

Resin Cookbook - Article: **Amazon EC2 Deployment**

Recently many deployments have been taking advantage of hosted cloud site deployments. While Resin can be used as it's own private, public, or hybrid cloud for hosting web applications, some users choose to deploy Resin in a cloud hosted environment for many reasons, among them:

- cost effectiveness
- unlimited store space
- hardware scalability
- device diversity
- security
- agility

This tutorial will cover deploying to the popular Amazon EC2 instances. If you are not familiar with Amazon, a brief tutorial can be found at [Amazon WS](#). You will need an Amazon WS account. Amazon WS allows you to have free tier so you can learn Amazon WS (EC2, S3, Elastic Load Balancer, Block Storage, SimpleDB, Simple Queue Service, Simple Notification Service).

Create an Elastic IP and assign it to your Amazon Linux instance

Elastic IP Amazon EC2 instances are assigned two IP addresses at launch: a private (RFC 1918) address and a public address. The public address is mapped to the private IP address through Network Address Translation (NAT). Every time you start or restart an instance, it gets new public and private addresses.

Amazon EC2 provides a service called Elastic IP addresses. Elastic IP addresses are static IP addresses. Elastic IP addresses are associated with your account, not a specific instances. You can associate an Elastic IP address with an instance. It is good to associate your instance with an IP address so you can direct your calls to that instance when deploying an application.

If you are not familiar with Elastic IPs, read the Amazon documentation on Elastic IP Elastic IP, just scroll down a bit and you can see how to create an Elastic IP from the Amazon WS Web Console.

Step 1: Elastic IP

Create an Elastic IP and assign it to the Amazon WS Linux instance using the Amazon web console.

Example IP:

23.21.195.83

In order to use it, you must assign it to our instance that has Resin. In the console for managing Elastic IP address is an option under actions to assign the Elastic IP address to our instance.

Warning Remember after you restart the server, you need to reassign it the Amazon Elastic IP. This means go to the console. Find the list of Elastic IPs. Select the one you want. Click the Associate Address button and pick the instance ID of the server that has Resin. Every time an instance restarts it forgets its IP address unless you reassign it.

Download & Install Resin

The most recent version of Resin is available to be downloaded directly from the caucho site:

```
> wget "https://caucho.com/download/resin-pro-4.0.57.tar.gz"
```

After downloading, install Resin as if you were on your local machine.

Create password for your Amazon Cloud instance

```
> resinctl generate-password -user admin -password mypassword
```

Output:

```
admin_user : admin
```

```
admin_password : {SSHA}somerealpasswordNv3IzMO0G1qNmzk3J
```

Create a properties file and put it into user-data of Amazon AMI

Create a file that looks like this:

```
https      : 8443
admin_user : admin
admin_password : {SSHA}UaNyLiyzU1kepY7OSayrtdVGZf5/XJCE
web_admin_enable : true
remote_cli_enable : true
web_admin_external : true
```

To put the above in the user-data of that instance, you typically need to stop the instance from the Amazon EC2 console or command line. Then from the Amazon Management Console, select the instance in the instances list, then select the action "View/Change User Data" and add the contents above.

Reminder: Associate with Elastic IP after restarting instance Remember after you restart the server, you need to reassign it the Amazon Elastic IP. This means go to the console. Find the list of Elastic IPs. Select the one you want. Click the Associate Address button and pick the instance ID of the server that has Resin.

After you restart the Amazon instance, look at its security group and make sure the following inbound ports are open up.

InBound Port (Scheme) Source

22 (SSH)	0.0.0.0/0
80 (HTTP)	0.0.0.0/0
443 (HTTPS)	0.0.0.0/0
8080 (HTTP*)	0.0.0.0/0
8443 (HTTPS*)	0.0.0.0/0
6800-6899	sg-47yourSGGroupId

When you setup a security in Amazon WS, it is given a name. If you are later going to turn on clustering, you need to open up ports 6800-6899 to your own security group. Replace **sg-47yourSGGroupId** with you actual security group id.

With Resin 4.0.27 and above you can configure additional config files like mycustom.xml and drop them under /etc/resin/resin-inf/mycustom.xml, and these XML files get picked up by Resin and become part of the Resin configuration. This was added to make configuration simpler for cloud deployments.

The actual directive that looks for these additional xml files is in /etc/resin/cluster-default.xml which gets included by /etc/resin/resin.xml.

```
/etc/resin/cluster-default.xml
```

```
<resin:import fileset="${__DIR__}/resin-inf/**/*.*xml"/>
```

The above is why you can define xml files into /etc/resin/resin-inf/ and they get picked up by Resin.

ec2.xml file

Create the file that is listed below called ec2.xml and put it into /etc/resin/resin-inf/ec2.xml on the Amazon instance.

```
<resin xmlns="http://caucho.com/ns/resin"
  xmlns:resin="urn:java:com.caucho.resin">
  <resin:properties path="http://23.21.195.83/latest/user-data"
    optional="true" recover="true"/>
</resin>
```

The IP address (http://23.21.195.83) is a meta address that KVM, and Amazon EC2 use to configure per instance data about the instance you are running. The above is configured to read properties out of this user data location.

Log into Amazon AMI instance and add the above to /etc/resin/resin-inf/ec2.xml.

```
> ssh -i resin2.pem ec2-user@23.21.195.83
> sudo mkdir /etc/resin/resin-inf
> sudo nano /etc/resin/resin-inf/ec2.xml
```

XML files that you add to `/etc/resin/resin-inf/` get added to the resin app cluster configuration configuration. This allows you add just the additional config you need without changing the rest on the resin configuration.

Remember: the Amazon security group that this instance needs has to open up the following ports 8080, and 8443, 80, 443, & Resin's internal cluster communication port 6800-6899.

To test out the changes, from your local development machine execute this command:

```
> resinctl deploy --address 23.21.195.83 --port 8080 --user  
admin --password mypassword hello.war
```

You can see if the deploy worked by do the following:

```
> resinctl deploy-list --address 23.21.195.83 --port 8080 --user admin --password roofoo  
production/webapp/default/blog
```