RSA CONFERENCE 2014
FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Share.
Learn.
Secure.
Capitalizing on
Collective Intelligence

Virtualization and Cloud: Orchestration, Automation, and Security Gaps

SESSION ID: CSV-R02

Dave Shackleford
Founder & Principal Consultant
Voodoo Security
@daveshackleford

Introduction

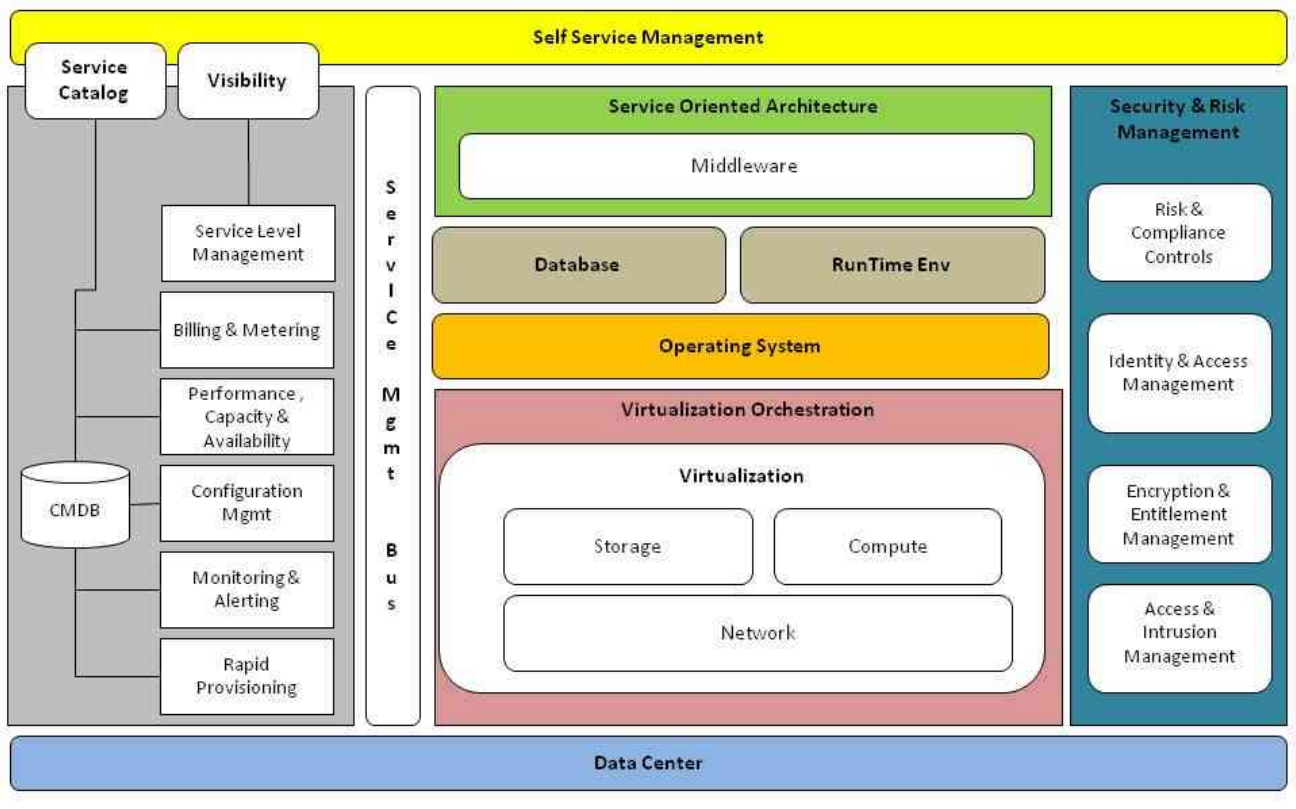
- Private cloud implementations incorporate a lot of “moving parts”
- With growth and maturity of a cloud infrastructure, most incorporate orchestration and automation functions
- These are rarely secured
  - Few vendor-integrated options
  - Little operational attention to risk and security
- Let’s delve into potential risks and what we can do about them.

Architecture and Definitions

Orchestration

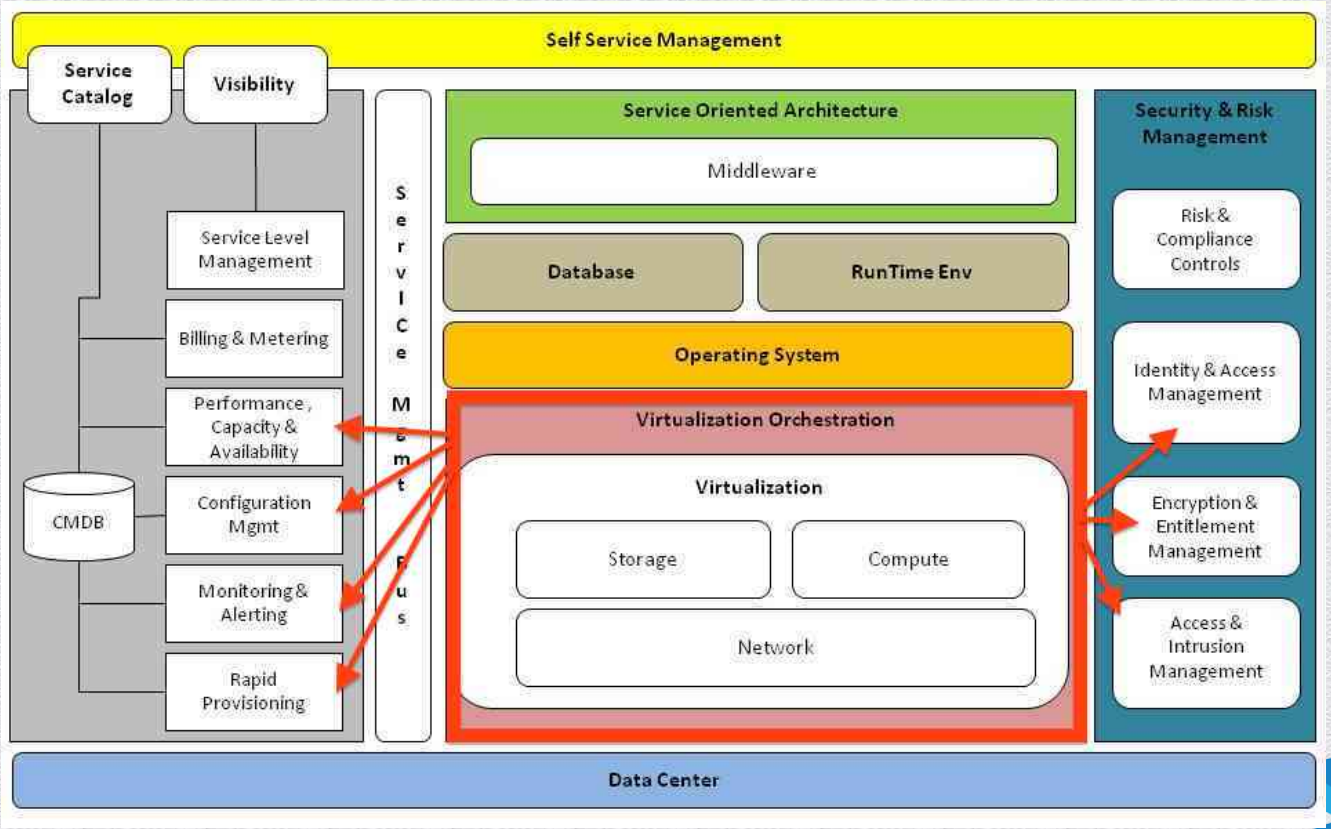
- The orchestration “layer” allows for planned automation and provisioning tasks within a cloud environment
- Typically managed by a distinct software platform
  - Can be open-source or commercial
- Often relies heavily on APIs
- Often focused on configuration, changes and change management, and provisioning
- Can also play a role in monitoring, security, and other functions

Private Cloud Architecture



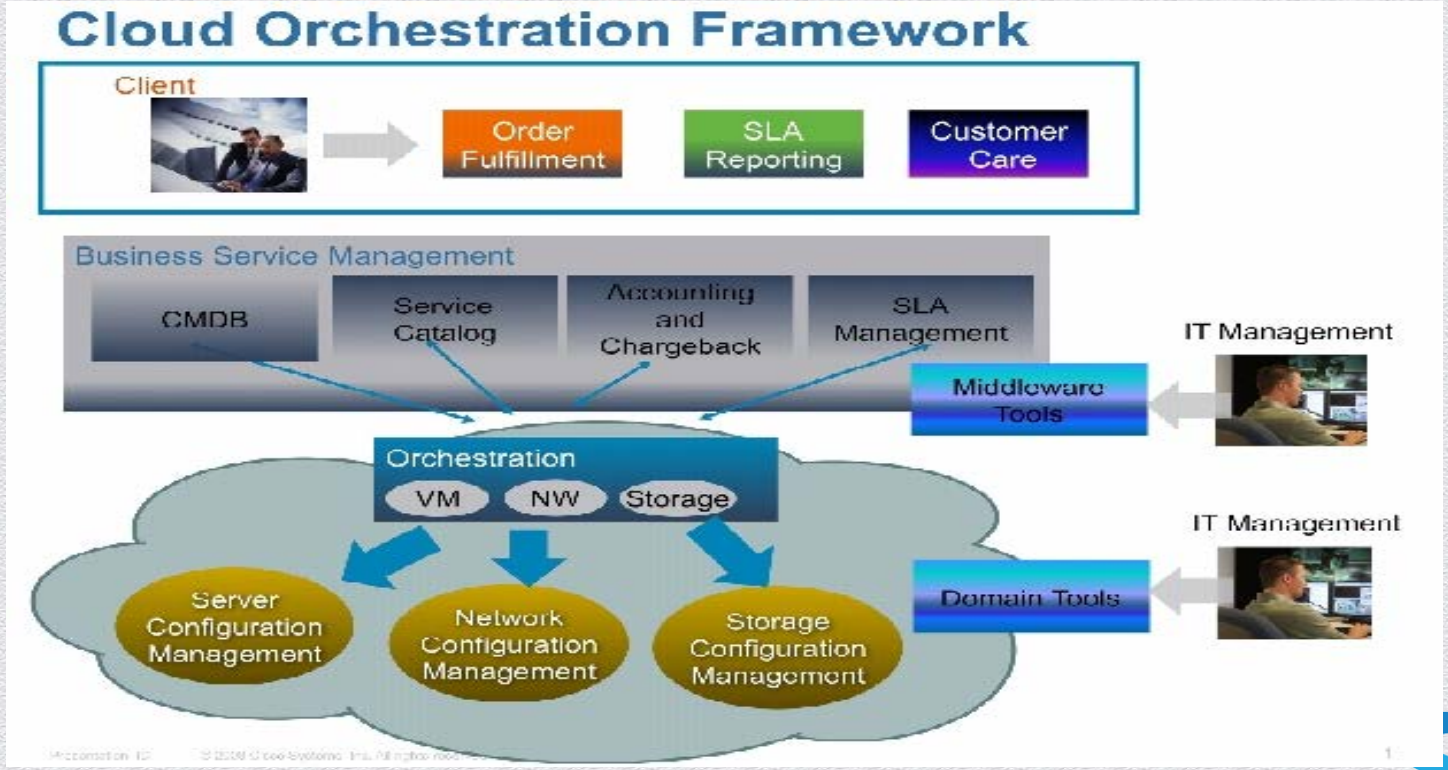
Reference: http://inthepassing.files.wordpress.com/2010/01/cloud-ref-arch.jpg

Private Cloud Architecture: Single Point of Failure?



Reference: http://inthepassing.files.wordpress.com/2010/01/cloud-ref-arch.jpg

Another Orchestration Model Example



Source: Cisco.com

What about automation?

- Orchestration relies heavily on automation tools and “rules”
- Automation tools can easily manage a number of common cloud “activities”
- If **misused**, however, automation could easily lead to chaos
  - Malicious commands
  - Service disruption
  - File/system/app modification

Automation Frameworks and Tools

- LOTS of tools emerging and available, both open and commercial
  - IBM Rational
  - Cisco Intelligent Automation for Cloud (CIAC)
  - Dell Cloud Manager
  - Puppet (Puppet Labs)
  - OpsCode Chef
  - CFEngine
- OASIS also defined Topology and Orchestration Specification for Cloud Applications (TOSCA)
  - XML-based language defined for service/template provisioning

More on Puppet and Chef

Puppet Labs' Puppet

- Centrally-defined resources are provisioned to systems and monitored
- Configuration management for OS, network, middleware, and application tiers is possible
- Integrates natively with AWS, VMware, OpenStack, etc.

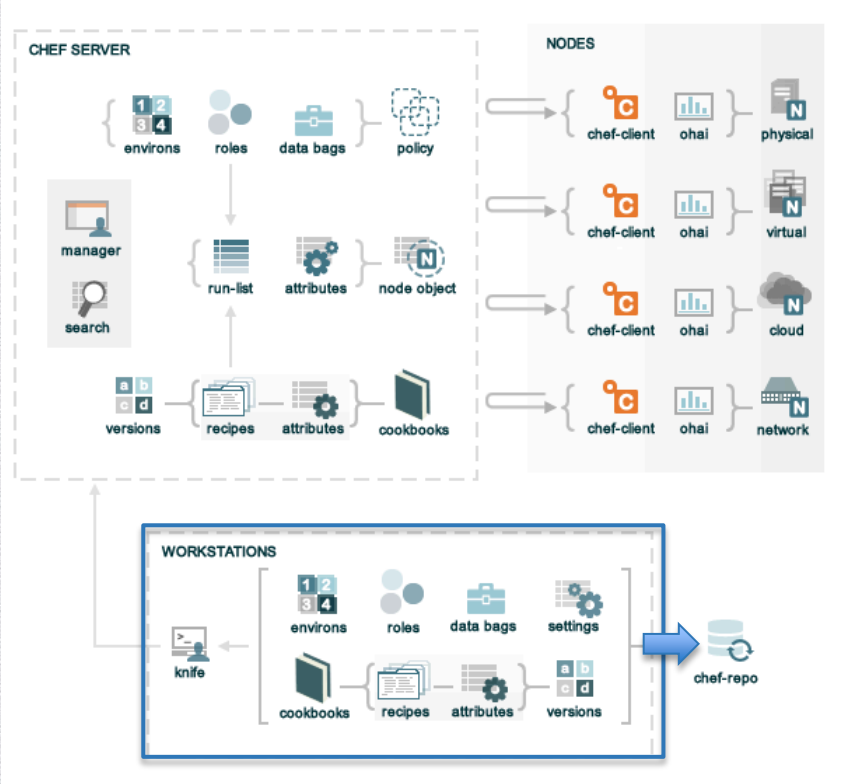
Opscode Chef

- 3-tier architecture:
  - Nodes
  - Chef Server
  - Workstations
- Leverages Ruby "recipes" that are loaded to configuration "cookbooks"

Common Orchestration Tasks

- Configuration Management
  - Storage
  - VM/Compute
  - Network
- Provisioning
  - VMs and application instances
- IT Automation and DevOps
- Security & Compliance assessment, monitoring, and reporting

An Example Use Case

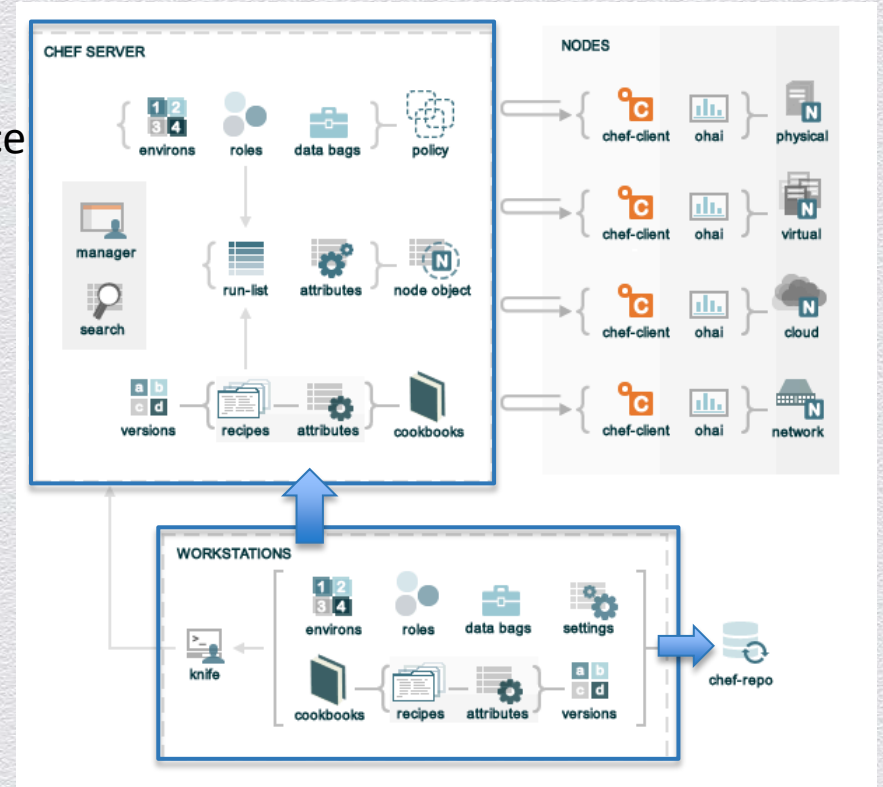


1. Orchestration Engineer defines a resource and commits to the repository

Source: http://docs.opscode.com/chef_overview.html

An Example Use Case

2. Automation Tools write the new resource definition to the main server, where it's added to a defined workflow and policy

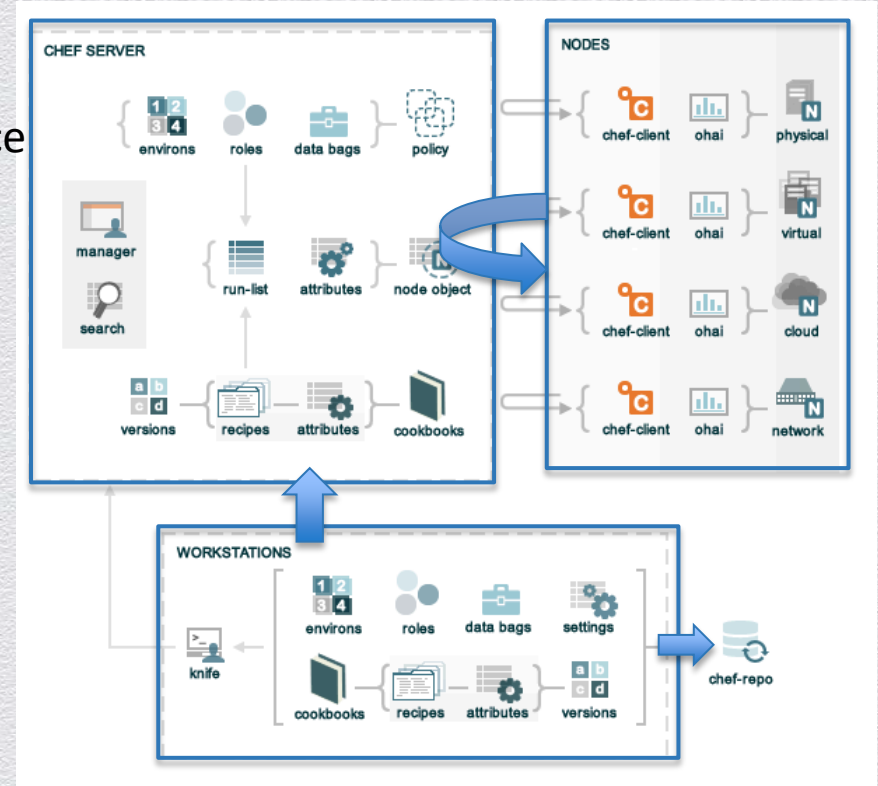


1. Orchestration Engineer defines a resource and commits to the repository

Source: http://docs.opscode.com/chef_overview.html

An Example Use Case

2. Automation Tools write the new resource definition to the main server, where it's added to a defined workflow and policy

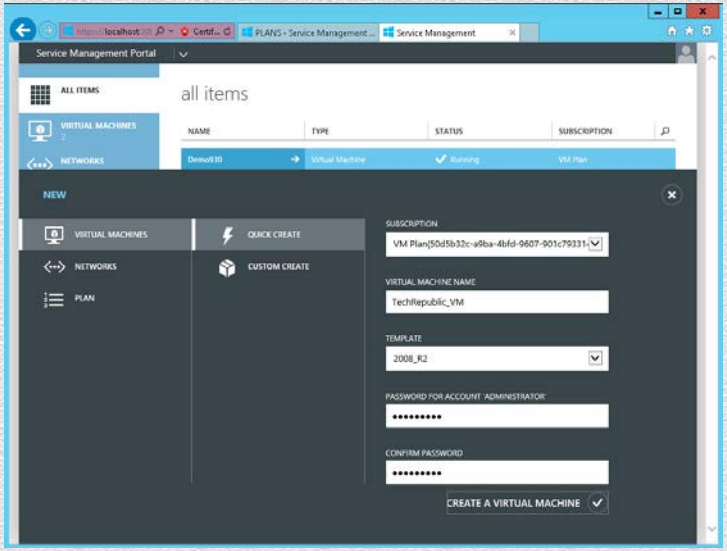


3. Nodes pull the new resource config, making configuration and local policy changes as needed

1. Orchestration Engineer defines a resource and commits to the repository

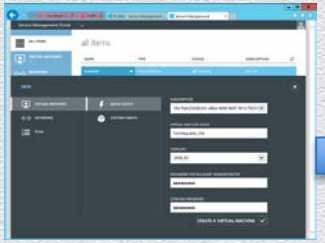
Source: http://docs.opscode.com/chef_overview.html

Another Example Use Case

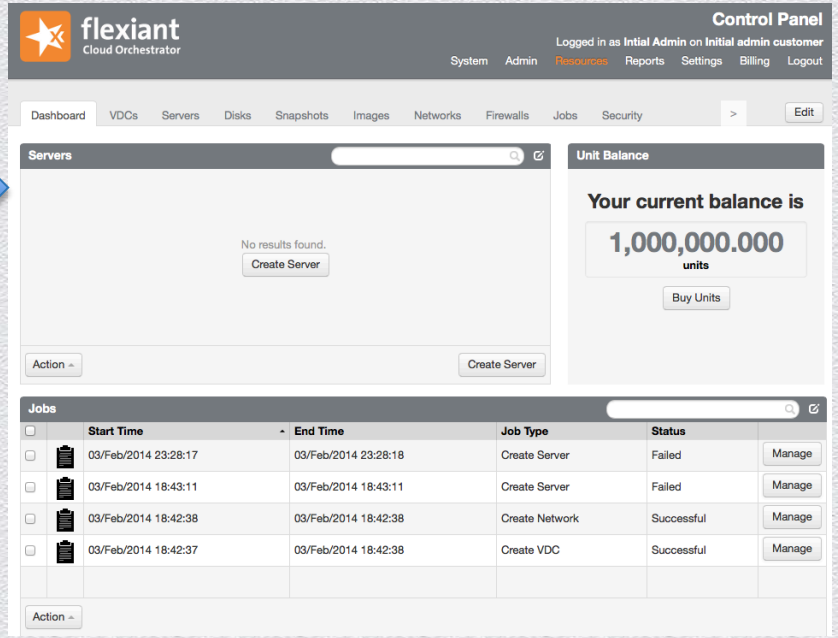


1. Developer navigates to internal self-service portal and requests a new virtual machine resource

Another Example Use Case

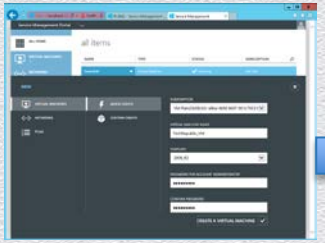


1. Developer navigates to internal self-service portal and requests a new virtual machine resource

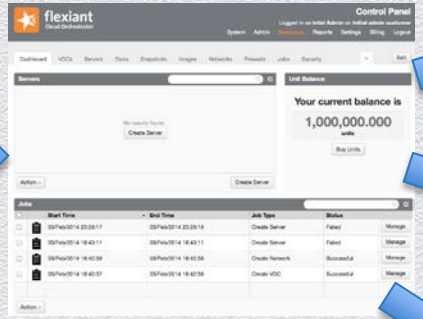


2. Request is sent to orchestration platform. Resource definition is verified, as is requester role and permissions.

Another Example Use Case

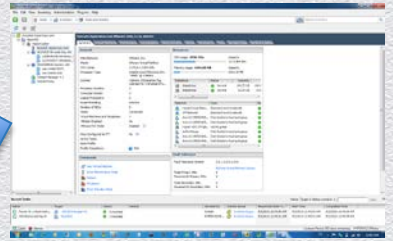


1. Developer navigates to internal self-service portal and requests a new virtual machine resource



2. Request is sent to orchestration platform. Resource definition is verified, as is requester role and permissions.

3a. A new VM is created.



3b. FW rules are opened.

3c. Storage is provisioned.

A final example use case…

A final example use case…

- The Orchestration platform becomes self aware…

A final example use case…

- The Orchestration platform becomes self aware…

Orchestration Tools

- Commercial:
  - CSC ServiceMesh Agility
  - Flexiant
  - IBM SmartCloud
  - HP Operations Orchestration
  - VMware vCenter Orchestrator
  - Oracle Nimbula
- Open-Source:
  - Abiquo
  - CloudStack
  - Eucalyptus
  - OpenStack
  - Puppet / Chef

Orchestration and Automation Risks

- Control of and interaction with automation platforms can be very risky
  - Poor development, scripting, resource design and instantiation
  - System availability issues or resource hijack/compromise
  - Malicious insiders or lack of “least privilege”
  - Vendor lock-in (architecture, language, etc.)
  - Poor authentication/credential management
  - Weak or non-existent integration with security products
- Configuration management and access control are critical

Key Risk 1: Modification of Critical Files

- All orchestration platforms have critical configuration files and/or files that include sensitive data
- Examples:
  - Puppet: /etc/puppetlabs/installer/database_info.install
  - Chef: knife.rb or JSON Data Bag files
  - Flexiant: /etc/extility/local.cfg
- Modifying these files could grant illicit access, change provisioning parameters, modify database or other users, etc.

Examples of critical platform files

```
[root@learn installer]# less database_info.install
q_backup_and_purge_old_database_directory=n
q_database_host=localhost
q_database_install=y
q_database_port=5432
q_database_root_password=[redacted]
q_database_root_user=pe-postgres
q_pe_database=y
q_puppet_enterpriseconsole_auth_database_name=console_auth
q_puppet_enterpriseconsole_auth_database_password=[redacted]
q_puppet_enterpriseconsole_auth_database_user=console_auth
q_puppet_enterpriseconsole_database_name=console
q_puppet_enterpriseconsole_database_password=[redacted]
q_puppet_enterpriseconsole_database_user=console
q_puppetdb_database_name=pe-puppetdb
q_puppetdb_database_password=[redacted]
q_puppetdb_database_user=pe-puppetdb
```

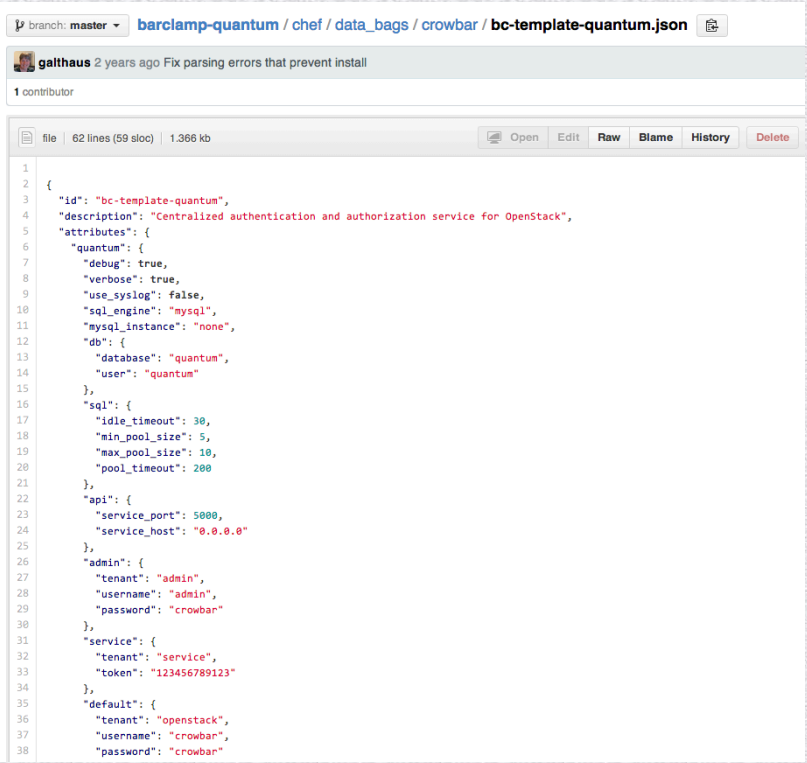
Puppet:
/etc/puppetlabs/installer/database_info.install

```
# CEPH support - set to 1 to support CEPH
CEPH=0
INITIAL_ADMIN_USER = dshackleford@voodoosec.com
INITIAL_ADMIN_PASSWORD = [redacted] CLEARTEXT
XVPADMIN_ADMIN_PASSWORD = [redacted]
HYPERVISOR = KVM
LICENCE_USER=85d081ea-6125-4825-899f-e292173[redacted]
LICENCE_PASSWORD=1fef8b6b-0430-4eb3-8e0d-505fc[redacted]
SSH_PUBLIC_KEY = ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDCCE4wuugeOwSEzkSFooxUyOR
VkrjfN3X82jXJwOetUsSIHUkOmXEUXJVjh3UexBhitt6C8AKFdFOYaG9sbgmV/Aa07FP5Tz21fJk3qi2
cvCOUc3fHoHA19IX+1XFrS2FbtzLIrl7F58MIuv7pNqDctC/iiMOK3uZLcoKX95yngYs4CXclX8VS464
9OwDdnE+FSHxO18A3ZRFLXLpTwqXMLqnQB9q8P9zjPZCiJxUKPS2QM8xsOa86Dgi1RcclPWdVBm3Fa4/
AnpiSKsaJ29EqSnjuPe6OKbY1rju5K146Yzc0T+yt8ukeCrDJIA72mNrUceNNLQRjbGsp+aOKHCYX ex
tility
```

Flexiant:
/etc/extility/local.cfg

Critical platform files…on the Internet

- Google query: chef data_bags filetype:json password -metadata



branch: master ▾ barclamp-quantum / chef / data_bags / crowbar / bc-template-quantum.json

galthaus 2 years ago Fix parsing errors that prevent install

1 contributor

file | 62 lines (59 sloc) | 1.366 kb — Open | Edit | Raw | Blame | History | Delete

```
{
  "id": "bc-template-quantum",
  "description": "Centralized authentication and authorization service for OpenStack",
  "attributes": {
    "quantum": {
      "debug": true,
      "verbose": true,
      "use_syslog": false,
      "sql_engine": "mysql",
      "mysql_instance": "none",
      "db": {
        "database": "quantum",
        "user": "quantum"
      },
      "sql": {
        "idle_timeout": 30,
        "min_pool_size": 5,
        "max_pool_size": 10,
        "pool_timeout": 200
      },
      "api": {
        "service_port": 5000,
        "service_host": "0.0.0.0"
      },
      "admin": {
        "tenant": "admin",
        "username": "admin",
        "password": "crowbar"
      },
      "service": {
        "tenant": "service",
        "token": "123456789123"
      },
      "default": {
        "tenant": "openstack",
        "username": "crowbar",
        "password": "crowbar"
```

Key Risk 2: Modification to Work Flows

- Orchestration platforms all function with defined “runbooks”
  - These include resource definitions, configuration options, scheduling and policy preferences, credentials/roles, and more
- Most work flow steps involve:
  - Integration with a cloud management platform (OpenStack, vSphere)
  - API calls to network devices, applications, or middleware
  - Pre-authenticated remote command execution
- Changing any of these could dramatically impact nodes or resources

Example of workflow modification:

- A workflow is defined that:
  - Provisions a new application VM
  - Opens numerous Check Point firewall rules to facilitate traffic to/from the new VM
  - Performs periodic health/security checks of the VM and app configuration
- An attacker is able to modify the workflow definition:
  - Adds malicious files to the VM configuration
  - Opens a new firewall port for data exfiltration and C2
  - …for ALL NEW INSTANCES.

Key Risk 3: Changes to Roles and Privileges

- Access **to** orchestration platforms needs to be carefully controlled
- In addition, defined roles and privileges should be designed and implemented with extreme caution
  - Too many privileges could easily allow insider attacks to proliferate
- Example: Puppet Console system has a simple Web username/password field combination, and is exposed to the entire management network
  - Brute force password guessing…and no lockout.
- Example 2: A business unit IT operator role is set up improperly to allow unfettered API access to network nodes and all hypervisor instances
  - The user accidentally crashes hypervisors with API calls…or worse.

Key Risk 4: Availability Sabotage

- Availability of cloud nodes, middleware, applications, and even network devices could be severely impacted if:
  - API access is changed or corrupted
  - Credentials are compromised/changed/deleted
  - Shutdown commands are issued
  - Network access paths are changed/degraded
- The orchestration platform **itself** is a single point of failure
  - Many implementations I have seen have ZERO redundancy

Threat Models

Attacking Orchestration

- In a cloud environment, the orchestration layer is a potential weak point with much to gain for attackers
- An attacker or malicious insider that gains control over orchestration could:
  - Modify the SAN allocation for VMs
  - Modify VM templates
  - Modify user/group roles
  - Impact availability of orchestration++
- These are just starting points!

Threat Model 1: SAN Allocation

- Most cloud implementations rely heavily on large-scale storage infrastructure
- Orchestration workflows incorporate automated disk provisioning for workloads
- Modification of the storage workflow parameters for disk allocation could easily lead to a SAN becoming full or over allocated
- Deliberate or accidental configuration changes could easily lead to this threat becoming realized
- Impact: Availability and/or loss/corruption of data

Threat Model 2: VM Template Modification

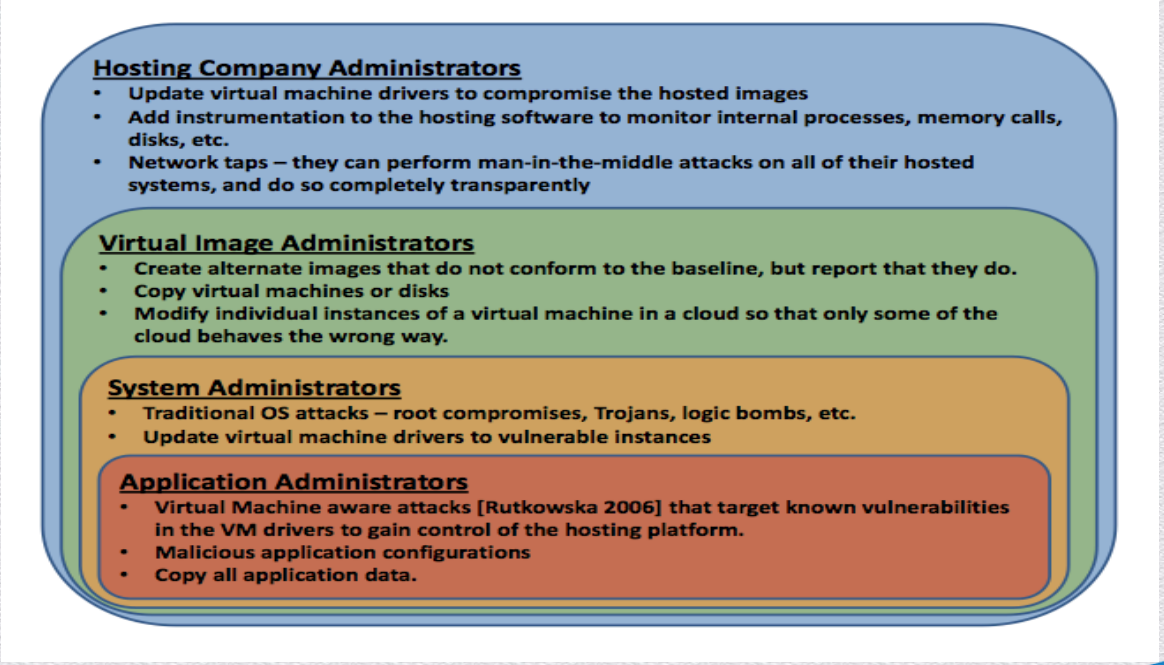
- A very common use case for orchestration is deployment of new VM workloads from templates
- Templates may exist on the SAN and hypervisor platforms
  - Orchestration resource templates will modify as needed
- Modification could:
  - Add malicious programs into a template
  - Open new ports / start new services
  - Disable security features or programs

Threat Model 3: Role Modification

- Modifying orchestration roles could easily lead to:
  - Undetected backdoor/privileged access by “low privilege” users
  - Accidental configuration changes/mishaps
  - Escalation of privilege scenarios
  - “Shadow IT” or other changes
- Role definition and privileged user monitoring is critical
- Many orchestration platforms don’t natively integrate with Identity Management systems

CERT's Cloud Insider Guide

- CERT breaks down the insiders and risks in a 2012 paper
- Lists roles and likely attack vectors
- Where's the Orchestration Admin?

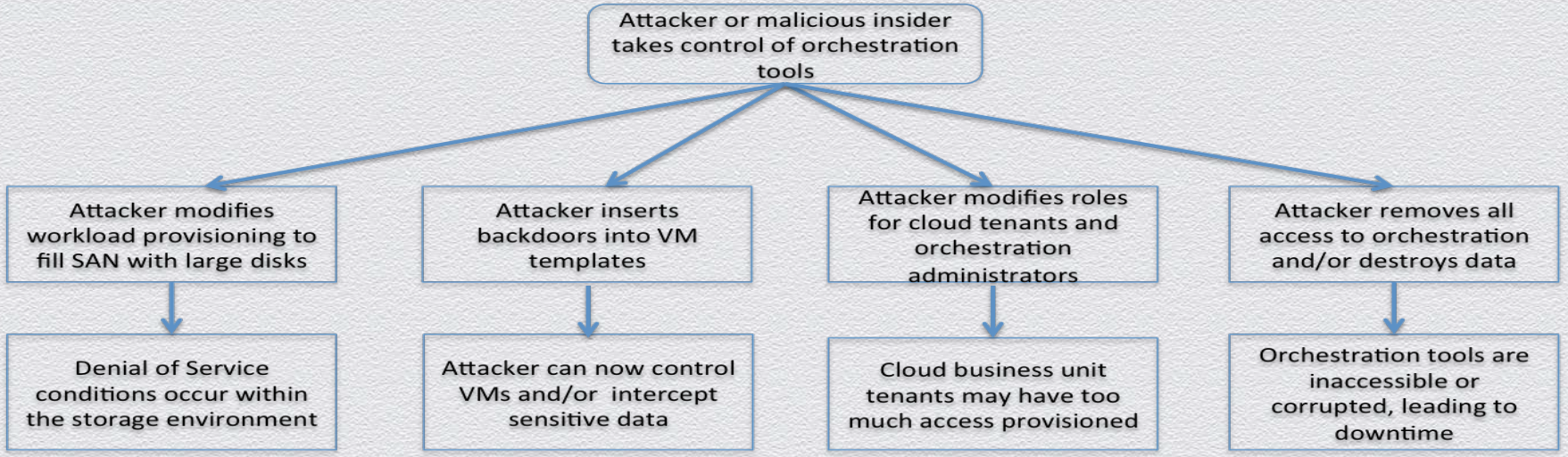


Full paper available at www.cert.org/archive/pdf/CERT_cloud_insiders.pdf

Threat Model 4: Availability Impact

- Any modification to the orchestration platform itself, or various settings, could have major availability impact:
  - Locking out admin accounts
  - Changing resource definitions
  - Modifying workflow steps or parameters
  - Changing/closing local ports for communication
  - Starting/stopping orchestration services
- The orchestration platform could be a single point of failure, too.

Orchestration Attack Tree



Remediation Options and Tools

Key Areas Of Focus

- Orchestration Platforms
  - Often multi-tiered
  - Focus on code/data repos, master servers, and client configs
- Databases
  - Usernames and passwords, config files containing sensitive data
- Automation platforms
  - Separate repos or “workstations” (Chef) used for configuration and resource management

Key Areas Of Focus

- Operations teams
  - Social engineering attacks targeting orchestration and automation teams - more focus on security awareness
- API calls and logging
  - Local access and calls of APIs
  - Remote API logging at nodes and infrastructure
- “Failsafes” – affected platforms and systems
  - “Deny All” stance and “triggers”/”tipping point” fallbacks

What To Do Now

A Checklist for Security Teams

- **Review security options available within orchestration platforms**
  - Most offer role-based access
  - Privilege creation and assignment is often limited, though
  - Key- and cert-based authentication
  - Look for integration with Privileged User Management and IAM tools
  - Assess depth and breadth of API integration
  - Look for logging and event generation

A Checklist for Security Teams (cont.)

- Review security options available within orchestration platforms
- **Evaluate whether file integrity monitoring tools can run on the orchestration management platforms**
  - Many attacks are focused on modification of critical files or configuration parameters
  - FIM is likely “unsupported”, especially with “appliance” form factors

A Checklist for Security Teams (cont.)

- Review security options available within orchestration platforms
- Evaluate whether file integrity monitoring tools can run on the orchestration management platforms
- **Consider dual-factor authentication to the orchestration servers, if possible**
  - May help to mitigate attack vectors coming from compromised Ops workstations
  - Can also require access from a “jump box” for control and audit

A Checklist for Security Teams (cont.)

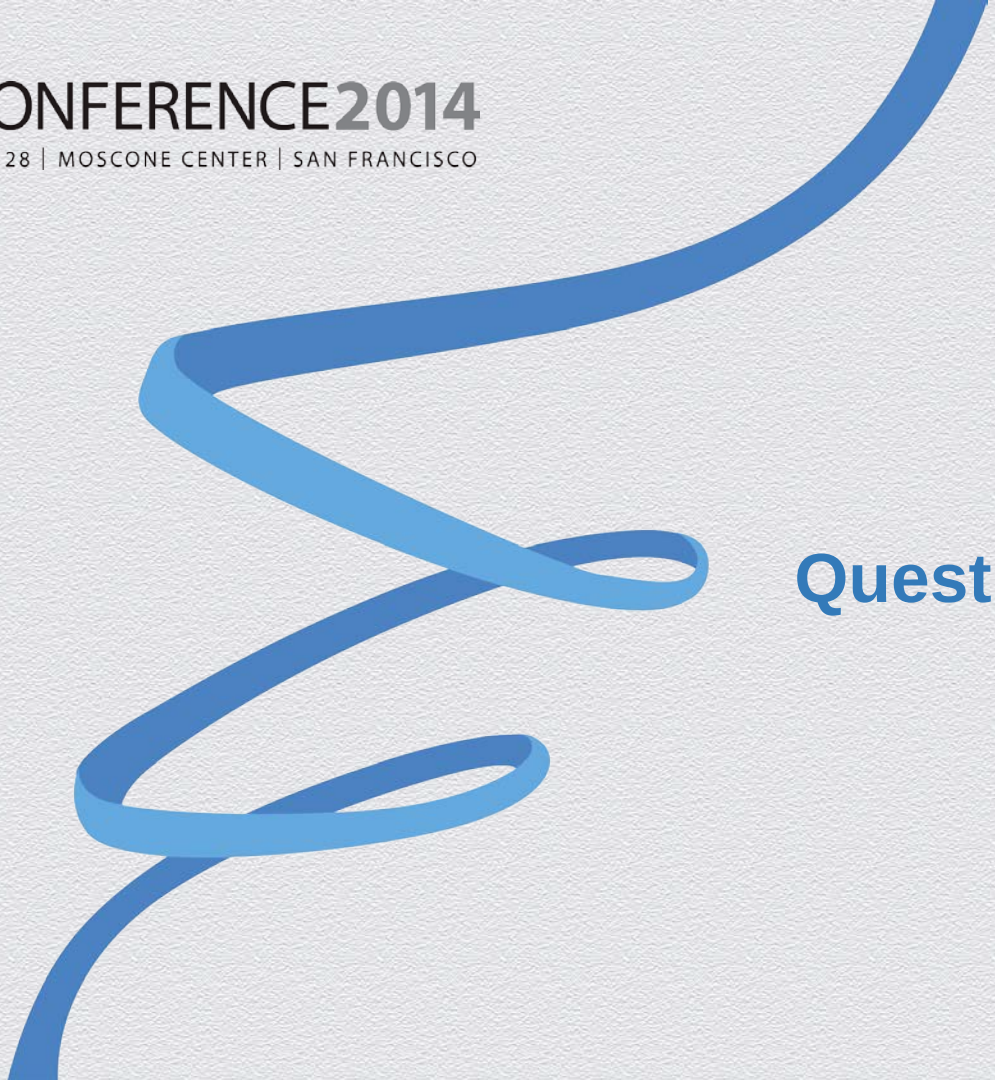
- Review security options available within orchestration platforms
- Evaluate whether file integrity monitoring tools can run on the orchestration management platforms
- Consider dual-factor authentication to the orchestration servers, if possible
- **Integrate orchestration logs and events into your monitoring/SIEM strategy**
  - Develop behavioral profiles for admin-level tasks and operations

A Checklist for Security Teams (cont.)

- Review security options available within orchestration platforms
- Evaluate whether file integrity monitoring tools can run on the orchestration management platforms
- Consider dual-factor authentication to the orchestration servers, if possible
- Integrate orchestration logs and events into your monitoring/SIEM strategy
- **Heighten security awareness for Orchestration teams!**

Conclusion

- Orchestration and automation platforms have the potential to streamline cloud operations
  - Properly implemented, can improve effectiveness & efficiency
- Many orchestration platforms are lacking in security, however
- Many security teams also aren't aware of the risks these systems pose!
  - Perform a security/risk assessment of orchestration platforms and governance/usage of them
- If well-managed, these systems can **improve** security, too!



Questions?