Bastion Hosts: Protected Access for Virtual Cloud Networks

August 2021, version 2.0

Public

Disclaimer

This document in any form, software, or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. 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 in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Revision History

The following revisions have been made to this document since its initial publication.

DATE	REVISION
August 2021	Updated the template and edited for clarity
February 2020	• Added a note about ensuring secured access (top of page 4) • Added figure titles and alt text for accessibility
February 2018	Initial publication

Table of Contents

Security note: When following the guidance in this paper, pay particular attention to ensuring secured access for administrators, as with all users. Follow best practices, including limiting privileges to only users who require them, enforcing strong authentication including multi-factor authentication, and planning for the security of the device and the physical location from which these connections are originating.

Overview

The term bastion comes from the fortifications that arose when cannons started dominating the battlefield. Then, a bastion was an angularly shaped part of an outer wall, placed around the corners of a fort to allow defensive fire in many directions. Like 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 (OCI) 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 OCI instances.

Note: This paper focuses on Linux bastion hosts. For a Windows environment, consider Remote Desktop gateway deployment to simplify management.

Network Security Best Practices

A multitiered security approach dictates network segmentation and firewall insertion at different entry points, which OCI simplifies through policy configuration.

In OCI, 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, such as source or destination CIDR, IPv4 protocol, and port.

The example in this paper has multiple virtual hosts deployed across two availability domains an d 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 subnet, for fine-grained access control:

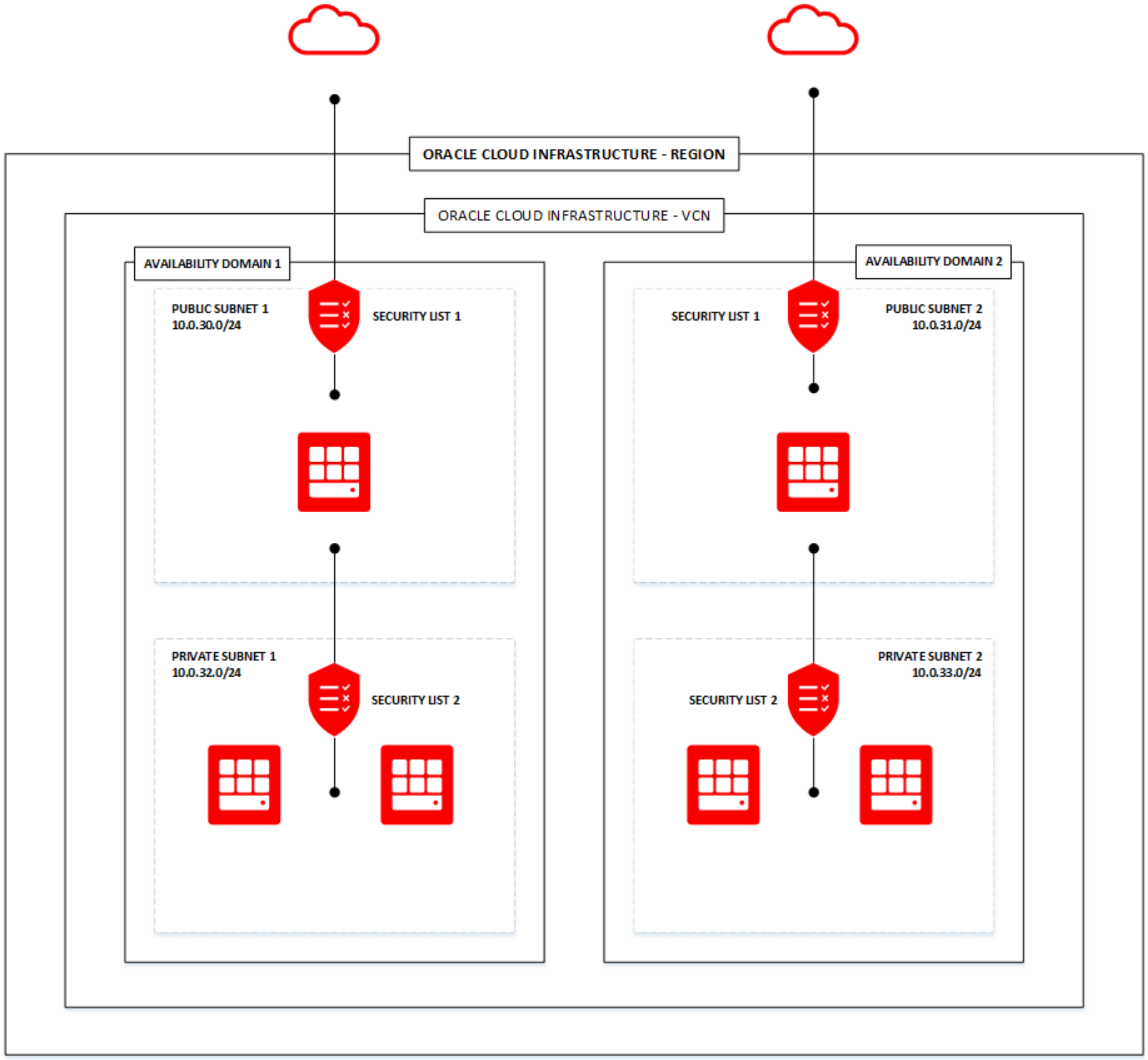
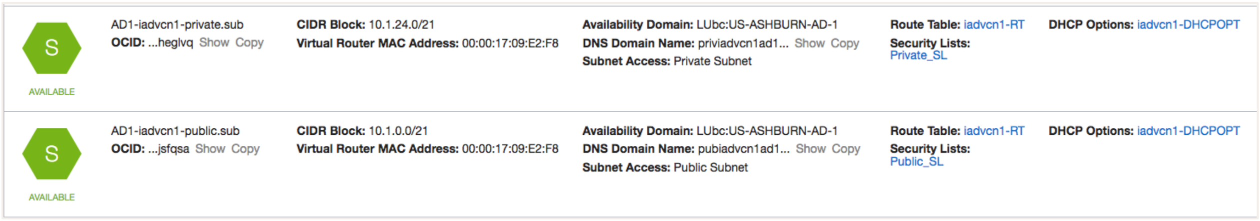


Figure 1: Security Lists on Each Subnet

Configure each availability domain with a public and a private subnet, as shown in the following image:



S AVAILABLE	AD1-iadvcn1-private.sub **OCID:** ...heglvq Show Copy	**CIDR Block:** 10.1.24.0/21 **Virtual Router MAC Address:** 00:00:17:09:E2:F8	**Availability Domain:** LUbc:US-ASHBURN-AD-1 **DNS Domain Name:** priviadvcn1ad1... Show Copy **Subnet Access:** Private Subnet	**Route Table:** iadvcn1-RT **Security Lists:** Private_SL	**DHCP Options:** iadvcn1-DHCPOPT
S AVAILABLE	AD1-iadvcn1-public.sub **OCID:** ...jsfqsa Show Copy	**CIDR Block:** 10.1.0.0/21 **Virtual Router MAC Address:** 00:00:17:09:E2:F8	**Availability Domain:** LUbc:US-ASHBURN-AD-1 **DNS Domain Name:** pubiadvcn1ad1... Show Copy **Subnet Access:** Public Subnet	**Route Table:** iadvcn1-RT **Security Lists:** Public_SL	**DHCP Options:** iadvcn1-DHCPOPT

Figure 2: Public and Private Subnet Configurations

Assign the correct security list to each subnet.

- **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. Modify those rules 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 in the private subnets. You can establish point-to-network VPN, but it increases the complexity and management necessary for the setup. A secure and convenient method 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
```

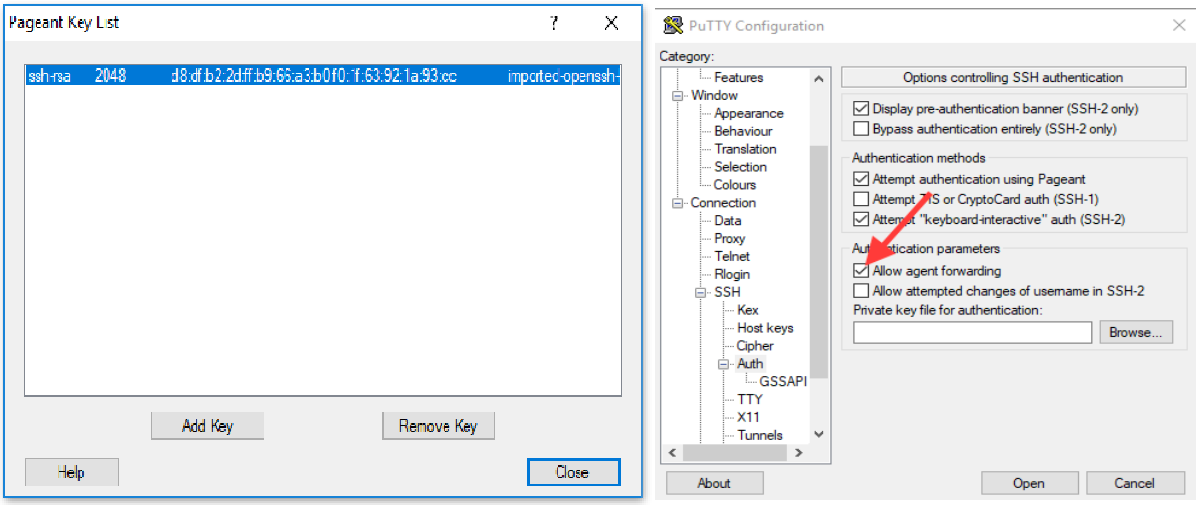
Mac OS X users can configure the `~/.ssh/config` file to enable loading keys into the agent with the following command:

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



ssh-rsa 2048 d8:df:b2:2d:ff:b9:66:a3:b0:f0:1f:63:92:1a:93:cc imported-openssh-

Options controlling SSH authentication

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

Figure 3: Using Pageant and PuTTY to Allow Agent Forwarding

Although an attacker can exploit the forwarded key on the remote host to initiate new connections, the key itself is secure. You can enable extra 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 automatically connects 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 a Windows system, this can be accomplished by 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's connected to the `IP:port` from your SSH server.

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

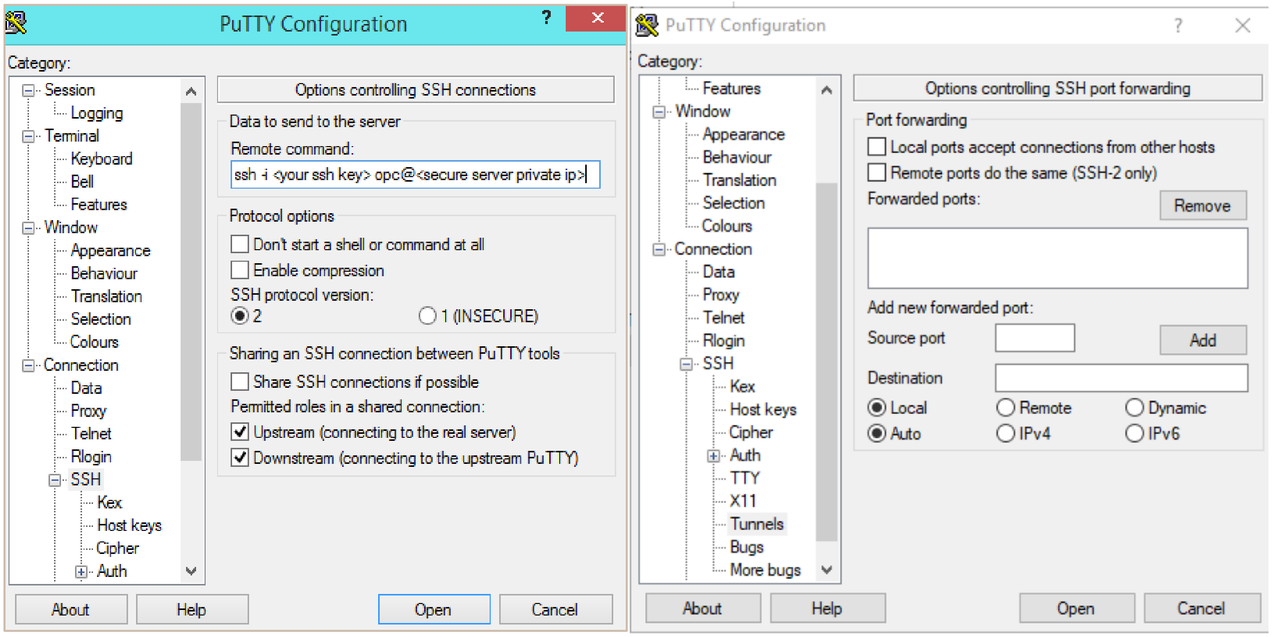
  ```
  $ SSH opc@bastion_host -L 8080:web_server_ip:80
  ```

- The **remote** tunnel is outside 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
  ```

These techniques are simple replacements that often require a VPN connection and can be combined with ProxyJump or ProxyCommand connections.

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



Options controlling SSH connections

Data to send to the server

Remote command:

ssh -i <your ssh key> opc@<secure server private ip>

Protocol options

□ Don't start a shell or command at all

□ Enable compression

SSH protocol version:

● 2 ○ 1 (INSECURE)

Sharing an SSH connection between PuTTY tools

□ Share SSH connections if possible

Permitted roles in a shared connection:

☑ Upstream (connecting to the real server)

☑ Downstream (connecting to the upstream PuTTY)

Options controlling SSH port forwarding

Port forwarding

□ Local ports accept connections from other hosts

□ Remote ports do the same (SSH-2 only)

Forwarded ports: Remove

Add new forwarded port:

Source port Add

Destination

● Local ○ Remote ○ Dynamic

● Auto ○ IPv4 ○ IPv6

Figure 4: Configuring Tunneling in Windows

You can use port forwarding, especially a local one, 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
```

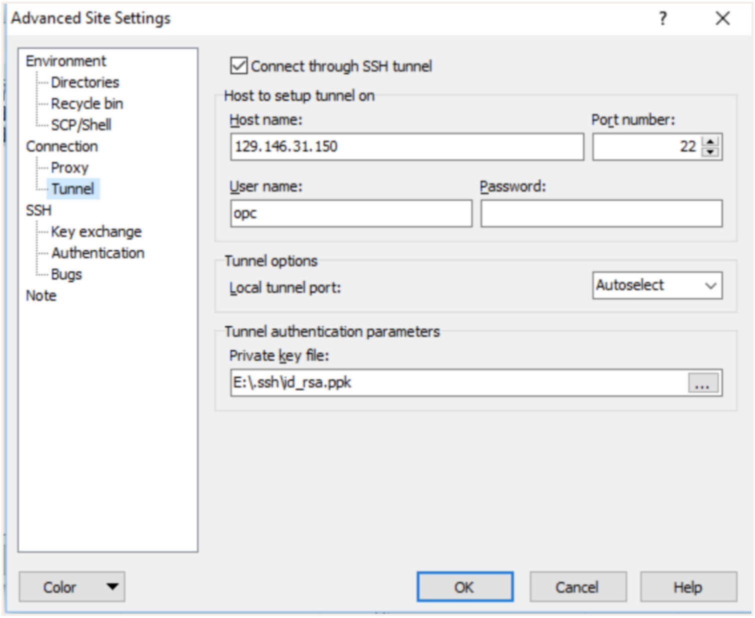
File Transfers

For a Linux client and servers, you can use secure copy protocol (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 applications for SCP is WinSCP. To transfer the files through the bastion host to a remote Linux instance, use the following steps:

1. Create a session with a private host IP address without a password. The Linux instance is configured with the SSH key.
2. In the left navigation menu, click **Advanced** and select **Tunnel**.
3. Enter your bastion host IP address and username. In the **Private key file** field, navigate and select the private key to authenticate with the bastion host.



☑ Connect through SSH tunnel

Host name: 129.146.31.150

Port number: 22

User name: opc

Password:

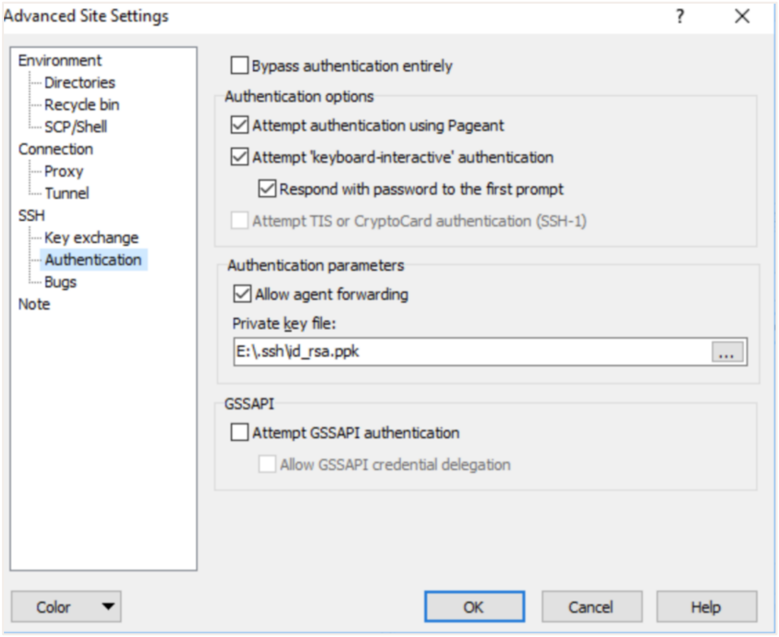
Local tunnel port: Autoselect

Private key file: E:\.ssh\id_rsa.ppk

Figure 5: Selecting a Private Key to Authenticate with the Bastion Host

4. In the left navigation menu, under **SSH**, select **Authentication**.
5. Ensure that **Allow agent forwarding** is selected.

6. Select the private key that authenticates with the private host. This example uses the same key, but you can use multiple keys for added security.



□ Bypass authentication entirely

Authentication options

☑ Attempt authentication using Pageant

☑ Attempt 'keyboard-interactive' authentication

☑ Respond with password to the first prompt

□ Attempt TIS or CryptoCard authentication (SSH-1)

Authentication parameters

☑ Allow agent forwarding

Private key file:

E:\.ssh\id_rsa.ppk

GSSAPI

□ Attempt GSSAPI authentication

□ Allow GSSAPI credential delegation

Figure 6: Selecting a Private Key to Authenticate with the Private Host

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

For Windows hosts behind a Linux bastion, you can transfer files by using Remote Desktop Protocol (RDP) and tunneling. This method is effective and secure for 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, RDP, 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 you can use a VPN to access internal networks, bastion hosts are simpler to deploy, easier to operate, and have significantly less management overhead.

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

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