



MongoDB Management Pain Relief

Michael Lynn

Sr. Solutions Architect, MongoDB

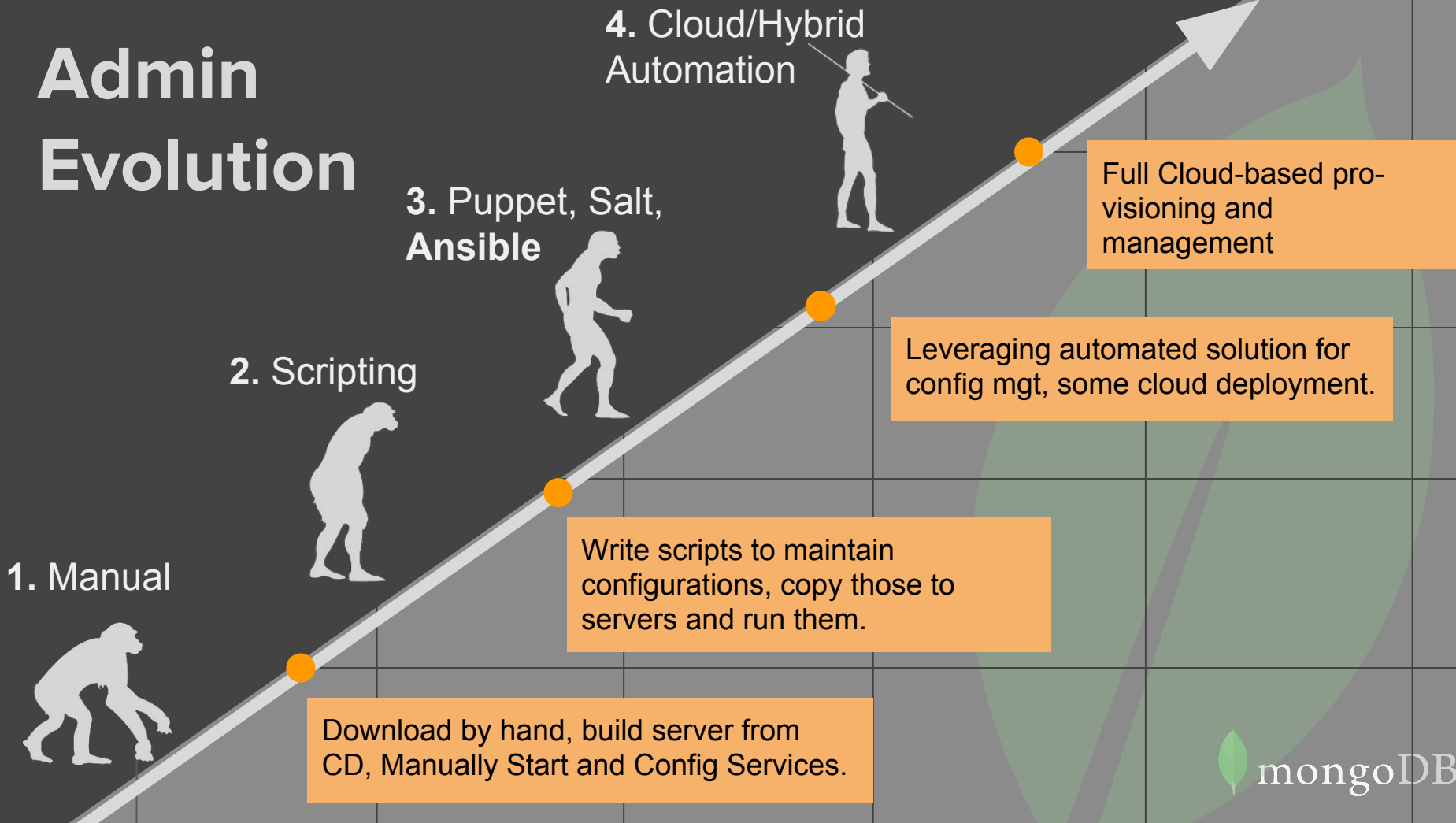
michael.lynn@mongodb.com

Goals

- **Ops Manager, Cloud Manager, Atlas**
 - What are they?
 - How do they differ?
 - Why do I need them?
- **Ansible**
 - What is it and how can I use these tools to **relieve some pain in my life?**

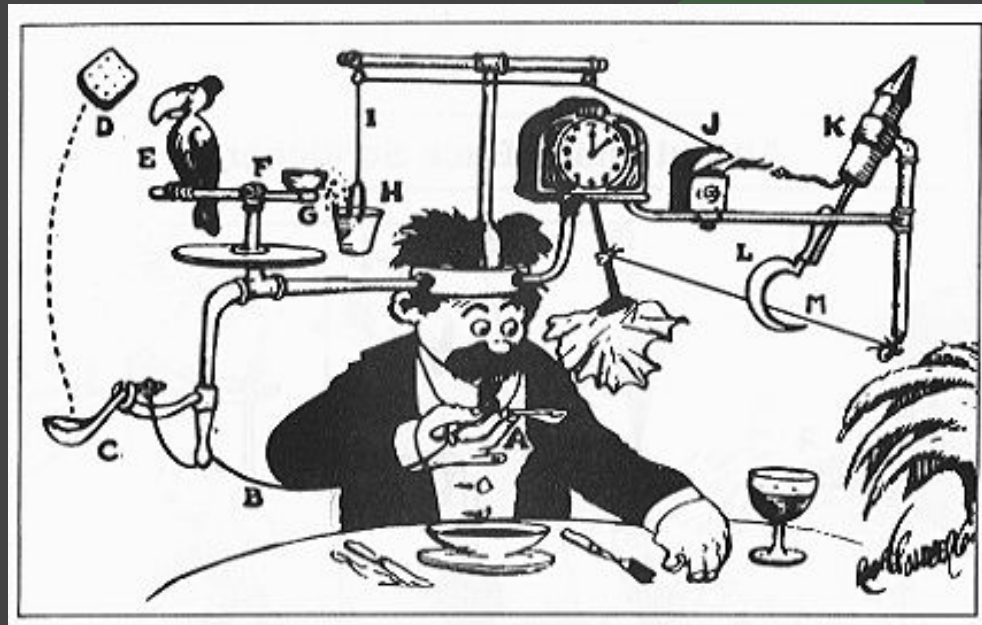
Show of Hands

Admin Evolution



Managing databases manually can be...

- Painful
- Risky
- Complex
- Susceptible to human error*
- Unnecessary...



* 80% of all downtime due to people/process error (Gartner)

100

Deploying

42

Upgrading

35

Maintaining

23

0

Play

What it Takes – 12-Server System

It can take a lot of manual effort to care for a MongoDB system in production



Deploy

- **Install + Configure**
 - 150+ steps

Upgrade

- **Upgrades, downgrades**
 - 100+ steps

Maintain

- **Scale out, move servers, resize oplog, etc**
 - 10 - 180+ steps

*Up to 95%
Reduction in
Operational
Overhead*

MongoDB Ops Manager

The Best Way to Manage MongoDB In Your Data Center

Up to 95% Reduction in Operational Overhead



Single-click provisioning, scaling & upgrades, admin tasks

Monitoring, with charts, dashboards and alerts on 100+ metrics

Backup and restore, with point-in-time recovery, support for sharded clusters

Demo of Ops Manager

mongoDB Ops Manager 2.0.6.363 GROUP dba

Group/RBAC

Deployment

Deployment Host Mappings Security Version Manager Authentication & Users Authorization & Roles Mongo Logs Global Admins Only Pings Deleted Hosts Raw AutomationConfig

PROCESSES SERVERS

Name	status	Version	Members	Actions
almrs		3.2.9	3 mongods	...
ip-172-31-21-19.ec2.internal:27018	11 secs ago	3.2.9		...
ip-172-31-21-20.ec2.internal:27018	11 secs ago	3.2.9		...
ip-172-31-21-14.ec2.internal:27018	11 secs ago	3.2.9		...
mlrs		3.2.9	3 mongods	...
newsr		3.2.9	3 mongods	...
nymug		3.2.9	3 mongods	...

Logical/Physical

MongoDB Version

Replica Set Members

Deployment

PROCESSES

SERVERS

Name	status	Version	Members
almrs		3.2.9	3 mongods
mlrs		3.2.9	3 mongods
newrs		3.2.9	3 mongods
nymug		3.2.9	3 mongods

Add Replica Set

1

- MongoDB Deployment
- New Cluster
- New Replica Set
- New Standalone
- Existing MongoDB Deployment

Last Login: 98.115.191.140 Version: 2.0.6.363

©2016 MongoDB, Inc.

ec2-52-201-246-152.compute-1.amazonaws.com:8080/v2/57db2b4ae4b0e512ab204cee#

mongoDB Ops Manager 2.0.6.363 GROUP dba

Deployment

Host Mappings

Security

Version Manager

Authentication & Users

Authorization & Roles

Mongo Logs

Global Admins Only

Pings

Deleted Hosts

Raw AutomationConfig

Deployment

PROCESSES

SERVICES

Name	status	Version	Members
▶ almrs		3.2.9	3 mongods
▶ mlrs		3.2.9	3 mongods
▶ newrs		3.2.9	3 mongods
▶ nymug		3.2.9	3 mongods

Last Login: 98.115.191.140 Version: 2.0.6.363

©2016 MongoDB, Inc.

New ReplicaSet

EDITING CANCEL APPLY

REPLICA SET CONFIGURATION

Name
e.g., myReplicaSet

Version
3.2.9-ent

Auth Schema Version
5 (3.0 Style)

Eligible Server RegExp
Regexp Matching Hostnames, e.g., ^hostPrefix

Eligible Port Range
e.g., 27000 e.g., 28000

MongoDs Per Replica Set Limit 12
- 3 +

MEMBER OPTIONS
Votes Priority Delay

Repset Name 1

Version of MongoDB 2

Server Pool Hostnames 3

Port to Run MongoDB 4

- Deployment
- Host Mappings
- Security
- Version Manager
- Authentication & Users
- Authorization & Roles
- Mongo Logs
- Global Admins Only
- Pings
- Deleted Hosts
- Raw AutomationConfig

Deployment

PROCESSES



SERVICES



Name	status	Version	Members
almrs		3.2.9	3 mongods
mlrs		3.2.9	3 mongods
newrs			
nymug		3.2.9	3 mongods

Last Login: 98.115.191.140 Version: 2.0.6.363

©2016 MongoDB, Inc.

CLICK APPLY 4

Member Roles 1

DB Path on Server 2

Advanced Options Settings 3

New ReplicaSet

EDITING

APPLY

Default 1 1 0

Automatic

Default 1 1 0

ADD

DB Path Prefix

e.g., /data

INDEX CONFIGURATION

ADVANCED OPTIONS

Startup Option	Value
bind_ip	0.0.0.0
engine	<input checked="" type="checkbox"/> mmapv1 <input type="checkbox"/> wiredTiger

MongoDB Enterprise Advanced

24 x 7 Support

MongoDB Ops Manager



Monitoring &
Alerting

Query
Optimization

Automation &
Configuration

Backup &
Recovery

REST API

MongoDB Compass



Schema
Visualization

Data Exploration

Ad-Hoc Queries

MongoDB Connector for BI



Visualization

Analysis

Reporting

Emergency
Patches

Customer
Success
Program

On-Demand
Online Training

Platform
Certifications

Encryption at Rest

LDAP & Kerberos

Auditing

FIPS 140-2

MongoDB Enterprise Server

Commercial License



Ops Manager

Activity



Deploy



Upgrade



Configure



Administer Database



Maintain OS - Mongod



Maintain OS - OpsMgr



Cloud Manager

Activity



Deploy



Upgrade



Configure



Administer Database



Maintain OS - Mongod



Maintain OS - OpsMgr



Atlas

Activity



Deploy



Upgrade (Done For You)



Configure



Administer Database



Maintain OS - Mongod



Maintain OS - OpsMgr

CONTROL

CONVENIENCE



Ops Manager

Where are my servers?

Your Data Center
AWS
Azure
GCP

MongoDB Versions?

Community, Enterprise



Cloud Manager

Where are my servers?

Your Data Center
AWS
Azure
GCP

MongoDB Versions?

Community, Enterprise



Atlas

Where are my servers?

AWS

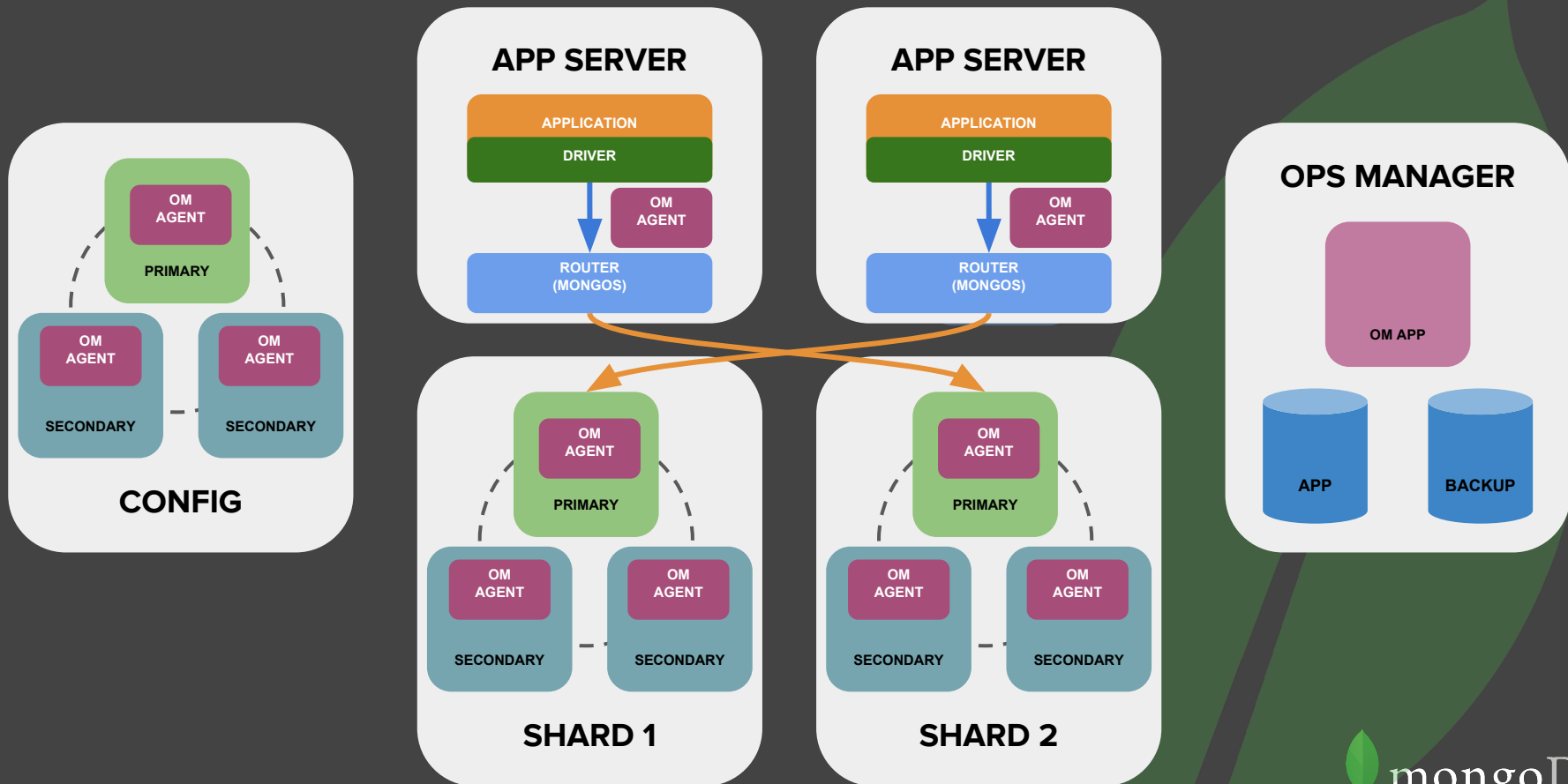
MongoDB Versions?

Community
Only Most Recent

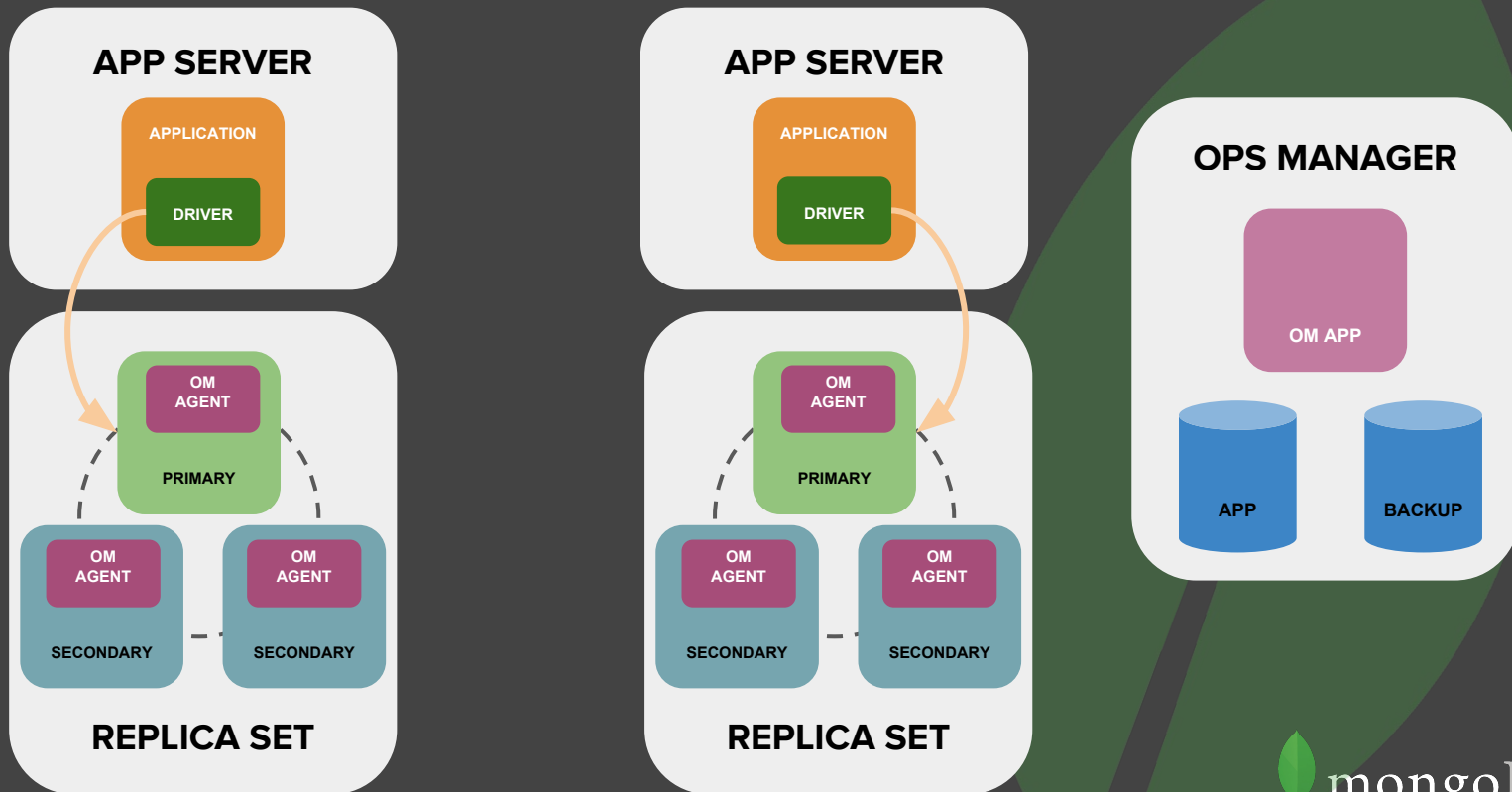


mongoDB

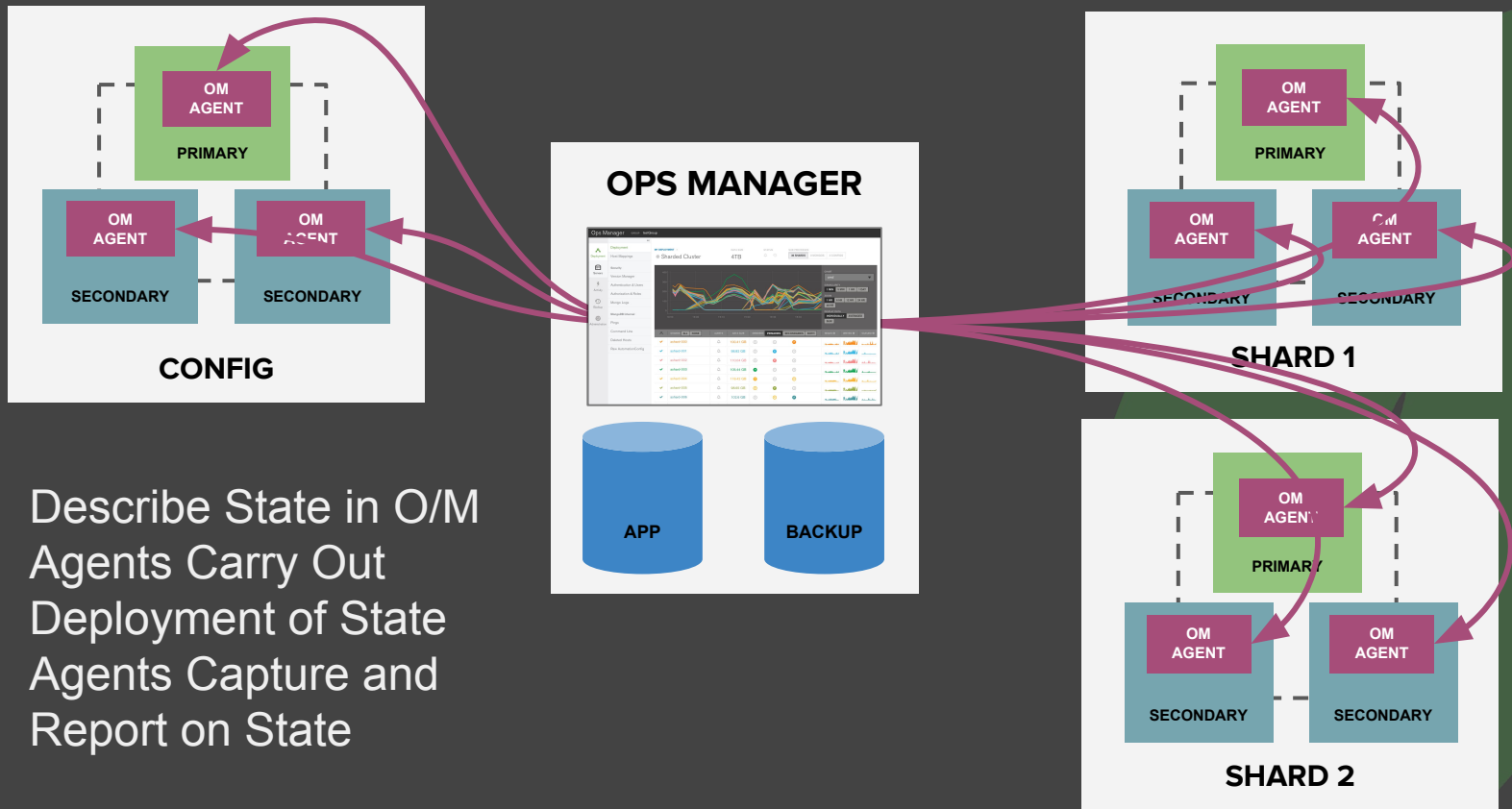
How Ops Manager Works



Replica Set Architecture (w/ Ops Manager)

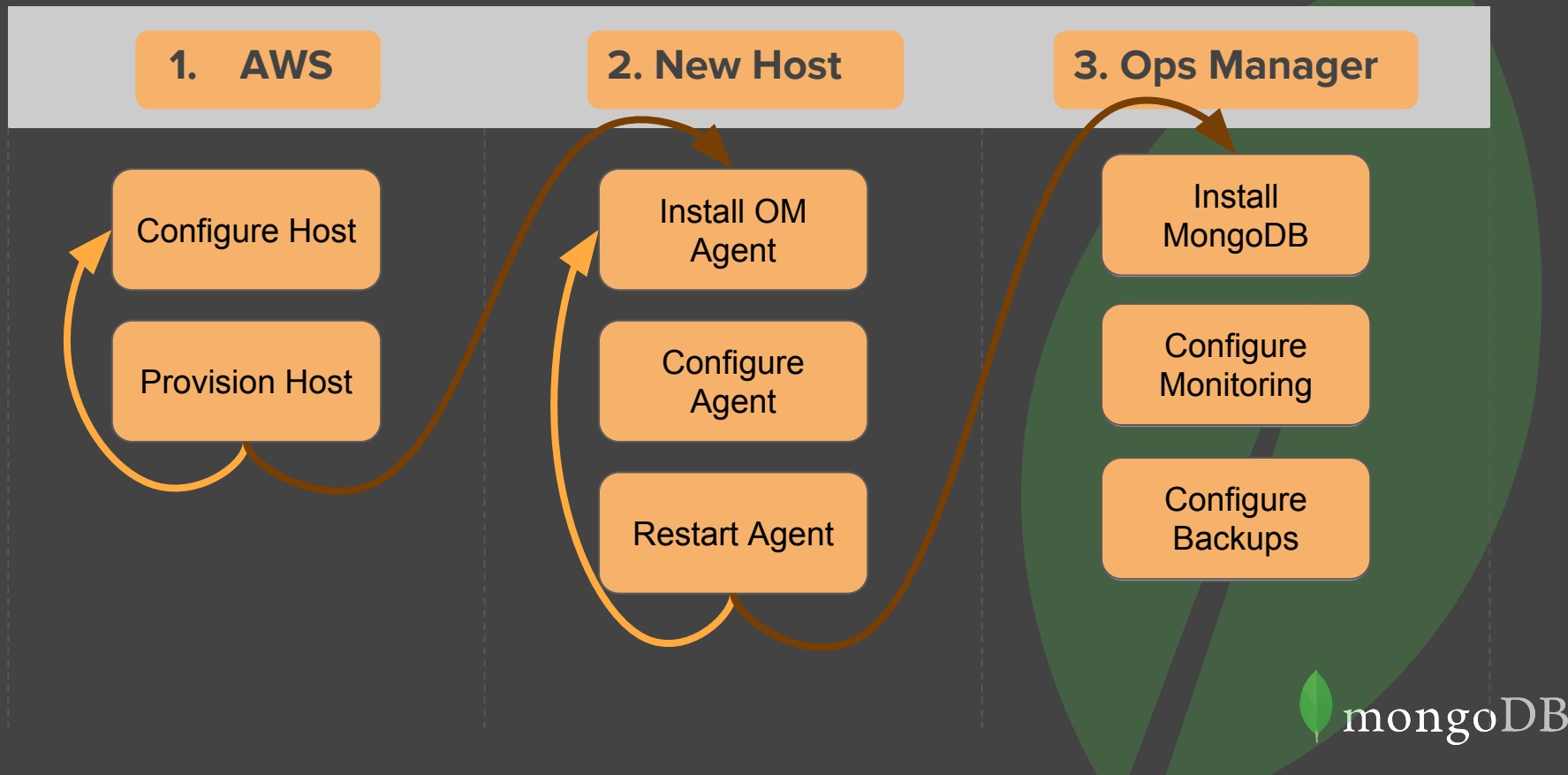


Sharded Architecture w/ Ops Manager



- Describe State in O/M
- Agents Carry Out Deployment of State
- Agents Capture and Report on State

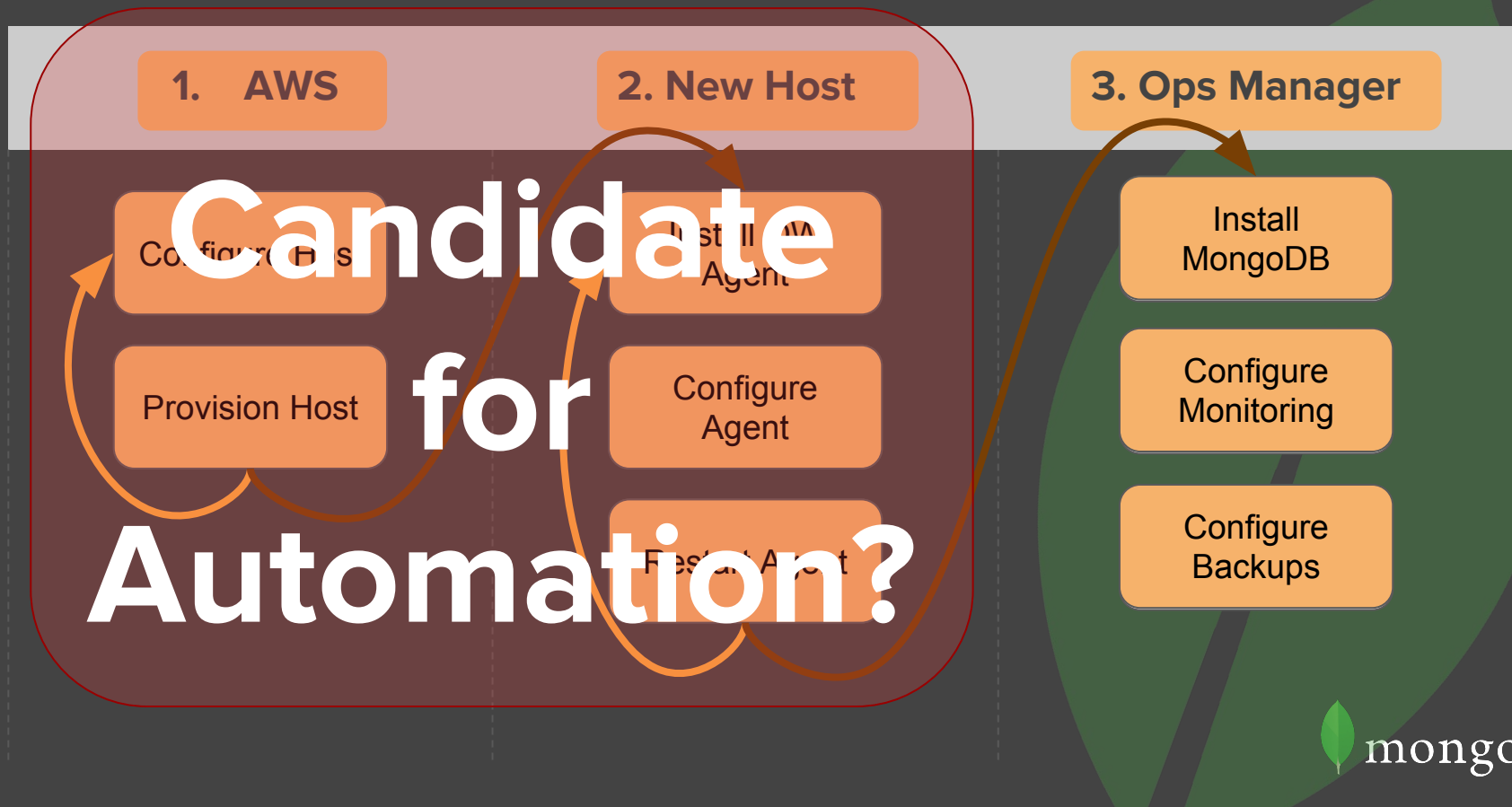
MongoDB Deployment - Ops Manager



MongoDB Deployment - Cloud Manager



MongoDB Deployment - Ops Manager





ANSIBLE

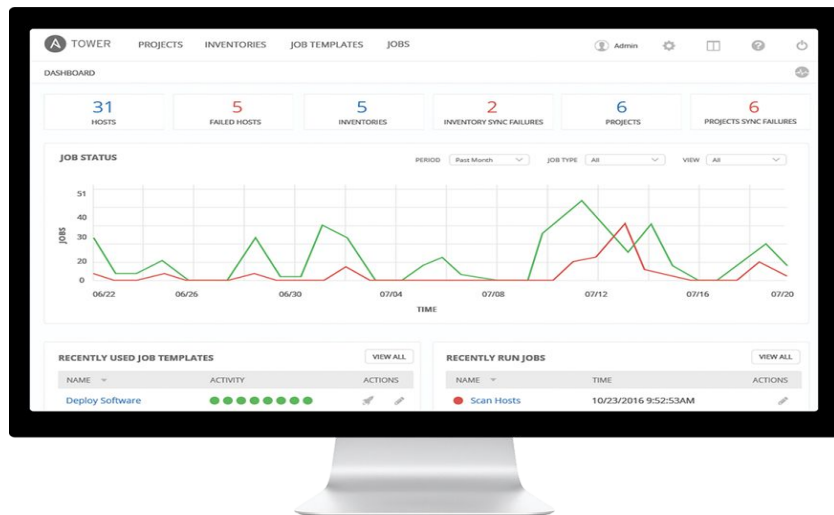
WHAT IS ANSIBLE?

ANSIBLE

It's a **simple automation language** that can perfectly describe an IT application infrastructure in Ansible Playbooks.

It's an **automation engine** that runs Ansible Playbooks.

Ansible Tower is an **enterprise framework** for controlling, securing and managing your Ansible automation with a **UI and RESTful API**.





SIMPLE

Human readable automation
No special coding skills needed
Tasks executed in order
Get productive quickly



POWERFUL

App deployment
Configuration management
Workflow orchestration
Orchestrate the app lifecycle



AGENTLESS

Agentless architecture
Uses OpenSSH & WinRM
No agents to exploit or update
More efficient & more secure



**ANSIBLE
TOWER**
by Red Hat®

TOWER EMPOWERS TEAMS TO AUTOMATE

CONTROL

Scheduled and
centralized jobs

KNOWLEDGE

Visibility and
compliance

DELEGATION

Role-based access
and self-service

SIMPLE

Everyone speaks the
same language

POWERFUL

Designed for
multi-tier deployments

AGENTLESS

Predictable, reliable,
and secure

AT ANSIBLE'S CORE IS AN **OPEN-SOURCE** AUTOMATION ENGINE

Getting Started with Ansible - Lexicon

- **Commands**

- ansible, ansible-playbook

- **Hosts File**

- How ansible finds the servers you want to manage

- **Plays, Playbooks**

- The execution tools to carry out your management

- **Tasks & Modules**

- The components that connect ansible to the servers

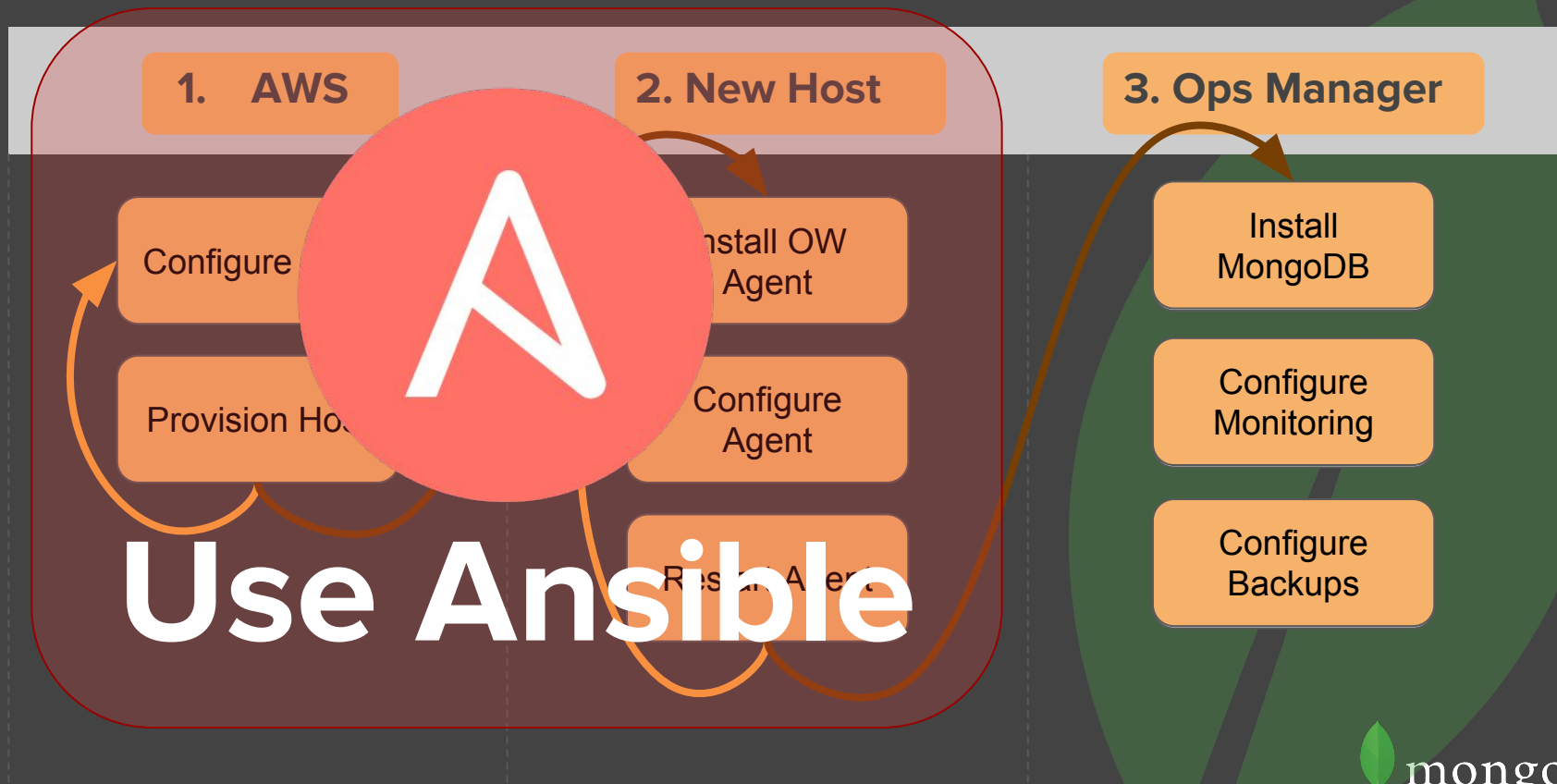
Getting Started with Ansible - Groups

```
ansible -m ping -i ansible-hosts  
opsManager
```

Here I'm telling ansible to use the ping module against the opsManager group in my ansible-hosts file.



MongoDB Deployment - Ops Manager



Demo of Ansible

Demonstration Setup

Command	Description
<code>ansible -m ping -i ansible-hosts all</code>	Using the ping module and the hosts file <code>ansible-hosts</code> in the current directory, ping all hosts
<code>export ANSIBLE_HOSTS=./ansible_hosts</code>	Save some time - ansible uses environment variables extensively
<code>ansible -m ping all</code>	Same as before - but now ansible leverages the env var to find the ansible hosts file.

Demonstration

Command	Description
<code>ansible -m script test.sh all</code>	Here, we take a local script and execute it across all of our hosts. Ansible takes that script and delivers it via scp to the hosts, executes it and captures the output.
<code>export ANSIBLE_HOSTS=./ansible_hosts</code>	Save some time - ansible uses environment variables extensively
<code>ansible -m ping all</code>	Same as before - but now ansible leverages the env var to find the ansible hosts file.

Appendix A - /etc/ansible/hosts sample

```
[opsManager]
```

```
ec2-54-93-114-205.eu-central-1.compute.amazonaws.com ansible_user=ec2-user
```

```
[amlReplicaSet]
```

```
ec2-54-93-79-122.eu-central-1.compute.amazonaws.com ansible_user=ec2-user
```

```
ec2-54-93-176-246.eu-central-1.compute.amazonaws.com ansible_user=ec2-user
```

```
ec2-54-93-207-148.eu-central-1.compute.amazonaws.com ansible_user=ec2-user
```

```
[amlReplicaSet:vars]
```

```
opsmanagerurl=http://ec2-54-93-114-205.eu-central-1.compute.amazonaws.com:8080
```

```
opsmanager=ec2-54-93-114-205.eu-central-1.compute.amazonaws.com
```



mongoDB

Appendix B - ansible.cfg

[defaults]

host_key_checking = False

private_key_file = PATH/TO/AWS/KEY.FILE

[ssh_connection]

control_path = %(directory)s/%%C



Post-Demo Check-in

- **Ops Manager**
 - Automation for all the things you do with MongoDB - except deploying the agent.
- **Ansible**
 - Automation for configs, deployment and more - especially deploying O/M Agents
- I hope you learned some ways you can relieve the pain associated with managing your MongoDB deployment.

Questions?

Get These Scripts and Slides:

<http://github.com/mrlynn/ansible-demo>



mongoDB