Create the SFTP private key on one of the Linux machines using the following command to generate the keys:

ssh-keygen -f sftp-server-key

Once the key generation process is complete, the file will be generated in the specified location. Alternatively, you can search for it using the name "sftp-server-key." Copy the contents of the "sftp-server-key.pub" file and paste it into the server's authorized keys location.

Note: When creating an SSH key, refrain from setting any passwords for those keys.

Convert the private key file to RSA format before copying it. By default, the private key file is in the OpenSSH format, and it needs to be modified to PEM format. Utilize the following command to convert it:

ssh-keygen -p -m PEM -f sftp-server-key

This command will overwrite the file in the same location, so ensure that the keys are updated with the latest timestamp. If the key is already in the RSA format, avoid performing any conversion.

Next, convert the key into a base64-encoded format for storage in the secret manager. Execute the following command to convert the content from the private key file, giving it a different name:

base64 -w 0 < sftp-server-key > sftp-server-key-base64

Save the file and provide it to the person responsible for handling the secret manager. They will then be able to share the credentials.

Additionally, ensure to provide the following information to the concerned party:

1. SFTP Server Hostname: [insert hostname here]
2. Username Details: [insert username details here]

Pgp Encryption and Decryption key Generation

In this document, we have outlined the process of creating decryption and encryption key pairs using GPG (GNU Privacy Guard) for data security. To begin, we generate a private key for decryption by running the command:

- gpg –gen-key

  # username bca
  # user email address bca@pm.me

During this step, we provide essential information such as the username and email address. Afterward, we can list the secret and public keys, if needed, with the following command:

- gpg --list-secret-keys --keyid-format=long
- gpg --list-keys

Following this, we create a 'gpg' directory to store the private key file and navigate to it using the commands:

- mkdir gpg
- cd gpg

To export the private key for safekeeping, we execute the command:

- gpg --export-secret-keys -a "bca" > bca-private-key.asc

Note:

Additionally, we can import the private key on other systems if required with the command:

- gpg --import bca-private-key.asc

For encryption, we use the public key associated with an email address and apply it to the desired file, 'demo.txt', for example:

- gpg --recipient "bca@pm.me" --encrypt demo.txt

To create a public key for encryption, we list the available keys and export the specific public key into a designated file, 'januo-public-key.asc', using the command:

- gpg --list-keys
- gpg --armor --export 9A35AFFC9C70CB43D160343C37A89C98857A7D57 > bca-public-key.asc

It is vital to maintain utmost security and regular backups of private keys to ensure data confidentiality and integrity throughout the process.

Next, convert the key and public key into a base64-encoded format for storage in the secret manager. Execute the following command to convert the content from the private key file, giving it a different name:

- base64 -w 0 < januo-public-key.asc > bca-public-key-base64
- base64 -w 0 < januo-private-key.asc > bca-private-key-base64

Save the file and provide it to the person responsible for handling the secret manager.

Additionally, ensure to provide the following information to the concerned party:

1. Agent name: [insert agent name here]
2. Agent receipt email address: [insert receipt email address here]