ORACLE
Cloud Infrastructure

Bastion Hosts

Protected Access for Virtual Cloud Networks

ORACLE WHITE PAPER | FEBRUARY 2018

ORACLE

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Table of Contents

Overview

The term *bastion* comes from the fortifications that arose when cannons started dominating the battlefield. At that time, a bastion was an angularly shaped part of an outer wall, usually placed around the corners of a fort to allow defensive fire in many directions.

Similar to Medieval and Renaissance structures, computer networks need layers of protection against intruders. *Bastion hosts*, like their physical counterparts, are a part of this defensive perimeter.

Nodes deployed within Oracle Cloud Infrastructure must be assigned a public IP address to connect to the internet. Although virtual cloud network (VCN) functionality provides network security control, we suggest using a multi-tiered approach that includes bastion hosts. This paper presents best practices for bastion hosts and securing access to Oracle Cloud Infrastructure instances.

NOTE: This paper focuses mainly on Linux bastion hosts. For a Windows environments, consider Remote Desktop Gateway deployment to simplify management.

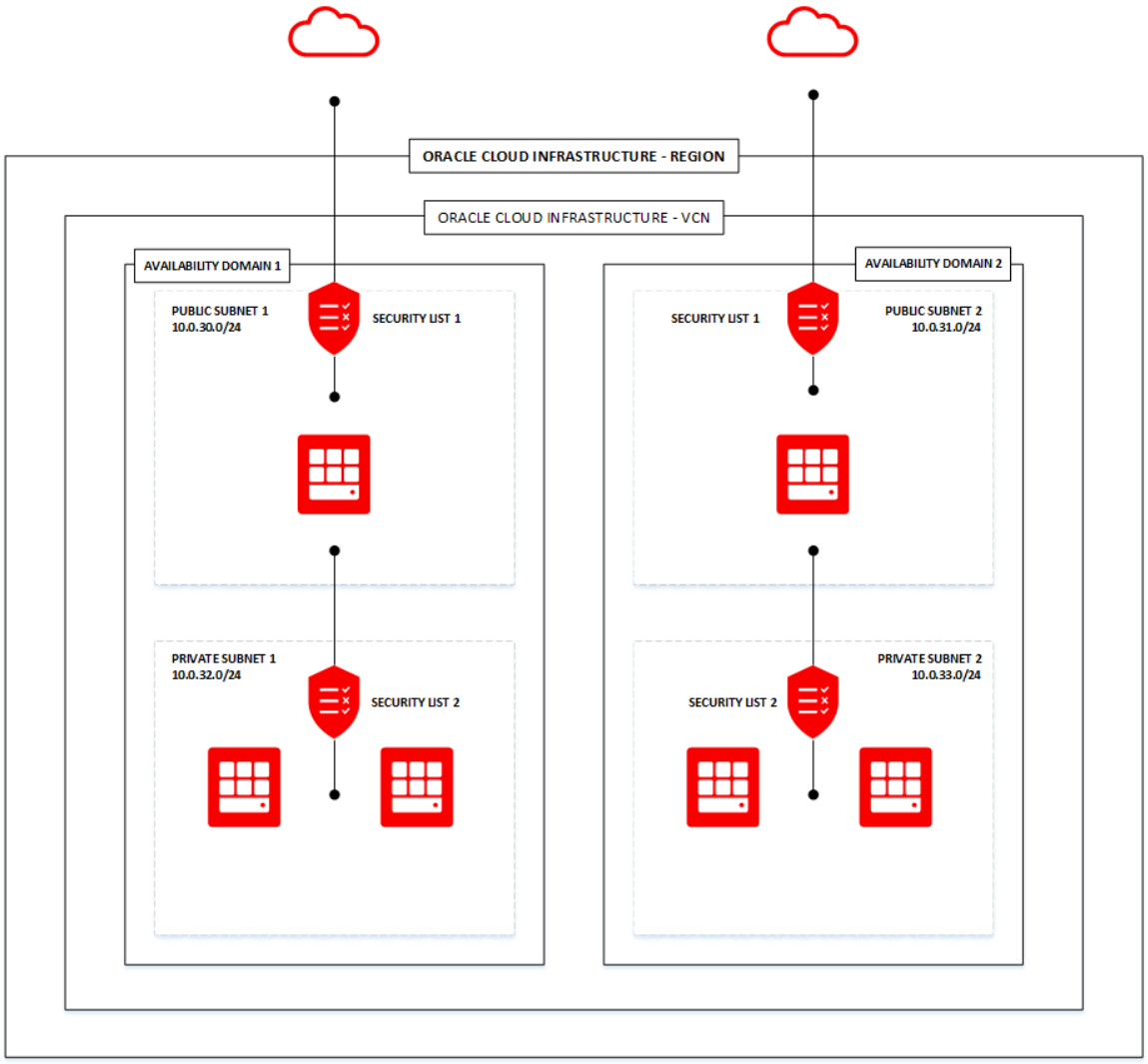
Network Security Best Practices

A multi-tiered security approach dictates network segmentation and firewall insertion at different entry points, which Oracle Cloud Infrastructure simplifies through policy configuration.

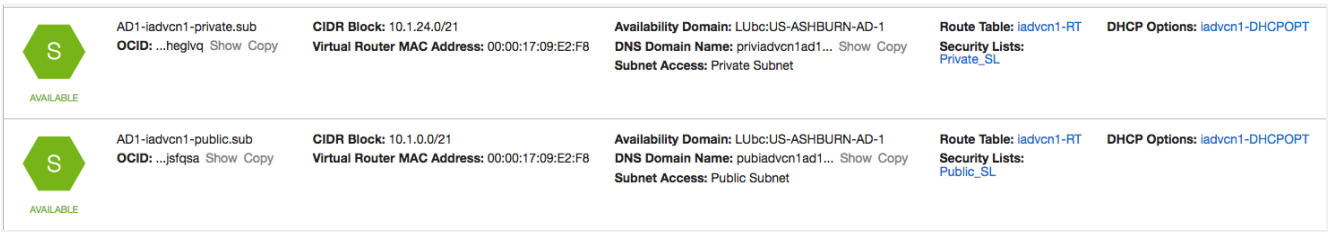
In Oracle Cloud Infrastructure, firewall rules are configured through *security lists*. Each security list can be stateless or stateful and can contain one or more rules, each rule allowing either ingress traffic or egress traffic. For each of the rules, multiple parameters are available for matching (for example, source or destination CIDR, IPv4 protocol, and port).

The example in this paper has multiple virtual hosts deployed across two availability domains and split into four subnets. Two of the subnets are public, contain bastion hosts, are configured with public IP addresses, and are connected to the internet. The remaining two subnets use private addresses, and the instances attached to each are in isolated environments.

We recommend creating a separate public subnet solely for bastion hosts to ensure that the appropriate security list is assigned to the correct host. The following diagram shows security lists configured on each segment, for fine-grained access control.



Each availability domain should be configured with a public and a private subnet, as shown in the following image:



Each subnet should be assigned the correct security list.

Security List 1 allows a particular public CIDR block of the customer network and port 22/TCP for SSH remote access to the public subnet.

INGRESS RULES FOR SECURITY LIST 1

Source	Protocol	Port
Management network CIDR	TCP	22
Management network CIDR	ICMP	Not applicable

EGRESS RULES FOR SECURITY LIST 1

Destination	Protocol	Port
0.0.0.0/0	ANY	ANY

Security List 2 allows only SSH access from the bastion hosts in the private subnet.

INGRESS RULES FOR SECURITY LIST 2

Source	Protocol	Port
Bastion Subnet AD1	TCP	22
Bastion Subnet AD2	TCP	22
Bastion Subnet AD3	TCP	22
Bastion Subnet AD1	ICMP	Not applicable
Bastion Subnet AD2	ICMP	Not applicable
Bastion Subnet AD3	ICMP	Not applicable

EGRESS RULES FOR SECURITY LIST 2

Destination	Protocol	Port
0.0.0.0/0	ANY	ANY

Each Linux or Windows host image provided by Oracle also includes a preconfigured and enabled host firewall. Those rules need to be modified to match the security groups.

On Oracle Linux, iptables can be managed using a `firewallcmd` command.

Using ssh-agent to Connect Through the Bastion Host

Because most of the infrastructure denies remote access, a method is needed for logging in to the servers located in the private subnets. Point-to-network VPN can be established, but that increases the complexity and management necessary for the setup. One method that is both secure and convenient is to connect to the bastion hosts by using the SSH protocol.

By default, access to the server is configured to use only SSH public key authentication. We recommend using `ssh-agent` instead of storing SSH keys (especially without a passphrase) on the bastion hosts. This way, private SSH keys exist only on your computer and can be safely used to authenticate to the next server.

To add a key to the authentication agent, use the `ssh-add` command. If the key is `~/.ssh/id_rsa`, it's added automatically. You can also specify which key to use by running the following command:

```
$ ssh-add [path_to_keyfile]1
```

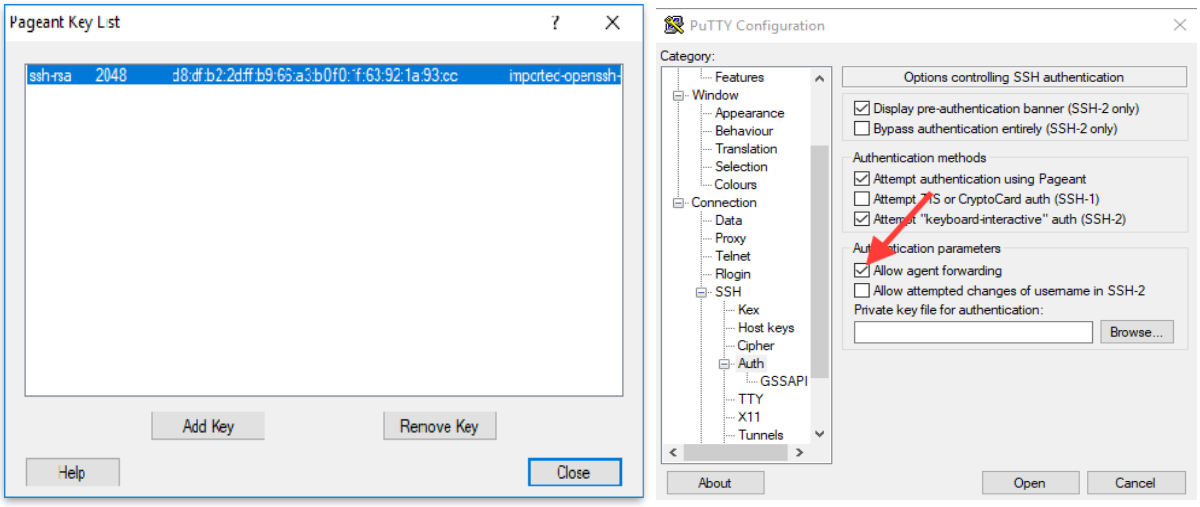
Mac OS X users can configure the `~/.ssh/config` file to enable loading keys into the agent:

```
AddKeysToAgent yes
```

Using the following command to connect to the bastion host enables agent forwarding and allows logging in to the next server by forwarding credentials from your local machine:

```
$ ssh -A opc@bastion_host
```

Windows users should use the Pageant application and import their private key file there, and then enable agent forwarding by selecting **Connection**, then **SSH**, and then **Auth** in the PuTTY Configuration window.



Although the forwarded key could be exploited by an attacker on the remote host to initiate new connections, the key itself is secure. You can enable additional protection by using the confirmation feature in `ssh-agent`.

Although the Mac OS X SSH implementation ships without the `/usr/libexec/ssh-askpass` command, multiple open-source projects provide a viable workaround.

To simplify SSH access and configuration, add the `-J` (ProxyJump) parameter to the `ssh` command. Following is an example of ProxyJump usage:

```
$ ssh -J opc@Bastion-1.oraclecloud.com opc@server2.oraclecloud.com
```

As a result, the SSH client will automatically connect to `server2.oraclecloud.com`.

If you're using an older SSH client, ProxyJump is not available. Instead, you can use ProxyCommand to achieve the same result, using the `stdio` forwarding mode to proxy connect through the remote host.

```
$ ssh -o ProxyCommand="ssh -W %h:%p opc@bastion-1.oraclecloud.com"
opc@server2.oraclecloud.com
```

This approach also helps to achieve port forwarding without any other required configuration.

On Windows system, this can be accomplished using PuTTY SSH configuration and the Remote command window when agent forwarding is enabled, as described previously. Enter `ssh opc@<secure_server_private_ip>` or specify the local SSH key on the bastion host by using the `-i` parameter.

Service Access Through SSH Tunneling

Sometimes SSH access might not be enough to perform the task. In this case, SSH tunneling can provide an easy way to access a web application or other listening service.

The main types of SSH tunneling are *local*, *remote*, and *dynamic*.

- The local tunnel provides an exposed port on the local loopback interface that is connected to the `IP:port` from your SSH server.

  For example, you can connect local port 8080 to `web_server_ip:80` that is accessible from your bastion host and point your web browser to `http://localhost:8080`:

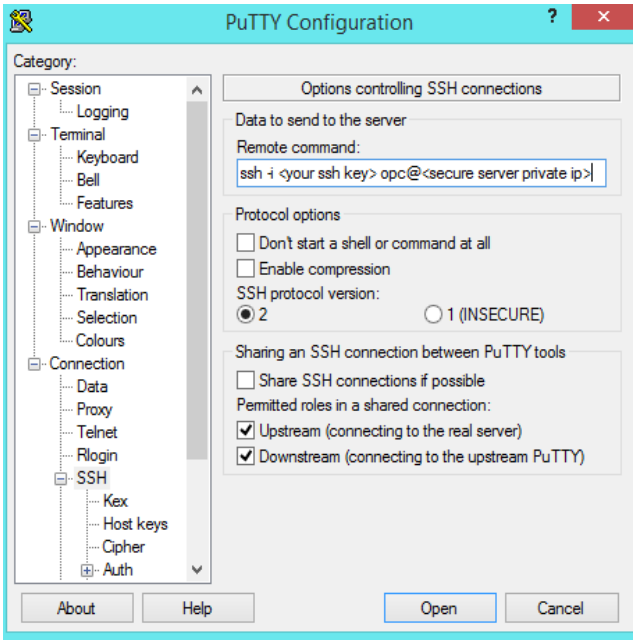
  ```
  $ ssh opc@bastion_host -L 8080:web_server_ip:80
  ```

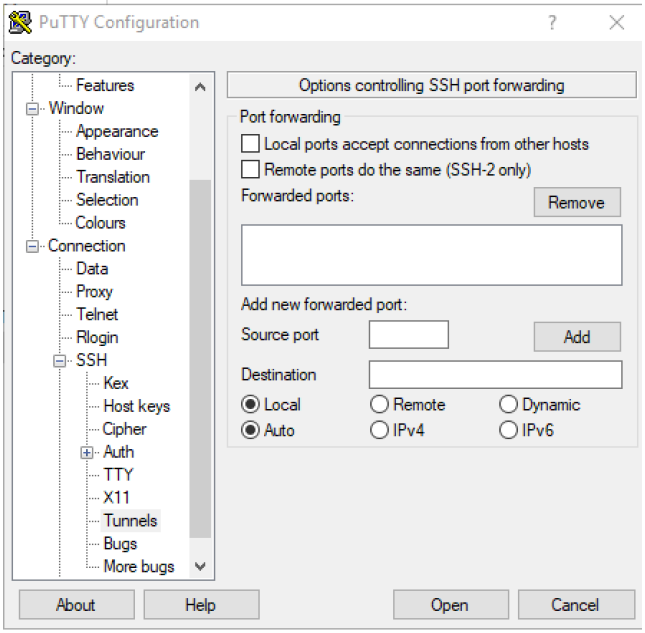
- The remote tunnel is outside of the scope of this tutorial, but it works the opposite of local forwarding: it exposes a local port to connections coming to the remote server.
- The dynamic tunnel provides a SOCKS proxy on the local port, but connections originate from the remote host. For example, you can set up a dynamic tunnel on port 1080 and configure it as SOCKS proxy in the web browser. As a result, you can connect to all the resources available from your bastion host that are in the private subnet.

```
$ ssh opc@bastion_host -D 1080
```

Those techniques are a simpler replacement that in many cases would require VPN connection and can be combined with ProxyJump or ProxyCommand connections.

Windows users can find the tunnel configuration in PuTTY by selecting **Connection**, then **SSH**, then **Tunnels**, as shown in the following images:





Port forwarding, especially a local one, can be used to easily establish the connection to Remote Desktop Services–enabled Windows hosts in the cloud, by tunneling port 3389 and connecting to localhost from a Remote Desktop client. If RDS is already listening on the local machine, you can select another port, as shown in the following example:

```
$ ssh opc@bastion_host -L 3390:windows_host:3389
```

File Transfers

For the Linux client and servers, you can use SCP to securely transfer files to and from hosts through the bastion host by using the same ProxyCommand or ProxyJump options specified from the SSH command line. For example:

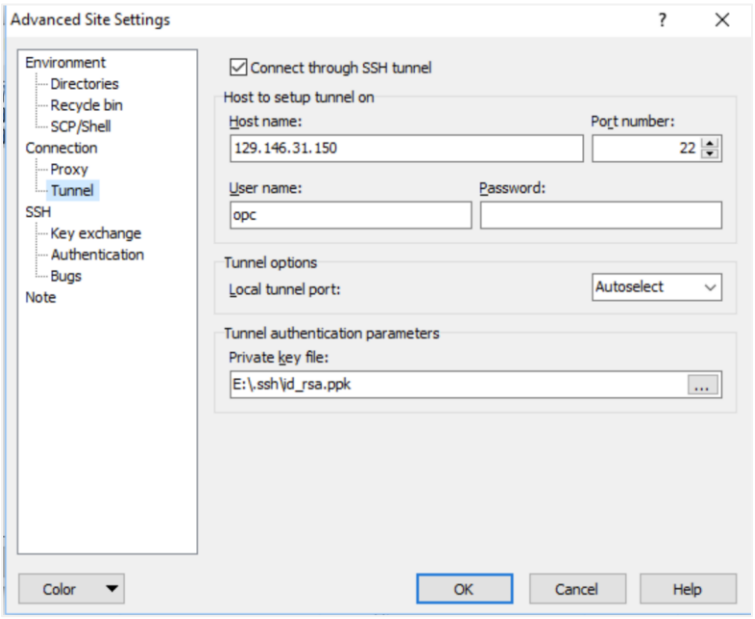
```
$ scp -o "ProxyJump opc@bastion_host" filename opc@private_host:/path/to/file
```

If you're using a Windows client, one of the most popular application for SCP is WinSCP. To transfer the files through the bastion host to a remote Linux instance, follow these steps:

1. Create a session with a private host IP address without a password (since the Linux instance will be configured with the SSH key).

2. Click **Advanced**, and select **Tunnel** from the left navigation menu.

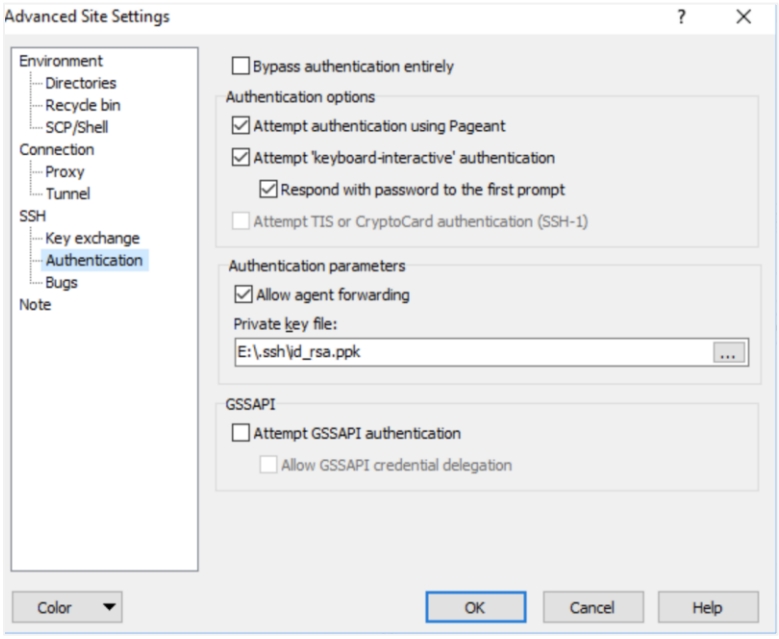
3. Enter your bastion host IP address and username. In the **Private key file** field, navigate and select the private key that will be used to authenticate with the bastion host.



4. In the left navigation menu, select **Authentication** (under **SSH**).

5. Ensure that **Allow agent forwarding** is selected.

6. Select the private key that will be used to authenticate with the private host. In this example, it's the same key but it doesn't have to be; you might want to use multiple keys for added security.



This setup allows direct file transfer between your Windows machine and Linux private host, protected by bastion.

For Windows hosts behind a Linux bastion, you can transfer files by using Remote Desktop Protocol (RDP) and tunneling. This is an effective and secure method of transferring files.

Bastion Gateway

You can also create a bastion gateway that provides web-based access to the servers behind it.

Multiple software solutions can deliver an SSH web console, such as shellinabox, KeyBox, or Apache Guacamole. The Guacamole project also provides access to Windows hosts using VNC and RDP, as well as a file transfer interface, remote disk functionality, and even remote sound and printing support.

Bastion gateway software provides easier access (especially from mobile devices), can be deployed using any popular web server application (such as Nginx or Apache), and can be launched in the container using LXC or Docker.

Conclusion

Bastion hosts are an important part of the network security layer for both cloud and data center deployments. Combined with firewall policies, bastion hosts can protect your environment from external access to management interfaces.

Although VPN can be used to access internal networks, bastion hosts are simpler to deploy, easier to operate, and have significantly less management overhead.

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Integrated Cloud Applications & Platform Services

Bastion Hosts: Protected Access for Virtual Cloud Networks
February 2018