Third-party Apps and Tools Integration with AWS

As Amazon Web Services is one of the leading Cloud service providers by any metric, it provides many tools for integration and application development. Still, in the database and many other sections, there are a few more providers, such as MySQL and MarkLogic, who provide more flexible features. So, AWS provides integration with tools such as MySQL or MarkLogic. This chapter will introduce several third-party applications that can be used in conjunction with AWS. We will also discuss the basic concepts behind those applications, for a better understanding of the purpose of AWS. We'll cover the following topics in this chapter:

- Enabling infrastructure for third-party app integration
- Third-party tools for integration and development
- Integration techniques best practices

Enabling infrastructure for third-party app integration

All third-party applications will access the AWS infrastructure through API calls. Most of the tools are loosely coupled, so they don't require a special preparation of the infrastructure.
However, there are a few points that help us to prepare our AWS infrastructures, which are listed as follows:

- The primary preparation of infrastructure includes providing proper credentials to the third-party tools so that they can interact with AWS on our behalf. Some tools ask us to provide our access key and security key, which we are not supposed to share with anyone else. So it's handy to change these values frequently or we can say a tool-specific key with a very restrictive set of permissions can be created so that we can be sure that the third-party tools don't cause any trouble.

- The AWS infrastructure must be managed based on the user's identity. So we need to create a new IAM role and rule through which all the third-party tools communicate.

- Some of the third-party tools also look for payment for their services. We don't want too many cooks to spoil the broth, so we should either pay through AWS or use a centralized monitoring tool (such as Ylastic), which helps us manage and monitor our account with multiple cloud providers and third-party tools. Even though these tools are expensive, they were proven to be effective in many cases.

- The third-party tools must work in conjunction with an AWS infrastructure. So the closer the infrastructure and server in which the tool is running, the faster and more efficient the access. So we need to choose a proper region of the AWS service (working with third-party tools).

- If the third-party tool being obtained through AWS marketplace is a kind of database (for example, MarkLogic, which is discussed further), obviously it will be running as an EC2 instance. So starting the instance that explicitly stops will change the public DNS and IP address. So we should take an Elastic IP from AWS.

**Third-party tools for integration and development**

There are a handful of pioneers who provide expert services in certain areas in which AWS is yet to explore. For most of those tools, AWS provides support and helps us to integrate them easily with our AWS infrastructure. AWS already has a single place in which we can find all the trusted third-party tools; the place is called the AWS marketplace. Whatever tools are available in the marketplace, everything is trusted and these tools will not save our AWS credentials anywhere in their database.
Usually, these tools will be installed in an EC2 instance running in our account, whose inbound and outbound traffic can be controlled by us. Stopping or terminating the instance will be equivalent to uninstalling the software. There are more than 2,000 third-party tools available in the AWS marketplace and millions of third-party tools outside AWS. We will discuss the following three tools, out of which, one (MarkLogic), is available in the marketplace and the other two (Extended S3 Browser and Elasticfox) outside AWS. Before discussing third-party tools, we should know the licensing models in the marketplace. AWS supports three kinds of license models:

- Free — You can use it directly without worrying about the licensing.
- Included — Licensing charges will be included for a resource in the resource cost. So you don't need to pay explicitly for the licensing cost.
- BYOL — Bring Your Own License (BYOL) which means you have to buy the product license and provide the license credentials while using the product on AWS.

So, let's start the discussion about third-party tools beginning with MarkLogic.

**MarkLogic AMI – a third-party database**

MarkLogic is one of the emerging enterprise NoSQL proprietary software with integration to Hadoop and related technologies. MarkLogic offers several features related to transactional processing, which is lacking in most NoSQL databases. It began its development as a simple XML database, which turned into an enterprise NoSQL database over 13 years. MarkLogic landscapes contain replication, rollback, automated failover, point-in-time recovery, restore, backup to Amazon S3, JSON, can run unswervingly on Hadoop Distributed File System (HDFS), parallelized ingest, role-based security, full text search, location services, geospatial alerting, RDF triple store, and SPARQL query support.

MarkLogic Server is presented under several licensing and delivery prototypes. These were publicized in October 2013:

- **MarkLogic Developer**: The full-featured version is openly available. It includes APIs outspread to all versions of MarkLogic. It is not for production practice.
- **MarkLogic Essential Enterprise**: This is a fully loaded Enterprise NoSQL database that involves search engine, replication, backup, high availability, recovery, fine-grained security, location skills, and alerting. Semantics and superior language packs are alternatives. It is available as a long-lasting license, term/yearly license, or hourly on AWS.
Third-party Apps and Tools Integration with AWS

- **MarkLogic Global Enterprise**: This version is designed for practice in large, globally disseminated applications. Semantics, tiered storage, geospatial alerting, and superior language packs are options.

All these specialized servers are available as AMIs in AWS. We can choose one (or more) of these and make it available only to us or for the entire world to witness.

MarkLogic's Enterprise NoSQL database platform is broadly practiced in advertising, government, economics, and other areas, with hundreds of large-scale systems in production. The following are the organizations using MarkLogic:

- **MarkMail**: This is an openly available public mailing list archive service that accentuates interactivity and search analytics. MarkMail has approximately four million e-mail messages. As of 24 November 2013, the service claims the inclusion of 66,058,071 messages, 8,761 lists, of which 2,975 are active lists. The library includes complete history list for Apache, FreeBSD, GNOME, Jabber, Java, .net, KDE, Mozilla, MySQL, OpenOffice.org, Perl.org, PostgreSQL, Python, Red Hat, Ruby, W3C, and Xen, among others.

- **Boeing**: It includes nationalized security and intellect functions.

- **Federal Aviation Administration**: It is a disaster operations network

- **Library of Congress**: This is the leading library in the world and a liable for manufacturing substances obtainable by Congress and the American public practices MarkLogic to search, regain, and display video, data, and digitized documents from the Library's assemblies.

- **McGraw Hill**: Here it is used for revenue producing functions and is a prototyping program that builds new market prospects in as little as two weeks.

- **Warner Bros**: It tops digital supply chain, dealing with metadata throughout the organization.

- **Wiley**: Here it is used for strategic publishing function.

**Getting the MarkLogic AMI**

MarkLogic developers don't have to rent an EC2 instance separately, and then install MarkLogic in it and manage it separately. As it is available as an AMI, the scalability and operability is provided by AWS. The link [https://aws.amazon.com/marketplace/seller-profile?id=e86516ea-dbbb-4e05-a2f0-6d6c0af22dc4](https://aws.amazon.com/marketplace/seller-profile?id=e86516ea-dbbb-4e05-a2f0-6d6c0af22dc4) provides more information about MarkLogic and the list of products available in the AWS marketplace.
Chapter 14

The page consists of two sections. The left-hand section has an overview of the MarkLogic product and its usage in various organizations. The right-hand section consists of a list of servers available as EC2 AMIs. The first two products, **MarkLogic Developer** and **MarkLogic Developer (HVM)**, require us to purchase a license by ourselves from MarkLogic (or continue with an evaluation for 30 days). The third product, **MarkLogic Essential Enterprise**, comes with a license for which we have to pay approximately 1 USD per hour. AWS will pay MarkLogic on our behalf. All we need to do is click on the respective product. Here, for demonstration (purposes and to evade billing), we will see the second product.

Clicking on the second product will take us to the following page. Here, the MarkLogic server's details will be displayed in the left-hand section. On this page, we can see that **MarkLogic 7.0-4** will be installed on the **64-bit Amazon Linux 2014.09** operating system. It requires an EBS volume attached to the instance.
The right-hand section consists of a **Continue** button to proceed with the product. Just below this button, we can see a region drop-down list (with **US East Virginia** selected), which lists the pricing details of different instance types in the corresponding regions.
Clicking on the **Continue** button will take us to the following page. Here, we need to read agreements from AWS and MarkLogic and then click on the **Accept Terms** button. Clicking on the **Launch with EC2 Console** button will launch the EC2 instance with MarkLogic pre-installed and port 22 and 7997 to 8100 open for all connections. However, it is currently disabled since the user is yet to accept the terms.
Once the user clicks on the **Accept Terms** button, the following page will be shown with the instructions displayed in **Next Steps**. As the first step, we need to confirm the subscription by clicking on the link sent by an e-mail.

After confirming the subscription, the **Launch with EC2 Console** button will be enabled (as shown in the following screenshot). All we need to do is click on the **Launch with EC2 Console** button corresponding to the **Region** in which we need the MarkLogic server to be hosted.
Clicking on the **Launch with EC2 Console** button will take us to the EC2 launch instance page with MarkLogic ami-id selected. We need to take care of two things (namely **Add Storage** and **Security Group**) minutely. The MarkLogic installation requires the `/dev/sdf` device or media to store all the data. MarkLogic will use the port number from 7997 to 8100. Failing the first pre-requisite will result in a failure to create a MarkLogic instance; failing the second step will make the server inaccessible to the external world (application or endpoint). Other steps for configuring the MarkLogic server instance are not of much importance. The EBS volume must not be deleted on termination because it will delete the entire MarkLogic database.

**Step 4: Add Storage**

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

<table>
<thead>
<tr>
<th>Type</th>
<th>Device</th>
<th>Snapshot</th>
<th>Size (GB)</th>
<th>Volume Type</th>
<th>IOPS</th>
<th>Delete on Termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>/dev/xvda1</td>
<td>snap-70e72968</td>
<td>8</td>
<td>General Purpose (SSD)</td>
<td>24 / 3000</td>
<td>✔</td>
</tr>
<tr>
<td>EBS</td>
<td>/dev/xvd1</td>
<td>Search (case-insensitive)</td>
<td>8</td>
<td>General Purpose (SSD)</td>
<td>24 / 3000</td>
<td>✔</td>
</tr>
</tbody>
</table>

Add New Volume

After following all the steps, we can see the instance running. Once the instance is running, we can be sure that MarkLogic is installed on the instance and all the services corresponding to MarkLogic are started and running as shown in the following image:
After successful installation, we can copy either the Public DNS or Public IP and we can try to access port number 8001. It will display the following page, which configures the MarkLogic server. As we chose to use our own license in the first step, the following page displays No license key has been entered. We can click on the ok button and start with MarkLogic.

Further information about MarkLogic is available at http://developer.marklogic.com/ and it is an enterprise NoSQL database worth spending our time on.

Browser and device-based tools
Along with several other tools available in AWS, there are a handful of tools that serve as add-ons to browsers and Android devices as well. We will see a few of these tools here.

Android apps
The AWS Console mobile app, delivered by Amazon Web Services, lets you look out for resources for particular services. The app also provides a regulated set of management utilities for select resource types, so you can practice the app to support occurrence response while you’re on the go. EC2, S3, Route 53, ELB, RDS, AWS Elastic Beanstalk, CloudFormation, DynamoDB, Auto Scaling, and AWS OpsWorks can use the app to browse resources and view configuration aspects, metrics, and alarms. Management functionality is sustained for EC2, RDS, AWS Elastic Beanstalk, DynamoDB, Auto Scaling, and AWS OpsWorks.

Cloud Hub is a freely available Android app that specifies the basic level management of EC2, S3, SNS, RDS, Route 53, and CloudFront services.
Decaf Amazon EC2 Client is an advanced level application, priced at $14.16, which lets us direct and monitor EC2 instances. Though it doesn't provision other AWS services (such as RDS or Route 53), it does compromise pronounced functionality for EC2, incorporating SMS alerts and SSH client integration with ConnectBot.

Cloud services
Ylastic (http://www.ylastic.com) is a Cloud-based service that supports accomplished and monitor manifold AWS services from a web browser or mobile device, starting at $25 per month. Ylastic provides EC2, S3, Route 53, RDS, Auto Scaling, IAM, CloudWatch, CloudFront, SDB, SNS, SQS, CloudFormation, and VPC. It deals with monitoring proficiencies with alerts, live visualization, and slating functionality, including automated backups. It also stipulates AWS Account Advisor, Amazon Simple Email Service, and AWS Employing Analytics tools, as well as security and disaster recovery landscapes. Each big spender (AWS bills more than 100 USD periodically) will notice that Ylastic has carried their monthly bill because of its account mentor.

You can also check other Cloud services, such as Rightscale and Newvem, as they are also widely adopted services around the globe.

Browser-based tools
Extended S3 Browser is an extension that enables quick access to S3 files in the Chrome browser. It adds an emblem to the Chrome toolbar (as shown in the following screenshot) that creates the web interface to view, delete, and upload files. You can also form folders, view the belongings and metadata of designated files, and add and manage buckets.

The add-ons can be downloaded from https://chrome.google.com/webstore/detail/extended-s3-browser/ddmmmnbkhpkgnkafpflhaoohifdpkmg/related in our Chrome browser. We can add the free add-ons, which will add a new icon (with three buckets or cubes) to the browser:
Clicking on the toolbar icon will ask us to provide our AWS Access Key and Security Key and click on the Save button. Without authenticating ourselves, we cannot perform any S3 operation.

![S3 Credentials](image)

Correct authentication will display all the buckets (in the drop-down list with DynamoDBbook selected) available with the account. We can select the corresponding bucket, folder, or file and perform whatever operation (Add Folder or Upload files to the current location) we desire.

![S3 Interface](image)
Clicking on the button next to the **DynamoDBbook** drop-down will help us to manage buckets. We can decide on a default bucket to be displayed every time we access this add-on. We can even delete the bucket and its data here by clicking on the cross mark that precedes the bucket name.

In order to get the file property and metadata, all we need to do is click on the corresponding file. Clicking on the **DynamoDB2.docx** file will show its property, as shown in the following screenshot. An option to download or delete the file is also shown:
ElasticFox-EC2tag is a modernized version of a well-known ElasticFox extension for Mozilla Firefox, which presents the capability to manage the EC2 infrastructure. Through the ElasticFox-EC2tag, the creation and management of AMIs, instances, volumes, Elastic IPs, and key pairs will not get much easier.

Unlike ElasticFox, ElasticFox-EC2tag is not a Firefox extension; it is a standalone software to be installed. It is available to download at http://elasticfox-ec2tag.s3-website-ap-northeast-1.amazonaws.com/.

Accessing Elasticfox for the first time will look like the following screenshot. It consists of five icons in the first line. The Regions icon (followed by a drop-down) is used to choose the region in which we need to manage EC2 instances. The Credentials icon is used to configure Elasticfox with the AWS credentials. The Account IDs icon is used to configure multiple AWS account IDs. The Tools icon is used to configure the Elasticfox tool. The About icon has details about the selected tool.

Clicking on the Credentials icon will open the following window, which asks for Account Name, Access Key and Secret Access Key. After providing proper values, clicking on the Add icon will configure and add an AWS account to the tool.
After adding the AWS account, **Account Name** will be populated in the drop-down list next to the **Credentials** icon. In the case of multiple accounts, we can choose one of the accounts from the populated drop-down list.

![Account Name populated in drop-down list]

Clicking on the **Tools** icon will open the window to configure the tool. It has information about the puTTY executable location, SSH username, and key-pair location and name.

![Manage EC2 Tools]

- **SSH Command**: \$\{Program Files\Putty\putty.exe
- **SSH Arguments**: -i "$\{key\}" $\{user\}@$\{host\}
- **SSH Key Template**: $\{home\}\ecc\ecc\$\{keyname\}.ppk
- **EC2 Private Key Template**: $\{home\}\ecc\ecc\$\{keyname\}.pem
- **SSH User**: root
- **RDP Command**: $\{\%Windows\%\}System32\mstsc.exe
- **RDP Arguments**: /v $\{host\}

- $\{home\}$ - Your home directory.
- $\{keyname\}$ - The EC2 keypair name the instance was launched with.
- $\{host\}$ - The public DNS name / IP address of the EC2 instance.
- $\{publicDnsName\}$ - The public DNS name of the EC2 instance.
- $\{privateDnsName\}$ - The private DNS name of the EC2 instance.
- $\{privateIpAddress\}$ - The private IP address name of the EC2 instance.
- $\{name\}$ - The "Name" tag of the EC2 instance.
- $\{key\}$ - The (interpolated) key template.
- $\{user\}$ - The configured user name.
- $\{pass\}$ - The Administrator password for a Windows instance.
Choosing the correct account and region will display all the instances associated with it. The first tab is the **Instances** tab, which shows all the instance-related parameters. In the following figure, we can see the instance name as **tagElastixFox**, the instance ID as **i-279d562d**, the state as **running**, the public DNS as **ec2-54-69-61-138.us-west-2.compute.amazonaws**, the private DNS as **ip-172-31-40-67.us-west-2.compute.amazonaws**, the private IP, key name, security groups, instance type, and launch time.

The preceding screenshot also shows several icons in red, green, and blue. The first icon is used to refresh, the second icon is used to start the instance, the third icon is used to stop, the fourth icon is used to reboot, and the fifth icon is used to terminate the selected EC2 instance. The sixth icon shows the console output of the instance. The seventh icon is used to connect to the instance using the PuTTY client. The final icon is used to rename the instance tag.

Selecting and then clicking on the stop instances icon will display the following popup. After clicking on **OK**, if we go back to the Elasticfox, we can see the status of the instance turn to **stopping**.

In case of any error or the invalid selection of an option, the following popup will be visible. Here, we are trying to attempt a reboot on the stopped instance (which is kind of absurd), which is the reason behind this error message. In case we feel that the error can be avoided by performing the action again, we can click on the **Retry** button.
Clicking on the Show Console Output icon will show the output as follows:

![Instance Console Output](image)

All the EC2 key pairs of the AWS account will be listed here by selecting the **KeyPairs** tab. It will display the key pair name and its fingerprint.

![KeyPairs Section](image)

**S3Fox** has the capability to upload, download, and synchronize files from S3 with an FTP client-like interface on a Firefox tab or popup. You can drag and drop files, delete files/folders, and form folders. You can also achieve S3 CloudFront Distributions, alter Access Control Policies, and create time-limited URLs.
Dome9 Instant Access can securely access cloud servers, incorporating EC2 instances. This extension adds an icon to the Chrome toolbar, delivering quick and secure access to cloud servers set up over Dome9.

SdbNavigator provides rapid and simple access to the SimpleDB (managed NoSQL) database. It adds an icon to the Chrome toolbar that carries a simple web interface to create, read, update, and delete both domains and records. You can also locate records using your own queries or sorting properties.

Integration techniques best practices
As we are well aware that AWS and our third-party tools are given authentication to use AWS on our behalf using our credentials, so, of course, it will incur charges to the associated account. Therefore, as a safety measure and for the efficient handling of third-party tools, we must follow most, or all, of the following practices:

• As AWS allows us to have two sets of access key and secret access key enabled for an account or use IAM as best practices, we need to use one of those for third-party tools and keep changing it as frequently as possible. This will restrict third-party tools from misusing our account.

• There are hundreds of third-party tools that provide the same functionalities but they differ in their policy agreements. So, we must read the agreement carefully and make sure that the tool will not create any resource by itself or store our credential information anywhere (except in the PC where the software is installed).

• Some tools (such as MarkLogic or Oracle) are licensed. So there is an option either to buy the license directly by paying the product or the tool owner. Most of the time, the license provided by AWS will be cheaper. So it's better to go with AWS. However, the product purchased through AWS might not cover for support (call or assistance) from the product owner. So we need to take this into consideration as well.

• Some of the tools are browser add-ons. Almost all the add-ons have a feature to save our AWS credentials too. We need to make sure that the auto-saving of AWS credentials does not happen in the browser (unless of course it’s a personal or portable device browser).

• When buying a third-party tool from the AWS marketplace, there will be two kinds of agreement. The first agreement from AWS and the second from the product owner. We need to read both the agreements and understand whether there are any security flaws incorporated because of one person (AWS) thinking that the other person (product owner) handles the security concerns of the tool.
• Some third-party tools may install a few other pieces of software (along with what we requested), which might act as a security stealer. So it is crucial to know the developer of the tool and read the user reviews before providing our AWS security access keys on the tool.

The following are the names of some useful tools for management, security, analytics, and monitoring that you should also consider and try to use:

<table>
<thead>
<tr>
<th>Use</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Billing and Management</td>
<td>Newvem</td>
</tr>
<tr>
<td></td>
<td>Netflix Ice</td>
</tr>
<tr>
<td></td>
<td>Netflix AMGard</td>
</tr>
<tr>
<td></td>
<td>RightScale</td>
</tr>
<tr>
<td></td>
<td>Cloud8</td>
</tr>
<tr>
<td>Compute resources</td>
<td>Storm</td>
</tr>
<tr>
<td></td>
<td>Apache Mesos</td>
</tr>
<tr>
<td></td>
<td>MIT Star Cluster</td>
</tr>
<tr>
<td></td>
<td>Apache Ambari</td>
</tr>
<tr>
<td>Monitoring</td>
<td>DataDog</td>
</tr>
<tr>
<td></td>
<td>Stack Driver</td>
</tr>
<tr>
<td></td>
<td>Librato</td>
</tr>
<tr>
<td></td>
<td>CopperEgg</td>
</tr>
<tr>
<td></td>
<td>New Relic</td>
</tr>
<tr>
<td>Logging and Analytics</td>
<td>Logstash</td>
</tr>
<tr>
<td></td>
<td>Kibana</td>
</tr>
<tr>
<td></td>
<td>Fluentd</td>
</tr>
<tr>
<td></td>
<td>Apache Flume</td>
</tr>
<tr>
<td></td>
<td>Splunk</td>
</tr>
<tr>
<td></td>
<td>Logly</td>
</tr>
<tr>
<td></td>
<td>SumoLogic</td>
</tr>
<tr>
<td></td>
<td>Woopra</td>
</tr>
<tr>
<td></td>
<td>Google Analytics</td>
</tr>
<tr>
<td>Performance check</td>
<td>Cloudping</td>
</tr>
<tr>
<td></td>
<td>YSlow</td>
</tr>
</tbody>
</table>
Third-party Apps and Tools Integration with AWS

<table>
<thead>
<tr>
<th>Use</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Sync / Replication / Migration</td>
<td>LSYNCD</td>
</tr>
<tr>
<td></td>
<td>Tsunami UDP</td>
</tr>
<tr>
<td></td>
<td>Gluster FS</td>
</tr>
<tr>
<td></td>
<td>AppScale</td>
</tr>
<tr>
<td></td>
<td>ARQ (sync for Mac)</td>
</tr>
<tr>
<td>Search</td>
<td>DynamoDB Geo</td>
</tr>
<tr>
<td>Provisioning</td>
<td>Madeira Cloud</td>
</tr>
<tr>
<td>Deployment</td>
<td>Codeship</td>
</tr>
<tr>
<td></td>
<td>Wercker</td>
</tr>
<tr>
<td></td>
<td>Beanstalk Maven plugin</td>
</tr>
<tr>
<td>Fault Tolerance / High Availability</td>
<td>Netflix Simian Army</td>
</tr>
<tr>
<td>Spot Usage</td>
<td>Chaordic Tio Patinhas</td>
</tr>
</tbody>
</table>

So, you can use the tools as mentioned in the preceding table based on your requirements.

**Summary**

Third-party tools are not something that we need to be scared of or use as little as possible. These are pretty useful when used in a proper way. We began this chapter by learning about the preparation of the infrastructure before using the third party tools. Then, we discussed three such third party tools, which provide database, S3 access, and EC2 access. Finally, we saw a few of the best practices, mostly the security practices that make use of third party tools in efficient ways.