

Deploying, Scaling, and Running Grails on AWS and VPC

Learn the ropes of networking to maximize
your potential

June 2nd - 4th 2014
Copenhagen, Denmark



GRAILS

Groovy



gradle

Created by [Ryan Vanderwerf](#) / [@RyanVanderwerf](#)

About Me

Learn the ropes of networking to maximize your potential

- Chief Architect @ ReachForce
- Co-Chair Austin Groovy and Grails User Group
- Help maintain Grails Quartz plugin
- Maintain GVPS Plugin (Grails Video Pseudo Streamer)
- Maintain Struts-1 Plugin
- Submit pull requests for others when I can!

What We Will Cover

- VPC
- S3
- **ElasticWolf**
- sshoogr and gramazon
- **Grails Plugins**
- **Elastic Load Balancers**
- **Autoscaling**
- **Others misc tips**

VPC - Virtual Private Cloud

Routing

NAT

ACL Firewall

Suggested Subnets

Security Groups

DHCP Tips

Others misc tips

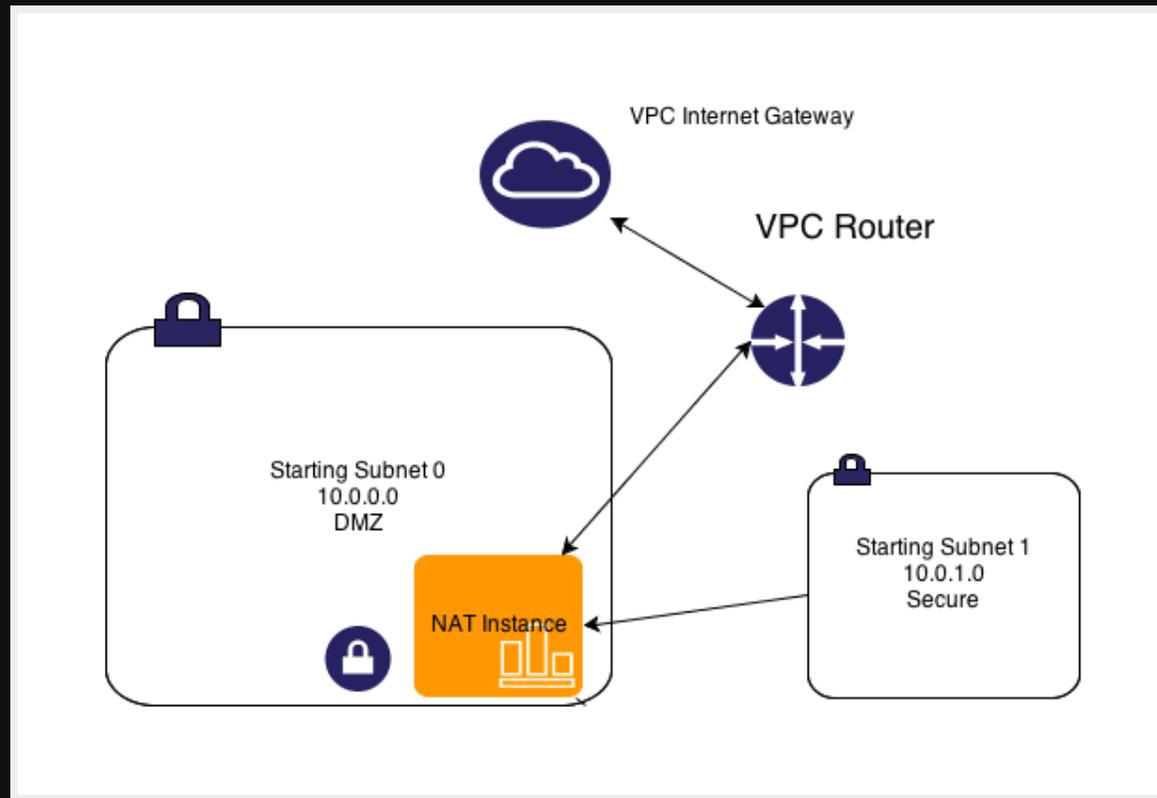
VPC Overview

Now required on new AWS accounts

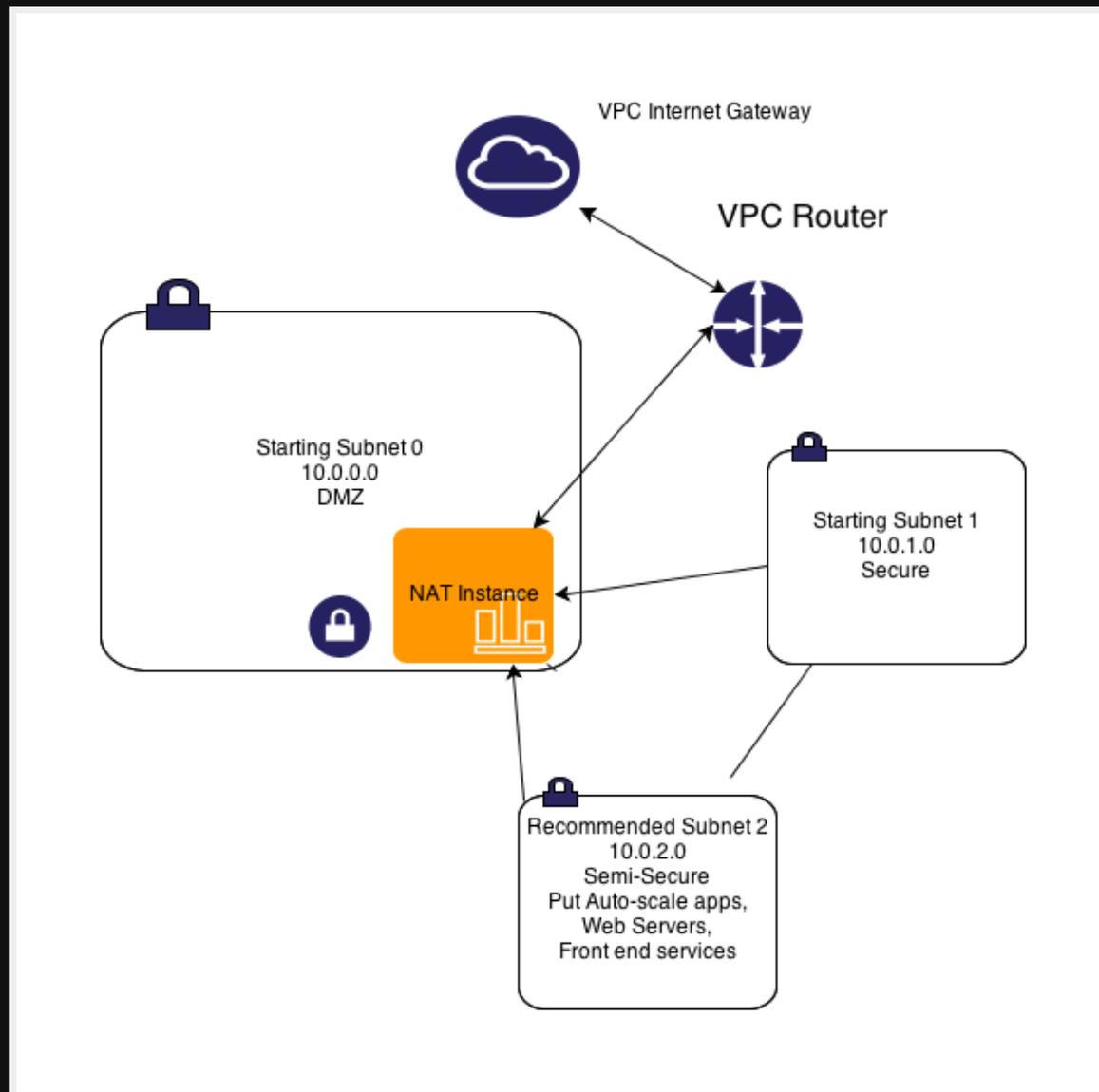
Very different than EC2 'Classic' which has no private network layer

Costs nothing extra - you have nothing to lose

VPC - Routing/Subnets



VPC - Routing/Subnets



VPC - Routing/Subnets

adding a 3rd subnet - this gives a helping hand

auto-scaling instances can access internet through NAT

Web Servers, Web facing apps can live behind ELB yet be able to run OS updates

S3 Buckets accessible

can access your secure subnet as well

VPC - Routing/Subnets

VPC - Routing/Subnets

The screenshot shows the AWS Management Console interface for the Subnets page. The top navigation bar includes 'Services', 'Edit', 'RF Research', 'N. Virginia', and 'Help'. The left sidebar lists various services under 'VIRTUAL PRIVATE CLOUDS', 'SECURITY', and 'VPN CONNECTIONS'. The main content area features a 'Create Subnet' and 'Delete' button, a 'Viewing: All Subnets' dropdown, and a table of subnets. Below the table, it indicates '0 Subnets selected' and prompts the user to 'Select a Subnet above'. The footer contains copyright information and a 'Feedback' button.

	Subnet ID	State	VPC ID	CIDR	Available IPs	Availability Zone	Route Table	Network ACL	Default Subnet
<input type="checkbox"/>	subnet-03373a6f	available	vpc-7790c01c	10.0.3.0/24	249	us-east-1a	rtb-b87c70d4	Default	false
<input type="checkbox"/>	subnet-4890c023	available	vpc-7790c01c	10.0.1.0/24	150	us-east-1a	rtb-7590c01e	Default	false
<input type="checkbox"/>	subnet-9e1b22f5	available	vpc-7790c01c	10.0.2.0/24	246	us-east-1a	rtb-eade6186	Default	false
<input type="checkbox"/>	subnet-4e90c025	available	vpc-7790c01c	10.0.0.0/24	230	us-east-1a	rtb-4390c028	Default	false

0 Subnets selected
Select a Subnet above

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#) [Feedback](#)

VPC - Routing/Subnets

The screenshot displays the AWS Management Console interface for a subnet. The top navigation bar includes 'Services', 'Edit', and user information 'RF Research', 'N. Virginia', and 'Help'. The left sidebar shows a navigation menu with categories like 'VIRTUAL PRIVATE CLOUDS', 'SECURITY', and 'VPN CONNECTIONS'. The main content area shows the 'Subnet' details for 'subnet-65573d0f'. It includes buttons for 'Create Subnet' and 'Delete', and tabs for 'Details' and 'Tags'. The details section shows the CIDR as 10.0.3.0/24, VPC as vpc-7790c01c, and Availability Zone as us-east-1a. Below this, the 'Route Table' is listed as 'rtb-b87c70d4 (replace)'. A table shows the route table entries:

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	i-cdab71b1

Below the route table, the 'Network ACL' is listed as 'Default (replace)'. Under 'Inbound:', a table shows the network ACL rules:

Rule #	Port (Service)	Protocol	Source	Allow/Deny
1	7	TCP	10.0.1.0/24	ALLOW
2	22 (SSH)	TCP	0.0.0.0/0	ALLOW
8	8080 (HTTP*)	TCP	0.0.0.0/0	ALLOW
100	ALL	ALL	0.0.0.0/0	ALLOW
101	ALL	TCP	5.5.0.0/24	ALLOW
102	16000	TCP	5.5.0.0/24	ALLOW

At the bottom of the console, there is a footer with copyright information: '© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved.' and links for 'Privacy Policy' and 'Terms of Use'. A 'Feedback' button is also present in the bottom right corner.

VPC - Routing/Subnets

The screenshot displays the AWS Management Console interface for VPC routing. The top navigation bar includes 'Services', 'Edit', 'RF Research', 'N. Virginia', and 'Help'. The left sidebar shows a navigation menu with categories like 'VIRTUAL PRIVATE CLOUDS' and 'SECURITY'. The main content area is titled 'VPC: All VPCs' and features buttons for 'Create Route Table' and 'Delete'. Below this, a 'Viewing:' dropdown is set to 'All Route Tables', and a pagination indicator shows '1 to 6 of 6 Items'. A table lists four route tables, with 'rtb-b87c70d4' selected. Below the table, a section titled '1 Route Table selected' shows the configuration for 'Route Table: rtb-b87c70d4'. This section includes tabs for 'Routes', 'Associations', 'Route Propagation', and 'Tags'. The 'Routes' tab is active, displaying a table with columns for 'Destination', 'Target', 'Status', 'Propagated', and 'Actions'. Two routes are listed: one for '10.0.0.0/16' with a 'local' target, and another for '0.0.0.0/0' with a target of 'eni-5894c433 / i-cdab71b1'. Both routes are 'active' and not 'Propagated'. The bottom of the page contains a copyright notice for Amazon Web Services, Inc. (2008-2013), links for 'Privacy Policy' and 'Terms of Use', and a 'Feedback' button.

Services Edit RF Research N. Virginia Help

VPC: All VPCs

Create Route Table Delete

Viewing: All Route Tables 1 to 6 of 6 Items

	Route Table ID	Associated With	Main	VPC
<input type="checkbox"/>	rtb-4390c028	1 Subnet	No	vpc-7790c01c (10.0.0.0/16)
<input checked="" type="checkbox"/>	rtb-b87c70d4	1 Subnet	No	vpc-7790c01c (10.0.0.0/16)
<input type="checkbox"/>	rtb-8d7724e6	0 Subnets	No	vpc-7790c01c (10.0.0.0/16)
<input type="checkbox"/>	rtb-911a23fa	0 Subnets	No	vpc-7790c01c (10.0.0.0/16)

1 Route Table selected

Route Table: rtb-b87c70d4

Routes Associations Route Propagation Tags

Destination	Target	Status	Propagated	Actions
10.0.0.0/16	local	active	No	Remove
0.0.0.0/0	eni-5894c433 / i-cdab71b1	active	No	Remove
	select a target			Add

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

VPC - Routing/Subnets

Security ACLs

ACLs like a full firewall - unlike groups

Protect subnets instead of EC2 instances

Processed in order of rule #

ACLs are stateless - responses to inbound bound by outbound rules

To accommodate various clients and OSs open ephemeral ports 1024-65535 then block malicious ports

VPC - Routing/Subnets

Security Groups

VPC Security Groups are different than EC2 Groups - use different and descriptive names if both used

You can use a security group ID (starting with sgXXXXX) in most places where IPs can be entered

Processed in order of rule #

ACLs are stateless - responses to inbound bound by outbound rules

To accommodate various clients and OSs open ephemeral ports 1024-65535 then block malicious ports

VPC - Routing/Subnets

DHCP Options

Determine What DNS DHCP clients use

Can do things like set default domain, whether to use internal or external DNS for your VPC

From Web UI, assign only 1 option set at a time

Using amazon provided DNS, using naming scheme of 10.0.1.x, default domain ec2.internal

Can assign your own DNS server names, or even Netbios name servers or NTP servers

Options are semicolon delimited name value pairs, i.e. 'domain-name=something.com; domain-name-severs=AmazonProvidedDNSs

VPC - IAM Roles

Don't use root account anymore. Traditional security page is retired
Set up your IAM roles for each user and use designated login url they give you
Use resource level security with IAM to tier access to instances/resources

VPC - Misc EC2 Tips

Reboot != Reboot be careful

Rebooting a machine from console will keep instance in place and ephemeral storage(OS reboot faster)

If you issue an 'stop' command you machine will move and lose ephemeral state

VPC - Misc EC2 Tips

VPN

Use OpenVPN instead of metered AWS VPN - it runs fine on a tiny instance- save \$\$

Use OpenVPN client to leave remote servers connected to VPC - it auto-reconnects

VPC - Misc EC2 Tips

Load Balancer

60s timeout on idle max - can file ticket for extension to 15+ minutes

Use Haproxy for anything very advanced, can accommodate many options

VPC - Misc EC2 Tips

EBS Based Instances

Use EBS backed instances for anything not build for cloud

If you can build server on the fly with Chef/Puppet like tools, go for ephemeral based

EBS backed instances have no swap by default - be sure to specify ephemeral disks on launch, use as swap

EBS backed instances have no ephemeral disks on by default be sure to use them on launch of instance and AMIs

S3 Storage

Limitations

Common tools

Regions with different functionality

Glacier

Grails S3 Plugin

S3 Storage

Limitations - Eventual Consistency

This means after write/update it will EVENTUALLY be consistent

Make your app retry on read fail - it might not be synced yet

Different regions have different consistency rules

US-West and EU Buckets have read after write consistency - but not update or delete (and cost more)

US-East is so large it cannot handle any kind of consistency after write/update/delete - except patience!

S3 Storage

Limitations - large files

Files over 5GB supported, but most tools don't handle properly
S3 Tools must support mime/multipart
s3cmd(Linux) / CyberDuck 4(Mac/Win32) / S3 Browser(Win32) /
Cloudberry Explorer(PRO Win32) / Bucket Explorer
File > 5GB files work with these tools, it is EXTREMELY slow

S3 Storage

Tools

s3fs - mount as filesystem - but >5GB files broken, beware of consistency!

Make sure FUSE is in kernel

s3cmd is best free command line tools

Bucket Explorer & CloudBerry Backup are good solid windows clients that parallelize multi-part uploads to ease the pain

s3 Browser is ok free tool

For install on Ubuntu: <http://zentraal.com/docs/installing-s3fs-on-ubuntu/>

S3 Storage

Different Function Between Different Regions

US-East-1 Cheapest for full redundancy otherwise to save \$ you can use RRS (Or Glacier)

US-West and EU Buckets have read after write consistency - but not update or delete (and cost more)

US-East is so large it cannot handle any kind of consistency after write/update/delete - except patience!

Barring these limitations (and budget!), use the region closed to your VPC instances and regions

S3 Storage

Grails S3 Plugin

Looks unmaintained, but still works fine on latest Grails versions (no JIRA bugs pending!)

Delete Buckets (See [org.grails.s3.BucketService](#))

Uploads and catalogs assets (will use bucket name you give as base for its bucket name)

Names files inside bucket with UUIDs to avoid collisions

Can give each asset a bucket and key pair or global

Demo - s3-demo project

General AWS Grails Plugins

Grails AWS Plugin

AWS SDK Plugin

DynamoDB GORM Plugin - not covered

Amazon Flexible Payments - not covered

SimpleDB GORM Plugin - not covered

CDN Asset Pipeline plugin - not covered

General AWS Grails Plugins - Grails AWS Plugin

**Actual Groovy/Grails Code to Manage S3
storage and SES Service (vs Java wrapper)**

General AWS Grails Plugins - Grails AWS Plugin

**Has handy Gant scripts installed - used for
SES stats**

Aws-Ses-Get-Send-Quota

Aws-Sws-Get-Send-Statistics

Aws-Ses-List-Verified-Emails

Aws-Ses-Send-Ping-Mail

Aws-Ses-Verify-Email

General AWS Grails Plugins - Grails AWS Plugin

Aws-Ses-Get-Send-Quota

Gets your current Quota for Simple Email Service
Shows email limit per day, per second, # of emails
Output looks like this:

[AWS SES] The maximum number of emails the user is allowed to send in a 24-hour interval: 10000.0

[AWS SES] The maximum number of emails the user is allowed to send per second: 5.0

[AWS SES] The number of emails sent during the previous 24 hours: 15.0

General AWS Grails Plugins - Grails AWS Plugin

Aws-Sws-Get-Send-Statistics

Gets your email sending statistics

The output is a list of items, for the last two weeks of sending activity.

Example output:

```
[AWS SES] -----  
[AWS SES] | time range | attemps | rejects (SES) | complaints  
(recipient) | bounces |  
[AWS SES] |-----|  
[AWS SES] | 2013/06/26 22:42 | 1 | 0 | 0 | 0 |
```

General AWS Grails Plugins - Grails AWS Plugin

aws-ses-list-verified-emails

Shows a list of all verified emails with

The output is a list of emails authorized to send email FROM

Example output:

```
[AWS SES] 1) support@reachforce.com
```

General AWS Grails Plugins - Grails AWS Plugin

aws-ses-send-ping-mail

Script will verify a given email has been 'verified' by Amazon

General AWS Grails Plugins - Grails AWS Plugin

aws-ses-verify-email

Script will submit an email for verification to amazon, and send an email to that address

General AWS Grails Plugins - Grails AWS Plugin

aws-ses-verify-email

Script will submit an email for verification to amazon, and send an email to that address

General AWS Grails Plugins - Grails AWS Plugin S3 File Management

Demo

General AWS Grails Plugins - Grails AWS Plugin

S3 File Management

Handy for storing existing assets into S3

Convert local File to S3 Storage

```
def s3file = new File("/tmp/test.txt").s3upload { path  
"folder/to/my/file/" }
```

Upload directly from Stream:

```
def file = request.getFile('photo') def uploadedFile =  
file.inputStream.s3upload(file.originalFilename) { bucket "file-upload-  
from-inputstream" }
```

General AWS Grails Plugins - Grails AWS Plugin

SES Management

AFAIK skip this - the mail plugin handles all of this

```
// settings for mail plugin to work with SES grails { mail { host = "email-  
smtp.us-east-1.amazonaws.com" port = 25 username =  
"SESUsername" password = "SESPassword" props =  
["mail.smtp.starttls.enable":"true",  
"mail.smtp.port":"587","mail.debug":"true","mail.smtp.auth":"true"]} //  
uncomment to force all emails to one address //  
grails.mail.overrideAddress="test@address.com"  
grails.mail.default.from = "support@reachforce.com" } }
```

General AWS Grails Plugins - AWS SDK Plugin

Wrapper for AWS Java SDK

This is the one to use for most powerful work of services - can access almost all of AWS services that JDK supports

Uses AWS Web Service API wrapped in a Java Library

Error handling can be more difficult than using web service api directly

General AWS Grails Plugins - AWS SDK Plugin

Services Supported

EC2 - ELB, CloudWatch, Elastic Beanstalk

RDS - Elastic Mapreduce, SDB, Dynamo, Redshift, SimpleDB

S3/Glacier

SES (us-east-1, us-west-2, eu-west-1 only)

SQS, SWF

CloudFormation, CloudFront

Elastic Beanstalk, Transcoder, Opsworks (Old Chef)

Cloudformation, CloudSearch, ElastiCache

General AWS Grails Plugins - AWS SDK Plugin

EC2

Grails service wrapper is 'amazonWebService'

Call `amazonWebService.ec2`. - for default region, call
`amazonWebService.getEc2('region')`. for others

`RunInstancesRequest` class defines parameters to launch an instance
(It seems picky about nulls)

use `amazonWebService.runInstances()` to start instances

Complete docs at:

<http://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/>

General AWS Grails Plugins - AWS SDK Plugin

RDS

Not sure why you would spin up entire databases, but you can!
Most practical use would be to automate snapshots of the database
If you have a small simple database, service is good
Cannot start/top RDS instances, and pay premium to use their
licenses if using commercial DBs
If you have the skill or have large DB requirements, set up your own
database - there is probably an AMI for it
Has good multi-az failover (but pricey!) with PIOPS EBS Volumes (You
can raid your own PIOPS drives!)

General AWS Grails Plugins - AWS SDK Plugin

S3

manage buckets, upload/download and delete files

Transfer Manager - handles multipart uploads (big files, batch, fast!)

full docs here:

[http://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/com/amazonaws/
summary.html](http://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/com/amazonaws/summary.html)

General AWS Grails Plugins - AWS SDK Plugin

Glacier

cold storage for data - not immediately available on demand
vaults - organize your archives and policies (where your files go)

Upload files via

`amazonWebService.glacier.uploadArchive(UploadArchiveRequest
uploadArchiveRequest)` or `UploadMultiPart`

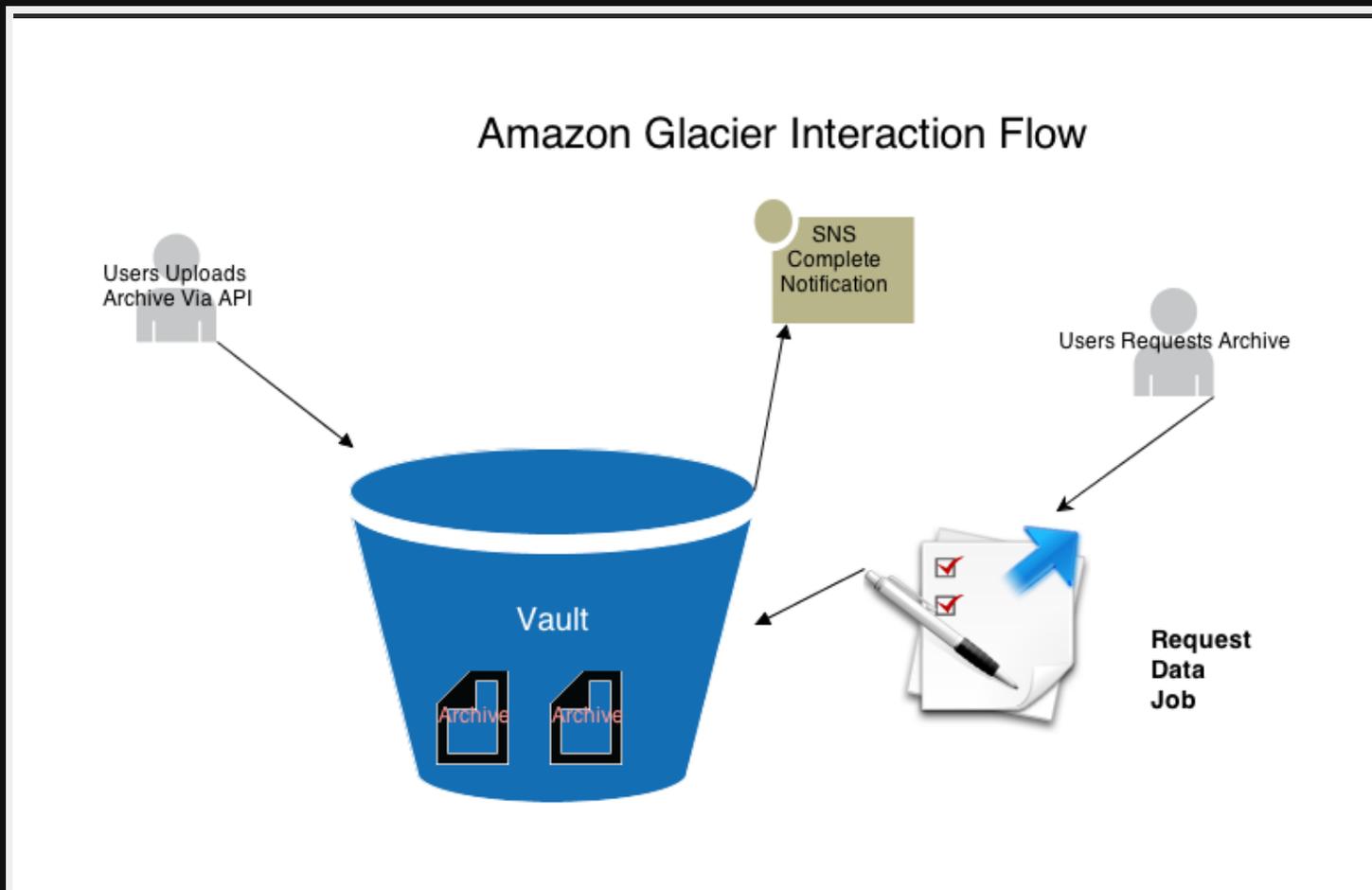
Files are retrieved via 'initiateJob'

Call 'listJobs' until your job is done.

Can use Amazon SNS to notify you when jobs are complete

When done get files from your vault via `GetJobOutput`

General AWS Grails Plugins - AWS SDK Plugin Glacier



General AWS Grails Plugins - AWS SDK Plugin

SQS

Simple queue service

Create / Delete Queues

Receive Message

Can also assign permissions and batch messages

Messages can remain in queue for 12 hours

ElasticWolf

More powerful than web console gui

Can control things like auto-scaling

Created partly for new GovCloud because they had no UI at all initially

Managed by sales organization instead of IT

Actively Maintained on Github

ElasticWolf

Demo

sshooogr and gramazon

sshooogr - Groovy-based DSL library for working with remote servers through SSH - DSL Allows:

- connecting
- executing remote commands
- copying files and directories
- creating tunnels in a simple and concise way
- gradle plugin also for
- project very active (last commit 1 month ago)

gramazon - Groovy based API for Amazon EC2

- interface library that can be used to interact with the Amazon EC2 system and control server resources on demand from your Groovy scripts or from Gradle, using a plug-in.>
- uses gradle to run commands
- maybe could be used as basis for a groovy version of chef or puppet?
- project very active (last commit 1 month ago)
- gradle project template available as well as 'gradle-ssh-plugin'

Repos for these: <https://github.com/aestasisit/sshooogr> and <https://github.com/aestasisit/gramazon-gradle>

Autoscaling Overview

Autoscaling can be used for fault-tolerance (min 1)

Most efficient user of instance

Set Policies using templates for how more servers created/terminated

Control Spot instance bit price

You app must be able to handle 'sudden death'

Make sure you debug your AMI BEFORE adding to a autoscale group
(debugging failed autoscaled instances is no way to go through life
son!)

Autoscaling Setup

Step 1: Create Launch Config(Just like launching instance but a template)

Step 2: Create Autoscale Group

Step 3: Create Policies

Demo with ElasticWolf (can use web console now too)

Elastic Beanstalk

Finally a good solution to push your application to Elastic Beanstalk on Grails!

Ken Liu now has first class citizen support in Grails for Elastic Beanstalk!

Easy to set up and configure, just add keys and params to Config.groovy

Use command 'aws-eb-deploy' and 'aws-eb-stop' - that's it!

Detailed online manual available at <http://kenliu.net/grails-elastic-beanstalk/manual.html>

Useful resources

Elastic Beanstalk plugin by Ken Liu: <http://grails.org/plugin/aws-elastic-beanstalk>

Elastic Beanstalk Intro @ Bobby Warner's Blog:

<http://www.bobbywarner.com/2011/10/14/grails-on-aws/>

Another Beanstalk:

<http://malderhout.wordpress.com/2011/02/18/deploy-grails-apps-in-3-simple-steps-to-amazon-beanstalk/>

Sample Beanstalk app: <https://github.com/4np/grailsOnAWS>

Oracle, EBS, and other Tips from AWS Architect Tom Laszewski

<http://cloudconclave.blogspot.com/>

Building an S3 Browser in Grails

<http://aws.amazon.com/articles/Amazon-S3/4000>

