Bringing Your First Application Into The Cloud

Julien Simon, Principal Technical Evangelist

12/04/2016

Create a Git repository with AWS CodeCommit

```
$ aws codecommit create-repository \
--repository-name blog --region us-east-1 \
--repository-description "ElasticBeanstalk
demo” \

$ git clone ssh://git-codecommit.us-
east-1.amazonaws.com/v1/repos/blog
```

Create a new Rails application

```
$ rails new blog

$ cd blog

$ git add .

$ git commit -m "Initial version"
```

Create a new Rails application

```
$ rails new blog

$ cd blog

$ git add .

$ git commit -m "Initial version"
```

Add a ‘post’ resource to the application

```
$ rails generate scaffold post title:string body:text

$ bundle exec rake db:migrate

$ git add .

$ git commit -m "Add post resource"

$ rails server

$ open http://localhost:3000/posts
```

Initialize a Ruby application in Elastic Beanstalk

```
$ eb init blog -p Ruby -r eu-west-1

$ git add .gitignore

$ git commit -m "Ignore .elasticbeantalk
directory"
```

Create a ‘blog-dev’ environment

Single instance (no auto-scaling, no load balancing),
t2.micro instance size (default value)

```
$ eb create blog-dev
--single
--keyname aws-eb
--envvars SECRET_KEY_BASE=`rake secret`
```

Update the code and redeploy on ‘blog-dev’

```
$ vi app/views/posts/index.html.erb
--> "Blog de dev"

$ git add app/views/posts/index.html.erb
$ eb deploy blog-dev --staged

$ git commit -m "Add message on post page"
$ eb deploy blog-dev
```

Create a production branch for the blog

```
$ git branch prod
$ git checkout prod
```

Now we have to modify 3 files to add support for Postgres:

`Gemfile`
`config/database.yml`
`.ebextensions/packages.config`

Gemfile

```
group :development, :test do
  # Use sqlite3 as the database for Active Record
  gem 'sqlite3’
end

group :production do
    # Use PostgreSQL as the database for Active Record
    gem 'pg', '~> 0.18.1’
end
```

config/database.yml

```
production:
    <<: *default
    adapter: postgresql
    encoding: unicode
    database: <%= ENV['RDS_DB_NAME'] %>
    username: <%= ENV['RDS_USERNAME'] %>
    password: <%= ENV['RDS_PASSWORD'] %>
    host: <%= ENV['RDS_HOSTNAME'] %>
    port: <%= ENV['RDS_PORT'] %>
```

These environment variables will be automatically declared by Elastic Beanstalk when we create an RDS instance

.ebextensions/packages.config

```
packages:
    yum:
      postgresql94-devel: []
```

This directive will install the postgres94-devel package on your instances.
It is required to install the ‘pg’ Ruby gem.

.ebextensions provides lots of options to configure and customize your Elastic Beansltalk applications. The documentation is your friend ☺

https://docs.aws.amazon.com/fr_fr/elasticbeanstalk/latest/dg/ebextensions.html

Commit these changes to the production branch

```
$ git add Gemfile config/database.yml
.ebextensions

$ git commit -m "Use Postgres for production”

$ git push
```

Now let’s create a proper production environment :
running in a VPC, auto-scaled, load-balanced,
with larger instances and backed by RDS Postgres.

Ready? ☺

Create a ‘blog-prod’ environment

```
$ aws ec2 describe-subnets --query "Subnets[].SubnetId"
$ export VPC_SUBNETS=subnet-63715206,subnet-cbf5bdbc,subnet-59395b00

$ eb create blog-prod -k aws-eb
--vpc.id=vpc-def884bb --vpc.elbpublic --vpc.publicip
--vpc.elbsubnets $VPC_SUBNETS
--vpc.ec2subnets $VPC_SUBNETS
--vpc.dbsubnets $VPC_SUBNETS
--instance_type m4.large
--database.engine postgres --database.version 9.4.5
--database.instance db.m4.large --database.size 5
--database.username YOUR_USERNAME --database.password YOUR_PASSWORD
--envvars SECRET_KEY_BASE=`rake secret`
```

Connect on ‘blog-prod’ instances

```
$ ssh -i ~/.ssh/aws-eb ec2-user@IP_ADDRESS

$ psql -h RDS_INSTANCE -p 5432 -U YOUR_USERNAME -d ebdb

Describe tables: \dt
Show posts: select * from posts
```

Terminate environments

```
$ eb terminate blog-dev --force

$ eb terminate blog-prod --force
```


AWS
SUMMIT