How to Upload an SSH Public Key to an Existing Droplet

Validated on 15 April 2019 • Posted on 19 June 2018

DigitalOcean Droplets are Linux-based virtual machines (VMs) that run on top of virtualized hardware. Each Droplet you create is a new server you can use, either standalone or as part of a larger, cloud-based infrastructure.

For security reasons, you can't add or modify the SSH keys on your Droplet using the control panel after you create it, but you have several options to add and modify them via the command line. If you currently have SSH access to the Droplet, you can upload keys:

- **From your local computer using** `ssh-copy-id`, which is included in many Linux distributions' OpenSSH packages.
- **From your local computer by piping the contents of the key** into the `~/.ssh/authorized_keys` file. This is a good choice if you don't have `ssh-copy-id`.
- **By SSHing to your Droplet and adding the public key manually**, which is necessary if you do not have password-based SSH access.

If you currently can't connect to your Droplet at all, use the Droplet console to reset the root user password. Once logged in on the console, you can either add your key manually from the console or temporarily enable password authentication to add the key via SSH.

If you have password-based access to your Droplet, you can copy your SSH key using `ssh-copy-id`. Substitute the IP address of your Droplet.

```
ssh-copy-id username@203.0.113.0
```

This prompts you for the user account's password on the remote system:

```
    The authenticity of host '203.0.113.0 (203.0.113.0)' can't be established.
    ECDSA key fingerprint is fd:fd:d4:f9:EX:AM:PL:E0:e1:55:00:ad:d6:6d:22:fe.
➥
    /usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any t
    /usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it
➥
```

After typing in the password, the contents of your `~/.ssh/id_rsa.pub` key are appended to the end of the user account's `~/.ssh/authorized_keys` file:

```
Number of key(s) added: 1

Now try logging in to the machine, with:   "ssh 'username@203.0.113.0'"
and check to make sure that only the key(s) you wanted were added.
```

After entering the password, it copies your key, and you can log in without a password.

Locally by Piping into ssh with Password-Based Access

If you do not have the `ssh-copy-id` utility available, but still have password-based SSH access to the remote server, you can pipe the contents of the key into the `ssh` command.

On the remote side, make sure the `~/.ssh` directory exists, and then append the piped contents into the `~/.ssh/authorized_keys` file. Substitute the IP address and your username for your Droplet.

```
    cat ~/.ssh/id_rsa.pub | \
➥
```

```
    The authenticity of host '203.0.113.0 (203.0.113.0)' can't be established.
    ECDSA key fingerprint is fd:fd:d4:f9:EX:AM:PL:E0:e1:55:00:ad:d6:6d:22:fe.
➡
➡
```

After entering the password, it copies your key, and you can log in without a password.

Manually from the Droplet without Password-Based Access

If you do not have password-based SSH access available, you have to add your public key to the remote server manually.

> **Note**
>
> These instructions assume you're connecting to your Droplet with a terminal and SSH without password access. If you can't connect using those methods, you can use the Droplet console to recover access by resetting your Droplet's root password, and then use `ssh` to add your keys.

On your local machine, output the contents of your public key.

```
cat ~/.ssh/id_rsa.pub
```

Copy the output.

```
ssh-rsa EXAMPLEzaC1yc2EAAAADAQABAAACAQCqql6MzstZYh1TmWWv11q5O3pISj2ZFl9HgH1JLknLLx44+tXfJ7mI
```

Log in to your Droplet using your local terminal and create the `~/.ssh` directory if it does not already exist:

```
mkdir -p ~/.ssh
```

Create and open the `~/.ssh/authorized_keys` file for editing using a terminal-based text editor, like `nano`.

```
nano ~/.ssh/authorized_keys
```

Paste the contents of your SSH key into the file by right-clicking in your terminal and choosing **Paste** or by using a keyboard shortcut like `CTRL+SHIFT+V`. Then, save and close the file. In `nano`, save by pressing `CTRL+O` and then `ENTER`, and exit by pressing `CTRL+X`.

Alternatively, instead of opening the file in an editor and pasting your key, you can create the `authorized_keys` file with your public key added with a single command. If you use this, make sure to substitute the contents of your public key into the `echo` command.

```
echo "ssh-rsa EXAMPLEzaC1yc2E...GvaQ== username@203.0.113.0" \
>> ~/.ssh/authorized_keys
```

Once the `authorized_keys` file contains the public key, you need to update permissions on some of the files. The `~/.ssh` directory and `authorized_keys` file must have specific restricted permissions (`700` for `~/.ssh` and `600` for `authorized_keys`). If they don't, you won't be able to log in.

Make sure the permissions and ownership of the files are correct.

```
chmod -R go= ~/.ssh
chown -R $USER:$USER ~/.ssh
```

You can now log out of your Droplet. The next time you log in, you can do so without a password.