CIMA Implementing the UK's 1st Production SOA Cloud Service (Oracle Excellence Award 2016 – Cloud Integration)

Kiran Tailor
CIMA
London UK

Keywords: **SOA, Cloud, DBCS, Database, Monitoring, EM13C, Cloud Control, Integration, PayPal, Oracle Excellence Award**

Introduction

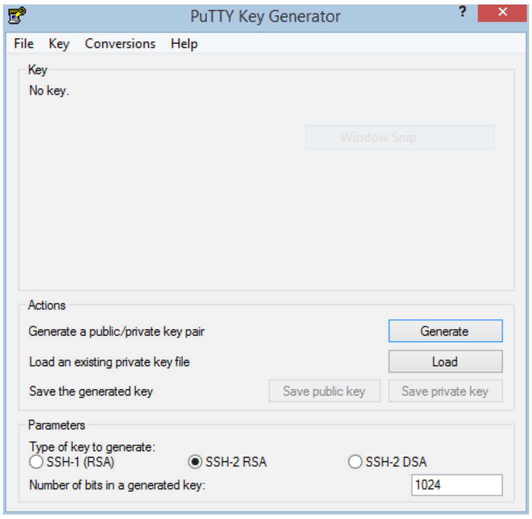
My name is Kiran Tailor,I am an Oracle Ace Associate, with extensive experience in Database Management, Performance and BI. At CIMA I am the Global Lead DBA/BI Technical Lead. In this role I lead and manage CIMA's DBA functionality and 3rd party service providers in efficiently and effectively installing,configuring,administering,monitoring and maintaining the database management systems that underpin CIMA's software applications to meet the dynamic business needs of CIMA's business units. I also lead the BI technical stream, managing the database warehouse, ETL implementations and Exalytics environment. Over 16 years' experience in the Administration of Oracle 7,8,10g,11g on Linux, Solaris 9 & 10 and Windows comprising of installations, configurations and upgrades (Oracle ASM,RAC,Dataguard & RMAN). Over 5 years' experience in OBIEE, Informatica, Exalytics and the Oracle DAC. I have presented at Oracle Openworld, UKOUG, Oracle Analytics Summit, Oracle Ireland Conference. I have articles published in the Oracle Scene as well as a video on the OTN Network.

CIMA

CIMA is the world‘s largest and leading professional body of management accountants, who have more than 229,000 members and students in 176 countries. They work at the heart of business in industry, commerce and not for profit organisations. CIMA recently went live with ERP Cloud. In this talk we will look at the real-world experiences when implementing the UK's first production SOA CS system. Seasoned administrators won't be surprised that implementing SOA CS, Java CS or Database CS is not quite as simple as many would have you believe, nevertheless with a bit of know-how you can build enterprise grade production and test environments on Oracle PaaS with far less effort than it used to take for on-premises systems.

Key Generation

First let's looks at public and private keys. One of the first steps when provisioning servers is that you will be requested for a public key, you can look at this as being a lock, i.e. your front door. When you generate this public key you will also have a private key, i.e. your key, this opens the lock to your server. You should guard this just like you do with your house key, you don't want to be losing it. Let's have a look at generating and using the key. What we will be using is the Putty Key generator, this is a freeware application and can be downloaded from the web. Open this application and you will see a screen that looks like fig 1.



PuTTY Key Generator

File Key Conversions Help

Key

No key.

Actions

Generate a public/private key pair — Generate

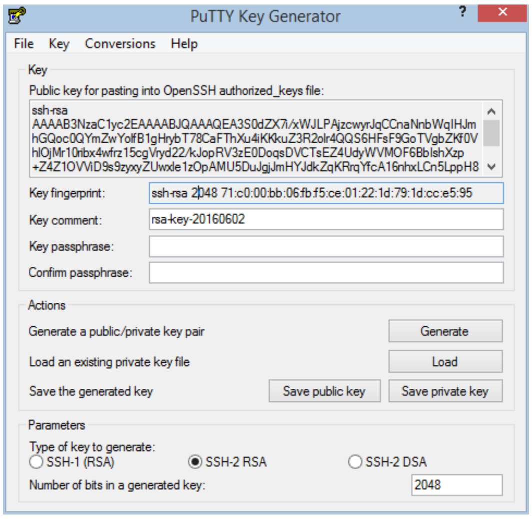
Load an existing private key file — Load

Save the generated key — Save public key — Save private key

Parameters

Type of key to generate:
○ SSH-1 (RSA) ● SSH-2 RSA ○ SSH-2 DSA

Number of bits in a generated key: 1024



PuTTY Key Generator

File Key Conversions Help

Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEA3S0dZX7i/xWJLPAjzcwyrJqCCnaNnbWqIHJm
hGQoc0QYmZwYolfB1gHrybT78CaFThXu4iKKkuZ3R2olr4QQS6HFsF9GoTVgbZKf0V
hlOjMr10ribx4wfrz15cgVryd22/kJopRV3zE0DoqsDVCTsEZ4UdyWVMOF6BbIshXzp
+Z4Z1OVViD9s9zyxyZUwxle1zOpAMU5DuJgjJmHYJdkZqKRqYfcA16nhxLCn5LppH8
```

Key fingerprint: ssh-rsa 2048 71:c0:00:bb:06:fb:f5:ce:01:22:1d:79:1d:cc:e5:95

Key comment: rsa-key-20160602

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair — Generate

Load an existing private key file — Load

Save the generated key — Save public key — Save private key

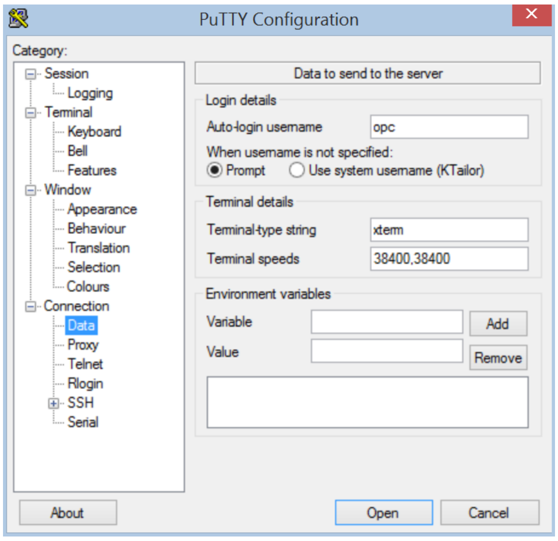
Parameters

Type of key to generate:
○ SSH-1 (RSA) ● SSH-2 RSA ○ SSH-2 DSA

Number of bits in a generated key: 2048

Fig. 1 Fig.2

In the section ‘number of bits in a generated key’ change this to ‘2048’ you will generate a stronger key. Click ‘Generate’ – At this point you will be asked to move your mouse randomly in the key box. Once finished you will see the key as in figure 2.
Copy the key from the box labelled ‘Public key for passing into Open SSH..’ and paste into notepad and save with a suitable name and extension of .pub ie test.pub.
Also click ‘Save private key’ – this is the private key and normally the extensions end in .ppk. Now we have a public and private key, the public key will be used in the steps of provisioning the server in the cloud.
Once the servers have been provisioned, how do we connect to the server using the keys we generated? One way is using a freeware product like putty and we can connect as follows. We would set up putty as we would normally connect to the server but make the following amendments, generally for Oracle they use the user opc for SOA, so we would enter this information under the section of connection/data as seen in fig 3. Now we will not be able to access the server without the key, so under the section connection/ssh/auth you select your private key fig 4.



PuTTY Configuration

Data to send to the server

Login details

Auto-login username: opc

When username is not specified:
● Prompt ○ Use system username (KTailor)

Terminal details

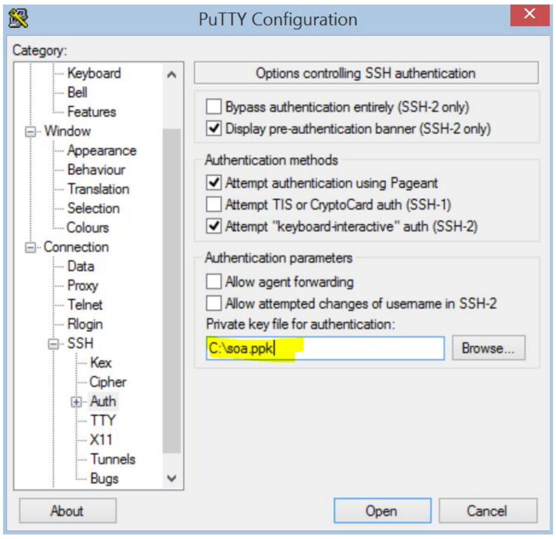
Terminal-type string: xterm

Terminal speeds: 38400,38400

Environment variables

Variable — Add

Value — Remove



PuTTY Configuration

Options controlling SSH authentication

☐ Bypass authentication entirely (SSH-2 only)
☑ Display pre-authentication banner (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant
☐ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

☐ Allow agent forwarding
☐ Allow attempted changes of username in SSH-2

Private key file for authentication: C:\soa.ppk — Browse...

Fig 3. Fig 4.

Once this is done and we select Open, we should connect to the cloud environment.
What we wanted to show here was the importance of the key, and how if you start distributing this key anybody can access your environment one they know your public IP. These keys should be kept secure.

User Management

If a user requires access to the environment, you can create another pair of public and private keys as we did above. Thereafter on the server create a new user and assign them the newly generated public key. Here we are going to be creating the user ukoug.

```
#useradd ukoug
# mkdir /home/ukoug/.ssh
```

The next step we copy and paste the public key we have generated for the new user

```
# echo "<Paste Public key here ie ssh-rsa ……..." > /home/ukoug/.ssh/authorized_keys
# cat /home/ukoug/.ssh/authorized_keys
ssh-rsa ...
# vi /etc/ssh/sshd_config
```

Add the user to AllowUsers

```
#       ForceCommand cvs server
AllowUsers  opc   otools oracle ukoug
```

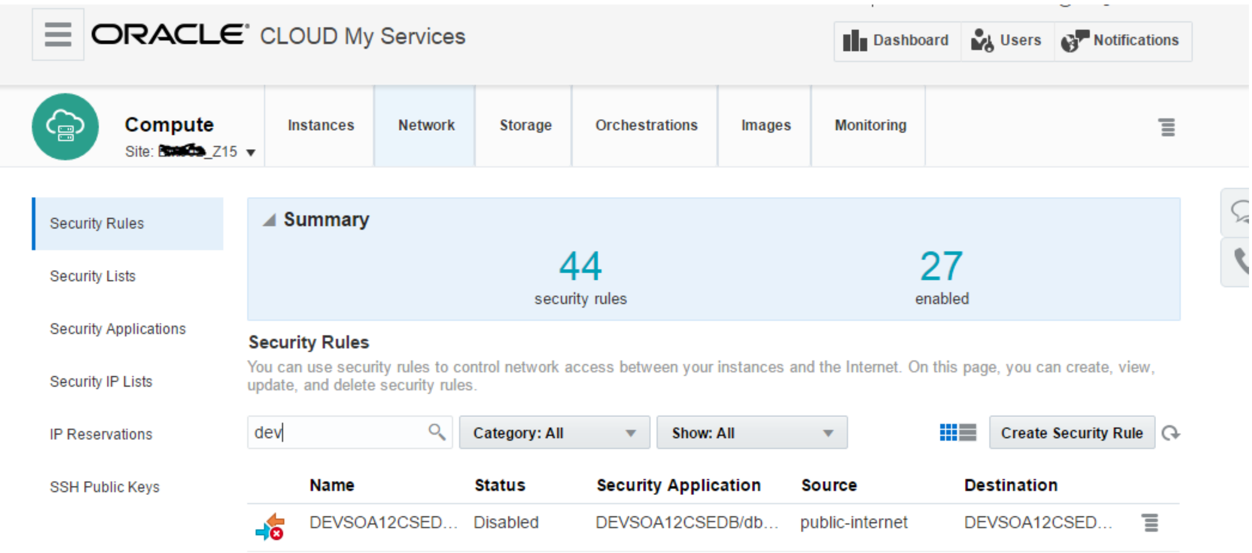
```
#chown -R ukoug:ukoug /home/ukoug/.ssh
#/sbin/service sshd restart
Stopping sshd:                                    [  OK  ]
Starting sshd:                                    [  OK  ]

#
```

The user ukoug should be able to now log in using the key we generated for them. The advantage of this is that when the user leaves or has finished their work, you can revoke their access by removing the key on the server. Here we have given a new user access to your server without compromising your master private key.

Compute Cloud Service

When you provision a server in the cloud, the server can be reached by anybody in the world if they know the public IP. Let's have a quick look at how we can control this.
Once you have provisioned servers in your dashboard you will have a list of services you can access, one is 'Compute Cloud Service'. In the header on this page select 'Network'



44 security rules

27 enabled

Security Rules

You can use security rules to control network access between your instances and the Internet. On this page, you can create, view, update, and delete security rules.

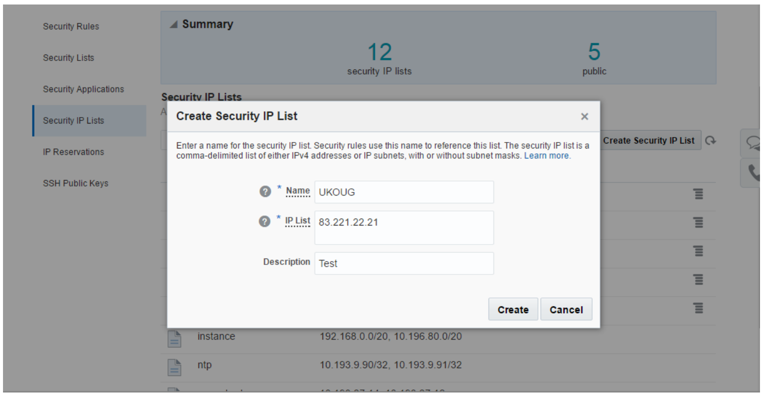
dev

Name	Status	Security Application	Source	Destination
DEVSOA12CSED...	Disabled	DEVSOA12CSEDB/db...	public-internet	DEVSOA12CSED...

Fig 5

On the left pane is a list of sub headings, best way to read this and configure is to start from the bottom SSH Public Keys and work your way up to Security Rules. What we will do here is explain these headings and at the same time set up a rule to only allow the IP 83.221.22.21 ssh access as an example.

SSH Public Keys shows you all the keys you have across your environment, with this selected you can add new and disable keys. IP reservations displays all the IP's that have been assigned to your servers, you should make sure the status is permanent otherwise on a reboot there is every chance your IP could change. Under Security IP Lists' you can create a list of IP's that you want to provide certain access rather than allowing the whole internet. In this section we create a new list and give our IP, as in Figure below.



12 security IP lists

5 public

Create Security IP List

Enter a name for the security IP list. Security rules use this name to reference this list. The security IP list is a comma-delimited list of either IPv4 addresses or IP subnets, with or without subnet masks. Learn more.

* Name: UKOUG

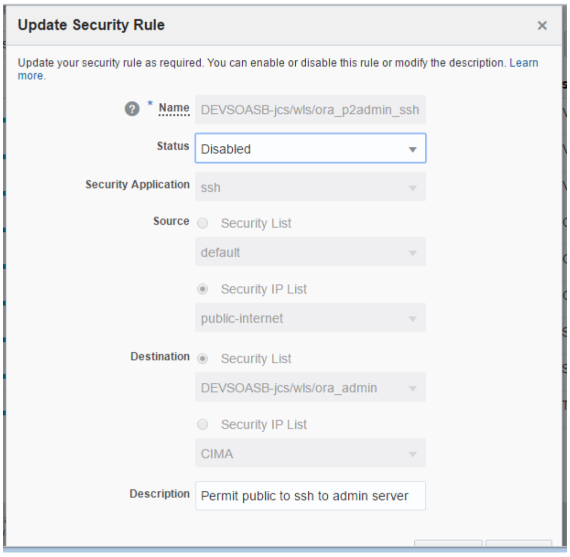
* IP List: 83.221.22.21

Description: Test

instance	192.168.0.0/20, 10.196.80.0/20
ntp	10.193.9.90/32, 10.193.9.91/32

Fig 6

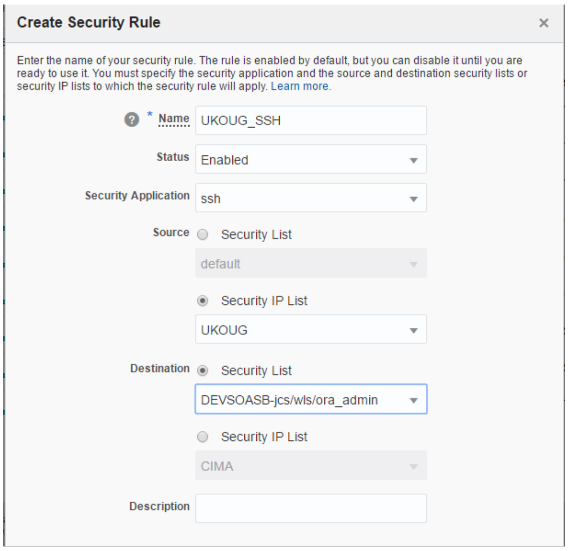
'Security Applications' – shows you all the applications and the relevant ports that the application is listening on, here you can disable/enable ports. 'Security Lists' displays the instances that are running across all the servers and that will be used when we set up security rules. The top of the tree 'Security Rules', is where everything we have looked at all comes together. We created a Security IP list called UKOUG, here we will disable the default ssh to our server and create a new rule, disabling is as simple as updating the rule to 'Disabled' Fig 7. Thereafter create a new rule as per fig 8. In the security application drop down we selected ssh, for the source we have selected the list we created earlier and the destination is the server.



Update Security Rule ×

Update your security rule as required. You can enable or disable this rule or modify the description. Learn more.

* Name	DEVSOASB-jcs/wls/ora_p2admin_ssh
Status	Disabled
Security Application	ssh
Source	○ Security List
	default
	● Security IP List
	public-internet
Destination	● Security List
	DEVSOASB-jcs/wls/ora_admin
	○ Security IP List
	CIMA
Description	Permit public to ssh to admin server



Create Security Rule ×

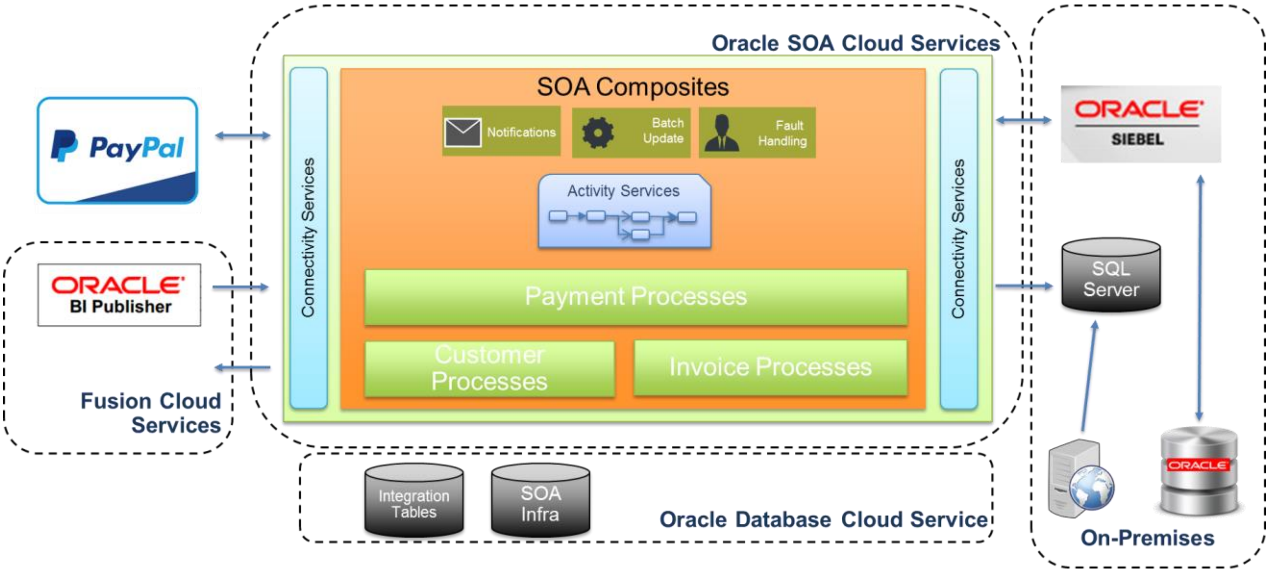
Enter the name of your security rule. The rule is enabled by default, but you can disable it until you are ready to use it. You must specify the security application and the source and destination security lists or security IP lists to which the security rule will apply. Learn more.

* Name	UKOUG_SSH
Status	Enabled
Security Application	ssh
Source	○ Security List
	default
	● Security IP List
	UKOUG
Destination	● Security List
	DEVSOASB-jcs/wls/ora_admin
	○ Security IP List
	CIMA
Description	

Fig7 Fig 8

Our Integration

Figure below shows, how our Premise and Cloud is integrated.



Contact address:

Kiran Tailor, CIMA
The Helicon, One South Place
EC2M 2RB, London

Phone: +44 (0)203 814 2421
Email kiran.tailor@cimaglobal.com
Internet: cimaglobal.com