Stack requirements

Security Groups

MasterSG

MasterSG is the name given to the Security Group used to control access to the Master Server. The following rules must be in place:

```
allow inbound traffic from MasterELBSG on TCP port 8080
allow inbound traffic from AgentSG on TCP port 8080
allow inbound traffic from AgentSG on TCP port 38228
allow inbound traffic from AgentSG on TCP port 49817
```

MasterELBSG

MasterELBSG is the Security Group used to filter access to the ELB, it has the following rules:

```
allow inbound traffic from 0.0.0.0/0 on TCP port 80
allow inbound traffic from 0.0.0.0/0 on TCP port 443
```

Port 443 is not actually open, but it could be used to enable HTTPS on the ELB/Jenkins.

MasterStorageSG

MasterStorageSG filters inbound traffic to the NFS mount points. Rules:

```
allow inbound traffic from AgentSG on TCP port 2049
allow inbound traffic from MasterSG on TCP port 2049
```

AgentSG

Agents Security Group with the following rules:

```
allow inbound traffic from MasterSG on TCP port 8080
allow inbound traffic from AgentELBSG on TCP port 8080
```

AgentELBSG

```
allow inbound traffic from AgentELBSG on TCP port 8080
```

IAM

MasterIAMRole

Instructions:

- Go to *IAM > Roles > Create new role*, we named our role "MasterIAMRole", select "AWS Service Roles" then select "Amazon EC2". Don't add any policy, just create the role.
- Go to the Role Sumary, you should see something like:

```
Role ARN                  arn:aws:iam::XXXXXXXX:role/MasterIAMRole
Instance Profile ARN(s)   arn:aws:iam::XXXXXXXX:instance-profile/MasterIAMRole
Path                      /
```

Scroll down to "Inline Policies" to create the following custom policies. Please note that you can adjust those policies as required in order to have a more restrictive permissions schema.

Autoscaling

Name: autoscaling.

Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "autoscaling:CompleteLifecycleAction",
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "write"
    }
  ]
}
```

Cloudwatch

Name: cloudwatch.

Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "cloudwatch:PutMetricData",
      "Resource": "*",
```

```
      "Effect": "Allow",
      "Sid": "write"
    }
  ]
}
```

Logs

Name: logs.

Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "arn:aws:logs:*:*:*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

SQS

Name: sqs.

Policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sqs:DeleteMessage",
                "sqs:ReceiveMessage"
            ],
            "Resource": "arn:aws:sqs:us-west-2:571780515387:*",
            "Effect": "Allow",
            "Sid": "write"
        }
    ]
```

```
}
```

MasterIP

The MasterIP refers to the Master Instance Profile, it's the same as the role name (`MasterIAMRole` in this case):

```
Role ARN                  arn:aws:iam::XXXXXXXX:role/MasterIAMRole
Instance Profile ARN(s)   arn:aws:iam::XXXXXXXX:instance-profile/MasterIAMRole
Path                      /
```

AgentIAMRole

Follow the same procedure explained above and create a new role, we named it "AgentIAMRole".

Add the following policies:

EC2

Name: ec2.

Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeInstances",
        "autoscaling:DescribeAutoScalingGroups"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "read"
    }
  ]
}
```

Logs

Name: logs.

Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "arn:aws:logs:*:*:*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

AgentIP

On the top of the AgentIAMRole summary page, you can see something like this:

```
Role ARN                arn:aws:iam::XXXXXXXX:role/AgentIAMRole
Instance Profile ARN(s) arn:aws:iam::XXXXXXXX:instance-profile/AgentIAMRole
Path                    /
```

The AgentIP has the same value of the role name (AgentIAMRole).

AgentTerminatingLifecycleHookIAMRole

Follow the same procedure explained above, but instead of selecting "Amazon EC2" when choosing the "AWS Service role", select "AutoScaling Notification Access". Then, create the following inline policy:

SQS

Name: sqs.

Policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sqs:SendMessage",
                "sqs:GetQueueUrl"
            ],
```

```
            "Resource": "arn:aws:sqs:us-west-2:571780515387:*",
            "Effect": "Allow",
            "Sid": "write"
        }
    ]
}
```

The Jenkins role

We have created a Jenkins role and attached the following policy (this code is for documentation purposes only, please refer to the `jenkins_iam.json` file):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1476201203000",
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Stmt1476202533000",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:AttachVolume",
        "ec2:DescribeNetworkInterfaceAttribute",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:DescribeNetworkInterfaces",
        "ec2:CreateNetworkInterface",
        "ec2:DeleteNetworkInterface",
        "ec2:DescribeInstances"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Stmt1476202775000",
      "Effect": "Allow",
      "Action": [
        "sqs:CreateQueue",
        "sqs:DeleteQueue",
        "sqs:GetQueueAttributes"
      ],
      "Resource": [
        "*"
      ]
```

```
    },
    {
      "Sid": "Stmt1476202855000",
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:CreateLoadBalancer",
        "elasticloadbalancing:DeleteLoadBalancer",
        "elasticloadbalancing:CreateTargetGroup",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:ModifyTargetGroup",
        "elasticloadbalancing:DeleteTargetGroup",
        "elasticloadbalancing:AddTags",
        "autoscaling:CreateOrUpdateTags",
        "elasticloadbalancing:RemoveTags",
        "elasticloadbalancing:CreateListener",
        "elasticloadbalancing:DescribeListeners",
        "elasticloadbalancing:DeleteListener"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Stmt1476202972000",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:DeleteLogGroup",
        "logs:PutRetentionPolicy"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Stmt1476203095000",
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:CreateFileSystem",
        "elasticfilesystem:DeleteFileSystem",
        "elasticfilesystem:DescribeFileSystems",
        "elasticfilesystem:CreateTags",
        "elasticfilesystem:DescribeTags",
        "elasticfilesystem:DeleteTags",
        "elasticfilesystem:CreateMountTarget",
        "elasticfilesystem:DeleteMountTarget",
        "elasticfilesystem:DescribeMountTargets",
        "elasticfilesystem:DescribeMountTargetSecurityGroups",
        "elasticfilesystem:ModifyMountTargetSecurityGroups"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Stmt1476205815000",
```

```
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Stmt1476206935000",
      "Effect": "Allow",
      "Action": [
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:DeleteAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:CreateLaunchConfiguration",
        "autoscaling:DeleteLaunchConfiguration",
        "autoscaling:DescribeScalingActivities",
        "autoscaling:DescribeLaunchConfigurations",
        "autoscaling:DescribeAutoScalingInstances",
        "autoscaling:PutLifecycleHook",
        "autoscaling:DeleteLifecycleHook",
        "autoscaling:PutScalingPolicy",
        "autoscaling:DeletePolicy"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Stmt1476215473000",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:DeleteAlarms"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```