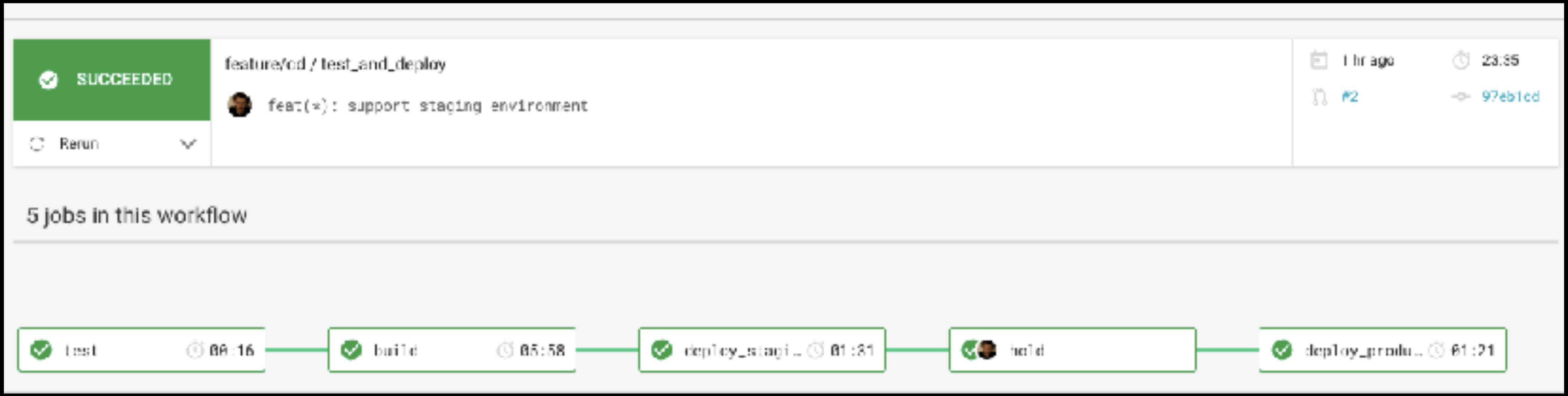
Notejam

Deployment Pipeline Architecture



SUCCEEDED

Rerun

feature/cd / test_and_deploy

feat(*): support staging environment

1 hr ago 23:35

#2 97eb1cd

5 jobs in this workflow

test 00:16

build 05:58

deploy_stagi… 01:31

hold

deploy_produ… 01:21

Deployment Pipeline

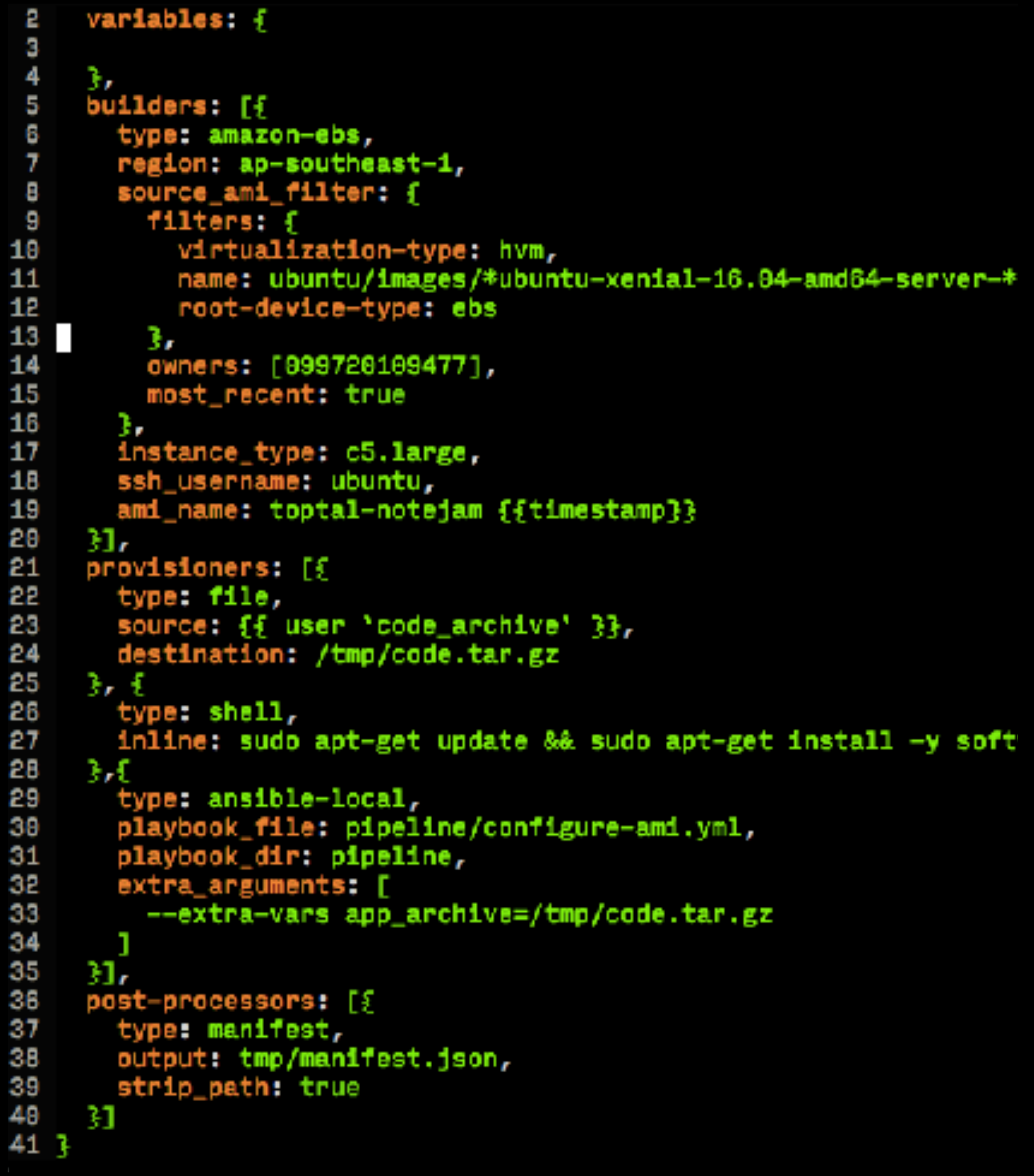
DevOps Standard CI & CD

Summary

- Immutable Golden Image Pipeline
- Automatic deploys to master
- Manual promotion to master
- CloudWatch Telemetry
- Ansible & CloudFormation for orchestration
- Postgres RDS for database
- Using Rails sample app

Build w/Packer

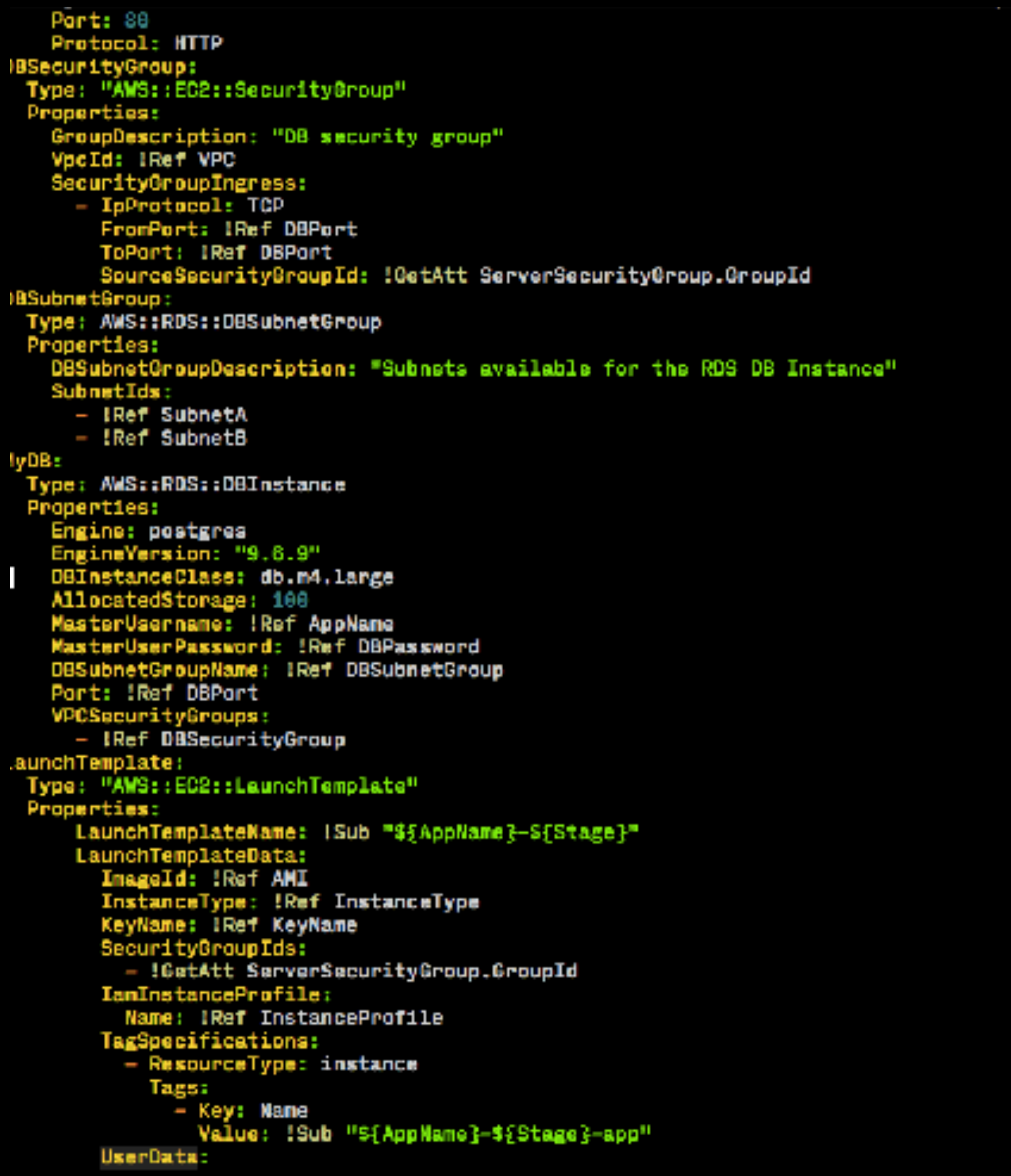
- Packer is the industry standard
- Ansible playbook (pipeline/configure-ami.yml) does the heavy lifting
- AMI includes logging and telemetry config
- AMI usable across all pipeline stages



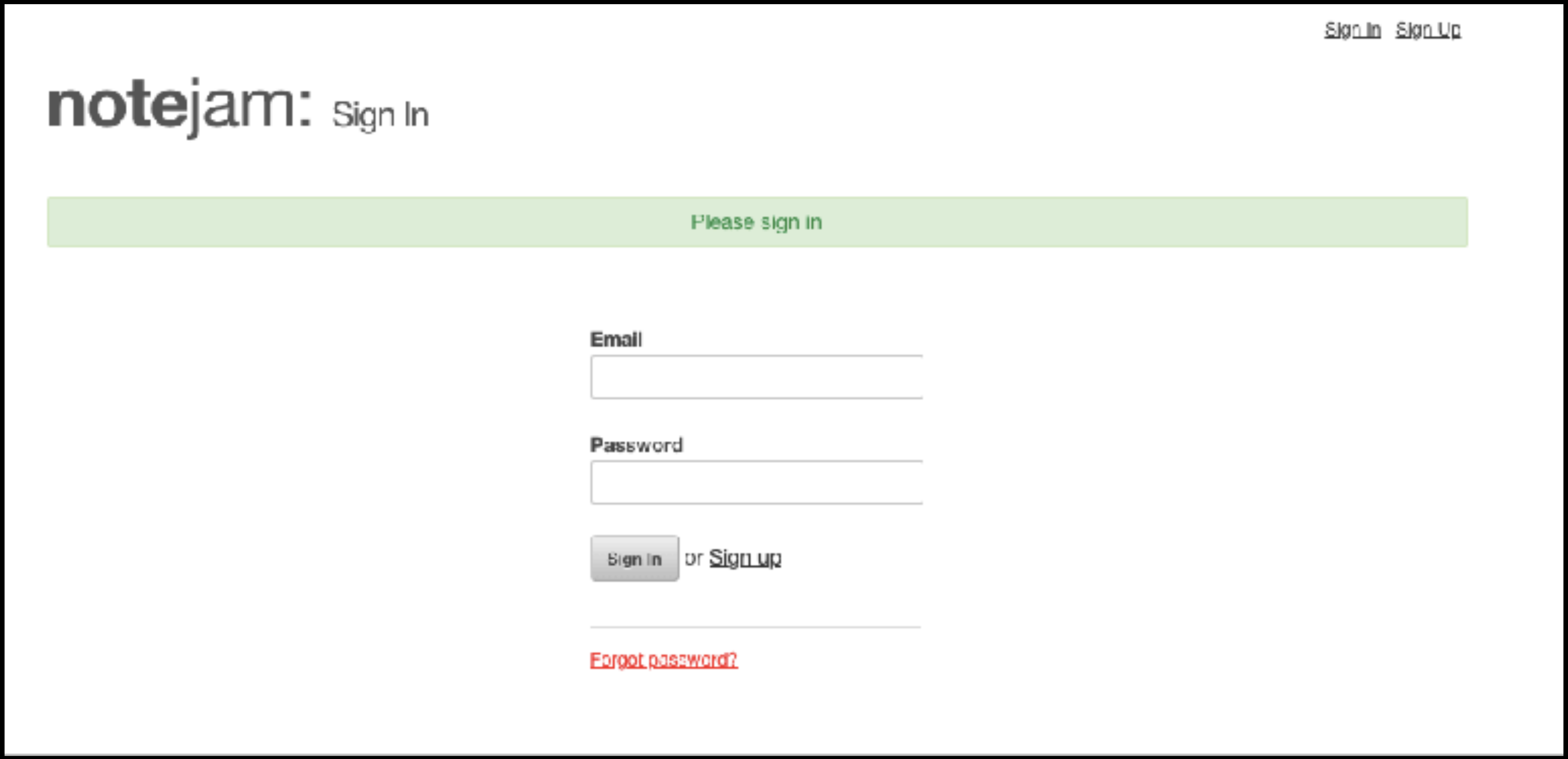
```
  variables: {

  },
  builders: [{
    type: amazon-ebs,
    region: ap-southeast-1,
    source_ami_filter: {
      filters: {
        virtualization-type: hvm,
        name: ubuntu/images/*ubuntu-xenial-16.04-amd64-server-*
        root-device-type: ebs
      },
      owners: [099720109477],
      most_recent: true
    },
    instance_type: c5.large,
    ssh_username: ubuntu,
    ami_name: toptal-notejam {{timestamp}}
  }],
  provisioners: [{
    type: file,
    source: {{ user 'code_archive' }},
    destination: /tmp/code.tar.gz
  }, {
    type: shell,
    inline: sudo apt-get update && sudo apt-get install -y soft
  },{
    type: ansible-local,
    playbook_file: pipeline/configure-ami.yml,
    playbook_dir: pipeline,
    extra_arguments: [
      --extra-vars app_archive=/tmp/code.tar.gz
    ]
  }],
  post-processors: [{
    type: manifest,
    output: tmp/manifest.json,
    strip_path: true
  }]
}
```

Deploy w/CloudFormation

- Infrastructure as Code from t-zero
- Postgres RDS instance
- 2AZ setup with ALB
- Autoscaling Triggers
- Reused for all deployment stages
- Running app tested in deployment pipeline

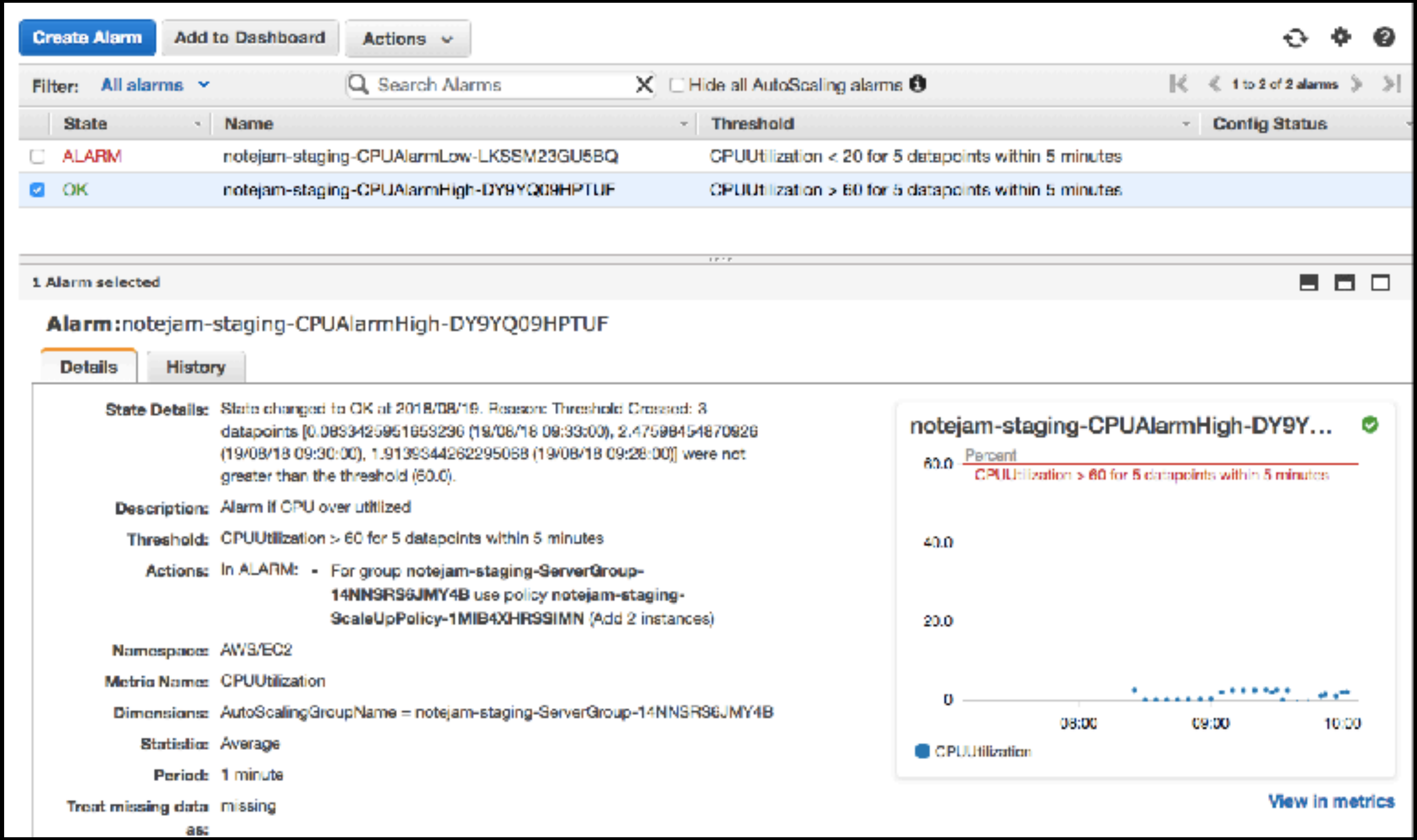


```
    Port: 80
    Protocol: HTTP
BSecurityGroup:
  Type: "AWS::EC2::SecurityGroup"
  Properties:
    GroupDescription: "DB security group"
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: TCP
        FromPort: !Ref DBPort
        ToPort: !Ref DBPort
        SourceSecurityGroupId: !GetAtt ServerSecurityGroup.GroupId
BSubnetGroup:
  Type: AWS::RDS::DBSubnetGroup
  Properties:
    DBSubnetGroupDescription: "Subnets available for the RDS DB Instance"
    SubnetIds:
      - !Ref SubnetA
      - !Ref SubnetB
yDB:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: "9.6.9"
    DBInstanceClass: db.m4.large
    AllocatedStorage: 100
    MasterUsername: !Ref AppName
    MasterUserPassword: !Ref DBPassword
    DBSubnetGroupName: !Ref DBSubnetGroup
    Port: !Ref DBPort
    VPCSecurityGroups:
      - !Ref DBSecurityGroup
aunchTemplate:
  Type: "AWS::EC2::LaunchTemplate"
  Properties:
      LaunchTemplateName: !Sub "${AppName}-${Stage}"
      LaunchTemplateData:
        ImageId: !Ref AMI
        InstanceType: !Ref InstanceType
        KeyName: !Ref KeyName
        SecurityGroupIds:
          - !GetAtt ServerSecurityGroup.GroupId
        IamInstanceProfile:
          Name: !Ref InstanceProfile
        TagSpecifications:
          - ResourceType: instance
            Tags:
              - Key: Name
                Value: !Sub "${AppName}-${Stage}-app"
        UserData:
```



App in Production

Each deploy tests infra with GET /ping



State	Name	Threshold	Config Status
ALARM	notejam-staging-CPUAlarmLow-LKSSM23GU5BQ	CPUUtilization < 20 for 5 datapoints within 5 minutes	
OK	notejam-staging-CPUAlarmHigh-DY9YQ09HPTUF	CPUUtilization > 60 for 5 datapoints within 5 minutes	

1 Alarm selected

Alarm:notejam-staging-CPUAlarmHigh-DY9YQ09HPTUF

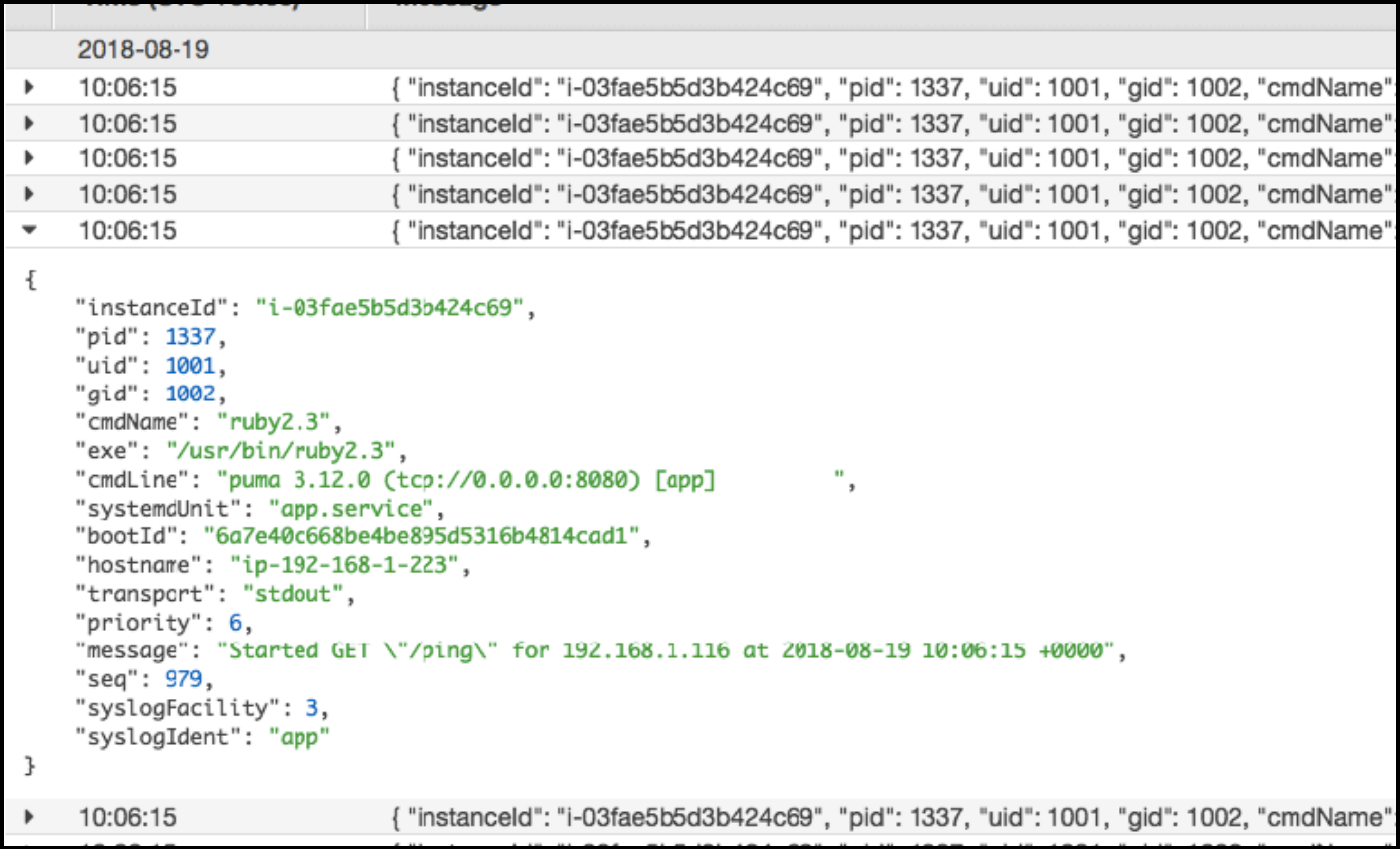
Details | History

State Details:	State changed to OK at 2018/08/19. Reason: Threshold Crossed: 3 datapoints [0.0833425951653236 (19/08/18 09:33:00), 2.47598454870826 (19/08/18 09:30:00), 1.9139344262295068 (19/08/18 09:28:00)] were not greater than the threshold (60.0).
Description:	Alarm if CPU over utilized
Threshold:	CPUUtilization > 60 for 5 datapoints within 5 minutes
Actions:	In ALARM: - For group **notejam-staging-ServerGroup-14NNSRS6JMY4B** use policy **notejam-staging-ScaleUpPolicy-1MIB4XHRSSIMN** (Add 2 instances)
Namespace:	AWS/EC2
Metric Name:	CPUUtilization
Dimensions:	AutoScalingGroupName = notejam-staging-ServerGroup-14NNSRS6JMY4B
Statistic:	Average
Period:	1 minute
Treat missing data as:	missing

notejam-staging-CPUAlarmHigh-DY9Y...
Percent
CPUUtilization > 60 for 5 datapoints within 5 minutes
60.0
40.0
20.0
0
08:00
09:00
10:00
CPUUtilization
View in metrics

AutoScaling Alarms

Commit to CloudFormation template and deploy



```
2018-08-19
▸ 10:06:15    { "instanceId": "i-03fae5b5d3b424c69", "pid": 1337, "uid": 1001, "gid": 1002, "cmdName"
▸ 10:06:15    { "instanceId": "i-03fae5b5d3b424c69", "pid": 1337, "uid": 1001, "gid": 1002, "cmdName"
▸ 10:06:15    { "instanceId": "i-03fae5b5d3b424c69", "pid": 1337, "uid": 1001, "gid": 1002, "cmdName"
▸ 10:06:15    { "instanceId": "i-03fae5b5d3b424c69", "pid": 1337, "uid": 1001, "gid": 1002, "cmdName"
▾ 10:06:15    { "instanceId": "i-03fae5b5d3b424c69", "pid": 1337, "uid": 1001, "gid": 1002, "cmdName"
{
    "instanceId": "i-03fae5b5d3b424c69",
    "pid": 1337,
    "uid": 1001,
    "gid": 1002,
    "cmdName": "ruby2.3",
    "exe": "/usr/bin/ruby2.3",
    "cmdLine": "puma 3.12.0 (tcp://0.0.0.0:8080) [app]          ",
    "systemdUnit": "app.service",
    "bootId": "6a7e40c668be4be895d5316b4814cad1",
    "hostname": "ip-192-168-1-223",
    "transport": "stdout",
    "priority": 6,
    "message": "Started GET \"/ping\" for 192.168.1.116 at 2018-08-19 10:06:15 +0000",
    "seq": 979,
    "syslogFacility": 3,
    "syslogIdent": "app"
}
▸ 10:06:15    { "instanceId": "i-03fae5b5d3b424c69", "pid": 1337, "uid": 1001, "gid": 1002, "cmdName"
```

Remote Log Access

Stream logs by CW utils

App Preparation

- Sample Rails application broken due to bit rot
- Locked ruby to a specific version
- Converted to 12Factor app
- Added GET /ping to for ALB health check that's independent from application functionality
- Added "asdf" version manager and dev env bootstrap script

Next Steps

- Review Technology Choices
- Lock down SSH access
- Speed up deployment pipeline by using a base AMI
- Evaluate Spinnaker or alternate deployment options (e.g. containers)
- Create bastion for application console access
- Configure cost allocation tags
- Create per-topic branch environments