Masterless SaltStack at Scale

Ryan Lane - Mar 5, 2015

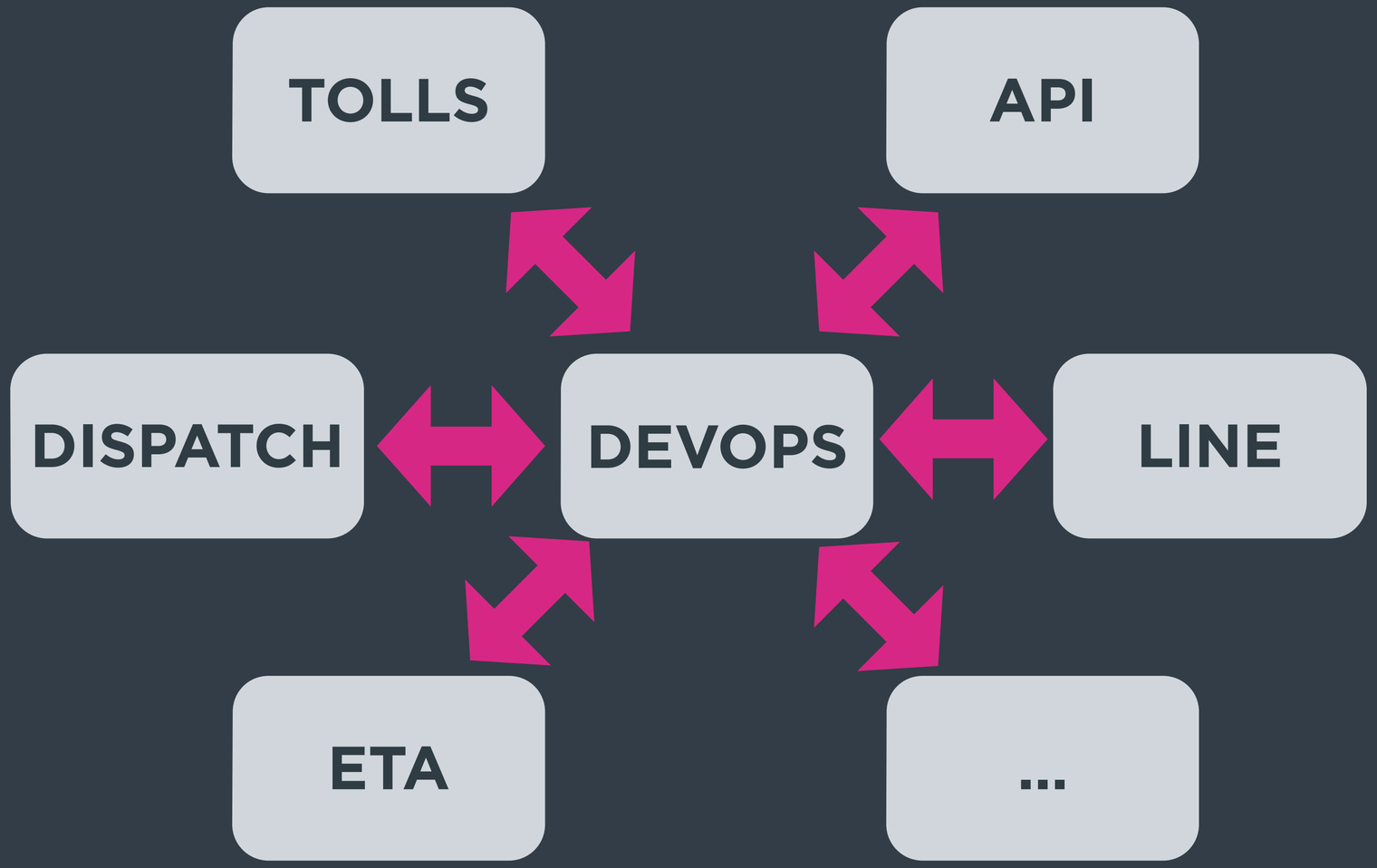
Ryan Lane

DevOps Engineer

Engineering culture

If you build it, you run it.

DevOps as consultants



Constraints Culture

service
+
base (shared)

Constraints Culture

Infrastructure as code

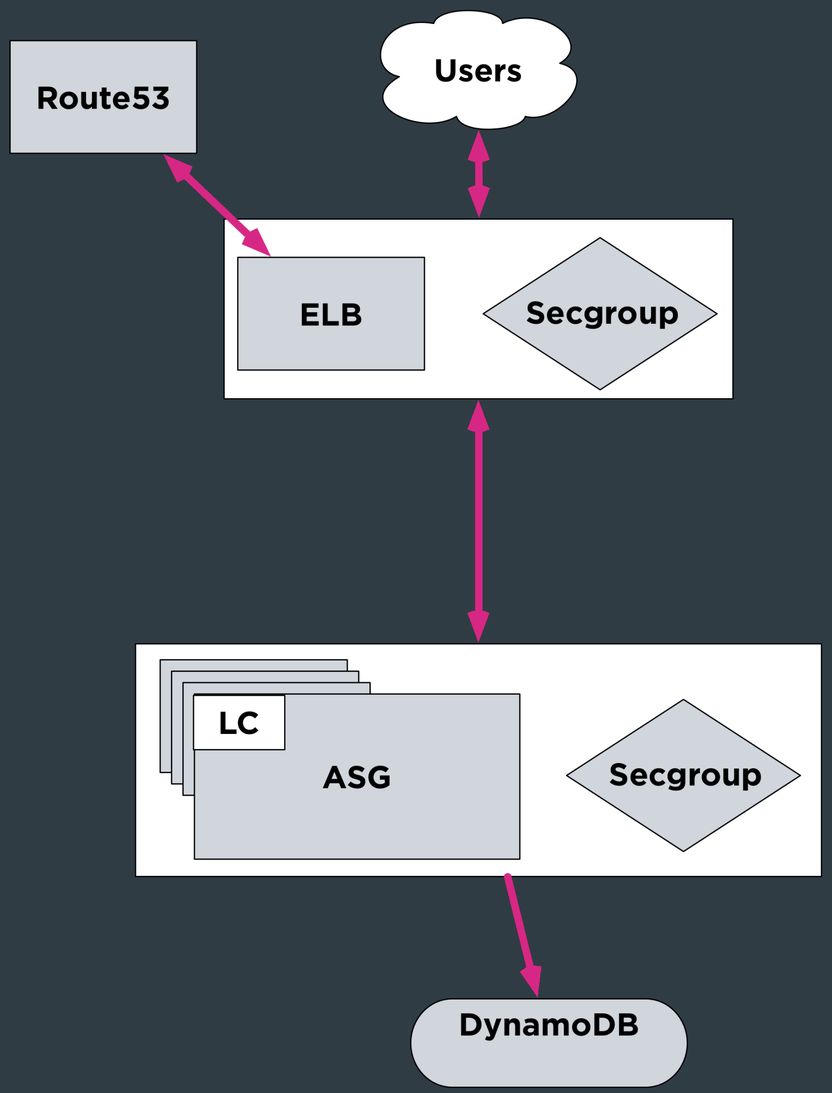
Constraints Scalability/Availability

No masters

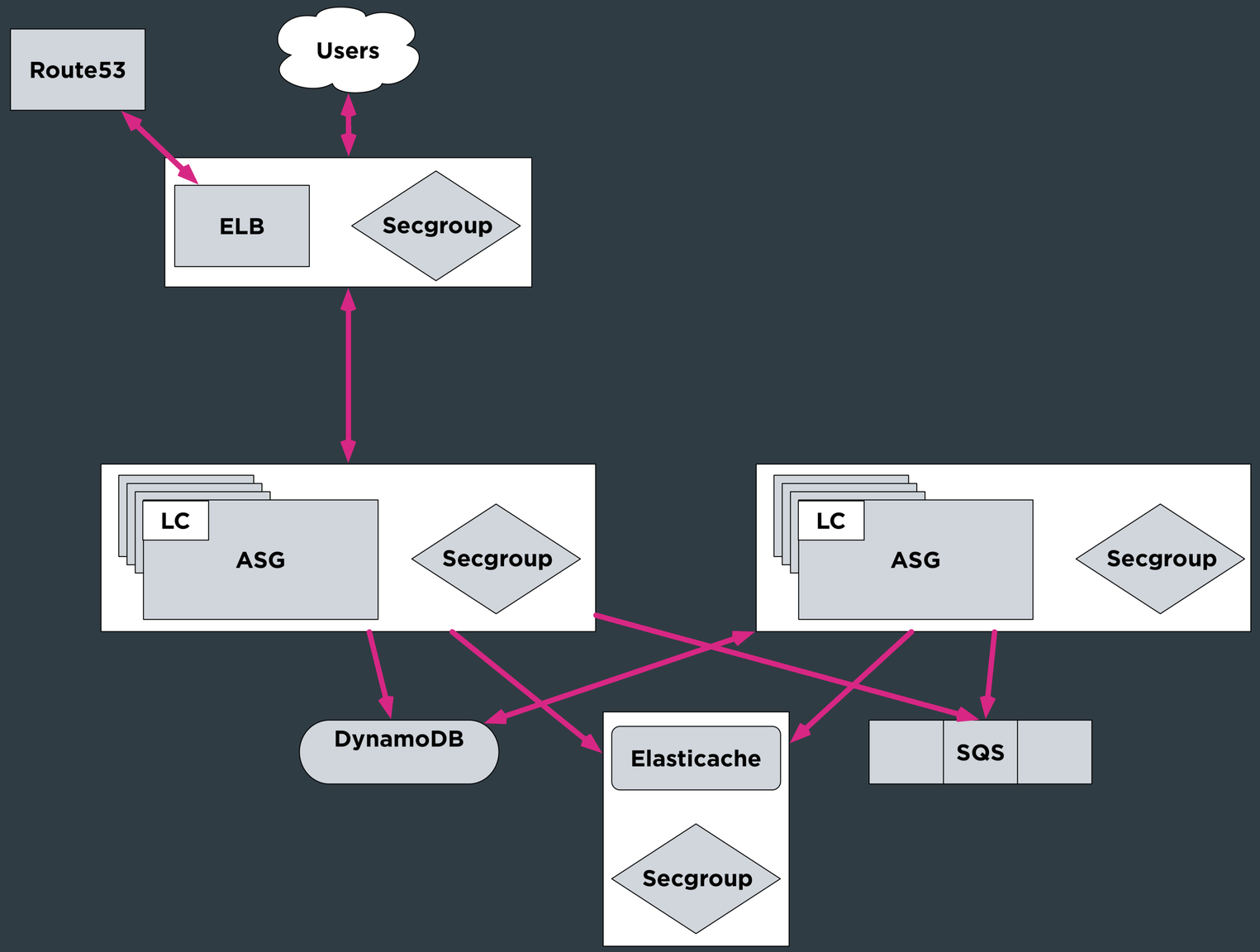
Constraints Consistency

production
staging
development
...

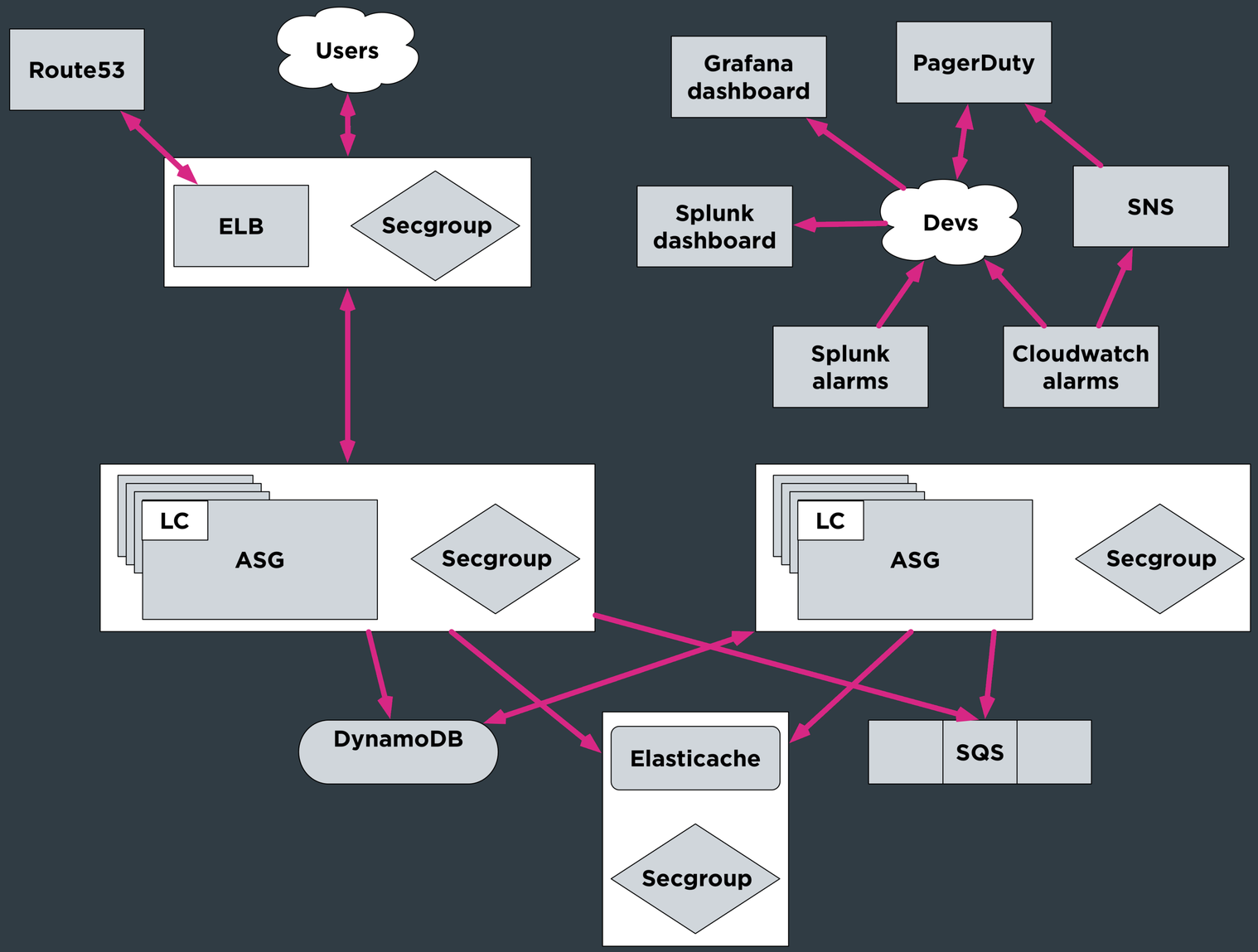
Example service



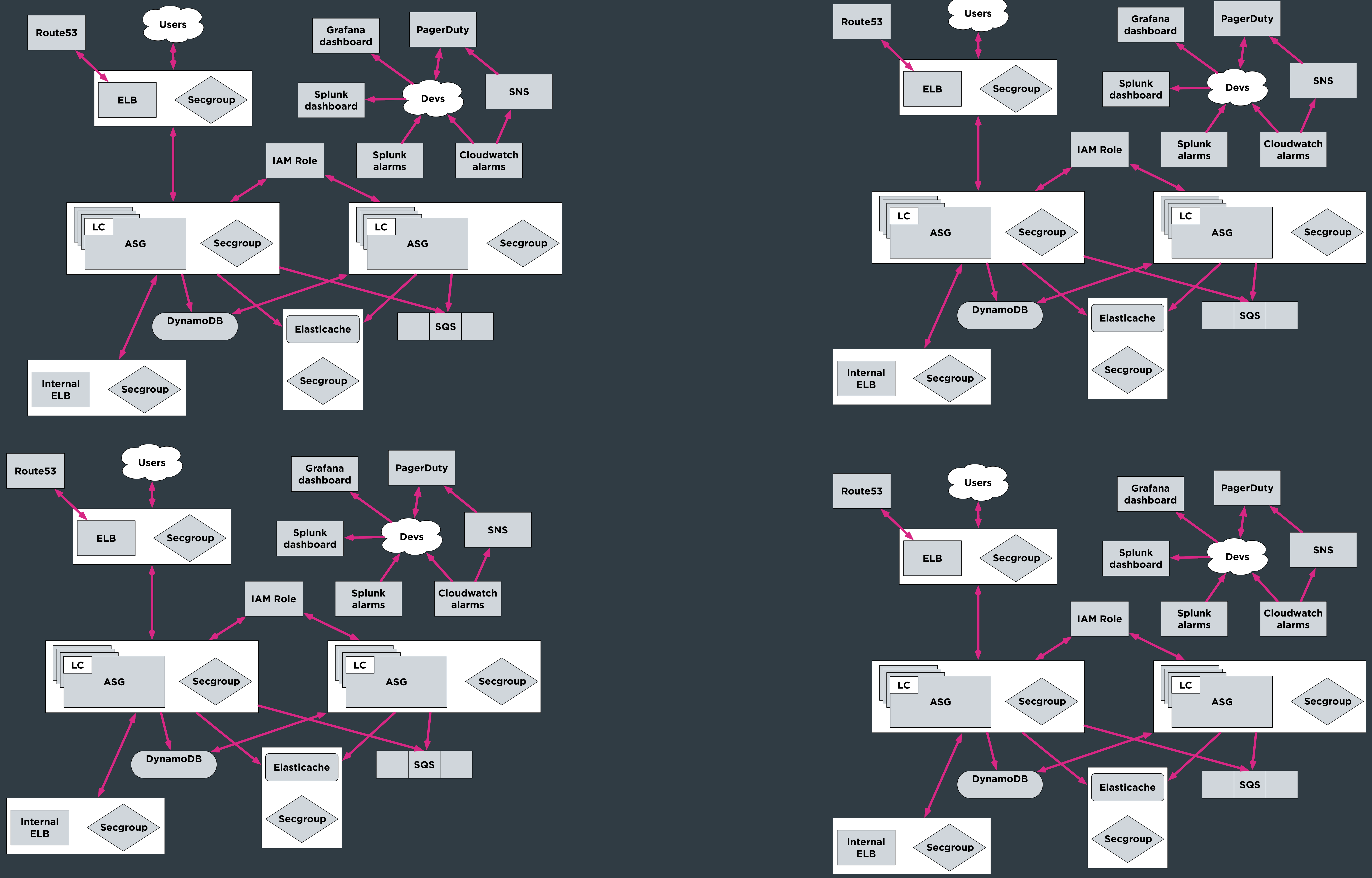
Example service



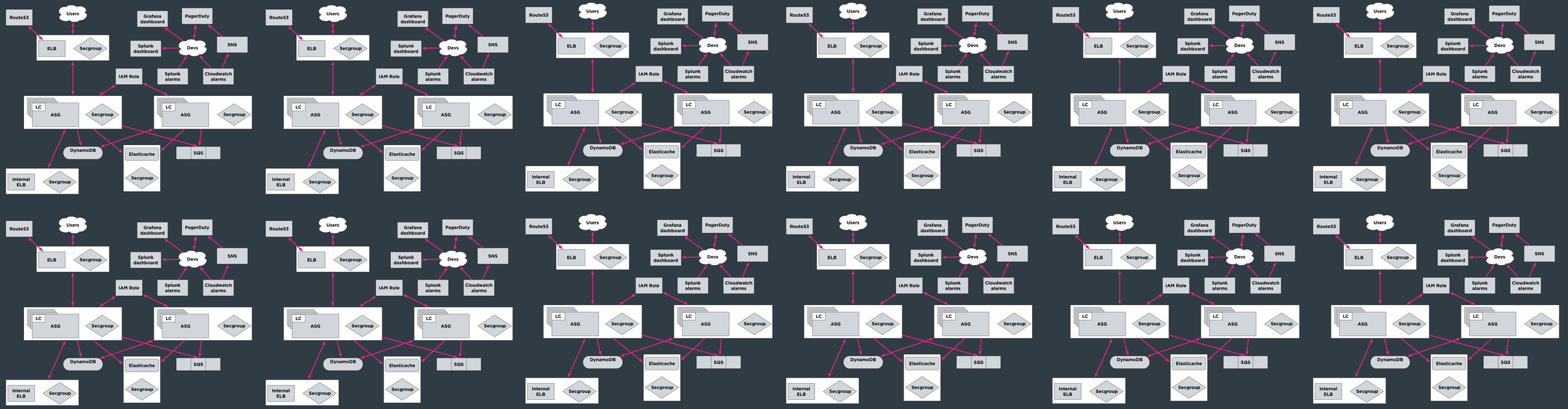
Example service



Example service Multiple environments



Example services Multiple environments



Repo layout

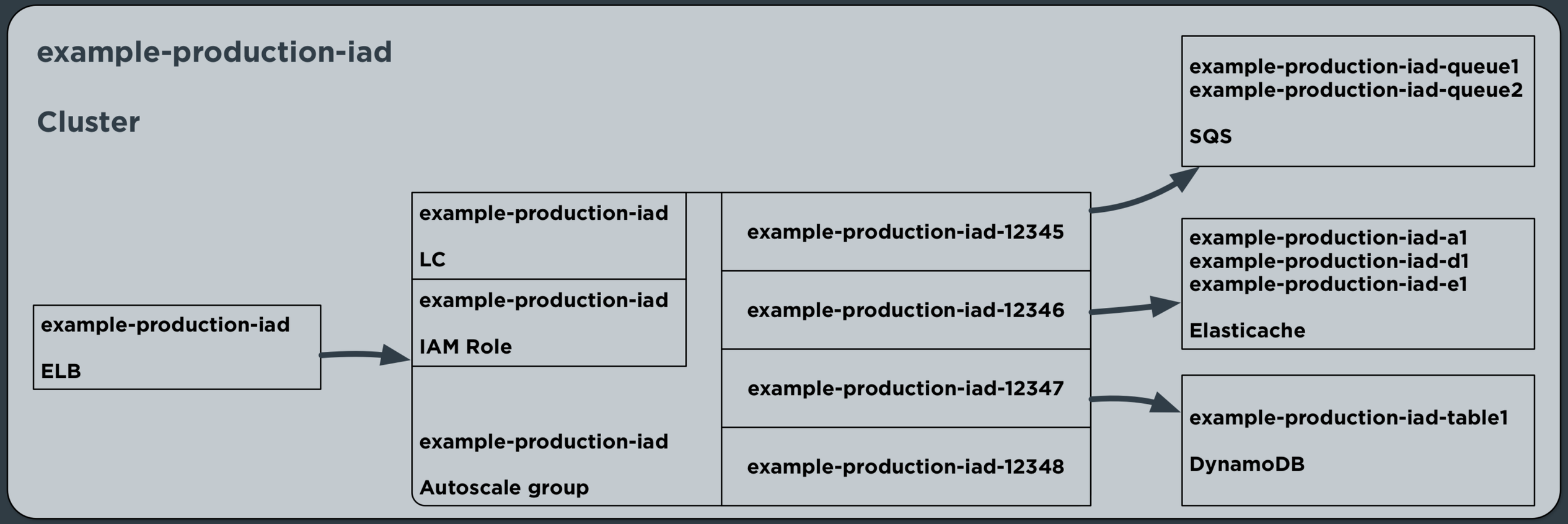
Makefile
salt/orchestration/states
salt/orchestration/pillar
salt/orchestration/modules
salt/config/states
salt/config/pillar
salt/config/modules

Naming conventions and custom grains

example-production-iad-12345.example.com

service_name: example
service_instance: production
region: iad
service_node: 12345
domain: example.com
service_group: example-production
cluster_name: example-production-iad

Naming conventions



Orchestration IAM (roles)

```
Ensure {{ grains.cluster_name }} iam role exists:
  boto_iam_role.present:
    - name: {{ grains.cluster_name }}
    - policies_from_pillars:
        - bootstrap
    - policies:
        sqs:
          Version: '2012-10-17'
          Statement:
            - Action:
                - 'sqs:*'
              Effect: 'Allow'
              Resource:
                - ‘arn:aws:sqs:*:*:{{ grains.cluster_name }}-*'
              Sid: 'sqs1'
```

Orchestration Secgroup

```
Ensure {{ grains.service_name }} secgroup exists:
  boto_secgroup.present:
    - name: {{ grains.service_name }}
    - description: {{ grains.service_name }}
    - rules:
      - ip_protocol: tcp
        from_port: 443
        to_port: 443
        source_group_name: elb
   - vpc_id: vpc-1234
```

Orchestration ELB

```
Ensure {{ grains.cluster_name }} elb exists:
  boto_elb.present:
    - name: {{ grains.cluster_name }}
    - subnets: [subnet-1234, subnet-1235, subnet-1236]
    - security_groups:
      - {{ grains.cluster_name }}
    - listeners:
        - elb_port: 443
          instance_port: 80
          elb_protocol: HTTPS
          instance_protocol: HTTP
          certificate: 'arn:aws:iam::12345:server-certificate/mycert'
    - health_check:
        target: ‘HTTP:80/check'
    - attributes_from_pillars: boto_elb_attributes
    - cnames:
      - name: {{ grains.cluster_name }}.{{ grains.domain }}.
        zone: {{ grains.domain }}.
```

Orchestration Cloudwatch ELB

```
boto_elb_alarms:
  UnHealthyHostCount:
    name: 'ELB UnHealthyHostCount **MANAGED BY SALT**'
    attributes:
      metric: UnHealthyHostCount
      namespace: AWS/ELB
      statistic: Average
      comparison: '>='
      threshold: 1.0
      period: 600
      evaluation_periods: 6
      unit: null
      description: ELB UnHealthyHostCount >= 1
      alarm_actions: ['arn:aws:sns:us-east-1:12345:alarm-myservice']
      ok_actions: ['arn:aws:sns:us-east-1:12345:alarm-myservice']
      insufficient_data_actions: []
```

Orchestration ASG + LC

```
Ensure {{ grains.cluster_name }} asg exists:
  boto_asg.present:
    - name: {{ grains.cluster_name }}
    - launch_config_name: {{ grains.cluster_name }}
    - launch_config:
      - image_id: {{ pillar.ec2_ami.iad.ubuntu14.instance }}
      - key_name: example-key
      - security_groups: [base, {{ grains.service_group }}]
      - instance_type: c3.large
      - associate_public_ip_address: true
      - cloud_init:
          scripts:
            salt: |
              {{ pillar.salt_bootstrap | indent(14)}}
    - vpc_zone_identifier: [subnet-1234, subnet-1235, subnet-1236]
    - availability_zones: [us-east-1a, us-east-1d, us-east-1e]
    - min_size: 12
...
```

Orchestration ASG + LC

```
...
  - max_size: 12
  - tags:
    - key: 'Name'
      value: '{{ grains.cluster_name }}'
      propagate_at_launch: true
  - scaling_policies:
    - name: ScaleDown
       adjustment_type: ChangeInCapacity
       scaling_adjustment: -5
       cooldown: 1800
    - name: ScaleUp
       adjustment_type: ChangeInCapacity
       scaling_adjustment: 15
       cooldown: 1800
```

Orchestration Cloudwatch ASG

```
boto_asg_alarms:
    CPU:
      name: 'ASG ScaleUp CPU **MANAGED BY SALT**'
      attributes:
          metric: CPUUtilization
          namespace: AWS/EC2
          statistic: Average
          comparison: '>='
          threshold: 65.0
          period: 60
          evaluation_periods: 30
          unit: null
          description: 'ASG CPU'
          alarm_actions: ['scaling_policy:{{ grains.cluster_name }}:ScaleUp']
          insufficient_data_actions: []
          ok_actions: []
```

Orchestration Splunk alarms

```
Ensure error alarm is present:
  splunk_search.present:
    - name: Errors in log file
    - action.email.format: plain
    - action.email.inline: '1'
    - action.email.sendresults: 'True'
    - action.email.to: example@myorg.pagerduty.com
    - actions: email
    - alert.expires: 1d
    - alert.severity: '4'
    - alert.suppress: '1'
    - alert.suppress.period: 30m
    - alert.track: '1'
    - alert_comparator: greater than
    - alert_threshold: '0'
    - alert_type: number of events
    - cron_schedule: '*/5 * * * *'
    - description: '**MANAGED** Errors in log file'
    - dispatch.earliest_time: -6m
    - dispatch.latest_time: -1m
    - dispatch.ttl: 1p
    - is_scheduled: '1'
    - search: host="example-production-iad*" source="/var/log/.../error.log"
```

Orchestration Grafana

```
Ensure {{ grains.service_name }} grafana dashboard is
managed:
  grafana.dashboard_present:
    - name: {{ grains.service_name }}
    - dashboard_from_pillar: 'grafana_dashboards:default'
    - rows_from_pillar:
      - 'grafana_rows:service'
      - 'grafana_rows:elb'
      - ‘grafana_rows:systemhealth’
```

testme

Zoom Out | 2 hours ago to a few seconds ago

PANELS/ROWS MARKED WITH (M) ARE MANAGED BY ORCHESTRATION. DON'T EDIT THEM!

testme (M)

The testme service (example/testme.git)

Notify On-Call

Call Volume (M)

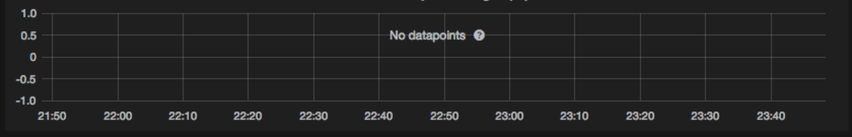
Response Time (M)

Error Rate (M)

ELB TRAFFIC (EXTERNAL) (M)

External ELB traffic (2xx,5xx) (M)

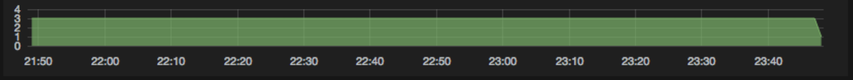
External ELB queue length (M)



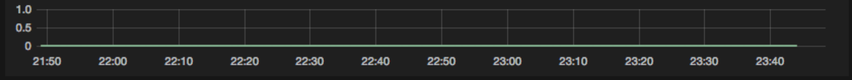
External ELB Latency (M)

ELB HEALTH (M)

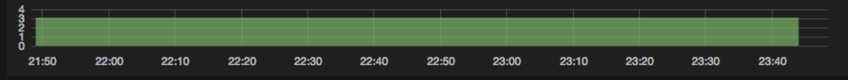
instance count (total) (M)



Unhealthy Instances by AZ (ELB) (M)

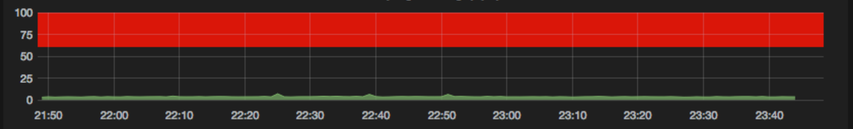


Healthy Instances (ELB) (M)

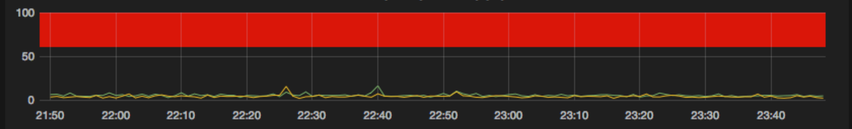


SYSTEM HEALTH (M)

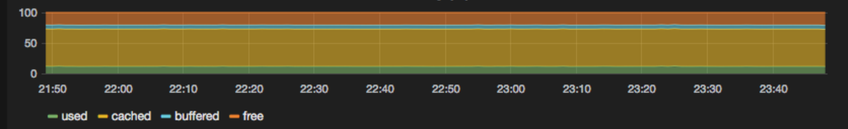
CPU (asg average) (M)



CPU (max per CPU) (M)

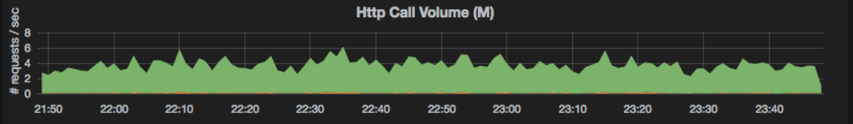


Memory (M)



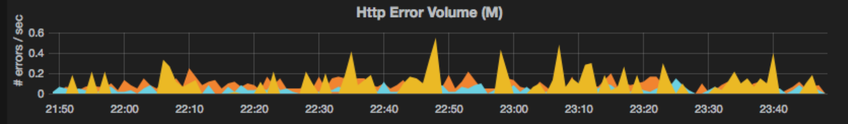
HTTP TRAFFIC (M)

Http Call Volume (M)



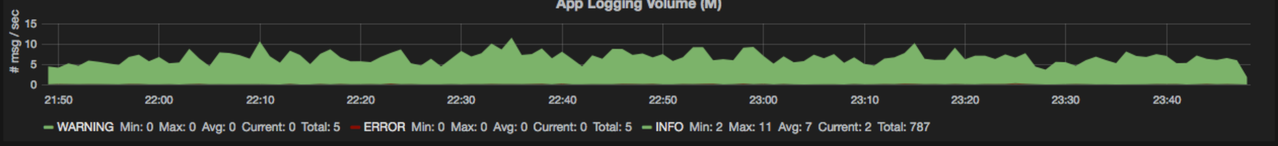
Http Response Time (p95) (M)

Http Error Volume (M)

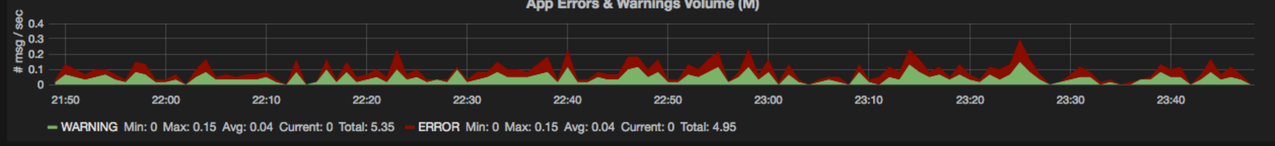


APP LOGGING (M)

App Logging Volume (M)



App Errors & Warnings Volume (M)



Orchestration Modules

Autoscale groups
Cloudwatch
DynamoDB
Elasticache
ELB
Grafana
IAM (roles)
Launch config
MMS*
Route53
Security groups
SNS
Splunk
SQS

* Not yet upstreamed

Orchestration

Custom grain + wrapper

env SERVICE_NAME=example SERVICE_INSTANCE=production salt-wrapper.sh state.sls example

Salt config generated by wrapper

```
root_dir: $TEMPDIR/
pki_dir: $TEMPDIR/pki/minion
cachedir: $TEMPDIR/cache/minion
pillar_roots:
  base:
    - $ORCHESTRATION_DIR/pillar
ipc_mode: tcp
file_client: local
local: true
file_roots:
  base:
    - salt/orchestration
module_dirs:
  - $ORCHESTRATION_DIR/modules
```

Config management

Salt config

```
failhard: True
state_output: mixed
state_verbose: False
log_level: info
file_client: local
file_roots:
  base:
    - /srv/service/next/salt/config/states
    - /srv/base/next/salt/config/states
pillar_roots:
  base:
    - /srv/service/next/salt/config/pillar
    - /srv/base/next/salt/config/pillar
module_dirs:
  - /srv/service/next/salt/config/modules
  - /srv/base/next/salt/config/modules
ext_pillar:
  <redacted>
```

Pillar Top

```
base:
  '*':
    - base
    - {{ grains.service_name }}
  'service_group:{{ grains.service_group }}':
    - match: grain
    - {{ grains.service_group }}
    - ignore_missing: true
```

State Top

```
base:
  '*':
    - base
    - {{ grains.service_name }}
```

Config management References

```
{% set local = ‘local-{{ grains.service_instance }}-{{ grains.region }}-
{{ grains.service_node }}.{{ grains.domain }}’ %}
environment:
  development:
    DYNAMODB_URL: ‘http://{{ local }}:8000’
    DYNAMODB_TABLE: {{ grains.cluster_name }}-exampletable
    REDIS_URL: ‘redis://{{ local }}:6379’
```

Deployment

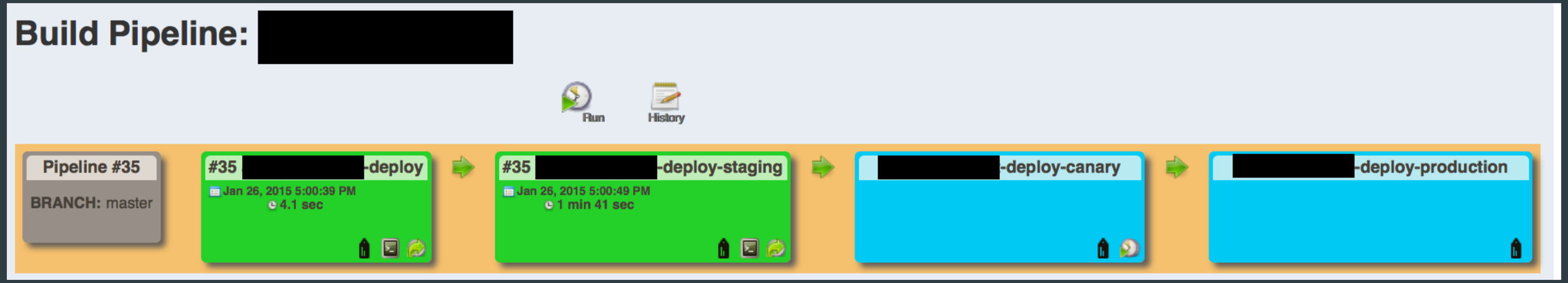
- Make feature branch
- Get review
- Ensure tests pass
- Merge PR
- Start deployment via jenkins pipeline

Deployment

- Generate artifact
- Push artifact to S3
- Release to environment
- Run orchestration

- Fetch artifacts
- Switch ‘next’ links to current SHAs
- Run pre-deploy hook
- Run Salt
- Switch ‘current’ links to current SHAs
- Run post-deploy hook
- Report status

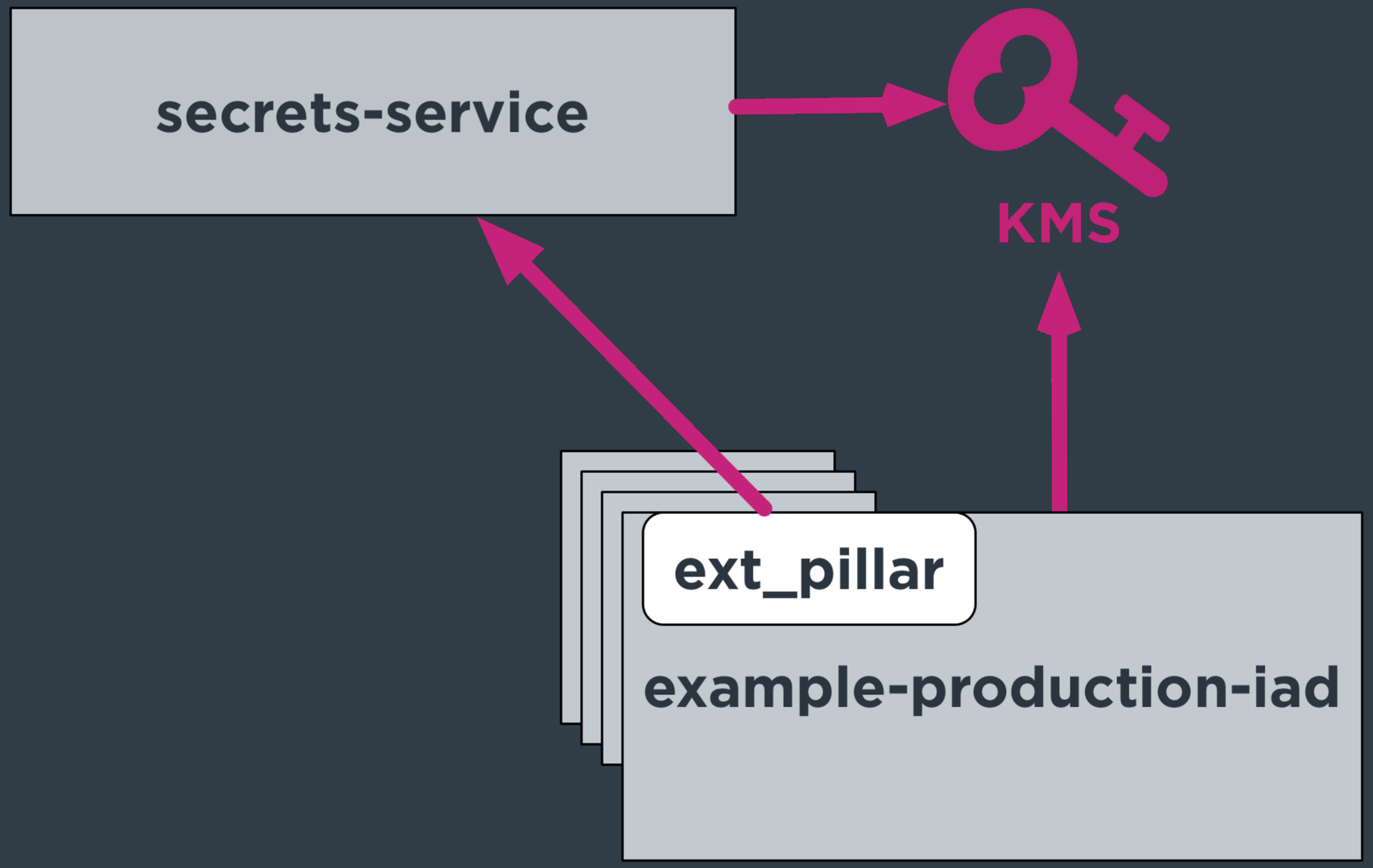
Deployment


Build Pipeline: [redacted]
Run
History
Pipeline #35
BRANCH: master
#35 [redacted]-deploy
Jan 26, 2015 5:00:39 PM
4.1 sec
#35 [redacted]-deploy-staging
Jan 26, 2015 5:00:49 PM
1 min 41 sec
[redacted]-deploy-canary
[redacted]-deploy-production

Masterless issues

Remote execution
Vulnerability remediation
Vulnerability tracking
Service discovery

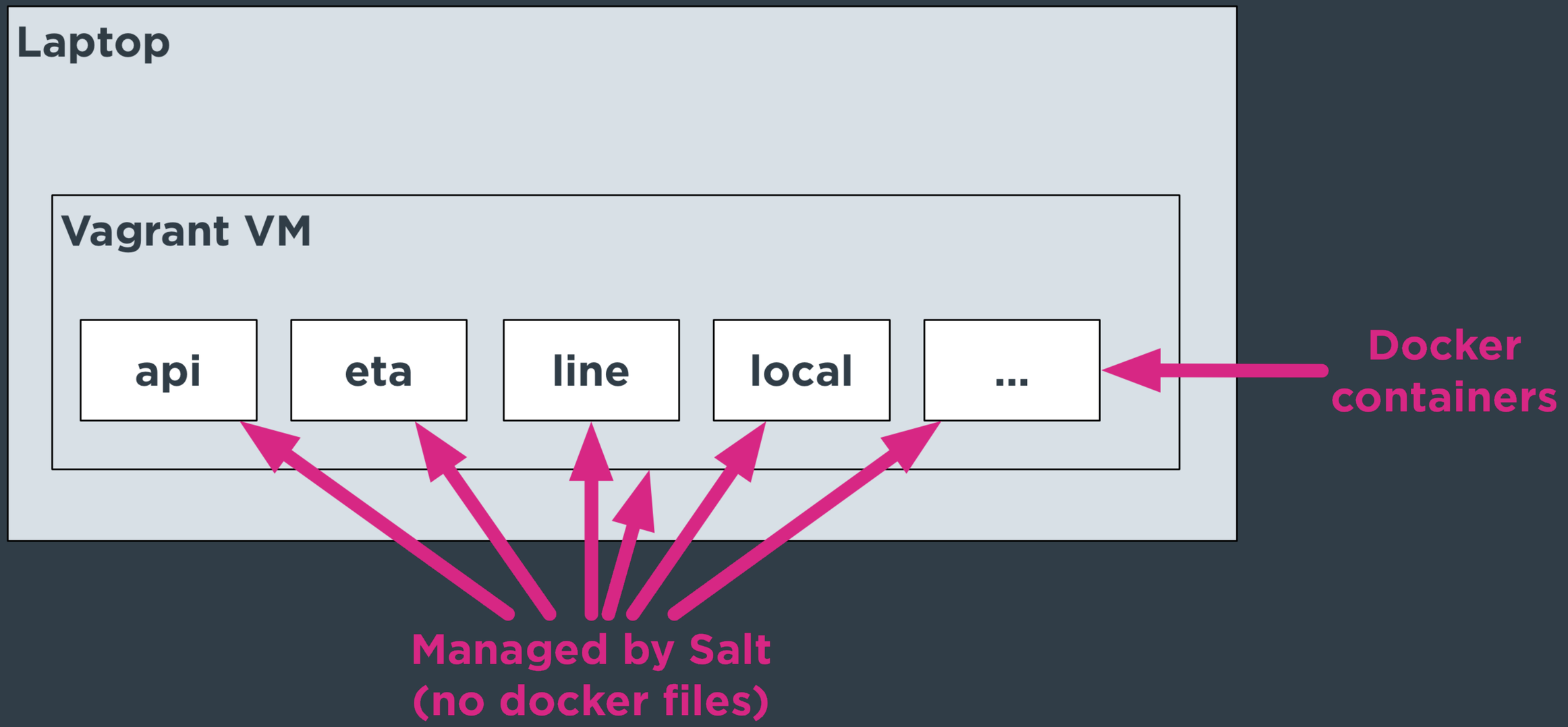
Masterless issues Secret management



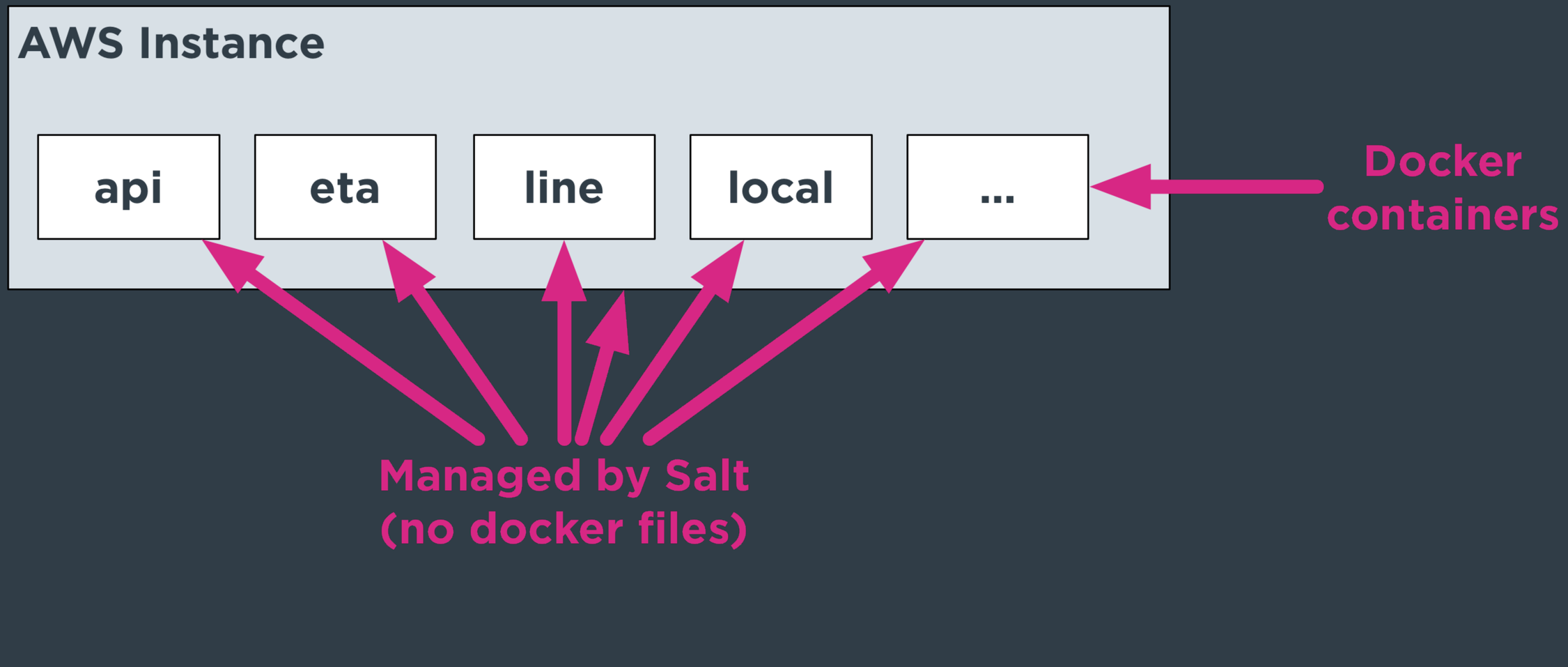
Masterless wins

Node registration
Master scaling
Master HA
Secret management

Development



Test/CI


AWS Instance
api
eta
line
local
...
Docker containers
Managed by Salt
(no docker files)

Docker + Salt

Docker module on host
Phusion base image
Lyft base image
Run + commit = service image
Tags for versioning

Thank You.

@squiddlane

rlane@lyft.com