



# How to Build GazApps

Prepared by:

Brad Starkie,  
Chief Scientist  
Gazunti Pty Ltd.

Revision:

Draft 1

Implementation Date: 11 March 2015

***Unless otherwise marked, this is an Uncontrolled Copy***

© Gazunti Pty. Ltd. 2013. All rights reserved.

Use of this publication is permitted by Gazunti Pty. Ltd. only on the condition that:

1. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise;
2. No part of this publication may be used for any commercial purpose; and
3. No part of this publication may be given to any entity other than the entity to which Gazunti Pty. Ltd. or its authorised representative has directly provided this publication without the prior written permission of Gazunti Pty. Ltd..

## How to build GazApps

---

### Document Control Sheet

#### Contact for Enquiries and Proposed Changes

If you have any questions or suggestions regarding this document, or if you require any assistance with implementing the procedures contained within this document, please contact:

Name: Brad Starkie  
Designation: Chief Scientist  
Phone: (+61) 409 861 861

### Revision History

<b>Issue No</b>	<b>Issue Date</b>	<b>Nature of Amendment</b>
1	28/2/2013	Working Draft as part of Workplan
2	24/7/2013	Updated version as sent to UCLA
3	11/03/2015	Updated Version based upon AWS server image
4	20/08/15	GazCloud v 3.2

## Table of Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>5</b>
1.1	Purpose and scope of this Document .....	5
1.2	What is A GazApp and What can it Do? .....	5
1.3	Motivation for the GazWeb Product .....	5
1.4	The AWS Server Image .....	6
<b>2</b>	<b>BLOCK DIAGRAMS.....</b>	<b>7</b>
<b>3</b>	<b>QUICK START.....</b>	<b>8</b>
3.1	Prerequisites.....	8
3.2	Creating an Amazon Account.....	8
3.3	Creating An Amazon Server Instance .....	9
3.4	Logging into your Server and Changing Passwords.....	12
3.5	Crawling a Web Site.....	14
3.6	Using the Chatbot.....	17
3.7	Using the Android GazClient.....	19
<b>4</b>	<b>ADVANCED APPLICATION DEVELOPMENT .....</b>	<b>22</b>
4.1	Using NoMachine and Eclipse .....	22
4.2	Java Properties File.....	24
4.3	SQL Database .....	27
4.4	Crawler .....	29
4.5	Default Prompts and Answers.....	29
4.5.1	Default Prompts.....	29
4.5.2	Frequently Asked Questions.....	30
4.6	String Rewriting Rules.....	30
4.7	Noun Phrase Classifier Module.....	32
4.8	Parser Module .....	33
4.9	Modifying The Java Code.....	35
4.9.1	Adding New question Types .....	35

# 1 INTRODUCTION

## 1.1 PURPOSE AND SCOPE OF THIS DOCUMENT

The purpose of this document is to enable a developer to build a conversational application (a GazApp) on their own instance of the Gazunti Hosting platform (The GazCloud server). To do this the developer creates an Amazon web Server account and then creates an instance of the GazCloud. You will then have your own server in the cloud running the Gazunti software.

## 1.2 WHAT IS A GAZAPP AND WHAT CAN IT DO?

A GazApp is a conversational application that runs on the Gazunti Hosting platform (The GazCloud server). A GazApp uses speech or text on a variety of devices including smart phones and web browsers. GazApps can answer natural language questions like

- Do you have a store in Camberwell?
- How long will delivery take?
- What are my rights as a creditor when a company is in external administration?

GazApps can perform actions, and link into your existing web forms responding to sentences like

- I want to purchase that now.
- Is the trading name "Pets are us" available?

A GazApp can be described as "knowledge navigator" or "virtual employee" that enables your customers to access your information and services.

When a user is interacting with a GazApp they can be transferred to another GazApp on another server. This is similar to the Hyperlink on the world web, and on VoiceXML applications. In the next version of the GazCloud GazApps will be able to hand over to either

- Another GazApp,
- A VoiceXML application,
- A web page or
- A person or system at the end of a telephone

## 1.3 MOTIVATION FOR THE GAZWEB PRODUCT

The old way of surfing the web is becoming outdated. Most people are using their smart phones, devices are getting smaller, speech recognition is getting better. There are exciting new devices like Google glasses, Dash and Smart Watches, that will change the way in we get access to information and services. But when we use the world wide we're still expected to type in or say some keywords into search engine, select some links and then read through a page.

We've already seen the first wave of Artificial Intelligence (AI) Personal Assistants (PAs) on smart phones. The technology is only going to get better. One problem with the existing PAs is that they are a single app. They are expected to do everything.

· That's bad for the end user, because it limits what your AI can do. If you want to do something one off it's probably just going to pass it to Google, then you're going to have to select from a list, and then interact via the web. That's hard to do if you're hands free.

· It's bad for developers because you're locked out. There are hundreds of millions of developers that aren't making the end users experience better. Developers are reliant on a small number of players agreeing to add their functionality to the end users experience. To be fair it's hard for the PA manufacturers to allow millions of the developers to add work their work into the PAs product. It's too hard to manage even if they wanted to.

· It's bad news for the owners of content and services. They can't improve the PA's functionality by allowing it to access their content and services.

The answer is to add AI to the World Wide Web. Every site can then form part of what you can do on your device. Billions of sites. Tens of Millions of developers.

### 1.4 THE AWS SERVER IMAGE

To use the GazCloud you need to create your own Amazon Web Server account. You then create an instance of the GazCloud server which contains the following

- Working versions of the following GazApps
  - GazuntiServer a GazApp that crawls a website and allows users to ask questions contained on the website
  - GazuntiAdmin A GazApp that allows users to manage the GazuntiServer GazApp
  - A crawler app which crawls your website and populates your local database
- A working version of the Eclipse SDK, and source code for the 3 applications. Developers can extend the capabilities of these applications by adding additional Java code. Access to the Eclipse SDK is via the NoMachine Server.
- A MySQL database

The use of an AWS server image has the following benefits for the developer

- It is a turnkey solution in which all of the components are functioning correctly
- The developer has complete freedom to customise the server in any way they like

## 2 BLOCK DIAGRAMS

Figure 1 Shows the GazApp architecture. There are two parts to a GazApp running on a smart device.

- The thin Client which runs on your device
- The GazApp chat bot which run on the GazCloud server

The two components interact by sending JSON encoded messages using http.

Figure 2 shows the architecture of the GazApp itself.

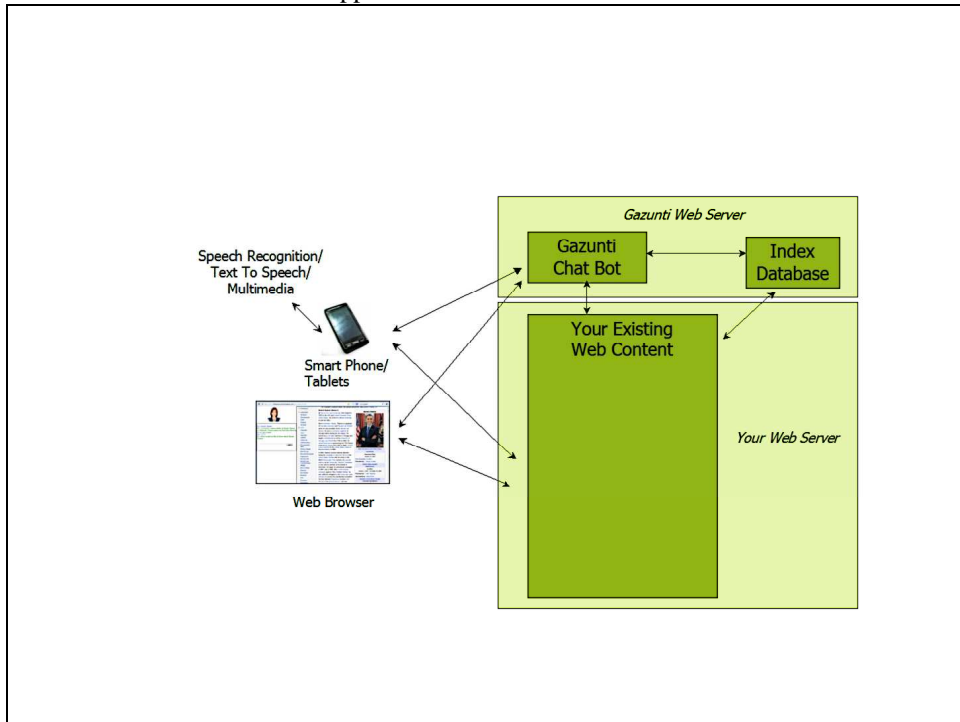


Figure 1 The GazApp Architecture

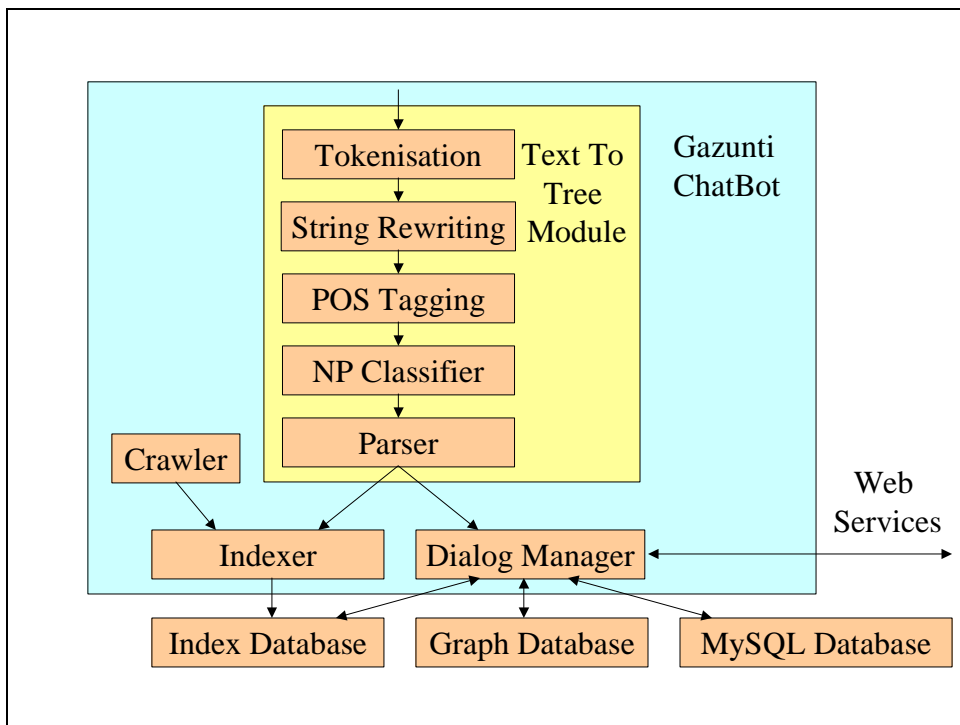


Figure 2 The ChatBot Architecture

### 3 QUICK START

To build your GazApp all you need to do is

- Provide a set of URLs for your GazApp to trawl.
- Provide a set of predefined prompts

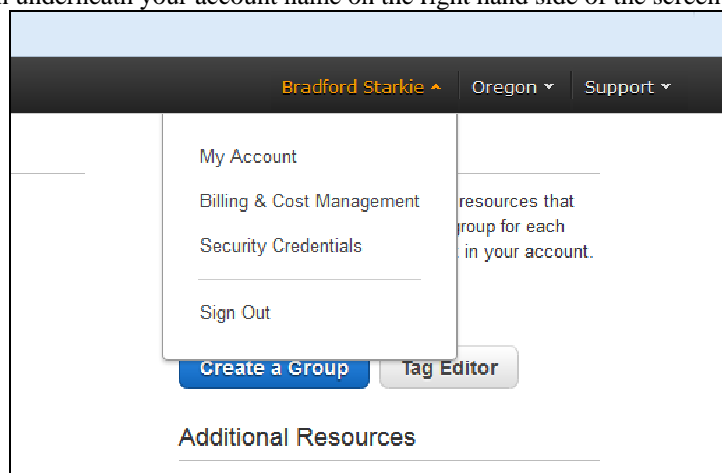
#### 3.1 PREREQUISITES

For GazApp to crawl your website, the following prerequisites need to apply

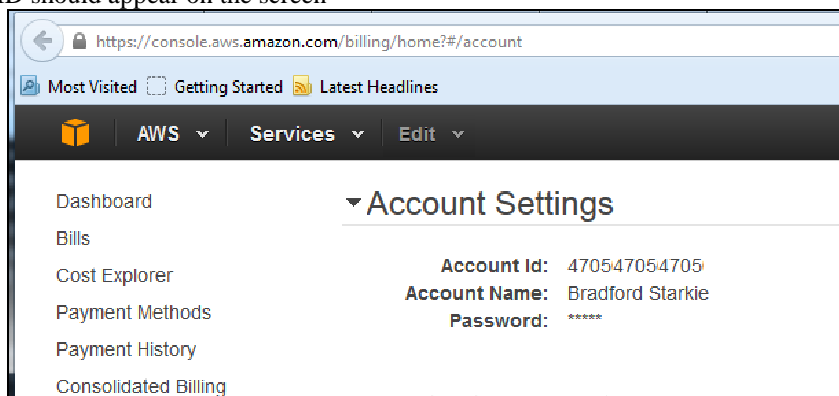
1. The crawler needs to be able to crawl your website, which involves downloading a large number of pages from it in succession. Some websites such as Facebook do not allow this and may have crawl detection scripts that prevent Gazunti from crawling it.
2. For the pages to be viewed using the chatbot the target site needs to allow pages to be viewed within an iframe. This will not be possible if the target uses Clickjacking prevention methods such as X-Frame-Options. (See <http://en.wikipedia.org/wiki/Clickjacking#X-Frame-Options>)
3. Not all of the text on the web pages that you crawl will be valid English sentences. Gazunti reads the text on websites, and where possible interprets them as English sentences.
4. You need an Amazon Web Services account. This will not cost you anything, but depending upon your application you may need to run a sufficiently large servers that may incur fees from Amazon Web Servers.

#### 3.2 CREATING AN AMAZON ACCOUNT

1. Open <http://aws.amazon.com/>, and click Sign Up.
2. Once you have created your account, log in and locate your Amazon Account Id by selecting "My Account" from the drop down underneath your account name on the right hand side of the screen on the AWS console



3. Your account ID should appear on the screen

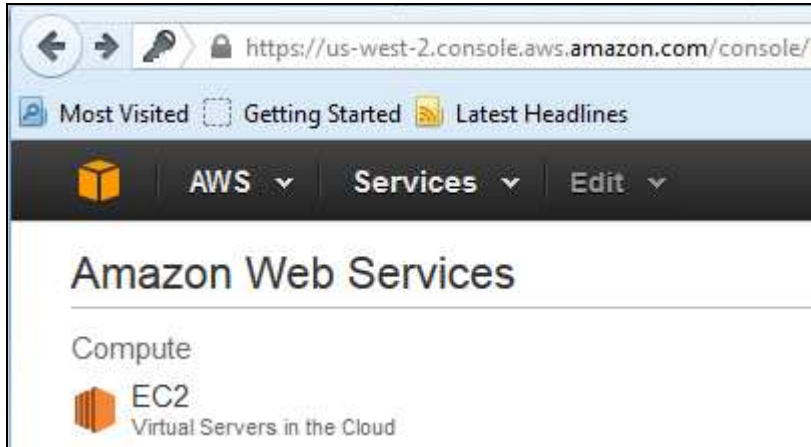


4. Email your account ID to [brad.starkie@gazunti.com](mailto:brad.starkie@gazunti.com) and request access to the server image. We will email you when we have granted you permission to use the server image.

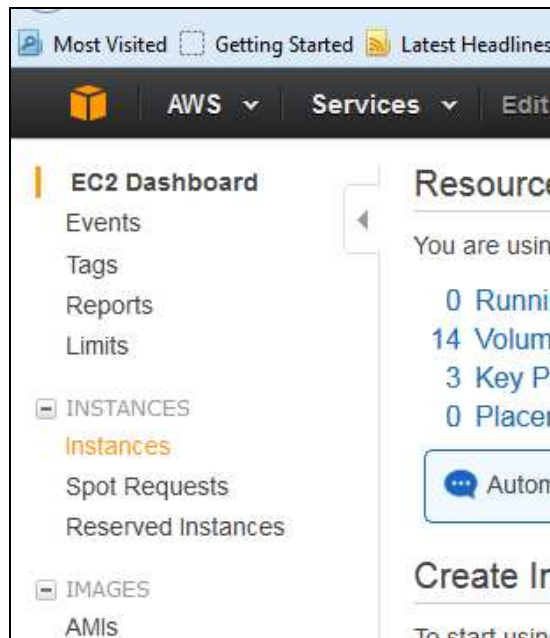


### 3.3 CREATING AN AMAZON SERVER INSTANCE

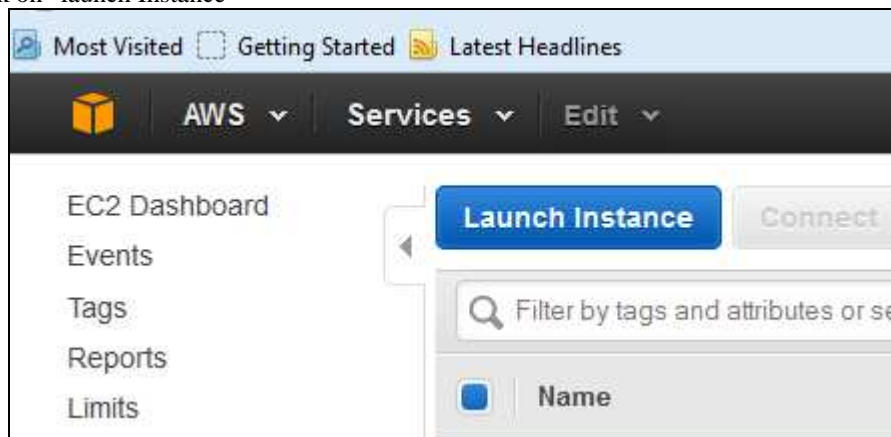
1. Once you have received your email confirming that you have been granted permission to use the server image Click on "EC2 Virtual Servers in the Cloud" as shown below



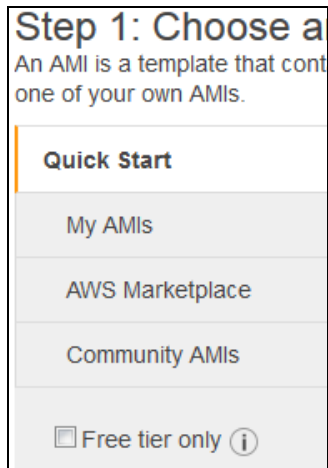
2. Click on Instances



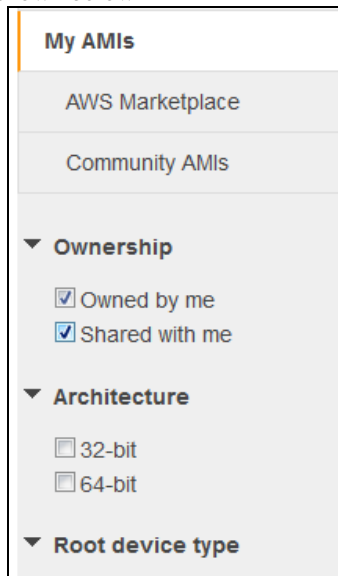
3. Click on "launch Instance"



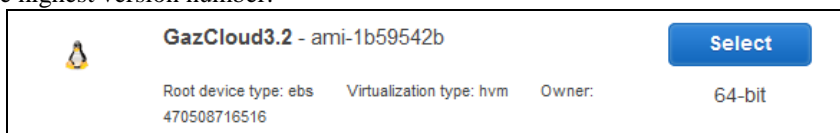
4. Click on my AMIs.



5. Click on "Shared with me" as shown below



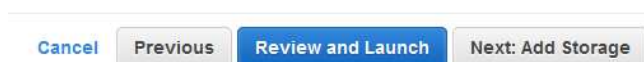
6. After clicking "Shared with me" the server image labelled "GazCloud3.2" will appear in the right hand side of the window pane as shown below. If there are more than one GazCloud server images, select the version with the highest version number.



7. Click "Select" and select the instance type that you want to use. We recommend the use of the M3.medium instance type. For a description of the different AWS instance types see <https://aws.amazon.com/ec2/instance-types/> and for a description of the pricing of the different AWS instance types see <https://aws.amazon.com/ec2/pricing/>. Then click "Configure Instance Details".

<input type="checkbox"/>	General purpose	m4.xlarge	40	160	EBS only	Yes	10 Gigabit
<input checked="" type="checkbox"/>	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate
<input type="checkbox"/>	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate
<input type="checkbox"/>	General purpose	m3.xlarge	4	15	1 x 16 (SSD)	Yes	High

8. Typically you will not need to modify any of the defaults on the next screen. Click "Add Storage".



## How to build GazApps

- On the next page you will need to add the 30 Giga Byte storage. If you do not add the 30 Gigabyte disk your server image will be missing software. If you increase the size of the additional storage to be greater than 30 GigaBytes, it will not be used until you expand the volume on the disk. Next click "Tag Instance".

The screenshot shows the AWS Management Console interface for 'Step 5: Tag Instance'. At the top, there are navigation tabs for '1. Choose AMI', '2. Choose Instance Type', '3. Configure Instance', '4. Add Storage', '5. Tag Instance', '6. Configure Security Group', and '7. Review'. Below the tabs, the title 'Step 5: Tag Instance' is followed by a brief explanation: 'A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.' The main form has two input fields: 'Key (127 characters maximum)' with the value 'Name' and 'Value (255 characters maximum)' with the value 'AgServer'. Below these fields is a 'Create Tag' button and a note '(Up to 10 tags maximum)'. At the bottom of the form are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Security Group'. The footer contains 'Feedback', 'English', copyright information '© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.', and links for 'Privacy Policy' and 'Terms of Use'.

- If you like you can give the server image name such as AgServer shown above, but this is not necessary. Click "Configure Security Group". Then select "Create a new security group". You will need to open TCP ports 22,443, and 4000. If need be you can restrict the allowable IP addresses to the IP from which you will be logging into the server. however you will also need to open up ports 8085 and 8086 to all ip addresses. Then click "Review and Launch".

The screenshot shows the AWS Management Console interface for 'Step 6: Configure Security Group'. It starts with a title 'Step 6: Configure Security Group' and a description: 'A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.' Below this is a section 'Assign a security group:' with two radio buttons: 'Create a new security group' and 'Select an existing security group'. The 'Select an existing security group' option is selected. Below this is a table of existing security groups. The table has columns for 'Security Group ID', 'Name', 'Description', and 'Actions'. One group is listed: 'sg-cda54ca9', 'default', 'default VPC security group', and 'Copy to new'. Below the table is a section for adding rules. It has a table with columns for 'Type', 'Protocol', 'Port Range', and 'Source'. The table contains five rows: 'SSH' (TCP, 22, 0.0.0.0), 'Custom TCP Rule' (TCP, 8086, 0.0.0.0), 'Custom TCP Rule' (TCP, 8085, 0.0.0.0), 'HTTPS' (TCP, 443, 0.0.0.0), and 'Custom TCP Rule' (TCP, 4000, 0.0.0.0). At the bottom right of the form are buttons for 'Cancel', 'Previous', and 'Review and Launch'.

- Then click launch.

The screenshot shows three navigation buttons: 'Cancel', 'Previous', and 'Launch'. The 'Launch' button is highlighted in blue.

- Now create a new key pair and download it to your local disk by clicking "Download key pair". Save this file to your local disk. Click "Launch Instances".

### Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name  
mykey

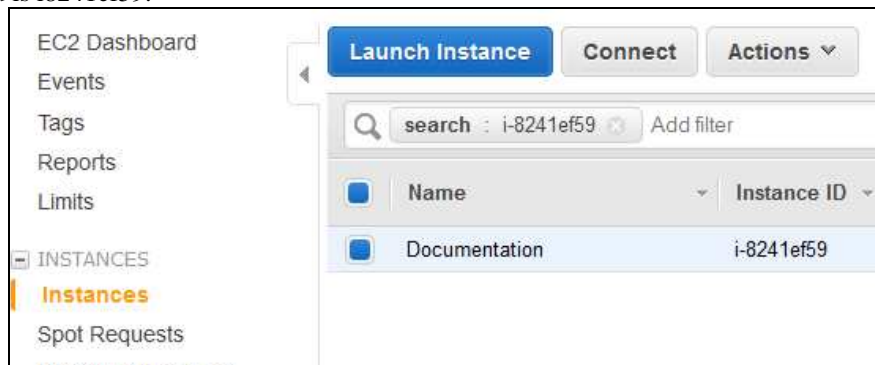
Download Key Pair

You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances

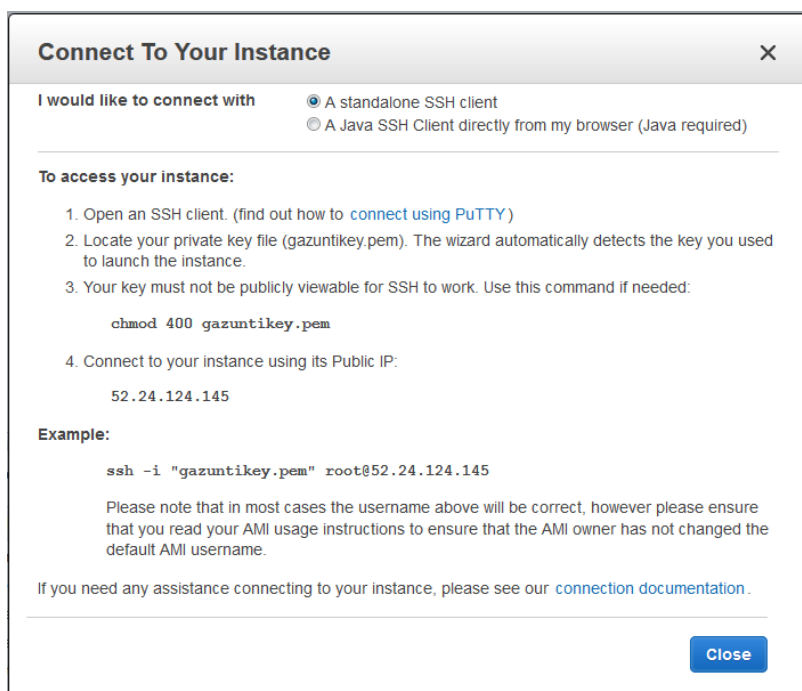
### 3.4 LOGGING INTO YOUR SERVER AND CHANGING PASSWORDS

When you create your server instance, it will come with a default password. To identify this password, click on the "Instances" menu item in the left pane of the AWS console. The instance ID will appear in the second column of the table listing the running instances. For instance in the screen shot below, the instance ID is i-8241ef59. The default password for the server is the instance ID of the server with the "-" symbol removed. I.E. in the screen above the default password is i8241ef59.



If you click the check box next to the server ID and then click on Connect you will get the following screen describing how to connect to your instance.

## How to build GazApps



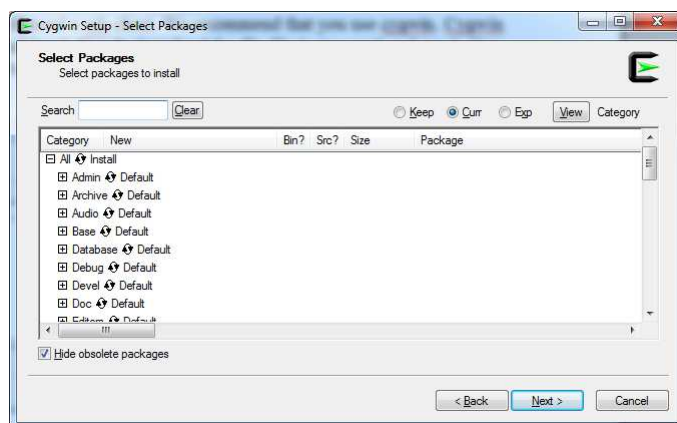
the text in this screen is incorrect in that you should connect using the username "samba" and not "root". Specifically the command line argument would be

```
ssh -i "gazuntikey.pem" samba@52.24.124.145.
```

Note the IP address of the server will change every time the server restarts unless you use an elastic IP address. This is not required if you are simply experimenting with the server, then you do not require the use of an elastic IP, but if you are running a permanent server you should

- Obtain and attach an elastic IP address to your server and
- add an A record to your DNS server to associate a host name with your elastic IP address

To secure shell into your GazCloud server you will need a ssh client. We recommend that you use cygwin. Cygwin can be downloaded from <https://cygwin.com/install.html>. Simply download the file file "setup.exe" and run it. You can select the default options for all of the screens, with the exception of the "Select Packages screen" Here you can elect to either select all packages as shown in Figure 3below or alternatively to select the default installation plus the package openssh as show in Figure 3 below.



**Figure 3 Electing to Install all Packages**

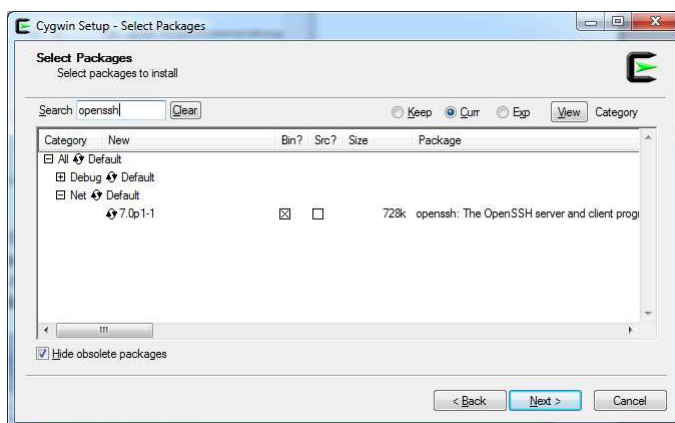


Figure 4 Selecting the Openssh Package

After you have logged on you can change your password by entering  
`sudo /root/bin/password_manager.sh`

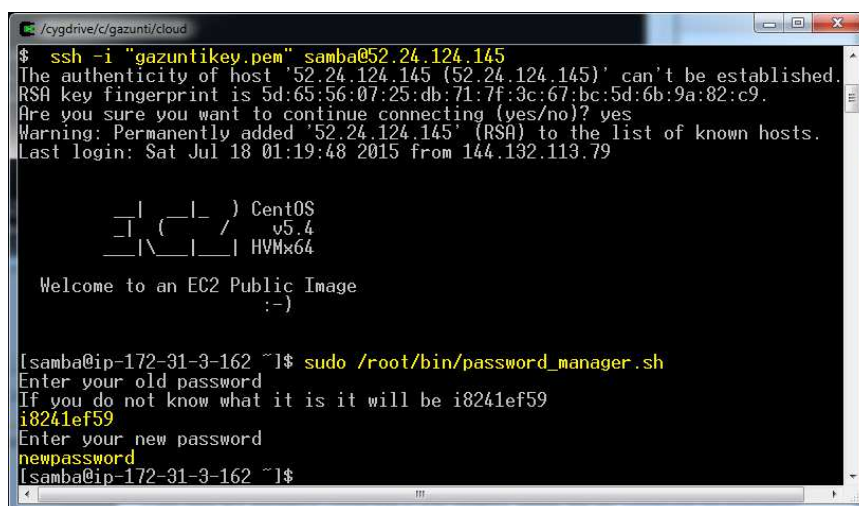
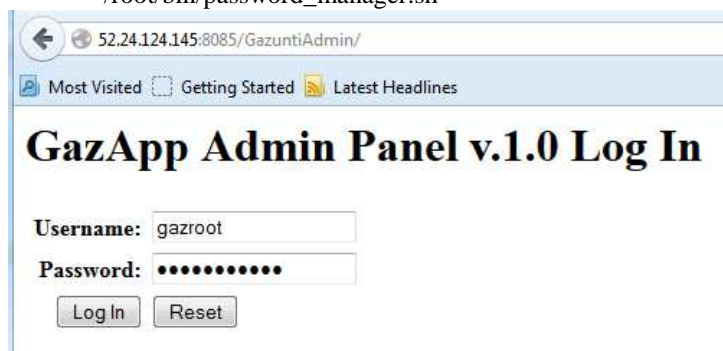


Figure 5 Changing the Default Password

### 3.5 CRAWLING A WEB SITE

To point the GazApp at your web site perform the following

1. Enter the gazbuilder URL. This will be of the form `SERVERADDRESS:8085/GazuntiAdmin`, where `SERVERADDRESS` is the address of your AWS server. You will then be requested to enter your username and password. Use the username "gazroot" and the password you just entered when you ran "`sudo /root/bin/password_manager.sh`"



2. Enter the details and the application will navigate to the main admin page, which is as shown below in Figure 6.

## How to build GazApps

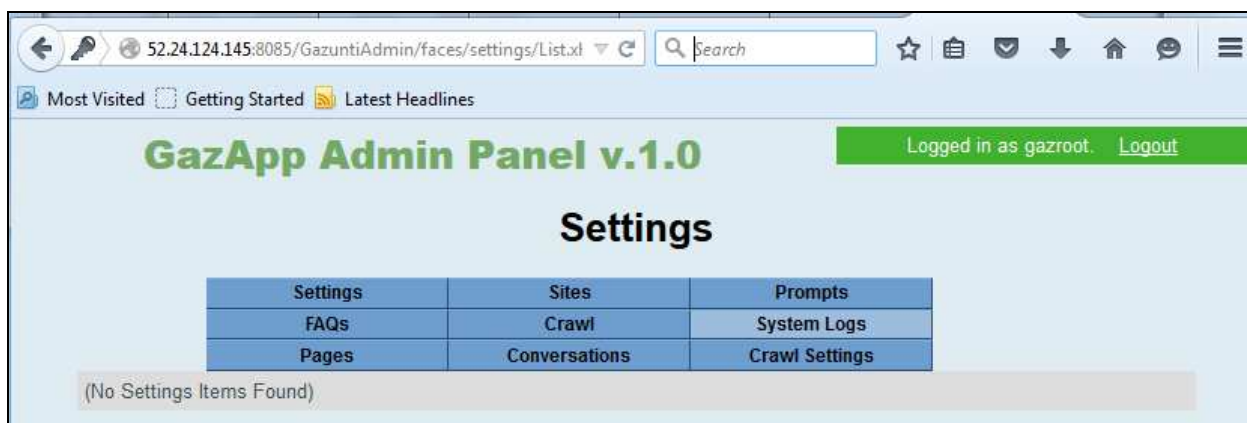
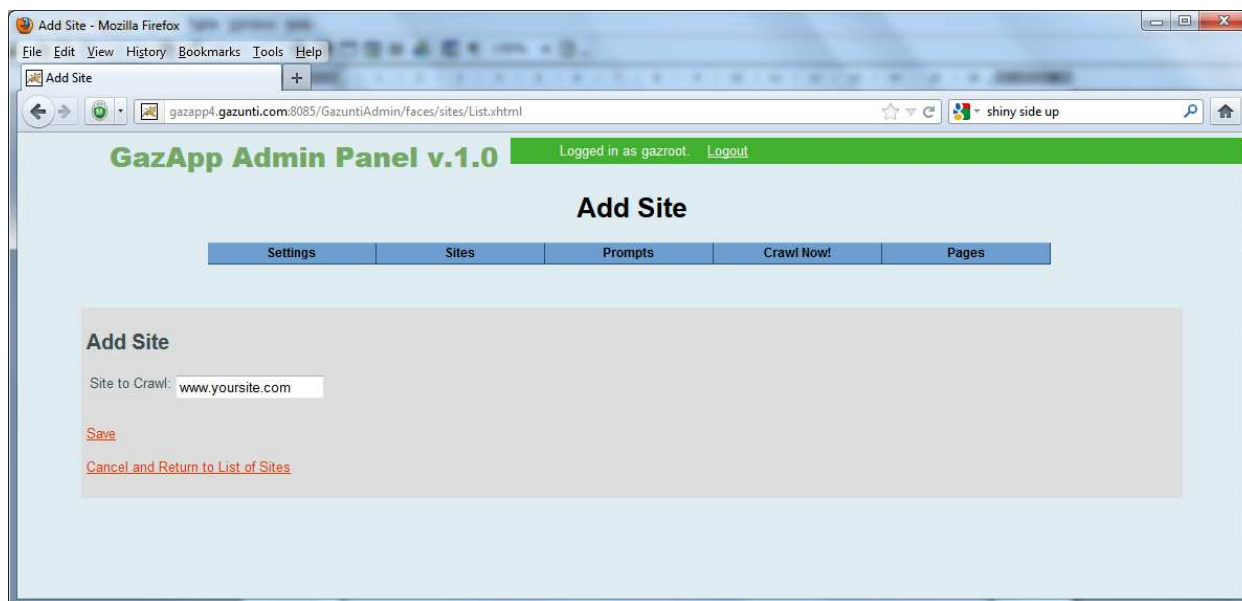


Figure 6 Main Admin Page

3. Now click on the button marked "Sites" on the menu bar. Click delete and the "Add Site". Enter your site you wish to crawl and then press save. For experimentation purposes you can use [http://aws1.gazunti.com/small\\_crawl/](http://aws1.gazunti.com/small_crawl/)



4. When you have completed adding your site URLs, click on "Cancel and return to list of sites".





## How to build GazApps

- To modify prompts click on the "Prompts" button in the men bar at the top of the screen.
- You will then be presented with a list of prompts to modify. You can edit the prompt by clicking on the edit hyperlink in the rightmost column. It should be intuitive what each prompt is used for, and the default prompts give you some guidelines as to what text should be replaced. Each prompt has an optional placeholder "@" that will be replaced by the GazApp when the prompt is spoken or displayed. The meaning of the placeholder should be obvious to you.

The screenshot shows the 'Prompts' page in the GazApp Admin Panel. At the top, there's a navigation menu with 'Settings', 'Sites', 'Prompts', 'FAQs', 'Crawl', 'System Logs', 'Pages', 'Conversations', and 'Crawl Settings'. Below the menu, a table lists prompts:

Question Type	Answer Type	Answer Text	Weight	
intro	intro	This prompt should be changed. You can do so by logging into your GazApp portal. Welcome to the INSERT NAME HERE. I can answer your questions about INSERT TOPIC HERE. To ask or answer a question, press the button on the screen and speak. What would you like to talk about?;	1	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
default	reprompt	What would you like to know about @?	1	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
menu	menu	Please select one from the following list	1	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
default	several_answers	There are several ways I can answer that question.	1	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

- You can then "Reset Webpages and Crawl Again" by clicking on "Crawl" in the menu bar and then "Reset Webpages and Crawl Again". All web pages will then have been cleared and will be reloaded again from "Sites" table starting within the next minute.

The screenshot shows the 'Crawl' page in the GazApp Admin Panel. At the top, there's a navigation menu with 'Settings', 'Sites', 'Prompts', 'FAQs', 'Crawl', 'System Logs', 'Pages', 'Conversations', and 'Crawl Settings'. Below the menu, a prominent link 'Reset Webpages and Crawl Again' is visible.

- You can click on "System Logs" and "Pages" links in the menu bar to view the progress of crawl. Also see "Crawl Settings" and "Sites".



## How to build GazApps



### 3.6 USING THE CHATBOT

1. You can test your gazapp by entering the default chatbot for your site". This URL will be of the form SERVERADDRESS:8085/GazuntiServer. You can then enter your text and chat with your gazapp. If you loaded the site use [http://aws1.gazunti.com/small\\_crawl/](http://aws1.gazunti.com/small_crawl/) you should see the page shown above in Figure 1. The first question that is asked relates to setting the topic. In response to this question you should answer "brad". Table 1 below lists some sample questions that can be asked and the answers to those questions.

## How to build GazApps



**Figure 7 GazApp Chat Bot**

**Table 1 Sample Questions**

Question	Answer
what colour jacket does brad wear	Brad Starkie wears a blue jacket . What else would you like to know?
who wears a jacket	I think that would be Brad Starkie . Brad Starkie wears a blue jacket . What else would you like to know?
how does he eat lunch	He eats lunch every day at 12:00 pm . What else would you like to know?
how far does he live from the CBD	I think that would be 10 kms . Brad lives 10 kms from the Cbd . What else would you like to know?
how long is his foot	There are several ways I can answer that question. Firstly, He is 5 feet 10 inches tall . Secondly, His foot is 20 cm long , so the answer would be 20 cm . Thirdly, Brad lives 10 kms from the Cbd , so the answer would be 10 kms . What else would you like to know?
how much did they spend on lunch today	I think that would be \$ 6 . He spent \$ 6 on lunch today . What else would you like to know?
why does he wear a jacket	He wears it because it looks good . What else would you like to know?
is brad a person	Yes. Brad Starkie is a person . What else would you like to know?
when was brad born	I think that would be 1967 . He was born in 1967 . What else would you like to know?
what percentage of his children are female	I think that would be 100 percent . 100 percent of his children are female . What else would you like to know?
what is a coat	There are 3 definitions. The first is an outer garment that has sleeves and covers the body from shoulder down; worn outdoors. The first is a thin layer covering something; "a second coat of paint". The first is growth of hair or wool or fur covering the body of an animal. What else would you like to know?
show me a picture of brad	I'm sorry , but I don't have a picture of that. What else would you like to know?
what does brad wear	Brad Starkie wears a blue jacket . What else would you like to know?
which coat does brad wear	Brad Starkie wears a blue jacket . What else would you like to know?
how often does he eat lunch	He eats lunch every day at 12:00 pm . What else would you like to know?
how many children does he have	There are several ways I can answer that question. Firstly, He has two children , so the answer would be two children . Secondly, 100 percent of his children are female , so the answer would be 100 . What else would you like to know?
where does he live	I'm sorry , but I don't know where, or even if it ever did or will happen. What else would you like to know?

## How to build GazApps

---

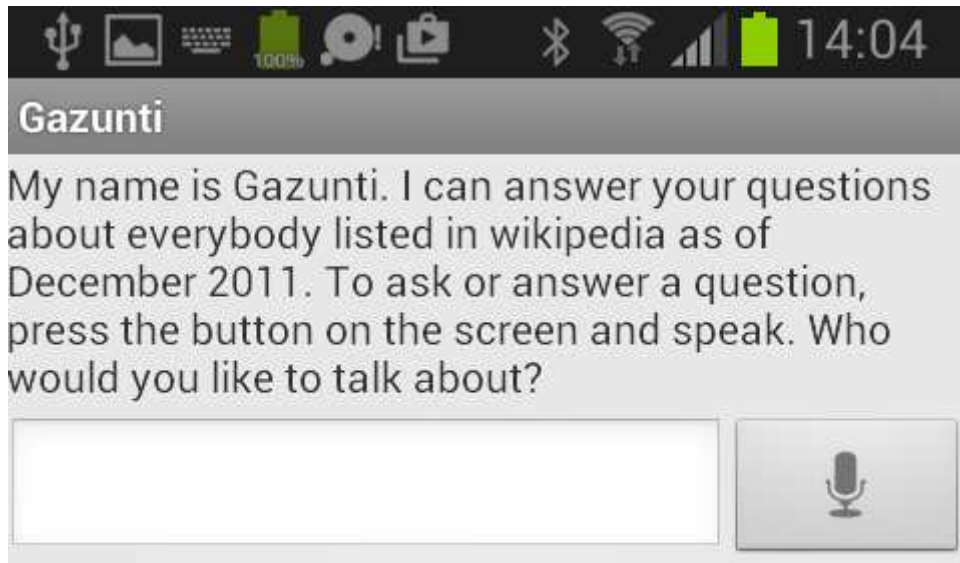
how long has brad been in melbourne	I think that would be 10 kms . Brad lives 10 kms from the Cbd . What else would you like to know?
how fast is his motorcycle	I think that would be 130 kilometres_per_hour . His motorcycle has a top speed of 130 kilometres_per_hour . What else would you like to know?
how big is his swimming pool	I think that would be 375,000 litres . His swimming pool holds 375,000 litres of water . What else would you like to know?

### 3.7 USING THE ANDROID GAZCLIENT

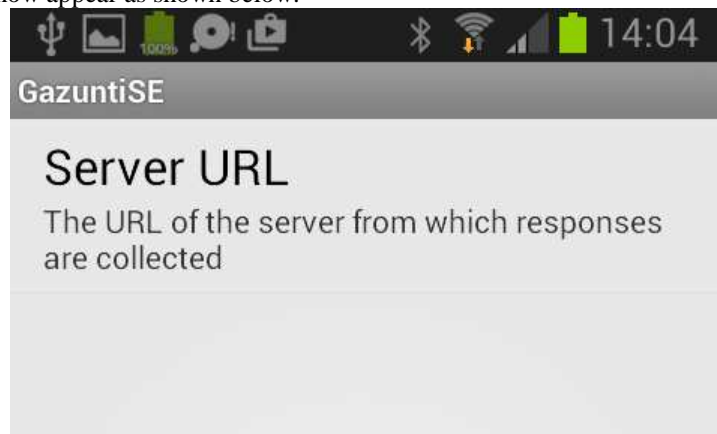
2. You can also download the Android GazClient from [http://SERVERADDRESS /GazuntiSE.apk](http://SERVERADDRESS/GazuntiSE.apk) and point it at your site. The following images are screenshots of an Android phone. To access your server open the GazClient App as shown below.



3. When you opened the application it should appear as shown below. Press the menu button. The Menu button is the button to the left of the home button at the bottom of the screen.



4. The screen should now appear as shown below.




5. Now click on "Server URL" and the screen should now appear as shown below.



6. Enter the same URL that you used to access the chat bot. Make sure to include a trailing "/" character. The URL should be of the form SERVERADDRESS:8085/GazuntiServer. Press OK and then exit the application by pressing the back button until you return to the screen in which the Gazunti icon appears as shown below.



7. You can now ask the same questions that you asked when accessing the chat bot as listed in Table 1 above. The first question that is asked relates to setting the topic. In response to this question you should answer "brad". To use speech input click the  icon. To enter text click on the text box and enter using the keyboard.



### 4 ADVANCED APPLICATION DEVELOPMENT

To create a custom application, you can begin with the GazuntiServer app and modify it.

Modification can be as simple as

- o Modifying the Java properties file that allows you to select which component you want to customise, including which indexer to use, or whether the default place and person tables are used.
- o Replacing the String rewriting rules. (See section 4.6).
- o Replacing Context free grammars for concept spotting (see Section 4.8)

If you want to interwork with a structured database, want to create custom question types or write code, you will need to use the Eclipse SDK installed on the server. To do so requires that you have skills in the following areas

- o Java
- o Tomcat
- o Eclipse
- o XWindows
- o MySQL

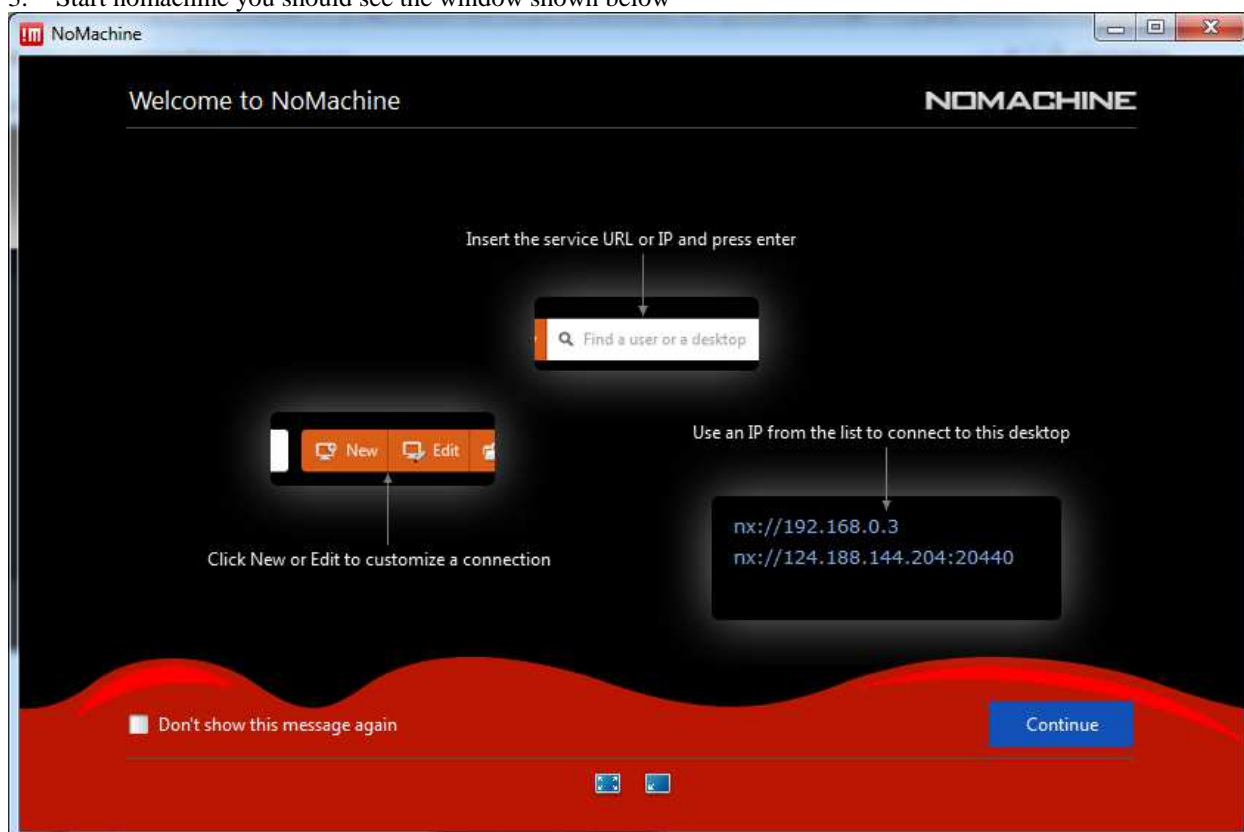
If you wish to write your own version of the parser you will also need expertise in BYACC/J.


The remainder of this document describes the underlying components that can be modified (see section 4.7).

#### 4.1 USING NOMACHINE AND ECLIPSE

The Amazon GazCloud server image comes with all of the software and source code required to develop the source code. To access the SDK perform the following

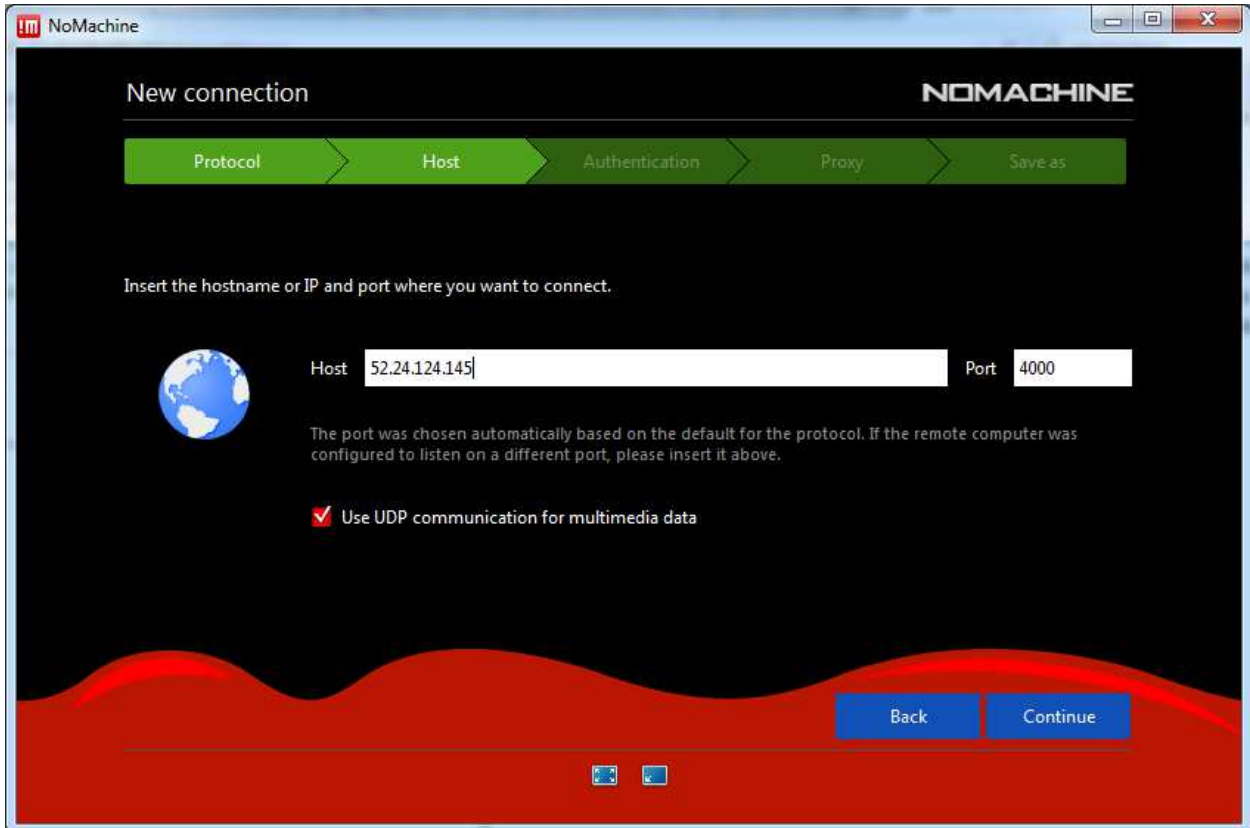
1. First you need to start the NX server. By default the server does not run to prevent unauthorised access to the machine. If you do start the server you should consider modifying the inbound rules on the security group to limit the IP addresses that can be connected to 4000 on the server.
2. Download and install the nomachine executable from <https://www.nomachine.com/download>
3. Start nomachine you should see the window shown below



4. Click Continue then click on Connect. Then click 
5. Select NC protocol

## How to build GazApps

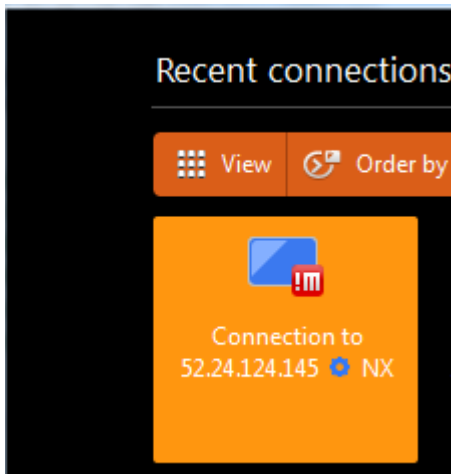
6. Enter the host in the next window



7. Use Password authentication

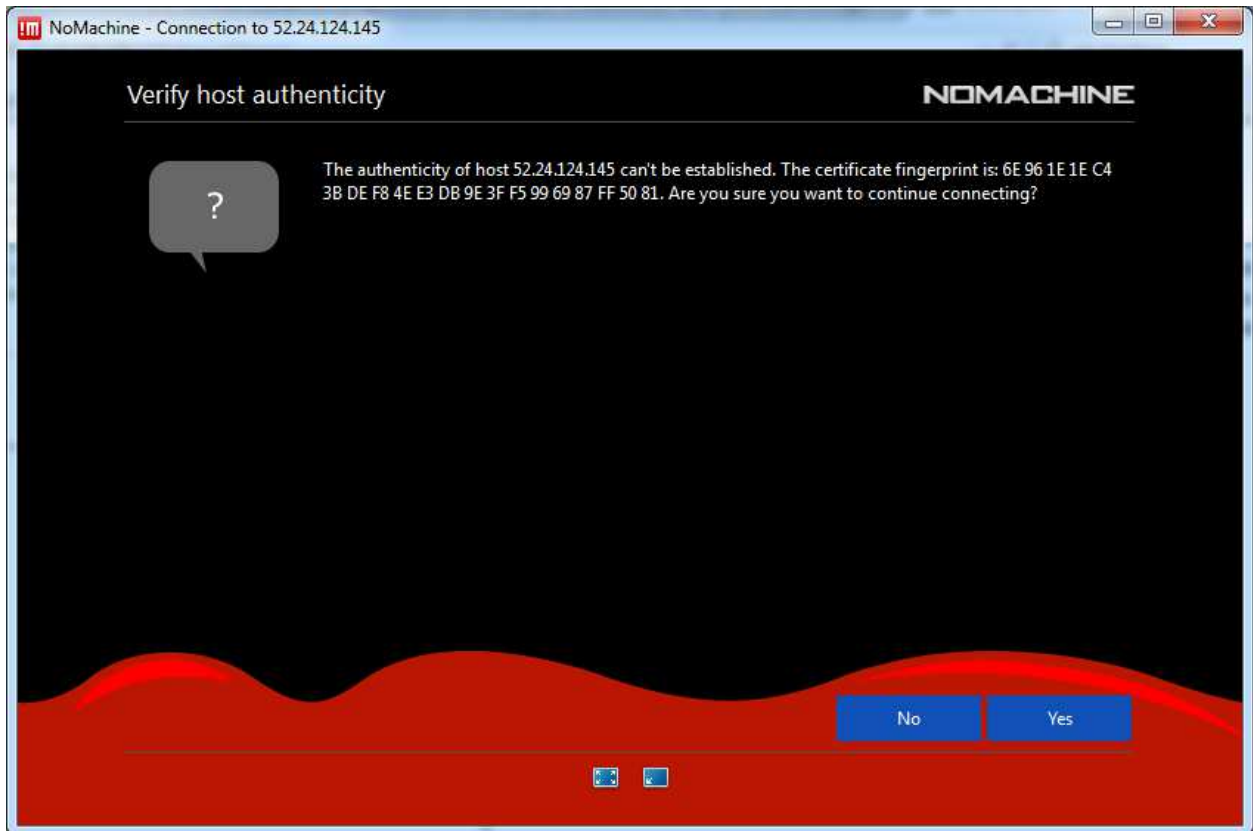
8. Don't use a proxy

9. Click on the new connection



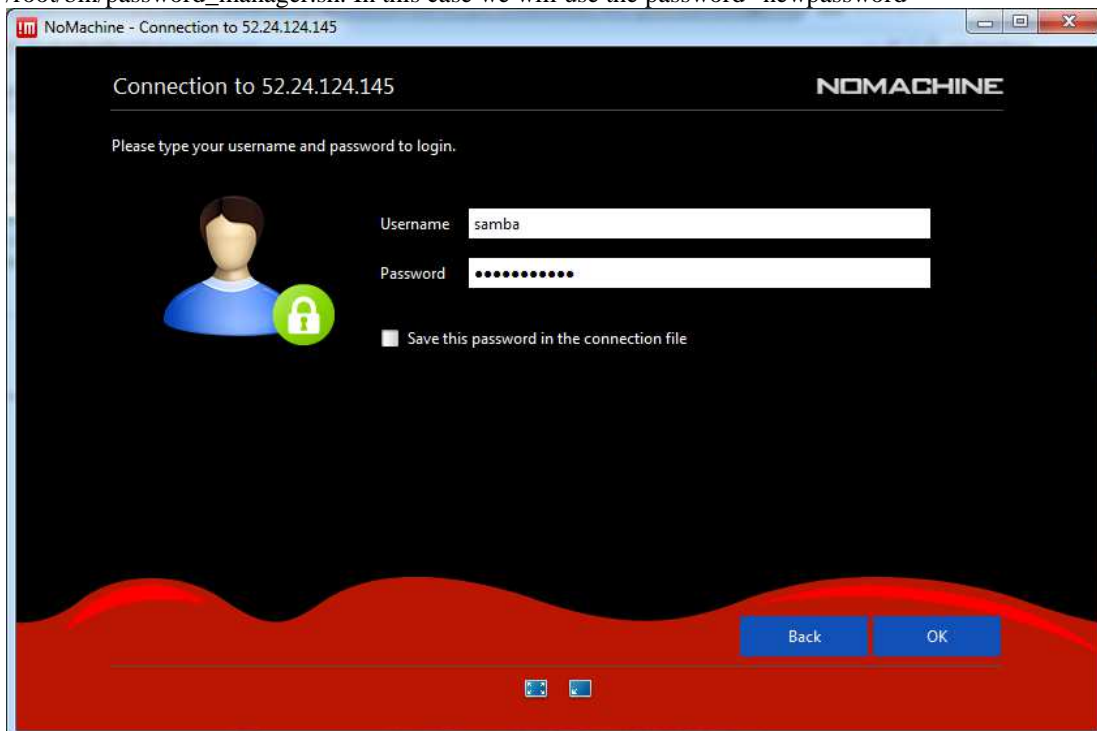
10. You should now see the following on your display

## How to build GazApps



11. Click "Yes"

Use the username samba and the password that you entered when you changed your password by entering `sudo /root/bin/password_manager.sh`. In this case we will use the password "newpassword"



12.

## 4.2 JAVA PROPERTIES FILE

Each GazApp contains the file `gazanti2.properties` which defines a number of implementation options for the server. The properties file for the main server is located at



## How to build GazApps

/opt/tomcat/GazApp1/GazuntiServer/WEB-INF/classes/gazunti2.properties

If you modify the java code and reinstall the war file this file will be clobbered. Therefore before you update the war file perform the following

- o Log in as root
- o cd /opt/tomcat
- o bash backup\_properties.sh

After you reinstall the app do the following

- o cd /opt/tomcat
- o bash restore\_properties.sh
- o restart the app using the tomcat manager <http://YOURSERVER:8085/manager/html>

The properties that can be contained in the file are shown below. The properties are listed in the order of importance. The lower the number the more likely it is that you will want to modify the property.

**Table 2 Properties Used by the Gazunti Server**

Property	Purpose	Possible Values	Example Value
dbuser.database	Name of MYSQL database and encoding used	The value of dbuser.url and dbuser.database are concatenated and passed to dbuser.url.DriverManager.getConnection()	GazApp1?characterEncoding=utf8
dbuser.url	URL used to access database	The value of dbuser.url and dbuser.database are concatenated and passed to dbuser.url.DriverManager.getConnection()	jdbc:mysql://example.com:3306/
dbuser.user	SQL user		root
dbuser.passwd	Password of database		password
indexer.type	The type of indexer used	every_word/titles_and_sections/headings_and_guts/solr/gazstar/email_thread/gazcrunch	gazcrunch
resolver.set	Determines which set of question types are supported	mail/wikipedia	wikipedia
partial_parser	The type of partial parser used	ConceptParser/ChartParser	ConceptParser
decode_anaphora	if set to 1 will decode pronouns such as I and you into proper nouns, mainly used in GazMail	0 or 1	1
question_answerer.s	file prefix of term		C

## How to build GazApps

et_topic_prefix	rewriting system for setting the topic "what would you like to talk about"		
page_loader.config_file_prefix	file prefix of term rewriting system for loading pages		../GazApp1/GazuntiServer/stage13
question_answerer.config_file_prefix	file prefix of term rewriting system for answering questions		../GazApp1/GazuntiServer/questions
gazmail_indexer.config_file_prefix	file prefix of term rewriting system for setting the topic "what would you like to talk about"		../GazApp1/GazuntiServer/gazmail
people_rules_index_table	MySQL table containing named entities		`people_rules_index`
people_rules_table	MySQL table containing indexes to named entities		`people_rules`
classifier_table	MySQL table containing words for noun type classifier		`noun_words`
noun_type_table	MySQL table containing prior probabilities for noun type classifier		np_classes
localstoreprefix	A prefix to be added to the URL to display an indexed item	Any Valid URL prefix	http://www.example.com/GazMail/dynamic?id=
wordnet.directory	Directory of wordnet files		../GazApp1/GazuntiServer/dict
bag_of_words.max_keyword_len	The Maximum length of an index	Any Integer	40
gazcrunch.server	URL of the GazCrunch Server for transferring from one server to another		http://aws2.gazunti.com:8085/GazCrunch/
wikipedia.server	URL of the GazStar Server for transferring from one server to another		http://aws2.gazunti.com:8085/GazStar/
test.data	Directory of files for unit tests		/opt/tomcat/GazApp1/GazStar
crawl_cmd	A linux command to crawl a site	The path of an executable	/opt/nutchdata/GazApp1/crawl.sh
nutch_dir	A directory containing	Any valid path	/opt/nutchdata/GazCrunch

## How to build GazApps

	executables used by nutch		
seedfile	Solr Seed file		/opt/nutchdata/GazApp1/conf/seed.txt
solr_prefix	URL for SOLR web services		http://127.0.0.1:8085/solr/gazapp1/collecti on1
solr_url	URL for SOLR web services		http://127.0.0.1:8085/solr/gazapp1
tagger.posdb	database used for part of speech tagging		GazCommon

**Table 3** Possible Values of the `indexer.type` Property

Value	Meaning
every_word	The indexer takes into account every word in a page.
titles_and_sections	The indexer considers only the titles of pages and sections within the page.
headings_and_guts	The indexer considers the titles of pages and sections within the page, but if it cannot find a match it also considers every word in a page.
solr	Uses the SOLR indexer. The use of SOLR and Nutch is not recommended
gazstar	The GazStar indexer. Uses the people table. The number of pages is fixed.
email_thread	The GazMail indexer.
gazcrunch	The GazCrunch indexer, a more general version of the GazStar indexer

### 4.3 SQL DATABASE

The GazCloud server comes with both MySQL and PHPMyAdmin installed. You can access PHPMyAdmin via <http://SERVERADDRESS/phpMyAdmin> or <https://SERVERADDRESS/phpMyAdmin>. By default all GazApps share a common database of proper nouns, specifically people and places. Each GazApp can also have its own proper noun database. Table 4 lists the tables used in GazApp databases. These tables can exist in the one database or can be split and shared across multiple databases, according to the values of properties in the properties file as described in Table 2. There are scripts for backing up and restoring the MySQL databases in `/samba/sql/gazunti`.

**Table 4** Tables in the GazApp Database used in Gazunti Lite

Table	Records	Module	Purpose
people_rules	7,236,110		A list of all proper nouns
sentenceindex	31874	Bag of Words	Index to the sentence table
sentences	2361	Bag of Words	A list of all sentences in the sentence table
people	788892	matching name	A list of people in wikipedia as defined by dbpedia
loaded_topics	6	wiki caching	List of loaded wikipages
topics	1	wiki caching	List of loaded wikipages
conversation_log			Stores all conversations with the chat bot

## How to build GazApps

users			Stores information about sent cookies
Infobox			An SQL implementation of a graph triple database

**Table 5 MYSQL Tables**

Database	Function
GazApp1	The main GazApp database
GazCommon	POS data
GenericPeopleRules	Named Entity Recognition
authority	tomcat users and passwords

**Table 6 The Sentence Table**

Field	Type	Null	Default
<i>Id</i>	int(12)	Yes	NULL
Context	varchar(150)	Yes	NULL
Text	text	Yes	NULL
ordered_keywords	text	Yes	NULL
BECAUSE	int(3)	Yes	0
Absorbant	int(3)	Yes	0
Age	int(3)	Yes	0
Angle	int(3)	Yes	0
Density	int(3)	Yes	0
Distance	int(3)	Yes	0
Percent	int(3)	Yes	0
Elastic	int(3)	Yes	0
Frequency	int(3)	Yes	0
Luminosity	int(3)	Yes	0
Money	int(3)	Yes	0
Pain	int(3)	Yes	0
Power	int(3)	Yes	0
Quantity	int(3)	Yes	0
Radioactivity	int(3)	Yes	0
Resistance	int(3)	Yes	0
Score	int(3)	Yes	0
Size	int(3)	Yes	0
Speed	int(3)	Yes	0
Strength	int(3)	Yes	0
Temperature	int(3)	Yes	0
Viscosity	int(3)	Yes	0
Volume	int(3)	Yes	0
Weight	int(3)	Yes	0
When	int(3)	Yes	0
Where	int(3)	Yes	0
Who	int(3)	Yes	0
BECAUSE_val	text	Yes	NULL
absorbant_val	text	Yes	NULL
age_val	text	Yes	NULL
angle_val	text	Yes	NULL
density_val	text	Yes	NULL
distance_val	text	Yes	NULL
percent_val	text	Yes	NULL

## How to build GazApps

elastic_val	text	Yes	NULL
frequency_val	text	Yes	NULL
luminosity_val	text	Yes	NULL
money_val	text	Yes	NULL
pain_val	text	Yes	NULL
power_val	text	Yes	NULL
quantity_val	text	Yes	NULL
radioactivity_val	text	Yes	NULL
resistance_val	text	Yes	NULL
score_val	text	Yes	NULL
size_val	text	Yes	NULL
speed_val	text	Yes	NULL
strength_val	text	Yes	NULL
temperature_val	text	Yes	NULL
viscosity_val	text	Yes	NULL
volume_val	text	Yes	NULL
Weight_val	text	Yes	NULL
when_val	text	Yes	NULL
Where_val	text	Yes	NULL
who_val	text	Yes	NULL
num_words	int(5)	Yes	NULL
heading_depth	int(2)	Yes	0

### 4.4 CRAWLER

The directory /samba/cronjobs/GazuntiServer contains a shell script that crawls through web sites. The shell script is run every minute using cron under the user id of samba. The results of these cronjob runs can be read by logging into <http://YOURSERVER:8085/GazuntiAdmin> and clicking on "System Logs". A log file of the output of the last execution of the cronjob can also be found at /samba/cronjobs/GazuntiServer/crawl.log

The following document types are supported in GazCloud 1.1

.html .htm .php	HTML	text/html
.txt	PlainText	text/plain
.doc,.word	MS word	application/msword
.pdf	PDF	application/pdf <sup>1</sup>

### 4.5 DEFAULT PROMPTS AND ANSWERS

To give your GazApp the persona that you want, you can customise the snippets of text that are added to the answers that your GazApp adds to the response it finds in your texts and databases. These prompts are modified using the GazuntiAdmin application described in the section entitled "Creating a Quick Start App" listed above. This full list of prompts is listed below. Most of these prompts will not need to be modified by the developer.

#### 4.5.1 DEFAULT PROMPTS

To give your GazApp the persona that you want, you can customise the snippets of text that are added to the answers that your GazApp adds to the response it finds in your texts and databases. These snippets are stored in the prompts database. Table 7 lists some of the entries in the prompts table.

**Table 7 Answer Wrappers**

hashtag	AnswerType	Answer	Weight
intro	intro	This prompt should be changed. You can do so by logging	1

<sup>1</sup> Bitmapped pdfs e.g. scanned documents are not currently supported, and the accuracy of the read text would be dependent upon the accuracy of the Optical Character Recognition system used.

## How to build GazApps

		into your GazApp portal. Welcome to the INSERT NAME HERE. I can answer your questions about INSERT TOPIC HERE. To ask or answer a question, press the button on the screen and speak. What would you like to talk about?;	
default	reprompt	What would you like to know about @?	1
menu	menu	Please select one from the following list	1
default	several_answers	There are several ways I can answer that question.	1
none_of_the_above	none_of_the_above	OK. We won't talk about any of them. What would you like to talk about?	1
load_wait	load_wait	I haven't loaded that page before, so it will take a few minutes to load it. In the meantime I'd like to tell you some things about @	1
default	factoid_prompt	The best answer I have is @ .	1
match_topic	many_matches	I've found @ matches.	1
match_topic	no_match	I cant locate any information about @. Perhaps I misunderstood what you said. Let's talk about something else. What would you like to talk about?	1
match_topic	one_match	I found one page containing information about @.	1
match_topic	one_match_continue	If that doesn't sound like the right topic then say Let's talk about something else.	1
completely_lost	completely_lost	I'm sorry I did not understand you. What would you like to talk about?	1
error	error	There has been an error loading the page about @. Let's talk about something else. What would you like to talk about?	1
default	no_factoid	I don't know exactly but	1
default	unknown	Sorry, I have no idea .	1
please_wait	please_wait	Please wait.	1

### 4.5.2 FREQUENTLY ASKED QUESTIONS

You can also specify answers to frequently asked questions explicitly by clicking on the FAQ tab. Your GazApp comes predefined with the following FAQs.

Question	Answer
hello	hello
how are you	i am fine, how are you
i am fine	good to hear that
goodbye	see you later

### 4.6 STRING REWRITING RULES

The first step in processing text in a GazApp is to apply string rewriting rules. String rewriting rules are used to convert different ways of expressing the same thing into a "normal form" as also to convert sentences into question types on the basis of prefixes of the strings. Detailed information on string rewriting systems can be found in (Book and Otto 1993). The string rewriting rules are similar to the production rules of a context free grammar, with the exception that

- The left hand side of the rule defines the text to be replaced and
- The right hand side defines the text to replace the left hand side with.

In this respect when applied to the task of parsing text they are written in a different order to context free grammars, where the right hand side would define the text to be replaced.

Some other differences include

- The right hand side of the rewrite rule can be longer than one word long,
- The rules can be applied to any sub-string and
- The final outcome of applying the rules need not be a single symbol.

**Figure 8 An example String Rewriting system.**

```
^ how -> ^ #how .  
^ why -> ^ #why .  
^ #how come -> ^ #why .  
^ #length is -> ^ #length .  
^ #length a -> ^ #length .  
^ #how long -> ^ #length .  
^ #length does -> ^ #time_length .  
^ #time_length it take to -> ^ #time_length .  
^ #why is -> ^ #why .
```

The purpose of string rewriting rules in Gazunti is to enable different strings with the same meaning to be treated in the same manner, while converting long complicated sentences to simpler, shorter ones. Gazunti string rewriting rules differ from symbol reduction in AIML in that as many rules are applied to the string, until no more rules can be applied.

**Termination** is a property of a string rewriting systems which denote whether or not there is a guaranteed point at which no more rules can be applied. Take for instance the following string rewriting rules

**Figure 9 A non terminating String Rewriting System**

```
A -> B  
B -> C  
C -> A
```

This string rewriting system is not guaranteed to terminate. If you begin with the sentence Z A C, the rules can be applied ad infinitum as follows

Z A C => Z B C => Z C C => Z A C => Z B C => Z C C ...

A string rewriting system will terminate if and only if, all of the rules are ordered according to a well-ordering.

Gazunti uses the length lexicographical ordering which is defined as follows

A < B if and only if  
(|A| < |B|) or  
(|A| == |B|) and (lex(A) < lex(B))

where

lex(A) is analogous to the alphabetic ordering, but using words instead of letters.

For instance in Figure 9 the rule A-> B is not well-ordered. It can be made well-ordered by swapping the left and right hand sides of some rules resulting in the system of Figure 10 below.

**Figure 10 A non terminating String Rewriting System**

```
B -> A  
C -> B  
C -> A
```

When these rules are applied to the sequence Z A C, there are two possible reduction sequences, either  
Z A C => C A A or  
Z A C => Z A B => C A A.

In this example above there are two ways in which Z A C can be resolved to C A A. One these involves the use of the rule "C -> B" and one doesn't. In this case it doesn't matter what sequence of rules is used, the outcome is the same. String rewriting systems that have this property are called **confluent**.

For all string rewriting systems there is a normal form which is defined as the string rewriting system with the following properties.

- For all rules of the form A -> B , A < B for some defined well-ordering.

- For all rules of the form A -> B , there does not exist any other rule of the form C -> D where C is a substring of A.

A consequence of the second rule, is that there will not be any duplicate rules.

The GazCloud server includes code for testing for string rewriting systems as well as normalising systems. Table 8 lists these executables.

**Table 8 Executables used for Constructing and Executing Stage 2**

Executable	Purpose
Normalise	Normalises a String Rewriting System
stage2	Applies Stage2 of Gazunti to plain text file

Gazunti string rewriting rules include the following symbols

- ^ denotes the beginning of a sentence
- \$\$ denotes the end of a sentence.

In Gazunti the string rewriting system is also used to perform the following functions

- To convert compound words to single words, in particular those related to units and
- To classify questions in Gazunti-lite.

We will now demonstrate this by applying the string rewriting system of Figure 10 to three different strings, as shown below in Figure 11.

**Figure 11 Examples being Reduced using a Term Rewriting System**

```
^ How long is a piece of string $$
^ #length is a piece of string $$
^ #length a piece of string $$
```

```
^ How long does it take to learn to fly $$
^ #how long does it take to learn to fly $$
^ #length does it take to learn to fly $$
^ #time_length it take to learn to fly $$
^ #time_length learn to fly $$
```

```
^ How come the sky is blue $$
^ #how come the sky is blue $$
^ #why the sky is blue $$
```

```
^ why is the sky blue $$
^ #why is the sky blue $$
^ #why the sky blue $$
```

## 4.7 NOUN PHRASE CLASSIFIER MODULE

As shown in Figure 2 the fourth stage is the Noun Phrase Classifier module. The GenericPeopleRules database includes the full names of all people listed in Wikipedia in the people\_rules table. It also contains the first and last names of all of these people. It also contains all of those places that are referenced in wikipedia. This information is useful for identifying who and where in a sentence. If you want your GazApp to recognise people places and things that are not in Wikipedia you need to add rows to the people\_rules and people\_rules index, in your own version of the database.

```
CREATE TABLE IF NOT EXISTS `people_rules` (
  `context` varchar(150) NOT NULL,
  `name` varchar(200) NOT NULL,
```



```
`id` int(11) NOT NULL auto_increment,
`class` varchar(50) NOT NULL default 'PERSON',
`num_words` int(2) NOT NULL default '0',
PRIMARY KEY (`id`),
UNIQUE KEY `context` (`context`,`name`),
KEY `context_2` (`context`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `people_rules_index` (
`id` int(11) NOT NULL,
`keyword` varchar(70) NOT NULL,
UNIQUE KEY `id` (`id`,`keyword`),
KEY `keyword` (`keyword`),
KEY `id_2` (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

**Figure 12 The Schema of the people\_rules and people\_rules\_index table**

For a phrase within a sentence to match an entry in the people\_rules\_table the following must occur.

- The first word in the name must be assigned the tag NNP by the tagger.
- There must be an entry in the people\_rules table. The name column must match the words to be matched. The phrase is then reduced to a single symbol with the type as listed in the class column. This class name can be used to interact with the grammar in the next stage.
- The first word in the phrase must exist in the people\_rules\_index table.

## 4.8 PARSER MODULE

The parser module adds hierarchical structure to sentences entered into Gazunti. It takes as its starting point the output of the np classifier and POS tagger stages.

There are two possible kinds of parser that can be used

- A right to left bounded right context free chart parser is used. The context free grammar does not attempt to parse the entire sentence.
- A default LALR(1) parser generated using BYACCJ

If the partial\_parser property is set to ConceptParser the LALR(a) parser is used otherwise the Chart parser is used.

Parsing converts sentences into structured representations using a model of a language known as a grammar (Allen 1995).

The default context-free grammar that comes with your GazApp, identifies times, dates and people, address and quantifier phrases such as "three kilograms"

If you are using the chart parser you can modify the following grammars

- `#{question_answerer.config_file_prefix}.rulelist` contains the grammar used when answering questions,
- `#{page_loader.config_file_prefix}.rulelist` contains the grammar used when loading pages and
- `#{question_answerer.set_topic_prefix}.rulelist` contains the grammar used when selecting the topic or thread.

```
RTLRulesstart : rules ;

rules :
  rule |
  rules rule
;

rule :
  symbol BECOMES symbols '~' FLOAT EOL { mk_rule_with_prob($1,$3,$5); } |
  symbol symbols FLOAT FLOAT EOL { rtl_mktyped_rewrite_rule($1,$2); } |
  symbol symbols EOL { rtl_mktyped_rewrite_rule($1,$2); }
```

```
EOL
;
symbols :
  symbols symbol { $$=rtl_join_rhs($1,$2);} |
  symbol { $$=rtl_mk_rhs($1);} ;
symbol :
  NONTERMINAL |
  TERMINAL
;
```

**Figure 13 Syntax of the GazApp Context Free Grammar**

```

CAPACITANCE -> CD CAPACITANCE_UNITS ~1
CAPACITY -> CD CAPACITY_UNITS ~1
CURRENT -> CD CURRENT_UNITS ~1
DATE -> CD TIME_UNITS TRP ~1
DATE -> DOW ~1
DATE -> DOW MONTH OD YEAR ~1
DATE -> DT TIME_UNITS ~1
DATE -> DOW CD MONTH ~1
DATE -> DOW OD MONTH ~1
DATE -> DOW CD MONTH YEAR ~1
DATE -> DOW OD MONTH YEAR ~1
    
```

Figure 14 Example GazApp Context Free Grammar

## 4.9 MODIFYING THE JAVA CODE

The AWS server image contains Eclipse and code for all of the GazApps resident on the server. The Java code can be modified, and the installed on the tomcat server by exporting the war file. In addition there are JUnit tests to perform regression testing on the code after modifications have been made. The server contains a fully featured debugger (the Eclipse debugger) to enable debugging of the code on the server.

The server image comes with access to the public CVS repository.

### 4.9.1 ADDING NEW QUESTION TYPES

To add a new question type you will need to do the following

- o Create a new instance of the abstract class `com.gazunti.server.functions.QueryType`
- o Create a new instance of the abstract class `com.gazunti.server.functions.FunctionResolver`
- o Change the value of the property `resolver.set` in the properties file to some new value e.g. `myresolver`.
- o modify `com.gazunti.server.functions.QuestionAnswerer.load()` to create an instance of your new function resolver when it reads in a value of `resolver.set` that refers to your new resolver set e.g. `myresolver`.

When you write the code to your question type you can take advantage of the following methods in the `QueryType` class. Use of these methods means that creation of a new question type is often a matter of providing parameters to the relevant method.

Table 9 Useful QueryType Methods

QueryType Method	Answering Method	Example Question Type
<code>process_query_say_factoid</code>	Factoid is extracted at load time, answer returns the factoid and the sentence	DistanceQuery
<code>process_query_dont_say_factoid</code>	Factoid is extracted at load time, answer returns the sentence	TemperatureQuery
<code>process_query_essential</code>	Answer must also contain a specified word, answer returns the sentence	CausesQuery
<code>process_query_say_factoid_needs_keywords</code>	Answer must also contain one of several words, answer returns the sentence	LastBillQuery
<code>process_query_infobox</code>	Uses the Graph Database stored in the infobox table	ChildQuery

Adding a new question type also involves writing new string rewriting rules (see section 4.6). For each new question type there needs to be at least one rule in the string rewriting rules, such that for at least one sentence, the output of the string rewriting system, begins with a the symbol `#X` where `#X` is the question type.

## How to build GazApps

---

Factoid extraction during page loading can be implemented using the parser module (see section 4.8). To add a new factoid type do the following

- Create a new non-terminal X in your grammar
- Add a new column Y to the sentences table in MySQL
- add a line of the form `add_decoration_rule("X", "Y")` to the `BagOfWordsOperator` constructor.