

```

{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "Template for stack",

  "Parameters" : {

    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the instances",
      "Type" : "String",
      "MinLength" : "1",
      "MaxLength" : "255",
      "AllowedPattern" : "[\\x20-\\x7E]*",
      "ConstraintDescription" : "can contain only ASCII characters."
    },

    "HostKeys" : {
      "Description" : "Public Key",
      "Type" : "String"
    },

    "SecretAccessKey" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the instances",
      "Type" : "String"
    },

    "InstanceType" : {
      "Description" : "Chef Server EC2 instance type",
      "Type" : "String",
      "Default" : "m1.small",
      "AllowedValues" : [ "t1.micro", "m1.small" ],
      "ConstraintDescription" : "must be a valid EC2 instance type."
    },

    "SSHLocation" : {
      "Description" : " The IP address range that can be used to SSH to the EC2 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\.\\.\\.\\.\\. (\\d{1,3})\\.\\.\\.\\.\\. (\\d{1,3})\\.\\.\\.\\.\\. (\\d{1,3})/ (\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    }
  },

  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"      : { "Arch" : "64" },
      "m1.small"     : { "Arch" : "64" }
    },

    "AWSRegionArch2AMI" : {
      "us-east-1"    : { "32" : "ami-d7a18dbe", "64" : "ami-bba18dd2", "64HVM" : "ami-0da96764" },
      "us-west-2"   : { "32" : "ami-def297ee", "64" : "ami-ccf297fc", "64HVM" :
"NOT_YET_SUPPORTED" },
      "us-west-1"   : { "32" : "ami-923909d7", "64" : "ami-a43909e1", "64HVM" :
"NOT_YET_SUPPORTED" }
    }
  },

  "Resources" : {

    "ChefClient" : {
      "Type" : "AWS::EC2::Instance",
      "Metadata" : {
        "Description" : "Chef Client",

        "AWS::CloudFormation::Init" : {
          "config" : {

```

```

    "packages" : {
      "yum" : {
        "git" : []
      }
    }
  },
},

"Properties": {
  "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" }, {
"Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" : "InstanceType" }, "Arch" ] } ] },
  "InstanceType" : { "Ref" : "InstanceType" },
  "SecurityGroups" : [ { "Ref" : "WebServerSecurityGroup" } ],
  "KeyName" : { "Ref" : "KeyName" },

  "UserData" : { "Fn::Base64" : { "Fn::Join" : [ "", [
    "#!/bin/bash -v\n",

    "yum update -y aws-cfn-bootstrap\n",

    "function error_exit\n",
    "{\n",
    "  cfn-signal -e 1 -r \"\$1\" \"\", { "Ref" : "WaitHandle" }, \"'\n",
    "  exit 1\n",
    "}\n",

    "yum update -y\n",
    "yum install git -y\n",
    "/sbin/service iptables stop\n",
    "/sbin/service ip6tables stop\n",
    "/sbin/chkconfig iptables off\n",
    "/sbin/chkconfig ip6tables off\n",
    "yum install git -y\n",

    "/usr/bin/curl -L https://www.opscode.com/chef/install.sh | bash\n",
    "cd /root/\n",
    "/usr/bin/git git://github.com/opscode/chef-repo.git\n",
    "/bin/mkdir -p /root/chef-repo/.chef\n",
    "/bin/mkdir -p /etc/chef\n",

    "/bin/mkdir /root/.aws\n",
    "/bin/touch /root/.aws/config\n",
    "/bin/echo '[default]' >> /root/.aws/config\n",
    "/bin/echo 'region = ", { "Ref" : "AWS::Region" }, "' >> /root/.aws/config\n",
    "/bin/echo 'aws_access_key_id = ", { "Ref" : "HostKeys" }, "' >> /root/.aws/config\n",
    "/bin/echo 'aws_secret_access_key = ", { "Ref" : "SecretAccessKey" }, "' >>
/root/.aws/config\n",

    "/usr/bin/aws s3 cp s3://storage/admin.pem /root/chef-repo/.chef\n",
    "/usr/bin/aws s3 cp s3://storage/chef-validator.pem /root/chef-repo/.chef\n",
    "/usr/bin/aws s3 cp s3://storage/knife.rb /root/chef-repo/.chef\n",
    "/usr/bin/aws s3 cp s3://storage/client.rb /etc/chef\n",
    "/usr/bin/aws s3 cp s3://storage/json_attribs.json /etc/chef\n",
    "/bin/cp -p /root/chef-repo/.chef/chef-validator.pem /etc/chef/validation.pem\n",
    "/usr/sbin/ntpdate -q 0.europe.pool.ntp.org\n",

    "/bin/echo '\nchef_server_url \"\", { "Ref" : "ChefServerURL" }, \"'\n" >>
/etc/chef/client.rb\n",
    "/bin/echo '\nchef_server_url \"\", { "Ref" : "ChefServerURL" }, \"'\n" >> /root/chef-
repo/.chef/knife.rb\n",

    "/usr/bin/chef-client\n",

    "/opt/aws/bin/cfn-signal -e 0 -r \"ChefClient setup complete\" \"\", { "Ref" : "WaitHandle"
}, \"'\n"

```

```

    ]}}
  }
},

"WaitHandle" : {
  "Type" : "AWS::CloudFormation::WaitConditionHandle"
},

"WaitCondition" : {
  "Type" : "AWS::CloudFormation::WaitCondition",
  "DependsOn" : "ChefClient",
  "Properties" : {
    "Handle" : {"Ref" : "WaitHandle"},
    "Timeout" : "1200"
  }
},

"ChefServer" : {
  "Type" : "AWS::EC2::Instance",
  "Metadata" : {
    "Description" : "Bootstrap ChefServer",

    "AWS::CloudFormation::Init" : {
      "config" : {
        "packages" : {
          "yum" : {
            "wget" : []
          }
        }
      }
    }
  },

  "Properties": {
    "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" }, {
"Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" : "InstanceType" }, "Arch" ] } ] },
    "InstanceType" : { "Ref" : "InstanceType" },
    "SecurityGroups" : [ {"Ref" : "WebServerSecurityGroup"} ],
    "KeyName" : { "Ref" : "KeyName" },

    "UserData" : { "Fn::Base64" : { "Fn::Join" : [ "", [
      "#!/bin/bash\n",

      "cfn-init --region ", { "Ref" : "AWS::Region" },
      " -s ", { "Ref" : "AWS::StackId" }, " -r ChefServer ", " -c orderby ",
      " --access-key ", { "Ref" : "HostKeys" },
      " --secret-key ", {"Ref" : "SecretAccessKey"}, " || error_exit 'Failed to run
cfn-init'\n",

      "yum update -y aws-cfn-bootstrap\n",

      "function error_exit\n",
      "{\n",
      "  cfn-signal -e 1 -r \"\$1\" ", { "Ref" : "WaitHandle" }, "'\n",
      "  exit 1\n",
      "}\n",

      "yum update -y\n",
      "/sbin/service iptables stop\n",
      "/sbin/service ip6tables stop\n",
      "/sbin/chkconfig iptables off\n",
      "/sbin/chkconfig ip6tables off\n",

      "#Install ChefServer package\n",
      "cd /root/\n",

```

```

"/usr/bin/wget https://opscode-omnibus-packages.s3.amazonaws.com/el/6/x86_64/chef-server-
11.0.10-1.el6.x86_64.rpm\n",
"/bin/rpm -ivh /root/chef-server-11.0.10-1.el6.x86_64.rpm\n",
"/usr/bin/wget https://s3.amazonaws.com/storage/default.rb\n",
"/bin/cp -f default.rb /opt/chef-server/embedded/cookbooks/runit/recipes/default.rb\n",

"#Configure ChefServer\n",
"su - -c '/usr/bin/chef-server-ctl reconfigure'\n",
"su - -c '/usr/bin/chef-server-ctl restart'\n",

"#AWS creds installation\n",
"/bin/mkdir /root/.aws\n",
"/bin/touch /root/.aws/config\n",
"/bin/echo '[default]' >> /root/.aws/config\n",
"/bin/echo 'region = ", { "Ref" : "AWS::Region" }, "' >> /root/.aws/config\n",
"/bin/echo 'aws_access_key_id = ", { "Ref" : "HostKeys" }, "' >> /root/.aws/config\n",
"/bin/echo 'aws_secret_access_key = ", { "Ref" : "SecretAccessKey" }, "' >>
/root/.aws/config\n",

"#Upload files for client\n",
"/usr/bin/aws s3 cp /etc/chef-server/admin.pem s3://storage/\n",
"/usr/bin/aws s3 cp /etc/chef-server/chef-validator.pem s3://storage/\n",

"#Chef client and dirs for it\n",
"/usr/bin/curl -L https://www.opscode.com/chef/install.sh | /bin/bash\n",
"/bin/mkdir /root/.chef\n",
"/bin/mkdir /etc/chef\n",
"/bin/mkdir /etc/chef/cookbooks\n",
"/bin/mkdir /etc/chef/roles\n",

"#Knife client config files from S3\n",
"/bin/cp /etc/chef-server/admin.pem /etc/chef/client.pem\n",
"/usr/bin/aws s3 cp s3://storage/knife_admin.rb /root/.chef/knife.rb\n",

"#Roles and cookbooks from S3\n",
"/usr/bin/aws s3 cp s3://storage/roles/ /etc/chef/roles/ --recursive\n",
"/usr/bin/aws s3 cp s3://storage/cookbooks/ /etc/chef/cookbooks/ --recursive\n",

"#Cookbooks from community\n",
"/usr/bin/knife cookbook site download cron\n",
"/usr/bin/knife cookbook site download jenkins\n",
"/usr/bin/knife cookbook site download ntp\n",
"/usr/sbin/ntpdate -q 0.europe.pool.ntp.org\n",
"yum remove ruby -y\n",
"yum install ruby19 -y\n",

"#Unpack and move cookbooks\n",
"/bin/mv /root/*.tar.gz /etc/chef/cookbooks\n",
"for i in `bin/ls /etc/chef/cookbooks/*.tar.gz`; do /bin/tar xzf $i -C
/etc/chef/cookbooks/; /bin/rm -f $i; done\n",
"for i in `bin/ls /etc/chef/cookbooks`; do /usr/bin/knife cookbook upload $i; done\n",

"#Upload cookbooks and roles\n",
"/usr/bin/knife cookbook upload * -c '/root/.chef/knife.rb'\n",
"/usr/bin/knife role from file /etc/chef/roles/*.rb\n",

"/bin/echo -e \"*/5 * * * * root /usr/bin/knife exec -E
'nodes.find(\\\"!roles:BaseRole\\\") { |n| puts n.run_list.add(\\\"role[BaseRole]\\\"); n.save}' -c
'/root/.chef/knife.rb'\" >> /etc/crontab\n",
"/bin/echo -e \"*/5 * * * * root /usr/bin/knife exec -E
'nodes.find(\\\"env_role:master AND !roles:master\\\") { |n| puts
n.run_list.add(\\\"role[master]\\\"); n.save}' -c '/root/.chef/knife.rb'\" >> /etc/crontab\n",
"/bin/echo -e \"*/5 * * * * root /usr/bin/knife exec -E 'nodes.find(\\\"env_role:slave
AND !roles:slave\\\") { |n| puts n.run_list.add(\\\"role[slave]\\\"); n.save}' -c
'/root/.chef/knife.rb'\" >> /etc/crontab\n",

```

```

    "/opt/aws/bin/cfn-signal -e 0 -r \"ChefServer setup complete\" \"\", { \"Ref\" : \"WaitHandle\"
}, \"\"\n\"

    ]]]}
  }
},

\"WaitHandle\" : {
  \"Type\" : \"AWS::CloudFormation::WaitConditionHandle\"
},

\"WaitCondition\" : {
  \"Type\" : \"AWS::CloudFormation::WaitCondition\",
  \"DependsOn\" : \"ChefServer\",
  \"Properties\" : {
    \"Handle\" : {\"Ref\" : \"WaitHandle\"},
    \"Timeout\" : \"1200\"
  }
},

\"WebServerSecurityGroup\" : {
  \"Type\" : \"AWS::EC2::SecurityGroup\",
  \"Properties\" : {
    \"GroupDescription\" : \"Enable HTTP access via port 80 and SSH access\",
    \"SecurityGroupIngress\" : [
      {\"IpProtocol\" : \"tcp\", \"FromPort\" : \"80\", \"ToPort\" : \"80\", \"CidrIp\" : \"0.0.0.0/0\"},
      {\"IpProtocol\" : \"tcp\", \"FromPort\" : \"8080\", \"ToPort\" : \"8080\", \"CidrIp\" : \"0.0.0.0/0\"},
      {\"IpProtocol\" : \"tcp\", \"FromPort\" : \"443\", \"ToPort\" : \"443\", \"CidrIp\" : \"0.0.0.0/0\"},
      {\"IpProtocol\" : \"tcp\", \"FromPort\" : \"22\", \"ToPort\" : \"22\", \"CidrIp\" : { \"Ref\" :
\"SSHLocation\"}}
    ]
  }
},
}

```