Klucze GitLab i SSH

Git to rozproszony system kontroli wersji, co oznacza, że ​​pracując lokalnie możesz także udostępniać lub „wypychać” swoje zmiany na serwery zewnętrzne. GitLab obsługuje bezpieczną komunikację między Git a jego serwerami przy użyciu kluczy SSH. Protokół SSH zapewnia to zabezpieczenie i umożliwia uwierzytelnianie w Zdalny serwer GitLab bez podawania nazwy użytkownika i hasła za każdym razem. Ta strona może pomóc w skonfigurowaniu bezpiecznych kluczy SSH, których możesz użyć do zabezpieczenia połączenia z repozytoriami GitLab. Jeśli potrzebujesz informacji na temat tworzenia kluczy SSH, zacznij od naszych opcji kluczy SSH. Jeśli masz klucze SSH dedykowane dla swojego konta GitLab, możesz być zainteresowany pracą z niestandardowymi ścieżkami par kluczy SSH. Jeśli masz już parę kluczy SSH, możesz przejść do tego, jak dodać klucz SSH do konta GitLab. Wymagania Aby obsługiwać SSH, GitLab wymaga instalacji klienta OpenSSH, który jest preinstalowany w systemach GNU / Linux i macOS, a także w systemie Windows 10. Upewnij się, że twój system zawiera SSH w wersji 6.5 lub nowszej, ponieważ wyklucza to teraz niezabezpieczony schemat podpisu MD5. Następujące polecenie zwraca wersję SSH zainstalowane w twoim systemie: ssh -V Chociaż GitLab nie obsługuje instalacji w systemie Microsoft Windows, możesz skonfigurować klucze SSH, aby skonfigurować system Windows jako klienta. Opcje kluczy SSH GitLab obsługuje klucze RSA, DSA, ECDSA i ED25519. GitLab wycofał klucze DSA w GitLab 11.0. Jak zauważono w Practical Cryptography With Go, kwestie bezpieczeństwa związane z DSA dotyczą również ECDSA. UWAGA: Dostępna dokumentacja sugeruje, że ED25519 jest bezpieczniejszy. Jeśli używasz klucza RSA, amerykański Narodowy Instytut Nauki i Technologii w publikacji 800-57 część 3 (PDF) zaleca klucz o rozmiarze co najmniej 2048 bitów. Dlatego nasza dokumentacja koncentruje się na używaniu kluczy ED25519 i RSA. Administratorzy mogą ograniczyć dozwolone klucze i ich minimalne długości. Przejrzyj istniejące klucze SSH Jeśli masz istniejące klucze SSH, możesz ich użyć do zabezpieczenia połączeń z GitLab repozytoria. Domyślnie klucze SSH w systemach Linux i macOS są przechowywane w katalogu domowym użytkownika, w podkatalogu .ssh /. Poniższa tabela zawiera domyślne nazwy plików dla każdego algorytmu klucza SSH: Algorytm Klucz publiczny Prywatny klucz ED25519 (preferowane) id\_ed25519.pub id\_ed25519 RSA (co najmniej 2048-bitowy rozmiar klucza) id\_rsa.pub id\_rsa DSA (wycofane) id\_dsa.pub id\_dsa ECDSA id\_ecdsa.pub id\_ecdsa Aby uzyskać zalecenia, zobacz opcje dotyczące kluczy SSH. Generowanie nowej pary kluczy SSH Jeśli chcesz stworzyć: Klucz ED25519, przeczytaj klucze ED25519 SSH. Klucz RSA, odczytaj klucze RSA SSH. ED25519 Klucze SSH Książka Practical Cryptography With Go sugeruje, że klucze ED25519 są bezpieczniejsze i wydajniejsze niż klucze RSA. Ponieważ OpenSSH 6.5 wprowadził klucze SSH ED25519 w 2014 roku, powinny one być dostępne dla każdego prądu system operacyjny. Możesz utworzyć i skonfigurować klucz ED25519 za pomocą następującego polecenia: ssh-keygen -t ed25519 -C "<comment>" Flaga -C z cytowanym komentarzem, takim jak adres e-mail, jest opcjonalnym sposobem oznaczania kluczy SSH. Zobaczysz odpowiedź podobną do: Generowanie pary kluczy publiczny / prywatny ed25519. Wpisz plik, w którym chcesz zapisać klucz (/home/user/.ssh/id\_ed25519): Aby uzyskać wskazówki, przejdź do typowych kroków. Klucze RSA SSH Jeśli używasz kluczy RSA do SSH, zaleca to amerykański Narodowy Instytut Standardów i Technologii że używasz klucza o długości co najmniej 2048 bitów. Domyślnie polecenie ssh-keygen tworzy 1024-bitowy klucz RSA. Możesz utworzyć i skonfigurować klucz RSA za pomocą następującego polecenia, zastępując w razie potrzeby minimalnym zalecanym rozmiarem klucza 2048: ssh-keygen -t rsa -b 2048 -C "email@example.com" Flaga -C z cytowanym komentarzem, takim jak adres e-mail, jest opcjonalnym sposobem oznaczania kluczy SSH. Zobaczysz odpowiedź podobną do: Generowanie pary kluczy publiczny / prywatny rsa. Wpisz plik, w którym chcesz zapisać klucz (/home/user/.ssh/id\_rsa): Aby uzyskać wskazówki, przejdź do typowych kroków. UWAGA: Jeśli masz OpenSSH w wersji 7.8 lub niższej, rozważ związane z tym problemy z kodowaniem. Typowe kroki generowania pary kluczy SSH Niezależnie od tego, czy tworzysz klucz ED25519, czy RSA, zacząłeś od polecenia ssh-keygen. W tym momencie zobaczysz następujący komunikat w wierszu poleceń (dla kluczy ED25519): Generowanie pary kluczy publiczny / prywatny ed25519. Wpisz plik, w którym chcesz zapisać klucz (/home/user/.ssh/id\_ed25519): Jeśli nie masz jeszcze pary kluczy SSH i nie generujesz klucza wdrażania, zaakceptuj sugerowany plik i katalog. Twój klient SSH używa wynikowa para kluczy SSH bez dodatkowej konfiguracji. Alternatywnie możesz zapisać nową parę kluczy SSH w innej lokalizacji. Możesz przypisać wybrany katalog i nazwę pliku. Możesz także przydzielić tę parę kluczy SSH do konkretnego hosta. Po przypisaniu pliku do zapisania klucza SSH możesz skonfigurować hasło do Twojego klucza SSH: Wpisz hasło (puste, jeśli nie ma hasła): Wprowadź ponownie to samo hasło: Jeśli się powiedzie, zobaczysz potwierdzenie