AWS and Ansible

Automating Scalable (and Repeatable) Architecture

Timothy Appnel, Principal Product Manager, Ansible by Red Hat
David Duncan, Partner Solutions Architect, Amazon Web Services
Ryan Brown, Senior Software Engineer, Ansible by Red Hat

May 4, 2017
SIMPLE  POWERFUL  AGENTLESS
DATA CENTER-LESS
SERVERLESS
AWS re:Invent Game Day 2016

(AWS + Ansible + Fun + Vegas!)
A fun way to learn
A fun way to learn
Best Practices
The Debrief
Deliver Results
AWS & ANSIBLE INTEGRATION
Out of the box, Ansible has dynamic EC2 inventory support and nearly 100 modules supporting AWS capabilities, including:

- AMI Management
- Autoscaling Groups
- CloudFormation
- CloudTrail
- CloudWatch
- DynamoDB
- ElastiCache
- Elastic Block Store (EBS)
- Elastic Cloud Compute (EC2)
- EC2 Container Service (ECS)
- Elastic IPs (EIP)
- Elastic Load Balancers (ELB)
- Identity Access Manager (IAM)
- Kinesis
- Lambda
- Redshift
- Relational Database Service
- Route53
- Security Groups
- Security Token Service
- Simple Storage Service (S3)
- Virtual Private Cloud (VPC)
Database Modules

Influxdb

- influxdb_database - Manage InfluxDB databases
- influxdb_retention_policy - Manage InfluxDB retention policies

Misc

- elasticsearch_plugin - Manage Elasticsearch plugins
- kibana_plugin - Manage Kibana plugins
- redis - Various redis commands, such as:"
MORE BATTERIES SOON

● Serverless Application Model (SAM)
● Direct Connect
● Data Pipeline
● Key Management Service (KMS)
● Lightsail
ANSIBLE TOWER INTEGRATION

- Dynamic Inventory
- Credentials Management
- Provisioning Callbacks (Autoscaling)
- Tower CLI tool (such as CodeDeploy integration)
RED HAT CLOUDFORMS

Deliver Services Across Your Hybrid Cloud

- Hybrid Cloud Management
- Self-Service Provisioning
- Policy-Driven Compliance
ANSIBLE & SERVERLESS
AGENTLESS TO SERVERLESS

- Network device connection plugins
- Container connection plugins
- Cloud APIs as another plugin
  - Lambda
  - DynamoDB
  - RDS
  - Route53
  - CloudFront
ANSIBLE ALL-IN

- Ansible skills 100% transferrable
- Add to existing playbooks
- Great for “Lambda Light” workloads
EXAMPLE

- lambda:
  
  name: sendReportMail
  zip_file: "{{ deployment_package }}"
  runtime: python2.7
  timeout: 20
  handler: handler.handler
  memory_size: 1024
  role: "{{ iam_exec_role }}"

  register: lambda
- cloudwatchevent_rule:
  description: "Send out incident summary daily"
  name: incident_summary_schedule
  schedule_expression: 'rate(1 day)'
  targets:
  - id: SendSummary
    arn: "{{ lambda.configuration.function_arn }}"
    input: '{“Test”: “Cronjob”}'
INVOKE

- execute_lambda:
  name: myNewFunction
  payload:
    arbitrary: json
    can: “go here”
  wait: true
DOWNSIDE

- Not all event sources supported (yet)
- Ansible release cycle doesn’t follow Lambda/API Gateway releases
ANSIBLE + SERVERLESS FRAMEWORK

Best of both worlds

- Serverless Framework primarily outputs CloudFormation
- Resources from Serverless can be gathered by Ansible
- Optimized for Lambda+event sources
ANSIBLE + CLOUDFORMATION

Best of both worlds

- cloudformation:
  
  stack_name: prod-vpc
  state: present
  template: base_vpc.yml

- cloudformation_facts:
  
  stack_name: external_stack
  register: stack_info
ANSIBLE + CLOUDFORMATION

In base_vpc.yaml

Outputs:

VpcId:
  Value: !Ref AppVPC
  Export: {Name: !Sub "${Stage}-VPCID"}

SubnetA:
  Value: !Ref SubnetA

SubnetB:
  Value: !Ref SubnetB
ANSIBLE + SERVERLESS

- serverless_deploy:
  stage: prod
  region: us-east-1
- cloudformation_facts:
  stack_name: petstore-prod
  register: sls_stack
ANSIBLE + SERVERLESS FRAMEWORK

Migration planning

- Attention to detail - who owns resources needed by both?
- Make sure data is close enough to new & old systems
- Don’t couple serverless & legacy too tightly
- Remember Ansible can populate Serverless variable files
ANSIBLE + AWS SAM

Soon...

- Serverless Application Model is mostly transforms in front of CloudFormation
- Planned for 2.4 as extension to CloudFormation module
- Expect same facts/vars/params to work with SAM
DEMONS & WALKTHROUGHS
AWS & ANSIBLE NEXT STEPS

Would you like to dig deeper with Ansible? It’s easy to get started:
ansible.com/get-started

The power of AWS meets the simplicity of Ansible:
ansible.com/aws

Because container orchestration is never just container orchestration:
ansible.com/containers

Have you used Ansible already? Try Tower for free:
ansible.com/tower-trial
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos
facebook.com/redhatinc
twitter.com/RedHatNews
LEARN. NETWORK. EXPERIENCE OPEN SOURCE.