

The Silence of the Lambda

Automate Windows AMI updates to stop the screaming

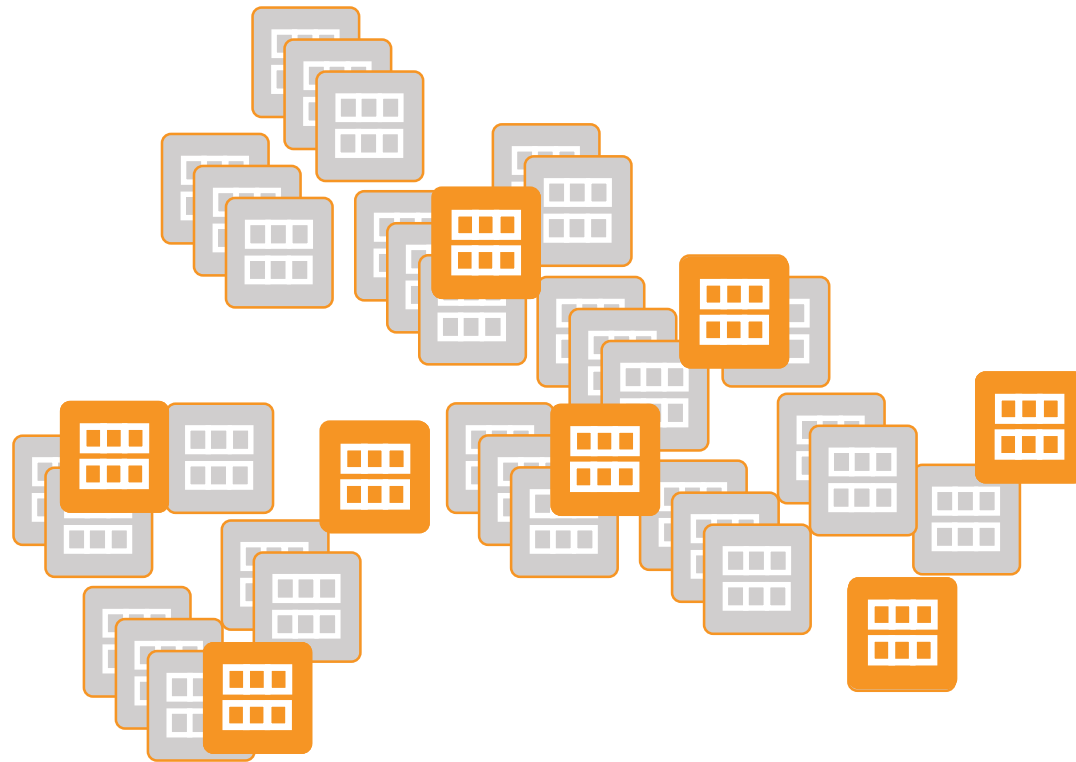


Why automate AMI updates?



- Security
- Consistency
- Cost

Stage 1 – Identify AMLs (to update)



Stage 1 Technical Detail: Identify AMIs



Tagging with “AutoUpdate = yes”

EC2 Dashboard

- Events
- Tags
- Reports
- Limits
- INSTANCES
 - Instances
 - Spot Requests
 - Reserved Instances
 - Scheduled Instances
 - Commands
 - Dedicated Hosts
- IMAGES
 - AMIs**
 - Bundle Tasks
- ELASTIC BLOCK STORE
- NETWORK & SECURITY

Launch Actions

Owned by me search : silence Add filter

Name	AMI Name	AMI ID	Source
silence-of-the-lambda	silence-of-the-lambda	ami-dbc67ca8	234277

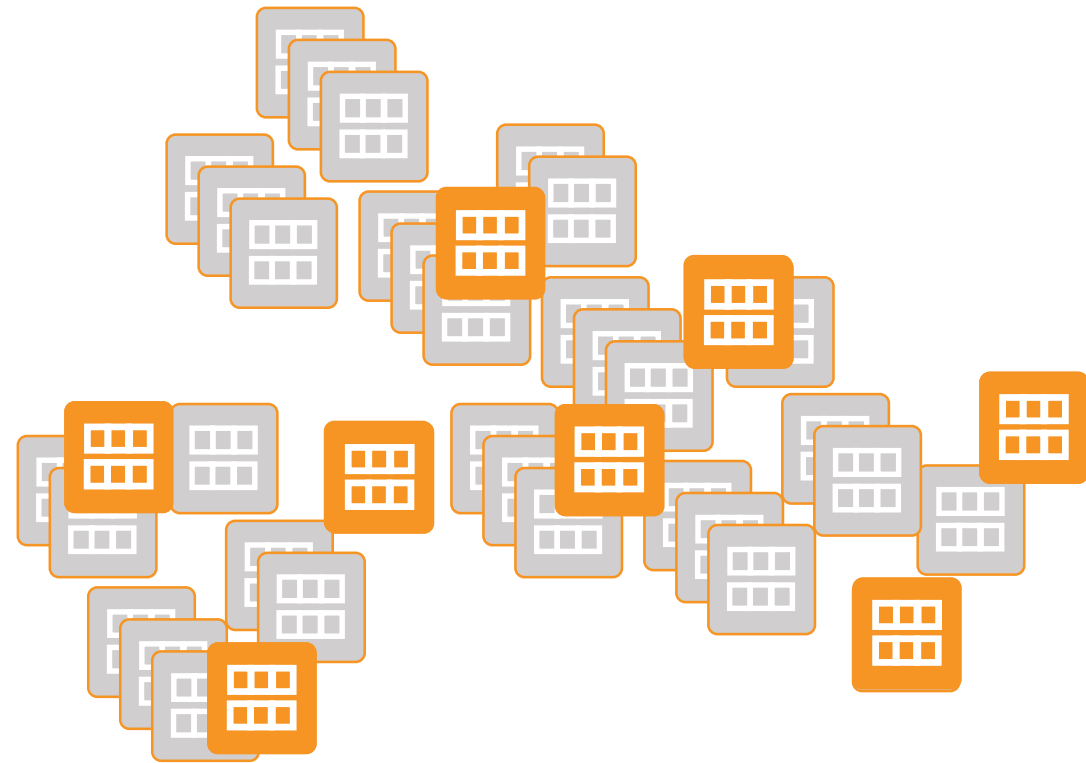
Image: ami-dbc67ca8

Details Permissions **Tags**

Add/Edit Tags

Key	Value
Name	silence-of-the-lambda
AutoUpdate	yes

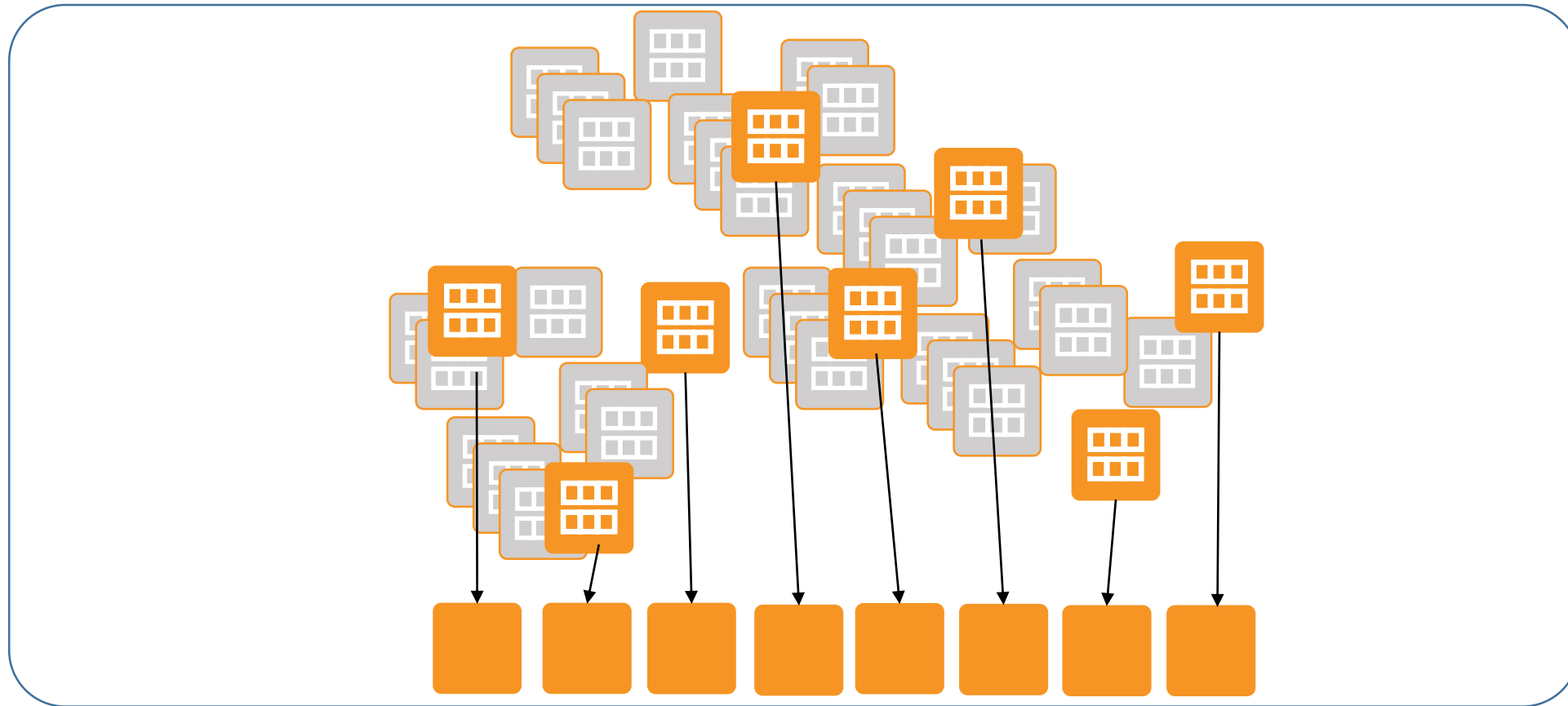
Stage 1 – Identify AMIs (✓ Completed)



Stage 2 – Update AMIs



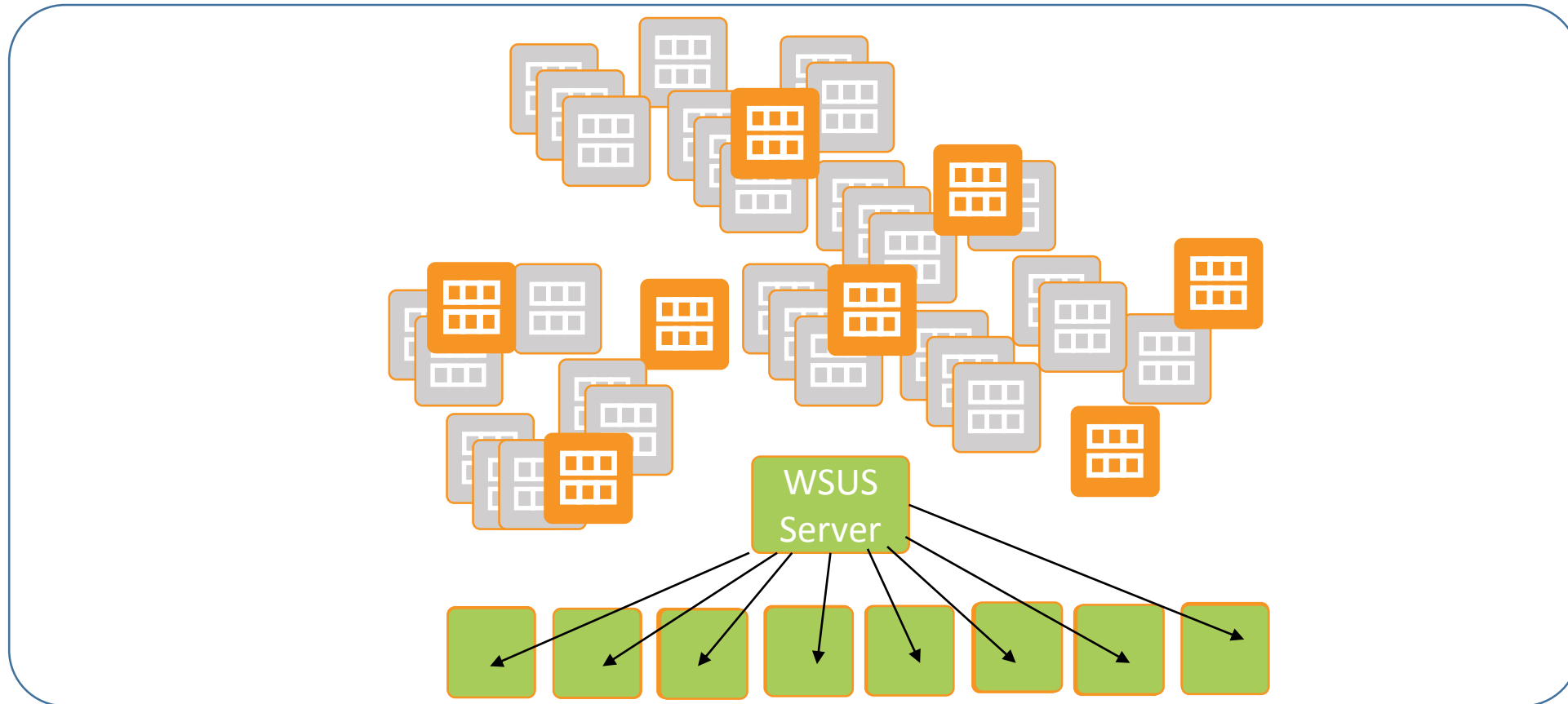
Step 1: Launch EC2 Instances



Stage 2 – Update AMIs



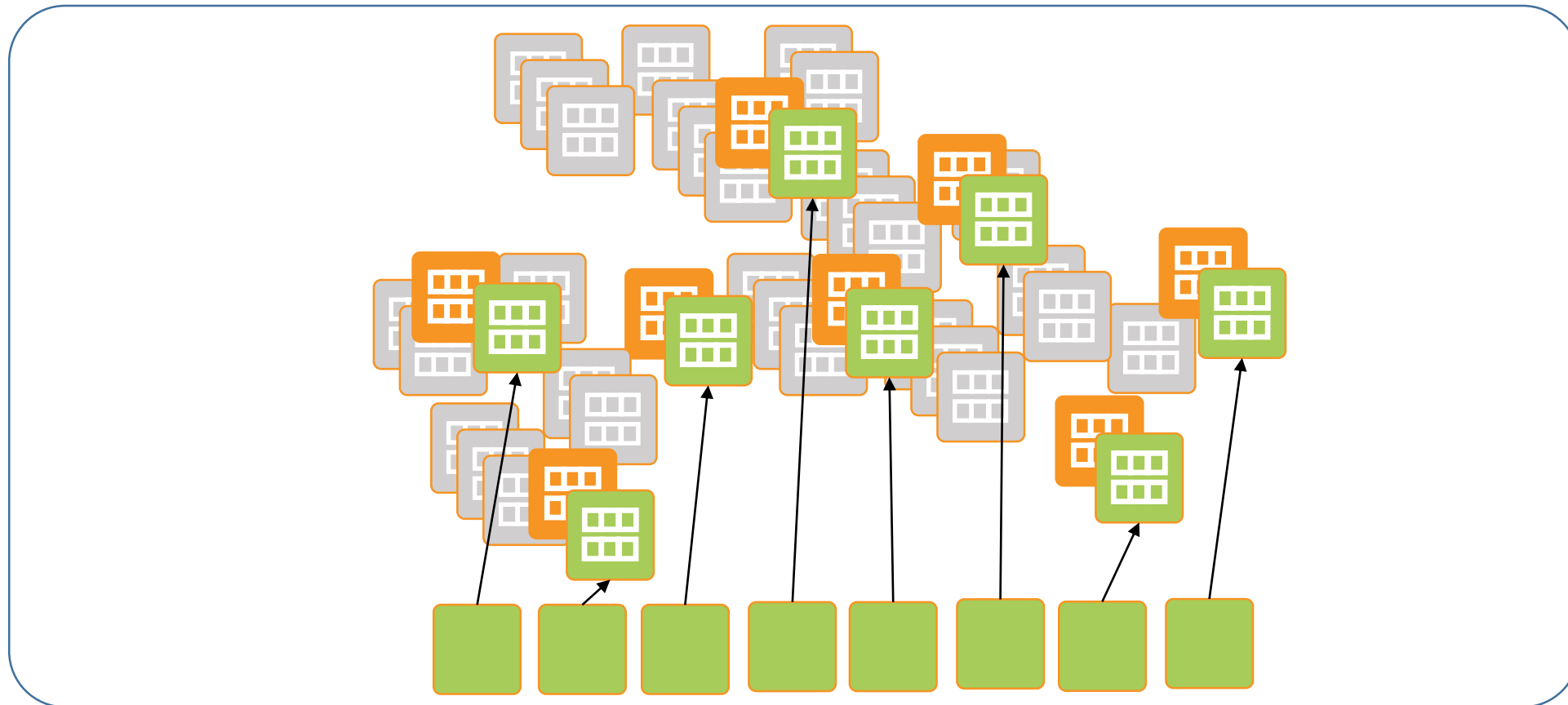
Step 2: Run Update on EC2 Instances



Stage 2 – Update AMIs



Step 3: Create Updated AMIs



Stage 2 Technical Detail: Update AMI



Step 1 – Launch EC2 Instance

```
client = boto3.client('ec2')
resource = boto3.resource('ec2')

def launch_existing_ami(ami_id, Groups, SubnetId, IamInstanceProfile, InstanceType):

def update_when_in_state(instance_id, ami_id):

def run_update_on_instance(instance_id, ami_id):
    ssm_client = boto3.client('ssm')
    params = {"commands":["wuauctl.exe /detectnow /updatenow"],"workingDirectory":["C:\Windows\System32"],"executionTimeout":["3600"]}
    response = ssm_client.send_command(InstanceIds=[instance_id],
                                       DocumentName='AWS-RunPowerShellScript',
                                       TimeoutSeconds=10800,
                                       Comment='Lambda AutoUpdate Function',
                                       Parameters=params,
                                       OutputS3BucketName='fc-automation',
                                       )
    print "Completed update on instance"
    print response

def lambda_handler(event, context):
    """
    Entry point to read in configuration details.
    Get those AMI's that have been tagged with AutoUpdate: yes
    """
    filters = [{"Name":'tag:AutoUpdate', 'Values':['yes']}]}
    response = client.describe_images(Filters=filters)
    if len(response.get("Images")) > 0:
        images = response.get("Images")
        for item in images:
            launch_existing_ami(item["ImageId"], item["Name"], event["Groups"],
                               event["SubnetId"], event["IamInstanceProfile"], event["InstanceType"])
    else:
        print "Failed to get list of images"
        raise Exception("Failed to get list of images")
```

resource.create_instances(ImageId=ami_id, ...)

Stage 2 Technical Detail: Update AMI



Step 2 – Update EC2 Instance

```
client = boto3.client('ec2')
resource = boto3.resource('ec2')

def launch_existing_ami(ami_id, Groups, SubnetId, IamInstanceProfile, InstanceType):

def update_when_in_state(instance_id, ami_id):

def run_update_on_instance(instance_id, ami_id):
    ssm_client = boto3.client('ssm')
    params = {
        "commands":["wuauclt.exe /u /q /w /s /o /n /c /d /e /f /g /h /i /j /k /l /m /n /o /p /q /r /s /t /u /v /w /x /y /z /AA /AB /AC /AD /AE /AF /AG /AH /AI /AJ /AK /AL /AM /AN /AO /AP /AQ /AR /AS /AT /AU /AV /AW /AX /AY /AZ /BA /BB /BC /BD /BE /BF /BG /BH /BI /BJ /BK /BL /BM /BN /BO /BP /BQ /BR /BS /BT /BU /BV /BW /BX /BY /BZ /CA /CB /CC /CD /CE /CF /CG /CH /CI /CJ /CK /CL /CM /CN /CO /CP /CQ /CR /CS /CT /CU /CV /CW /CX /CY /CZ /DA /DB /DC /DD /DE /DF /DG /DH /DI /DJ /DK /DL /DM /DN /DO /DP /DQ /DR /DS /DT /DU /DV /DW /DX /DY /DZ /EA /EB /EC /ED /EE /EF /EG /EH /EI /EJ /EK /EL /EM /EN /EO /EP /EQ /ER /ES /ET /EU /EV /EW /EX /EY /EZ /FA /FB /FC /FD /FE /FF /FG /FH /FI /FJ /FK /FL /FM /FN /FO /FP /FQ /FR /FS /FT /FU /FV /FW /FX /FY /FZ /GA /GB /GC /GD /GE /GF /GG /GH /GI /GJ /GK /GL /GM /GN /GO /GP /GQ /GR /GS /GT /GU /GV /GW /GX /GY /GZ /HA /HB /HC /HD /HE /HF /HG /HH /HI /HJ /HK /HL /HM /HN /HO /HP /HQ /HR /HS /HT /HU /HV /HW /HX /HY /HZ /IA /IB /IC /ID /IE /IF /IG /IH /II /IJ /IK /IL /IM /IN /IO /IP /IQ /IR /IS /IT /IU /IV /IW /IX /IY /IZ /JA /JB /JC /JD /JE /JF /JG /JH /JI /JJ /JK /JL /JM /JN /JO /JP /JQ /JR /JS /JT /JU /JV /JW /JX /JY /JZ /KA /KB /KC /KD /KE /KF /KG /KH /KI /KJ /KK /KL /KM /KN /KO /KP /KQ /KR /KS /KT /KU /KV /KW /KX /KY /KZ /LA /LB /LC /LD /LE /LF /LG /LH /LI /LJ /LK /LL /LM /LN /LO /LP /LQ /LR /LS /LT /LU /LV /LW /LX /LY /LZ /MA /MB /MC /MD /ME /MF /MG /MH /MI /MJ /MK /ML /MM /MN /MO /MP /MQ /MR /MS /MT /MU /MV /MW /MX /MY /MZ /NA /NB /NC /ND /NE /NF /NG /NH /NI /NJ /NK /NL /NM /NN /NO /NP /NQ /NR /NS /NT /NU /NV /NW /NX /NY /NZ /OA /OB /OC /OD /OE /OF /OG /OH /OI /OJ /OK /OL /OM /ON /OO /OP /OQ /OR /OS /OT /OU /OV /OW /OX /OY /OZ /PA /PB /PC /PD /PE /PF /PG /PH /PI /PJ /PK /PL /PM /PN /PO /PP /PQ /PR /PS /PT /PU /PV /PW /PX /PY /PZ /QA /QB /QC /QD /QE /QF /QG /QH /QI /QJ /QK /QL /QM /QN /QO /QP /QQ /QR /QS /QT /QU /QV /QW /QX /QY /QZ /RA /RB /RC /RD /RE /RF /RG /RH /RI /RJ /RK /RL /RM /RN /RO /RP /RQ /RR /RS /RT /RU /RV /RW /RX /RY /RZ /SA /SB /SC /SD /SE /SF /SG /SH /SI /SJ /SK /SL /SM /SN /SO /SP /SQ /SR /SS /ST /SU /SV /SW /SX /SY /SZ /TA /TB /TC /TD /TE /TF /TG /TH /TI /TJ /TK /TL /TM /TN /TO /TP /TQ /TR /TS /TT /TU /TV /TW /TX /TY /TZ /UA /UB /UC /UD /UE /UF /UG /UH /UI /UJ /UK /UL /UM /UN /UO /UP /UQ /UR /US /UT /UU /UV /UW /UX /UY /UZ /VA /VB /VC /VD /VE /VF /VG /VH /VI /VJ /VK /VL /VM /VN /VO /VP /VQ /VR /VS /VT /VU /VV /VW /VX /VY /VZ /WA /WB /WC /WD /WE /WF /WG /WH /WI /WJ /WK /WL /WM /WN /WO /WP /WQ /WR /WS /WT /WU /WV /WW /WX /WY /WZ /XA /XB /XC /XD /XE /XF /XG /XH /XI /XJ /XK /XL /XM /XN /XO /XP /XQ /XR /XS /XT /XU /XV /XW /XX /XY /XZ /YA /YB /YC /YD /YE /YF /YG /YH /YI /YJ /YK /YL /YM /YN /YO /YP /YQ /YR /YS /YT /YU /YV /YW /YX /YY /YZ /ZA /ZB /ZC /ZD /ZE /ZF /ZG /ZH /ZI /ZJ /ZK /ZL /ZM /ZN /ZO /ZP /ZQ /ZR /ZS /ZT /ZU /ZV /ZW /ZX /ZY /ZZ
```

params = {
 "commands":["wuauclt.exe /u /q /w /s /o /n /c /d /e /f /g /h /i /j /k /l /m /n /o /p /q /r /s /t /u /v /w /x /y /z /AA /AB /AC /AD /AE /AF /AG /AH /AI /AJ /AK /AL /AM /AN /AO /AP /AQ /AR /AS /AT /AU /AV /AW /AX /AY /AZ /BA /BB /BC /BD /BE /BF /BG /BH /BI /BJ /BK /BL /BM /BN /BO /BP /BQ /BR /BS /BT /BU /BV /BW /BX /BY /BZ /CA /CB /CC /CD /CE /CF /CG /CH /CI /CJ /CK /CL /CM /CN /CO /CP /CQ /CR /CS /CT /CU /CV /CW /CX /CY /CZ /DA /DB /DC /DD /DE /DF /DG /DH /DI /DJ /DK /DL /DM /DN /DO /DP /DQ /DR /DS /DT /DU /DV /DW /DX /DY /DZ /EA /EB /EC /ED /EE /EF /EG /EH /EI /EJ /EK /EL /EM /EN /EO /EP /EQ /ER /ES /ET /EU /EV /EW /EX /EY /EZ /FA /FB /FC /FD /FE /FF /FG /FH /FI /FJ /FK /FL /FM /FN /FO /FP /FQ /FR /FS /FT /FU /FV /FW /FX /FY /FZ /GA /GB /GC /GD /GE /GF /GG /GH /GI /GJ /GK /GL /GM /GN /GO /GP /GQ /GR /GS /GT /GU /GV /GW /GX /GY /GZ /HA /HB /HC /HD /HE /HF /HG /HH /HI /HJ /HK /HL /HM /HN /HO /HP /HQ /HR /HS /HT /HU /HV /HW /HX /HY /HZ /IA /IB /IC /ID /IE /IF /IG /IH /II /IJ /IK /IL /IM /IN /IO /IP /IQ /IR /IS /IT /IU /IV /IW /IX /IY /IZ /JA /JB /JC /JD /JE /JF /JG /JH /JI /JJ /JK /JL /JM /JN /JO /JP /JQ /JR /JS /JT /JU /JV /JW /JX /JY /JZ /KA /KB /KC /KD /KE /KF /KG /KH /KI /KJ /KK /KL /KM /KN /KO /KP /KQ /KR /KS /KT /KU /KV /KW /KX /KY /KZ /LA /LB /LC /LD /LE /LF /LG /LH /LI /LJ /LK /LL /LM /LN /LO /LP /LQ /LR /LS /LT /LU /LV /LW /LX /LY /LZ /MA /MB /MC /MD /ME /MF /MG /MH /MI /MJ /MK /ML /MM /MN /MO /MP /MQ /MR /MS /MT /MU /MV /MW /MX /MY /MZ /NA /NB /NC /ND /NE /NF /NG /NH /NI /NJ /NK /NL /NM /NN /NO /NP /NQ /NR /NS /NT /NU /NV /NW /NX /NY /NZ /OA /OB /OC /OD /OE /OF /OG /OH /OI /OJ /OK /OL /OM /ON /OO /OP /OQ /OR /OS /OT /OU /OV /OW /OX /OY /OZ /PA /PB /PC /PD /PE /PF /PG /PH /PI /PJ /PK /PL /PM /PN /PO /PP /PQ /PR /PS /PT /PU /PV /PW /PX /PY /PZ /QA /QB /QC /QD /QE /QF /QG /QH /QI /QJ /QK /QL /QM /QN /QO /QP /QQ /QR /QS /QT /QU /QV /QW /QX /QY /QZ /RA /RB /RC /RD /RE /RF /RG /RH /RI /RJ /RK /RL /RM /RN /RO /RP /RQ /RR /RS /RT /RU /RV /RW /RX /RY /RZ /SA /SB /SC /SD /SE /SF /SG /SH /SI /SJ /SK /SL /SM /SN /SO /SP /SQ /SR /SS /ST /SU /SV /SW /SX /SY /SZ /TA /TB /TC /TD /TE /TF /TG /TH /TI /TJ /TK /TL /TM /TN /TO /TP /TQ /TR /TS /TT /TU /TV /TW /TX /TY /TZ /UA /UB /UC /UD /UE /UF /UG /UH /UI /UJ /UK /UL /UM /UN /UO /UP /UQ /UR /US /UT /UU /UV /UW /UX /UY /UZ /VA /VB /VC /VD /VE /VF /VG /VH /VI /VJ /VK /VL /VM /VN /VO /VP /VQ /VR /VS /VT /VU /VV /VW /VX /VY /VZ /WA /WB /WC /WD /WE /WF /WG /WH /WI /WJ /WK /WL /WM /WN /WO /WP /WQ /WR /WS /WT /WU /WV /WW /WX /WY /WZ /XA /XB /XC /XD /XE /XF /XG /XH /XI /XJ /XK /XL /XM /XN /XO /XP /XQ /XR /XS /XT /XU /XV /XW /XX /XY /XZ /YA /YB /YC /YD /YE /YF /YG /YH /YI /YJ /YK /YL /YM /YN /YO /YP /YQ /YR /YS /YT /YU /YV /YW /YX /YY /YZ /ZA /ZB /ZC /ZD /ZE /ZF /ZG /ZH /ZI /ZJ /ZK /ZL /ZM /ZN /ZO /ZP /ZQ /ZR /ZS /ZT /ZU /ZV /ZW /ZX /ZY /ZZ

ssm_client.send_command (
 instance_id,
 params,
 ...
)

Stage 2 Technical Detail: Update AMI



Step 3 – Create Updated AMI

```
client = boto3.client('ec2')
resource = boto3.resource('ec2')

def launch_existing_ami(ami_id, Groups, SubnetId, IamInstanceProfile, InstanceType):

def update_when_in_state(instance_id, ami_id):

def run_update_on_instance(instance_id, ami_id):
    ssm_client = boto3.client('ssm')
    params = {"command": "powershell C:\Windows\System32\cmd.exe /c powershell {0} >>> C:\Windows\System32\"
    response = ssm_client.send_command(InstanceIds=[instance_id],
                                       Parameters=[{"Name": "AWS-RunPowerShellScript",
                                                     "Value": ["powershell {0} >>> C:\Windows\System32\"
                                                             .format(ami_id, ami_id)]}],
                                       Targets=[{"KeyPrefix": "windowsupdates-{0}-{1}-".format(instance_id, ami_id)}],
                                       Comment="Lambda AutoUpdate Function",
                                       MaxAttempts=3,
                                       MaxConcurrency=1,
                                       OutputsKeyPrefix="windowsupdates-{0}-{1}-".format(instance_id, ami_id))
    print "Command Status: ", response["Command"]["Status"]
    print response

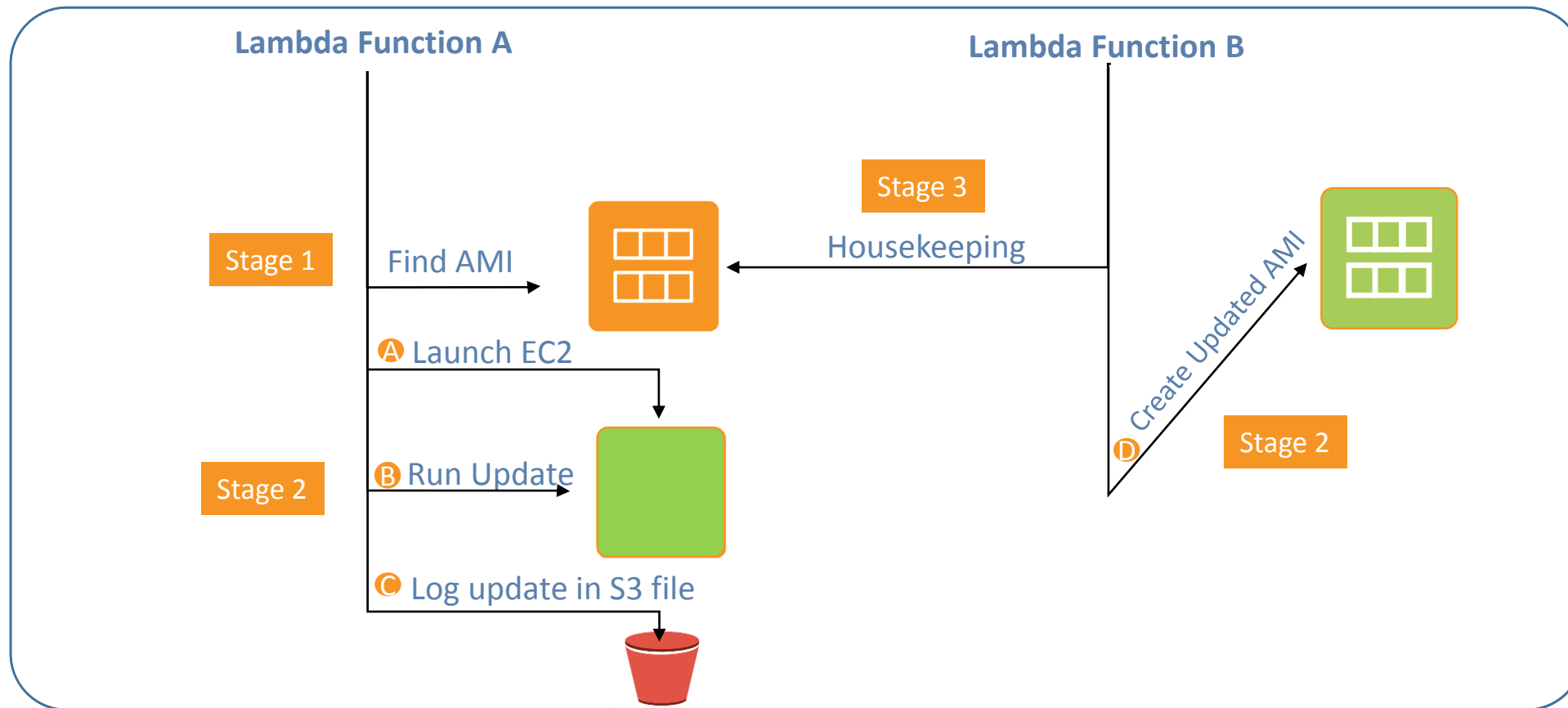
def lambda_handler(event, context):
    """
    Entry point to read in configuration details.
    Get those AMIs that have been tagged with AutoUpdate: yes
    """
    filters = [{"Name": "tag:AutoUpdate", "Values": ["yes"]}]]
    response = client.describe_images(Filters=filters)
    if len(response.get("Images")) > 0:
        images = response.get("Images")
        for item in images:
            launch_existing_ami(item["ImageId"], item["Name"], event["Groups"],
                               event["SubnetId"], event["IamInstanceProfile"], event["InstanceType"])
    else:
        print "Failed to get list of images"
        raise Exception("Failed to get list of images")
```

ami = client.create_image(
DryRun=False,
InstanceId=instance_id,
Description='Automated AMI creator',
Name="{0}-
{1}" .format(name,int(time.time())))

Stage 2 Technical Detail: Update AMI



Appendix : 2 Lambda Functions linked via S3 Bucket



Stage 2 Technical Detail: Update AMI



Appendix : S3 Bucket Configuration

Bucket: fc-automation
Region: Ireland
Creation Date: Tue Mar 15 11:01:05 GMT+000 2016
Owner: jphayes11


▶ Permissions

▶ Static Website Hosting

▶ Logging

▼ Events

[Event Notifications](#) enable you to send alerts or trigger workflows. Notifications can be sent via [Amazon Simple Notification Service \(SNS\)](#) or [Amazon Simple Queue Service \(SQS\)](#) or to a [Lambda function](#) (depending on the bucket location).

Name	Event(s)	Filter	Type
AMIUpdate-HouseKeeping	Post,Put		Lambda  

Stage 2 Technical Detail: Update AMI



Appendix – S3 Bucket Call

```
client = boto3.client('ec2')
resource = boto3.resource('ec2')

def launch_existing_ami(ami_id, event["Groups"], SubnetId, IamInstanceProfile, InstanceType):

def update_when_in_state(instance_id, ami_id):

def run_updates_on_instance(instance_id, ami_id):
    ssm_client = boto3.client('ssm')
    params = {"commands":["wuauclt.exe /detectnow /updatenow"],"workingDirectory":["C:\Windows\System32"],"executionTimeout":["3600"]}
    response = ssm_client.send_command(InstanceIds=[instance_id],
                                       DocumentName='AWS-RunPowerShellScript',
                                       Parameters=params,
                                       OutputS3BucketName='fc-automation',
                                       OutputS3KeyPrefix="windowsupdates-{0}-{1}-".format(instance_id, ami_id))
    print "Command Status: ", response["Command"]["Status"]
    print response

def lambda_handler(event, context):
    Entry point for the lambda function
    Get those AMI's that have been tagged with AutoUpdate: yes
    """
    filters = [{'Name': 'tag:AutoUpdate', 'Values': ['yes']}]}
    response = client.describe_images(Filters=filters)
    if len(response.get("Images")) > 0:
        images = response.get("Images")
        for item in images:
            launch_existing_ami(item["ImageId"], item["Name"], event["Groups"],
                               event["SubnetId"], event["IamInstanceProfile"], event["InstanceType"])
    else:
        print "Failed to get list of images"
        raise Exception("Failed to get list of images")
```

OutputS3BucketName='fc-automation',

OutputS3KeyPrefix="windowsupdates-{0}-{1}-

".format(instance_id, ami_id)

Stage 2 Technical Detail: Update AMI



Appendix : S3 Bucket Updates

Upload Create Folder Actions

Search by prefix None Properties Transfers

All Buckets / fc-automation

Name	Storage Class	Size	Last Modified
The bucket 'fc-automation' is empty			

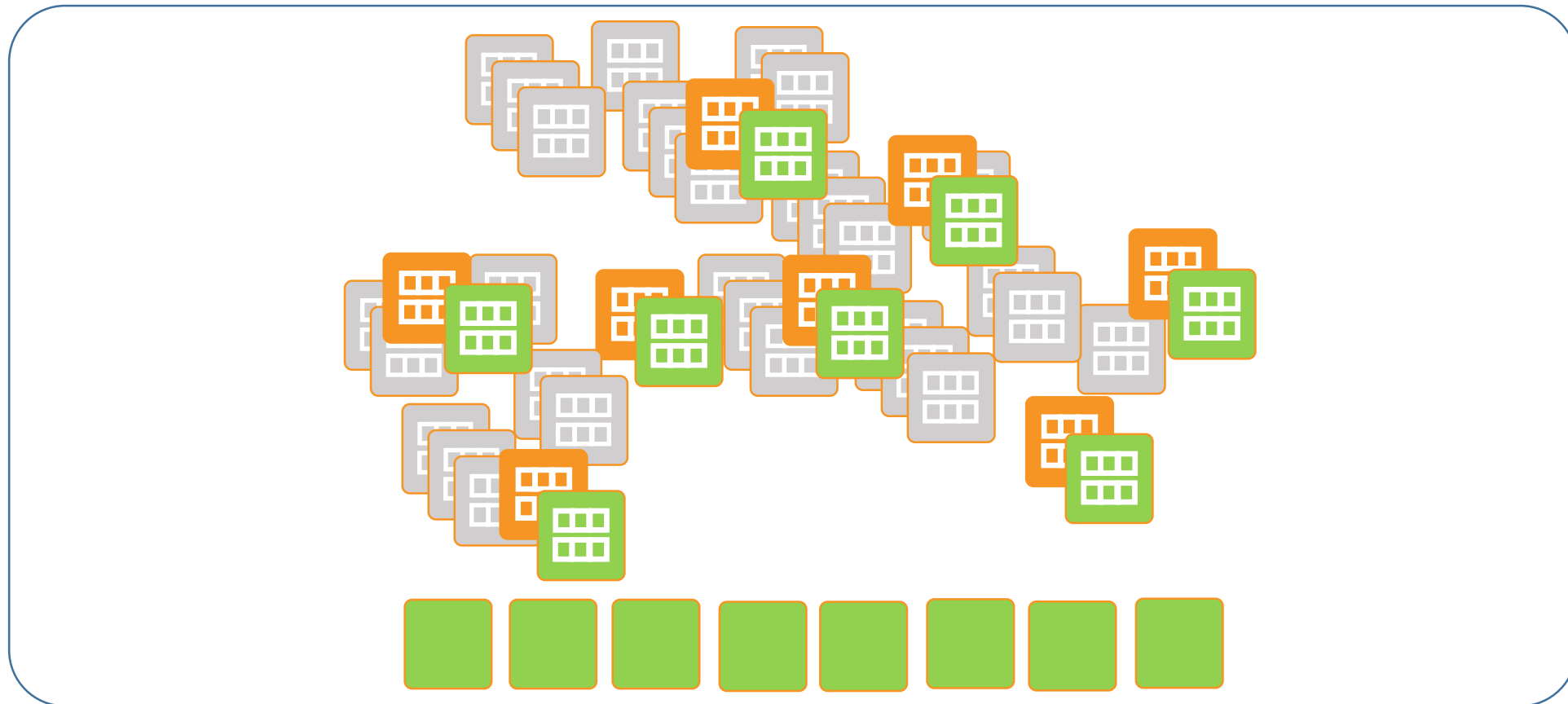
Upload Create Folder Actions

Search by prefix None Properties Transfers

All Buckets / fc-automation / windowsupdates-i-06916d8a-ami-f9bf058a-

ssmaccessstest.txt	Standard	24 bytes	Tue Mar 15 17:11:16 GMT+000 2016
--------------------	----------	----------	----------------------------------

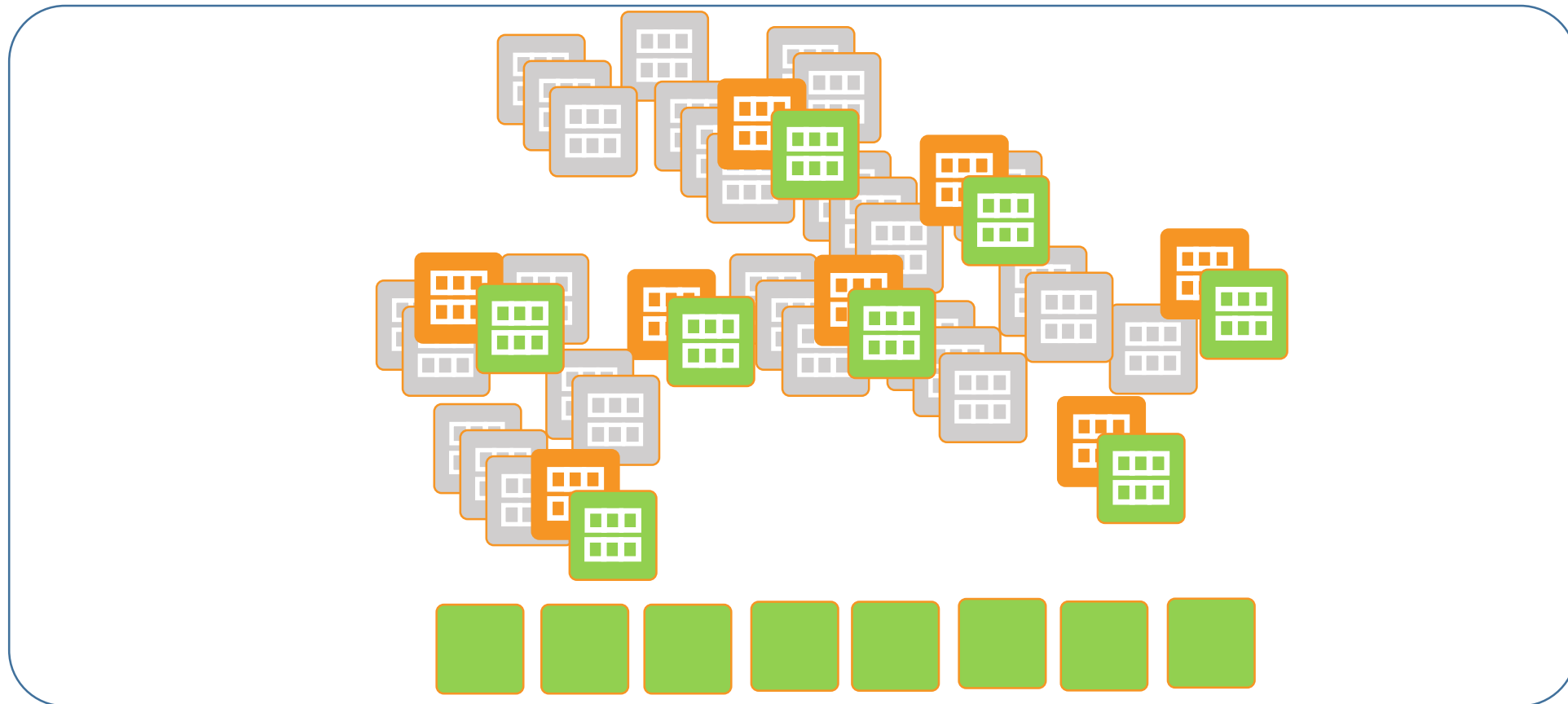
Stage 2 –Update AMIs (✓Complete)



Stage 3 – Housekeeping



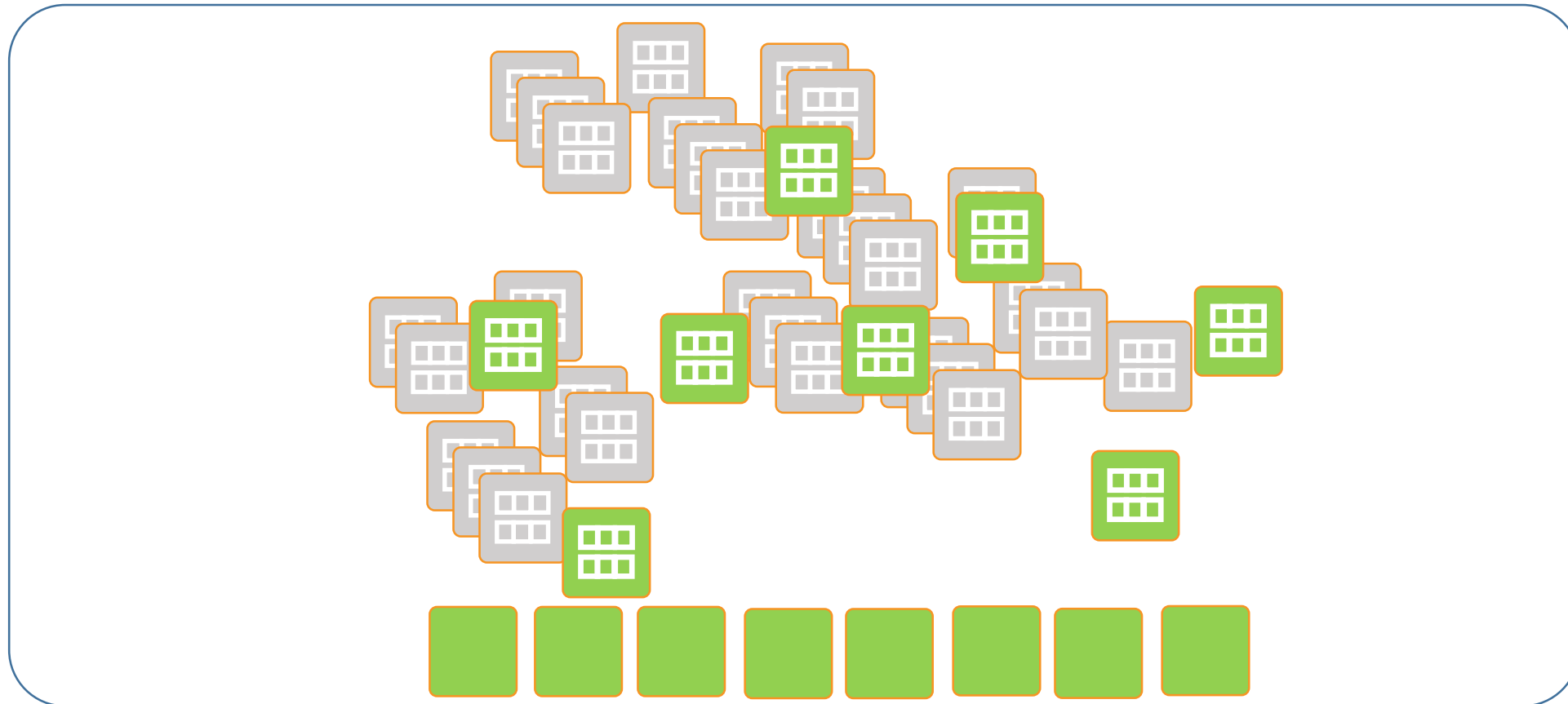
Step 1: De-register Earlier Version AMIs



Stage 3 – Housekeeping



Step 2: Terminate EC2 Instances



Stage 3 Technical Detail: Housekeeping



Deregister Earlier AMIs & Terminate EC2 Instances

```
client = boto3.client('ec2')
resource = boto3.resource('ec2')

def launch_existing_ami(ami_id, Groups, SubnetId, IamInstanceProfile, InstanceType):

def update_when_in_state(instance_id, ami_id):

def run_update_on_instance(instance_id, ami_id):
    ssm_client = boto3.client('ssm')
    params = {"command": "powershell -Command 'aws ssm put-command --name 'update-ami' --description 'Update AMI' --timeout: [\"3600\"] --document-name 'AWS-RunPowerShellScript' --document-version 'AWS-RunPowerShellScript-2019-09-17' --parameters 'Parameters=params, OutputS3BucketName='fc-automation', OutputS3KeyPrefix='windowsupdates-{0}-{1}-'.format(instance_id, ami_id)'}
    response = ssm_client.put_command(**params)
    print "Command Status: ", response["Command"]["Status"]

def lambda_handler(event, context):
    """
    Entry point to read in configuration details.
    Get those AMI's that have been tagged with AutoUpdate: yes
    """
    filters = [{'Name': 'tag:AutoUpdate', 'Values': ['yes']}]
    response = client.describe_images(Filters=filters)
    if len(response.get("Images")) > 0:
        images = response.get("Images")
        for item in images:
            launch_existing_ami(item["ImageId"], item["Name"], event["Groups"],
                               event["SubnetId"], event["IamInstanceProfile"], event["InstanceType"])
    else:
        print "Failed to get list of Images"
        raise Exception("Failed to get list of Images")
```

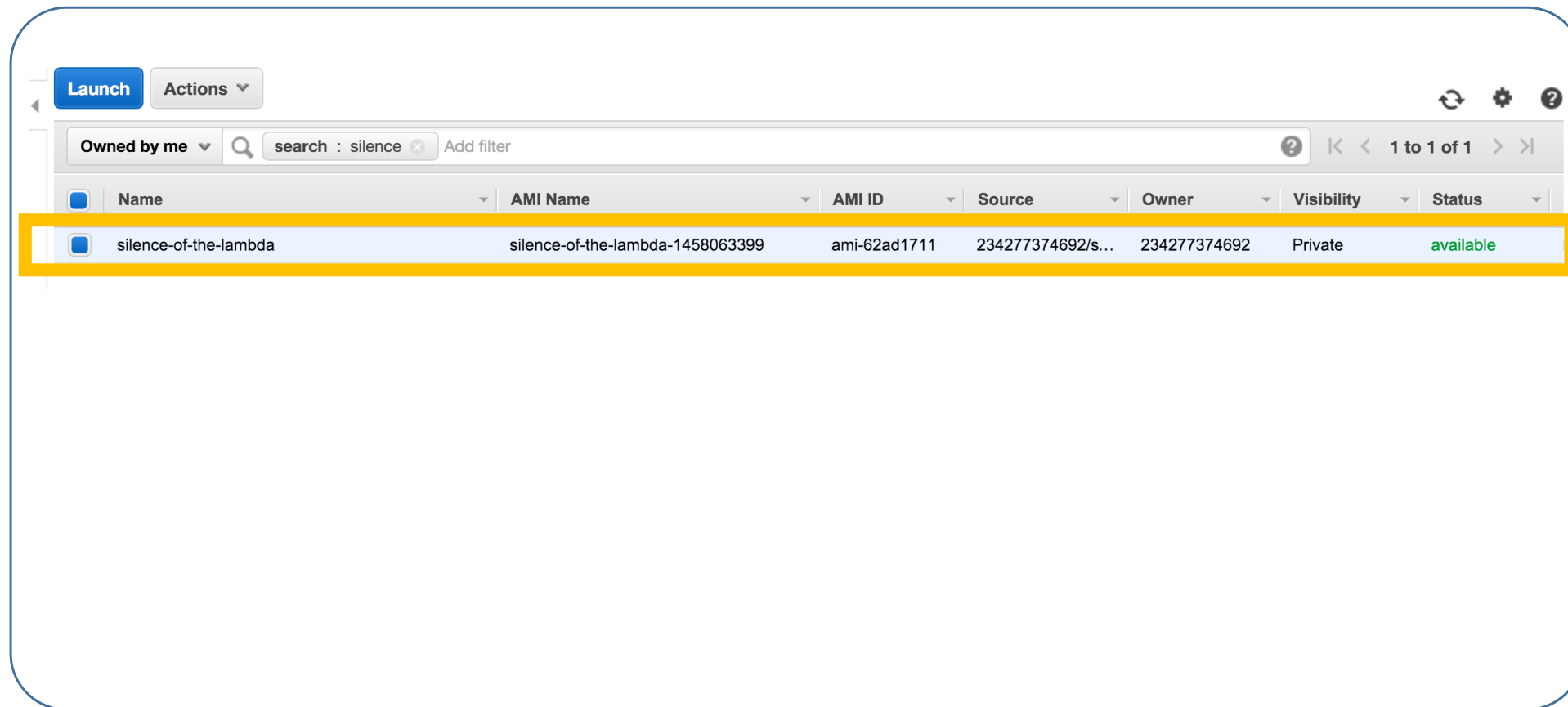
`client.deregister_image(ImageId=ami_id)`

`client.terminate_instances(InstanceIds=[instance_id])`

Stage 3 Technical Detail: Housekeeping

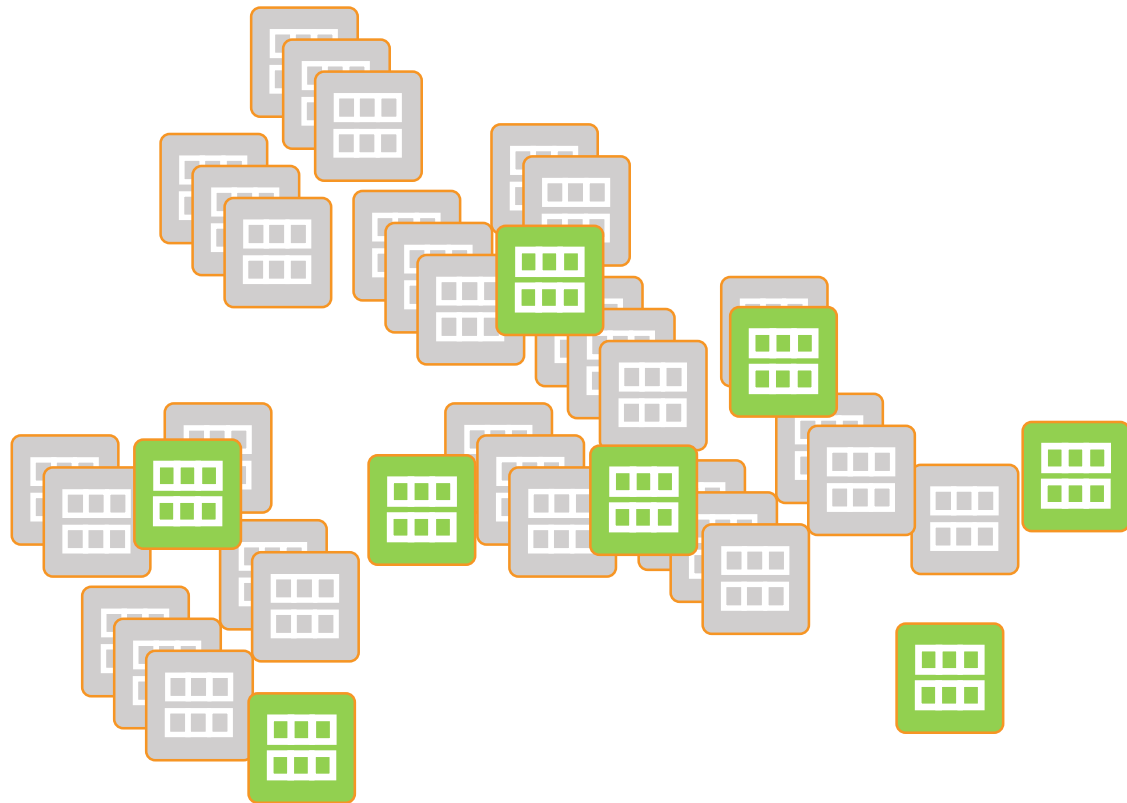


Outcome: **Updated AMIs**



<input type="checkbox"/>	Name	AMI Name	AMI ID	Source	Owner	Visibility	Status
<input type="checkbox"/>	silence-of-the-lambda	silence-of-the-lambda-1458063399	ami-62ad1711	234277374692/s...	234277374692	Private	available

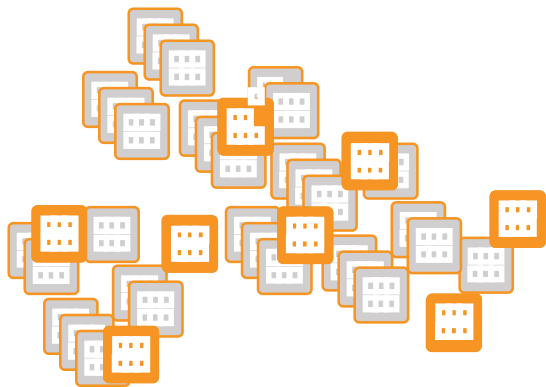
Stage 3 – Housekeeping (✓Complete)



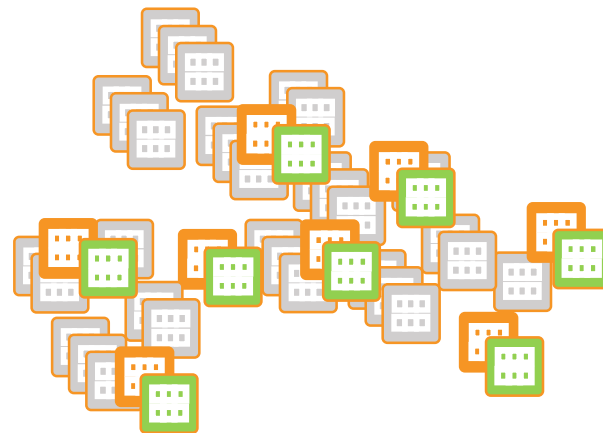
Automated AMI Update (✓ Complete)



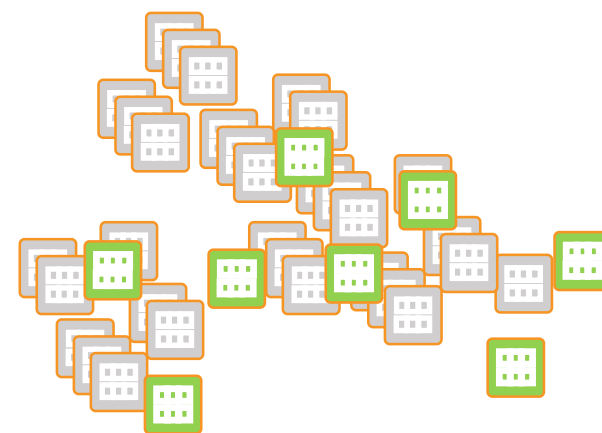
Stage 1
Identify AMIs ✓



Stage 2
Update AMIs ✓



Stage 3
Housekeeping ✓



Contact Details



www.terraalto.com



jobs@terraalto.com
