### **Create Cloud Config**

The [cloud config](https://bosh.io/docs/cloud-config.html) is a YAML file that defines IaaS specific configuration used by the Director and all deployments. It allows us to separate IaaS specific configuration into its own file and keep deployment manifests IaaS agnostic.

Now we are going to extract IaaS specific information from our greeter release and put it into the cloud config. We also will use the cloud config in all further deployments. Our initial cloud config will look like the following. Lets save this file as ~/deployment/cloud-config.yml

azs:  
- name: z1  
 cloud\_properties: {availability\_zone: REPLACE\_WITH\_AZ}  
vm\_types:  
- name: t2.small  
 cloud\_properties:  
 instance\_type: t2.small  
- name: t2.large  
 cloud\_properties:  
 instance\_type: t2.large  
 ephemeral\_disk: {size: 30000, type: gp2}  
 root\_disk: {size: 50000, type: gp2}  
  
networks:  
- name: private  
 type: manual  
 subnets:  
 - range: 10.0.0.0/24  
 gateway: 10.0.0.1  
 az: z1  
 static: [10.0.0.7 - 10.0.0.10]  
 reserved: [10.0.0.2 - 10.0.0.6]   
 dns: [10.0.0.2]  
 cloud\_properties:   
 subnet: $REPLACE\_WITH\_SUBNET\_ID  
 security\_groups: [app, bosh]  
- name: public  
 type: vip  
  
compilation:  
 workers: 5  
 reuse\_compilation\_vms: true  
 az: z1  
 vm\_type: t2.large  
 network: private

Next we need to upload cloud config to the Director.

bosh -n update-cloud-config ~/deployment/cloud-config.yml

### **Update deployment manifest**

When using cloud config manifest schema is different. See [here](https://bosh.io/docs/manifest-v2.html) for details. Manifest for our greeter-release will look like the following. You need to save this file as ~/deployment/greeter.v2.0.yml

name: greeter-release  
  
releases:  
- name: greeter-release  
 version: latest  
  
instance\_groups:  
- name: app  
 instances: 1  
 azs: [z1]  
 vm\_type: t2.small  
 stemcell: ubuntu  
 jobs:  
 - name: app  
 properties: {}  
 networks:  
 - name: private  
 static\_ips:   
 - 10.0.0.7  
  
- name: router  
 instances: 1  
 azs: [z1]  
 vm\_type: t2.small  
 stemcell: ubuntu  
 jobs:  
 - name: router  
 properties:  
 upstreams:  
 - 10.0.0.7:8080  
 networks:  
 - name: private  
 static\_ips:  
 - 10.0.0.8  
 default: [dns, gateway]  
 - name: public  
 static\_ips:  
 - $REPLACE\_WITH\_ELASTIC\_IP  
  
stemcells:  
- alias: ubuntu  
 os: ubuntu-trusty  
 version: $REPLACE\_WITH\_STEMCELL\_VERSION  
update:  
 canaries: 1  
 canary\_watch\_time: 30000  
 update\_watch\_time: 30000  
 max\_in\_flight: 10  
 max\_errors: 1

The following changes were made:

1.compilation and networks sections were moved to cloud config.

1. stemcells section was added
2. jobs section now becomes instance\_groups and following changes also were applied
3. azs property, that is responsible for configuring availability zoned were added.
4. former templates block now is renamed to jobs
5. properties now are defined as a subsection of jobs They should be defined for each job separately
6. Instead of resource\_pool we now specify vm-type and stemcell

### **Redeploy greeter release**

Before we will be able to redeploy greeter release using cloud config, we need to delete the old deployment.

bosh -n -d greeter-release delete-deployment

Now lets redeploy our greeter release

bosh -n -d greeter-release deploy ~/deployment/greeter.v2.0.yml

You can check if everything has been deployed as intended:

curl "http://$REPLACE\_WITH\_ELASTIC\_IP:8080"