Install Kubernetes on AWS

Log into all servers

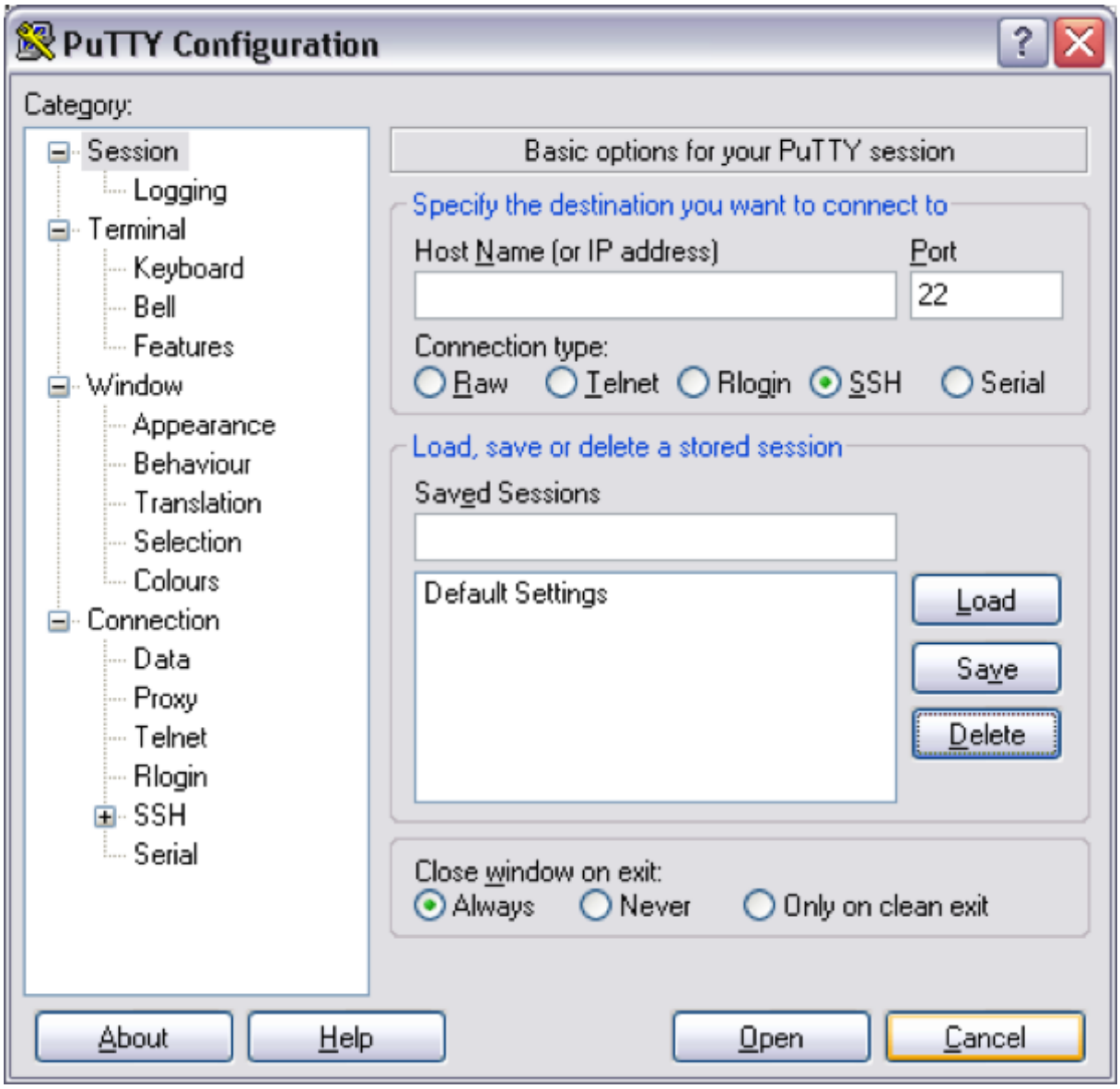
MacOS

Run the following commands in a terminal

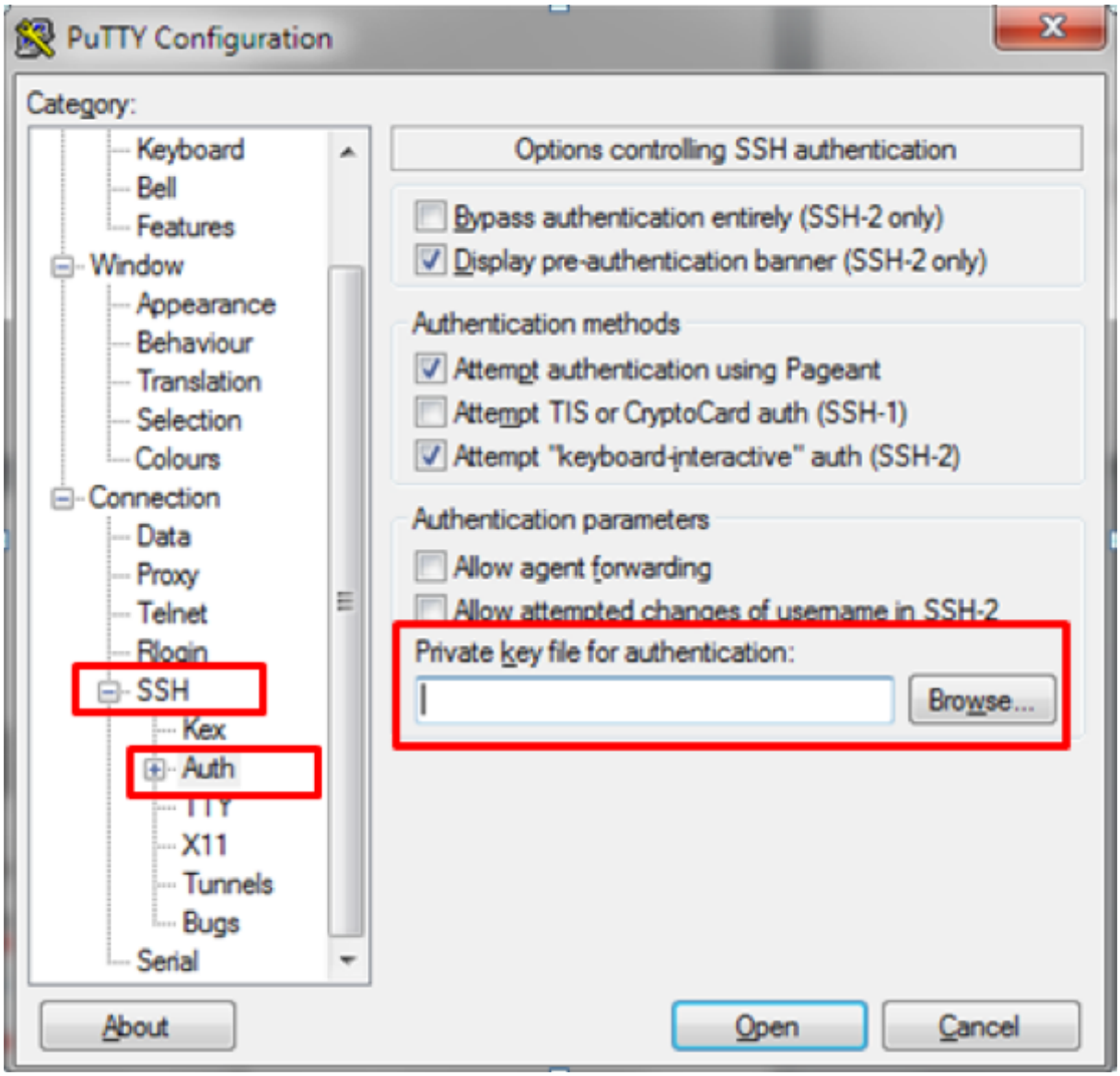
```
chmod 600 /path/to/lab.pem
ssh -i /path/to/lab.pem ubuntu@<server IP>
```

Windows

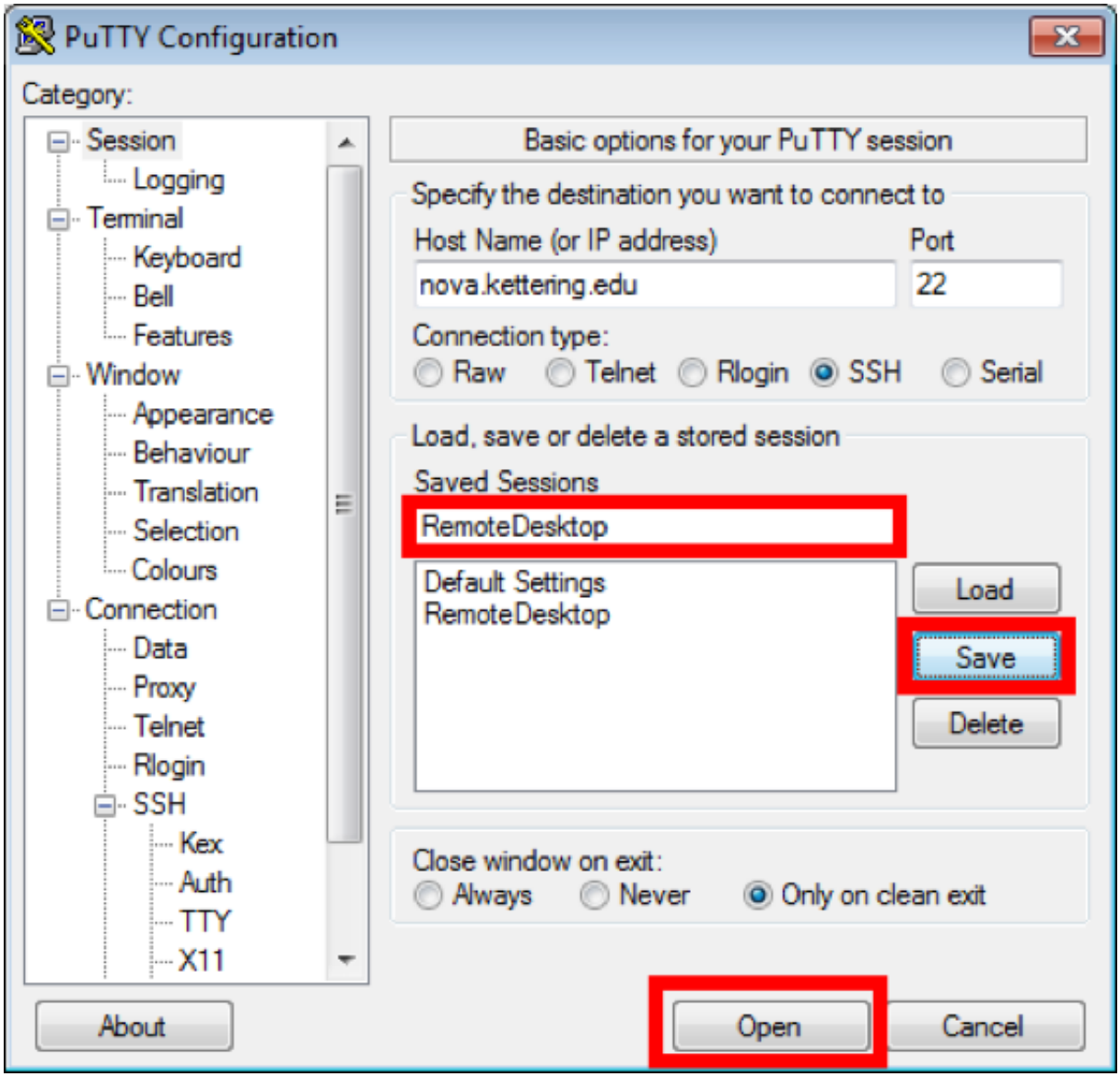
Open Putty and configure a new session.



Expand “Connection/SSH/Auth and then specify the PPK file



Now save your session



Install Kubernetes on all servers

Following commands must be run as the root user. To become root run:

```
sudo su -
```

Install packages required for Kubernetes on all servers as the root user

```
apt-get update && apt-get install -y apt-transport-https
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
```

Create Kubernetes repository by running the following as one command.

```
echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a /
etc/apt/sources.list.d/kubernetes.list
```

Now that you've added the repository install the packages

```
apt-get update
apt-get install -y kubelet=1.15.0-00 kubeadm=1.15.0-00 kubectl
```

The kubelet is now restarting every few seconds, as it waits in a `crashloop` for `kubeadm` to tell it what to do.

Initialize the Master

Run the following command on the master node to initialize

```
kubeadm init --kubernetes-version=1.15.0 --ignore-preflight-errors=all
```

If everything was successful output will contain

```
Your Kubernetes master has initialized successfully!
```

Note the `kubeadm join...` command, it will be needed later on.

Exit to ubuntu user

```
exit
```

Now configure server so you can interact with Kubernetes as the unprivileged user.

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Run following on the master to enable IP forwarding to IPTables.

```
sudo sysctl net.bridge.bridge-nf-call-iptables=1
```

Pod overlay network

Install a Pod network on the master node

```
export kubever=$(kubectl version | base64 | tr -d '\n')
curl -SL "https://cloud.weave.works/k8s/net?k8s-version=$kubever" \
| kubectl apply -f -
```

Wait until `coredns` pod is in a `running` state

```
kubectl get pods -n kube-system
```

Join nodes to cluster

Log into each of the worker nodes and run the join command from `kubeadm init` master output.

```
sudo kubeadm join <command from kubeadm init output> --ignore-preflight-
errors=all
```

To confirm nodes have joined successfully log back into master and run

```
watch kubectl get nodes
```

When they are in a `Ready` state the cluster is online and nodes have been joined.

Congrats!