



# Puppet modules for AWS and Docker

DevOps Practitioner

transforming performance  
through learning

## Outline

- **Puppet and EC2**
  - Controlling your infrastructure with Puppet
  - Creating EC2 instances
- **Puppet and Docker**
  - The Docker module
  - Launching Docker containers

## Objective

- **By the end of this session you should be able to**
  - Create ec2 resources using puppet
  - Deploy docker containers with puppet

## Using Puppet to provision servers in EC2

- **Puppet has a set of different plugins which allow it to define and control cloud system setups**
  - We are going to look at using amazon webservice
  - But similar options exist for Azure or open source clouds
- **It is possible to maintain your entire cloud infrastructure from within Puppet**
  - Define security groups
  - Ensure that servers exist
  - There are options to connect to most AWS services
- **See <https://puppetlabs.com/blog/provision-aws-infrastructure-using-puppet> for more information**

<https://forge.puppetlabs.com/puppetlabs/aws>

## Creating new EC2 instances with puppet

### ▪ Step 1: System Setup

- Verify that you have installed a puppet agent locally on your machine:

```
$ which puppet  
/opt/puppetlabs/bin/puppet
```

- Verify the version of puppet (recommend version is 4.6.2):

```
$ puppet -V  
4.6.2
```

<https://linuxacademy.com/howtoguides/posts/show/topic/11889-deploying-ec2-resources-with-puppet>

## Creating new EC2 instances with puppet

### ▪ Step 1: System Setup

- Install the module and dependencies

```
sudo /opt/puppetlabs/puppet/bin/gem install \
                                     aws-sdk-core retries
sudo puppet module install puppetlabs-aws
```

- Add your AWS credentials

```
export AWS_ACCESS_KEY_ID=your_access_key_id
export AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

6

```
export AWS_ACCESS_KEY_ID=lina
export AWS_SECRET_ACCESS_KEY=lina_key
```

## Creating new EC2 instances with puppet

- **Step 2: Create a security group**
  - We need port 8080 open to access the hosted services

```
ec2_securitygroup { 'puppets':  
  ensure      => present,  
  region      => 'eu-west-1',  
  description => 'ports required for puppets',  
  ingress     => [{  
    protocol => 'tcp',  
    port     => 80,  
    cidr     => '0.0.0.0/0',  
  }, {  
    protocol => 'tcp',  
    port     => 22,  
    cidr     => '0.0.0.0/0',  
  }, {  
    protocol => 'tcp',  
    port     => 8080,  
    cidr     => '0.0.0.0/0',  
  }],  
}
```

7

## Creating new EC2 instances with puppet

- **Step 3: Define the instance**

```
ec2_instance { 'puppet-created':  
  ensure      => present,  
  region      => 'eu-west-1',  
  image_id    => 'ami-47a23a30', #EU West Ubuntu  
  instance_type => 't2.micro',  
  key_name     => 'qwikLABS-L407-10860',  
  security_groups => ['puppets'],  
  subnet      => 'subnet-name',  
}
```

- You need to ensure that you have named your subnets in AWS
  - Do not use the ID of the subnet!

8

You can add user data (a bootstrap script) with the user\_data property

```
user_data      => template('agent-setup.sh.erb'),
```



## Creating new EC2 instances with puppet

- **Step 4: Tell puppet to run the manifest**
  - Have a look at your dashboard

```
root@ip-172-31-42-219:~/puppetscripts# puppet apply ec2-helloworld.pp
Notice: Compiled catalog for agent1.eu-west-1.compute.internal in environment production in 0.03 seconds
Warning: Ambiguous subnet name '' resolves to subnets subnet-32d32059 (vpc: vpc-3dd32056), subnet-33d32058 (vpc: vpc-3dd32056), subnet-3cd32057 (vpc: vpc-3dd32056) - using subnet-32d32059
Notice: /Stage[main]/Main/Ec2_instance[puppet-created]/ensure: changed absent to running
Notice: Finished catalog run in 14.67 seconds
root@ip-172-31-42-219:~/puppetscripts#
```

Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
<input checked="" type="checkbox"/>	puppet-created	i-ac26c314	t2.micro	eu-west-1a	running	Initializing	None
<input type="checkbox"/>	Puppet Agent	i-ae2d1b0f	t2.medium	eu-west-1c	running	2/2 checks...	None
<input type="checkbox"/>	Puppet Master	i-4d2016ec	m3.large	eu-west-1c	running	2/2 checks...	None

## Puppet and Docker Configs

- We can add Docker information to the puppet manifest file
  - There's a module to help

```
$ sudo puppet module install garethr-docker
```

- Now create the manifest using the docker information:

```
include 'docker'

docker::image { 'ubuntu':
  ensure => 'absent',
}

docker::run { 'helloworld':
  image    => 'ubuntu',
  command => '/bin/sh -c "while true; do echo hello
                                world; sleep 1; done"',
}
```

When you apply the script you will see the Docker container running:

```
$ puppet apply -t docker-helloworld.pp
```

```
Notice: /Stage[main]/Main/Docker::Run[helloworld]/File[/etc/init.d/docker-helloworld]/ensure: created
Info: /Stage[main]/Main/Docker::Run[helloworld]/File[/etc/init.d/docker-helloworld]: Scheduling refresh of Service[docker-helloworld]
Notice: /Stage[main]/Main/Docker::Run[helloworld]/Service[docker-helloworld]/ensure: ensure changed 'stopped' to 'running'
Info: /Stage[main]/Main/Docker::Run[helloworld]/Service[docker-helloworld]: Unscheduling refresh on Service[docker-helloworld]
Notice: Finished catalog run in 53.61 seconds
root@ip-172-31-23-246:~/puppetscripts# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
b2b8b83116a2       ubuntu             "/bin/sh -c 'while t  16 seconds ago      Up 15 seconds              helloworld
root@ip-172-31-23-246:~/puppetscripts#
```

## Puppet and Docker Configs

- **You can define all aspects of the docker config from the manifest file**

```
docker::run { 'helloworld':  
  image => 'ubuntu',  
  command => '/bin/sh -c "while true; do echo hello world;  
                                     sleep 1; done"',  
  ports => ['4444', '4555'],  
  volumes => ['/var/lib/couchdb', '/var/log'],  
  volumes_from => '6446ea52fbc9',  
  memory_limit => 10485760, # bytes  
  username => 'example',  
  hostname => 'example.com',  
  env => ['FOO=BAR', 'FOO2=BAR2'],  
  dns => ['8.8.8.8', '8.8.4.4'],  
}
```

11

See: <https://puppetlabs.com/blog/simplify-managing-docker-puppet> or  
<https://docs.docker.com/articles/puppet/>

## Exercise

- **Use puppet to automate deploying your docker containers**
  - Create a new module to host some some docker containers
  - Create a new agent
    - (Manually to start with)
  - Add your agent to a group and assign the docker class to it!
- **Stretchers**
  - Look at how you can manage your infrastructure with puppet
  - (And read about whether you should or not)

## To read more about Puppet

### Documentation:

- <https://puppet.com/blog/simplify-managing-docker-puppet>
- <https://puppet.com/blog/puppet-docker-running-puppet-container-centric-infrastructure>
- <https://docs.docker.com/engine/admin/puppet/>

## Summary

- **Puppet and EC2**
  - Controlling your infrastructure with Puppet
  - Creating EC2 instances
- **Puppet and Docker**
  - The Docker module
  - Launching Docker containers