(../..)

Purpose of this lab

- How to view application security groups
- How to view the details of an application security group
- What commands that administrators can use to manage security groups
- How security group rules can impact your application

Estimated Time: 10 minutes

Review Application Security Groups

1. Review the documentation (http://docs.pivotal.io/pivotalcf/adminguide/app-sec-groups.html) on application security groups (https://docs.pivotal.io/pivotalcf/concepts/security.html#network-traffic).

2. List the security groups in your environment.

```
cf security-groups
```

3. Pivotal Cloud Foundry ships with multiple security groups. Pick two security groups from the list and note the differences in the details of those security groups.

```
cf security-group public_networks
```

Questions

- Do application security groups use a whitelist or blacklist approach to firewall rules?
- What are the differences in the two application security groups that you looked at?
- What are some reasons why security groups could be used?

- How could security groups affect the staging and running of your application?

Administration of application security groups

The cf CLI exposes a dozen or so commands for `admin` users to use to create and manage application security groups. The rest of this lab provides an overview.

NOTE: Unless you're an administrator of the PCF instance you're working with, you won't be able to experiment with these commands. Nevertheless it's important that you be aware of them.

Managing ASGs

When logged in as `admin`, the command `cf security-groups` lists all security groups:

```
cf security-groups
```

The following commands support managing this list:

```
cf create-security-group
cf update-security-group
cf delete-security-group
```

The *create* and *update* commands require that you supply the security group definition as a JSON file that encodes a list of egress rules.

The creation of a security group is only a first step. Additional commands exist to control which security groups are applied in different situations.

Binding ASGs to Running Applications

The command `cf running-security-groups` lists all security groups that apply to all running applications:

```
cf running-security-groups
```

Use these commands to apply (or remove) a security group to all running applications:

```
cf bind-running-security-group all_open
cf unbind-running-security-group all_open
```

Binding ASGs to Application Staging

Security groups can also control the behavior of scripts that stage an application. To list all security groups that apply to the staging of applications, use the command:

```
cf staging-security-groups
```

Use these commands to apply (or remove) a security group to the application staging phase:

```
cf bind-staging-security-group all_open
cf unbind-staging-security-group all_open
```

Org and Space specific bindings

The *running-security-groups* binding is coarse-grained; it applies to all running applications in a PCF installation. For finer-grained control of security groups to specific running applications, these commands can scope the binding to a specific org or space:

```
cf bind-security-group all_open my-org my-space
cf unbind-security-group all_open my-org my-space
```

Questions

- What are the differences between the different types (running and staging) of security groups?

Imagine a situation in which an application is bound to a backing database service and deployed to PCF.

- In the absence of ASG bindings, would the application be able to communicate with the backing service?

Let's assume that the backing database is a MySQL instance, which by default accepts connections on port 3306.

- As a Cloud Foundry administrator, what would you have to do in order to allow *any* running application to communicate with this type of backing service?
- What would your security group rules JSON file look like? Should the rules allow all ports over all ip addresses? Can the rules be constrained to allow only port 3306 over a specific range of ip addresses? Try to construct a security group rules file (JSON) that you'd use to create and then

bind to running applications. What command would you use to define the security group? What would you name it? What command would you use to bind it?

(https://pivotal.io)