



# Jenkins

Remote Nodes

# Section Contents

- Remote Node Architecture
- Remote Node Strategy
- Setting up a Node
- Associating Jobs with Nodes
- Node Monitoring

# Remote Node Architecture

- Extra Nodes on Remote Machines:
  - can take on extra load when the "master" resources are full
  - can build on a machine that requires a different OS system that is different from the "master" OS
  - collectively known as farms
  - originally called "slaves" but is moving to "nodes"










# Remote Node

- A remote node
  - is a small java based executable that runs on a remote machine
  - communicates via TCP/IP
- Can be established using
  - SSH (Unix based machines)
  - JNLP (Other Operating Systems)

# Setting up a Managed Node (via SSH)

- Manage Jenkins Screen
- Click on the "New Node"
- Select "Dumb Slave"
- Provide a Name for the new node (No spaces preferred)

# Setting up Node Properties

Name	<input type="text" value="Some_Linux_Box"/>	
Description	<input type="text" value="Some Linux Box"/>	
# of executors	<input type="text" value="3"/>	
Remote FS root	<input type="text" value="/home/myname/.jenkins-node"/>	
Labels	<input type="text" value="linux mint fast"/>	
Usage	<input type="text" value="Utilize this slave as much as possible"/>	
Launch method	<input type="text" value="Launch slave agents on Unix machines via SSH"/>	
<div>Host <input type="text"/></div> <div>Credentials <input type="text" value="danno"/></div> <div> Add</div>		
<div>Advanced...</div>		
Availability	<input type="text" value="Keep this slave on-line as much as possible"/>	

## Node Properties

- ☐ Environment variables
- ☐ Tool Locations

Save

# Setting up Node Properties

- Name - Override the name of the node if desired (no spaces)
- Description – Description of the node
- Executors – Number of executors
- Remote FS root – Remote Folder where the jenkins will store workspaces
- Labels – Labels are used to control how where the jobs should run

# Node Usages

- SSH – Using SSH to setup the remote node
- Java Web Start – Allow a windows machine to start up a node
- Launch Slave Via Execution – Launch a node using the command line
- Let Jenkins Control Via a Windows Service – Control the node via Windows service



# Host and Credentials

- Enter the host of the machine you wish to connect to
- Enter in SSH credentials
  - As a name or password
  - A user with a private key
  - Or a certificate to the server

# Host and Credentials (Advanced)

- Port – The SSH port of the remote system
- JavaPath – The path that should be used on the remote system (it will search the remote system, if not added)
- JVM Options – Options used to start the JVM, like memory options (Xms, Xmx)
- Prefix and Suffix Command – Ability to wrap the the start "slave" command with some other commands if needed

# Availability

- Keep this slave on as much as possible -
  - Tries to keep the slave on-line.
  - Can start the slave without user assistance,
  - It will periodically attempt to restart the slave if it is unavailable.
  - Will not take the slave off-line.
- Take this slave online according to schedule
  - Uses cron time to indicate start time
  - Scheduled uptime in minutes to indicate amount of time
- Take this slave online when in demand and offline when idle
  - In demand delay – Length in minutes of how long it should be in queue
  - Idle delay – Length in minutes of how long it should be online doing nothing before shutting down

# Node Properties

**Node Properties**

---

☒ Environment variables

☐ Tool Locations

# Node Properties

- Environment Variables – The environment variables that you wish to create before running remote jobs
- Tool Locations – Locations of the tools that you wish to override from "Jenkins" section.

# Lab: Connect to the Instructor's Linux Machine!

- Set up remote node called "Linux Node"
- Select "Launch Slave Agents on Unix machines via SSH"
- Set up credentials based on the credentials that the instructor provides
- Set up a host name based on what IP the instructor provides
- Set up a JDK to run on the remote machine given what the instructor provides
- Set up Labels: Linux Instructor

# Setting up Windows Nodes









- Typical solution for Windows machines or machines behind a proxy
- Can really be used on any Operating System
- Drawbacks
  - Server cannot be started
  - Server cannot be restarted

# Create Managed Node

- Same setup as before with some exceptions
- Remote File system should reflect Windows (e.g. backslashes, C:, D:, etc)



# Typical Setup of Windows Node


Name	<input type="text" value="Windows-Node"/>	
Description	<input type="text" value="Windows 8.1. VM Node"/>	
# of executors	<input type="text" value="3"/>	
Remote FS root	<input type="text" value="C:\jenkins-node"/>	
Labels	<input type="text" value="Windows 8.1"/>	
Usage	<input type="text" value="Utilize this slave as much as possible"/>	
Launch method	<input type="text" value="Launch slave agents via Java Web Start"/>	
<div>Advanced...</div>		
Availability	<input type="text" value="Keep this slave on-line as much as possible"/>	

# Establishing the Connection

- Opening a browser on the remote machine,
- Opening the slave node page on the Jenkins master (may require login)
- Open the "Slave Node" that was just created
- Launching the slave using JNLP icon.
- Once you have launched the slave, you can install it as a Windows service
- Warning: Be sure to have the proper URL in "Manage Jenkins > Configure > Jenkins Location"
- Works best with Chrome or Firefox

# Connecting from the Node

Connect slave to Jenkins one of these ways:

-  Launch Launch agent from browser on slave
- Run from slave command line:

```
java -jar slave.jar -jnlpUrl http://192.168.56.1:8080/computer/Windows-Node/slave-agent.jnlp -secret  
55a214730a6caaff0f7df285797fb9c5887efa5f62b9dfd8c113fda5cd9f7273
```

# Lab: Connect to your Windows partner

- Select a partner, particularly one with the same setup
- Create a "Partner Node" on your machine.
- For the remote FS, select C:\jenkins-remote
- Provide Labels to describe your partner's machine
- Remember to delete C:\jenkins-remote on your machine after class.

# Establishing a Job on Remote Nodes

- Remember to "labels" that are used to create the nodes?
- You can assign those labels to any job by selecting "Restrict where this job is run" and entering the label in "Label Expression"
- By default, Jenkins will use the first available node
- If a label has a space, the entire name must be included in quotes

# Boolean expression on remote nodes

- windows
- linux
- !windows
- linux && ubuntu
- ubuntu && (inhouse || remote)
- (windows8 || windows7)

# Adding Labels to the Job



Restrict where this project can be run



Label Expression



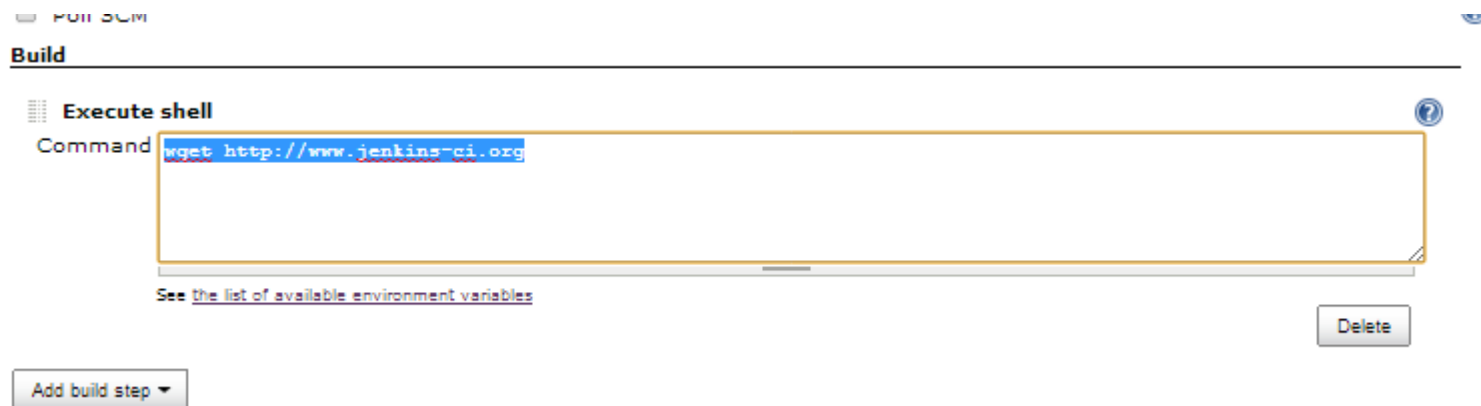
# Lab: Running on Remote Windows Machines

- Go to either your "simple-msbuild-project" or your "simple-ant-project" or "simple-maven-project" job
- Select your partners machine by selecting "Select where this project can be run"
- Add Labels that will trigger this job on the remote machine
- Manually Build the Job on the remote machine



# Lab: Running on a Linux Machine

- Create a free style job called "linux-build"
- Make sure that the job only runs on a Linux machine by restricting the build
- Add "Execute Shell" as a build step.
- In the shell, execute `wget http://www.jenkins-ci.org`



# Remote Node Last Thoughts

- All nodes are monitored
- Load statistics charts are available for each node
- System information is also available to view environments variables and system properties

Thanks!