

# Automation with Ansible and EC2

Michael Menzel  
Vineet Hingorani  
September 02, 2015



# Agenda

---

## Recap: Cloud Computing & Compute Clouds

- Definition, Characteristics, Delivery Models, Core Concepts

## Infrastructure Automation with OSCM

- Concept of OSCM, Opscode Chef, Saltstack Salt

## Ansible

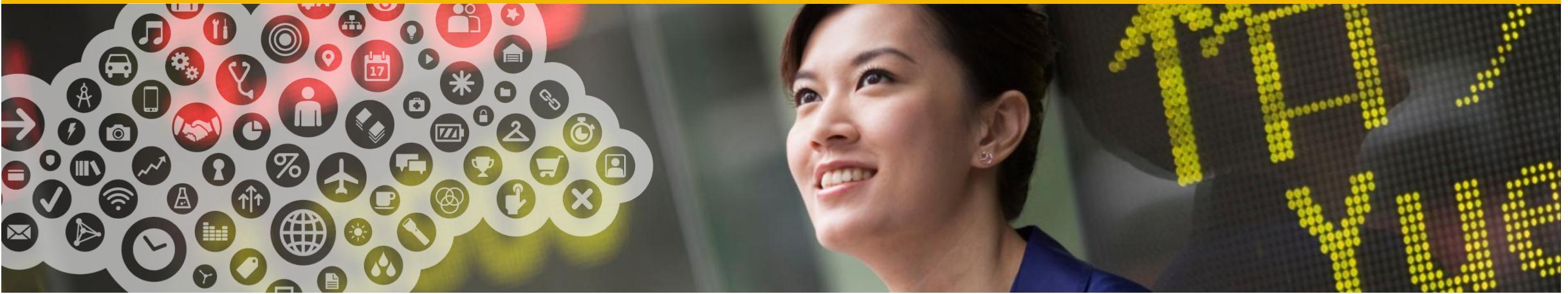
- Explain the Ansible Approach, Share our Experiences

## Examples with Ansible

- Start Example App on multiple Instances with Load Balancer, Start MetricStorageService and use Auto-Scaling

## Outlook

- Ansible Tower, Discussion on how to leverage



# Recap: Cloud Computing

# Cloud Computing Definition

---

*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*

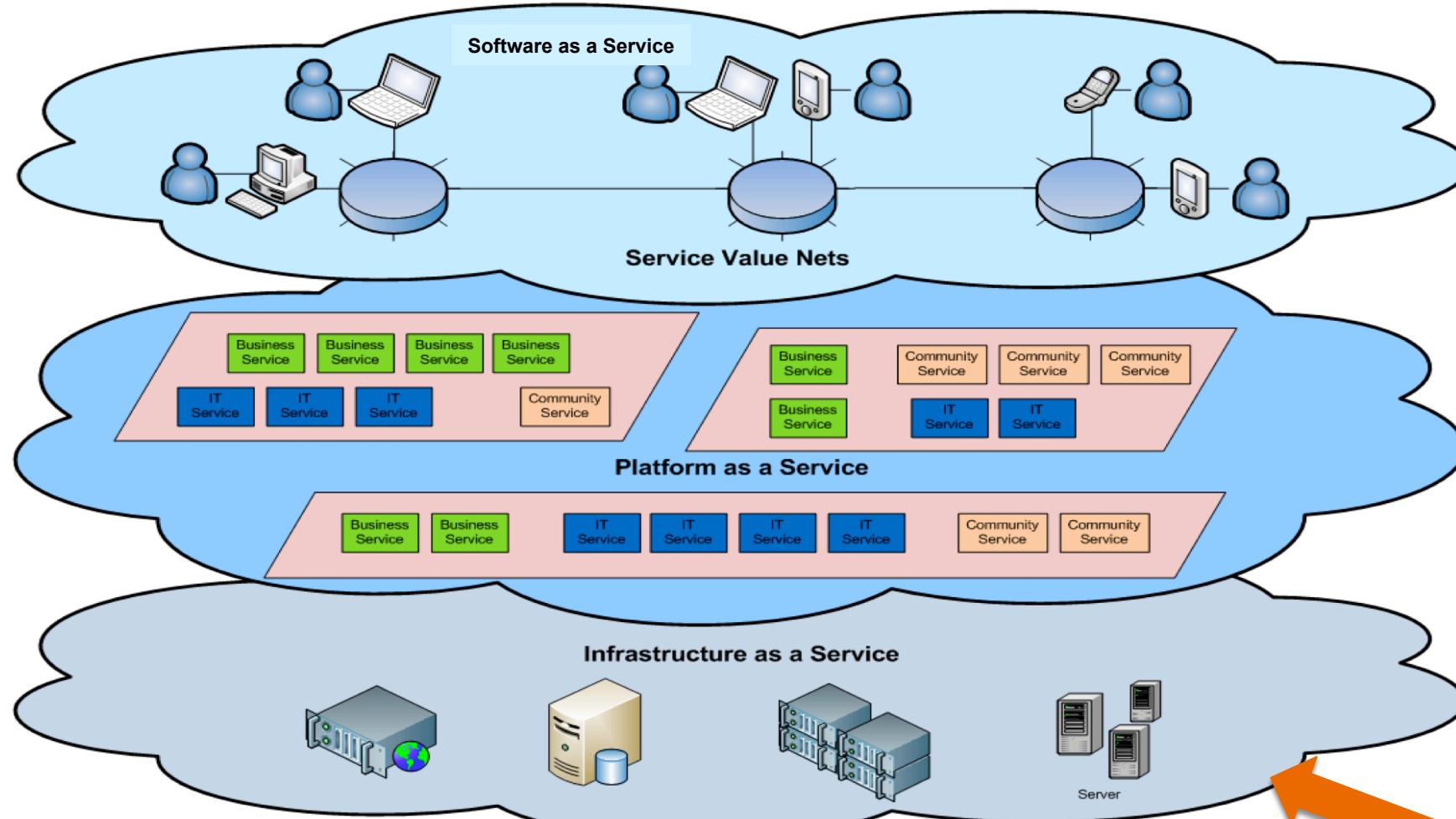
Source: National Institute for Standards & Technology (NIST)  
<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>

# Cloud Computing Characteristics



Relevant to this talk

# Cloud Delivery Models



## SaaS

Web Applications like Google Apps, Salesforce etc.

## PaaS

Development or execution environments like Google App Engine, Windows Azure

## IaaS

Offerings like Google Storage, Amazon Web Services etc.

**Focus of this talk**

# Amazon Web Services

## Amazon Web Services

### Compute

- EC2**  
Virtual Servers in the Cloud
- EC2 Container Service**  
Run and Manage Docker Containers
- Elastic Beanstalk**  
Run and Manage Web Apps
- Lambda**  
Run Code in Response to Events

### Storage & Content Delivery

- S3**  
Scalable Storage in the Cloud
- CloudFront**  
Global Content Delivery Network
- Elastic File System** PREVIEW  
Fully Managed File System for EC2
- Glacier**  
Archive Storage in the Cloud
- Storage Gateway**  
Integrates On-Premises IT Environments with Cloud Storage

### Database

- RDS**  
MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
- DynamoDB**  
Predictable and Scalable NoSQL Data Store
- ElastiCache**  
In-Memory Cache
- Redshift**  
Managed Petabyte-Scale Data Warehouse Service

### Networking

- VPC**  
Isolated Cloud Resources
- Direct Connect**  
Dedicated Network Connection to AWS
- Route 53**  
Scalable DNS and Domain Name Registration

### Developer Tools

- CodeCommit**  
Store Code in Private Git Repositories
- CodeDeploy**  
Automate Code Deployments
- CodePipeline**  
Release Software using Continuous Delivery

### Management Tools

- CloudWatch**  
Monitor Resources and Applications
- CloudFormation**  
Create and Manage Resources with Templates
- CloudTrail**  
Track User Activity and API Usage
- Config**  
Track Resource Inventory and Changes
- OpsWorks**  
Automate Operations with Chef
- Service Catalog**  
Create and Use Standardized Products

### Security & Identity

- Identity & Access Management**  
Manage User Access and Encryption Keys
- Directory Service**  
Host and Manage Active Directory
- Trusted Advisor**  
Optimize Performance and Security

### Analytics

- EMR**  
Managed Hadoop Framework
- Data Pipeline**  
Orchestration for Data-Driven Workflows
- Kinesis**  
Real-time Processing of Streaming Big Data
- Machine Learning**  
Build Smart Applications Quickly and Easily

### Mobile Services

- Cognito**  
User Identity and App Data Synchronization
- Device Farm**  
Test Android, Fire OS, and iOS apps on real devices in the Cloud
- Mobile Analytics**  
Collect, View and Export App Analytics
- SNS**  
Push Notification Service

### Application Services

- API Gateway**  
Build, Deploy and Manage APIs
- AppStream**  
Low Latency Application Streaming
- CloudSearch**  
Managed Search Service
- Elastic Transcoder**  
Easy-to-use Scalable Media Transcoding
- SES**  
Email Sending Service
- SQS**  
Message Queue Service
- SWF**  
Workflow Service for Coordinating Application Components

### Enterprise Applications

- WorkSpaces**  
Desktops in the Cloud
- WorkDocs**  
Secure Enterprise Storage and Sharing Service
- WorkMail** PREVIEW  
Secure Email and Calendaring Service

EC2 is the elastic compute cloud of AWS offering “computers” as a service.



# EC2 Core Concepts

---

## Amazon Machine Image (AMI)

An encrypted file stored in Amazon storage, containing all the information necessary to boot instances of a customer's software

- An AMI is like a bootable root disk, which can be pre-defined or user-built.
  - Public AMIs: Pre-configured, template AMIs. Eg. Ubuntu 14.04, Red Hat Enterprise 7
  - Private AMIs: User-built AMI containing private applications, libraries, data and associated configuration settings. Eg. Ubuntu 14.04 image with HANA as a software.
- There is a community of AMI creators, e.g. [bitnami.com](http://bitnami.com), [thecloudmarket.com](http://thecloudmarket.com), [aws.amazon.com/marketplace](http://aws.amazon.com/marketplace)

## Instance

The running system based on an AMI

- All instances based on the same AMI begin executing identically. An instance can be launched in a few minutes.
- Any information on them is lost when the instances are terminated or if they fail.
- Data can be made persistent by use of Amazon storage services.



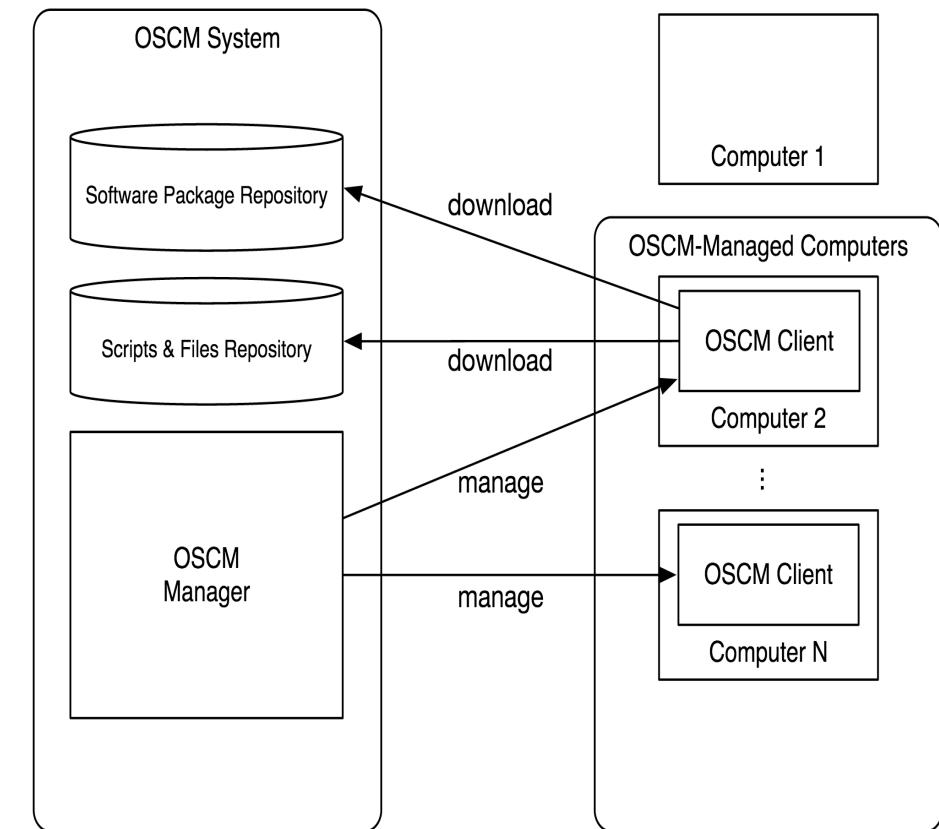
# Infrastructure Automation

# Infrastructure Automation Approaches

## Scripting Environments

Applying configuration to nodes (single or hundreds)

- Includes automating everything
  - Installing OS
  - Configuring servers on instances
  - Configuring how different softwares communicate with each other
- Setting up and replicating environments in less error-prone manner
  - Automated testing
  - Documentation
  - Versioning of infrastructure setup



# OpsCode Chef

---

## Configuration Management Tool

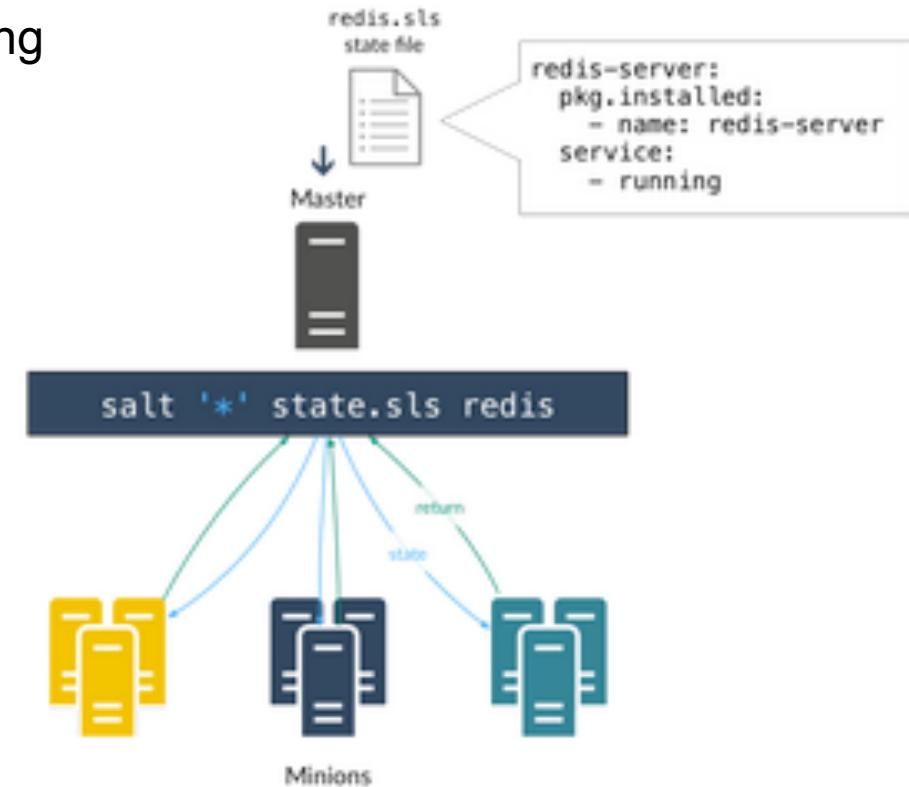
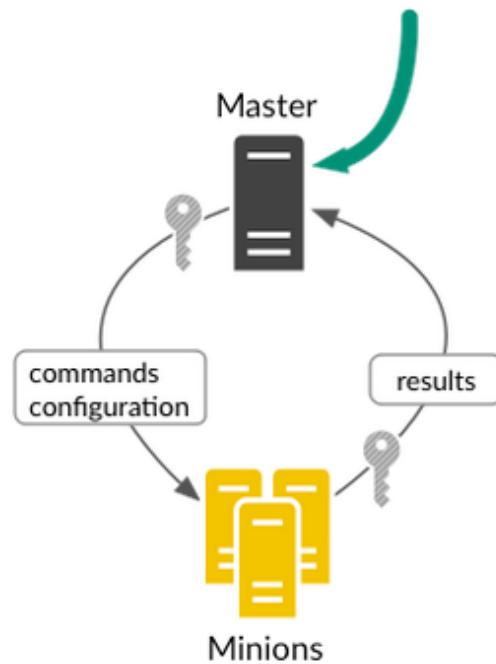
- Uses Ruby based Domain Specific Language for writing recipes
- Can integrate with cloud-based platform but not originally built for that
- The API is written both in Ruby and Erlang
- Supports multiple platforms for clients Solaris, FreeBSD, OS X, Ubuntu, Debian, Fedora, Microsoft Windows
- Server is supported on Oracle Linux/CentOS and Ubuntu



# SaltStack Salt

## Another Management Software for CloudOps, ITOps, DevOps

- Written in Python
- Configuration files written in YAML or PyDSL
- Master-Client architecture and needs a Master always running





# Ansible





# What is Ansible?

## YAML based language

- Descriptive language for automation of infrastructure setups

## Concepts

- Owns concepts to structure and reuse definition files:
  - Variables (vars)
  - Tasks
  - Profiles
  - Handlers
  - Templates (jinja2)
- Provides reusable modules to run tasks
- Can manage known hosts and cloud services

```
YAML: true
Fruits:
- name: Apple
  price: .50
- name: Banana
  price: .80
```



## Playbooks

- Ansible's configuration, deployment and orchestration language.
- The design plan. The set of steps. The work process.

## Modules

- The tasks that are actually executed in the playbooks.
- They do the actual work.
  - Core Modules
    - Maintained by the Ansible team shipped with Ansible itself.
  - Extra Modules
    - Maintained by the community and may not ship with Ansible in future.

```
---
```

```
- hosts: webservers
```

```
  vars:
```

```
    http_port: 80
```

```
    max_clients: 200
```

```
  remote_user: root
```

```
  tasks:
```

```
    - name: ensure apache is at the latest version
```

```
      yum: pkg=httpd state=latest
```

```
    - name: write the apache config file
```

```
      template: src=/srv/httpd.j2 dest=/etc/httpd.conf
```

```
      notify:
```

```
        - restart apache
```

```
    - name: ensure apache is running (and enable it at boot)
```

```
      service: name=httpd state=started enabled=yes
```

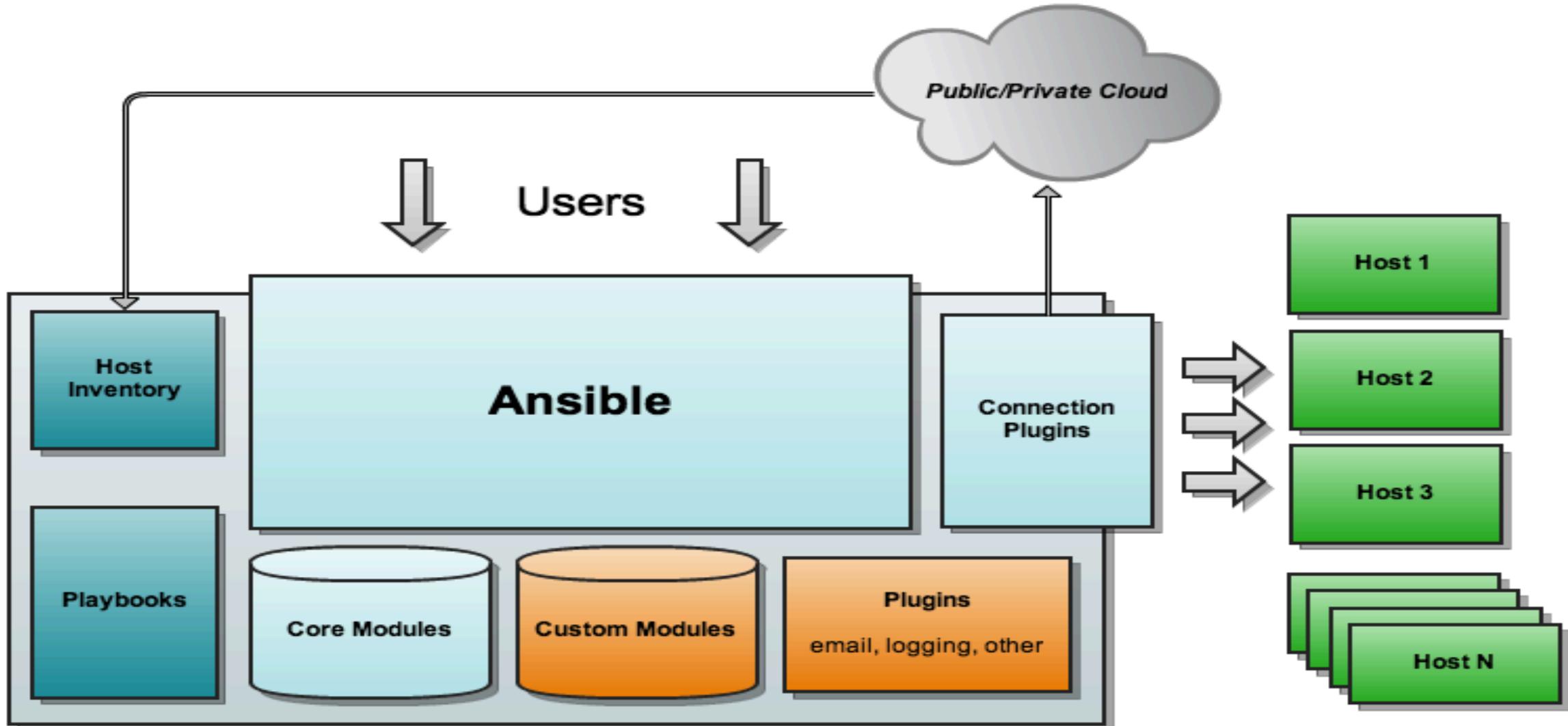
```
  handlers:
```

```
    - name: restart apache
```

```
      service: name=httpd state=restarted
```



# Ansible Setup

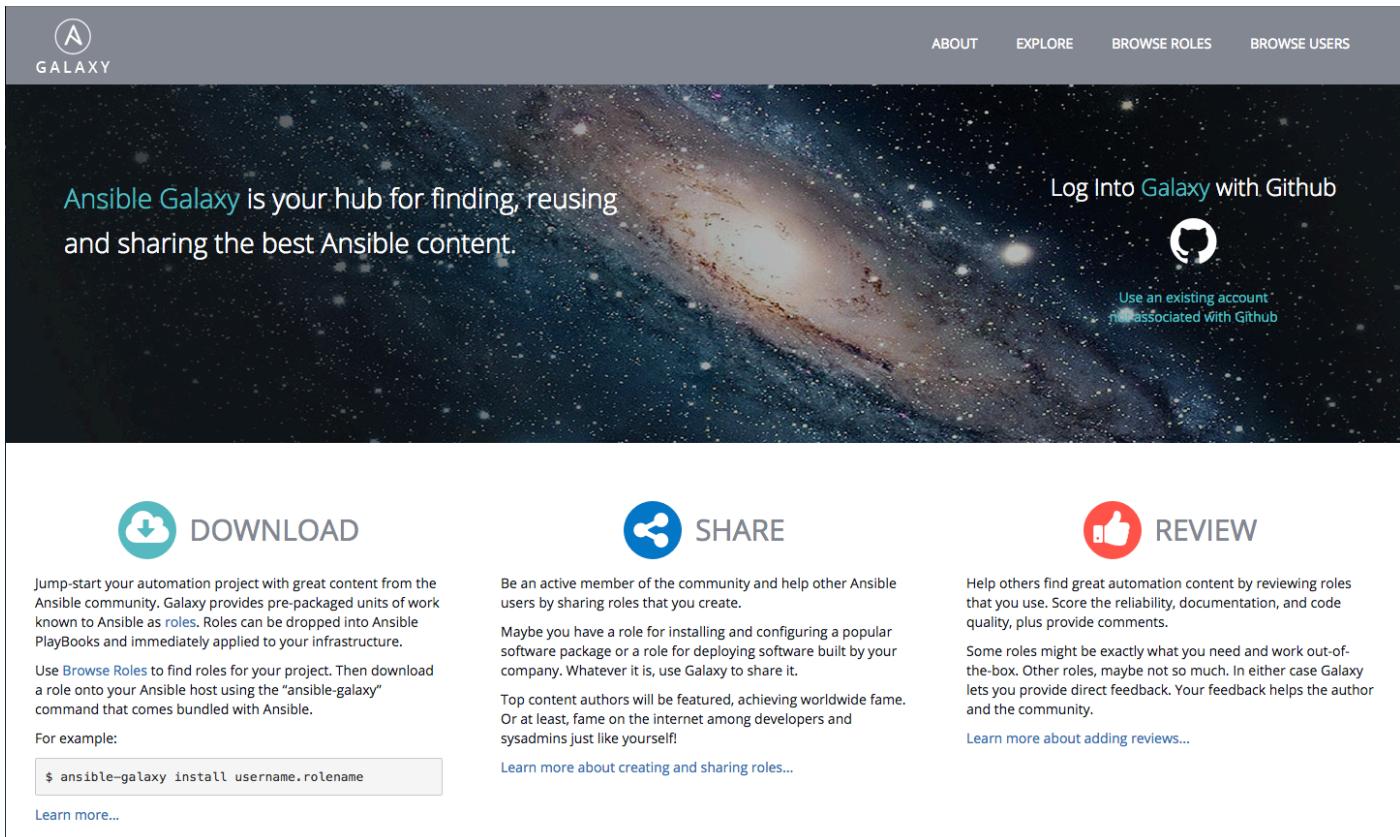


# Ansible Galaxy

**Ansible modules repo maintained by a big community**

**Many custom modules and code snippets (roles) are shared by ansible users**

**Also integrations with clouds, vagrant, etc.**



The screenshot shows the Ansible Galaxy homepage. At the top right, there are navigation links: ABOUT, EXPLORE, BROWSE ROLES, and BROWSE USERS. On the far right, there's a "Log Into Galaxy with Github" button with a GitHub icon, and below it, a link to "Use an existing account associated with Github". The main header features the "GALAXY" logo with a stylized 'A' icon. Below the header, a large banner image of a galaxy serves as the background for the text: "Ansible Galaxy is your hub for finding, reusing and sharing the best Ansible content." In the bottom left, there's a "DOWNLOAD" section with a cloud icon, explaining how to jump-start an automation project using pre-packaged roles. It includes a command-line example: "\$ ansible-galaxy install username rolename". In the bottom right, there are three sections: "SHARE" with a share icon, "REVIEW" with a thumbs-up icon, and both sections have descriptive text about contributing to the community and providing feedback.

ABOUT EXPLORE BROWSE ROLES BROWSE USERS

Log Into Galaxy with Github

Use an existing account associated with Github

GALAXY

Ansible Galaxy is your hub for finding, reusing and sharing the best Ansible content.

DOWNLOAD

Jump-start your automation project with great content from the Ansible community. Galaxy provides pre-packaged units of work known to Ansible as [roles](#). Roles can be dropped into Ansible PlayBooks and immediately applied to your infrastructure.

Use [Browse Roles](#) to find roles for your project. Then download a role onto your Ansible host using the "ansible-galaxy" command that comes bundled with Ansible.

For example:

```
$ ansible-galaxy install username.rolename
```

Learn more...

SHARE

Be an active member of the community and help other Ansible users by sharing roles that you create.

Maybe you have a role for installing and configuring a popular software package or a role for deploying software built by your company. Whatever it is, use Galaxy to share it.

Top content authors will be featured, achieving worldwide fame. Or at least, fame on the internet among developers and sysadmins just like yourself!

Learn more about creating and sharing roles...

REVIEW

Help others find great automation content by reviewing roles that you use. Score the reliability, documentation, and code quality, plus provide comments.

Some roles might be exactly what you need and work out-of-the-box. Other roles, maybe not so much. In either case Galaxy lets you provide direct feedback. Your feedback helps the author and the community.

Learn more about adding reviews...

# With Ansible it's like...

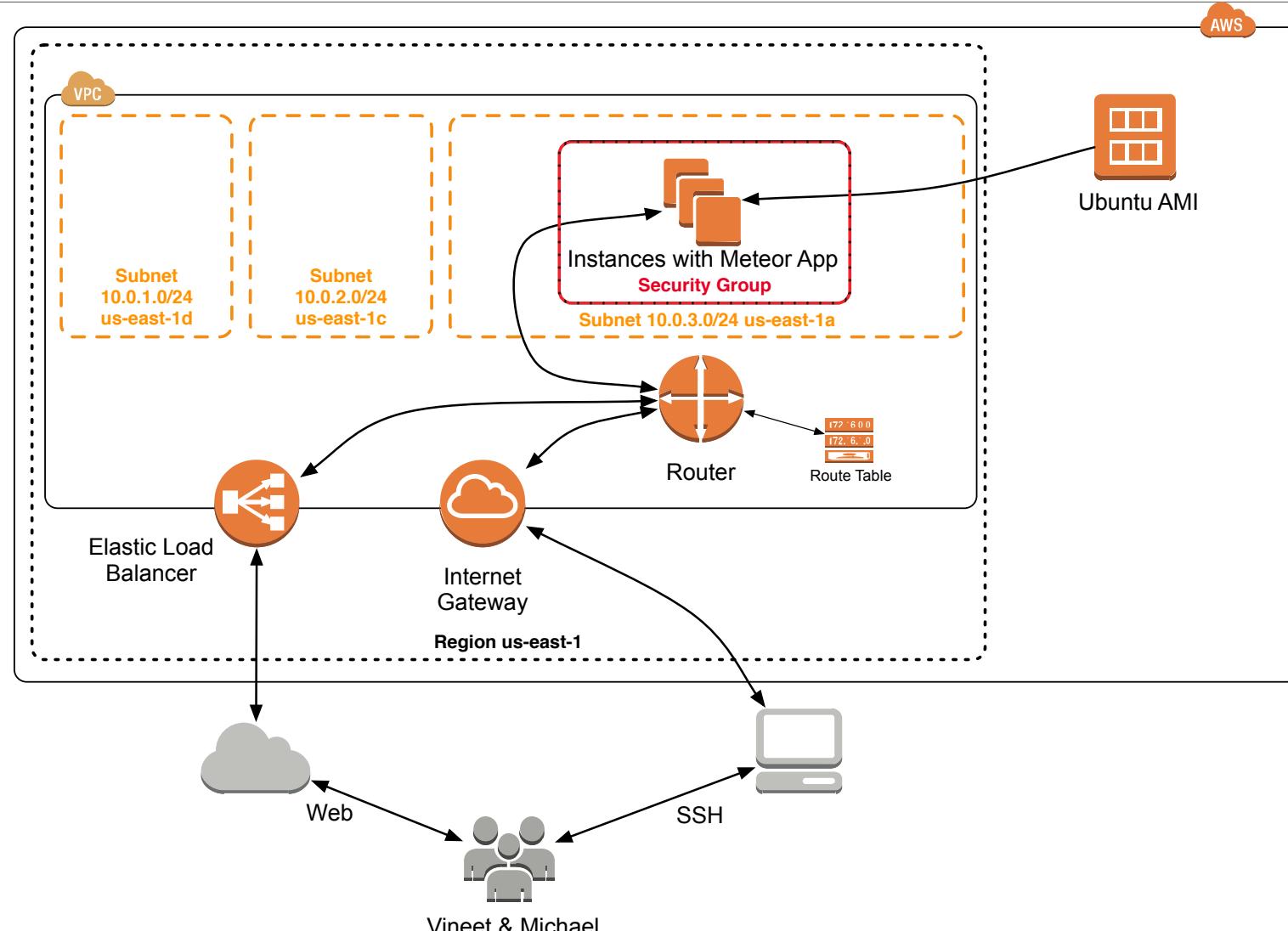
---

- **Defining and documenting** your infrastructure
- Thinking about a **final state** (idempotence)
- **Modules pave the way** to simple setups. Guides are available & reuse is king!
- **Iteration works** in descriptive files, too!
- No more (or less) scripting, but **still thinking!**
- Support for **continuous delivery and rolling upgrades**
- **Problems with Python or SSH** may occur (but could be with any technology)

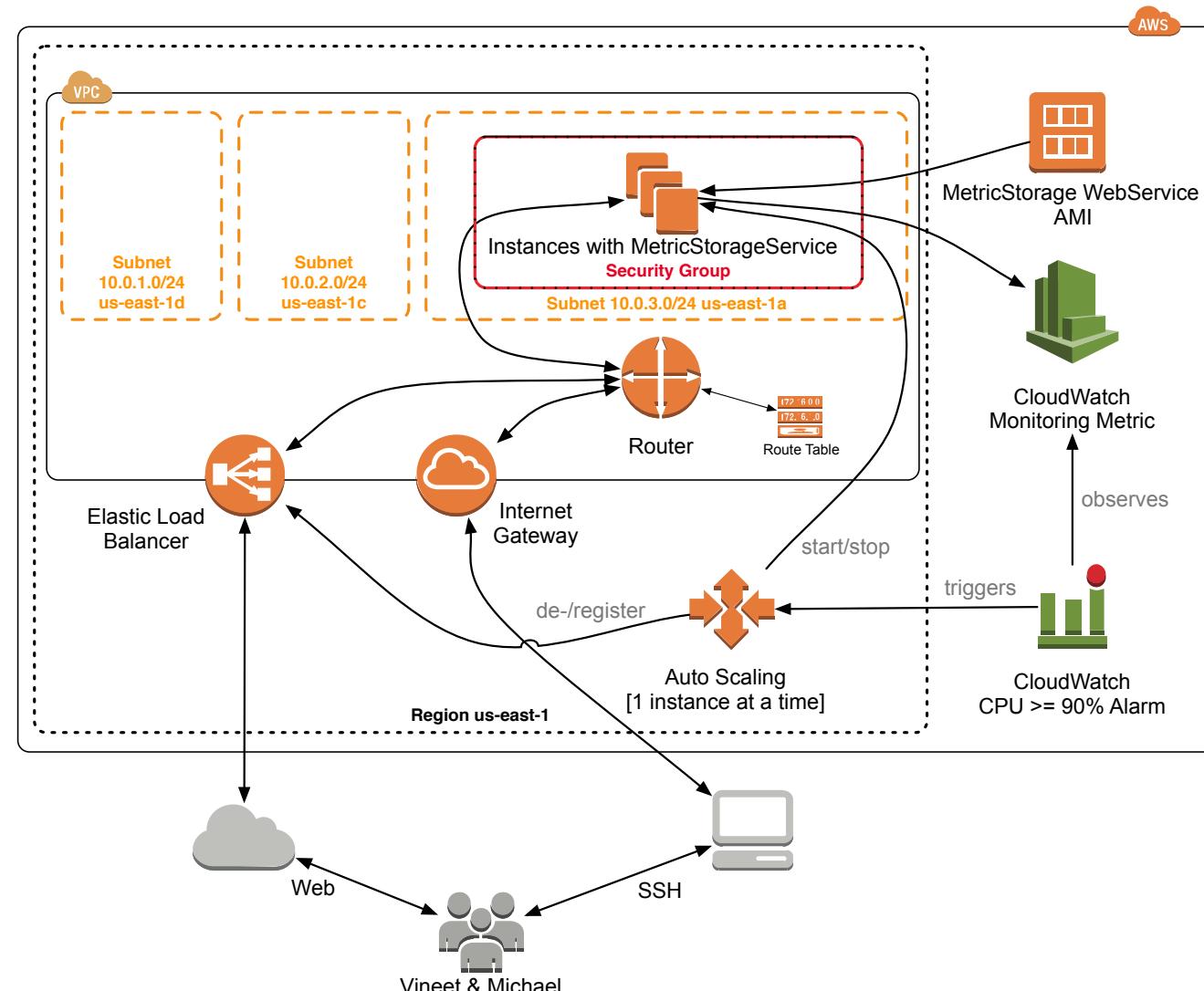


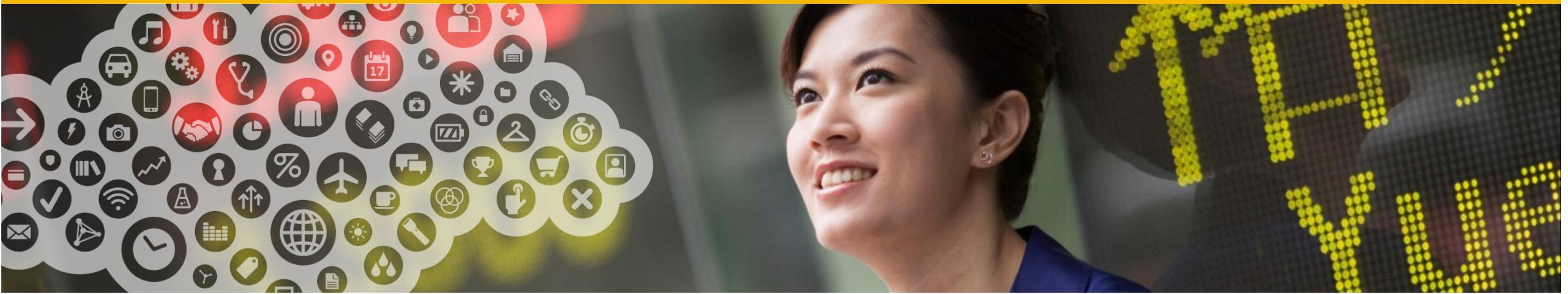
# Examples with Ansible

# Ansible Example 1: Target Architecture



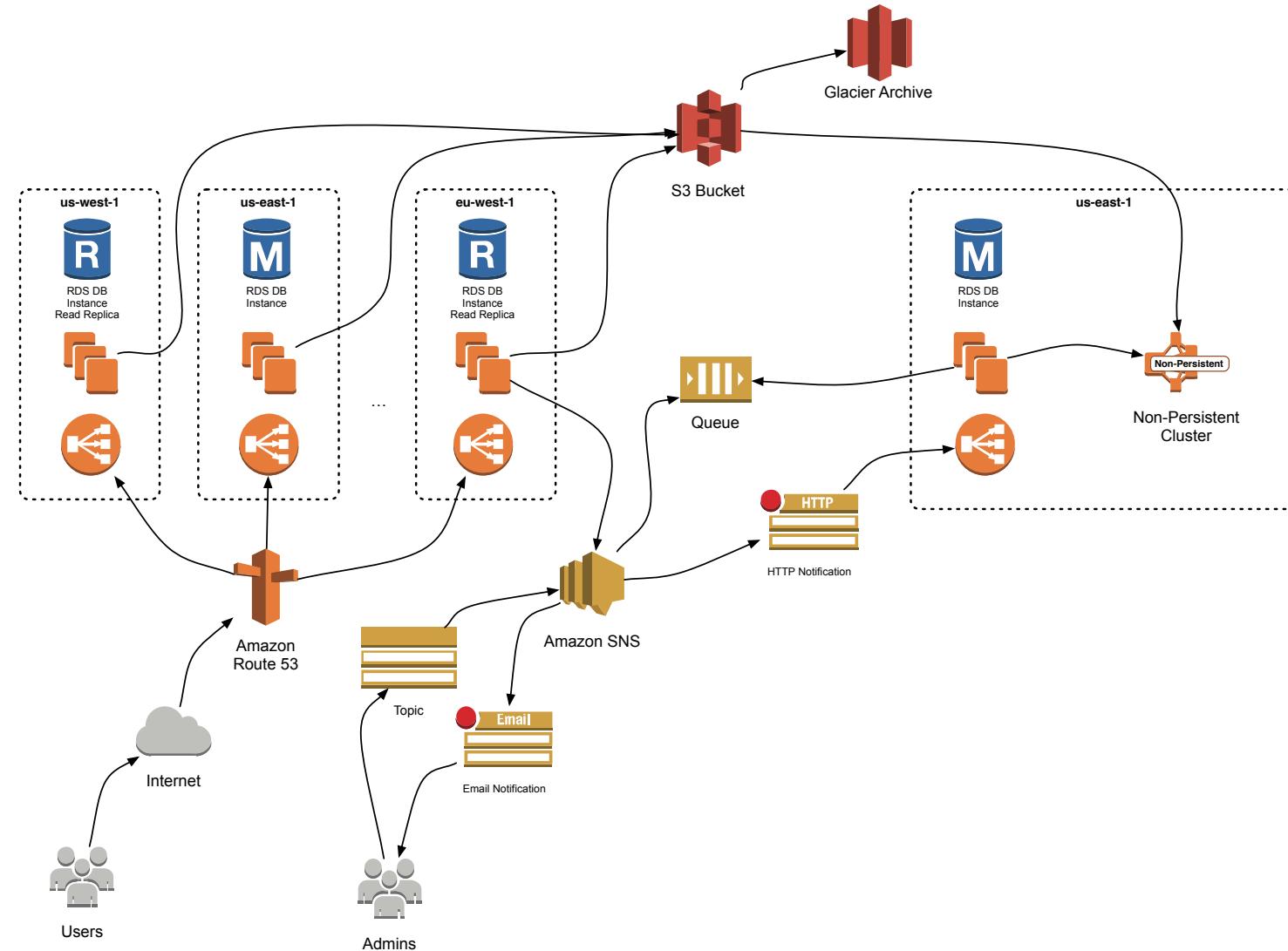
# Ansible Example 2: Target Architecture





# Outlook

# Imagine more complex architectures



# Ansible Tower

Provides a comprehensive management GUI

Costs scale with number of managed machines (**good & bad**)

Adds features such as:

- Scheduling of ansible tasks
- Direct machine access
- Logging of user actions

The screenshot shows the Ansible Tower web interface. At the top, there is a navigation bar with links for Home, Organizations, Users, Teams, Credentials, Projects, Inventories, Job Templates, and Jobs. The 'Jobs' link is highlighted. On the left, there is a sidebar with a tree view of the organization structure. The main content area is titled 'Jobs' and contains four sections: 'Completed', 'Active', 'Queued', and 'Scheduled'. Each section has a table with columns for Job ID, Status, Started/Finished On, Type, Name, and Actions. The 'Completed' section shows five entries, the 'Active' section shows one, and the 'Queued' and 'Scheduled' sections show none. There are also search bars and pagination controls (e.g., 1, 2, 3, 4, 5, >) at the bottom of each table.

Completed					
Job ID	Status	Finished On	Type	Name	Actions
26	green	06/03/14 15:42:40	Playbook Run	Deploy LAMP HA Proxy to AWS	View
25	green	06/03/14 15:42:14	Playbook Run	Deploy Rolling Update to Rackspace	View
24	green	06/03/14 15:41:40	SCM Update	Update from Production SCM Repository	
23	red	06/03/14 15:40:31	Playbook Run	Deploy LAMP HA Proxy to Rackspace	View
22	green	06/03/14 15:05:05	Inventory Sync	AWS (AWS)	

Page 1 of 5 for 25 completed jobs

Active					
Job ID	Status	Started On	Type	Name	Actions
27	green	06/03/14 15:43:35	Playbook Run	Provision QA Environment on EC2	View

Page 1 of 1 for 1 running jobs

How can we leverage

---

**Questions?**

**How can we use this technology?**



# Thank you

Contact information:

Michael Menzel  
Vineet Hingorani  
C3.05