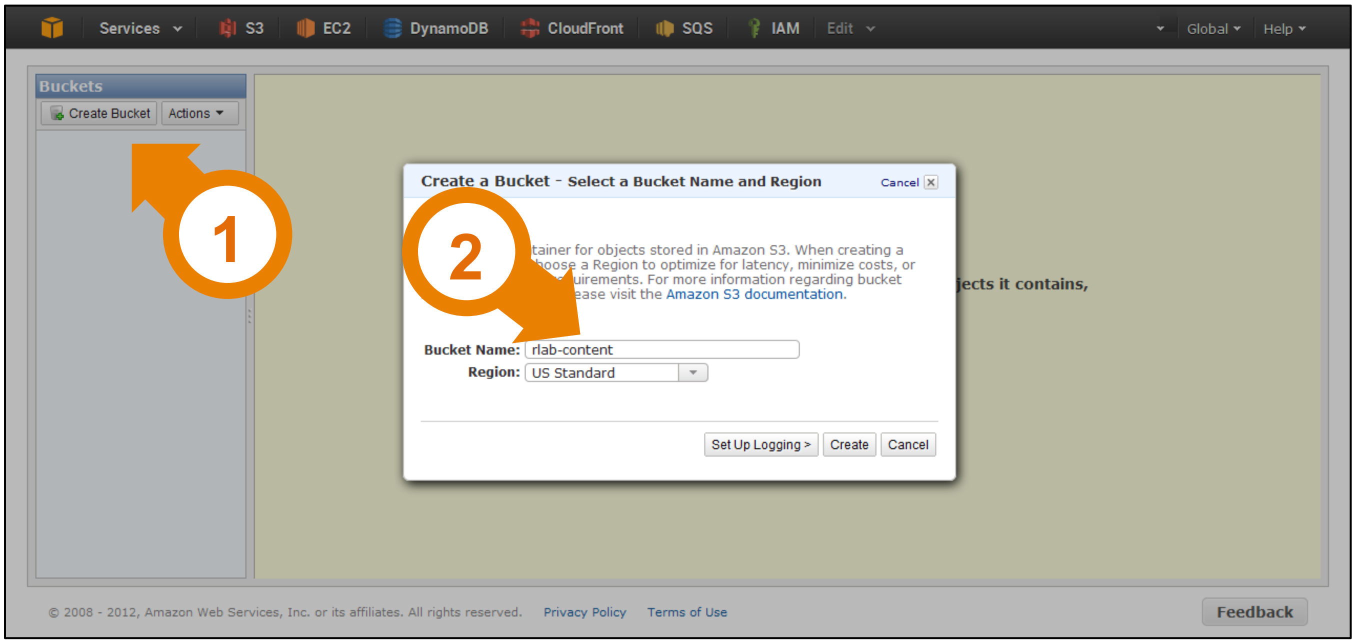
Creating an Amazon S3 Bucket

Open the Amazon S3 (Services / S3) section of the AWS Web Console and click “Create Bucket” (1). Then choose a bucket name (2). Your bucket name should comply with the following rules:

- Bucket names must be **lowercase**, at least 3 and no more than 63 characters long
- Bucket name must be a series of one or more labels separated by a period (.), where each label can contain lowercase letters, numbers and dashes
- Bucket names must not be formatted as an IP address (e.g., 192.168.5.4)

The bucket names on S3 are Global. Within a bucket, you can use any names for your objects, but bucket names must be unique across all of Amazon S3. In case the name you are using is already in use, please provide a different option.

Important: Don’t forget to write down the name of your bucket as it will be required to complete future steps.



tainer for objects stored in Amazon S3. When creating a hoose a Region to optimize for latency, minimize costs, or uirements. For more information regarding bucket ease visit the Amazon S3 documentation.

jects it contains,

Bucket Name: rlab-content
Region: US Standard

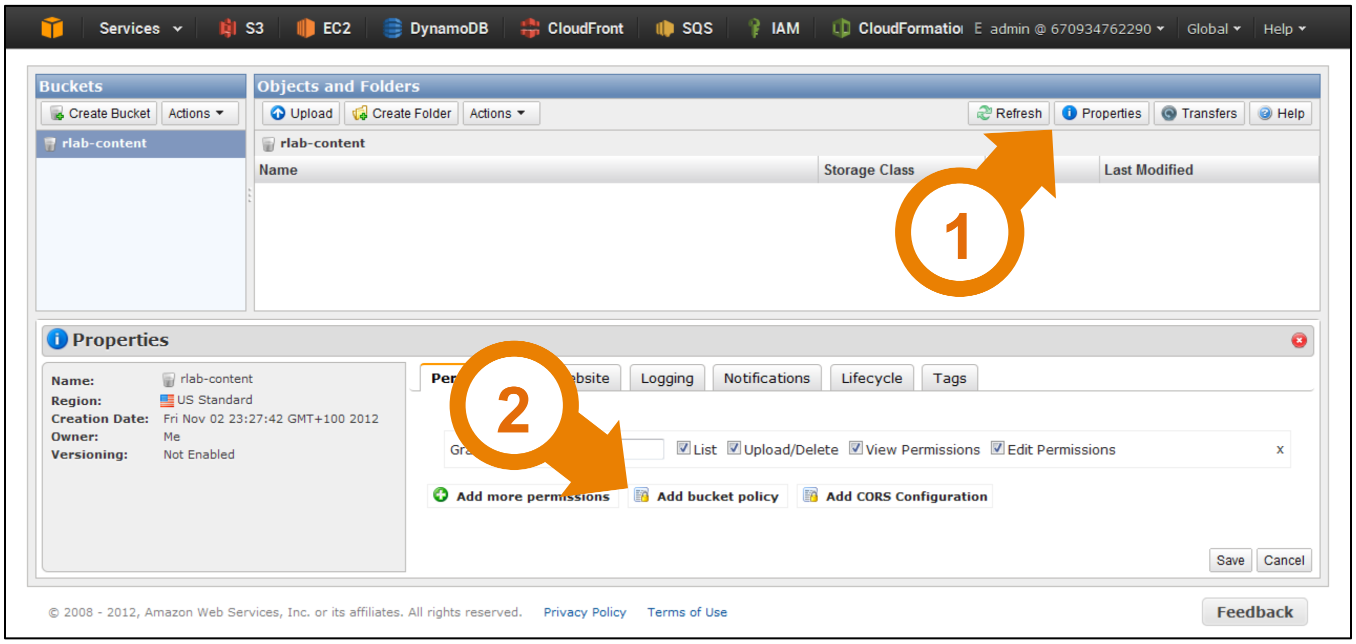
© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Assign a Bucket Policy

By default, content stored in Amazon S3 buckets cannot be accessed publicly. For our use case, we need to allow public access to the bucket, so users can download content. While we want to allow downloads from the bucket, we don’t want to allow users to list the contents of the bucket or delete any objects

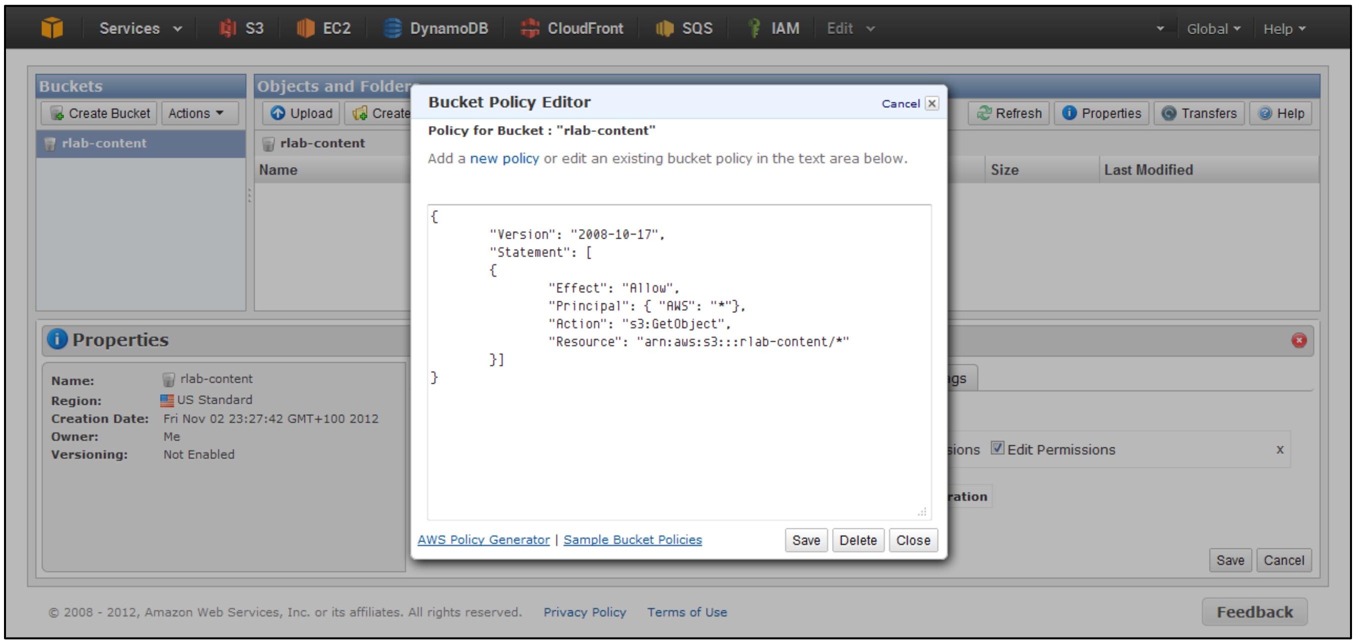
Amazon S3 enables you to manage access to objects and buckets using bucket policies. Bucket policies provide access control management at the bucket level for both a bucket and the objects in it. Bucket policies are a collection of JSON statements written in the access policy language. For more information about bucket policies, visit: http://docs.amazonwebservices.com/AmazonS3/latest/dev/UsingBucketPolicies.html.

To add a policy to your bucket, go to the Amazon S3 section of the AWS web console. Select your newly created bucket, click “**Properties**” (1) and then click the “**Add bucket policy**” (2) button from the permissions tab:



In the "**Bucket Policy Editor**" dialog box, add the following policy to your bucket (the policy file can be downloaded here: http://rlab-content.s3.amazonaws.com/lab11-bucket_policy.json) :

```
{
        "Version": "2008-10-17",
        "Statement": [
                {
                        "Sid": "AddPerm",
                        "Effect": "Allow",
                        "Principal": { "AWS": "*" },
                        "Action": "s3:GetObject",
                        "Resource": "arn:aws:s3:::YOUR_BUCKET_NAME/*"
                }
        ]
}
```



Bucket Policy Editor

Policy for Bucket : "rlab-content"

Add a new policy or edit an existing bucket policy in the text area below.

```
{
        "Version": "2008-10-17",
        "Statement": [
        {
                "Effect": "Allow",
                "Principal": { "AWS": "*"},
                "Action": "s3:GetObject",
                "Resource": "arn:aws:s3:::rlab-content/*"
        }]
}
```

Name: rlab-content
Region: US Standard
Creation Date: Fri Nov 02 23:27:42 GMT+100 2012
Owner: Me
Versioning: Not Enabled

© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Click “**Save**”, to apply the policy to your bucket.

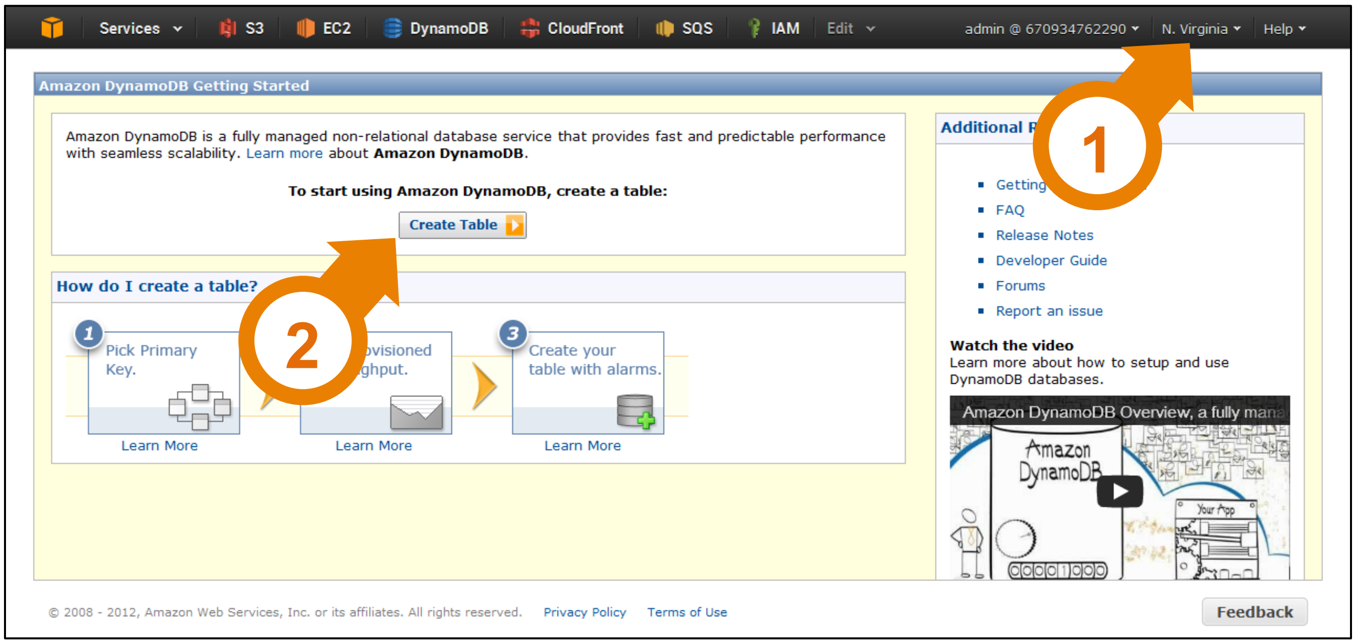
Step 2 – Media Database

For each entry, we need to store metadata information such as the title, comment, publication date, entry type, etc. We need a database to store this information. Since we don’t know how many entries our system will have to contain, and since we want our system to be scalable, we will use Amazon DynamoDB to store metadata information.

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. With a few clicks in the AWS Management Console, you can launch a new Amazon DynamoDB database table, scale up or down their request capacity for the table without downtime or performance degradation, and gain visibility into resource utilization and performance metrics. Amazon DynamoDB enables you to offload the administrative burdens of operating and scaling distributed databases to AWS, so you don’t have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling. For more information about Amazon DynamoDB, visit: http://docs.amazonwebservices.com/amazondynamodb/latest/developerguide/Introduction.html.

Creating an Amazon DynamoDB table

Open the DynamoDB section of the AWS Web Console. Select the **US East (N.Virginia)** region (1) and then click the “**Create Table**” button (2):



Amazon DynamoDB Getting Started

Amazon DynamoDB is a fully managed non-relational database service that provides fast and predictable performance with seamless scalability. Learn more about **Amazon DynamoDB**.

To start using Amazon DynamoDB, create a table:

How do I create a table?

1 Pick Primary Key. Learn More

2 Set Provisioned Throughput. Learn More

3 Create your table with alarms. Learn More

Watch the video
Learn more about how to setup and use DynamoDB databases.

© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Choose a name for the table (1). The table name can include characters a-z, A-Z, 0-9, '_' (underscore), '-' (dash), and '.' (dot). Names can be between 3 and 255 characters long. Then set the Primary Key. Primary Key Type has to be "**Hash / String**" and Hash Attribute Name is "**eib**" (2):

Important: Don't forget to write down the name of your table as it will be required to complete future steps.

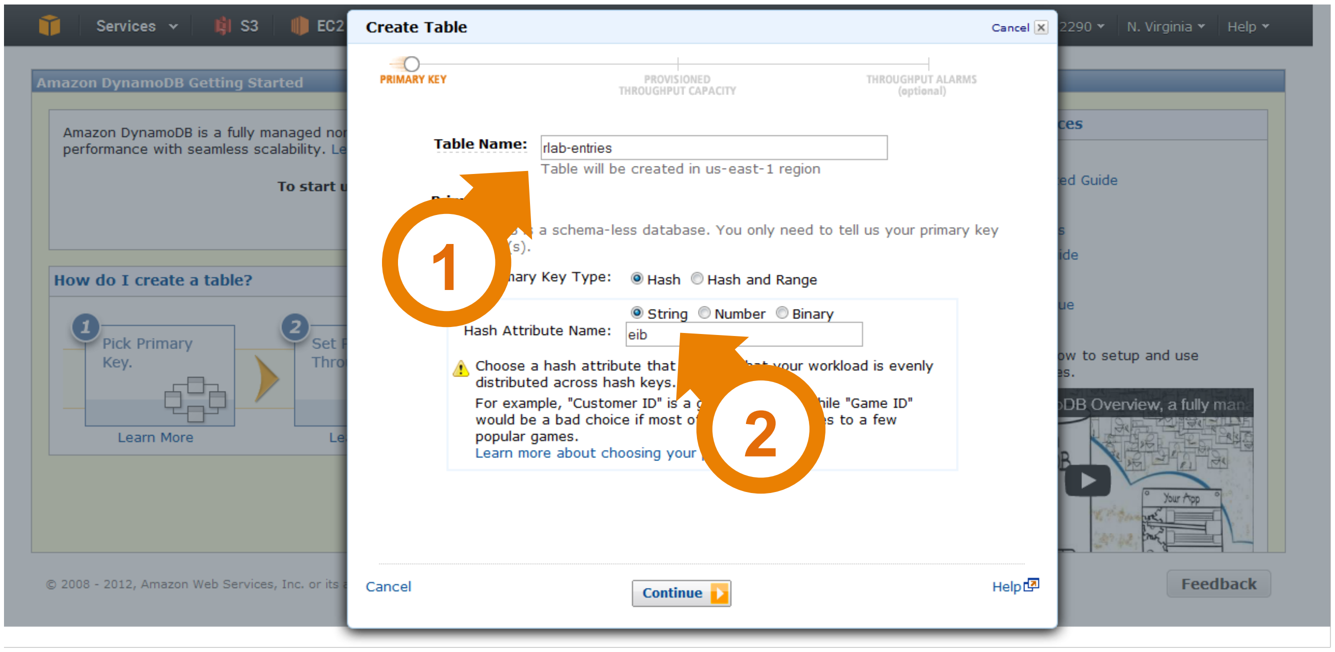


Table Name: rlab-entries
Table will be created in us-east-1 region

a schema-less database. You only need to tell us your primary key (s).

Primary Key Type: [✓ Hash] Hash and Range

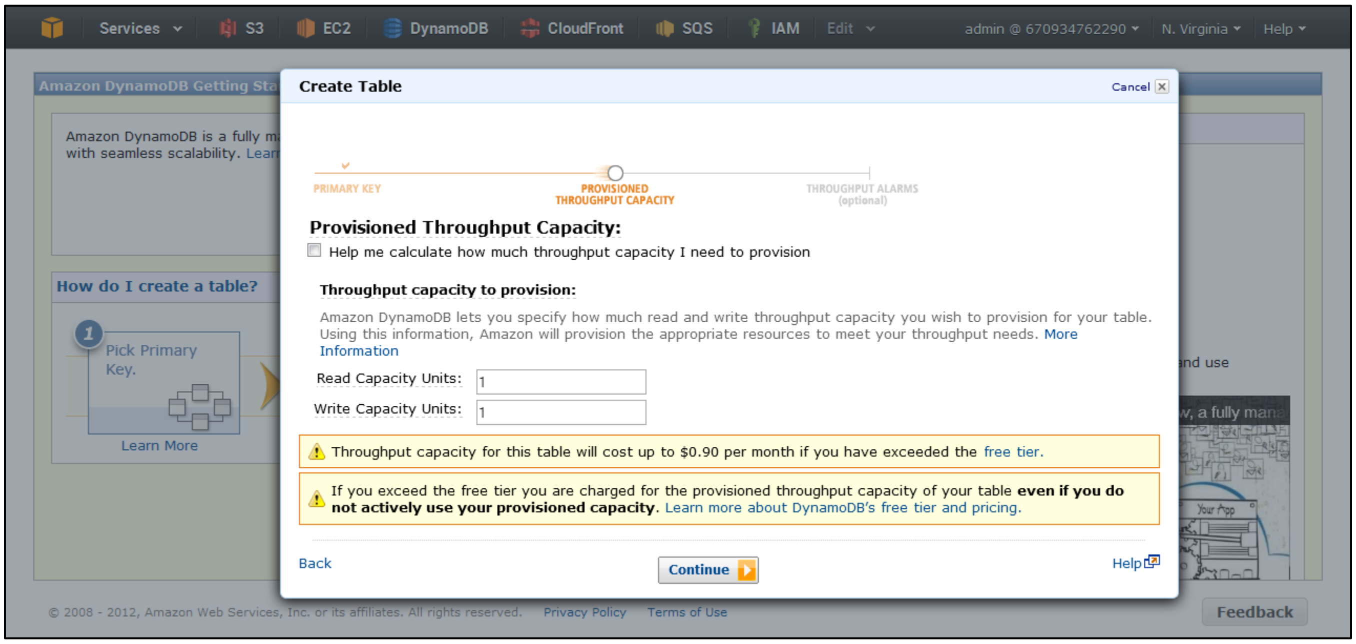
[✓ String] Number Binary

Hash Attribute Name: eib

Choose a hash attribute that your workload is evenly distributed across hash keys.
For example, "Customer ID" is a good hash key, while "Game ID" would be a bad choice if most of your traffic relates to a few popular games.
Learn more about choosing your

For more information about DynamoDB data model, visit:
http://docs.amazonwebservices.com/amazondynamodb/latest/developerguide/DataModel.html.

Next step is tuning table's capacity. When you create or update your Amazon DynamoDB table, you specify how much capacity you wish to reserve for reads and writes. Amazon DynamoDB will reserve the necessary machine resources to meet your throughput needs while ensuring consistent, low-latency performance. For this lab, the default settings will be enough, click "**Continue**":



Provisioned Throughput Capacity:

☐ Help me calculate how much throughput capacity I need to provision

Throughput capacity to provision:

Amazon DynamoDB lets you specify how much read and write throughput capacity you wish to provision for your table. Using this information, Amazon will provision the appropriate resources to meet your throughput needs. More Information

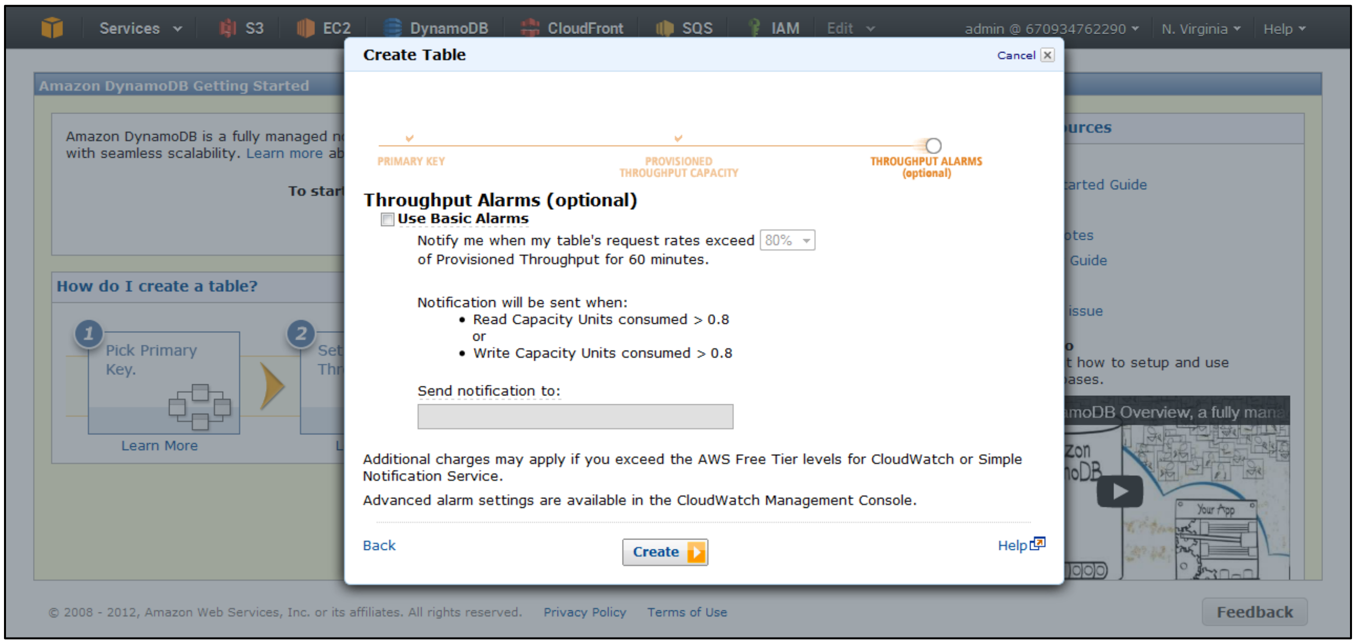
Read Capacity Units:	1
Write Capacity Units:	1

Throughput capacity for this table will cost up to $0.90 per month if you have exceeded the free tier.

If you exceed the free tier you are charged for the provisioned throughput capacity of your table **even if you do not actively use your provisioned capacity**. Learn more about DynamoDB's free tier and pricing.

The last step in the table creation process is setting Throughput Alarms. In production systems, such alarms will allow you to understand when to adjust your table's throughput capacity. For more information about DynamoDB provisioned throughput model, visit:
http://docs.amazonwebservices.com/amazondynamodb/latest/developerguide/ProvisionedThroughputIntro.html.

For this lab, you can disable basic alarms, and click "**Create**":



Throughput Alarms (optional)

☐ Use Basic Alarms

Notify me when my table's request rates exceed 80% of Provisioned Throughput for 60 minutes.

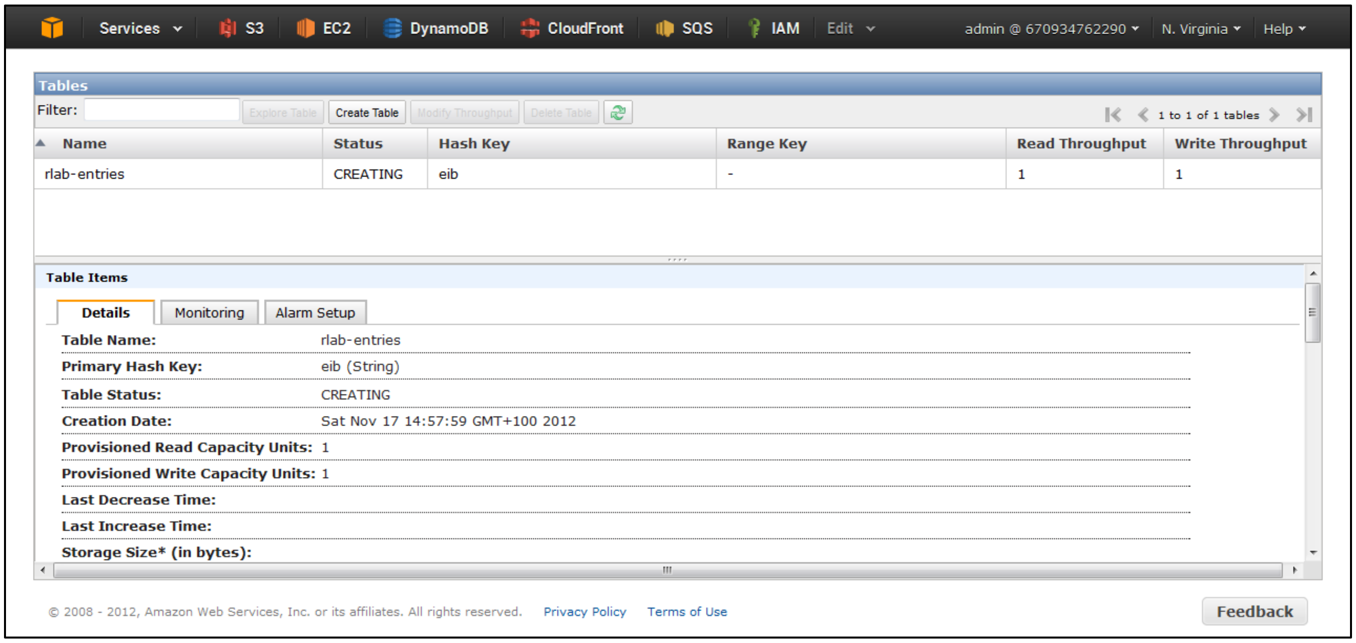
Notification will be sent when:
- Read Capacity Units consumed > 0.8
  or
- Write Capacity Units consumed > 0.8

Send notification to:

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service.

Advanced alarm settings are available in the CloudWatch Management Console.

The table's status will be noted as "**CREATING**". The table creation process might take a few minutes:



Name	Status	Hash Key	Range Key	Read Throughput	Write Throughput
rlab-entries	CREATING	eib	-	1	1

Table Name:	rlab-entries
Primary Hash Key:	eib (String)
Table Status:	CREATING
Creation Date:	Sat Nov 17 14:57:59 GMT+100 2012
Provisioned Read Capacity Units:	1
Provisioned Write Capacity Units:	1
Last Decrease Time:	
Last Increase Time:	
Storage Size* (in bytes):	

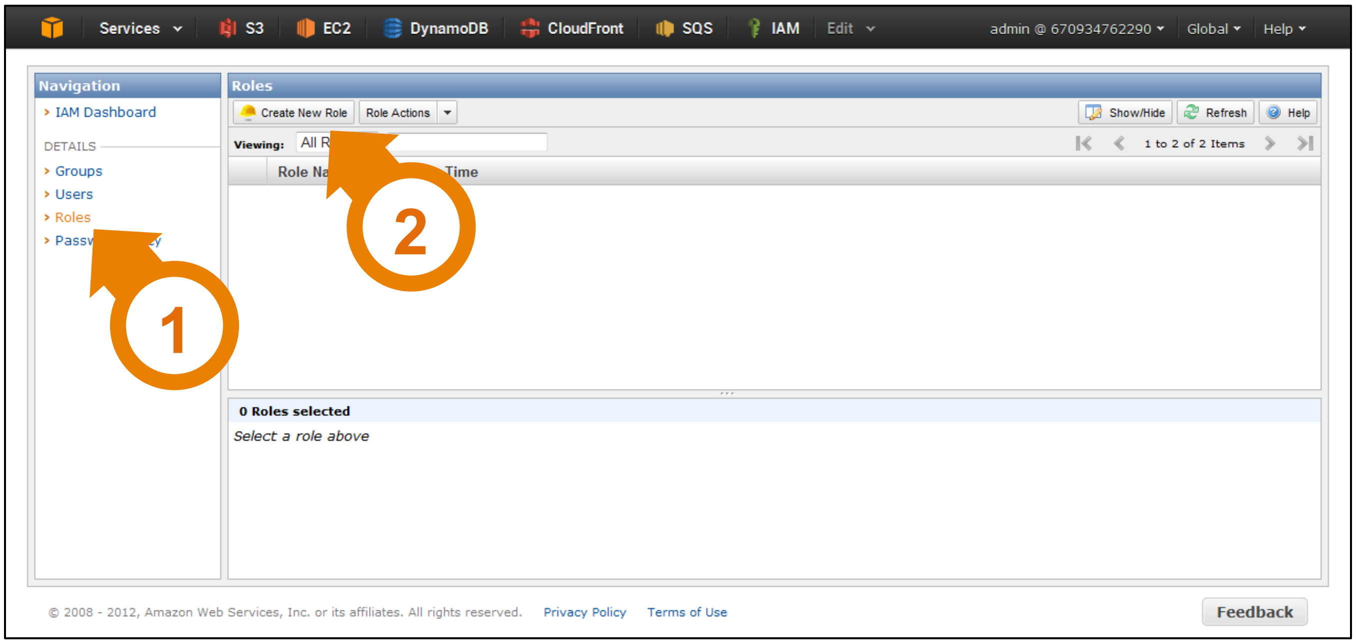
Step 3 – Access Control

So far, we created two resources on AWS: an Amazon S3 Bucket and an Amazon DynamoDB table. To access both of them, we need specific security credentials. While we could use the AWS Account Credentials to access those resources, such practice is strongly discouraged in production systems since AWS Account Credentials have full access to all AWS resources. AWS Identity and Access Management (IAM) helps you securely control access to Amazon Web Services and your account resources. AWS IAM provides roles for EC2 instances, a feature that makes it easy to securely access AWS service APIs from EC2 instances. You can create an IAM role, assign it a set of permissions, launch EC2 instances with the IAM role, and then AWS access keys with the specified permissions are automatically made available on those EC2 instances.

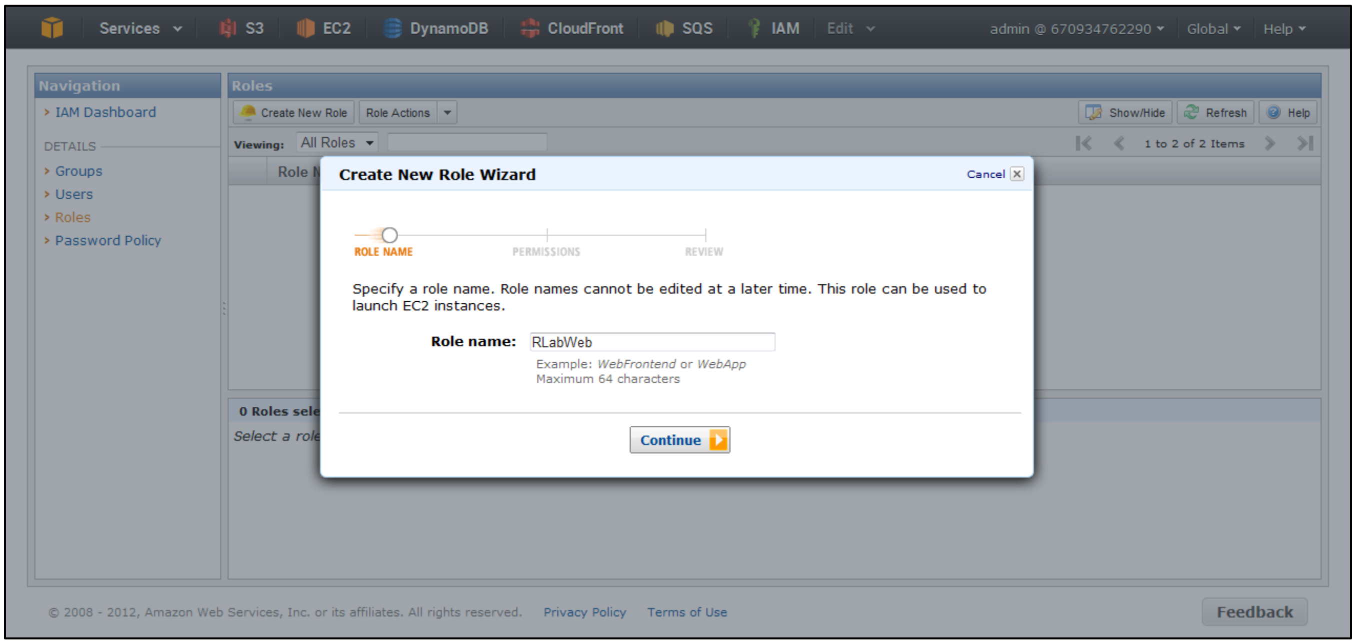
For more information about AWS IAM, visit: http://docs.amazonwebservices.com/IAM/latest/UserGuide/Welcome.html. A short video explaining IAM roles for EC2 instances is available here: http://www.youtube.com/watch?v=XuRM4Id6uDY.

Creating an IAM role

Open the AWS IAM section of the AWS Web Console, select the "**Roles**" section (1) then click the "**Create New Role**" button (2):



Choose a name for the IAM role:



Create New Role Wizard

Specify a role name. Role names cannot be edited at a later time. This role can be used to launch EC2 instances.

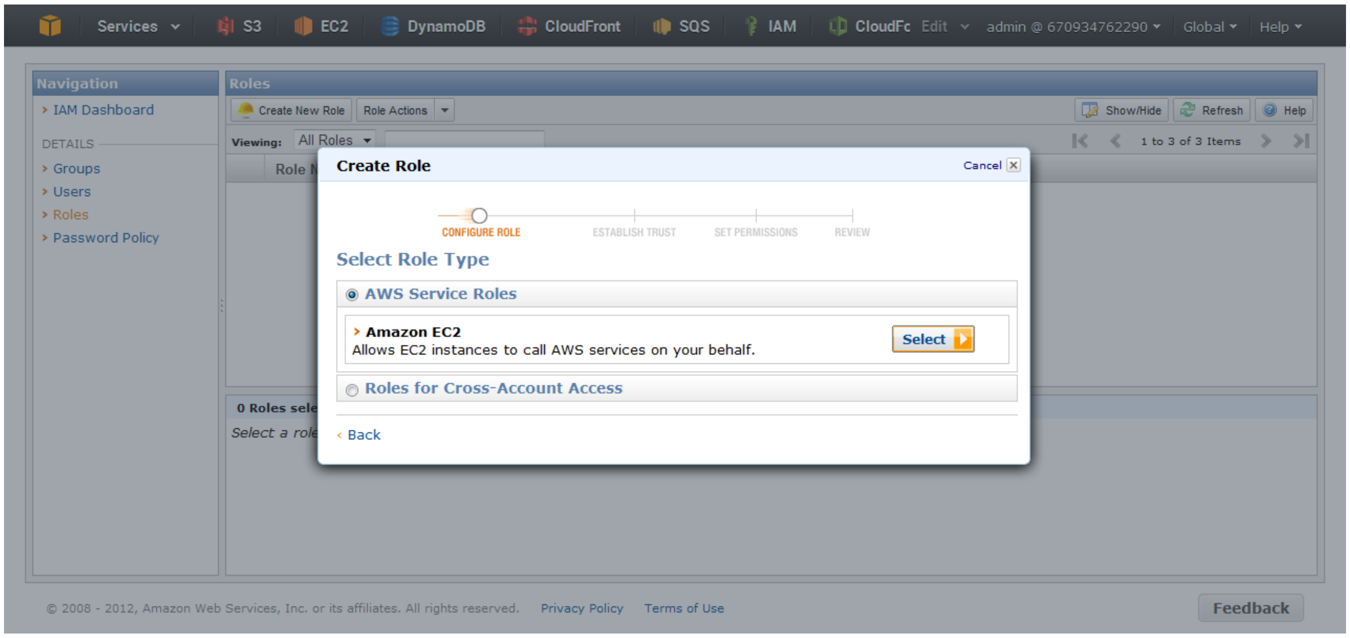
Role name: RLabWeb

Example: *WebFrontend* or *WebApp*
Maximum 64 characters

© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Now you need to specify a policy describing what actions are allowed for this role. A policy is a document that formally states one or more permissions. The distinction between a *permission* and a *policy* is important. To give a particular IAM role a permission, you simply write a policy according to the access policy language IAM uses, then attach the policy to the role you want it to apply to (a particular user or group in your AWS account). You don't actually specify the entity in the policy itself; the act of attaching the policy to the role grants it the permission stated in the policy. Policies are described in JSON format. For information about the access policy language, see http://docs.amazonwebservices.com/IAM/latest/UserGuide/AccessPolicyLanguage.html.

On the option "**Select Role Type**" and choose "**AWS Service Roles**":



Create Role

Select Role Type

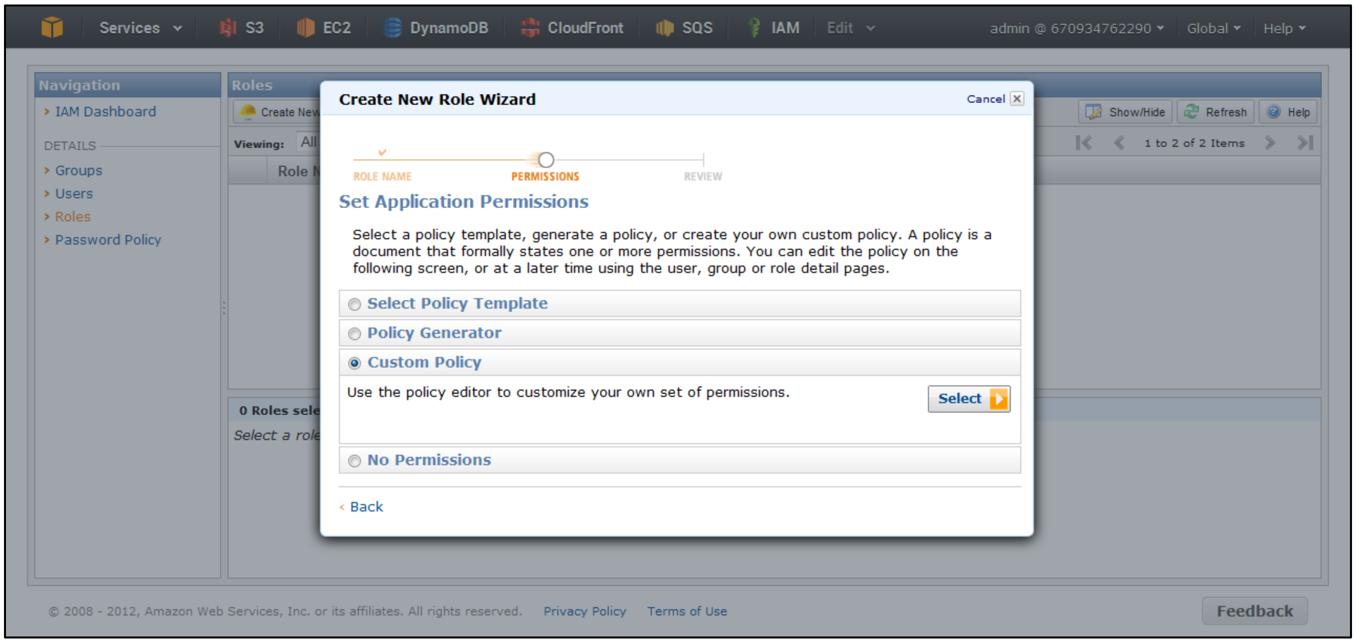
(●) AWS Service Roles

Amazon EC2
Allows EC2 instances to call AWS services on your behalf.

( ) Roles for Cross-Account Access

© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

There are various ways to provide a policy, like selecting from one of the available templates, using the policy generator, or typing your own policy. For this lab, we provide the policy below, so chose “**Custom Policy**”:



Create New Role Wizard

Set Application Permissions

Select a policy template, generate a policy, or create your own custom policy. A policy is a document that formally states one or more permissions. You can edit the policy on the following screen, or at a later time using the user, group or role detail pages.

- Select Policy Template
- Policy Generator
- [✓ Custom Policy]

Use the policy editor to customize your own set of permissions.

- No Permissions

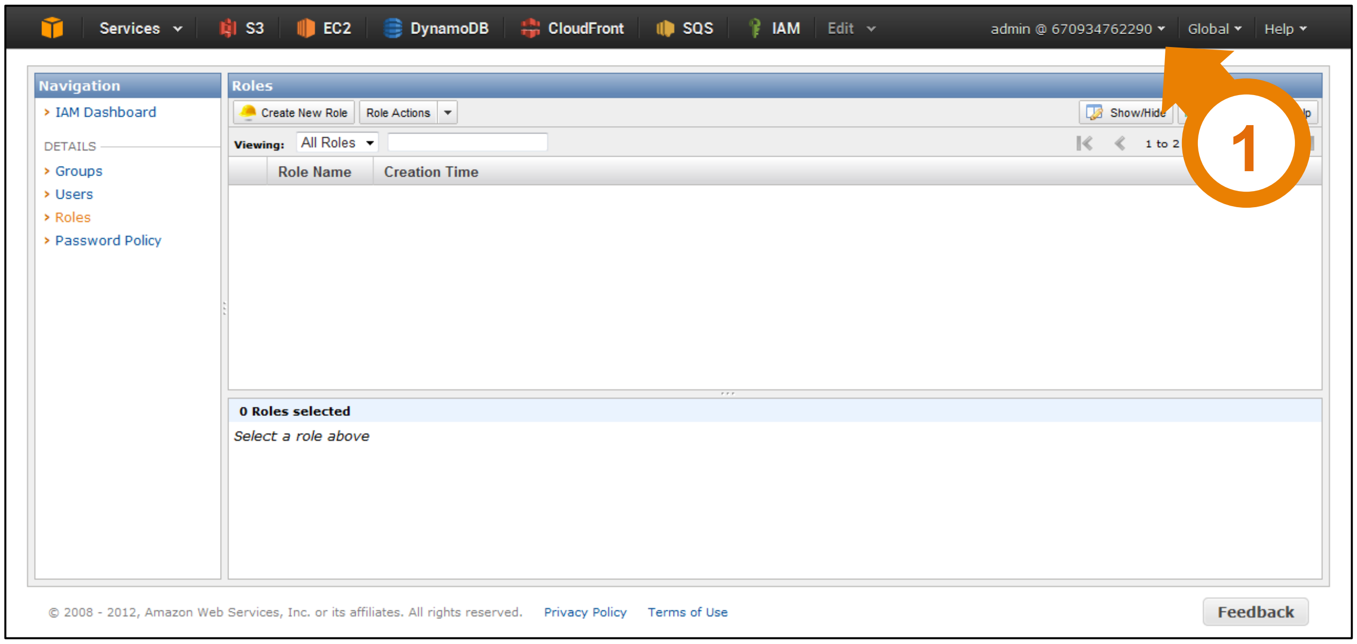
The web server needs to be able read, write and delete objects from the S3 bucket as well as listing the contents of the bucket. The web server also needs to query, scan, read, write and delete items from the DynamoDB table.

Use the following policy to set those permissions, and replace the strings highlighted by your own values (the policy file can be downloaded here: http://rlab-content.s3.amazonaws.com/lab11-role_policy.json) :

```
{
  "Statement": [
    {
      "Sid": "Stmt1351896212463",
      "Action":
["dynamodb:BatchGetItem","dynamodb:DeleteItem","dynamodb:DescribeTable",
        "dynamodb:GetItem","dynamodb:PutItem","dynamodb:Query","dynamodb:Scan",
        "dynamodb:UpdateItem"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:dynamodb:us-east-1:YOUR_ACCOUNT_ID:table/YOUR_TABLE_NAME"
    },
    {
      "Sid": "Stmt1351896363046",
      "Action": ["s3:DeleteObject","s3:GetObject","s3:PutObject","s3:ListBucket"],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::YOUR_BUCKET_NAME/*", "arn:aws:s3:::YOUR_BUCKET_NAME"
      ]
    }
```

```
        ]
    }
```

Your account ID can be found on the top-right of the AWS Web Console (1):



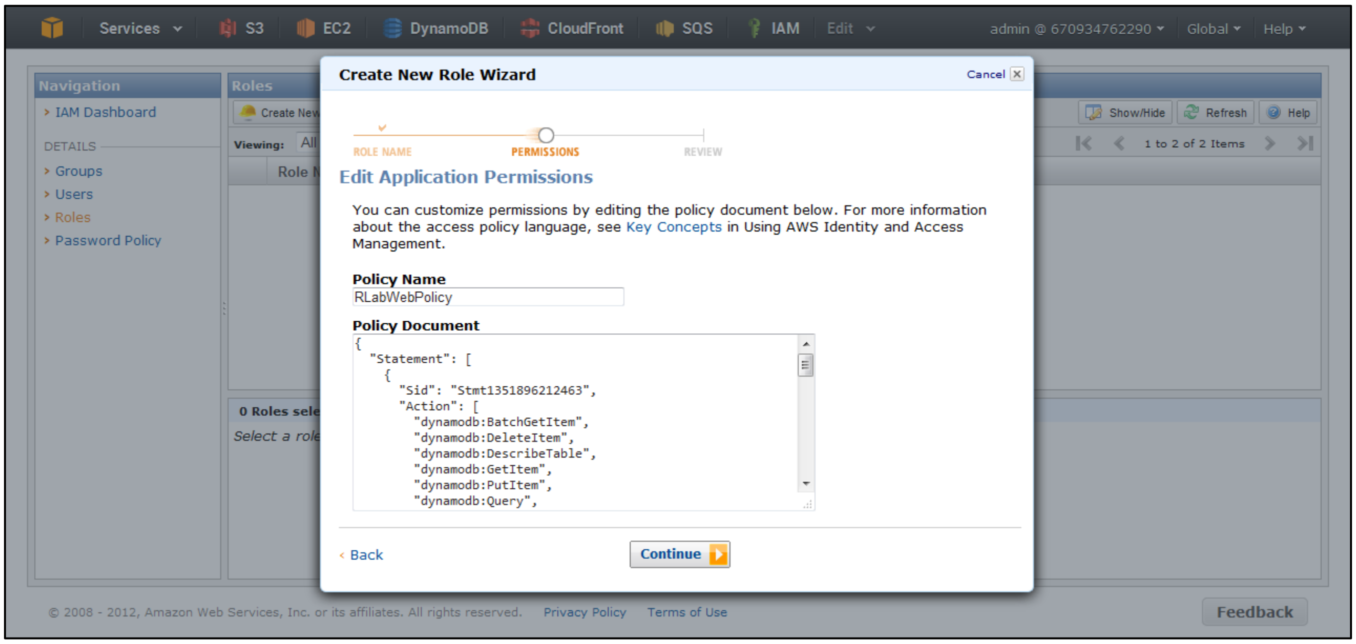
The values for your S3 Bucket Name and Dynamo DB Table Name are the ones that you specified earlier in this lab.

Once you have your own value regarding YOUR_ACCOUNT_ID, YOUR_TABLE_NAME, YOUR_BUCKET_NAME you can update them on the previously provided JSON policy.

Copy and paste this policy in the "**Policy Document**" area.

Select a name for role policy and click continue:

Important: Don't forget to write down the name of your IAM role as it will be required to complete future steps.



Create New Role Wizard

Edit Application Permissions

You can customize permissions by editing the policy document below. For more information about the access policy language, see Key Concepts in Using AWS Identity and Access Management.

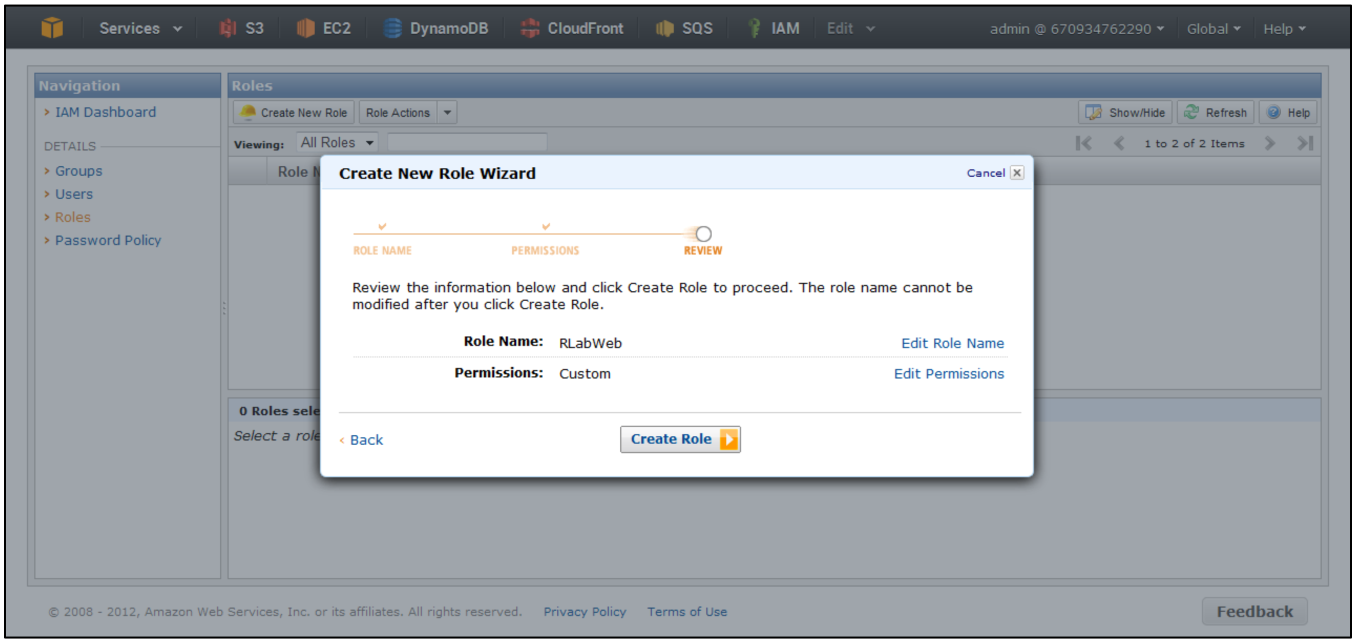
Policy Name
RLabWebPolicy

Policy Document

```
{
  "Statement": [
    {
      "Sid": "Stmt1351896212463",
      "Action": [
        "dynamodb:BatchGetItem",
        "dynamodb:DeleteItem",
        "dynamodb:DescribeTable",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:Query",
```

© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Click “**Create Role**” to complete the role creation process:



Create New Role Wizard

Review the information below and click Create Role to proceed. The role name cannot be modified after you click Create Role.

Role Name:	RLabWeb
Permissions:	Custom

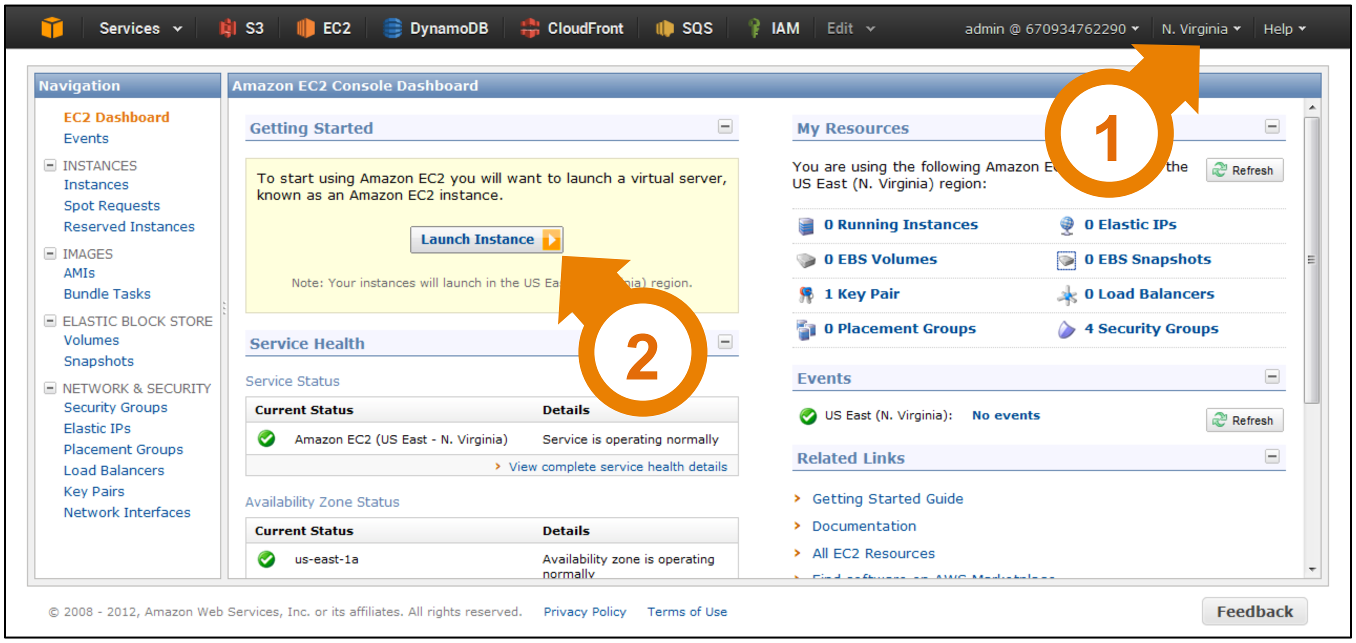
© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Step 4 – Web Front-End

You have now everything ready to create the web server. You will use an Amazon EC2 instance to host the web server. The web application used for this lab is already packaged in an Amazon Machine Image (AMI).

Deploying the web server

Open the Amazon EC2 section of the AWS Web Console. Select the **US East (N.Virginia)** region (1), and then click the "**Launch** Instance" button (2):



To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US Ea... ...ia) region.

My Resources

You are using the following Amazon E... ...the US East (N. Virginia) region:

0 Running Instances	0 Elastic IPs
0 EBS Volumes	0 EBS Snapshots
1 Key Pair	0 Load Balancers
0 Placement Groups	4 Security Groups

Events

US East (N. Virginia): No events

Service Status

Current Status	Details
✓ Amazon EC2 (US East - N. Virginia)	Service is operating normally

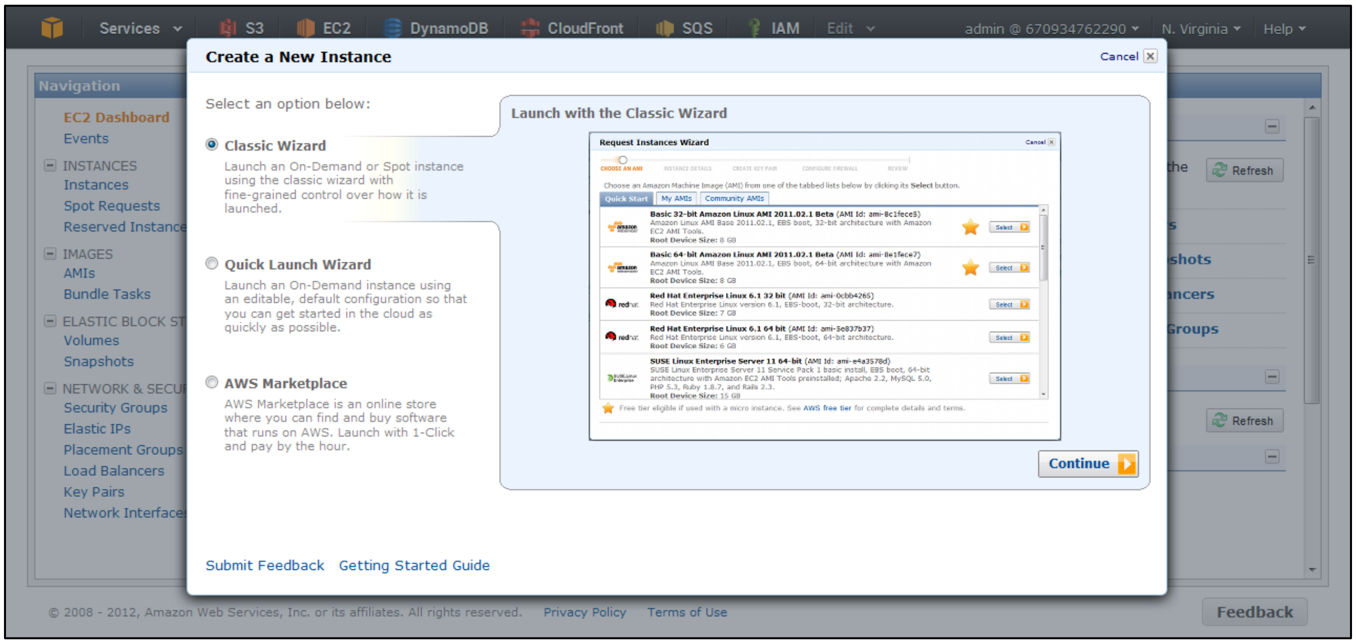
› View complete service health details

Availability Zone Status

Current Status	Details
✓ us-east-1a	Availability zone is operating normally

© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Choose the "**Classic Wizard**" option and click continue:



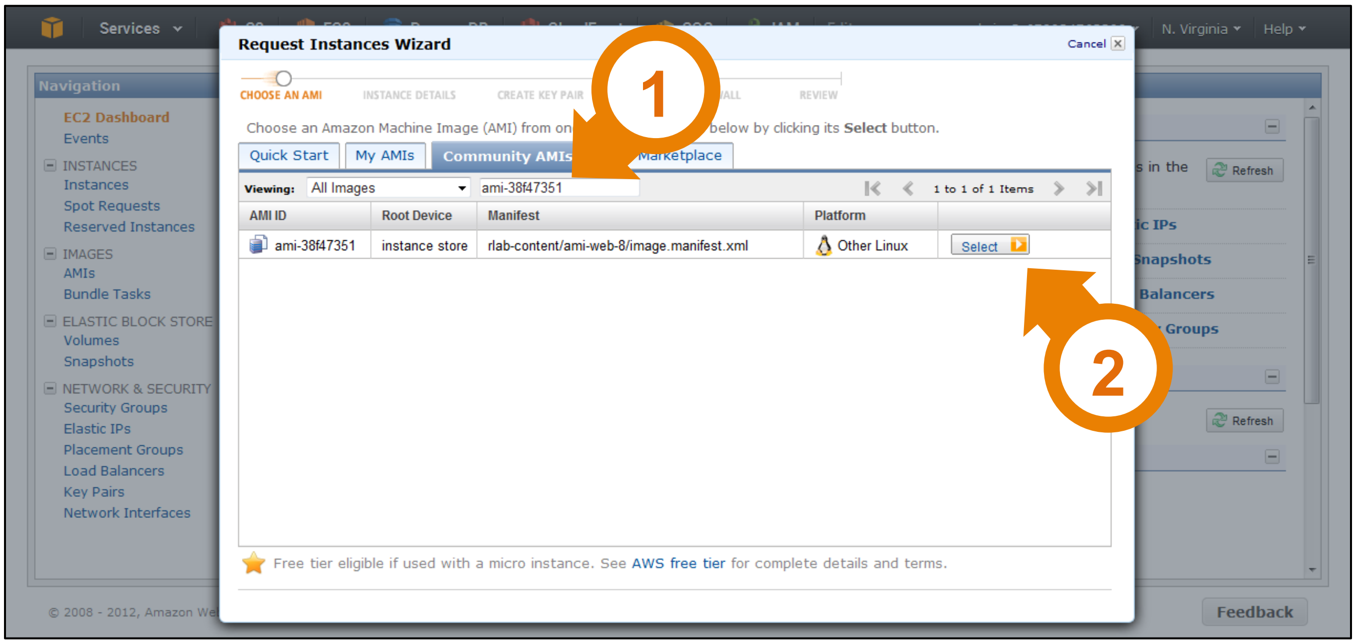
Select an option below:

[✓ Classic Wizard]
Launch an On-Demand or Spot instance using the classic wizard with fine-grained control over how it is launched.

Quick Launch Wizard
Launch an On-Demand instance using an editable, default configuration so that you can get started in the cloud as quickly as possible.

AWS Marketplace
AWS Marketplace is an online store where you can find and buy software that runs on AWS. Launch with 1-Click and pay by the hour.

From the "**Community AMIs**" tab, search for the AMI "**ami-38f47351**" (1) and click "**Select**" (2):



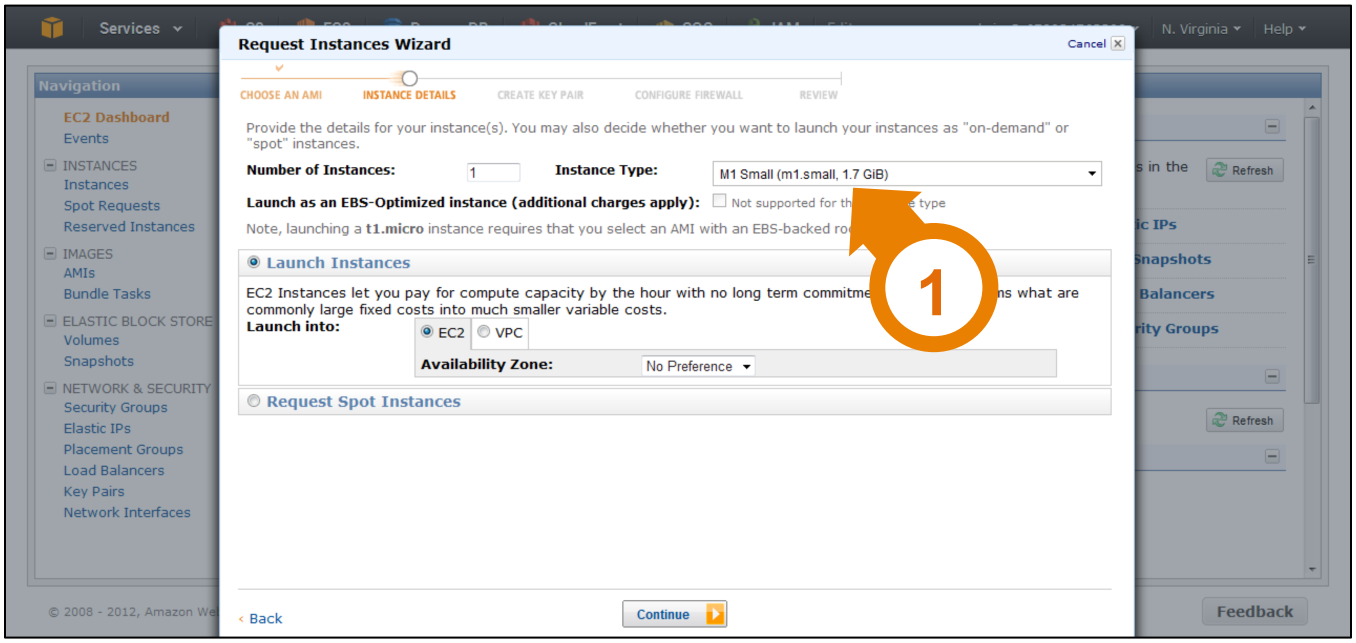
Choose an Amazon Machine Image (AMI) from on... below by clicking its **Select** button.

Viewing: All Images | ami-38f47351 | 1 to 1 of 1 Items

AMI ID	Root Device	Manifest	Platform	
ami-38f47351	instance store	rlab-content/ami-web-8/image.manifest.xml	Other Linux	Select

Free tier eligible if used with a micro instance. See AWS free tier for complete details and terms.

You can now choose the instance type for the web server. For this lab, "**M1 Small**" will be enough (1). As the services used by the instance (Amazon DynamoDB and Amazon S3) are spread across multiple availability zones within the same region, we don't need to choose a specific availability zone for the web server. Click "**Continue**".



Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: 1 **Instance Type:** M1 Small (m1.small, 1.7 GiB)

Launch as an EBS-Optimized instance (additional charges apply): Not supported for th... type

Note, launching a **t1.micro** instance requires that you select an AMI with an EBS-backed ro...

[✓ Launch Instances]

EC2 Instances let you pay for compute capacity by the hour with no long term commitme... ns what are commonly large fixed costs into much smaller variable costs.

Launch into: [✓ EC2] VPC

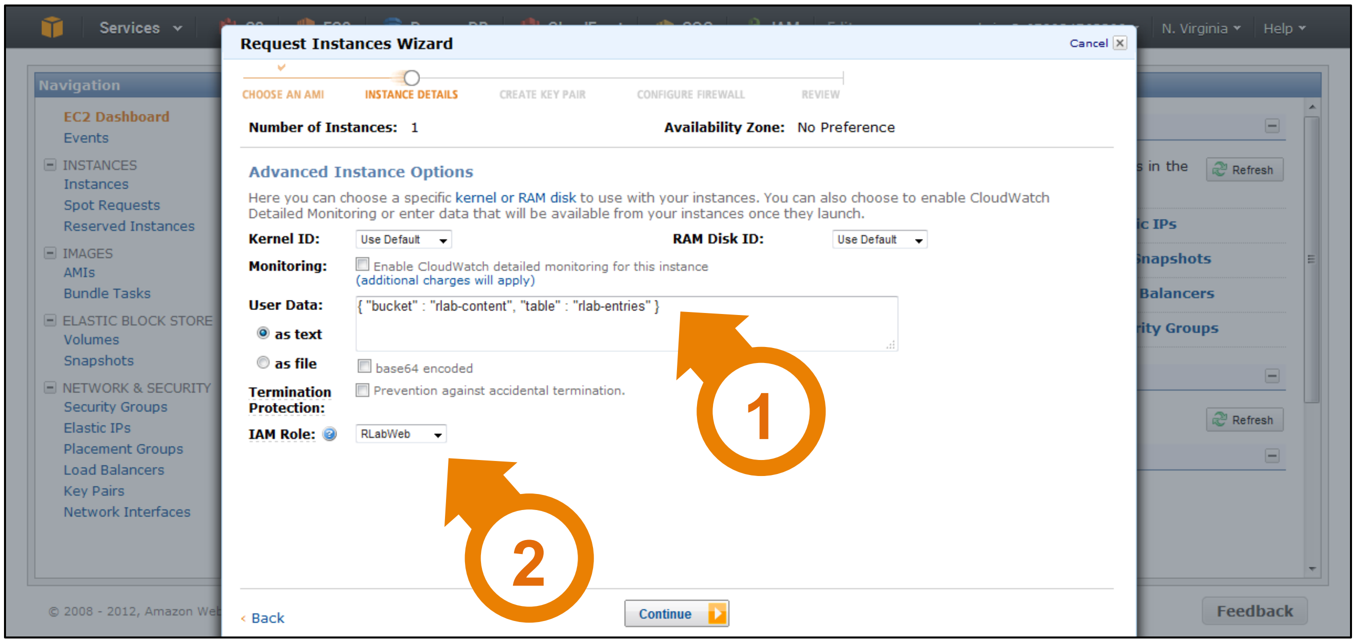
Availability Zone: No Preference

Request Spot Instances

Next, we need to specify the web server which DynamoDB table and which S3 bucket to use. Amazon EC2 instances can access instance-specific metadata, as well as data supplied when launching the instances. We can pass information to the instance using the "**User Data**" field (1). Paste the following JSON document and replace the highlighted strings with your own values:

```
{ "bucket" : "YOUR_BUCKET_NAME", "table" : "YOUR_TABLE_NAME" }
```

Apply the **IAM role** (2) created previously to the instance to ensure access to those resources, and click "**Continue**"


Request Instances Wizard

Number of Instances: 1 Availability Zone: No Preference

Advanced Instance Options

Here you can choose a specific kernel or RAM disk to use with your instances. You can also choose to enable CloudWatch Detailed Monitoring or enter data that will be available from your instances once they launch.

Kernel ID: Use Default RAM Disk ID: Use Default

Monitoring: Enable CloudWatch detailed monitoring for this instance (additional charges will apply)

User Data: { "bucket" : "rlab-content", "table" : "rlab-entries" }

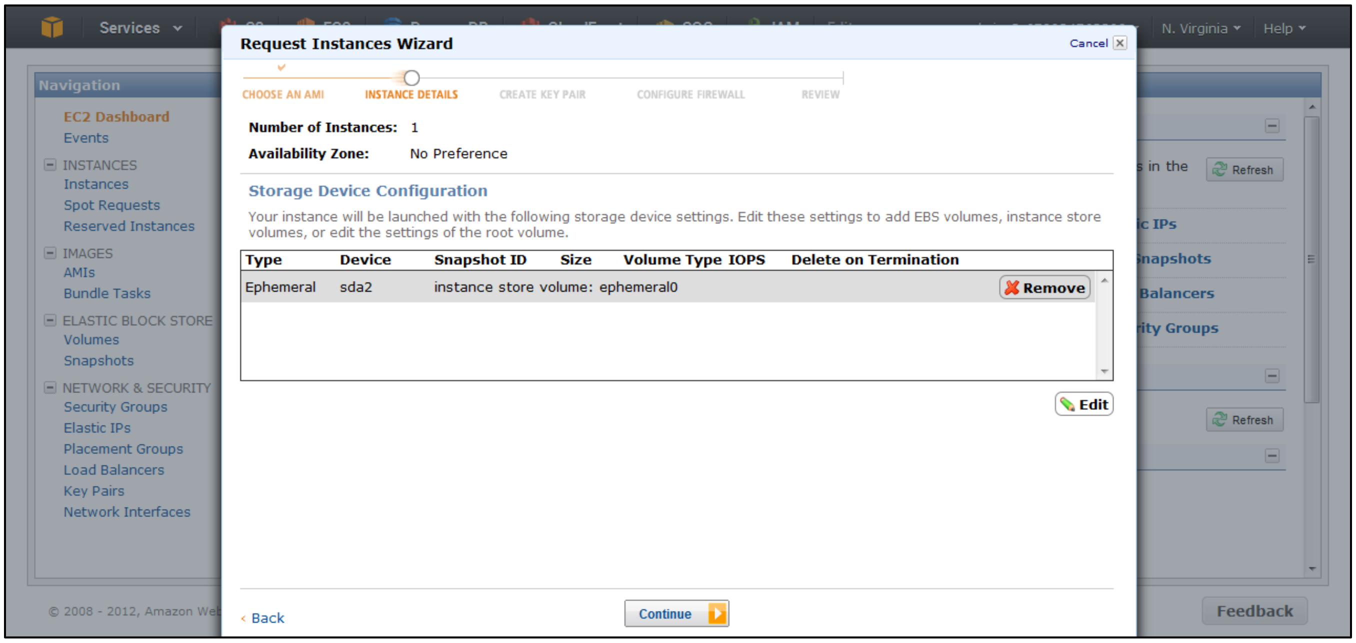
[✓ as text] as file

base64 encoded

Termination Protection: Prevention against accidental termination.

IAM Role: RLabWeb

As we don't need specific storage for the web server, you can ignore the Storage Device Configuration section and click "**Continue**":


Request Instances Wizard

Number of Instances: 1

Availability Zone: No Preference

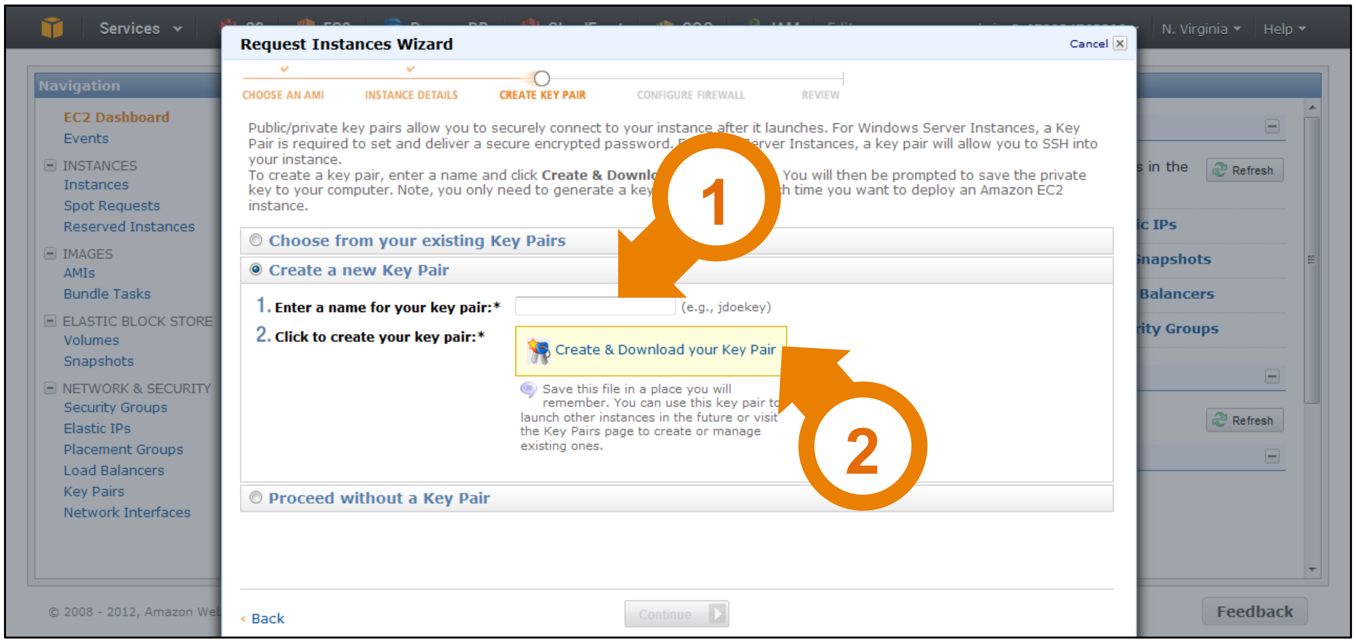
Storage Device Configuration

Your instance will be launched with the following storage device settings. Edit these settings to add EBS volumes, instance store volumes, or edit the settings of the root volume.

Type	Device	Snapshot ID	Size	Volume Type	IOPS	Delete on Termination
Ephemeral	sda2	instance store volume: ephemeral0				

The web server instance you're creating is based on Linux operating system. To enable remote login via SSH, you need to provide a key pair.

To create a Key Pair and remotely connect to the web server instance, enter a **Key Pair name** (1) and click the "**Download your Key Pair**" (2). Otherwise, you can select "**Proceed without a Key Pair**" and click "**Continue**":



Public/private key pairs allow you to securely connect to your instance after it launches. For Windows Server Instances, a Key Pair is required to set and deliver a secure encrypted password. F ... erver Instances, a key pair will allow you to SSH into your instance.
To create a key pair, enter a name and click **Create & Downlo** ... You will then be prompted to save the private key to your computer. Note, you only need to generate a key ... h time you want to deploy an Amazon EC2 instance.

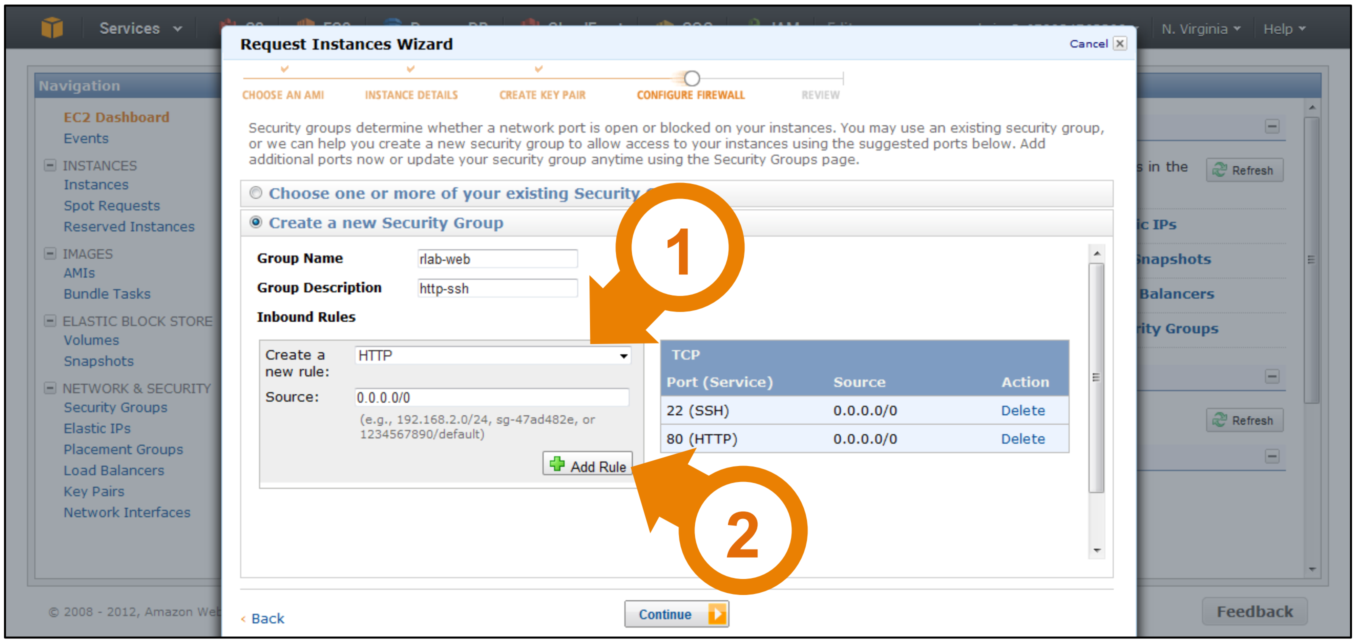
1. **Enter a name for your key pair:*** (e.g., jdoekey)
2. **Click to create your key pair:***

Save this file in a place you will remember. You can use this key pair to launch other instances in the future or visit the Key Pairs page to create or manage existing ones.

© 2008 - 2012, Amazon Web

Next step is to configure firewall settings for this instance. Amazon EC2 provides **security groups** acting as a firewall that controls the traffic allowed to reach one or more instances. You can add rules to each security group that control the inbound traffic allowed to reach the instances associated with the security group. All other inbound traffic is discarded.

The instance we're creating is a web server so we need to assign a rule for inbound HTTP traffic. Create a new rule with "**HTTP**" (1), leave the 0.0.0.0/0 IP address range (any IP address allowed) and click "**Add Rule**" (2). If you created a Key Pair in the previous step, you also need to create a rule to allow inbound SSH traffic. Click "**Continue**".



Security groups determine whether a network port is open or blocked on your instances. You may use an existing security group, or we can help you create a new security group to allow access to your instances using the suggested ports below. Add additional ports now or update your security group anytime using the Security Groups page.

Choose one or more of your existing Security

Create a new Security Group

Group Name: rlab-web

Group Description: http-ssh

Inbound Rules

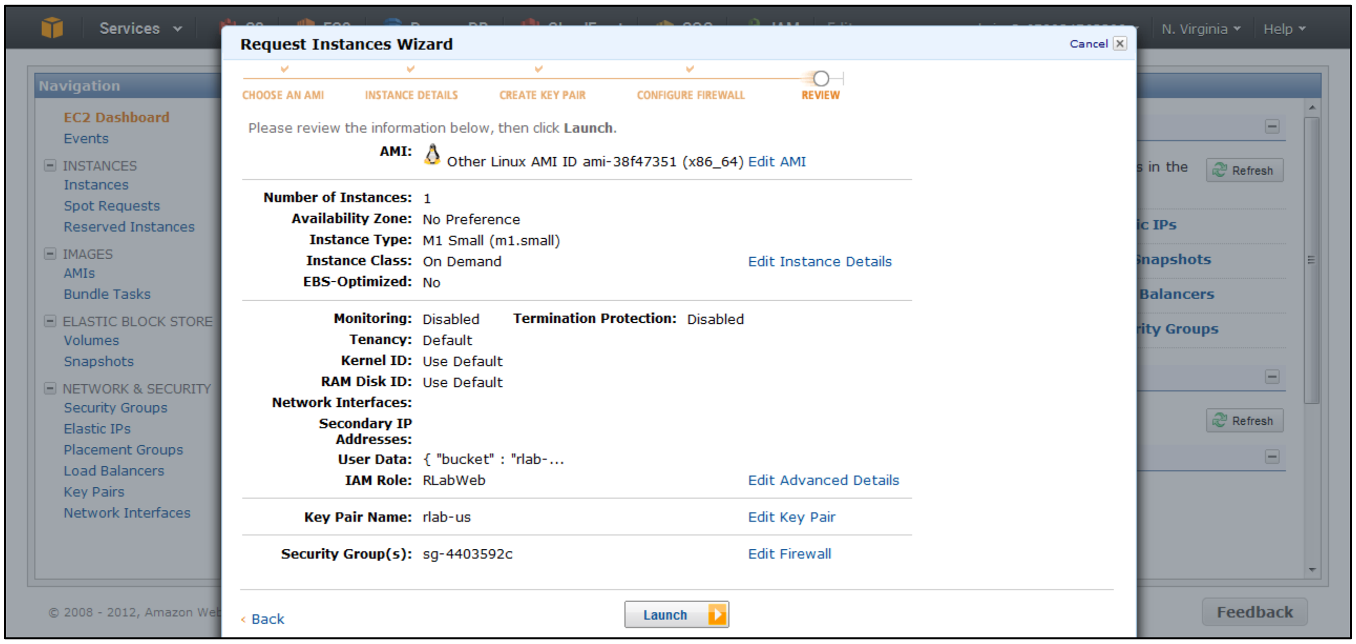
Create a new rule: HTTP

Source: 0.0.0.0/0

(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

TCP		
Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete
80 (HTTP)	0.0.0.0/0	Delete

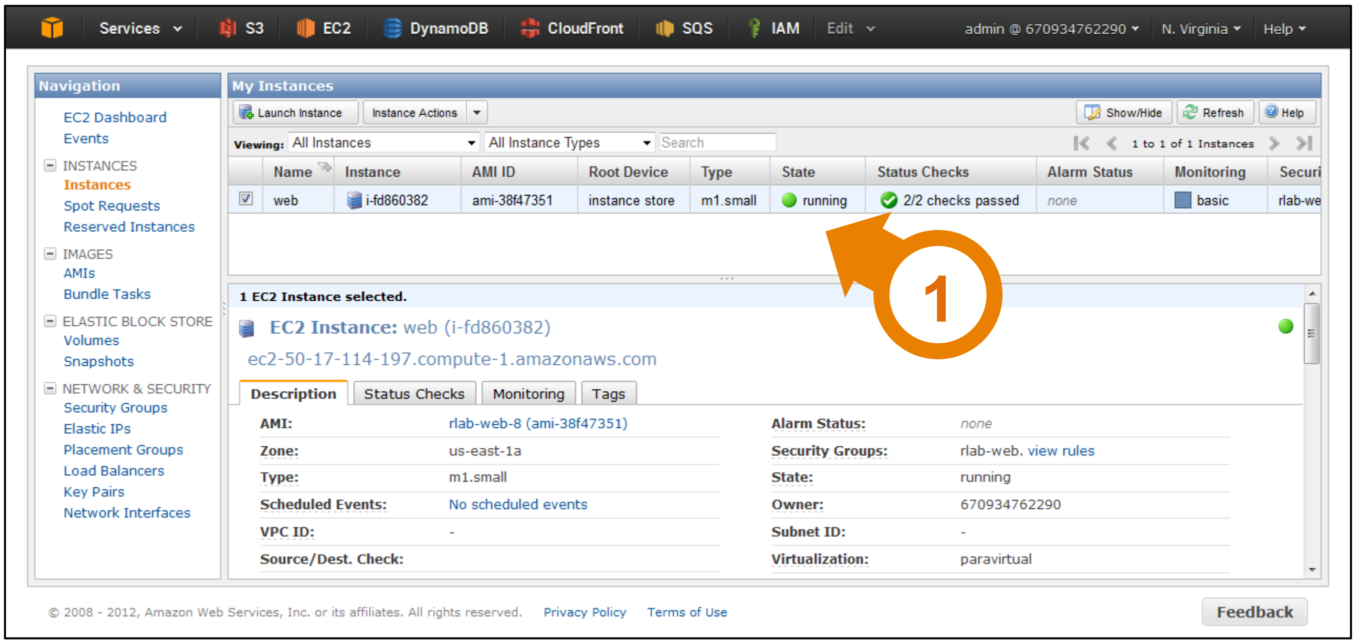
Review your instance settings and click "**Launch**":



Please review the information below, then click **Launch**.

AMI:	Other Linux AMI ID ami-38f47351 (x86_64)	Edit AMI
Number of Instances:	1	
Availability Zone:	No Preference	
Instance Type:	M1 Small (m1.small)	
Instance Class:	On Demand	Edit Instance Details
EBS-Optimized:	No	
Monitoring:	Disabled	**Termination Protection:** Disabled
Tenancy:	Default	
Kernel ID:	Use Default	
RAM Disk ID:	Use Default	
Network Interfaces:		
Secondary IP Addresses:		
User Data:	{ "bucket" : "rlab-...	
IAM Role:	RLabWeb	Edit Advanced Details
Key Pair Name:	rlab-us	Edit Key Pair
Security Group(s):	sg-4403592c	Edit Firewall

The console will show the instance as "**pending**". Once ready, the instance's status will switch to "**running**" as shown below:



Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Alarm Status	Monitoring	Securi
web	i-fd860382	ami-38f47351	instance store	m1.small	running	2/2 checks passed	none	basic	rlab-we

1 EC2 Instance selected.

EC2 Instance: web (i-fd860382)

ec2-50-17-114-197.compute-1.amazonaws.com

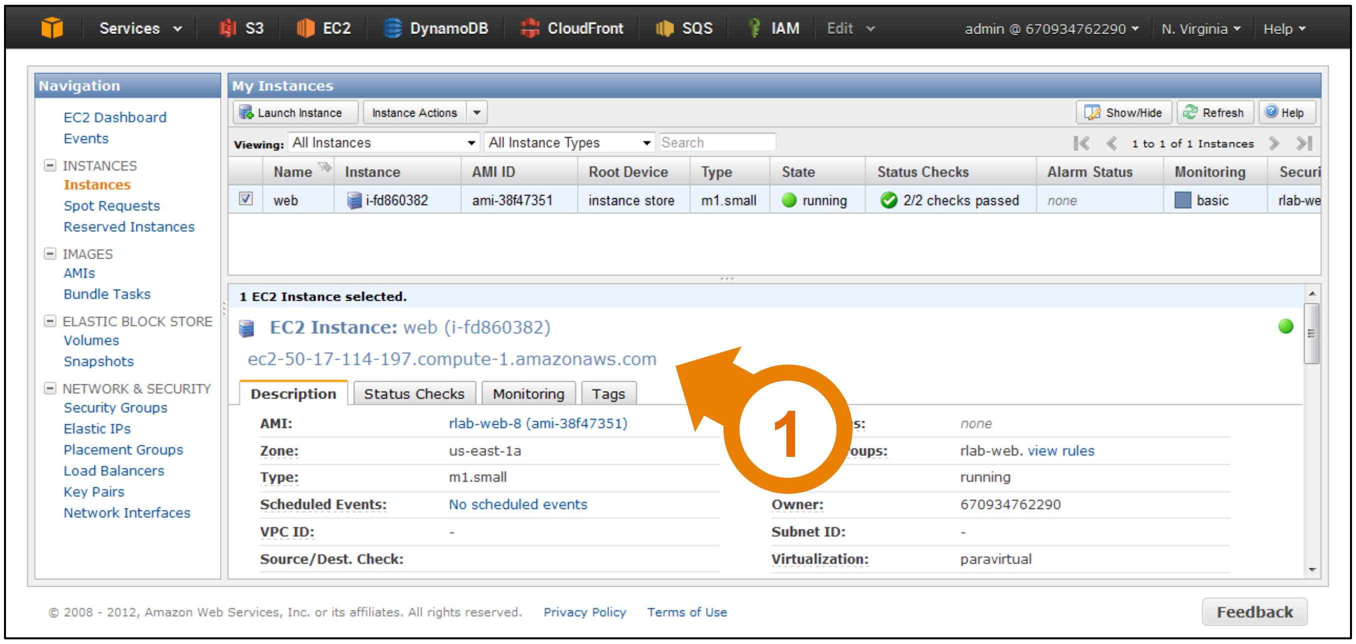
Description | Status Checks | Monitoring | Tags

AMI:	rlab-web-8 (ami-38f47351)	Alarm Status:	none
Zone:	us-east-1a	Security Groups:	rlab-web. view rules
Type:	m1.small	State:	running
Scheduled Events:	No scheduled events	Owner:	670934762290
VPC ID:	-	Subnet ID:	-
Source/Dest. Check:		Virtualization:	paravirtual

© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

Testing the deployment

To test the deployment, copy the **domain name** of the instance (1) and paste it in a new tab of your web browser.


Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Alarm Status	Monitoring	Securi
web	i-fd860382	ami-38f47351	instance store	m1.small	running	2/2 checks passed	none	basic	rlab-we

1 EC2 Instance selected.

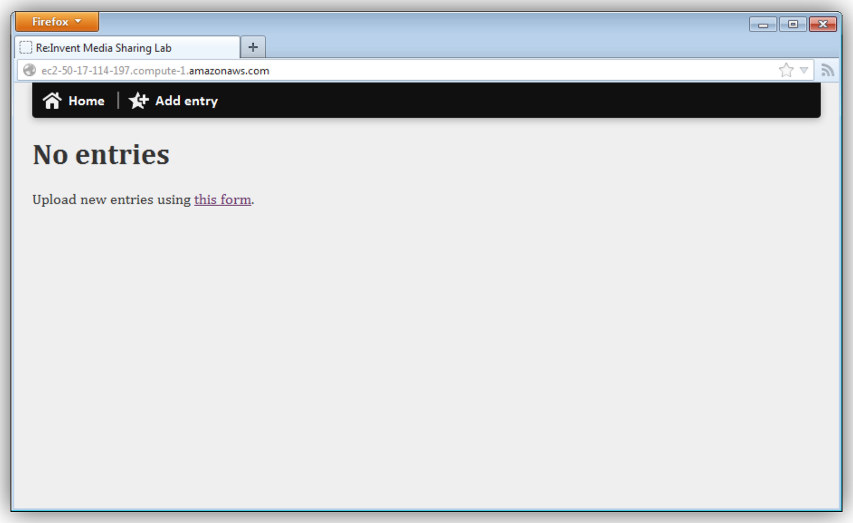
EC2 Instance: web (i-fd860382)

ec2-50-17-114-197.compute-1.amazonaws.com

1

AMI:	rlab-web-8 (ami-38f47351)	s:	none
Zone:	us-east-1a	oups:	rlab-web. view rules
Type:	m1.small		running
Scheduled Events:	No scheduled events	Owner:	670934762290
VPC ID:	-	Subnet ID:	-
Source/Dest. Check:		Virtualization:	paravirtual

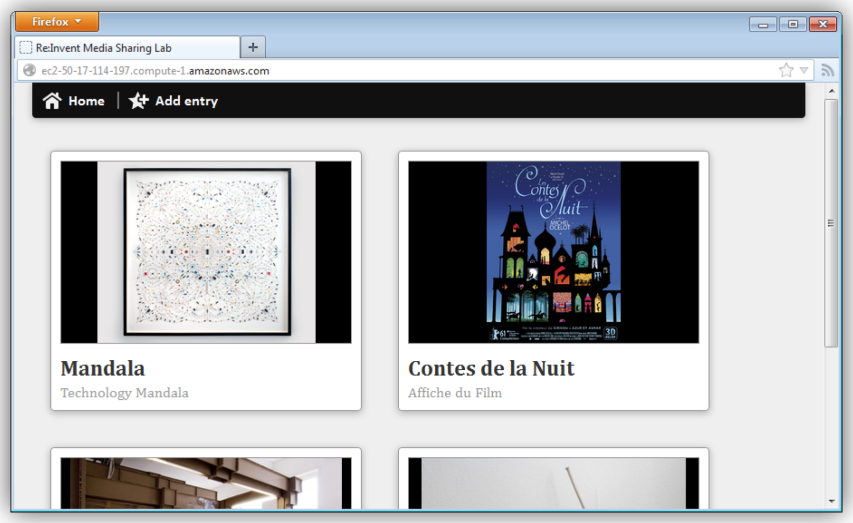
You should see the home page of the web application:


Firefox
Re:Invent Media Sharing Lab
ec2-50-17-114-197.compute-1.amazonaws.com
Home
Add entry

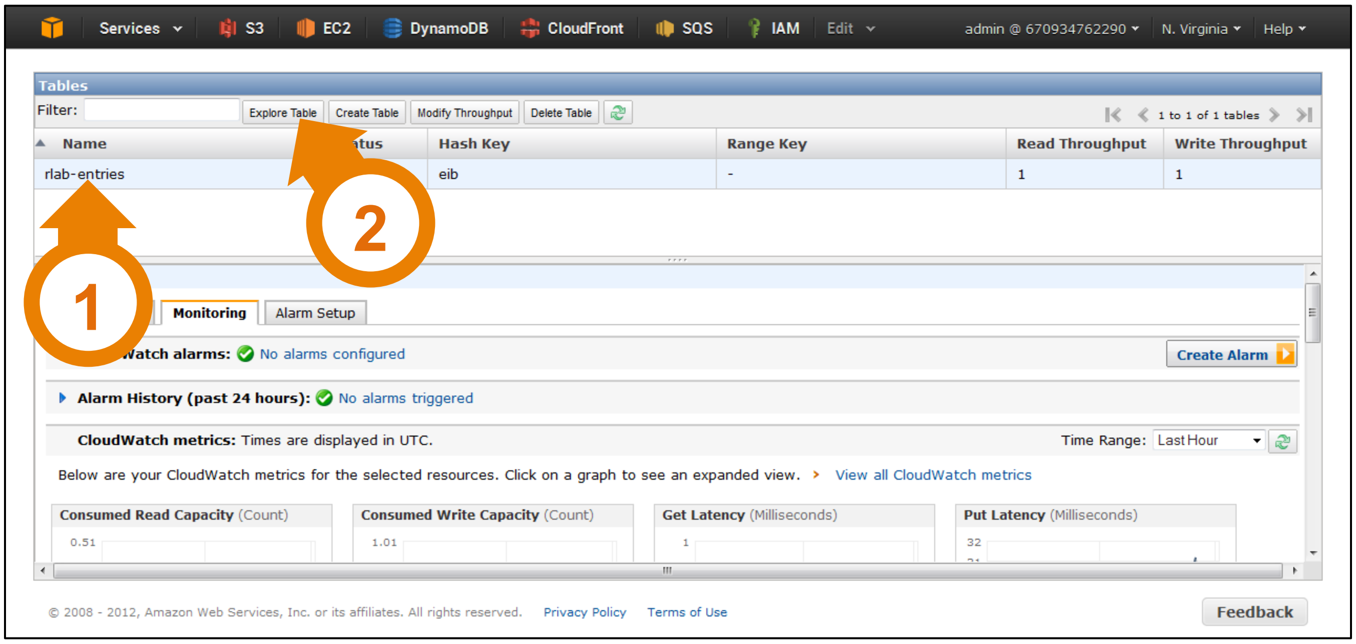
No entries

Upload new entries using this form.

Add a couple of images (use images from the laptop) and play around with the application:



Once you added a couple of images in the web application, go back to the AWS Web Console, and open the DynamoDB section. Select the table you created for the application (1) and click the "**Explore Table**" button (2):



Name	Status	Hash Key	Range Key	Read Throughput	Write Throughput
rlab-entries		eib	-	1	1

Watch alarms: No alarms configured

Alarm History (past 24 hours): No alarms triggered

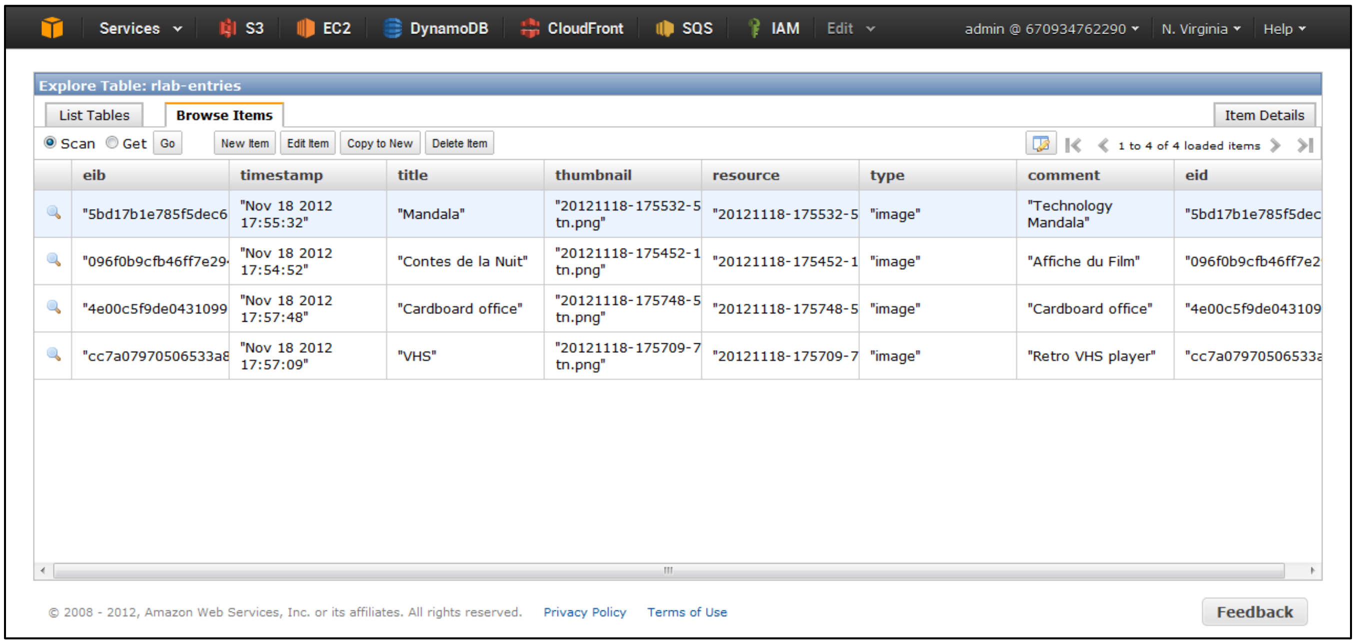
CloudWatch metrics: Times are displayed in UTC.

Below are your CloudWatch metrics for the selected resources. Click on a graph to see an expanded view.

Consumed Read Capacity (Count) 0.51 | Consumed Write Capacity (Count) 1.01 | Get Latency (Milliseconds) 1 | Put Latency (Milliseconds) 32

© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

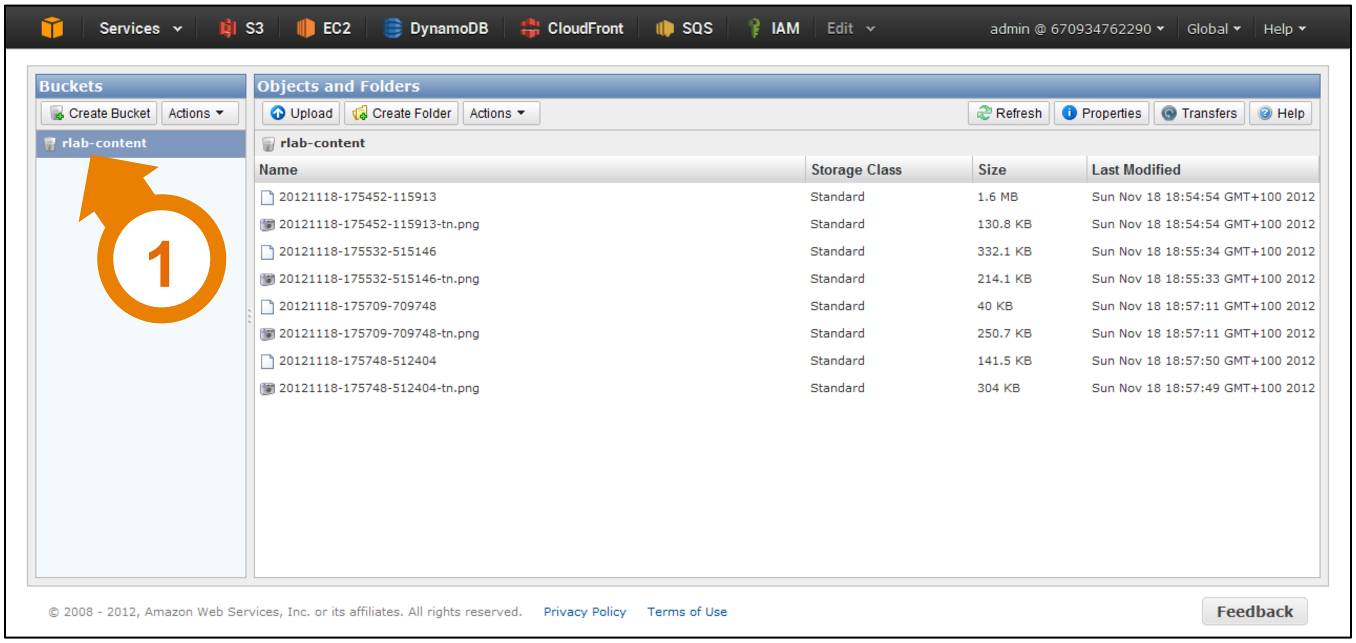
You should see your content metadata as stored by the web application:



Explore Table: rlab-entries

eib	timestamp	title	thumbnail	resource	type	comment	eid
"5bd17b1e785f5dec6	"Nov 18 2012 17:55:32"	"Mandala"	"20121118-175532-5 tn.png"	"20121118-175532-5	"image"	"Technology Mandala"	"5bd17b1e785f5dec
"096f0b9cfb46ff7e29-	"Nov 18 2012 17:54:52"	"Contes de la Nuit"	"20121118-175452-1 tn.png"	"20121118-175452-1	"image"	"Affiche du Film"	"096f0b9cfb46ff7e2
"4e00c5f9de0431099	"Nov 18 2012 17:57:48"	"Cardboard office"	"20121118-175748-5 tn.png"	"20121118-175748-5	"image"	"Cardboard office"	"4e00c5f9de043109
"cc7a07970506533a8	"Nov 18 2012 17:57:09"	"VHS"	"20121118-175709-7 tn.png"	"20121118-175709-7	"image"	"Retro VHS player"	"cc7a07970506533a

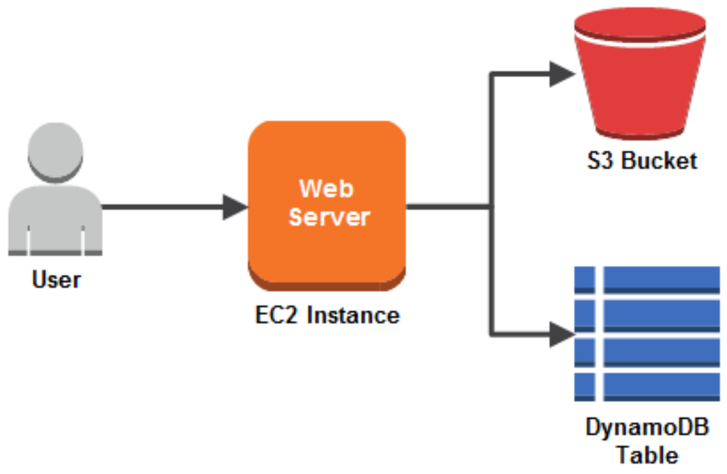
You can also check how media files have been stored in the S3 bucket. From the AWS Web Console, go to the Amazon S3 section and select the bucket you created for this lab (1). You will see the list of image files uploaded by the application in the bucket along with the thumbnail images (with the "-tn.png" extension):



Name	Storage Class	Size	Last Modified
20121118-175452-115913	Standard	1.6 MB	Sun Nov 18 18:54:54 GMT+100 2012
20121118-175452-115913-tn.png	Standard	130.8 KB	Sun Nov 18 18:54:54 GMT+100 2012
20121118-175532-515146	Standard	332.1 KB	Sun Nov 18 18:55:34 GMT+100 2012
20121118-175532-515146-tn.png	Standard	214.1 KB	Sun Nov 18 18:55:33 GMT+100 2012
20121118-175709-709748	Standard	40 KB	Sun Nov 18 18:57:11 GMT+100 2012
20121118-175709-709748-tn.png	Standard	250.7 KB	Sun Nov 18 18:57:11 GMT+100 2012
20121118-175748-512404	Standard	141.5 KB	Sun Nov 18 18:57:50 GMT+100 2012
20121118-175748-512404-tn.png	Standard	304 KB	Sun Nov 18 18:57:49 GMT+100 2012

Architecture Overview

So far, we created the following system:



When the user uploads an image, the web server receives it and creates a thumbnail. It will then upload the image and the thumbnail to the S3 bucket and insert the image metadata in the DynamoDB table.

While Amazon S3 and Amazon DynamoDB are both scalable and fault-tolerant systems, our web server running on a single Amazon EC2 instance is clearly a single point of failure (if the web application fails, the system is not accessible and cannot recover) and a bottleneck (with an important load of incoming requests, the system might even become unavailable). You will fix both issues in the next section.

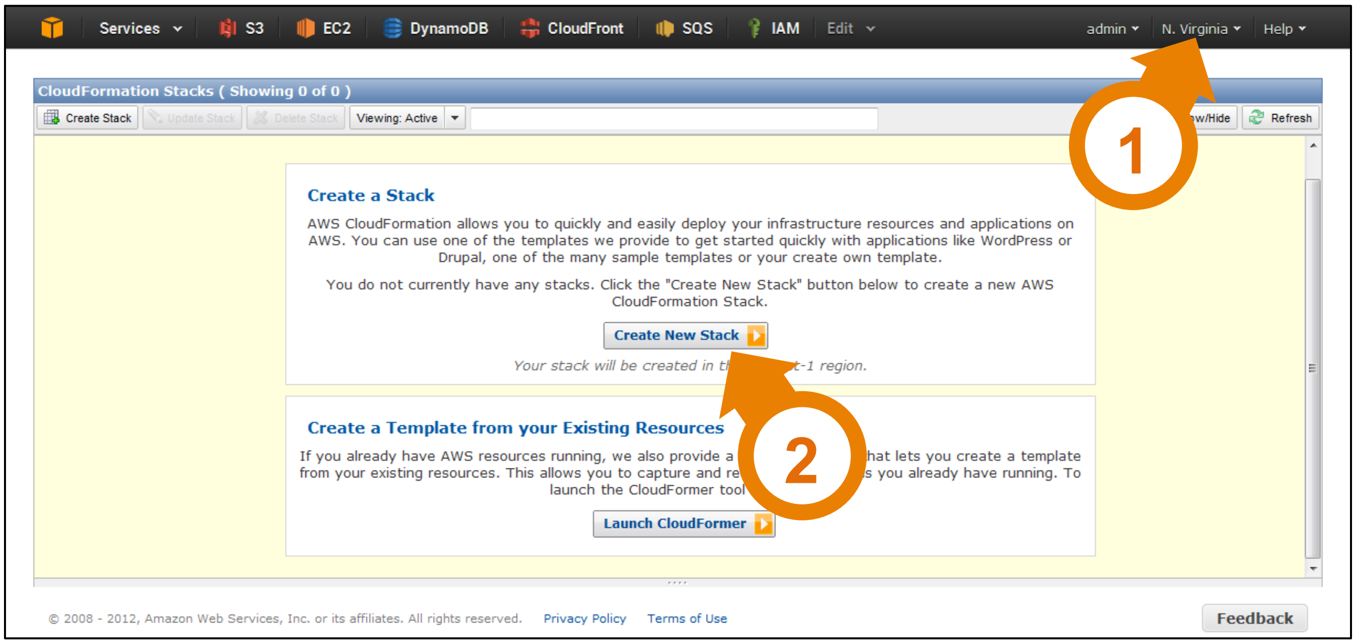
Step 5 – Scalable Architecture Deployment

To solve both the scalability and the single point of failure problem, you will use **Auto Scaling**. Auto Scaling allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define, removing the bottleneck problem. Now with multiple web servers in the front-end, you need to distribute incoming HTTP traffic to them. **Elastic Load Balancing** automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic. Elastic Load Balancing detects unhealthy instances within a pool and automatically reroutes traffic to healthy instances until the unhealthy instances have been restored, or replaced by Auto Scaling.

To deploy the new front-end, you will use an AWS CloudFormation script which will create the Auto Scaling groups and Elastic Load Balancers for you. AWS CloudFormation enables you to create and delete related AWS resources together as a unit called a stack. You define the characteristics of a stack parameters, mappings, resource properties, and output values) using a template (a JSON-compliant text file).

Deployment using CloudFormation

Open the AWS CloudFormation section of the AWS Web Console. Select the **US East (N.Virginia)** region (1), and then click the "**Create New Stack**" button (2):



Create a Stack

AWS CloudFormation allows you to quickly and easily deploy your infrastructure resources and applications on AWS. You can use one of the templates we provide to get started quickly with applications like WordPress or Drupal, one of the many sample templates or your create own template.

You do not currently have any stacks. Click the "Create New Stack" button below to create a new AWS CloudFormation Stack.

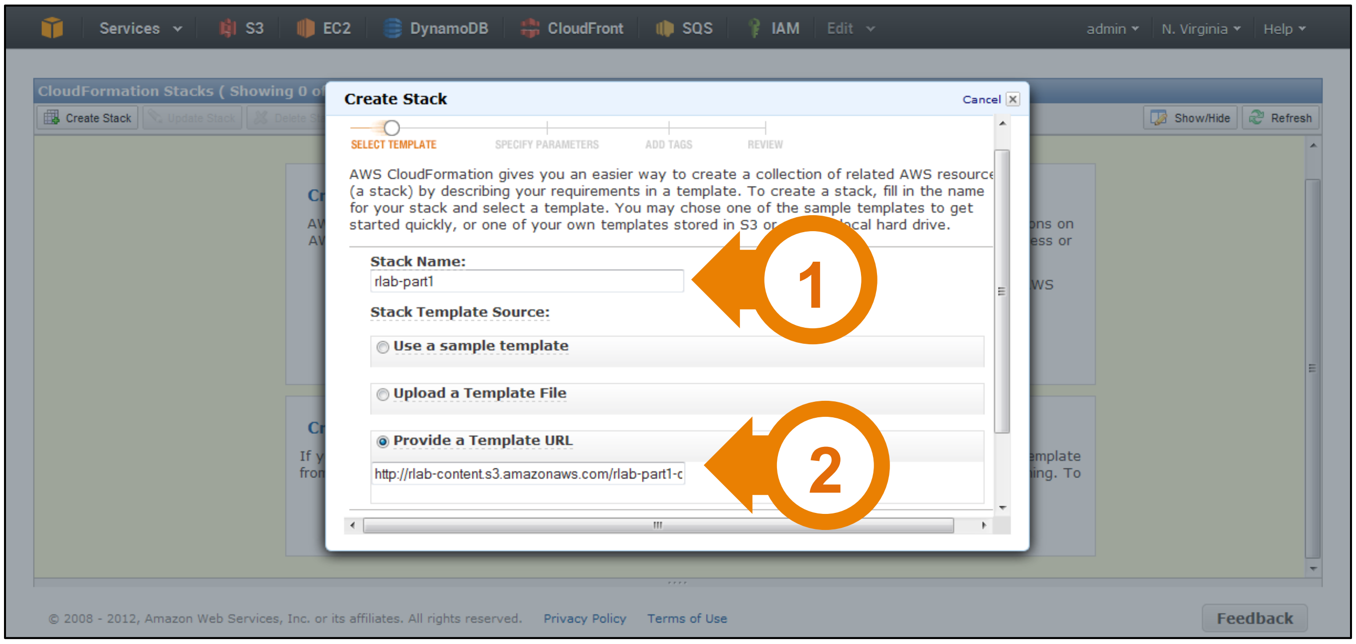
Your stack will be created in th…t-1 region.

Create a Template from your Existing Resources

If you already have AWS resources running, we also provide a … hat lets you create a template from your existing resources. This allows you to capture and re… s you already have running. To launch the CloudFormer tool

© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Provide a name to the CloudFormation stack (1) and choose "**Provide a Template URL**" for the origin of the stack (2). The URL containing the stack for this part of the lab is http://rlab-content.s3.amazonaws.com/rlab-part1-cfn.template :



AWS CloudFormation gives you an easier way to create a collection of related AWS resource (a stack) by describing your requirements in a template. To create a stack, fill in the name for your stack and select a template. You may chose one of the sample templates to get started quickly, or one of your own templates stored in S3 or …ocal hard drive.

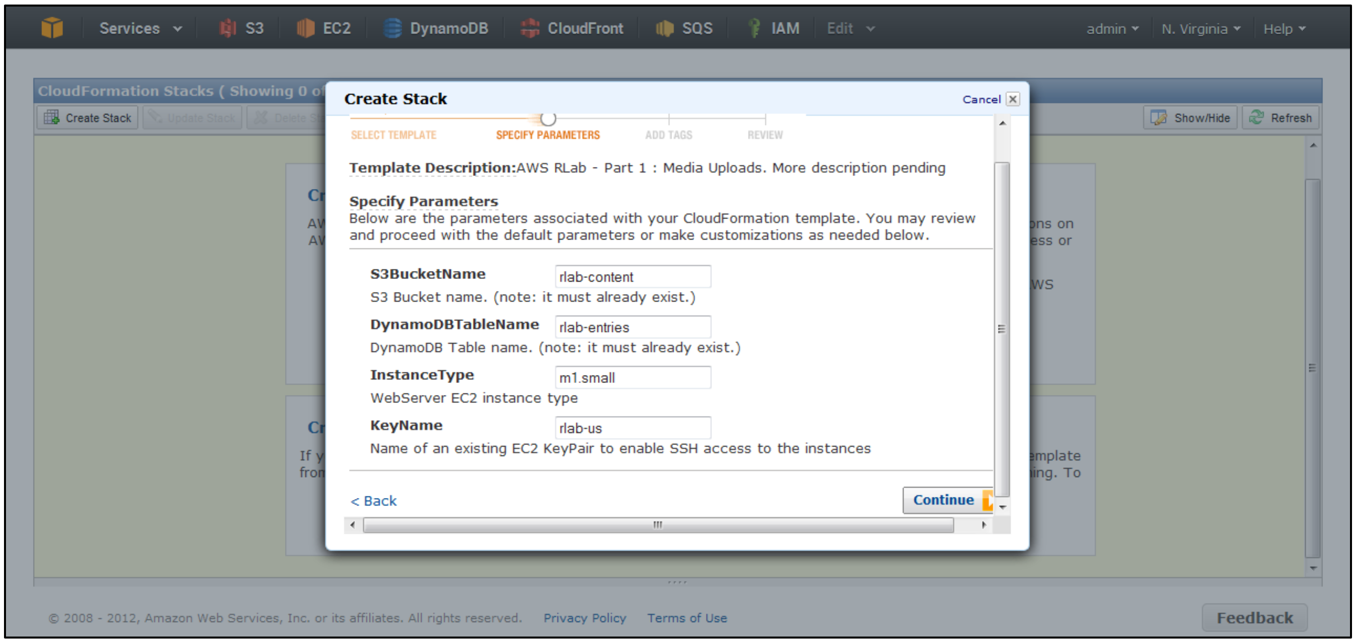
Stack Name: rlab-part1

Stack Template Source:

- Use a sample template
- Upload a Template File
- [✓ Provide a Template URL] http://rlab-content.s3.amazonaws.com/rlab-part1-c

© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The stack contains parameter placeholders that allows you to specify your DynamoDB table, your S3 bucket, customize the instance size and provide a Key Pair name (if applicable) for remote login:

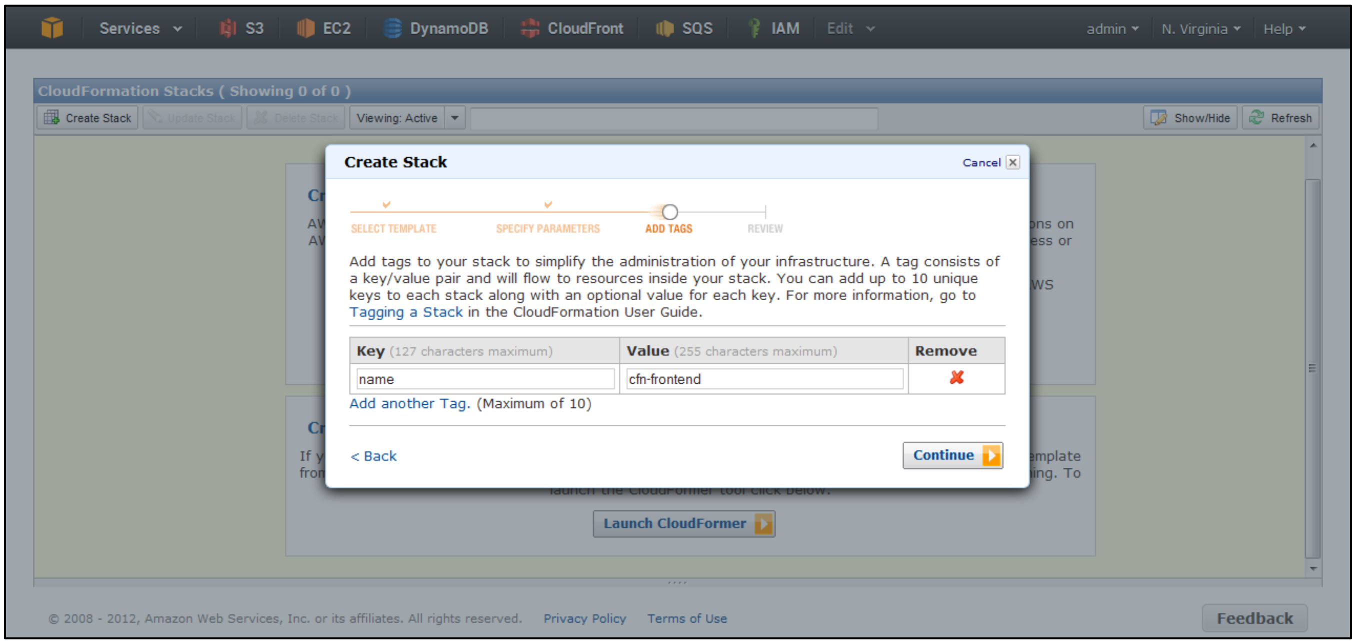


Template Description:AWS RLab - Part 1 : Media Uploads. More description pending

Specify Parameters
Below are the parameters associated with your CloudFormation template. You may review and proceed with the default parameters or make customizations as needed below.

S3BucketName	rlab-content
S3 Bucket name. (note: it must already exist.)	
DynamoDBTableName	rlab-entries
DynamoDB Table name. (note: it must already exist.)	
InstanceType	m1.small
WebServer EC2 instance type	
KeyName	rlab-us
Name of an existing EC2 KeyPair to enable SSH access to the instances	

You can provide some tags which will be assigned to the resources created by this stack:

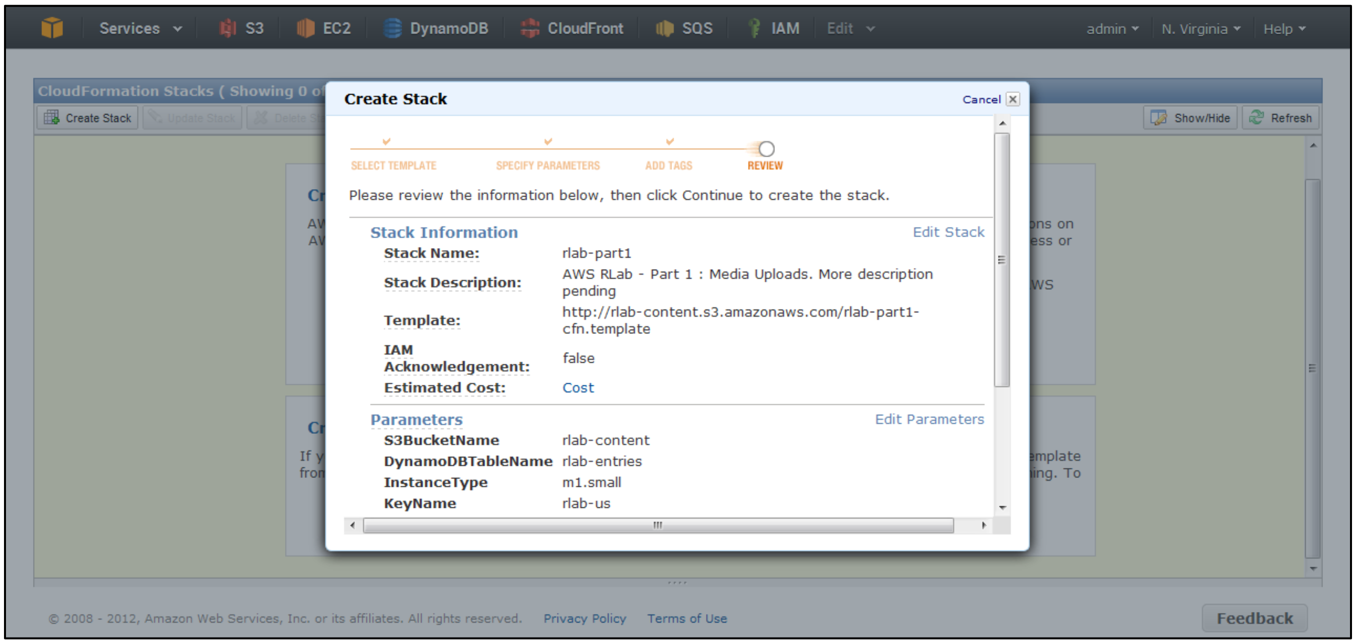


Add tags to your stack to simplify the administration of your infrastructure. A tag consists of a key/value pair and will flow to resources inside your stack. You can add up to 10 unique keys to each stack along with an optional value for each key. For more information, go to Tagging a Stack in the CloudFormation User Guide.

Key (127 characters maximum)	Value (255 characters maximum)	Remove
name	cfn-frontend	✘

Add another Tag. (Maximum of 10)

Finally, review your stack parameters and click “**Create**”:



Create Stack

SELECT TEMPLATE · SPECIFY PARAMETERS · ADD TAGS · REVIEW

Please review the information below, then click Continue to create the stack.

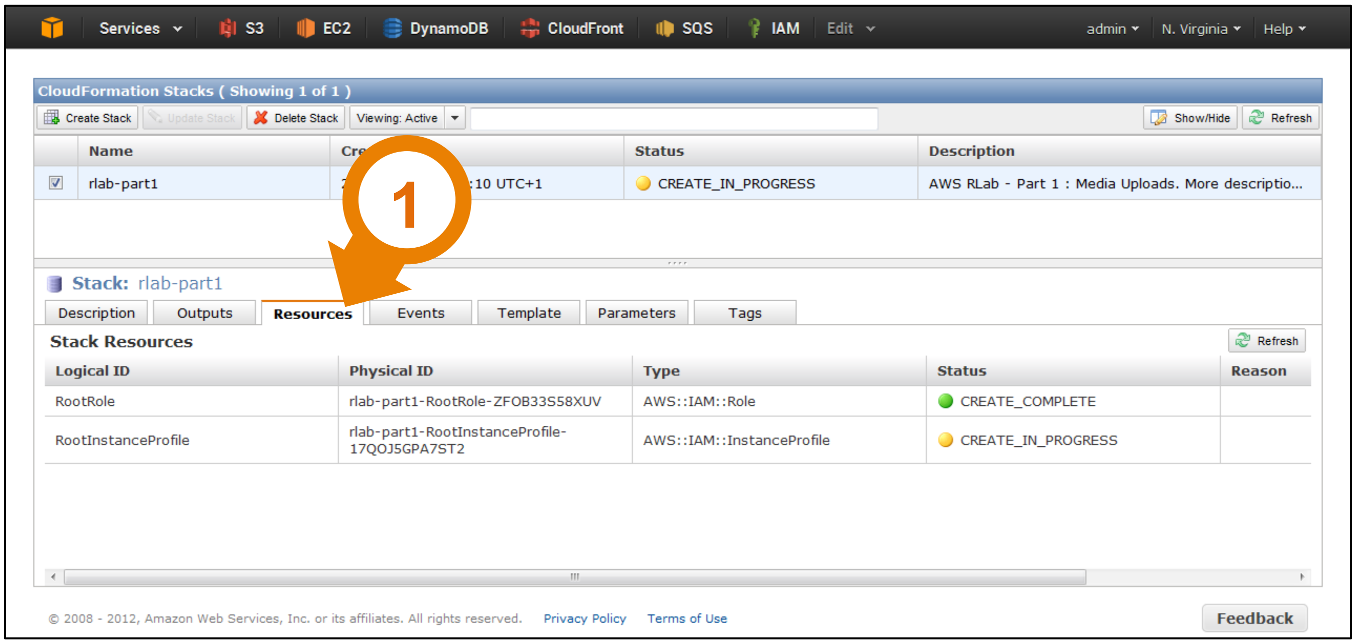
Stack Information — Edit Stack

Stack Name:	rlab-part1
Stack Description:	AWS RLab - Part 1 : Media Uploads. More description pending
Template:	http://rlab-content.s3.amazonaws.com/rlab-part1-cfn.template
IAM Acknowledgement:	false
Estimated Cost:	Cost

Parameters — Edit Parameters

S3BucketName	rlab-content
DynamoDBTableName	rlab-entries
InstanceType	m1.small
KeyName	rlab-us

The stack creation process will take a couple of minutes. You can follow the stack creation process by checking the “**Resources**” tab (1):



CloudFormation Stacks (Showing 1 of 1)

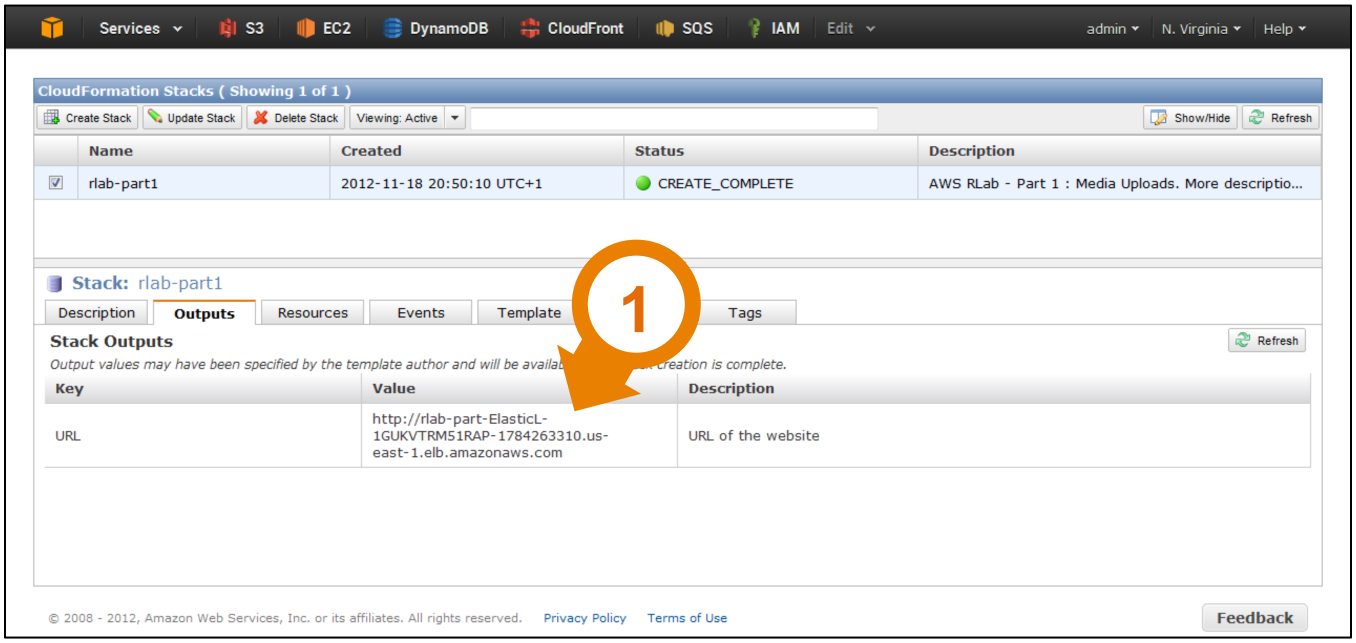
Name	Cre…	Status	Description
rlab-part1	…10 UTC+1	CREATE_IN_PROGRESS	AWS RLab - Part 1 : Media Uploads. More descriptio...

Stack: rlab-part1

Stack Resources

Logical ID	Physical ID	Type	Status	Reason
RootRole	rlab-part1-RootRole-ZFOB33S58XUV	AWS::IAM::Role	CREATE_COMPLETE	
RootInstanceProfile	rlab-part1-RootInstanceProfile-17QOJ5GPA7ST2	AWS::IAM::InstanceProfile	CREATE_IN_PROGRESS	

Once the stack creation is completed, the "**Outputs**" tab will show the URL endpoint of the Elastic Load Balancer (1):



CloudFormation Stacks (Showing 1 of 1)

	Name	Created	Status	Description
☑	rlab-part1	2012-11-18 20:50:10 UTC+1	● CREATE_COMPLETE	AWS RLab - Part 1 : Media Uploads. More descriptio...

Stack: rlab-part1

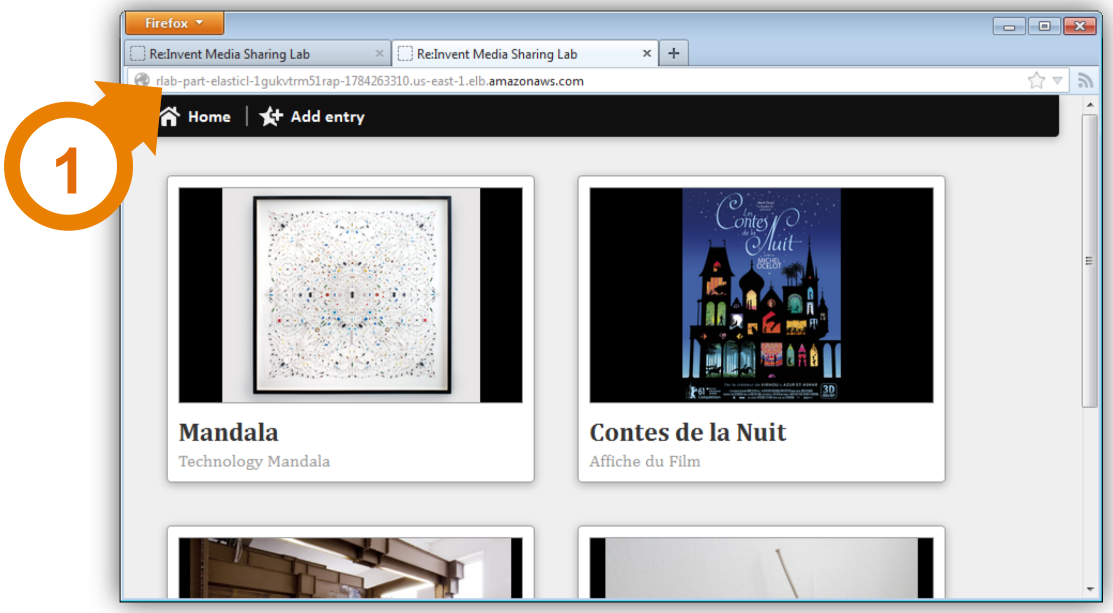
Stack Outputs

Output values may have been specified by the template author and will be availab... creation is complete.

Key	Value	Description
URL	http://rlab-part-ElasticL-1GUKVTRM51RAP-1784263310.us-east-1.elb.amazonaws.com	URL of the website

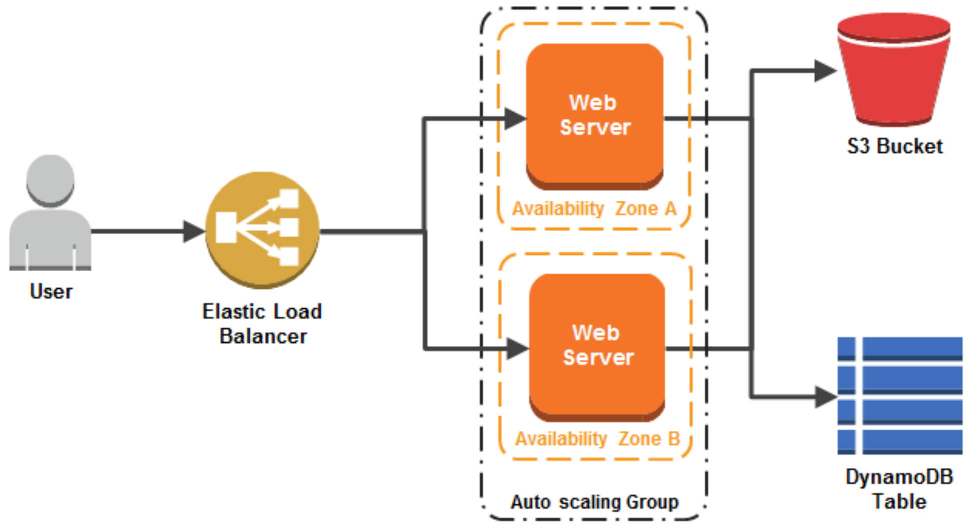
© 2008 - 2012, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Copy and paste this URL in a new tab of your web browser (1). You should see the web application home page:


Firefox
Re:Invent Media Sharing Lab
Re:Invent Media Sharing Lab
rlab-part-elasticl-1gukvtrm51rap-1784263310.us-east-1.elb.amazonaws.com
Home
Add entry
1
Mandala
Technology Mandala
Contes de la Nuit
Affiche du Film

Final Architecture Overview

The following diagram represents the final architecture you built in this lab:



The front end is now as fault-tolerant and scalable as the data storage systems. In case of increased inbound traffic, the Auto Scaling group will automatically provision new instances, and replace instances in case of failure.