respawn Documentation

Release 0.0.1

Victor Mena, Kuber Kaul, Vishal Shah

Mar 08, 2018

Contents

Contents:

CHAPTER 1

Getting Started

This page describes how to download, install and use the basic functionality of respawn.

1.1 Installation

To install respawn, simply:

1.1.1 Windows/Unix/Mac OS X

- Open command prompt and execute pip command :

```
pip install respawn
```

1.2 Usage - Template Generation

to use respawn, in your command prompt/terminal :

```
$ respawn pathToYAML.yaml
```

to create & validate the JSON against AWS using boto and pipe output to a file:

```
$ respawn --validate pathToYAML.yaml > pathToJSON.json
```

to pipe the output to a file :

```
$ respawn pathToYAML.yaml > pathToJSON.json
```

where:

- pathToYAML.yaml = the YAML file that needs to be processed into JSON.

- pathToJSON.json = the JSON file containing AWS cloudformation.

For exhaustive documentation and help with specific keywords to be used with resources , got to usage section.

1.3 Dependencies

- boto==2.32.1
- nose==1.3.3
- cfn-pyplates==0.4.3
- Jinja2==2.7.3
- enum34
- pytest==2.7.1

1.4 Next Steps

That concludes the getting started guide for respawn. Hopefully you're excited about the possibilities of respawn and ready to begin using respawn with your applications.

We've covered the basics of respawn in this guide. We recommend moving on to the usage next, which serves as a complete reference to all the features of respawn.

CHAPTER 2

Keywords - YAML/JSON

2.1 Resource Index

Following is the documentation of keywords required to add the following resources in your yaml file.

2.1.1 Auto Scaling Group

The AWS::AutoScaling::AutoScalingGroup type creates an Auto Scaling group resource for your stack.

JSON Syntax for auto scaling group.

```
"Type" : "AWS::AutoScaling::AutoScalingGroup",
"Properties" : {
    "AvailabilityZones" : [ String, ... ],
    "Cooldown" : String,
    "DesiredCapacity" : String,
    "HealthCheckGracePeriod" : Integer,
    "HealthCheckType" : String,
    "InstanceId" : String,
    "LaunchConfigurationName" : String,
    "LoadBalancerNames" : [ String, ... ],
    "MaxSize" : String,
    "MetricsCollection" : [ MetricsCollection, ... ]
    "MinSize" : String,
    "NotificationConfigurations" : [ NotificationConfigurations, ... ],
    "PlacementGroup" : String,
    "Tags" : [ Auto Scaling Tag, ..., ],
    "TerminationPolicies" : [ String, ..., ],
    "VPCZoneIdentifier" : [ String, ... ]
}
```

Sample YAML Syntax for Auto Scaling Group.

```
auto_scale_groups:
    *AutoScalingName*:
        hostname: sampleTestName
        availability_zones:
            - AZName1
            - AZName2
        min_size: 1
        max_size: 10
        desired_capacity: 10
        instance_id: ami-xxxxxxxx
        cooldown: 10
        launch_configuration: LaunchConfigName
        load_balancer_names:
            - LBName
            - ref(SampleLoadBalancer)
        max_size: 2
        min_size: 1
        metrics_collection:
            - granularity: 1Minute
            - granularity: 1Minute
        metrics:
            - Metric1
            - Metric2
        notification_configs:
            - notification_type:
                - Type1
                - Type2
              topic_arn: "arn:aws:[service]:[region]:[account]:resourceType/
↪resourcePath"
            - notification_type:
                - Type3
              topic_arn: "arn:aws:[service]:[region]:[account]:resourceType/
↪resourcePath"
        placement_group: PlacementGroupName
        tags:
            - key: Key1
```

```
              value: Value1
              propagate_at_launch: true
            - key: Key2
              value: Value2
              propagate_at_launch: false
        termination_policies:
            - Policy1
            - Policy2
        vpc_zone_identifier:
            - ZoneIdentifier1
            - ZoneIdentifier2
```

2.1.2 CloudWatch

Respawn supports CloudWatch for AutoScaling/EC2 instances. The AWS::CloudWatch::Alarm type creates a CloudWatch alarm.

JSON syntax for the resource CloudWatch.

```
"Type" : "AWS::CloudWatch::Alarm",
"Properties" : {
    "ActionsEnabled" : Boolean,
    "AlarmActions" : [ String, ... ],
    "AlarmDescription" : String,
    "AlarmName" : String,
    "ComparisonOperator" : String,
    "Dimensions" : [ Metric dimension, ... ],
    "EvaluationPeriods" : String,
    "InsufficientDataActions" : [ String, ... ],
    "MetricName" : String,
    "Namespace" : String,
    "OKActions" : [ String, ... ],
    "Period" : String,
    "Statistic" : String,
    "Threshold" : String,
    "Unit" : String
}
```

Sample YAML syntax for the resource CloudWatch.

```
cloud_watch:
    *CloudWatchName*:
        actions_enabled: true
        alarm_actions:
            - AlarmAction1
            - AlarmAction2
        alarm_name: SampleAlarm
        alarm_description: "Sample alarm description"
        comparison_operator: GreaterThanOrEqualToThreshold
        dimensions:
            - name: Dimension1
              value: Value1
            - name: Dimension2
              value: Value2
        evaluation_periods: 15
        insufficient_data_actions:
            - InsufficientDataAction1
```

```
            - InsufficientDataAction2
        metric_name: SampleName
        namespace: SampleNamespace
        ok_actions:
            - OkAction1
            - OkAction2
        period: 12
        statistic: Average
        threshold: 10
        unit: Milliseconds
```

2.1.3 Instances

The AWS::EC2::Instance type creates an Amazon EC2 Instance.

JSON syntax for the resource Instances.

```
"Type" : "AWS::EC2::Instance",
"Properties" : {
    "AvailabilityZone" : String,
    "BlockDeviceMappings" : [ EC2 Block Device Mapping, ... ],
    "DisableApiTermination" : Boolean,
    "EbsOptimized" : Boolean,
    "IamInstanceProfile" : String,
    "ImageId" : String,
    "InstanceInitiatedShutdownBehavior" : String,
    "InstanceType" : String,
    "KernelId" : String,
    "KeyName" : String,
    "Monitoring" : Boolean,
    "NetworkInterfaces" : [ EC2 Network Interface, ... ],
    "PlacementGroupName" : String,
    "PrivateIpAddress" : String,
    "RamdiskId" : String,
    "SecurityGroupIds" : [ String, ... ],
    "SecurityGroups" : [ String, ... ],
    "SourceDestCheck" : Boolean,
    "SubnetId" : String,
    "Tags" : [ Resource Tag, ... ],
    "Tenancy" : String,
    "UserData" : String,
    "Volumes" : [ EC2 MountPoint, ... ],
    "AdditionalInfo" : String
}
```

Sample YAML syntax for the resource Instances.

```
instances:
    *InstanceName*:
        hostname: SampleHostname
        instance_type: m3.xlarge
        ami_id: ami-xxxxxxxx
        ebs_optimized: true
        iam_role: SampleIAMRole
        security_groups:
            - sg-00000001
            - sg-00000002
```

```
        ramdisk_id: SampleRamDiskID
        source_dest_check: true
        network_interfaces:
        Interface1:
        public_ip: true
        delete_on_termination: true
        device_index: 0
        subnet_id: subnet-xxxxxxxx
        private_ips:
            - private_ip: 1.1.1.1
        primary: false
            - private_ip: 2.2.2.2
        primary: true
        block_devices:
            /dev/sda:
                ebs:
                    delete_on_termination: false
                    encrypted: false
                    iops: 1000
                    size: 100
                    type: standard
            /dev/sdb:
                ebs:
                    snapshot_id: snap-xxxxxxxx
            /dev/sdc:
                virtual_name: ephemeral0
            /dev/sdd:
                no_device: true
        volumes:
            - device: ref(SampleVolume1)
              volume_id: /dev/sdd
            - device: vol-xxxxxxx
              volume_id: /dev/sde
        tags:
            - key: Key1
              value: Value1
        user_data:
            file: path/to/script.sh  # Jinja2 Template
            params:
                param1: hello
                param2: world
```

2.1.4 Launch Configuration

The AWS::AutoScaling::LaunchConfiguration type creates an Auto Scaling Launch Configuration that can be used by an Auto Scaling Group to configure Amazon EC2 Instances in the Auto Scaling Group.

JSON Syntax for Launch Configuration.

```
"Type" : "AWS::AutoScaling::LaunchConfiguration",
"Properties" : {
    "AssociatePublicIpAddress" : Boolean,
    "BlockDeviceMappings" : [ BlockDeviceMapping, ... ],
    "ClassicLinkVPCId" : String,
    "ClassicLinkVPCSecurityGroups" : [ String, ... ],
    "EbsOptimized" : Boolean,
    "IamInstanceProfile" : String,
```

```
    "ImageId" : String,
    "InstanceId" : String,
    "InstanceMonitoring" : Boolean,
    "InstanceType" : String,
    "KernelId" : String,
    "KeyName" : String,
    "PlacementTenancy" : String,
    "RamDiskId" : String,
    "SecurityGroups" : [ SecurityGroup, ... ],
    "SpotPrice" : String,
    "UserData" : String
}
```

YAML Syntax for Launch Configuration.

```
launch_configurations:
    *LaunchConfigurationName*:
        instance_type: t2.small
        ebs_optimized: false
        ami_id: ami-xxxxxxxx
        iam_role: SampleIAMRole
        key_pair: SampleKey
        ramdisk_id: SampleRamDiskID
        public_ip: true
        security_groups:
            - sg-00000001
            - sg-00000002
        block_devices:
            /dev/sda:
                ebs:
                    delete_on_termination: false
                    encrypted: false
                    iops: 1000
                    size: 100
                    type: standard
            /dev/sdb:
                ebs:
                    snapshot_id: id-testSnapshot
            /dev/sdc:
                virtual_name: ephemeral0
            /dev/sdd:
                no_device: true
        user_data:
            file: path/to/script.sh  # Jinja2 Template
            params:
                param1: hello
                param2: world
```

2.1.5 Security Group

Creates an Amazon EC2 security group. To create a VPC security group, use the VpcId property. This type supports updates.

JSON Syntax for Security Group.

```
"SampleSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
```

```
  "Properties": {
    "SecurityGroupIngress": [
      {
        "FromPort": 443,
        "IpProtocol": "https",
        "ToPort": 443
      }
    ],
    "VpcId": "SampleVPC",
    "Tags": [
      {
        "Key": "Key1",
        "Value": "Value1"
      }
    ],
    "GroupDescription": "SampleDescription",
    "SecurityGroupEgress": [
      {
        "FromPort": 80,
        "IpProtocol": "http",
        "ToPort": 80
      }
    ]
  }
}
```

YAML Syntax for Security Group.

```
security_group:
  *SecurityGroupName*:
    group_description: SampleDescription
    security_group_egress:
      - from_port: 80
        ip_protocol: http
        to_port: 80
    security_group_ingress:
      - from_port: 443
        ip_protocol: https
        to_port: 443
    tags:
      - key: Key1
        value: Value1
    vpc_id: SampleVPC
```

2.1.6 Lifecycle Hooks

The AWS::AutoScaling::LifecycleHook creates a Lifecycle Hook to control the state of an instance in an Auto Scaling Group after it is launched or terminated. The Auto Scaling Group either pauses the instance after it is launched (before it is put into service) or pauses the instance as it is terminated (before it is fully terminated).

JSON Syntax for Lifecycle Hook.

```
"Type" : "AWS::AutoScaling::LifecycleHook",
"Properties" : {
    "AutoScalingGroupName" : String,
    "DefaultResult" : String,
```

```
    "HeartbeatTimeout" : Integer,
    "LifecycleTransition" : String,
    "NotificationMetadata" : String,
    "NotificationTargetARN" : String,
    "RoleARN" : String
}
```

YAML Syntax for Lifecycle Hook.

```
lifecycle_hooks:
    *LifecycleHookName*:
        asg_name: ref(SampleAutoScaleGroup)
        lifecycle_transition: autoscaling:EC2_INSTANCE_TERMINATING
        notification_target_arn: ref(SampleSNSTopic) # SNS Topic
        role_arn: SampleIAMRole
        heartbeat_timeout: 1800
        default_result: CONTINUE
        notification_metadata: SampleMetadata
```

2.1.7 Load Balancer

The AWS::ElasticLoadBalancing::LoadBalancer type creates a LoadBalancer. In the case where the resource has a public IP address and is also in a VPC that is defined in the same template, you must use the DependsOn attribute to declare a dependency on the VPC-gateway attachment.

Note - You need to have a listener in your load balancer for it to be created successfully. There are 4 types of load balancer protocol that AWS allows you :

- HTTP
- HTTPS
- TCP
- SSL

in respawn we ask of you to use the sample to create your load balancer listener with the second level being the protocol you want to create the listener with. You can repeat the protocol in a list in case you need multiple ports

to attach on that.

JSON Syntax for Load Balancer.

```
"Type": "AWS::ElasticLoadBalancing::LoadBalancer",
"Properties": {
    "AccessLoggingPolicy" : AccessLoggingPolicy,
    "AppCookieStickinessPolicy" : [ AppCookieStickinessPolicy, ... ],
    "AvailabilityZones" : [ String, ... ],
    "ConnectionDrainingPolicy" : ConnectionDrainingPolicy,
    "ConnectionSettings" : ConnectionSettings,
    "CrossZone" : Boolean,
    "HealthCheck" : HealthCheck,
    "Instances" : [ String, ... ],
    "LBCookieStickinessPolicy" : [ LBCookieStickinessPolicy, ... ],
    "LoadBalancerName" : String,
    "Listeners" : [ Listener, ... ],
    "Policies" : [ ElasticLoadBalancing Policy, ... ],
    "Scheme" : String,
    "SecurityGroups" : [ Security Group, ... ],
```

```
    "Subnets" : [ String, ... ],
    "Tags" : [ Resource Tag, ... ]
}
```

YAML Syntax for Load Balancer.

```
load_balancers:
*LoadBalancerName*:
    scheme: internet-facing
    connection_settings:
    idle_timeout: 40
    cross_zone: True
    security_group:
        - sg-xxxxxxx1
        - sg-xxxxxxx2
    instances:
        - ref(SampleInstance)
    policies:
        - policy_name: SamplePolicyName1
    attribute:
        - name: SampleName1
          value: SampleValue1
        - name: SampleName2
          value: SampleValue2
    instance_ports:
        - 2121
        - 2424
    load_balancer_ports:
        - 32323
        - 2424
    policy_type: SSLNegotiationPolicyType
        - policy_name: SamplePolicyName2
    attribute:
        - name: SampleName1
          value: SampleValue1
    instance_ports:
        - 1212
        - 4242
    load_balancer_ports:
        - 23232
        - 4141
    app_cookie_stickiness_policy:
        - policy_name: SamplePolicy1
    cookie_name: SampleCookie1
        - policy_name: SamplePolicy2
    cookie_name: SampleCookie2
    connection_draining_policy:
    enabled: True
    timeout: 10
    availability_zones:
        - "Fn::GetAZs": ""
    health_check:
        healthy_threshold: 2
        interval: 10
        target: /healthcheck
        timeout: 10
        unhealthy_threshold: 2
    lb_cookie_stickiness_policy:
```

```
        - policy_name: SamplePolicyName1
    cookie_expiration_period: 300
        - policy_name: SamplePolicyName2
    cookie_expiration_period: 600
    load_balancer_name: SampleLoadBalancer1 # Unique name used by AWS
    access_logging_policy:
        emit_interval: 20
        enabled: True
    s3_bucket_name: SampleS3BucketName
    s3_bucket_prefix: SampleS3BucketPrefix
    listeners:
        https:
            load_balancer_port: 83
            instance_port: 84
            instance_protocol: tcp
        tcp:
            load_balancer_port: 8443
            instance_port: 8443
            instance_protocol: http
            ssl_certificate_id: SampleSSLARN
    tags:
        - key: Key1
          value: Value1
        - key: Key2
          value: Value2
```

2.1.8 Network Interface

The AWS::EC2::NetworkInterface type creates a network interface for an EC2 Instance.

JSON Syntax for Network Interface.

```
"Type" : "AWS::EC2::NetworkInterface",
"Properties" : {
    "Description" : String,
    "GroupSet" : [ String, ... ],
    "PrivateIpAddress" : String,
    "PrivateIpAddresses" : [ PrivateIpAddressSpecification, ... ],
    "SecondaryPrivateIpAddressCount" : Integer,
    "SourceDestCheck" : Boolean,
    "SubnetId" : String,
    "Tags" : [ Resource Tag, ... ]
}
```

YAML Syntax for Network Interface.

```
network_interfaces:
    *NetworkInterfaceName*:
        description: "Sample Description"
        group_set:
            - SampleGroup1
            - SampleGroup2
        private_ip_address: String
        private_ip_addresses:
            - private_ip: String
              primary: True
            - private_ip: String
```

```
              primary: False
        secondary_private_ip_address_count: 4
        source_dest_check: true
        subnet_id: String
        tags:
            - key: Key1
              value: Value1
            - key: Key2
              value: Value2
```

2.1.9 Network Interface Attachment

The AWS::EC2::NetworkInterfaceAttachment type creates a Network Interface Attachment that attaches additional network interfaces to an EC2 Instance without interruption.

JSON Syntax for Network Interface Attachment.

```
"Type" : "AWS::EC2::NetworkInterfaceAttachment",
"Properties" : {
    "DeleteOnTermination": Boolean,
    "DeviceIndex": String,
    "InstanceId": String,
    "NetworkInterfaceId": String
}
```

YAML Syntax for Network Interface Attachment.

```
network_interface_attachments:
    *NetworkInterfaceAttachmentName*:
        delete_on_termination: False
        device_index: 1
        instance_id: ref(SampleInstanceName)
        network_interface_id: ref(SampleNetworkInterfaceName)
```

2.1.10 RDS

The AWS::RDS::DBInstance type creates a Relation Database Instance.

JSON Syntax for RDS Instance.

```
"Type" : "AWS::RDS::DBInstance",
"Properties" : {
    "AllocatedStorage" : String,
    "AllowMajorVersionUpgrade" : Boolean,
    "AutoMinorVersionUpgrade" : Boolean,
    "AvailabilityZone" : String,
    "BackupRetentionPeriod" : String,
    "CharacterSetName" : String,
    "DBClusterIdentifier" : String,
    "DBInstanceClass" : String,
    "DBInstanceIdentifier" : String,
    "DBName" : String,
    "DBParameterGroupName" : String,
    "DBSecurityGroups" : [ String, ... ],
    "DBSnapshotIdentifier" : String,
```

```
    "DBSubnetGroupName" : String,
    "Engine" : String,
    "EngineVersion" : String,
    "Iops" : Number,
    "KmsKeyId" : String,
    "LicenseModel" : String,
    "MasterUsername" : String,
    "MasterUserPassword" : String,
    "MultiAZ" : Boolean,
    "OptionGroupName" : String,
    "Port" : String,
    "PreferredBackupWindow" : String,
    "PreferredMaintenanceWindow" : String,
    "PubliclyAccessible" : Boolean,
    "SourceDBInstanceIdentifier" : String,
    "StorageEncrypted" : Boolean,
    "StorageType" : String,
    "Tags" : [ Resource Tag, ..., ],
    "VPCSecurityGroups" : [ String, ... ]
}
```

YAML Syntax for RDS Instance.

```
rds:
    *RDSName*:
        allocated_storage: 100
        instance_class: db.m1.small
        engine: MySQL
        allow_major_version_upgrade: True
        allow_minor_version_upgrade: True
        availability_zone: SampleAZ
        backup_retention_period: 10
        character_set_name: UTF8
        instance_identifier: SampleRDSName # Unique name used by AWS
        db_name: SampleDB
        db_parameter_group_name: SampleDBParameterGroup
        db_security_groups:
            - SampleSecurityGroup
        snapshot_identifier: SampleSnapshot
        subnet_group_name: SampleSubnetGroup
        engine: MySQL
        engine_version: 1.0.0
        iops: 1000
        kms_key_id: SampleKMSKeyID
        license_model: SampleLicenseModel
        master_username: SampleUsername
        multi_az: False
        option_group_name: SampleOptionGroup
        port: 3306
        preferred_backup_window: Mon:03:00-Mon:11:00
        preferred_maintenance_window: Tue:04:00-Tue:04:30
        publicly_accessible: False
        source_db_instance_identifier: SampleSourceDBIdentifier
        storage_encrypted: True
        vpc_security_groups:
            - SampleVPCSecurityGroup
```

2.1.11 Scheduled Action

The AWS::AutoScaling::ScheduledAction type creates a scheduled scaling action for an Auto Scaling Group to change the number of Instances available.

JSON Syntax for Scheduled Action.

```
"Type" : "AWS::AutoScaling::ScheduledAction",
"Properties" : {
"AutoScalingGroupName" : String,
    "DesiredCapacity" : Integer,
    "EndTime" : Time stamp,
    "MaxSize" : Integer,
    "MinSize" : Integer,
    "Recurrence" : String,
    "StartTime" : Time stamp
}
```

YAML Syntax for Scheduled Action.

```
scheduled_actions:
    *ScheduledActionName*:
        asg_name: SampleAutoScaleGroup
        desired_capacity: 0
        max_size: 0
        min_size: 0
        recurrence: 0 7 * * *
```

2.1.12 Sns Topic

The AWS::SNS::Topic type creates an Amazon SNS Topic with subscriptions.

JSON Syntax for SNS Topic.

```
"Type" : "AWS::SNS::Topic",
"Properties" : {
    "DisplayName" : String,
    "Subscription" : [ SNS Subscription, ... ],
    "TopicName" : String
}
```

YAML Syntax for SNS Topic.

```
sns_topic:
    *SNSTopicName*:
    display_name : SampleSNSTopic
    topic_name : SampleTopic
    subscription:
        - protocol : https
          endpoint : Endpoint1
        - protocol : http
          endpoint : Endpoint2
```

2.1.13 Volume

The AWS::EC2::Volume type creates a new Amazon Elastic Block Store Volume.

JSON Syntax for Volume.

```
"Type":"AWS::EC2::Volume",
"Properties" : {
    "AvailabilityZone" : String,
    "Encrypted" : Boolean,
    "Iops" : Number,
    "KmsKeyId" : String,
    "Size" : String,
    "SnapshotId" : String,
    "Tags" : [ Resource Tag, ... ],
    "VolumeType" : String
}
```

YAML Syntax for Volume.

```
volumes:
    *SampleVolume:
        availability_zone: SampleAZ
        snapshot_id: snap-xxxxxxxx
        size: 1000
        iops: 4000
        kms_key_id: SampleKMSKeyID
        volume_type: standard
        encrypted: true
        tags:
            - key: Key1
              value: Value1
        deletion_policy: Retain
```

2.1.14 Other Required Keywords

Properties:

```
stack_name: SampleStackName
environment: int
```

2.2 Parameter Index

- *Parameters*

2.2.1 Parameters

Respawn supports String, Integer and Boolean parameters.

YAML Syntax for Parameters

```
parameters:
    *ParameterName*:
        default: String
        type: String
        description: "Sample Description"
        allowed_values:
```

```
            - String
            - String
        allowed_pattern: [A-Za-z0-9]+
        no_echo: true
        max_length: String
        min_length: String
        max_value: String
        min_value: String
        constraint_description: "Parameter must only contain upper and lower case␣
↪letters"
```

2.3 UserData Index

- *UserData*

2.3.1 UserData

Jinja2 template rendered and base64-encoded made available to the Instances and Launch Configurations.

```
user_data:
    file: /path/to/script.sh #Absolute/Relative path to your user data Jinja2␣
↪template.
    params:
        param1: hello
        param2: world
```

2.4 References Index

- *Reference*
- *Get_Attribute*

2.4.1 Reference

References can be specified in the YAML to reference resources created within the template.

```
dimensions:
    - name: SampleName
      value: ref(RefName)
```

2.4.2 Get_Attribute

Get_Attributes can be specified in the YAML to get attributes from resources created within the template.

```
dimensions:
    - name: SampleName
      value: get_att(ResourceName, AttributeName)
```

CHAPTER 3

Source Code Documentation

3.1 Resource Index

- *EC2*
- *Load Balancer*
- *Auto Scaling*
- *CloudFormation*
- *CloudWatch*
- *RDS*
- *Sns Topic*

3.1.1 EC2

3.1.2 Load Balancer

3.1.3 Auto Scaling

3.1.4 CloudFormation

3.1.5 CloudWatch

3.1.6 RDS

3.1.7 Sns Topic

3.2 Parameter Index

3.3 Setup Class Index

CHAPTER 4

Sample YAML

Sample YAML syntax for respawn. : Please note that this contains most of the resources that respawn supports at this moment. We will keep adding on as we keep building resource support.

```yaml
# Globals
 stack_name: sampleStack
 environment: sampleEnvironment
 team: &team sampleTeam
 default_windows_ami: &win_ami sampleAMI
 multi_az: True
 eap: True
 ebs_optimized: &ebs_optimized false
 periodic_chef: false
 service_name: &service sampleServiceName


 parameters:
   testWeb:
     default: String
     type: String
     description: "Creating test param"
     allowed_values:
       - "value1"
       - "value2"
     allowed_pattern: "[A-Za-z0-9]+"
     no_echo: true
     max_length: String
     min_length: String
     max_value: String
     min_value: String
     constraint_description: "Malformed input-Parameter MyParameter must only contain␣
↪upper and lower case letters"


 # Default Security Groups
 SgDevsample: &dev_djin_fcm String
```

```
ELBSubnet: &elb_subnet  String

security_groups:
    Web: &web_sgs

load_balancers:
      SampleLoadBalancer:
        scheme: internet-facing
        connection_settings:
          idle_timeout: 40
        cross_zone: True
        security_group:
          - sg-xxxxxxx1
          - sg-xxxxxxx2
        instances:
          - ref(SampleInstance)
        policies:
           - policy_name: SamplePolicyName1
             attribute:
                - name: SampleName1
                  value: SampleValue1
                - name: SampleName2
                  value: SampleValue2
             instance_ports:
                - 2121
                - 2424
             load_balancer_ports:
                - 32323
                - 2424
             policy_type: SSLNegotiationPolicyType
           - policy_name: SamplePolicyName2
             attribute:
                - name: SampleName1
                  value: SampleValue1
             instance_ports:
                - 1212
                - 4242
             load_balancer_ports:
                - 23232
                - 4141
        app_cookie_stickiness_policy:
          - policy_name: SamplePolicy1
            cookie_name: SampleCookie1
          - policy_name: SamplePolicy2
            cookie_name: SampleCookie2
        connection_draining_policy:
          enabled: True
          timeout: 10
        availability_zones:
          - "Fn::GetAZs": ""
        health_check:
          healthy_threshold: 2
          interval: 10
          target: /healthcheck
          timeout: 10
          unhealthy_threshold: 2
        lb_cookie_stickiness_policy:
          - policy_name: SamplePolicyName1
```

```
            cookie_expiration_period: 300
          - policy_name: SamplePolicyName2
            cookie_expiration_period: 600
        load_balancer_name: SampleLoadBalancer1 # Unique name used by AWS
        access_logging_policy:
          emit_interval: 20
          enabled: True
          s3_bucket_name: SampleS3BucketName
          s3_bucket_prefix: SampleS3BucketPrefix
        listeners:
          https:
            load_balancer_port: 83
            instance_port: 84
            instance_protocol: tcp
          tcp:
            load_balancer_port: 8443
            instance_port: 8443
            instance_protocol: http
            ssl_certificate_id: SampleSSLARN
        tags:
          - key: Key1
            value: Value1
          - key: Key2
            value: Value2

instances:
    SampleInstance:
      hostname: SampleHostname
      instance_type: m3.xlarge
      ami_id: ami-xxxxxxxx
      ebs_optimized: true
      iam_role: SampleIAMRole
      security_groups:
        - sg-00000001
        - sg-00000002
      ramdisk_id: SampleRamDiskID
      source_dest_check: true
      network_interfaces:
        Interface1:
          public_ip: true
          delete_on_termination: true
          device_index: 0
          subnet_id: subnet-xxxxxxxx
          private_ips:
            - private_ip: 1.1.1.1
              primary: false
            - private_ip: 2.2.2.2
              primary: true
      block_devices:
        /dev/sda:
          ebs:
            delete_on_termination: false
            encrypted: false
            iops: 1000
            size: 100
            type: standard
        /dev/sdb:
          ebs:
```

```
            snapshot_id: snap-xxxxxxxx
        /dev/sdc:
          virtual_name: ephemeral0
        /dev/sdd:
          no_device: true
      volumes:
        - device: /dev/sdd
          volume_id: ref(SampleVolume1)
        - device: /dev/sde
          volume_id: vol-xxxxxxx
      tags:
        - key: Key1
          value: Value1
      user_data:
        file: path/to/script.sh  # Jinja2 Template
        params:
          param1: hello
          param2: world


volumes:
    SampleVolume1:
      availability_zone: SampleAZ
      instance: ref(SampleInstance)
      size: 100

    SampleVolume2:
      availability_zone: SampleAZ
      snapshot_id: snap-xxxxxxxx
      size: 1000
      iops: 4000
      kms_key_id: SampleKMSKeyID
      volume_type: standard
      encrypted: true
      tags:
        - key: Key1
          value: Value1
      deletion_policy: Retain

auto_scale_groups:
    SampleAutoScaleGroup:
      hostname: sampleTestName
      availability_zones:
          - AZName1
          - AZName2
      min_size: 1
      max_size: 10
      desired_capacity: 10
      instance_id: ami-xxxxxxxx
      cooldown: 10
      launch_configuration: LaunchConfigName
      load_balancer_names:
          - LBName
          - ref(Sample_LB)
      max_size: 2
      min_size: 1
      metrics_collection:
          - granularity: 1Minute
```

```
          - granularity: 1Minute
            metrics:
                - Metric1
                - Metric2
      notification_configs:
          - notification_type:
                - Type1
                - Type2
            topic_arn: "arn:aws:[service]:[region]:[account]:resourceType/
↪resourcePath"
          - notification_type:
                - Type3
            topic_arn: "arn:aws:[service]:[region]:[account]:resourceType/
↪resourcePath"
      placement_group: PlacementGroupName
      tags:
          - key: Key1
            value: Value1
            propagate_at_launch: true
          - key: Key2
            value: Value2
            propagate_at_launch: true
      termination_policies:
          - Policy1
          - Policy2
      vpc_zone_identifier:
          - ZoneIdentifier1
          - ZoneIdentifier2

 launch_configurations:
     SampleLaunchConfiguration:
      instance_type: t2.small
      ebs_optimized: false
      ami_id: ami-xxxxxxxx
      iam_role: SampleIAMRole
      key_pair: SampleKey
      ramdisk_id: SampleRamDiskID
      public_ip: true
      security_groups:
        - sg-00000001
        - sg-00000002
      block_devices:
        /dev/sda:
          ebs:
            delete_on_termination: false
            encrypted: false
            iops: 1000
            size: 100
            type: standard
        /dev/sdb:
          ebs:
            snapshot_id: id-testSnapshot
        /dev/sdc:
          virtual_name: ephemeral0
        /dev/sdd:
          no_device: true
      user_data:
          file: path/to/script.sh  # Jinja2 Template
```

```
          params:
            param1: hello
            param2: world

lifecycle_hooks:
  SampleLifecycleHook:
      asg_name: ref(SampleAutoScaleGroup)
      lifecycle_transition: autoscaling:EC2_INSTANCE_TERMINATING
      notification_target_arn: ref(SampleSNSTopic) # SNS Topic
      role_arn: SampleIAMRole
      heartbeat_timeout: 1800
      default_result: CONTINUE
      notification_metadata: SampleMetadata


scheduled_actions:
  SampleActionDown:
      asg_name: SampleAutoScaleGroup
      desired_capacity: 0
      max_size: 0
      min_size: 0
      recurrence: 0 7 * * *

  SampleActionUp:
      asg_name: SampleAutoScaleGroup
      desired_capacity: 5
      max_size: 5
      min_size: 5
      recurrence: 0 9 * * *

rds:
  SampleRDS:
    allocated_storage: 100
    instance_class: db.m1.small
    engine: MySQL
    allow_major_version_upgrade: True
    allow_minor_version_upgrade: True
    availability_zone: SampleAZ
    backup_retention_period: 10
    character_set_name: UTF8
    instance_identifier: SampleRDSName # Unique name used by AWS
    db_name: SampleDB
    db_parameter_group_name: SampleDBParameterGroup
    db_security_groups:
        - SampleSecurityGroup
    snapshot_identifier: SampleSnapshot
    subnet_group_name: SampleSubnetGroup
    engine: MySQL
    engine_version: 1.0.0
    iops: 1000
    kms_key_id: SampleKMSKeyID
    license_model: SampleLicenseModel
    master_username: SampleUsername
    multi_az: False
    option_group_name: SampleOptionGroup
    port: 3306
    preferred_backup_window: Mon:03:00-Mon:11:00
    preferred_maintenance_window: Tue:04:00-Tue:04:30
```

```
    publicly_accessible: False
    source_db_instance_identifier: SampleSourceDBIdentifier
    storage_encrypted: True
    vpc_security_groups:
        - SampleVPCSecurityGroup

network_interfaces:
  SampleNetworkInterface:
    description: "Sample Description"
    group_set:
      - SampleGroup1
      - SampleGroup2
    private_ip_address: 10.20.03.20
    private_ip_addresses:
     - 10.23.23.23
     - 12.13.3.4
    secondary_private_ip_address_count: 4
    source_dest_check: true
    subnet_id: 131.3.13.1
    tags:
        - key: Key1
          value: Value1
        - key: Key2
          value: Value2

network_interface_attachments:
     TestNetworkIntefaceAttachment:
       delete_on_termination: False
       device_index: 1
       instance_id: ref(SampleInstanceName)
       network_interface_id: ref(SampleNetworkInterfaceName)

sns_topics:
  SampleSNSTopic:
    display_name : SampleSNSTopic
    topic_name : SampleTopic
    subscription:
      - protocol : https
        endpoint : Endpoint1
      - protocol : http
        endpoint : Endpoint2

cloud_watch:
  SampleCloudWatch:
    actions_enabled: true
    alarm_actions:
      - AlarmAction1
      - AlarmAction2
    alarm_name: SampleAlarm
    alarm_description: "Sample alarm description"
    comparison_operator: GreaterThanOrEqualToThreshold
    dimensions:
      - name: Dimension1
        value: Value1
      - name: Dimension2
        value: Value2
    evaluation_periods: 15
    insufficient_data_actions:
```

```
      - InsufficientDataAction1
      - InsufficientDataAction2
    metric_name : SampleName
    namespace : SampleNamespace
    ok_actions :
      - OkAction1
      - OkAction2
    period : 12
    statistic : Average
    threshold : 10
    unit : Milliseconds

security_group:
  SampleSecurityGroup:
    group_description: SampleDescription
    security_group_egress:
      - from_port: 80
        ip_protocol: http
        to_port: 80
    security_group_ingress:
      - from_port: 443
        ip_protocol: https
        to_port: 443
    tags:
      - key: Key1
        value: Value1
    vpc_id: SampleVPC
```

CHAPTER 5

Overview

Infrastructure templates and utilities for building AWS CloudFormation stacks. Respawn uses cfn-pyplates to generate CloudFormation templates. A pyplate is a class-based python representation of a JSON CloudFormation template and resources, with the goal of generating CloudFormation templates based on input python templates (pyplates!) that reflect the CloudFormation template hierarchy.

Respawn is a Python package that provides interfaces to Amazon Web Services - Cloudformation. It allows for easier and more user friendly and concise YAML keywords to create resources/parameters/userdata in CloudFormation stacks. This is used in Dow Jones professional information business pipeline and with success and has been modified to be as generic and serve all. Currently the library supports Python 2.7 because of its dependency on cfn-pyplates.

CHAPTER 6

Summary

Respawn is template and utility for spawning AWS CloudFormation stacks from simpler YAML specifications. Respawn will consume a YAML file with documented keywords and spit out a CloudFormation stack json specification.

CHAPTER 7

Key Features

The key features of Respawn are:

- Automatic CloudFormation creation: Respawn detects your application type and builds a CloudFormation JSON for your application tailored to your use based on your YAML. It supports multiple resources/parameters/user-data that AWS supports. Please go through usage to see the list of resources respawn supports.
- Validates CloudFormation: Respawn validates the JSON created against AWS resources to confirm the correctness of your CloudFormation script. It utilizes boto3 and AWS credentials stored in your environment.