Chef Automate

Azure Partner Quickstart Template

Hands-On Lab User Manual

Table Of Contents

1. About this test drive

The Continuous Delivery & Compliance with Chef Automate Azure Partner Quickstart Template launches a DevOps stack that provides an automated provisioning, configuration and integration of Chef Automate that is needed for continuous delivery & compliance of applications, as well as infrastructure code. This is intended as a pilot solution and is not production ready.

Chef Automate gives you a full-stack continuous delivery pipeline, automated testing for compliance and security, and visibility into everything that's happening along the way. It builds on Chef for infrastructure automation, InSpec for compliance automation, and Habitat for application automation. You can transform your company into a highly collaborative, software driven organization with Chef Automate as the engine.

Automate enables you to scan your entire infrastructure for security risks and compliance issues, get reports on risks and issues classified by severity and impact levels, and build automated testing into your deployment pipelines. Chef Compliance includes pre-built profiles that scan for CIS benchmarks to help you get started quickly.

Solution Summary

The Chef Automate Azure Partner Quickstart solution is a 30 minute long walkthrough of using Automate's Compliance profiles to rapidly scan a series of CentOS 6.8 machines and remediate them to OS-hardening specifications.

The Quickstart template consists of a Chef Automate cluster, several infrastructure nodes and a user's Workstation. The Nodes are managed by Automate, and their policy is configured to scan the node and report scan results back to Automate.

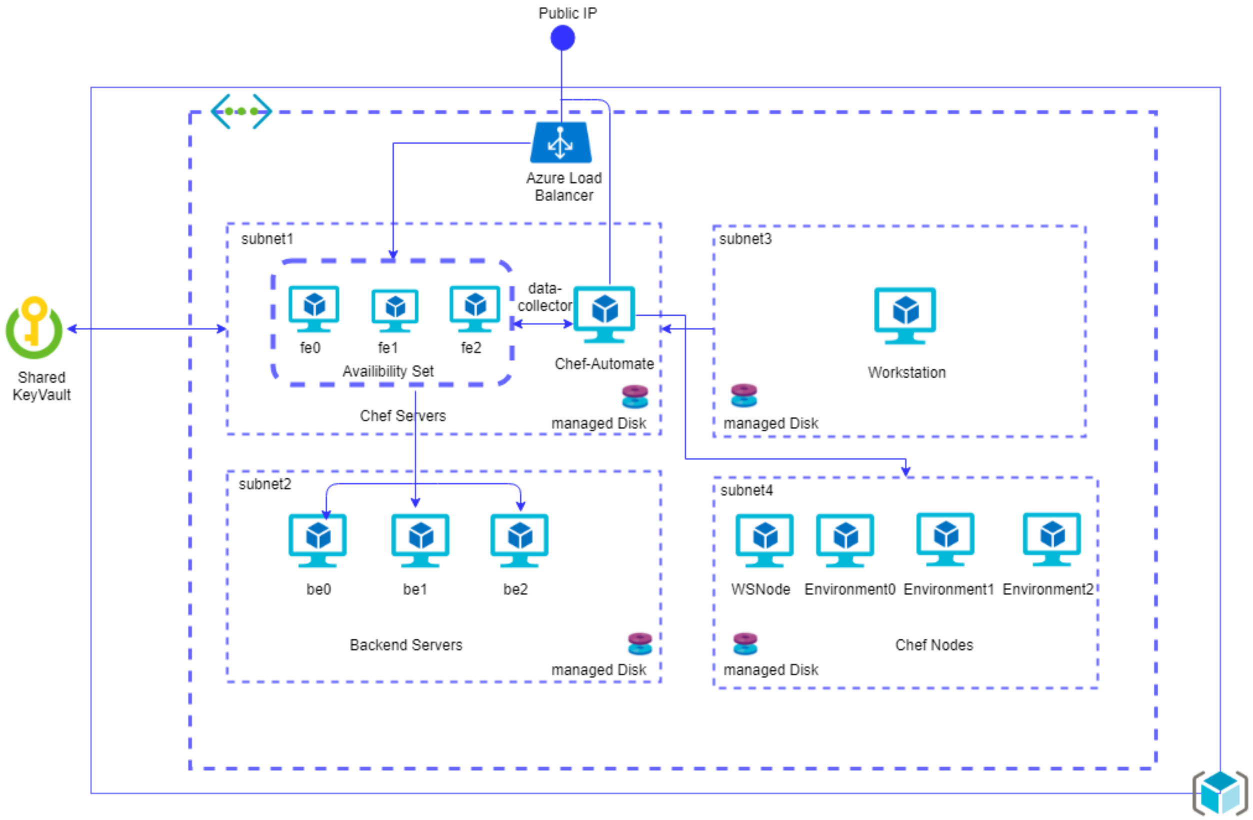
The user will go through the process of triggering a Chef execution on managed machines from the Workstation using knife-ssh (the remote execution over SSH feature) in the CLI (command-line interface). This simulates the automated periodic Chef execution on nodes in a convergent infrastructure.

Once nodes converge (run chef), they report their status to Automate. The user will examine Automate's visibility dashboards to assess the health and compliance of the node.

The user will then add the OS-hardening cookbook (a configuration policy) to the node's run_list (a list of configuration policies that a node has to process).

Finally, the user will again trigger a Chef execution on the node. The user will observe the OS-hardening policies being applied. Returning to the Automate dashboard, the user can review the improved state of the node's compliance.

2. Architecture



3. What will you learn?

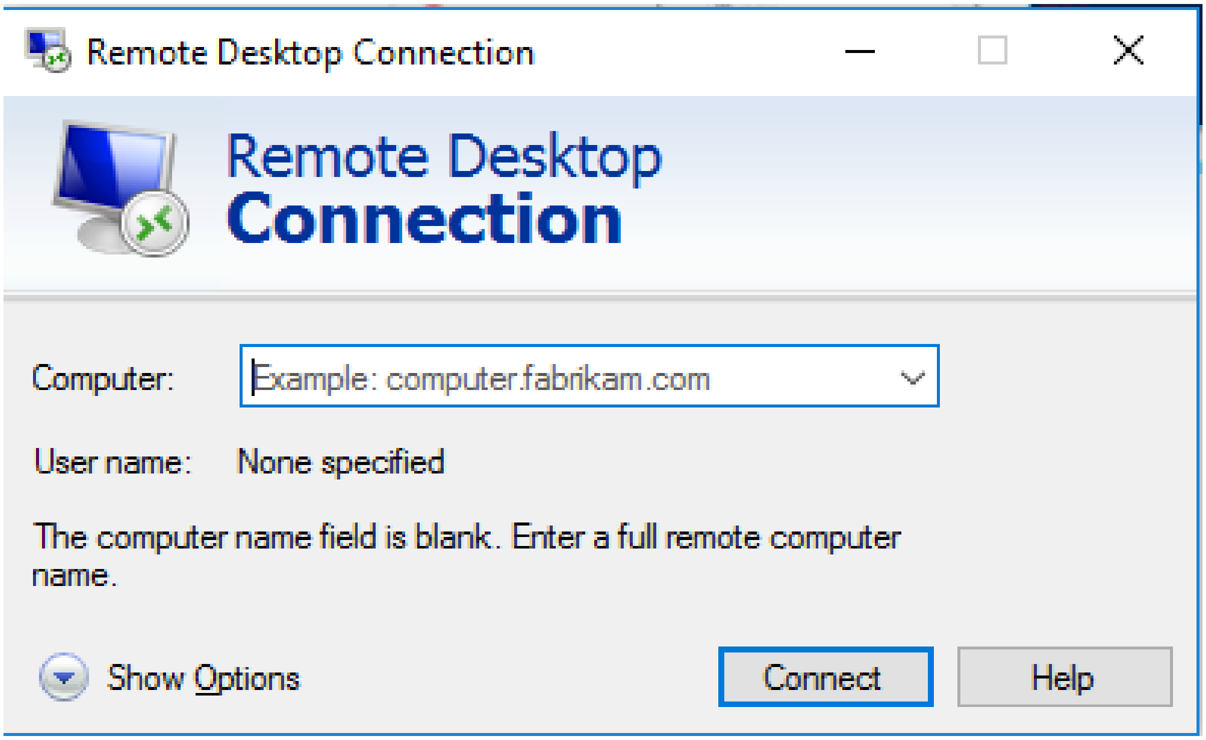
The goal of this Chef Automate solution stack is to provide a continuous delivery & application compliance experience. By walking through the lab in this Launch & Learn session, attendees will learn how this Quickstart template can be used to make the nodes compliant build and deploy a powerful DevOps solution using Chef products.

4. Lab Steps

4.1. Connect to the workstation

Use your preferred Microsoft Remote Desktop client to connect to the Workstation with the Fully Qualified Domain Name (FQDN) and credentials provided in your email.

You can login to the workstation using FQDN Username and Password of that machine. Launch the windows RDP client and enter the FQDN as shown below and click on connect.



Remote Desktop Connection

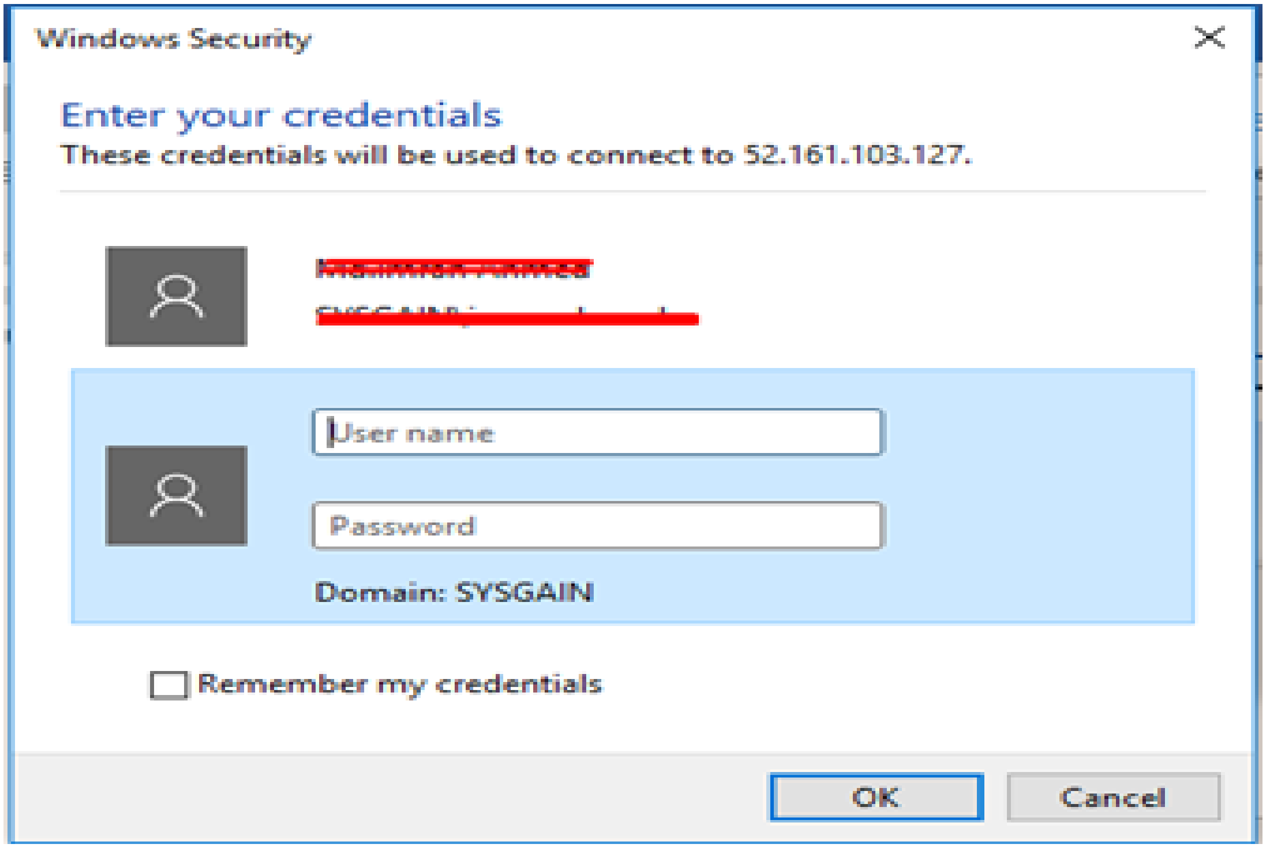
Remote Desktop Connection

Computer: Example: computer.fabrikam.com

User name: None specified

The computer name field is blank. Enter a full remote computer name.

Enter the credentials of the windows Workstation node



Windows Security

Enter your credentials

These credentials will be used to connect to 52.161.103.127.

[redacted]

User name

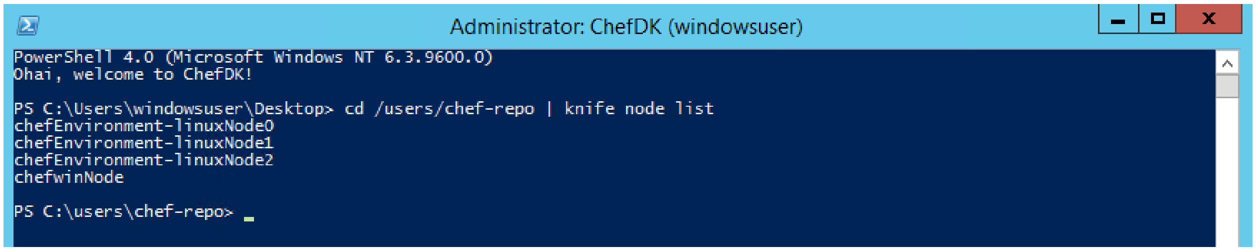
Password

Domain: SYSGAIN

☐ Remember my credentials

Open PowerShell and execute the below command to know the node list in Chef Automate:

cd /users/chef-repo | knife node list



```
PowerShell 4.0 (Microsoft Windows NT 6.3.9600.0)
Ohai, welcome to ChefDK!

PS C:\Users\windowsuser\Desktop> cd /users/chef-repo | knife node list
chefEnvironment-linuxNode0
chefEnvironment-linuxNode1
chefEnvironment-linuxNode2
chefwinNode

PS C:\users\chef-repo> _
```

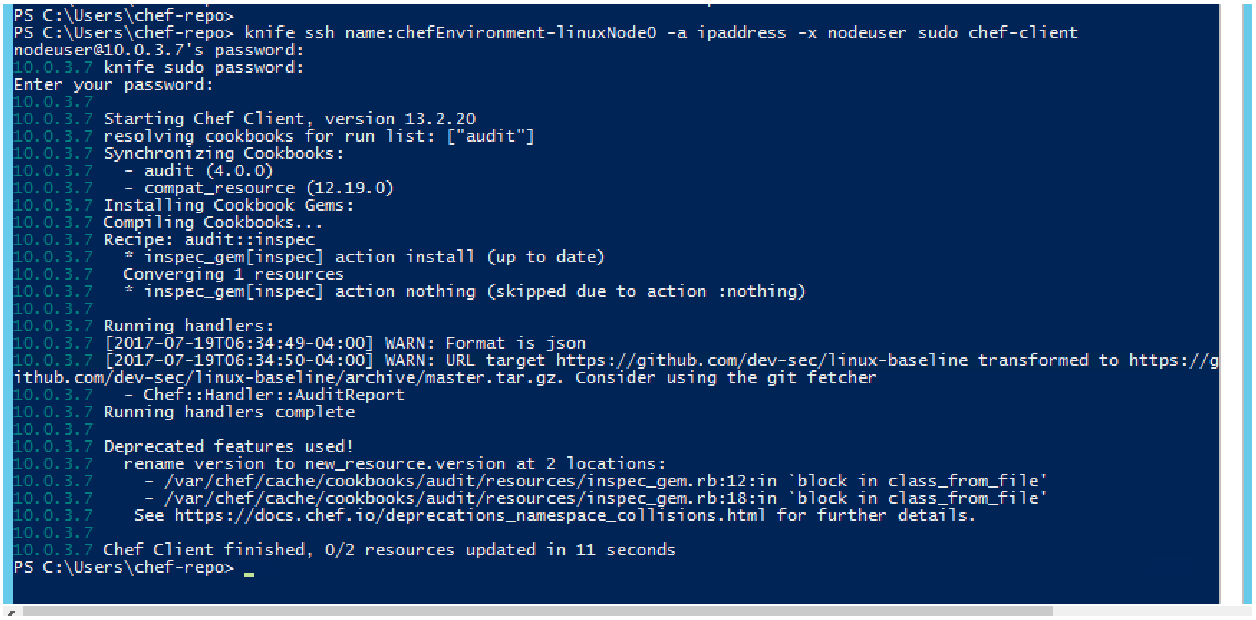
4.2. Remotely trigger Chef execution on one or many nodes

Run this command to execute the **chef-client** remotely from workstation:

knife ssh name:chefEnvironment-linuxNode0 -a ipaddress -x nodeuser sudo chef-client

Enter **Password@1234** when prompted for the password.

This establishes an ssh connection to the node named chefEnvironment-linuxNode0 over its IP address as the user nodeuser to run the command sudo chef-client.



```
PS C:\Users\chef-repo>
PS C:\Users\chef-repo> knife ssh name:chefEnvironment-linuxNode0 -a ipaddress -x nodeuser sudo chef-client
nodeuser@10.0.3.7's password:
10.0.3.7 knife sudo password:
Enter your password:
10.0.3.7
10.0.3.7 Starting Chef Client, version 13.2.20
10.0.3.7 resolving cookbooks for run list: ["audit"]
10.0.3.7 Synchronizing Cookbooks:
10.0.3.7   - audit (4.0.0)
10.0.3.7   - compat_resource (12.19.0)
10.0.3.7 Installing Cookbook Gems:
10.0.3.7 Compiling Cookbooks...
10.0.3.7 Recipe: audit::inspec
10.0.3.7   * inspec_gem[inspec] action install (up to date)
10.0.3.7   Converging 1 resources
10.0.3.7   * inspec_gem[inspec] action nothing (skipped due to action :nothing)
10.0.3.7
10.0.3.7 Running handlers:
10.0.3.7 [2017-07-19T06:34:49-04:00] WARN: Format is json
10.0.3.7 [2017-07-19T06:34:50-04:00] WARN: URL target https://github.com/dev-sec/linux-baseline transformed to https://g
ithub.com/dev-sec/linux-baseline/archive/master.tar.gz. Consider using the git fetcher
10.0.3.7   - Chef::Handler::AuditReport
10.0.3.7 Running handlers complete
10.0.3.7
10.0.3.7 Deprecated features used!
10.0.3.7   rename version to new_resource.version at 2 locations:
10.0.3.7     - /var/chef/cache/cookbooks/audit/resources/inspec_gem.rb:12:in `block in class_from_file'
10.0.3.7     - /var/chef/cache/cookbooks/audit/resources/inspec_gem.rb:18:in `block in class_from_file'
10.0.3.7    See https://docs.chef.io/deprecations_namespace_collisions.html for further details.
10.0.3.7
10.0.3.7 Chef Client finished, 0/2 resources updated in 11 seconds
PS C:\Users\chef-repo> _
```

Run the below commands for both **linuxNode1** and **linuxNode2**:

knife ssh name:chefEnvironment-linuxNode1 -a ipaddress -x nodeuser sudo chef-client

knife ssh name:chefEnvironment-linuxNode2 -a ipaddress -x nodeuser sudo chef-client

This simulates the automated periodic Chef execution on nodes in a convergent infrastructure.

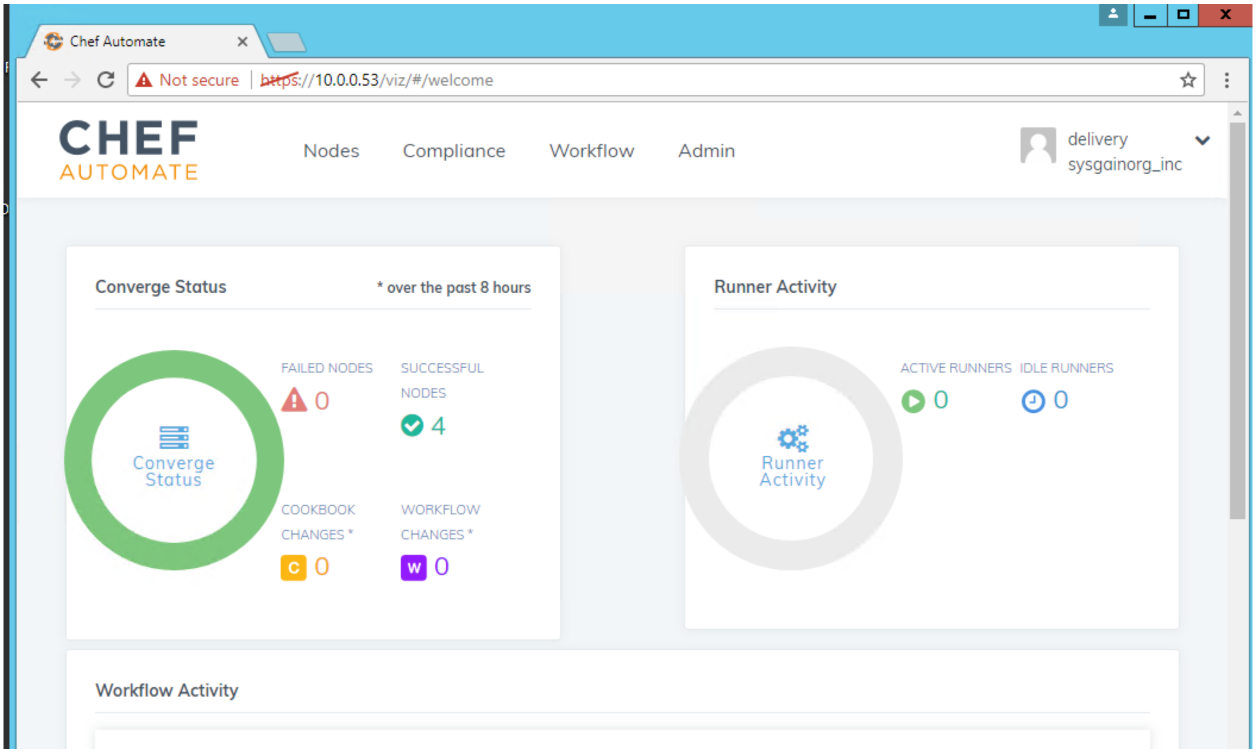
4.3. Log in to Automate

Open the web browser and enter the below URL in address bar to access the Chef Automate web console to check the Nodes compliance status:

https://10.0.0.53

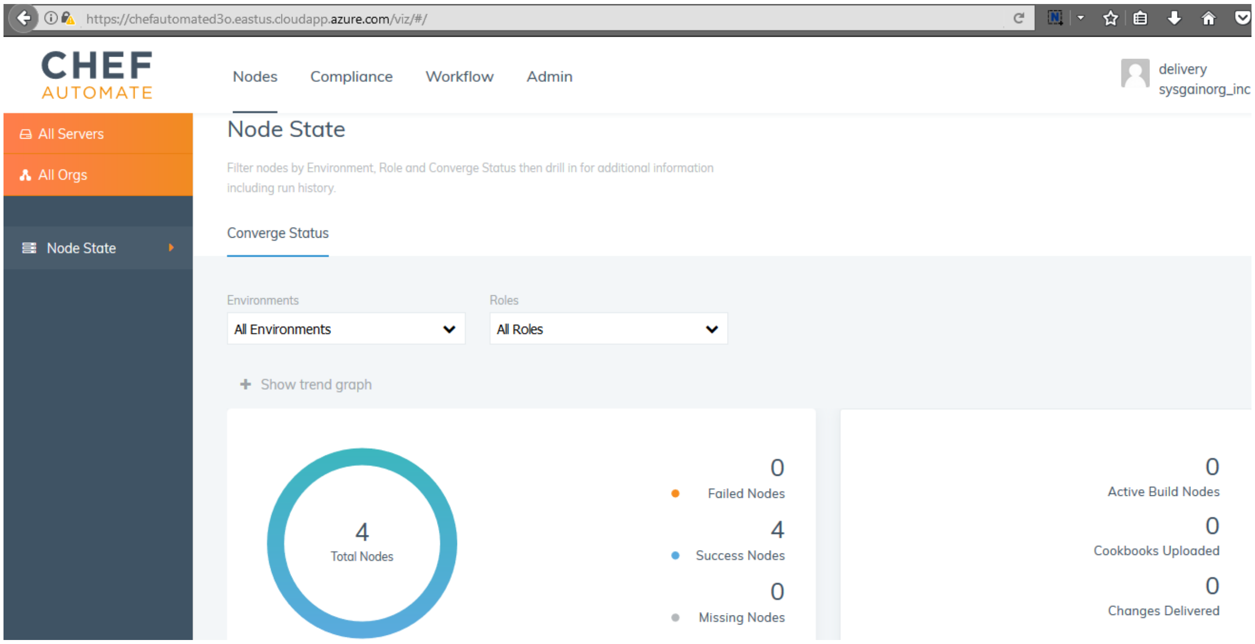
Login to the Chef Automate web console using the credentials provided via email.

After successful login, you can view the Chef-Automate dashboard.



4.4. Review the nodes compliance state in CHEF AUTOMATE

Click the Nodes tab to examine the state of your infrastructure fleet.



https://chefautomated3o.eastus.cloudapp.azure.com/viz/#/

Node State

Filter nodes by Environment, Role and Converge Status then drill in for additional information including run history.

Environments: All Environments

Roles: All Roles

4 Total Nodes

0 ● Failed Nodes

4 ● Success Nodes

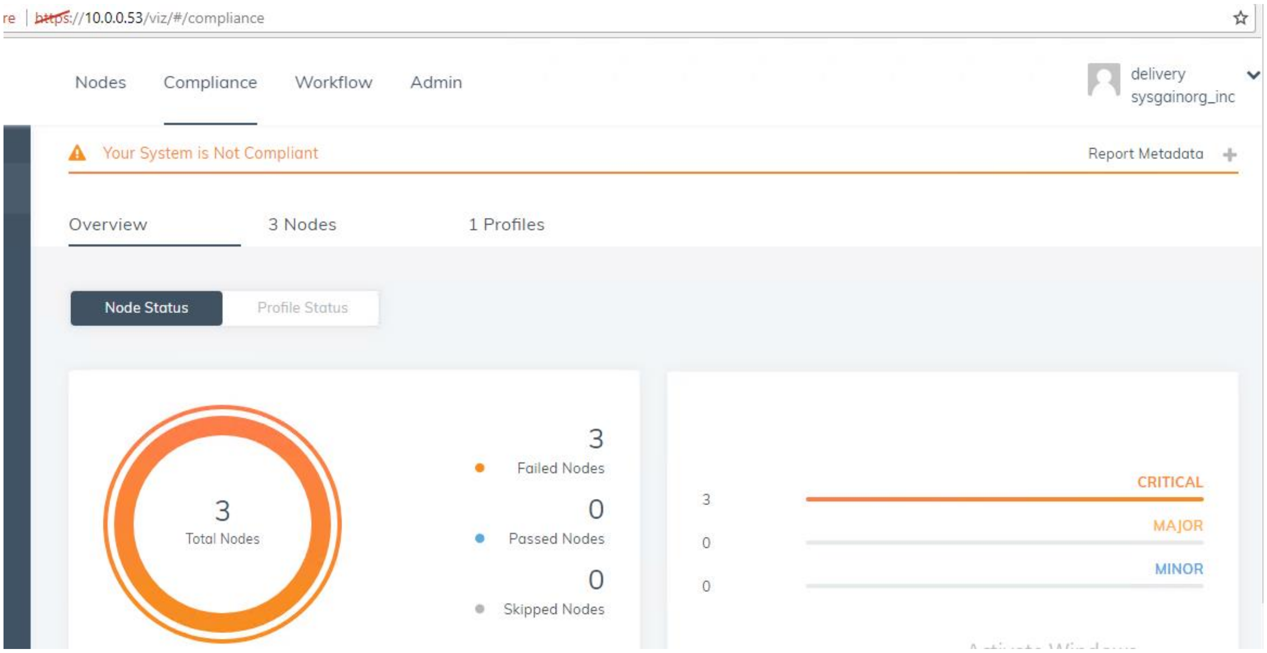
0 ● Missing Nodes

0 Active Build Nodes

0 Cookbooks Uploaded

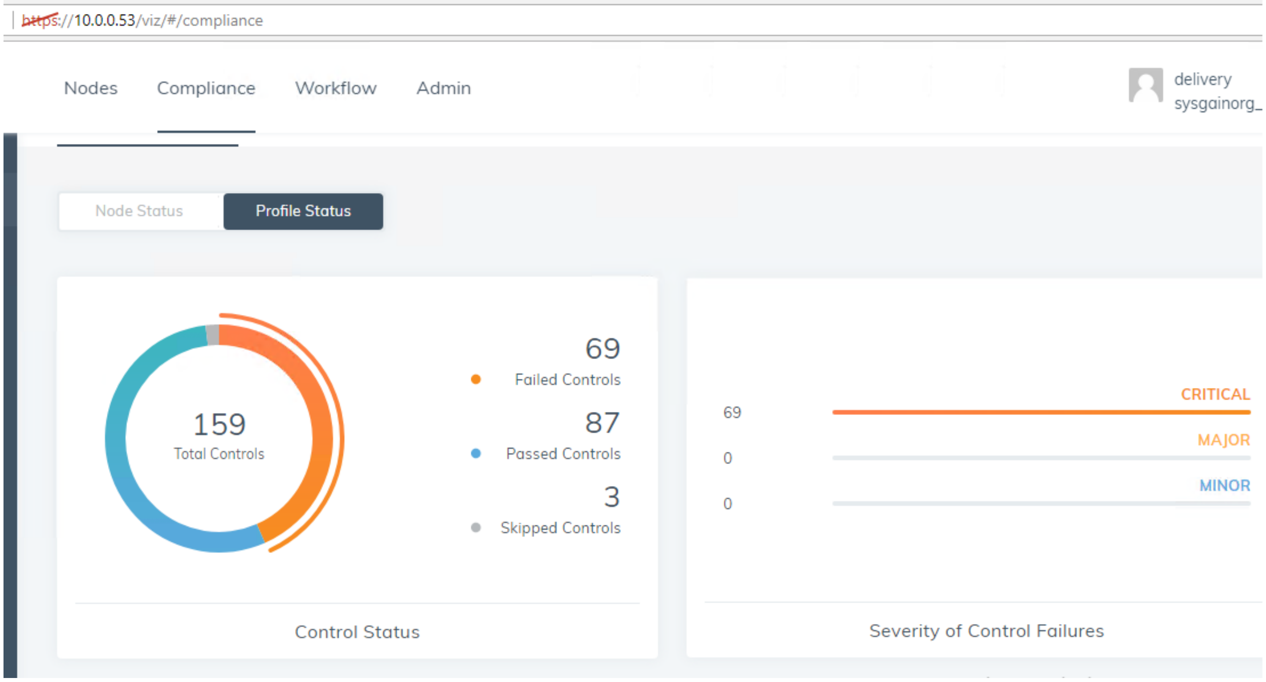
0 Changes Delivered

Now click the **Compliance tab**, here you can see the failed nodes.

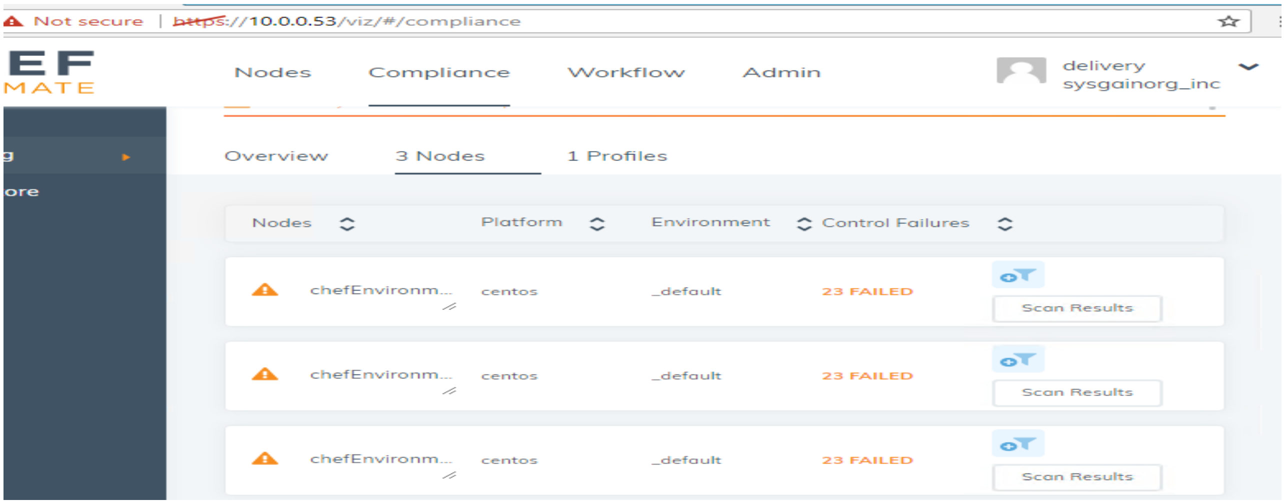


Your System is Not Compliant

Click on **Profile Status** to view the number of Critical failures on your nodes.

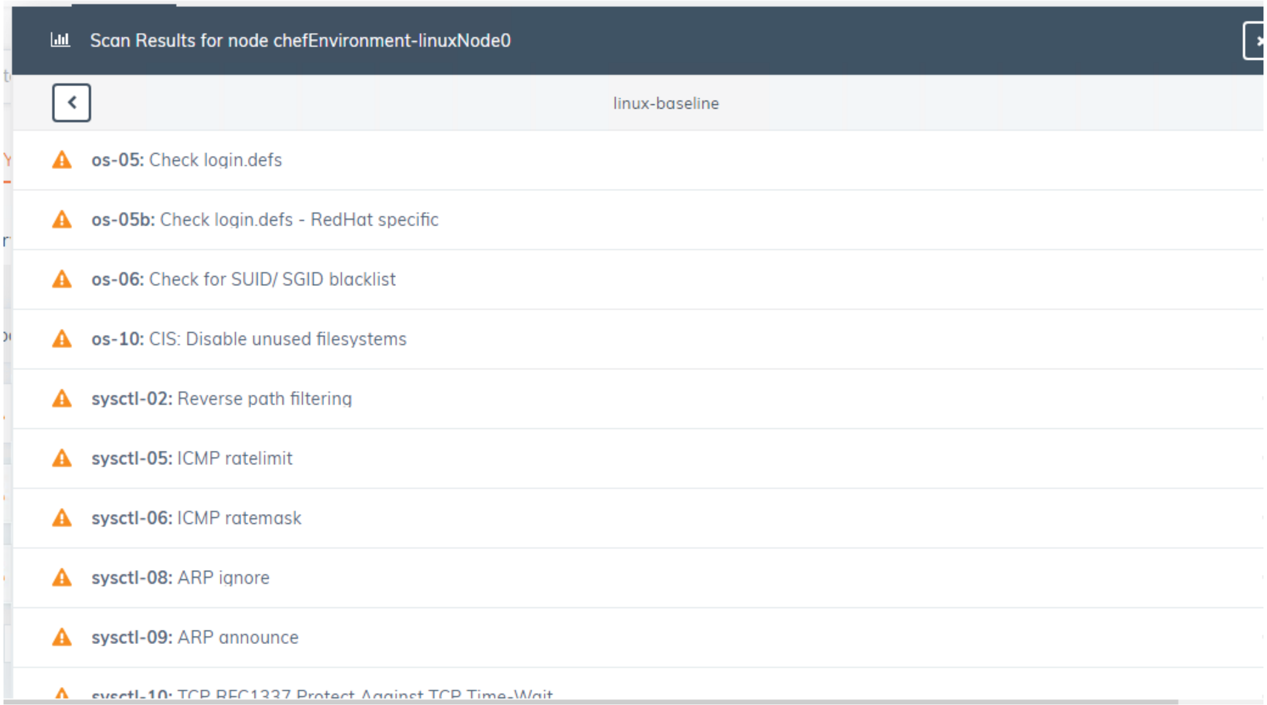


Under compliance tab Click on **Nodes** to view individual nodes in detail.



Nodes	Platform	Environment	Control Failures	
chefEnvironm...	centos	_default	23 FAILED	Scan Results
chefEnvironm...	centos	_default	23 FAILED	Scan Results
chefEnvironm...	centos	_default	23 FAILED	Scan Results

Click on the **Scan Results>linux-baseline** to examine the specific audit controls failed.



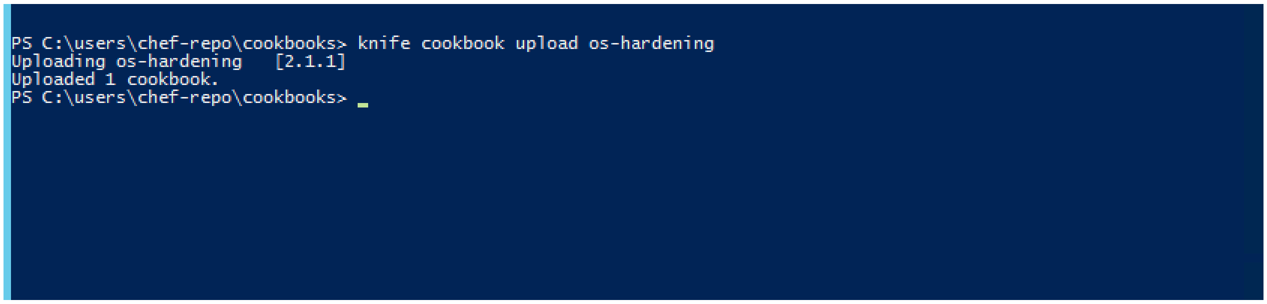
Scan Results for node chefEnvironment-linuxNode0

- **os-05:** Check login.defs
- **os-05b:** Check login.defs - RedHat specific
- **os-06:** Check for SUID/ SGID blacklist
- **os-10:** CIS: Disable unused filesystems
- **sysctl-02:** Reverse path filtering
- **sysctl-05:** ICMP ratelimit
- **sysctl-06:** ICMP ratemask
- **sysctl-08:** ARP ignore
- **sysctl-09:** ARP announce
- **sysctl-10:** TCP RFC1337 Protect Against TCP Time-Wait

4.5. Upload the os-hardening cookbook into the Chef Server.

Go back to your command prompt and issue a Chef API command to upload the os-hardening cookbook into the chef server

knife cookbook upload os-hardening



```
PS C:\users\chef-repo\cookbooks> knife cookbook upload os-hardening
Uploading os-hardening   [2.1.1]
Uploaded 1 cookbook.
PS C:\users\chef-repo\cookbooks> _
```

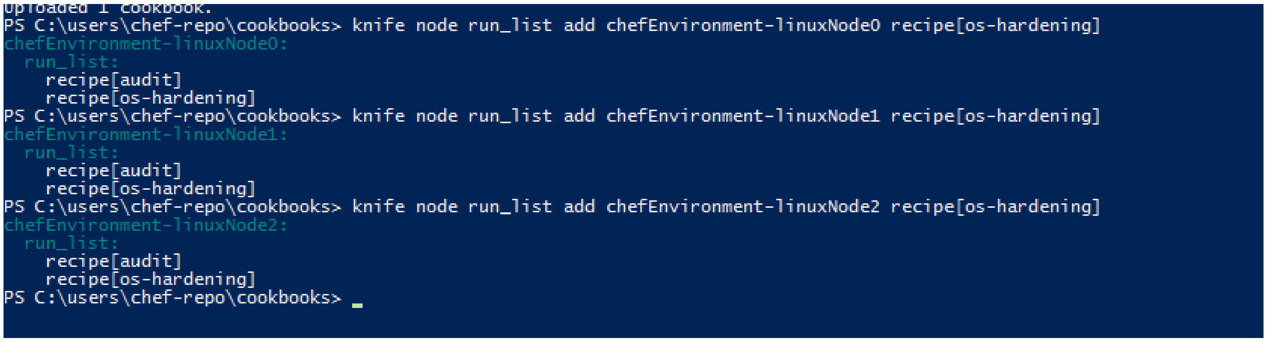
4.6. Update the node's run-list to run the os-hardening cookbook

In the command prompt issue a Chef API command to add the OS-hardening recipe to the node's run list so that we can apply this cook book in the following steps to resolve the compliance issues:

knife node run list add chefEnvironment-linuxNode0 recipe[os-hardening]

The command should confirm to you that the recipe is now in the list.

Run the command for both **linuxNode1** and **linuxNode2.**



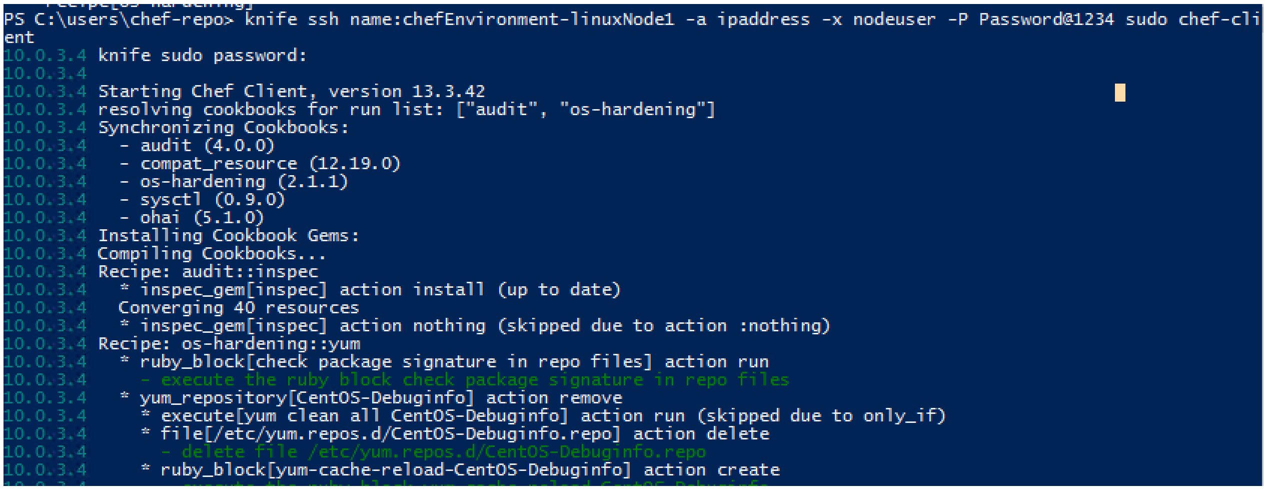
```
Uploaded 1 cookbook.
PS C:\users\chef-repo\cookbooks> knife node run_list add chefEnvironment-linuxNode0 recipe[os-hardening]
chefEnvironment-linuxNode0:
  run_list:
    recipe[audit]
    recipe[os-hardening]
PS C:\users\chef-repo\cookbooks> knife node run_list add chefEnvironment-linuxNode1 recipe[os-hardening]
chefEnvironment-linuxNode1:
  run_list:
    recipe[audit]
    recipe[os-hardening]
PS C:\users\chef-repo\cookbooks> knife node run_list add chefEnvironment-linuxNode2 recipe[os-hardening]
chefEnvironment-linuxNode2:
  run_list:
    recipe[audit]
    recipe[os-hardening]
PS C:\users\chef-repo\cookbooks> _
```

4.7. Remotely trigger Chef execution again

Now that we modified the configuration policy for our node, let's simulate what a convergent node would do automatically in production – execute Chef:

knife ssh name:chefEnvironment-linuxNode0 -a ipaddress -x nodeuser sudo chef-client

Enter **Password@1234** when prompted for the password.



```
PS C:\users\chef-repo> knife ssh name:chefEnvironment-linuxNode1 -a ipaddress -x nodeuser -P Password@1234 sudo chef-cli
ent
10.0.3.4 knife sudo password:
10.0.3.4
10.0.3.4 Starting Chef Client, version 13.3.42
10.0.3.4 resolving cookbooks for run list: ["audit", "os-hardening"]
10.0.3.4 Synchronizing Cookbooks:
10.0.3.4   - audit (4.0.0)
10.0.3.4   - compat_resource (12.19.0)
10.0.3.4   - os-hardening (2.1.1)
10.0.3.4   - sysctl (0.9.0)
10.0.3.4   - ohai (5.1.0)
10.0.3.4 Installing Cookbook Gems:
10.0.3.4 Compiling Cookbooks...
10.0.3.4 Recipe: audit::inspec
10.0.3.4   * inspec_gem[inspec] action install (up to date)
10.0.3.4   Converging 40 resources
10.0.3.4   * inspec_gem[inspec] action nothing (skipped due to action :nothing)
10.0.3.4 Recipe: os-hardening::yum
10.0.3.4   * ruby_block[check package signature in repo files] action run
10.0.3.4     - execute the ruby block check package signature in repo files
10.0.3.4   * yum_repository[CentOS-Debuginfo] action remove
10.0.3.4     * execute[yum clean all CentOS-Debuginfo] action run (skipped due to only_if)
10.0.3.4     * file[/etc/yum.repos.d/CentOS-Debuginfo.repo] action delete
10.0.3.4       - delete file /etc/yum.repos.d/CentOS-Debuginfo.repo
10.0.3.4     * ruby_block[yum-cache-reload-CentOS-Debuginfo] action create
```

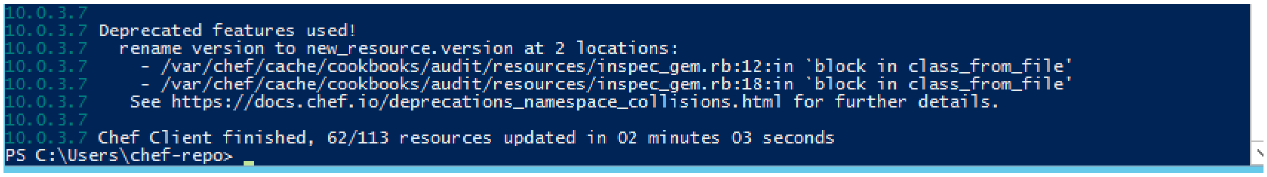
Run the same command for the **linuxNode1** and **linuxNode2:**

knife ssh name:chefEnvironment-linuxNode1 -a ipaddress -x nodeuser sudo chef-client

knife ssh name:chefEnvironment-linuxNode2 -a ipaddress -x nodeuser sudo chef-client

Watch as the Chef execution hardens the node to OS-hardening specifications. It will take a few minutes.

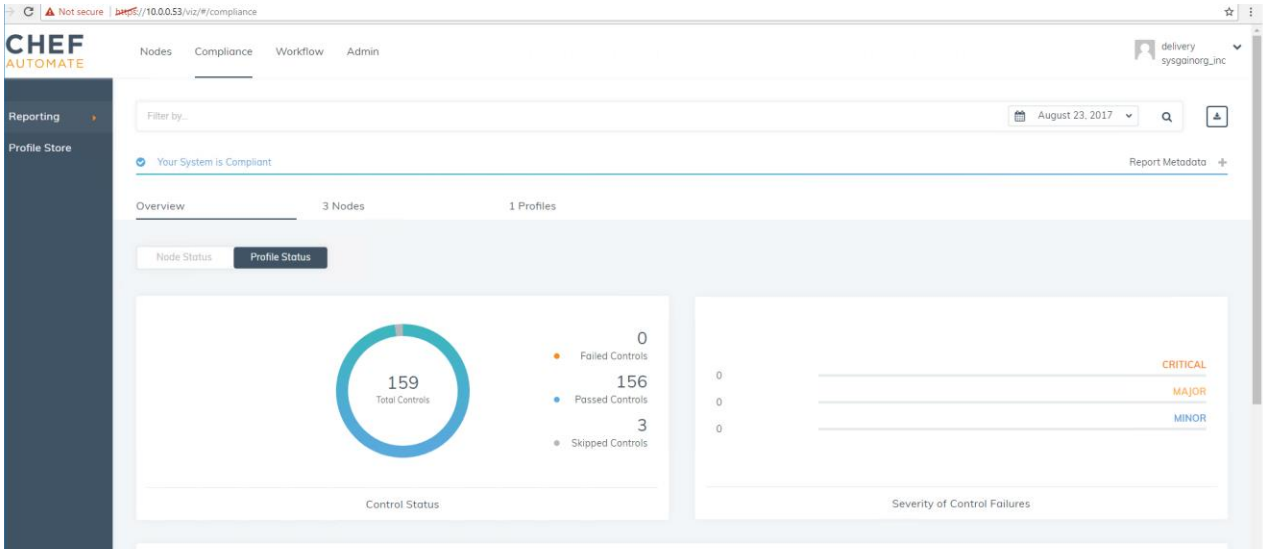
After completion of command execution, you will see the number of resources updated.



```
10.0.3.7
10.0.3.7 Deprecated features used!
10.0.3.7   rename version to new_resource.version at 2 locations:
10.0.3.7     - /var/chef/cache/cookbooks/audit/resources/inspec_gem.rb:12:in `block in class_from_file'
10.0.3.7     - /var/chef/cache/cookbooks/audit/resources/inspec_gem.rb:18:in `block in class_from_file'
10.0.3.7    See https://docs.chef.io/deprecations_namespace_collisions.html for further details.
10.0.3.7
10.0.3.7 Chef Client finished, 62/113 resources updated in 02 minutes 03 seconds
PS C:\Users\chef-repo> _
```

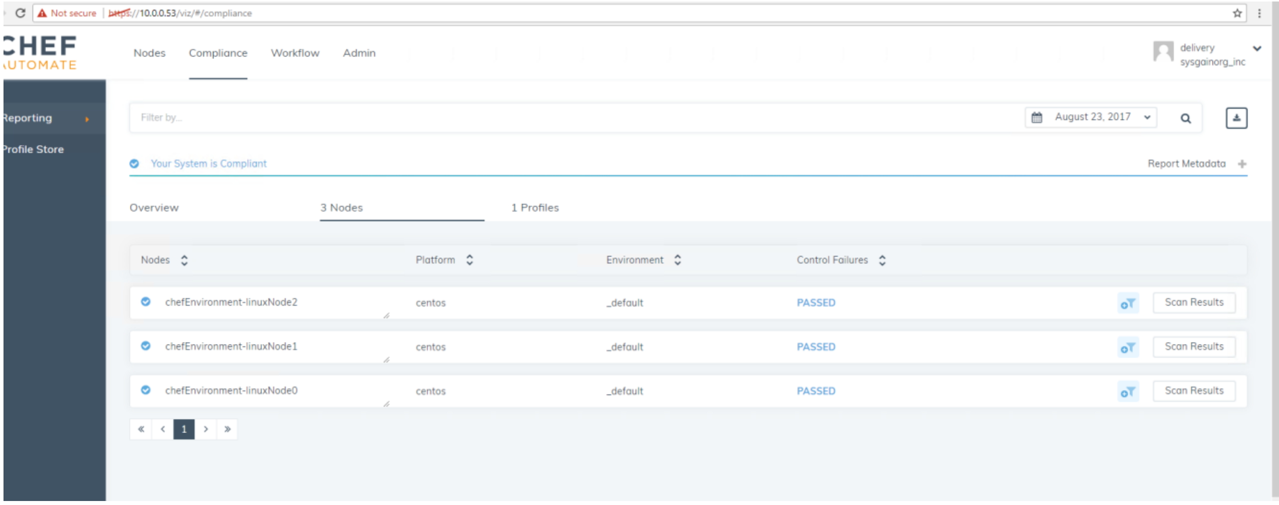
4.8. Review the improved state of the node’s compliance

Go back to your browser. Refresh the Automate page, hit the Compliance tab, then click on Profile Status. We should now be able to see our node with an improved compliance profile:



Not secure | ~~https~~://10.0.0.53/viz/#/compliance

Your System is Compliant



Your System is Compliant

Nodes	Platform	Environment	Control Failures	
chefEnvironment-linuxNode2	centos	_default	PASSED	Scan Results
chefEnvironment-linuxNode1	centos	_default	PASSED	Scan Results
chefEnvironment-linuxNode0	centos	_default	PASSED	Scan Results

5. Key takeaways

With Chef Automate, it's easy to maintain the integrity of your infrastructure and keep track of any security or compliance issues. The platform enables automatic remediation and continuous audit capabilities for any vulnerabilities that the system detects.